UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Richard Erelt

Remote monitoring and data collection for 3D printing

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisor:

MSc Laura Viidik

Tartu 2021

Remote monitoring and data collection for 3D printing

Abstract:

This paper gives a comprehensive guide on how to build a custom 3D printing monitoring system. It provides an overview of common failures occurring during 3D printing. It also maps out the available 3D printer monitoring systems.

Keywords:

3D printing, data collection, remote monitoring

CERCS:

T125 Automation, robotics, control engineering

3D printimise kaudne jälgimine ja andmete kogumine

Lühikokkuvõte:

See töö kirjeldab põhjalikult, kuidas ehitada 3D printerile kaudne jälgimissüsteem, annab ülevaate levinumatest vigadest 3D printimisel ja kaardistab juba saadaolevad 3D printerite jälgimissüsteemid.

Võtmesõnad:

3D printimine, andmete kogumine, kaudne jälgimine

CERCS:

T125 Automatiseerimine, robootika, control engineering

INTRODUCTION

3D printing has gained a lot of popularity in the last decade. Printers have become affordable and relatively easy to use. However, 3D printing still has a long way to go before it can be considered a reliable means of production. Many mistakes can happen during setting up the print and also during printing. In a lot of cases, this leads to an incorrectly printed object already during the printing process. If the print fails, the printer should be stopped as soon as possible to avoid further wasting of materials and other resources such as electricity.

The problem is that due to the long time that 3D printing takes, most users tend to leave the room that the printer is in. This however necessitates the need to constantly go back to check on the printer. If the printer is not checked periodically, the user risks missing the critical failure point and therefore wasting resources on an already failed print. This also becomes critical when testing new materials for printing, where suitable printing conditions need to be developed. The solution for this problem is to use a remote monitoring system.

In this work, I will take a look at why 3D printing failures happen and map out existing solutions that are available for remote monitoring. Based on this knowledge I will build a custom monitoring system. This system has several advantages over the existing solutions, is relatively low cost, and can be replicated by people with no prior knowledge about this kind of monitoring system.

1 LITERATURE REVIEW

1.1 Most common 3D printing technologies

3D printing is an additive manufacturing technique allowing the layer-by-layer production of previously designed objects. During the last decade 3D printing has become a lot more popular than ever before. In 2015 the ISO/ASTM 52900 standard was created. The aim was to standardize terminology and classify 3D printer types. Currently, there are seven different categories of additive manufacturing processes, from which we get eleven distinct types of 3D printing methods. These categories differ from each other by the material type and layer formation method. Notable 3D printing categories are Material Extrusion, Vat Polymerization, and Powder Bed Fusion.[1]

The most common category of 3D printing is Material Extrusion (Figure 1) [1]. It is a process where a filament of solid thermoplastic material is squeezed through a heated nozzle and is melted in the process [1,2]. Then the printer lays the material on a build platform, where the molten filament cools and solidifies, forming a solid object [1,2]. Fused Deposition Modelling (FDM) is a type of Material Extrusion [1,2]. FDM printers are cheap and readily available [1,3]. They work by having a motor that feeds the spool of filament through a heated nozzle [1,2]. Then the extrusion head is moved to predetermined coordinates, and a precise amount of the material is pushed out of the nozzle onto a build plate, where it cools and solidifies [1,2]. After a layer is completed, the extrusion head is raised, and the new layer is laid on top of the last one [1]. The process is repeated until the object is fully printed [1].

Material extrusion allows using a range of different thermoplastic filaments, such as polylactic acid (PLA), acrylonitrile butadiene styrene (ABS), and thermoplastic polyurethane (TPU) [1,3,4]. PLA is the most extensively applied filament in FDM printing [4]. PLA is easy to print with because it has a low melting point, is nontoxic, non-irritating, and also biodegradable [4]. Polylactic acid allows good print quality and best detail reproduction of the typical 3D printing filaments [4]. It is also available in many different colors and blends [4,5]. PLA is constantly being improved by modifying its properties [4]. Modifying PLA is really promising for 3D printing from prototyping tool into a robust and functional manufacturing technique [4].

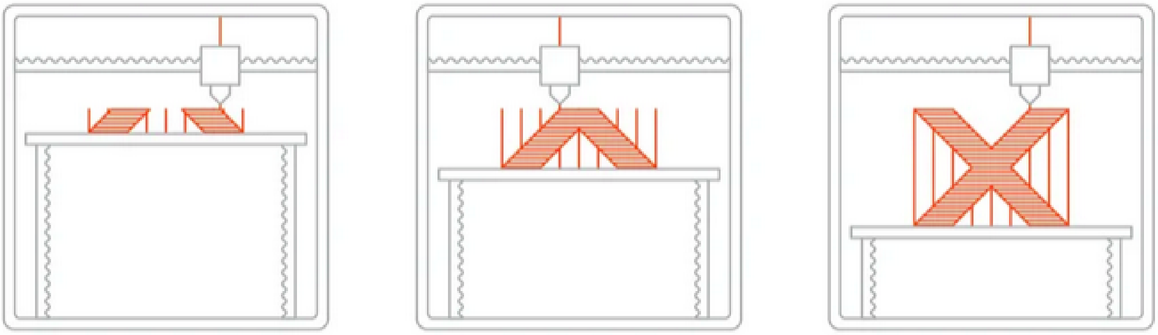

Figure 1. Schematic of Material extrusion printer.[1]

Vat Polymerization is a 3D printing process where UV light is used to selectively cure a photopolymer resin (Figure 2) [1,6]. The most common types of Vat Polymerization are SLA (Stereolithography) and DLP (Digital Light Processing) [1]. The main difference between SLA and DLP is the light source they use for curing the resin [1]. SLA printers use a point laser [1]. However, the DLP printer uses a digital light projector to flash an entire image of each layer simultaneously [1]. Therefore, DLP printers have faster print times than SLA printers [1].

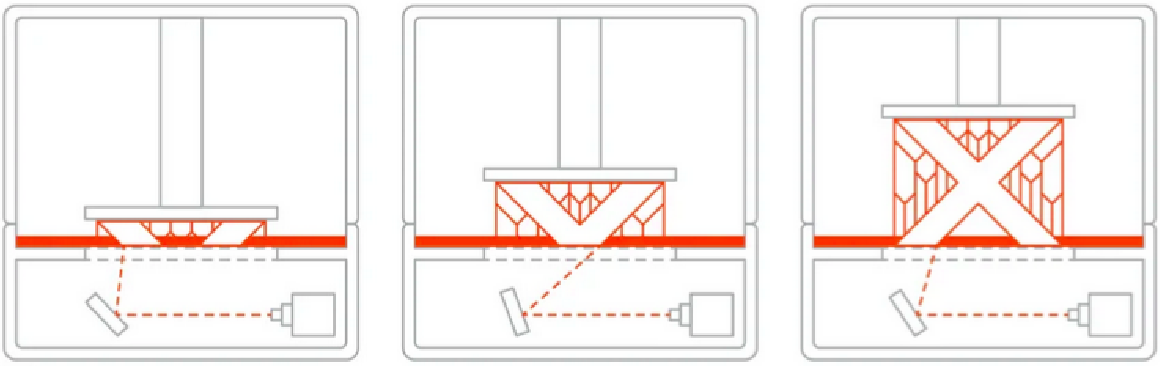

Figure 2. Schematic of Vat Polymerization printer.[1]

Powder Bed Fusion is a type of 3D printing where a thermal energy source is used to selectively induce fusion between powder particles (Figure 3). A lot of Powder Bed Fusion printers use a mechanism to add powder continuously while the object is being printed. Therefore the finished item is surrounded and supported in unused powder. This type of 3D printing allows printing with both thermoplastic and metal powders, such as nylon, aluminum, and stainless steel.[1]

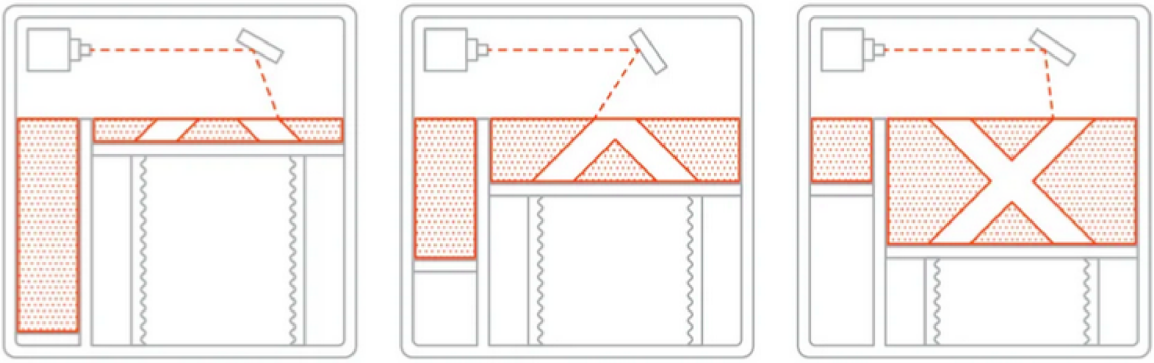

Figure 3. Schematic of Powder Bed Fusion printer.[1]

1.2 Common failures and their prevention that happen during Material Extrusion 3D printing

Material Extrusion is the most common category of 3D printing [1]. Since this work is also focused on this method, the most common failures and ways to prevent them during the FDM process are analyzed. The most common failures include running out of filament, filament breaking, filament stripping, blocked nozzle, printed object coming loose and the printed object warping.

Running out of filament while printing is a problem that should be unmissable in many printers, such as Ender 3 [7,8]. However, other printers like Cel Robox have the filament hidden behind the printer, and because of that, running out of filament can happen [7,8]. While some 3D printers feature built-in failsafe software that warns the user if the filament reel is getting close to running out, most printers have no way of detecting the amount of filament [7]. If an FDM printer runs out of material, then the print will stop halfway and will be not be usable in most cases [7,8,9].

From my experience, running out of filament is especially a problem when you are not constantly monitoring the printer while printing. Current 3D printing software is good at estimating the amount of filament used for a print. However, it is not easy to assess the amount of filament left in the spool visually. When trying to use the entire filament roll to reduce wasted material, estimation mistakes can happen, and filament can run out before finishing the print.

The filament can also break [7,8]. This is mainly caused by an old or cheap filament, but also by wrong storing conditions [7,8]. When storing filament improperly, for example, in direct sunlight, it can become brittle [7,8]. Breaking can also be caused by varying filament diameter. Idler tension can become too tight and cause the filament to snap [7,8].

In addition to running out and snapping, the filament can also strip or slip [7,9,10]. This could happen because of several reasons, such as a loose idler tensioner, wrong hot end temperature, or blocked nozzle [7,9,10]. This problem can happen at any point in the printing process [7]. Most 3D printers use a toothed gear to push the filament through the hot end [7,9]. Usually, the toothed gear moves the filament forward, but if the gear does not have enough grip, it will start to spin [7,9,10]. This results in the gear wearing the filament away, and instead of moving, the filament will stay in one place [7,9,10].

A blocked nozzle is a problem that happens due to a small piece of filament left into the nozzle after swapping filament reels [7]. This occurs when removing filament from the printer [7]. A small part from the end of the filament breaks off and remains in nozzle [7]. When that happens, the new filament cannot be appropriately pushed through [7]. This often results in minor defects in the printed object, such as dark filament, a small amount of missing extruded filament, or print quality changes [7,9]. However, in some cases, it can lead to clogging the nozzle and therefore being unable to continue the print [7,9].

One of the common problems when printing is that the print can become loose from the print platform [7,10]. This happens when the print does not correctly adhere to the printing surface [7,10]. The printed object can break free at any time during the printing process [7].

A similar problem to the print becoming loose is warping [7,8,10]. Warping is caused by the characteristics of the materials used to print with, common with PLA [7,8], ABS [7,8,10], and Nylon [10]. When the plastic cools, it contracts slightly. This causes the edges of the printed object to bend upwards away from the print platform [7,8,10].

In addition to the aforementioned problems, there are a lot of other issues that can happen during 3D printing with FDM printers [7,8,9,10].

1.3 Existing solutions for monitoring and data collection of 3D printer

There have been attempts to make an easy and inexpensive monitoring system for FDM 3D printers [11]. Usually, remote monitoring systems can be divided into two different categories: monitoring systems that transmit video feed of the printer [11] and monitoring systems that transmit the printer controlling software [12]. However, there are some solutions that combine both video feed and software control (Figure 4) [13,14,15,16].

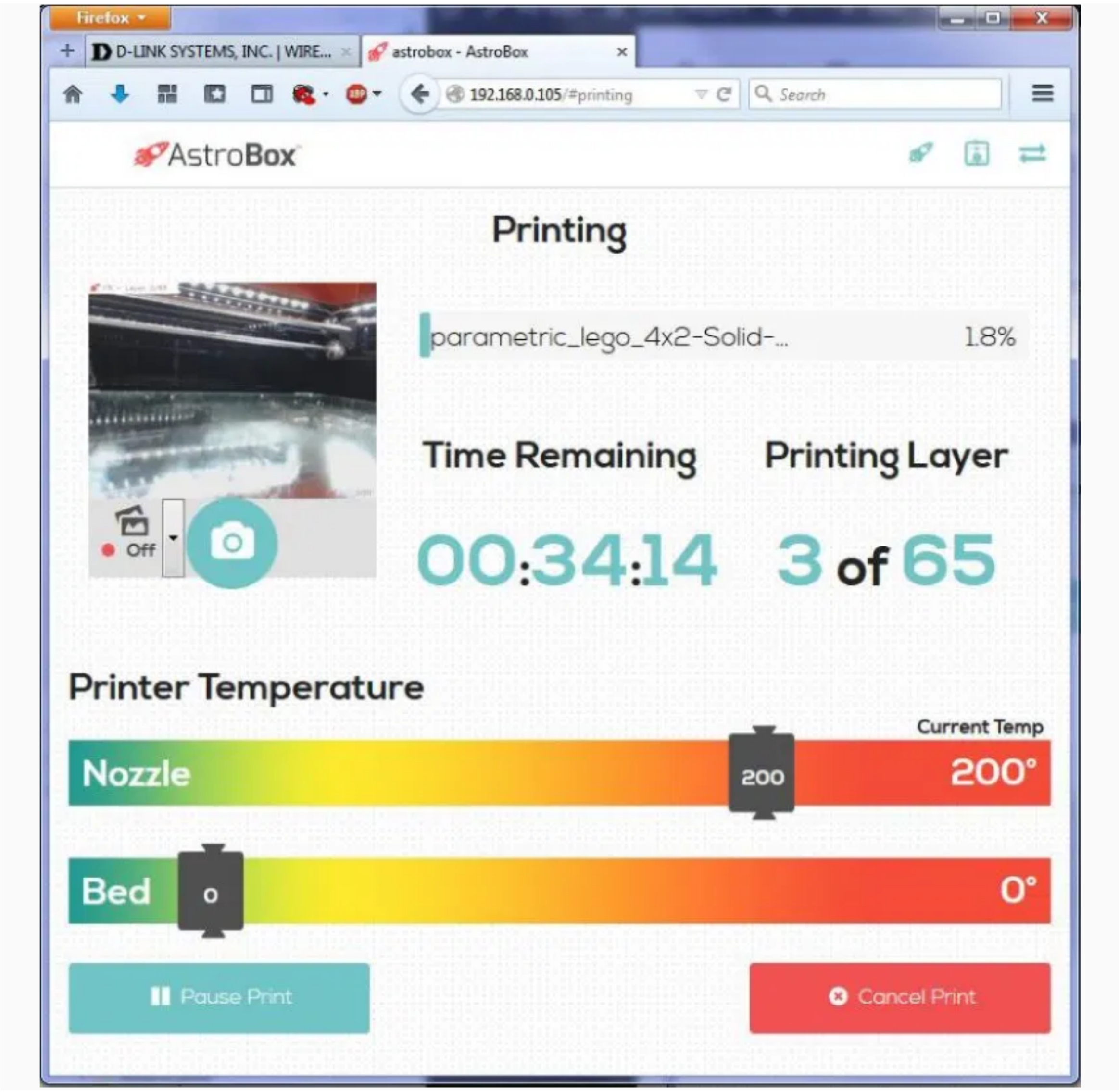

Figure 4. AstroBox software that allows to transmit video feed and to control the 3D printer remotely. [13]

One of the main problems with this kind of monitoring system is that, while they can provide video feed and information from the printer's built-in sensors, they are limited on the capability of adding task-specific sensors and hardware [15], such as infrared temperature sensors [17]. While there are some companies [18], that offer this kind of customizability, they tend to be expensive and impractical for relatively cheap 3D printers [19].

I will now quickly go over some of the solutions that are currently available. The focus will be on the price, ease of usage, and the ability to customize the control system to different 3D printers and monitoring requirements.

AstroBox (Figure 4) is a software made by AstroPrint solution that allows both remote printing and live monitoring in addition to other features [13,20]. However, it seems to be focused more on business [21] and school [22] environments, where are multiple printers or users. AstroBox is available with both free and premium plans for most 3D printers [13,23]. AstroBox requires a Raspberry Pi [24] to be used with every connected printer [13,21]. AstroBox is customizable in the sense that it runs on Raspberry Pi [13,21], but the software is itself limited in the capability of adding additional sensors, therefore requiring to run different software [13].

OctoPrint is software that AstroBox was based on (Figure 5) [13]. Similar to AstroBox, OctoPrint allows both remote printing and live monitoring and can be used on a variety of 3D printers [16,25]. OctoPrint is entirely open-sourced, with no payments required [16,25]. It is mainly used on Raspberry Pi [24]. However, it can be run on almost all computers or phones [16,25]. There is also a special Raspberry Pi [24] operating system called OctoPi [25,26]. OctoPi includes the OctoPrint software and is tweaked to get maximum printing performance out of Raspberry Pi [24,25,26].

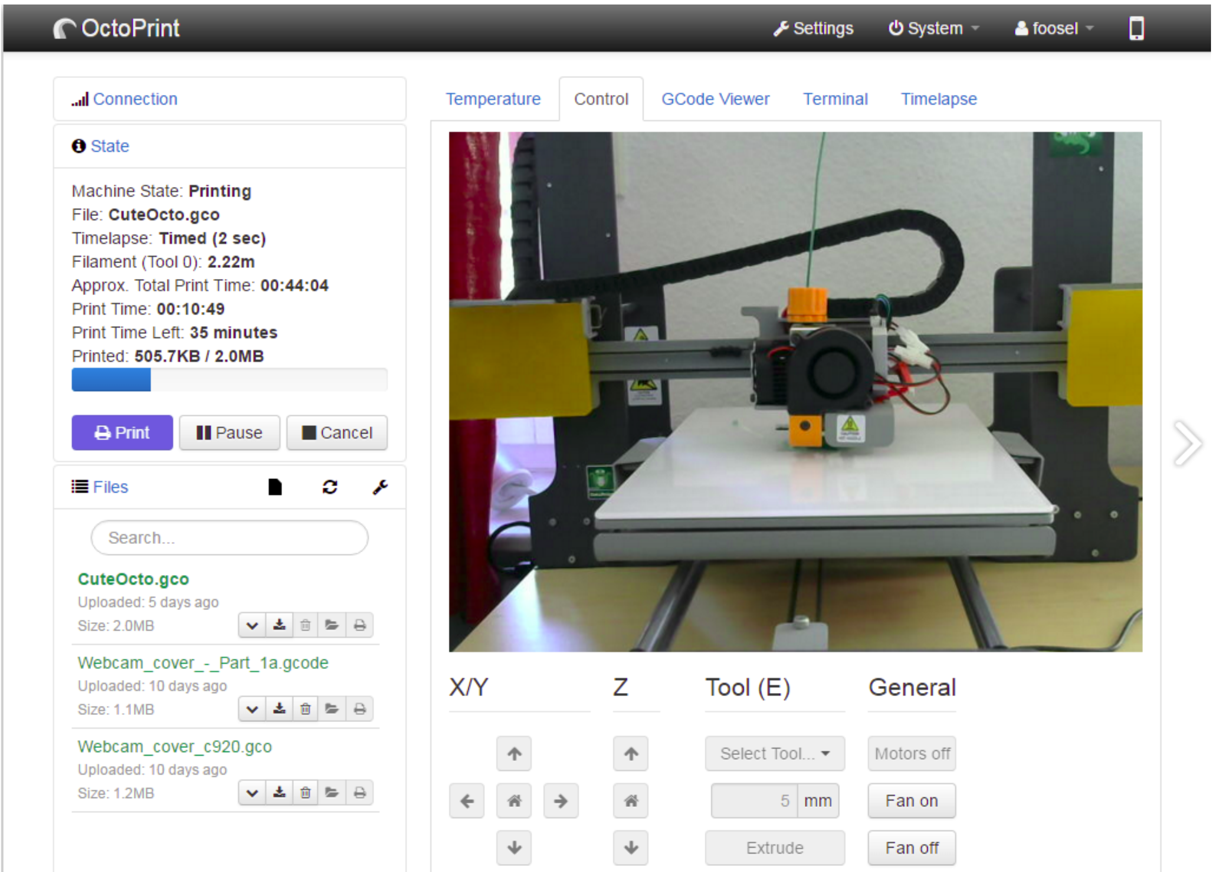

Figure 5. OctoPrint software. [16]

1.4 Possible ways of making a custom 3D printer monitoring system

When making a custom monitoring system for a 3D printer, many different aspects should be considered. First, a budget should be set because that determines the other options and solutions. Secondly, should be considered, what capabilities are necessary, what capabilities are nice to have, and what is not required. Thirdly, complexity and ease of use should be taken into account. This work focuses on making an easy and inexpensive remote monitoring and data collection system. Therefore, the budget is set to be as low as possible, with the goal of reusing old and already purchased hardware. Wherever possible, free or open-source software will be used. Also, possible future upgrades and additions should be easily addable.

1.4.1 Options for streaming live video feed

Being able to see the live stream from everywhere, on any device, and with minimal effort are crucial. One possible solution for that is to use a camera to record the live video feed and then use software to stream the video on a streaming platform. Another solution would be to use virtual network computing (VNC). However, this method is more complicated and can not be easily used on phones. Now I will take a look at some streaming software and streaming platforms.

There are plenty of free software for live streaming, and one of the most popular ones is Open Broadcast Studio (OBS) [27,28,29]. OBS is open-source software for video recording and streaming that is available for Windows, macOS, and Linux [27]. OBS has an excellent and easy-to-use layout both for recording and streaming, which gives complete control of the user's media [29].

Lightstream is a good alternative to OBS [28]. Lightstream is free and browser-based, meaning that it does not require any downloads [28,30]. It supports many popular streaming-related tools and has a lot of adjustability [28,30].

One of the most used streaming platforms is Twitch [31,32]. Twitch is free and is available on many different devices and also on the web [31]. It has many various features and options for streaming live video (Figure 6) [31]. Twitch even offers free software for creating the actual stream called Twitch Studio, which eliminates the need for third-party software, such as OBS [31,33]. However, Twitch Studio is currently in beta testing and is only available for Windows [31,33].

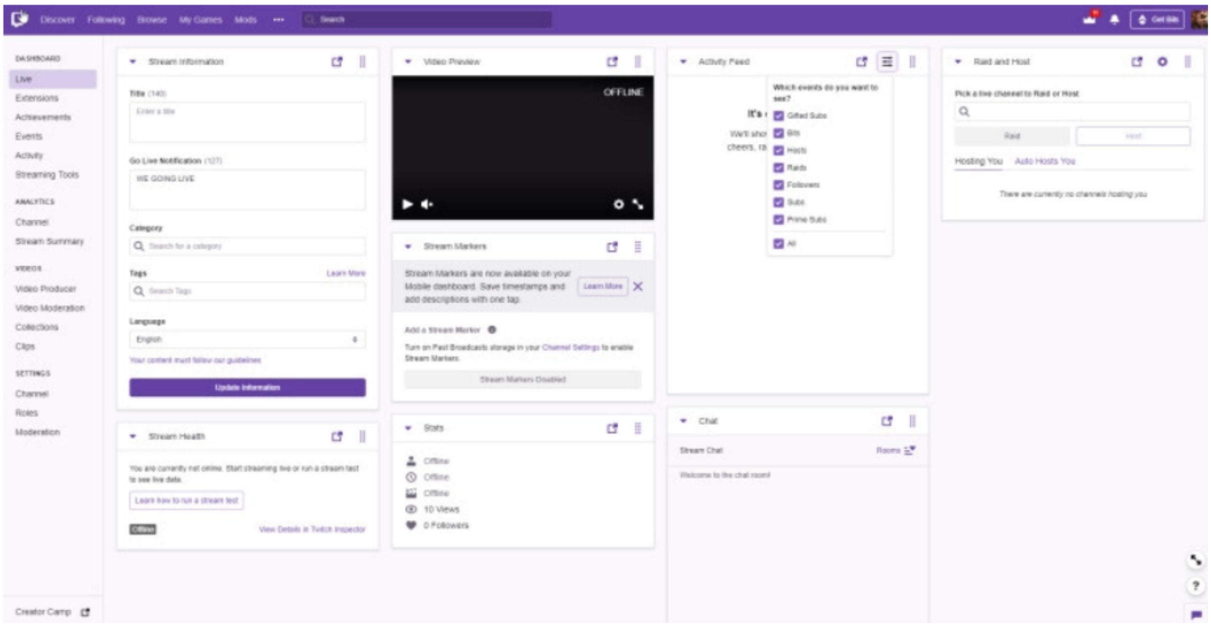

Figure 6. Twitch options for streaming. [31]

YouTube Live is another popular live streaming platform [34]. It is free to use, has an easy set-up, and has similar features to Twitch [34,35]. YouTube Live requires third-party software, such as OBS, to have reasonable control over the streaming quality [35].

1.4.2 Possibilities of adding additional sensors

Having the ability to gather more information about the printing process can be helpful. One way of doing that is by using different sensors, such as the infrared temperature sensors [17]. There are various possibilities for adding additional sensors to the control system. A simple way is to use a sensor that can be connected to the USB port [36]. However, finding a suitable sensor with USB connectivity can be difficult, and the sensors are usually relatively expensive [36,37].

Another possibility is to use an Arduino to connect sensors [38]. This approach allows for a wide range of different sensors for different applications [39]. The available Arduino sensors are usually cheaper than the USB alternatives [39,40]. However, the cost of the Arduino has to be also considered [41]. There are plenty of guides and information on how to set up Arduino with sensors (Figure 7) [42].

In my opinion, Arduino requires some coding and electronics experience and is typically less compact than a sensor with USB connectivity.

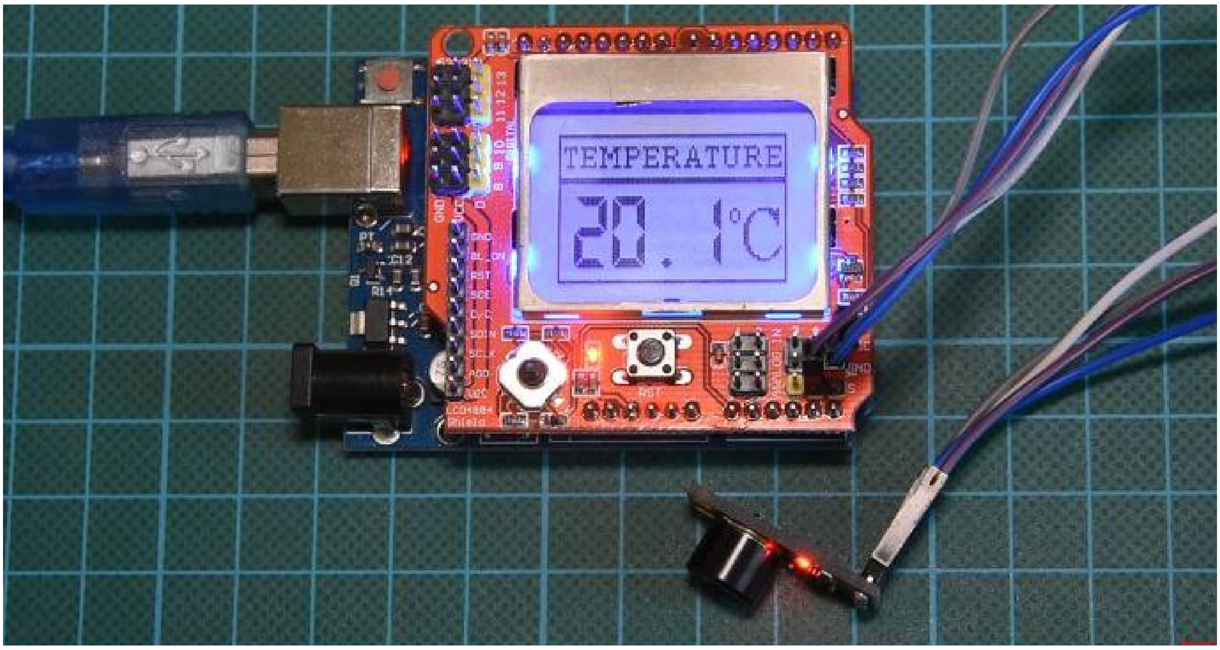

Figure 7. Arduino with IR temperature sensor. [42]

1.5 Detecting 3D printing errors with machine vision

A remote monitoring system allows the observation of a printing process while not being in the same room as the printer. However, this approach still requires checking the camera feed periodically in order to be helpful. There are ways to build an automatic warning system that notifies the user when something goes wrong. One way of doing that is to use different sensors, as mentioned previously, to check if the printing process is working correctly. There has also been research done on implementing machine vision to detect errors in 3D printing [43,44,45]. Even though machine vision will not be implemented in this work, I will briefly go over it to introduce the possibilities for future upgrades for the monitoring system.

Machine vision can be implemented by viewing the side of the object while it is printing and comparing it to the model that was made in 3D software [43,44,45]. The simple and cheap way is to compare the outlines of the object and the model by using OpenCV [44]. Only a regular and readily available camera is needed [44]. However, this method is not reliable [44]. Out of the five classes of proposed error classification, only three were able to be detected [44]. The used algorithm is capable of a 60 to 80 percent failure detection rate, with a false positive detection rate of 60 to 80 percent [44].

To increase the accuracy of the failure detection rate, much more complicated machine vision-based systems can be made. An adaptive algorithm can be further developed, but it is a complex problem (Figure 8). It can compare the object's outer contour but also the inner structures to the G-Gode. However, this requires the printer to be configured and calibrated optimally. Currently, this system is more an intelligent printing suspension tool than a failure correction algorithm. [45]

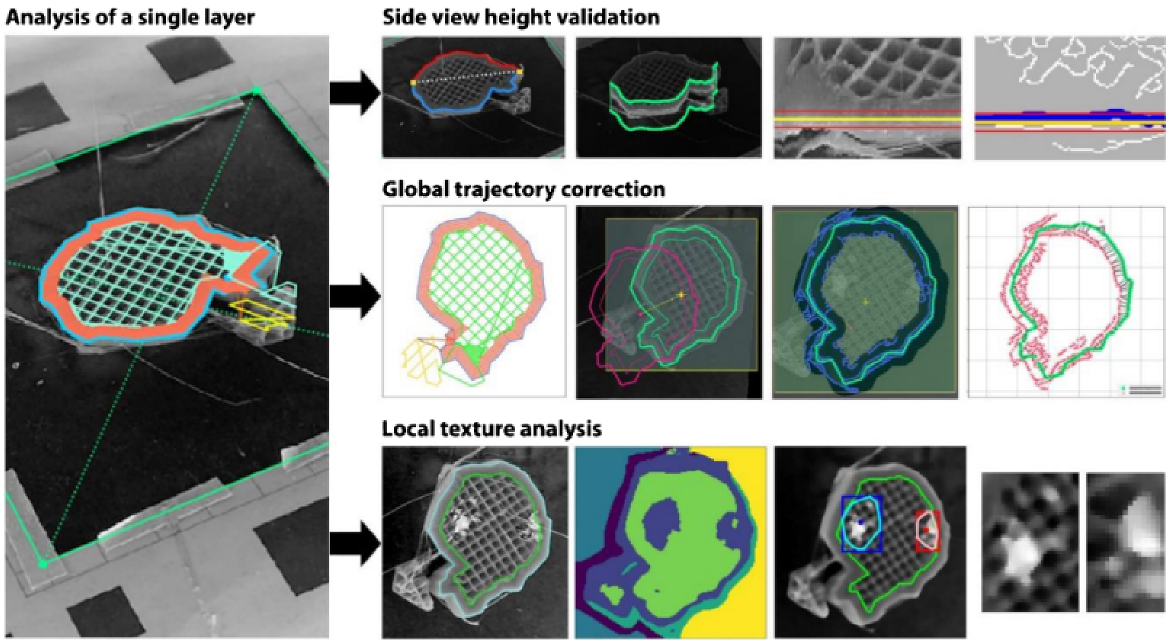

Figure 8. Image processing pipeline for computer vision-based layer-wise algorithm. [45]

2. THE AIMS OF THE THESIS

The aims of the thesis are to:

- Categorize and review the information on monitoring the 3D printing process found in literature;
- Choose and prepare a low price and customizable monitoring/data collection system for 3D printer;
- Evaluate the functionality of the prepared system;

3. EXPERIMENTAL PART

3.1 MATERIALS AND METHODS

3.1.1 Requirements

The requirements for the monitoring system were following:

- Low cost;
- Easy replicability for people with no or minimal prior experience of this kind of systems;
- Ability to stream live video feed;
- Ability to stream 3D printing software stream;
- Ability to measure extruded material temperature accurately;
- Ability to record and save both video feed and the temperature data;
- Ability to quickly set up the monitoring system;
- Ability to easily open and watch the video stream;
- Ability to modify the monitoring system in the future;

3.1.2 Materials

The 3D printer used for building the monitoring system in this thesis was HYREL 3D System 30M. This printer has a multi-head system that allows quickly swapping between all Hyrel print heads. This allows printing with numerous materials, such as plastics, homemade filaments, clays, pastes, hydrogels, etc [46]. This printer was located in the University of Tartu Institute of Pharmacy where it is mostly used for FDM and hydrogel printing. However, it was taken into consideration that the built monitoring system could be incorporated into any commercially available 3D printer..

For the video feed, two computer web cameras were used: Logitech C270 and a non-brand Mini Webcam from Aliexpress. However, for the purpose of recreating a similar monitoring system, most web cameras with USB connectivity will work.

For adding sensors Arduino Nano [41] was used. MLX90614ESF-BCC infrared temperature sensor was used with a lens that has a field of view of 35 degrees [40]. For connecting the Arduino and sensor, mini breadboard, breadboard jumper wires, and micro USB were used.
A computer is required in order to connect web cameras, connect Arduino and also stream and record video feed with sensor data. Dell OptiPlex 3050 with Intel Core I5-7500 was used as a computer. However, for the purpose of recreating a similar monitoring system, most computers with USB interfaces can be used. For the computer, a keyboard, mouse, and monitor were used. The computer also has to be connected to the internet in order to stream video feed and sensor data. Pop!_OS (operating system based on Linux Ubuntu distribution) version 20.04 LTS was used as the operating system for the computer. However, for the purpose of recreating a similar monitoring system, Windows, macOS, and different Linux distributions can be used.

3.1.3 Monitoring system computer setup

The most important part of this monitoring system is the computer. The computer is required in order to connect cameras, sensors and also to create the video stream. For this, I used an available computer. This computer was a Dell OptiPlex 3050 with Intel Core I5-7500, 8GB of RAM, and 256GB SSD (Figure 9). However, any other computer that is powerful enough for video streaming can be used instead. I used Pop!_OS 20.04 LTS as the operating system for this computer. Pop!_OS is the modified version of Ubuntu made by System76. Ubuntu is a Linux distribution. I selected Pop!_OS, because this is the operating system that I'm familiar with, and Linux has good support for Arduino and also free/open-source software. Although, other operating systems, such as Windows or macOS can be used. I recommended using the operating system that the user is most familiar with since it makes solving software issues easier. I clean installed the Pop!_OS 20.04 LTS onto the OptiPlex 3050, did all necessary steps for the operating system setup, and also updated software packages. I connected the keyboard and 1980x1080px monitor to the computer. For a stable internet connection, I used an Ethernet cable that connected to the university network. I also created a standard procedures guideline so people without prior knowledge can set this system up (Appendix I).

3.1.4 Camera setup

For the video feed, I used two web cameras. I already had one camera from the previous usage: Logitech C270. For the second camera, I bought the cheapest web camera from Aliexpress, which happened to be a generic Mini Webcam. Logitech C270 is a 1980x1080px camera and Mini Webcam is 1280x720px camera. I tried different camera placements for the optimal viewing angles, both inside and outside the printer case. Originally I wanted to mount one camera on top of the printer, viewing downwards onto the print plate, however, this camera view was blocked by the print head. The best placement for the C270 ended up being the front of the Hyrel 3D System 30M 3D printer case. For mounting the C270, I used a suction cup that connected to the camera with a connecting arm (Figure 9). I designed the connecting arm in Onshape (3D modeling software), and 3D printed it from ABS on my Creality Ender 3 3D printer. I positioned the second camera to film from the right side of the 3D printer. The best mounting place for this camera was the top of the computer monitor (Figure 9). I mounted the Mini Webcam with double-sided tape. Used software will be covered in chapter 3.1.6.

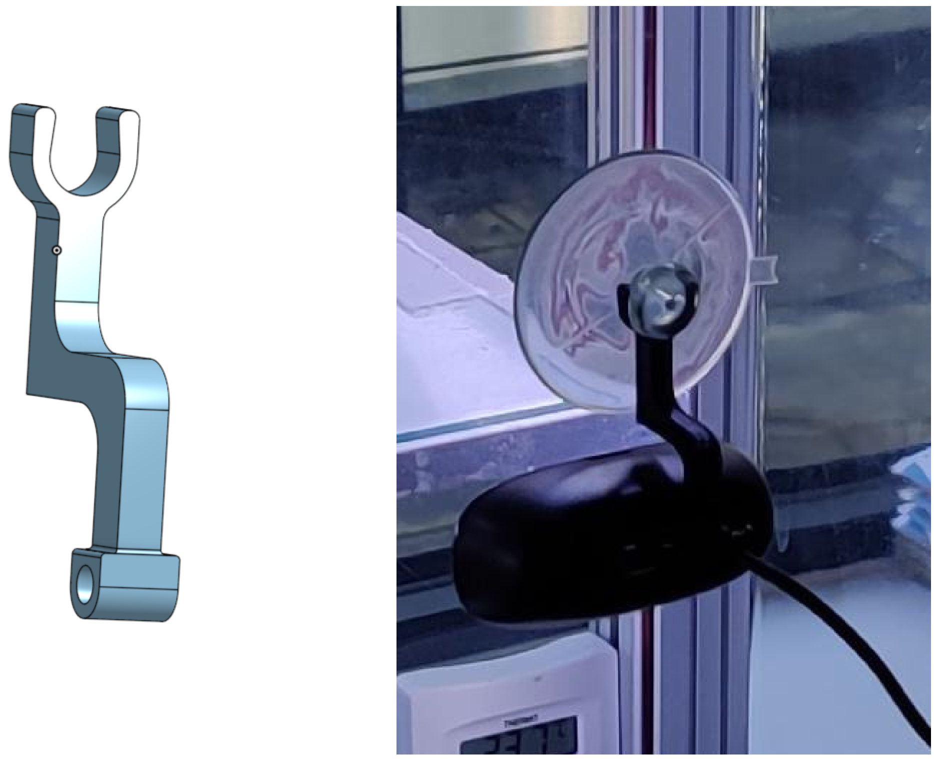

Figure 9. The 3D model of the camera connecting arm and the picture of the mounted camera.

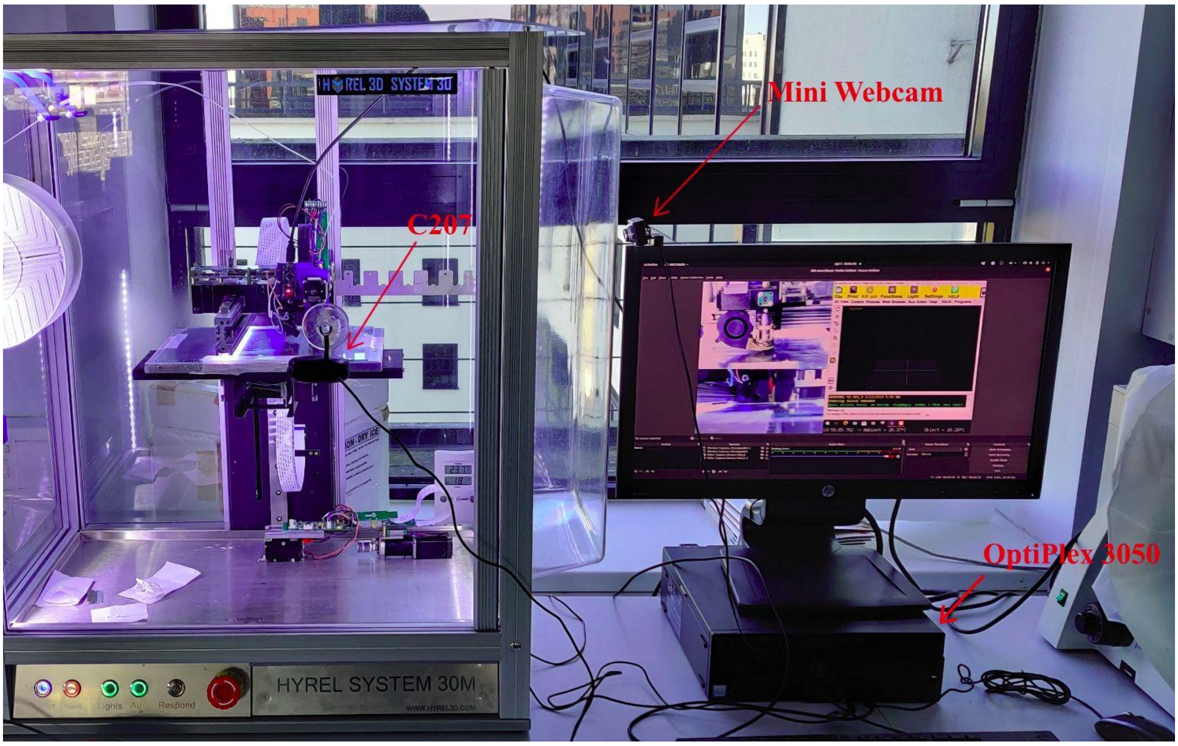

Figure 10. Monitoring system setup.

3.1.5 Adding sensor to the monitoring system

Since the printing temperature has a great effect on the printing quality [7] and also impacts a lot of different printing failures, having the possibility to constantly measure the temperature is important. Therefore, we decided that adding the possibility to measure the temperature of the printed object will give good information about how the printing process is going. For this, I considered many different solutions. I also contacted Karl Gifford from HYREL to get his opinion on how to add a system for measuring temperature. We made a Skype call where we talked about different solutions. He recommended using a thermal camera for this application since it would give accurate information over different spots of the printed object in a quickly understandable visual form. However, I had difficulties finding a suitable camera. Most of the suitable thermal cameras were too expensive and the only reasonably priced camera was out of stock.

Therefore, I decided that the next best solution would be to use an infrared temperature sensor instead of a thermal camera. First, I tried to find an IR temperature sensor with USB connectivity, so it would be easy to integrate into the monitoring system. However, I ran into a similar problem as with the thermal camera. All suitable IR temperature sensors with USB connectivity were too expensive to use on this system. Therefore, I had to use an Arduino to connect the IR temperature sensor. I ended up using Arduino Nano with the MLX90614ESF-BCC infrared temperature sensor. I chose this sensor because it had a low price, it had a suitable working temperature range of -70*C to 380*C and it came with a lens that had a 35-degree field of view (FOV). This sensor was also available with a lens that had a 15-degree FOV, which might even be a better choice for this application, than the 35-degree FOV lens. However, the 35-degree version was much cheaper in terms of price.

For mounting the IR temperature sensor, I tried many different solutions. I designed and 3D printed several mounting brackets for the sensor together with Arduino. I found the best location for the sensor to be mounted on a bracket that can be inserted into a slot on the 3D printer that is meant to hold an additional extruder head. I used Onshape for designing this sensor mount to be adjustable to allow the user to change sensor angle and distance relative to the printed object (Figure 11).

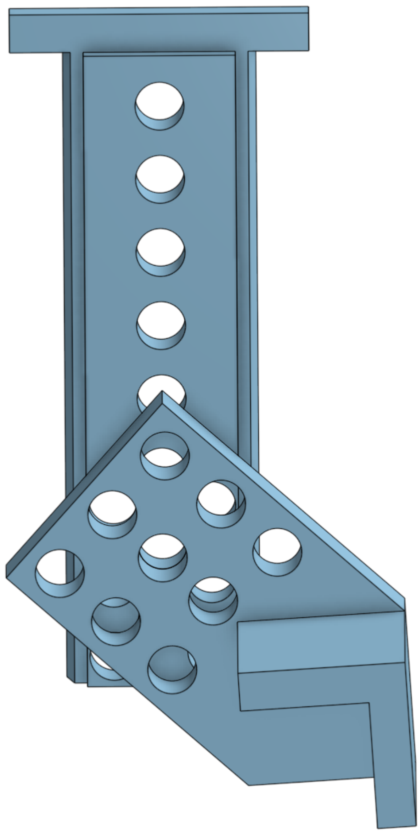

Figure 11. IR temperature sensor mount model in Onshape.

I put Arduino Nano on a mini breadboard and connected the sensor with short breadboard jumper wires. I used double-sided tape to mount Arduino on the same mount as the IR sensor and ran a USB cable from Arduino to the computer (Figure 12). Used software will be covered in chapter 3.1.6.

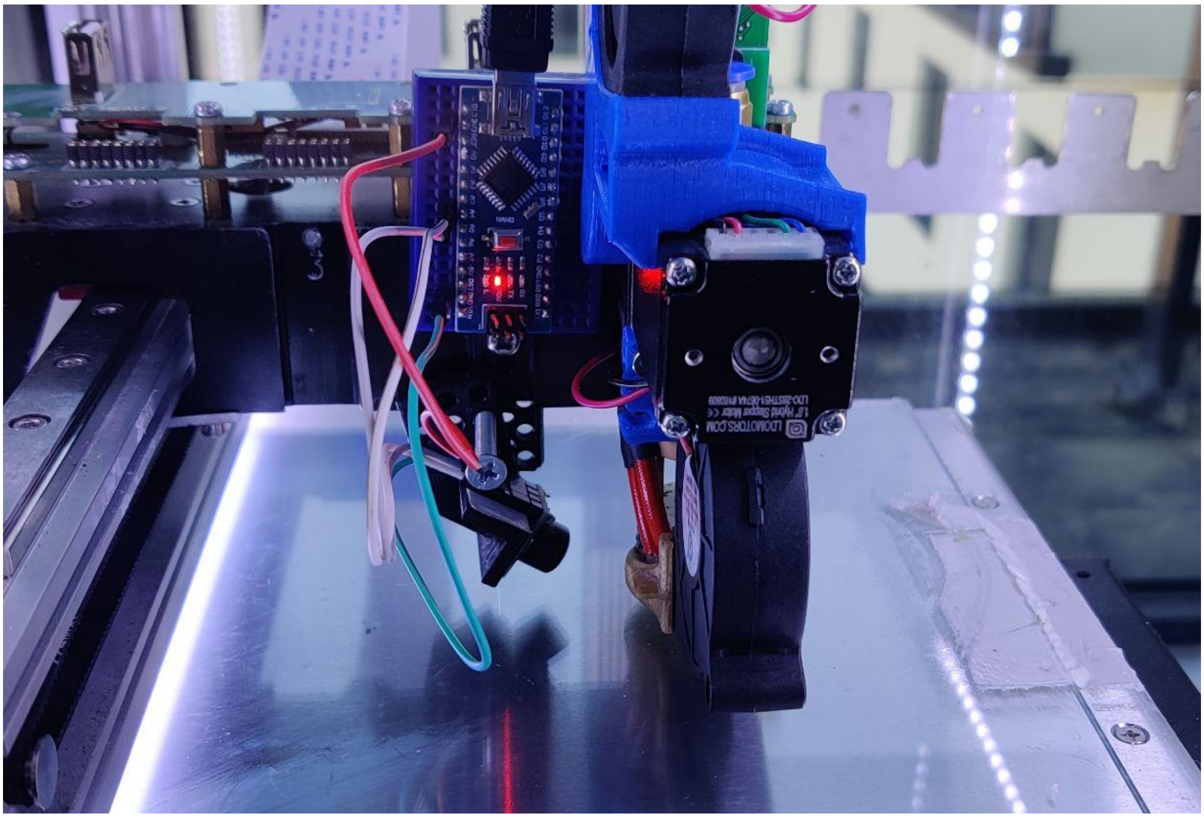

Figure 12. IR temperature sensor with Adruino on a custom mount next to the extruder head.

3.1.6 Used software

For the video streaming software, I decided to use OBS Studio since it is open-source, easy to use, runs on Linux, and has a lot of different options for streaming, including the possibility of saving the video. The camera feed was shown in OBS Studio and configured to a suitable size and location on the stream video (Figure 11).

The System 30M 3D printer has a separate tablet that runs the 3D printing software, however, this computer is not powerful enough to run constant multiple camera and sensor feed. Therefore in order to get 3D printing software live feed into the OptiPlex 3050, I used TeamViewer software. TeamViewer is a VNC software that allows remote controlling another computer over the network.

In order to run the Arduino with an IR sensor, I had to create a code in Arduino IDE (Appendix II). Arduino IDE is also constantly running with Serial Monitor open in order to read the sensor output. Using the Arduino IDE to read sensor output eliminates the need to run a different script just to read the sensor output, therefore making the overall system simpler.

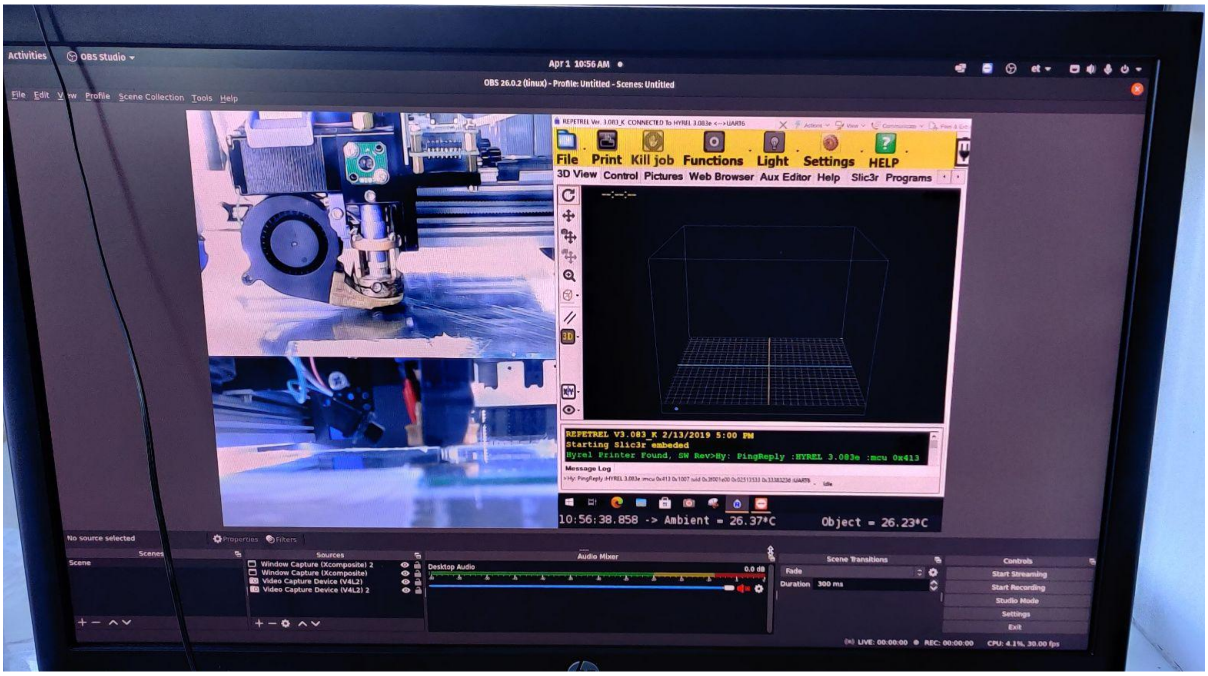

Figure 13. OBS Studio running camera, 3D printing software, and sensor feed.

I decided to use Twitch as the video streaming platform. Twitch was free, easy to set up, and also allowed watching the live feed on any device by using a stream link. The downside of using Twitch is that potentially anyone can watch the video stream.

3.2 RESULTS

3.2.1 Testing the system

During and after the completion of the 3D printer monitoring system I ran tests to evaluate the effectiveness of the system. These tests consisted of setting up the monitoring system, starting a 3D print, and then remotely monitor the printer from a mobile phone. Usually, the system set up takes around 5 minutes, less if the system is already partially set up. This consists of:

- Turning on both the computer running the 3D printer and the monitoring computer;
- Setting up the cameras to proper locations;
- Setting up the IR temperature sensor and connecting Arduino with the computer;
- Connecting computers via the TeamViewer;
- Running Arduino IDE Serial Monitor to get the sensor output;
- Running OBS Studio and configuring the video stream to user preferences;
- Opening Twitch and starting video stream from the preferred Twitch user;

After this setup has been done, the video feed can be seen on a mobile phone or computer by opening the used Twitch user's stream. The stream link can be easily shared and viewed by everyone. A Twitch account is only needed for the streamer but not for the viewer. Both OBS and Twitch allow recording the stream, which is useful for data collection. If any failures were to happen during the printing process, I can rewatch the video stream and get a much better understanding of what went wrong. I can compare the constant temperature sensor readings to the reading gathered while the printer is working properly. This kind of data allows me to change printer settings/configurations in order to avoid future failures. The downside of using Twitch is the fact that everyone who finds it, can also see the stream. However, actually finding this stream without having the stream link of the streamer username is extremely difficult. Together, with the fact that most 3D prints are not information sensitive, it means that having a public stream is not a big problem.

From testing, this custom remote monitoring and data collection system for 3D printing works well. I did not encounter any problems with the video stream (Figure 14). The sensor data helps with failure prevention, by giving a better overview of the extruded material and ambient temperatures.

Figure 14. Mobile phone screenshot of the video stream at the beginning of the print.

From the video stream, we can clearly see the printed object (Figure 15).

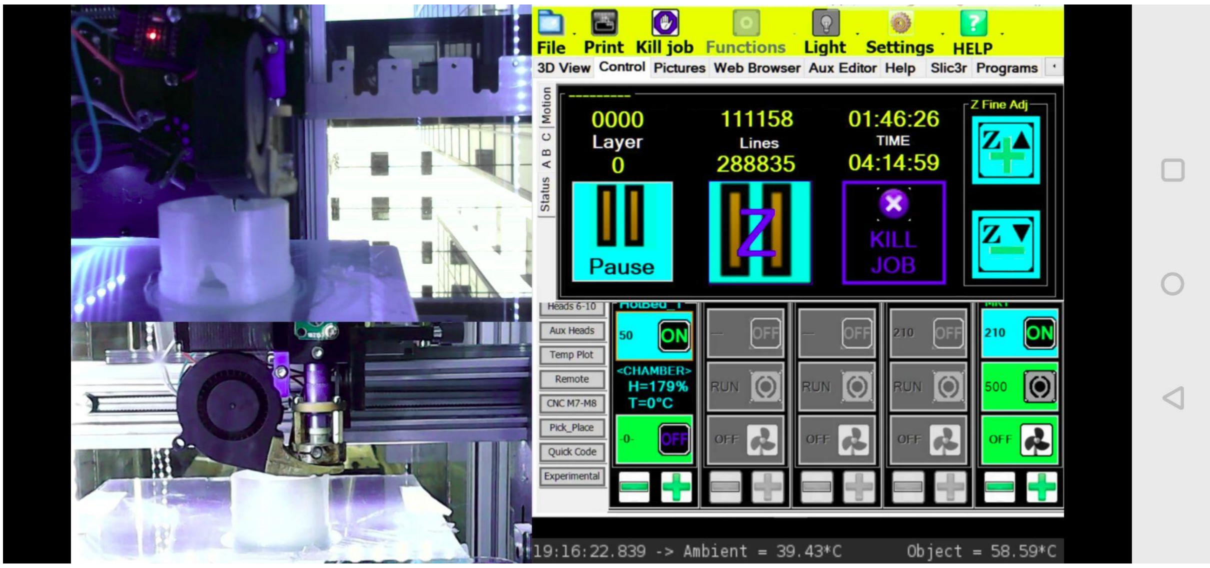

Figure 15. Mobile phone screenshot of the video stream at the middle of the print.

Printing failures can quickly be seen and identified (Figure 16).

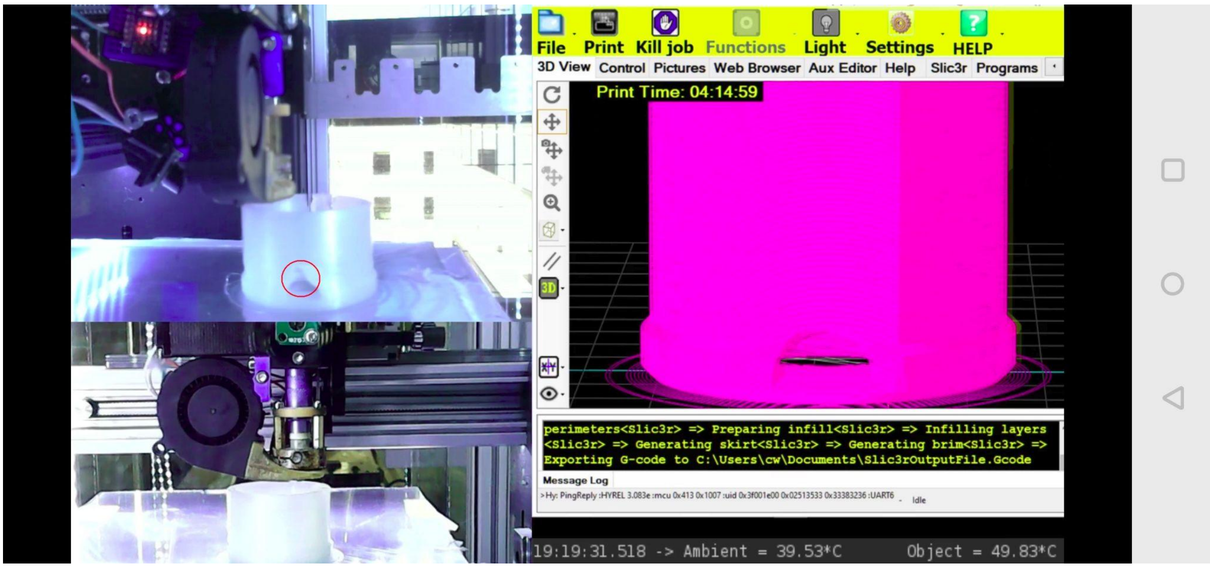

Figure 16. Print failure, marked by a red circle and compared to the original 3D model.

3.2.2 Comparison to the existing solutions

Since one of the goals for my thesis was to explore the existing solutions for remote monitoring of 3D printers, it is important to compare them to my custom system.

Compared to other available remote monitoring systems, my custom system is cheaper and more customizable according to user preferences (Table 1). My system can be built entirely free if using surplus hardware. All the used software are free.

Table 1. Comparison between my custom system, Astrobox and OctoPrint.[20,26]

System:	**My custom system**	**AstroBox**	**OctoPrint**
Price:	Depending on the used hardware. Can be made for free using surplus hardware.	125 euros for Raspberry Pi 3 kit, including software (no cameras or sensors). 39 euros for only the software.	Software is free. The user has to provide the hardware. Made to run on Raspberry Pi.
Functionality:	live camera feed, live 3D printing software feed, IR temperature sensor feed, data	live camera feed, live 3D printing software feed, data collection, controlling the printer	live camera feed, live 3D printing software feed, data collection, controlling the printer

	collection.	from the software.	from the software.
Customizability:	Customizable to the user requirements.	Limited customizability, only what AstroPrint supports.	More customizable than AstroBox but less customizable than my system.
Compatibility:	All printers, also other equipment that needs monitoring.	Most 3D printers.	Most 3D printers.

3.3 DISCUSSION

This type of custom remote monitoring system works well for 3D printers, since it eliminates the need of being in the same room as the printer but still getting information on how the print process is going. It allows the user to easily and quickly check the printing process from a mobile phone and in case of a failure, the user can go to the printer and restart the printing process with better parameters. This system also gathers information that is helpful when optimizing 3D printer for failure prevention.

This custom monitoring system is easy to set up and operate for everybody. This is essential since a lot of people who work with 3D printers do not have prior experience in complicated computer systems. But at the same time, this system is flexible enough to be changed and upgraded.

One advantage of the custom monitoring system is that it can be relatively easily adapted for use with other lab equipment, not just 3D printers. This is useful in applications where equipment requires monitoring but not necessarily the physical presence of the operator.

There are several ways to upgrade my custom system. One useful addition would be to integrate an automatic warning system that would notify the user if something looks wrong in terms of a video feed or sensor data. This addition would eliminate the need for the user to even check the video stream, therefore making the system even more effortless to use. This can be done by sending a message to the user’s phone.

Another potentially useful addition would be making a custom server to stream the video feed. This would allow private and secure video transfer, which is useful when streaming sensitive information about the equipment or the experiment. However, setting this up is more complicated than using existing streaming platforms. There are other solutions for private streaming, but they are usually with monthly fees.

Adding the possibility to remotely control the printer can also be done. This would allow the user to stop the print remotely. It could theoretically be used to also change printer settings during the printing process. However, when the print has already started, changing the printer settings is not good practice and can potentially lead to print failure. The easiest way to add remote control is to connect the entire monitoring system via VNC to a remote computer.

SUMMARY

The aim of the thesis to categorize and review the information on monitoring the 3D printing process found in the literature was completed. Starting with the research into 3D printing technologies and common failures. This was followed by looking at the available monitoring systems and software.

The aim of the thesis to choose and prepare a low price and customizable monitoring/data collection system for the 3D printer was completed. First, the research for suitable ways of building a custom monitoring system was completed. Based on the gathered information a custom remote monitoring and data collection system was built.

The aim of the thesis to evaluate the functionality of the prepared system was completed. The system was tested to verify that it meets the requirements and works as expected. The custom system was also compared to commercially available solutions.

REFERENCES

1. The types of 3D printing technology: https://all3dp.com/1/types-of-3d-printers-3d-printing-technology/ (26.02.2021)
2. Material extrusion 3D printing: https://apiumtec.com/en/material-extrusion-fff-3d-printing (26.02.2021)
3. Pros and cons of a fused deposition modeling (FDM) 3D printer: https://tfmindustrial.com/prosof-fused-deposition-modeling-3d-printer/ (26.02.2021)
4. Liu Z.; Wang Y.; Wu B.; Cui C.; Guo Y.; Yan C.; A critical review of fused deposition modeling 3D printing technology in manufacturing polylactic acid parts, *The International Journal of Advanced Manufacturing Technology*, 2019, vol. 102.
5. Polylactic acid (PLA): The environment-friendly plastic: https://3dinsider.com/what-is-pla/ (26.02.2021)
6. VAT photopolymerisation: https://www.lboro.ac.uk/research/amrg/about/the7categoriesofadditivemanufacturing/vatphotopolymerisation/ (26.02.2021)
7. Troubleshooting guide to common 3D printing problems: https://all3dp.com/1/common-3d-printing-problems-troubleshooting-3d-printer-issues/ (26.02.2021)
8. Ten common 3D printing troubleshooting problems you may have: https://www.dobot.cc/resource/10-common-3d-printing-problems-troubleshooting.html (26.02.2021)
9. 3D printer not extruding? Here are some troubleshooting tips: https://3dinsider.com/3d-printer-not-extruding/ (26.02.2021)
10. 3D printer troubleshooting guide: https://www.matterhackers.com/articles/3d-printer-troubleshooting-guide (26.02.2021)
11. How to monitoring your 3D prints: https://3duniverse.org/2014/01/06/how-monitoring-your-3d-prints/ (28.02.2021)
12. Remote 3D printing - easy and reliable solution: https://www.flexihub.com/remote-3d-printing/ (28.02.2021)

13. Monitor your 3D printer with AstroPrint: https://solidutopia.com/monitor-your-3d-printer-with-astroprint/ (28.02.2021)
14. Repetier: https://www.repetier.com/ (28.02.2021)
15. The all-new MakerBot print: https://www.makerbot.com/3d-printers/apps/makerbot-print/ (28.02.2021)
16. OctoPrint: https://octoprint.org/ (28.02.2021)
17. Understanding the infrared temperature sensor: https://www.sensortips.com/temperature/infrared-temperature-sensor/ (28.02.2021)
18. Flexible quality assurance for 3D printer: https://www.eos.info/en/additive-manufacturing/software-3d-printing/monitoring-software (28.02.2021)
19. Creality Ender 3 3D printer: https://www.creality3dofficial.com/products/official-creality-ender-3-3d-printer (02.03.2021)
20. AstroPrint: https://www.astroprint.com/ (02.03.2021)
21. Software to control your 3D printer fleet: https://www.astroprint.com/3d-printer-farm-software (02.03.2021)
22. Educators' preferred 3D printing software: https://www.astroprint.com/3d-printer-school-university (02.03.2021)
23. AstroPrint plans and pricing: https://www.astroprint.com/plans-and-pricing (02.03.2021)
24. Raspberry Pi: https://www.raspberrypi.org/products/ (02.03.2021)
25. What is OctoPrint, and how is it used with 3D printers?: https://3dprintingmentor.com/what-is-octoprint-and-how-is-it-used-with-3d-printers/ (02.03.2021)
26. Download & setup OctoPrint: https://octoprint.org/download/ (02.03.2021)
27. OBS Studio: https://obsproject.com/ (06.03.2021)
28. Top 5 best OBS alternatives to record computer screen: https://www.vidmore.com/record-game/obs-alternative/ (06.03.2021)
29. How to record your computer screen for free with OBS: https://slurptech.com/record-your-screen-obs/ (06.03.2021)
30. Lightstream Studio: https://www.golightstream.com/ (06.03.2021)

31. Twitch review: https://www.pcmag.com/reviews/twitch (06.03.2021)
32. Twitch: https://www.twitch.tv/ (06.03.2021)
33. Twitch studio: https://www.twitch.tv/broadcast/studio
34. Ten best twitch alternatives for online streaming: https://www.winosbite.com/twitch-alternatives/#YouTube_Live (06.03.2021)
35. How to easily go live on YouTube from a computer: https://www.socialmediaexaminer.com/how-to-easily-go-live-on-youtube-from-computer/ (06.03.2021)
36. USB Temperature & humidity sensor with data-logging software: https://www.omega.com/en-us/temperature-measurement/temperature-and-humidity-data-loggers/p/RH-USB-Series (11.03.2021)
37. Infrared temperature sensor with USB interface: https://www.omega.com/en-us/temperature-measurement/noncontact-temperature-measurement/fixed-infrared-temperature-sensors/p/IR-USB-Series (11.03.2021)
38. Arduino: https://www.arduino.cc/ (14.03.2021)
39. List of Arduino sensors and modules: https://www.thegeekpub.com/wiki/list-of-arduino-sensors-and-modules/ (14.03.2021)
40. MLX90614 infrared sensor: https://www.aliexpress.com/item/32678232952.html?spm=a2g0s.9042311.0.0.13834c4d8Ih9RW (14.03.2021)
41. Arduino Nano: https://store.arduino.cc/arduino-nano (14.03.2021)
42. Infrared thermometer with arduino and MLX90614 temperature sensor: https://www.electronics-lab.com/project/infrared-thermometer-arduino-mlx90614-temperature-sensor/ (20.03.2021)
43. Delli U.; Chang S. Automated Process Monitoring in 3D Printing Using Supervised Machine Learning, *ScienceDirect: Procedia Manufacturing*, 2018, vol. 26.
44. Baumann F.; Roller D. Vision based error detection for 3D printing processes, *MATEC Web of Conferences*, 2016, vol. 59.
45. Petsiuk L.A; Pearce M.J. Open source computer vision-based layer-wise 3D printing analysis. *ScienceDirect: Additive Manufacturing*, 2020, vol. 36.

46. HYREL 3D System 30M: http://www.hyrel3d.com/portfolio/system-30m/ (01.05.2021)

Appendix

I. SOP for the monitoring system

For streaming cameras, sensor and Teamviewer (takes 5-10min):

1. Turn on both tablet and desktop PC

2. Log in to the desktop user: “****” password: “****”

3. Open Teamviewer on both tablet and desktop, connect devices so that you can control the tablet from the desktop.

4. Mount the thermal sensor to the printer, open Arduino IDE, connect the sensor, run “temp_sensor.ino” so you can get the sensor feed into the serial monitor (if Arduino gives an error then check if Arduino nano and old bootloader are selected in the settings)

5. Mount cameras outside the printer (choose a suitable setup according to preference)

6. Open OBS studio on desktop and set up cameras, serial monitor, and Teamviewer feed, start streaming into twitch (correct stream key should be already entered, if not then do step 7. and get the stream key from Twitch), press start stream

7. Open twitch.com, log into user: “****” password: “****” or use other preferred users (different stream key is required in OBS)

8. Open Twitch on different device and search for user: “****”, check if the stream is working properly

For streaming-only camera feed (takes <5 min):

1. Turn on desktop PC

2. Log in to the desktop user: “****” password: “****”

3. Mount cameras outside the printer (choose a suitable setup according to preference)

4. Open OBS studio on desktop and set up cameras’ feed, start streaming into twitch (correct stream key should be already entered, if not then do step 7. and get the stream key from Twitch), press start stream

5. Open twitch.com, log into user: “****” password: “****” or use other preferred users (different stream key is required in OBS)

6. Open Twitch on different device and search for user: “****”, check if the stream is working properly

II. Arduino code for IR temperature sensor

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
void setup() {
  Serial.begin(9600);
  Serial.println("temp_sensor");
  mlx.begin();
}
void loop() {
  Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());
  Serial.print("*C\tObject = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C");
  Serial.println();
  delay(500);
}
```

NON-EXCLUSIVE LICENCE TO REPRODUCE THESIS AND MAKE THESIS PUBLIC

I, Richard Erelt

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Remote monitoring and data collection for 3D printing

supervised by Laura Viidik

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Richard Erelt

20/05/2021