SoftMod: A Soft Modular Plug-and-Play Kit for Prototyping Electronic Systems

Mannu Lambrichts, Jose Maria Tijerina, Raf Ramakers
Hasselt University - tUL - Flanders Make, EDM
Diepenbeek, Belgium
firstname.lastname@uhasselt.be

ABSTRACT

We present SoftMod, a novel modular electronics kit consisting of soft and flexible modules that snap together. Unlike existing modular kits, SoftMod tracks the topology of interconnected modules and supports basic plug-and-play behavior as well as advanced user-specified behavior. As such, the shape of a SoftMod assembly does not depend on the desired behavior and various 2D and 3D electronic systems can be realized. While the plug-and-play nature of our modules stimulates play, the advanced features for specifying behavior and for making a variety of soft and flexible shapes, offer a high-ceiling when experimenting with novel types of interfaces, such as wearables, and interactive skin and textiles.

CCS CONCEPTS

• **Human-centered computing** → *User interface toolkits.*

KEYWORDS

Electronic Toolkit; Sensor Systems; Prototyping; Physical Computing

ACM Reference Format:
Mannu Lambrichts, Jose Maria Tijerina, Raf Ramakers. 2020. SoftMod: A Soft Modular Plug-and-Play Kit for Prototyping Electronic Systems. In *Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '20), February 9–12, 2020, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3374920.3374950

TEI '20, February 9–12, 2020, Sydney, NSW, Australia

ACM ISBN 978-1-4503-6107-1/20/02…$15.00
https://doi.org/10.1145/3374920.3374950

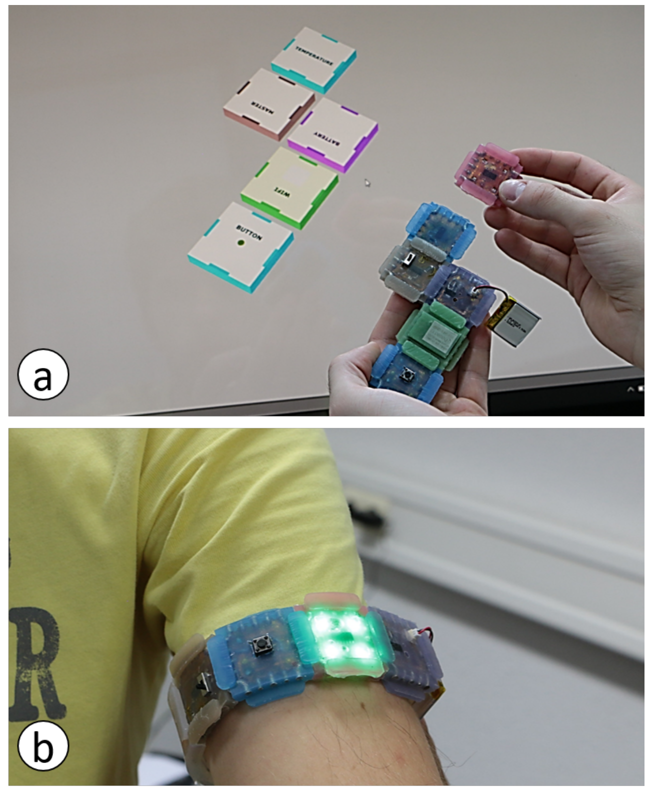

Figure 1: (a) snapping together SoftMods to (b) prototype an interactive armband.

1 INTRODUCTION

Electronic development boards, such as Arduino [3], Raspberry Pi[36], and Phidgets [15], support hobbyists in making electronic systems. However, these boards require basic knowledge of electronics and programming. To further lower the barrier for prototyping with electronics, plug-and-play electronic modules allow for making sensor systems by simply connecting together modules. Popular examples include LittleBits [5] that snap together and MakerWear modules [22] that interconnect through a flexible carrier. While these modules make prototyping electronic behavior convenient, the shape of the resulting electronic systems is mostly dictated by the functionality as re-configuring modules in different shapes results in different behavior. These modular kits are therefore often used for play and curiosity-driven learning.

To support maker enthusiasts in prototyping electronic systems with more intricate form-factors and sensors embedded in materials, researchers explored techniques to lower the barrier for crafting with electronics on paper [37, 39], textiles [16], and silicones [32] through workshops [29], online tutorials [35], and software tools [39]. Although these approaches result in electronic systems that are more permanent (sturdy) and closely resemble the look-and-feel of a final product, the workflows are often tedious which significantly increases the time required for making design variations.

In this paper, we present SoftMod (Soft Modules), a soft modular electronic kit that empowers people without an engineering background to make electronic systems in a desired shape by snapping together soft flexible modules (Figure 1a). SoftMod supports modules for sensing, output, processing, power, and wireless communication. Unlike existing plug-and-play modular kits, such as LittleBits [5] and MakerWear modules [22], the topology of a SoftMod assembly is tracked and the default behavior is adjusted accordingly. As such, the topology of interconnected modules can fit the desired shape of the electronic system and does not influence the behavior. More advanced logic is specified using the SoftMod IDE. SoftMods can also interconnect through soft stretchable cables which makes it possible to realize larger 2D or even 3D structures. On the one hand, SoftMod empowers and stimulates designers and enthusiasts to explore ideas for new types of interactive systems, such as wearables and interactive skin and textiles. On the other hand, SoftMod also appeals to children and adults to play and thinker with sensors without risking any damage to components or one's safety.

The fabrication process for our modules is fully DIY compatible and consist of flexible PCBs produced with a laser cutter and magnetic connectors precisely mounted and casted in place using laser cut molds. Therefore, we also provide details on the engineering and fabrication procedure for making SoftMod in order for the community to learn and adopt these approaches and to join our effort in maintaining and extending the SoftMod kit. The main contribution of this paper is a soft modular plug-and-play electronic kit to rapidly prototype sensor systems with diverse form-factors. This comprises three parts:

(1) The DIY engineering and design principles behind our modules, including the soft form factor, the electronic design, and mechanical interconnections.
(2) An embedded software framework, including communication protocols, to track the topology of interconnected modules and allow for basic plug-and-play behavior as well as end-user specified behavior.
(3) A set of example designs and use cases that showcase the novel possibilities and the utility of SoftMold for prototyping novel types of interactive systems.

2 WALKTHROUGH

The following walkthrough illustrates the process of making a soft interactive armband that serves as an interval timer while running. It consists of indicators notifying the user to switch between walking and running and a button to configure the interval. The final design is shown in Figure 1b and has the same functionality and a similar look-and-feel as the interval timer prototyped in Silicone Devices [32], using state-of-the-art crafting techniques. With SoftMod a similar prototype is realized in a fraction of the time (30min vs a full working day).

The process starts with snapping together a SoftMod master module, an indicator module, and a push-button module. As shown in Figure 1a, the embedded magnetic connectors make it very convenient to link modules. When powering the master module with a battery, SoftMod recognizes the topology of all connected modules and assigns a default behavior to the sensor system. In this configuration, all four LEDs of the LED module light up when touching the button. To update this default behavior, the master module is connected to a desktop computer running the SoftMod IDE. This software environment visualizes the topology and allows for reconfiguring the behavior of all modules using IF-This-Then-That (IFTTT) rules or a scratch-like visual programming environment (Figure 1a). In this example, the interval timer is configured with four modes: 3min running - 1min walking, 5min running - 5min walking, and 10min running - 5min walking. The push-button is programmed to switch the mode and the LEDs are configured to be powered while walking and blink when running.

Modules can be easily added, removed, or swapped during the prototyping stage. In this example, the user tests different cable modules to connect the first and last module and realizes a armband that matches the size of the arm (Figure 1b).

3 RELATED WORK

This work builds upon and relates to tools for lowering the barrier to prototyping electronic systems and Do-It-Yourself (DIY) crafting of electronic systems.

Tools for Lowering the Barrier to Prototyping Electronic Systems

Electronic development boards, such as Arduino [3] and Rasberry Pi [36] make it affordable and convenient for maker enthusiasts to prototype electronic systems. However, these toolkits still require users to have basic engineering expertise, including electronics and programming, to correctly wire and drive the sensor system. To allow users with limited to no expertise engineering expertise to build electronic systems, several products, such as Phidgets [15], .Net Gadgeteer [18], Lego Mindstorms [26], and MakeBlock [21], come with electronic modules that communicate over a bus by interconnecting modules with ribbon cables. As such, they allow for prototyping with high-level electronic modules without requiring the user to handle low-level components, such as resistors, capacitors, shift-registers, or amplifiers. Instead of using flat cables or networking cables, researchers also experimented with embedding a bus communication system in layered materials. In these approaches, conductive and insulating rigid [23], flexible [44], or textiles [14] materials are stacked and the connector pins of electronic components make contact with each of these layers. Such approaches avoid having many cables but still allow components to be configured in various layouts. To avoid wires entirely, modular toolkits, such as Data Flow [9] and Sifteo [30] communicate wireless.

Besides electronics, researchers also aim to lower the barrier for programming electronic systems using IFTTT [2, 4], state-charts [17], and scratch-like approaches [10, 11, 31]. To empower true novices and children to build electronic systems, several toolkits for sensor systems do not require programming and assign a basic behavior to every module in a construction kit [5, 8, 22, 41]. By composing these building blocks, various types of behavior can be realized as the signal sequentially passes through all the modules. Popular examples include LittleBits [5] for rigid sensor systems, MakeWear [22] for prototyping on textiles, and roBlocks [41] for robotic systems. Although these plug-and-play kits make it very convenient for making sensor systems, the shape and thus the form-factor is dictated by the desired sensor behavior as modules are composed to realize a certain behavior.

Popular toolkits that track the topology of modules include ActiveCube [45], roBlocks [41], Triangles [13], StrutModeling [25] and Sifteo [30]. These toolkits consist of modules that provide a visual overview of the construction by tracking their topology in real-time. For example, Triangles [13] and Sifteo [30] use topology information to empower children to compose interactive stories and puzzles. Similar to SoftMod, roBlocks [41] uses topology information to assign default interactive behavior to modules.

To allow modules to intercommunicate, various approaches have been explored. Similar to traditional breadboards, the Printoo flexible electronic kit [1] embeds header pins between modules. Although electronically reliable, header pins require mechanical force to assemble and are not fool-proof. To make interconnecting modules more convenient to novices, various connector and communication techniques have been explored. Triangles [13] and roBlocks [41] embed magnetic connectors that make connecting modules as convenient as snapping together magnets. ActiveCube [45], in contrast, uses mechanical hooks to allow for strong interconnections between modules. Sifteo [30] entirely avoids physical connectors and uses a wireless communication and near-field object sensing as topology detection method. Alternatively, infrared can be used to sense the presence and identity of nearby modules as demonstrated by Data Flow [9]. Although offering modules that are entirely self-contained, wireless communication techniques use more power.

In comparison to these approaches, SoftMod empowers non-engineers to prototype desired 2D and 3D shapes using a soft plug-and-play kit. By tracking the topology of modules in real-time, a basic sensor behavior is assigned. At the same time, more advanced users can alter this basic sensor behavior using visual programming paradigms without requiring changes to the form-factor of a SoftMod assembly.

Crafting Electronic Systems

In contrast to construction kits, embedding off-the-shelf electronic components in substrates oftentimes result in circuits that are seamlessly embedded in artefacts. Examples include, the Arduino Lilypad [7] and Adafruit Flora [42] that embed electronic circuits in fabrics by stitching conductive threads. These approaches, however, require crafting skills that can be acquired through workshops [29, 38] and online tutorials [35]. Perner Wilson et al. [35] argued, that unlike construction kits which are too general purpose, crafting approaches encourage personal expression and learning.

Various crafting approaches have been presented to embed electronic circuits in materials. Examples include the e-TAGs [27] that embed ribbon cable bus connectors in textiles to facilitate attaching and swapping electronic components. Instead of using rigid cables and connectors, i*CATch [34] uses spring snap buttons embedded in textiles for attaching electronic components. These buttons are interconnected using conductive strips that attach to fabrics using an iron-on adhesive. The TeeBoard [33] uses a similar technique but offers a breadboard-like layout, embedded in a shirt, as a universal electronic prototyping platform for textiles. To partially automate stitching of electronic systems in textiles, Sketch&Stitch [16] uses sketches of electronic circuit in combination with computer vision to drive a computerized embroidery machine.

In addition to textiles, several research projects also explored crafting techniques for embedding electronics in paper by interconnecting electronic sticker-like components [12, 19] using copper [38, 40] or conductive inks [39]. With more advanced crafting techniques, circuits can also be embedded on small-scale polymer films [28] or silicone [32, 46, 47]. However, despite these efforts to streamline crafting processes, these techniques often remain tedious and error-prone. In contrast, SoftMod is a modular platform that bridges

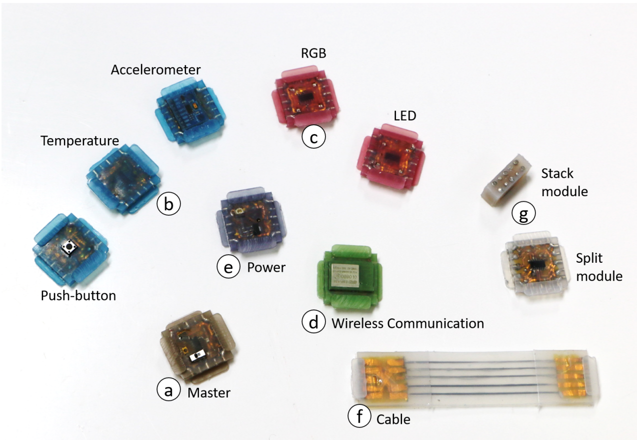

Figure 2: SoftMod supports six types of modules: (a) a master module, (b) input modules, (c) output modules, (d) wireless communication modules, (e) power modules, (f) cable modules, (g) adaptor modules.

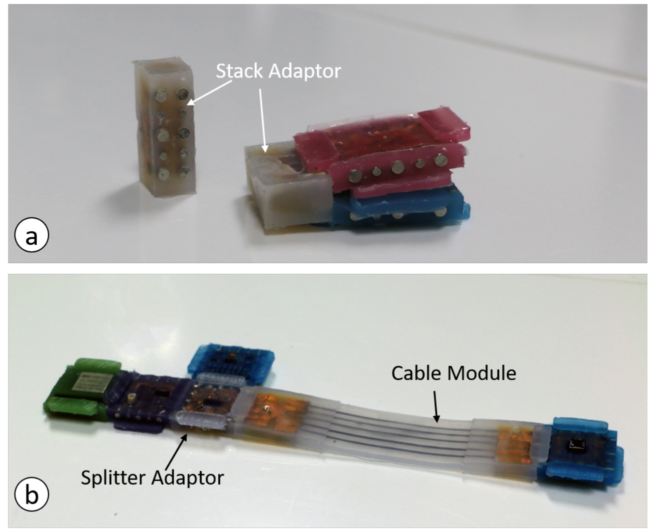

Figure 3: Two types of adaptor modules: (a) a stacking module, (b) a splitter module

the gap between crafting and construction kits. As our modules are soft and allow for making a wide variety of 2D and 3D shapes, they can be used to prototype early-stage wearable and textile interfaces that traditionally would require advanced crafting techniques.

4 SOFTMOD: SOFT PLUG-AND-PLAY MODULES

Supported Modules

Every interconnected SoftMod assembly consists of a single master module and slave modules. While slave modules embed electronic functionalities, the master module controls the behavior of the slave modules and can be programmed. All modules have the same dimensions of 30x30x5mm. As shown in Figure 2, the current version of the SoftMod system consist of (a) a master module (brown), and six types of slave modules: (b) input modules (blue), including a temperature module, push-button module, and accelerometer module; (c) output modules (red), including a standard LED module and RGB module; (d) wireless communication modules (green); (e) power modules (purple); (f) two cable modules (transparent-white); and (g) two adaptor modules (transparent-white). While exactly one master module needs to be present in any SoftMod assembly, up to 124 slave modules can be controlled by a single master module.

All modules, except the cable modules, embed four magnetic connectors to realize flexible 2D shapes. Cable modules shown in Figure 2f, embed two magnetic connectors. They do not add functionality to the assembly but allow for configuring modules in intricate shapes beyond 2D grid layouts. SoftMod supports cable modules of different lengths. Each cable module embeds two magnetic connectors. In contrast to other types of slave modules, cable modules are fully stretchable, in addition to being soft and flexible, as they consist of a liquid conductor (Section 6). SoftMod also comes with two adaptor modules that allow for (1) stacking of modules (Figure 3a) and (2) branching a single magnetic connector into three connectors (Figure 3b). The latter adaptor split module and has the same shape as the slave modules. We noticed, stacking of modules is especially useful for components that do not require direct visual attention or user interactions, such as the master module, power modules, accelerometer modules, and wireless communication modules. Stacking these modules oftentimes facilitates making specific shapes, such as interactive armbands that fit the arm.

SoftMod Topology

To accurately track the topology of an assembly, every module detects and identifies its direct neighbors connected to the top, left, right, and bottom side. This information is gathered by the master module which builds up a topology graph of all connected modules. Topology information is used for the automated assignment of default behavior as well as for offering an accurate real-time visualization of interconnected modules in the IDE to facilitate programming user-specified behavior (see Section 4).

SoftMod Behavior

SoftMod supports two techniques for defining the behavior of an assembly of modules:

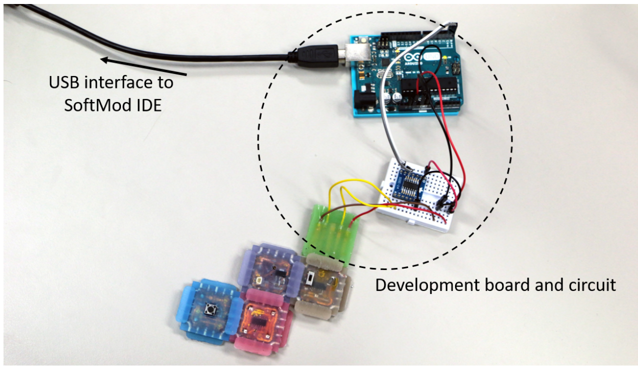

Figure 4: The master module is connected via an additional development board and circuit to a desktop computer running the SoftMod IDE.

- Default behavior: The master is configured to assign a default behavior to the slave modules based on their geometric topology. This mode makes best use of the plug-and-play nature of SoftMod as modules connected to the assembly immediately demonstrate some functionality. With current default behavior an output module will turn on when controlling neighboring input modules. More advanced and intelligent default behaviors can be supported in the future similar to LittleBits [5].
- User-specified behavior: The master module is connected to a desktop computer through an additional development board and circuit that connects to the master module using a magnetic connector Figure 4. This additional development board offers a USB interface for serial communication with the desktop. As shown in Figure 1a, the geometric topology of the assembly is visualized in real-time in the SoftMod IDE. The SoftMod IDE also allows end-users to specify the behavior a SoftMod assembly using basic IF-This-Then-That (IFTTT) rules. During this procedure, the user selects an input and output module after which they are linked with a trigger-action rule. Selecting modules and components is convenient as the geometric topology of the SoftMod assembly is exactly rendered in the IDE. As such, a one-to-one mapping exist between the tangible and digital representation. Alternatively, more advanced behavior is specified using S4A scratch-like programming environment [1] or by writing C-code in the Arduino IDE. User-specified code is then uploaded to the master module.

These two modes allow for using SoftMod both as a basic plug-and-play kit as well as an modular programmable electronic kit. Once user-specified behavior is loaded on a master module, the user can simply switch back to the default behavior by pressing the button on the master module for five seconds.

[1]http://s4a.cat

5 ARCHITECTURE AND ELECTRONIC DESIGN

Magnetic Connector Design

Figure 5a shows our connector design and the five electrodes it embeds: VCC (3.3V), SDA (serial data), trigger, SCL (serial clock), and ground. To keep the modules as small as possible, the five electrodes also serve as magnetic connectors to easily attach modules. As shown in Figure 5a, connectors consist of three 3x3mm and two 2x2mm neodymium disc magnets. This provides enough magnetic strength to bend our soft modules up to 90 degrees without electrically disconnecting the modules (Figure 6a). All modules have a hardness of Shore A 5 which is one of the softest silicones. The thickness of the cable modules (3mm) allows for stretches up to 27% before the magnetic connector releases (Figure 6b). Connectors are Poka-Yoke constrained to ensure modules only connect when properly oriented. This is key to the plug-and-play nature of our modules and avoid modules not functioning correctly or short circuits. Poka-Yoke constraints however require a careful design especially when connectors consist of soft silicone and thus can stretch. Figure 5b shows how a connector of a module has a magnetic attraction to two connectors of another module and a magnetic repulsion from the two other connectors. Additionally, some of the magnetic electrodes in a connector are swapped to protect modules that are not properly aligned to snap together. All modules come with an additional hood to avoid connecting a module upside down. Even when users would enforce such inappropriate connections, the connectors are designed to prevent short circuits. The spacing of the magnets and their pole layout also ensures that stretching a module cannot cause misalignment of pins.

Electronic Components

All modules, including the programmable master module embed an ATtiny1614 microcontroller operating at 10MHz. We choose this microcontroller as it is a low-cost (0.55 euros) and low-power (3.1 mA) microcontroller with a sufficient number of GPIO pins (12 pins) to control our modules' features. Even though the battery and link modules do not provide computational functionality, they also embed a microcontroller in order to track their position in a SoftMod assembly (see Section 5). Our wireless communication modules embed an ESP8266 chip as its cheap, small, and uses the IEEE 802.11 standard which allows SoftMod to communicate with a wide variety of existing devices. All modules run on 3.3v offered by power modules embedding a 190 mAh LiPo battery. Multiple power modules can be present in a single SoftMod assembly to offer more power or extend the assembly's lifetime without recharging. Figure 7 lists the power consumption of all modules when active. SoftMod's IDE uses this knowledge

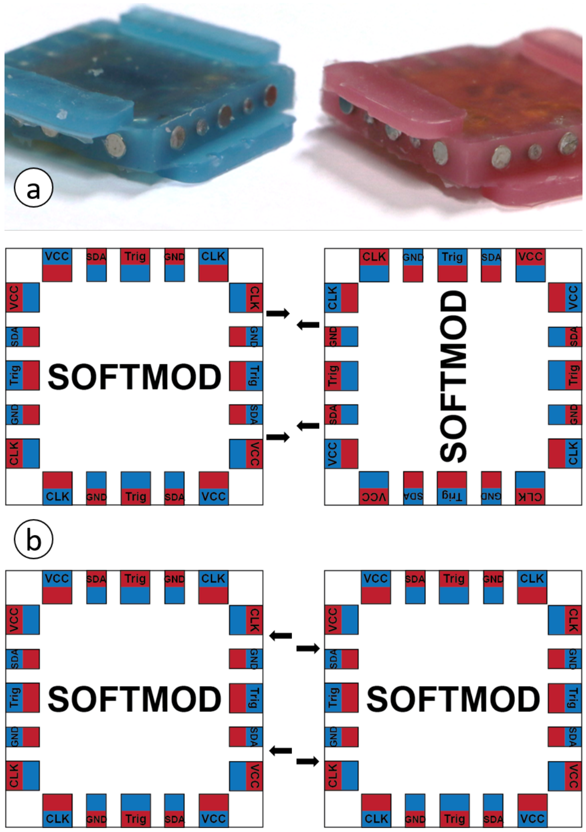

Figure 5: (a) SoftMod's magnetic connectors embed magnets that (b) attract two connectors of another module and repel from two other connectors of that module.

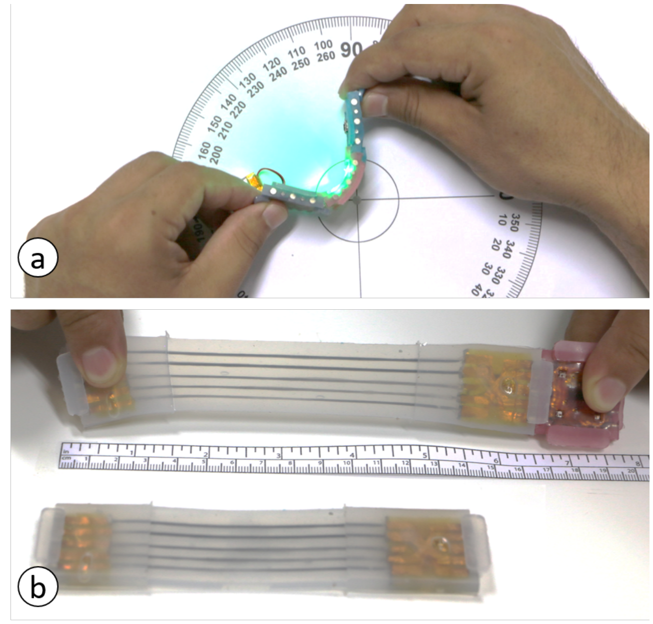

Figure 6: The magnetic connectors are strong to ensure electrical conductivity while (a) bending and (b) stretching modules.

Temperature module	8.1 mA	RGB module	45 mA
Push-button module	8 mA	Wireless communication module	85 mA
Accelerometer module	6.2 mA	Cable and adaptor modules	3.1 mA
LED module	13 mA		

Figure 7: The power consumption of all modules

to estimate the lifetime of an assembly and inform the user how many power modules to connect.

Communication and Protocols

Unlike LittleBits [5] and MakerWear [22], the behavior of a module is not hard-coded on that module but controlled by the master module based on the physical configuration of modules, or the user specified behavior in the master module (Section 4). Therefore the master module needs to control, and thus be able to address, slave modules and track their topology.

Addressing Slave Modules

A SoftMod master module controls connected slave modules over an I2C bus (SDA and SCL electrodes of the magnetic connector). We choose this bus standard over popular alternatives, such as SPI and 1-wire as it respectively requires less pins/lines and has superior transmission rates. The I2C standard uses 7-bit addressing space and thus supports uni-cast addressing of maximum 127 modules on a single bus. While this would be enough for making advanced prototypes with SoftMod, hard-coding the I2C address in every module would result in duplicate addresses when combining different SoftMod sets. One solution would be to broadcast all messages and add the ATTiny's 64-bit unique identifier as data. However, as I2C transmits data byte per byte, transmitting one byte of data would first require transmitting a 4-byte address. Therefore, we implemented a dynamic address allocation technique inspired by the ARP protocol of SMBus. This technique, which we explain below, dynamically assigns a 7-bit I2C address to slave modules connected to the master module (directly or through another slave module). The master module ensures that the 7-bit address for every slave module is unique within a SoftMod assembly.

When powered, slave modules have I2C address 127 (ARP address). The master module continuously sends a read request to this ARP address and immediately receives a response from new slave modules. Before the master module can assign unique I2C addresses to the new slave modules,

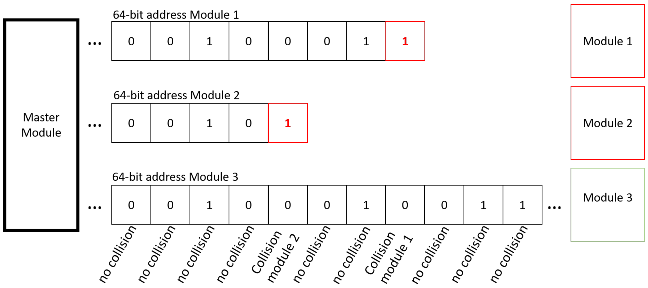

Figure 8: SoftMod uses a dynamic address allocation technique to assign an 7-bit I2C address to all modules. During this procedure, a single slave module is selected by having all modules transmit their 64-bit ATTiny address over the I2C bus while listening for collisions.

it needs to address every new slave module individually. Therefore, the master module sends out a request to initiate a self-selection strategy on the ARP address. Slave modules that receive this request, all start transmitting their ATTiny 64-bit unique identifier over I2C by pulling the data line (SDA) high or low every clock cycle (SCL) as shown in Figure 8. When a slave module detects a collision, by reading that the data line is high, after pulling it to ground in the same clock cycle, it stops transmitting its identifier. After 64 clock cycles, only one new slave module could transmit the entire unique identifier and accepts the unique 7-bit I2C address that the master hands-out right after 64 cock cycles. Finally, the slave module replies to the master module and communicates its module type. This strategy continues until there are no slave modules answering to the ARP address. As four I2C addresses are reserved (ARP, broadcast, and master module address), 124 addresses are available for slave modules.

Topology Tracking

When all new slave modules have a unique I2C address, the topology of the SoftMod assembly is updated. Therefore, the master module transmits four read requests to all new slave modules to retrieve the addresses and the orientation of the four connected neighboring modules. For one module to request the address of a neighbor module, it cannot use the I2C lines as this bus transmits the request to all modules in the assembly. Therefore a fifth electrode, the trigger, is present in the magnetic connector. This electrode directly connects a digital pin of one module's microcontroller to a digital pin of a neighboring module's microcontroller. As there are four edges on a module, four digital pins on every microcontroller are reserved for trigger electrodes (Figure 9).

Neighboring modules transmit their 7-bit I2C address, plus an additional two bits to identify the connector (orientation

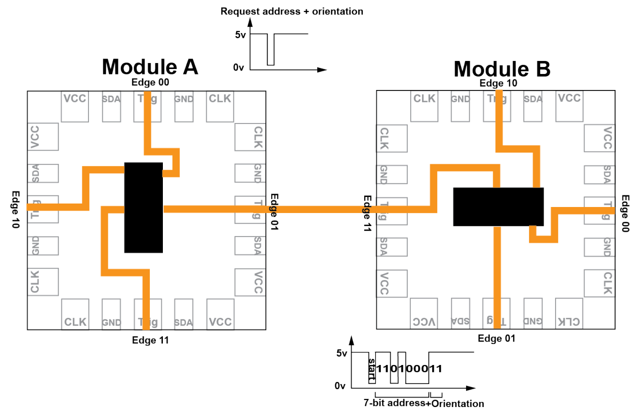

Figure 9: Slave modules identify neighbors and their orientation by transmitting the 7-bit address and connector code over the trigger electrode.

of the module), over this trigger electrode by pulling it high or low for all 9 bits. In contrast to I2C, no clock is present and the transmission is strictly time-based. To correctly read and write all bits, both modules go into an atomic state during which they cannot be interrupted by other requests. Figure 9 illustrates this process: (1) when module A needs the address and orientation of a neighbor module B, module A generates an interrupt at module B by pulling the trigger from high to low for 5ms after which module A goes into the atomic state and is ready to read bits. (2) From the moment module B is idle and detects this interrupt, it goes into the atomic state and transmits a start signal by pulling the trigger low for 5ms. When no start signal is received after a 100ms timeout at module A, the process stops and module A reports to the master module that no neighbor is attached at one edge. (3) When both modules are in the atomic state, module B transmits the 9-bits by pulling the trigger low or high every 5ms while module A reads at the same interval. (4) After the transmission is finished, module A transmits all 9-bits to the master module over I2C which in turn updates the topology accordingly.

Controlling Slave Modules

Assigning a unique I2C address to all new slave modules and updating the topology takes a few 100 milliseconds depending on the number of new modules and the complexity of the topology. After this automated initialization process, the master modules proceeds executing the default or user-specified behavior. During this process, it continuously reads updates from input modules and updates the state of output modules whenever needed. These messages are transmitted over I2C.

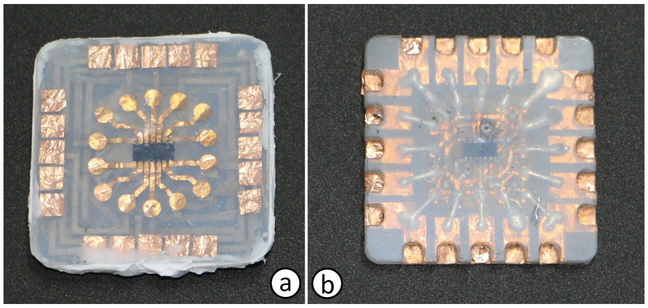

Figure 10: SoftMod's failed attempts: (a) Galinstan used for all traces, (b) Galinstan used for traces between PCB and connectors.

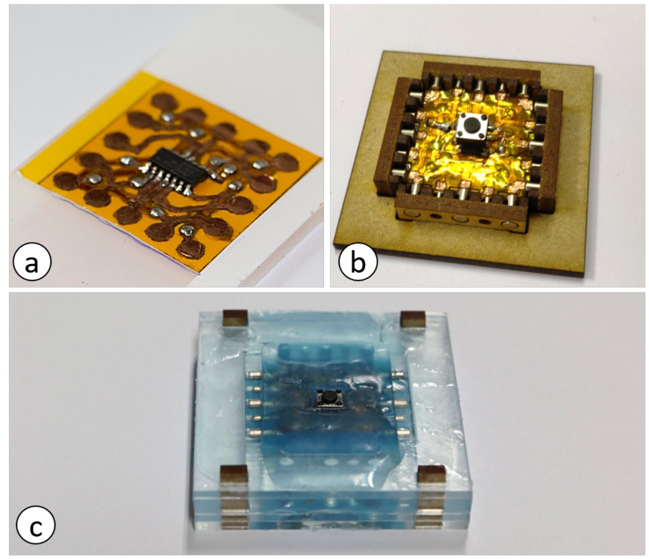

Figure 11: SoftMod's manufacturing process: (a) creating flexible PCB's using a CO2 and fiber laser cutter, (b) precisely soldering magnets using a fixture, (c) casting the silicone using a mold.

6 PRODUCING SOFT MODULES

SoftMod is the first soft modular electronic kit. Therefore we could not use standard manufacturing techniques and assembly lines that are used for making rigid PCBs and connectors. Instead, we adopted and extended state-of-the-art DIY fabrication procedures [24, 32, 43] for making our soft modules, including the electronics design, flexible form factor, magnetic connectors, as well as stretchable cable modules. As a result, the SoftMod electronic kit can be replicated and extended by the DIY and maker community with machinery available in many FabLabs and makerspaces. This section provides details and insights on our DIY production procedure and gets readers started with making Soft Modules. Technical drawings will also be made available online to facilitate replication.

Initially, our goal was to make all modules stretchable using liquid conductors (i.e. Galinstan), according to the procedure of Nagels et al. [32]. However, as this is mostly a manual process, it was too tedious and error-prone for making a set of modules. Figure 10 shows some results of these attempts. Therefore, we only made the cable modules entirely stretchable using this procedure. As an alternative to liquid metal, laser patterning a meander structure in copper could be used as a stretchable conductor [6]. However, since copper traces are fully enclosed in silicone they can not move freely and therefore could break easily.

All other modules embed a flexible PCB (FPCB) that interconnect the majority of electronic components. These flexible PCBs, consisting of copper and kapton tape, are produced using a laser cutter machine supporting both a fiber and CO2 laser, according to the process of Lambrichts et al. [24]. Figure 11a shows the three-layered flexible PCB for the LED module.

The FPCBs are embedded in silicone together with the magnetic connectors to realize a soft and flexible module. This process starts by soldering magnets and electronic components to the FPCB. Soldering the magnets to the FPCB requires significant precision to maintain the spacing between magnets and respect the distance between the edge of a module and the magnets. To facilitate the soldering process, a fixture was laser cut to precisely positioning the magnets with respect to the FPCB as shown in Figure 11b. This fixture also houses additional magnets to ensure the small connector magnets do not loose or switch polarity as a result of the heat of the soldering process. The magnets used have a Curie Point of 310°C at which they which they irreversible lose their magnetism. Key to soldering magnets is to solder as fast as possible to keep the temperature well below that point. An additional stronger magnet is embedded in the fixture as an additional force field to protect the magnetic properties of the smaller magnets while also acting as a heat sink.

During our DIY production experiments, we also tested laser cut and 3D printed connectors to hold the magnets precisely in place. We noticed, however, that such connectors limited the flexibility of modules. In addition, 3D printing or laser cutting connectors was too time-consuming and fragile even for small-scale fabrication.

To precisely cast the electronics design in the shape of our soft modules using silicone, we produced a mold by stacking laser cut acrylic parts as shown in Figure 11c. This molding approach, inspired by StackMold [43], holds all parts in place while it is filled up with casting silicone Shore A5. This silicone has a low viscosity and ensures all gaps between electronic components and magnets are filled during the casting process.

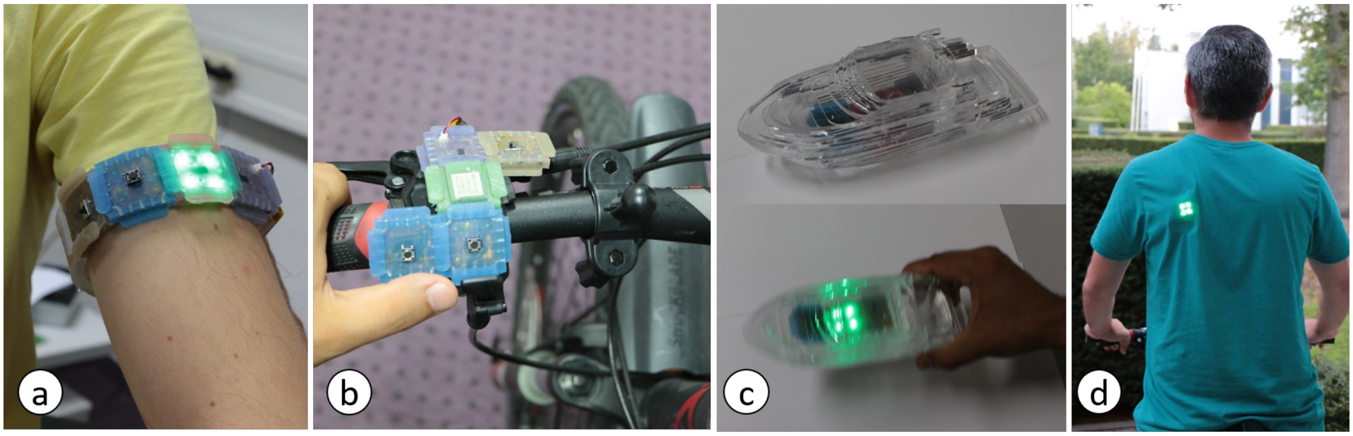

Figure 12: SoftMod allows for prototyping various novel types of interfaces: (a) an armband, (b) enriching the handle bar of a bike with blinker controls, (c) embedding interactivity in fabricated objects, (d) blinkers embedded in a shirt.

As described in the sections above, we used only DIY fabrication techniques and machinery to speed up our design and research iterations. As a side effect this resulted in a toolkit for and by makers. As a result, we will make an Instructable[2] available online for DIY enthousiasts interested in building their own SoftMod toolkit, variations, or extensions. As making SoftMods requires is tedious and requires experience with PCB making, soldering, and silicone molding, we also envision this toolkit to become commercially available to children and people outside the DIY community.

7 EXAMPLE DESIGNS AND USE CASES

SoftMod assemblies are easy to change in shape using plug-and-play magnetic connectors. Additionally, the variety of programmable features allows for specifying the desired behavior. Therefore, SoftMod stimulates design explorations and allows users to materialize and test their ideas in short design iterations.

As our modules support making a wide variety of shapes, they can augment objects with interactivity. To transition SoftMod assemblies to more permanent and robust electronic systems, we experimented with casting an additional layer of DIY casting silicone and dissolvable beeswax over top of the modules, to reinforce or attach SoftMod assemblies permanently or temporarily. Such techniques allow for moving low-fidelity modular designs to high-fidelity electronic systems. Below, we discuss a number of use cases and example designs that SoftMod enables.

Curiosity, Play, and Experimentation

The plug-and-play magnetic connectors make it convenient to snap together modules and make a working electronic system without training. Therefore, it empowers children and adults to play and thinker with sensors without risking any damage to components or one's safety.

Wearables

SoftMods are flexible and allow for connecting modules in a loop. As shown in Figure 12a, this enables prototyping of interactive bands that fit, for example, the chest, arm, or wrist. Such bands are easy to wear and take off using the magnetic connectors or the cable modules that allow for stretching over the body. When lots of modules are required, stacking modules can help to make small bands. In contrast to commercially available interactive bands, such as health monitoring wristbands, SoftMod wearables have a personalized functionality and shape and are easy to reconfigure, as demonstrated in the Walkthrough (Section 2).

Enriching Existing Objects

By attaching SoftMod assemblies to existing objects, they can alter, extend, or add interactivity to objects. Our soft modules attach to objects with various curves using tape, fasteners, or Velcro. Figure 12b shows the handlebar of a bike augmented with two buttons that control blinkers on the biker's back (Figure 12d). The blinking LEDs are also made with SoftMod and intercommunicate with the buttons using SoftMod's wireless communication module.

Making new Interactive Objects

As SoftMod supports making interactive 2D and 3D shapes, they can be embedded inside a fabricated object to add interactivity. Figure 12c shows an example of a laser cut acrylic object embedding a SoftMod assembly. In this example, embedded LEDs light up when shacking the object. Alternatively, a SoftMod assembly can be inserted in a laser cut mold

[2]www.instructables.com

structure that is then filled with casting material to realize more permanent interactive forms, similar to StackMold [43].

Interactive Skin and Textile

Our soft modules also conform well to skin and textiles and are therefore suitable for prototyping interactive tattoos and clothes. Best results are achieved when casting a thin layer of DIY casting silicone on top the modules to adhere it to skin or textiles. This also ensures the magnetic connectors are sealed and thus waterproof. Figure 12d shows an example in which an LED, wireless communication, and master module are embedded in a T-shirt to realize blinkers while riding the bike. Traditionally, prototyping interactive tattoos and textiles, require lots of materials, machinery, and DIY knowledge [16, 46]. SoftMod makes prototyping low-fidelity versions of these novel interfaces available to true novices.

8 LIMITATIONS AND FUTURE WORK

SoftMod has three limitations, which we hope to address in future versions of our system:

First, while the SoftMod IDE supports specifying basic If-This-Then-That (IFTTT) behavior, more advanced logic requires composing code in scratch-like programming environments supporting Arduino (i.e. S4A) or the Arduino IDE. These programming paradigms are still quite challenging for novices, especially for distributed systems communicating over WiFi. Therefore, future versions of SoftMod could support techniques that further lower the barrier for end-users to specify sensor behavior, such as IBM NodeRed [20], Pulsation [39] or Trigger-Action-Circuits [2].

Second, a modular kit can always be extended and upgraded with more modules. For example, cable modules can be extended with deflection sensing to offer new modalities for input and further increase the accuracy of the 3D assembly reconstruction in SoftMod's IDE. The wireless communication module can become flexible when thin-film flexible antennas become available. The power module could embed a capacitive pad and three LEDs for users to check the battery status and recharge them in time. SoftMod could also be extended with entirely new modules, such as modules for actuation and advanced sensing.

Third, all modules embed a ATtiny1614 microcontroller for processing. Although this is sufficient for all slave modules as they only run the software framework for communication, the master module stores the topology, the default behavior, as well as the user defined behavior. This could exceed the 2kb of RAM of the ATtiny1614. In future versions, we plan to use a more powerful microcontroller for the master module, such as the Microchip SAM L10 that embeds a Cortex-M23 CPU and has 16kb of RAM. This is especially useful for more advanced end-user specified behavior as these programs are oftentimes less efficient in terms of memory consumption.

Last, to evaluate the usability and utility of SoftMod for end-users we plan to conduct a user evaluation to assess the ease of use of our modules and the user programmed behavior. In addition, organizing a larger workshop with designers, artists, children, and makers could reveal new use-cases and applications for SoftMod.

9 CONCLUSION

In this paper, we presented SoftMod, a modular electronics kit consisting of soft and flexible modules that snap together for prototyping novel interactive systems. SoftMod offers qualities for both children and adults as it comes with simple plug-and-play behavior as well as advanced user-specified behavior. In this paper, we contributed and detailed our software framework, electronic design, mechanical connector design, and prototyping procedures for making SoftMods. We hope our toolkit empowers designers, researchers, and artists, to explore new ideas for electronic systems, including wearables, interactive skin and textiles. As the process to make our soft modules is entirely DIY compatible, we hope the community builds further on our ideas and joints our efforts to build easy-to-use electronic toolkits.

ACKNOWLEDGMENTS

This research was supported by the Research Foundation - Flanders (FWO), project G0E7317N End-User Development of Intelligible Internet-of-Things Objects and Applications.

REFERENCES

[1] [n. d.]. Printoo: Paper-Thin, Flexible Arduino-Compatible modules! https://www.kickstarter.com/projects/1030661323/printoo-paper-thin-flexible-arduinotm-compatible-m

[2] Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2017. Trigger-Action-Circuits: Leveraging Generative Design to Enable Novices to Design and Build Circuitry. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 331–342. https://doi.org/10.1145/3126594.3126637

[3] Arduino. 2019. Arduino Home. (2019). Retrieved August 8 2019, from: www.arduino.cc.

[4] Autodesk. 2019. Arduino Home. (2019). Retrieved August 8 2019, from:https://ifttt.com/.

[5] Ayah Bdeir and Ted Ullrich. 2011. Electronics As Material: LittleBits. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 341–344. https://doi.org/10.1145/1935701.1935781

[6] Shantonu Biswas, Johannes Reiprich, Joerg Pezoldt, Matthias Hein, Thomas Stauden, and Heiko O. Jacobs. 2018. Stress-adaptive meander track for stretchable electronics. *Flexible and Printed Electronics* 3, 3, 032001. https://doi.org/10.1088/2058-8585/aad583

[7] L. Buechley and M. Eisenberg. 2008. The LilyPad Arduino: Toward Wearable Engineering for Everyone. *IEEE Pervasive Computing* 7, 2 (April 2008), 12–15. https://doi.org/10.1109/MPRV.2008.38

[8] L. Buechley, N. Elumeze, C. Dodson, and M. Eisenberg. 2005. Quilt Snaps: a fabric based computational construction kit. In *IEEE International Workshop on Wireless and Mobile Technologies in Education*

(WMTE'05). 3 pp.–221. https://doi.org/10.1109/WMTE.2005.55

[9] Alvaro Cassinelli and Daniel Saakes. 2017. Data Flow, Spatial Physical Computing. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*. ACM, New York, NY, USA, 253–259. https://doi.org/10.1145/3024969.3024978 event-place: Yokohama, Japan.

[10] Marina Conde. 2019. Scratch for ARduino. (2019). Retrieved August 8 2019, from: http://s4a.cat/.

[11] Susan Cotterell, Ryan Mannion, Frank Vahid, and Harry Hsieh. 2005. eBlocks: An Enabling Technology for Basic Sensor Based Systems. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*. IEEE Press, Piscataway, NJ, USA, Article 58. http://dl.acm.org/citation.cfm?id=1147685.1147754

[12] Natalie Freed, Jie Qi, Adam Setapen, Cynthia Breazeal, Leah Buechley, and Hayes Raffle. 2011. Sticking Together: Handcrafting Personalized Communication Interfaces. In *Proceedings of the 10th International Conference on Interaction Design and Children (IDC '11)*. ACM, New York, NY, USA, 238–241. https://doi.org/10.1145/1999030.1999071

[13] Matthew G. Gorbet, Maggie Orth, and Hiroshi Ishii. 1998. Triangles: tangible interface for manipulation and exploration of digital information topography. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '98*. ACM Press, Los Angeles, California, United States, 49–56. https://doi.org/10.1145/274644.274652

[14] M. M. Gorlick. 1999. Electric suspenders: a fabric power bus and data network for wearable digital devices. In *Digest of Papers. Third International Symposium on Wearable Computers*. 114–121. https://doi.org/10.1109/ISWC.1999.806684

[15] 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 (UIST '01)*. ACM, New York, NY, USA, 209–218. https://doi.org/10.1145/502348.502388

[16] Nur Al-huda Hamdan, Simon Voelker, and Jan Borchers. 2018. Sketch&Stitch: Interactive Embroidery for E-textiles. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 82, 13 pages. https://doi.org/10.1145/3173574.3173656

[17] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective Physical Prototyping Through Integrated Design, Test, and Analysis. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 299–308. https://doi.org/10.1145/1166253.1166300

[18] Steve Hodges, James Scott, Sue Sentance, Colin Miller, Nicolas Villar, Scarlet Schwiderski-Grosche, Kerry Hammil, and Steven Johnston. 2013. .NET Gadgeteer: A New Platform for K-12 Computer Science Education. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*. ACM, New York, NY, USA, 391–396. https://doi.org/10.1145/2445196.2445315

[19] 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 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1743–1746. https://doi.org/10.1145/2556288.2557150

[20] IBM. 2019. Node-Red: Flow-based programming for the Internet of Things. (2019). Retrieved August 8 2019, from: https://nodered.org/.

[21] Wang Jianjun. 2019. Makeblock Co., Ltd. (2019). Retrieved August 8 2019, from:https://www.makeblock.com/.

[22] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. 2017. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 133–145. https://doi.org/10.1145/3025453.3025887

[23] Kristof Van Laerhoven, Albrecht Schmidt, and Hans-Werner Gellersen. 2002. Pin&Amp;Play: Networking Objects Through Pins. In *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp '02)*. Springer-Verlag, Berlin, Heidelberg, 219–228. http://dl.acm.org/citation.cfm?id=647988.741497

[24] Mannu Lambrichts, Jose Maria Tijerina Munoz, Tom De Weyer, and Raf Ramakers. 2020. DIY Fabrication of High Performance Multi-Layered Flexible PCBs *(Submitted to TEI '20 Extended Abstract/Poster)*. New York, NY, USA.

[25] Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 471–479. https://doi.org/10.1145/3126594.3126643 event-place: QuÃľbec City, QC, Canada.

[26] Lego. 2019. Lego Mindstorm. (2019). Retrieved August 8 2019, from:https://www.lego.com/en-us/mindstorms.

[27] David I Lehn, Craig W Neely, Kevin Schoonover, Thomas L Martin, and Mark T Jones. 2004. e-TAGs: e-textile attached gadgets. (2004).

[28] Joanne Lo and Eric Paulos. 2014. ShrinkyCircuits: Sketching, Shrinking, and Formgiving for Electronic Circuits. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 291–299. https://doi.org/10.1145/2642918.2647421

[29] David A. Mellis, Sam Jacoby, Leah Buechley, Hannah Perner-Wilson, and Jie Qi. 2013. Microcontrollers As Material: Crafting Circuits with Paper, Conductive Ink, Electronic Components, and an "Untoolkit". In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 83–90. https://doi.org/10.1145/2460625.2460638

[30] David Merrill, Emily Sun, and Jeevan Kalanithi. 2012. Sifteo Cubes. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 1015–1018. https://doi.org/10.1145/2212776.2212374 event-place: Austin, Texas, USA.

[31] Amon Millner and Edward Baafi. 2011. Modkit: Blending and Extending Approachable Platforms for Creating Computer Programs and Interactive Objects. In *Proceedings of the 10th International Conference on Interaction Design and Children (IDC '11)*. ACM, New York, NY, USA, 250–253. https://doi.org/10.1145/1999030.1999074

[32] Steven Nagels, Raf Ramakers, Kris Luyten, and Wim Deferme. 2018. Silicone Devices: A Scalable DIY Approach for Fabricating Self-Contained Multi-Layered Soft Circuits Using Microfluidics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 188, 13 pages. https://doi.org/10.1145/3173574.3173762

[33] Grace Ngai, Stephen C.F. Chan, Joey C.Y. Cheung, and Winnie W.Y. Lau. 2009. The TeeBoard: An Education-friendly Construction Platform for e-Textiles and Wearable Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 249–258. https://doi.org/10.1145/1518701.1518742

[34] Grace Ngai, Stephen C.F. Chan, Vincent T.Y. Ng, Joey C.Y. Cheung, Sam S.S. Choy, Winnie W.Y. Lau, and Jason T.P. Tse. 2010. I*CATch: A Scalable Plug-n-play Wearable Computing Framework for Novices and Children. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 443–452. https://doi.org/10.1145/1753326.1753393

[35] Hannah Perner-Wilson, Leah Buechley, and Mika Satomi. 2011. Handcrafting Textile Interfaces from a Kit-of-no-parts. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 61–68. https://doi.org/10.1145/1935701.1935715

[36] Raspberry pi. 2019. Raspberrypi Home. (2019). Retrieved August 8 2019, from: www.raspberrypi.org.

[37] Jie Qi and Leah Buechley. 2010. Electronic Popables: Exploring Paper-based Computing Through an Interactive Pop-up Book. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '10)*. ACM, New York, NY, USA, 121–128. https://doi.org/10.1145/1709886.1709909

[38] Jie Qi and Leah Buechley. 2014. Sketching in Circuits: Designing and Building Electronics on Paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1713–1722. https://doi.org/10.1145/2556288.2557391

[39] 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)*. ACM, New York, NY, USA, 2457–2466. https://doi.org/10.1145/2702123.2702487

[40] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: Fabricating Custom Capacitive Touch Sensors to Prototype Interactive Objects. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 579–588. https://doi.org/10.1145/2380116.2380189

[41] Eric Schweikardt and Mark D. Gross. 2006. roBlocks: A Robotic Construction Kit for Mathematics and Science Education. In *Proceedings of the 8th International Conference on Multimodal Interfaces (ICMI '06)*. ACM, New York, NY, USA, 72–75. https://doi.org/10.1145/1180995.1181010

[42] Becky Stern and Tyler Cooper. 2015. *Getting started with Adafruit FLORA: making wearables with an Arduino-compatible electronics platform.* Maker Media, Inc.

[43] 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 (UIST '19)*. ACM, New York, NY, USA, 687–699. https://doi.org/10.1145/3332165.3347915 event-place: New Orleans, LA, USA.

[44] Nicolas Villar, Florian Block, Dave Molyneaux, and Hans Gellersen. 2006. VoodooIO. In *ACM SIGGRAPH 2006 Emerging Technologies (SIGGRAPH '06)*. ACM, New York, NY, USA, Article 36. https://doi.org/10.1145/1179133.1179170

[45] Ryoichi Watanabe, Yuichi Itoh, Masatsugu Asai, Yoshifumi Kitamura, Yoshifumi Kitamura, Fumio Kishino, and Hideo Kikuchi. 2004. The Soul of ActiveCube: Implementing a Flexible, Multimodal, Three-dimensional Spatial Tangible Interface. *Comput. Entertain.* 2, 15–15. https://doi.org/10.1145/1037851.1037874

[46] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2991–3000. https://doi.org/10.1145/2702123.2702391

[47] Jaeyoung Yoon, Yunsik Joo, Eunho Oh, Byeongmoon Lee, Daesik Kim, Seunghwan Lee, Taehoon Kim, Junghwan Byun, and Yongtaek Hong. 2019. Soft Modular Electronic Blocks (SMEBs): A Strategy for Tailored Wearable Health-Monitoring Systems. *Advanced Science* 6, 5 (2019), 1801682. https://doi.org/10.1002/advs.201801682 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.201801682