Received: 5 February 2021 | Revised: 23 March 2021 | Accepted: 24 March 2021

DOI: 10.1002/pol.20210086

RESEARCH ARTICLE

JOURNAL OF POLYMER SCIENCE WILEY

PolyChemPrint: A hardware and software framework for benchtop additive manufacturing of functional polymeric materials

Bijal B. Patel[1] | Yilong Chang[2] | Sang Kyu Park[1] | Siqing Wang[1] | John Rosheck[3] | Kush Patel[1] | Dylan Walsh[1] | Damien Guironnet[1] | Ying Diao[1]

[1]Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois

[2]Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois

[3]School of Chemical Sciences, University of Illinois at Urbana-Champaign, Urbana, Illinois

Correspondence
Ying Diao, Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Urbana, IL 61801.
Email: yingdiao@illinois.edu

Funding information
NASA Early Career Faculty Award, Grant/Award Number: 80NSSC21K0070; NSF CAREER, Grant/Award Number: NSF DMR 18–47828; NSF under DMREF, Grant/Award Number: DMR-1727605

Abstract

Additive manufacturing (AM, 3D Printing) of hierarchical polymer structures for a targeted function represents a grand challenge in the field of polymer science and engineering. Because advanced functional materials often do not possess suitable mechanical and rheological properties for conventional fused deposition modeling, a key challenge that researchers face is in integrating custom deposition tool heads that enable printing of non-filamentary materials while preserving synchrony with the motion axes. In this article, we demonstrate a highly versatile hardware and software platform for melt and solution-phase benchtop AM and highlight patterning and post-deposition processing of a series of non-filament forming functional polymers, including PS-b-PLA bottlebrush block copolymers, semiconducting polymer DPP2T-TT, conducting polymer PEDOT:PSS [poly(3,4-ethylenedioxythiophene)-poly(styrenesulfonate)], and an SiO_2-PE nanoporous polymer matrix composite. We present a free and open-source Python module, PolyChemPrint, which serves as a research-optimized AM control software. The cross-platform (Windows/Linux) software is designed to be extremely flexible in terms the hardware that can be connected and a detailed user manual and developer guide are provided for use by researchers without extensive computer programming experience. Finally, we provide extensive details of the hardware used for operation of low- and high-pressure pneumatic extruders and a laser module as rapidly interchangeable tool heads.

KEYWORDS

3D printing, additive manufacturing, control software, polymers

1 | INTRODUCTION

The additive manufacturing (AM) (3D Printing) paradigm has dramatically impacted manufacturing and rapid prototyping. Over the past 30 years, improvements in the reliability and ease-of-use of 3D printing hardware and software have led to the proliferation of fused deposition modeling (FDM) systems in laboratories, community centers, and for personal use. Now, even home users have ready access to the mature FDM pipeline: from computer aided design (CAD) of complex parts, through slicing and generation of executable G-code instructions, to printing a variety of plastic feedstocks on reliable hardware with micron-scale dimensional accuracy. In parallel, industrial systems using a variety of deposition techniques have enabled bottom-up processing of materials ranging from concrete and metal to living tissue, with commercial viability in the construction, aerospace, and biomedical industries.[1]

An ongoing research focus has been to expand the material palette of 3D printing to functional materials with desirable biological, electronic, or stimuli-responsive properties. In doing so, researchers must overcome a fundamental practical challenge: functional materials often do not have suitable thermal and rheological properties for filament drawing and extrusion via FDM. Even when material properties are suitable, it can be prohibitively costly to produce high-purity functional materials on the kilogram scale for feeding into commercial filament spooling equipment. While this incompatibility can sometimes be circumvented by dispersing active material into polymer composites,[2–5] a more versatile approach is to use non-FDM methods, with custom tool heads tailored to particular material system. The literature has many successful demonstrations of “microextrusion” or “direct-writ” techniques for electronic,[6–12] structural,[4,13–15] and biological materials,[16–20] generally delivering materials of varying viscosity through syringes driven by pneumatic or mechanical pressure sources.

Using custom tooling introduces a major challenge, however, that may preclude the use of the commercially mature CAD-to-print pipeline. To successfully print arbitrarily complex patterns, the motion axes and tool head must operate in synchrony, starting and stopping flow in time with motion commands. There have hereto been two commonly reported approaches to solving this problem. The first is to base the system on a consumer 3D printer and simply replace the FDM tool head with the custom tool head. This approach has key advantages in its low cost, and guaranteed compatibility of the motion axes with G-code, the language commonly generated by 3D slicing programs. Unfortunately, it is nontrivial to establish a communication link between the 3D printer's onboard microcontroller and a custom tool head. Unless the tool head can be built around the printer's original stepper motor,[21] tool commands must be executed either manually, or by a separate program, restricting the complexity of printed patterns and leading to defects caused by poorly synchronized motion and tooling. The second approach is to build the printer using a combination of industrial linear stages and custom tooling. In this case, proprietary software solutions that run on a dedicated control PC are sometimes available for purchase from hardware manufacturers or are developed by researchers in environments such as LABVIEW for a specific combination of axes and tooling. This approach requires significant monetary and time investment and creating home-grown solutions often requires a level of software engineering knowledge not readily accessible to chemists and materials scientists whose main interest is in studying material property tuning during 3D printing.

In this article, we present a free and open-source hardware and software framework for benchtop AM that enables integration of non-FDM tooling into the standardized 3D printing workflow. This article introduces the Python package PolyChemPrint which serves as control software to manage synchronized communication between hardware controllers for motion axes (such as that of a consumer 3D printer) and custom tooling (pneumatic extruders, lasers, etc.) along with providing hardware details for the implementation used in our lab. We also provide full user and developer manuals (SI 1,2, https://polychemprint3.readthedocs.io) with detailed instructions on software operation, modification, and the setup of new hardware. We hope that by doing so we can help to reduce a substantial barrier to entry for research in this field. Presently, countless hours being wasted by individual research groups in duplicating programming and hardware design efforts, and reproducing published results is made unnecessarily challenging by the use of closed-source software and hardware. Widespread accessibility of precise modulation of printing parameters has particular importance to the burgeoning study of 3D-printed functional materials, where deposition conditions can drastically modify self-assembly behavior and unlock novel material properties.[22,23]

2 | SYSTEM DESIGN PRINCIPLES AND OVERVIEW

Fundamentally, the AM (3D Printing) paradigm requires three components: (1) a method to input software instructions that specify a 2D/3D pattern, (2) hardware capable of precise motion and dispensing control, and (3) control software that ensures that the tool head(s) and motion stages act in synchrony. Additional peripherals may return data back to the control computer for monitoring or feedback. The interplay between these components at runtime is shown schematically in Figure 1. In this section, we present a high-level overview of our implementation of each of these aspects and the key design principles of PolyChemPrint, leaving specific details of software and hardware implementation for Sections 3 and 4, respectively.

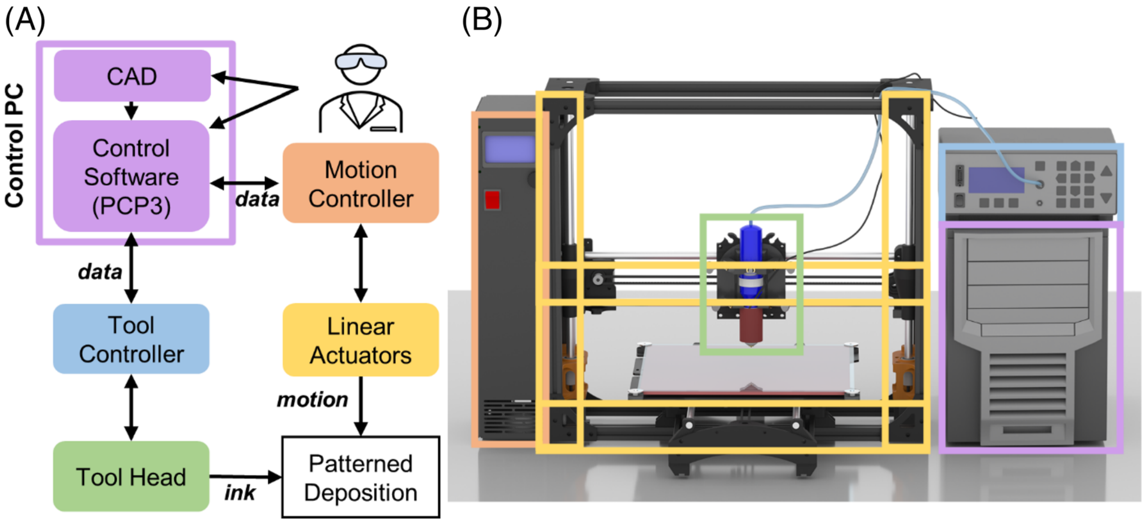

FIGURE 1 Outline (A) and rendering (B) of the 3D printing paradigm as implemented in PolyChemPrint. Solid lines in (B) are colored to correspond with boxes in (A). A photograph of the system is shown in Figure S2

The first step of 3D printing is to generate a design digitally and convert it into a series of commands to be executed sequentially by the hardware. In the PolyChemPrint framework, all print routines are defined as hardware agnostic. That is, regardless of the specific identity of the axes or tool used, all print routines draw from a single "library." To enable this level of generality, it was necessary to define a common set of commands that all hardware devices would execute. For motion commands, a subset of the G-code (RS-274[24]) syntax was chosen, as it is by far the most common protocol used for 3D printing and computer numerical control (CNC) machining. For toolheads (extruders, syringe pumps, LASERs, etc.) three basic functions are defined: turning the tool on or off and setting the tool Value. As we will describe, users use these basic functions to build up print sequences of arbitrary complexity. Before execution, these high-level commands are converted to their respective hardware-specific protocols and then sent sequentially from the control computer.

Users can define print routines in several ways depending on the pattern complexity. At the simplest level, users can directly control motion axes and tools from the terminal by entering commands line-by-line into the command line interface (CLI) of PolyChemPrint. For multi-line "sequences," users can draw from the built-in library, or add new sequences hardcoded as individual python files. Existing sequences include common test geometries such as cubes, meanderlines, and circles, with adjustable parameters (printing speed, length, width, etc.) that can be set from within the CLI at runtime. Alternatively, routines of arbitrary length can be imported from G-code files generated by the myriad free and fully featured software programs that convert 2D vector images (Inkscape) or 3D CAD files (Cura, Slic3r) into G-code. When the recommended slicing configurations are used (see Section 6.2 of the User Manual) these can be readily pre-processed within the CLI to change parameters such as axes feedrates (printing speed), and tool behavior, including converting from FDM extruder motor commands (e.g., "G0 E10") to tool on/off signals. In this way, PolyChemPrint allows users to leverage the consumer FDM workflow and unlock vastly higher complexity of motion paths and print sequences than can be achieved by manually specifying paths. Finally, the "recipe" function allows any number of sequences to be chained together, saved, loaded, and executed, for example, for performing a combinatorial set of prints to rapidly screen for printing parameters, or for performing specific pre-print and post-print actions.

The PolyChemPrint framework is designed to be extremely flexible in terms of hardware (axes/tools). The key requirement is bi-directional communication between the hardware devices and the control computer. The difference between the PolyChemPrint framework and stock FDM printers is that when using PolyChemPrint, the task of timing commands sent to the motion axes and tool head is shifted from the onboard 3D printer microcontroller to the control PC. This means that the control computer must receive confirmation from the axes when a motion step has completed to know when to send the next command. For many industrial linear stages, this functionality is either enabled by default or can be enabled in the controller settings. For 3D printers running open source Marlin firmware (most consumer 3D printers), instructions for altering the firmware to enable this function are described in Section 2.6 of the PolyChemPrint User manual (SI 1.0, https://polychemprint3.readthedocs.io/). In our lab, we have used PolyChemPrint to control the Aleph Objects Lulzbot Taz 6 consumer 3D printer, and a range of toolheads including an Ultimus V pneumatic dispenser (Figure 1(B)) and an Arduino-based Laser controller. After initial setup of connection parameters, hardware can be activated or deactivated within the software on-demand. In this way, the software can easily accommodate a library of tools (or axes) which are readily interchangeable.

The final component necessary for 3D printing is the control software, released as a free and open source Python package for Windows/Linux (project homepage [https://publish.illinois.edu/polychemprint3]). Two key design criteria for this package are that it should streamline bench-scale AM research and not require extensive computer science expertise to either operate or modify/extend. To this end, the software is written in a modular, object-oriented fashion, with templates provided for additions to the library of tools, sequences, axes, and even interface menus. The full user manual includes instructions for setup and operation, and a programming guide for modifying the software. Key research-friendly features include the previously mentioned parameterized sequences and recipe builder for combinatorial screening, automatic data logging during each print routine, and the ability to easily plan for and include pauses for tool head cleaning/ sample switching within recipes, along with robust error-handling.

3 | POLYCHEMPRINT PYTHON SOFTWARE PACKAGE

PolyChemPrint is released under the University of Illinois NCSA Open Source License and fully documented source code is available on GitHub (https://github.com/BijalBPatel/PolyChemPrint3). Three branches are maintained: (1) the master branch featuring the latest stable release, (2) the beta branch that contains new updates and is the version being used in our lab, and (3) the dev (unstable) branch for implementing and testing new features. The master branch of the package is also available for installation from the python package index (PyPI) and it can be installed with all dependencies using the standard python package manager (pip).

The user interacts with PolyChemPrint through a CLI with a menu tree structure as depicted in Figure 2(A). Submenus branching from the main menu are separated

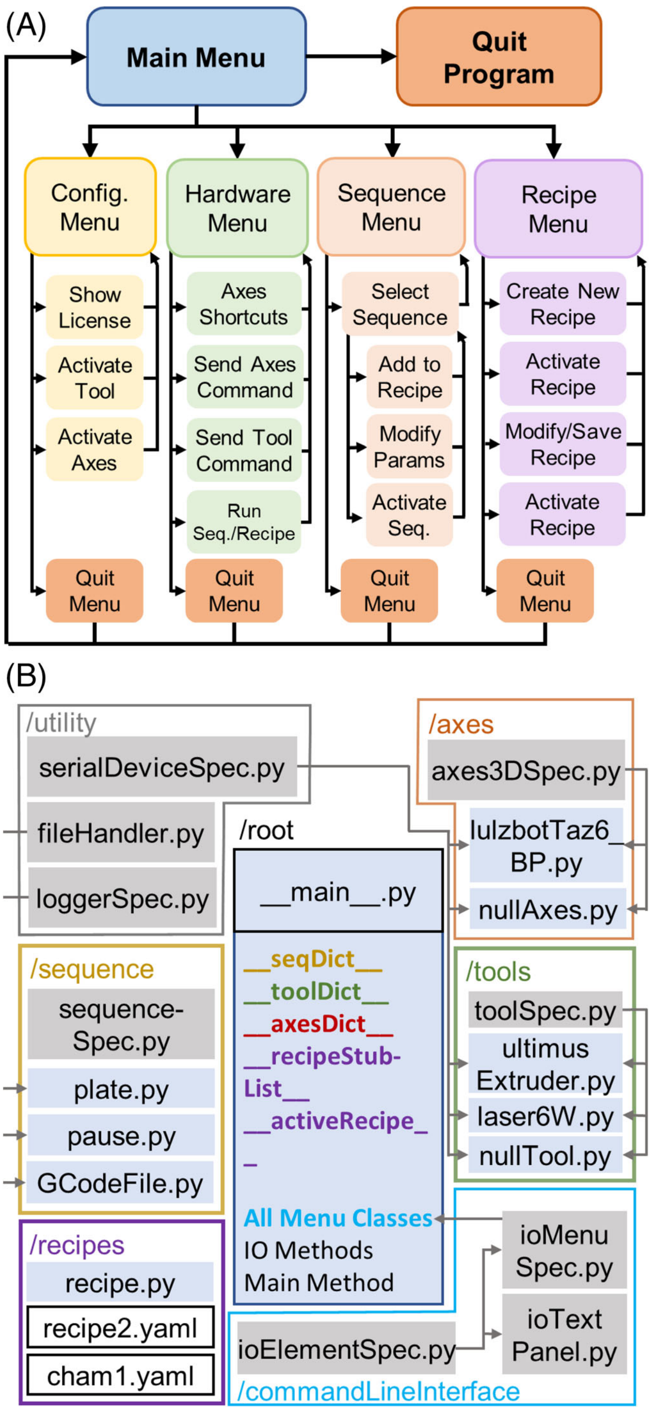

FIGURE 2 Flowcharts mapping (A) the menu tree structure and (B) the PolyChemPrint package structure. In (B), light blue boxes represent class declarations (containing methods and attributes) and gray boxes represent abstract base classes (containing methods only), with gray arrows showing inheritance relationships. Boxes with white fill indicate text files. Parameters and methods are omitted for clarity. The main method contains data structures holding the instantiated objects whose colored frame correspond to the text color

to correspond to the three main operating modes (manual, line-by-line commands and sequence/recipe; parameterized sequences; and multi-sequence recipes) and a configuration menu for viewing program details and switching the active tool or axes. A text-based CLI was chosen (as opposed to a graphical user interface) in the interest of streamlining both user operation of the software and easing the addition of new features by researchers who create forks of the main program for their specific equipment. The main drawback of CLIs (the steeper learning curve) is actively mitigated through the following approaches. First, all menus show detailed prompts and descriptions for menu options; no commands need to be memorized. Second, except for manual communication to the hardware, all user input is validated against the menu options to minimize damage caused by input errors. Finally, robust error handling is included at all stages, and the user can issue a "break" command at any time with the key combination "ctrl + c."

PolyChemPrint is written in Python following object-oriented programming (OOP) principles, meaning that data and methods are organized in terms of distinct "objects" with clearly defined attributes and behaviors, which are only instantiated (given specific values) in the "__main__.py" file. This modular approach (Figure 2(B)) greatly simplifies the process of adding additional hardware and sequences by isolating these blocks of code into distinct submodules (folders) that are separate from the more complex user interface and main methods. In order to add a new sequence to the program, for example, the user can simply clone one of the existing light blue .py files, rename it and modify the contents to contain the desired motion/tool commands, and place it in the sequence folder. On startup, PolyChemPrint automatically attempts to load any new sequences, axes, or tool . py files, after checking for compiler errors.

PolyChemPrint also takes advantage of the OOP concept of inheritance to streamline addition of new code files to the program. The gray boxes in Figure 2(B) are effectively blueprints ("Abstract Base Classes," in Python) for the class declarations (light blue .py files) that inherit from them (signified by gray arrows from the parent to the child class). This system enforces a standardized format for each type of code file. All "sequences" are required to behave following the rules of the parent sequenceSpec class to compile properly, or else they are not loaded when PolyChemPrint starts. In addition to minimizing runtime errors, this approach provides another way to isolate the user from having to deal with repetitive boilerplate code common across many different objects. For example, there is no need to explicitly write logging methods for each sequence, because they already inherit them from the loggerSpec Abstract Base Class. Finally, the main method contains the data structures, which hold all the instantiated objects as well as containing all Menu classes and driving the user interface.

4 | HARDWARE IMPLEMENTATION AND EXAMPLES

In our laboratory, we have implemented the PolyChemPrint framework for three rapidly interchangeable configurations; (1) low-pressure extrusion, (2) high-pressure, variable-temperature extrusion, and (3) laser patterning. In this section, we will focus on the design and capabilities of each configuration and provide illustrative examples reflecting the range of materials that can be deposited. For completeness, extended hardware specifications, part numbers, mechanical drawings, and all CAD files for custom parts are provided inSI Section 3. Further details of printing parameters and materials used for demonstration are provided in the Methods section. Photographs of the printer (Figure S2) and toolheads (Figures S5, S6, S13) are provided in the SI.

4.1 | Commonalities: Motion control and software integration

A commonality between all three printing configurations is the motion platform. Our implementation is built around a consumer 3D printer (Aleph Objects Lulzbot Taz 6). We have found this platform to be versatile and convenient for studying a wide range of materials due to its large and open build area, maximum bed temperature of up to 120°C, and X/Y axes speed range of 15 mm/min–12 m/min. While the open-source hardware and software documentation provided by this manufacturer were particularly convenient for development, in practice any consumer 3D printer running the very popular Marlin firmware could be incorporated relatively easily (as noted in Section 2). In modifying the printer for materials research, we have not impaired the original FDM capabilities of the printer in any way, and in fact frequently switch back to FDM mode to print the holders and peripheral parts needed for these tools. Aside from swapping out tool heads and FDM peripherals and the addition of a glass or aluminum plate to protect the PEI bed surface (SI 3.1), the printer is used without hardware modification. A limitation of using consumer 3D printers is that printhead lateral movement is generally accomplished by belt-driven motors, with precision generally limited to ~10 μm, and dimensional accuracy on the order of 100 μm. For applications requiring higher precision, a combination of industrial linear stages must be used (at much higher cost). While somewhat more challenging to implement, the PolyChemPrint software can also accommodate the use of even mismatched linear stages simply through incorporating a new "axes" driver file that either directly addresses the commercial device via pySerial, or through a manufacturer specific communications interface.

Regardless of the specifics of the dispensing and motion hardware, well-synchronized hardware and software integration is critical for accurate pattern deposition. We highlight this point in Figure 3, using low-pressure direct-write printing of a bottlebrush block copolymer in a simple meanderline pattern. Panel (A) depicts a film with good synchronization between hardware and software, with the resulting pattern having sharp 90-degree corners. In Panel (B) we illustrate the effect of a gross hardware synchronization issue: in this case a fault in timing, where the y-axis motion is delayed by 10% versus perfect synchronization, leading to clear distortion of the pattern at the trailing edge of corners and over-extrusion (blob formation) at the leading edge. This type of timing issue can readily occur if using independently controlled X and Y stages rather than routing control through a unified control software.

Even when a unified control software is used, care must be taken in hardware configuration and software design to minimize latency in command execution during printing. Panel C contains an example of how these more subtle issues can lead to loss of pattern fidelity. In this example, the rate at which the software checks the hardware status (read time-out threshold) was set to a value of 0.1 s (vs. 0.001 s in Panel A). The resulting slight pause between execution of successive motion commands results in over-extrusion of solution at corners. A similar effect was observed when, in earlier versions of the program, the software loop that generated the list of hardware commands from user-specified parameters was executed at run time (as opposed to prior). In that scheme, after printing a pair of lines, an internal loop counter would increment, and the code would evaluate the counter versus the user's desired number of lines before either printing the next pair or ending the sequence. This again resulted in momentary pauses at some corners, leading to a similar over-extrusion issue. A shift in the software design to decouple pattern generation and execution into two steps has resolved that issue.

We note that for our experiments, we have found computational demands to not be very high, and that running PolyChemPrint on a salvaged (low-spec) Linux PC to be entirely sufficient. Communication with all hardware is accomplished across either USB or RS-232 serial cables. Modern RS-232 to USB adapters are also compatible with the PolyChemPrint software and are available at low cost. We anticipate that incorporating feedback or simultaneous recording of video or spectroscopic data will require higher performance computer hardware.

4.2 | Low-pressure extrusion

The low-pressure extrusion configuration (SI 3.2.1) is targeted at printing of low to medium viscosity fluids and is rendered in Figure 4(A) (photographs provided in Figure S5 C,D). The tool head in this case is a disposable polypropylene syringe connected via gas tubing to a pneumatic dispenser (Nordson EFD Ultimus V). Syringe barrels are terminated with a Luer-lock fitting, allowing for a wide selection of printing needles with task-specific diameter (generally, we use 0.11–0.60 mm inner

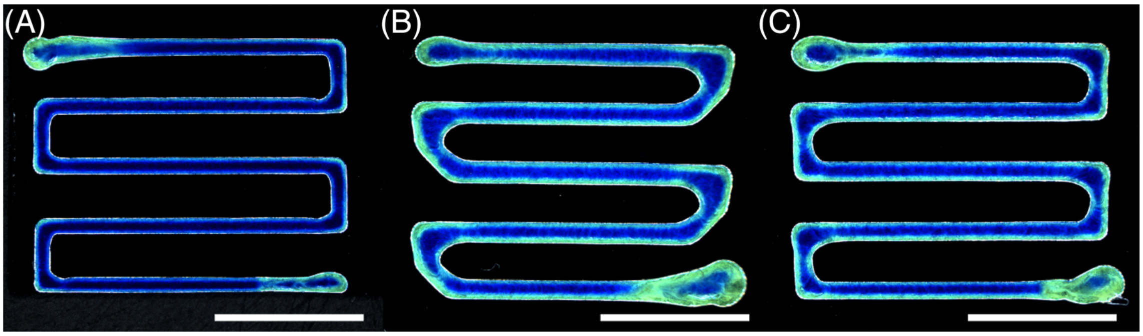

FIGURE 3 Bottlebrush block copolymer meanderline patterns printed using the low-pressure configuration (see Section 4.2). Panel (A) demonstrates good pattern fidelity with sharp, well-defined corners. Panel (B) demonstrates the effect of poor coordination between the X- and Y-axis motion commands leading to poor pattern fidelity. Panel (C) demonstrates the effect of poor optimization of hardware configuration or pattern generation in the software leading to over-extrusion at corners. In all cases, the white scale bar represents 0.5 cm

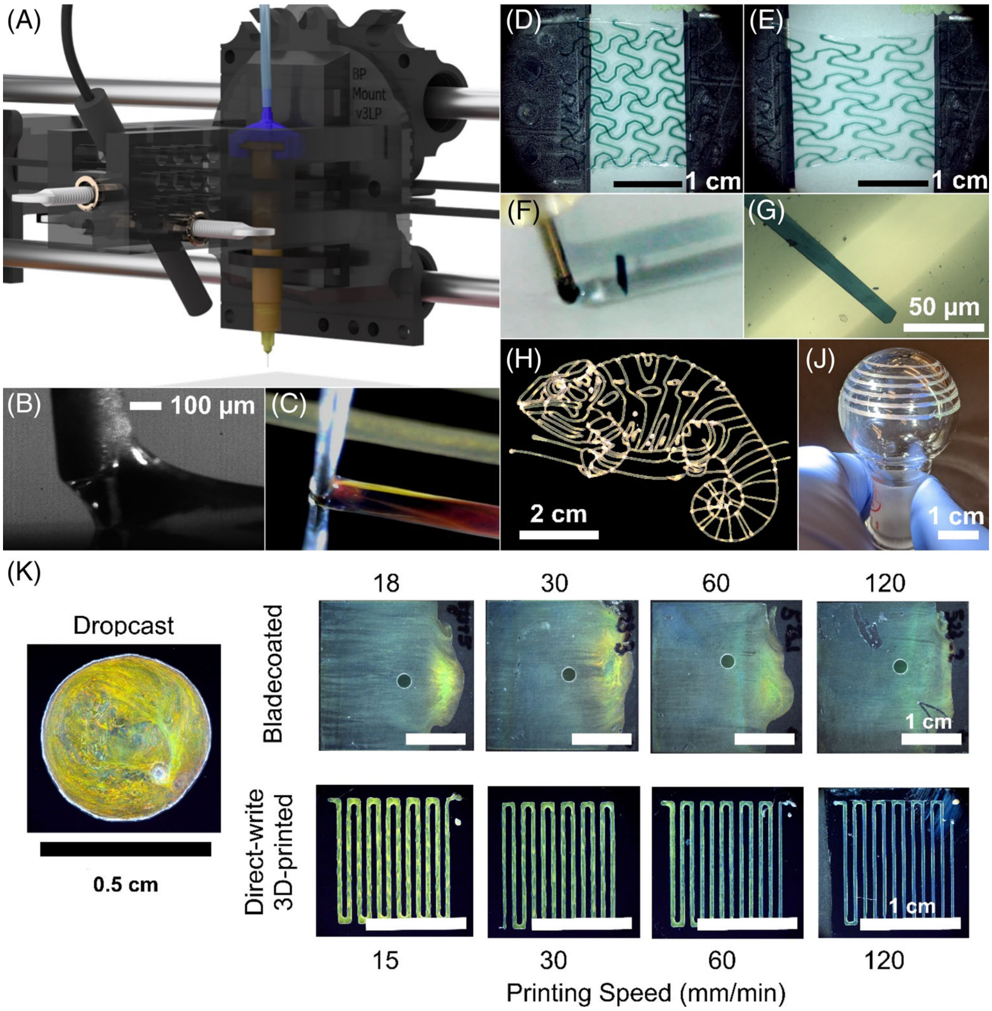

FIGURE 4 (A) Render of the low-pressure extrusion setup. Photographs provided in Figure S5. 3D-printed nozzle/camera holder is rendered as partially transparent for ease of viewing. (B and C) side and top view of the meniscus during printing of a PLA-b-PDMS bottlebrush block copolymer ink. (D and E) image of a DPP2T-TT conjugated polymer pattern printed onto a stretchable SEBS rubber film at 0% (D) and 50% (E) strain. (F) Pen camera still image from during printing of PEDOT:PSS electrodes onto a single crystal of the organic semiconductor TIPS-pentacene placed on a PVP:HDA-coated PET substrate. (G) Microscope image of deposited PEDOT:PSS electrodes. (H) Composite microscope image of a PLA-b-PDMS bottlebrush block copolymer film printed in the pattern of a chameleon. (J) The same PLA-b-PDMS bottlebrush block copolymer printed onto the surface of a round-bottom flask. Images B, C, H reprinted with permission from Patel et al.[23] under the CC BY-NC license. (K) Images of PLA-b-PDMS bottlebrush block copolymer films prepared at room temperature by dropcasting, bladecoating, and direct-write 3D printing from a 100 mg/ml solution in THF. PEDOT:PSS, poly(3,4-ethylenedioxythiophene)-poly(styrenesulfonate); PET, polyethylene terephthalate; PVP:HDA, poly(4-vinylphenol):4,4′-(hexafluoroisopropylidene)diphthalic anhydride; SEBS, styrene-ethylene-butylene-styrene

diameter, 20–32 gage, although similar printing setups using nozzle diameters as small as 1 μm have been reported[25]). Polyethylene pistons are used when printing from volatile organic solvents and when vacuum backpressure is applied to prevent dripping of solutions of sufficiently low viscosity. With this setup, the syringe backpressure during dispensing can be adjusted between 0 and 100 psi (0–689 kPa). For ease of positioning the nozzle and in situ monitoring of the printing process, we have 3D-printed a holder with a pen-camera mount as shown depicted in the figure.

We have used this configuration to print a range of materials in the style of “direct-write”[26] 3D printing. In this scheme, the nozzle drags a liquid ink meniscus across the surface, leaving a thin film of solute as the volatile solvent dries. Figure 4(B),(C) present a typical side and top view of an ink meniscus during printing. This technique has wide applicability for patterning functional materials with diverse applications. In Figure 4(D),(E) and Video S1 we demonstrate printing of a semiconducting polymer with diketopyrrolopyrrole acceptor and thienothiophene donor groups, DPP2T-TT, in chlorobenzene onto a flexible styrene-ethylene-butylene-styrene (SEBS) rubber film. By modulating printing speed and applied pressure, film thickness was optimized to allow for subsequent stretching up to 50% strain without apparent cracking or buckling of the film. We have also found the technique to be convenient for deposition of conductive poly(3,4-ethylenedioxythiophene)-poly (styrenesulfonate) (PEDOT:PSS) electrode patterns onto organic semiconductor films and crystals,[27] as demonstrated in Figure 4(F),(G). After optimizing printing speed and ink formulation, reproducible channel lengths as small as ~50 microns could be achieved when printing on polyethylene terephthalate (PET) substrates with a thin poly(4-vinylphenol):4,4′-(hexafluoroisopropylidene) diphthalic anhydride (PVP:HDA) passivation layer. PEDOT:PSS conductivity for deposited films was measured to be 169.75 ±18 S/cm (SI 4.1), comparable to the 150 S/cm previously reported for spincast films.[28] Finally, we demonstrate the ability to deposit highly complex 2D and 3D patterns in Figure 4(H),(J), using a PLA-b-PDMS bottlebrush block copolymer in tetrahydrofuran as the model system. The chameleon pattern shown in Figure 4 (H) was illustrated and converted to G-code in Inkscape before executing through the PolyChemPrint software. The 3D spiral pattern was modeled in Autodesk Inventor and sliced in Cura before execution through the PolyChemPrint software. As we report in detail elsewhere,[23] the photonic properties (color) of the film were found to be highly sensitive to easily modified printing parameters such as nozzle speed and substrate temperature. Compared to deposition by drop-casting and blade-coating, 3D-printed films exhibited more uniform color, and readily tunable domain spacing (Figure 4(K), SI 4.2).

4.3 | High-pressure, temperature-controlled extrusion

The high-pressure extrusion configuration (SI 3.2.2, Figure 5(A), Figure S6) is suitable for extruding high viscosity and thermoplastic materials and is a variation of a design previously reported by White et al.[29] Here, the tool head is a pressure-multiplying attachment (hp3cc from Nordson EFD) with gas supply from the same Ultimus V dispenser as in the low-pressure case. To provide stable

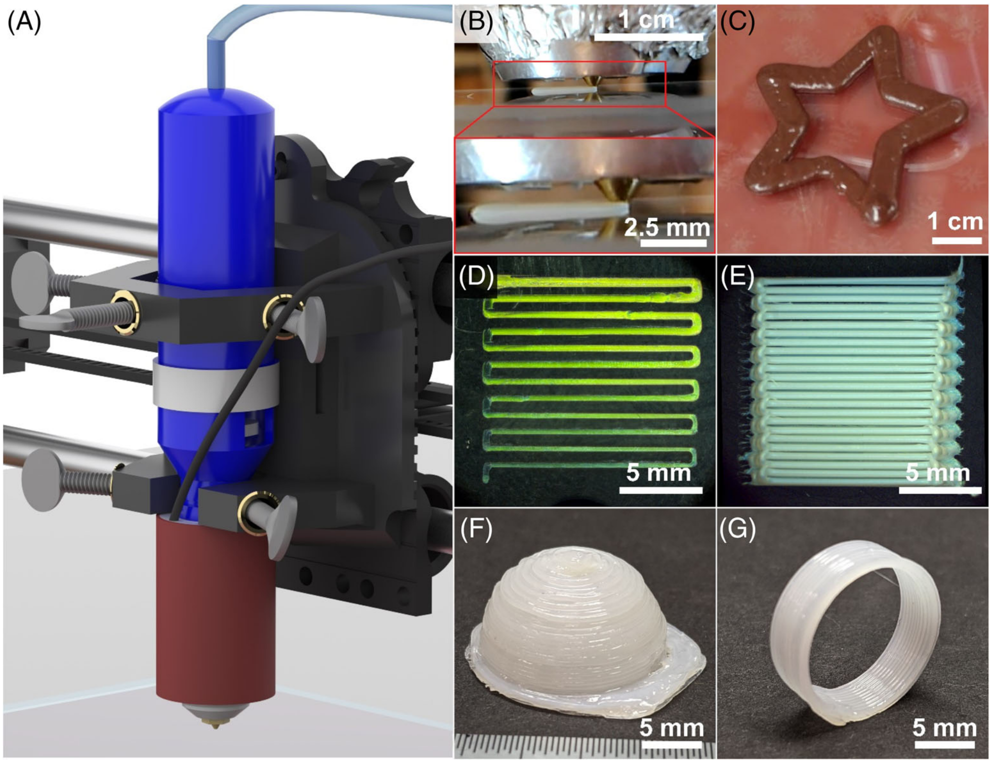

FIGURE 5 (A) Render of the high-pressure, temperature-controlled extrusion setup. Photographs provided in Figure S6. (B) Image of the nozzle during printing of a thermoplastic polymer with a closeup in the inset. (C) Image of a single-layer print using chocolate. (D) Image of a meanderline pattern printed using a PS-b-PLA bottlebrush block copolymer where each pair of lines is printed at progressively increasing speed. (E) Image of the same PS-b-PLA bottlebrush block copolymer printed at a higher pressure as a five-layer stack. (F,G) image of a SiO_2-PE-oil composite printed into the shape of a domed hat and a ring. Image G reprinted with permission from Zhou et al.[30] Copyright (2020) American Chemical Society

heating, we use a cylindrical resistive heater fitted to the extruder, with machined spacers with external insulation. The temperature is set and maintained by a closed-loop controller (Watlow EZ-Zone) connected to the barrel with an embedded thermocouple probe. Because we frequently need to use temperatures exceeding the softening point of the disposable plastic syringe barrels, we instead use a machined aluminum barrel (volume ~ 3 ml) with an aluminum piston sealed by a greased Viton O-ring. We have modified the endpiece of the hp3cc attachment to accommodate the metal barrel mated to either metal Luer-Lock syringe tips, or brass consumer 3D printer nozzles. Based on the degradation temperature of the Viton O-ring, the system is capped at ~210 °C. Using the hp3cc attachment, the pressure applied to the back of the piston can reach up to 700 psi (~4800 kPa), although the actual pressure applied to the fluid is reduced somewhat by the friction of the piston.

The key benefit of this configuration versus traditional FDM 3D-printing is that a wider (non-filament-forming) material palette is made accessible by the ability to load the barrel with powders or pre-formed cylinders of feedstock that can then be heated and fused within the barrel prior to and during extrusion to produce uniform, bubble-free, extrudate (Figure 5(B)). Because of the interchangeable nature of tools in the PolyChemPrint software, we retain full flexibility in terms of pattern generation. Figure 5(C), Video S2, demonstrates printing of a single-layer chocolate pattern from G-code generated from a 2D vector image in Inkscape. Figure 5(D),(E) demonstrate printing of single and multi-layered meanderline patterns using the PolyChemPrint Sequence library, with the former taking advantage of a parameter-screening routine that allows for systematic variation of printing speed between successive sets of lines. Finally, by generating a CAD drawing and slicing in Cura to obtain a G-code file, it was possible to print the complex three-dimensional "domed hat" and ring patterns shown in Figure 5(F),(G); Video S3. Here the material used was a SiO_2-PE-oil composite[30] and the resulting print has well-defined layers comparable to those prepared using traditional FDM feedstocks and tooling.

4.4 | Laser patterning

Finally, we have adapted the printer into a laser etching device (SI 3.2.3, Figure 6(A), Figure S13) by mounting using a 6-watt laser (Opt Laser PLH3D-6 W-XF) as the toolhead and sending instructions to an Arduino-based controller. When using the printer for this purpose, we add a fire-resistant fabric curtain around the enclosure and replace the glass bed plate with a thin sheet of anodized aluminum capable of withstanding the direct beam. The system is fully capable of traditional laser-cutting tasks such as patterning on combustible substrates (e.g., the burned paper sample in Figure 6(B), Video S4). In addition, patterning of functional materials can be accomplished. In Figure 6(C),(D), we show the results of irradiating a thin photonic crystal-forming PS-b-PLA bottlebrush block copolymer film that was spin-cast on a glass substrate with a sacrificial paper layer beneath it. Combustion of the sacrificial paper leads to local heating, achieving laser annealing of the polymer without direct absorption. UV–Vis spectroscopy corroborates the shift in peak reflectance within irradiated regions, corresponding to reorganization and growth of the block copolymer lamellar domains.

5 | MATERIALS AND METHODS

Poly(lactic acid)-b-poly(dimethyl siloxane) (PLA-b-PDMS) bottlebrush block copolymers with total backbone degree of polymerization of 400 were synthesized as reported elsewhere[23] and dissolved in tetrahydrofuran

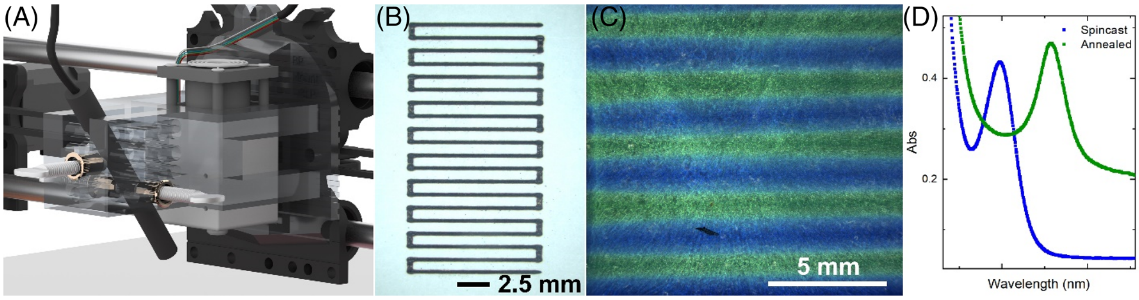

FIGURE 6 (A) Render of the laser toolhead. Photographs provided in Figure S13. (B) Image of a meanderline pattern burned into paper by the laser. (C) Image of a PS-b-PLA bottlebrush block copolymer film after laser irradiation in a repeated line pattern. (D) UV–Vis absorption spectra confirming the shift in reflected wavelength by the block copolymer before and after laser annealing

(Macron). Poly(lactic acid)-b-poly(styrene) (PLA-b-PS) bottlebrush block copolymers with total backbone degree of polymerization of 400 were dissolved in Toluene (Sigma Aldrich) and synthesized according to a previously published procedure.[31]

The conjugated polymer DPP2T-TT, with diketopyrrolopyrrole acceptor and thienothiophene donor groups was provided by the group of Dr. Jianguo Mei at Purdue University and dissolved in chlorobenzene (Alfa Aesar). The styrene-ethylene-butylene-styrene (SEBS) substrate was prepared by spin coating 200 mg/ml SEBS (Tuftec H1052 by AK elastomer) in toluene solution at 2000 rpm for 30 s on an n-octyldecyltrichlorosilane (OTS, 95%; Acros) treated glass slide. (OTS treatment protocol reported elswhere[32]). The stretchable DPP2T-TT/SEBS film on the glass slide was transferred to a computer-controlled motorized stage (Thorlabs MTS50-Z8) for a stretching test via a double-sided-tape-assisted peel-off method.

A 1.3 wt% poly(3,4-ethylenedioxythiophene)-poly (styrenesulfonate) (PEDOT:PSS) ink was prepared from dry re-dispersible pellets (Sigma-Aldrich) dissolved in (6:4, vol:vol) deionized water and isopropyl alcohol. The protocol for preparation of 6,13-Bis(triisopropylsilylethynyl) pentacene (TIPS-P, ≥99%, HPLC, Sigma-Aldrich) crystals and deposition onto PET substrates coated with a thin layer of crosslinked PVP-HDA ([Poly(4-vinylphenol] [Mw = 25,000 g mol − 1, Sigma- Aldrich], 4,4′-(hexafluoroisopropylidene)diphthalic anhydride [99%, Sigma-Aldrich]) from propylene glycol monomethyl ether acetate (PGMEA, ≥ 99.5%, Sigma-Aldrich) is reported elsewhere.[27]

The nanoporous SiO_2-PE-oil feedstock was prepared by mixing SiO_2 particles (< 10 μm, Sigma-Aldrich), high-density polyethylene pellets (Sigma Aldrich), ultrahigh molecular weight PE powder (Alfa Aesar) and paraffin oil (light, Fisher Chemical) as reported elsewhere.[30]

PS-b-PLA bottlebrush block copolymer films for laser annealing were prepared by spin-coating 100 μl of 128 mg/ml solution in Toluene (Sigma Aldrich) onto plasma-treated Corning glass substrates. The best film uniformity was achieved following a 3-step procedure: for 30s, first accelerating at 100 rpm/s to hold at 1000 rpm, then for 15 s accelerating @1200 rpm/s to hold at 3000 s, and finally spinning down to 0 rpm @1200 rpm/s. Solution was injected during the first step.

6 | CONCLUSION

In conclusion, we present a free and open source hardware and software framework for benchtop AM of functional materials. These tools provide the necessary infrastructure for precise control of non-FDM tool heads in generating arbitrarily complex patterns leveraging existing 3D slicing software. Illustrative examples of patterning materials with a wide range of functional and mechanical properties demonstrate the widespread utility of this approach. We hope this work helps to reduce the barrier to entry in this exciting and rapidly developing field.

ACKNOWLEDGMENTS

We would like to thank Prof. Jianguo Mei and graduate student Xuyi Luo for providing the DPP2T-TT semiconducting polymer and K. Zhou, R. Tao, and L. Cai for providing the SiO_2-PE composite material. Experiments were carried out, in part, at the MRL Central Research Facilities, University of Illinois. This work was supported by the NSF under DMREF award no. DMR-1727605. Ying Diao acknowledges partial support by the NSF CAREER award under grant no. NSF DMR 18–47828. Siqing Wang and Ying Diao acknowledge partial support by the NASA Early Career Faculty Award under grant no. 80NSSC21K0070.

CONFLICT OF INTEREST

The authors declare that they have no competing interests.

DATA AVAILABILITY STATEMENT

All data needed to evaluate the conclusions in the article are present in the article and/or the Supplementary Materials. Additional data related to this article may be requested from the authors.

ORCID

Bijal B. Patel https://orcid.org/0000-0002-8015-9075
Damien Guironnet https://orcid.org/0000-0002-0356-6697
Ying Diao https://orcid.org/0000-0002-8984-0051

REFERENCES

[1] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Q. Nguyen, D. Hui, *Compos., Part B* **2018**, *143*, 172. https://doi.org/10.1016/j.compositesb.2018.02.012.

[2] X. Wang, M. Jiang, Z. Zhou, J. Gou, D. Hui, *Compos,, Part B* **2017**, *110*, 442. https://doi.org/10.1016/j.compositesb.2016.11.034.

[3] S. J. Leigh, R. J. Bradley, C. P. Purssell, D. R. Billson, D. A. Hutchins, *PLoS One* **2012**, *7*(11), e49365. https://doi.org/10.1371/journal.pone.0049365.

[4] R. D. Farahani, M. Dubé, D. Therriault, *Adv. Mater.* **2016**, *28* (28), 5794. https://doi.org/10.1002/adma.201506215.

[5] M. Nadgorny, A. Ameli, *ACS Appl. Mater. Interfaces* **2018**, *10* (21), 17489. https://doi.org/10.1021/acsami.8b01786.

[6] J. A. Lewis, *Adv. Funct. Mater.* **2006**, *16*(17), 2193. https://doi.org/10.1002/adfm.200600434.

[7] B. Y. Ahn, S. B. Walker, S. C. Slimmer, A. Russo, A. Gupta, S. Kranz, E. B. Duoss, T. F. Malkowski, J. A. Lewis, *J. Visualized Exp.* **2011**, *58*, e3189. https://doi.org/10.3791/3189.

[8] Z. Zhu, S.-Z. Guo, T. Hirdler, C. Eide, X. Fan, J. Tolar, M. C. McAlpine, *Adv. Mater.*, **2018**, *30*(23), 1707495. https://doi.org/10.1002/adma.201707495.

[9] A. D. Valentine, T. A. Busbee, J. W. Boley, J. R. Raney, A. Chortos, A. Kotikian, J. D. Berrigan, M. F. Durstock, J. A. Lewis, *Adv. Mater.* **2017**, *29*(40), 1703817. https://doi.org/10.1002/adma.201703817.

[10] J. J. Adams, E. B. Duoss, T. F. Malkowski, M. J. Motala, B. Y. Ahn, R. G. Nuzzo, J. T. Bernhard, J. A. Lewis, *Adv. Mater.* **2011**, *23*(11), 1335. https://doi.org/10.1002/adma.201003734.

[11] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, J. A. Lewis, *Adv. Mater.* **2014**, *26*(36), 6307. https://doi.org/10.1002/adma.201400334.

[12] J. H. Kim, S. Lee, M. Wajahat, H. Jeong, W. S. Chang, H. J. Jeong, J.-R. Yang, J. T. Kim, S. K. Seol, *ACS Nano* **2016**, *10*(9), 8879. https://doi.org/10.1021/acsnano.6b04771.

[13] M. K. Gelber, G. Hurst, T. J. Comi, R. Bhargava, *Addit. Manuf.* **2018**, *22*, 38.

[14] G. Siqueira, D. Kokkinis, R. Libanori, M. K. Hausmann, A. S. Gladman, A. Neels, P. Tingaut, T. Zimmermann, J. A. Lewis, A. R. Studart, *Adv. Funct. Mater.* **2017**, *27*(12), 1604619. https://doi.org/10.1002/adfm.201604619.

[15] M. K. Hausmann, P. A. Rühs, G. Siqueira, J. Läuger, R. Libanori, T. Zimmermann, A. R. Studart, *ACS Nano* **2018**, *12* (7), 6926. https://doi.org/10.1021/acsnano.8b02366.

[16] J. M. McCracken, A. Badea, M. E. Kandel, A. S. Gladman, D. J. Wetzel, G. Popescu, J. A. Lewis, R. G. Nuzzo, *Adv. Healthcare Mater.* **2016**, *5*(9), 1025. https://doi.org/10.1002/adhm.201500888.

[17] T. J. Hinton, Q. Jallerat, R. N. Palchesko, J. H. Park, M. S. Grodzicki, H.-J. Shue, M. H. Ramadan, A. R. Hudson, A. W. Feinberg, *Sci. Adv.* **2015**, *1*(9), e1500758. https://doi.org/10.1126/sciadv.1500758.

[18] Q. Gao, Y. He, J. Fu, A. Liu, L. Ma, *Biomaterials* **2015**, *61*, 203. https://doi.org/10.1016/j.biomaterials.2015.05.031.

[19] B. G. Compton, J. A. Lewis, *Adv. Mater.* **2014**, *26*(34), 5930. https://doi.org/10.1002/adma.201401804.

[20] D. B. Kolesky, R. L. Truby, A. S. Gladman, T. A. Busbee, K. A. Homan, J. A. Lewis, *Adv. Mater.* **2014**, *26*(19), 3124. https://doi.org/10.1002/adma.201305506.

[21] H. H. Hwang, W. Zhu, G. Victorine, N. Lawrence, S. Chen, *Small Methods* **2018**, *2*(2), 1700277. https://doi.org/10.1002/smtd.201700277.

[22] B. B. Patel, Y. Diao, *Nanotechnology* **2018**, *29*(4), 044004. https://doi.org/10.1088/1361-6528/aa9d7c.

[23] B. B. Patel, D. J. Walsh, D. H. Kim, J. Kwok, B. Lee, D. Guironnet, Y. Diao, *Sci. Adv.* **2020**, *6*(24), eaaz7202. https://doi.org/10.1126/sciadv.aaz7202.

[24] Kramer, T. R.; Proctor, F. M.; Messina, E. R. The NIST RS274NGC Interpreter - Version 3. **2000**.

[25] B. Y. Ahn, E. B. Duoss, M. J. Motala, X. Guo, S.-I. Park, Y. Xiong, J. Yoon, R. G. Nuzzo, J. A. Rogers, J. A. Lewis, *Science* **2009**, *323*(5921), 1590. https://doi.org/10.1126/science.1168375.

[26] I. Gibson, D. W. Rosen, B. Stucker, *Additive Manufacturing Technologies*, Springer, Boston, MA **2010**. https://doi.org/10.1007/978-1-4419-1120-9.

[27] S. K. Park, H. Sun, H. Chung, B. B. Patel, F. Zhang, D. W. Davies, T. J. Woods, K. Zhao, Y. Diao, *Angew. Chem. Int. Ed.* **2020**, *59*(31), 13004. https://doi.org/10.1002/anie.202004083.

[28] H. Bednarski, B. Hajduk, J. Jurusik, B. Jarząbek, M. Domański, K. Łaba, A. Wanic, M. Łapkowski, *Acta Phys. Pol. A* **2016**, *130*(5), 1242. https://doi.org/10.12693/APhysPolA.130.1242.

[29] I. D. Robertson, M. Yourdkhani, P. J. Centellas, J. E. Aw, D. G. Ivanoff, E. Goli, E. M. Lloyd, L. M. Dean, N. R. Sottos, P. H. Geubelle, J. S. Moore, S. R. White, *Nature* **2018**, *557*(7704), 223. https://doi.org/10.1038/s41586-018-0054-x.

[30] K. Zhou, W. Li, B. B. Patel, R. Tao, Y. Chang, S. Fan, Y. Diao, L. Cai, *Nano Lett.* **2021**, *21*, 1493. https://doi.org/10.1021/acs.nanolett.0c04810.

[31] M. A. Wade, D. Walsh, J. Ching-Wei Lee, E. Kelley, K. Weigandt, D. Guironnet, A. Rogers, S. Color, *Soft Matter* **2020**, *16*(21), 4919.

[32] K. S. Park, J. J. Kwok, R. Dilmurat, G. Qu, P. Kafle, X. Luo, S.-H. Jung, Y. Olivier, J.-K. Lee, J. Mei, D. Beljonne, Y. Diao, *Sci. Adv.* **2019**, *5*(8), eaaw7757. https://doi.org/10.1126/sciadv.aaw7757.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Patel BB, Chang Y, Park SK, et al. PolyChemPrint: A hardware and software framework for benchtop additive manufacturing of functional polymeric materials. *J Polym Sci*. 2021;59:2468–2478. https://doi.org/10.1002/pol.20210086