ModElec: A Design Tool for Prototyping Physical Computing Devices Using Conductive 3D Printing

LIANG HE, Paul G. Allen School of Computer Science & Engineering, University of Washington, USA
JARRID A WITTKOPF, HP, USA
JI WON JUN, HP Labs, USA
KRIS ERICKSON, HP Incorporated, USA
RAFAEL ‘TICO’ BALLAGAS, HP Inc., USA

Integrating electronics with highly custom 3D designs for the physical fabrication of interactive prototypes is traditionally cumbersome and requires numerous iterations of manual assembly and debugging. With the new capabilities of 3D printers, combining electronic design and 3D modeling workflows can lower the barrier for achieving interactive functionality or iterating on the overall design. We present *ModElec*—an interactive design tool that enables the coordinated expression of electronic and physical design intent by allowing designers to integrate 3D-printable circuits with 3D forms. With *ModElec*, the user can arrange electronic parts in a 3D body, modify the model design with embedded circuits updated, and preview the auto-generated 3D traces that can be directly printed with a multi-material-based 3D printer. We demonstrate the potential of *ModElec* with four example applications, from a set of game controls to reconfigurable devices. Further, the tool was reported as easy to use through a preliminary evaluation with eight designers.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**.

Additional Key Words and Phrases: Conductive 3D printing, Electronic design, 3D modeling, Physical computing

ACM Reference Format:
Liang He, Jarrid A Wittkopf, Ji Won Jun, Kris Erickson, and Rafael ‘Tico’ Ballagas. 2021. ModElec: A Design Tool for Prototyping Physical Computing Devices Using Conductive 3D Printing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 159 (December 2021), 20 pages. https://doi.org/10.1145/3495000

1 INTRODUCTION

3D printers have evolved as personal-scale manufacturing tools to rapidly create physical shapes [2, 4]. Adding electronic components and circuitry to mechanical designs can turn 3D-printable objects into functional and interactive devices [22, 31, 32]. To create a successful integration of electronics into a 3D form, designers need to go through two mostly isolated workflows: 3D modeling and circuit assembly. Circuit assembly awaits the completion of 3D modeling and fabrication and may take numerous manual steps, including arranging electronic components and wiring. As a result, there is traditionally a high penalty for making changes to the 3D design after the circuitry has been added.

Authors’ addresses: Liang He, lianghe@cs.washington.edu, Paul G. Allen School of Computer Science & Engineering, University of Washington, 185 E Stevens Way NE, Seattle, WA, USA, 98195; Jarrid A Wittkopf, HP, 1501 Page Mill Road, Palo Alto, CA, USA, 94304-1126; Ji Won Jun, HP Labs, 1501 Page Mill Road, Palo Alto, CA, USA, 94304-1126; Kris Erickson, HP Incorporated, 1501 Page Mill Road, Palo Alto, CA, USA, 94304-1126; Rafael ‘Tico’ Ballagas, HP Inc. 1501 Page Mill Road, Palo Alto, CA, USA, 94304-1126.

2474-9567/2021/12-ART159 $15.00
https://doi.org/10.1145/3495000

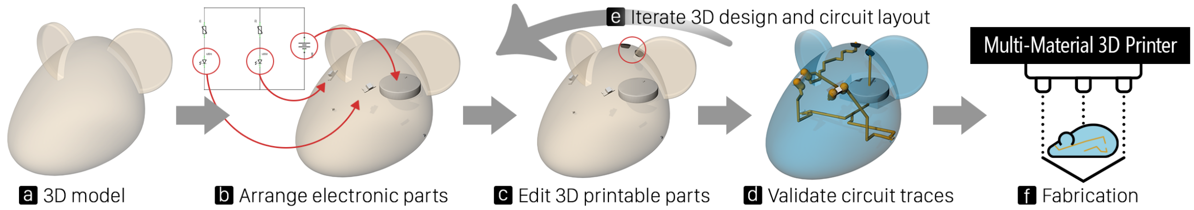

e Iterate 3D design and circuit layout
Multi-Material 3D Printer
a 3D model
b Arrange electronic parts
c Edit 3D printable parts
d Validate circuit traces
f Fabrication

Fig. 1. We introduce *ModElec*, an interactive design tool that allows designers to coordinate electronics in custom 3D models for prototyping physical computing devices. The workflow above shows how to integrate two LEDs as eyes into a rat body: (a) create a rat model in Rhino; (b) arrange electronic parts in the 3D design based on a circuit schematic; (c) manually add and edit parts that are conductive and 3D printable; (d) inspect the auto-generated trace layout and manually connect the two 3D printable touch pads to the circuit as a switch; (e) iterate the model and the circuit layout; and finally, (f) 3D print the internal traces, 3D printable parts, and the plastic body with a multi-material-based 3D printer that supports conductive 3D printing.

Conductive 3D printing has extended the design space of 3D printing interactivity [1] by converting 3D-printed objects into interactive tangible gadgets [35], speakers [13], and touch sensors [34]. Recently, researchers have explored novel fabrication techniques to add highly conductive routes to 3D objects for circuit wiring [12, 33, 38, 40, 43, 50]. Combining 3D-printable objects with electronic components and circuitry has been possible with the advent of these emerging new techniques for conductive 3D printing. These shifts have led to a re-thinking about what software tools can support the coordination of electronics and 3D models [10] while enabling *low threshold*, *high ceiling*, and *wide walls* in creative design and making [29]?

To answer this question, our work explores how to integrate the two isolated design workflows and support designers with the digital fabrication of 3D-printable interactive devices using highly conductive 3D printing. Recent work has explored approaches for adding external electronic parts and wires to 3D-printed objects, including assembly-aware tools for the arrangement of commodity electronic modules [5, 7], on-surface circuitry design [12, 40, 41, 43], embedded in-plane 3D printable circuit traces [26, 38], and 3D printable electronic components [6, 21, 24, 45]. However, these methods rarely investigate the design and control of non-planar tracing for an embedded circuit in a 3D form. In this work, we present *ModElec*, an interactive design tool built in Rhino, as a case study to examine how software tools support integrated 3D circuit design with novel conductive 3D printing. With *ModElec*, designers can experiment with different 3D circuit layouts while iterating the 3D form design without fabricating the model and assembling the electronic parts and wires (Figure 1), which are time consuming and cumbersome. *ModElec* fully automates the generation of internal electrical connections by using the entire 3D object volume for routing, which obviates the need of placing traces on the complex exterior surfaces and reduces the interference caused by accidental user touching in capacitive sensing-based applications. Further, unique part sockets with snap-fit designs are auto-generated for the ease of post-print electronic components assembly. We demonstrate the potential of *ModElec* with four interactive prototypes created with the design tool and the multi-material testbed printer and validate the usability of the proof-of-concept design tool through a preliminary user study.

In summary, this paper makes the following contributions: (i) an interactive design tool that enables the designer to iterate a 3D circuit layout in a custom 3D shape and facilitates designers; (ii) four example physical computing prototypes created with *ModElec* tool that showcases the benefits of our approach; and (iii) a preliminary user study that examines the usability of the design tool.

2 RELATED WORK

We build upon prior work on making interactive 3D-printed objects by embedding electronic components and conductive routes. Our work also relates to the design tools for adding electronics and circuitry to 3D-printed objects.

2.1 Interactive 3D-Printed Devices Using Embedded Electronic Parts

Researchers have explored using electronic components and devices to add functionality and interactivity to 3D-printed objects [1, 2]. One way is to incorporate commodity devices that have computation capabilities such as mobile phones [16–18, 47], sensors [11, 31, 37, 45], and actuators [20, 23, 27, 48] with 3D-printed objects. For example, Vázquez *et al.* [42] created a series of 3D-printed pneumatic controls such as buttons and knobs that detect user force actuation using embedded air pressure sensors. *Romeo* [19] embeds external electromotors in a 3D-printed object and converts the 3D model into a transformable robot for augmented functionalities. Besides computing devices, designers can also quickly create interactive prototypes using easy to use and readily available electronic modules and components [5, 7, 8]. For example, *SurfCuit* [40] allows the user to place electronic components such as resistors and integrated circuits on the exterior surface of a 3D-printed object and connect them using manually attached conductive traces. *MorphSensor* Zhu *et al.* [51] enables designers to reform plug-and-play sensor modules on 2D shapes into free-form arrangements for the 3D shape of a physical device. Recently, *MorphingCircuit* [43] turns a flat substrate where a mix of functional electronic components and modules are assembled into an interactive and programmed 3D object when triggered by heating. Moreover, researchers have also explored 3D printable electronic components [6, 9, 13, 28, 35]. For example, microelectronic components such as capacitors and passive wireless sensors can be 3D printed using fused deposition modeling technology and a multiple-nozzle printer [46]. While all these approaches have the potential to add computation and interactivity to 3D-printable designs, coordinating external electronic parts and 3D-printed highly conductive routes in the iterative design of a 3D model has not yet fully demonstrated. Our work not only supports the 3D arrangement of electronic parts, including commodity components (*e.g.*, diodes and resistors) and plug-and-play modules (*e.g.*, microcontrollers), but also provides design aids throughout the iterative 3D design process.

2.2 Fabrication of Circuit Routes for 3D-Printed Objects

To connect electronic parts that are attached to 3D-printed objects, researchers have explored automated and manual fabrication techniques creating electrical connections in 3D-printed devices [33, 40, 50]. One approach is to establish conductive traces on the surface of the 3D-printed object [12, 28, 30, 34, 43, 44, 51]. For example, highly conductive copper tapes are firmly affixed on the surface by melting the surface plastic as traces in *SurfCuit* [40]. *MorphingCircuit* [43] has conductive circuit traces created on a flat 2D substrate surface using silver screen printing and the traces conform to a programmed 3D shape under heating. Another method is to create internal channels inside 3D-printed objects for embedded routes [36, 41, 50]. For example, *FiberWire* [38] deposits silver epoxy on exposed carbon-fiber with plastic substrate to create 3D objects with embedded electrical traces. *CurveBoards* [50] creates pin holes such as those for a breadboard on an arbitrary 3D object's surface for circuit prototyping and connects those holes by filling conductive silicone compound in interconnected channels inside the 3D object. These methods do not fully utilize the entire object body for electrical routing due to fabrication limitations and complexity. Most relevant to our work, Savage *et al.* [33] used the entire volumetric space of the 3D object to generate internal pipes for connecting electronics. Those internal pipes become conductive after copper paint is filled in the post-print process. *ModElec* takes a similar method to generate internal electrical routes but applies electrical design rules to generated traces for 3D printing. Unlike the free-from and manually established connections in [33], our design tool automates the generation of 3D circuit traces based on a circuit schematic while allowing the user to manually add conductive connections in the 3D object.

2.3 Design Tools for Coordinating Electronics with 3D Mechanical Designs

Unlike printed circuit board (PCB) design on a flat sheet in common electronic design automation (EDA) software such as *Eagle*[1] and *Cadence*[2], designing circuits in 3D objects has two challenges: (i) arranging electronic parts in 3D is more difficult than planning on a 2D board [5, 27, 30] and (ii) routing the electrical connections in 3D is error prone due to the geometric constraints [28, 40]. To accommodate external electronics in 3D objects, researchers have explored placing custom containers that house electronic modules inside a 3D-pritned object [7, 49] or using sculpt-and-scan approaches to position electronics in fabricated objects [15, 32]. The design tools in such approaches are used for validating the spatial layout of electronics in the 3D object. For example, Desai *et al.* [5] created a computational design tool that optimizes the 3D enclosure based on user input shape to encapsulate inserted electronic components and devices. *PHUI-kit* [14] enables the user to drag and drop the digital models of physical widgets on curved 3D surfaces and the widgets can be re-positioned with implementation details hidden beneath the surface. These tools only take electronic components as input and the integrated wiring is not a part of the layout design. In contrast, *ModElec* allows the user not only to control the placement and orientation of individual electronic part, but also to plan conductive connections during the digital design phase. While previous works have developed novel techniques to fabricate highly conductive 3D-printable routes on flat layers [12, 38, 43, 51], our work adapts the PCB design principles to the 3D domain and extend [33] and [51] by enabling the control of each integrated electronic part, the adaptation of 3D circuit to 3D model changes, and the automation of 3D conductive routes complying with the electrical specifications.

3 FORMATIVE INTERVIEW

To understand the characteristics of the design tool that supports the integration of electronics and 3D-printable routes in 3D models, we conducted semi-structured interviews with three mechanical designers and two engineers, who know conductive 3D printing and have experiences in building interactive systems. The interview included a retrospect of one of their past projects that involved any sort of combining circuits with physical objects and elaborations on the pain points they faced with during the development of a physical computing system. We also solicited thoughts and suggestions on the user interface of the design tool.

Based on the interview results, we identified three key characteristics should be supported by computer-aided design (CAD) tools for the workflow of combining electronics and 3D forms. First, providing 3D representations of electronic components with accurate dimensions is critical for the integration. Without knowing the part dimensions, the user has difficulties gauging if electronics can fit in the desired places of the object. The lack of digital representation of electronic parts may lead to overall mechanical design changes, aesthetic sacrifices, extra fabrication time, or even extra manual assembly. Second, simplifying the cumbersome and error-prone wiring process for the circuit assembly. Traditionally, electrical wires are manually added to connect electronic components in the 3D-printed form, including thoughtful wire planning, extra crafting, and connection testing. Any failure in these processes may result in non-functional circuitry or modifications to the 3D form. Interviewees felt positive about the 3D-printable conductive routes and expected an automated routing mechanism in the design tool, which can reduce the human effort for wiring and the risk of problematic connections. However, electrical engineers expressed concerns about the resistivity of the 3D-printed traces and emphasized that certain design rules should be used for regulating the trace routing (*e.g.*, the minimum trace spacing). Finally, supporting the exploration of the layout of electronic components in the 3D object while ensuring a robust electronic assembly. Since the 3D modeling and circuit assembly workflows are combined, interviewees strongly suggested that the user can place electronic components freely and create the desired electrical layout through trial-and-errors with the design tool. While installing electronic parts on curved surfaces is often difficult, interviewees also questioned

[1] Autodesk Eagle: https://www.autodesk.com/products/eagle/

[2] Allegro PCB editor: https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html

how the 3D models created with such a tool could guarantee the successful assembly of electronic components after the device is 3D printed.

4 ModElec DESIGN TOOL OVERVIEW

Informed by the formative interview and the PCB design process, we developed *ModElec*—an interactive design tool that enables designers to create a 3D circuit layout, including interconnected electronic parts and 3D-printable conductive traces, in a custom 3D object via an extension to Rhino3D[3]. The *ModElec* design tool is built in C# using *RhinoCommon API*[4] and *Grasshopper Human UI*[5] and it is open source[6]. The tool aims to provide the following design aids for integrating an electrical design into a 3D model:

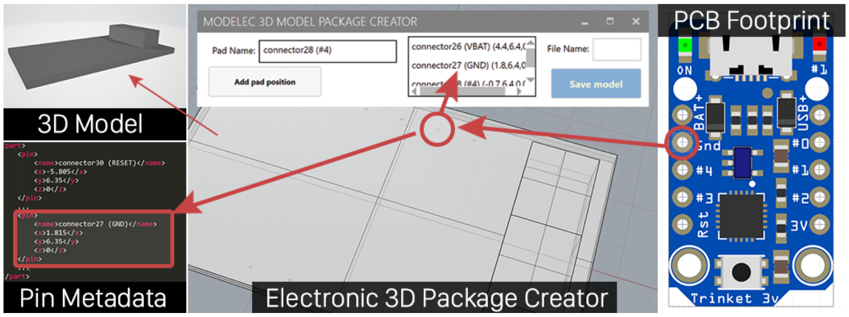
- Accurate 3D representations of electronic parts are provided for electronic arrangement in the 3D design.
- Edits on 3D model and the embedded circuit layout (i.e., the placement and orientation of electronic parts) are allowed at any point of digital design process.
- 3D-printable conductive routes are auto-generated and allow the user to manually edit routes. Other 3D-printable parts such as capacitive touchpads and resistors can be created with the tool.
- Unique sockets are automatically generated for the ease of the post-print assembly of electronic components.

4.1 Circuit Preparation and 3D Representations of Electronic Parts

To support the integration of circuits with 3D models, we developed *ModElec* design tool to recognize circuit schematics, which are diagrammatic abstractions of circuits. These circuit schematics created with three kinds of third-party EDA software—*Fritzing*[7], *Autodesk Eagle*, and *Cadence*—can be recognized by our design tool. When the circuit is loaded in the tool, three files are needed: a *Circuit Schematic* file that contains all the electronic parts and traces, a *Netlist* file that contains all logic connections within the circuit, and an *Image* filet that is used for displaying the circuit in the tool.

Since 3D representations of electronic parts are needed for the electronic arrangement, we use a library that contains 255 in total 3D representations of electronic parts downloaded from *SparkFun*[8]. These 3D models can already auto-match many parts provided by online platforms such as *Adafruit*[9] and *SparkFun*. In addition, to support a wider variety of electronic parts, we also developed a standalone *"Electronic 3D Package Creator"* tool for associating a custom 3D model with a specific electronic part (Figure 2). Like creating custom footprints for electronic components on a PCB, the user can create a 3D model for a specific electronic part using the dimensions defined in the part's datasheet and annotate the pin positions with points in Rhino following the part's PCB footprint. For each pin, the user types in the pin number and name, which can be found in the part's PCB footprint, and then selects the point that corresponds to the pin on the part's footprint. For example, Figure 2 shows that the user marks the *GND* pin on a Trinket board 3D model with the tool. Finally, the tool saves the 3D model of the electronic part as a STL file and exports all the pin information such as pin positions into an XML file, which are both recognizable by the *ModElec* design tool.

[3] Rhinoceros: https://www.rhino3d.com/
[4] RhinoCommon API: https://developer.rhino3d.com/api/
[5] Human UI: https://www.food4rhino.com/app/human-ui
[6] The GitHub repository of ModElec: https://github.com/EdigaHe/modelec
[7] Fritzing: https://fritzing.org/
[8] SparkFun: https://www.sparkfun.com/
[9] Adafruit: https://www.adafruit.com/



```
<part>
    <pin>
        <name>connector30 (RESET)</name>
        <x>-5.805</x>
        <y>6.35</y>
        <z>0</z>
    </pin>
    ...
    <pin>
        <name>connector27 (GND)</name>
        <x>1.815</x>
        <y>6.35</y>
        <z>0</z>
    </pin>
    ...
</part>
```

Pad Name: connector28 (#4)

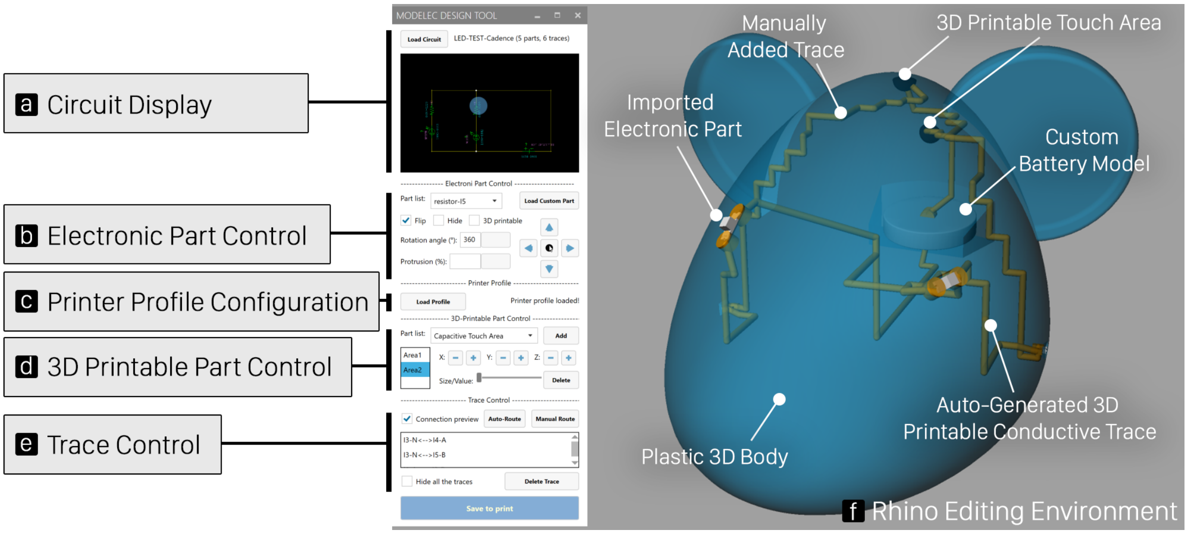
connector26 (VBAT) (4.4,6.4,0
connector27 (GND) (1.8,6.4,0
connector28 (#4) (-0.7,6.4,0

File Name:

Fig. 2. The user can customize a 3D representation of an electronic part and record all the pin information.

4.2 ModElec Workflow and User Interface

The *ModElec* design tool is composed of five parts: circuit display, electronic part control, printer profile configuration, 3D printable part control, and trace control (Figure 3). With the tool, the user creates a 3D model in the traditional Rhino editing environment, adjusts the placement and orientation of a selected electronic part from the imported circuit, adds 3D printable parts such as capacitive touch area if needed, previews the auto-generated traces in a 3D layout or manually adds routes, and iterates the 3D circuit layout and the model design for printing. Below, we describe each part of the user interface and implementation details of *ModElec* workflow.



LED-TEST-Cadence (5 parts, 6 traces)

Part list: resistor-I5

☑ Flip ☐ Hide ☐ 3D printable

Rotation angle (°): 360

Protrusion (%):

Printer profile loaded!

Part list: Capacitive Touch Area

Area1
Area2

X: − + Y: − + Z: − +

Size/Value:

☑ Connection preview

I3-N<-->I4-A
I3-N<-->I5-B

☐ Hide all the traces

Fig. 3. *ModElec* user interface includes (a) a circuit display, (b) an electronic part control panel, (c) a printer profile configuration panel, (d) a 3D printable part control panel, and (e) a trace control panel. All the imported 3D representations of electronic parts, auto-generated conductive traces, and the converted plastic body are displayed in (f) the Rhino 3D model editing environment.

4.2.1 *Electronic Part Control.*

Upon the *ModElec* design tool reads in the circuit schematic, the user needs to select a target 3D body to host the circuit, followed by an auto-arrangement of all electronic parts. Electronic parts are added to the selected body based on their logic connections defined in the Netlist file. For example, if a LED's anode is connected to the positive of the battery and its cathode is connected to a resistor, the LED is placed geometrically between the battery and the resistor in the 3D body. The circuit schematic appears in the circuit display panel and the total amount of parts and traces are also shown (Figure 3a). Then, the user can start adjusting the placement and orientation of the added electronic parts (Figure 4).

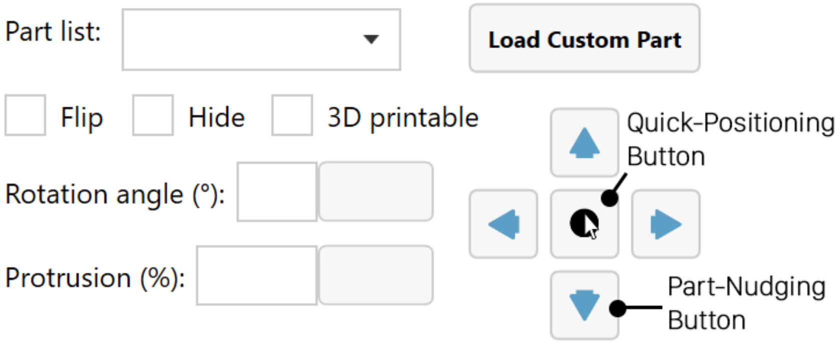


Fig. 4. The electronic part control panel in the *ModElec* design tool.

(a) *Select an electronic part and update the 3D representation of the part.* The user can select a target electronic part from a dropdown list and the selected part is highlighted in both the Rhino 3D scene and the circuit schematic (see the blue spotlight circle in Figure 3a). If the part does not have a corresponding 3D representation or the user has an alternative 3D model for the part, the user clicks on the *Load Custom Part* button to use a custom 3D representation for that part. For example, the coin battery is customized by the user in Figure 3.

(b) *Move, quickly relocate, and nudge the electronic part.* Besides freely dragging the part in the Rhino 3D scene, the tool also offers two other part moving options: *Quick-Positioning* and *Part-Nudging*. For *Quick-Positioning*, the user selects a position to insert the part from candidate points shown on the model surface by clicking on the *Quick-Positioning* button (Figure 4). When the user hovers the cursor over those points, the closest point to the cursor turns into a blue sphere with a red line indicating the position and the normal direction of that position on the surface (Figure 5a). With such visual assistance, the user can distinguish overlapping points that locate on two opposite surfaces but are both approachable and clickable by the mouse in Rhino. After a desired position is selected, the electronic part moves immediately to that place and all the candidate points disappear. For *Part-Nudging*, the user can finely move the part to the nearby positions in four directions by clicking on the four directional buttons (Figure 4). Since all the external electronic parts will be manually assembled after the body and traces are printed, the part snaps to the body's exterior surface automatically after each part moving and faces in the normal direction for the ease of post-print assembly. Moreover, the tool also inspects if electronic parts overlap with each other and reports an error if intersections exist.

(c) *Flip, rotate, and protrude the electronic part in place.* The user can also execute a series of operations on individual electronic part, including in-place part flip, part rotation, and part protrusion. If the electronic part's pins are outside of the 3D body, the user can flip the part by checking the Flip checkbox (Figure 5b). Electronic parts are constrained to rotate only around the normal axis at the part position on the surface for the easy part assembly (Figure 5c). When the electronic part is added to the 3D body, the part fully submerges under the surface by default but can be exposed by the user (Figure 5d). The part updates in real time in response to all these user operations.

(d) *Other electronic part controls.* Finally, the user can hide/show each electronic part and opt to use a 3D printable version for a specific part (Figure 4). For example, the tool currently can replace pin headers with 3D printable pads on the surface. It is open to more electronic components that are interchangeable with 3D-printable parts such as resistors and capacitors. The user can modify the 3D model in Rhino at any point and the embedded electronic parts reacts to model transformations (*i.e.*, translation, rotation, and scale) and morphing by auto-adapting to the updated body. If the 3D model scales and the electronic parts no longer can find sufficient room in the new model, the electronic parts are suspended outside the body and indicate the user to resize the shape.

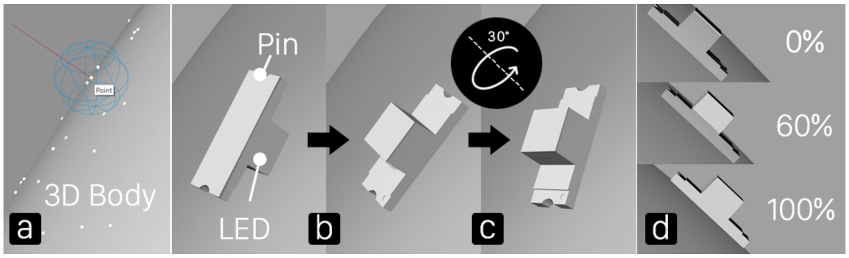


Fig. 5. The tool supports (a) quick part positioning, (b) part flip, (c) in-place part rotation, and (d) part protrusion.

4.2.2 Printer Profile for Conductive 3D Printing. Similar to the design rules that regulate the tracing in PCB design and manufacturing, design criteria for conductive 3D printing are critical in the design and fabrication of 3D-printable routes and electronic parts. For example, the resistivity of the conductive region and the minimum spacing between two conductive traces are important for routing in a 3D design. To support distinct design rules for different compatible 3D printers, the *ModElec* tool reads in and parses printing specifications that are defined in a JSON formatted file—*printer profile*—before 3D-printable electronic parts and routes are generated (Figure 3c).

4.2.3 3D Printable Electronic Part Control. To broaden the spectrum of 3D printable electronics enabled by conductive 3D printing, the tool provides three different electronic components for custom circuit design: *connection nodes*, *capacitive touch areas*, and *resistors* (Figure 6).

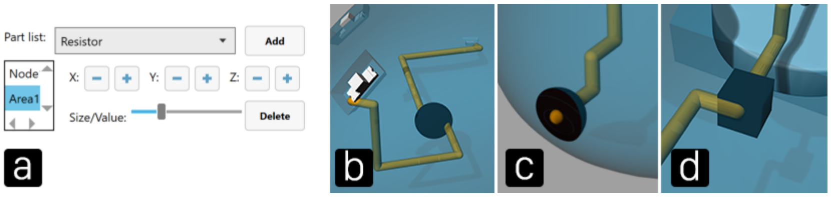

Part list: Resistor

Fig. 6. The user can control (b) a connection node, (c) a capacitive touch area, and (d) a resistor through the (a) 3D printable part control panel in the user interface of the *ModElec* design tool.

(a) *Connection nodes.*Spheres can be generated inside or on the surface of the 3D body as junction nodes such as those for connecting multiple traces and electronic parts in the circuit schematics (Figure 6b). For example, the user can use these nodes as intermediate "detour" nodes for trace re-routing under the circumstance that no direct route can be established between two pins. The added node moves along the X/Y/Z axis and resizes as the user controls through the user interface (Figure 6a).

(b) *Capacitive touch areas.* A conductive region can be added and exposed on the surface as interactive touchpads using capacitive sensing [34] (Figure 6c). Upon selected from the dropdown list, all the available locations are shown as points and the user needs to select one point like that for *Quick-Positioning* to place the touchpad. The touchpad moves only on the object's surface when the user clicks on the directional buttons and resizes in respond to the slider changes.

(c) *Resistors.* Like the connection nodes and capacitive touch areas, a printable resistor can be added to the 3D body and moves around inside the body (Figure 6d). However, the slider is used for controlling the resistor's resistance. Currently, the tool provided an estimated resistance for the added 3D-printable resistor by controlling the slider.

4.2.4 Trace Control. Finally, the tool allows the designer to preview the logic connections between electronic parts, execute auto-route based on the circuit schematic, and manually add traces (Figure 7).

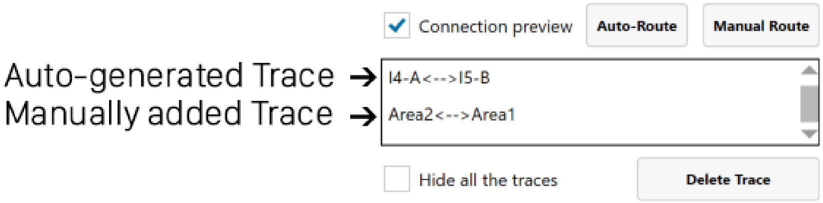


Fig. 7. The trace control panel in the design tool.

(a) *Logic connection preview.* Before the actual traces are deployed, the tool shows all the part connections by displaying red straight curves that connect part pins in the 3D space (Figure 8a). Like the logic connections in the traditional PCB design, these curves indicate the logic connections between pins rather than the real routing. The user can use these curve indicators to inspect if the routes have potential crossings before the actual traces are generated. If all the circuit traces are generated successfully, the red connection curves disappear. A residual curve indicates that the tool fails to generate a 3D trace between the two pins.

(b) *Sockets for hosting the electronic parts.* To host the external electronic parts on the surface, the tool creates sockets for the electronic parts as the traces are generated (Figure 8b). These sockets are inversed truncated pyramids that are perpendicular to the body surface. If the socket is large and shallow, two snap-fit guides are also automatically generated on the longer sides. The electrical part can slide into the socket and stay firmly with the protection of the two snap-fits. Figure 8b shows examples with (small LED) and without (large speaker) snap-fit designs. The tool generates part sockets in two ways for electronic parts placed on curved surfaces. For those part pins that are below the body surface, the body simply opens a socket for hosting the part. When the part pins are outside the body, the body protrudes the surface at the part position and creates a base for the part pins to stand (Figure 8b). If the electronic part is switched to 3D printable part by the user (see 4.2.1), no part socket will be created (Figure 8c).

(c) *Auto-route.* The goal of auto-routing is to establish the shortest 3D connections between intact positions of electronic components for minimal resistance rather than reducing fabrication time, which is largely dependent on the shape of the model and how the model is printed. The tool takes the following steps for auto-routing. First, the tool recognizes all the connection pairs and finds the center position if multiple pins are logically connected. Each pair of the center position and the connected part pin position constructs a new connection pair. For all these new connection pairs, the tool then examines if a straight solid pipe can be created within the object body and without intersecting with other pipes where the pipe diameter is defined by the printer profile. For those point pairs that cannot establish straight connections, the tool sorts the point pairs based on an ascending order

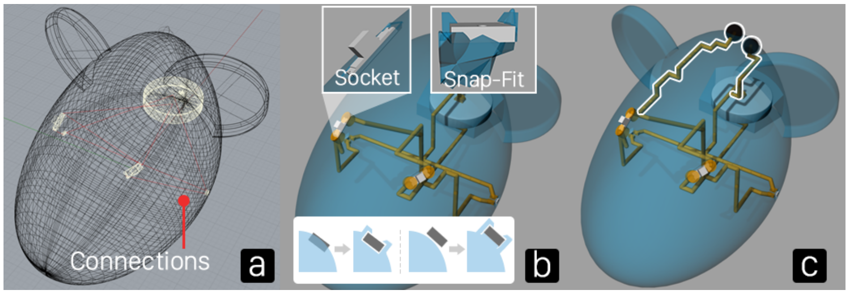


Fig. 8. The tool provides (a) logic connection preview for the user and converts the body by creating (b) part sockets and auto-generated traces. The user can also (c) manually add traces to the body.

of the distance between the two points and then generates a 3D path between each two-point pair in this group using A^* *path search algorithm* [33]. As a result, the two points that have shorter Euclidean distance have higher priority to generate the trace. To avoid the overlapping with already-generated pipes and ensure the minimum route spacing defined in the printer profile, the algorithm is revised to exclude those positions that are within or too close (less than the defined minimum trace spacing) to the generated pipes for the shortest path searching. After computation, the tool uses the series of points output by the algorithm to construct a polyline and then sweeps a circle along this polyline using Rhino Sweep function to create a solid route (Figure 8b). If no route can be generated with a two-point pair, the user needs to manually create the trace, which will be discussed below.

Finally, the tool optimizes the route terminals by creating straight extrusions that are perpendicular to the part pin pads in the socket. Since the finishing of the conductive regions exposed to the air can easily get over-fused with the printer (see 5), the tool extends the route terminals outside the 3D body and creates excessive regions to protect the top from over-fused. As a result, the generated route terminals can easily touch each other in the socket (Figure 9a), which makes it even more difficult to clean if the socket is extremely small such as that for a 0603 surface-mount device (SMD) packaged resistor. To avoid such issue, the tool first creates a cylindrical extrusion in the inversed normal direction at the pin position. Then, the tool finds an appropriate position that has a minimal distance to both the cylindrical extrusion and the generated route (Figure 9b). Lastly, a pipe is generated through the found position to connect the route and the extrusion. The original route terminal portion are removed (Figure 9c).

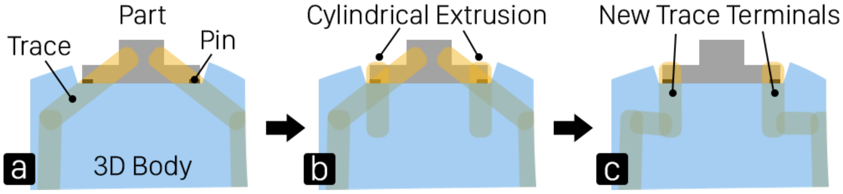


Fig. 9. Routes have straight terminals that are perpendicular to the part pin pads in the socket to avoid trace interference.

(d) *Manual-route.* Beside the auto-generated routes, the user can manually add routes to connect different components. After the user clicks on the *Manual Route* button, the tool prompts to ask the user to specify the position of the start point for the route. For the point position, the tool accepts three types of objects: a part pin, a 3D printable part, and an already generated trace. Only those positions in the selected type of object are highlighted when the user moves around the cursor in the Rhino 3D scene. Upon the confirmation of the start point position, the user needs to specify the end point for the trace in the same manner. A trace is generated using the same algorithm as auto-route (Figure 8c).

(e) *Other route operations.* The user can select a generated route in the interface (Figure 7) and the selected route is highlighted in the Rhino 3D scene. In addition, all the generated routes can be hidden, displayed, or deleted by the user.

4.2.5 Example Walk-through. Here we showcase the entire workflow of using *ModElec* to embed a LED circuit into a 3D rat model (Figure 1): the user (i) creates a custom rat model in Rhino 3D; (ii) imports a LED circuit created with *Fritzing*; (iii) arranges each electronic part in the 3D rat model (the two LEDs are placed as the rat eyes); (iv) adds 3D-printed touchpads to the rat's back for a potential circuit switch; (v) previews the auto-generated routes and manually adds routes for the touchpad-based switch; (vi) modifies the circuit layout or even the model until feels satisfied with the overall 3D design; and finally (vii) sends the saved plastic 3D model, conductive routes, and 3D-printable touchpads to print. The user only needs to attach the electronic components to the printed body and the rat lights up the two eyes when the switch is closed manually.

5 FABRICATION AND PART ASSEMBLY

The converted 3D body, added 3D printable parts, and generated routes are exported from the tool in separate STL files for 3D printing. We use a Multi Jet Fusion technology^{TM} (MJF) based 3D printing testbed at HP Labs with multi-agents/materials capability to print both conductive internal routes and non-conductive body sections of parts [46]. The testbed printer deposits plastic powder layers followed by selectively fusing areas of interests by jetting a proprietary special fusing agent (Figure 10a). The special fusing agent contains materials that absorb the energy radiated from the fusing and overhead warming lamps on the testbed printer. As a result, the part geometry is formed by controlling whether the powder is fused or not. For an object created with *ModElec*, the non-conductive regions were created with the special fusing agent only, and the electrically conductive traces were created with and a conductive agent (CA)—a metal nanoparticle dispersion that has been formulated to work with HP thermal inkjet pens. With our current CA and process control in our testbed printer, the scan axis conductive traces and the build axis conductive traces can achieve resistivities of approximately $0.000001\Omega m$.

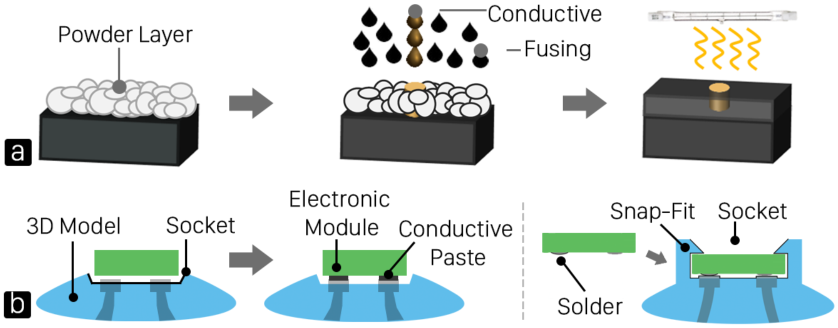


Fig. 10. The created object is printed on a PBF-based testbed printer that follows (a) print processes including dispensing a powder layer, jetting conductive and fusing agent, and fusing the finish. (b) Electronic parts are assembled in the generated sockets.

To assemble the electronic part, we first prepare the conductive terminals on the printed part to insure uniform contact resistance for all contact points. For sockets without the snap-fit designs, we place the electronic part in the socket, align the pins with the exposed pads, and then add sliver conductive adhesive (NAMICS© UNIMEC

H9626D) to strengthen the contact between the pin and the pad. For the other sockets that have snap-fit guides, we first apply solder on the part pin pads and then slide the electronic part into the socket, where the cooled solder raises the part body against the snap-fit guides and thus locks the part firmly in the socket (Figure 10b).

6 APPLICATIONS

To demonstrate the potential of our approach, we built four interactive prototypes with the *ModElec* design tool, including a set of 3D printable input devices, a personalized game controller, a tangible storytelling character, and a reconfigurable handheld device. These applications showcase the integration of a mix of electronic components and plug-and-play module such as microcontroller boards with 3D-printed objects. All the devices were printed on the testbed printer with a limited printing bed size (*i.e.*, 3×10×10cm).

6.1 3D Printable Flippers and Launcher for a Pinball Game

The first application is a set of input controls that includes a pair of flippers and a spring launcher for a Pinball game. In this application, we created two flexure-based flippers that are bendable under finger pressing and a spring-based ball launcher. These deformable controls solely have 3D printable routes and touch areas embedded. For the flippers, two conductive touch points are added to the flipper head and the base individually. One conductive connector in the front is connected to the touch point in the base and two other conductive connectors are attached to the rear of the flipper (Figure 11a). By wiring the front connector and one of the rear connectors with an external Arduino board and an auxiliary circuit (the external circuitry was not integrated in the 3D-printed form due to the limited print bed size), the two flippers work as tactile push buttons (Figure 11b). When the player presses the flipper head, the flipper bends and the two conductive points touch each other, which closes the circuit. For the ball launcher, the exposed area is used as a large touchpad, which converts touch capacitance into the applied force to launch the ball in the game. The detected capacitance increases as the finger keeps touching and pressing the pad on the spring. The Pinball game was built using Processing and communicates with the Arduino board via Serial.

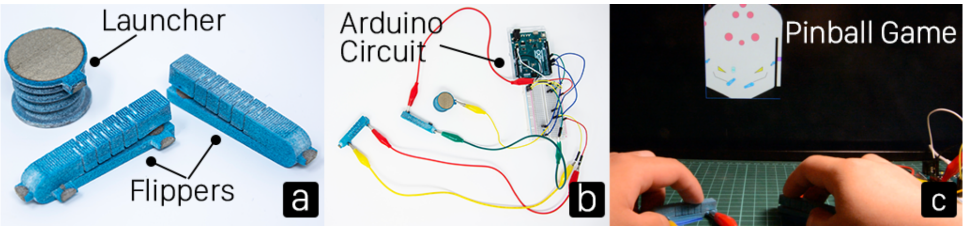


Fig. 11. 3D printable circuits are embedded in (a) two flippers and a spring launcher for controlling (c) a Pinball game.

6.2 Personalized Controller for a Car Racing Game

To demonstrate the ability to use *ModElec* to design custom I/O device, we created a car-like game controller for a Processing car racing game. The controller consists of two parts: a car body that hosts a 6 degree-of-freedom (DOF) module and an Adafruit Itsy Bitsy board (Figure 12a), and a car top that has one LED and two touch pads embedded. The embedded circuit is closed when the top is stacked on the body (Figure 12b). The inertial module can be hidden in the car top (ideally the car is printed in one shot with a bigger printing bed). To play the game, the player holds the controller and leans toward right or left to control the car direction and control the accelerator and the brake by touching the right and left touch pads individually (Figure 12c). The LED on the top blinks to indicate not too much time left for the lap. With *ModElec*, the user can customize their own game controller by integrating complex off-the-shelf sensor modules.

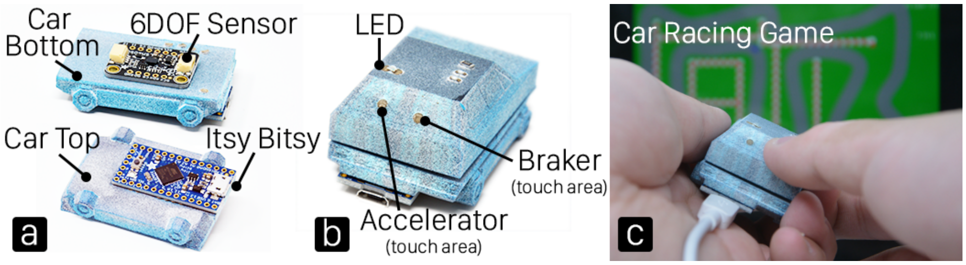


Fig. 12. A car-like game controller has (b) a car top stacked on (a) a car body to control (c) a digital car racing game.

6.3 Turtle with Complex Texture for Storytelling

As the circuit traces can be hidden in the 3D body, we created a spikey shell that is safe to touch on a 3D-printed turtle prop for interactive storytelling. The turtle body hosts an *Adafruit Trinket M0* controller board, a buzzer, and a vibration motor (Figure 13a). Two capacitive touch pads were added on the shell to detect finger touch on the turtle's back, which activates the turtle to tremble and make intermittent sounds to express disturbed emotion (Figure 13b). Another touch pad is placed on the top of the turtle head for the child to record/replay a story narrative (Figure 13c). By long pressing the pad, the storyteller starts speaking and the story narrative is recorded on the tethered computer. The story narrative stops recording when the child releases the touch. When the pad is tapped, the computer replays the recorded story.

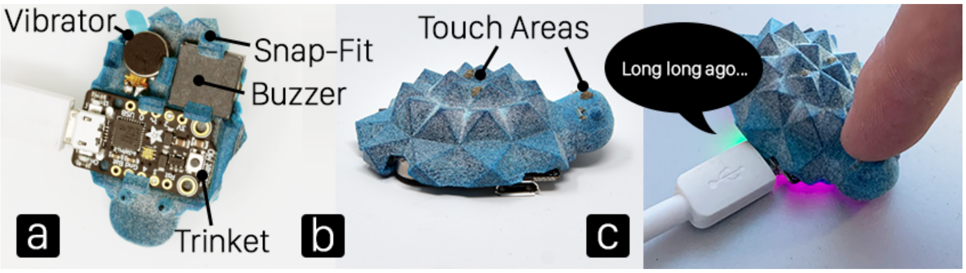


Fig. 13. A turtle with spiky shell texture has (a) a buzzer, a vibration motor, and (b) two touchpads for (c) interactive storytelling.

6.4 Reconfigurable Handheld Flashlight and Fan

Finally, we created a reconfigurable handheld device that can be used as a flashlight and a fan with different heads attached to the same body (Figure 14). To achieve this modular design, we first decomposed a complete circuit into three parts: a controlling body that consists of an *Adafruit Trinket* board and two touchpads, a light head with an embedded LED, and a fan head with a motor and a driver circuit. These two heads share the same controlling circuit embedded in the body but uses different connectors, which are 3D-printed and exposed at the junction place (Figure 14c). When the light head is snapped into the connector, the user can adjust the brightness of the light by touching the two 3D-printed touchpads. When the fan head is attached, the user change the spinning speed of the propellers.

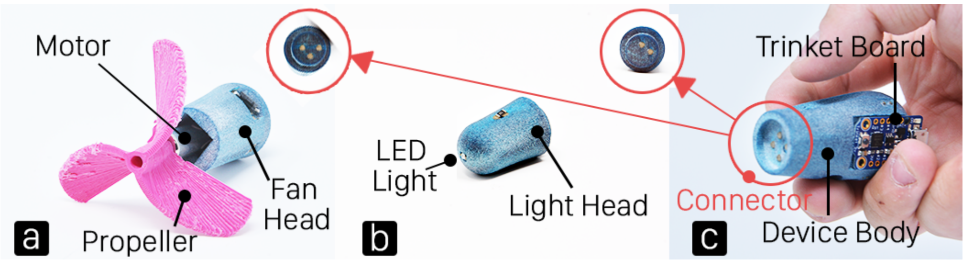


Fig. 14. A handheld device can be reconfigured as (a) a fan or (b) a flashlight using (c) a connector to match different heads.

7 PRELIMINARY STUDY

To examine if designers understand how to prepare 3D circuit in custom 3D objects with *ModElec* for conductive 3D printing and the usability of the tool, we conducted an informal qualitative study.

7.1 Participants

We recruited eight participants (aged 19-33, female=3). Five participants had design background and three had Electrical Computing and Engineering background. All participants had experience using CAD tools and 3D modeling with Rhino. Although seven participants had experience in building physical computing systems and circuit design, three of them identified themselves as beginners in electronics. The other one participant did not prototype circuits but understands circuit schematics.

7.2 Apparatus, Tasks and Procedure

We investigated how participants used *ModElec* to arrange electronic parts and create custom circuit layout in 3D design, rather than building circuitries in third-party EAD tools. Therefore, some basic circuits were provided unless participants needed to edit the circuit. In the study, participants designed 3D models with integrated circuits via Zoom's remote desktop to interact with *ModElec* running on the experimenter's computer. Due to the COVID-19, the 3D printer is limited for access thus we asked the participant to validate their designs in Rhino with the help of the experimenter.

7.2.1 Design Tasks. To start, each participant was introduced the concept of conductive 3D printing and how to use *ModElec* to arrange electronics and routes in a 3D model through the LED circuit-based rat example. Then, the participant was asked to complete two design tasks. In the first task, the participant added the LED circuit into the snake model and created an open switch by adding capacitive touch areas on both the snake's tail and lower jaw. The goal of this task was to examine if the participant understood the *ModElec* design tool. In the second task, the participant was open to create a custom model of their own choice and added either the LED circuit (with the option to add more LEDs) or a motion control-based game controller as a Christmas gift. The goal of this open design task was to investigate if the participant could create both the 3D mechanical design and circuit layout for an interactive prototype with *ModElec*. The two design tasks were videotaped with the participant's consent and the participant had a 10-minute break between the two tasks.

7.2.2 Metrics & Measurements. After both tasks were completed, the participant filled out a questionnaire to answer how difficult to understand the user interface and use the tool, whether the resulting circuit design was expected, and whether the tool was useful for integrating electronics and traces with 3D objects. We evaluated the tool usability using System Usability Scale (SUS) [3]. At the end of the study, we also conducted a brief interview soliciting the participant's feedback on the overall user experience and suggestions for tool improvements.

7.3 Results

Overall, all participants were able to complete both two design tasks without assistance and mostly thought *ModElec* as a convenient and useful tool for arranging electronic parts during the 3D modeling process. By interpreting the scores of the rated items by SUS, we got average 74.06 (*SD*=4.50) for the rating, which shows that our tool has acceptable usability [3], but improvements can be made. Figure 15 shows all the creations during the study and the time distribution on 3D modeling, electronic arrangement, and routing in the two tasks. Two participants chose to create a controller and the other participants stayed with the simple LED circuit. Overall, participants spent average 501.6s (*SD*=80.1s) for task 1 and 1293.1s (*SD*=373.2s) for task 2. At least six participants iterated their electronic arrangement and routing more than once in both task 1 and 2.

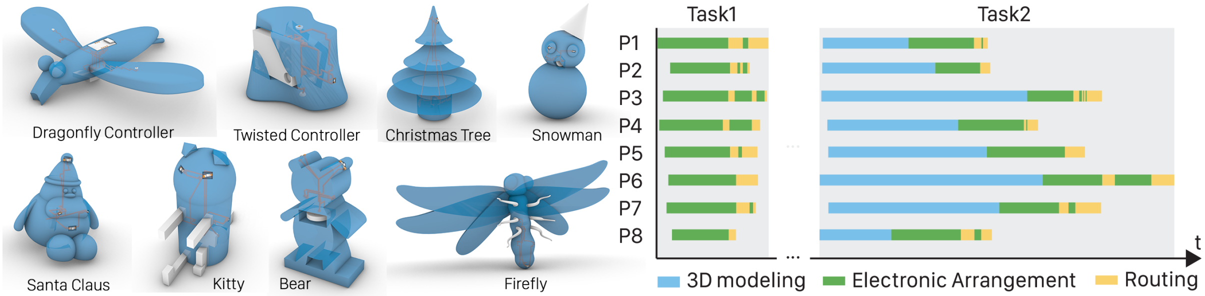


Fig. 15. All the creations in the study and observed time distributions on different operations in task 1 and 2.

7.4 Findings

We summarize the findings that we observed in the study.

7.4.1 Electronic Part Control. The currently provided set of part control operations in the design tool are sufficient for electronic part arrangement in a 3D form and the workflow mimics the physical prototyping course in the real world. All the participants enjoyed the straightforwardness of the electronic part control panel and agreed that the part selection, moving, rotation, flip, and protrusion are easy to understand and use. The only confusion about the in-place part rotation occurred since the participants had a hard time associating the part rotation with the post-print part assembly when they arranged each electronic part. This indicates that assembly-aware controls built in such a design tool may help the user create a valid application. Amongst the part control operations, the *Quick-Positioning* feature was praised and one of the most used functions in the study. Participants thought the Quick-Positioning as natural as the physical operation in the real world: *"[Quick-Positioning] is awesome and this feels like I am putting the LED directly in the body although this is virtual. I don't need to drag around the part to find a perfect position"* (P4). While coarse positioning support like the *Quick-Positioning* is useful, fine tuning on the part such as the *Part-Nudging* and part protrusion were also considered helpful for the aesthetics of the 3D design: *"I want to align these two LEDs on the Santa Claus's mustache" (P5). "I want this [LED] to be at the eye position and come out like a real eyeball"* (P2).

7.4.2 Circuit Trace Control. Knowing that the circuit routes are 3D-printable and no need to manually wire electronic components in the 3D object, participants felt very positive about the auto-routing as part of the design process: *"I don't need to connect [parts] myself... I will never need to worry about the wires"* (P3) *"This is super convenient...I can imagine when the circuit is getting more complex [auto-routing] will save me a lot of work"* (P5). However, an intelligent mechanism that automatically adjusts the placement and orientation of electronics for successful routing was wanted. Although the tool provides the *Connection Preview* for the user to inspect the places that may cause problematic routing, there is a learning cost for the user to manually fix all the issues. Finally, participants had difficulties understanding how manual traces would affect the circuit: *"...I see you need to remove that trace first and connect [the two touch areas] with the LED and the battery... Can I manually wire these components?"* (P1). The tool may eliminate such confusion by predicting the affected traces when the user edits the routes and providing visual assistance for managing both auto-generated and manually added routes.

7.4.3 Iterative Design in the New Integrated Workflow. We observed that designers were still used to following a "3D-modeling-and-then-circuit-modeling" method with the tool. In the second task, participants spent a lot of time creating and polishing the custom 3D models in the first place and then transited to electronic deployment. Participants rarely modified the model after the model was created. This is echoed in the post-study interview by participants: *"I spent a lot of time thinking about what I should make and before I got something that I felt happy about I wouldn't start adding the electronics"* (P6) *"I am not sure if I will make [the circuit] worse if I make any changes [to the model]"* (P1). However, all the participants liked to modify the electrical layout several times throughout the design process due to the low cost of changing the routing: *"It's really interesting to see how those traces change... can't imagine this is happening automatically"* (P4). These emerging patterns provide good lessons learned for adding more helpful features to CAD tools that support such an integrated 3D modeling and circuit design workflow. For example, a history track of 3D mode edits and circuit changes can potentially make the designer feel more comfortable about switching between these two traditionally isolated workflows.

8 DISCUSSION AND FUTURE WORK

While *ModElec* is validated as easy to use, we enumerate limitations and the next steps for this work.

8.1 Constraints, Optimizations, and Trade-offs for Routing

A main benefit of *ModElec* is to automatically generate traces for the inserted electronics. However, route generation may fail due to three geometric constraints. First, it is hard to generate routes in an extreme 3D shape that does not have sufficient room (*e.g.*, a porous object or a slender body). Second, the route generation will be problematic when two interconnected electronic parts are placed too close to each other or on the same surface. Finally, routing may fail when the host 3D form is an enclosure rather than a solid object. The design tool can potentially recognize those geometries that cannot afford insufficient room for routing and detect where electronic parts are too close in plane. As suggested in the study, electronic parts can self-adjust its placement and orientation for a more approachable circuit layout. We will implement these mechanisms as part of the future work to address the first two problems. For adding traces to thin-shell based enclosures, we plan to improve the tool by enabling trace routing on poly-surfaces where connecting electronic parts are attached to the interior surface of an enclosure or exposed on the exterior surface for aesthetics [40].

Another area to improve is to optimize the auto-routing algorithm by minimizing the overall impedance of wiring for a functional circuit design. Currently, the tool only considers the shortest path for individual two-pin pairs and prioritizes those pins that are geometrically closer to each other. However, the wiring length between two target parts that are not directly connected may not always be the shortest, which leads to more energy loss and is critical in power-sensitive circuits such as an antenna. For the next step, we will implement intelligent algorithms to guarantee the working current and power for each electronic part in the circuit. For example, similar to [39], the resistance of conductive traces can be automatically tuned as the circuit topology changes in the 3D model. Additionally, electronic part positions can also self-adjust towards an optimal topology to balance the current distribution.

8.2 Compatibility with Electronic Part Assembly and Circuit Modifications

Another limitation of our approach is that all the external electronic components can only be attached to the surface of the 3D-printed object. This can be addressed by using a pick-and-place manufacturing method that selectively picks electronic parts and places them in the 3D body during the layer-wised printing process. To achieve the printable design that can be used in such a pick-and-place approach, we will improve the design tool by allowing the designer to select and freely place non-interactive electronics such as resistors and diodes within the 3D body. Those interactive modules such as sensors, actuators, displays are still installed on the surface.

It is easy to position SMD parts on the pads since these parts have flat pin base and larger pad contact. Compared with SMD components, through-hole devices are error-prone for installation due to the insecure contact between the device pin and the 3D-printed conductive pad. From a comparison study on two LED circuit-based designs (the snake and the rat model in Figure 16), we find that through-hole components need more strength to penetrate the pin into the conductive pad. Drilling a hole in the pad before inserting the part pin can greatly increase the contact but is very likely to seal more air between the pin and the conductive pad, which reduces the conduction. Therefore, SMD components are more recommended for *ModElec*. Commodity electronic modules without soldered pins such as sensor breakouts are compatible to use since the vias on the module board can serve as SMD pads, plus reinforced by the snap-fit guides.

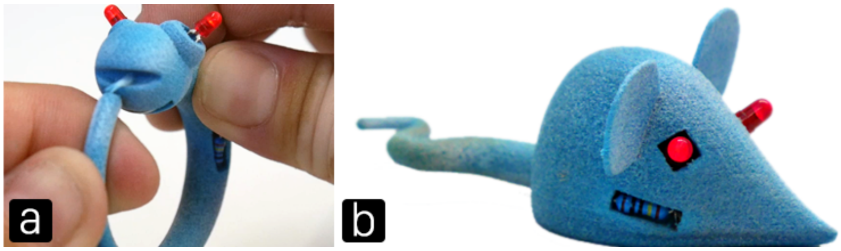


Fig. 16. Two LED circuit-based applications with through-hole devices: (a) a snake and (b) a rat.

While the design tool allows the user to quickly deploy a 3D circuit design in a 3D model, debugging the embedded circuit and fixing circuit errors are not supported in the current prototype. We envision that designer can quickly fix circuit errors such as circuit breaks or short circuits by re-arranging electronic parts and wiring and re-printing the updated model, since both the 3D model and circuit design pipelines are incorporated and thus accelerated with *ModElec*. Another alternative solution to this problem is to use on-the-fly editing and printing techniques like those in [25] that allow the user to modify the circuit design while the model is being printed on the 3D printer. We will investigate these strategies to support circuit editing and debugging in the integrated workflow with *ModElec*.

8.3 Support for 3D-Printable Electronics and Various Printing Processes

3D-printed electronics offers new capabilities to enable novel interactive applications for 3D printing and gives rise to new development of design tools to support the making with 3D-printable electronics. While *ModElec* introduces 3D printable parts as an add-on to the circuit design in 3D objects, the support for the design and control of 3D printable electronics is still limited. To investigate how to incorporate a wider library of 3D-printable electronics in the design tool, we will explore more 3D-printable passives and sensors, such as inductors, capacitors, LC-tank, or even antennas, and exploit the tunable parameter space of those 3D-printable electronics.

ModElec aims to enable the integration of circuitries into 3D models that can be printable with all kinds of conductive 3D printing processes, including the testbed printer used in our work and other mature conductivity fabrication machines and methods such as Fuji's FPM-Trinity system[10] and laser direct structuring (LDS)-based manufacturing[11]. Ideally, all the machine/printing process-related specifications are manually defined by the designer or engineer in a JSON-format print profile file and *ModElec* design tool operates accordingly by parsing the information logged in the print profile. Like exiting common 3D printing process, the created model (usually a mesh) needs to be carefully and purposely converted into *gcode* lines using a slicer software and then printed on a designated 3D printer. For a successful print with *ModElec*, it is essential to understand the desired printing process and prepare the print profile.

9 CONCLUSION

In this paper, we present *ModElec*, an interactive design tool that examines how circuits and 3D models can be coordinated to support new fabrication paradigms where conductive 3D printing is used for physical prototyping. With *ModElec*, designers can arrange electronic parts in a custom 3D design, preview auto-generated traces and edit manual traces, and iterate model design and circuit layout for interactive applications. We detail the workflow and user interface of the tool, show the potential of our approach with a set of example applications created with the tool, and validate that the tool is easy to use through a preliminary study.

ACKNOWLEDGMENTS

This work was supported and funded by HP Inc., specifically by 3D Lab at HP Labs. We would like to thank the extensive collaboration within 3D Lab. We would also like to thank Eric Luna-Ramirez for his help with 3D printing and all the participants for their time and valuable feedback.

REFERENCES

[1] Rafael Ballagas, Sarthak Ghosh, and James Landay. 2018. The Design Space of 3D Printable Interactivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol* 2, 2 (2018), 1–21. https://doi.org/10.1145/3214264

[2] Patrick Baudisch and Stefanie Mueller. 2017. Personal fabrication. *Foundations and Trends in Human-Computer Interaction* 10, 3-4 (2017), 165–293. https://doi.org/10.1561/1100000055

[3] John Brooke. [n.d.]. *System Usability Scale (SUS)*. Retrieved October 22, 2021 from https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

[4] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J. Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K. Kane. 2015. Sharing is Caring: Assistive Technology Designs on Thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 525–534. https://doi.org/10.1145/2702123.2702525

[5] Ruta Desai, James McCann, and Stelian Coros. 2018. Assembly-aware design of printable electromechanical devices. *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (2018), 457–472. https://doi.org/10.1145/3242587.3242655

[6] David Espalin, Danny W. Muse, Eric MacDonald, and Ryan B. Wicker. 2014. 3D Printing multifunctionality: Structures with electronics. *International Journal of Advanced Manufacturing Technology* 72, 5-8 (2014), 963–978. https://doi.org/10.1007/s00170-014-5717-7

[7] Wei Gao, Yunbo Zhang, Diogo C. Nazzetta, Karthik Ramani, and Raymond J. Cipra. 2015. RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (2015), 437–446. https://doi.org/10.1145/2807442.2807476

[8] Saul Greenberg and Chester Fitchett. 2001. Phidgets: Easy development of physical interfaces through physical widgets. *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* 3, 2 (2001), 209–218.

[9] Daniel Groeger, Martin Feick, Anusha Withana, and Jürgen Steimle. 2019. Tactlets: Adding tactile feedback to 3D objects using custom printed controls. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (2019), 923–936. https://doi.org/10.1145/3332165.3347937

[10] Bjorn Hartmann and Paul K. Wright. 2013. Designing Bespoke Interactive Devices. *Computer* 46, 8 (Aug. 2013), 85–89. https://doi.org/10.1109/MC.2013.274

[10] Fuji FPM-Trinity System: https://www.fuji.co.jp/en/about/fpm-trinity/

[11] Laser Technology for Mechatronic Interconnect Devices (MIDs): https://www.lpkf.com/en/industries-technologies/electronics-manufacturing/3d-mids-with-laser-direct-structuring-lds

[11] Liang He, Gierad Laput, Eric Brockmeyer, and Jon E. Froehlich. 2017. SqueezaPulse: Adding interactive input to fabricated objects using corrugated tubes and air pulses. *Proceedings of the 11th ACM International Conference on Tangible, Embedded, and Embodied Interaction, TEI 2017* (2017), 341–350. https://doi.org/10.1145/3024969.3024976

[12] Freddie Hong, Connor Myant, and David E Boyle. 2021. Thermoformed Circuit Boards: Fabrication of highly conductive freeform 3D printed circuit boards with heat bending. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–10. https://doi.org/10.1145/3411764.3445469

[13] Yoshio Ishiguro and Ivan Poupyrev. 2014. 3D printed interactive speakers. *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems* (2014), 1733–1742. https://doi.org/10.1145/2556288.2557046

[14] Michael D. Jones, Zann Anderson, Casey Walker, and Kevin Seppi. 2018. PHUI-kit: Interface layout and fabrication on curved 3D printed objects. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* 2018-April (2018), 1–11. https://doi.org/10.1145/3173574.3173684

[15] Michael D. Jones, Kevin Seppi, and Dan R. Olsen. 2016. What you sculpt is what you get: Modeling physical interactive devices with clay and 3D printed widgets. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), 876–886. https://doi.org/10.1145/2858036.2858493

[16] Gierad Laput, Eric Brockmeyer, Moshe Mahler, Scott E. Hudson, and Chris Harrison. 2015. Acoustruments: Passive, acoustically-driven, interactive controls for handheld devices. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015). https://doi.org/10.1145/2782782.2792490

[17] David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. 2017. Pineal: Bringing passive objects to life with embedded mobile devices. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), 71–71. https://doi.org/10.2307/j.ctt1ffjdz1.47

[18] Dingzeyu Li, David I.W. Levin, Wojciech Matusik, and Changxi Zheng. 2016. Acoustic voxels: Computational optimization of modular acoustic filters. *ACM Transactions on Graphics* 35, 4 (2016). https://doi.org/10.1145/2897824.2925960

[19] Jiahao Li, Meilin Cui, Jeeeun Kim, and Xiang Anthony Chen. 2020. Romeo: a design tool for embedding transformable parts in 3d models to robotically augment default functionality. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (2020), 897–911. https://doi.org/10.1145/3379337.3415826

[20] Jiahao Li, Jeeeun Kim, and Xiang 'Anthony' Chen. 2019. Robiot: A Design Tool for Actuating Everyday Objects with Automatically Generated 3D Printable Mechanisms. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (2019), 673–685. https://doi.org/10.1145/3332165.3347894

[21] Eric MacDonald, Rudy Salas, David Espalin, Mireya Perez, Efrain Aguilera, Dan Muse, and Ryan B. Wicker. 2014. 3D printing for the rapid prototyping of structural electronics. *IEEE Access* 2 (2014), 234–242. https://doi.org/10.1109/ACCESS.2014.2311810 Publisher: IEEE.

[22] Kathryn Mcelroy and Kathryn Mcelroy. 2016. *Prototyping for designers: Developing the best digital and physical products.* O'Reily Media.

[23] Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. 2017. A computational design tool for compliant mechanisms. *ACM Transactions on Graphics* 36, 4 (2017). https://doi.org/10.1145/3072959.3073636

[24] Huaishu Peng, François Guimbretière, James McCann, and Scott E. Hudson. 2016. A 3D printer for interactive electromagnetic devices. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (2016), 553–562. https://doi.org/10.1145/2984511.2984523

[25] Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 887–896. https://doi.org/10.1145/2858036.2858106

[26] Giovanni Postiglione, Gabriele Natale, Gianmarco Griffini, Marinella Levi, and Stefano Turri. 2015. Conductive 3D microstructures by direct 3D printing of polymer/carbon nanotube nanocomposites via liquid deposition modeling. *Composites Part A: Applied Science and Manufacturing* 76 (2015), 110–114. https://doi.org/10.1016/j.compositesa.2015.05.014 Publisher: Elsevier Ltd.

[27] Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2016. RetroFab: A Design Tool for Retrofitting Physical Interfaces Using Actuators, Sensors and 3D Printing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), 409–419. https://doi.org/10.1145/2858036.2858485

[28] Steven Ready, Fred Endicott, Gregory L. Whiting, Tse A. Ng, Eugene M. Chow, and Jeng Ping Lu. 2013. 3D printed electronics. *International Conference on Digital Printing Technologies* (2013), 9–12.

[29] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design Principles for Tools to Support Creative Thinking. (2005), 16.

[30] Claudia Daudén Roquet, Jeeeun Kim, and Tom Yeh. 2016. 3D folded PrintGami: Transform passive 3D printed objects to interactive by inserted paper Origami circuits. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (2016), 187–191. https://doi.org/10.1145/2901790.2901891

[31] Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (2013), 447–456. https://doi.org/10.1145/

2501988.2501992

[32] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' marks: Physical markup for designing and fabricating functional objects. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (2015), 103–108. https://doi.org/10.1145/2807442.2807508

[33] Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. 2014. A Series of Tubes: Adding Interactivity to 3D Prints Using Internal Pipes. *Proceedings of the 27th annual ACM symposium on User interface software and technology* (2014). https://doi.org/10.1145/2642918.2647374

[34] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhaüser, and Jühlhürgen Steimle. 2015. Capricate: A fabrication pipeline to design and 3D printcapacitive touch sensors for interactive objects. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (2015), 253–258. https://doi.org/10.1145/2807442.2807503

[35] Martin Schmitz, Urgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Uhlhäuser. 2017. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), 1001–1014. https://doi.org/10.1145/3025453.3025663

[36] Martin Schmitz, Martin Stitz, Florian Müller, Markus Funk, and Max Mühlhäuser. 2019. ./Trilaterate: A Fabrication Pipeline to Design and 3D Print Hover-, Touch-, and Force-Sensitive Objects. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), 1–13. https://doi.org/10.1145/3290605.3300684

[37] Ronit Slyper and Jessica Hodgins. 2012. Prototyping robot appearance, movement, and interactions using flexible 3D printing and air pressure sensors. *2012 IEEE RO-MAN: The 21st IEEE International Workshop on Robot and Human Interactive Communication* 1 (2012), 6–11. https://doi.org/10.1109/ROMAN.2012.6343723

[38] Saiganesh Swaminathan, Kadri Bugra Ozutemiz, Carmel Majidi, and Scott E. Hudson. 2019. FiberWire: Embedding Electronic Function into 3D Printed Mechanically Strong, Lightweight Carbon Fiber Composite Objects. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* Figure 1 (2019), 1–11. https://doi.org/10.1145/3290605.3300797

[39] Tung D. Ta, Fuminori Okuya, and Yoshihiro Kawahara. 2017. LightTrace: auto-router for designing LED based applications with conductive inkjet printing. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*. ACM, Cambridge Massachusetts, 1–10. https://doi.org/10.1145/3083157.3083160

[40] Nobuyuki Umetani and Ryan Schmidt. 2017. SurfCuit: Surface-Mounted Circuits on 3D Prints. *IEEE Computer Graphics and Applications* 38, 3 (2017), 52–60. https://doi.org/10.1109/MCG.2017.40

[41] Voxel8. [n.d.]. *Voxel8 3D printer.* Retrieved October 22, 2021 from https://www.voxel8.com

[42] Marynel Vázquez, Eric Brockmeyer, Ruta Desai, Chris Harrison, and Scott E. Hudson. 2015. 3D Printing Pneumatic Device Controls with Variable Activation Force Capabilities. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015), 1295–1304. https://doi.org/10.1145/2702123.2702569

[43] Guanyun Wang, Fang Qin, Haolin Liu, Ye Tao, Yang Zhang, Yongjie Jessica Zhang, and Lining Yao. 2020. Morphingcircuit: An integrated design, simulation, and fabrication workflow for self-morphing electronics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020). https://doi.org/10.1145/3432232

[44] Tianyi Wang, Ke Huo, Pratik Chawla, Guiming Chen, Siddharth Banerjee, and Karthik Ramani. 2018. Plain2Fun: Augmenting ordinary objects with interactive functions by auto-fabricating surface painted circuits. *Proceedings of the 2018 Designing Interactive Systems Conference* June (2018), 1095–1106. https://doi.org/10.1145/3196709.3196791

[45] Karl D.D. Willis, Eric Brockmeyer, Scott E. Hudson, and Ivan Poupyrev. 2012. Printed Optics: 3D printing of embedded optical elements for interactive devices. *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (2012), 589–598.

[46] Jarrid Wittkopf, Kris Erickson, Paul Olumbummo, Aja Hartman, Howard Tom, and Lihua Zhao. 2019. 3D Printed Electronics with Multi Jet Fusion. Society for Imaging Science and Technology. *NIP & Digital Fabrication Conference 2019* 1 (2019), 29–33. https://doi.org/10.2352/ISSN.2169-4451.2019.35.29

[47] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K. Nayar. 2019. Vidgets: Modular mechanical widgets for mobile devices. *ACM Transactions on Graphics* 38, 4 (2019). https://doi.org/10.1145/3306346.3322943

[48] Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2018. Bend-it: Design and Fabrication of Kinetic Wire Characters. *ACM Transactions on Graphics* 37, 6 (2018), 1–15. https://doi.org/10.1145/3272127.3275089

[49] Junichi Yamaoka, Mustafa Doga Dogan, Katarina Bulovic, Kazuya Saito, Yoshihiro Kawahara, Yasuaki Kakehi, and Stefanie Mueller. 2019. FoldTronics: Creating 3D objects with integrated electronics using foldable honeycomb structures. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019). https://doi.org/10.1145/3290605.3300858

[50] Junyi Zhu, Lotta-Gili Blumberg, Yunyi Zhu, Martin Nisser, Ethan Levi Carlson, Xin Wen, Kevin Shum, Jessica Ayeley Quaye, and Stefanie Mueller. 2020. CurveBoards: Integrating Breadboards into Physical Objects to Prototype Function in the Context of Form. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), 1–13. https://doi.org/10.1145/3313831.3376617

[51] Junyi Zhu, Yunyi Zhu, Jiaming Cui, Leon Cheng, Jackson Snowden, Mark Chounlakone, Michael Wessely, and Stefanie Mueller. 2020. MorphSensor: A 3D electronic design tool for reforming sensor modules. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (2020), 541–553. https://doi.org/10.1145/3379337.3415898