

1. Introduction

3D printers are currently seen as symbols of digital and technological progress. While demand for many tech products is declining, the 3D printing market seems to be growing steadily (Footnote 1). Even for the private user, it seems to be a technology that one might not have yet but would definitely wish to (Footnote 2). In major electronics stores, the visionary possibilities of a 3D printer are presented to astonished customers as if at a trade show booth from the future. However, it usually comes down to printing items that are either put on a shelf for display or thrown away. The vendors are happy to omit any reference to the negative impact on nature and the environment. Does it really feel good to finally be able to produce plastic items at home? Not for us. At present, the production of high-quality plastic items should still be left to professionals. Unfortunately, the availability of affordable 3D printers (still) comes at the expense of quality and the environment. However, many people have already bought a 3D printer and may find that they hardly use it. Instead of throwing it away, we want to expand the uses of just such cheap 3D printers, so that even without using plastic, the benefits of this technology can be put to good use. We turn 3D printers into musical instruments and want to share our somewhat crazy idea with all fans of the OER community.

With our tips and tricks, a dusty 3D printer can be turned into a device that inspires interaction without waste (upcycling). Many parts of the printer remain intact and do not have to be thrown away (sustainability).

We decided to convert the printer into a musical instrument because there are some musicians in our team. However, the printer could also be used as a “controllable brush” for creating two-dimensional works of art. Either by hand (e.g., by a joystick) or by a programming algorithm. Regardless of how you convert the printer, what matters is not only the result but also the process of understanding. Thus, a discarded 3D printer is also well suited for interdisciplinary teaching with an orientation towards projects, actions, and products that combines technical and aesthetic education.

Footnote 1)

The market value of 3D printers was over \$10 million in 2019 and is expected to double every three years (cf. AMFG, 40 3D Printing Industry Facts You Absolutely Need to Know, URL: <https://amfg.ai/2020/02/01/40-fakten-der-3d-druckindustrie-die-sie-unbedingt-wissen-sollten-2020/>, retrieved on 09/13/2020).

Footnote 2)

Almost everyone in Germany owns a laptop or a cell phone. However, only 14% of Germans have a 3D printer, but more than half could imagine buying one (cf. Survey: 3D printers on the advance into German private households, URL: <https://www.reichelt.de/magazin/studien/umfrage-3d-drucker/>, retrieved on 09/13/2020).

2. Which instruments are compatible and appropriate?

In principle, 3D printers can be used to produce sound with many different instruments. Both rhythm and melody or harmony instruments are compatible. To cover these types of instruments, we have chosen the güiro, the zither, and the ukulele as examples.

- The güiro is capable of generating simple rhythms. For this, the printhead is capable of rubbing the instrument in one direction as well as two directions.
- The zither was specially made for the 3D printer so that it can play as many different notes as possible. It has room for ten strings, eight of which form an octave in E-flat major (Eb5, F5, G5, Ab5, Bb5, C6, D6, Eb6). The remaining two strings are a minor second lower (D5) or fourth (Bb4). This allows many simple melodies to be played with a guitar pick on the printhead. It should be emphasized that the latency of steps (conjunct motions) is significantly lower than that of skips (disjunct motions), which require the instrument contact surface to lower briefly. Therefore, the musical result of melodies consisting mostly of steps is rhythmically much more appropriate for the 3D printer (e.g., “Alle meine Entchen”, and the beginning of “Freude schöner Götterfunken”, etc.). In addition, two pickups are installed so that the generated tones can be further processed electronically (amplifier, effect units, etc.).
- The ukulele works in principle like the zither, except that it comprises only 4 strings and can thus produce a chord with a quick movement of the printhead. One could thus program a simple strumming pattern where, for example, a power chord (root, fifth, octave, fifth) is played in a loop.

3. The basics of 3D printing

The standard 3D printers we are familiar with belong to the additive manufacturing technologies (figure 1), which create finished products from the layer-by-layer application of shapeless material through a variety of techniques. The materials used have diversified after years of development. Not only plastics, but also powdered metal alloys, biological materials such as living cells, and even ceramics are used in manufacturing today to produce finished products with complex geometries that cannot be reproduced in any other way. Some say that 3D printing is driving a new industrial revolution. Among the fields in which these technologies are being used are the medical, aerospace, automotive, and energy industries.

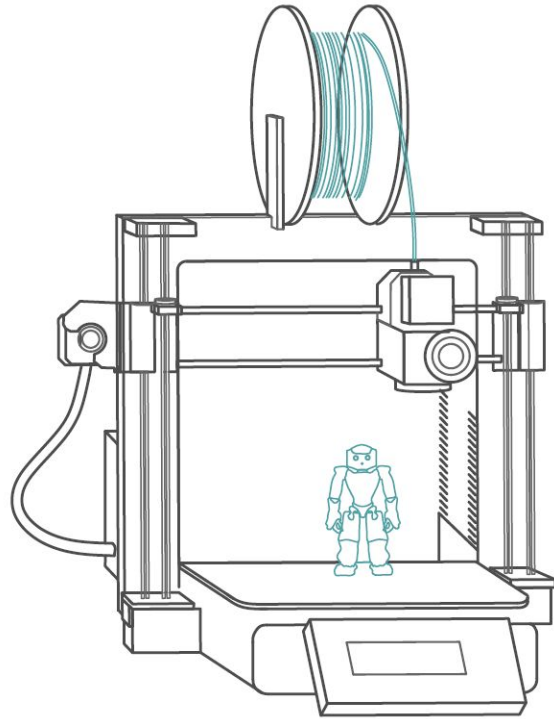


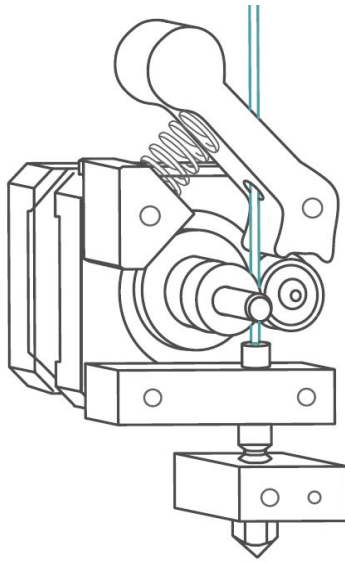
Figure 1: 3D printer (type: Cartesian)

The actual process of 3D printing starts with the designing of objects in CAD (computer-aided design) programs. For the printer to print the geometry of the object in layers, a process called “tessellation” is used to approximate its geometry with triangles in the CAD program. If you want to find more information on the design of 3D printable objects, you can check out these sites:

- <https://www.sculpteo.com/blog/category/3d-design-tips-software/>
- <https://all3dp.com/1/best-free-3d-printing-software-3d-printer-program/>

In this OER, we will focus only on 3D printers that use plastics because they are more accessible and less expensive. One of the most popular technologies in this field is “Fused Filament Fabrication” (FFF), also called “Fused deposition modeling” (FDM). If you are interested in other technologies, we recommend this site:

<https://www.sculpteo.com/en/3d-learning-hub/>



FFF uses a plastic filament that is wound in a spool. This filament is continuously fed through an extruder and at the same time partially melted. The material is then extruded through a nozzle and applied to a plate or print bed. The printed layers cool down and begin to harden. Meanwhile, new material is deposited. Some printers are additionally equipped with fans on the print head, which accelerate the hardening process and thus reduce the printing time. The three-dimensional object is created in this way by the repeated deposition of layers of molten material on top of each other. Figure 3 names the various parts of the printhead. Keep in mind that each printer manufacturer has its variations in the design of the parts.

Figure 2: Extruder

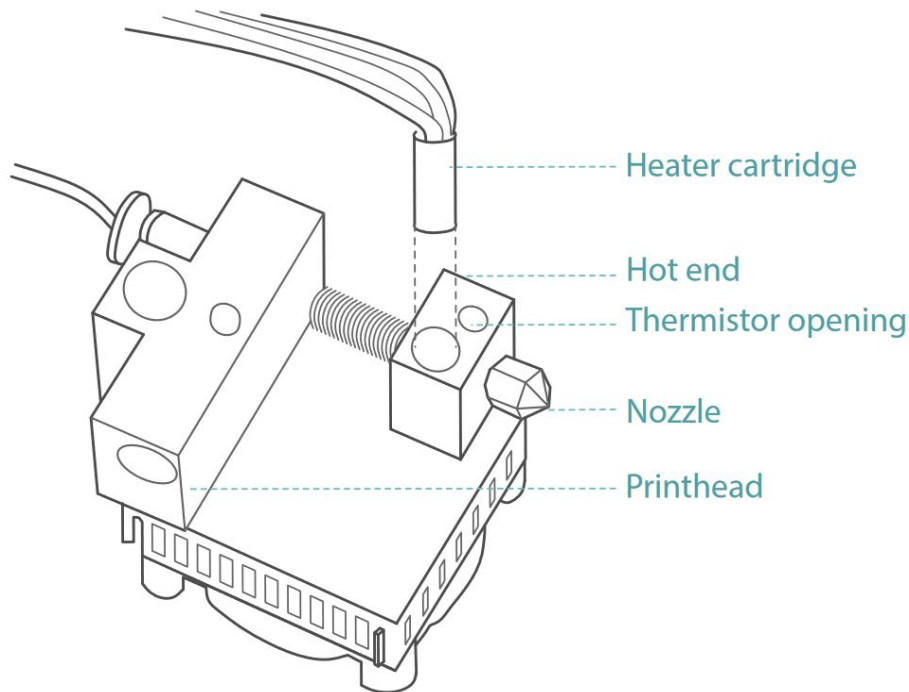


Figure 3: Names of important parts

The nozzle and the whole printhead can be moved by a mechanical system of stepper motors and toothed belt drives, but how does the 3D printer know how fast and in which directions to move the printhead or even what temperature to bring the filament to? These tasks are performed by a microcontroller running specific firmware. An example of 3D printer firmware is

Marlin, which controls the motors, heater cartridge, fan, thermistor (temperature sensor), and other sensors and actuators. Marlin is an open-source firmware developed for the collaborative project RepRap (**re**plicating **rap**id prototyper) and can be configured via the Arduino development platform. RepRap aims to develop machines that can replicate themselves and thus manufacture complex products without extensive infrastructure. For more information about RepRap visit their website:

- <https://reprap.org/wiki/RepRap>

Thanks to its open-source code, we found in Marlin an interface through which we can communicate with the printer. This enables us to configure the printer to play a musical instrument. The software we used to accomplish this is made available for free in this OER for you to use. If you want to learn more about Marlin and Arduino to write your own codes, you can visit the following sites:

- <https://marlinfw.org/>
- <https://marlinfw.org/docs/configuration/configuration.html>
- <https://www.arduino.cc/en/Tutorial/HomePage>

To make the printer play a musical instrument, one must understand how the printhead moves in the print area. 3D printers are classified depending on the directions in which the various parts move and the extent to which the movement is constrained. In the printer we use, the print bed can move in the Z direction (vertical) and the printhead can only move in the X and Y axes (horizontal plane). Printers of this type are called Cartesian printers. Another type is the polar printers, which have a round print bed that rotates through a certain range of angles. The movement mechanism of the delta printers is much more complicated, making them more flexible. If you are not sure whether your printer is Cartesian or not, you can find all printer types and examples on this page:

- <https://3dsourced.com/3d-printers/types-of-fdm-3d-printer-cartesian-delta/>

The software we will present in this OER has been tested on Cartesian printers. You may need to modify the software for a different type of printer.

4. 3D Printer modification

Duration: 30 - 60 minutes

Difficulty: easy

 [Tutorial Video](#) 

Preparation:

To modify the 3D printer, you will need the following tools:

- Screwdriver (PH 0, slotted).
- Pliers
- Side Cutter (Diagonal pliers)
- Hex key
- 1 guitar pick in various thicknesses
- 1 screw with matching nut (slightly longer than the length of a heater cartridge) and washer, if necessary.
- Cordless screwdriver with a 3mm metal or PVC drill bit.
- 1 cotton swab and wooden or metal sticks, if necessary.
- Clips or clamps in three different sizes and Powerstrips if necessary.
- Twine or string.

Since all 3D printers are different, we will show an overview of each step rather than provide detailed documentation. It basically involves removing two parts from the 3D printer: the *extruder* and the *hot end*. The video provides a visual explanation of the main parts of the 3D printer and their location.

Step 1: Remove the 3D printer parts

We recommend that you remove the filament from the 3D printer in advance. Put the 3D printer on a table and unplug all cables. In case you used the 3D printer in the last hour, you must wait until the *hot end* is cold again. Do you have all the tools at hand? Then you are ready to go!

- *Filament and spool holder*

If the 3D printer still has *filament*, you must first separate the *filament* from the *extruder* (break the *filament* by hand). You can remove the *spool holder* from the 3D printer with the pliers and screwdriver.

- *Extruder*

Unplug the *extruder cable*, unscrew the *extruder* and remove it. Depending on the length of the *extruder tube*, you can either put the *extruder* on the table or just leave it hanging. You will remove this later.

- *Fan and hot end*

To make the next steps easier, remove the cable tie that secures the *protective spiral* to the enclosure.

Push with your hand the *printhead* into a corner. This will make it easier for you to separate the *fan* from the *hot end* with the screwdriver. The *hot end* contains the *thermistor* (temperature sensor) and the *heater cartridge*. Both have to be removed from the *hot end*. To remove the *heater cartridge*, you need the Hex key. You may have to push the *printhead* again with your hand.

The *protective spiral* must also be removed from the *printhead*. It is attached to the *printhead* with a screw.

Now you can unscrew the *hot end* from the *printhead*. A pair of pliers would be handy here. Put the *hot end* on the table, you will need it later.

Now you need to unscrew and remove the *extruder tube*. You may need the pliers for this step. If there is still *filament* in the tube, apply some force to pull the *filament* out of the *printhead*.

The *extruder* and *hot end* are no longer attached to the 3D printer. The cables of the *fan*, *extruder*, and *heater cartridge* are still connected to the *printer controller board*.

- *Printer controller board*

To access the *printer controller board*, you must turn the print bed upward by hand and remove the bottom cover of the 3D printer (attached with screws). Now you can work on the *printer controller board*.

Remove the *heater cartridge cables* from the *printer controller board* with the screwdriver, unplug the *extruder motor cable* and the *thermistor cable* from the *printer controller board*. **Important:** make sure you remember the *thermistor* connector; *the thermistor will be plugged in again later*.

The fan cable must also be removed from the power supply. Plug the *thermistor* back in and leave it on the bottom of the 3D printer.

For safety reasons, 3D printers do not function without a *thermistor*. You can just leave it on the bottom of the 3D printer. Put the bottom cover back on and now you are done with this step.

Step 2: Modify the hot end

In order for the 3D printer to play a musical instrument, you need to modify the *hot end*. We have designed two variants: one variant for stringed instruments (zither and ukulele) and another for percussion instruments (güiro).

Depending on the variant (stringed instrument or percussion instrument) you have to separate the *nozzle* from the *hot end* (stringed instrument) or you can leave it where it is (percussion instrument).

If you need to remove the nozzle, proceed as follows: the *nozzle* is probably stuck in place due to the melted *filament*, in this case you need to apply a lot of force using one or even two pairs of pliers. Applying a little heat to the nozzle will also help. Fortunately, we did not need to do this...

Variant 1: Guitar pick

Here you need a guitar pick and a screw that is slightly longer than the length of the *heater cartridge*, so that you can attach a suitable nut to the other end. If the opening in the *hot end* is bigger than the nut, you can simply put a washer between the *hot end* and the nut.

You need to find the right guitar pick for your instrument. Guitar picks come in different thickness (very soft to very hard). You need to find out the right thickness by yourself. We have opted for a soft thickness.

To attach the guitar pick, drill a small hole in the upper middle area of the pick. You can then simply attach it to the *hot end* with the screw. The guitar pick must be attached firmly, otherwise, you may have problems later at the programming stage.

Variant 2: Stick

The tube joint from the extruder tube will be helpful. With it, you can easily insert a cotton swab. To separate the tube joint from the extruder tube you will need a side cutter. The cotton swab must be inserted into the tube joint. The cotton swab is now our güiro stick. You can also use different materials for the güiro stick, such as a long screw or a round stick.

If you want more, you can also use the opening in the *hot end* that you used for mounting the guitar pick to attach a second wooden or metal stick with which you could play a second percussion instrument.

What instruments come to your mind? We do not have any programming for this expansion yet.

Step 3: Fasten the modified hot end and instrument

The modified *hot end* with guitar pick or stick is screwed to the cylinder (with screw thread) under the printhead so that it is attached to the 3D printer.

After that, you can attach your instrument to the print bed. Since the print bed is mobile, the instrument must be well secured, otherwise you may have problems later at the programming stage.

1. To attach the zither, we decided to use a clamp or clip. It is very stable and it attaches to the print bed at the middle of the zither.
2. To attach the ukulele, we decided to use Powerstrips, because they are easy to remove. This way you can easily “stick” the ukulele to the printing table. You can also use a large clamp.
3. You can also attach the güiro to the 3D printer with a twine. It is also possible to attach it with screws. However, we did not want to drill a hole in the güiro and therefore decided to use twine.

5. Construction of the zither:

Our zither prototype is custom made from wood and metal. Holes or cavities were milled into the wooden board (52 x 15 x 1.8 cm) to insert the various electric guitar parts.

The important question now is how to best attach the strings to the wooden board. You can use simple bridges/nuts over which the strings can be stretched. However, we decided to go with hardtail bridges like you find on electric guitars. Since we have 10 strings, we need a total of 4 electric guitar bridges (hardtail bridges). This makes it easier to change the intonation of the individual strings as well as their height. In addition, two single-coils were used as pickups, because they take less space than humbuckers, for example. Single-coils with a chrome-plated cap make the whole conversion easier. The area accessible by the printhead (130 mm Y-axis) offers space for a maximum of 10 strings. To cover as many melodies as possible with it, we decided to use a diatonic scale in E-flat major (Eb5, F5, G5, Ab5, Bb5, C6, D6, Eb6), which offers a minor second (D5) as well as a pure fourth (Bb4) downwards. This allows for playing, for example, upbeat melodies that begin with a fourth (e.g., the well-known "Tedeum" by Marc-Antoine Charpentier). For our 10 strings we use guitar strings. Since a guitar string set usually consists of six strings, we used the five higher ones (A, D, G, B, E) twice and tuned them as described above. Since some are more tensioned than others, (e.g., to get from an A to a D) it was necessary to use strings of different gauges (see table). The reason for raising the notes of our zither by an octave is that the string lengths of our zither are only half as long as those of an ordinary guitar. This provides a higher pitch range, which seems to be more suitable for melodies in particular.

String	Pitch	Guitar string	Gauge	Interval
10	Eb6	E	9	8
9	D6	E	11	7
8	C6	B	11	6
7	Bb5	B	14	5
6	Ab5	G	16	4

5	G5	G	18	3
4	F5	D	24	2
3	Eb5	D	28	1
2	D5	A	32	7
1	Bb4	A	38	5

Construction of the zither step by step:

Duration: 60 - 120 minutes

Difficulty: hard

 [Tutorial Video](#) 

Preparation:

To build the zither yourself you will need some electric guitar components and some tools.

General tools	Components
Screwdriver PH 0 Drill Ruler or tape measure Tools for the pickups: 15mm Forstner drill bit Flat and round files Chisel Jigsaw Side Cutter (Diagonal pliers) Tools for the tuner set: Top: 8 mm wood drill Bottom: 10 mm wood drill Round file	Light wooden board (52 x 15 x 1.8 cm) Guitar jack socket 2 single-coil pickups (preferably both with chrome-plated caps) 4 electric guitar bridges (hardtail bridge) 2 tuner sets 4 electric guitar string sets (spare parts included)

Tools for the guitar jack socket:	
12 mm Forstner drill	
Soldering station and soldering iron	

Once you have all the necessary components and tools, you need to determine the available space on your 3D printer. The zither will be designed on paper according to the size and print area (X or Y) of the 3D printer.

How many strings will fit on the print area? *The first string should not be on the edge of the print area. The guitar pick should be separated from the first and the last string by at least 1cm, otherwise it will not be able to pluck the strings.*

How big should the wooden board be? *Since your zither will be moved inside the 3D printer, the zither and the 3D printer should be at least 3 mm apart.*

You can use different patterns for your design: the zither can also look like a guitar. But this will require more time to make the wooden board. Our pattern is quite simple and basic. It does not look very fancy... yours could be prettier.

You should make sure that the pickups are placed on the print bed of the 3D printer, so that the weight of the zither is better distributed on the 3D printer. Also, the wooden board must be positioned in the middle of the 3D printer.

Once you have all your ideas on paper, you can cut the wooden board with the jigsaw. Then you can place all the parts on the wooden board in the right position and mark these positions (the pickups in the middle, the electric guitar bridges and the tuners towards the strings, and the jack near the pickups).

Drilling the wooden board

The best place to start is with the microphone holes. To make these holes, you first need to drill two holes at the corners using the Forstner drill bit. The 15 mm Forstner drill fits perfectly for the single-coil pickups. Once the two holes are done, you can use the jigsaw to finish the job. Since the pickups are larger in the lower part, you will need to use the chisel and files to widen the hole so that the microphone can be inserted into the wooden board. The upper part of the pickup may be up to 3 mm above the wooden board. Once the pickups fit into both holes, you can move on to the next step. You will install and attach the pickups later.

The holes for the tuners are an easier task compared to the microphone holes. You can drill the holes with an 8 mm wood drill bit. Since the tuning posts of the tuners have two different sizes, the hole on the back part of the wooden board has to be a bit wider. We drilled through about 5

mm with the 10 mm wood drill bit. Now you can check if the tuner fits into the hole. You can enlarge the hole with the file. Depending on how many strings your zither has, you will have to drill a corresponding number of tuners holes. You will install and fasten the tuners later.

The last hole is for the guitar jack socket. With it, you can connect the zither to your amplifier. Here we will drill a hole in the shape of an "L". We will start from one side of the wooden board and the hole will end on the back part of the wooden board. Using a 12 mm Forstner drill (or the size of your jack socket), you need to drill a hole 7cm deep (or the length of your jack socket) in the side of the zither. This hole should be near the pickups. Since your wooden board is very thin (about 18 mm), you need to be very precise. The Forstner drill could easily break the wooden board if it has even a minimal tilt. Did everything work well? Now you have to drill through the wooden board from behind about 10 mm until you hit the first hole. Now the "L" hole is ready.

Using the chisel, you need to remove about 2 mm from the wooden board between the pickup and the jack socket. This is where the pickup cable will be placed later.

You are now done with the holes. To make your zither look fancier, you can paint and varnish it now. This is not covered in this tutorial.

Assemble the parts

Since the microphone holes are difficult, it's best to attach the **pickups** first. The pickups must be inserted from behind. The upper part of the pickups can be up to 3 mm above the wooden board. You can now fasten the screws that come with the pickups with a PH-0 screwdriver. You will take care of the electronics in a moment.

When the pickups are fixed in place, you have to fasten all the **electric guitar bridges** with the supplied screws and the PH-0 screwdriver. Are the bridges in the correct position in relation to the pickup? If not, you need to place the bridges on their correct positions and then fasten them. You can also use a 3 mm wood drill to drill the hole for the bridge screws.

Then insert the **tuners** into the holes and fix them in place.

Now you will work on the **jack socket**. Insert the jack socket from the side and fasten it. Once the jack socket is fastened, you have to connect the pickup cables to the jack socket and solder them. Please pay attention to the diagram in the animation.

Now you need to cover the bottom of the wooden board with cardboard and put on the guitar strings, tune them and, of course, test your homemade zither.

6. PROGRAMMING


I. Introduction to Pd

Pure Data, for short **Pd**, is a visual open-source programming environment that runs easily on all kinds of computers and devices (like smartphones or Raspberry Pi). Moreover, Pd can be controlled by any input devices, such as Nintendo Wii remotes, keyboards, and other music devices, sensors, or the like.

In the past, it was common for computer programmers to use text-based programming languages to develop applications. Programmers wrote lines of code in a file and then executed them to get results. For non-programmers, this was not an intuitive way to creatively engage in coding.

Using a Visual Programming Language such as Pd, a user can create highly complex interactions by editing graphical elements rather than specifying them with textual commands. The text lines describing the functions of a program are replaced by visual “building blocks” that can be copied, exchanged, or edited many times and at will. In a similar process to connecting, disconnecting or reconnecting real wires to a breadboard, we can effortlessly create new programs, also called **patches**. In doing so, we change the behavior of our program by connecting the visual components together and allowing or preventing communication between them.

In this sense, Pd offers a unique form of programming that is much closer to our intuitive and familiar way of dealing with physical things, such as chaining guitar effects together or building a Lego set.

 Pd comes from the family of programming languages originally developed by Miller Puckette at the Research Institute of Acoustics/Music (IRCAM) in France, which also includes Max/MSP (a commercial program). Programs in this class are often used for live electronic music performances, interactive art installations, for audio programming in video games, and for controlling robots, as expansions for acoustic instruments, and all kinds of other creative interactions.

For example, a simple microphone can be used to change the chromaticity of a video depending on the pitch sung. Similarly, the shadow image of a dancer can change the playback speed or volume of a sound recording by using a webcam.

In Pd, the operation steps shown are actually executed directly. This means that each action is already effective during programming, i.e. at the very moment it is completed, which has an immediate effect on the output.

This is an advantage of Pd that allows both users and audiences to immediately perceive the results of any action, which makes Pd a powerful tool for artists of all kinds who wish to use data from a wide variety of sources with unusual devices as musical instruments, for example.

In our case, outdated 3D printers can create music in a unique way. The possibilities are endless!

II. Pre-use checklist for Windows-based devices

- ✓ [Download and install](#) Arduino 1.8.13. When installing, you must also install the Arduino drivers.
- ✓ Connect your own 3D printer to the computer via USB and set it up according to the model. Write down the COM-Port number used.
- ✓ [Download and install](#) Pure Data 0.51. After successful installation of Pd, the program is ready for use. Before we can play with our **Pimp my 3d-printer** patch, we first need to establish and calibrate the connection between computer and printer. Let's go!
- ✓ [Download and extract](#) the Pimp my 3d-printer ZIP file.

III. But how does it work? The essentials of Pd

The Pd console

When you first open Pd, you will notice an empty window. This is the console, which can be accessed at any time by selecting the appropriate option in the "Window" menu. You can safely ignore this window as a beginner, but it is certainly of interest for people who want to take a look at the inner workings of Pd. Here you can see all the notifications that are important for running the program.

Edit mode and execute mode

Once a patch has been opened, Pd offers two modes of operation: edit mode and execute mode. To see which mode we are in, we just have to look at the mouse pointer: if you move the mouse over a button in execute mode, you will see an arrow on the pointer. If you are in edit mode, the mouse pointer becomes a pointing hand.


When you open a Pd patch, it automatically starts in execute mode. In this mode, you can only operate the buttons and knobs that are necessary for the intended applications, without being able to make changes in the basic operation.

However, it is very handy to be able to switch to edit mode, where you can look “behind the scenes” and manipulate the patch structure. As long as a Pd patch is open and in edit mode, we can freely connect its basic elements by lines (“cables”) to graphs that represent the data flow within the program. As mentioned above, this has an immediate effect!

Basic elements

The building blocks of Pd are mainly **objects**, **messages**, **numbers** and **symbols**, plus the connections between them mentioned above. If you connect several of them to a useful, reusable function, they are called **abstractions**.

For more info, see the **help-intro.pd** patch.



 *The help patch with an overview of (almost) all objects can be easily accessed by right-clicking on an empty space in any patch and selecting “Help”.*

Alternatively, you can read the detailed documentation by selecting “Pd Help” in the main “Help” menu.

Each **abstraction** is basically a Pd program itself, whose inputs and outputs not only allow it to be linked into complex structures, but also to be used several times at will.

Expansion using externals

However, this requires further expansions in Pd. These are additional functions that the user community of Pd has developed and implemented over the years to upgrade the basic environment of the program through so-called **externals**. We can use these for a variety of purposes. For example: modeling virtual physical objects, video processing, or streaming and manipulating MP3 audio files are all exciting uses that are not supported out of the box in Pd. Externals exist for exactly these purposes, and can be easily installed from within the base program. Externals are usually found in compact collections of related uses, also known as **libraries**.

If you want to install externals in Pd, you can watch the following  [video](#) .

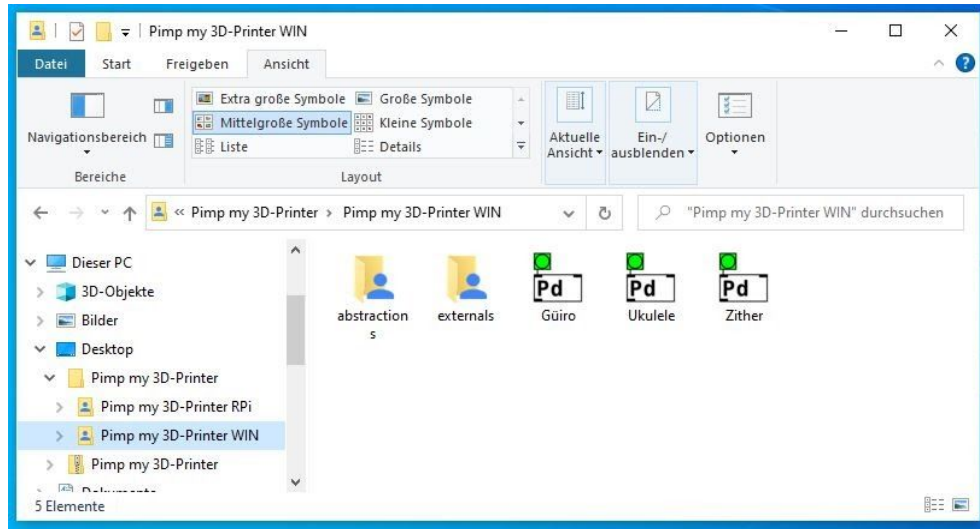
Fortunately, our patch is already ready to use with all the necessary externals and libraries in one folder. It means you do not need to install any additional externals.

Below we explain all three pre-made instrument patches for our 3D printer.

IV. Pimp my 3D-Printer explained

Introduction

As soon as you open the *{Pimp my 3D-Printer WIN}* folder, you will see the following directory:



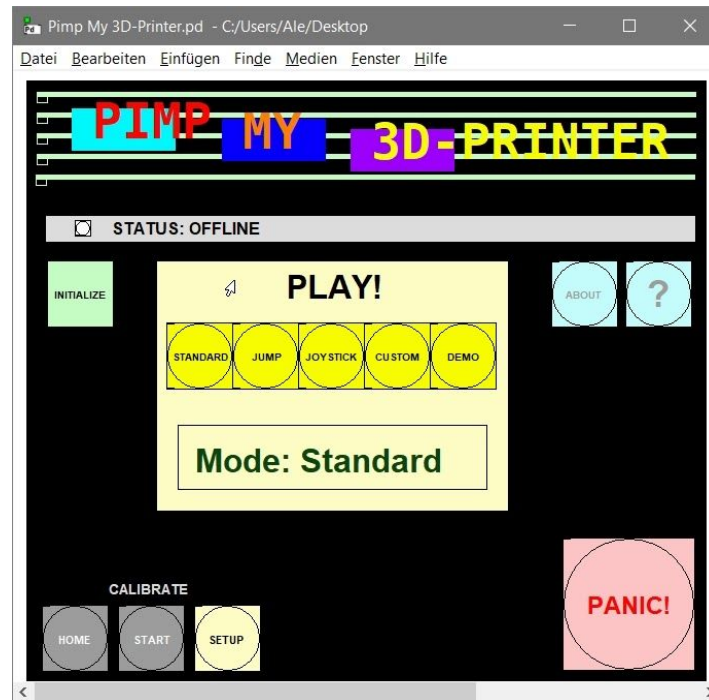
There you will find our instruments, the zither, ukulele and güiro, each in the form of a Pd patch. We will first explain how to control and edit the program using the most complex of these three: **zither.pd**. At relevant passages, you will also find hints for using the patches **güiro.pd** and **ukulele.pd**.

In addition to the individual Pd patches, you will find two important folders:

1. The **abstractions** folder contains some necessary Pd snippets provided by the Pd community, which we are allowed to use in our patch. If you want to extend your own Pimp my 3d-printer patch with external **abstractions**, you can add them here.
2. Likewise, the **externals** folder contains additional material that our program uses in the form of libraries. These files are very important, because their functions affect our patch extensively.

Let's get started with the **zither.pd**!

Overview of the user interface



When opening the patch, you can see several buttons for selecting the playing modes. Above them there is a status bar with a gray background, which shows whether there is a connection to the 3D printer (online) or not (offline).

In order to be able to use the 3D printer as a zither (or güiro or ukulele) at all, you must perform a one-time, small-scale calibration. For this you will find instructions under the **?** button, which you have to follow first. The corresponding setup options can be found under the yellow **SETUP** button. After the setup you can reset the printer to its zero position (**HOME**) or bring it to any calibrated playing position (**START**).

The **PANIC!** button ensures that in the event of a mechanical override or error, the current operation can be aborted to prevent damage to the hardware and motor.

Important: If the **PANIC!** button does not work, you should remove the power cord from the 3D printer and close Pd, then try again.

Bonus:

If you scroll to the right on the interface, you will see all the subpatches that are important for the program. As a beginner, you should not change them and may safely ignore them. Advanced Pd users: feel free to check them out for inspiration!


Communication and data transmission

The most important prerequisite for the communication between the program and the 3D printer is the serial connection, the so-called COM port. After the printer is connected to the computer via a USB cable, you must enter the corresponding port number once. To do this, click on the **SETUP** button, which opens the **Calibration Panel**. The yellow main display now shows us that we are in calibration mode.

In this window you will find on the top row a two-part module, **COM-Port** and **PrinterComm**. Here you must first enter the serial port (COM port) used by your printer. To be sure, click on **Show available ports** to see the correct port number in the Pd console.

Since most 3D printers work in a similar way, you can safely ignore the port speed (**Serial_Port_Speed**) as well as the **PrinterComm** module for now and only change or fine-tune it if it is not working properly. After entering the port number save this setting with **SET!**.

Now you can click on the **INITIALIZE** button back in the main window to establish the connection with the printer.

 *If you are unsure at this point, follow the links to **G-Code** below and refer to your 3D printer's documentation for guidance on your specific model.*

After you initialize the printer, the display will show “*Waiting for Start Position*”. This is where we wait for both the printhead and the print bed to move to the playing position:

- 1) The printhead moves to the zero position of the printer, called **HOME** in our patch.
- 2) The print bed moves to **HOME** accordingly.
- 3) The printhead moves to the preset playing position, which is the position where the guitar pick is closest to the first string. In our patch this position is called **START**. Later, the program will adopt the values you set for **START** during calibration.
- 4) The print bed moves to **START** accordingly.

The whole process takes about one minute. As soon as this has been done successfully, the display will show “Online”. The printer is ready!

The G-Code language

Most 3D printers run using the special **G-Code** standard. To do this, Pd must communicate with the printer in **ascii** coding language using a **Pd-abstraction** that correctly converts our commands to move the printhead linearly.

For example: the **G-Code message** “G1 X30 F5000” commands the printer to go to “position 30” on the X-axis at a speed of 5000 millimeters per second.

Follow the link to learn [more about G-Code](#).

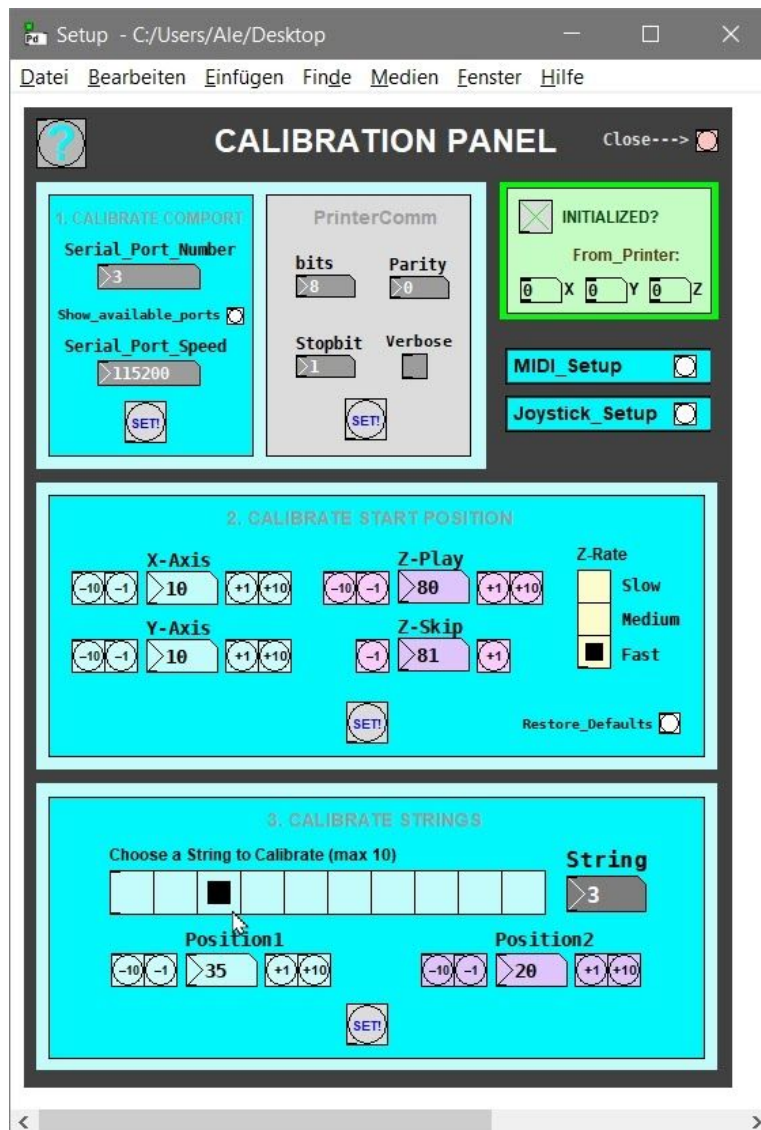
🔧 Each machine uses ready-made subroutines (cycles) that are adjusted with parameters or variables. It is recommended to check the documentation of your 3D printer model to avoid errors or problems. You can read a more detailed explanation on setting up 3D printers [here](#).

Calibration of the printer depending on the instrument

🔧 Before you start with the digital part of the calibration, make sure that all parts of the printer are movable and that the print bed is at the bottom so that you can mount the instrument.

Warning: It is essential to perform this setup before the first use or when changing the instrument.

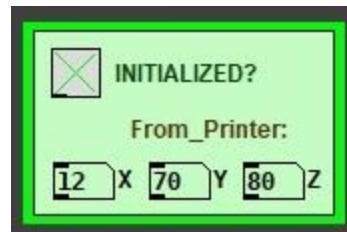
If the printer indicates that it is online, we can open the Calibration Panel by clicking on **SETUP** in the main window of the corresponding instrument patch.



In the **Calibration Panel**, we must look carefully and enter the correct values for all variables depending on the instrument or model of the 3D printer:

0. Check the data transmission

First of all, we can check on the right side on the green **Initialized?** area to see if everything is working properly. It shows us the status of the connection with the printer as well as the incoming axis values, i.e. the values that are transmitted from the device through the COM port into the Pd patch.



1. Calibrate the COM-Port

In the first module on the left, you can enable or configure the COM port, as described above under **Communication and Data Transmission**. This step is not only necessary for the calibration that follows, but also for the correct functioning of the instrument patches.

2. Calibrate the Start Position

To set the playing position we use the module on the second row. First, you set the desired values by pressing the buttons in 1- or 10-millimeter increments: the printer will respond immediately. If that does not suit your instrument, keep fine-tuning the values.

The purple values for the Z variable indicate the height of the print bed. You can choose from three different speed levels (**Z-Rate**). **Z-Play**, for example, tells us the height at which the strings are plucked. And **Z-Skip** determines the minimum height necessary to skip strings and thus efficiently drive the guitar pick through the playing surface. This means you should pay attention to both values and their correlation. Then click **SET!** to set your values for the start position and save them.

Now you can check at the main window if everything works correctly by clicking on **START**. Both the printhead and the print bed should now move to the desired start position and display your own values for X, Y and Z respectively on the green **Initialized?** module.

When you open the patch afterwards, this motion should always happen automatically at steps 3) and 4) of the printer initialization.

Default guitar pick positions or axis values:

Home:

X = 0 = Left

Y = 0 = Bottom

Z = 100 = Bottom

Maximum range:

X = 150 = Right

Y = 130 = Top

Z = 0 = Top

3. Calibrate the strings

Each string can be plucked by the guitar pick from the “front” or the “back” as desired. In order to play our instrument correctly, we now need to adjust the exact heights of the two playing positions (forward and backward) for each string individually. It is tedious, but it is worth the effort. For this purpose, select a string for manual adjustment in millimeter steps, just like in the previous step. Click **SET!** at the end to save the last values.

Bonus:

Advanced users can look into the **Midi_Setup** or **Joystick_Setup** subpatches to configure other input devices and try out advanced playing modes.

Let it rip! Overview of the playing modes

Once you have calibrated your 3D printer, you need to select the playing mode and now your 3D printer is finally ready to make music!

Zither:

- **Standard:** plays single strings without skipping other strings. Can play up to 10 strings either with a keyboard (keys 1 to 0, where the “1” key corresponds to the first string and the “0” key to the tenth string) or with the MIDI device. In the chapter “Construction of the zither”, you will find the pitch of the strings.
- **Jump:** plays single strings while skipping other strings. Works with a keyboard or with the MIDI device.
- **Joystick:** moves the guitar pick with the help of the joystick without skipping strings. Recommended for advanced players.
- **Custom:** offers the possibility to code your own subpatch. Recommended for advanced users.
- **Demo:** plays “Alle meine Entchen” automatically.

Ukulele:

- **Standard:** plays single strings without skipping other strings.

- **Rhythm:** automatically plays a pre-programmed rhythm. Here you can change the tempo of the rhythm.
- **Joystick:** moves the guitar pick using the joystick without skipping strings. Recommended for advanced players.
- **Custom:** offers the possibility to code your own subpatch. Recommended for advanced users. Here you could program your own rhythm.

Güiro:

- **Standard:** moves the stick using the keyboard or MIDI device.
- **Rhythm:** automatically plays a pre-programmed rhythm. Here you can change the tempo of the rhythm.
- **Joystick:** Moves the stick with the help of the joystick. Recommended for advanced players.
- **Custom:** offers the possibility to code your own subpatch. Recommended for advanced users. Here you could program your own rhythm.

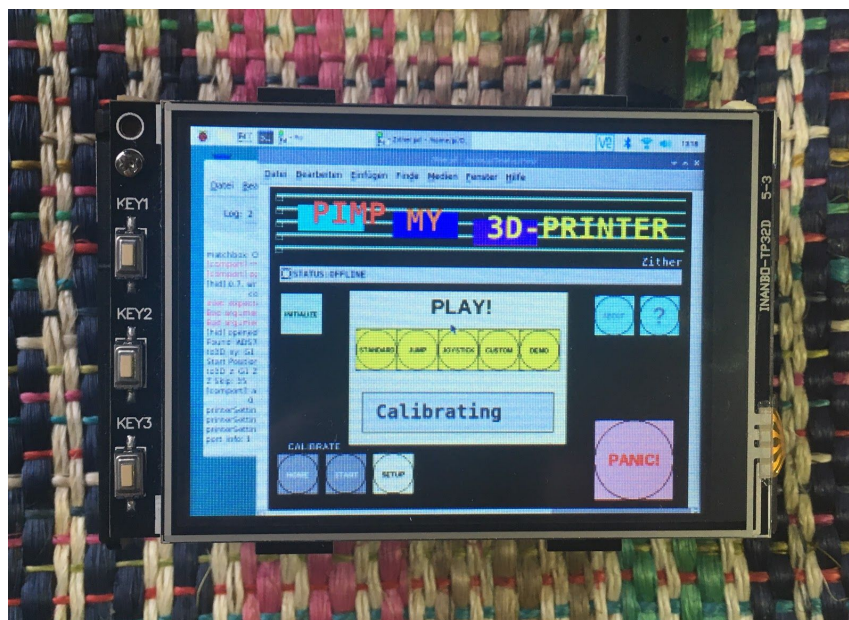
 [Video - 3D Printer in action](#) 

V. RPi

Lastly, we made a patch for the RaspberryPi for advanced users. It allows you to control the 3D printer without a laptop or a computer. You could even install a touch display on the RaspberryPi and mount it on the printer.

We tested the patch with a RaspberryPi 4 (RaspberryPi OS). The possibilities offered by the RaspberryPi 4 are great and could enable you to get even more out of your 3D printer.

If you want to know more about the RaspberryPi, you can learn more [here](#).



Pimp my 3D Printer

Dario Quiñones
Fabian Emde
Alejandro Weyler
Eduardo Delgado
Juan Diego Otero



<https://www.houu.de/>

