

# Augmenting Physical Prototype Activities in Early-Stage Product Development

Sampsa M. I. Kohtala<sup>1</sup>, Jørgen A. B. Erichsen<sup>1</sup>, Heikki Sjöman<sup>1</sup>, Martin Steinert<sup>1</sup>

<sup>1</sup> Norwegian University of Science and Technology (NTNU),  
Department of Mechanical and Industrial Engineering (MTP)

*smkohtal@stud.ntnu.no*  
*jorgen.erichsen@ntnu.no*  
*heikki.sjoman@ntnu.no*  
*martin.steinert@ntnu.no*

## Abstract

Prototyping in the early-stage of product development is widely used for exploring solutions and generating knowledge. Prototypes in the form of tangible artefacts have many advantages, including expressing and transferring tacit knowledge, creating proof of concepts, learning by doing and testing ideas. In the era of digitalization, this paper attempts to discover opportunities in engineering design research by transforming physical prototypes into digital 3D models, and methods for converting hand drawn sketches to physical parts, in addition to capturing microcontroller output.

With the increasing development and robustness of computer vision and photogrammetry algorithms over the past few years, simple methods for generating digital models of real objects have surfaced. By providing pictures of a prototype, these algorithms can generate a digital 3D representation, including color and texture.

A system for capturing information and knowledge from early-stage product development has been developed, and consists of a digital repository for collecting, storing and sharing data from design output (prototypes), and a physical instrument for capturing the input data. The physical instrument consists of several cameras used for taking pictures of prototypes, and a turntable to capture many angles for further processing to generate a 3D model. In addition, also a tool for producing laser cut pieces from sketches and a microcontroller logger is developed. It is aimed at advancing the discovery and understanding of causalities in the early stage of PD. As a positive side effect of enabling better research, the system can benefit its users (designers) by providing a basis for documentation and feedback.

Through practical experimentation and testing, we aim to discover the potential of methods such as photogrammetry in aiding practitioners reflect through design output (i.e. prototypes) in the early-stages of product development, as well as discussing the limitations of capturing design output from product development projects. Various ways of representing the prototype repository will be discussed, where making the prototypes accessible through virtual reality is

one possible concept that is discussed. The digital repository currently consists of data from various projects, including mechanical engineering student projects, a start-up developing a coreless ring motor, as well as projects from a multi-national product manufacturing company located in Norway.

***Keywords: Prototypes, Capturing Prototypes, 3D models, Protobooth, Product Development, 3D scanning***

## **1 Introduction**

Prototyping in the early-stage of product development is widely used for exploring solutions and generating knowledge. Prototypes in the form of tangible artefacts have many advantages, including expressing and transferring tacit knowledge, creating proof of concepts, learning by doing and testing ideas (Erichsen, Pedersen, Steinert, & Welo, 2016). In the era of digitalization, this paper attempts to discover opportunities in engineering design research by documenting performance of microcontroller-based prototypes, providing an effortless way to laser cut hand drawn sketches and transforming physical prototypes into digital 3D models.

With the increasing development and robustness of computer vision and photogrammetry algorithms over the past few years, simple methods for generating digital models of real objects have surfaced. By providing pictures of a prototype, these algorithms can generate a digital 3D representation, including color and texture.

One of the main problems with researching tools and methods in early-stage product development research is access to representative cases and companies that want to disclose their methods and findings from such projects. That is why the authors developed a tool named Protobooth that helps designers to capture and remember their prototypes better than before it was possible while allowing researchers to tap into the companies' and individuals' documenting process.

### **1.1 Research background**

In the pre-requirement stages, or the fuzzy front end (Herstatt & Verworn, 2004) of engineering design, designers are often working in large solution spaces along with ambiguous information and uncertainty. The greatest potential for discovering and testing new innovative solutions lie in these early stages, as they will greatly affect the cost and quality of the later converging development activities, such as optimization and manufacturing.

One of the key elements in early-stage PD is the generation and utilization of knowledge (Nonaka & Takeuchi, 1995; Ringen & Welo, 2015; Sutcliffe & Sawyer, 2013). Learning mechanisms in PD can be categorized into three loops (Leifer & Steinert, 2011). Learning loop one is based on explicit knowledge and aims to retain knowledge from the development projects. Learning loop two involves the informal space between PD team members and their coach. Tacit knowledge (Polanyi, 2009), learning loop three, is the skill and learnings of the individuals (i.e. designers). We argue that interacting with prototypes, in the form of tangible artifacts, is one of the most valuable dimensions of tacit knowledge as a means of knowledge acquisition (Nonaka, Toyama, & Konno, 2000). Prototypes, both external and internal reflective prototypes (Erichsen et al., 2016), can be used as learning tools by conceptualizing and sharing ideas, often with the intention of testing functionalities or suggesting the appearance of a product concept (Lim, Stolterman, & Tenenberg, 2008). With the benefits of interacting

with prototypes (and prototyping), more is often better, thus requiring a certain amount of time to be created. A solution for reducing both time and cost is the use of low fidelity prototypes (Bryan-Kinns & Hamilton, 2002).

Thus, our aim in this paper is to augment physical prototyping activities by simplifying and speeding up the process, ultimately allowing the generation of more design iterations to emerge in the early pre-requirement stages of product design. If we succeed, both experienced and inexperienced product developers can benefit from our system, to ultimately make better products faster.

## **1.2 The Proto booth Project**

In order to address the problem of getting a consistent method for documenting early-stage PD projects, a system for capturing project output from early-stage PD projects has been developed by the authors. This system is comprised of three main parts; a physical sensor platform for capturing data from project output (i.e. prototypes), a repository for storing aforementioned captured data and a user interface for interacting with the repository.

The system, which originates from the efforts described by Sjöman, Erichsen, Welo, and Steinert (2017), uses a multitude of sensors, including 7 cameras, load cells and RFID readers to capture data on prototypes (as well as sketches) from PD projects. Although rough details were presented (Sjöman et al., 2017), the system has seen some major upgrades in fidelity, performance and scale. Consequently, usage of the system has increased, and there are now around 50 users (of which about 20 are active each week), and the repository has roughly 400 prototype scans to date (early March 2018). One of the core principles of this system is that adding data to the repository should be effortless, thus increasing the chance of users wanting to utilize the system and leading to the capture of more project data.

The main objective of this research is to enable research of early-stage product development through capturing design output from projects (as detailed by Sjöman et al. (2017)). A goal of performing this research is to be able to feed this knowledge back into the PD process, in order to make better products. Consequently, this paper aims to present some key concepts that have been explored in a research setting with mechanical engineering graduate students, in order to aid designers in early-stage PD projects.

## **1.3 Aiding Designers in Early-Stage PD Projects**

Currently, the system can be categorized as a tool that users (designers) can use for documenting project progress. However, in this paper, we aim to highlight how we can experiment with adding more incentives and features to the system, thus making the system more useful to designers doing early-stage product development.

As the authors' research laboratory closely relates (both in proximity and activity) to the product realization lab at the mechanical engineering department at our university, the authors experience that users aiming to realize products (and prototypes) often have a certain set of tools that they are familiarized with and frequently use. The same users have a broader spectrum of tools and equipment that could be used, yet they tend to stick with methods and machinery that are familiar. A real example of such under-utilization is users 3d-printing small boxes for various micro-controllers (Arduino Uno, Raspberry Pi, etc.), and using such simple geometry (e.g. rectangles) that a laser cutter would produce more durable and more accurate models in less time than a FDM 3D printer. Therefore, we believe that lowering the threshold of learning

new equipment and machinery for realizing new products and prototypes will also make the lab users better equipped to solving their various tasks.

Based on both observation and experience from PD activities and some insight into start-up activity, we also believe that helping developers realize their ideas by producing prototypes will ultimately result in better solutions (and in less time). By contributing to more design alternatives, through helping design iterations emerge simpler and quicker, a single individual or a team of developers can gain more knowledge with less time. If this is the result of the methods presented in this paper, we have successfully achieved our goal of augmenting physical prototype activities in early-stage product development.

## **2 Prototype Realization Scenarios**

In our prototyping laboratory, most activities consist of building prototypes, in either individual or team-based projects. Projects usually revolve around high novelty product challenges in the fuzzy front end (pre-requirement stages) of product development. Attempting to solve these challenges are done through iterations (Steinert & Leifer, 2012) where designing, building and testing is done in several probes along the project timeline. The core concept of this model is to promote iterative learning cycles (Eris & Leifer, 2003) driven by rapid conceptual and tangible prototyping.

In this chapter, we present some common usage scenarios, and some challenges that we have identified in these scenarios. The aim of doing this is to highlight various prototype realization challenges, and later address how they could be solved.

### **2.1 Scenario 1: Designing and producing physical parts**

A common tool for creating physical prototypes is the laser cutter, mainly used for cutting materials such as Medium Density Fiberboard (MDF) and acrylic glass. It is popular due to its speed, accuracy and reliability, but requires training in order to be utilized properly and efficiently. Experienced users will use software, often Computer Aided Design (CAD) - software, to create a 2D sketch, export it as a vector file, save it to a removable drive, transfer it to the computer connected with the laser cutter, modify the cutting sequence and material properties in the laser cutting software, and finally press print to start cutting. Even experienced developers can spend a lot of time on making a laser cuttable file with the desired dimensions and geometry. Having a computer and measuring tools near the prototype is commonly observed in the laboratory during the building and testing of prototypes. Conceptual properties can be lost in the process of making prototypes for the laser cutter, as the physical connection between the idea and its practical use can be lost in all the software. Not until having produced the parts is it possible to perceive its real properties and applicability. This can result in designs that are not properly scaled, and not based on initial concepts of size and fit, thus promoting rework or results that are 'just good enough' and stays as is.

Many new and inexperienced users are more hesitant in using the laser cutter, mostly due to the uncertainty with learning new machines. Beginners have been observed to instead use more traditional tools, such as a saw, to form materials when creating prototypes. The threshold for becoming familiar with new software can be high, especially in early stages of development combined with a lack of experience. While learning and generating new knowledge is important in new product development (NPD), it can be time-consuming and best spent on the challenge rather than learning to operate new tools and machines. Consequently, building might take up

more time and focus compared to testing and developing new ideas. The time from concept to physical prototype should therefore be reduced if possible.

## **2.2 Scenario 2: Documenting performance of microcontroller-based prototypes**

It has become more common to use microcontrollers in product development and mechanical engineering courses (Slåttsveen, Steinert, & Aasland, 2016). Microcontrollers, such as the Arduino, provides a simple and fast method for interacting with the physical environment by controlling actuators and using sensors. Mechanical engineering students are encouraged to learn how to utilize them to build functional prototypes.

It is a less frequent practice to document such prototypes in an effective and effortless way. Usually they are documented by providing the script filled with comments explaining the code. When the actual prototype is used or demonstrated, it is difficult to show how it works programmatically. Printing serial data is used for debugging and understanding how the prototype works, but this content is often only used by the programmer who is writing and testing the code or prototype.

## **2.3 Scenario 3: Utilization of 3D technology**

CAD is a tool mostly used later in the development process, when requirements and specifications are made (and set). In the early stages of development, it is in many scenarios considered to be less effective, especially when developing novel products with a large solution space. In this case, it is often more rewarding to build prototypes that can be tested in the real world, to get a better sense of the concept(s) being tested and how they perform and to discover unknown unknowns (Gerstenberg, Sjöman, Reime, Abrahamsson, & Steinert, 2015). However, when prototypes need higher resolution it is common to utilize rapid prototyping tools such as 3D printing. The normal approach is then to create a 3D model with CAD. Designing and drawing 3D models can be a slow process, especially if the shapes are complex and/or based on an abstract concept. As in the case for making 2D parts for laser cutting, the perception can be skewed in the process, as the CAD can perfectly simulate an ideal part but differ in the real world. The scale of a model is an example that is often underestimated, resulting in small, unstable and weak parts that are difficult to produce. It is also common for newer engineering students to find CAD difficult and cumbersome.

With powerful computers, it is possible with 3D scanning to generate accurate 3D models of real objects quickly. It could potentially be feasible to, within a few seconds, have an augmented reality representation of the real model at a remote location, or 3D-print it to get a physical copy. It would also be possible to combine the real-world prototypes with virtual reality prototypes to make assemblies at several different locations at the same time, thus keeping production separate at long distances. The authors imagine having sessions with teams where people work at different locations around the globe but are able to virtually assemble each part together to simulate and test the overall functionality. Another possibility could be that customers could test the physical prototypes or user experience prototypes before costly production takes place.

## **3 Technology solutions**

In this paper we highlight experiments with new features for aiding designers in early-stage PD activities. Additionally, we will present the technologies behind these experiments, including

the hardware used and the software implementations. Furthermore, we will provide possible benefits that this system can bring to early-stage PD in general and present an in-depth look at user activity in our research laboratory with examples of how the system can help these users with realizing prototypes and documenting development. First, we present existing solutions and argue why our system is solving the challenges in a more efficient way.

### **3.1 Technical solutions**

#### *3.1.1 Existing solutions*

3D scanning has become a more common tool in the industry, used for reverse engineering and quality control. In PD, it is often used to accurately model complex geometries, such as faces, hands, bones or limbs (for prosthetics), to make custom products. However, it is generally not associated with the pre-requirement stages of PD, therefore often avoided or not considered until later development stages. Some of the main drawbacks with most commercial 3D scanning equipment is cost, scanning time and complexity. HP 3D Structured Light Scanner is an example, which is a professional-level instrument using a camera and a projector. The instrument is highly accurate (up to 0.05 mm) and fast, but is an advanced system consisting of many elements that needs to be manually adjusted to work properly, in addition high cost. Matter and Form 3D Scanner is a user-friendly system and a low-cost alternative. It is however slow and more limited in the object size it can 3D-scan. There are also many handheld scanners that provide more flexibility during the scanning process, and mobile apps that utilize photogrammetry to reconstruct 3D models from pictures through cloud processing. Depth-sensor based systems does not have very high resolution or are very expensive and advanced. The method of Photogrammetry is often more suitable due to availability of good cameras with high resolution. Generally, commercial Photogrammetry programs are interactive and fixed in its capabilities. Photogrammetry can be difficult, but is very flexible, accessible and low cost, thus having the potential to be tailored for specific usage and automation.

Using hand drawn sketches to directly produce physical parts is not common practice in PD activities to the knowledge of the authors. Converting hand drawings to digital copies or g-code for engraving is used mostly in the visual arts. A drawing is then typically scanned with a normal paper scanner and manually processed with software such as Inkscape and Photoshop.

Ready-made microcontrollers (e.g. Arduino, etc.) are widely used in early-stages of PD to quickly make functional prototypes. It is a great way to facilitate ideas and test them in the real world and show how the product works. It is however less common to intuitively demonstrate and document how the physical prototype works while simultaneously demonstrating how it works internally (programmatically). Usually, data from the microcontroller is displayed through software, such as the Arduino integrated development environment or Processing.

#### *3.1.2 What we offer through Protobooth*

Our system provides an all-in-one solution to different challenges in product design and development. The different technologies discussed in the previous chapter are integrated into one instrument, in such a way that it is as simple to use as possible. Furthermore, we use open source software developed by the community, which enables more control and automation possibilities while being free of charge. In Protobooth, we control the programs and the physical environment (lighting and background color), which enables us to tailor its functions to the users' needs. In this way we can promote new innovative approaches for producing prototypes and help developers realize ideas.

Photogrammetry is usually done by experts that are familiar with its limitations. In the controlled environment of Protoboost, we can experiment with different materials and textures, and use this information to automatically adjust parameters based on what is put inside. Users can then utilize this technology without being experts in photogrammetry or having to learn new 3D scanning equipment. Photogrammetry is also a very fast way to capture the required spatial data compared to other sensor-based methods; however model reconstruction time can vary greatly based on processing power and data quality. Another benefit with photogrammetry is that the scanned model can vary in size, from small components to large structures. Another benefit is that the reconstructed model will always include detailed texture, providing accurate, yet simple, possibilities for visual communication. This is promising for virtual reality applications in the near future.

Making hand drawn sketches to directly produce physical parts in the laser cutter is seldom observed in PD activities, due to existing software being more directed towards other applications. We believe this approach can help designers make parts faster and with less effort in many scenarios. By implementing this function in Protoboost, users can make simple designs and quickly have results they can use.

Even though it is common to use the IDEs (and serial monitors) for displaying how microcontrollers work, it is often not convenient for documenting or sharing this knowledge. It can also be a steep learning curve to simply log sensor data from a microcontroller to a computer, to enable plotting and other forms of data processing. With Protoboost, these steps are accomplished by simple plug and play, linking the physical prototype to the actual (real-time) code output. The users can thus focus more on the actual prototyping while getting data out of their activities and prototypes for free.

### **3.2 Hardware**

The main elements of the physical Protoboost, some of which are depicted in Figure 1, consist of:

- Structure made of an aluminum frame, wooden walls and a table
- Several Logitech C930e and C920 web cameras
- A turntable
- LEDs
- Intel NUC with Debian 9
- RFID reader
- Arduino Mega
- Nikon D5300 camera
- Cross laser



**Figure 1.** The Protoboost used for testing new features. The camera used for detecting and capturing sketches is shown on the top right image.

The Arduino Mega handles the LEDs, cross laser and a stepper motor driving the turntable. The Arduino, RFID reader and cameras are connected to and controlled through NUC.

### 3.3 Software

Most of the software used is open source, except laser cutting software “Gravostyle”. Logitech web cameras are used with an open source computer vision library [OpenCV], while the Nikon camera is controlled through command line arguments provided by gPhoto2. Node-red, a Node.js framework with its browser-based flow editor, connects hardware and programs in an orderly manner, shown in Figure 2. The main programs are written in C++.

A program for controlling an Arduino mega through serial communication is used for simply setting turntable parameters, activation and changing color of LEDs.

The object classification program is used to capture an image of the current object placed in Protoboost and classifying it with a trained neural network, using YOLOv2 by Redmon and Farhadi (2016). It can detect if a sketch is present or not, as such determine the next program to run.

Video recording is accomplished using OpenCV. If an Arduino microcontroller is connected while recording, the serial data from the microcontroller is displayed on the video in real time. A separate text document with the data is also provided.



The sketch to laser cutter program captures an image of the sketch placed in Protobooth. It converts the image to a scaled PDF file and a binary image. The binary image is further processed into a vectorized DXF file using KVEC.

For 3D-scanning, a program was made, utilizing gPhoto2, to capture several images with Nikon D5300 while the object rotates on the turntable. 3D reconstruction from the images is accomplished through several steps, using COLMAP (Schönberger, Zheng, Frahm, & Pollefeys, 2016) and OpenMVS. Steps include feature detection, extraction and matching, generating a point cloud, densifying the point cloud, generating a mesh, refining the mesh and finally adding texture to the model.

Google drive is used for sending files between Protobooth and laser cutter PC. This procedure is used in the experimentation stage. In the future, all data can be sent to the users' repository, accessible through a web interface.

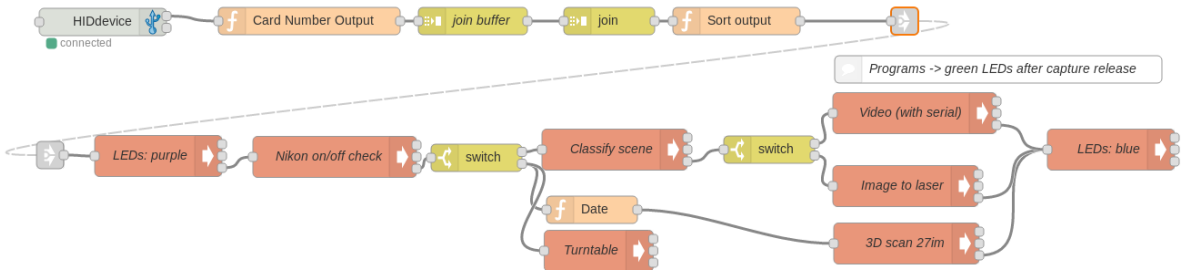


Figure 2. Node-red interface linking functions and executable programs.

One of the goals of Protobooth is to keep it as simple and nonintrusive as possible, lowering the threshold for using it while helping both experienced and inexperienced product developers realize and convey their ideas. The users can simply place their prototype in Protobooth and scan their personal ID access card on the RFID reader. This activates a series of programs, illustrated in Figure 2, based on what is put inside Protobooth. For 3D scanning it is necessary to activate and position the camera manually first. LEDs are used to indicate at which stage the program is currently at, where purple means the prototype is being captured, green means capturing is complete and the user can remove the prototype, and blue meaning Protobooth has processed every stage and is ready to be used again. The cross laser is used to mark the center and will automatically turn off while capturing prototypes.

## 4 Experimenting with aiding designers

### 4.1 Sketch to laser

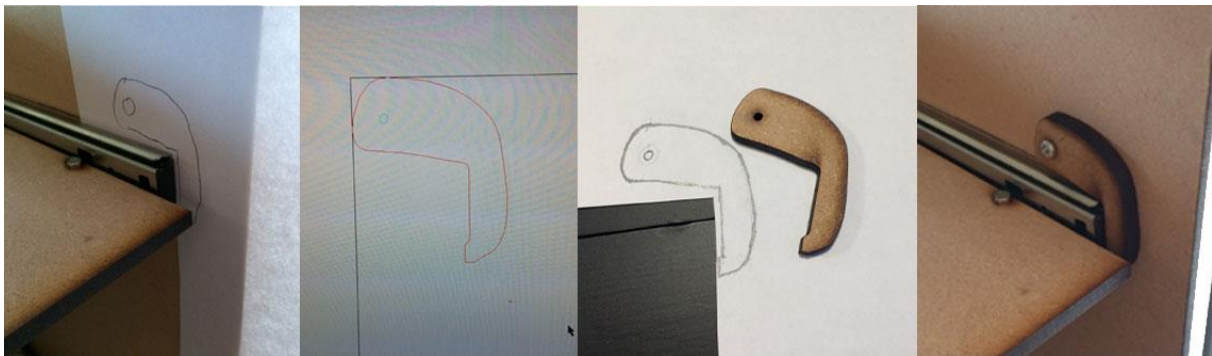
Protobooth is capable of capturing hand drawn sketches and convert them to a scaled PDF and vector (DXF) file. After detecting centerlines in the sketch and applying a simulated Bezier algorithm, both provided by KVEC, a DXF file is generated containing vectors made up of approximated polylines. The files are automatically sent to a shared folder on the computer that has the laser cutter software installed. After a user has captured the sketch with Protobooth, it takes approximately 10 seconds before the files are ready and received at the laser PC. By simply dragging and dropping these files into the laser cutting software, they can be modified and prepared for cutting.

When a vector is selected in the software every connected vector is highlighted, which is essential when applying the cutting sequence to different parts of the design and not having to select each vector individually. The PDF is a scaled version of the original image captured of the sketch. When this file is opened in the software it is possible to manually draw vectors over the image to make refined and customized cutting patterns of the original sketch.

#### 4.1.1 Practical example

An example case, where a locking mechanism was needed for a drawer, was used to test and demonstrate the system. Without knowing the dimensions or exactly how it should look, the normal approach would be to take measurements, make a few designs and build them for testing. Instead, a piece of paper was placed at the area of interest, and an outline was drawn directly where it should fit. Using a lead pencil made it simple to refine the sketch after the first rough version was drawn.

After scanning the sketch in Protobooth, the DXF file was dropped into the laser cutting software. Like in any case when using the software to cut materials, the only required change made to the vectors was changing color to specify the cutting sequence, which was done with a few mouse clicks. After setting laser the parameters and choosing the material, the part was cut smoothly.



**Figure 3.** From left: pencil drawing, vectors generated from the drawing (colors are selected for cutting sequence), comparing the result with the original sketch, and lastly showing its final application. Note that the leftmost sketch was slightly modified before scanning, as shown on the middle-right image.

#### 4.1.2 Limitations and possible solutions

A reoccurring problem with this approach has been a presence of many vectors not part of the sketch, in addition to not discovering thin or weak lines. The simple explanation is the quality of the picture taken with the Logitech web camera, or the lighting condition. Although decent for video recording, 2MP is low for taking pictures. In some cases, this might have caused noise in the binary image due to the difference in observed color between white paper and drawn lines being too small. Using a higher resolution camera with different lighting can resolve this issue.

## 4.2 Video recording with serial output

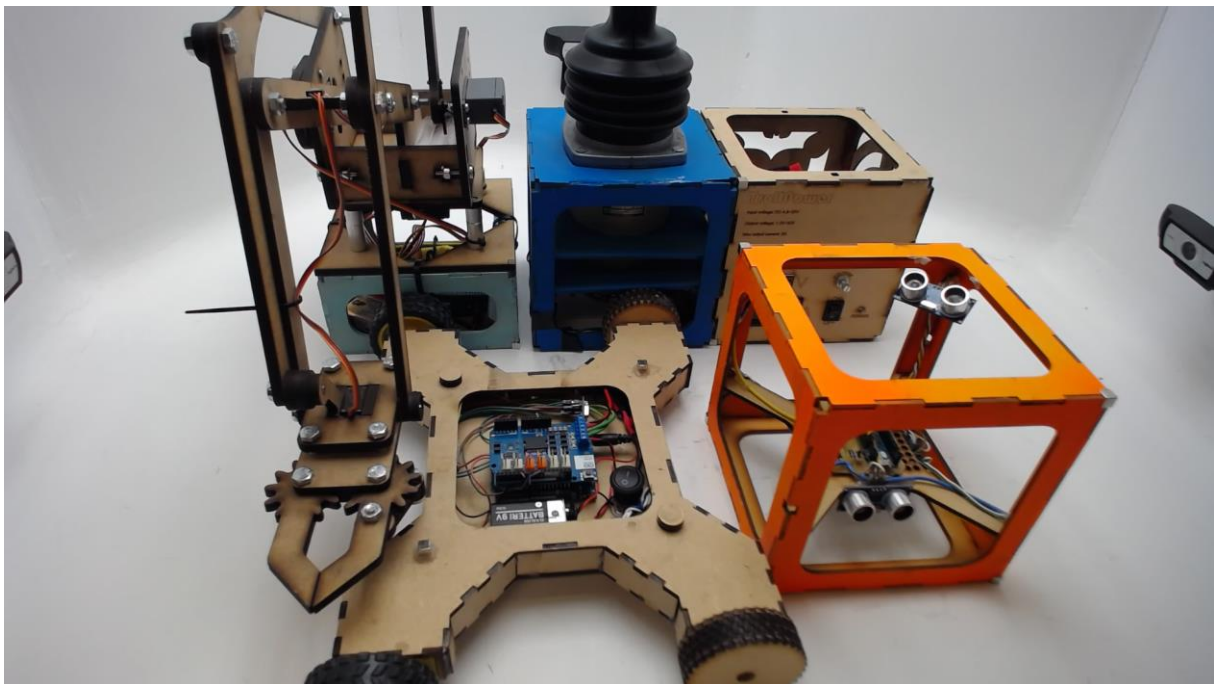
When connecting a microcontroller to the available USB port, Protobooth (NUC) will open the port and read the serial data that is received. Currently only Arduino based microcontrollers are tested and accepted by the program. When a user scans their RFID card a signal is sent to the Arduino which will reset it. Video recording is then started and synced with the serial output.

The output text is written to the video frames using OpenCV. With the current hardware setup, it is possible to record 1080p video up to 30fps.

#### 4.2.1 Practical example

A microcontroller-based system was developed at a course by a team of three students including one of the authors. It is a modular robot system consisting of different modules (boxes) that can communicate with each other through radio signals. Each box is equipped with an Arduino and a transceiver.

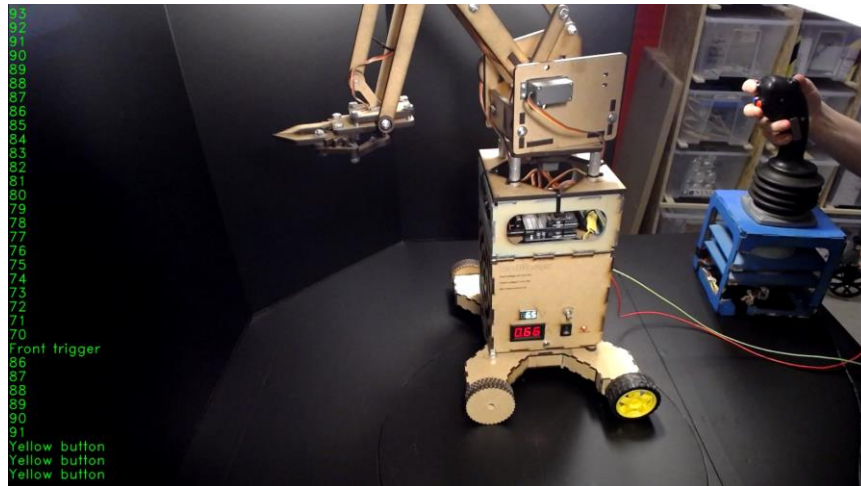
To showcase how the prototypes could be used, a handful of modules was created: a joystick, car, robot arm, power supply and sensor module (see Figure 4). Even though the team behind the project consisted of only three students, it was not entirely clear to everyone how these modules worked together, as they were built and programmed by different team members.



**Figure 4. Prototypes from the modular robot system.**

The system has since been documented using Protobooth. One of the videos made, shown in Figure 5, demonstrates how different modules react to signals received by the joystick module. By simply connecting the joystick module to Protobooth, it was easy to demonstrate what types of signals were sent to the other modules. In Figure 5 the numbers represent the angle of one of the servos on the arm, which are changing based on the joystick movement at different rates while the arm moves accordingly. It is shown on the bottom that the yellow button is pressed on the joystick, and moments later the car is controlled instead of the arm. In this way it is clearly demonstrated how the modules react and how they work together. A 36 seconds long video was needed to properly demonstrate the functionality of this setup.

Other short videos were also made (Kohtala, 2018), showing how to set up and initiate communication with the different modules. Not only can this improve communication within teams, but other users can play with these prototypes now that they can simply learn how to use them on their own.



**Figure 5. Snapshot from a video captured with Protoboosth showing how different modules work together. The joystick module on the right is connected to Protoboosth while recording, and its serial output is displayed in real time on the left side of the video frame.**

#### 4.2.2 Limitations

It requires good programming practices to maximize the potential of this method. Even if data is printed to the screen, it does not guarantee that the video with serial text makes sense to other viewers.

### 4.3 3D-scanning

Protoboosth has a semi-automated process for 3D-scanning a prototype. A DSLR camera must be aimed manually to capture the object, before an automated process rotates the object while capturing a total of 28 images. This process takes about 28 seconds, restricted by the capturing and downloading speed of the camera using gPhoto2.

Photogrammetry is used to reconstruct a 3D model from a set of images. COLMAP and OpenMVS is used to automatically generate a complete 3D model. Through several steps, the algorithms outputs point clouds and meshes with MVS and PLY file formats. The last stage generates a refined mesh with texture. Processing time can vary based on the complexity of the object and the image quality, in addition to processing power. MeshLab can be used to further modify the outputs or to simply export the model to STL format, which is supported by most CAD and 3D printing software.

#### 4.3.1 Practical example

An attachment for a Raspberry Pi to a Nikon camera was made with modelling clay, as an experiment for testing the system. Different colored clay was mixed to add texture and improve feature detection. The clay was quickly molded by pressing it against the camera and Raspberry, and then shaped by hand. A small piece of clay was also used to hold the model upright while scanning, to capture the largest and most critical surfaces in one scan.

Markers were used to further assist the algorithms to detect features and align photos. The markers and support were removed in MeshLab after the reconstruction was complete. A function in NX was used to automatically fill holes on the model, which came from removing the clay support in MeshLab, in addition to scaling the model after measuring two points on both the clay and digital model. A feature was also added with NX for the locking mechanism which was not present on the clay model, shown as the light brown part in Figure 6.

To save time another computer than the NUC was used. With an i7-7700HQ CPU running at around 3.5GHz with a 16GB memory capacity, reconstruction finished after 10 minutes. Different stages of the process are shown in Figure 6. The final 3D-printed prototype (see Figure 7) fits nicely to the camera and is easy to connect while remaining rigid in place.

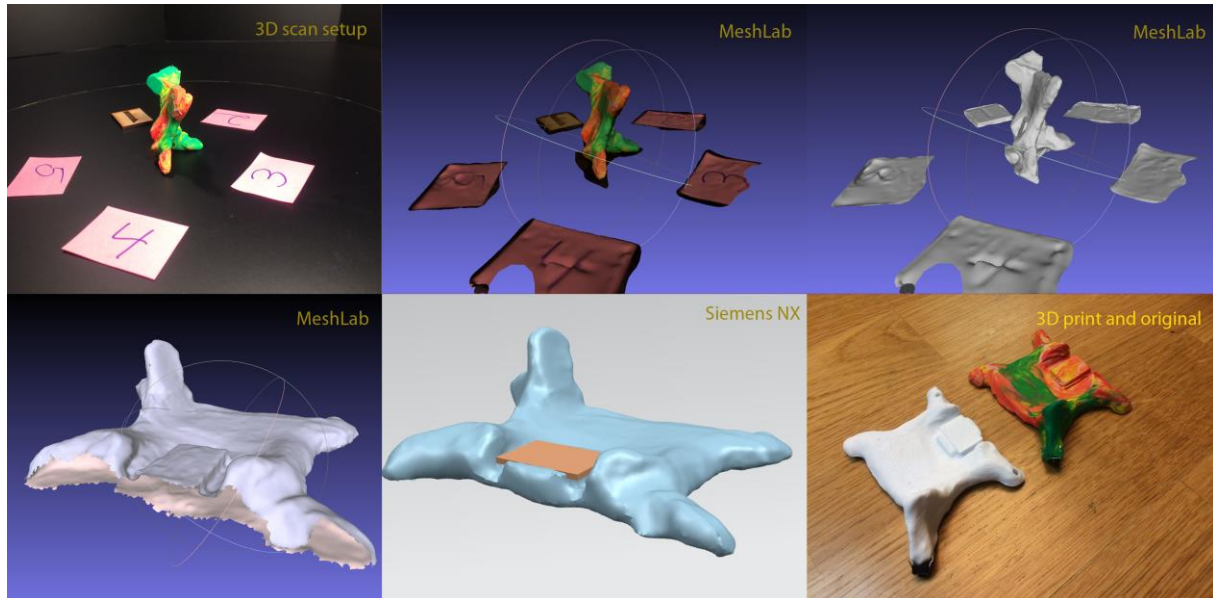


Figure 6. 3D-scan to 3D-print.

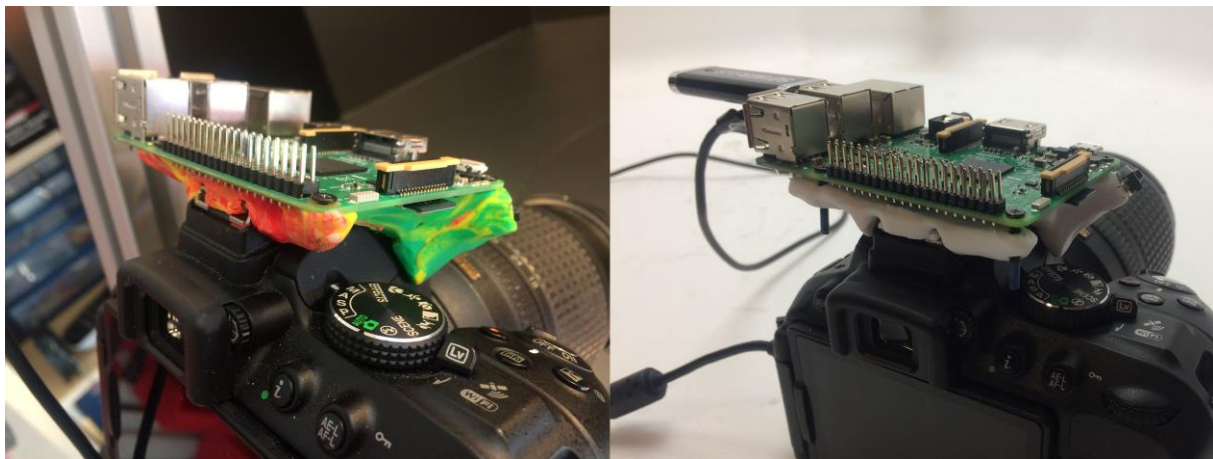


Figure 7. Modelling clay to 3D-print result.

#### 4.3.2 Limitations and possible solutions

Low memory can result in reconstruction failure, thus a decent computer is required to execute the programs properly and in less time. Results from scanning does currently requires some manual processing to remove unwanted features before printing the model. However, solutions exist in MeshLab where specific colors on the mesh can be selected and removed. It is then possible to use markers, and support if needed, with specific colors to be more easily removed from the final model. The reconstructed model is not scaled and must be manually calculated and set in software to get a correct digital representation. It might be possible to implement methods to apply and calculate scale factor based on camera parameters such as focal length and distance to object.

## 5 Discussion and conclusion

We have presented observations from early-stage PD activities and common prototype realization scenarios that can be challenging for some designers. To address these challenges, new features has been added to the prototype capturing system (Protobooth) developed by the authors, including 3D scanning, techniques for converting hand drawn sketches to laser cut parts and a way to document microcontroller-based prototypes.

Through practical examples, we have proposed how these methods can aid designers develop physical prototypes in early-stage PD and discussed their limitations and possible solutions. The main methods explored to augment physical prototype activities are simplifying the utilization of 3D scanning technology, a fast and simple way to make laser cut parts and a plug-and-play method for documenting microcontroller-based prototypes. Additionally, we have tested object recognition methods to make the user experience of Protobooth even simpler, by automatically detecting what is inserted before running the next appropriate step. Results from experimenting with these features has shown the potential to aid designers. However, it requires more testing and feedback from users to determine its true potential.

We aim to further improve the system based on user feedback, by keeping Protobooth as an evolutionary yet functional tool.

### Acknowledgement

This research is supported by the Research Council of Norway through its user-driven research (BIA) funding scheme, project number 236739/O30.

### References

- Bryan-Kinns, N., & Hamilton, F. (2002). *One for all and all for one?: case studies of using prototypes in commercial projects*. Paper presented at the Proceedings of the second Nordic conference on Human-computer interaction.
- Erichsen, J. A., Pedersen, A. L., Steinert, M., & Welø, T. (2016). *Using prototypes to leverage knowledge in product development: Examples from the automotive industry*. Paper presented at the Systems Conference (SysCon), 2016 Annual IEEE.
- Eris, O., & Leifer, L. (2003). Facilitating product development knowledge acquisition: interaction between the expert and the team. *International Journal of Engineering Education*, 19(1), 142-152.
- Gerstenberg, A., Sjöman, H., Reime, T., Abrahamsson, P., & Steinert, M. (2015). *A Simultaneous, Multidisciplinary Development and Design Journey—Reflections on Prototyping*. Paper presented at the International Conference on Entertainment Computing.
- Herstatt, C., & Verworn, B. (2004). The ‘fuzzy front end’ of innovation. In *Bringing technology and innovation into the boardroom* (pp. 347-372): Springer.
- Kohtala, S. (2018). TrollBot video serial playlist. Retrieved May 30, 2018, from [https://www.youtube.com/playlist?list=PLj9XjbRcnJ\\_0AMBmT0Idtu9-BngYM0qmp](https://www.youtube.com/playlist?list=PLj9XjbRcnJ_0AMBmT0Idtu9-BngYM0qmp)
- Leifer, L. J., & Steinert, M. (2011). Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning. *Information Knowledge Systems Management*, 10(1-4), 151-173.

- Lim, Y.-K., Stolterman, E., & Tenenberg, J. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 15(2), 7.
- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*: Oxford university press.
- Nonaka, I., Toyama, R., & Konno, N. (2000). SECI, Ba and leadership: a unified model of dynamic knowledge creation. *Long range planning*, 33(1), 5-34.
- Polanyi, M. (2009). *The tacit dimension*: University of Chicago press.
- Redmon, J., & Farhadi, A. (2016). YOLO9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*.
- Ringen, G., & Welo, T. (2015). *Knowledge based development practices in systems engineering companies: A comparative study*. Paper presented at the Systems Conference (SysCon), 2015 9th Annual IEEE International.
- Schönberger, J. L., Zheng, E., Frahm, J.-M., & Pollefeys, M. (2016). *Pixelwise view selection for unstructured multi-view stereo*. Paper presented at the European Conference on Computer Vision.
- Sjöman, H., Erichsen, J. A. B., Welo, T., & Steinert, M. (2017). Effortless Capture of Design Output. 7.
- Slåttsveen, K. B., Steinert, M., & Aasland, K. E. (2016). *Increasing student confidence and motivation in a project-based Machine Construction and Mechatronics course*.
- Steinert, M., & Leifer, L. J. (2012). 'Finding One's Way': Re-Discovering a Hunter-Gatherer Model based on Wayfaring. *International Journal of Engineering Education*, 28(2), 251.
- Sutcliffe, A., & Sawyer, P. (2013). *Requirements elicitation: Towards the unknown unknowns*. Paper presented at the Requirements Engineering Conference (RE), 2013 21st IEEE International.