



---

**Themenheft Nr. 56: Making & more: gemeinsam Lernen gestalten.**

Herausgegeben von Bernadette Spieler, Manuela Dahinden, Klaus Rummler und Tobias M. Schifferle

## **From Smartphone to Fabric: Mobile Embroidery Programming**

**Improving the maker workflow to reduce entry barriers in STEAM**

Sarina Gursch<sup>1</sup> , Stephan Robinig<sup>1</sup> , Patrick Ratschiller<sup>1</sup> , Stefan Kutschera<sup>1</sup> , and Wolfgang Slany<sup>1</sup> 

<sup>1</sup> Technische Universität Graz

### **Abstract**

*Equipping the next generation with skills in science, technology, engineering, art, and mathematics (STEAM), including computational thinking and graphic design, is crucial. Yet it can be difficult for young people to sustain interest in technology – particularly for girls due to persistent differences in their socialization. Project-based learning combining hands-on experience with fashion and design offers a promising means of sparking and sustaining interest in technology. Mobile embroidery programming is one such project, but there are still challenges to be overcome in transferring embroidery files to programmable embroidery machines as part of school courses or coding weeks. This report presents a device we developed that facilitates wireless file transfer. The embroidery file, encoded in a common format like DST, can be wirelessly transferred to the embroidery machine. We optimized the process of the embroidery workflow in a series of 14 workshops. Furthermore, we tested additional hardware for embroidery file transfer based on Raspberry Pi on a group of 18 participants. Our findings suggest that wireless file transfer may be useful with other maker devices such as 3D printers. The results will benefit schools, maker spaces, code clubs, and individual users seeking to engage in STEAM activities.*

**Vom Smartphone zum Stoff: Mobile Stickprogrammierung. Verbesserung des Arbeitsablaufs für Maker Education zur Reduzierung von Einstiegshürden in MINKT**

### **Zusammenfassung**

*Die Vorbereitung der nächsten Generation mit Fähigkeiten in Mathematik, Informatik, Naturwissenschaft, Kunst und Technik (MINKT), einschliesslich des Denkens in Bezug auf Computer und der Gestaltung von Grafiken, ist von entscheidender Bedeutung. Die Aufrechterhaltung des Interesses an der Technik, insbesondere bei Mädchen, kann jedoch*



*eine Herausforderung darstellen. Projektbasiertes Lernen, das praktische Erfahrungen mit Mode und Design kombiniert, bieten einen vielversprechenden Ansatz für Mädchen. Eine Herausforderung ist die Übertragung von Stickdateien auf programmierbare Stickmaschinen während Schulkursen oder Coding Weeks. In diesem Bericht wird ein von uns modifiziertes Gerät vorgestellt, das die drahtlose Übertragung von Dateien ermöglicht. Die Stickdatei, die in einem gängigen Format wie DST kodiert ist, kann drahtlos zur Stickmaschine übertragen werden. Anhand von 14 Workshops haben wir den Ablauf des Stickworkflows optimiert. Darüber hinaus haben wir die Hardware für die Übertragung von Stickdateien auf der Basis des Raspberry Pi an einer Gruppe von 18 Teilnehmern getestet. Unsere Ergebnisse deuten darauf hin, dass die drahtlose Dateiübertragung auch mit anderen Maker-Geräten, wie z. B. 3D-Druckern, nützlich sein kann. Die Ergebnisse kommen Schulen, Maker Spaces, Code Clubs und Einzelpersonen zugute, die sich an MINKT-Aktivitäten beteiligen möchten.*

## 1. Introduction

There is a growing gender gap in science, technology, and innovation in developed countries (Gursch et al. 2022). At the same time, these fields are evolving more rapidly than ever before. To ensure that men and women have equal opportunities to shape innovation, young people must be equipped with key skills in science, technology, engineering, and mathematics (STEM) for their future jobs and ventures. STEAM extends the STEM concept by adding the arts – for instance, graphic design skills (The Conversation 2018). STEAM education encourages students to transfer their knowledge across disciplines (Bertrand and Namukasa 2020). Sustaining young people's interest in technology remains challenging, however – especially due to persistent differences in socialization between boys and girls. Initiatives such as coding clubs and special programs have been developed to counteract this problem (Buhnova and Prikrylova 2019; Kelleher and Pausch 2007). Project-oriented access to technology through hands-on experience, especially in combination with art, offers a promising approach to STEAM education for girls. It provides the opportunity to see how different fields are connected through interdisciplinary and even transdisciplinary learning (Liao 2016). One approach, the use of programmable embroidery machines, has proven effective in encouraging girls to program (Gursch et al. 2021). Children and teenagers can create their own designs, program them, and have them embroidered onto fabric with an embroidery machine. Barriers to this creative process still exist, however: the availability of the necessary machines, the skills to operate them, and the actual file transfer to the embroidery machine in the context of school courses or coding weeks. Transferring data from one or even multiple student devices to an embroidery machine involves various intermediate steps, complicating course implementation.

This report describes a process to facilitate file transfer. We show how to transfer a design onto textiles with the help of an embroidery machine. All that is needed is a smartphone with an app such as Embroidery Designer for Android and iOS to implement the user's design. The next step is to transfer the created embroidery file, encoded in a common embroidery file format such as DST, to the embroidery machine. Next, the embroidery machine is used in interaction with an operator, often the users themselves, to stitch the design onto a textile. The optimization of the process builds on studies carried out in the context of the Code'n'Stitch project. In this project, fourteen workshops were conducted that served in evaluating the approach. Furthermore, additional hardware based on Raspberry Pi for transferring the embroidery files was tested on a group of 18 participants. Using devices like Raspberry Pi Zero W allowed us to create a small device that interfaces the programmable embroidery machines via a USB-OTG (on-the-go) connection. Users can send their creations via Bluetooth or a network connection to the interfacing Raspberry Pi. A Samba network share is used for network connectivity, while a modified version of a BlueZ agent manages the Bluetooth interface. A file system watchdog is used to move new files to a folder that serves the USB-OTG connection. We show that this embroidery and programming approach is suitable for young people and present its use in the school or maker education context. Through these findings, anyone with a smartphone and access to an embroidery machine can create their own design on a textile. Although these maker platforms enhance the user experience, they come with a slightly higher entrance barrier for educators due to the necessary steps for setting up the hardware.

Here we show that the increasing availability of maker platforms can be used to improve the workflow and scalability of programmable embroidery machines. This approach allows us to provide users with a seamless experience for learning new skills. Furthermore, we present the possibilities of the required tools: a smartphone with a corresponding app, a Raspberry Pi, and an embroidery machine. Through the additional use of the Raspberry Pi, the DST files no longer have to be transferred to a USB stick but can be sent directly by Bluetooth. This approach simplifies course implementation and will benefit schools, maker spaces, code clubs, individual users, and businesses providing embroidery services to customers. Our results are useful for users including educators, artists, fashion designers, and young people. Moreover, using a Raspberry Pi to transfer files is not limited to the embroidery machine use case. We found that the Raspberry Pi can be used with many maker devices such as 3D printers. It is expected that our Raspberry Pi-based solution will support a number of maker scenarios in which data have to be transferred from mobile devices to maker machinery.

CatroPi is specifically tailored to transferring files quickly via a wireless or Bluetooth connection to the maker device without any need for other tools or devices. Compared to other Raspberry Pi-based solutions (such as Octoprint<sup>1</sup> or Klipper<sup>2</sup>), CatroPi is a lightweight, general-purpose device and not tailored to any specific maker device, which makes it usable for any device that accepts file transfer through USB. Our solution is beneficial for a broad spectrum of applications and will help empower young users in their learning processes, enabling them to more easily create embroidery works from their own designs. This promises to spark enthusiasm and foster computational thinking and STEAM skills. The report answers the following questions:

- What advantages can the CatroPi prototype offer in maker education?
- Which steps are needed for creating an embroidery work using a smartphone?

## **2. Related Work**

In recent years, many initiatives and research projects have been conducted in the context of maker spaces for different age groups. Maker spaces offer the possibility to provide learners with hands-on experience. STEM and specifically STEAM education significantly benefits students because it allows them to apply theoretical concepts in practice (Connor, Ferri, and Meehan 2013). Maker spaces aim to bring STEAM subjects closer to students and foster their creativity and innovation (Galaleldin et al. 2016). This report focuses on the aspects of file transfer in maker spaces and the creation of an embroidery design, which are briefly outlined below.

### **2.1 File Transfer in Maker Spaces**

Maker spaces often provide devices from 3D printers and laser cutters to embroidery machines that have WiFi connectivity and therefore offer a simplified file transfer mechanism. Some 3D printers can be connected to a network, making it possible to send files directly (Figuera, Dong, and El Saddik 2013). However, embroidery machines with WiFi connectivity are expensive, and while those without such hardware may offer data transfer through USB sticks or SD Cards, there is often little documentation available on this feature. Some publications have demonstrated how embroidery can be done using PC software, and mention that a USB connection, flash card, or USB stick is needed to transfer the design to the machine, or explain how the data can be streamed to a micro controller (Devi et al. 2019). In the context of file transfer for maker spaces, the issue of file transfer has been discussed, especially for 3D printers. Marques, Williams and Zhou (2017) use an Arduino Yun on top of a

---

1 <https://octoprint.org/>.

2 <https://www.klipper3d.org/>.

small mobile 3D printing robot. The machine code instructions, which are similarly used on computer numerical control (CNC) machines, known as G-code, also must be transferred to the robot. Successful transfer requires a wireless connection to a PC that streams G-code data containing the printing information to an Arduino Mega that operates as the main circuit for the printing robot. A WiFi-enabled USB memory stick was sold in 2016 by the company SanDisk (SanDisk Connect Wireless Stick), although the device seems to have been discontinued. However, while this device made it possible to send files to the USB stick using WiFi, this was only possible when the device was not plugged into a USB port. Thus, the use case would still demand unplugging the stick from the device in the maker space (embroidery machine, 3D printer, etc.) each time a file is sent to it, which makes the process more complicated than with the CatroPi we present. There are a few microcontrollers that can be used for 3D printing (Evans 2012). An interesting aspect is that Printrboard uses a USB interface connected to the host PC or alternatively an SD card reader to supply the files. The Sanguinololu board uses USB only. Various free software programs can control 3D printers. For this purpose, a Raspberry Pi is connected to the 3D printer and controlled using the OctoPrint software. OctoPrint also makes it possible to control other maker devices with additional plugins. Like Klipper, OctoPrint has the main function of communicating with the 3D printer via G-code. It also requires that software be installed on the end device. With OctoPrint, in contrast to CatroPi, no further software is needed for the smartphone (or laptop). There is, however, a smartphone app available where the smartphone itself can be connected to the 3D printer instead of a Raspberry Pi board. This shows the relevance and significance of CatroPi, since it improves the workflow when transferring files to a maker space device with more users simultaneously. Since file transfer with the USB stick takes time, with multiple steps such as reconnecting to the embroidery machine and having an overview of the right USB stick with the suitable file in groups, the connection with CatroPi can be simplified by sending a file during the embroidery process. Connecting CatroPi with the local network provides file access through the network share, simplifying the maker process logistically. Additionally, the hardware cost is lower than a maker device with comparable connectivity options.

## **2.2 *Creating an Embroidery Design***

The development of physical computing enriches maker spaces to benefit learning in STEAM fields. Physical computing through robotics and wearables encourages students to acquire STEAM skills (Merkouris, Chorianopoulos, and Kameas 2017). In particular, e-textiles and wearable computing offer promising introductions to computer science and foster learning in STEAM fields (Buechley, Eisenberg, and Elumeze 2007; Kafai et al. 2019; Lau et al. 2009). The combination of art and STEM in STEAM

is applied in pattern design and implementation. The platform Turtlestitch, for example, which is based on Scratch, uses pattern design and programming (Wolz, Auschauer, and Mayr-Stalder 2019). Previous work shows that embroidery programming is an appealing approach to teaching computational thinking skills (Spieler et al. 2018; Gursch et al. 2021; Gursch 2022).

This report is founded on past research on the workflow for the realization of self-created embroidery work in the context of teaching STEAM skills. Some of this research gave examples of a possible embroidery workflow – from drawing, creating, and implementing a design to the final result (Schneider and Boufflers 2019; Wolz, Auschauer, and Mayr-Stalder 2019) – utilizing vector drawing, programming (using Scratch), and image manipulation, among other methods. However, previous research still lacked an optimized workflow, which was the focus of the present work. The use of a smartphone to program the design for the embroidery work offers a particular advantage of our approach: it can be programmed anywhere, and no additional hardware needs to be purchased.

### 3. Methodology

This report focuses on two key aspects: first, the process of creating a design with a smartphone and embroidering it onto fabric, and second, the improvement of the process by developing a new option for wireless data transfer.

#### 3.1 Setting

First, a smartphone (e.g., Samsung Galaxy A50) is needed with the Embroidery Designer app. The app is available in the Google Play Store, the App Store, and the Huawei AppGallery. A smartphone with the Embroidery Designer app can be used to create a DST file, a format that can be read by an embroidery machine. The wireless connection is provided by a Raspberry Pi with a certain software (see Section 3.2). For this, we developed a software prototype that we call CatroPi. The purpose of CatroPi is to transfer a DST file wirelessly to an embroidery machine, thus avoiding the use of a USB stick. Finally, an embroidery machine that can read DST files is required. We have worked with the BROTHER Innov-is M240E Disney and the Innov-is NV2600 sewing and embroidery machine.

### 3.2 *Setting up CatroPi*

This section describes the installation and setup of the prototype CatroPi. This technical description of scripts and other features required for the Raspberry Pi to establish a wireless connection between, in this case, an embroidery machine and a terminal device such as a smartphone or laptop/computer. It should be noted that only a Bluetooth function is required for the end device for file transfer.

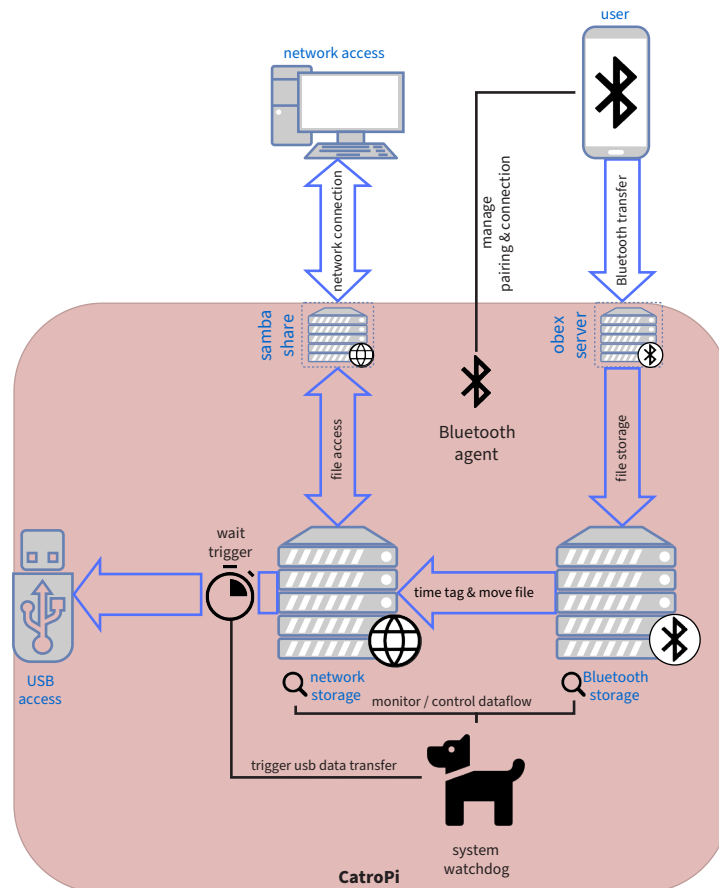
#### 3.2.1 *Wireless connection*

One result of the report is the wireless data transfer with the CatroPi. Figure 1 shows the comparison of the file transfer between the CatroPi and a USB stick. The file transfer through USB stick requires data transfer by e-mail or cloud storage (for instance, Google Drive) to a computer or laptop with a USB port. As a next step, the USB stick can be plugged into the embroidery machine. For the file transfer through CatroPi, the file can be sent directly to the Raspberry Pi that is plugged into the embroidery machine. Therefore, the watchdog monitors the main data flow and handles the file system behavior. As the watchdog is implemented in Python, it is easily adaptable to the specific needs of the individuals using the solution. A network share implemented with Samba provides easy file access via the local network and allows for data management through a simple file explorer. Finally, the Bluetooth agent, which is based on a sample implementation of a BlueZ Bluetooth agent,<sup>3</sup> can be used and modified for the specific security needs of the use case. Alongside the Bluetooth agent, an OBEX server accepts incoming files transferred via Bluetooth. However, the main functionality that enables the device's functionality is its hardware support of USB-OTG,<sup>4</sup> which allows it to act as both a host and a peripheral device.

---

<sup>3</sup> For reference implementation, see: <https://github.com/bluez/bluez/blob/master/test/simple-agent>.

<sup>4</sup> For further information on USB-OTG, see: <https://gist.github.com/gbaman/50b6cca61dd1c3f88f41>.



**Fig. 1:** CatroPi's data flow.

### 3.2.2 Development of the software

Our software implementation was initially designed for use with Raspberry Pi Zero devices. Due to hardware supply issues, however, it was found that it is beneficial to provide software support for a broader range of hardware devices such as the Banana Pi Zero. The installation script should work out of the box, with minimal user input required.<sup>5</sup>

### 3.2.3 Installation script

An installation script has been created to make setting up devices more accessible and less error-prone. Through the script, many parameters of the behavior can be tweaked. The default settings for newly created devices are designed to create an 8GB binary file, which will act as the USB storage for CatroPi and the necessary

<sup>5</sup> For details, see: <https://github.com/Stofflr/CatroZero>.



folder structure for handling incoming files by Bluetooth, and for the network share. Furthermore, the installation script enables the essential modules for the USB-OTG functionality. It configures the Bluetooth interface and connectivity settings, the network share, and all the necessary dependencies and scripts run at boot to enable the device's functionality. At the top of the file `install.sh`, the variables `delay`, `hostname`, `wififile`, `blfile`, `mountfile`, and `datafile` can be modified to customize the behavior and structure of the device. To adjust the file size of the emulated USB stick, the `dd if=/dev/zero of=$datafile bs=1M count=8K &`<sup>6</sup> command in line 34 of the installation script can be modified.

For the installation script to function, the CatroPi Github Repository needs to be cloned or copied onto the device.<sup>7</sup> After the files are on the device, the install script can be executed with root privileges using `sudo bash ./CatroZero/install.sh`.

#### 3.2.4 Bluetooth agent and file handling

A Bluetooth agent is necessary to allow incoming Bluetooth connections and pairing. In our case, we modify a simple reference BlueZ agent. The behavior of this Bluetooth agent can be modified to the specific end user's needs. With a basic understanding of the GPIO of the used hardware device, it would be recommended to display the Bluetooth pairing code, which is not displayed on the current implementation, on a small attachable LCD screen. The Bluetooth agent handles the pairing and connection through the program `obexpushd`,<sup>8</sup> The OBEX (object exchange) protocol over Bluetooth is used to handle the incoming files from the previously paired devices and store them in the corresponding folder defined in the installation script.

The main software component of the device is the filesystem watchdog. It ties together all of the features above and ensures a proper data stream from the network and Bluetooth connection. Once file system changes are detected, they are processed by the watchdog and stored in a process queue. The queue ensures proper handling of larger files, which tend to be processed in chunks. Files transferred by Bluetooth are timestamped, as sending files with the same name would fail otherwise, and the files are moved to the network storage of the system. Once the watchdog detects no further file transfers within the defined delay, it proceeds to transfer the data to the USB-OTG side of the system. Due to the limitations of USB-OTG, the data transfer is only unidirectional. This leads to files needing to be updated and displayed once they are added to the CatroPi. For this reason, a virtual replug is realized by disabling and enabling the USB-OTG mass storage interface, which updates the displayed files on the device the CatroPi is plugged into.

---

<sup>6</sup> For further details, see `dd` command at: <https://manpages.org/dd>.

<sup>7</sup> For the CatroPi Github Repository, see: <https://github.com/StoffIR/CatroZero>.

<sup>8</sup> For further details, see `obexpushd` at: <https://manpages.org/obexpushd>.

### 3.3 Assessment

The observation of the entire workflow from the Code'n'Stitch project framed the crucial steps for the work of this report. In total, 14 workshops (four units of 2x50 minutes) were carried out with 229 students (106 boys, 123 girls) between 10 and 14 years of age. Furthermore, qualitative data were collected on the designs created by the students and their final work (programmed DST file). The workshops were coordinated with the teachers and held in four units (two double lessons each). These four units were:

- Unit 0: drawing a design to be stitched,
- Units 1 and 2: programming the drawn design,
- Unit 3: embroidering the design onto the fabric, laying the foundation for a possible workflow.

In addition, two week-long initiatives were held at Graz University of Technology to assess the workflow and its improvement through wireless data transfer. The first coding initiative in 2021 served to carry out and verify the four steps. Subsequent observations helped to identify existing barriers (such as file transfer). Another obstacle was the realization of the programmable design within the available time. After the coding initiative in 2021, the target state was defined and implemented as described in Section 3.2. The new technology for wireless file transfer, the CatroPi, was used in the second coding initiative in 2022. The observations and feedback of trainers verified the success of this new approach (see Section 4).

### 3.4 Coding Initiative

The one-week event took place as part of a summer course for students at Graz University of Technology. The courses were held for three hours in the morning and three hours in the afternoon. Students at the Graz University of Technology supervised the participants. The supervisors were experts in STEAM fields and had years of experience teaching and guiding young people in STEAM subjects. The *embroidery programming units* were held following the workflow of the Code'n'Stitch project, as demonstrated in Figure 2. Both the girls themselves and their legal guardians gave consent to the report on the coding initiative in 2021 and in 2022. Information on the age and number of participants is shown in Table 1.

	Participants	Average age	Age range
coding initiative	21	13.86	12-18
coding initiative	18	12.56	10-15

**Tab. 1:** Total participants and age range of the coding initiative.

## **4. Results**

The entire process of this work produced two main results. One was the process of a (possible) workflow for creating an embroidery work using a smartphone. The second was the use of CatroPi in an educational setting (the coding initiative Girls Coding Week 2022). This led to the introduction of the first wireless connection for file transfer in the workflow for creating an embroidered product (bag or T-shirt).

### **4.1 *CatroPi: First Usage in a Coding Initiative***

One result of the report was the successful wireless data transfer with CatroPi in an educational setting. First, the entire Raspberry Pi was modified (see section 3.2.) to transfer files via a wireless / Bluetooth connection to a maker device: in this context, an embroidery machine. This made it possible to send a DST file to the embroidery machine with any end device (smartphone, laptop, etc.) wirelessly.

In the coding initiative, participants drew a design that they self-programmed. CatroPi was used to send the design from a smartphone through a wireless Bluetooth connection to the embroidery machine, and the embroidery machine embroidered the DST file onto T-shirts and bags. As the embroidery process can take more than 10 minutes depending on the design, there is usually a queue. Multiple participants had to wait for their designs to be embroidered. While CatroPi is plugged into the embroidery machine, additional files can be sent during the embroidery process. The file then only has to be selected, and the next design can be embroidered directly without waiting or plugging another USB stick in.

The designs can also be changed at the last minute and sent again. Most of the participants programmed with their own smartphones. Many steps were saved here, such as sending the file to a USB-compatible device (laptop), creating cloud space folders, and accessing them. The file transfer process was almost trivial, as most participants managed to send the file to the CatroPi without help.

### **4.2 *Optimized Workflow***

During the Code'n'Stitch project (229 participants), 217 ideas were guided through the workflow (Steps 1–4) from the initial idea to the final embroidered work. We found that the procedure for creating a self-programmed embroidery work using a mobile device includes four steps (Figure 2).

1. Design
2. Programming
3. File transfer
4. Embroidery of design onto fabric.

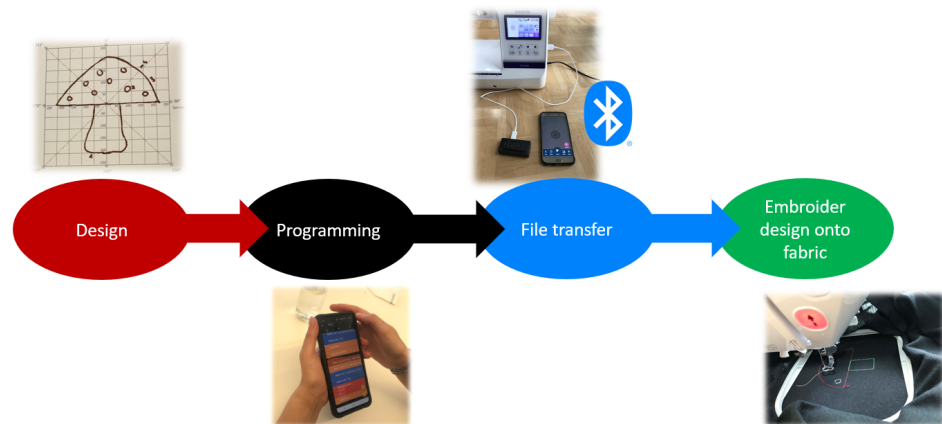


Fig. 2: Process from the design to the final embroidery.

The following subsections describe the results of the workflow for mobile embroidery programming. They also show how it is possible to embroider fabric with an embroidery machine and a self-programmed DST file through a smartphone.

#### 4.2.1 Design

The first step is to create an embroidery design. An idea needs to be drawn or illustrated using a program. The embroidery design should be designed in such a way that it can be logically programmed. We have found that straight lines, circles, or geometric patterns are easy to realize. Shapes, animals, or logos can be implemented as long as they are made up of simple lines and curves. Figure 3 shows programmable embroidery designs.

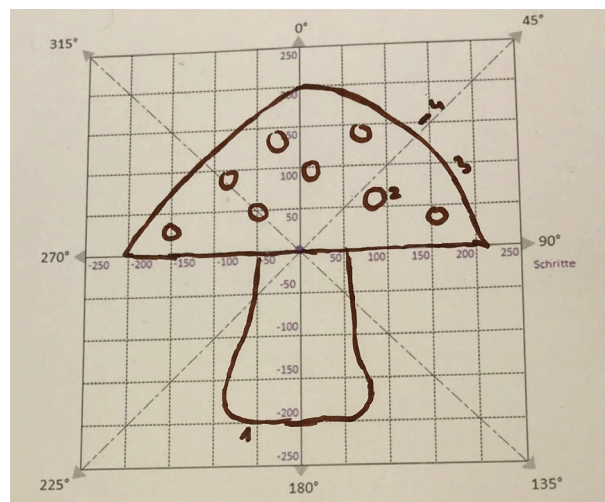


Fig. 3: Step 1, Design.

#### 4.2.2 Programming

The next step is to implement the drawn design (Figure 4). With the help of the Embroidery Designer app (or Pocket Code) on the smartphone, this can be programmed easily. The first steps on how to use the app, how to program anything from simple shapes to complex figures, and how to fill areas, are described on the project's wiki.<sup>9</sup> Once the design has been programmed, it is still necessary to check the program for several key points: One is the size of the self-programmed design (the hoop size sets the bounds and is therefore considered the maximum size). Another are loose threads (jumps) in the program that must be programmed accordingly; otherwise, the work will disintegrate over time after embroidering.

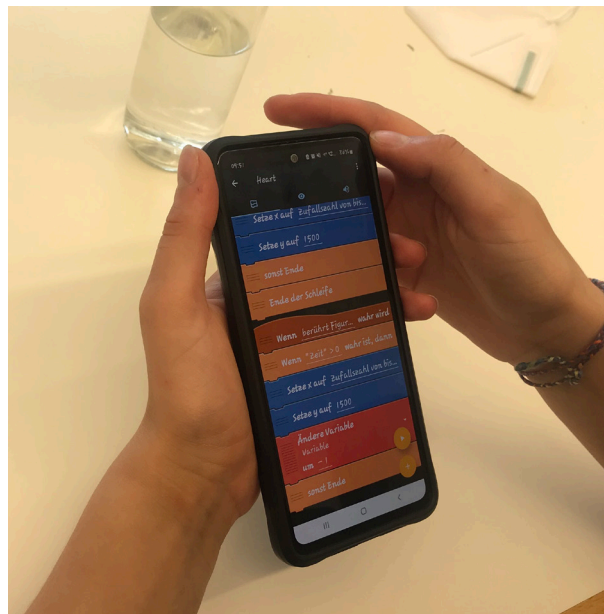


Fig. 4: Step 2, Programming.

#### 4.2.3 File transfer

Once the implementation has been completed according to the design, it can be transferred to an embroidery machine as a DST file. We therefore developed a new way to transfer the file wireless directly from to smartphone to the embroidery machine. The CatroPi must be connected to the machine's USB port. When the smartphone is paired with the CatroPi, the file can be sent directly and will be visible in the embroidery machine's user interface (see Figure 5).

<sup>9</sup> See: <https://wiki.catrobat.org/bin/view/Education/Embroidery/>.



**Fig. 5:** Step 3, File Transfer.

#### *4.2.4 Embroidering the design onto fabric*

First, the fabric to be embroidered must be clamped in the hoop. Depending on the fabric, an embroidery stabilizer should be used. The embroidery thread needs to be selected according to the design. The yarn is threaded through the embroidery machine. The next step is to adjust the embroidery machine settings. This includes choosing the correct needle, thread tension, and embroidery program. Once the settings have been selected, the actual embroidery process can begin. The embroidery machine reads the DST file's design and moves the needle accordingly to embroider the fabric (see Figure 6). When the embroidery machine is finished, check the pattern to ensure there are no mistakes and everything has been embroidered correctly. If necessary, corrections can be made before the hoop is removed from the embroidery machine.



**Fig. 6:** Step 4, embroidering the design onto fabric.

## 5. Discussion

The main findings of this report show that the process from self-designed embroidery idea to embroidered work always depends on specific steps. In this work, these were summarized in the context of an educational project. The report describes four steps: 1) creating a design, 2) programming, 3) transferring the file, and 4) embroidering the design onto the fabric. These steps are crucial to a structured workflow. From the observations, the following has emerged: The design should be kept simple (only lines, circles, or similar), making it easier to create an embroidery work. Depending on the user's programming skills, however, more complex designs or shapes are possible. More advanced users can create code to automatically fill areas, insert variables to change the shape of the programmed design, and much more. There are almost no limits to programming a design as long as the final size of the design does not exceed the hoop size. These findings build on the Code'n'Stitch project with its 14 workshops. The course design aimed to guide each participant (299) from a self-drawn design to an embroidered shirt or bag as the final product. In total, 217 ideas were implemented and thus guided through the entire workflow. With four defined steps, it is possible to guide groups or courses through the process and indirectly introduce basic programming skills. Both coding initiatives followed the workflow in blocked units. One part of this week-long event was mobile embroidery programming. Several participants used more units for this purpose and embroidered more complex designs.

A further result of this report is data transfer (DST files), especially in the context of maker education. We demonstrate a new approach to file transfer through CatroPi. Devices like the Raspberry Pi Zero W allow the creation of a small device

that interfaces with programmable embroidery machines. Once CatroPi is plugged into the embroidery machine, it is possible to send the files (DST file) directly from the smartphone via Bluetooth or a network connection to the interfacing Raspberry Pi. However, especially for collaboration in groups, CatroPi offers the possibility to transfer multiple files at the same time or at random intervals that are not easily predictable. Even though only one person is using the embroidery machine, there is no need to plug another USB device in for a new file, which saves time when it is most needed. The DST file can be sent to the embroidery machine by Bluetooth from different devices. If CatroPi were not used, conducting courses with a smartphone would require additional steps. Since most people use smartphones, and since smartphones have different USB cable connections, connecting the smartphone directly to the embroidery machine by USB creates struggles. In addition, iPhone users face additional obstacles to transferring files via USB and may have to send the files to a computer, copy them onto a USB stick, and plug this into the embroidery machine. We demonstrate how CatroPi can replace the conventional way of using a USB stick. Although this requires setting up hardware (like Raspberry Pi) in advance, once CatroPi is ready, the advantage of sending files directly from the smartphone improves the maker's education. CatroPi has been used on a group of 18 participants. Although this was only done with a small group, it showed a clear advantage over previous courses. The time saved was substantial. If there was no e-mail account on the smartphone, it was necessary to set it up, send the file to a laptop, and then transfer it via USB stick to the embroidery machine. Once the device was paired via Bluetooth, it was possible to send files to the CatroPi plugged into the embroidery machine.

The time saved in groups, especially in maker spaces, is significant with CatroPi. We found that an average skilled educator takes around 45 seconds using a USB stick to get the files from the embroidery app on the smartphone to the embroidery machine (maker device). In comparison, using CatroPi, the time needed to transfer files is either ~10 seconds on the first file sent (due to the pairing process) or ~3 seconds if the smartphone has previously been paired with CatroPi. The regular workflow of sending the files includes sending them via a messaging/mail app to a personal computer, where they can be moved onto the USB drive. It has been found that not only is the time needed for helping students reduced; the perceived impact of interruptions is also less disruptive for the educator workflow, which leads to a better teaching environment. CatroPi allows multiple students to send their work to the device in parallel without having to keep track of a physical USB drive. In addition, CatroPi can receive other files while it is connected to the embroidery machine. In contrast, a USB stick must be plugged in and unplugged from the computer and embroidery machine when a new file is to be transferred. Some of the unpredictable overhead can be avoided through the use of multiple USB sticks.



The time saved for transferring files in courses by using CatroPi is significant. Users can start embroidering as soon as the first file has been sent to CatroPi for embroidery. It is possible to send further files to CatroPi and use them afterward, even if the embroidery process is running. There is no need to reconnect a USB stick with new files and keep track of several sticks. Furthermore, most participants managed to send the file to the CatroPi themselves. No additional support is needed, such as sending the file to an e-mail account or cloud storage and transferring it to a USB stick. Furthermore, the connection via Bluetooth and the file sending does not require the intermediate solution of the USB stick. CatroPi makes it possible to send several files to the embroidery machine simultaneously. Since the embroidery process can take several minutes, another file can be received while it is running. As soon as one file has started the embroidery process, additional files will be received. Furthermore, most participants manage to send the files themselves. The smartphone can be easily connected to the CatroPi by Bluetooth or network.

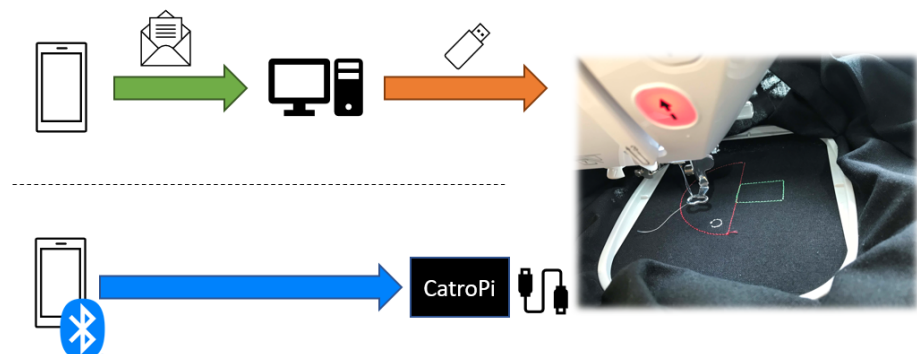


Fig. 7: File transfer, USB vs. CatroPi.

Figure 7 shows the steps for the DST file transfer from USB versus CatroPi. CatroPi allows the DST file to be sent directly to the embroidery machine. With a USB stick, on the other hand, if not all files have been sent, the stick has to be plugged back into the computer and the embroidery machine. Otherwise, several USB sticks might be used, but one must know which file is on which USB stick. There was only one USB port with the machines we have used so far. Most of the time, the participants needed support to send the files by e-mail or save them in a drive folder to transfer them from the laptop to the USB stick. With CatroPi, however, most participants sent the files directly to the embroidery machine without needing any additional assistance.

Overall, CatroPi makes data transfer almost trivial, especially for coding initiatives or courses in an educational setting. The plugged-in CatroPi enables a Bluetooth connection between the embroidery machine and the end devices, users' smartphones. The participants usually accomplish data transfer without any

support from the supervisor, which simplifies course conduct. The trainers, supervisors, or instructors can therefore spend more time supporting the embroidery process or programming the design. Furthermore, CatroPi can be used with other end devices such as 3D printers, offering a further advantage for maker education.

Throughout this report, four steps for creating an embroidery work with an embroidery machine have emerged. These four steps (see Section 4.2) can be used as a basis for implementation, especially in coding initiatives. Teachers, educators, and practitioners can also expand or elaborate on these steps and adapt them to create an individual practice framework. We suggest that computer science and handicraft teachers collaborate in school settings and adapt these steps appropriately. Handicraft teachers could help students create the design and carry out the embroidery process, and computer science teachers could work with the students throughout the programming process.

Further findings proved helpful during the implementation and conduct of these coding initiatives:

1. Depending on the users' programming skills, designs should start simpler (lines, circles, etc.), especially with novices. After beginning to program, users can approach more complex designs.
2. Educators should plan enough time to use the embroidery machine. The process of hooping and embroidering requires time, especially in groups.

## **6. Outlook**

The workflow demonstrated here provides future teachers, curriculum designers, maker educators, and code clubs a possible framework for their mobile embroidery programming lessons. The use of smartphones opens up new possibilities: participants can use apps such as Embroidery Designer or Pocket Code on their smartphones. The course designer only requires an embroidery machine, which costs around 600 euros, to enable participants to embroider self-created work on clothes, bags, and other textiles. Hands-on experience is precious in the field of STEAM: it allows learners to translate concepts and theories into practical solutions. This helps them to understand concepts and systems and how they work in practice. Art and computer science teachers are in an ideal position to offer engaging STEAM lessons.

In addition, there are many possibilities for focusing not only on embroidery works but even on e-textiles (see Figure 8) or even going one step further toward smart textiles (see Figure 9). The control of the smartphone also makes it possible to use its sensors, as the app Embroidery Designer is based on the app Pocket Code and has inherited all its functionalities.



**Fig. 8:** Smart textiles.



**Fig. 9:** E-textiles.

Future work could apply continuous delivery methods to supply the maker devices (embroidery machines, 3D printer) with files. This can be achieved by adding a web interface for easier reachability from the outside. The maker device does not have to be in the same room. In addition, for multiple machines, a load-distributing queue could be implemented that would automatically distribute work to machines that are free to use or available at the moment. Designs can even be embroidered if participants are not physically present at the event. Therefore, multiple events can be hosted in different places, and a code-based login could be used to divide the various locations into separate workspaces. At a later point in time, cooperation with companies (such as clothing manufacturers) would be possible since they could download the embroidery files from a trusted source. In addition, the interface emulating a USB device can be used on machines with similar use cases (e.g., 3D printers). All file types are supported. The machine only needs to be able to use USB devices as a file source.

### References

- Bertrand, Marja G., and Immaculate K. Namukasa. 2020. "Steam Education: Student Learning and Transferable Skills". *Journal of Research in Innovative Teaching & Learning* 13 (1): 43–56. <https://doi.org/10.1108/JRIT-01-2020-0003>.
- Bertrand, Merkouris, Alexandros, Konstantinos Chorianopoulos, and Achilles Kameas. 2017. "Teaching programming in secondary education through embodied computing platforms: Robotics and wearables". *ACM Transactions on Computing Education (TOCE)* 17, (2): 1–22. <https://doi.org/10.1145/3025013>.
- Buechley, Leah, Mike Eisenberg, and Nwanua Elumeze. 2007. "Towards a Curriculum for Electronic Textiles in the High School Classroom". In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 28–32. <https://doi.org/10.1145/1268784.1268795>.
- Buhnova, Barbora, and Dita Prikrylova. 2019. "Women Want to Learn Tech: Lessons from the Czechitas Education Project". In *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*, 25–28. <https://doi.org/10.1109/GE.2019.00013>
- Connor, Kenneth A., Bonnie H. Ferri, and Kathleen Meehan. 2013. "Models of Mobile Hands-On STEM Education". In *2013 ASEE Annual Conference & Exposition*, Paper ID 23-910. <https://doi.org/10.18260/1-2--22295>.
- Devi, Sarita, Nirmal Yadav, Nisha Arya, and Sushila. 2019. "Digital Embroidery: An Imagination". *Journal of Pharmacognosy and Phytochemistry* 8 (6): 127–130.
- Evans, Brian. 2012. *Practical 3D Printers: The Science and Art of 3D Printing*. Apress.

- Figueroa, Nadia, Haiwei Dong, and Abdulmotaleb El Saddik. 2013. "From Sense to Print: Towards Automatic 3D Printing from 3D Sensing Devices". In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 4897–4904. <https://doi.org/10.1109/SMC.2013.833>.
- Galaleldin, Mohamed, Francois Bouchard, Hanan Anis, and Claude Lague. 2016. "The Impact of Makerspaces on Engineering Education". In *Proceedings of the Canadian Engineering Education Association (CEEA)*. <https://doi.org/10.24908/pceea.v0i0.6481>.
- Gursch, Sarina, Vesna Krnjic, Katja Urak, Michael Herold and Wolfgang Slany. 2021. "How to Encourage Girls to Code Through Embroidery Patterns". In *4th International Conference on Gender Research: ICGR 2021*, Curran Associates, Inc., 122–129. <https://doi.org/10.34190/IGR.21.041>.
- Gursch, Sarina, Katja Urak, Michael Herold, Stefan Kutschera, Silvia de los Rios Perez, Rebecca García-Betances, Maria F. Cabrera-Umpierrez, Yolanda Ursa, Wolfgang Slany, and Vesna Krnjic. 2022. "Inequalities for Women in Science, Technology and Innovation". In *ICGR 2022 5th International Conference on Gender Research*, Academic Conferences and Publishing Limited.
- Gursch, Sarina. 2022. "Coding Initiative Provides Different Approaches to Inspire Girls for Programming". In *30th International Conference on Computers in Education Conference. Proceedings Volume I*, 655–659.
- Kafai, Yasmin B., Debroah A. Fields, Debora A. Lui, Justin T. Walker, Mia S. Shaw, Gayithri Jayathirtha, Tomoko M. Nakajima, and Joey Goode. 2019. "Stitching the Loop with Electronic Textiles: Promoting Equity in High School Students' Competencies and Perceptions of Computer Science". In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1176–1182. <https://doi.org/10.1145/3287324.3287426>.
- Kelleher, Caitlin, and Randy Pausch. 2007. "Using Storytelling to Motivate Programming". *Communications of the ACM* 50 (7): 58–64. <https://doi.org/10.1145/1272516.1272540>.
- Lau, Winnie W., Grace Ngai, and Stephen CF Chan, Joey CY Cheung. 2009. "Learning Programming Through Fashion and Design: A Pilot Summer Course in Wearable Computing for Middle School Students". In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 504–508. <https://doi.org/10.1145/1508865.1509041>.
- Liao, Christine. 2016. "From Interdisciplinary to Transdisciplinary: An Arts-Integrated Approach to Steam Education". *Art Education* 69 (6): 44–49. <https://doi.org/10.1080/00043125.2016.1224873>.
- Marques, Lucas G., Robert A. Williams, and Wenchao Zhou. 2017. "A Mobile 3D Printer for Cooperative 3D Printing". In *2017 International Solid Freeform Fabrication Symposium*, University of Texas at Austin.
- Schneider, Daniel K., and Lydie Boufflers. "Digital embroidery to teach ict skills". Poster presented at AECT (2019).

Spieler, Bernadette, Vesna Krnjic, Wolfgang Slany, Karin Horneck, and Ute Neudorfer. "Design, code, stitch, wear, and show it! mobile visual pattern design in school contexts". In *2020 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2020, pp. 1–9. <https://doi.org/10.1109/FIE44824.2020.9274120>.

The Conversation, 2018. Accessed November 12 2023. "Explainer: what's the difference between stem and steam?" <https://theconversation.com/explainer-whats-the-difference-between-stem-and-steam-95713>.

Wolz, Ursula, Michael Auschauer, and Andrea Mayr-Stalder. "Code crafting with turtlestitch". In *ACM SIG-GRAPH 2019 Studio*, 2019, pp. 1–2. <https://doi.org/10.1145/3306306.3328009>.

### **Acknowledgements**

This project received funding through Talente: FEMtech Research projects 2017, Code'n'Stitch, eCall/FFGNr. 14975187 / 866755. Our research was supported by the Office for Gender Equality and Equal Opportunity of Graz University of Technology. We thank our colleagues from the Institute of Software Technology (Graz University of Technology) who provided insight and expertise that greatly assisted us in our research. We would also like to thank the students at Graz University of Technology for their support in carrying out the coding initiatives.