TECHNISCHE UNIVERSITÄT WIEN

Development of an Inkjet Printer for Electrolyte Layers of Oxygen Ion Batteries

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Embedded Systems

eingereicht von

Raphael Hauk, BSc
Matrikelnummer 01326340

an der Fakultät für Elektrotechnik und Informationstechnik und ausgeführt am Institut für Chemische Technologien und Analytik
der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Phys. Dr.rer.nat. Jürgen Fleig
Mitwirkung: Senior Scientist Dipl.-Ing. Dr.techn. Tobias Huber
Projektass.(FWF) Dipl.-Ing. Dr.techn. Alexander Viernstein

Wien, November 2024

Raphael Hauk	Jürgen Fleig

Acknowledgements

First, I want to thank Prof. Dr. Jürgen Fleig for giving me the opportunity to pursue this thesis under his academic guidance and supervision. Furthermore, I want to express my gratitude towards the entire working group for their friendly welcome, their great interest and the natural manner in which suggestions and assistance are offered. Special thanks to Dr. Tobias Huber for his inspiration and motivation to kick off the project and to get me started, for his enthusiasm, and for providing pragmatic solutions to difficult problems along the way. Dr. Alexander Viernstein did not only share his expertise in electrochemistry with me, but also helped with ink preparation, patiently explained the measurement methods, and guided this thesis towards the finish line. Thank you very much for that! A special shout-out goes to Daniel Marik for the fruitful brainstorming sessions, discussions of all the small and big design decisions, and troubleshooting during and in the extension of our coffee breaks. I deeply thank Sonja Achleitner for sharing her profound experience in academic writing and the compassionate mental support throughout my studies.

Finally, I would like to thank everyone who supported my work and made the completion of this multi-faceted thesis possible.

Kurzfassung

Die Sauerstoffionenbatterie ist ein neuartiger Energiespeicher, der dabei helfen könnte, die variable Verfügbarkeit von Energie aus erneuerbaren Energiequellen, die zunehmend in die Stromnetze integriert werden, auszugleichen. Verschiedene Verfahren zur industriellen Herstellung dieser Batterien werden diskutiert, wie zum Beispiel der Tintenstrahldruck. In dieser Arbeit wird ein Tintenstrahldrucker zum Drucken von aus YSZ (Yttrium-stabilisiertes Zirkondioxid) bestehenden Elektrolytschichten entwickelt und angewendet.

Die Tätigkeiten umfassten die Erarbeitung der Anforderungen, die Auswahl der Komponenten, das elektrische und mechanische Design sowie die Softwareentwicklung des Drucksystems. Eine detaillierte Beschreibung der Designentscheidungen, der verwendeten Komponenten und der Implementierung soll dabei helfen, diesen Drucker, einen ähnlichen oder Teile davon zu reproduzieren. Die Hauptelemente des gebauten Druckers sind der Druckkopf, die Druckkopf-Steuerungselektronik und deren Firmware, das Tintenversorgungssystem, die Bewegungsplatform, Kamerasysteme zur Überprüfung und Optimierung des Druckprozesses sowie die Software zur Steuerung und Bedienung des Druckers.

YSZ-Schichten wurden mit dem implementierten Drucker auf Einkristallsubstrate aufgebracht, wobei selbst hergestellte, Lösungsmittelbasierte Tinten mit YSZ-Partikeln zum Einsatz kamen. Die gedruckten Schichten wurden gesintert und anschließend mittels optischer sowie Elektronenmikroskopie, Profilometermessungen, XRD (X-Ray Diffraction) sowie EIS (elektrochemische Impedanzspektroskopie) charakterisiert.

Es wurden erfolgreich verschiedene Modelle mit der Keramikpartikel beinhaltenden Tinte gedruckt, jedoch wiesen die aus bis zu 200 Schichten übereinander gedruckten Schichten bestehenden 4 µm dünnen Filme eine substantielle Porosität auf. Elektrochemische Messungen zeigten, dass die Dichte und die ionische Leitfähigkeit der hergestellten Schichten zu derzeitigem Entwicklungsstand noch nicht für den Einsatz als Elektrolyt in Sauerstoffionenbatterien ausreichen. Allerdings konnte aus der gemessenen Aktivierungsenergie der Leitfähigkeit geschlossen werden, dass es sich beim Leitungsprozess um den gewünschten Ionentransport in YSZ handelt. Durch Verbesserung der Tintenzusammensetzung wird erwartet die erforderlichen Schichteigenschaften für den Einsatz als Elektrolyt zu erreichen.

Im Zuge der Entwicklung und Anwendung des Druckers wurden Faktoren identifiziert, die eine wichtige Rolle für die Weiterentwicklung oder bei Auswahl eines zukünftigen Drucksystems für diese spezielle Anwendung spielen können. Dazu gehören Eigenschaften des Druckkopfs, zum Beispiel sollte die Kompatibilität zu wasserbasierten Tinten oder die Möglichkeit eine zirkulierende Tintenversorgung einzusetzen evaluiert werden. Die Installation in einem Gehäuse um Verunreinigungen vorzubeugen wird empfohlen. Weiter Erkentnisse betreffen das Zusammenspiel von Tropfengröße und Anforderungen an die Wiederholbarkeit der Bewegungsplatform, sowie die besonderen Vorteile der beiden Kamerasysteme, die für das erfolgreiche Drucken der keramischen Dünnfilme entscheidend waren.

Abstract

Oxygen-ion batteries are a newly developed all-solid-state energy storage device that could help balance the load of variable-power, renewable energy sources that are increasingly integrated into power grids. Different processes are discussed to fabricate these batteries on large scale, including inkjet printing. This work investigates the development and application of an inkjet printer for the task of printing the electrolyte layer, consisting of yttria stabilized zirconia (YSZ).

Development and implementation involved defining requirements, component selection as well as electrical, mechanical and software design. A detailed description of the design decisions, the components used and the implementation is given with the intention to assist in reproducing this printer, a similar one or parts thereof. The main elements of the built printer are the printhead, printhead drive electronics and associated firmware, an ink supply system, the motion stage, vision systems to inspect and optimize the printing process, as well as software to control and operate the printer.

Solvent-based inks containing YSZ particles were created and used to print YSZ layers with the implemented printer onto single crystal substrates. The printed layers were sintered and then characterized using optical and electron microscopy, profilometer measurements, X-ray diffraction (XRD) as well as electrochemical impedance spectroscopy (EIS).

While printing various patterns with the ceramic particle-filled ink was successful, the resulting layers with a thickness of up to 4 µm had substantial porosity, even though up to 200 layers were deposited on top of each other. Measurements indicated that density and ionic conductivity was not yet sufficient to use the printed layers as electrolyte layers for batteries. However, the detected activation energy was that of ion conduction in YSZ. However, based on the detected activation energy, it could be concluded that the conduction process is the desired oxide ion motion in YSZ. Moreover, it is expected to achieve the required layer properties for use as an electrolyte by further improving the ink composition.

Discussing the development and application of the printer, guidelines were identified that can be used for improvement or selecting/designing a future printer for this specific task. These include printhead properties like water-based ink compatibility or recirculating ink supply, repeatability of the motion stage related to droplet size, benefits of the camera systems and the use of an enclosure.

Contents

Kurzfassung v

Abstract vii

1 Introduction 1

2 Implementation 7
2.1 Printhead . . . 9
2.1.1 Requirements . . . 9
2.1.2 Component selection . . . 9
2.1.3 Implementation . . . 11
2.2 Printhead drive electronics . . . 12
2.2.1 Printhead power supply, waveform control and reset signal . . . 14
2.2.2 Level shift . . . 23
2.2.3 Strobe . . . 23
2.2.4 Firmware . . . 23
2.3 Ink supply system and printhead maintenance . . . 25
2.3.1 Requirements . . . 25
2.3.2 Implementation . . . 26
2.4 Motion stage . . . 30
2.4.1 Requirements . . . 30
2.4.2 Implementation . . . 31
2.5 Print system software . . . 34
2.5.1 Print job creation . . . 34
2.5.2 Print planner and track rasterizer . . . 36
2.5.3 Print job execution . . . 40
2.5.4 Software stack and architecture . . . 42
2.6 Dropwatcher . . . 45

3 Application 47
3.1 Ink . . . 48
3.1.1 Influence of the waveform . . . 50
3.2 Printed samples and printing process optimization . . . 51
3.3 Measurements . . . 55
3.3.1 Optical microscope . . . 55
3.3.2 Scanning Electron Microscopy . . . 58
3.3.3 Profilometer . . . 60
3.3.4 X-ray diffraction . . . 62
3.3.5 Electrochemical Impedance Spectroscopy . . . 63

4 Discussion and conclusion 69
4.1 Discussion . . . 70

4.2 Summary and outlook 74

References **75**

Acronyms **79**

List of Figures **80**

List of Tables **82**

A Xaar 128 Driver Shield Schematics **83**
A.1 Overview 83
A.2 Regulator 84
A.3 Printhead piezo drive voltage supply 84
A.4 Printhead reset signal 85
A.5 Microcontroller 85
A.6 Level Shift 86
A.7 FFC connector 86
A.8 Strobe 87
A.9 IO 87

B Equations **89**
B.1 Ohnesorge number 89

Chapter 1

Introduction

The European Unions Green Deal aims for net-zero greenhouse gas emissions by 2050, and photovoltaics plays a major role in this energy transition [1]. Energy storage can compensate for fluctuations in the supply of renewable energy. Grids that deploy high levels of photovoltaics are expected to require even more electrical energy storage power and capacity [2].

The recently developed oxygen ion battery [3] is a novel all-solid-state device for multi-hour energy storage that could help to overcome the challenges posed by renewable energy sources. It exhibits advantages compared to lithium ion batteries, a major established chemical energy storage [4]. For example, contrary to them, oxygen ion batteries do not contain flammable elements and therefore pose less of a fire hazard [5]. Additionally, in the long term, only materials with low environmental and social impact shall be used. Consequently, no materials that are mined under questionable conditions like cobalt are utilized in oxygen ion batteries [6]. Also, the possibility of regeneration after degradation and resulting loss in capacity, promises longevity.

The simplified structure of such an oxygen ion battery is an anode and cathode stacked together, separated by a thin layer of a solid electrolyte which consists of ceramic yttria stabilized zirconia (YSZ). The electrodes must be isolated from the atmosphere, while current collectors provide the electrical interface to them. A high current density can be achieved by using a very thin but dense electrolyte layer at sufficiently high temperatures (>250 °C).

Different processes to fabricate such batteries on a larger scale are discussed in the field. For example, a reactive sputtering technique to create dense YSZ thin films was recently developed [7]. Additive manufacturing processes such as inkjet printing have been discussed as a promising tool for the production of all solid-state batteries [8]. As an additive manufacturing technology, it wastes less material (in comparison to subtractive processes). The dispensed droplet patterns are entirely digitally controllable allowing arbitrary structures, and the patterns can be changed easily. Because of this, products can be adapted quickly, and the process is also interesting for prototyping and use in research and development (R & D). It is scalable, by fitting many printheads in a row, and can be used in a continuous production process. By mixing solvents with particles, various inks for use in material jetting can be created, including e.g. metal or ceramic filled inks. [9] [10]

While this sparks the idea to investigate inkjet printing for manufacturing the battery structure as a whole, this work focuses on printing the electrolyte layer. More specifically, a printer that is tailored to the use case of printing the electrolyte layer for oxygen ion batteries is built and applied.

In [11], various successful approaches in inkjet printing ceramics are summarized, and the basic process outlines as follows: A drop on demand (DOD) printhead with one or more nozzles dispenses picoliter-sized droplets of ink filled with ceramic particles on a substrate. To create the desired patterns, the printhead and the substrate are moved relative to each other and the drop ejection is timed precisely. The liquid contents of the ink evaporate, leaving a layer of particles on the substrate. To create 3D structures, this process is repeated. After

the printing is finished, a sintering step can be applied, for example to receive a dense film in the case of printing an electrolyte layer.

Regarding ceramic printed YSZ layers, many applications concern solide oxide cells (SOCs), particularly solid oxide fuel or electrolytic cells. In the following, work related to this thesis, including inkjet printing ceramics for SOCs, will be presented, leading to the definition of the aim of this work, concluding this chapter.

Related work In [12], SOCs are constructed by a combined approach of robocasting and inkjet printing, where a YSZ layer is printed on/sandwiched between a thicker robocasted electrode material. The YSZ layer is printed using a Pixdro LP50 inkjet printer, equipped with a Dimatix DMC 2810 (Fujifilm) printhead. A water based ink with a ≈1.8 vol.% particle load was created that was further diluted 3:1:1 using an isopropyl alcohol and propylene glycol solvent mixture before printing. After co-sintering the layers, the resulting YSZ layer appears highly dense in scanning electron microscope (SEM) images and has an approximate thickness of ≈2.8 µm. Holes found in the printed YSZ layer, where the two electrodes seem to be in contact were attributed to imperfections in the printing process.

The same combination of printer and printhead was used in [13] to print YSZ films. The synthesis of a YSZ dispersion in water using a continuous hydrothermal reactor led to exceptionally small single crystal particles with a size of ≈10 nm with very little agglomeration. The dispersion basis featuring a 5.5 wt.% particle loading was further adjusted in viscosity and surface tension using additives. With an optimized waveform the droplet size was found to be 70 µm. Arranging the drops into a square grid of droplets with 500 dots per inch (DPI) spacing in both directions and printing 10 layers resulted in dense layers of 250 - 500 nm in thickness after sintering at 1000 °C and in full densification after sintering at 1200 °C.

Another type of R & D printing platform, the Dimatix DMP-2850 (Fujifilm) was used in [14]. Again, a water-based ink was prepared from commercially available powder by ball-milling, and the viscosity and surface tension was adjusted using additives. This time, the particles are a combination of YSZ and lanthanum strontium manganese oxide (LSM), having an average hydrodynamic diameter of 370 nm suspended in the ink. The loading of the final ink is reported to be 3.9 wt.%. The printing waveform[a] was optimized using a dropwatcher camera, and the resulting droplet size is 39 µm. The deposited film consists of 90 layers with drops overlapping in single layers at 891 DPI. With the substrate heated to 60 °C, the drying time for each layer was 8 min. The resulting thickness shown in cross section SEM images is approximately 9 µm.

Even more 3D-like YSZ structures, in form of pillars, were printed in [15] using a CeraPrinter X-Serie (Ceradrop) equipped with a Dimatix Sapphire QS-256/30 AAA printhead (Fujifilm). A water-based dispersion ink was produced from particles of 194 nm and 121 nm by sonication together with printability-adjusting additives. Next to the pillars that contained 50 or 180 layers of single drops or two drops overlapping, also planar electrolyte films were printed consisting of 12 layers of square lattice patterns with a drop overlap of 20 to 40 %. The layers dried between printing for 240 s at ambient temperature. The resulting dense YSZ layer had a thickness of 23 µm after sintering at 1000 °C for 3 h.

While the information on ink preparation and its characterization is very detailed, and in most cases there is some description of the printing process and its parameters, there are fewer hints on the printer itself, e.g. what the advantages and shortcomings of the different systems are, with respect to the specific application. Common patterns can be found in the different approaches mentioned above: For one, piezo-based actuation is clearly preferred, while different approaches exist. For example, thermal-based inkjet actuation[b] is used in

[a] Waveform: The electrical signal applied to the piezo elements inside the printhead that are responsible for drop ejection

[b] Thermal-based inkjet actuation: The ejection of a droplet is triggered by creating a vapor bubble in the ink using miniature heating resistors in the nozzle chamber.

[16], where a commercial desktop/office inkjet printer was modified. Printing in multiple layers and post-processing by sintering are techniques that lead to dense layers. However, printing parameters are ink- and system-dependent. The droplet size depends on the ink, the interaction of the ink and the substrate, and the jetting parameters as can be seen by differences in related work that were using the same printhead, presented earlier this section. On the other hand, it is expected that different printheads produce different droplet sizes with the same ink. Printing with the right amount of overlap of droplets in the layers is important to form a smooth deposition as indicated in [11]. This makes it difficult for printer distributors or printhead/drive electronics manufacturers to recommend a one-size-fits-all printhead.

Available inkjet systems The before mentioned Dimatix Materials Cartridges[a] are sometimes recommended as entry level printheads. They are designed as disposables, with just 12/16 [b] nozzles with a larger pitch (distance between adjacent nozzles) of $\approx 338.67/254\,\mu m$. Their main advertised purpose, according to the website, is for prototyping, iterating through ink formulations utilizing the low fill volume of 1.5 ml. The price per head is still in the lower hundred euro range.

Commonly used printers in R & D are the Pixdro LP50 (SUESS)[c] and the Dimatix DMP-2850 (Fujifilm)[d]. While the latter is designed to be used with the Dimatix Materials Cartridge, the Pixdro LP50 is compatible with various piezo-based printheads from different manufacturers. It is more expensive than the Dimatix DMP-2850, but has a higher repeatability motion stage, and features advanced drop analysis and print optimization routines, as well as different printhead maintenance procedures. Both have a heatable vacuum plate to hold the substrate, and two camera systems, one for the dropwatcher and one monitoring the substrate, for alignment and inspection purposes. The Pixdro LP50 also features a print simulator to preview the exact drop placement.

Requirements and entry cost Prices for ready-made entry-level R & D inkjet systems as described above start at around 100.000 euros.[e] Undoubtedly, there is a lot of features packed into them, justifying the price. However, if a department invests in a new system to research a specific application, it should be the right one for the job. The requirements for a specific task are not always clear at the start of a project.

The minimum resolution and repeatability of the motion stage to form dense layers does not only depend on the droplet size and nozzle pitch and consequently also ink and printhead selection. The patterns and their smallest features that should be printed give constraints on required precision and droplet size too.

The next question, after a decision for a system has been made, is, which printhead fits the application best. There is a wide variety available[17], and printhead drive electronics manufacturers offer great compatibility by head personality boards[f]. A group starting out with the inkjet process might want to start out with a less expensive printhead, and upgrade later on project success. But project success might suffer from a suboptimal head choice. Furthermore, the price of head personality boards can be a multiple of a single printhead,

[a] Dimatrix Materials Cartridge: (https://www.fujifilm.com/us/en/business/inkjet-solutions/inkjet-technology-integration/materials-cartridge)

[b] 12 or 16 nozzles, version dependent

[c] Pixdro LP50: https://www.suss.com/en/products-solutions/inkjet-printing/lp50

[d] Dimatix DMP-2850: https://www.fujifilm.com/de/en/business/inkjet-solutions/inkjet-technology-integration/dmp-2850

[e] The rough price ranges given in the following for guidance purposes are based on consultations of dealers and their offers.

[f] Head personality board: an adapter electronics that sits between the main printer control electronics and the actual head, converting the more general firing instructions into the specific control signals of the head.

increasing the weight on the decision further. Also, changing to a different printhead might require adaptions in the ink supply system.

Availability of secondary features of the printer and their properties can be of importance. Examples are the maximum temperature of the substrate heating or the possibility to fit a fan or a curing lamp to speed up the drying process.

The combination of high entry cost and limited knowledge of the requirements encourages further research before making a purchase decision.

Building a printer Another way to start research of an inkjet application is to build a printer. The biggest advantage is the great customizability of all the different aspects. One approach is to combine different readymade distinct subsystems into a tailored device. Most printhead drive electronics suppliers, like Global Inkjet Systems[a] or Meteor Inkjet[b] provide the printheads, drive electronics and control software in a single package. Companies like ImageXpert[c] specialize in customizable components from ink supplies and vision systems up to complete R & D print setups.

Added Scientific offers a Driver Board for the Xaar 128 printhead[d] which supports up to four printheads. It allows to control the waveform by adjusting the printhead clock input from 50 to 150 % of its nominal value of 1 MHz and the piezo supply voltage from 15 to 36 V. It also provides an encoder input and a heater for the printhead, which allows to jet higher viscosity inks, like UV inks. An example for its application is to use it on a low cost 3D fused deposition modeling (FDM) printer platform.

A different approach is to build the printer from basic parts (from scratch). The RepRap project [18] sparked a movement of open- and closed-source desktop 3D printers, which are now also commonly found in workshops and laboratories due to their suitability for rapid prototyping. Inkjet printing is also a topic that attracts increased interest in the 3D printing community, which can be observed by pages in the RepRap wiki describing the use of inkjet printheads on 3D printers, including the Xaar 128 printhead[e].

Another step towards an open source inkjet platform was contributed in [19]. Here, an open source DNA synthesizer was developed that constructs oligonucleotides (short DNA or RNA segments) by inkjet printing. It uses five Xaar 128 printheads with different inks loaded: one to dispense an activator and one for each of the bases of DNA. While the impact of this work on open source inkjet platforms remains to be seen, it is a good example to show the advantage of customizing a printer to the specific task: Syringe pumps to dispense bulk reagents in between printing, specialized drying of the substrate and performing the process in argon/nitrogen atmosphere are examples for that.

Building a printer, also has an advantage in the event of repair, since many parts can be reproduced in-house. Additionally, it is a great opportunity to learn and familiarize with the technology. The biggest downside is the additional time required to build such a printer that increases with every custom-built component.

Considering the aspects of the different approaches, to kick off researching the inkjet printing process for the solid oxygen ion battery production, the aim of this thesis is defined as follows:

Goal of this work A prototyping inkjet printer is built that is tailored to the task of printing YSZ electrolyte layers for oxygen ion batteries. During the design and application,

[a] Global Inkjet Systems: `https://www.globalinkjetsystems.com/`

[b] Meteor Inkjet: `https://www.meteorinkjet.com/`

[c] ImageXpert: `https://imagexpert.com/`

[d] Added Scientific Products: `https://www.addedscientific.com/products`, Jetforge Driver Board Datasheet: `https://static1.squarespace.com/static/6167f11bbc1d430fd6c867d6/t/637f713e994f004027f49cd6/1669296447951/Low+Cost+Datapath+Datasheet.pdf`

[e] Xaar 128 printhead in the RepRap wiki: `https://reprap.org/wiki/Getting_Started_with_Xaar_128`

requirements are identified, that provide a basis for future decisions when purchasing an inkjet printing system for this task. Examples include the characteristics of a printhead that are most important for this application, the constraints on the movement stage and what other features are required for successfully building up the layers and geometries. In this kind of bottom-up approach, components are added when requirements are detected, which should lead to a guideline of how a minimal system should look like.

This thesis is structured as follows: In the following chapter, the printers implementation is presented in detail, and its components are explained individually. Subsequently, the application of the printer is discussed, including information on the printing process and the ink used. The evolution of printing parameters is described and the printed samples are presented and characterized. An analysis of the findings is given in the final chapter, where most important properties of an inkjet system for this task are identified. An outlook on further investigations as well as possible improvements when reproducing this printer build are suggested.

Chapter 2

Implementation

The following chapter provides a detailed insight into the implemented printer, what components it consists of and their purpose, how they interact, and what challenges had to be solved during implementation. In Fig. 2.1 one can see an overall picture in which most of the

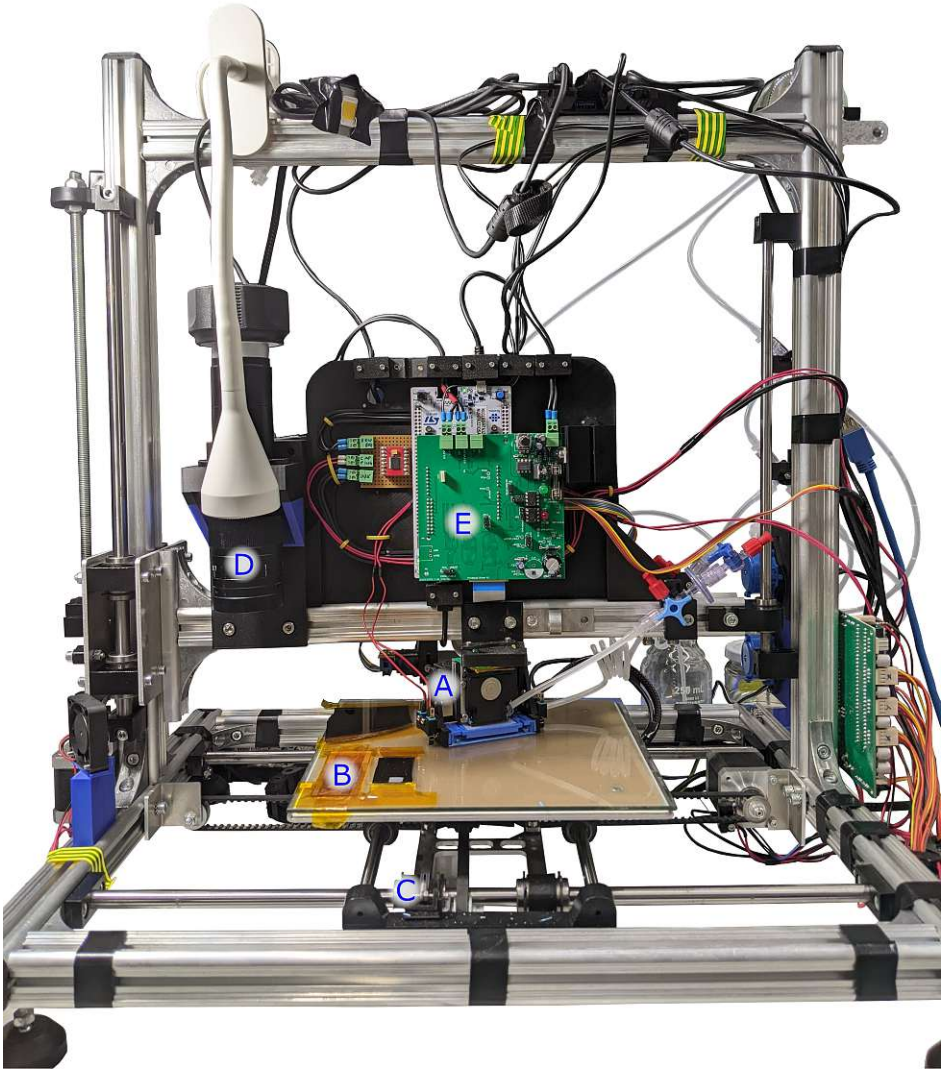


Figure 2.1: Photo of the implemented printer.

implemented components are shown.

An inkjet printhead (A in Fig. 2.1) ejects droplets of solvent ink filled with ceramic particles onto a substrate. The substrate is mounted to a build plate (B in Fig. 2.1). A

motion stage positions the build plate relative to the printhead to place the droplets. An encoder (C in Fig. 2.1) provides a signal synchronized to the motion, allowing the firing pulses of the printhead and thus the droplet positioning to be sufficiently precise. This way the printhead moves over the substrate area until a complete layer is printed. After a curing time the solvent evaporates and the process is repeated to print the next, and the subsequent layers.

The model to print is created by computer aided design (CAD). A slicer and rasterizer software extracts the separate layers from the model and converts them to a continuous stream of nozzle drop ejection commands that are executed at the specific encoder positions.

A print system software orchestrates the components to print the complete model while monitoring the process and providing information. It enables the user to supervise the process and interact with the system.

A successful printing process depends on several additional features, which were implemented in this work:

- an ink supply system that maintains the correct pressure of ink, filters the ink and prevents ink shortage or air entering the printhead,
- different printhead maintenance functions providing capping, flushing, pressure and vacuum purges closely connected to the ink supply system,
- a dropwatcher, which is a camera system capable of capturing the droplet ejection process to identify problems, characterize droplet formation and tune printing parameters,
- a build plate substrate camera (D in Fig. 2.1), for alignment purpose and inspection of the placed droplets and printed layers.

The next sections describe the components in detail, starting from the printhead at the core of the machine. Then, the components closest to the printhead are discussed: The printhead drive electronics (E in Fig. 2.1), the ink supply system and the motion stage. Finally, the print system software and the vision systems are explained. Each section first establishes the requirements that were set for the specific subsystem. Based on them, component selection and implementation are discussed, focusing on the challenges that had to be solved.

2.1 Printhead

2.1.1 Requirements

Most DOD printheads use either a thermal or piezo actuation principle. Thermal based printheads are mainly used in desktop graphics printing, and impose limits on compatible inks [20]. However, material jetting using thermal based HP office printers has been shown [21].

A great resource to get an overview of the industrial printhead market is The Ultimate Industrial Inkjet Printhead Comparison Chart[17] compiled by ImageXpert. The printheads are listed there together with some of their characteristics:

- number of nozzles and print width (swathe)
- fluid compatibility, categorized to oil-, solvent- or water-based fluids, viscosity range
- droplet size (or range), and maximum jetting frequency
- compatible ink viscosity range
- whether a recirculating ink supply system is required
- number of different colors (inks)

For the application of printing a thin electrolyte layer, one has to keep in mind that the ink which is printed contains particles, and there are trade-offs to consider:

Small nozzle size and corresponding drop size increases resolution [22], which comes together with higher restrictions on solid particle size in the ink [23], which should be significantly smaller than the nozzle diameter.

A sufficiently high nozzle density makes it possible to print in a single pass or multi pass scanning mode. This means that with every pass of the printhead over the substrate, a continuous area with a width of the printhead swathe is covered. When the nozzle pitch is larger, the printhead has to be moved in smaller increments, to cover the space in between the nozzles, which is called multi-cycle printing [24].

Water-based inks are more environmentally friendly, however, non-aqueous solvent-based inks evaporate fast and have a simpler drying process. [25]

A printhead with a recirculating ink supply has an inlet and an outlet port, and the ink is pumped through the printhead while printing by a separate pump. A recirculating ink supply can help to prevent particle sedimentation in the ink and subsequent clogging of the printhead, as well as to remove air bubbles from the system [26] [27]. On the other hand, a recirculating printhead ink supply requires additional tubing, potentially also more pumps, sensors, reservoirs or filters that increase the minimum volume of ink required to fill the printhead.

2.1.2 Component selection

In order to choose a printhead for the particular case described here, different printhead and drive electronics suppliers were contacted and the requirements discussed. However, it was difficult to propose and evaluate printheads due to a number of reasons:

- Many parameters of the process were not yet known, e.g. the nature and composition of the ink used. Also the project being an evaluation of the inkjet process for the application did not impose strict limits on accuracy or layer height.
- Many printheads suggested are not primarily designed for material jetting. Therefore, official specification sheets often come short to help assessing compatibility. They often are designed to be used with vendor-authorized ink only.

- Most of the information about internals of the printheads are confidential, like whether internal filters are installed and their size.

There were other factors influencing the decision as well: Examination of commercial desktop inkjet printers showed that clogged nozzles would be a problem, caused by e.g. incompatible or dried out ink. The selected printhead should therefore be able to withstand regular maintenance including cleaning, flushing and unclogging procedures. The price and availability should make it possible to replace when its unrecoverable.

Some inkjet drive electronics manufacturers split their products into a more generalized printhead drive electronics that supports multiple printheads, and head personality boards or heard interface cards, which then connect a specific printhead (family) to the drive electronics. This way it is possible to only replace a part of the electronics, if another printhead should be used. The GbitE Print Controller from Meteor can be used for example with different head interface cards for example with Fujifilm or RICOH printheads.[a] The AEWA Print Manager Board[b] connects with different printhead drivers to interface a printhead. But even if it is interchangeable, the head interface cards can sell for a multiple of the price of the printheads, accordings to offers from vendors received, making a premature suboptimal printhead decision costly.

It was decided to use a comparably inexpensive printhead in this work to explore the parameter space, resulting in a firm basis for the decision which printhead should be used for future research.

The Xaar 128 showed to be a fitting candidate with the following advantages:

- It features 128 rather large diameter (50 µm / 35 µm) nozzles than can be controlled with a simple interface and eject comparably large droplets of 40 pl or 80 pl (version dependent). Starting with bigger nozzles increases the chances of success for the first experiments, as it decreases the limits for an ink to be printable: The Ohnesorge number (see Section B.1) that is used as a guidance of ink printability should be < 1 for ink to eject [28], the nozzle diameter being in the denominator. Also, there are less restrictions on particle size [23].

- It generates the piezo drive waveform internally, so no special waveform drive electronics are required.

- It has been explored by the community surrounding the RepRap project[c], with the main focus in binder jetting and UV-based 3D printing. In continuation, there is a forum thread [29] with more than 300 posts, where users documented their experiences with this printhead and frequently occurring problems were discussed. Naturally, building on the experiences of others is a big advantage.

- It is durable, when operated within specification. At preliminary tests, Epson desktop inkjet printheads were quickly damaged when cleaning and unclogging procedures needed to be performed. The Xaar 128 is reported [29] to tolerate cleaning procedures including gentle ultrasonic cleaning[d]. The manual also lists other maintenance procedures, allowing pressures as high as 0.5 bar.

[a] Meteor printhead controllers: `https://www.meteorinkjet.com/products-services/printhead-controllers/`

[b] AEWA Print Manager Board & Printhead Drivers: `https://www.aewa.de/index.php/en/products.html`

[c] RepRap: self replicating 3D printer [18], topic about the Xaar 128 printhead in the RepRap wiki: `https://reprap.org/wiki/Getting_Started_with_Xaar_128`

[d] The Print Head Doctor (`https://www.printheaddoctor.com/`) is a solution specialized to recovering and cleaning several printhead types. Its manual lists various printhead cleaning sensitivities, and reports medium for the Xaar 128.

- Availability: It is developed and advertised for coding and marking, and is in stock at a supplier in Europe at a price range around 300 euro per head[a].

However, it also comes with the following disadvantages in the scope of the task:

- Since the waveform is internally generated, it cannot be changed. The supply voltage can be varied though, supposedly influencing the amplitude of the firing pulse. Also, the internal clock of the printhead is provided externally, and by variation of that, the firing pulse width of the waveform may be influenced. With these two parameters the waveform is customizable to some extent, which is shown in this work using the dropwatcher.
- The printhead employs a so-called shared wall design for its piezo channels. This means that one piezo wall is shared between two channels. Since the wall can only shear towards one direction at once, adjacent nozzles cannot fire simultaneously. Therefore, the nozzles are sorted into three groups that fire sequentially.
- There is an internal last chance ink filter installed that cannot be exchanged. Therefore, ink filtering prior to the printhead with a smaller filter is of great importance [30]. It was found that, if air bubbles exist in the tubing, they tend to accumulate in the installed filter in the ink supply system, and assumed that there could be a similar problem at the internal ink filter.
- According to the data sheet, the printhead is to be used with oil or solvent ink, and users reported damaged printheads after being used with water-based inks [29]. Presumably, this is due to piezo actuators being exposed to the fluid. While many piezo printheads are not compatible with water-based ink, however there is increasing market demand for aqueous ink supporting printheads leading to new products [31].

2.1.3 Implementation

The printhead is fixed to the printers z-axis carriage aluminum extrusion using a 3D-printed mount, shown in Fig. 2.2 on the left and labeled A installed on the printer on the right. It allows to adjust the printhead's angle (with respect to the printing axis of the stage).

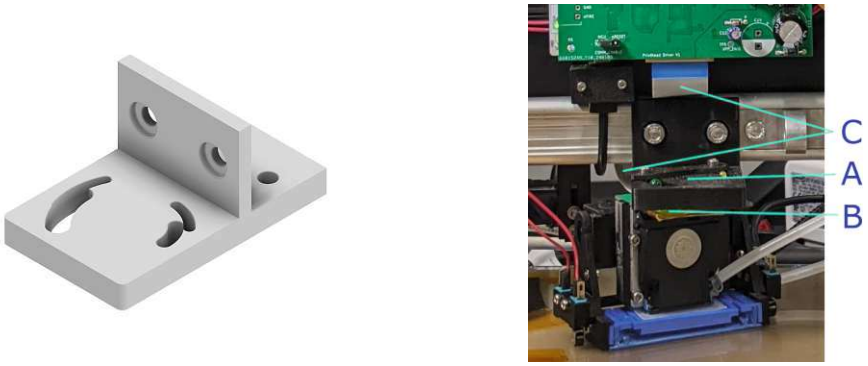


Figure 2.2: Printhead mounting and electrical connection.

A steeper angle increases the drop density across the printing axis from the default nozzle pitch of 137.1 µm in perpendicular orientation, which is further detailed on in Section 2.4.1.

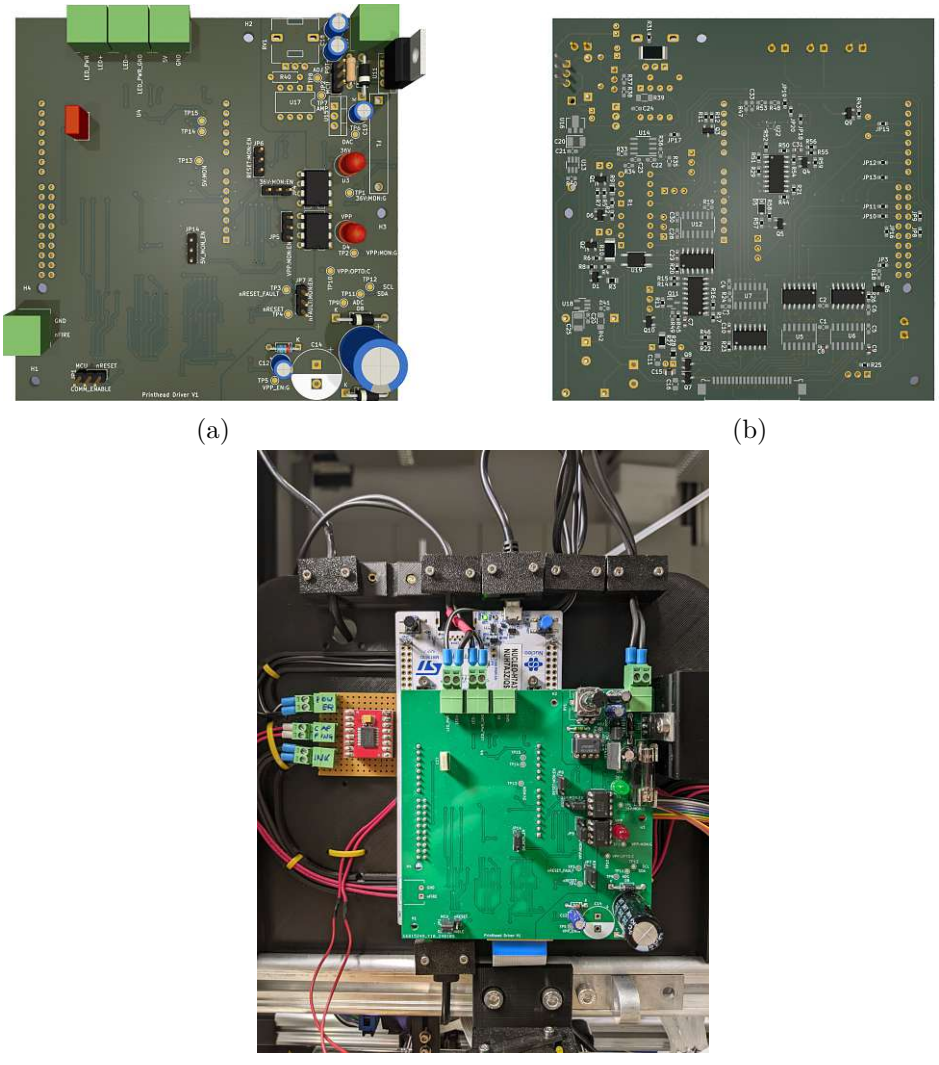
A small printed circuit board (PCB), labeled B, which interfaces the printhead's electrical connector to a 30 pin flexible flat cable (FFC), labeled C, are also shown Fig. 2.2. They connect the printhead to the printhead drive electronics, which will be presented in the next section.

[a]Xaar 128 at Digiprint Supplies: https://www.digiprint-supplies.com/de/xaar-128-80-printhead-xp12800005-pphxa128bt

2.2 Printhead drive electronics

The printhead drive electronics provide different functions to the higher level software in order to execute print, diagnostic and maintenance jobs.

It is based on a NUCLEO-H7A3ZI-Q development board from ST Microelectronics. Several reasons led to the decision to use this component: When starting out with the project, the complete hardware architecture was not yet clear, and the STM32H7 Series microcontrollers (MCUs) allows for a lot of options: It is supported as a target for Marlin Firmware[a] as well as Klipper[b], which are the two most common 3D FDM type printer firmwares, as well as Zephyr Project[c], which is ultimately used as the basis for the firmware in this project. The maximum possible 280 MHz clock speed and more than 1 MB RAM provide some headroom, for a price of less than 30 euro. The NUCLEO-144 development board form factor comes with a lot of pins, which makes it simpler to add additional functionality that was not expected in the first iteration. A subset of its pinsocket connectors are compatible to the layout of Arduino UNO.



(a) (b)

(c)

Figure 2.3: Xaar 128 driver shield renderings. Front side (a), back side (b) and the completed printhead drive electronics assembly including the NUCLEO development board (c).

[a] Marlin Firmware: `https://marlinfw.org/`
[b] Klipper: `https://www.klipper3d.org/`
[c] Zephyr Project: `https://zephyrproject.org/`

The Xaar 128 driver shield that is discussed in detail in the following, is an extension board for the NUCLEO-144 development board, and was designed in this work. The 2-sided shield PCB was created using KiCad[a], and ordered together with a surface mount technology (SMT) stencil from JLCPCB[b]. It was then assembled by SMT soldering using the stencil and hand-soldering the through-hole components. Renderings of the shield and the assembled printhead drive electronics are shown in Fig. 2.3. The design is kept compatible in a way that the most important functions could also be executed by using it with an Arduino UNO.

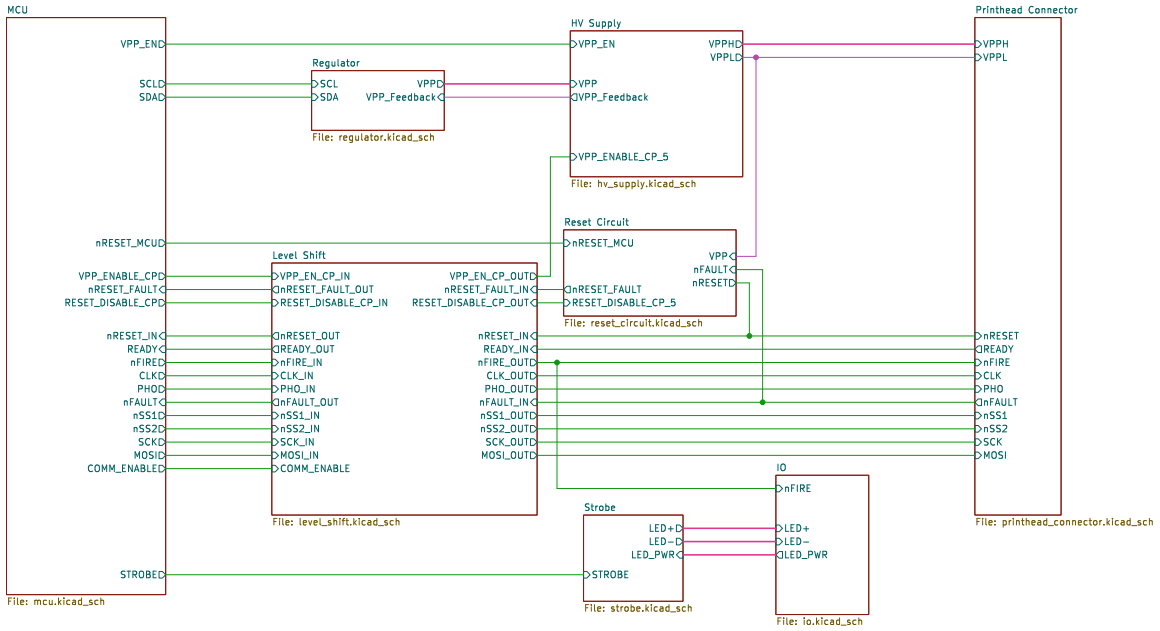


Figure 2.4: Xaar 128 Driver Shield overview.

An overview of the shields components and connections is given in Fig. 2.4, which will be further referred to when explaining the separate components. Next to its main job, sending nozzle data to the printhead, and precisely triggering the jetting, it fulfills a variety of important tasks:

- printhead power supply and interface, including adjusting the waveform by varying supply voltage and printhead clock signal,
- decoding of the encoder signal,
- speed and direction control of the ink supply pumps and readout of the pressure sensor to provide an ink supply interface to the print system software,
- triggering the strobe of the dropwatcher,
- monitoring and failure handling: The components parameters, like power supply voltages, printhead temperature or ink pressure are monitored so that a safe state can be entered when errors occur.

Some of these functions (pump control, pressure sensor, encoder) were not considered in the first developed PCB design iteration and were fitted to the NUCLEO development board afterwards, by building smaller perfboards.

In the following sections, first the goals and corresponding requirements of the particular jobs are explained in more detail. Then the implementation of the relevant hardware and the firmware is described, focusing on the challenges that had to be solved.

[a]Kicad Electronics Design Automation Suite: `https://www.kicad.org/`
[b]JLCPCB PCB prototype manufacturer `https://jlcpcb.com/`

2.2.1 Printhead power supply, waveform control and reset signal

The data sheet of the printhead specifies three power supplies: a 5 V logic level power supply and two piezo driving supplies, one low-current-low-noise and one with more current capability and less requirements on noise. The nominal voltage of the piezo driving supplies is 35 V. An additional requirement is that the grounds of the logic power supply, and the two piezo driving voltage supplies need to be separated.

The printhead data sheet also states that a power supply sequencing scheme must be strictly followed to not damage the printhead:

1. The logic power supply is turned on.
2. The piezo driving supplies are turned on.
3. The reset line of the printhead is released.

For turning off the printhead, the sequence must be followed in reverse order.

The Xaar 128 generates its waveform internally. However, it is reported that user waveform control should be possible to some extent: The Jetforge driver board introduced in Section 1 allows to change the printhead's clock input from 50 to 150 %, to influence the waveform timing, and to vary the piezo driving voltage from 15 to 36 V. It was also suggested by other users to use a step down converter in series with the power supply to adjust the voltage [29]. In this work, variable piezo drive voltage and clock signal are implemented and the influence on the jetting is reported in Section 3.1.

Fig. 2.5 shows an overview of the circuits developed to meet the above requirements, an adjustable isolated piezo drive voltage and strict adherence to the power sequence. These circuits and their interaction are discussed in more detail below.

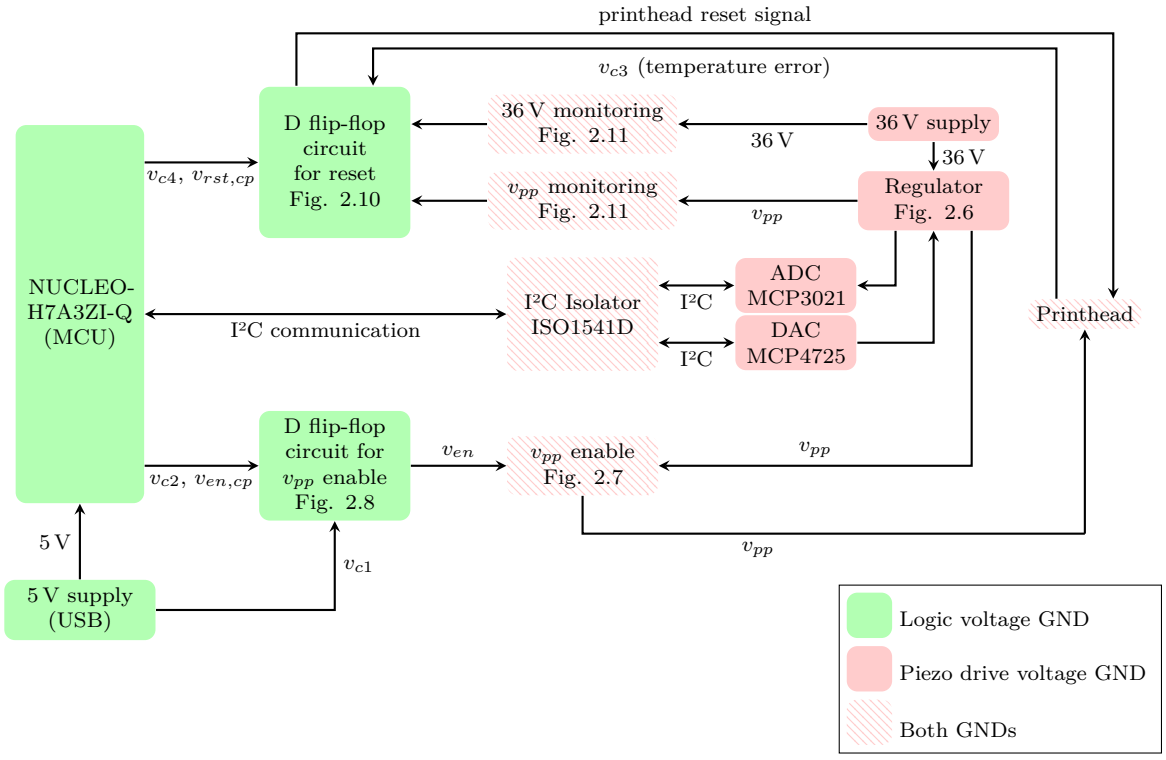


Figure 2.5: Overview of the circuits for printhead power supply, waveform control and printhead reset signal.

The 5 V logic level power supply is directly used from the NUCLEO development board, which gets powered by 5 V USB from the host PC. A second 36 V 1 A switching mode power supply, a LRS-35-36 (Meanwell), is used to create the piezo driving voltage.

Regulator design The piezo drive supply voltage is adjustable from 15 to 34.75 V, by using a LM317 linear voltage regulator as shown in the simplified circuit in Fig. 2.6. It is largely based on the reference circuit in the LM317 datasheet, but in addition, the output voltage, v_{pp} (which is used as piezo drive supply voltage here) can be controlled digitally by applying a voltage to the v_{dac} pin.

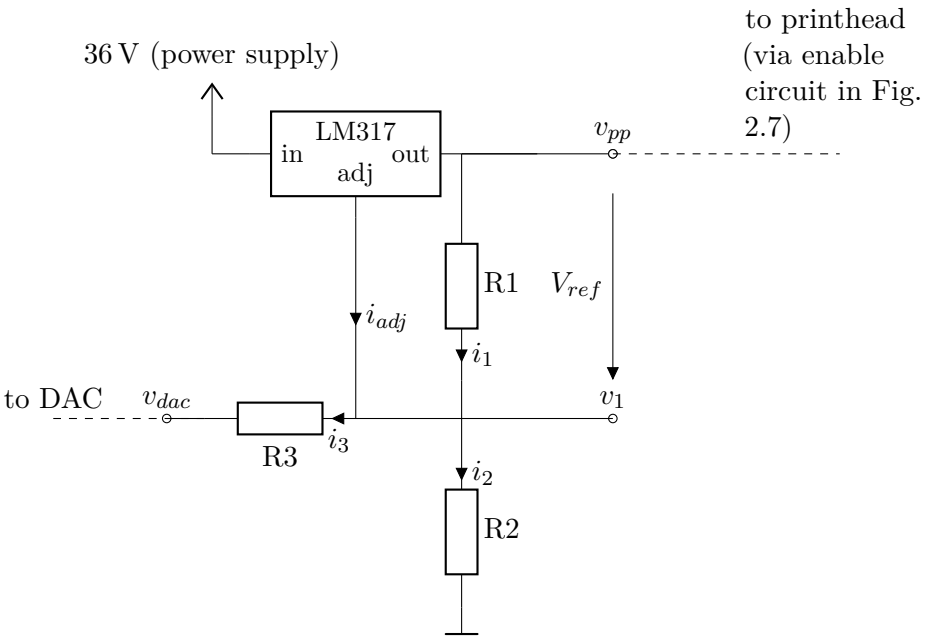


Figure 2.6: Simplified regulator circuit.

Using the two definitions for v_1 stated in eqn. 2.1 and 2.2 and Kirchhoff's current law (KCL) applied in equation 2.3, the following equality in eqn. 2.4 can be constructed that can be rearranged into an expression for i_3.

$$v_1 = R_2 i_2 \tag{2.1}$$

$$v_{dac} - v_1 = -i_3 R_3 \implies v_1 = i_3 R_3 + v_{dac} \tag{2.2}$$

$$i_2 = i_1 + i_{adj} - i_3 \tag{2.3}$$

$$R_2(i_1 + i_{adj} - i_3) = i_3 R_3 + v_{dac} \implies i_3 = \frac{R_2 i_{adj} + R_2 i_1 - v_{dac}}{R_2 + R_3} \tag{2.4}$$

When substituting i_3 from in eqn. 2.4 and the i_1 as in eqn. 2.5 again into eqn. 2.3, i_2 can be expressed as shown in eqn. 2.6.

$$i_1 = \frac{V_{ref}}{R_1} \tag{2.5}$$

$$i_2 = \frac{V_{ref}}{R_1} + i_{adj} - \frac{R_2 i_{adj} + V_{ref}\frac{R_2}{R_1} - v_{dac}}{R_2 + R_3} \tag{2.6}$$

Then v_{pp} can be calculated using Kirchhoff's voltage law (KVL) from V_{ref} and v_1 by substituting i_2 into eqn. 2.1.

$$v_{pp} = V_{ref} + v_1 = V_{ref} + R_2\left(\frac{V_{ref}}{R_1} + i_{adj} - \frac{R_2 i_{adj} + V_{ref}\frac{R_2}{R_1} - v_{dac}}{R_2 + R_3}\right) \tag{2.7}$$

Conforming to the LM317 data sheet i_{adj} is neglected as it is typically only 50 µA, giving the final eqn. 2.8. This way, the firmware can control the output voltage by using a digital analog converter (DAC) at the v_{dac} input[a].

$$v_{pp} = V_{ref}\left(1 + \frac{R_2}{R_1}\right) - R_2\frac{V_{ref}\frac{R_2}{R_1} - v_{dac}}{R_2 + R_3} \tag{2.8}$$

To determine suitable resistor values, eqn. 2.8 is optimized using MATLAB[b], to satisfy the following conditions: R_1, R_2 and R_3 should be in the range of 200 Ω to 100 kΩ. The two additional optimization variables $v_{dac-min}$ and $v_{dac-max}$ describe the voltage range the DAC must provide, and form two additional constraints, where the output voltage should be at the maximum possible $36\,\text{V} - V_{ref} = 36\,\text{V} - 1.25\,\text{V} = 34.75\,\text{V}$ when $v_{dac} = v_{dac-max}$ and 15 V at $v_{dac-min}$. Finally, the power lost over the resistors is constrained to be less than 0.5 W. Using the optimization goal of minimizing $v_{dac-max}$ to simplify the DAC design the resistor values are determined iteratively, by fixing the value of one resistor to the closest value of the E24 resistor series, and running optimization again. Using the resistor values $R_1 = 200\,\Omega$, $R_2 = 100\,\text{k}\Omega$ and $R_3 = 2.2\,\text{k}\Omega$, the v_{dac} must provide a range of 0.3 V to 20.49 V to get the desired 15 - 34.75 V for v_{pp}. This is achieved using a MCP4725 5 V DAC together with a non inverting operational amplifier circuit to multiply its output by 5.

To provide the high and low current piezo drive voltages, v_{pp} is split using two Schottky diodes. Each of the resulting power lines have a separate set of capacitors to stabilize. The firmware can connect/disconnect these supply lines to the printhead using p-channel MOSFET transistors.

Isolated ground The two piezo drive voltage lines grounds must, per Xaar 128 data sheet, be isolated from the logic voltage supply ground. For the firmware to be able to switch the gates of the transistors to the piezo drive voltage side while maintaining isolation an optocoupler is used. A simplified circuit diagram is shown in Fig. 2.7. In the following the logic voltage ground is labeled GND_1 and the piezo drive voltage ground GND_2.

When 5 V is applied to v_{en}, the current i_2 pulls down the gate of the MOSFET M_1 and the piezo drive voltage v_{pp} is connected to the printhead. The optocoupler LTV-356T has a current transfer ratio (CTR) of min. 50 % at $i_f = 5\,\text{mA}$. R_1 is therefore set to 1 kΩ. The function of the Zener diode D_1 is to protect the MOSFET: V_{SG} must not be larger than 20 V, and D_1 limits it to its Zener voltage of 5.1 V. With the minimum CTR of 50 %, i_2 is expected to be at least 2.5 mA, for the case where the piezo drive voltage is at its highest level of 34.75 V, however i_2 can be larger if the CTR is larger. Therefore, R_2 is introduced to protect D_1 from overcurrent, and calculated using eqn. 2.9.

$$R_2 = \frac{(V_{max} - V_{fzener})}{i_2} = \frac{34.75\,\text{V} - 5.1\,\text{V}}{2.5\,\text{mA}} = 11.86\,\text{k}\Omega \approx 12\,\text{k}\Omega \tag{2.9}$$

[a] When regarding the regulator schematic in Fig. A.2 of Appendix A, it can be seen that by switching a jumper on the PCB, v_{pp} can alternatively be set by a potentiometer, bypassing the firmware and DAC for testing purposes.

[b] MATLAB Problem-Based Optimization Workflow `https://www.mathworks.com/help/optim/ug/problem-based-workflow.html`

When the piezo drive voltage is at its lowest level of 15 V, it must still be ensured that, when it is enabled, V_{SG} is sufficiently high to ensure low resistance over the MOSFET. The MOSFET used is a Si2319CDS and specifies 0.09 Ω at $V_{SG} = 4.5\,\text{V}$.

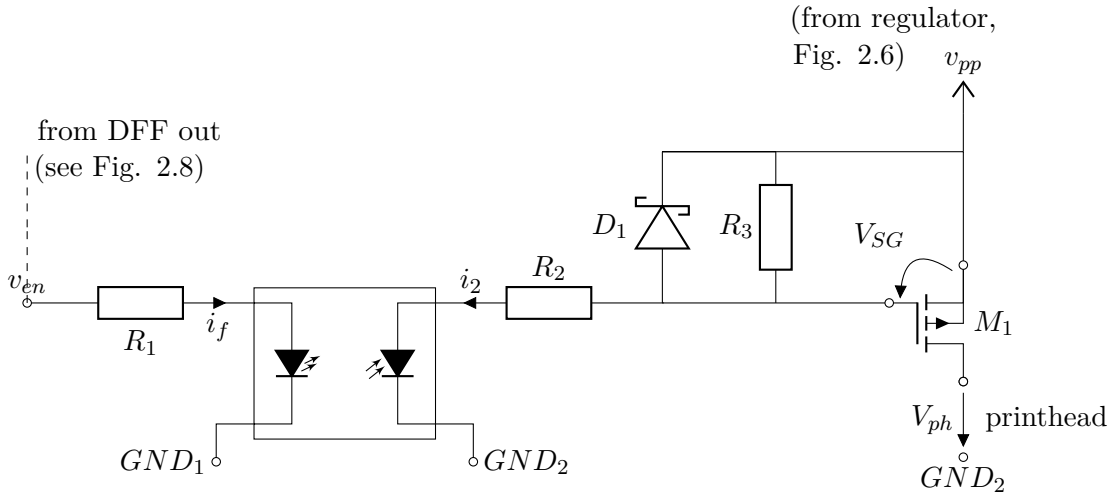


Figure 2.7: Optocoupler to enable the piezo drive voltage.

If D_1 was conducting, V_{SG} would be 5.1 V, so for this calculation, assume D_1 is not conducting. Therefore, R_2 and R_3 can be viewed as resistor dividers. Using eqn. 2.10 and eqn. 2.11, R_3 can be expressed as in eqn. 2.12

$$V_G = V_S - V_{SG} \tag{2.10}$$

$$V_G = \frac{R_2}{R_3 + R_2} \cdot V_S \tag{2.11}$$

$$V_S - V_{SG} = \frac{R_2}{R_3 + R_2} \cdot V_S \implies R_3 = R_2\left(\frac{V_S}{V_S - V_{SG}} - 1\right) \tag{2.12}$$

With the lowest $V_S = 15\,\text{V}$, R_3 is calculated to:

$$R_3 = 12\,\text{k}\Omega \cdot \left(\frac{15\,\text{V}}{15\,\text{V} - 4.5\,\text{V}} - 1\right) = 5.142\,\text{k}\Omega \approx 5.1\,\text{k}\Omega \tag{2.13}$$

As mentioned before, the the piezo drive voltage is set by the MCU using the MCP4725 DAC. Since this part is located on the printhead side of the isolated grounds, the MCU can not directly talk to the DAC over its I²C interface[a]. With the Texas Instruments ISO1541D I²C digital isolator in between the MCU and the DAC, communication is possible without connecting the grounds.

Additionally, to setting the piezo drive voltage, the MCU can also read the actual voltage and report it to the user interface. The voltage is read by a simple to use MCP3021 analog digital converter (ADC). The maximum piezo drive voltage is 34.75 V, which exceeds the maximum voltage (5 V) of the ADC. Therefore, a resistor divider is used, which transforms the piezo drive voltage into the allowable voltage range. The MCP3021 ADC is connected to the same I²C bus, as the MCP4725 DAC.

(Power) failure detection Since the logic voltage supply and the piezo drive supply are independent, there is a possibility that the logic voltage supply fails while the piezo drive supply is active. This poses a risk of damaging the printhead, as it violates the power sequence.

[a] I²C is a protocol that integrated circuits (ICs) on a PCB use to communicate. [32]

To mitigate this risk, the logic voltage supply is monitored using a simple Zener diode circuit[b]. This forces the piezo drive supply voltage to be disconnected from the printhead if the logic voltage supply falls below 4.3 V.

However, short time intermittent failures might stay undetected. To mitigate this risk a D flip-flop type 74HC173D is used, which was selected for featuring an active high asynchronous reset. Instead of directly connecting a pin of the MCU to the v_{en} signal in Fig. 2.7, the D flip-flop is used in between.

In the following explanation, digital signals are used, where high means 5 V, low means 0 V or ground. A signal rising means a transition from 0 V to 5 V, and falling the opposite, going from 5 V to 0 V. The function of the D flip-flop is that on a rising edge of the clock pin $v_{en,cp}$ the digital signal at the D-pin (input) is copied to the Q-pin [33]. Whenever the active high reset pin RST is high, the Q-pin is set low immediately, and the Q-pin can only rise again after the reset pin is low and another rising edge is applied to $v_{en,cp}$.

In case of a short time intermittent failure that stays unrecognized by the MCU, the output of the D flip-flop v_{en} is pulled to ground automatically, which disconnects the piezo drive voltage v_{pp} from the printhead. The circuit in Fig. 2.8 shows how the short time intermittent failures are detected, in a simplified circuit.

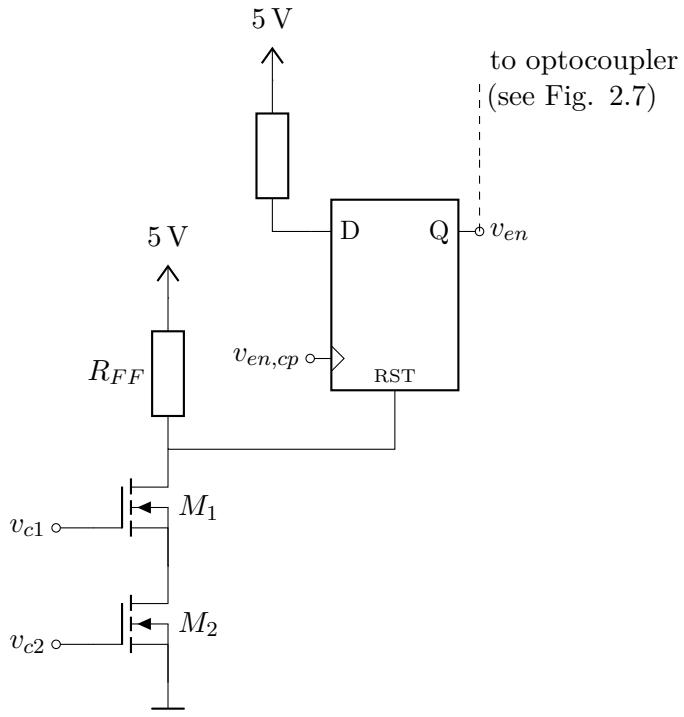


Figure 2.8: D flip-flop to detect intermittent failures.

Two conditions were implemented to allow the piezo drive supply to be enabled: The signal v_{c1} models the first condition and is high when the logic voltage supply is higher than 4.3 V. The second condition, v_{c2}, is connected to a pin of the MCU. It is set high by the MCU, to explicitly request connection of the piezo drive supply to the printhead.

[b]This circuit can be found in Fig. A.2 of Appendix A

Fig 2.9 illustrates normal operation and failure detection. When the printhead drive electronics is supplied with 5 V by connecting to the host PC with its USB connection, v_{c1} changes from low to high.

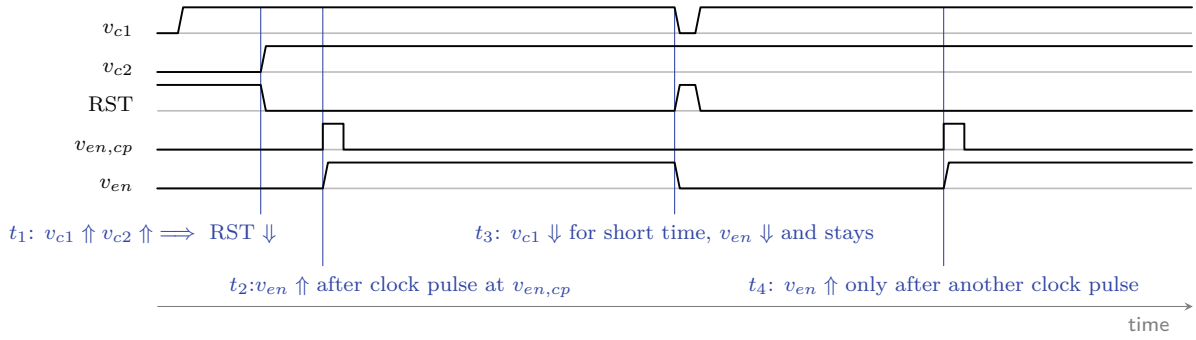


Figure 2.9: Timing diagram for failure detection using a D flip-flop.

Signal v_2 is low at startup, thus no current can flow through M_2, and RST is pulled high by R_{FF}. Therefore v_{en} is set to low by the D flip-flop, and the piezo drive supply is disconnected from the printhead.

At t_1, the user chooses to start printing, and the MCU requests to connect the piezo drive voltage supply to the printhead, and switches v_{c2} to high as well. Because both v_{c1} and v_{c2} are high, M_1 and M_2 pull RST to ground. At t_2 the MCU applies a clock pulse to $v_{en,cp}$, which causes v_{en} to rise and the piezo drive supply to be connected to the printhead. Now, printing operations can start.

At t_3, a power failure occurs at the logic voltage supply and v_{c1} falls for a short moment. Immediately RST rises, and subsequently v_{en} falls, disconnecting the piezo drive supply from the printhead. Altough v_{c1} rises again, v_{en} stays low, as another rising edge at $v_{en,cp}$ would be required to enable v_{en} again. This is also the case when the firmware has an error, and the $v_{en,cp}$ is stuck at high. As soon as the MCU has detected the failure condition, it is shown in the user interface, and the user can investigate.

If the user decides that the error is resolved, printing can continue: At t_4, the MCU applies another clock pulse to $v_{en,cp}$ and the piezo drive voltage supply gets connected to the printhead again.

More conditions can be added to such a circuit, by connecting more n-channel MOSFETs in series.

The complete circuit for printhead piezo drive voltage power supply and voltage control can be found in Fig. A.2 and Fig. A.3 of Appendix A.

Printhead reset signal The reset signal of the printhead is the last step in the printhead's power supply sequencing scheme. This means it is the last signal to switch when activating the printhead, and the first when deactivating. Different failure modes can lead to this power sequence to be violated:

1. The piezo drive voltage power supply fails, while the printhead is activated (not in reset condition).

2. The firmware has a bug and switches the reset signal on while it is not allowed to.

3. The logic voltage supply fails.

4. The printhead itself signals a failure condition (e.g. over-temperature error) and it is not shut down afterwards.

To protect the printhead, the following conditions are monitored: The over-temperature error signal of the printhead, a MCU pin set high by the firmware, and the piezo drive supply voltage must be in the valid range. The reset signal can only be activated if all these conditions are met. To ensure that the reset is not accidentally switched off and on again, if a failure occurs only for a short time, an additional D flip-flop circuit, similar to the one in Fig. 2.8, is used. This way, if any error caused the printhead to be reset, the firmware has to toggle a MCU pin to activate it again.

This time conditions v_{c3} and v_{c4} represent the over-temperature error signal of the printhead and the MCU pin that requests the printhead to be activated. However, two more conditions are added to the second D flip-flop circuit by connecting two instances of the subcircuit shown in Fig. 2.11. One of them is for monitoring the 36 V power supply, and another for monitoring the 15-34.75 V output of the regulator. By monitoring both voltages, it can be detected whether the fuse blew, which is situated directly after the regulator circuit. The second advantage is that failure of the piezo drive power supply might already be detected while the capacitors after the regulator are still charged. The modified D flip-flop circuit, with the two additional conditions added, is shown in Fig. 2.10.

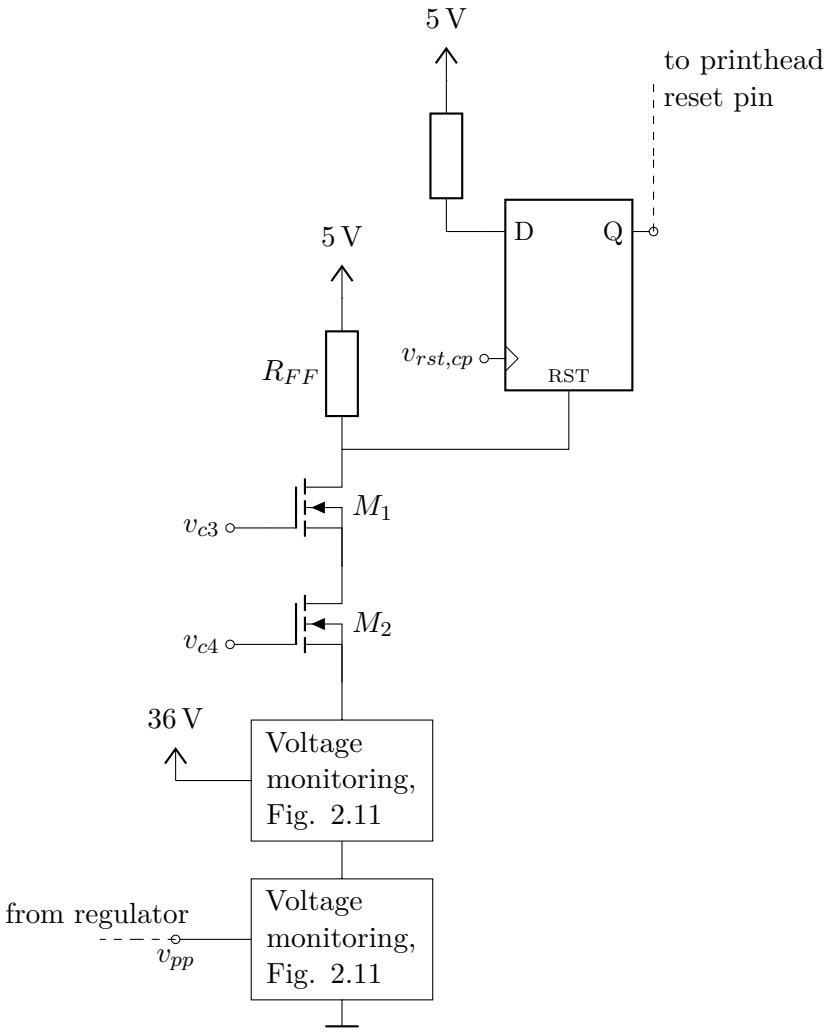


Figure 2.10: Second D flip-flop circuit to monitor the conditions for the printhead reset signal.

Power supply and regulator voltage monitoring Since the reset signal is part of the logic voltage side, and the piezo drive voltage grounds are isolated, a 6N135 high speed optocoupler is used to cross the boundary. A signal is transferred, indicating whether the

piezo drive voltage is in an acceptable range. A simplified circuit diagram is shown in Fig. 2.11.

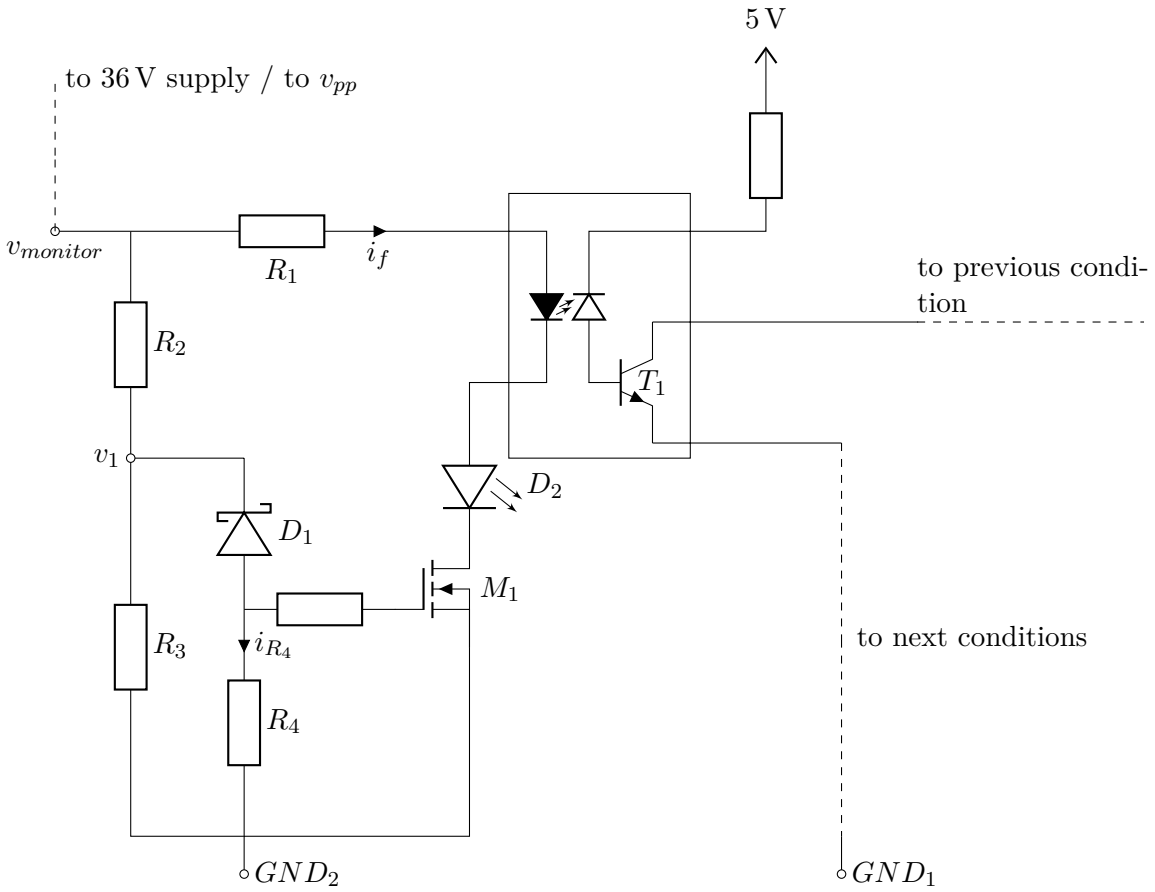


Figure 2.11: Optocoupler circuit to monitor piezo drive voltage.

The voltage to monitor is applied to $v_{monitor}$. If $v_{monitor}$ is lower than a threshold voltage $V_{shutdown}$, the n-channel MOSFET M_1 is not conducting. Consequently, also T_1 is not conducting, and the RST pin of the D flip-flop is at 5 V (pulled up by R_{FF} in Fig. 2.10). This prevents the printhead's reset signal from being switched by the MCU. When the $v_{monitor}$ is higher than $V_{shutdown}$, the n-channel MOSFET M_1 and subsequently T_1 are conducting. If the rest of the conditions that are implemented in the second D flip-flop circuit are met as well, the MCU can activate the printhead by switching the printhead reset signal. D_2 is an indicator light for the user, which is on when the voltage is over $V_{shutdown}$.

The 6N135 optocoupler comes with a higher speed than the LTV-356T, but exhibits lower CTR of 5 %. To make sure that the transistor in the optocoupler can pull the RST line of the second D flip-flop to ground, the relation of R_1 and R_{FF} (R_{FF} in Fig. 2.10) must be correct. Assuming $R_{FF} = 20\,\text{k}\Omega$, the current through it should be 0.25 mA. With the CTR of 5 %, i_f needs to be 5 mA.

$$R_1 = \frac{v_{monitor} - v_{f,optocoupler} - v_{f,D_2}}{i_f} \tag{2.14}$$

Using eqn. 2.14, which factors in the forward voltage of the optocoupler's diode $v_{f,optocoupler} = 1.4\,\text{V}$ and of the indicator light, R_1 must be $\approx 5.1\,\text{k}\Omega$ for the 36 V case and $\approx 2\,\text{k}\Omega$ in the 15 V case.

With the Zener voltage V_{D_1} and equivalent series resistance of the conducting Zener diode Z_{D_1}, the voltage at v_1 can be expressed as:

$$v_1 = V_{D_1} + Z_{D_1} i_{R_4} + v_{gs} \tag{2.15}$$

Using eqn. 2.15 and 2.16 in the KVL of eqn. 2.17, the gate voltage of the n-channel MOSFET is written in eqn. 2.18.

$$i_{R_4} = \frac{v_{gs}}{R_4} \tag{2.16}$$

$$v_{monitor} - R_2(\frac{v_1}{R_3} + i_{R_4}) - v_1 = 0 \tag{2.17}$$

$$v_{gs} = -\frac{R_2R_4V_{D_1} - R_3R_4(v_{monitor} - V_{D_1})}{R_2R_3 + R_2R_4 + R_3R_4 + R_2Z_{D_1} + R_3Z_{D_1}} \tag{2.18}$$

To find the values of the resistors R_1-R_3 and the Zener voltage V_{D_1}, optimization using MATLAB was carried out. $V_{shutdown}$ being the threshold voltage, where the optocoupler stops pulling the reset low, a range $[V_{shutdown} - v_{down}, V_{shutdown} + v_{up}]$ is defined. The optimization process should minimize this range, so that the transition happens fast. Therefore, the optimization variable is $v_{up} + v_{down}$. The following constraints were applied:

1. $v_{gs}(v_{monitor} = V_{shutdown}) = V_{th}$, placing the transition at the correct point (V_{th} is the threshold voltage of the MOSFET).

2. $v_{gs}(v_{monitor} = V_{shutdown} - v_{down}) \leq 0.3\,\mathrm{V}$: When $v_{monitor}$ is below the shutdown voltage range, v_{gs} should be sufficiently low to turn M_1 off.

3. $v_{gs}(v_{monitor} = V_{shutdown} + v_{up}) \geq 1.5\,\mathrm{V}$: When $v_{monitor}$ is above the shutdown voltage range, v_{gs} should be sufficiently high to turn M_1 on.

4. $v_{gs}(v_{monitor} = V_{max}) \geq 4\,\mathrm{V}$: At the maximum voltage at $v_{monitor}$, v_{gs} should turn M_1 fully on.

5. i_{R_4} should be at least 1 mA when $v_{monitor} = V_{shutdown}$ to ensure a steady Zener voltage and not more than 4 mA when $v_{monitor} = V_{max}$, to avoid overcurrent at the Zener diode.

The optimization was carried out multiple times, fixing a variable after each run. First, the Zener diode was selected and a prior estimated Z_{D_1} value was replaced with values from the data sheet. Then, a resistor was selected from the E24 resistor series. For monitoring the 36 V supply, the following values were determined: $V_{D_1} = 27\,\mathrm{V}$, $R_2 = 1.2\,\mathrm{k\Omega}$, $R_3 = 31.6\,\mathrm{k\Omega}$ and $R_4 = 1.6\,\mathrm{k\Omega}$. Therefore, $v_{gs} = 0.3\,\mathrm{V}$ when $v_{monitor} \approx 28.6\,\mathrm{V}$ which turns the transistor off, and $v_{gs} \approx 4.3\,\mathrm{V}$ when $v_{monitor} \approx 36\,\mathrm{V}$ which turns the transistor fully on.

For monitoring the 15-34.75 V regulated voltage the following values were determined: $V_{D_1} = 10\,\mathrm{V}$, $R_2 = 5.1\,\mathrm{k\Omega}$, $R_3 = 68\,\mathrm{k\Omega}$ and $R_4 = 1.6\,\mathrm{k\Omega}$. This leads to a $v_{gs} = 0.3\,\mathrm{V}$ at $v_{monitor} \approx 12\,\mathrm{V}$ which turns the transistor off, and $v_{gs} \approx 1.7\,\mathrm{V}$ at $v_{monitor} \approx 17\,\mathrm{V}$ which turns the transistor on.

Testing performed on the PCB that was built, showed that the circuits function as designed, in the laboratory environment where the printer was used. Tests for the failure detection and mitigation mechanisms were conducted with an oscilloscope, without a printhead connected, so that no printheads were damaged in the course of printing experiments. However, there is room for improvement: The threshold voltage of the DMN5L06K MOSFET is stated as 0.4 V to 1 V, indicating variations, moreover threshold voltage of MOSFETs is also temperature dependent [34]. Therefore temperature influence, also on the resistor network, should be analyzed, as it could cause variation in the actual shutdown voltages.

A different approach, using comparators together with a reference voltage generator, with a wide supply voltage range and buffering capacitors to ensure that they can fulfill their function correctly when the supply voltage drops, may be a better solution.

2.2.2 Level shift

The STM32H7 MCU is configured to run on 3.3 V, while the printhead uses 5 V logic levels. This means logic levels of the signals to and from the printhead need to be shifted. To provide this function, 74AHCT125 3 state buffers are used. They have an output-enable signal that can set the outputs in high impedance mode, to prevent any signals to the printhead while it is deactivated. The output-enable signal can be controlled by a MCU pin.

2.2.3 Strobe

An LED light can be connected to the printhead drive electronics that serves as the strobe for the dropwatcher. It is driven by an n-channel MOSFET. The LED should have a high brightness to function as a strobe light, but since the LED is only turned on for a very short time (e.g. 3-4 µs), a heatsink is omitted in this design. To mitigate the risk of burning out the LED light when there is a failure in the MCU firmware, e.g. when the the control pin is kept on, a highpass filter is added before the gate of the transistor. This way, the LED turns off after a short time, even if the MCU pin is stuck at high.

2.2.4 Firmware

In the following, the program that was implemented in this thesis to run on the STM32 MCU will be referred to as firmware. While it is programmed in C, different approaches were considered as a basis, to not start completely from scratch. One option was to extend an existing open source 3D printer firmware, examples being Marlin[a] and Klipper[b]. These firmwares typically communicate with another software on a host PC. In case of Marlin, the host PC's role is to stream a prepared file in GCODE, which is a list of instructions, most of them movement commands that the machine should execute, via a serial connection. When using Klipper, the GCODE is further processed on the host PC respecting machine kinematics, and even lower level commands are sent via the serial connection, such as stepper positions and GPIO values. In both cases, it would have been possible to extend the protocol by additional commands for controlling printhead, ink supply, etc. However, this approach would be more suitable if the motion stage controller and the printhead drive electronics were combined in a single module controlled by a single firmware. The motion stage for this project is adapted from a discarded 3D printer, which included a still functioning controller.

A different option is to use a hardware abstraction layer (HAL), or a real time operating system (RTOS), like it was used here, as basis for the printhead drive electronics firmware. A HAL, such as STM32 HAL or STM32 LL for the STM32 MCU family, provides abstractions, higher-level functions that a developer can use to implement functionality without knowing the exact details of how the underlying hardware must be configured. It also creates code that can be ported between different microcontrollers (of the family) easily [35]. Using the STM32 HAL, peripherals like timers or communication controllers can be configured, including their advanced functionalities. If many tasks should be executed on one MCU, a RTOS can be beneficial. It is called OS (operating system) because it provides ways to schedule tasks, the separate jobs that the firmware executes. RT (real time) refers to the scheduling and execution being reproducible, and predictable. [36]. In this project the main concurrent tasks of the firmware are:

- controlling the pumps of the ink supply, e.g. to control the pressure within the ink system to a set value,
- sending the nozzle print data to the printhead at the correct time,

[a] Marlin: `https://marlinfw.org/`
[b] Klipper: `https://www.klipper3d.org/`

- processing the encoder input and triggering the nozzles to fire,
- communication between the printer and the host PC.

Zephyr Project The Zephyr Project[a] is an open source RTOS that is actively developed by many organizations and the open source community. It has a small kernel that allows program execution in multiple threads. It also provides work queues, where tasks can be scheduled, synchronization primitives like events and semaphores, logging/debugging functions all the way to a framework to build state machines with. Using the concepts of a device tree and drivers for peripherals, including attached sensors, storage devices, communication interfaces it is highly modular and portable from/to different platforms. A lot of drivers have been implemented by open source contributors for various sensors and other devices. The Zephyr Project includes many communication stacks (e.g. ethernet, bluetooth), for this project USB was used to communicate with the host PC.

An abstraction model and device specific drivers make it modular. For example, the same application code can be used with any pressure sensor, by exchanging the driver, meaning that sensors can be exchanged effortlessly [37]. An adaptable, future proof design is very valuable in a product. However, following these patterns can slow down the prototyping phase. It took a while to grasp the concepts of the device tree, and when lower level hardware had to be used, device drivers had to be created for it.

Also, even if standard drivers exist for a piece of hardware, special functionality of it is often not included in the generalized device interface. For example, here, the timing of the print signal and the strobe is critical for the dropwatcher to work. To provide perfect synchronization, specialized hardware timer functionality was used that links two hardware timers together (further explained in Section 2.4.2). To support this, a custom developed driver was needed, because the default timer driver interface does not support timer synchronization configuration.

[a] Zephyr Project documentation: `https://docs.zephyrproject.org/latest/index.html`

2.3 Ink supply system and printhead maintenance

The ink supply system and the printhead maintenance share several components and are therefore discussed together in this section.

2.3.1 Requirements

The following requirements were followed for the use case presented in this work. Some of them were also reported to be important for inkjet printing in general:

- Printheads have an ink dependent range of ink pressure specified at which they operate [30]. For the Xaar 128 a negative pressure in the range of 1 to 10 mbar is specified.
- Filtering the ink before loading it into the printhead prevents too large particles from entering [30].
- Air/Vapor bubbles in the printhead can prevent nozzles from jetting [30] [38]. During development it happened that air bubbles entered the system, either when components where exchanged or by user error. Therefore, there should be a way to flush them out before they can enter the printhead.
- A capping station seals the printhead's nozzle plate to prevent drying out of nozzles and to protect from contamination. It can also be used to perform vacuum purges.
- By monitoring the pressure, errors like leaks, obstructions or sensor failures can be detected and the system can be shifted into a safe state.
- For a prototyping system, it is important that only small amounts of ink are needed. Additionally, quick changes between different inks without wasting vast amounts of ink need to be possible.
- For the ink produced, described in Section 3.1, the particles stayed in suspension only for some amount of time before sedimenting and/or agglomerating. Damage to the printhead should be reduced, by purging the ink using pressure or vacuum purging techniques, and by flushing the system with cleaning fluid, after the printing is finished.

2.3.2 Implementation

The assembly of the separate components of the ink supply system is presented in Fig. 2.12.

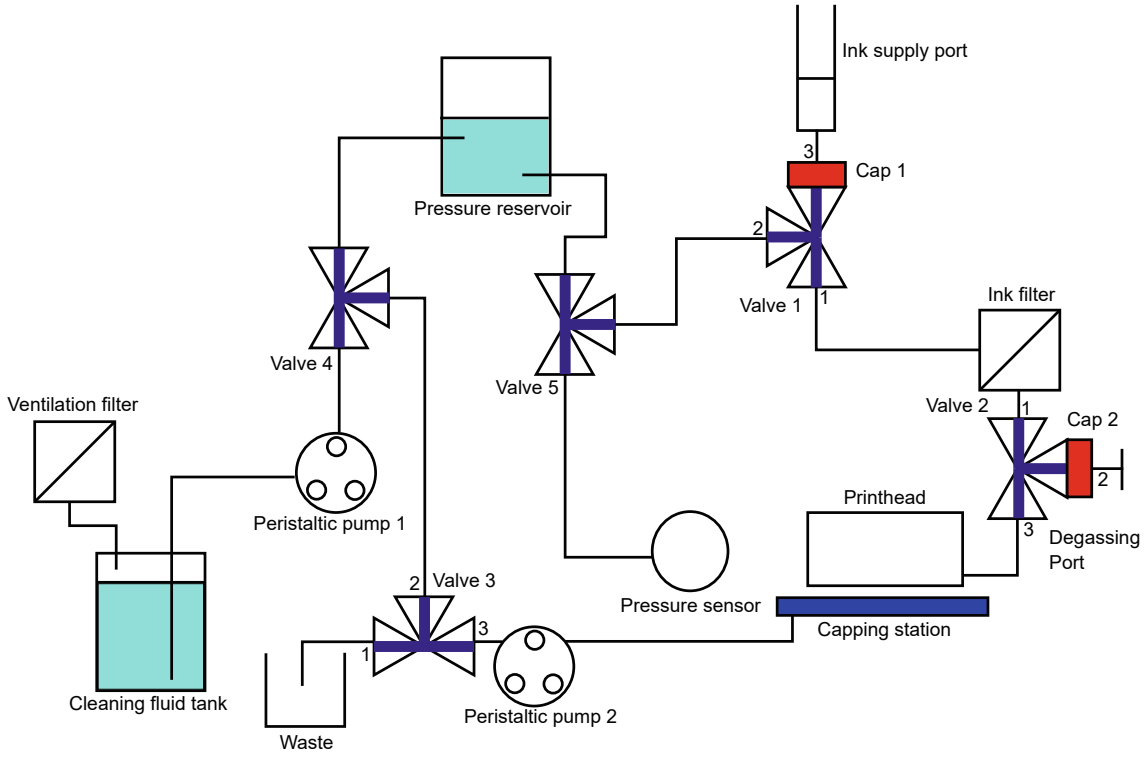


Figure 2.12: A schematic view of the implemented ink supply system.

The capping station is assembled from 3D-printed parts that hold a machined polycarbonate insert. When closed, the print head is pressed onto the insert, separated by a silicone seal (A in Fig. 2.13), by moving the z-axis of the motion stage down. Flexures (B in Fig. 2.13) in the 3D-printed part allow the insert to align in such a way that a seal can be created with the nozzle plate. A tubing connected to the insert (C in Fig. 2.13) allows applying a vacuum to the nozzle plate. While printing, the capping station folds away using a rotary mechanism (D in Fig. 2.13), and is magnetically locked.

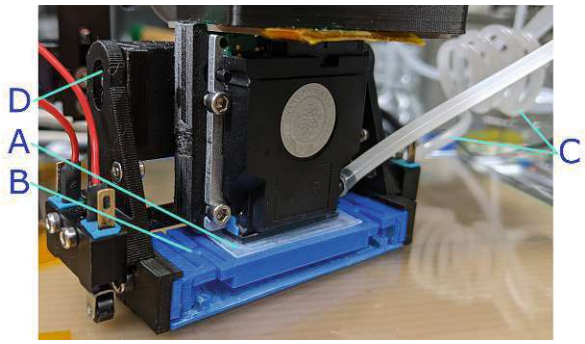


Figure 2.13: Printhead docked in the capping station: silicone seal (A), 3D-printed flexures (B), tubing (C) connected to the insert, and rotary mechanism (D).

The printhead can be loaded with ink either by applying a vacuum to the nozzles via the capping station or by pushing an ink manually into the ink supply port using a syringe.

The path between the ink supply port and the printhead is kept as short as possible, to reduce the amount of ink required for a single loading. Its main element is a permanently

installed 5 µm filter to prevent any contamination and large particles with the ability to clog the nozzles. The 2.5 mm ID polyethylene tubing can be pushed over the printhead's ink connector with a little force and creates a good seal.

Valve 2 is used to remove air bubbles trapped in the filter, so that they do not enter the printhead.

The pressure reservoir provides a small working volume to adjust the pressure. When the peristaltic pump 1 adds cleaning fluid to the pressure reservoir, air in the pressure reservoir is compressed. This allows more fine grained pressure control, as one rotation of the peristaltic pump leads to a smaller pressure change, at the cost of speed. The reservoir sits at the top position of the system with its ports on the bottom, below the liquid's surface, so that air bubbles entering are trapped within the pressure reservoir and do not further circulate through the system.

The pressure sensor is located at the bottom of the system, as close as possible to the printhead in terms of height, in order to reflect the ink pressure in the printhead itself.

The use of two pumps is not strictly necessary, however using two pumps reduced the amount of installed valves, and thus decreased the number of operational errors. Only valve 1, 2 and 3 are manually switched to execute the different operations and maintenance procedures. The valves 4 and 5 are always in the "connect all" position and therefore act as T-pieces.

Both pumps and the pressure sensor are connected to the printhead control electronics. It monitors the pressure, and provides two basic operations of the two pumps:

- Target pressure: The pump is actuated by a propotional-integral (PI) controller [39]. The controller minimizes the error (in mbar) between the pressure setpoint and the value measured by the pressure sensor. The controller's output are a pulse width modulation (PWM) duty cycle and direction value, which are applied to the pump motor to influence its speed and pumping direction. The control frequency is set to 10 Hz as a compromise of reaction speed and load to the MCU. The two parameters $k_p = 0.15$ and $k_i = 0.5$ of the PI controller were determined empirically: The initial proportional coefficient was chosen by measuring the pressure change of the pump running at full speed over time and scaling it for the control window of 100 ms. It became clear that the pump motor cannot spin below a PWM duty cycle of 0.3, which defined the minimum duty cycle to turn on the motor. This introduces even more steady state error, and the integral controller part is needed to mitigate that. However, because the maximum duty cycle is 1, and it can take several seconds to reach the target pressure when a big change in pressure is requested, the integral error was capped at 1.

- Feed with limit: The pump is actuated in a specified direction, for a certain amount of time with a user-defined PWM duty cycle. This is for example used to vacuum purge residual ink from the printhead using the capping station. A pressure range is specified that must not be exceeded. If the pressure approaches the limit, the PWM duty cycle is set to $D = min(D_{requested}, k_p \cdot |p_{limit} - p_{sensor}|)$, and therefore slows down and finally stops completely.

Any of the two operations can be issued for any of the pumps individually. During the actual printing process, only peristaltic pump 1 is used to create the suitable negative ink pressure requirement of the printhead, by leveraging the target pressure operation. The print system software provides several maintenance routines that can be executed by the user.

In the following, frequently used routines are described shortly.

- Fill pressure reservoir: To top up the pressure reservoir, the path to the printhead is closed by valve 1, and a vacuum of −250 mbar is created by pump 1. If the fluid level is low, air is pumped out to the cleaning fluid tank, and is replaced by fluid after pump 1 restores the pressure to 0 mbar.

- Keep-alive state: Valves 1-3 are set in a way that pump 2 recirculates cleaning fluid through the printhead, capping station and pressure reservoir every hour for five seconds, while the system is idle, to keep the nozzles ready and free from air bubbles.

- Supply tubing/filter degassing: After the path to the printhead is shut off by valve 1, valve 2 is switched to direct fluid to the degassing port, also closing the path to the printhead. Pump 1 builds up pressure of 350 mbar, which is subsequently used to flush by opening valve 1 for a short moment, letting air escape via the degassing port.

- Power pressure purge: It is performed after the bulk of the residual ink is removed by vacuum purge or mild pressure purges after printing, or to remove air in the printhead. Implemented similar to 'Supply tubing/filter degassing', but this time valve 2 is directed to the printhead forcing the cleaning fluid through the nozzles instead.

- Mild pressure purge: Can be used to remove residual ink from the printhead, without installing the capping station. Pump 1 is used in the feed with limit mode to force ink through the printhead with a maximum pressure of 30 mbar for 5 s.

- Load ink: By applying vacuum to the nozzle plate with the capping station and pump 2, ink can be sucked into the printhead from the ink supply port if valve 1 is switched to connect the ink supply port to the printhead.

- Vacuum purge nozzles: Residual ink is purged by applying vacuum to the nozzle plate using the capping station and pump 2, in the feed with limit mode for 15 s. Valve 3 is set to direct fluid into the waste container. Meanwhile, pump 1 prevents a vacuum to appear on the printhead's supply side, by regulating pressure to 0 mbar in target pressure mode.

- Nozzle plate wetting: The wetting nozzle plate version of the Xaar 128/Irix printhead specify a thin film of ink spread on the nozzle plate to work correctly. To introduce such a thin film of ink, the pressure in the printhead is regulated to 4 mbar, and to -0.5 mbar after 3 s. This creates a small drop of ink on the nozzle plate that is further spread by moving the printhead over a small piece of silicon rubber on the build plate, without touching the nozzle plate directly.

A leak detection safety function was implemented as follows: While the target pressure operation is in progress, the system monitors whether the error is actually decreasing by calculating the moving average of $error_i - error_{i-1}$ of the last 8 control cycles. If the result is positive, a low pressure change flag is activated. This flag can be cleared when the error has decreased again for 0.5 s, but if it persists for 5 s, the printhead control electronics enters the failure state to prevent any further damage, as it indicates a leak or obstruction of some kind.

Component selection Pump 1 and 2 are peristaltic pumps of the type WPM447 (Whadda). The TB6612FNG dual H-Bridge IC (Toshiba) on a breakout board can control the pumping direction and intensity via PWM signals for two pumps. It is used in combination with a 6 V 600 mA power supply.

The pressure sensor is a ABPDRRV015PDSA3 (Honeywell). It was chosen because it was added after the first iteration of the printhead drive electronics, and could be easily soldered to a perfboard in its DIP-6 package. It can measure differential pressure between its two ports and operates in the range of ± 1034 mbar, features a digital output and supports liquids.

Millex-SV 5 µm, PVDF, 25 mm filters are used as ink filters.

As connection components, Luer lock three way valves (Discofix, B Braun) and Luer lock fittings (CPC LM 41/LF 41) are used together with 2.5 mm ID/4 mm OD polyethylene tubing. At first, flexible silicon tubing (type Rotilabo silicone standard design 2.5 mm ID/4.5 mm

OD) was tested, but it leaked ethanol vapor, which lead to ethanol vapor bubble formation inside the tubing, impacting the printing performance and ultimately emptying the system over longer pause periods. It is important to consider fluid compatible of the components used. For example, the body of the Discofix valves is made of polycarbonate, which was not compatible with ink containing propylene carbonate solvent, as it caused stress cracks after just a few seconds of contact.

2.4 Motion stage

The job of the motion stage is to move the substrate relative to the printhead, to place the droplets at the correct positions. Because of this, the accuracy of the printer is directly linked to the accuracy of the motion stage.

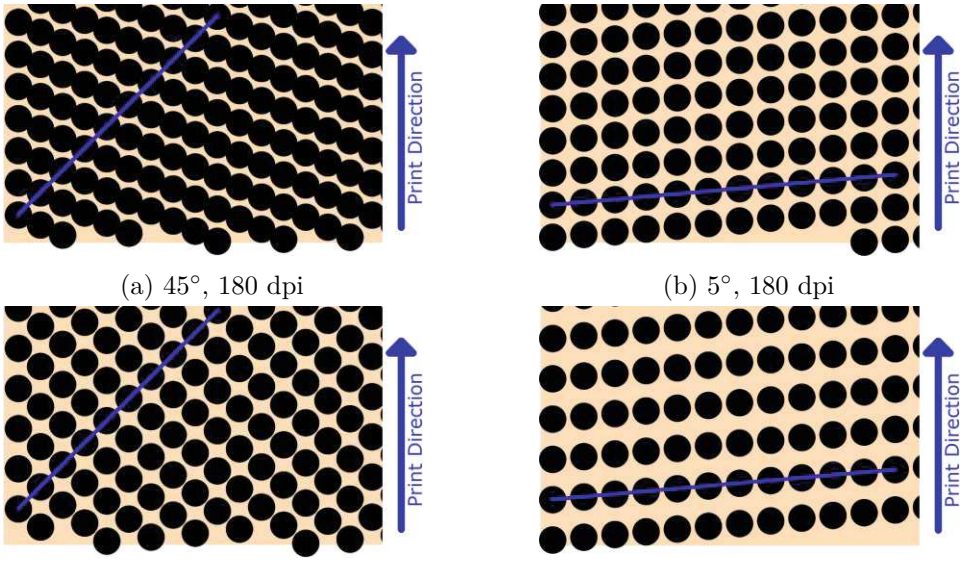
2.4.1 Requirements

It is helpful to dissect the requirements into the separate dimensions of the 3-dimensional space.

The z direction controls the distance between the printhead's nozzle plate and the substrate, called the throw distance [10]. The Xaar Irix data sheet specifies a print distance of $< 5\,\text{mm}$. Since the printed layer's thickness is only a few micrometers, it was not necessary to move the printhead up as layers build up. However, the user needs to adjust the printhead distance to the thickness of the substrate that was placed before starting the print. For printing, a print distance of $\approx 1\,\text{mm}$ in addition to the substrate thickness was used.

In the scanning multi-pass process [24] used in this project, the printhead moves relative to substrate in tracks. For each track, the printhead fires the nozzles in the right moment to print the droplets of this track from start to end. Then the motion stage moves so that the next, adjacent track can be printed. This process is repeated until all tracks are printed. As an optimization, tracks where no objects are placed can be skipped, and the start/end positions of a track are set to only move the printhead in the area where objects are placed.

The axis in the direction of the tracks will be called "print axis" in the following, and is the y-axis for this printer, while the axis across the printing tracks, x-axis in this case, will be called "move axis", to match the names used in the software code. Inside one track, the placement of the drops across the track is dependent on the pitch of the nozzles of the printhead and the angle the printhead is mounted with respect to the print axis. Along the track, the placement depends on the accuracy of the firing signal with respect to the position of the printhead. This is shown in Fig. 2.14: Different DPI values vary the overlap in the print axis direction, the droplet density can be varied as a function of both dpi and angle.



(a) 45°, 180 dpi (b) 5°, 180 dpi

(c) 45°, 120 dpi (d) 5°, 120 dpi

Figure 2.14: Printing pattern of a track at different DPI and different printhead angles. The angled blue line indicates the printhead nozzles, these dots are printed at the same time.

The repeatability of the move axis directly influences how well the tracks are stitched together. Inaccuracies lead to gaps between, or overlap of the tracks. When printing multiple

layers on top of each other, droplets can be placed with a specific overlap to the underlying droplets, dependent on the repeatability of the movement stage.

Generally, low repeatability limits the amount of details that can be printed, as features get more blurred, when the same feature is slightly offset across layers due to imprecise positioning.

2.4.2 Implementation

A modified Vellemann K8200 3D FDM printer provides the motion stage. This older 3D printer model has been available since 2013. A curious aspect of this printer that sets it apart from most modern printers, is that the extruder only moves in z direction, while the build plate moves both in x and y directions. From the motion design perspective, this is not ideal, as the build plate is usually the part with more mass. Also, when it has to move in two directions, the motor for the first axis must be moved by the second one. Higher moving mass leads to lower resonance frequencies and therefore limiting the print speed. However, this particular property simplified the design of the prototype printer: An inkjet printhead typically has more electrical signals than an FDM extruder, and ink supply lines come on top of that. It is easier to route these cables and tubing if the printhead rests at this almost static position, as it only has to move a few millimeters in z direction.

Most of the modifications on the printer are mounts for the various components of the electronics, printhead, vision and ink supply system. The extruder was removed, but the steppers motors, smooth rods and linear bearings, drive belts, endstops and the control electronics were used as was. An additional endstop was added to the printhead's capping station, which prevents it from crashing into the build plate. In order to improve reliability, the cables connected to the printbed were deep cleaned and rewired and a 3D-printed cable chain was added. The latest version of Marlin has been compiled and flashed onto the control electronics so that all G-codes commonly used today are understood.

Tests were showing that an increase in temperature of the heated bed drastically improved drying times of layers. By using an additional 24 V power supply and a small circuit with an n-channel MOSFET, the heatbed could reach 100 °C instead of 45 °C that were possible with its original power supply, without need to further modify the drive electronics.

However, the most important modification was adding a linear incremental encoder to the print axis (y-axis). The encoder is salvaged from an Epson L355 desktop inkjet printer, and consists of a transparent strip of about 30 cm length with dark lines spaced ≈141 µm apart. This distance corresponds to a 180 DPI spacing. A readout sensor like this consists typically of a light source and photo detectors, together with comparators forming a digital output signal [40]. Two readout positions are spaced with 90° phase difference, producing a quadrature encoded output signal. By counting the pulses of the two output signals, the direction being detectable due to the phase difference of the signals, the printer can keep track of the position of the stage.

Encoder based drop placement One way to place the droplets accurately is to command the stage to move to the exact position and then fire the nozzles. For this to work, the movement and the fire signals must be synchronized. It must be assured that the stage has reached the position before the nozzles are fired.

A different approach is to move the printhead relative to the build plate at constant speed and fire the nozzles at the right moment in time. This way, it is not necessary to move the stage to an exact position, rather to know where it is and to fire the printhead at the right place. This is what the encoder signal is used for. Every time an encoder line is scanned by the sensor, the nozzles of the printhead can be fired. The resolution can be further increased by triggering drop ejection on every rising, and falling edge of the two quadrature signals respectively. The resulting resolution is 720 DPI, which means, a dot can be placed every

≈35 µm. To achieve repeatability, any random delay between the sensor scanning a line and the printhead ejecting must be avoided. Typically, a random delay could be introduced by software processing, for example a task can be queued if a task of higher priority is executed at the moment. Also switching context between tasks takes some time. To keep the latency between the sensor signal and the nozzle firing signal constant, in other words minimize jitter, a specific configuration of timers of the STM32H7 MCU is used. Fig. 2.15 shows the different signals when printing a track.

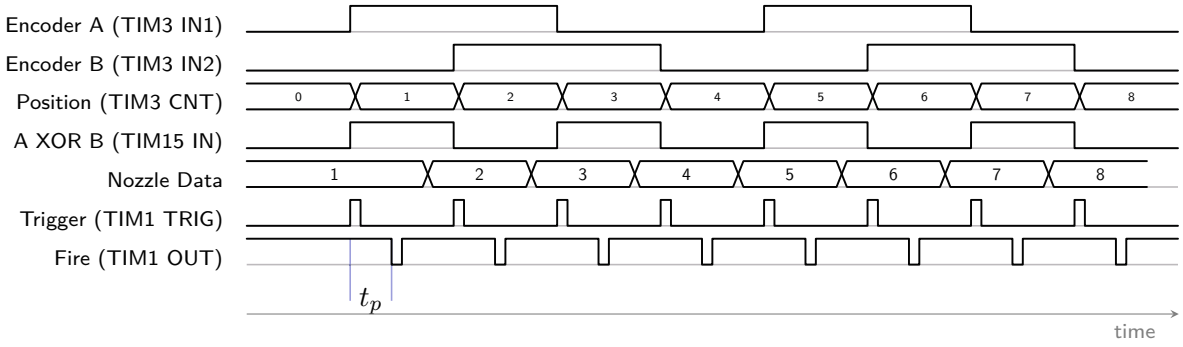


Figure 2.15: Encoder/Timing signals during normal printing operation over time. *A* and *B* are the signals from the encoder sensor. The *Position* signal is the MCU internal representation of the position in response to *A* and *B*. *A XOR B* is the result of the logical XOR operator on the A and B signals that is further used as *Trigger* signal for Timer 1. At the *Fire* pulse the printhead fires the nozzles. The constant delay t_p between *Trigger* and *Fire* provides a window for computations. The *Nozzle Data* encodes which nozzles eject at *Fire* and have to be written before the *Fire* pulse.

The quadrature encoded signals A and B are connected to the input pins of Timer 3 (TIM3 IN1/2), which is configured in encoder mode to track the position of the axis. The encoder signal is connected by a shielded cable to the MCU, additionally an input filter[a] is activated in the timer configuration. With the timer running at 96 MHz, the tracking works without missing any steps at the fastest speed the axis can move. This was tested by moving back and forth to the (mechanical) end of the axis and comparing the encoder values. As shown by the position row in Fig. 2.15, the timer counts on both edges of both inputs.

Timers of STM32 microcontrollers can be used in primary/subordinate mode, which means that another timer can be started (triggered) by an event of a primary timer. This can be used to create constant delays between input and output events, here to create a constant delay between the encoder and the fire signal. Since the encoder position cannot be used to trigger Timer 1 directly, the following workaround was implemented: Timer 15 has the special feature, where the XOR (logical operation) of the input signals of Timer 3 (TIM3 IN1/2) can be used as an input signal. The resulting signal of the XOR operation (TIM15 IN) has a rising or falling edge on any transition of one of the encoder signals. For this channel, the input capture interrupt is activated, set to both edges. This means the input capture interrupt will be triggered every time when the input to Timer 15 (TIM15 IN) rises or falls. The interrupt signal is selected as trigger output, and Timer 1 is set to use this signal as trigger signal, labeled as Trigger (TIM1 TRIG) in Fig. 2.15. Consequently, Timer 1 is triggered every time the enocder position (TIM3 CNT) changes.

The output signal of Timer 1 (TIM1 OUT) is the printhead fire signal. Timer 1 sends a fire pulse after a fixed delay of $t_p = 10.3\,\mu s$. This allows the firmware a short processing window: The firmware gets an interrupt signal when Timer 1 is triggered, and can decide, using its internal state and the current position value (TIM3 CNT) tracked by Timer 3, if the

[a]Input filter: A signal level change must be stable for n samples to not be discarded, here n was set to 32.

printhead should be fired. If it decides not to fire, it can abort Timer 1, and the fire pulse will not be sent. Otherwise, the fire pulse is sent after the fixed delay.

Due to the fixed delay, an ideal motion stage that travels with a constant speed, has perfect repeatability, while providing a small time window for computations between the encoder pulse and the fire signal. Computations were required because in practice, some jitter on the encoder position was found. Presumably they occur when encoder and strip are relatively moved by vibrations in the motion stage due to the movement and the stepper motors. This could create two edge cases:

1. Encoder jumped back: The vibrations caused the position value to decrement, and Timer 1 is now triggered to fire for a position that has already been printed.

2. Encoder skipping: The encoder moved over to the next position so quickly that processing/printing the position (or even multiple positions) before has not yet completed.

The first case can be simply mitigated by detecting the condition, and aborting the fire pulse, once Timer 1 has been triggered with a past encoder position. After some time, the next encoder position to print will occur again, and printing is resumed. The second case is a little more complicated: The fire pulse at the current position must be aborted as well because the nozzle data (which nozzle should eject at this encoder position) in the printhead does not match to what is to be printed at this position. This lost print cycle is rather used to load the data for the next position, so that from there on printing can continue normally. However, all positions that were skipped are lost.

In practice the second case happens rarely. This is because moving forward is the default behavior - the encoder sensor has to scan *two* lines in a short time for it to count as skipping.

2.5 Print system software

The print system software is part of the control software, and runs on a host PC. Its main tasks are:

- Creating print jobs: This includes importing CAD files of models to print, placing them on the virtual build plate, and adjusting the print settings, according to which the model's layers are rasterized. Thorough examination is possible by simulation.

- Executing and monitoring the printing process: The host PC communicates with the printhead drive electronics and the motion stage controller to perform the printing. It orchestrates the jetting and movements next to printbed temperature, fans and pumps. Relevant variables are displayed to the user next to pictures from the substrate microscope camera allowing process monitoring.

- Inspection of recorded images after printing.

- An entire screen is dedicated to performing different printhead maintenance jobs like loading ink or flushing/purging operations.

- Optimization: The jetting is recorded and can be analyzed by using the dropwatcher function, or the drop size and nozzle reliability can be inspected using the substrate microscope. Different automated parameter sweeps simplify ink/waveform characterization.

In the following subsections *creating print jobs* and *executing and monitoring the printing process* are explained further, as printhead maintenance is already described in Section 2.3, and the dropwatcher functionality has its own Section 2.6. Specific challenges that had to be solved will be highlighted. Finally, the section concludes with an overview of the software stack and architecture used.

2.5.1 Print job creation

Each print job starts with the model. Like when using a conventional 3D printer, models are designed using any CAD software. The solid bodies are cut into separate slices parallel to the build plate in a process called slicing. For this, the open source software Slic3r[a] is used with the following command in its command line interface (CLI) mode that will create an svg-file[b] containing the geometry information of the separate layers.

```
Slic3r-console.exe --layer-height 0.1 --export-svg
  .\input-file.stl -o output-file.svg --first-layer-height 0.1
```

The number of layers is adjusted by setting the –layer-height and –first-layer-height parameters. If the CAD model has a height of 1 mm, setting both values to 0.1 will cut it into 10 layers. To streamline this process in the future, libslic3er, the core of this software package could be directly integrated into this project to perform the slicing.

[a] Slic3r: `https://slic3r.org/`

[b] svg: scalable vector graphics, an image format containing vectory geometry rather than pixel graphics

The resulting svg-file is then imported by drag and drop on the virtual build plate in the "Print" tab, displayed in Fig. 2.16 Models imported this way can be positioned on the build plate by dragging them around.

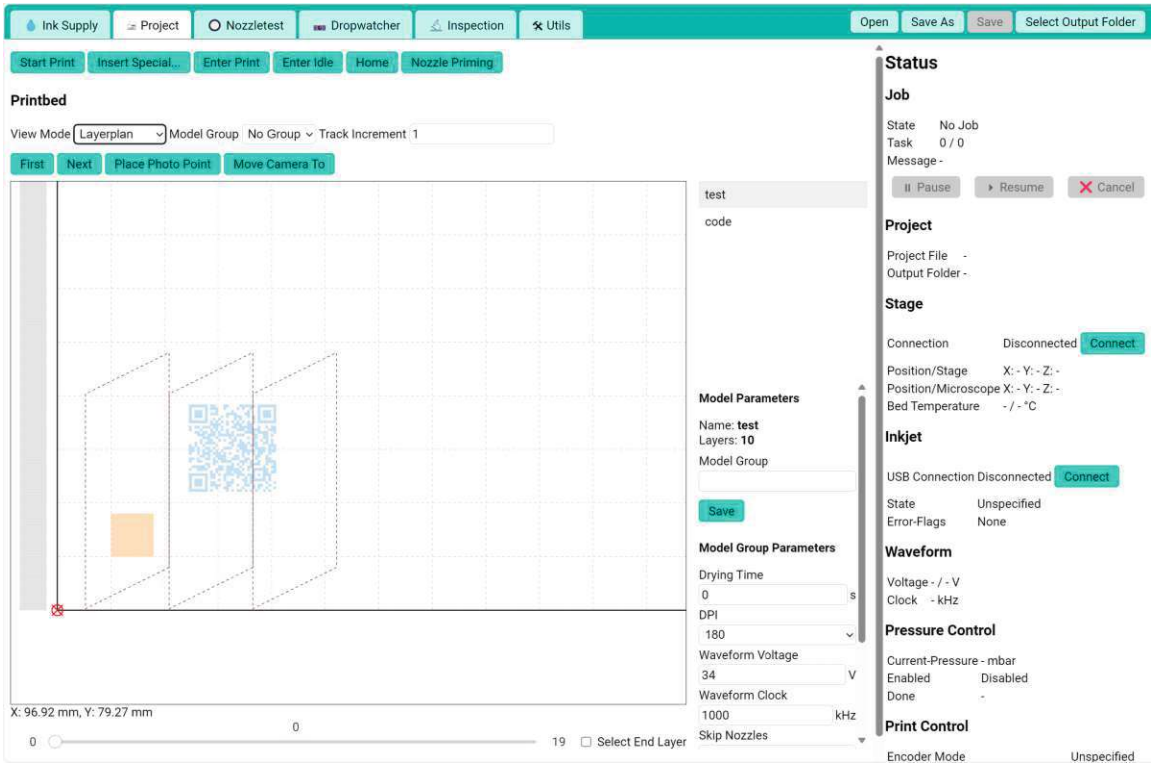


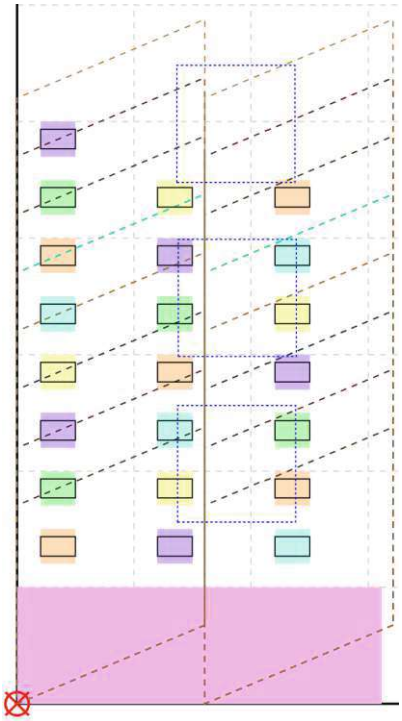
Figure 2.16: Print tab: An orange square and a blue QR code model placed on the build plate of the developed user interface.

Print settings Apart from the position the most important adjustable print settings are:

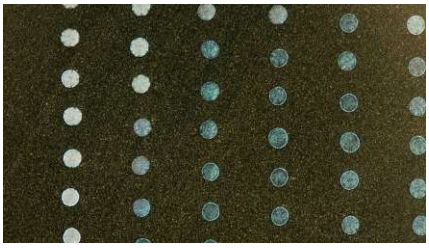
- DPI: Sets the droplet spacing in the print axis. The available options 720, 360, 240, 180, 120, 90 correspond to the signals of the linear encoder, 720 meaning drops are ejected at every possible position, 360 at every second, 240 at every third and so on. This setting should be adjusted according to droplet size and desired droplet overlap.
- Skip nozzles: Restrict jetting to every first/second/third/etc. nozzle. This is mainly for ink characterization purposes, to space the droplets further apart and be able to inspect them individually.
- Heatbed temperature
- Drying time: Time (in seconds) to wait after printing one layer before printing the next one onto the previous one.
- First layer height: The distance at which the first layer is printed, needs to be roughly adjusted to the substrate thickness.
- Waveform voltage and clock: Adjust the width and amplitude of the firing pulse.

Photo points Multiple photo points can be selected using the "place photo point" button. The substrate camera will save a photo at this point after each layer was printed. This allows to inspect the layer growth, which is useful for example to investigate why specific structures emerge.

Model groups and parameter sweeps Models can also be assigned to groups. A useful feature for prototyping is that in a group, some of the settings above can be overridden: Different DPI, drying time and waveform parameter settings can be tested in a single print job this way. This is especially useful for performing parameter sweeps. For example, to characterize a new ink, models with different waveform settings can be placed automatically using the "Insert Special" menu on the top left of Fig. 2.16 creating an individual group for each model. It inserts single layer rectangular models with the size of the viewport of the substrate camera, setting different waveform parameters and adding photo points to each model as demonstrated in Fig. 2.17a. The drops are set to be spaced further apart, so that after printing has finished, the influence of the waveform on the nature of the deposited droplets can be investigated in the recorded images, an example is shown in Fig. 2.17b.



(a)



(b)

Figure 2.17: Auto-generated waveform test patterns, the small black rectangles indicate the photo points (a), one of the photo points recorded after printing (b).

The prepared print jobs can be saved to a file, which is helpful to repeat a previously performed experiment or to revisit the printing parameters after evaluating the printed sample.

2.5.2 Print planner and track rasterizer

After the print job is designed by the user, it needs to be converted into a series of instructions the machine can execute. The imported models that were obtained from the slicing step using Slic3r described earlier, are present as vector graphics. Specifically each layer consists of one or multiple disjoint polygons, the contours of the layers objects. Each of the contours can have additional sub-polygons, "holes", inside. To print the layers, the geometries have to be converted to patterns of dots, the positions at which the printhead needs to place the droplets.

Two components were implemented to accomplish this task: A print planner and a rasterizer. As outlined in Section 2.4.1, printing of a layer is carried out in tracks, with the printhead moving in adjacent, parallel paths. The width of each track corresponds to the nozzle line (perpendicular to the track, respecting the mounting angle of the printhead), and as further detailed in Section 2.4.2, at each encoder position droplet ejection of any of the

nozzles can be triggered. The print planner calculates a series of layer plans, each consisting of the individual tracks movements the printhead needs to perform, defined by the move axis' (x-axis) position and starting and ending defined by the print axis' (y-axis) position. The rasterizer then calculates the lines to print for each of the tracks. Lines correspond to the positions of the encoder, each line contains the information, which nozzle should eject at this position, and which should not.

Print planner The algorithm of the print planner is simple: For a given layer it fits a minimum bounding box around all the polygons together in the layer. Starting with the bottom left corner, adjacent tracks are added until the rectangular bounding box is completely filled. The angle of the printhead has to be considered, meaning that the beginning of the track is the encoder position at which the foremost nozzle just entered the bounding box. A distance of 2 mm is added at the start and the end of a track if possible, for acceleration and motion stabilization.

While evaluating the nature of placed droplets with different ink compositions and waveform settings, it was discovered that not only blocked nozzles that do not eject at all, are a problem, but working nozzles do not perform equally. Some were unstable, showing some missing jets in patterns, also a variation in drop size and particle loading was observed. Reasons for this may be unequal distribution and local changes of ink properties in the printhead, particle agglomeration or air bubble formation in the vicinity of a nozzle [38].

To reduce inhomogeneity of the thin films caused by uneven nozzle performance, an additional strategy was implemented: Instead of fitting the tracks to the bounding box by starting with the leftmost nozzle covering the leftmost areas of the object, the first track of each layer is additionally shifted to the left by a random amount between 0 mm and the width of the track. This moves effectively all tracks to the left by a random amount, adding another track at the end if necessary to cover the whole area. This way, the same nozzle passes over different positions of the object in different layers. It is expected that by uniform randomization, the unevenness will level out when printing a reasonable amount of layers. To keep the result deterministic, a pseudo random generator with a seed number is used, which will always produce the same layer plans.

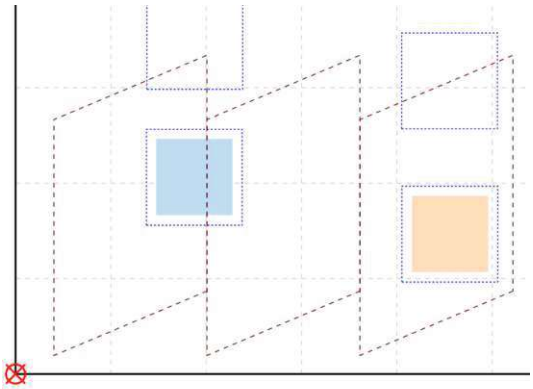


Figure 2.18: Print plan of a layer of two 8 mm x 8 mm squares (blue and orange). Tracks are shown with red dashed outlines. The blue dotted outlined suqares mark substrate holder positions.

Fig. 2.18 shows the print plan of the first layer of two 8 mm x 8 mm squares (blue and orange). The areas displayed as red dashed lines are the separate tracks. The blue dotted square outline marks specific printing positions where sample holders are placed on the build plate to mount substrates.

The tracks of objects that are in the same print parameter group (see Section 2.5.1) are planned together. This can lead to empty tracks, if objects are spaced further apart. To

prevent this, objects can be assigned to different groups, as each group gets an individual layer plan.

Track rasterizer The input to the track rasterizer is the definition for one track that the print planner determined beforehand, along with the model geometries of this layer and the print settings. Its job is to generate the precise drop placement, nozzle ejection information for every encoder position along the track. The nozzle data generated is in the form of 4 bytes for every encoder position, which corresponds to 128 bits, where each bit indicates whether a nozzle should eject at this position or not.

One way to generate this information that was implemented first, works as follows: The printhead is moved virtually along the track, and for every encoder position, the positions of all the nozzles (according to printhead angle) are calculated. Here, the DPI and skip nozzles print settings are considered. Then, for every nozzle position, and every polygon in the layer, a complete point in polygon test is performed. The point in polygon test is typically implemented using a ray casting algorithm [41], where from the point a ray in a fixed, arbitrary direction is casted. If the number of intersections with the polygon lines is odd, the point is inside, otherwise it is outside. Even though this process is sped up by using rectangular bounding boxes around the polygons, it is rather slow.

Later on, a more efficient method was implemented using another common computer graphics technique for polygon rasterization involving a scan line and an active edge list [41].

In this algorithm the scanline moves along the polygon, e.g. from bottom to top, in steps that correspond to the resolution in this direction. Every time a new edge of the polygon intersects with the scanline, it is added to the active edge list including its intersection point. For every step of the scanline the active edge list is checked: The part of the scanline from the first intersection point to the second is inside, from the second to the third is outside, from the third to the fourth is inside and so on. When the scanline advances, the active edge list must be updated. Edges that terminated have to be removed, and the intersection points have to be updated to match the next scanline. To do this efficiently the active edge list also stores a vector for each active edge that can be added to the previous intersection point to get to the next.

Contrary to the standard implementation as found in [41], in this case the nozzle line (the line between nozzle 1 and 128) was used directly as the scanline, which is, like the printhead is mounted, at an angle. This way the resulting spans that determine which parts of the line are inside, can directly be used to calculate which nozzles should activate. Fig. 2.19 illustrates the process: The parallel angled lines show the scanlines (only 3 are shown here), and the red circles their intersections. The areas of the scanlines colored in blue are the

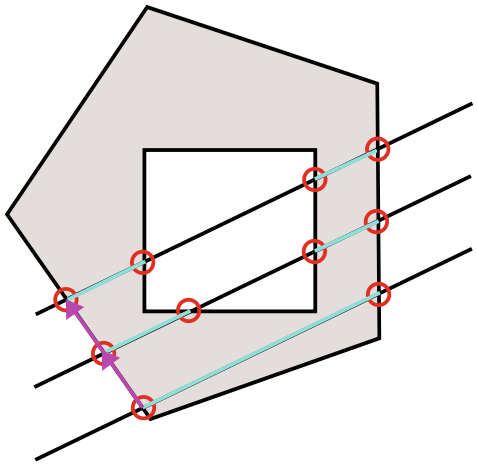


Figure 2.19: Illustration of the implemented scanline rasterization algorithm, showing the scanlines at the angle of the printhead.

regions considered inside, and the fuchsia colored vector shows, how the next intersection can be efficiently calculated from the previous one.

A special feature of the track rasterizer is the blocked nozzle correction. In the software, the user can configure which nozzles have problems. This is detected by nozzle checks, which are further discussed in Section 3.2. When the rasterization of a track has finished, and nozzles were used that are marked as blocked, the track rasterizer can suggest additional tracks to correct for the missing jets. These tracks are moved slightly to a different move axis (x-axis) position, so that different nozzles can cover the area, where blocked ones would have been used. The algorithm tries to optimize the movements, so that as many blocked nozzles are covered with as few passes as possible.

Simulation view The result of the track rasterization can be directly inspected on the virtual build pate, if the view is switched to "Rasterization" by using the drop-down in the top left shown in Fig. 2.16. In this mode the droplet placement is directly shown on the objects, each droplet being represented by a circle. Fig. 2.20 shows the simulation of the rasterization of a TU Wien logo model.

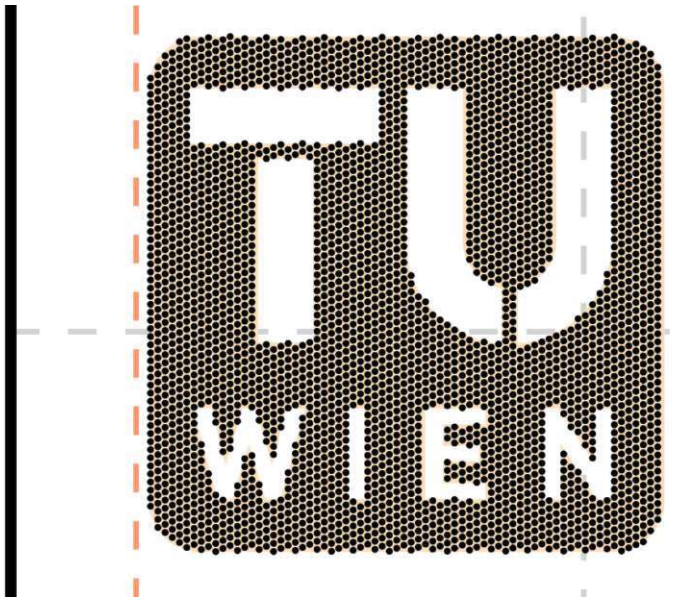


Figure 2.20: Rasterization of a TU Wien Logo.

This is especially useful when setting the print settings, to get a preview and directly see the influence of the parameters. At the same time it is very useful during the development process: The output of the track rasterizer is, as explained before, a byte array. For printing a track, this byte array is written to the printhead drive electronics memory, before activating the encoder print mode and moving the printhead. The simulation on the other hand also uses the same byte array. Therefore, the simulation can directly be used as a verification tool while developing and improving the rasterization algorithm.

Below the virtual build plate, the user can select which layers are currently shown. By activating the "Select End Layer" checkbox, a range can be selected, which is useful to inspect the droplet overlap across layers. In this mode, each droplet is shown at an opacity level that allows to estimate the homogeneity of the printed thin films in advance, as depicted in Fig. 2.21. In each panel one can see as many layers on top of each other as labeled. With 10 layers one can see that the pattern is already blurring, and more layers could lead to a more even distribution.

(a) Layer 1 (b) Layer 1-2 (c) Layer 1-3 (d) Layer 1-6 (e) Layer 1-10

Figure 2.21: Simulation displaying rasterization of multiple layers on top of each other. After 10 layers, the drops of the different layers already overlap significantly.

This also shows the effect of the randomly positioned tracks the print planner calculates. Contrary to intuition, the droplets do not just overlap in move axis (x-axis) direction, which is the one subjected to the random offsets, but also an overlap in print direction can be seen. The reason for this positive effect is the angle of the printhead, which results in a specific pitch of the nozzles in print direction. This y-pitch of the nozzles, however, does not coincide with the pitch of the encoder. Depicting this is the following example. The current position of the encoder is associated with the first/lowest nozzle, and the printhead prints a specific pattern. Now, if for the next layer, the printhead is moved exactly one nozzle pitch in x direction to the left, as sketched by the orange dots in Fig. 2.22, the second nozzle is now at the x position of the first nozzle. However, the dot the second nozzle prints is shifted up by one y-nozzle pitch as well. Since this pitch does not coincide with one encoder position pitch, the drop must overlap with the drops in the layer underneath.

Figure 2.22: Drop overlap in print (y) direction, indicated by blue and orange dots, is generated by moving the track in move (x) direction for subsequent layers.

2.5.3 Print job execution

When the user finished the print job design, the detailed instructions are created by the planner and rasterizer, and the machine is prepared by loading ink and placing the sample substrates, and printing can start. A typical print job consists of a start phase and a print phase.

Start phase When the printhead moves in z direction, the ink pressure changes. Therefore, the first task is to turn on the pressure control of the ink supply system. Next, the printbed starts to heat up. Meanwhile the motion stage moves to its home position at the origin point and zeros the encoder there. The printhead moves to a safe position and the start phase completes after the target heatbed temperature is reached.

Print phase The steps that have to be repeated for each layer, illustrated in Fig. 2.23 are explained in the following.

First, the motion stage moves so that the printhead is in a nozzle priming position, and a quick priming cycle (firing all nozzles 100 times using the fastest firing frequency) is performed. Then, each group, consisting of one or more tracks, is printed.

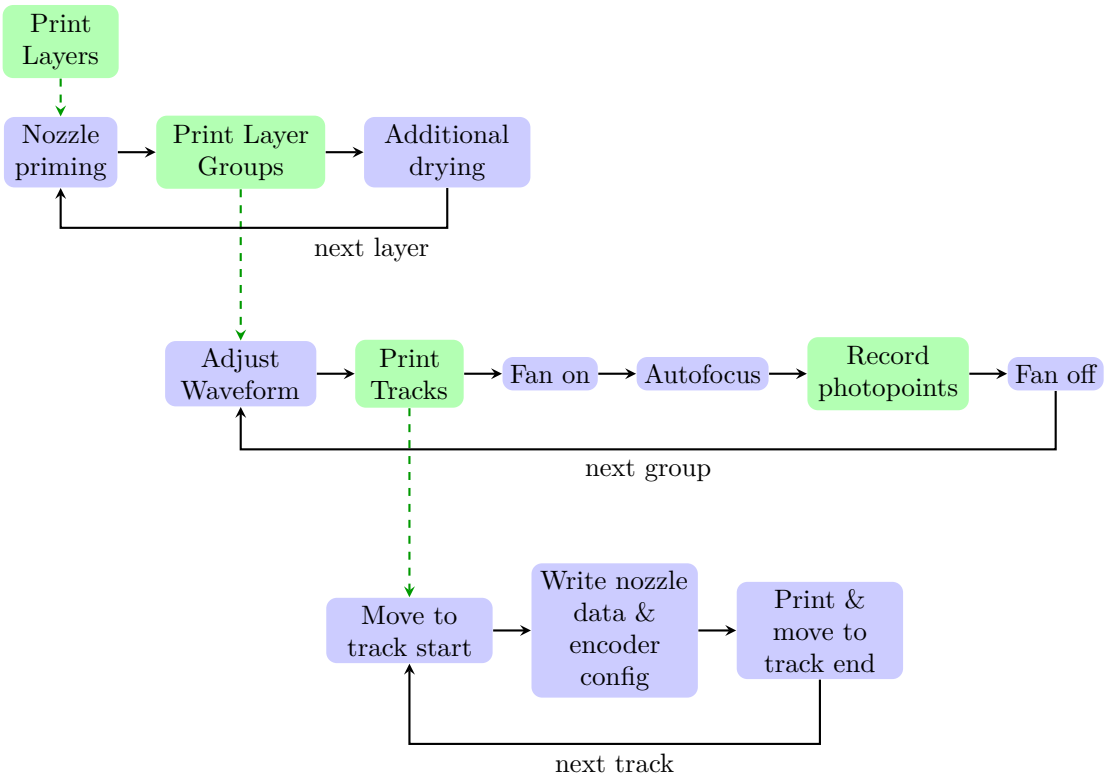


Figure 2.23: Tasks that are executed for each layer.

If necessary, the waveform parameters are adjusted for the print settings of the group. For this, the printhead is shutdown, by activating reset and disconnecting the piezo drive power supply. Then the voltage of the power supply is changed, and the printhead is activated again, after a pause of 3 s to allow for capacitors to adjust.

Then the tracks of the group are printed: For each track, the printhead first moves to the track's starting position. The nozzle data for this track is written to the printhead drive electronics memory, and the encoder printing parameters are set: number of lines to print, starting encoder position, DPI/print interval (how many encoder positions are between positions where the nozzles should fire). The printhead then is moved to the track end, while the droplet ejection is triggered by the printhead drive electronics and the nozzle data is written to the printhead in between ejections, as explained in Section 2.4.2. The ink pump pauses, while the track is printed to minimize any disturbances in ink pressure. After the track is printed the encoder print mode is set to paused, so it can move to the start of the next track.

When all tracks of a group are printed, the substrate microscope camera is used to record images at the predefined photo points. It moves to the specified location and performs an autofocus procedure using the z-stage.

The autofocus is implemented by moving the camera up and down in a 2 mm range, while taking images at the different positions. The focus is evaluated using a contrast based autofocus algorithm [42] [43]: The camera takes pictures at different distances, and evaluates a focus function for each image. On the grayscale converted image, a median blur is performed to remove noise, after which an edge detection using a Laplacian operator is executed. The

resulting edge image having the highest variance is considered the be-in-focus, as it has the most edges. The camera moves to this distance to record the images, which are saved in a folder the user chooses before starting the print. While recording the photos, the fan is turned on to speed up the drying process, by moving away solvent-saturated air near the substrate surface.

The timestamp, when the printing of all tracks of a group is finished, is recorded. After all groups are printed, and all the photos are recorded, the program calculates if any of the groups have not yet been dried for the time specified in their print settings. If so, the printhead moves to a safe position, and the fan is turned on again for the remaining drying time. Finally, the process is repeated for the next layer.

Pausing, crashes and advanced homing Print jobs can be paused and resumed. This is helpful, for example if a minor maintenance procedure has to be executed, like wetting the nozzle plate. Also, if insufficient drying is detected by the user, additional drying time can be added by pausing the print job. Automated nozzle tests can be executed after printing a user-selectable number of layers. After a nozzle test finished, the print is paused automatically, for the user to reselect, which nozzles should be used to continue the print. The automated nozzle test is discussed in more detail in Section 3.2.

In a software under development, especially with this many components, crashes that require restart must be anticipated. When starting a print, the user can select a specific layer number to start at. One problem that arose during testing was that the motion stage lost its absolute position when its serial port connection had been re-opened, which required re-homing. The limit switches used for homing do not provide a high repeatability.

To enhance the homing accuracy, a visual target of three tiny, dark dots is fixed to the build plate. After rudimentary homing using the limit switches, the three dots are recognized using the build plate microscope camera and computer vision algorithms: A contour finding function is executed on a binary thresholded image that has been blurred using a Gaussian blur. When fitting circles to the contours, unlikely-sized candidates are rejected. The build plate is then moved in tiny steps until the center between the three dots aligns with the center of the microscope image. Using this process it is possible to restart failed prints even after restarting system components or the host PC.

2.5.4 Software stack and architecture

On first glance the decision for the platform the software is built on and which technology is used might seem unconventional. Therefore, these choices are analyzed briefly in this section.

Web platform It is not uncommon to use websites, or rather web apps, to implement the user facing parts of software: Octoprint[a] is a popular open source web interface that allows to send print jobs to 3D printers and monitor them. Similarly for printers using Klipper firmware[b], several web interfaces are available.

The advantages of using the web platform include remote control, by opening the website on another computer, and a broad range of frameworks, libraries and tools being available, to speed up development. Web applications only require a browser to run, and are therefore inherently cross-platform compatible, often including mobile devices. The lack of an installation step makes accessing the application easier and offers convenience. Creating user interfaces is especially well supported, since it is a primary task of web applications [44] [45].

Usually, a web application is split into client and server parts. In case of 3D printing control software mentioned above, the user interface runs on the client part and provides

[a] Octoprint: `https://www.octoprint.org/`
[b] Klipper: `https://www.klipper3d.org/`

functions to plan print jobs and monitor printing. The server part executes the print jobs and its separate tasks, communicating with the firmware of the printer, for example via a serial port. To empower web applications even more, special application programming interfaces (APIs) have been added to browsers that allow communication to USB- or serial-port devices.

The print system software of this printer uses these APIs to communicate with the hardware, making it a client-only application. This way, no server had to be implemented for this prototype, which sped up development further. A nice side effect is that for running the printer, the only requirements for the host PC are an up-to-date (chromium based[a]) web browser and a USB port, and absolutely no setup is required. However, the application was designed with the introduction of a server in mind: Operations that would likely run on a server in the future are sufficiently abstracted so that eventual migration is simplified.

The main programming language used for the host PC software is TypeScript, a superset of the web platform's main language JavaScript that adds type safety on top. Webpack is used as a build tool, comparable to a compiler in native development. It bundles the separate modules that make up the application together, and converts the TypeScript code into browser executable JavaScript. A very effective development feature used is hot module reloading: With it, single parts of the application can be reloaded, immediately after the code is changed, without completely closing and restarting the application. In other words, parts of the program can be changed while running. This saves a lot of time, and additionally the hardware devices, motion stage, printhead drive electronics, stay connected to the application, and connections are not reset. For example, while parts of a computer vision algorithm are optimized, the motion stage and camera stay connected and changes can be tested immediately.

The user interface (UI) application uses a reactive state management architecture. This type of software architecture is encouraged by some frameworks, for example React[b], a similar example is the Flux architecture[c] coined by Facebook. The common denominator is that the state of the application is separated from the UI components, and state changes need to be performed via a defined interface. In this project a single instance, called "Store" holds the state of the application. Changes to the state are requested by components by posting "Actions" to the Store. A change is for example when the user moves an object on the build plate to a new position. UI components subscribe for updates to the Store, and change their appearance as soon as they get informed. For example, the virtual build plate renders the object at its new position. Apart from organizing the responsibilities of different components of the application very well, this architecture allows hot module reloading in the first place: While components reload, the state still lives in the Store, and upon code change, the changed component is initialized in the current state. This decoupling also provides a good point to move some parts of the state to a server in the future, without changing the components.

The communication to the motion stage works via a serial port. An abstraction layer was implemented providing specific movement, heating and fan functions to the application components that internally sends G-code commands to the Marlin firmware running on the motion stage controller. WebUSB[d] was chosen as the communication method to the printhead drive electronics. While USB is more complicated to set up than a serial port interface, it has some advantages: It provides reliable communication using re-transmissions, supports a higher data rate, and the host software can identify the device automatically using descriptors (in contrast to serial ports, where the correct port, COM port on windows, must be selected by the user) [46]. Furthermore, WebUSB can be used without the need to install a device driver (in contrast to standard USB). Zephyr project, which was used to develop the firmware also provides example code for WebUSB communication.

[a] The WebUSB and serial port APIs are for now only available in chromium based browsers like Edge, Chromium, Chrome, Opera, etc.

[b] React web application framework: `https://react.dev/`

[c] Flux architecture: `https://facebookarchive.github.io/flux/`

[d] WebUSB: `https://wicg.github.io/webusb/`

The USB communiation works in packets of bytes, so a protocol is needed to give meaning to the messages sent between host PC software and printhead drive electronics. Protobuf[a] is a library that can be used to define a language-neutral protocol in a simple way. The messages that are exchanged are defined in a special definition language. An example for a message would be the request to activate the ink pump and control the ink pressure to a specified value. The protobuf compiler can then generate typescript code and C code from the definition, for the host PC software and the firmware, respectively, to convert the message to a byte array that can be sent over the wire, and back. This way, it is easy to add new messages or extend existing ones on both sides of the communication.

A popular library used for image processing and computer vision is OpenCV[47]. There is a version available that can be used in browsers, which is possible by compiling the OpenCV C++ code to WebAssembly. It is used in this project for computer vision related tasks like the autofocus of the cameras, detection of the calibration target on the build plate as well as automated droplet size measurement.

[a]Protobuf: `https://protobuf.dev/`

2.6 Dropwatcher

A dropwatcher allows to analyze the jetting process of a printhead in detail at the single drop level. Its main purpose is to optimize the waveform for a new ink to be used with a printhead. Its components are a camera and a strobe light (flash). A camera equipped with a microscope objective typically needs a bright light source to illuminate the sample and displays a dark image under ambient light conditions. The camera is pointed at the nozzle plate of the printhead and is configured to continuously take pictures (frames of a video). The printhead is fired after the exposure of the image sensor for a frame started. The flash is mounted at the opposite site of the printhead, pointing directly into the camera. Exact alignment of flash and camera was found not to be required. Right when the droplet is ejected, the strobe light turns on for a short time. Only in this short timeframe, light hits the camera's image sensor to produce an image. Some light is inhibited by the drop, which allows to photograph a still image of it. The time between frames (exposure time) must be sufficiently long, so that the exposure of the sensor with the flash completed and the flash is turned off, before the next frame starts.

By varying the time delay between the print head firing signal and the flash being switched on, images of the droplet can be captured at different times of its formation and trajectory. Stitching together images with incremental delay times, even a video of the entire lifetime of the droplet can be created. This is useful to diagnose misfiring jets, analyze droplet speed and the formation of satellite droplets [48].

The requirements of a dropwatcher are therefore:

- Precise timing of jetting signal and the strobe
- The light source must be able to deliver sufficient light in the short time.
- Synchronization of camera and jetting: The printhead must be fired inside the timeframe of exposure.
- The magnification and resolution of the microscope must be able to image individual drops.

A simple USB microscope, TOOLCRAFT USB Mikroskop 2, is used as a camera device. It was chosen because it allows to set the exposure time via the host PC. A generic white 10 W SMD led module functions as strobe light, powered by a 12 V 500 mA power supply. The led light and the power supply are connected to the printhead drive electronics, as described in Section 2.2.3.

To use the dropwatcher, the user selects the "Dropwatcher" tab in the print system software as shown in Fig. 2.16 in Section 2.5.1. There, the camera stream of the webcam opens, displaying a video stream. By using the integrated led light of the webcam, its position can be adjusted, either manually, or by moving the motion stage, were it is attached to, using arrow keys. Afterwards, this LED light is turned off. Then, the printhead drive electronics is put into "Dropwatcher" mode.

Unfortunately, the camera shutter signal is not transmitted to the host PC. However, the software can detect, when a new frame arrived from the video stream. The exposure time is increased sufficiently, lowering the frames per second. If the delay between shutter and frame, transferred via USB and rendered to the screen, is short in comparison to the frame-to-frame time, the arrival of a new frame event can be substituted for the shutter signal. Each time a new frame arrives, the next drop ejection is requested, as it is assumed that the exposure window has just begun. If the exposure time is chosen too short, the (rolling) shutter might have already started reading out pixel values from parts of the image before the drop ejected. Since the light turns on mid readout in this case, half lit images are generated. Tests showed that the exposure time needs to be at least 125 ms to capture stable images.

Precise synchronization of the jetting and the light is accomplished again by using hardware timers of the STM32 MCU, similar to printing using the encoder signal, as explained in Section 2.4.2. Timer 1 has, next to the printhead fire signal, a second output, which is connected to an n-channel MOSFET to switch the light. Both of these outputs are configured in "combined PWM mode", which basically means that for each output, a specific pulse can be configured by setting two values, corresponding to the start time and the end time of the pulse.

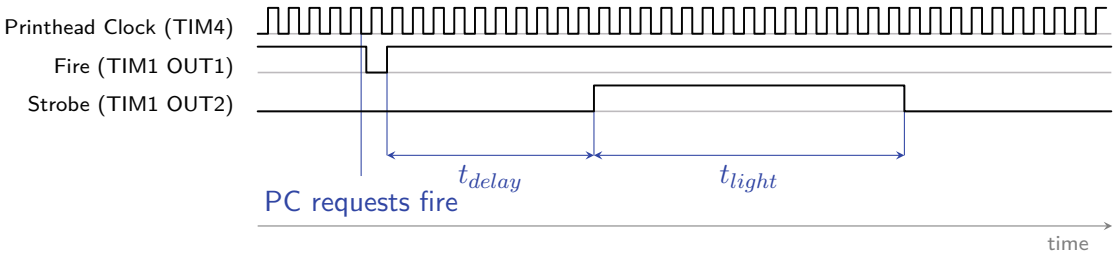


Figure 2.24: Jetting and strobe synchronization signals over time. The printhead is requested to fire, and after a delay t_{delay} the strobe light is turned on for the duration of t_{light}.

Fig. 2.24 shows the timer signals for imaging a single nozzle fire cycle. This time, Timer 4, which provides the printhead with its main clock signal, is set as the primary timer, and Timer 1 as the subordinate that is started automatically at a falling edge of Timer 4 and stopped after a single iteration. When the host PC requests jetting after recording a frame has started, Timer 1 is activated. It starts to run on the next falling edge of Timer 4, and pulls the fire pin low for exactly the next rising and falling edge of the printhead clock. This makes sure that the delay between setting the fire pin low, and the printhead sampling the signal is always constant. After the user-configurable time t_{delay}, the light is turned on by switching the MOSFET. It is turned off again after the configurable delay t_{light}.

To record the droplets at their different stages t_{delay} is varied between 0 and 400 µs. An on-time of the light t_{light} of 3 to 4 µs resulted in a good illumination.

The precise timing works well, and the printhead's drop generation is very repeatable, so that the video stream resembles a still looking image of droplets standing in mid air. An example frame is shown in Fig. 2.25. As explained in Section 2.1.2, the shared wall design of the Xaar printheads required to fire the nozzles in three groups, which can be observed by the three different stages of the droplets in the image. This way, it would even be possible to get an estimate for drop velocity, using the distance between two drops photographed with different delay settings.

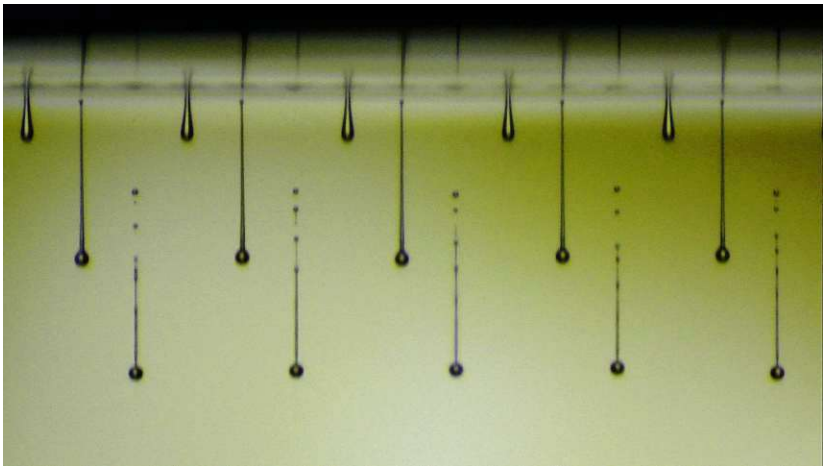


Figure 2.25: Dropwatcher image of the Xaar Irix (40 pl, wetting nozzle plate) jetting, t_{light} = 2 µs, t_{delay} = 106 µs. The firing cycle split into three groups is clearly visible.

Chapter 3

Application

After the developed inkjet printer was introduced in the previous chapter along with insight into its design process, its application to the task of printing electrolyte layers is described in the next one. Several samples were printed with different inks and print settings, and characterized. First the developed ink compositions that were used to print the samples, and their preparation is explained. This includes additional experimental results like viscosity measurements, drying behaviour and the influence of the waveform parameters on the printing. The printing process of the samples and the adjustments made to the ink, printing parameters, and the procedure itself, between each print run, are then discussed. The results of different measurements to assess the morphology and to analyze electrochemical properties of the printed layers are presented in the final part of this chapter.

3.1 Ink

In this section the ink development process is presented. First the final compositions used to print the samples are given, and the process of ink preparation is explained. Results and findings of additional tests that led to those final compositions are discussed along with challenges found while printing with the inks. Finally the influence of the waveform is analyzed using recordings made with the dropwatcher.

Table 3.1: Ink compositions used to print the samples.

Ink	YSZ	Ethanol	Terpineol	BYK-399	BYK-106
A	1 wt.%	24.75 wt.%	74.25 wt.%	0 wt.%	0 wt.%
B	2 wt.%	24.50 wt.%	73.50 wt.%	0 wt.%	0 wt.%
C	5 wt.%	36.80 wt.%	55.20 wt.%	2 wt.%	1 wt.%

8 mol% Y_2O_3-stabilized zirconia powder (TZ-8YSB, TOSOH Corporation) was used as received in ink A and B. Ink C was prepared by ball-milling for 6 h in ethanol using YSZ balls, which were dried afterwards. After the YSZ powder was mixed with the ethanol and terpineol (α-terpineol, 97 %, Thermo Scientific Chemicals) solvents, and for ink C also with the dispersant additive BYK-106 (BYK), a 1 hour long sonication was carried out in an ultrasonic bath. The surfactant additive BYK-399 (BYK) was added after sonication to ink C. If the ink was stored for more than one day, it was again sonicated for 30 min before printing followed by at least 30 min of settling time.

2.5 ml are sufficient to fill the tubings of the printhead completely. Before filling the printer with this ink, it was filtered with a Millex-SV 5 µm, PVDF, 25 mm filter, which is the same type that is installed in the ink path. Without the settling time the filter clogged very quickly, so that it was impossible to press the 2.5 ml of ink through it, indicating that larger particle agglomerates are still present and need to sink to the bottom first.

Solvent composition When jetting pure ethanol many satellite droplets were observed, as shown in Fig. 3.1. Also, it was found that air/vapor bubbles entering the printhead, could lead to missing jets. Suspecting that volatile ethanol could be evaporating out of the nozzles, a mixture of ethanol and terpineol, with latter having a higher boiling point of 214 -224 °C, was tested.

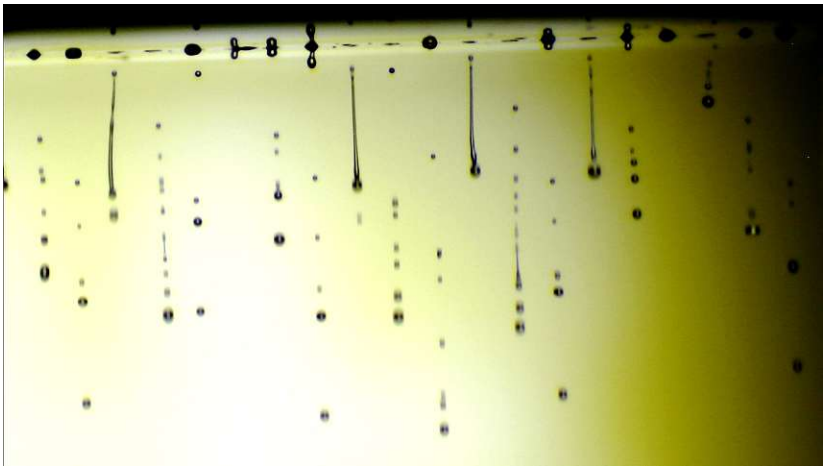


Figure 3.1: Jetting pure ethanol showing satellite droplets.

Viscosity measurements of terpineol/ethanol mixtures were performed with a MCR300 rheometer (Anton Paar) at 25 °C, the results are plotted in Fig. 3.2. In [17], the viscosity range for the Xaar Irix is given as 6-20 mPa s. Using the viscosity measurements as a guideline, it was decided to use inks with ethanol content of 25-40 wt.% to be in the appropriate range.

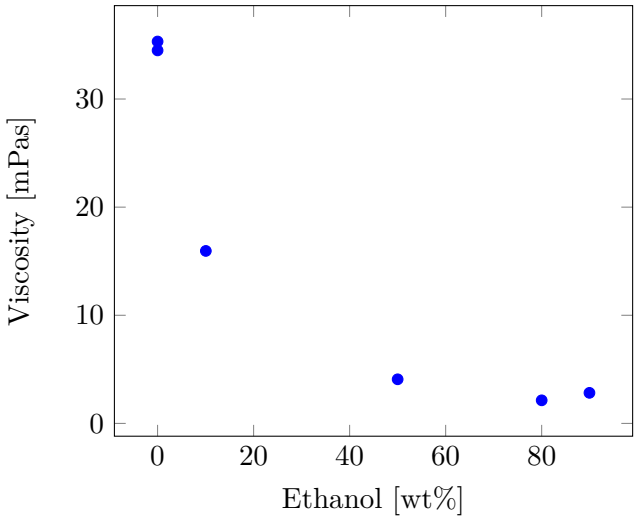


Figure 3.2: Viscosity measurements of ethanol/terpineol mixtures as a function of ethanol content.

Drying A disadvantage of the increased terpineol content could be observed by more drying time required between printing the layers. If a previous layer does not dry completely, an uneven particle distribution pattern starts to emerge as more layers are built up, as can be seen in Fig. 3.3.

Figure 3.3: Uneven pattern on the surface when drying between layers is too short.

A possible explanation could be residual solvent that, if not dried completely, builds up over time in specific areas. With the heatbed set to 90 °C, the drying time could be reduced to 35 s to prevent this effect.

3.1.1 Influence of the waveform

As mentioned in Section 2.1, the Xaar 128 (and Irix) generate the jetting waveform internally, but it is possible to influence it by varying the supply voltage and the clock signal, as can be viewed in Fig. 3.4 and Fig. 3.5.

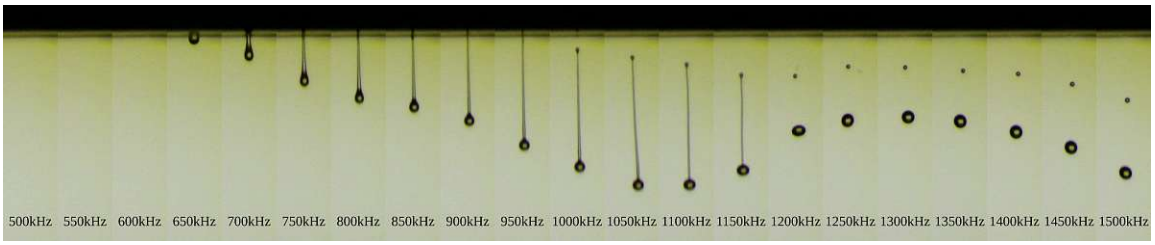


Figure 3.4: Clock sweep using the Xaar Irix Core 40 pl printhead and a reference ink.

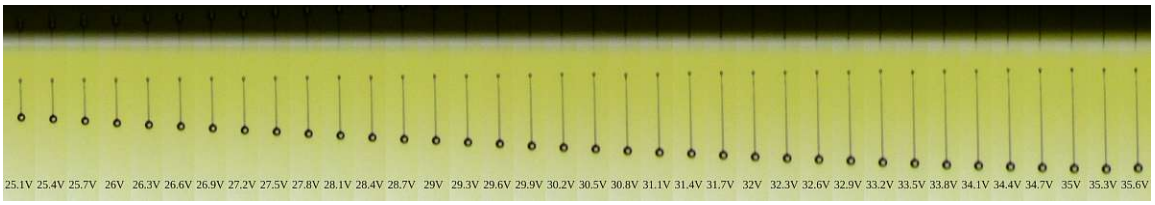


Figure 3.5: Voltage sweep using the Xaar Irix Core 40 pl printhead and a reference ink.

The images are created using the dropwatcher mode of the print system software, by performing parameter sweeps while recording images as it has already been shown in [48]. It automatically steps through different voltages, or clock frequencies, by sending the respective commands to the printhead drive electronics[a]. For every setting, a photo is recorded at the same time delay. Afterwards the stichted image is created by cutting the same rectangular area from every image, displaying a single nozzle, and stichting the image together.[b]

To select the waveform before printing a reference ink was created with the same composition as ink C, but without YSZ particles. The optimization process is adapted from [49] and [50], however it is simplified as only two parameters, voltage and clock can be set here. First, the clock sweep was performed at the highest possible voltage setting. The fastest drop is at 1050 kHz in Fig. 3.4, as all the images are taken at the same delay after the fire command, and it traveled the furthest. Repeating the sweep with smaller steps around this position revealed a maximum at 1080 Hz. At this frequency, the voltage sweep was carried out. Here a maximum was found at 35 V. At higher voltage, the speed did not increase any more, just more satellite drops were visible, therefore these settings were chosen to print with.

[a] Similar to when the waveform is changed during printing (see Section 2.5.3), the printhead is shutdown for a short period during the change.

[b] The stitching functionality is not yet included in the user interface at the time of writing, and needs to be performed afterwards by executing a script.

3.2 Printed samples and printing process optimization

A range of samples were printed, using different inks, print settings and sintering temperature curves. This section gives an overview of the samples, and explains how the printing process was further adjusted using the intermediate results. Table 3.2 lists the samples that are going to be discussed, as well as their major parameters. Samples 4 and 8 were damaged were and discarded.

Table 3.2: Samples & major printing parameters.

Sample	Pattern	Substrate	DPI	Ink	Layers
1	8 mm x 8 mm square	YSZ Single Crystal	144	A	103
2	8 mm x 8 mm square	YSZ Single Crystal	144	A	120
2a	8 mm x 8 mm square	Sapphire Single Crystal	144	A	120
3	8 mm x 8 mm square	YSZ Single Crystal	180	B	200
5	TU Wien Logo	Sapphire Single Crystal	180	C	200
6	8 mm x 8 mm square	Sapphire Single Crystal	180	C	200
7	8 mm x 8 mm square	Sapphire Single Crystal	180	C	200
9	8 mm x 8 mm square	Magnesium Oxide Single Crystal	180	C	200
10	8 mm x 8 mm square	YSZ Single Crystal + Pt Layer	180	C	200

For the first tests, the substrate YSZ was chosen because the ink is more likely to adhere when printed on the same material. Sapphire and Magnesium Oxide were used to perform in-plane conductivity measurements. Substrate of samples 10 was coated with a platinum layer before printing to enable cross plane conductivity measurement.

Printhead Samples 1 to 3 were printed using the Xaar 128 80 pl printhead with a non-wetting coated nozzle plate, while sample 5 to 10 were printed using a Xaar Irix Core 40 pl printhead with a wetting nozzle plate.

Drop spacing The finest drop spacing, in which drops would not overlap/merge in the print axis while printing a single layer was achieved when using 144 DPI. This was chosen as a starting point, because preliminary tests showed uneven distribution of ceramic particles if drops are allowed to run into each other before they dry (see Fig. 3.3 in Section 3.1). After samples 1 and 2 were inspected by SEM, it became clear that the ceramic particle density was not sufficient to form a dense layer. To increase the amount of material printed for sample 3, drop spacing was slightly increased to 180 DPI, leading to a theoretical 25 % of more material deposited with the same amount of layers.

Ink particle load Priting tests were done using 1 wt.% and 5 wt.% particle loaded ink, showing that the 5 wt.% being more difficult to handle. If the ink was loaded in the printhead for three or more hours, particles settled visibly inside valves and tubing, presumably also inside the printhead, requiring disassembly of significant parts of the ink supply system for cleaning. Degradation of the printhead by this could happen. However, the mean time per layer of printing sample 3 was 1.5 minutes, which totals to 5 hours for 200 layers. This average time per layer consists mainly of drying time and the time of the nozzle checks performed every 10 layers. In addition, as explained in Section 3.1, a settling time of 30 minutes was required for the ink to pass through the 5 µm ink filter, raising the suspicion that the actual particle load of the ink consumed from the top of the vessel differed from the original mixture. Similarly to increasing the DPI, after detecting the sparsely printed surface of samples 1 and 2, the particle load was increased to 2 wt.% (ink B) for printing sample 3.

Examples given in the introduction (Section 1) show that other groups were able to produce thicker thin films above 2 µm with a fraction of printed layers. Further effort was

made to optimize the ink, by the use of a dispersant and a surfactant additive, to increase the particle load and decrease precipitation, and the YSZ powder was prepared in a ball mill before adding it into the solution. Samples 5 - 10 were printed using a 5 wt.% ink (ink C) this way. However, the ink loaded in the printhead was again removed from the upper part of the vessel after a settling time of 24 hours in order to avoid larger particle agglomerates.

Bed temperature Even with the highest possible temperature of 90 °C, drying takes several seconds, which is why all samples were printed at this temperature. The actual sample temperature was measured with an infrared thermometer and was significantly lower. The heatbed is a PCB style heater for 3D FDM printers, equipped with one thermistor. However, on the 4 mm glass surface, a microscope slide was attached, which was modified by gluing Kapton tape to reveal a square hole used to hold the sample substrate. Values between 60 °C and 85 °C were measured at sample surface with samples mounted at different positions, showing an uneven heat distribution of the heatbed.

Missing jets Missing jets were one of the biggest problems, and can have various possible reasons:

- Particle sedimentation, agglomeration buildup in the nozzle chamber or foreign particles, large enough to block or impact the jetting of the nozzle [28] [51]. An example of particle sedimentation in the nozzle chambers is shown in Fig. 3.6.
- Local change of ink properties like viscosity or surface tension. This can be caused for example by drying or solvent evaporation in the nozzle [38].
- Air or solvent bubbles in the vicinity of the nozzle [38].

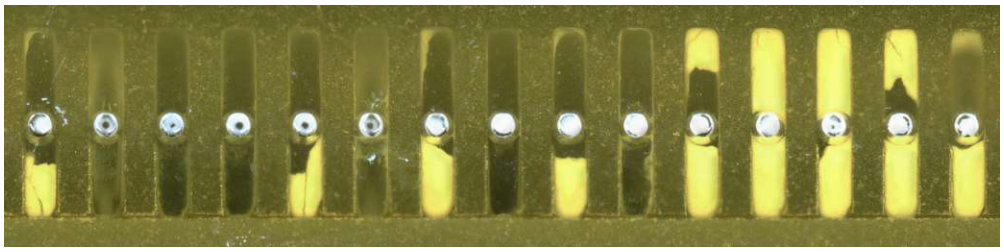


Figure 3.6: Part of the nozzle plate of the Xaar 128, imaged from the bottom, showing the nozzles (circles) and the nozzle chambers. Ceramic particles sedimentation can be seen inside the nozzle chambers.

Aspects that help greatly to reduce the problems are a suitable ink composition that can suspend particles for a long time, and correct printhead maintenance, detailed in Section 2.3. However, especially while the printer and the inks are under development, optimal conditions did not always occur. To be able to print samples in the best possible way despite poor nozzle performance, a missing jet compensation feature was implemented. This can help in different scenarios, one example being decreased printhead performance. Another would be, when an expensive ink is loaded in the printhead, but a large air bubble is detected. In this case the air bubble cannot be flushed without wasting the ink, but with missing jet correction, printing can continue.

The missing jet correction was improved in iterations: The first version was used with sample 1. Before printing the actual sample, a nozzle test pattern was printed onto a transparent film. Each nozzle prints a small rectangle, and a calibration card can be placed

underneath, like shown in Fig. 3.7. The calibration card has corresponding rectangular outlines and the nozzle number. If the square is filled, it means the nozzle is working.

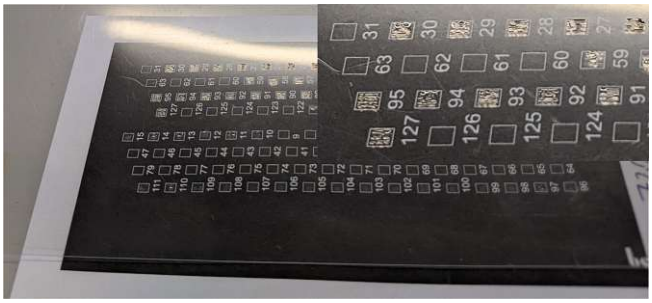


Figure 3.7: Calibration card to detect blocked nozzles.

The numbers of the working nozzles are entered into the software by the user. When the actual sample is printed, the rasterization algorithm detects when a nozzle marked as blocked should be used, and plans additional passes using working nozzles to compensate for the missing jets in the first pass, as shown in Fig. 3.8.

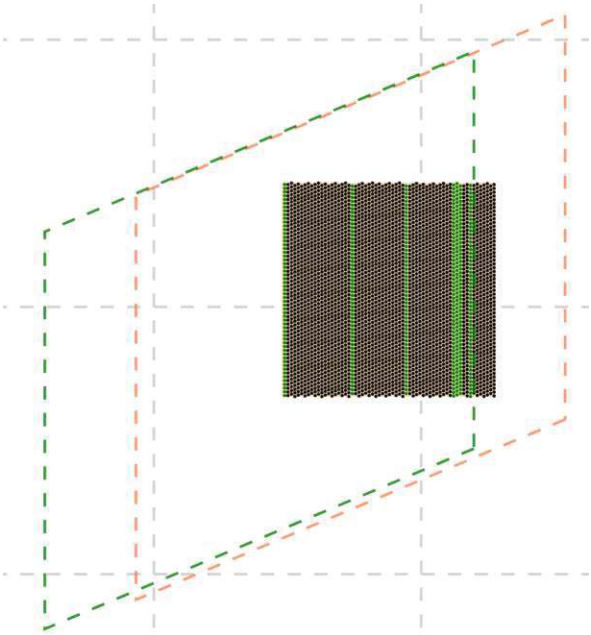


Figure 3.8: Compensation of blocked nozzles: the outlines and dots in green show the correction pass required to compensate for missing jets in the first pass.

An improvement was introduced when printing samples 2, 2.a and 2.b: Remaining unevenness and lines of lesser density along the print axis suggested that either nozzles failed after the nozzle test was conducted, during the print of the sample, or nozzles were performing unevenly. To level out the unevenness the algorithm was extended to select the nozzles that compensate for others from the working set at random.

The disadvantage of this method is that it can still not detect a nozzle failing or recovering during a long print with many layers.

Starting with sample 3, a different method was implemented. Every 10 layers (user selectable), a tiny dot matrix is printed using each nozzle onto a reserved space of the build plate. The system takes images using the build plate microscope camera of each square. Then

the print is paused and the user can select which nozzles should be used to continue the print in the nozzle test screen shown in Fig. 3.9

Figure 3.9: Nozzle test screen of the software: A small calibration grid is printed using each nozzle as shown. The user can select which nozzles should be used to continue the print. Nozzle 2, 4, 7, and 12 show satellite drops, 17-19, 23, 25, and 27 are blocked completely, 21 and 24 are misfiring heavily, and 15, 20, 22, and 26 show missing jets.

While printing sample 3 it was confirmed that the working nozzles actually shifted during printing. Also, it was detected that some nozzles performed unevenly, showing satellite drops, missing, or misplaced jets, which could not have been detected by the first method. Another aspect is the additional drying time during the nozzle test: The sample was on the heatbed without printing on it for longer than the drying time after each layer, which could also have affected the overall layer quality. A disadvantage of this method is that the reserved area on the build plate needs to be cleaned for every nozzle test, requiring an operator to be present.

Sintering profile Sintering profiles of the samples are shown in Fig. 3.10. Most of the samples were sintered for 6 h at 1400 °C, after full densification was not achieved at first sintering tests with samples 2, 2.a and 2.b. Sample 10 was sintered at 1300 °C to prevent damage to the platinum layer. The sintering profile included a step at 400 °C to burn residual organic material from the ink. Another step at 500 °C was added for samples 5 and 6.

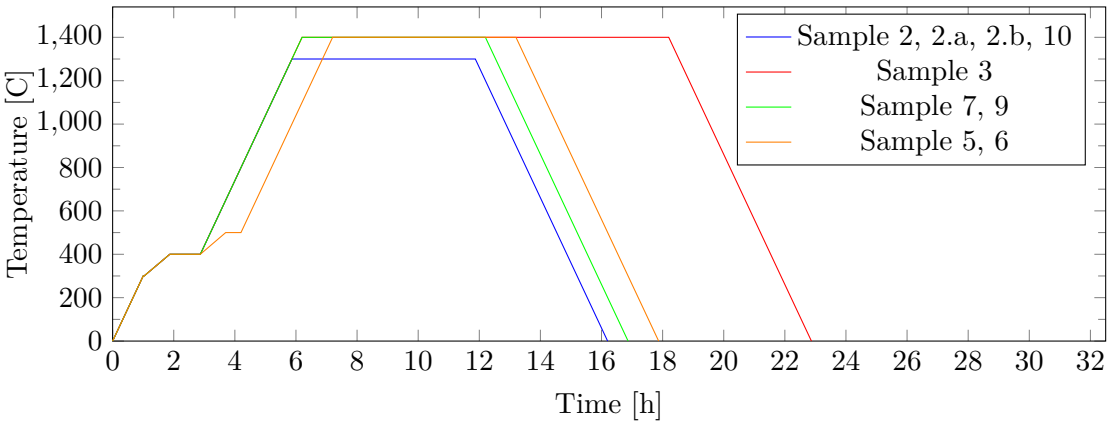


Figure 3.10: Sintering profiles.

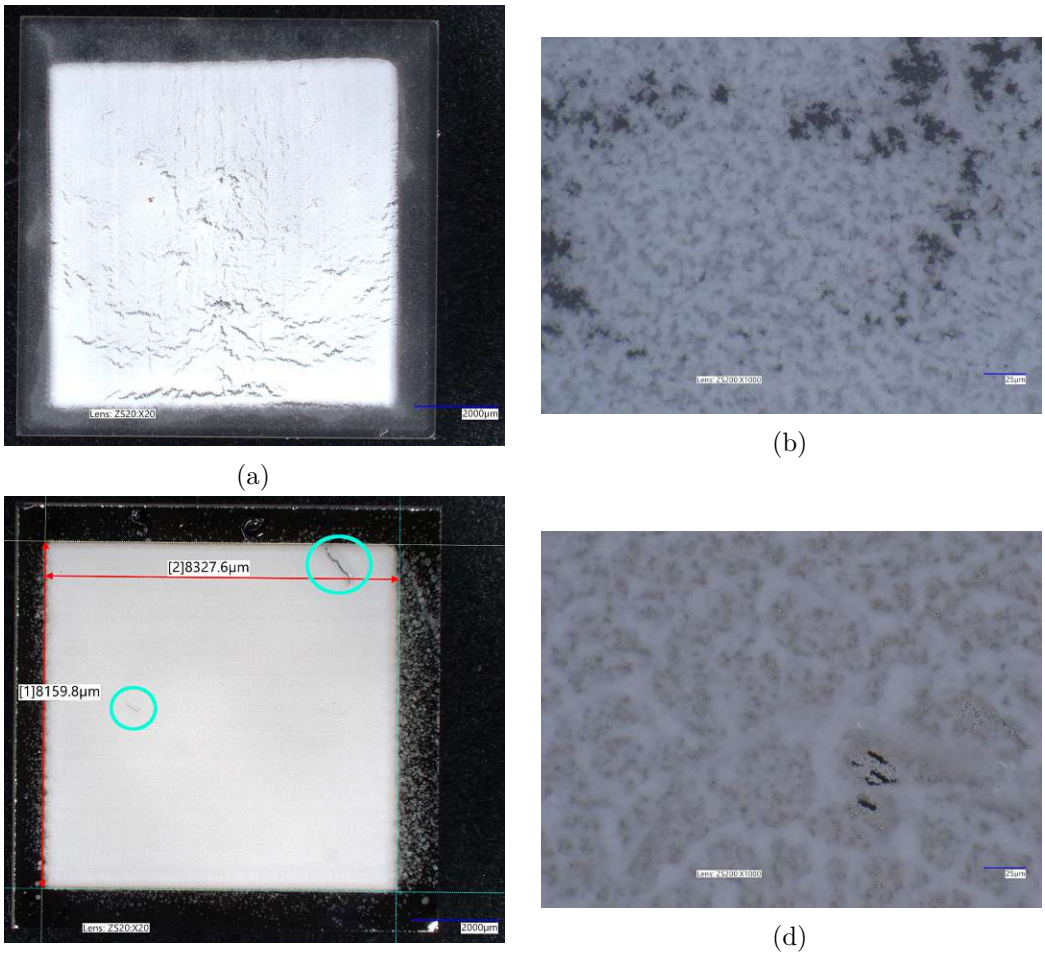
3.3 Measurements

In the last section, the evolution of the printing process was described along with different samples. The following chapter explains the individual measurement methods that were used to characterize the samples. The purpose of the methods for evaluating the printed layers as well as the measurement results are presented.

3.3.1 Optical microscope

The attached build plate microscope USB camera only has a limited magnification and measurement capability, compared to a modern (digital) optical microscope. While it proved useful for quick evaluations without the need to demount, and transfer samples, a VHX-7000N digital microscope (Keyence) was used to assess the macroscopic nature of the printed layers. Basic quality control checks that were performed, helped to decide whether to proceed with more elaborate measurements or to reject the specific sample.

Defects that can be found using optical microscopy are for example cracks as found on sample 9, shown in Fig. 3.11a.



(a) (b) (c) (d)

Figure 3.11: Samples 9 and 10 before sintering. (a) Overall image of sample 9 with visible cracks on the entire surface. (b) By zooming in one can examine a crack and also the arrangement of particle agglomerates. (c) Sample 10 including dimensions, and two dust particles marked with circles. (d) The platinum layer is revealed at scratches, and the deposit formed a fine and a rough structured layer on top of each other.

Comparing sample 9 and 10 the importance of ink composition compatibility to the

substrate become evident. They where printed in the same batch, the difference between them being the substrate. For sample 9 it was a magnesium oxide single crystal and for sample 10 the YSZ layer was printed on top of a platinum thin film that was sputtered on a YSZ single crystal. When zooming in to inspect the nature of the cracks of sample 9 in more detail, one can also see the arrangement of the particle agglomerates (Fig. 3.11b). Sample 10 in comparison shows no cracks, and also the structure up close looks different. The dark spots shown in Fig. 3.11d, small imperfections, reveal the platinum layer. It seems that for sample 10 a fine layer of particles was deposited on the platinum substrate, and more rough structured particle agglomerates were placed on top.

Dust fibres, encircled in Fig. 3.11c, were found on almost all samples indicating that using an enclosure for the print area would be beneficial.

The optical microscope is also useful for checking dimensional accuracy of the printer. The 8 mm x 8 mm square is slightly over-sized which is attributed to the rasterization algorithm implemented: Dots will be printed overlapping the contour of the model slightly, because a droplet position is counted to be inside, as long as the center point of the dot is inside the contour. This also leads to softened edges because fewer dots are printed near the edge than inside.

At the right edge of sample 10, stray drops can be seen, which could be due to nozzles spraying (not shooting straight down) or producing satellite drops, which was also observed for some nozzles at the nozzle tests during the print (see Fig. 3.9 in Section 3.2). While undesired, these stray drops were useful to estimate the solid content of single droplets, and the nature of the particles/agglomerates, as captured in Fig. 3.12.

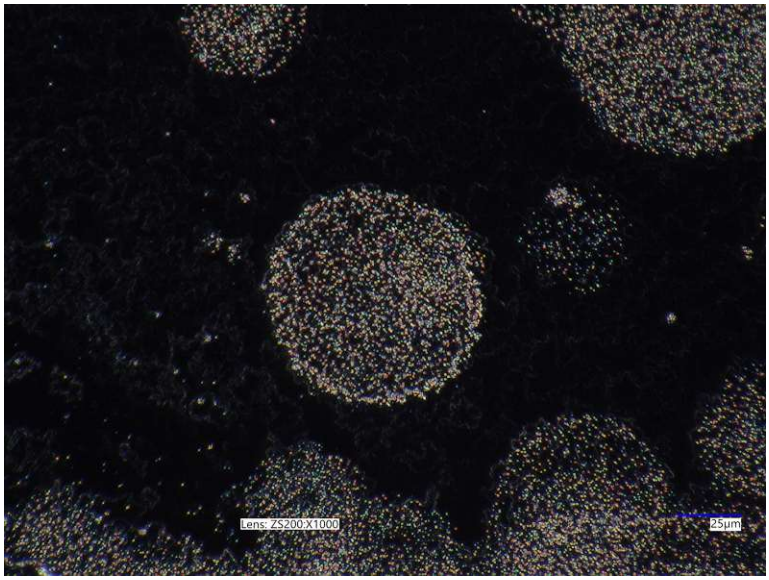
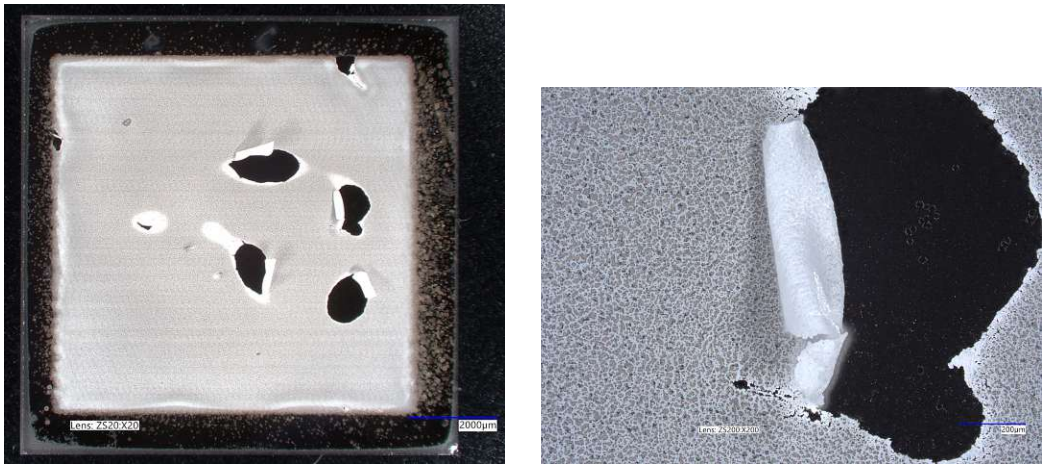


Figure 3.12: A stray drop of sample 10 allows the inspection of the solid content (ink C) and nature of particles/agglomerates.

If a sample passed this preliminary quality control, it proceeded to sintering (see sintering conditions specified in Section 3.2), after which again optical microscopy was carried out. For most of the samples the effect of sintering could not be further analyzed with optical

microscopy. For sample 10 however, severe damage could be observed as shown in Fig. 3.13.



(a) (b)

Figure 3.13: Sample 10 was severly damaged during sintering (a), the detached printed layer can be seen in the close up image (b).

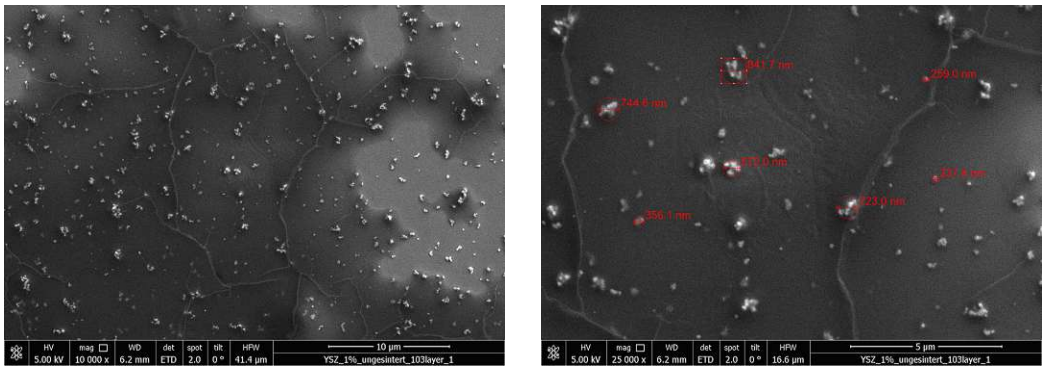
The printed layer seems to have detached from the platinum surface at multiple positions, however as displayed in 3.13b, parts of the layer are still intact in a rolled up state. The black spots that are spread over the entire surface indicate that small holes emerged. For other samples, without the platinum substrate, the color of the substrate was too similar to the deposited layer to identify holes without further analysis, for example by SEM.

3.3.2 Scanning Electron Microscopy

Measurements were carried out using a Quanta 250 FEG SEM using the secondary electron detector. The measurements were helpful in various ways.

An important property to test for the electrolyte layer of the oxygen ion battery, is whether the printed, and sintered layer is dense. The density and distribution of printed particles, as well as the porosity of the layer could be analyzed using SEM images. One could also see how the layers changed during sintering and assess the effects of different sintering profiles. When scanning single drops, the size of the particles and agglomerates of the ink could be estimated.

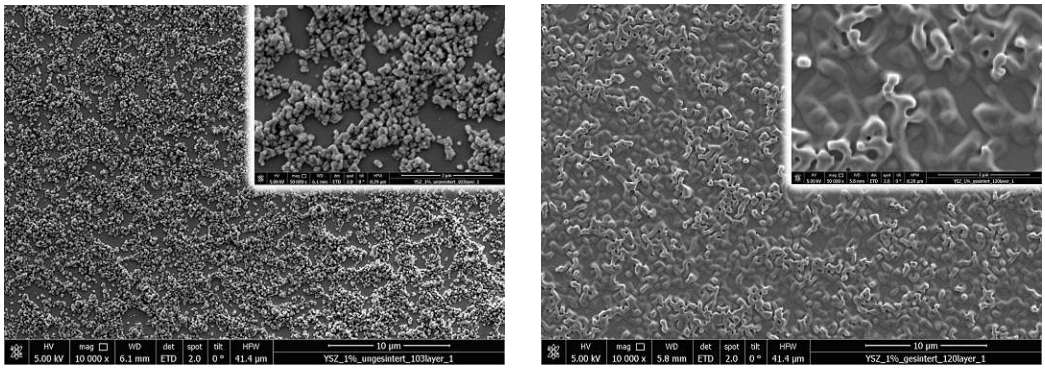
Particle/agglomerate size Similarly to the stray drop captured with the optical microscope shown previously in Fig. 3.12, a stray drop was imaged with SEM from the unsintered sample 1 which is printed with the ink A (1 wt.% particle load). Fig. 3.14 shows agglomerates with a size between 200 nm and 900 nm. The fluid-like substance visible in the background is suspected to be solvent residue that has not evaporated.



(a) (b)

Figure 3.14: A stray droplet from the unsintered sample 1 allows to inspect the size of particle agglomerates, which is in the range of 200 nm to 900 nm.

Sintering The transformation of the structure during sintering can be observed when comparing the top view of the unsintered sample 1 to sample 2 (same ink and print settings, sample 2 has 17 additional layers) as shown in Fig. 3.15.



(a) (b)

Figure 3.15: Top view of sample 1 (a) and sintered sample 2 (b). The length of the scale bar in the cutout in the upper right corner is 3 µm.

One can see that the particles are fused together after sintering, but the particle density was not enough to cover the whole surface with a homogeneous layer.

Porosity In the close up cutout of Fig. 3.15b, the surface of the YSZ single crystal substrate is still visible in the background, indicating high porosity. Sample 3 was printed using ink B with higher particle load, more layers (200) and higher drop density (180 DPI), and which was sintered at a higher temperature (1400 °C for 12 h) (see Section 3.2). Here, a much denser layer has formed, shown in Fig. 3.16.

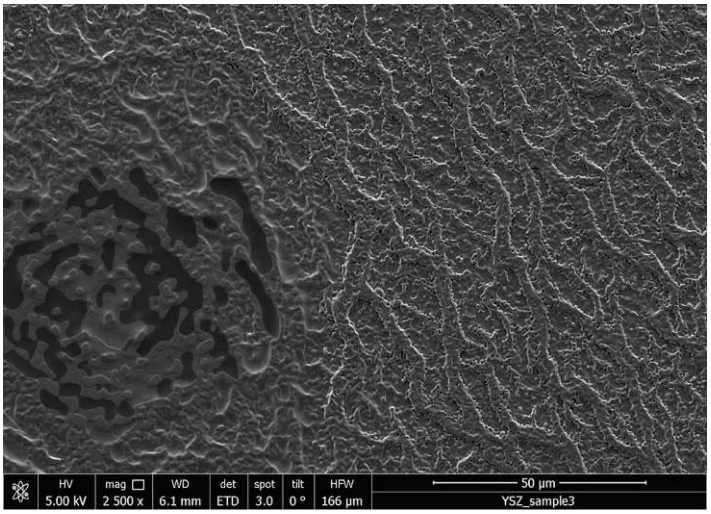
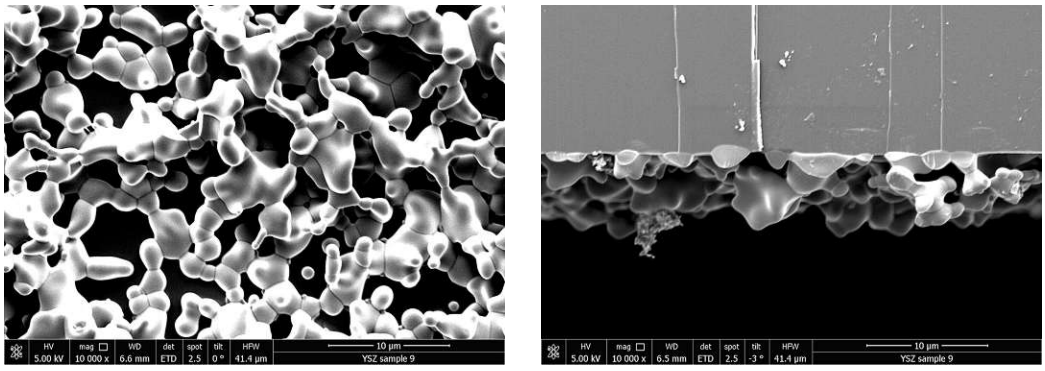


Figure 3.16: Top view of sample 3 (after sintering), where the single crystal substrate is only visible through a defect in the printed layer.

In addition to the increased density and greater fusion of the particles, sample 3 also shows a coarse structure on the surface of the printed layer. Ink C was created with an even higher particle loading of 5 wt.% with the help of the dispersant additive BYK-106 (see Section 3.1) with the expectation to finally receive a dense layer after sintering. However, an even more porous layer seems to have formed for sample 9, which may be due to a surplus of organic additives in the ink. It is suspected that the additives remained in the printed film after drying, and that their combustion during sintering formed pores in the YSZ layer.



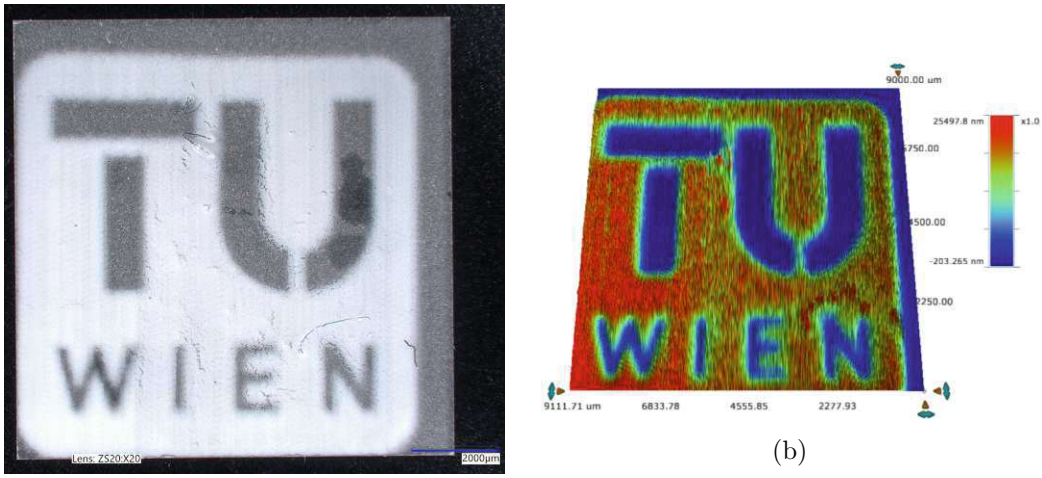
(a) (b)

Figure 3.17: Top view of sample 9 after sintering (a) and cross section (b) showing a high porosity.

Sintering necks and grain boundaries are well visible in 3.17a. The particles that fused together are even bigger compared to 3.15b but there are also large gaps between them. The cross section image in 3.17b shows the unevenness of the surface and illustrates the difficulty of defining the thickness of the printed layer.

3.3.3 Profilometer

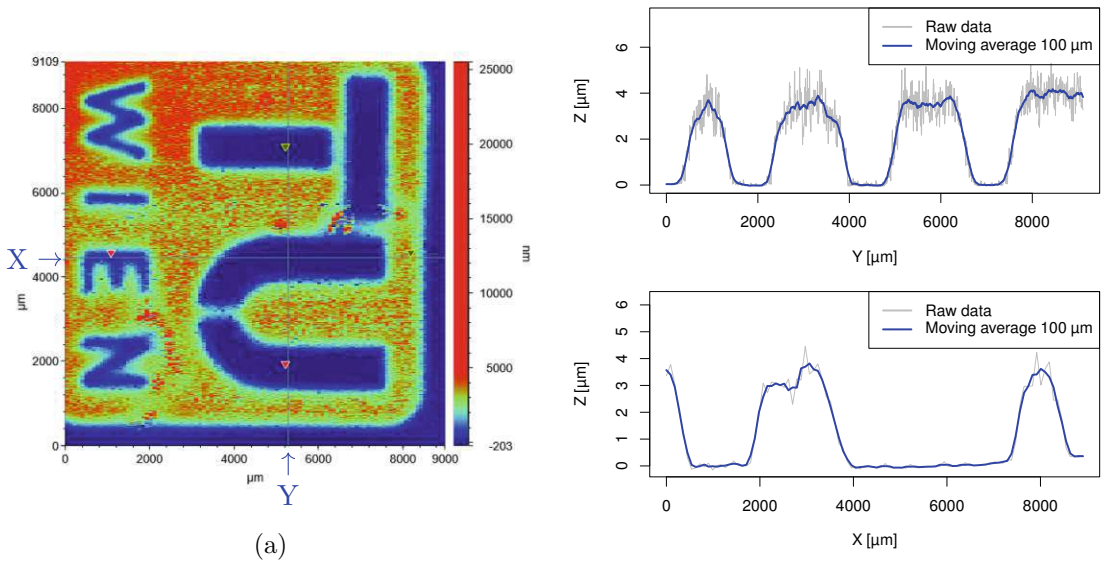
Another method to measure the layer thickness next to the inspection of cross sections by SEM, is to scan the sample using a profilometer. In the following the measurements on sample 5 performed using a DektakXT profilometer are shown. The print pattern was a TU Wien logo with the outer dimensions of 8 mm x 8 mm. For comparison an overview image (optical microscopy) is shown next to the 3D view of the recorded profile in Fig. 3.18.



(a) (b)

Figure 3.18: Sample 5 overview imaged using optical microscopy (a). The result of a 3D map scan using the profilometer (b).

Fig. 3.19 shows a top view of the scan and plots of one scan line each in X and Y direction, indicated by the blue arrows in Fig. 3.19a. The surface appears to be rough, so a moving average was applied to better estimate the average thickness, which seems to be in the range of 3.5-4 µm. The printed layer seems to get slightly thicker at the left side, which is also indicated by the red color in Fig. 3.19a.



(a) (b)

Figure 3.19: In the top view of the scan (a), an X and a Y line are indicated by blue arrows. The profiles at these lines are shown in the plots in (b).

While the thickness of the printed layer is very small compared to the models size, the

profilometer measurement shows that it was actually possible to print a 3-dimensional object.

At this point it is interesting to see, how the printed object compares to its rasterization, and the simulation of it. To illustrate this, the rasterization of one layer, as shown in Fig. 2.20 in Section 2.5.2 is partially superimposed onto the optical microscope image of Fig. 3.18a, the result being shown in Fig. 3.20. The diameter of the simulated drops was 120 µm, and droplets measured during nozzle tests while printing sample 5 were in the range of 100-130 µm. It can be observed that the lines of the smaller letters (the width of the I being ≈300 µm) are as narrow as just two droplets.

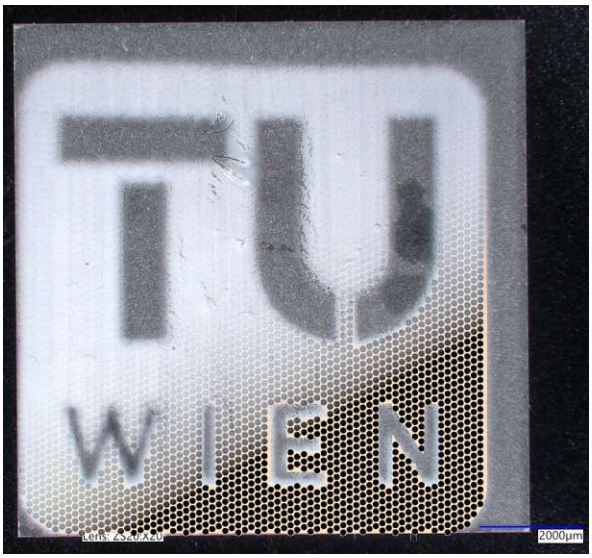


Figure 3.20: Rasterization of a layer partially superimposed on the microscopy image of the printed object.

3.3.4 X-ray diffraction

X-ray diffraction (XRD) was used to determine the crystallographic system of the printed YSZ layers. Thereby a copper X-ray source was used and all measurments were performed in Bragg-Brentano geometry. In the following, the diffractogram recorded from sample 9 is compared to diffractograms of:

- Magnesium oxide for the substrate, MgO (PDF#00-045-0946) [52].
- YSZ with a cubic crystal structure, $Y_{0.15}Zr_{0.85}O_{1.93}$ (PDF#00-030-1468) [53].
- YSZ with a tetragonal crystal structure, $Y_{0.15}Zr_{0.85}O_{1.93}$ (PDF#01-070-4431) [54].

Fig. 3.21 shows the comparison to the peaks of YSZ in cubic crystal structure, which are marked red, and Fig. 3.22 shows the comparison to the peaks of YSZ in tetragonal crystal structure, marked in green. The blue lines in both Figures show the peaks of MgO. The peaks marked with blue stars are also attributed to the MgO substrate (Cu K_β, W L_α radiation).

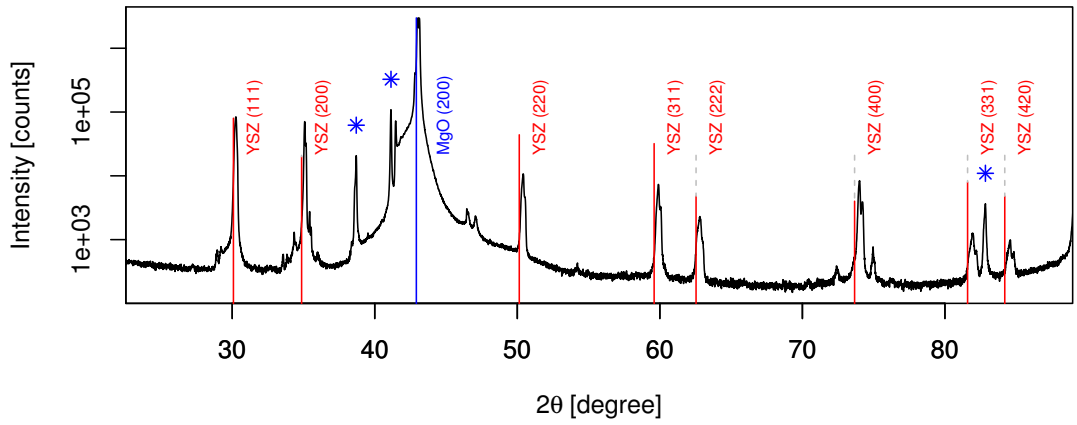


Figure 3.21: XRD diffractogram of sample 9 compared to YSZ (PDF#00-030-1468) [53] with a cubic structure.

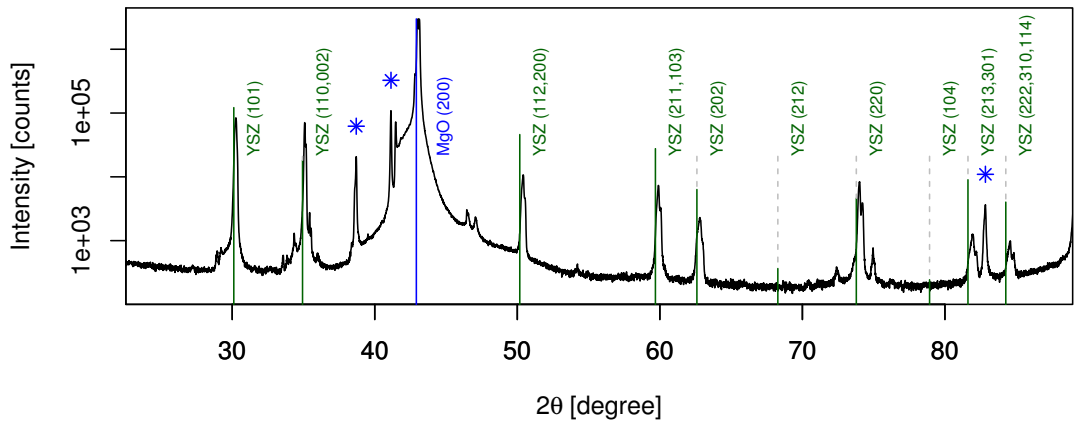


Figure 3.22: XRD diffractogram of sample 9 compared to YSZ (PDF#01-070-4431) [54] with a tetragonal structure.

While being difficult to assess because of the similar peaks of the two crystal structures, the printed layer is assumed to be of cubic structure, because in diffractogram of sample 9 the (212) peak at $2\theta = 68.3°$ of the tetragonal structured YSZ is missing. The (104) peak at 78.9° might be missing as well, but it might also simply be hidden in the noise.

3.3.5 Electrochemical Impedance Spectroscopy

The electrolyte layer for oxygen ion batteries should have a high ionic conductivity, but inhibit electron(hole) conduction between the electrodes [55]. To assess these electrochemical properties of the printed electrolyte layers, electrochemical impedance spectroscopy (EIS) was conducted in two different setups, one for in-plane and one for cross-plane measurement. The measurements were performed with an Alpha-A Frequency Analyzer (Novocontrol), which applies an alternate voltage at different frequencies to the sample, and records the electrical impedance measured for each frequency.

In Plane A lower conductivity of the electrolyte layer leads to a higher voltage drop across the electrolyte layer and consequently to lower performance of the battery [55]. To measure the conductivity, sample 7 was prepared by brushing on platinum electrodes as shown in Fig. 3.23. Then EIS in the range from 1 MHz to 0.1 Hz was conducted at different temperatures between ≈260 °C and ≈656 °C. All EIS measurements were conducted at 10-20 mV_{rms}. During the measurements, the sample was heated homogeneously in a tube furnace.

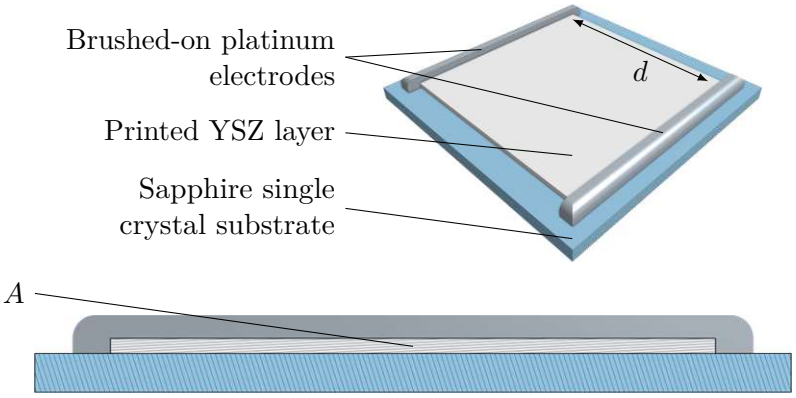


Figure 3.23: Sketch of the setup for in plane conductivity measurements, showing the length d and the cross-sectional surface A.

The resulting data points were fitted to the equivalent circuit model shown in Fig. 3.24, where in case of thick, dense polycristalline samples one $R \parallel CPE$ element can be attributed to the bulk and the other to the grain boundary conductivity. CPE indicates a constant phase element and is used to describe a non-ideal capacitor.

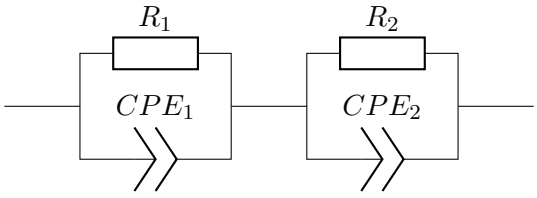


Figure 3.24: Equivalent circuit model to describe bulk and grain boundary conductivity.

Ideally, Nyquist plots of this equivalent circuit show two distinguishable semicircles. The fitting process (using ZView[a]), however, turned out to be difficult, as the semicircles seem to overlap to a large extent. This is attributed to the porosity of the printed layers and the measurement geometry.

In Fig. 3.25 the measurement data for sample 7 and fit to the the model at four temperatures are shown as examples. For each temperature, the outlier at 50 Hz was removed.

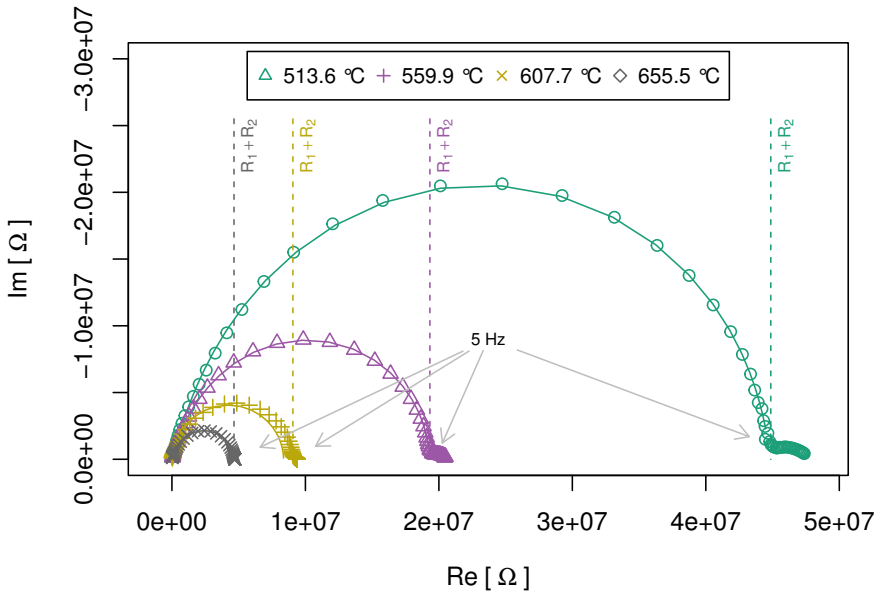


Figure 3.25: EIS measurements of sample 7 at four different temperatures. The symbols represent the actual measurements and the lines show the curves of the fitted model. The arrows point to the measurements at 5 Hz.

Using the sum of the fitted values $R_1 + R_2$ at different temperatures together with the dimensions d and A of the sample, the total ionic conductivity σ can be calculated as shown in eqn. 3.1.

$$R_{total} = \frac{1}{\sigma}\frac{d}{A} \implies \sigma = \frac{1}{R_1 + R_2}\frac{d}{A} \tag{3.1}$$

To evaluate the printed layers, the conductivity values at different temperatures are compared with measurements that were conducted on polycrystals in [56]. These measurements were chosen for comparison because the polycrystals were prepared from YSZ powder with an Y_2O_3 content of 8 mol.% (TOSOH), like it was used to prepare the inks in this work (see Section 3.1).

The conductivity at each temperature is calculated using eqn. 3.1. Using optical microscopy, the distance between the electrodes, d was measured to be ≈7.3 mm. To calculate the cross-sectional surface A, the length of the cross section, which was ≈8.1 mm, is multiplied by the thickness. However the thickness is difficult to measure because of the large porosity. SEM measurements of the cross section of sample 9 (shown in Fig. 3.17b of Section 3.3.2) indicate a range of 3-5 µm which agrees with the profilometer measurements on sample 5 (see Section 3.3.3). These samples were printed in the same batch, and in the following a thickness of 4 µm was used to demonstrate the calculations and the comparison. With $d = 4$ µm the

[a]ZView: Equivalent circuit modelling and fitting https://www.scribner.com/software/68-general-electrochemistr376-zview-for-windows/

cross sectional surface is $A = 32\,533.6\,\mu m^2$. The fitted values R_1 and R_2 and the calculated conductivity is shown in Table 3.3.

Table 3.3: Fitted values R_1 and R_2 and conductivity at four different temperatures

Temperature	R_1	R_2	σ
513.60 °C	35.2 MΩ	9.65 MΩ	$5.03 \times 10^{-5}\,Scm^{-1}$
559.90 °C	16.0 MΩ	3.32 MΩ	$1.17 \times 10^{-4}\,Scm^{-1}$
607.70 °C	8.25 MΩ	0.81 MΩ	$2.49 \times 10^{-4}\,Scm^{-1}$
655.50 °C	4.40 MΩ	0.25 MΩ	$4.85 \times 10^{-4}\,Scm^{-1}$

The calculated conductivities are further displayed in an Arrhenius plot together with the results from literature to facilitate comparison. In the Arrhenius plot, a temperature-dependence of a quantity, is shown by plotting the logarithm of the quantity over the reciprocate of the temperature. The temperature dependence of ionic conductivity of YSZ can be described by the Arrhenius equation shown in eqn. 3.2, where σ_0 is a material constant, E_a is the activation energy, k_B is the Boltzmann constant. [57].

$$\sigma = \frac{\sigma_0}{T} e^{\frac{-E_a}{k_B T}} \tag{3.2}$$

When taking the logarithm of both sides of eqn. 3.2, the result (eqn. 3.3) can be interpreted as a linear equation ($y = kx + d$) with $x = \frac{1}{T}$ and $y = \log(\sigma T)$.

$$\log(\sigma T) = \log(\sigma_0 e^{\frac{-E_a}{k_B T}}) = \log(\sigma_0) - \log(e) E_a \frac{1}{k_B} \frac{1}{T} \tag{3.3}$$

Therefore, when $\log(\sigma T)$ is plotted over $\frac{1}{T}$, a line is formed in the Arrhenius plot. In Fig. 3.26, the values from Table 3.3 are plotted together with measurements from literature in this manner.

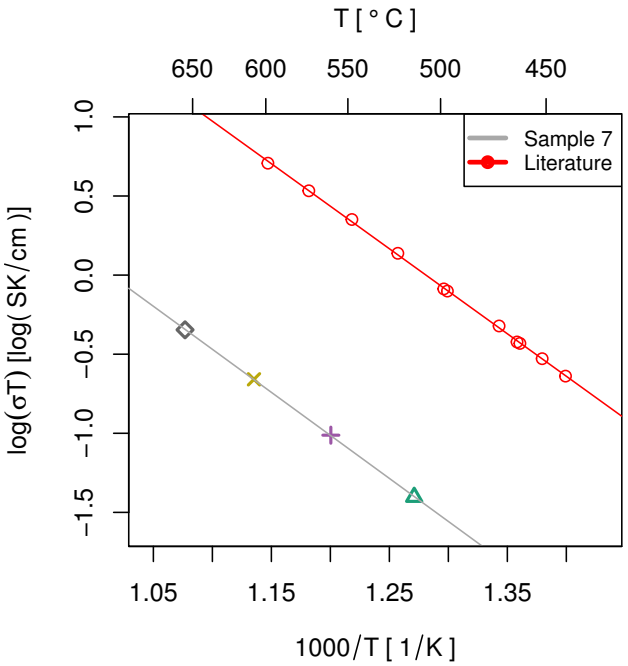


Figure 3.26: Arrhenius plot of the calculated conductivities together with the measurement results in [56]. The line fitted through the data points is parallel to the one from literature but shows an offset.

One can observe that the four points are actually on a line. From the slope of the line ($k = -5.439\,75\,\frac{\mathrm{K}}{1000}$ in the plot, $k = -5439.75\,\mathrm{K}$ with respect to eqn. 3.3) the activation energy for the printed film of sample 7 was calculated to 1.08 eV as shown in eqn. 3.4.

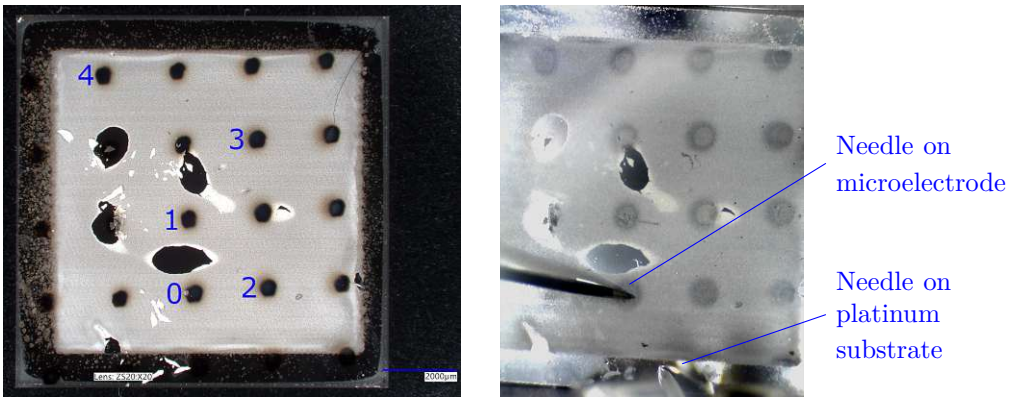
$$k = -\log(e)E_a\frac{1}{k_B} \implies E_a = -\frac{k}{\log(e)}k_B = -\frac{-5439.75\,\mathrm{K}}{0.43}8.62\times10^{-5}\,\frac{\mathrm{eV}}{\mathrm{K}} = 1.08\,\mathrm{eV} \quad (3.4)$$

Furthermore, the determined line runs almost parallel to the measurements described in the literature. Accordingly, they show nearly the same activation energy, indicating that indeed the same conduction process is observed, namely oxide ion motion in YSZ. The offset shows that conductivity of the printed film is lower, which was expected due to the large porosity of the measured sample.

Cross Plane As stated before, the electrolyte layers should isolate the electrodes from each other. In case of contact of the electrodes, leakage currents can flow, which leads to self-discharge of the battery. Therefore, the printed layers must be dense, and cross plane EIS was used to check if this is the case.

The measurements were performed on sample 10, where the substrate was a platinum-coated YSZ single crystal. After sintering, microelectrodes were formed on the surface of the printed layer by sputtering.

While the sample was placed on a heating element set to 500 °C, the lower platinum layer and several microelectrodes were contacted one after the other. Needles were used that were gently lowered to contact the platinum surface without scratching it, as shown in Fig. 3.27b. For each of the contact points marked in Fig. 3.27a EIS was conducted.



(a) (b)

Figure 3.27: Sample 10 showing the measured microelectrodes in (a) and the setup in (b).

All of the measured points except microelectrode 4 showed a low resistance of ≈7 Ω at low frequencies. At microelectrode 4 the low frequency resistance was ≈200 Ω, which is attributed to poor contact of the needle. This indicates that a short circuit had formed between the

microelectrodes and the platinum substrate, as the printed sample did not have sufficient density. As an example the plot of the recorded impedance of point 3 is shown in Fig. 3.28.

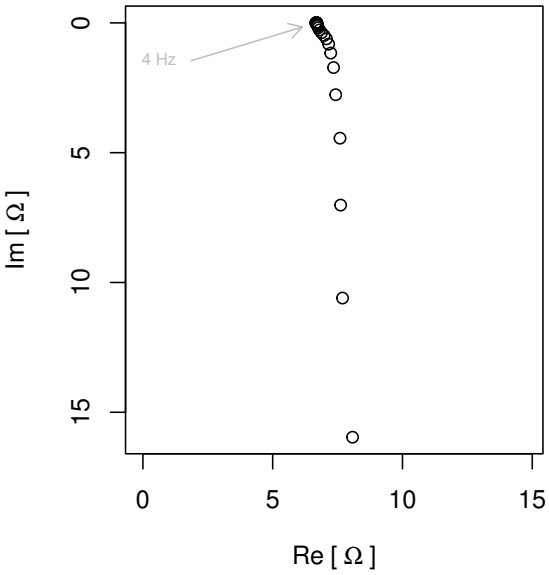


Figure 3.28: Cross plane impedance of microelectrode 3 of sample 10, showing low resistance at low frequencies indicating a short circuit.

In summary, printing with the developed printer and YSZ inks prepared from powder works. It was possible to print layers as well as fine 3-dimensional structures like the TU Wien Logo. The process how to characterize the printed layers was explained using examples. However, even though up to 200 layers were printed on top of each other, remaining porosity can be viewed in SEM images. In addition, EIS showed a lower conductivity than a comparable sample in the literature indicating that the density of the layer was not yet sufficient. On the other hand, the activation energy measured via in-plane experiments (1.08 eV) strongly suggest that indeed oxide ion conduction in YSZ takes place. Further optimizations, which are discussed in the next chapter, are intended to improve the properties of the printed layers sufficiently to use them in oxygen-ion batteries.

Chapter 4

Discussion and conclusion

The final chapter begins by discussing lessons learned from developing and using the printer. It highlights what worked well, as well as specific areas where improvements are needed or suggested that should be considered when rebuilding the printer or building the next version. The findings are used to develop guidelines that can be used when selecting a future printer system for inkjet-printing the electrolyte layer of oxygen ion batteries. Finally, reconsidering the printed samples and results of related work presented in Section 1 an outlook is given on further experiments as well as potential adjustments to the ink and printing process to obtain dense layers.

4.1 Discussion

Use of Zephyr Project Starting with the software development process, using Zephyr project as a base for the firmware development showed to have advantages and disadvantages. It was beneficial due to scheduling of concurrent tasks provided by the RTOS kernel and the ready-made modules for USB communication. Also, the logging and debugging tools helped to save time during development. On the other hand, it took time to understand the concepts of drivers and their creation in order to program low-level structures such as interfacing peripherals, accessing special features of timers or using interrupts. Therefore, no definite recommendation can be made to use Zephyr for projects of this scope, in the prototyping phase. It may be more advantageous, when turning the printer into a product: If more features are added and the code has to be maintained for the lifetime of the product, the well organized code and architecture, provided modules and tools will outweigh the additional time investment [36]. However, as already discussed in Section 2.2.4, using an open source 3D-printer firmware like Marlin or Klipper could be more suitable in this case.

Dropwatcher Effort spent on accurate and repeatable timing using the hardware timer functions of the MCU during firmware development seemed to pay off: It was interesting to see that even though the dropwatcher is built from relatively simple inexpensive components, like the USB microscope camera used, it was able to generate useful images that helped to assess the jetting behavior with different inks and different jetting parameters. A concrete downside of the implemented system is the minimum exposure time of 125 ms, explained in Section 2.6, that effectively limits the maximum frame rate to $\frac{1}{125\,\mathrm{ms}} = 8\,\mathrm{Hz}$. This could probably be improved by using a camera system that supports external shutter control.

Implementing a mode to capture jetting frequency sweeps could also help assessing the printing process further, since jetting can show different behavior at different print speeds [48].

Build plate microscope camera The second implemented vision system, the build plate microscope camera, also was a beneficial addition. It was primarily planned as a fast way to inspect printed patterns without transferring the sample to a microscope. However, it was used in various ways: The possibility to inspect single drops inspired to implement automatic parameter sweeps of waveform parameters, as shown in Section 2.5.1. The drying process could be monitored and the required drying time of layers could be estimated. Capturing photo points after every layer allowed to analyze how certain (undesirable) features in the layers arise. Fixing a calibration target to the build plate allowed more accurate homing of the motion stage. It was also used to measure alignment and angle of the printhead to provide these parameters to the rasterizer, and to position the printed structures at the right position on the substrates. The final version of the automated nozzle test that is executed in between printing layers, described in Section 3.2, heavily builds on the build plate microscope camera to record the droplets. However, the light setup could be improved. In order to make the most of the camera's features, a desk lamp was placed next to it to add more light, which could be better integrated into the printer. Also, the current mechanical installation of the camera does not allow to inspect every position where the printhead can print, which is something to address in an updated version of the printer. Conclusively, due to the versatility, the installation of a substrate inspecting camera is a definitive recommendation.

Xaar 128 vs. Irix Core As useful as compensation of blocked nozzles was, the primary strategy is to get all jets to work consistently. Therefore, it was planned to investigate, whether a printhead with a wetting nozzle plate would lead to improvements in jetting stability. The Xaar Irix Core is such a printhead and is electrically and mechanically compatible to the Xaar 128, which made it possible to compare the printing of these two printheads. Using the

Xaar Irix Core, the jetting was more reliable, with almost all nozzles working throughout the whole printing process of multiple hours, and also in the dropwatcher, where the jetting frequency was much lower. This consistency made it first possible to assess the influence of the waveform-changing parameters, the piezo drive voltage and the printhead clock signal on the jetting behavior with a specific ink, as shown in Section 3.1. With the Xaar 128, changing these parameters often had no distinguishable effect, and it seemed completely random, whether a nozzle worked or not. There were problems with blocked nozzles from the start with this printhead, however, it worsened over time, so that degradation of the printhead was suspected.

Both printheads are available in different configurations, for the Xaar 128 the 80 pl version with a non-wetting nozzle plate coating was used, and for the Xaar Irix Core the configuration was 40 pl and wetting nozzle plate. Next to degradation and being an improved version of the printhead, the wetting nozzle plate could also have contributed to more successful printing. One speculative explanation is that the ink film that is spread on the wetting nozzle plate while printing helps to prevent the nozzles from drying out.

It was found that a negative ink pressure of −2 mbar to −8 mbar was required to prevent leaking out of the nozzle for the Xaar 128. Effects of variation inside this range could not be detected. The Irix, on the other hand, was more sensitive: only a slight negative pressure of −0.6 mbar was used. When the pressure was more negative, the ink film on the nozzle plate would be sucked up over time, when it was any greater than 0 mbar, too much ink would wet the nozzle plate over time, preventing jetting in some regions. Also in the dropwatcher, effects of pressure variation could be observed by drops having different speeds.

Ink supply system Because of the increased pressure sensitivity of the Irix, adding a second pressure sensor, or a sensor with the same range but higher resolution, would be beneficial. While the installed ABPDRRV015PDSA3 sensor worked well, and the measurement range is useful for the power pressure purge maintenance routine, its resolution limits the pressure control, to ≈0.5 mbar. A second sensor with a smaller measurement range, therefore, having a finer resolution at the same number of bits, could improve that.

The increased pressure sensitivity of the Irix was also the reason, to turn off the ink pump pressure control for the time the printhead was ejecting, as explained in Section 2.5.3. This is possible due to a property of peristaltic pumps, to block the flow in the tube like a valve when they are turned off. This property as well as the mechanics not coming into contact with ink or cleaning fluid and the ability to achieve the necessary pressures of up to 600 mbar make the peristaltic pumps recommendable for the ink path. A useful improvement for the current setup would be to find electrically controllable valves that are compatible to the ink, so that printhead maintenance methods could be more automated.

The design of the capping station, to seal the printhead when it is not operational, was a time consuming part of this work that required multiple iterations. This could have been avoided if capping would have been a primary concern at the design of the motion stage: By planning a fixed, specific location for the capping station from the start, where the printhead can move to and automatically dock, could simplify the capping station design.

Motion stage Considering the motion stage, some observations should be considered when purchasing a new system or upgrading this printer: It was found, confirming to [58], that the drop spacing in relation to the diameter of an individual drop, is an important print setting to receive smooth layers. With it, also the amount of fluid per area dispensed varies, which had influence on the drying time that needs to be accounted for to prevent artifacts in the layers, as shown in Fig. 3.3 in Section 3.1. A suitable motion stage should therefore allow a drop placement repeatability and resolution in the print axis to be a fraction of the droplet size, to achieve the desired droplet overlap. In other words, for an average droplet size of e.g. 100 µm, and droplets overlapping by 25 % of their diameter, the drop spacing should

75 µm. A repeatability of 10 µm could lead to an overlap of about 15 % to 35 % which might be acceptable.

Using the encoder for droplet positioning in the print axis, as described in detail in Section 2.4.2, led to an average drop spacing repeatability of ≈8 µm (standard deviation of 128 drop spacing distance measurements) when printing droplets at 120 DPI (≈211.7 µm spacing). This and configuring to print not at every encoder tick, but only at e.g. every second, third, fourth, etc., made it possible to set a desired drop overlap in the print axis.

For the second axis that moves the printhead to the next track, a similar guideline was found: When printing tracks/layers on top of each other, the move axis should have a resolution/repeatability that allows to control the overlap between the droplets of the layers. With limited resolution and repeatability of the move axis in this work, this was not achieved, and bad alignment of two tracks that were printed on top of each other was observed. This was especially visible during blocked nozzle correction, when the correcting jet did not fill the gaps left by the previous printed track, but landed where a working nozzle had printed before instead. The artifacts produced by these random positioning errors leveled out when printing many layers on top of each other, like it was done here, where up to 200 layers were printed. However, to speed up and simplify the printing process, the number of layers should be reduced in the future, and thus the repeatability of the move axis gets more important. Adding an encoder to this axis cannot solve the problem, as it can only move to discrete positions limited by its resolution.

Ink Reconsidering the nature of the printed samples, it is clear that the required dense electrolyte layers for oxygen ion batteries were not produced yet. While the thickness of the printed layers was ≈4 µm, identified by profilometer measurements of sample 5 and SEM imaging of sample 6 and 9, they also showed a high porosity. In related work presented in Section 1, dense layers with with similar thickness were produced with significantly less layers. This suggests that the particle load of the ink used was higher. To achieve similar results, the ink's capacity for suspended particles should be improved. Smaller particle size can decrease sedimentation speed [59], which could be achieved by hydrothermal synthesis as shown in [13]. It should be further investigated, whether reducing the amount of dispersant and surfactant additives leads to reduced porosity after sintering. Also, contrary to the solvent-based inks prepared in this work, water-based dispersion ink was used in [12] [13] and [15]. Whether water-based ink can suspend YSZ particles better needs to be investigated further.

Printhead Either way, a printhead that can jet water-based ink is preferable, especially for a prototyping system, where different ink compositions are evaluated. Another way to increase particle loading, while also preventing sedimentation like shown in Fig. 3.6 in Section 3.2, is to use a printhead designed to be used with a recirculating ink supply, as briefly mentioned in Section 2.1.1. This should help to remove air bubbles from the nozzle vicinity and prevent sedimentation of particles by the agitation of the ink. In fact, many printheads use it for those reasons [27] [30] [59] [60]. The main reason why a recirculating printhead was not selected in this work, was to keep it simple, as it was expected that a recirculating supply needs more components like a pump and additional tubing. But it seems the requirements for the non-recirculating ink supply system have been underestimated: When comparing the exemplary graphic describing a recirculating supply in [30] to the ink supply system implemented, shown in Section 2.3, one can see many similarities, indicating that only few changes would be required. In fact, in the implemented keep-alive mode, cleaning fluid actually already circulates through the printhead, using the nozzles and the capping station like an outlet port. Consequently, evaluating the use of a recirculating printhead should be a next step.

A smaller nozzle orifice is more restrictive in terms of particle size [23], as discussed in Section2.1.1. However, printing with the smaller nozzle diameter of 35 µm of the Xaar Irix

Core, as opposed to 50 µm of Xaar 128 worked well. This suggests that the larger nozzle size is not a requirement. Other, minor recommendations for printhead selection would be to select one with more elaborate waveform customization options, so that further customization to the ink is possible. With this model tailoring the ink to the printhead seemed more effective, and the selected waveform parameter, discussed in Section 3.1, were close to the nominal values stated in the printhead data sheet. Furthermore, it would be preferable if the internal structure, including any internal filters can be cleaned or exchanged.

Enclosure Every sample that was checked after printing had at least one defect that was caused by dust particles that landed on the sample during the printing process. If a particular dust particle lands on the sample after only a small number of layers, it continues to be printed on. This could even have caused a hole after sintering. To conclude this list of recommendations, an enclosure is therefore suggested to mitigate this problem.

4.2 Summary and outlook

An inkjet printer was built for the task of printing electrolyte layers for oxygen ion batteries. Requirements were defined, components selected and integrated to a complex system including mechanical and electrical design, as well as software development. In iterations, the design was tested and optimized, and the design was extended as new requirements were found.

Several inks were developed and tested to achieve printable compositions. The printer was applied to the task and a number of samples were printed onto single-crystal substrates.

The samples were characterized with different methods. While the printing itself worked very well, and even fine (flat) 3D-structures could be printed (a small TU Wien logo, sample 5), the produced layers were not yet dense, and and still have to be optimized prior to their use as electrolyte layers.

Along with guidelines and recommendations to consider when updating or rebuilding the printer or purchasing a new printing platform, next steps to reach the goal of printing dense layers are outlined. The optimization of the ink by using smaller size particles, adapting the use of additives and increasing the particle load will be investigated. Furthermore, the use of water-based ink and a printhead with recirculating ink supply should be considered.

References

[1] I. Kougias, N. Taylor, G. Kakoulaki, and A. Jäger-Waldau, "The role of photovoltaics for the european green deal and the recovery plan," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 111017, 2021.

[2] F. Cebulla, J. Haas, J. Eichman, W. Nowak, and P. Mancarella, "How much electrical energy storage do we need? A synthesis for the U.S., Europe, and Germany," *Journal of Cleaner Production*, vol. 181, pp. 449–459, 2018.

[3] A. Schmid, M. Krammer, and J. Fleig, "Rechargeable oxide ion batteries based on mixed conducting oxide electrodes," *Advanced Energy Materials*, vol. 13, no. 11, p. 2203789, 2023.

[4] B. Scrosati and J. Garche, "Lithium batteries: Status, prospects and future," *Journal of Power Sources*, vol. 195, no. 9, pp. 2419–2430, 2010.

[5] C. Arbizzani, G. Gabrielli, and M. Mastragostino, "Thermal stability and flammability of electrolytes for lithium-ion batteries," *Journal of Power Sources*, vol. 196, no. 10, pp. 4801–4805, 2011.

[6] B. K. Sovacool, "When subterranean slavery supports sustainability transitions? Power, patriarchy, and child labor in artisanal Congolese cobalt mining," *The Extractive Industries and Society*, vol. 8, no. 1, pp. 271–293, 2021.

[7] F. Dolezal, "Reactive sputtering of dense yttria-stabilized zirconia," Master's thesis, TU Wien, Vienna, Austria, 2024.

[8] K. Sztymela, M. Bienia, F. Rossignol, S. Mailley, S. Ziesche, J. Varghese, and M. Cerbelaud, "Fabrication of modern lithium ion batteries by 3D inkjet printing: Opportunities and challenges," *Heliyon*, vol. 8, no. 12, pp. e12623–e12623, 2022.

[9] Y. Guo, H. S. Patanwala, B. Bognet, and A. W. Ma, "Inkjet and inkjet-based 3D printing: Connecting fluid properties and printing performance," *Rapid prototyping journal*, vol. 23, no. 3, pp. 562–576, 2017.

[10] J. R. Castrejon-Pita, W. R. S. Baxter, J. Morgan, S. Temple, G. D. Martin, and I. M. Hutchings, "Future, opportunities and challenges of inkjet technologies," *Atomization and Sprays*, vol. 23, no. 6, pp. 541–565, 2013.

[11] B. Derby, "Inkjet printing ceramics: From drops to solid," *Journal of the European Ceramic Society*, vol. 31, no. 14, pp. 2543–2550, 2011. Special Issue ICCPS 11: 11th International Conference of Ceramic Processing Science.

[12] S. Anelli, M. Rosa, F. Baiutti, M. Torrell, V. Esposito, and A. Tarancón, "Hybrid-3D printing of symmetric solid oxide cells by inkjet printing and robocasting," *Additive Manufacturing*, vol. 51, p. 102636, 2022.

[13] M. Rosa, P. Gooden, S. Butterworth, P. Zielke, R. Kiebach, Y. Xu, C. Gadea, and V. Esposito, "Zirconia nano-colloids transfer from continuous hydrothermal synthesis to inkjet printing," *Journal of the European Ceramic Society*, vol. 39, no. 1, pp. 2–8, 2019.

[14] M. Charalampakis, L. Zouridi, I. Garagounis, A. Vourros, G. E. Marnellos, and V. Binas, "Inkjet-printed LSM-YSZ thin films for enhanced oxygen electrodes in solid oxide fuel cells," *Energy & Fuels*, vol. 38, no. 15, pp. 14621–14631, 2024.

[15] N. Farandos, L. Kleiminger, T. Li, A. Hankin, and G. Kelsall, "Three-dimensional inkjet printed solid oxide electrochemical reactors. I. Yttria-stabilized zirconia electrolyte," *Electrochimica Acta*, vol. 213, pp. 324–331, 2016.

[16] V. Esposito, C. Gadea, J. Hjelm, D. Marani, Q. Hu, K. Agersted, S. Ramousse, and S. H. Jensen, "Fabrication of thin yttria-stabilized-zirconia dense electrolyte layers by inkjet printing for high performing solid oxide fuel cells," *Journal of Power Sources*, vol. 273, pp. 89–95, 2015.

[17] ImageXpert, "The ultimate industrial inkjet printhead comparison chart." `https://imagexpert.com/the-ultimate-industrial-inkjet-printhead-comparison-chart/`. [Online; accessed October 23, 2024].

[18] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer, "RepRap – the replicating rapid prototyper," *Robotica*, vol. 29, no. 1, p. 177–191, 2011.

[19] J. Kim, H. Kim, and D. Bang, "An open-source, 3D printed inkjet DNA synthesizer," *Scientific Reports*, vol. 14, no. 1, p. 3773, 2024.

[20] M. Willis, "Current inkjet technology and future directions," in *Inkjet Technology for Digital Fabrication* (I. M. Hutchings and G. D. Martin, eds.), ch. 15, pp. 343–362, John Wiley & Sons, Ltd, 2012.

[21] G. D. Han, K. Bae, E. H. Kang, H. J. Choi, and J. H. Shim, "Inkjet printing for manufacturing solid oxide fuel cells," *ACS Energy Letters*, vol. 5, no. 5, pp. 1586–1592, 2020.

[22] T.-M. Liou, C.-Y. Chan, and K.-C. Shih, "Effects of actuating waveform, ink property, and nozzle size on piezoelectrically driven inkjet droplets," *Microfluidics and nanofluidics*, vol. 8, no. 5, pp. 575–586, 2010.

[23] I. M. Hutchings and G. D. Martin, "Introduction to inkjet printing for manufacturing," in *Inkjet Technology for Digital Fabrication* (I. M. Hutchings and G. D. Martin, eds.), ch. 1, pp. 1–20, John Wiley & Sons, Ltd, 2012.

[24] J. Przybyla and S. Simske, "HP printhead technology*," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 18, pp. 465–502, John Wiley & Sons, Ltd, 2022.

[25] A. Kamyshny, E. Sowade, and S. Magdassi, "Inkjet ink formulations: Overview and fundamentals," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 5, pp. 93–124, John Wiley & Sons, Ltd, 2022.

[26] A. Tomotake, "Technology of Konica Minolta's inkjet printhead," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 19, pp. 503–521, John Wiley & Sons, Ltd, 2022.

[27] A. Condie and J. Brünahl, "Xaar's inkjet printing technology and applications," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 21, pp. 535–554, John Wiley & Sons, Ltd, 2022.

[28] G. D. Martin and I. M. Hutchings, "Fundamentals of inkjet technology," in *Inkjet Technology for Digital Fabrication* (I. M. Hutchings and G. D. Martin, eds.), ch. 2, pp. 21–44, John Wiley & Sons, Ltd, 2012.

[29] "Hacking the Xaar 128 printhead, ytec3d forum on 3D printers and 3D printing." `https://ytec3d.com/forum/viewtopic.php?t=47`. [Online; accessed October 18, 2024].

[30] N. Campbell, W. Eve, and S. Wilson, "Inca's experience of system integration," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 49, pp. 1151–1169, John Wiley & Sons, Ltd, 2022.

[31] A. Hudd, "Inkjet printing technologies," in *The Chemistry of Inkjet Inks* (S. Magdassi, ed.), ch. 1, pp. 3–18, World Scientific, 2009.

[32] W. Gay, "I2C," in *Beginning STM32* (W. Gay, ed.), Maker Innovations Series, pp. 253–285, Berkeley, CA: Apress, 2024.

[33] D. De and K. P. Ghatak, "Sequential logic circuit," in *Basic electronics*, ch. 13-8-2, London, United Kingdom: Pearson, 1st edition ed., 2010.

[34] S.M. Sze, Kwok K. Ng, "MOSFETs," in *Physics of Semiconductor Devices*, ch. 6, pp. 293–373, John Wiley & Sons, Ltd, 2006.

[35] J. Beningo, "API and HAL fundamentals," in *Reusable Firmware Development: A Practical Approach to APIs, HALs and Drivers*, pp. 29–60, Berkeley, CA: Apress, 2017.

[36] A. Eliasz, "A review of RTOS fundamentals," in *Zephyr RTOS Embedded C Programming: Using Embedded RTOS POSIX API*, pp. 19–67, Berkeley, CA: Apress, 2024.

[37] A. Eliasz, "Zephyr RTOS application development environments and Zephyr application building principles," in *Zephyr RTOS Embedded C Programming: Using Embedded RTOS POSIX API*, pp. 69–122, Berkeley, CA: Apress, 2024.

[38] H. Wijshoff, "Online and offline testing of printheads," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 34, pp. 801–819, John Wiley & Sons, Ltd, 2022.

[39] B. R. Young, W. Y. Svrcek, and D. P. Mahoney, "Fundamentals of single-input/single-output systems," in *A Real-Time Approach to Process Control*, United Kingdom: John Wiley & Sons, 2014.

[40] P. P. Regtien, "Optical sensors," in *Sensors for Mechatronics* (P. P. Regtien, ed.), ch. 7, pp. 161–217, Oxford: Elsevier, 2012.

[41] W. M. Newman and R. F. Sproull, "Solid-area scan conversion," in *Principles of interactive computer graphics*, McGraw-Hill computer science series, ch. 16, pp. 229–246, New York, NY: McGraw-Hill, 1979.

[42] D. Jia, C. Zhang, N. Wu, J. Zhou, and Z. Guo, "Autofocus algorithm using optimized laplace evaluation function and enhanced mountain climbing search algorithm," *Multimedia tools and applications*, vol. 81, no. 7, pp. 10299–10311, 2022.

[43] J. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia, "Diatom autofocusing in brightfield microscopy: a comparative study," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 3, pp. 314–317 vol.3, 2000.

[44] V. Panwar, "Web evolution to revolution: Navigating the future of web application development," *International journal of computer trends and technology*, vol. 72, no. 2, pp. 34–40, 2024.

[45] C. Bouras, A. Papazois, and N. Stasinos, "Cross-platform mobile applications with web technologies," *International Journal of Computing and Digital System*, vol. 4, no. 3, pp. 153–163, 2015.

[46] S. McDowell and M. D. Seyer, *USB explained*. Upper Saddle River, N.J.: Prentice Hall, 1st edition ed., 1999.

[47] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[48] Y. Kipman, P. Best, and K. Pucci, "Drop watcher technology and print quality analysis," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 30, pp. 717–737, John Wiley & Sons, Ltd, 2022.

[49] K. Pucci, "A novel process of automated waveform optimization," *NIP & Digital Fabrication Conference*, vol. 34, no. 1, p. 99, 2018.

[50] K. Pucci and ImageXpert, "How to optimize a waveform." `https://imagexpert.com/how-to-optimize-a-waveform/`. [Online; accessed October 25, 2024].

[51] S. Güttler and A. Gier, "Ceramic inks," in *The Chemistry of Inkjet Inks* (S. Magdassi, ed.), ch. 16, pp. 319–339, World Scientific, 2009.

[52] A. Kern, R. Doetzer, and W. Eysel. Mineralogisch-Petrographisches Inst., Univ. Heidelberg, Germany., ICDD Grant-in-Aid, 1993.

[53] Pfoertsch and McCarthy. Penn State University, University Park, Pennsylvania, USA., ICDD Grant-in-Aid, 1977.

[54] D. G. Lamas and N. E. Walsöe De Reca, "X-ray diffraction study of compositionally homogeneous, nanocrystalline yttria-doped zirconia powders," *Journal of Materials Science*, vol. 35, pp. 5563–5567, 2000.

[55] S. Sen and F. H. Richter, "Typology of battery cells – from liquid to solid electrolytes," *Advanced Science*, vol. 10, no. 33, p. 2303985, 2023.

[56] C. Ahamer, A. K. Opitz, G. M. Rupp, and J. Fleig, "Revisiting the temperature dependent ionic conductivity of yttria stabilized zirconia (ysz)," *Journal of the Electrochemical Society*, vol. 164, no. 7, pp. F790–F803, 2017.

[57] C. Zhang, C.-J. Li, G. Zhang, X.-J. Ning, C.-X. Li, H. Liao, and C. Coddet, "Ionic conductivity and its temperature dependence of atmospheric plasma-sprayed yttria stabilized zirconia electrolyte," *Materials Science and Engineering: B*, vol. 137, no. 1, pp. 24–30, 2007.

[58] D. Soltman and V. Subramanian, "Inkjet-printed line morphologies and temperature control of the coffee ring effect," *Langmuir*, vol. 24, no. 5, pp. 2224–2231, 2008.

[59] A. Martínez and B. Rudersdorf, "Ceramic inks," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 12, pp. 357–367, John Wiley & Sons, Ltd, 2022.

[60] B. Paulson, "Dimatix printhead technology," in *Inkjet Printing in Industry* (W. Zapka, ed.), ch. 20, pp. 523–533, John Wiley & Sons, Ltd, 2022.

Acronyms

ADC analog digital converter. 17

API application programming interface. 43

CAD computer aided design. 8, 34

CLI command line interface. 34

CTR current transfer ratio. 16, 21

DAC digital analog converter. 16, 17

DOD drop on demand. 1, 9

DPI dots per inch. 2, 30, 31, 35, 36, 38, 41, 51, 59, 72, 80

EIS electrochemical impedance spectroscopy. vii, 63, 64, 66, 67, 82

FDM fused deposition modeling. 4, 12, 31, 52

FFC flexible flat cable. 11, 82, 86

HAL hardware abstraction layer. 23

IC integrated circuit. 17

KCL Kirchhoff's current law. 15

KVL Kirchhoff's voltage law. 16, 22

LSM lanthanum strontium manganese oxide. 2

MCU microcontroller unit. 12, 17–21, 23, 27, 32, 46, 70, 80

PCB printed circuit board. 11, 13, 16, 17, 22, 52

PI propotional-integral. 27

PWM pulse width modulation. 27, 28, 46

RTOS real time operating system. 23, 24, 70

SEM scanning electron microscope. 2, 51, 57, 58, 60, 64, 67, 72

SMT surface mount technology. 13

SOC solide oxide cell. 2

UI user interface. 43

XRD X-ray diffraction. vii, 62, 82

YSZ yttria stabilized zirconia. vii, 1, 2, 4, 48, 50–52, 56, 59, 62, 64–67, 72, 82

List of Figures

2.1 Photo of the implemented printer. 7
2.2 Printhead mounting and electrical connection. 11
2.3 Xaar 128 driver shield renderings. Front side (a), back side (b) and the completed printhead drive electronics assembly including the NUCLEO development board (c). 12
2.4 Xaar 128 Driver Shield overview. 13
2.5 Overview of the circuits for printhead power supply, waveform control and printhead reset signal. 14
2.6 Simplified regulator circuit. 15
2.7 Optocoupler to enable the piezo drive voltage. 17
2.8 D flip-flop to detect intermittent failures. 18
2.9 Timing diagram for failure detection using a D flip-flop. 19
2.10 Second D flip-flop circuit to monitor the conditions for the printhead reset signal. 20
2.11 Optocoupler circuit to monitor piezo drive voltage. 21
2.12 A schematic view of the implemented ink supply system. 26
2.13 Printhead docked in the capping station: silicone seal (A), 3D-printed flexures (B), tubing (C) connected to the insert, and rotary mechanism (D). 26
2.14 Printing pattern of a track at different DPI and different printhead angles. The angled blue line indicates the printhead nozzles, these dots are printed at the same time. 30
2.15 Encoder/Timing signals during normal printing operation over time. *A* and *B* are the signals from the encoder sensor. The *Position* signal is the MCU internal representation of the position in response to *A* and *B*. *A XOR B* is the result of the logical XOR operator on the A and B signals that is further used as *Trigger* signal for Timer 1. At the *Fire* pulse the printhead fires the nozzles. The constant delay t_p between *Trigger* and *Fire* provides a window for computations. The *Nozzle Data* encodes which nozzles eject at *Fire* and have to be written before the *Fire* pulse. 32
2.16 Print tab: An orange square and a blue QR code model placed on the build plate of the developed user interface. 35
2.17 Auto-generated waveform test patterns, the small black rectangles indicate the photo points (a), one of the photo points recorded after printing (b). 36
2.18 Print plan of a layer of two 8 mm x 8 mm squares (blue and orange). Tracks are shown with red dashed outlines. The blue dotted outlined suqares mark substrate holder positions. 37

2.19 Illustration of the implemented scanline rasterization algorithm, showing the scanlines at the angle of the printhead. 38
2.20 Rasterization of a TU Wien Logo. 39
2.21 Simulation displaying rasterization of multiple layers on top of each other. After 10 layers, the drops of the different layers already overlap significantly. 40
2.22 Drop overlap in print (y) direction, indicated by blue and orange dots, is generated by moving the track in move (x) direction for subsequent layers. 40
2.23 Tasks that are executed for each layer. 41
2.24 Jetting and strobe synchronization signals over time. The printhead is requested to fire, and after a delay t_{delay} the strobe light is turned on for the duration of t_{light}. 46
2.25 Dropwatcher image of the Xaar Irix (40 pl, wetting nozzle plate) jetting, $t_{light} = 2\,\mu s$, $t_{delay} = 106\,\mu s$. The firing cycle split into three groups is clearly visible. 46

3.1 Jetting pure ethanol showing satellite droplets. 48
3.2 Viscosity measurements of ethanol/terpineol mixtures as a function of ethanol content. 49
3.3 Uneven pattern on the surface when drying between layers is too short. 49
3.4 Clock sweep using the Xaar Irix Core 40 pl printhead and a reference ink. . . . 50
3.5 Voltage sweep using the Xaar Irix Core 40 pl printhead and a reference ink. . . 50
3.6 Part of the nozzle plate of the Xaar 128, imaged from the bottom, showing the nozzles (circles) and the nozzle chambers. Ceramic particles sedimentation can be seen inside the nozzle chambers. 52
3.7 Calibration card to detect blocked nozzles. 53
3.8 Compensation of blocked nozzles: the outlines and dots in green show the correction pass required to compensate for missing jets in the first pass. 53
3.9 Nozzle test screen of the software: A small calibration grid is printed using each nozzle as shown. The user can select which nozzles should be used to continue the print. Nozzle 2, 4, 7, and 12 show satellite drops, 17-19, 23, 25, and 27 are blocked completely, 21 and 24 are misfiring heavily, and 15, 20, 22, and 26 show missing jets. 54
3.10 Sintering profiles. 54
3.11 Samples 9 and 10 before sintering. (a) Overall image of sample 9 with visible cracks on the entire surface. (b) By zooming in one can examine a crack and also the arrangement of particle agglomerates. (c) Sample 10 including dimensions, and two dust particles marked with circles. (d) The platinum layer is revealed at scratches, and the deposit formed a fine and a rough structured layer on top of each other. 55
3.12 A stray drop of sample 10 allows the inspection of the solid content (ink C) and nature of particles/agglomerates. 56
3.13 Sample 10 was severly damaged during sintering (a), the detached printed layer can be seen in the close up image (b). 57
3.14 A stray droplet from the unsintered sample 1 allows to inspect the size of particle agglomerates, which is in the range of 200 nm to 900 nm. 58
3.15 Top view of sample 1 (a) and sintered sample 2 (b). The length of the scale bar in the cutout in the upper right corner is 3 µm. 58
3.16 Top view of sample 3 (after sintering), where the single crystal substrate is only visible through a defect in the printed layer. 59
3.17 Top view of sample 9 after sintering (a) and cross section (b) showing a high porosity. 59
3.18 Sample 5 overview imaged using optical microscopy (a). The result of a 3D map scan using the profilometer (b). 60

3.19 In the top view of the scan (a), an X and a Y line are indicated by blue arrows. The profiles at these lines are shown in the plots in (b). 60
3.20 Rasterization of a layer partially superimposed on the microscopy image of the printed object. 61
3.21 XRD diffractogram of sample 9 compared to YSZ (PDF#00-030-1468) [53] with a cubic structure. 62
3.22 XRD diffractogram of sample 9 compared to YSZ (PDF#01-070-4431) [54] with a tetragonal structure. 62
3.23 Sketch of the setup for in plane conductivity measurements, showing the length d and the cross-sectional surface A. 63
3.24 Equivalent circuit model to describe bulk and grain boundary conductivity. . . 63
3.25 EIS measurements of sample 7 at four different temperatures. The symbols represent the actual measurements and the lines show the curves of the fitted model. The arrows point to the measurements at 5 Hz. 64
3.26 Arrhenius plot of the calculated conductivities together with the measurement results in [56]. The line fitted through the data points is parallel to the one from literature but shows an offset. 65
3.27 Sample 10 showing the measured microelectrodes in (a) and the setup in (b). . 66
3.28 Cross plane impedance of microelectrode 3 of sample 10, showing low resistance at low frequencies indicating a short circuit. 67

A.1 Xaar 128 Driver Schematics Overview. 83
A.2 Xaar 128 Driver regulator block (*Regulator* in Fig. A.1). 84
A.3 Xaar 128 Driver piezo drive voltage supply block (*HV Supply* in Fig. A.1). . . 84
A.4 Xaar 128 Driver printhead reset signal block (*Reset Circuit* in Fig. A.1). 85
A.5 Xaar 128 Driver microcontroller block (*MCU* in Fig. A.1). 85
A.6 Xaar 128 Driver level shift block (*Level Shift* in Fig. A.1). 86
A.7 Xaar 128 Driver FFC connector block (*Printhead Connector* in Fig. A.1). . . . 86
A.8 Xaar 128 Driver strobe block (*Strobe* in Fig. A.1). 87
A.9 Xaar 128 Driver IO block (*IO* in Fig. A.1). 87

List of Tables

3.1 Ink compositions used to print the samples. 48
3.2 Samples & major printing parameters. 51
3.3 Fitted values R_1 and R_2 and conductivity at four different temperatures 65

Appendix A

Xaar 128 Driver Shield Schematics

A.1 Overview

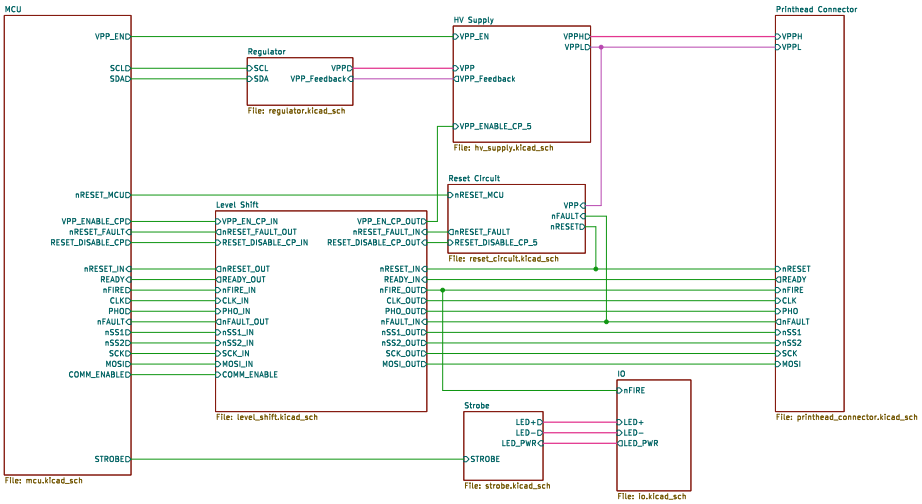


Figure A.1: Xaar 128 Driver Schematics Overview.

A.2 Regulator

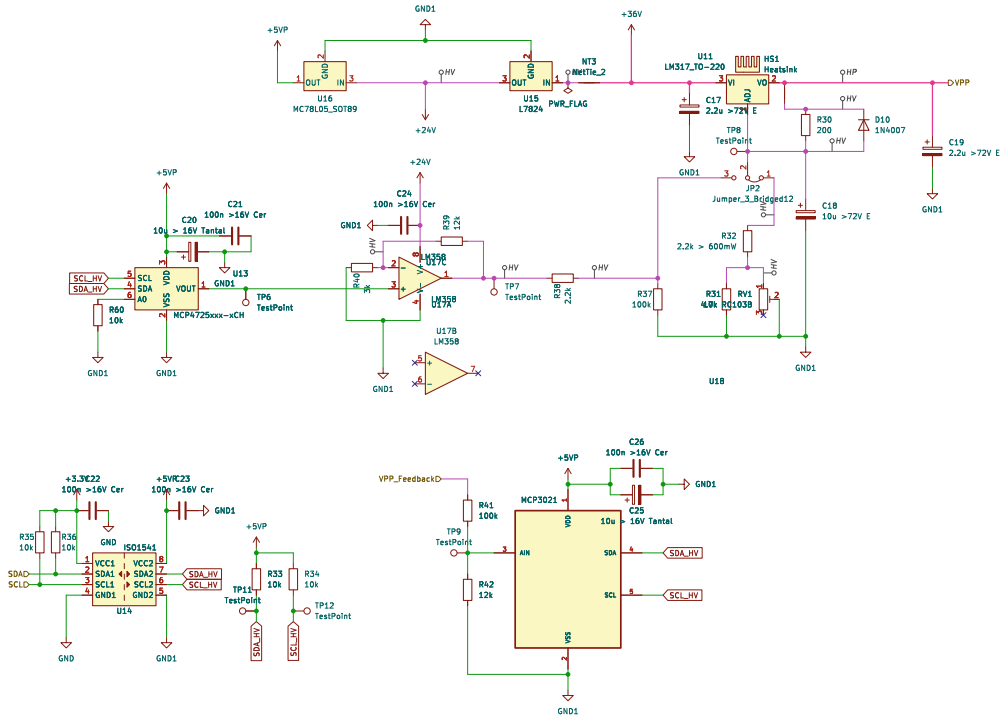


Figure A.2: Xaar 128 Driver regulator block (*Regulator* in Fig. A.1).

A.3 Printhead piezo drive voltage supply

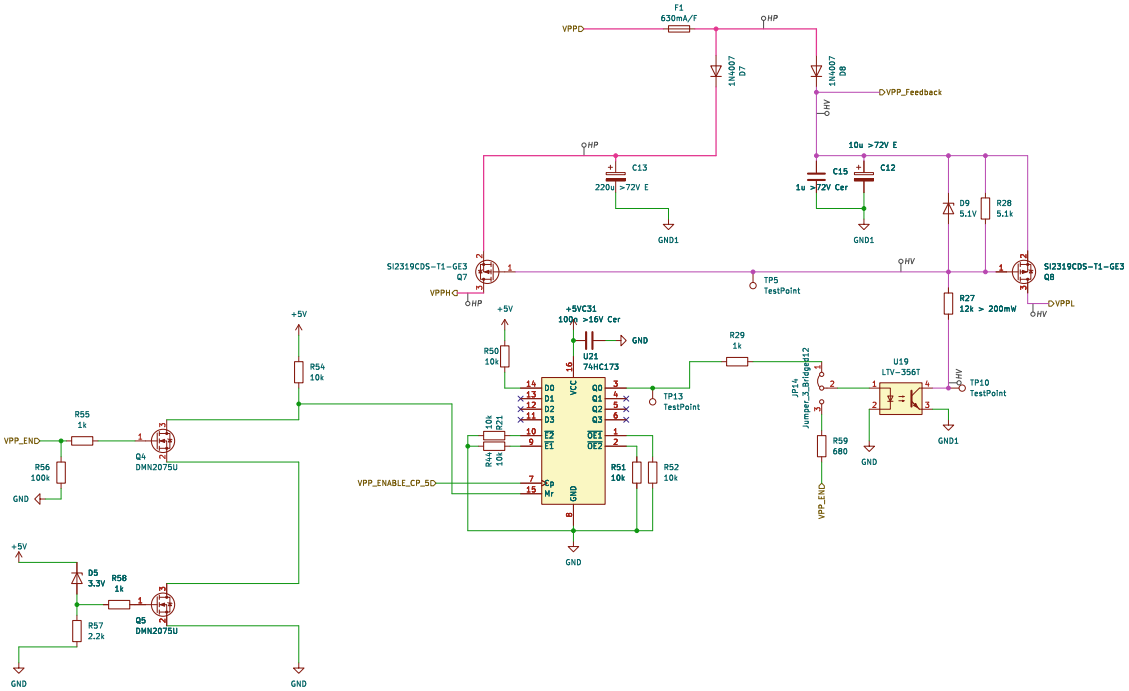


Figure A.3: Xaar 128 Driver piezo drive voltage supply block (*HV Supply* in Fig. A.1).

A.4 Printhead reset signal

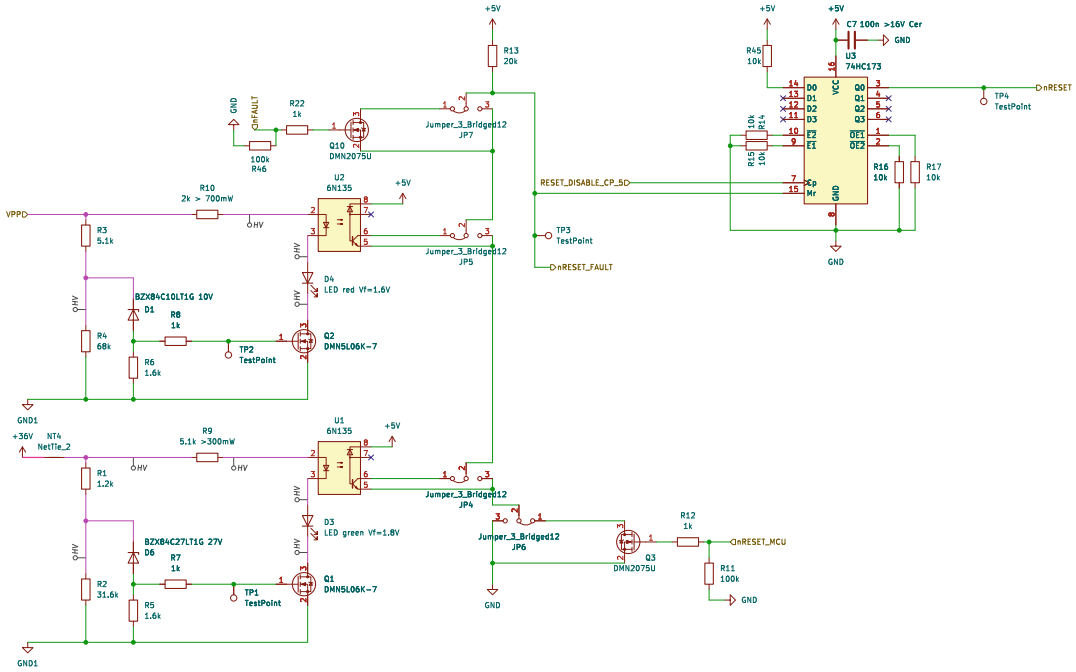


Figure A.4: Xaar 128 Driver printhead reset signal block (*Reset Circuit* in Fig. A.1).

A.5 Microcontroller

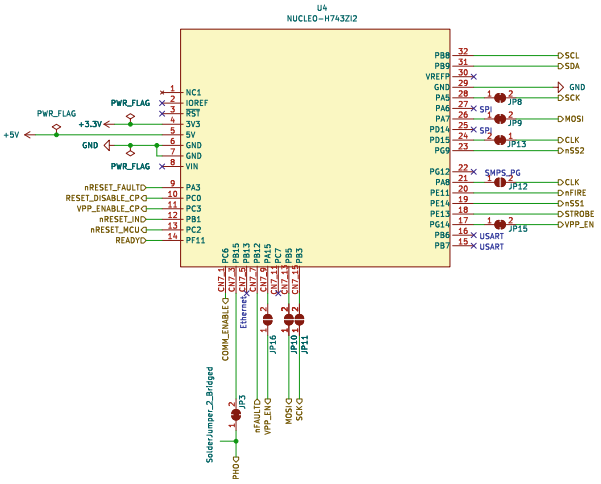


Figure A.5: Xaar 128 Driver microcontroller block (*MCU* in Fig. A.1).

A.6 Level Shift

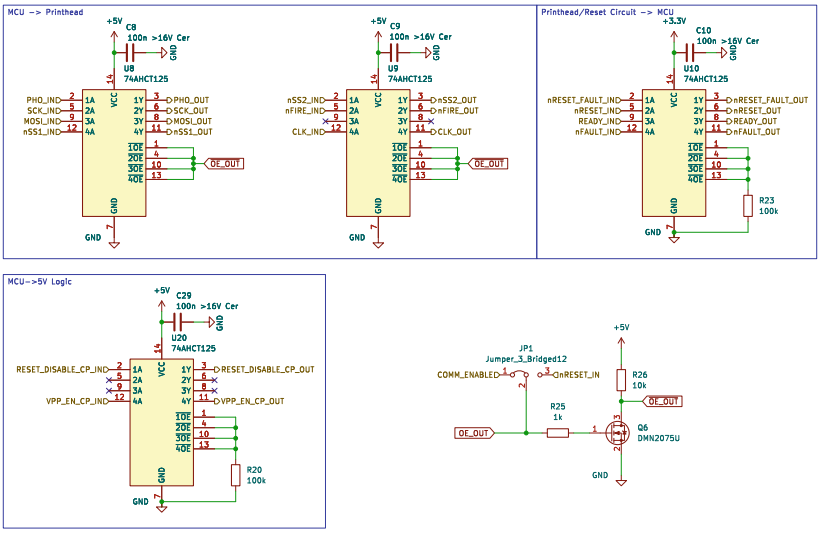


Figure A.6: Xaar 128 Driver level shift block (*Level Shift* in Fig. A.1).

A.7 FFC connector

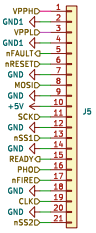


Figure A.7: Xaar 128 Driver FFC connector block (*Printhead Connector* in Fig. A.1).

A.8 Strobe

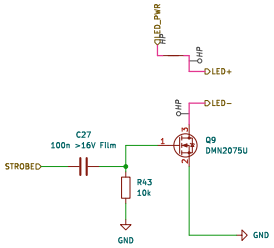


Figure A.8: Xaar 128 Driver strobe block (*Strobe* in Fig. A.1).

A.9 IO

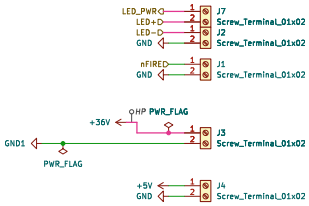


Figure A.9: Xaar 128 Driver IO block (*IO* in Fig. A.1).

Appendix B

Equations

B.1 Ohnesorge number

The ohnesorge number is used as a guideline to assess printability of an ink, and calculated as shown in eqn. B.1.

$$Oh = \frac{\eta}{\sqrt{\sigma \rho d}} \tag{B.1}$$

η is the dynamic viscosity, σ the surface tension, ρ the density, and for d, the nozzle diameter is used. The dimensionless value should be between 0.1 and 1 for an ink to be printable. [28]