AirLogic: Embedding Pneumatic Computation and I/O in 3D Models to Fabricate Electronics-Free Interactive Objects

Valkyrie Savage*
Department of Computer Science
University of Copenhagen
vasa@di.ku.dk

Carlos E. Tejada*
Department of Computer Science
University of Copenhagen
ct@di.ku.dk

Mengyu Zhong
University of Uppsala
mengyu.zhong@it.uu.se

Raf Ramakers
Hasselt University - Flanders Make
Expertise Centre for Digital Media
raf.ramakers@uhasselt.be

Daniel Ashbrook
Department of Computer Science
University of Copenhagen
dan@di.ku.dk

Hyunyoung Kim
School of Computer Science
University of Birmingham
h.kim.4@bham.ac.uk

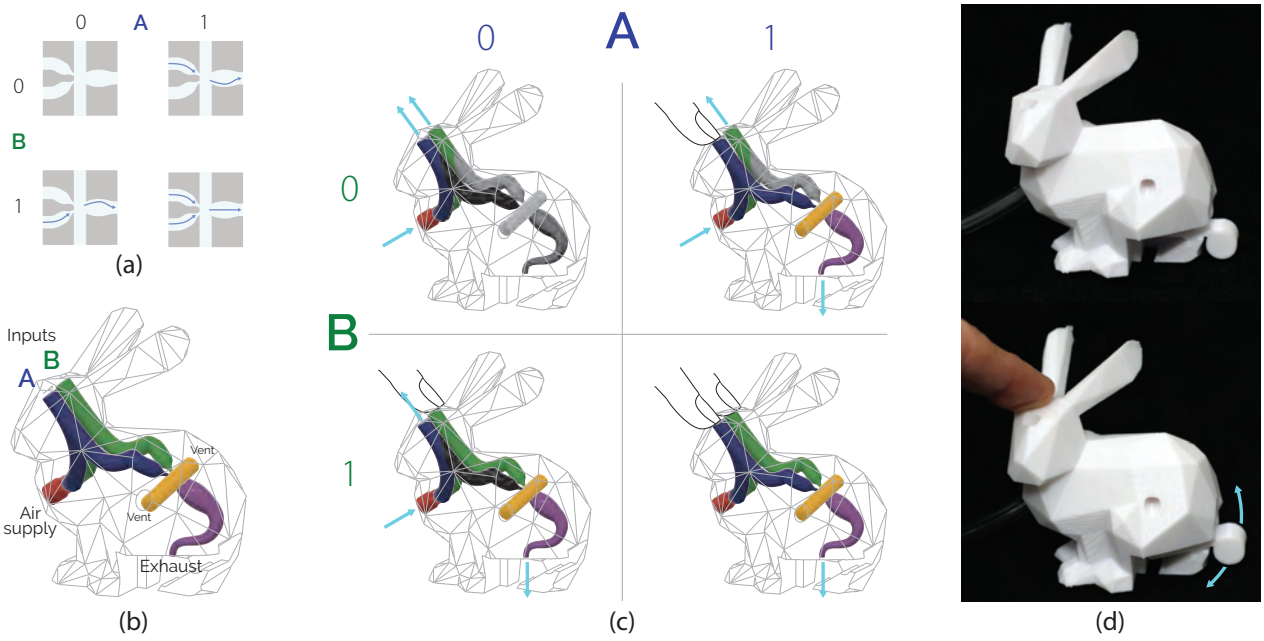

Figure 1: AirLogic enables 3D printing interactive objects that are powered by airflow. We integrate printed logical structures like OR gates (a) alongside tube-based inputs and outputs in 3D models (b). These route airflow through the device's interior based on the results of logical operations performed on the user's input (c). Our widgets enable creating fully-printed, stand-alone interactive objects with integrated sensing, computation, and actuation powered solely by air (d).

ABSTRACT

Researchers have developed various tools and techniques towards the vision of on-demand fabrication of custom, interactive devices. Recent work has 3D-printed artefacts like speakers, electromagnetic actuators, and hydraulic robots. However, these are non-trivial to instantiate as they require post-fabrication mechanical– or electronic assembly. We introduce AirLogic: a technique to create electronics-free, interactive objects by embedding pneumatic input, logic processing, and output widgets in 3D-printable models. AirLogic devices can perform basic computation on user inputs and create visible, audible, or haptic feedback; yet they do not require electronic circuits, physical assembly, or resetting between uses. Our library of 13 exemplar widgets can embed AirLogic-style computational capabilities in existing 3D models. We evaluate our widgets' performance—quantifying the loss of airflow (1) in each widget type, (2) based on printing orientation, and (3) from internal object geometry. Finally, we present five applications that illustrate AirLogic's potential.

*Authors contributed equally

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

UIST '22, October 29-November 2, 2022, Bend, OR, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9320-1/22/10.
https://doi.org/10.1145/3526113.3545642

CCS CONCEPTS

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

KEYWORDS

3D-printing, fluerics, fluidics, logic gates, pneumatic interfaces

ACM Reference Format:
Valkyrie Savage, Carlos E. Tejada, Mengyu Zhong, Raf Ramakers, Daniel Ashbrook, and Hyunyoung Kim. 2022. AirLogic: Embedding Pneumatic Computation and I/O in 3D Models to Fabricate Electronics-Free Interactive Objects. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3526113.3545642

1 INTRODUCTION

Recently, digital fabrication research in HCI has broadened its focus from fabricating 3D shapes to creating functional, interactive objects, like speakers [23], electromagnetic actuators [39], and hydraulic robots [33]. While such objects offer useful functionality, the fabrication process is often laborious—requiring end users to modify object geometry [32], assemble circuits [37], or manually insert non-printable materials [19]. We envision a future where such devices are instantly useful: without intervention during printing, post-print assembly, or training machine learning models.

As a step towards this vision, this paper presents AirLogic, a novel technique to fabricate interactive 3D-printed devices that encapsulate input, logic, and output as integral parts of the printed structure. These objects are immediately usable once they are printed and attached to a pressurized air source. AirLogic accomplishes this by updating classic work in *fluerics*[1] [8], a nearly forgotten area of research that uses jets of air to perform computation without electricity or moving parts. While flueric technology was actively developed in the 1960s through the 1980s, it became largely obsolete with the advent of smaller, cheaper, and faster transistors. In this paper, we show how advances in additive manufacturing enable current generations of off-the-shelf fused-deposition modeling (FDM) printers to produce pneumatic input, output, and logic structures. In contrast to approaches requiring embedding non-printable material into 3D prints, AirLogic's flueric transits are realized as 3D-printed voids in the structure of the object itself, enabling us to create fully-printed objects that accept user input, perform simple calculations based thereon, and respond with output. These objects can also have nearly arbitrary exterior shapes largely decoupled from their computational requirements. This gives our computing substrate many desirable properties: it is fully producible—both shape and function—on a single machine in a single process; it is capable of sensing, processing, and output; and objects leveraging it do not require resetting between uses (they are stateless). A basic printed interactive device, shown in Figure 1, takes user touch input, calculates whether input A `OR` input B has been activated, and actuates an oscillator based on that calculation.

This work contributes to the existing track of research on embedding functionality in objects during the fabrication process in order to facilitate and speed-up prototyping interactive devices [22, 39, 60]. AirLogic advances our vision of Print-and-Play Fabrication [55]: a future where tangible devices are printed on commodity hardware instead of assembled. In summary, we contribute:

(1) A set of 13 chainable pneumatic widgets—logic gates, inputs, and outputs—that can be fabricated with consumer-grade FDM 3D-printers.
(2) A widget library for a consumer CAD tool that makes the widgets available to users for designing AirLogic devices.
(3) A characterization of our widgets' performance, as well as a set of example devices illustrating AirLogic's potential.

2 RELATED WORK

2.1 Physical User Interface Toolkits

The pneumatic widgets developed as part of AirLogic contribute to the field of physical user interface toolkits. Early efforts in this area adopted a *prefabricated* approach, where the toolkit components are fabricated by a third-party, and end users simply assemble them. One of the first such toolkits in the HCI literature is Phidgets [17]. Its authors applied concepts from Graphical User Interface (GUI) widgets to construct physical interaction controls with reusable components. Further refinement of this concept introduced connections between the components [4], novel form-factors [21], or more powerful components [63]. While prefabricating different components of the toolkit reduces design and assembly work for end users, they are then limited to manufacturers' designs.

Later efforts assist designers in constructing *custom* widgets. Midas [47], Pineal [32], and PaperPulse [42] enable building interactive devices with custom touch sensors wrapped around existing objects, "remote widgets" on smartphones and watches, or predefined widgets made with conductive inkjet printing. These approaches' main advantage is that designers can customize widgets as needed.

AirLogic draws inspiration from both types of physical user interface toolkits: we provide a set of predefined input, logic, and output widgets to embed in existing 3D models and fabricate using commodity 3D printers. These widgets can also be customized during the design stage to suit the application.

2.2 Fabricating Interactive Objects

A growing body of work has explored different techniques for digitally fabricating interactive and functional objects. Ballagas et al. present a comprehensive overview of this design space, grouping previous efforts by the interaction mechanism used to enable *interactivity* [2]. In contrast, this section aims to highlight how previous works handle the *computation* requirements of their approaches.

To date, most work on interactive fabricated objects relies on external computation—that is, while the structure of the fabricated object is instrumental in enabling the interaction, the computing resources involved are usually not part of the object itself.

Many systems enable input via fabricated structures that, in concert with user input, create some kind of event that can be detected by a sensor. Some systems use the structure of the printed object as a passive transducer, transforming energy from user interaction into another form, such as sound [19, 45, 50, 56] or movement [43], which a computer then senses and acts upon. Other systems use

[1]Historically also called *fluidics* or *fluid logic*; we use *fluerics* to avoid confusion with microfluidics.

an active sensing approach where user interaction *modifies* an externally supplied signal, which a computer then senses and acts upon. Examples include those using acoustics [26, 31], pneumatics [38, 57, 62], optics [66], and electronics [48, 49].

Systems providing output via digitally-fabricated objects use similar approaches, relying on external computation and actuation to induce user-perceivable changes in the object that depend on the object's structure. Examples include using electricity to produce resistive heating [18] or electrotactile haptics [9], light for programmable appearance [24, 66], hydraulics for motion [33], or pneumatics for haptics [62] or shape change [28].

While the these objects have low computation requirements, and the requisite hardware could theoretically be miniaturized and embedded in the print [37], doing so requires the object designer to understand and engineer such embedding, and it requires post-print assembly. In contrast, AirLogic devices are printed as a single structure with minimal assembly, and they can capture input, perform simple computations, and display output—all due to the interaction of airflow with the object's internal geometry.

2.3 Non-electrical computing systems

AirLogic draws inspiration from the long history of non-electrical computation. The earliest computing devices, developed before the advent of electrical circuits, were mechanical: the earliest known computer, the Antikythera Mechanism (ca. 250–100 BCE) [10], was based on a complex system of gears, as was Babbage's later proposal for an Analytical Engine (1837) [7]. Liquids were also used for pre-electronic computation of complex algebraic [11]– and differential [34] equations, and to visualise the "flow" of money [3].

Despite the modern dominance of electronic computers, researchers continue to explore alternative computing substrates to overcome the limitations of electrical circuitry. Thorsen et al. developed microfluidic processors [58] with applications in biology and chemistry. However, fabricating them requires complex industrial processes, and—due to their "micro" nature—they operate at pressures and flow rates too low for actuating interactive devices. Aiming at robots composed entirely of soft components, Preston et al. created flexible pneumatic logic circuits based on kinking soft, embedded tubes [41]. While they demonstrate `AND`, `OR`, and `NOT` operations, fabricating the gates requires a complex manual molding process, and incorporating them in interactive objects requires complex assembly. On the other hand, AirLogic creates interaction-capable channels in a single print on commodity machines.

Recently, Ion et al. demonstrated fully functional 3D-printed interactive digital devices comprised of metamaterial-based logic cells [22]. Although theoretically capable of extending logical operations through any number of gates due to per-gate energy storage via a buckling mechanism, these devices must be manually reset after each use to recover the lost energy; AirLogic objects are stateless and do not require such resets.

2.4 Fluerics

Before modern interest in digital fabrication, a technique called *fluerics* enabled the manufacture and deployment of air-powered sensors, actuators, logic gates, and control systems [8].

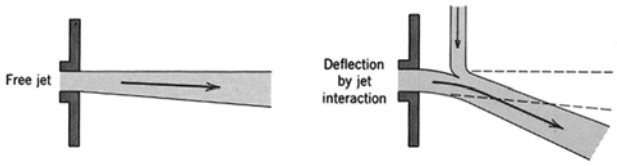

Figure 2: AirLogic works on the principle of momentum transfer between streams of fluid (specifically, air). There are multiple ways to deflect air jets; our technique relies on deflection by jet interaction. (Figure adapted from [6].)

The basic principle underlying fluerics is simple: a constant stream of fluid moving in one direction can be deflected by the momentum of a second, less powerful, stream applied at an angle to the first, by an amount proportional to the strength of the deflecting jet [8, p. 64] (Figure 2). By creating specific geometrical arrangements of channels that shape and direct the fluid streams, a multitude of operational elements can be created [16] (Figure 4).

Starting with the public disclosure of the fluid amplifier in 1960 [29, 54], fluerics was an active research area for nearly three decades, with widespread commercial application [1] throughout the Cold War era while concerns over power interruptions and radiation drove interest in non-electronic computing. However, the field was eventually eclipsed by the development of reliable, high-speed, integrated electronics. This timing means that the majority of work in the space occurred before the popularization of interactive computing in the 80s (e.g., [13]), limiting flueric interaction elements to simple buttons [1, p. 240] and one-bit displays [1, p.698]. Today, the remaining research in fluerics largely concentrates on its potential for aerospace [12] or industrial [30] applications, leaving the potential for fluerics-based interactive devices largely unexplored. One modern work integrates fluerics and electronic computation, but logical operations are performed by the electronics [15]. Another work uses deformable materials with microfluidics for interaction [35], but does not provide computation. For soft robotics, researchers have explored microfluidic logic structures [36, 65], fabricated with specialized machines; in contrast to AirLogic their structures are complex to manufacture and rely on fluid pressure and check valves rather than jet interaction for calculation.

To make them as small and efficient as possible, flueric logic devices were historically produced via chemical etching or machining, achieving channel sizes as small as 0.5 mm [54]. Due to their intricate manual operation and inaccessible manufacturing equipment, these processes are out of reach for hobbyists, makers, and non-experts. AirLogic starts with basic flueric concepts and structures, and updates and extends them to enable production on consumer-grade 3D printers. 3D printing allows a high level of integration: incorporating flueric inputs, outputs, and logic gates directly into an object's structure.

3 AIRLOGIC

3.1 Operating Principle

Our goal is to fabricate interactive devices with little to no user intervention in the production and operation process. For this vision, we require a computing substrate with the following properties:

- it should be able to be completely produced on a single printer with no user intervention;
- it should be capable of sensing, processing, and output;
- it should be embeddable in objects during printing;
- it should be stateless: objects should not require re-setting after use.

To create such a substrate, we rely on pneumatic airflow in combination with the principles of fluerics as described in the previous section. Unlike previous pneumatics-based approaches which required external sensing [56, 57], and complex fabrication techniques [19, 51, 62], AirLogic uses a single-step fabrication process, senses a variety of input events, performs simple computations based on those events, and creates output based on the computations. The key is that—inspired in part by fluerics—we use 3D-printed geometry to enable a continuous flow of air to act as a *power source*, allowing AirLogic-based structures to perform functions analogous to those performed by electrical circuits (see Table 1). Because the fluid has a specific origin and destination and does not travel in a loop, we refer to these structures as "transits." Here we briefly explain how each of AirLogic's three main parts (input, logic, output) work in the context of the sample object illustrated in Figure 1; later sections describe the components in greater detail.

Flueric transits behave analogously to electronic circuits. Once the bunny device illustrated in Figure 1 has finished printing, the first step is to provide it with a power source in the form of pressurized air. This air input is analogous to VIN or V+ in an electronic circuit, and can be seen entering the front of the bunny through a clear tube. The air flows through channels and splitters (analogous to wires) fabricated in the body of the model. The designer has specified two touch points on the bunny's surface. These are designed such that, in the absence of touch, air vents through them to the atmosphere (analogous to electrical ground). When blocked, however, the channels route the air to a flueric OR gate (described in-depth below). With either touch sensor covered, the air flows to the oscillating actuator (very roughly analogous to a motor) embedded in the bunny's tail, which then wiggles up and down with the force of the air striking the paddle on its way to the atmosphere.

While the functioning of the input and output widgets is fairly intuitive, the operation of flueric logic gates is less so. As noted, and similar to historical flueric gate designs, these operate on the principle of momentum transfer between jets of air: an air jet's course can be modified by striking it with another air jet. Each logic gate in AirLogic uses 3D-printed geometry to form streams of air into jets and to direct those jets into an "interaction region". In the case of the OR gate (Figure 1 left), a single air jet from either input proceeds at an angle through the interaction region, and the cupping wall of the output channel catches it and directs it to the output. When both are present, they collide, canceling each other's angle and forming a single coherent jet that exits the output.

3.2 Fabrication

Historic flueric components were produced via chemical etching or machining, achieving very fine detail. We used an Ultimaker S3 and a Creality Ender 3 Pro. Due to the limitations of FDM-based manufacturing, we cannot reproduce the same levels of smoothness in the air tubes, nor similar tiny diameters of channels. Historic components were also typically produced in multiple independent layers fused together into a stack (e.g., [61, Fig. 5]).

Electronic Component	AirLogic Analog
Circuit	Transit
Power Source	Pressurized Air
Wires	Tubes
Ground	Vent to Atmosphere
Logic Gate	Flueric Logic Gate
Motor	Oscillating actuator

Table 1: AirLogic components' rough analogs in electronics.

To compensate for these limitations, our components differ somewhat from classic flueric designs. To prevent printing errors from sealing the air channels, we increased the scale of our designs: where a historic gate might have channel features as narrow as 0.2 mm, our smallest opening is 1 mm. In addition, the stacked construction of flueric gates led to rectangular channels with heights of 1 mm or less; our channels are tube-shaped and 5 mm in diameter. We found that this combined with 2 mm jet-forming reductions produced the best trade-off between performance and airflow.

Our larger channels cause an increase in working volume as compared to classic components, correspondingly necessitating a greater mass-flow rate of air, and therefore a higher operating pressure. While our components can be seen as less efficient, the higher pressure connotes an important advantage: our devices are more-easily able to operate at human scales, with a higher pressure enabling actuators such as those illustrated in Figures 1 and 5.

Another drawback to FDM-based manufacturing is that the layer ridges cause turbulence inside the air channels. Classic flueric components often operated with laminar airflow [64], allowing them to take advantage of phenomena such as the Coandă effect for creating bistable flueric switches [16]. We mitigate layer-induced turbulence with layer heights of 0.06–0.12 mm. We experimented with a Formlabs Form 2 printer, but found that uncured resin residue in our jet-forming reductions affected performance.

Because we add pneumatic input and output capabilities to historic flueric logic designs, using single-material FDM printers also comes at the cost of (very minimal) assembly of moving parts. Multi-material printers can construct AirLogic devices as a single structure by embedding dissolvable, breakable or otherwise-removable support materials, however devices constructed as here must have moving parts (e.g., buttons, vibration motors) printed separately and manually assembled, or added with pauses mid-print.

Other features of our components remain similar to classic designs. We use reductions in air channels to form jets in interaction regions, and in some cases add vents (e.g., the holes in the sides of the bunny in Figure 1) to ensure back-pressure from output loading does not affect the upstream system [5, Ch. 3].

4 AIRLOGIC WIDGET TOOLKIT

We present a set of pneumatic structures for sensing input, providing output, and performing basic logic operations. The widgets are all interoperable within transits, and designers can customize the user-facing portions at design time (e.g., larger button, knurled

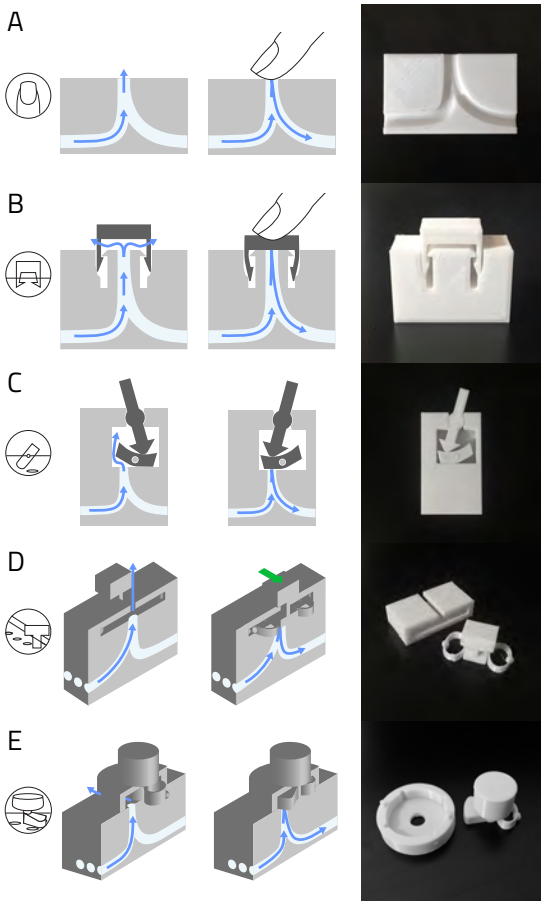

Figure 3: Our five input widgets: (A) Touch, (B) Button, (C), Switch, (D) Slider, (E) Dial. The leftmost column shows a schematic view of the widget in its default (non-interacting) state and the middle column shows how the airflow changes with user interaction. The right column shows a cutaway view of each printed widget illustrating its mechanism; in normal use, these structures would be seamlessly embedded into the surrounding object.

dial). We also note that this is not an exhaustive set of the potential widgets: we merely wish to illustrate the possibilities.

4.1 Input widgets

Our input widgets' internal designs are based on an inverted T-joint (Figure 3A). In its most basic design, pressurized air is injected on one arm of the T and will only continue its trajectory out the other arm of the T when the air vent at the T's stem is blocked.

(1) *Presence.* Touch widgets use the basic T-joint design (Figure 3A); touching (or otherwise obstructing) the vent at the top allows the air to continue its trajectory.
(2) *Push Button.* Embedding a cap and slots inside the T-joint structure realizes a push button (Figure 3B). The cantilever spring design ensures the button cap always returns to its original position when released.
(3) *Switch.* The switch widget design integrates a lever and wedge atop the basic T-joint (Figure 3C). Moving the lever causes the wedge to cover the sensing structure, allowing air to continue flowing inside the object.
(4) *Slider.* Our slider widget uses a series of our basic sensing structures, connected in parallel to the air source and arranged in a straight line. The user-facing slider sits in a linear rail and has a large, flat base which obstructs the air escape of the channel below. We added spring-loaded stops at each sensing location to form detents (Figure 3D).
(5) *Dial.* Similar to the slider, the dial widget uses multiple sensing structures, arranged in a circle. As the user twists the dial, a wedge-shaped obstruction is rotated into place above a given sensing channel. The dial widget also has detentes at each sensing location (Figure 3E).

4.2 Logic Widgets

Our logic widgets leverage interacting jets of air to compute logical operations, modifying airflow through a device based on sensed input.

While the majority of our input and output widgets use moving parts to operate, our logic widgets do not require mechanical parts. This has two main benefits. First, *printability*: because there is nothing to assemble, we can fabricate the core of an AirLogic device as a single structure, requiring only minimal assembly of external moving parts. Second, *reliability*: the lack of moving parts means the object's inner workings will not degrade with use, with the added bonus that it is robust against movement and vibration.

Below we describe the operation of our four logic widgets.

(1) `AND`. Our `AND` logic gate widget (Figure 4A1) has inputs to the left, a single output on the middle right, and vents at the top and bottom right. When only one input is present (Figure 4A2,3), the flow is directed to the corresponding vent channel. If both inputs are present, their jets collide, redirecting flow to the logical output channel (Figure 4A4).
(2) `OR`. Our `OR` logic gate widget (Figure 4B1) has two input channels on the left, an output channel to the right, and two vents in the top and bottom (to reduce backpressure in the system). This design operates as an "inclusive or": if either input is active (Figure 4B2,3), its flow is directed to the output channel. When both inputs are active, their flows combine and the resulting jet is also directed to the output channel (Figure 4B4).
(3) `XOR`. Our exclusive or (`XOR`) logic gate widget (Figure 4C1) uses the same basic design as our `AND` logic gate, but instead of vents redirects the single-input channels to a shared output. When one input is present, its air jet is directed to the output channel (Figure 4C2,3). If both inputs are present, their jets collide, redirecting flow to the out-of-plane central vent (Figure 4C3).
(4) `NOT`. Our `NOT` logic gate widget (Figure 4D1) uses the same basic design as `AND` and `XOR`, with some changes: logical output is on the top channel; the two bottom channels are vents. Here, the lower-left "input" channel is a "power" channel

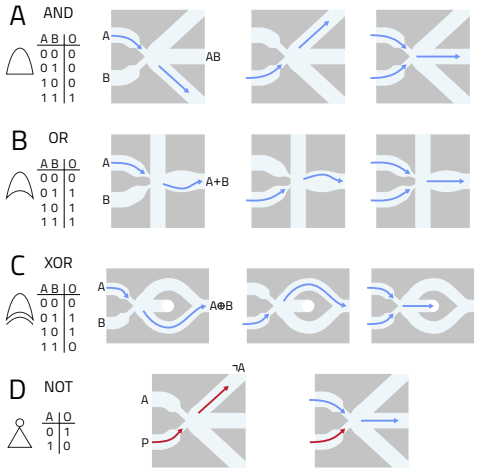

Figure 4: Our logic widgets: (A) AND, (B) OR, (C) XOR, (D) NOT.

P, with a constant flow of air regardless of the input A (Figure 4D, left). When A is present, P is redirected to the middle vent, providing 0 on the logical channel (Figure 4D, right).

4.3 Output Widgets

We developed air-powered widgets that present acoustic, visual, and vibrotactile output at the conclusion of input and logical calculation.

(1) *Pin.* Inspired by research in shape-changing interfaces [14], our pin widget provides visual output (Figure 5A). This widget is comprised of a cylindrical piston inside a chamber, actuated by the application of air pressure.
(2) *Whistle.* Our whistle widget (Figure 5B) provides acoustic output and is made up of three main components: an intake, a fipple, and a chamber. Air from logic operations enters through the intake and exhausts through the fipple, creating a tone. Varying internal chamber size changes the pitch [20].
(3) *Oscillating actuator.* Our wiggler widget (Figure 5C) can agitate sections of a device with a lever that is pushed by incoming jets of air. When moving, the lever shortly falls out of phase with the air jet and returns to its original position, causing it to be pushed once more. This widget relies on closure change instead of length change as in the pin widget [27] and enables an output that is less "binary" in its visual characteristics (its wiggle speed can be varied, or it can be popped fully).
(4) *Vibration motor.* This widget (Figure 5D) provides vibrotactile feedback. It operates similarly to electronic vibration motors commonly found on smartphones, where a—usually imbalanced—mass is spun to create different vibration patterns. In our design, incoming air spins a fan structure which is loosely coupled to its shaft, causing vibration.

5 DESIGNING AIRLOGIC OBJECTS

AirLogic offers two strategies for designing interactive objects: a prototyping workflow that allows for rapid testing, and a design pipeline for embedding AirLogic widgets inside existing 3D models.

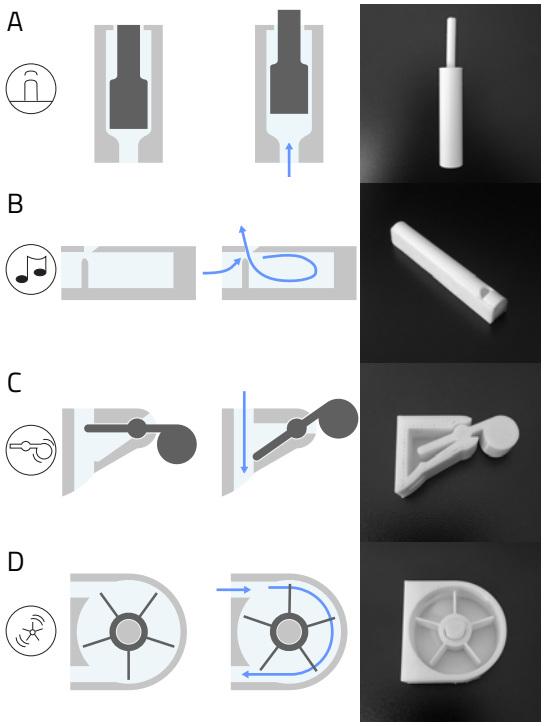

Figure 5: Output widgets. (A) pin, (B) whistle, (C) oscillating actuator, (D) vibration motor. The leftmost column shows a schematic view of the widget with no airflow and the middle column shows how the airflow causes output. The right column shows a cutaway view of each printed widget illustrating its mechanism; in normal use, these structures would be seamlessly embedded into the surrounding object.

We developed a plugin for Autodesk Fusion 360 that supports both flows (Figure 6), which we are releasing for community use[2].

For prototyping, a designer works with *encapsulated* widgets: these basic individual components in boxes are printed, connected with off-the-shelf tubing, and powered with a constant air source (Figure 6 A-B). After the designer is happy with their prototyped transit, they can add the requisite components directly to a 3D model (Figure 6 C). Some modifications to the widgets are possible without compromising functionality, like changing the shape of the user-facing dials; designers cannot re-size interior widget parts as the jet forming reductions and logic gates would be unlikely to work. Finally, the designer manually connects the widget models with pipes (this could be automated in a future tool using, e.g., PipeDream's curvature energy functions [46]) (Figure 6 D), and cuts the pipes and widgets from their model using the "subtraction" tool. They then print the object (Figure 6 E).

While prototyping with encapsulated widgets is a heavily manual process for the designer involving many connections, fabricating and assembling a device with internal widgets requires little intervention. The majority is printed as a single structure, with assembly only required for moving parts and aesthetic covers.

[2]Available at https://github.com/shape-changing-interfaces/AirLogic

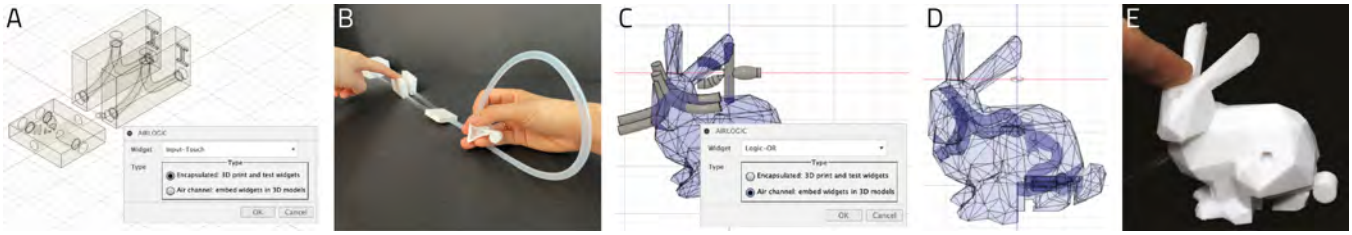

Figure 6: (A) A designer first selects and prints the encapsulated versions of the widgets they want to use in their object. (B) They can then prototype by physically connecting them with tubes. (C) Designers then add the unencapsulated versions of the widgets into their 3D model, (D) manually add tubes, and (E) print their final, interactive object.

6 VALIDATION

To validate AirLogic and assess its practical feasibility, we empirically evaluated our widget capabilities, and developed several illustrative applications.

6.1 Theoretical Validation

Classical flueric computation requires that the fluid system has a laminar flow, as determined by the Reynolds Number [52, 53] of the configuration: this unitless value relates the pipe diameter, fluid velocity, and fluid viscosity. Intuitively speaking, a high Reynolds Number implies that air is "piling up" on itself and creating eddies inside a pipe; a low value means it can flow smoothly. Smooth airflow was a key part of classic gates, as their small size left little room for error. We calculated the Reynolds Number of our various configurations and discovered that it is $\approx$ 113, 350, which is far larger than the $\leq$ 2100 that describes a system with laminar flow. Classic gates typically had openings for jet output that were approximately the same size as the jet-forming reductions; we have a larger "catch" opposite our jet reductions and our designs are larger overall, which may be what allows them to tolerate non-laminar flow. This merits further investigation.

6.2 Technical Evaluation

We empirically evaluated widget air operating requirements and losses, as well as the effects of print orientation and internal air channel bed angles. Using an anemometer and pressure sensor, we recorded airspeed into and out of our input and logic widgets at various input levels, airspeed through a single widget type with various printing parameters, and optimal activation pressures for our output widgets.

These experiments highlight two qualities of our designs: airflow needs and printing requirements. Taken together, the findings can inform mass flow needed for a given device, how widgets are best arranged for printing, and widget chaining possibilities.

We used a JunAir 2000-40PD air compressor, a Festo MS4-LR-1/4-D5-AS valve, an analog Panasonic PS-A (ADP5151) barometric sensor, and a Kestrel 3500 NV Pocket Weather Anemometer. We printed encapsulated versions of our widgets, connecting their output channels to our barometric pressure sensor and anemometer with off-the-shelf rubber tubes, OD 6 mm, ID 5 mm. Using the measured airspeed, cross-sectional area of our tubes, and density of air, we calculate and report mass flow rate [40] in kg/s. For the printing parameters test, we printed twenty-five copies of our `OR` widget: four sets of five with the internal tubes angled 0–90 degrees from the airflow direction, and five total with printed internal pipes of bend radiuses from 0–20 mm before the gates.

6.2.1 Results. **Air loss in widgets** Our tests showed that in general we lose proportionally less airflow when powering our systems with lower airflow (see Figure 7). On average, when powered with $5e^{-5}$ kg/s of air, our logic and input widgets lost 33.1% of the airflow, but at $18e^{-5}$ kg/s they lost 48.0%. This is likely related to laminarity: some widgets may work best when airflow is closer to laminar, such that it "sticks" properly to the interior walls of the printed tubes. Our logic widgets tended to lose less airflow than our input widgets (26.8% vs. 47.6%), with the `XOR` widget performing exceptionally well (average loss for non-`XOR` logic gates: 34.9% vs. 7.38% for `XOR`). We hypothesize this is related to `XOR`'s relative insensitivity to turbulence (`AND` is sensitive due to the two-jet interaction, `OR` is sensitive due to its escape valve geometry). The touch input widget also performed very well, likely because it can be arbitrarily well-sealed at the top—a squishy finger can close an air escape more completely than a rigid piece of plastic. We did not evaluate the pressure losses of our output widgets, as they are intended to be the last element in our transits. While we did not formally measure the escapes out the "wrong" holes, we experienced that with the tested gates there was very little "erroneous" air signal. We did experience more erroneous signal with the multi-way `AND` gate in our demo application (see Discussion). The button seemed to pass the most signal of our inputs while in the un-pressed configuration, likely since the cap is only slightly above the hole.

We thus recommend powering AirLogic devices with as little airflow as possible, given the constraints of downstream widgets.

Air loss from print orientation and printed pipe curvature Overall, more gradual printed curvatures led to better preservation of airflow in our widget designs, with losses ranging from 21.9% at 20 mm radius to 43.5% at 0 mm radius (see Figure 8). We saw no distinguishable pattern from printing angle (see Figure 9), in spite of having many datapoints; loss pattern were fairly consistent across all printings with the same orientation, but there was no progression tied to the specific angle. We suspect internal printing artifacts, e.g., how the layers of the printer met up with the internal geometry, affected jet formation at some orientations. The gates printed at 0, 22.5, and 90 degrees performed uniformly well (losing 27.0–30.5% of airflow on average).

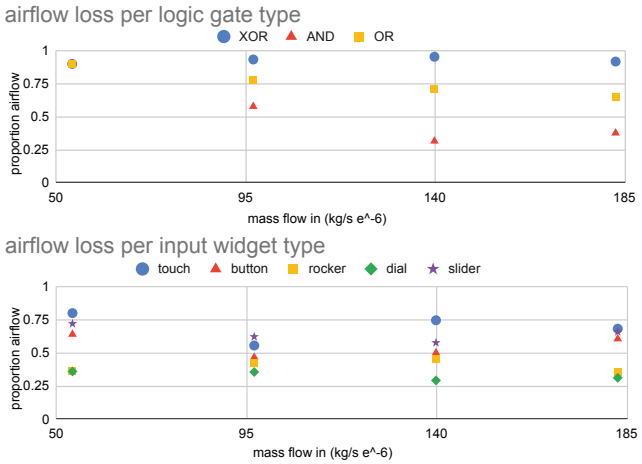

Figure 7: Our logic (top) and input (bottom) widgets caused varying amounts of air loss, dependent on input airflow.

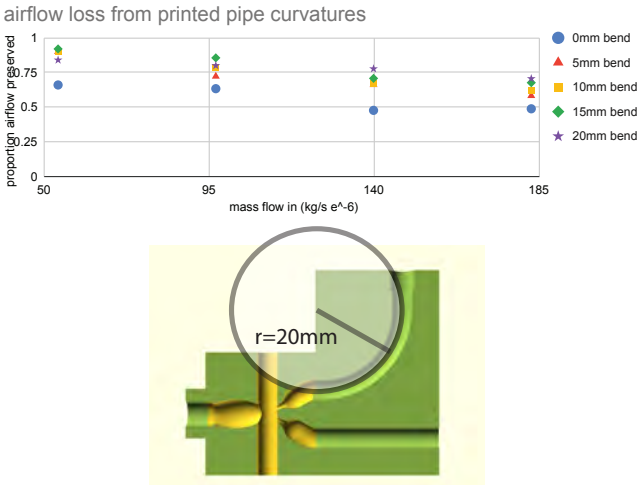

Figure 8: We tested drops in airflow depending on the sharpness of curves leading into our logic gates. Unsurprisingly, more gradual curves preserve airflow more effectively.

We thus suggest including gradually-curving pipes where possible; more work is needed to understand the best printing orientation given devices' complex internal geometry.

Optimal output pressure. Most of our output widgets operate best when actuated with pressures from 5+ kPa. They can likely be tuned for particular pressure systems (e.g., by adjusting counterweights or pressure-exposed surface area and shape), but in the particular configurations we tested we found the pin display works at 13.5+ kPa, the vibration motor works from 13.5–35 kPa, the whistle works from 5–7 kPa, and the wiggler wiggles at .3 kPa, working as a permanently-activated pin display above that.

Given the results of our previous investigations, designers can calculate the amount of input pressure required in their designs in order to optimally actuate their desired outputs.

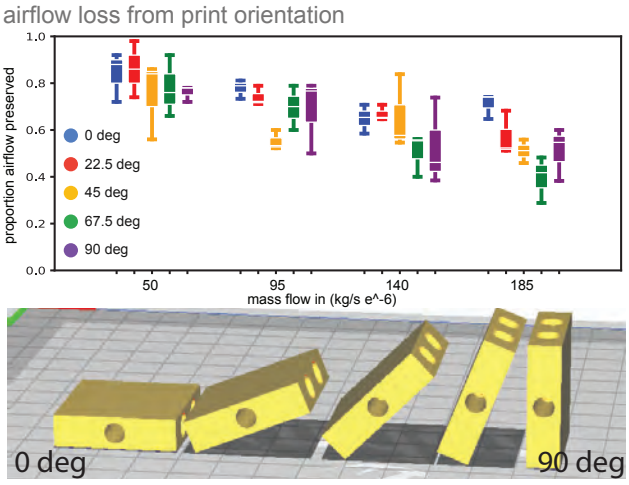

Figure 9: We tested for drops in airflow (top) depending upon our gates' orientation in the printer (bottom). The results did not follow a clear pattern and merit further study.

6.3 Example Applications

We present a series of applications of AirLogic, illustrating AirLogic's capabilities and potential for fabricating custom interactive objects without electronics.

6.3.1 Interactive Bunny. Using our touch, OR, and wiggler widgets, we constructed an interactive bunny that wiggles its tail when pet in one of the touch points on the forehead (Figure 1).

6.3.2 Block Puzzle. We constructed an interactive puzzle using the letters U, I, S, and T. When arranged to spell UIST, a pin with an attached flag is raised (Figure 10). This device splits the airstream into 4 parts and uses touch inputs to sense the blocks' presence (as the widgets' escape hole can be blocked by any object) and identity (each block has a particular void pattern underneath). The touch inputs are routed to a 4-way AND gate to determine if all holes are blocked. Due to air loss and stream splitting, this device requires ≈ 600 kPa to power. This device showcases our widgets' standalone capabilities: while, due to its scale, it takes up to 7 days to print the full box with integrated internal tubes, a smaller print of encapsulated widgets requires a matter of hours, and assembly with tubes takes only minutes. We envision our widgets could be used in this mode or to prototype larger interactive devices with techniques like Maker's Marks [44] or WYSIWYG [25].

6.3.3 Split or Steal Game. Modeled on the prisoner's dilemma and the classic English gameshow Split or Steal[3], this game box uses an XOR widget to determine if players are choosing to share (button pressed) or steal (button not pressed) the pile of money in the center (Figure 11). This application benefits from the AND widget embedded in the XOR widget. If players both choose to share, the airstream from the AND widget pushes the latch—a modified pin widget—and the money falls into a box between the players to share. If one player chooses to steal, the money is blown into the air by the

[3]https://en.wikipedia.org/wiki/Golden_Balls

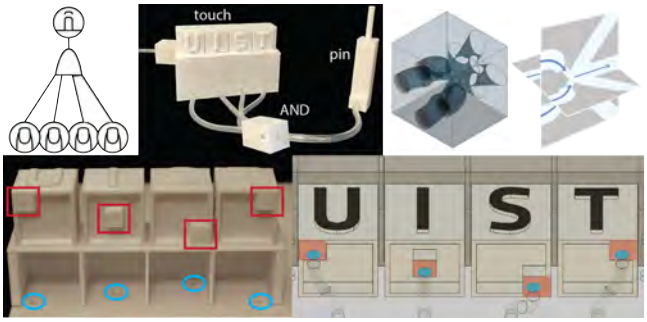

Figure 10: Our puzzle (top left) uses a 4-way air split and a 4-way AND (top right) with a radial arrangement to detect if the user correctly spelled UIST. The blocks (bottom) are identified by the locations of tube blockers (red) that match up with touch points (blue) when the blocks are put in place (render).

Figure 11: The Split or Steal game drops money if both players choose split (AND), or blows the air in and spreads the money if a player chooses steal. (XOR).

Figure 12: This clay dog head is modeled over a bare set of widgets that detect touch A OR touch B and actuate an oscillator; the same widgets form the core of the bunny.

airstream from the XOR widget. This also shows that the airstream itself can be used as output, without widgets.

6.3.4 Prototyped Dog. We printed just the sensing core of the bunny example and mocked up a new "case" around it using craft dough—a dog whose tongue wags when the user pets it (Figure 12).

6.3.5 Lung-powered pitch selector. To highlight alternative air sources, we used our slider and whistle widgets to fabricate a handheld, lung-powered pitch selector (Figure 13). The user selects the frequency she wants to play using a slider, and when she blows into the input, the desired tone is played using the respective whistle. This application also highlights that in some cases, a logic widget is not required to obtain the desired functionality.

Figure 13: Our lung-powered pitch slider has users blow into the air intake and slide the slider to generate different tones.

7 DISCUSSION & LIMITATIONS

7.1 Chaining logic widgets

While our applications illustrate using multiple logic widgets in an AirLogic object (Sections 6.3.2 and 6.3.5), this functionality is limited in our current implementation. In exploratory tests we found that our logic widget designs can be successfully connected three different ways: in parallel, balanced chained AND, and chained OR. We cannot support unbalanced chained AND, or combinations of different logic widget types.

The main reason for this is that our AND design relies on identical pressures from both input channels to function correctly. If one channel has more pressure—therefore more momentum—than the other, it pushes the combined jet towards an escape vent. This means that, if an AND widget were to be connected after an OR, which can be activated using one or two inputs, we would have to dynamically regulate the pressure of the second input for our AND widget depending on the number of inputs used in the previous OR widget. This is somewhat mitigated by the design of our multi-way AND gate, used in the Interactive Puzzle example: however, with this type of logic gate, the turbulence from multiple colliding input airstreams can lead to erroneous output.

We aim to tackle this issue by standardizing our logic widgets' output. Doing so will guarantee that the results from our logic operations will have the same pressure profile as our input widgets, no matter how many inputs are calculated on. Our current designs are based on *passive* flueric devices, where the a gate's output is a combination of its input flows. Active logic designs use flueric amplifiers [8, 54] as switches, so that a logical TRUE output is always the same regardless of the number of inputs.

7.2 Alternate Fabrication Methods

FDM-based printers are not the only fabrication method that is available to makers. During our explorations we constructed our widgets using both FDM and SLA printers, as well as a laser cutter. Interestingly, despite a higher layer-wise resolution of 25 microns, SLA-printed widgets performed worse than those constructed by our FDM printer. We fabricated a series of AND gates using a Form 2 printer, and most had poor performance due to blockages created by uncured resin residue trapped in the jet-formation reductions. These blockages, given the small size of our channels and high pressure sensitivity of our designs, adversely affect gate performance. We achieved promising results by flushing a gate with isopropyl alcohol before the residue cured, but more experimentation is necessary, and we anticipate that the continuing democratization of fabrication will enable more and more printers to create the types of geometry required for AirLogic devices. Smoother internal tubes may also

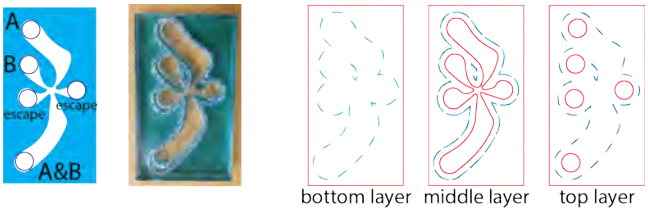

Figure 14: We recreated a classic, planar flueric AND gate (left) on a lasercutter (middle). The three-layer design (right) requires both cutting with material removal (red lines) and welding pairs of layers (blue lines).

mitigate the bend radius issues encountered in our evaluation and enable more flexibility in design.

Thanks to its high precision, clean cuts, and similarity to historical methods, laser cutting is also a promising fabrication method. We leveraged LaserStacker-style cutting mixed with welding [59] and removal/restacking of material to create voids as we recreated classic planar flueric geometries (see Figure 14). AirLogic devices constructed in this way require significantly more manual assembly than our printed implementation and would likely be limited to 2D (thus limiting the kinds of possible computation, per the three utilities problem[4], as well as the possibilities for input and output components). Future work could iterate on fabrication techniques and provide design tools for such machines.

7.3 Comparison with Electronics

While AirLogic enables embedding end-to-end computation in digitally-fabricated objects, its capabilities are considerably less-sophisticated than electrical circuits made with off-the-shelf electronic toolkits like Arduino[5]. Electronic toolkits allow a high degree of flexibility and variety of interactions (high ceiling), but this comes at the cost of a steep learning curve and high threshold to get started. For now, AirLogic targets both a lower threshold and a lower ceiling in terms of design complexity. We plan to explore novel structures representing more intricate operations like timers, proximity, temperature, and light sensors to expand our ceiling. Through the use of amplifiers, classical flueric gates could also operate on *analog* input signals: we look forward to development in this area with modern consumer fabrication technologies. Similarly, works like Aeromorph [38], MorpheesPlug [27], and PneUIs [67] have explored fabricatable inputs and outputs based on air; integrating these and similar techniques with AirLogic's computational structures could increase I/O diversity.

Even with improved capabilities, we still view AirLogic as a complement to—rather than a replacement for—traditional electronic components. AirLogic objects shine in use cases where, for example, (1) computation is simple but traditional electronics would be in danger of getting wet or irradiated: a designer could create an umbrella reminder that is triggered when a user walks towards the door and it is raining outside, an irrigation system could be configured using nothing more than the water already flowing through it, or AirLogic devices could be used in the International Space Station where primary cosmic rays are prevalent; (2) when electrical power sources are difficult to come by, such as for a water level monitoring device in a remote location in a national park; (3) where computation is incidental to the intended purpose of a designed object, like a 3D-printable musical instrument that (when air is applied) can help a learner understand if their grip is correct. Electronics are sensitive to various environmental factors; more work is needed to understand the effect of the environment on AirLogic devices, for example, their use in high wind.

7.3.1 Sustainability. As most electronically-interactive objects rely on tightly coupling the electronic components to the design as well as embedding them (with either mounting screws or pauses during prints), separating the constituent materials of and recycling such objects can be a challenge. AirLogic devices, on the other hand, can be fully recycled in a single piece based on their material.

7.4 Other air sources

All interactive devices require a power source to operate. In the case of electronic devices it is electricity, and for AirLogic devices it is air. The main issue with AirLogic objects is that, while electrical battery technology has been heavily researched, constant air sources (particularly portable ones) can be harder to come by. We powered our prototypes with an air compressor (JunAir 2000-40PD), however other air sources are possible. Informal experiments showed that users can power our transits with their lungs, and we believe designers can use our characterization results to calculate the mass flow requirements of their designs to choose their air source accordingly (compressor, lungs, or perhaps compressed air canisters such as those used for paintball guns). In countries such as Canada, central vacuum systems—where the vacuum motor and dirt collection are located in a central room, with access holes that provide suction spread through the house—are common, which suggests the possibility of an external *negative* pressure source. We are also interested in experimenting with other non-tethered air sources, such as the heat-differential-generated air- and steam-flow that powers classic pop pop boats[6] and the ancient Aeolipile[7].

8 CONCLUSION

This paper presented AirLogic, a technique to fabricate interactive 3D printed objects with integrated air-powered sensing, computation, and output. We discussed 13 pneumatic widgets demonstrating this technique. We also highlighted, through measurement, the opportunities and challenges of the current widget designs as fabricated on consumer 3D printers, and we demonstrated their use in exemplar interactive objects that do not require any electronics or coding. We believe our work moves towards the vision of instantly interactive, single-pass printed objects decoupled from external computing devices.

ACKNOWLEDGMENTS

This work was partially supported by a Novo Nordisk Fonden Starting Grant under grant number NNF21OC0072716.

[4]https://en.wikipedia.org/wiki/Three_utilities_problem
[5]https://www.arduino.cc/
[6]See, e.g., https://www.youtube.com/watch?v=u1X76MK5dHo
[7]https://en.wikipedia.org/wiki/Aeolipile

REFERENCES

[1] 1968. *Fluidic Components and Equipment 1968–9*. Elsevier. https://doi.org/10.1016/C2013-0-02239-6

[2] Rafael Ballagas, Sarthak Ghosh, and James Landay. 2018. The Design Space of 3D Printable Interactivity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (July 2018), 1–21. https://doi.org/10.1145/3214264

[3] Nicholas Bar. 1988. The Phillips Machine Project. 75 (June 1988), 3.

[4] Ayah Bdeir. 2009. Electronics as Material: littleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI '09)*. Association for Computing Machinery, New York, NY, USA, 397–400. https://doi.org/10.1145/1517664.1517743

[5] Charles Belsterling. 1971. *Fluidic Systems Design*. Wiley-Interscience. http://archive.org/details/BELSTERLINGFluidicSystemsDesign1971

[6] D Bouteille and C Guidot. 1973. *Fluid Logic Controls and Industrial Automation*. John Wiley & Sons.

[7] A. G. Bromley. 1998. Charles Babbage's Analytical Engine, 1838. *IEEE Annals of the History of Computing* 20, 4 (Oct. 1998), 29–45. https://doi.org/10.1109/85.728228

[8] Charles Belsterling. 1971. *Fluidic Systems Design*.

[9] Martin Feick Anusha Withana Jürgen Steimle Daniel Groeger. 2019. Tactlets: Adding Tactile Feedback to 3D Objects Using Custom Printed Controls. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. New Orleans, LA, USA, 1–14.

[10] Kyriakos Efstathiou and Marianna Efstathiou. 2018. Celestial Gearbox. *Mechanical Engineering* 140, 09 (Sept. 2018), 31–35. https://doi.org/10.1115/1.2018-SEP1

[11] DR Arnold Emch. 1901. Two Hydraulic Methods to Extract the Nth Root of Any Number. *The American Mathematical Monthly* 8, 1 (Jan. 1901), 10–12. https://doi.org/10.1080/00029890.1901.12000520

[12] Michele Ferlauto and Roberto Marsilio. 2017. Numerical Investigation of the Dynamic Characteristics of a Dual-Throat-Nozzle for Fluidic Thrust-Vectoring. *AIAA Journal* 55, 1 (2017), 86–98. https://doi.org/10.2514/1.J055044

[13] J. D. Foley, V. L. Wallace, and P. Chan. 1984. The Human Factors of Computer Graphics Interaction Techniques. *IEEE Computer Graphics and Applications* 4, 11 (Nov. 1984), 13–48. https://doi.org/10.1109/MCG.1984.6429355

[14] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: Dynamic Physical Affordances and Constraints through Shape and Object Actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/2501988.2502032

[15] M. Garrad, G. Soter, A. T. Conn, H. Hauser, and J. Rossiter. 2019. A Soft Matter Computer for Soft Robots. *Science Robotics* 4, 33 (Aug. 2019). https://doi.org/10.1126/scirobotics.aaw6060

[16] H. H. Glaettli. 1964. Digital Fluid Logic Elements. In *Advances in Computers*, Franz L. Alt and Morris Rubinoff (Eds.). Vol. 4. Elsevier, 169–243. https://doi.org/10.1016/S0065-2458(08)60221-1

[17] Saul Greenberg and Chester Fitchett. 2001. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, New York, USA, 209–218. https://doi.org/10.1145/502348.502388

[18] Daniel Groeger, Elena Chong Loo, and Jürgen Steimle. 2016. HotFlex: Post-Print Customization of 3D Prints Using Embedded State Change. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 420–432. https://doi.org/10.1145/2858036.2858191

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

[20] Herman L F Helmholtz. 1885. *On the Sensations of Tone as a Physiological Basis for the Theory of Music* (2 ed.). Longmans, Green, and Co., London, United Kingdom.

[21] Steve Hodges, Nicolas Villar, Nicholas Chen, Tushar Chugh, Jie Qi, Diana Nowacka, and Yoshihiro Kawahara. 2014. Circuit Stickers: Peel-and-Stick Construction of Interactive Electronic Prototypes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1743–1746. https://doi.org/10.1145/2556288.2557150

[22] Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. 2017. Digital Mechanical Metamaterials. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM Press, Denver, CO, USA, 977–988. https://doi.org/10.1145/3025453.3025624

[23] Yoshio Ishiguro and Ivan Poupyrev. 2014. 3D Printed Interactive Speakers *(Proceedings of the SIGCHI Conference on Human Factors in Computing Systems)*. 1733–1742.

[24] Yuhua Jin, Isabel Qamar, Michael Wessely, Aradhana Adhikari, Katarina Bulovic, Parinya Punpongsanon, and Stefanie Mueller. 2019. Photo-Chromeleon. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New Orleans, LA, USA, 701–712. https://doi.org/10.1145/3332165.3347905

[25] 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. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 876–886. https://doi.org/10.1145/2858036.2858493

[26] Shohei Katakura and Keita Watanabe. [n. d.]. ProtoHole: Prototyping Interactive 3D Printed Objects Using Holes and Acoustic Sensing. In *CHI '18: Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems* (2018-04). ACM, LBW112–6. https://doi.org/10.1145/3170427.3188471

[27] Hyunyoung Kim, Celine Coutrix, and Anne Roudaut. 2018. Morphees+: Studying Everyday Reconfigurable Objects for the Design and Taxonomy of Reconfigurable UIs. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3174193

[28] Hyunyoung Kim, Aluna Everitt, Carlos Tejada, Mengyu Zhong, and Daniel Ashbrook. [n. d.]. MorpheesPlug: A Toolkit for Prototyping Shape-Changing Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021-05-06). Association for Computing Machinery, 1–13. https://doi.org/10.1145/3411764.3445786

[29] Diamond Ordinance Fuse Laboratories. 1960. News release, March 2, 1960. (March 1960).

[30] Arnaud Lacarelle and Christian O. Paschereit. 2011. Increasing the Passive Scalar Mixing Quality of Jets in Crossflow With Fluidics Actuators. *Journal of Engineering for Gas Turbines and Power* 134, 021503 (Dec. 2011). https://doi.org/10.1115/1.4004373

[31] Gierad Laput, Eric Brockmeyer, Scott E Hudson, and Chris Harrison. 2015. Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2702123.2702416

[32] David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. 2017. Pineal: Bringing Passive Objects to Life with Embedded Mobile Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM Press, Denver, CO, USA, 2583–2593. https://doi.org/10.1145/3025453.3025652

[33] Robert MacCurdy, Robert Katzschmann, Youbin Kim, and Daniela Rus. 2016. Printable Hydraulics: A Method for Fabricating Robots by 3D Co-Printing Solids and Liquids *(2016 IEEE International Conference on Robotics and Automation (ICRA))*. IEEE, 3878–3885.

[34] A. D. Moore. 1936. The Hydrocal. *Industrial & Engineering Chemistry* 28, 6 (June 1936), 704–708. https://doi.org/10.1021/ie50318a022

[35] Hila Mor, Tianyu Yu, Ken Nakagaki, Benjamin Harvey Miller, Yichen Jia, and Hiroshi Ishii. 2020. Venous Materials: Towards Interactive Fluidic Mechanisms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376129

[36] Bobak Mosadegh, Chuan-Hsien Kuo, Yi-Chung Tung, Yu-suke Torisawa, Tommaso Bersano-Begey, Hossein Tavana, and Shuichi Takayama. 2010. Integrated elastomeric components for autonomous regulation of sequential and oscillatory flow switching in microfluidic devices. *Nature Physics* 6 (1 6 2010), 5. Issue 6. https://doi.org/10.1038/nphys1637

[37] Roderick Murray-Smith, John Williamson, Stephen Hughes, and Torben Quaade. 2008. Stane: Synthesized Surfaces for Tactile Input *(Proceedings of the SIGCHI Conference on Human Factors in Computing Systems)*. 1299–1302.

[38] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. 2016. aeroMorph - Heat-Sealing Inflatable Shape-Change Materials for Interaction Design. In *Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 121–132. https://doi.org/10.1145/2984511.2984520

[39] Huaishu Peng, François Guimbretière, James McCann, and Scott Hudson. 2016. A 3D Printer for Interactive Electromagnetic Devices. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*. ACM Press, Tokyo, Japan, 553–562. https://doi.org/10.1145/2984511.2984523

[40] Merle C. Potter and D. C. Wiggert. 2008. *Schaum's outline of Fluid Mechanics*. London.

[41] Daniel J. Preston, Philipp Rothemund, Haihui Joy Jiang, Markus P. Nemitz, Jeff Rawson, Zhigang Suo, and George M. Whitesides. 2019. Digital Logic for Soft Devices. *Proceedings of the National Academy of Sciences* 116, 16 (April 2019), 7750–7759. https://doi.org/10.1073/pnas.1820672116

[42] Raf Ramakers, Kashyap Todi, and Kris Luyten. 2015. PaperPulse: An Integrated Approach for Embedding Electronics in Paper Designs. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2457–2466. https://doi.org/10.1145/2702123.2702487

[43] Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 447–456. https://doi.org/10.1145/2501988.2501992

[44] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 103–108. https://doi.org/10.1145/2807442.2807508

[45] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 1277–1280. https://doi.org/10.1145/2702123.2702207

[46] Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. 2014. A Series of Tubes: adding interactivity to 3D prints using internal pipes. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 3–12. https://doi.org/10.1145/2642918.2647374

[47] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 1–9.

[48] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. : A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, New York, USA, 253–258. https://doi.org/10.1145/2807442.2807503

[49] 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. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow, Scotland, United Kingdom, 454–13. https://doi.org/10.1145/3290605.3300684

[50] Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. 2016. Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 4896–4907. https://doi.org/10.1145/2858036.2858507

[51] Ronit Slyper and Jessica Hodgins. 2012. Prototyping Robot Appearance, Movement, and Interactions Using Flexible 3D Printing and Air Pressure Sensors. In *The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 6–11. https://doi.org/10.1109/ROMAN.2012.6343723

[52] A. Sommerfeld. 1883 and 1895. Ein beitrag zur hydrodynamischen erklaerung der turbulenten fluessigkeitsbewegungen. 174 and 186 (1883 and 1895), 935 and 123.

[53] Sir George Gabriel Stokes. 1850. On the effect of the internal friction of fluids on the motion of pendulums. ix (1850), 8.

[54] J. W. Tanney. 1970. Fluidics. *Progress in Aerospace Sciences* 10 (Jan. 1970), 401–509. https://doi.org/10.1016/0376-0421(70)90008-4

[55] Carlos E. Tejada. 2021. *Print-and-Play Fabrication*. University of Copenhagen. https://doi.org/10.31237/osf.io/nd7e9

[56] Carlos E Tejada, Osamu Fujimoto, Zhiyuan Li, and Daniel Ashbrook. 2018. Blowhole: Blowing-Activated Tags for Interactive 3D-Printed Models. In *Proceedings on the 44th Anual Conference on Graphics Interface*.

[57] Carlos E. Tejada, Raf Ramakers, Sebastian Boring, and Daniel Ashbrook. 2020. AirTouch: 3D-Printed Touch-Sensitive Objects Using Pneumatic Sensing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376136

[58] Todd Thorsen, Sebastian J. Maerkl, and Stephen R. Quake. 2002. Microfluidic Large-Scale Integration. *Science* 298, 5593 (Oct. 2002), 580–584. https://doi.org/10.1126/science.1076996

[59] Udayan Umapathi, Hsiang-Ting Chen, Stefanie Mueller, Ludwig Wall, Anna Seufert, and Patrick Baudisch. 2015. LaserStacker: Fabricating 3D Objects by Laser Cutting and Welding. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 575–582. https://doi.org/10.1145/2807442.2807512

[60] Tom Valkeneers, Danny Leen, Daniel Ashbrook, and Raf Ramakers. 2019. StackMold: Rapid Prototyping of Functional Multi-Material Objects with Selective Levels of Surface Details. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New Orleans, LA, USA, 687–699. https://doi.org/10.1145/3332165.3347915

[61] J Van Der Heyden. 1968. Fluidic Displays. In *Recent Advances in Display Media*. 71–80.

[62] Marynel Vázquez, Eric Brockmeyer, Ruta Desai, Chris Harrison, and Scott E Hudson. 2015. 3D Printing Pneumatic Device Controls with Variable Activation Force Capabilities. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 1295–1304. https://doi.org/10.1145/2702123.2702569

[63] Nicolas Villar, James Scott, Steve Hodges, Kerry Hammil, and Colin Miller. 2012. .NET Gadgeteer: A Platform for Custom Devices. In *Pervasive Computing (Lecture Notes in Computer Science)*, Judy Kay, Paul Lukowicz, Hideyuki Tokuda, Patrick Olivier, and Antonio Krüger (Eds.). Springer, Berlin, Heidelberg, 216–233. https://doi.org/10.1007/978-3-642-31205-2_14

[64] Rolf E. Wagner. 1969. Fluidics—a New Control Tool. *IEEE Spectrum* 6, 11 (Nov. 1969), 58–68. https://doi.org/10.1109/MSPEC.1969.5214171

[65] Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. 2016. An Integrated Design and Fabrication Strategy for Entirely Soft, Autonomous Robots. *Nature* 536, 7617 (Aug. 2016), 451–455. https://doi.org/10.1038/nature19100

[66] Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed Optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. ACM, Cambridge, Massachusetts, United States, 589–598. https://doi.org/10.1145/2380116.2380190

[67] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: Pneumatically Actuated Soft Composite Materials for Shape Changing Interfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/2501988.2502037