

3D-Druck-Tutorial, Teil 4:
Druckaufbereitung und -steuerung

Unter Kontrolle



Klaus Knopper

Sind die Objekte konstruiert und das Filament gewählt, folgt die Umsetzung in Steuerungsbefehle für den 3D-Drucker. Hierbei gilt es, einige Klippen zu umschiffen.



- Zur Vorbereitung des Druckes müssen die 3D-Modelle im STL-Format mit einer Slicer-Software in druckerspezifischen G-Code gewandelt werden.
- Da vom Hersteller mitgelieferte Slicer den Benutzer teils stark einschränken, sind freie Pendanten wie Slic3r oft die bessere Alternative.
- Als Preis der Flexibilität müssen Anwender selbst darauf achten, keine Einstellungen zu verwenden, die die Druckerhardware beschädigen können.
- Mit einem Raspberry Pi und OctoPrint lässt sich eine Fernüberwachung des Druckvorgangs realisieren.

Nachdem sich Teil 3 des Tutorials mit der „Physik“ des Druckmaterials beschäftigt und einige Tricks zur Materialhaftung und -bearbeitung gezeigt hat, befasst sich dieser letzte Teil mit dem eigentlichen Druckvorgang. Dazu sind die zuvor konstruierten STL-Dateien in Steuerbefehle für den 3D-Drucker umzusetzen. Anschließend geht es um das Überwachen und gegebenenfalls Korrigieren der manchmal sehr viel Zeit in Anspruch nehmenden Druckvorgänge bei FDM-Druckern (Schmelzschicht – Fused Deposition Modeling).

Hat man sein Druckmaterial gefunden, generiert die Slicer-Software aus dem 3D-Volumenobjekt den Algorithmus zum Ansteuern des Druckers. Dabei erzeugt sie die Befehle, die die Motoren und die Heizung regeln. In den meisten Fällen, vor allem bei FDM-Druckern, entsteht hierbei eine Datei mit den Befehlszeilen in standardisiertem G-Code.

Die beliebten Open-Source-Slicer CuraEngine und Slic3r (wirklich mit „3“ geschrieben) sind viel flexibler als ihre proprietären Verwandten. Letztere werden bei bestimmten Druckern als einzige Möglichkeit mitgeliefert und beschränken Anwender oft hinsichtlich der Anpassung des Druckers und der nutzbaren Materialien. Die freien Vertreter „bevormunden“ die Benutzer weniger. Hier lassen sich auch Initialsequenzen in den G-Code einbauen, die typische Optimierungen und Qualitätsparameter beinhalten. Slic3r bietet im Expert-Modus viel mehr Einstellungsmöglichkeiten als Cura. Im Gegenzug ist derzeit Cura beim automatischen Erstellen von leicht entfernbarem Stützmaterial Slic3r noch geringfügig überlegen – Slic3r erfordert hier mehr manuellen Aufwand. Der Artikel wird sich im Weiteren aus Platzgründen auf Slic3r beschränken.

Der Beispielcode in Listing 1 heizt Druckbett und Druckkopf zunächst auf die Solltemperatur auf und extrudiert dann altes Filament in einer dicken Bahn auf das Druckbett, um beim Start des eigentlichen Druckobjekts mit „frischem“ Filament zu arbeiten.

Allerdings kann die Option der direkten Eingabe druckerspezifischen G-Codes und die kaum eingeschränkte, freie Eingabe aller Parameter bei den Open-Source-Slicern auch zu teils offensichtlichen Fehlern führen. Gibt man etwa – versehentlich oder absichtlich – unpassende Parameter an, kann dies bis hin zur Beschädigung von Druckkopf oder Druckbett, zum Überschreiten der Nullposition in negativer Richtung oder zum Überschreiten des „offiziell erlaubten“ Druckvolumens führen.

Listing 1: G-Code-Beispiel

```
M104 S[first_layer_temperature] ; set extruder temp, don't wait
M140 S[first_layer_bed_temperature] ; set bed temp, don't wait
M109 S[first_layer_temperature] ; wait for extruder temp to be reached
M190 S[first_layer_bed_temperature] ; wait for bed temperature to be reached
G28 ; home all axes
[...]
G1 X230 Y25 Z0.35 F5000 ; start position, 0.35mm up from bed
G1 X20 E25 F1000 ; move on X axis while quickly extruding 25mm Filament
```

Daher ist der erste Schritt beim Konfigurieren des Slicers, die Hardware des Druckers sowie die von der Firmware unterstützten Befehle korrekt einzugeben. Dies erfolgt bei Slic3r über einen Wizard, der nur die wichtigsten Hardwaredaten abfragt, Abbildung 1 fasst die entsprechenden Dialogboxen zusammen.

Je nach verwendetem Filament sind Temperaturen, Geschwindigkeit und Retract einzustellen. Die meisten Slicer bieten hierfür optional an, die Einstellungen als Profil zu speichern, sodass Anwender beim Wechsel von Filament oder Düse leicht diese wieder aktivieren können. Abbildung 2 verdeutlicht die Problematik, die richtige Einstellung zu finden. Nicht immer liefert die feinste Auflösung auch das gewünschte Druckergebnis.

Zum für die Stabilität absolut notwendigen Überlappen der Druckbahnen und Verbinden der Schichten ist mehr Material zu extrudieren als bei einer perfekt runden Bahn. Dadurch wird die Bahndicke größer als der Düsendurchmesser und der Druckkopf presst die Bahnen stärker an die darunterliegende Schicht oder das Druckbett.

Extrudiert man hierbei jedoch zu viel Material, sammelt sich der Überschuss zwischen den Bahnen an. Der Effekt verstärkt sich mit jeder Schicht, bis der Druckkopf „Furchen“ ins Material zieht und schließlich das Objekt vom Druckbett reißt. Einige Druckerbesitzer werden das kennen, beispielsweise wenn sie den Filamentdurchmesser zu gering eingestellt haben (etwa 2,85 statt 3 mm bei der „dicken“ Variante der Filamente). Man sollte daher immer mit einer Schieblehre

nachmessen, ob der angenommene Filamentdurchmesser stimmt; auch die Düsenbreite muss bekannt sein und ist im Slicer korrekt einzustellen.

Die für das Überlappen notwendige sogenannte Überextrusion nebst dem beabsichtigten Flachpressen der Druckbahnen durch den Druckkopf lässt sich jedoch durch einen geringfügig größer berechneten Abstand zwischen den Bahnen kompensieren. Hier ist das Gleichgewicht entscheidend für den Verbund und die Präzision. Die meisten Slicer bieten zusätzlich eine manuelle Einstellung, die Bahnenbreite, Extrusion und Bahnenabstand zusammen ändert, sowie einen Multiplikator für den Filamentdurchmesser zum leichteren Eingeben von Korrekturwerten.

Unterextrusion (zu wenig Material wird extrudiert) ist zwar weit weniger auffällig im Ergebnis, führt in Extremfällen aber zu instabilen Ausdrucken und zu Brüchigkeit (siehe Abbildung 3). Überextrusion führt hingegen zur Ansammlung überschüssigen Filaments in Hohlräumen und im Außenbereich des Druckobjekts.

Als Faustregel sollte man den Bahndurchmesser immer mindestens 10 Prozent größer als den Düsendurchmesser wählen. Eine Bahn darf sogar bis zum doppelten Düsendurchmesser breit sein, allerdings muss man die Extrusionsgeschwindigkeit dann etwas reduzieren,

Tutorialinhalt

- Teil 1: 3D-Scan und Umwandlung in Volumenmodelle
- Teil 2: 3D-Konstruktion versus „Zeichnen“
- Teil 3: Materialeigenschaften und -auswahl
- Teil 4: 3D-Druck-Steuerung mit Slic3r und OctoPrint**

um den Druck auf das Filament im Druckkopf zu begrenzen. Sonst besteht die Gefahr, dass der Extrudermotor zu schwach wird oder „durchdreht“. Abbildung 4 zeigt die Einstellungen für eine 0,4-mm-Druckdüse.

Für gute Haftung auf dem Druckbett und zum Ausgleichen von Unebenheiten sollte man die erste Schicht etwas dicker drucken. Auch hier gibt es Fehlermöglichkeiten, die mitunter falsch interpretiert werden. Ist der Abstand zum Druckbett zu groß, haftet die erste Schicht nicht richtig. Ist er zu klein, wird die erste Schicht uneben, der Druckkopf zieht „Furchen“ auf der Oberfläche und reißt unter Umständen die erste Schicht durch die auftretenden Zugkräfte in horizontaler Richtung wieder von Druckbett. Haftet die für den Druckerfolg kritische erste Schicht nicht, bedeutet dies also nicht zwangsläufig, dass

Firmware

Choose the type of firmware used by your printer, then click Next.

G-code flavor: RepRap (Marlin/Sprint) ▾

Filament

Enter the diameter of your filament, then click Next.

Good precision is required, so use a caliper and do multiple measurements along the filament, then compute the average.

Diameter: 1.75 mm

Bed

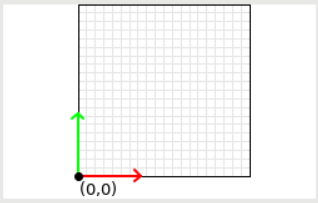
Set the shape of your printer's bed, then click Next.

Shape: Rectangular ▾

Settings

Size: x: 200 y: 200

Origin: x: 0 y: 0



Extrusion

Enter the temperature needed for extruding your filament, then click Next.

A rule of thumb is 160 to 230 °C for PLA, and 215 to 250 °C for ABS.

Temperature: 220 - +

Nozzle

Enter the diameter of your printer's hot end nozzle, then click Next.

Nozzle diameter: 0.4 mm

Bed

Enter the bed temperature needed for getting your filament to stick to your heated bed, then click Next.

A rule of thumb is 60 °C for PLA and 110 °C for ABS. Leave zero if you have no heated bed.

Bed temperature: 40 - +

Ein Wizard fragt in Slic3r diverse Basisdaten ab (Abb. 1).

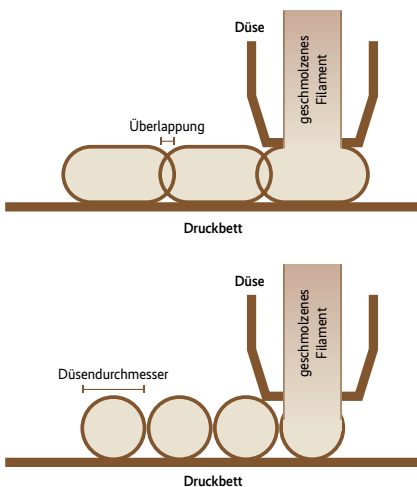
der Abstand zwischen Düse und Untergrund zu groß oder der Druck beim Auftragen der ersten Schicht zu klein ist. Im Gegenteil, der Abstand und damit der Materialfluss kann auch zu klein sein. Ist das Druckbett so dicht an der Düse, dass der Drucker kein Filament mehr nachschieben kann, führt das bis hin zum völligen

Blockieren des Filamentflusses. Folglich trägt das Gerät auch nicht genug Material auf, um den nachfolgenden Ebenen eine vernünftige Haftung zu bieten.

Eine weitere Faustregel besagt, dass man den „Null-Abstand“ zwischen Druckkopf und Druckbett so einstellt, dass zwei Blatt normales Druckerpapier dazwischen

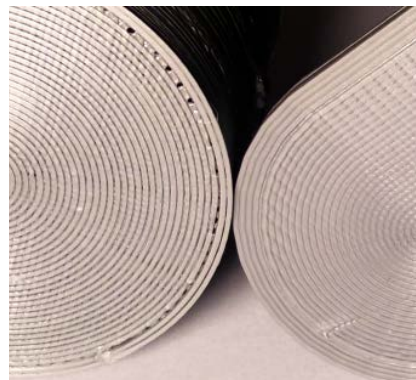
passen – so, dass sie sich gerade nicht verklemmen und sich an jeder potenziellen x-y-Position auf dem Druckbett einigermaßen leicht bewegen lassen. Aufbaubedingte größere Abstände durch nicht gut justierbare Endschalter in der z-Achse kann man im Slicer durch eine entsprechende Einstellung jedoch ausgleichen (Abbildung 5).

Bei Kopfbewegungen ohne Erzeugen von Druckbahnen (Springen/Jump zwischen Positionen) können jedoch materialspezifisch feine Fäden entstehen oder es bilden sich durch nachlaufendes Material Tropfen am Druckkopf. Dies lässt sich am einfachsten durch das Einstellen eines Retract beheben – ein Zurückziehen des Filaments in den Druckpausen. Aber Vorsicht: Nicht alle Materialien und Druckköpfe vertragen dies. Bei TPU verklemmt sich das Filament gerne im Druckkopf, und auch bei PLA können sich beim Zurückziehen sogenannte Residuen bilden, die das Filament beim erneuten Herausschieben „verklemmen“. Daher sollte man den Retract nur so groß wie notwendig wählen, um das „Nachtropfen“ zu verhindern. Bei den meisten Materialien sind 0,8 bis 1,4 mm sinnvoll. Bei TPU sollte man, trotz seiner Tendenz zum Fadenziehen, ganz auf Retract verzichten.



Mit der feinsten Druckauflösung zu arbeiten, funktioniert offensichtlich nicht, da die Druckbahnen sich nur punktuell berühren und nicht haften. Presst man mehr Material durch die Düse, überlappen sich die Bahnen, was zu besserer Haftung und zu einer glatteren Oberfläche führt (Abb. 2).

Oberflächen bei unterschiedlichen Einstellungen für die Bahnbreite: Zu dünne Bahnen verursachen, wie links zu sehen, Löcher im Ausdruck (Abb. 3).



PET(G) als Druckmaterial, eine wie im dritten Tutorialteil beschrieben gute und temperaturstabilere Alternative zu PLA, erfordert zum Eliminieren des Fädenziehens ein Retract bis gut 4 mm. Das ist je nach verwendeter Zuführung zur Düse und je nach Drucker nicht immer einfach.

Als Ersatz oder Ergänzung für Retract kann man die Geschwindigkeit erhöhen, mit der der Drucker zwischen den verschiedenen Druckpositionen beim Ab- und Ansetzen „springt“. Dadurch werden sich bildende Fäden in den Druckpausen „abgerissen“ und verbinden sich so nicht zu dauerhaften Clustern.

Grenzen der Mechanik

Schwächere Stepper-Motoren können sich an den „Umkehrpunkten“ beziehungsweise bei starkem Beschleunigen und Bremsen des Druckkopfes „versteppen“. Da das Gerät die Position des Druckkopfes nach der initialen Erfassung durch die Endabschalter nie nachmisst, verlässt sich die Firmware darauf, dass jeder Schritt der Schrittmotoren auch funktioniert. Sollte dies einmal nicht der Fall sein, entstehen dauerhafte Versätze im Druckobjekt, wie Abbildung 6 zeigt.

Hier gibt es einen Trick in allen dem Autor bekannten G-Code-Firmware-Versionen für Slicer-Programme, die einen eigenen Initialcode zulassen: Die Einstellung für die maximale Beschleunigung (nicht Geschwindigkeit) des Druckkopfes. Diese bewirkt, dass der Druckkopf vor

dem Ändern der Richtung sanft abbrems und im Verlauf einer längeren geraden zurückzulegenden Strecke wieder beschleunigt.

In Slic3r lässt sich diese Einstellung, die auch akustisch einen interessanten Effekt ähnlich wie die Bewegungen eines Industrieroboters hervorruft, in der G-Code-Präambel vor Start des eigentlichen Ausdrucks angeben (siehe RepRap-Wiki [4]):

```
; Acc printing
M201 X250 Y250 Z15
; Acc travel
M202 X250 Y250 Z15
```

Die Einstellungen M201 für Beschleunigung während des Druckens und M202 für Beschleunigung während des Sprungs zu einer anderen Stelle sind hier mit max. 250 mm/s² in der Horizontalen, und 15 mm/s² in der vertikalen Richtung angegeben. Dies bewirkt recht sanfte Abbrems- und Beschleunigungsvorgänge, die sich nicht direkt auf die Druckqualität auswirken. Sie erlauben aber das Setzen einer höheren Geschwindigkeit beim Positionieren des Druckkopfes über längere Strecken, da die beim Beschleunigen und Bremsen sowie Umkehren auftretenden Kräfte deutlich geringer ausfallen.

Objekte in Teilen ausdrucken

Neben dem Generieren von G-Code aus einem Volumenobjekt besitzt Slic3r auch ein Plug-in, mit dem man Objekte zerschneiden und in Teilen drucken kann, die sich sonst schwierig flach auf dem Druckbett positionieren lassen oder zu groß für den Drucker sind. Die Abbildungen 8 bis 10 verdeutlichen dies anhand der zum Tropfen verzerrten Kugel aus dem zweiten Teil des Tutorials [2].

Bei ABS und HIPS kann man mit diesem Verfahren geschnittene und einzeln gedruckte Teile durch kurzes Tunken der Schnittkanten in Aceton und anschließendes Zusammenpressen „autogenverschweißen“. Bei PLA ist hierfür starker Kleber (Pattex transparent, Sekundenkleber oder Ähnliches) erforderlich.

Natürlich kann Slic3r, so wie alle anderen Slicer, Objekte in alle Raumrichtungen skalieren. So lassen sich, sofern Maßhaltigkeit keine zwingende Prämisse ist, geringfügig zu große Objekte auch kleiner skalieren und so „am Stück“ drucken.

Nebenbei erlaubt das geschickte Rotieren und Schneiden von Objekten in vielen Fällen auch, dass man wie in Abbildung 11

| Size and coordinates | |
|----------------------|---------------------------------------|
| Bed shape: | <input type="button" value="Set..."/> |
| Z offset: | <input type="text" value="-0.1"/> mm |

Justieren der Nullposition in Slic3r: Der Druckkopf wird nach Ansprechen des Z-Schalters um den eingetragenen Wert vom Druckbett entfernt – positiver Wert – oder selbigem angenähert – negativer Wert (Abb. 5).



Hier haben einige Steps nicht geklappt: Das Druckobjekt hat Versätze, die sich sogar wiederholen (Abb. 6).

| Speed for print moves | |
|---------------------------|---|
| Perimeters: | <input type="text" value="20"/> mm/s |
| ↳ small: | <input type="text" value="10"/> mm/s or % |
| ↳ external: | <input type="text" value="15"/> mm/s or % |
| Infill: | <input type="text" value="25"/> mm/s |
| ↳ solid: | <input type="text" value="15"/> mm/s or % |
| ↳ top solid: | <input type="text" value="10"/> mm/s or % |
| ↳ gaps: | <input type="text" value="10"/> mm/s or % |
| Bridges: | <input type="text" value="5"/> mm/s |
| Support material: | <input type="text" value="20"/> mm/s |
| ↳ interface: | <input type="text" value="15"/> mm/s or % |
| Speed for non-print moves | |
| Travel: | <input type="text" value="120"/> mm/s |
| Modifiers | |
| First layer speed: | <input type="text" value="50"/> mm/s or % |

Einstellungen für die Positionierungsgeschwindigkeit in Slic3r (Abb. 7)

durch entsprechende Drehungen nach oben die Überhänge reduziert.

Komische Ideen und Hausmittelchen

Im Laufe der Zeit sammeln sich bei den Benutzern von 3D-Druckern mehr oder weniger exotische Tipps an. Einer davon, der dem Autor besonders aufgefallen ist: Rapsöl zum Einfetten von PLA oder ABS-Filament vor der Verarbeitung.

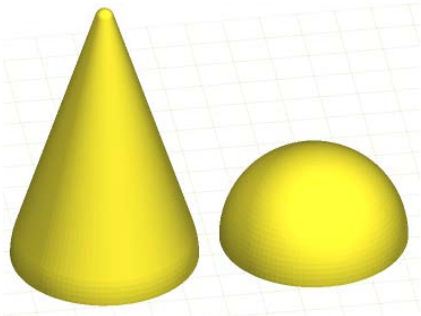
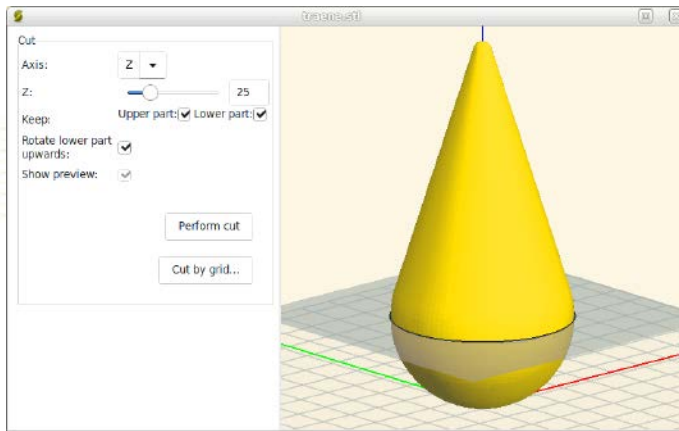
| Extrusion width | |
|----------------------------|--|
| Default extrusion width: | <input type="text" value="0.6"/> mm or % |
| First layer: | <input type="text" value="0.7"/> mm or % |
| Perimeters: | <input type="text" value="0.6"/> mm or % |
| ↳ external: | <input type="text" value="0.55"/> mm or % |
| Infill: | <input type="text" value="0.6"/> mm or % |
| ↳ solid: | <input type="text" value="0.6"/> mm or % |
| ↳ top solid: | <input type="text" value="0.6"/> mm or % |
| Support interface: | <input type="text" value="default"/> mm or % |
| Support material: | <input type="text" value="0.5"/> mm or % |
| Overlap | |
| Infill/perimeters overlap: | <input type="text" value="20%"/> mm or % |

Beispiel 0,4-mm-Düse: Die stabilen Perimeter sind relativ dick gewählt, was die Stabilität erhöht und weniger Bahnen ermöglicht und damit die Fertigstellung beschleunigt, während die Außenperimeter relativ fein eingestellt sind und feine Strukturen sowie dünne Wände erlauben (Abb. 4).



Manche Objekte lassen sich ohne zusätzlichen Aufwand nur schlecht stabil auf dem Druckbett positionieren (Abb. 8).

In Slic3r lassen sich solche Objekte zerschneiden ... (Abb. 9)



... und in Teilen ausdrucken (Abb. 10).



Dreht man den aus einem Scanstammenden Kopf nach hinten, bleiben die Überhänge in einem ohne Stützstrukturen druckbaren Rahmen (Abb. 11).

Hierbei wird vor oder hinter der Extruder-Mechanik, so nah wie möglich an der Düse, das Filament durch einen kleinen Schwamm gesteckt, der regelmäßig mit Rapsöl getränkt wird, was der Verbesserung der Druckqualität dienen soll. Rapsöl besitzt einen relativ hohen Siedepunkt, verdrängt Wasser, vermindert stark die Reibung zwischen Filament und der Düsenzuführung und fördert die gleichmäßige Hitzeverteilung. Daher erscheint die Verwendung physikalisch gesehen durchaus sinnvoll und in Experimenten gelungen Ausdrücke sowohl mit ABS als auch mit PLA sichtbar besser, wenn das Filament mit Rapsöl benetzt war.

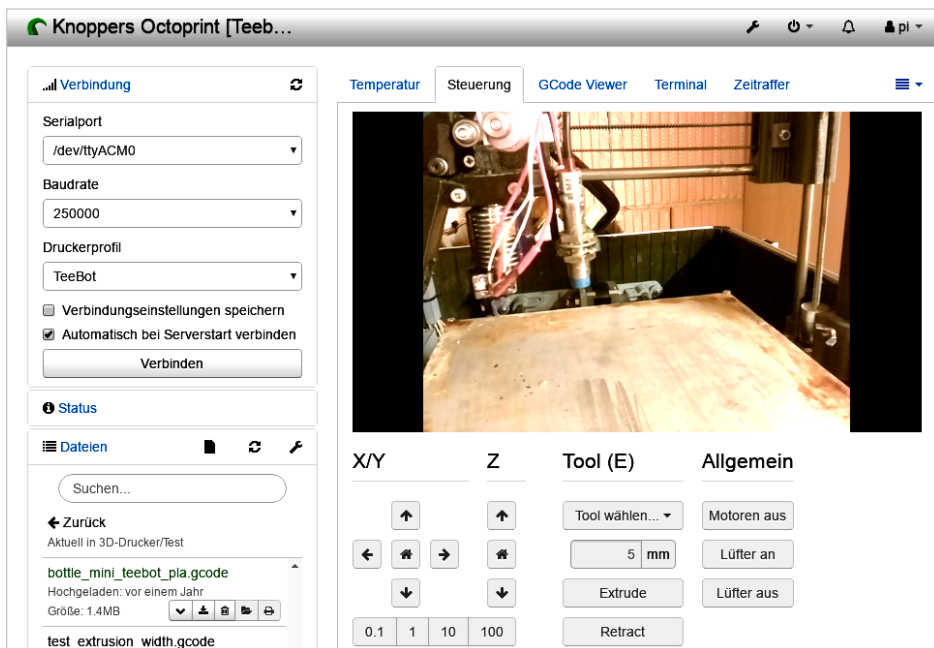
Ein weiterer nützlicher Nebeneffekt: Das Öl löst Staub oder andere Verunreinigungen des Filaments ab, sie verbleiben größtenteils im Schwamm. Was dem Autor allerdings Sorgen bereitet, ist die Frage, was mit dem Rapsöl im fertigen Ausdruck passiert. Wegen seiner hohen Temperaturbeständigkeit verdampft es ja nicht, sondern wird zusammen mit dem Filament „verdrukt“. Offenbar vermischt oder verbindet es sich mit dem aufgeschmolzenen Thermoplast. Ob dies einen Einfluss auf die Haltbarkeit der Ausdrücke hat, lässt sich noch nicht abschließend beurteilen.

OctoPrint als Kontrollzentrum für den 3D-Drucker

Ein absolutes Muss für Besitzer eines G-Code-verarbeitenden Druckers ist OctoPrint. Das von der Entwicklerin Gina Häußge in Python geschriebene WebGUI mit Druckersteuerungssoftware erleichtert die Kontrolle und Überwachung des Druckvorgangs ungemein.

Man installiert die Software auf einem handelsüblichen Raspberry Pi, der mit einer PiCam oder USB-Kamera ausgestattet sein sollte. Das Bundle befestigt man so am Drucker, dass es die Mechanik nicht stört und die Kamera einen freien Blick auf den Bereich hat, in dem sich der Druckkopf während des Ausdrucks bewegt.

Die Netzanbindung per LAN oder WLAN macht den 3D-Drucker netzwerkfähig und per Smartphone mittels Browser oder spezieller OctoPrint-Apps überwach- und bedienbar. OctoPrint erlaubt zudem das Speichern von Druckaufträgen und kann per Plug-in den für den Drucker eingesetzten Slicer auf dem Raspberry Pi einrichten. So lassen sich Druckaufträge auch mit entsprechend ausgewähltem Slicer-Profil direkt als STL-Datei an den Drucker schicken. Ein G-Code-Preview erlaubt das Begutachten der zu druckenden Schritte mit Fortschrittsanzeige und



Mit etwas Hardware und der freien Software OctoPrint können Anwender ihren 3D-Druckvorgang aus der Ferne überwachen (Abb. 12).

ein Statistik-Plug-in protokolliert verbrauchtes Druckmaterial, Laufzeit, Stromverbrauch, Erfolgsstatistik, Druckkosten und andere einstellbare Parameter.

Für die Installation von OctoPrint existieren für den Raspberry Pi fertige Images zum Flashen auf SD-Karte. Experten auf der Linux-Kommandozeile haben mit

```
unzip -p octopi-stretch-lite-0.15.1.zip \
2018-04-18-octopi-stretch-lite-0.15.1.img \
|dd of=/dev/mmcblk0 bs= 1M oflag=sync,direct
```

bereits alles erledigt. Windows-Anwender können auf die Software Etcher (etcher.io) zurückgreifen, um den Inhalt des Zip-Archivs auf eine SD-Karte zu bringen.

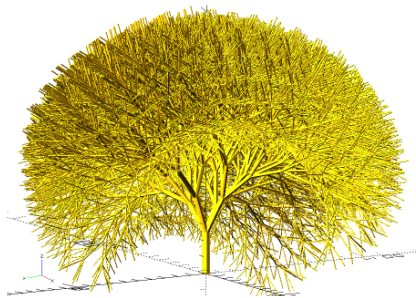
Die Initialkonfiguration des Netzwerks kann über eine Textdatei auf der ersten FAT32-Partition der SD-Karte erfolgen, der ganze Rest über das komfortable Web-GUI. Der Funktionsumfang von OctoPrint ist gigantisch, Erweiterungen sind übers Web download- und installierbar. Kosten: keine, es ist ein echtes Open-Source-Projekt. Wer die OctoPrint-Entwickler unterstützen möchte, kann dies über octoprint.org tun. Abbildung 12 zeigt, wie man mit der Software einen typischen Druckvorgang bequem vom Schreibtisch aus im Auge behalten kann.

Fazit

An der Länge des Tutorials kann man sehen, wie schnell Interessierte vor lauter Begeisterung und Experimentierfreudigkeit viel – sicher gut investierte – Zeit damit verbringen können, aus den Materialeigenschaften und der physikalischen Leistungsfähigkeit des angeschafften Druckers das Maximum herauszuholen. Das Stöbern in Foren und Tipps von 3D-Designern zum besten Ausdruck ihres Konstrukts lässt fast jeden regelmäßigen Anwender des 3D-Drucks nach kurzer Zeit zum Experten für seinen jeweiligen Drucker werden – weil die Technik an sich einfach ist, aber der Meister sich erst im Detail beweist. Der

Listing 2: Per Rekursion erzeugter Baum

```
module baum(bh=25, tr=5, l=5, a=30, bc=4, $fn=8){
  if(l>0){
    cylinder(h=bh, d=tr);
    translate([0,0,bh]){
      sphere(d=tr);
      for(i=[0:(bc-1)])
        rotate([0,0,i*(360/bc)])
        rotate([a,0,0])
        baum(bh*0.95, tr*0.75,
            l-1, a, bc);
    }
  }
}
baum(l=8);
```



Die Vorteile einer echten Programmiersprache zeigen sich auch in bekannten Konstrukten der Programmierung wie Rekursion. Das Beispielmodul aus Listing 2 erzeugt mit einer Rekursionstiefe von acht Levels eine Baumstruktur (Abb. 13).

Autor hofft, dass die Tipps in diesem Tutorial helfen, 3D-Projekte erfolgreich umzusetzen und auch Dinge, die man früher für unmöglich hielt, mit dem eigenen 3D-Drucker und ein paar Tricks doch zu realisieren. Für Fragen zum Artikel oder zum 3D-Druck allgemein steht der Autor per E-Mail gern zur Verfügung. Die Kontaktaufnahme kann über die Redaktion erfolgen.

Zum Abschluss noch ein kleines OpenSCAD-Experiment, an dem man sieht, wie sich scheinbar natürlich gewachsene Strukturen per Rekursion in einer Programmiersprache abbilden lassen. Das Zusammensetzen des Baums mithilfe einer rein grafisch arbeitenden 3D-CAD-Software wäre hier sehr zeitaufwendig gewesen, OpenSCAD schafft dies mit ein paar Befehlen. Die Frage, ob und wie man dieses Objekt mit einem 3D-Drucker tatsächlich ausdrucken kann, muss der Autor in den nächsten Tagen noch im Experiment klären. (avr@ix.de)

Quellen

- [1] Klaus Knopper; 3D-Workflow; Abgetastet; 3D-Druck-Tutorial, Teil 1: 3D-Scan mit Open-Source-Software; *iX* 2/2019, S. 118
- [2] Klaus Knopper; 3D-Workflow; Aufgezeichnet; 3D-Druck-Tutorial, Teil 2: Konstruktion mit Blender, OpenSCAD und Co.; *iX* 3/2019, S. 142
- [3] Klaus Knopper; 3D-Druck-Workflow; 3D-Druck-Tutorial, Teil 3: Bessere Druckergebnisse mit dem richtigen Material; *iX* 4/2019, S. 132
- [4] Codebeispiele: ix.de/ix1905116

Klaus Knopper

lehrt Softwareengineering am Fachbereich Betriebswirtschaft der Hochschule Kaiserslautern und forscht unter anderem im Bereich Open-Source-Systeme sowie räumlich-akustischer Repräsentation von Softwaremodellen.