A Survey and Taxonomy of Electronics Toolkits for Interactive and Ubiquitous Device Prototyping

MANNU LAMBRICHTS, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE
RAF RAMAKERS, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE
STEVE HODGES, Microsoft Research, Cambridge, UK
SVEN COPPERS, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE
JAMES DEVINE, Microsoft Research, Cambridge, UK

Over the past two decades, many toolkits for prototyping interactive and ubiquitous electronic devices have been developed. Although their technical specifications are often easy to look up, they vary greatly in terms of design, features and target audience, resulting in very real strengths and weaknesses depending on the intended application. These less technical characteristics are often reported inconsistently, if at all. In this paper we provide a comprehensive survey of interactive and ubiquitous device prototyping toolkits, systematically analysing their characteristics within the framework of a new taxonomy that we present. In addition to the specific characteristics we cover, we introduce a way to evaluate toolkits more holistically, covering user needs such as 'ease of construction' and 'ease of moving from prototype to product' rather than features. We also present results from an online survey which offers new insights on how the surveyed users prioritize these characteristics during prototyping, and what techniques they use to move beyond prototyping. We hope our analysis will be valuable for others in the community who need to build and potentially scale out prototypes as part of their research. We end by identifying gaps that have not yet been addressed by existing offerings and discuss opportunities for future research into electronics prototyping toolkits.

CCS Concepts: • **Human-centered computing** → **User interface toolkits**.

Additional Key Words and Phrases: electronics prototyping platforms, toolkits, interactive devices, ubiquitous computing

ACM Reference Format:
Mannu Lambrichts, Raf Ramakers, Steve Hodges, Sven Coppers, and James Devine. 2021. A Survey and Taxonomy of Electronics Toolkits for Interactive and Ubiquitous Device Prototyping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 70 (June 2021), 24 pages. https://doi.org/10.1145/3463523

1 INTRODUCTION

The human-computer interaction and ubiquitous computing communities have a long tradition of prototyping novel electronic circuits for sensing and interaction. General-purpose computing devices like personal computers, tablets and smartphones are often not suitable for this type of work because they can't support the vision of truly ubiquitous computing and the internet of things. Although it's possible to develop and research a bespoke

Authors' addresses: Mannu Lambrichts, mannu.lambrichts@uhasselt.be, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE; Raf Ramakers, raf.ramakers@uhasselt.be, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE; Steve Hodges, steve.hodges@microsoft.com, Microsoft Research, Cambridge, UK; Sven Coppers, sven.coppers@uhasselt.be, Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media, BE; James Devine, t-jamdev@microsoft.com, Microsoft Research, Cambridge, UK.

2474-9567/2021/6-ART70 $15.00
https://doi.org/10.1145/3463523

embedded solution, often based on a custom microcontroller (MCU) circuit [110], this can be expensive, complex and/or time-consuming.

Thankfully, the advent of hardware development systems such as Arduino [11, 15] and Phidgets [50], has lowered the barrier for making new electronic circuits using MCUs. Over the past two decades, a large number of prototyping toolkits that facilitate the construction of electronic devices have appeared, both as research projects and commercial products. These have a wide range of technical specifications encompassing aspects such as processor speed, amount of memory and nature of supported input and output modalities. But in addition, these toolkits differ in terms of design, features and target audience, resulting in a diversity of user experiences and very real strengths and weaknesses depending on the intended application. In short, the toolkits often look, feel and work differently to each other. To give specific examples, some toolkits are based on a single 'development board' [105] while others are modular [52, 57]; several toolkits require skills in either programming [11] or electronics [59] while others require no technical expertise or additional equipment at all [86]; sometimes the resulting prototype can run standalone [11, 86, 105] but sometimes it relies on continuous connection to a computer [109]; some toolkits are generic in nature while others are particularly suited for building specific types of prototype such as smart textiles [24], paper-based interfaces [59], or robots [36]; and some toolkits are developed for specific user groups such as educators and students [58].

With such a wide variety of electronics prototyping toolkits available, it can be challenging for those new to the area to understand the range of user experiences offered by electronics toolkits. Several articles have classified and reflected on electronics prototyping toolkits that are targeted towards primary and secondary school (also known as K-12) educational settings [21, 42, 58]. But the objectives for educators and students who are building electronic devices are very specific: in the education context the artifact being constructed is a side-effect of the process, which is instead optimized for the multi-faceted learning experience it delivers [57]. In contrast, other audiences, including makers, professional engineers, artists, designers and researchers often use electronics toolkits to build working prototypes, either as useful one-offs, as a way to evaluate a concept in more detail, or as the first step in the process of developing a product [65].

In this paper, we present a literature survey and product summary of the broad gamut of electronics prototyping toolkits beyond the previously-reported K-12 systems, and develop a taxonomy the goes beyond technical features and specifications. In this sense, our work is different from existing online comparisons of commercial toolkits such as [78] because our broader set of characteristics provide a more holistic view of the prototyping process and the artifact that results. For practitioners who must build a device that can be deployed reliably, or want a prototype that can be replicated many times, including need-based characteristics is an important aid to toolkit selection.

Given the plethora of electronics prototyping solutions that exist, it was not feasible for us to examine them all. We have tried to provide a complete review of systems reported in the research literature, but our coverage of commercial systems is not comprehensive—for example, we have not included every MCU development board or every variant of Arduino. Instead, we have tried to include representative examples of each. In total we have labeled 56 electronics toolkits. Our taxonomy is, however, applicable beyond the specifics of this set and we believe it will also be valuable for discussing and comparing future generations of electronics toolkits.

We also report on the results of a formative online survey with 122 respondents which indicates the relevance of the characteristics in our taxonomy, and offers new insights into current prototyping practices including the need to scale to multiple copies of a prototype. This data reveals some needs currently unaddressed by the established electronics toolkit offerings that could form the basis for future research.

The main contributions of this work are three-fold: Firstly, we report extensively on the literature relating to electronics prototyping toolkits, and we include representative commercial examples. Secondly, we present a novel taxonomy for classifying and comparing these toolkits in new ways that we think provide a useful perspective to others in the research community who are using electronics toolkits and/or developing new ones.

We use this taxonomy to label the toolkits we have reviewed and the resulting dataset is available online for ease of consumption. And thirdly, we report on the results of an online survey that indicates the relevance of the characteristics in our taxonomy to the practitioners we surveyed, and highlights some common practices for building individual prototypes and scaling up to larger numbers. Collectively, we hope these contributions will provide a common ground for discussions between researchers in the field and will highlight opportunities for future research into toolkits that support the development of ubiquitous computing and interactive devices.

2 THE ESTABLISHED APPROACHES TO ELECTRONICS PROTOTYPING

Since the birth of the electronics industry, engineers have naturally sought ways to accelerate the design, test and construction of prototype electronic devices. As described above, these tools and techniques have subsequently transitioned to a broader audience of practitioners. Blikstein [22] offers a historic analysis of 30 years of developments in electronics prototyping toolkits and identifies three levels of abstraction. Here we build on Blikstein's categorization but include a fourth category. To avoid confusion we refer to 'Types' of electronics prototyping rather than Blikstien's 'Levels'.

2.1 Type 1: Prototyping with Discrete Components

Blikstein's 'Level 1' electronics is based on discrete components—the same components that are used in the mass production of electronic devices. In this paper we refer to this approach to prototyping as 'Type 1'. The key to using production-ready electronic components for prototyping is making the desired electrical connections without needing to solder them to a custom-designed printed circuit board (PCB). A common approach to achieve this is to push the leads of components into a solderless breadboard.

Mellis et al. [83] advocate for Type 1 prototyping since it offers form-factor freedom and is closest to current engineering practices. With a solderless breadboard it's also quick and easy to make changes as a design iterates. However, although it is relatively easy to physically plug and unplug components, making sure that the connections are exactly as planned can be fiddly and error-prone, and the resulting "rat's nest" of circuitry is often susceptible to damage.

In an effort to facilitate Type 1 electronics prototyping, product designers and researchers have built various tools. These either provide novel ways for connecting discrete components together, for example Snap Circuits [115], or they help novice engineers detect and diagnose problems [40]. In recent years there is also an increased interest in using novel fabrication technologies to interconnect discrete electronic components. Researchers have explored novel approaches ranging from conductive pens [59] and inkjet printing conductive traces on paper [62] to techniques for making soft [53] and stretchable [88] circuits.

Despite the innovations listed above, two significant drawbacks remain when using Type 1 prototyping. Firstly, if multiple copies of a circuit are needed, the entire artifact must be built from scratch each time which can be time consuming. Secondly, and often more significantly, prototyping with discrete components requires significant knowledge of electronics, slowing down the process of creating and iterating a prototype.

2.2 Type 2a: Integrated Microcontroller Development Boards

Rather than prototyping devices with discrete components only, many prototyping practices involve extending or interconnecting ready-made PCBs such as purpose-built modules and development boards. Blikstein [22] defines 'Level 2' prototyping based on the use of an integrated MCU development board, i.e. a PCB built around a particular MCU that houses supporting components to ensure correct operation and eases access to the most common features. In this paper we call these MCU development boards 'Type 2a' electronics.

Example Type 2a products include the Arduino Uno [11, 15], the STM32 discovery [119], Teensy [121], and the TI Launchpad [122]. These integrate components for power delivery, programming, communications and basic user interaction via regulators, pin headers, USB ports, push buttons and LEDs. Type 2a also encompasses more featured MCU development boards such as the micro:bit [14, 86], Raspberry Pi [105] and BeagleBone [18]. These devices embed networking, mass storage, additional sensors, and/or displays and are often referred to as single-board computers as they function as stand-alone devices.

Type 2a prototyping is well-established in the electronics profession because, compared to the Type 1 approach, it simplifies experimentation with MCUs. However, for many prototype electronic devices, the MCU is only half of the story: any given design typically requires integration with components that are not present on the ready-made MCU development board. This shortcoming can of course be addressed by combining Type 2a with Type 1—wiring the MCU development board to a solderless breadboard with the necessary additional components.

2.3 Type 2b: Breakout Boards and Wireless Modules

Another approach that has become popular in the past decade or two is the use of breakout boards and wireless modules. These are similar to MCU development boards, but are typically smaller and rather than featuring an MCU they contain a particular chip such as an accelerometer, regulator or WiFi radio along with enough components to support its operation. Blikstein [22] does not label these explicitly, but we believe they form an interesting category of their own which we refer to as 'Type 2b' electronics prototyping tools.

Specific examples of Type 2b products include Adafruit's Bosch BNO055 sensor breakout board[3] which embeds sensor fusion and an I2C interface, and ESP8266/ESP32 WiFi modules[46]. Several groups of Type 2b modules are worth a particular mention: Seeed's Grove system [52], MikroElektronika's Click boards [30]; Digilent's Pmod modules [99]; SparkFun's Qwiic modules [117] and Adafruit's STEMMA and STEMMA QT [8]. These are all ecosystems of modules that are somewhat compatible with each other, making it relatively easy to interface several of them to a given MCU or to move between modules with different functionalities.

As with Type 2a components, a critical aspect of Type 2b components is that they are not sufficient to build a prototype; you need to combine them with Type 2a and/or Type 1 electronics. Although both Type 2a and Type 2b products may support common protocols like I2C, SPI and UART, they are typically manufactured by different companies and their operating voltage, physical connections and/or communications speeds and protocols are not necessarily compatible. Therefore selecting and interfacing components appropriately still requires a good understanding of electronics.

In summary, the main advantage of Type 2 electronics prototyping over Type 1 is speed and robustness—because fewer connections have to be specified and hand-made. The biggest disadvantage is reduced flexibility.

2.4 Type 3: Integrated Modular Systems

Consistent with Blikstein's 'Level 3', our definition of 'Type 3' electronics prototyping covers toolkits that consist of a complete set of modules specifically designed to work together without needing any other components. They often comprise a processing module combined with various input and output (I/O) modules. Many Type 3 platforms offer keyed connectors with poka-yoke constraints, or use modules that communicate wirelessly, to make the assembly fool-proof. Popular examples include .NET Gadgeteer [57], Phidgets [50], littleBits [17], LEGO Mindstorms [72] and SAM Labs [109].

Type 3 platforms simplify and expedite electronics prototyping compared to Type 2 because there is no need to find third party compatible components and work out how to connect them. Blikstein [22] noticed that Type 2 electronics expose more details of embedded electronic components compared to Type 3. For these reasons, Type 3 platforms are frequently used in educational and leisure settings as they allow users to get up to speed and experience success without knowing many details of the underlying electronics. However, this also means an important factor when selecting a Type 3 platform is the particular set of modules it supports; adding discrete components or building custom modules is often hard.

2.5 Bridging Between Types

As indicated above, it's possible to bridge between Types when prototyping. For example, it may be necessary to combine a Type 1 solderless breadboard with a Type 2a development board if suitable Type 2b modules are not available. Also, when the popular Type 2a systems Arduino, Raspberry Pi and BeagleBone are used with shields, HATs and capes (respectively), they effectively move into Type 3. The same is true for some microcontroller development boards that incorporate sockets for Gadgeteer or Click modules.

3 IDENTIFYING AND REVIEWING THE LITERATURE

Having provided a broad overview of the different approaches to electronics prototyping, in this section we describe our methodology for reviewing commercial products and academic projects in the electronics toolkit space. Data and insights from this literature review informed the taxonomy presented in the next section.

3.1 What is an Electronics Prototyping Toolkit?

Various research communities have contributed electronics prototyping platforms and these contributions have spanned many different research areas including HCI, UbiComp, electronics engineering and mechatronics. A wealth of contributions have also come from industry, with examples of electronics toolkits from multinational corporations like LEGO [72] and smaller startups such as SAM Labs [109]. With so many toolkits from a variety of stakeholders, we were keen to objectively define which prototyping toolkits should be considered in-scope or out-of-scope for this paper.

On this basis, we define electronics prototyping toolkits as having one or more higher-level electronic modules, composed of discrete electronic components, that empower users to prototype a wide variety of functional artifacts with control over both hardware and software. This definition is similar to the more general definition of toolkits in HCI reported in [69] but we explicitly exclude the following types of projects and products:

(1) The large set of discrete electronic components (Type 1) as they are not part of a coherent toolkit.
(2) Projects that only present a novel engineering workflow without a new physical toolkit, such as the wide variety of workflows for connecting discrete components (Type 1) e.g. Instant Inkjet Circuits [62], iSkin [125], PaperPulse [103], The ToastBoard [40], Scanalog [120], VirtualComponent [33], Makers' Marks [111] and Heimdall [61].
(3) Purely mechanical toolkits for making physical artefacts without embedded electronic functionality, such as basic LEGO bricks, StrutModeling [71] and ProtoPiper [9].
(4) Purely software electronic prototyping tools such as Fritzing [48, 66], Autodesk EAGLE [41], ConductAR [89], IFTTT [85] and Gumstix Geppetto [49].
(5) "Assembly kits" according to Eisenberg et al.'s [42] "specificity classification". These are interactive products, such as Sifteo cubes [84], Sphero [118], PlayPiper [98], and Cubetto[35] that require assembly or programming but only allow for making a single or a handful of pre-determined artifacts. This naturally includes products primarily deployed as-is such as the Niko Home Control [93] and Loxone [75] home automation systems.

3.2 Corpus of Products and Publications

To create a representative corpus of products and research prototypes of electronics prototyping platforms, we conducted a systematic search on Google Search, the ACM Digital Library and IEEE Xplore similar to the methodology of Grosse-Puppendahl et al. [51]. Our search terms included all possible combinations of *hardware*, *prototyping*, *construction*, *physical*, *robotic* and *electronic* with each of *platform*, *toolkit*, *prototyping* and *kit*. We then examined the referenced literature of the articles in these search results. Additionally, we browsed through popular online magazines, including Make Magazine [78] and retailers and manufacturers of electronic platforms, such as Seeed Studio [114], Sparkfun [116] and Adafruit [1]. For every product and article we verified whether it was in- or out-of-scope using the exclusion criteria listed above. This resulted in a representative set of 56 unique electronics toolkits.

3.3 Characteristics

An initial set of labels was developed by one of the authors following an open-coding process [28, 69] and this was complemented by deductive codes from existing literature on toolkits [22]. We collectively reflected on the set of labels and formulated more precise definitions. Next, we grouped the refined labels into objective characteristics that aligned with the hardware aspects of electronics toolkits that form the focus of this paper. After reaching consensus on characteristics and the associated labels, we did another pass over the dataset to ensure the labels were correctly applied. This procedure resulted in 13 distinct characteristics.

As our taxonomy focuses on hardware related aspects of electronics toolkits, the final set of characteristics do not cover the programming and debugging aspects of toolkits in detail. These largely depend on the availability, compatibility, and characteristics of software libraries and are therefore out of scope for this work. Likewise, our taxonomy mainly focuses on the user experience offered by electronics toolkits and is therefore complementary to existing comparisons of technical specifications such as the online overview of development boards by Make Magazine [78].

3.4 Data Points and Clusters

While labeling electronics toolkits, we identified four clusters of toolkits with very similar characteristics: (1) *generic breakout boards*, such as breakout boards for the Bosch BNO055 IMU sensor [3] and the MPR121 capacitive touch sensor [7], (2) programmable *low-cost WiFi modules*, such as those based on the ESP32 [44] and the ESP8266 [45], (3) *silicon vendor development boards*, such as the TI Launchpad [122] and the nRF52840 DK [94], and (4) *FPGA development boards*, such as the Arduino MKR Vidor [12] and the Alchitry FPGA development boards [10]. Despite listing only two examples for each of these clusters, tens if not hundreds of similar tools exist. Although they have a wide range of technical specifications, we discovered a remarkable similarity in terms of the characteristics used for evaluation in this work. Therefore, instead of labeling each of these tools individually, we classified them into four clusters in our taxonomy. Indeed, the characteristics of each of these four clusters are very similar, and arguably they could be further collapsed, but we didn't want to over-condense the representation of so many different toolkits.

Similarly, some of the named toolkits have many variants; examples include a huge range of Arduino boards [11] and many variants of Raspberry Pi, including the recently launched Pi Pico [105]. Entering each of these as separate entries into our dataset was not feasible, so we have again consolidated them into a single entry.

After this clustering process and accounting for our exclusion criteria, our taxonomy includes 56 unique electronics toolkits as they exist at the time of writing. We believe that our taxonomy is more valuable than these individual data points as it presents a new perspective for discussing and comparing the current and future generations of electronics toolkits.

4 ELECTRONIC PROTOTYPING PLATFORM TAXONOMY

This section presents all 13 characteristics that encapsulate the nature of each platform, the target audience, how widely adopted they are, and how prototypes are assembled, deployed and used.

4.1 Nature and Application

We start by considering the **type of electronics** supported by each platform. As shown in Figure 1, 66% of platforms are Type 3, 8% are Type 2b and the rest are Type 2a (using the definitions from Section 2). As a reminder, the large number of Type 1 discrete electronic component solutions are are out of scope as per the exclusion criterion 1.

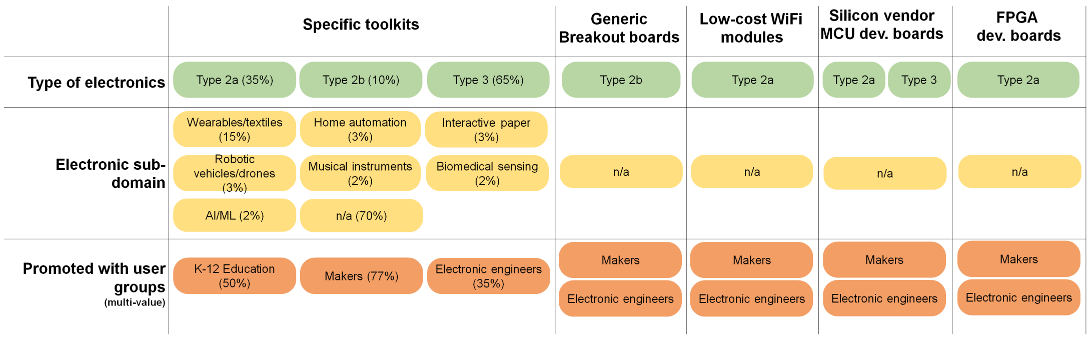

Fig. 1. Categories and labels concerning the nature and application of the platforms.

Similar to the "domain specificity" dimension of Eisenberg et al. [42], we examined each platform to determine if it was optimized for a particular **electronic sub-domain**. For example, mBot [80] and BITalino [19, 37] target *robotic vehicles* and *biomedical sensing* respectively. Platforms that have no specific focus, including Arduino, LEGO Mindstorms [72] and littleBits [17] are labelled *not applicable*. Resnick and Silverman [107] suggest that prototyping toolkits used in a K-12 education context should have "wide walls", meaning that there should be no particular focus on a sub-domain, because generic platforms allow children to expand their interests and passions. Outside of the classroom however, this does not necessarily hold true. In many cases—especially when considering makers—specialised toolkits are beneficial as they allow for faster prototyping in a particular sub-domain.

In addition to particular sub-domains, some electronics platforms are **promoted with user groups**. In particular, we saw toolkits that were advertised for *education*, *makers*, and *electronics engineers*, which we recorded accordingly. We noticed that this oftentimes is not the same group for whom the toolkit was originally designed. For example, the Arduino [11] originally targeted designers but has had a lot of impact in education. Similarly, the Raspberry Pi [105] was designed for education but also offers a compute module [104] very-much targeted at electronics engineers.

4.2 Assembly of Prototypes

The way(s) in which the modules in a prototyping toolkit are put together affects important qualities such as the speed of assembly and ease of modification of a prototype, its looks, and its durability. As shown in Figure 2 and described in this section, we identified three characteristics that consistently affect these 'assembly qualities'.

We identified four major **type of connections** used in prototyping platforms. 38% of toolkits require *individual conductors* to be manually connected, e.g. wires between an Arduino and a breadboard or WiFi module, copper tape for interconnecting Circuit Stickers [59], or conductive thread with the Lilypad [24]. Around a third of all toolkits use *multi-wire cables*, such as the ribbon cables used in Gadgeteer [57]. 36% of prototyping toolkits use a *direct module-to-module* approach that physically interconnects modules without wires, such as Arduino shields [13] that stack and littleBits [17] that clip together. Unfortunately this often results in prototypes that are increasingly tall or long. Finally, 2% of toolkits do not require physical connections as they communicate *wirelessly*, such as SAM Labs [109].

Multi-wire cables and *direct module-to-module* approaches help with interconnection as users can see and feel how modules can be combined without composing a detailed schematic diagram. Blikstein [22] refers to these affordances as "tangibility mappings". Tangibility mappings can "raise the ceiling" as they encourage exploring all possibilities. These approaches also encourage "a path of least resistance" [87] as errors are prevented. Some connectors are even entirely fool-proof by embedding a mechanical or magnetic poka-yoke constraint, such as littleBits [17].

The next characteristic further details the specifics of the **connection mechanism** which has implications for the durability of the assembly, the tools that are required during constructions, and the reusability of components. The majority of toolkits embed *friction fit* connectors, such as Gadgeteer's ribbon cables [57] and Arduino's headers [11]. 8% of toolkits support modules that interconnect mechanically using a *locking* mechanism, for example a locking JST connector [72, 80] or snaps [25]. 6% of toolkits use *crocodile clips*. To interconnect modules more easily, 10% of toolkits support *magnetic* connectors. Magnetic connectors simply snap together but can also accidentally disconnect when quite a light force is applied. A few toolkits offer modules with strong fixations, such as *screws* (2%) or stitches using *thread* (4%). While these connections require using external tools, they also allow for the construction of a greater ranges of pieces and materials [42]. Finally, in 2% of toolkits, namely Circuit Stickers [59], the modules interconnect using an adhesive. Similar to stitches, adhesives make it harder to re-use components.

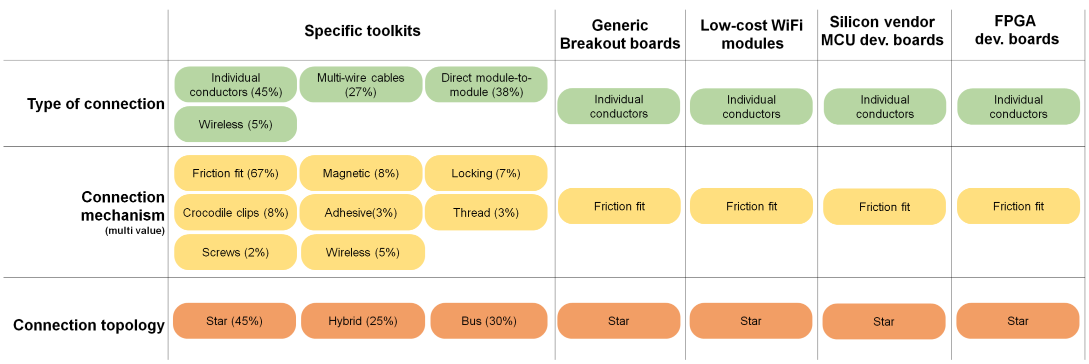

Fig. 2. Categories and labels concerning the assembly of individual elements into a working prototype.

The last characteristic related to the assembly of modules details the **connection topology**. The majority of toolkits (56%) require direct connections to the processing board in a *star* topology. In contrast 32% of toolkits employ a *bus* topology allowing modules to be daisy chained (e.g. Foxels [96] and Cubelets [34, 112]). Some toolkits (10%) also support a *hybrid* star/bus allowing daisy chaining for some modules while others need a direct connection to the processing board (e.g. Gadgeteer [57] and Pmod [99]). Some bus configurations support automatic detection of the topology of an assembly, a feature that further bridges the gap between computational and physical construction kits according to Eisenberg et al. [42].

4.3 Deploying and Configuring

Figure 3 shows the three categories that relate to deploying and configuring electronics prototyping toolkits. Typically Type 2b breakout boards do not require or support programming since they are operated through Type 2a components, so we added a *not applicable* label to all three categories.

The first characteristic **programming style** classifies all toolkits in two categories. One label groups all toolkits that are programmed using a *physical configuration* of modules in space (22%). Examples include MakerWear [64], ReWear [63], and littleBits [17] that allow specifying behavior by physically composing sensor and modifier modules. A second label groups all toolkits for which the behavior is specified in a *software configuration*

(74%). Examples include the micro:bit [86], which is programmed using visual building blocks (i.e. Microsoft MakeCode [39]) or Python.

	Specific toolkits			Generic Breakout boards	Low-cost WiFi modules	Silicon vendor MCU dev. boards	FPGA dev. boards
Programming style	Physical configuration (20%)	Software configuration (75%)	n/a (10%)	n/a	Software configuration	Software configuration	Software configuration
Dependencies for programming (multi-value)	Fully self-contained (23%) n/a (8%)	Connected wirelessly (12%)	Tethered to computer (73%)	n/a	Tethered to computer	Tethered to computer	Tethered to computer
Dependencies during deployment	Fully self-contained (80%) n/a (8%)	Connected wirelessly(3%)	Tethered to computer(8%)	n/a	Fully self-contained	Fully self-contained	Fully self-contained

Fig. 3. Categories and labels relating to deploying and configuring electronic prototypes built with the platforms in our survey. Note that the labels in two categories are not mutually exclusive.

The two last labeling categories document external computing **dependencies for programming** and **dependencies during deployment**. Eisenberg et al. [42] argue that communication between prototyping toolkits and desktop machines allows the toolkit to leverage the computational power of the desktop computer and the internet. While almost all modern electronics toolkits support communications with external computing devices, some different architectures are available. For example, both Cubelets [34, 112] and SAM Labs [109] are programmed wirelessly, but SAM Labs also requires a wireless connection to a desktop or tablet computer at all times to operate the final prototype.

4.4 Availability and Adoption

Figure 4 shows four characteristics of electronic prototyping platforms relating broadly to their availability and use. These are described in more detail in this section.

	Specific toolkits			Generic Breakout boards	Low-cost WiFi modules	Silicon vendor MCU dev. boards	FPGA dev. boards
Existing use	In commercial products (7%)	Multiple copies (7%)	Only used in one-offs (86%)	Multiple copies	In commercial products	Only used in one-offs	Only used in one-offs
Commercially available	Yes (67%)	No longer (2%)	Never (31%)	Yes	Yes	Yes	Yes
Third party use	Yes (86%)	No (17%)		Yes	Yes	Yes	Yes
Open source	Fully (45%)	Partial (42%)	Closed (13%)	Fully	Fully	Fully	Fully

Fig. 4. Categories and labels relating broadly to the availability and use of the electronic prototyping platforms in-scope for our survey.

The category **existing use** documents whether a platform has been used for *only one-off prototyping*, to make *multiple copies*, or if it has been embedded and shipped *in commercial products*. We define *multiple copies* as five or more exact copies built with the same prototyping platform. Multiple exact copies are often desired for long-term experiments with multiple setups [113]. Going further, sometimes *commercial products* ship using a prototyping platform, such as the Deltamaker 3D printer [38] that embeds a Raspberry Pi, and the TriggerTrap SLR Camera trigger device [60] that embeds an Arduino board. This suggests the platform is highly robust and competitively priced compared to a custom PCB. Interestingly, as far as we can tell, 84% of platforms (excluding the generic categories) have only been used to make one-offs.

The next category indicates if the platform was or still is **commercially available**. While many electronics toolkits are commercially available, several toolkits are *no longer* available (e.g. Gadgeteer [57]) or only available as research prototypes (e.g. Foxels [96] and ESLOV [43]).

The following category, **third party use**, records if the platform was used only by the team that created it (*first party*), or also by *third parties*. Naturally all toolkits that have been commercialised (60%) have also been used by many, but we were pleasantly surprised to find that an additional 19% of toolkits – about half of the platforms that have never been sold as products – have been used by third parties. Examples include the use d.tools [54] by third parties in workshops.

The final category in Figure 4 labels whether the engineering details of the platform are **open source**, either *fully* (36%), *partially* (48%, e.g. via well documented specifications or an open source schematic but closed PCB layout), or completely *closed* (16%, more common for commercial products).

5 ANALYZING THE CHARACTERISTICS

Figure 8 in the Appendix depicts most of the 13 characteristics we have captured across all 56 toolkits we reviewed. To make this information easier to access, our dataset and an interactive tool for exploring it are publicly available at http://etclassification.com. This allows researchers and practitioners to explore the dataset by sorting, filtering, and color-coding features, and to analyze our labels and characteristics in more detail. In light of the evolving nature of many electronics toolkits, we intentionally used the GitHub version control platform to host our dataset and interactive tool. Via GitHub 'pull requests', practitioners and researchers can update the dataset when existing toolkits evolve and new generations of prototyping toolkits become available.

Besides the objective characteristics covered in our taxonomy, there are important holistic attributes of electronic prototyping toolkits, such as ease of use and the level of expertise required. These attributes, which often map directly to user needs, are somewhat subjective which makes them harder to assess. We have developed an approach for comparing different toolkits by building on the objective characteristics of our taxonomy. In particular we have evaluated all 56 toolkits across four more holistic characteristics by assigning weights to the labels of our objective characteristics.

The first two of these more holistic characteristics we estimated are the level of electronics expertise required and the level of programming expertise required. These characteristics inevitably vary between toolkits, but the variation has not, to our knowledge, been quantified in the literature.

The third holistic characteristics evaluates the ease of construction of a prototype, independently of electronics and programming expertise. This relates to how fiddly and time-consuming the construction process is, and also encapsulates the time required to make multiple copies of the same prototype. Motivated by the work of Hodges et al. [55, 56, 65], the fourth and final more holistic characteristic that we evaluate is the ease of moving from a prototype to a product. This considers the complexity of the pathway from the prototype to a more integrated, robust, compact and cost-effective design that can be used for long term deployments, for more extensive evaluation, or even as a low-volume product.

Table 1 gives an overview of the objective characteristics in our taxonomy that contribute to each of the aforementioned holistic characteristics. For example, we postulate that it is more convenient to construct a prototype with a toolkit that (1) does not require connecting individual wires, (2) can be programmed by physically interconnecting blocks, (3) does not require additional tools for connecting modules, such as adhesives or stitches, and (4) can be connected in a bus topology. In our calculation, all relevant characteristics contribute equally and the weights are therefore distributed evenly across all labels within an objective characteristic. However, these weights can be adjusted according to personal preference or specific requirements, in order to identify the

Table 1. The four more holistic (and somewhat subjective) characteristics we evaluated (left) and the set of objective characteristics upon which they are based (right).

Expertise in electronics	Type of connection, promoted with user groups, connection topology.
Expertise in programming	Programming style, promoted with user groups.
Ease of construction	Type of connection, programming style, connection mechanism, connection topology.
Ease of moving from prototype to product	Programming style, existing use, dependency during deployment, open source.

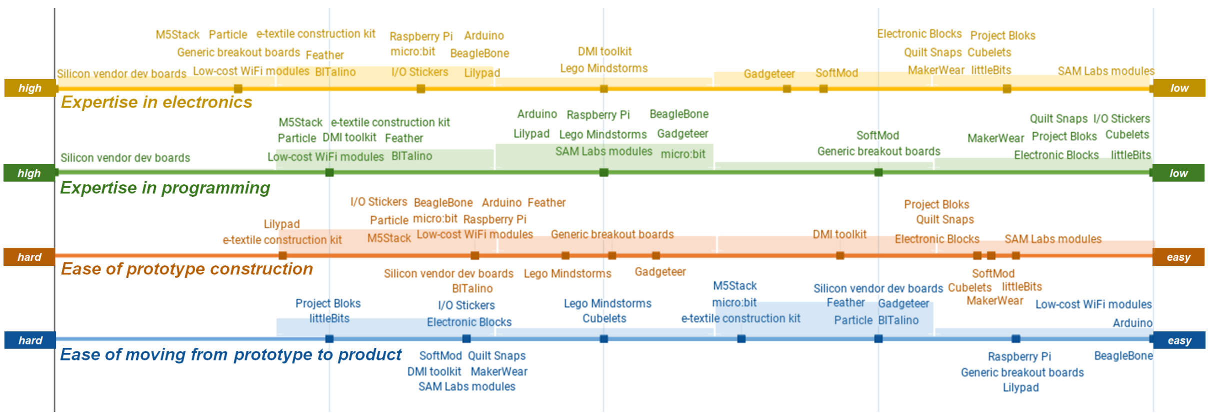

Fig. 5. Prototyping platforms ranked according to four holistic characteristics, with 'better' on the right. Not all platforms can be named due to space constraints, but the shading indicates the distribution of all 56 platforms from this study across each characteristic. Note that rankings are all relative to the dataset.

platforms that are most suitable for specific prototyping settings. To allow others to adjust the weights in this way, we have made the holistic characteristics configurable as part of our interactive dataset[5].

Figure 5 illustrates how the electronics toolkits compare based on our holistic characteristics, using our default weights. Although the ranking obviously changes if the weights change, we are not aware of a more objective metric for assessing and comparing holistic characteristics of toolkits. Due to space constraints, only the five highest and lowest ranking prototyping platforms for each of the four holistic characteristics are named in the figure. However, the histograms in the background of the figure visualize the distribution of all 56 electronics toolkits according to each of the more holistic characteristics.

An interesting observation from Figure 5 is that platforms which require expertise in electronics often also require expertise in programming; similarly platforms with low expertise requirements for electronics are often paired with low expertise in programming. There does not appear to be any obvious trade-off between these two holistic characteristics. The reverse is true of the other two holistic characteristics: very broadly we can see from Figure 5 that for many toolkits supporting easy prototype construction, it's hard to make a transition from prototype to product, and vice-versa.

6 UNDERSTANDING HOW ELECTRONICS TOOLKITS ARE USED

Having developed a taxonomy for electronics toolkits, we wanted to get a deeper understanding of how users value the different characteristics in practice, and uncover any characteristics we may have missed. During the course of our research we observed that there are hundreds of electronics toolkits targeted at professional electronics engineers and made available through numerous companies—but these are largely homogeneous, resulting in just a few categories of toolkit in our dataset. In contrast, a wide range of toolkits, many of which originate in academic research [24, 34, 57, 59, 64], are promoted for makers and for educators. Indeed, there seems to be a strong focus on the design and use of toolkits for education within the research community [20–22, 42, 107]. With such a strong focus on education, we were curious to evaluate if the needs of makers and electronics engineers are addressed by the current electronics toolkit offerings.

To this end, we conducted an online survey to get an understanding of preferences and experiences with prototyping electronics by non-professional (i.e. makers) and professional electronics engineers. The online survey was aimed at people who have built at least one electronic device prototype and consisted of both multiple-choice and free-form responses for a total of 75 questions. By formulating simple questions and grouping them in matrix Likert scales, we estimated filling in the survey would take 15 minutes on average. We employed a snowball sampling approach to reach as many electronics practitioners and experts as possible. We initially distributed the survey via email, social media and community-specific platforms. The study ran for one week in April 2020 and 122 people participated (21% response rate based on unique page views). The time to complete the survey was 5-57 minutes (median = 13.5 minutes).

6.1 The Demographics of our Respondents and Their Prototyping Experience

84% of our respondents identified as male, 13% as female, and 3% self described as 'other' or did not disclose gender. 16% of the participants were aged 18-24; 41% aged 25-34; 20% aged 35-44; 15% aged 45-54; 4% aged 55-64; and 2% aged 65-74. 2% of our respondents did not disclose their age. 46% of respondents were located in Europe, 32% in North America, and 20% in Asia, with 2% not disclosing a geographic region.

We also asked respondents about their backgrounds: 40% self-identified as an electrical or electronics engineer (hereafter abbreviated to electronics engineer), 18% as a mechanical or mechatronics engineer, 15% as an engineer with a different specialization, 17% as a product or industrial designer and 59% as a computer scientist or programmer. In addition, 69% self-identified as a researcher, 31% as a student (above K-12 education), 55% as a maker/DIY builder, 17% as a hobby programmer and 1% as retired. None of the preceding options were mutually exclusive—respondents were free to select all they felt applied. 3% of our respondents selected none.

We did an initial analysis of the survey data looking for differences based on the background of our 122 respondents. We found the most insightful way to pivot our dataset was based on whether respondents self-identified as having a background in electrical and electronics engineering or not. Put another way, the electrical and electronics engineers we surveyed reported different priorities and attitudes when it comes to electronics prototyping, compared to the respondents who don't have the same technical grounding in electronics. For this reason we split our respondents into two mutually exclusive groups of 49 electronics engineers vs. 73 respondents with backgrounds in other disciplines, and we have used that split for all the analysis presented hereafter.

Following the demographics questions, respondents were asked how frequently they build electronics prototypes in various device categories. They report having built electronic devices several times (meaning more than once or twice) for the following categories: learning and fun (70%), wearable electronics (45%), home automation systems (43%), robotic systems excluding wheeled robots (42%), games and toys (33%), office workplace devices (32%), wheeled robots (25%), interactive textiles (25%), biomedical sensing (24%), flying vehicles (12%), and in vehicle devices for cars (8%).

When splitting the data based on formal engineering expertise, we noticed electronics engineers build significantly more electronic devices across all categories compared to respondents from other disciplines (Mann-Whitney $U=200966.5$, $p<0.001$, $Z=-5.61$).

6.2 Use of Prototyping Platforms

We presented participants with a list of the commercially-available electronics toolkits covered in our detailed literature review, and asked them to specify in ranges how many they'd heard of, how many they had experimented with once or twice, and how many they used often. Participants were aware of the existence of many different platforms with 97% of respondents having heard of more than 4 and 75% of more than 7. In terms of the hands-on experience of our respondents, 33% had experimented with 7 or more platforms, 65% had experimented with 4 or more, and 98% had experimented with at least 2 platforms. This indicates that our snowball sampling approach had successfully solicited respondents with meaningful experience of prototyping toolkits.

Our data also shows that 8% of respondents use 2 or 3 platforms often, 28% use 4-6 platforms often, and 6% use 7 or more different platforms often. This is interesting because it shows that the respondents who are regularly building prototypes tend to leverage 4-6 different toolkits—they don't appear to get comfortable with just one or two tooklits and rely on only these. When comparing electronics engineers to all other respondents, we noticed that 42% often use more than 4 platforms compared to only 19% of the respondents from other disciplines (The Fisher's exact test: $N = 116$, $p < 0.05$, odds ratio = 3.0).

We were also curious why participants did not use a wider set of platforms, and we addressed this through a multiple choice (multiple answer) question. 70% of participants simply responded they were happy with the platforms they already use, with the majority of these (59%) indicating that it was not clear what benefits a new platform would bring.

In terms of the friction associated with adopting another platform, (30%) of our respondents indicated "I do not want to learn another platform", (27%) indicated "The platforms are not well established and might be deprecated in the future", (25%) checked "I do not want to pay for another platform ", (26%) indicated "The community support is not good enough", (24%) checked "Documentation or examples are too limited ", for (21%) "The platform seems too limited", (19%) indicated "The platform seems too complicated in use", and (15%) said that "Too little detailed information is available about the platform". The low response rate on these latter options suggests that despite the general trend of making electronics platforms simpler to learn and use, these factors are not necessarily sufficient for switching to another prototyping toolkit.

6.3 Important Characteristics of Prototyping Platforms

In a final series of questions on prototyping platforms, we asked participants to rate 26 characteristics based on how important they are when selecting a prototyping platform. These characteristics include the objective and more holistic characteristics listed earlier in this paper, reformulated to make them easier to understand where appropriate. We also added nine new characteristics in order to highlight potential gaps in our previous analysis. Figure 6 shows the importance of each characteristic for both electronics engineers and respondents from other disciplines. The Likert scale answers for all characteristics were: "always unimportant", "usually unimportant", "usually important" and "always important".

When analyzing the results, a Mann-Withney U test showed that the characteristic "easy to evolve to a custom PCB" is significantly more important to electronics engineers than other respondents ($U = 836.5$, $Z = -5.16$, $p < 0.001$, $r = 0.47$). A significant difference was also found for the characteristic "Quick to build more copies" (Mann-Withney $U = 1351$, $Z = -2.31$, $p < 0.05$, $r = 0.21$). These results suggest that electronics engineers find it more important to replicate prototypes, a process that often involves designing a custom PCB.

We also found a significant difference between electronics engineers and other disciplines for the characteristic "Use favorite programming language" (Mann-Withney $U = 1164.5$, $Z = -3.46$, $p < 0.001$, $r = 0.31$), with other disciplines being less concerned about the choice of programming language. Conversely, and quite understandably, respondents from other disciplines were keen to see "Little electronics expertise required" whereas the electronics engineers were almost ambivalent about this (Mann-Withney $U = 2393.5$, $Z = 3.79$, $p < 0.001$, $r = 0.35$).

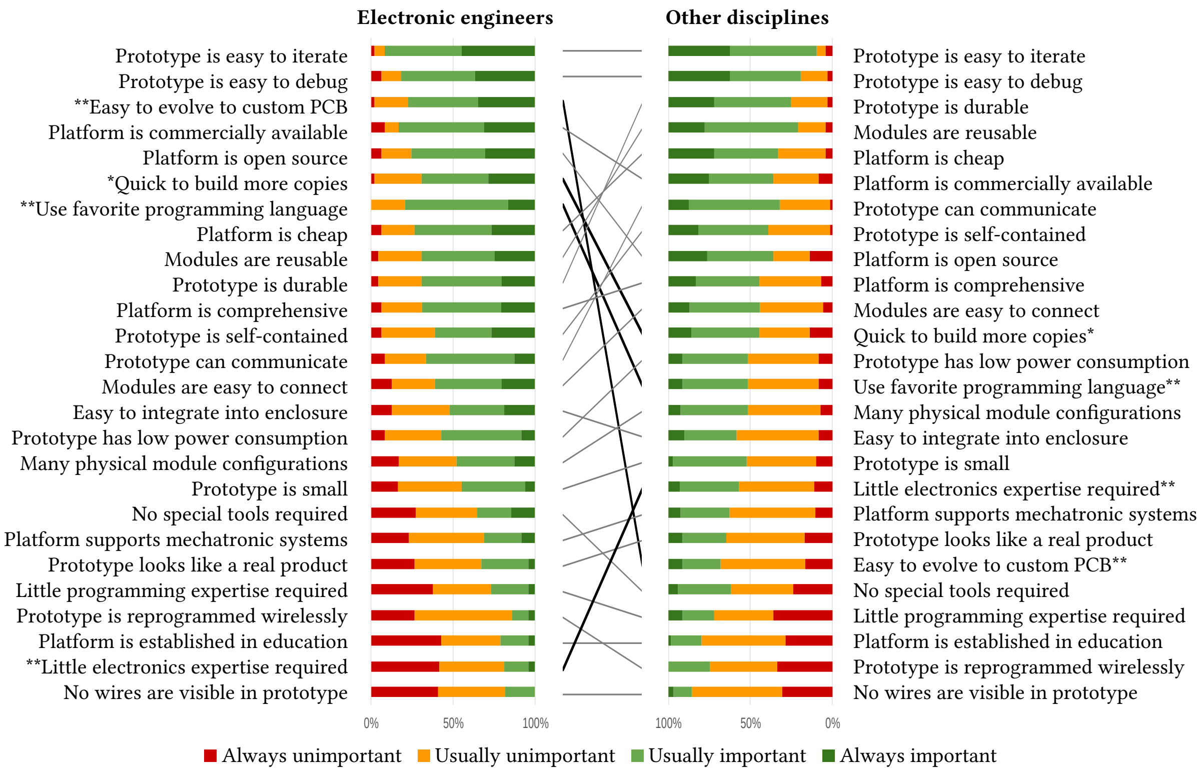

Fig. 6. Comparing the ranking of the importance of different characteristics between electric engineers and respondents with a different background. We determined significance using Mann Withney U tests (* $p < 0.05$, ** $p < 0.001$).

The characteristics in our survey included three technical characteristics not covered in our taxonomy: support for wireless communication, requirements for low power consumption, and the support offered for debugging. The first two of these, frequently reported in datasheets and online reviews [78], were not rated as important selection criteria by our participants. However, debugging is very important according to our respondents. Unfortunately, debugging is challenging to characterize and as explained in Section 4.3 is largely affected by software aspects of a toolkit. Evaluating this across different electronics toolkits remains an opportunity for future research (see Section 7). An open question allowed respondents to suggest additional important characteristics. Four valued compatibility with specific operating system(s) and two prioritized their current prototyping platforms because they were on-hand for immediate use.

6.4 Experiences of Type 1 Prototyping and Scaling Up to Multiple Copies

We also asked participants questions about their experience prototyping with Type 1 electronics, i.e. with discrete electronic components. The majority (64%) reported using solderless breadboards often, followed by soldering TH components (48%), soldering components to a custom PCB (43%), soldering SMD components (36%), using pre-built modules e.g. WiFi and battery charging modules (36%), soldering components to strip boards (37%), designing custom PCBs for SMD components (33%), using pre-built breakout boards for SMD components (22%), designing custom PCBs for TH components (21%). When using Fisher's exact test on the number of people who

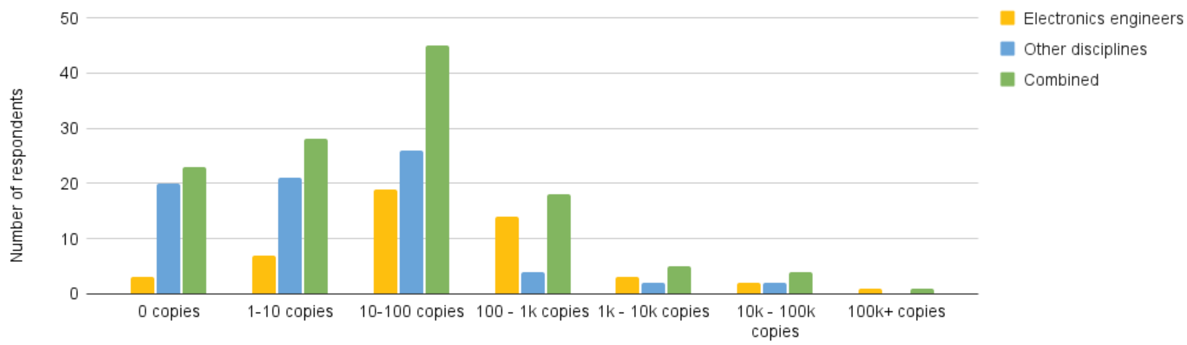

Fig. 7. The number of copies of electronic prototypes made by electronics engineers and respondents from other disciplines.

use these tools often, we detect significant differences between electronics engineers and other respondents regarding designing custom PCBs for SMD components (31 vs 8, N = 121, p < 0.001, odds ratio = 13.4), designing custom PCBs for TH components (18 vs 7, N = 120, p < 0.001, odds ratio = 5.5), soldering components to custom circuit boards (33 vs 18, N = 121, p < 0.001, odds ratio = 6.1), soldering components to strip boards (25 vs 19, N = 121, p < 0.01, odds ratio = 2.9), and soldering SMD components (33 vs 10, N = 121, p < 0.001, odds ratio = 12.4).

We also wanted to know to what extent those who prototype electronic devices start with one of the toolkits listed in this paper with a view to transitioning to a custom PCB later in the process. We learned that 53% of all respondents often start the process with Type 2a development boards, 44% often start with solderless breadboard, and 30% often start with Type 2b breakout boards. Of those using Type 2b toolkits, one half (15% of respondents) have used a system of modules such as Grove [52], Pmod [99], or Click [30]. Note that these responses are not mutually exclusive, which is consistent with our earlier observation that toolkits of Types 1 and 2 are often used in conjunction with each other.

Only 10% of our respondents often started the prototyping process with a Type 3 toolkit such as Gadgeteer or littleBits with the intent to transition to a custom PCB.

Next, we asked participants if they ever made multiple copies of a prototype and how many. We were surprised that 94% of electronics engineers and 73% of other disciplines reported making multiple copies; we thought it would be fewer. Our chi-squared test showed that this difference is significant (χ^2(6, N = 122) = 21.59, $p < 0.01$, ϕ=0.42). Figure 7 shows the distribution of the maximum number copies for both groups. While the majority of respondents who are not electronics engineers have made fewer than 10 copies, 48% of them have made 10 or more. Across all respondents 24% had made more than 100 copies, 9% more than 1000 and 5% more than 10,000. Again, these numbers were higher than we expected.

Finally, we asked participants how frequently they use different approaches to transition from a one-off prototype to multiple copies. 39% of respondents reported they often make copies using a custom PCB; there was a strong difference between electronics engineers and respondents in other disciplines (69% vs. 16%, the Fisher's exact test: N = 116 , p = < 0.001 , odds ratio = 11.2). In contrast only 12% of respondents often make copies using the same prototyping platform as used for the one-off prototype and only 4% switch electronics toolkits to facilitate multiple copies.

7 DISCUSSION

For the participants in our survey and the current generation of electronics toolkits they use, it appears that electronics and programming expertise requirements are not a significant barrier—either for those with an electronics engineering background or those from other disciplines. This contrasts with a theme we see across many toolkits, namely a drive to lower the programming and electronics skills required for device prototyping. This trend may be driven by the use of electronics toolkits by children for which researchers suggest to make the technology "as simple as possible – and maybe even simpler" [107]. Only a small portion of participants (19%) reported they would not use some of the toolkits in our taxonomy because they seem to complicated to use. This observation is also supported by the following free-form comment left by a participant at the end of our survey: *"Current modular kits make electronic design easier, but the fundamentals are missed out on, and users that use these platforms struggle [with] debugging (i.e. death by over abstraction)"*. Having said this, our snowball sampling approach may have solicited respondents with quite some experience in prototyping toolkit; as mentioned above 65% of them had experience with at least 4 different toolkits. Given this potential skew, we believe it's improtant to continue lowering the bar for electronics development to stimulate interest in STEM and to empower more individuals and communities to engage in electronics prototyping. However, for users familiar with prototyping toolkits, the level of expertise required in electronics or programming appears to be sufficiently low already, and this group of users might value other features more.

Our respondents did reveal several qualities of prototyping platforms that both electronics engineers and users from other disciplines do find important. The most highly rated characteristics across all respondents were ease of iteration and ease of debugging, attributes that naturally relate to our holistic characteristic of "speed of construction". It seems that a future toolkit that speeds up the prototyping process without constraining the artifacts that can be built (as many of the commercially available Type 3 toolkits currently do), would appeal broadly and could add much value. We would certainly ask participants more about this in a future survey.

Another characteristic that was ranked relatively high by all our respondents was low cost. This is corroborated by the free-form comment from one of our participants: *"They [the toolkits] are often ludicrously expensive to accommodate unneeded features, or feel too much like an end product in quality."* As researchers, when we develop new toolkits we tend to focus on adding features in service of adding value, and product designers may be tempted to improve the form, fit and finish of toolkits; but this user feedback is a timely reminder that constraining cost can be as valuable as adding functionality.

We were somewhat surprised that the vast majority of our respondents (81%), independent of their background, have engaged in making multiple copies of prototypes. Over 50% have made over ten copies, and over 20% have made over 100 copies of a prototype, again indicating a bias in our snowball sampling towards more experienced makers. This appears to reflect a significant group of people who are moving beyond demonstrating the feasibility of their ideas via a lab-bound prototype, to a deployment stage frequently involving tens or hundreds of devices, if not more. This transition is typically supported by moving from a toolkit-based prototype to custom-designed PCBs, a process that in turn benefits from the availability of open source design information. However, 92% of our non-electronics engineering respondents do not often design custom PCBs. This observation is also supported by comments of respondents, such as: *"I like to get into PCB design – but often find it daunting"* and *"When designing PCBs, I found it hard to understand differences and packaging sizes of components and its implications on my design"*.

The holistic characteristic of "Ease of moving from prototype to product" that we introduced in Section 5 should be a useful indicator of the ease of moving to a custom PCB, but we would like to evaluate this more rigorously and formalise the selection of weights assigned to the objective characteristics that underpin the calculation. It may be possible to personalize recommendations for prototyping platforms based on the preferences, expertise, and needs of individual users.

Our previous analysis also indicated that toolkits which more readily support a transition from prototype to product may be liable to complicate the prototyping process, something that warrants careful consideration. And finally, if more objective characteristics are made available in future versions of the taxonomy, additional holistic characteristics such as ease of debugging could be included.

One respondent highlighted a potentially fruitful research direction: *"A prototyping toolkit that makes this process [transitioning from prototypes to PCB] easier may make a huge difference."* Although systems like Scanalog [120] help with component selection and circuit design, to our knowledge no existing electronics toolkits have been designed with the transition from a prototype to a more integrated PCB solution in mind. However, the value of doing so has recently been highlighted in the literature [55, 56]. In addition to the PCB design process, other barriers to making multiple copies have been reported in the literature [65], and one of these was also raised by our respondents: *"it [prototyping] has gotten so much easier over the years... part identification/procurement for small volume runs that is affordable is the biggest hassle"*, pointing to another area of future research.

Our study results show that respondents use electronics toolkits for prototyping a variety of interactive and ubiquitous computing devices, such as systems for experimentation and fun, wearable electronics and home automation solutions. This is consistent with some of the specific domains targeted by electronics toolkits and reported in our literature survey (see Figure 8). In general, we believe there is a symbiotic relationship between electronics toolkit availability and adoption in that when more tools are available, more ideas are explored and novel potential is revealed, driving further demand. In the coming years we imagine there will be a growing set of electronics toolkits to support AI and machine learning applications, complementing Coral [32]. Having said this, our respondents did not report much interest in textile interfaces or biomedical sensing.

The current version of our taxonomy, consisting of 13 objective characteristics and their corresponding labels, mainly focuses on hardware aspects of toolkits. During the course of this work it has become clear that it could be extended to include aspects of programming such as development platform compatibility, the availability of software libraries, and support for debugging. In terms of the latter, we are aware of several novel debugging and inspection techniques and tools for electronic prototyping, such as BiFrost [81], WiFrost [82] and Scanalog [120], but more research is needed to characterize what makes platforms easy to debug.

On reflection, it could also be useful to include the more standard technical specifications that have been reported elsewhere, such as operating voltage requirements, processor speed and memory, so that all pertinent information is available in a consolidated form. This would likely involve breaking out our four 'generic' categories (e.g. 'Low-cost WiFi modules) and several other aggregated categories (e.g. 'Arduino') into specific products. 59% of our survey participants reported that they were not clear of the benefits offered by prototyping platforms other than the ones they already use, but such a central repository of characteristics could help with this.

Finally, we must acknowledge that while our survey involved a good number of respondents, our snowball sampling approach may have skewed towards more experienced users of prototyping toolkits that are mostly male (84%) and located in the Western world (78%). This might have led to biases in our data, that we are keen to address in future surveys. Indeed, with electronics prototyping becoming increasingly accessible to a global population with diverse backgrounds, experiences, and needs, more studies are needed to address the needs of specific user groups. We therefore see our work as a starting point that we hope other researchers can build upon.

8 CONCLUSION

In concluding, we hope that other researchers and practitioners from the research community find value in the analysis of 56 electronics prototyping toolkits presented in this paper. We have complemented existing surveys and summaries by developing and presenting 13 objective characteristics that differ from the technical specifications more commonly reported. We have also developed the concept of holistic characteristics, which more naturally represent typical user needs. These are more subjective but none-the-less can be encoded to facilitate direct comparisons between toolkits. We encourage readers to explore our dataset for themselves via http://etclassification.com and GitHub. We described the results of a survey of 122 electronics toolkit users that both corroborate and complement our first-hand analysis.

Having highlighted the strengths and weaknesses of existing toolkits, we presented future directions for electronics toolkit research that other researchers may be inspired to explore. We also hope our work will be valuable for those in the community who need to build and potentially scale out prototypes as part of their research. We envision a future where prototyping toolkits support the needs of novices and experienced users alike, in their pursuit of new interactive and ubiquitous computing solutions.

ACKNOWLEDGMENTS

This research was partially supported by the Special Research Fund (BOF) of Hasselt University, project BOF19KP04. We thank Gabriele D'Amone, Nicholas Chen, Teddy Seyed, Kris Luyten, and Florian Heller for the valuable input during the course of the research.

REFERENCES

[1] Adafruit 2021. Adafruit Industries, Unique & fun DIY electronics and kits. https://www.adafruit.com/ Accessed: 2020-09-17.

[2] Adafruit 2021. Breakout Boards. https://www.adafruit.com/category/42 Accessed: 2021-04-30.

[3] Adafruit BNO055 2021. Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055. https://www.adafruit.com/product/2472 Accessed: 2021-04-30.

[4] Adafruit Feather 2021. Feather - A complete line of development boards from Adafruit that are both standalone and stackable. https://www.adafruit.com/feather Accessed: 2021-04-30.

[5] Adafruit Flora 2021. FLORA - Wearable electronic platform. https://learn.adafruit.com/getting-started-with-flora Accessed: 2021-04-30.

[6] Adafruit Gemma 2021. Adafruit GEMMA v2 - Miniature wearable electronic platform. https://learn.adafruit.com/introducing-gemma Accessed: 2021-04-30.

[7] Adafruit MPR121 2021. Adafruit 12-Key Capacitive Touch Sensor Breakout - MPR121. https://www.adafruit.com/product/1982 Accessed: 2021-04-30.

[8] Adafruit Stemma 2021. What is STEMMA? https://learn.adafruit.com/introducing-adafruit-stemma-qt Accessed: 2021-04-30.

[9] Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. 2015. Protopiper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA, 2015-11-05) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 427–436. https://doi.org/10.1145/2807442.2807505

[10] Alchitry 2021. Alchitry homepage. https://alchitry.com/ Accessed: 2021-04-30.

[11] Arduino 2021. Arduino homepage. https://www.arduino.cc/ Accessed: 2021-04-30.

[12] Arduino MKR Vidor 2021. Arduino MKR Vidor 4000 product page. https://store.arduino.cc/arduino-mkr-vidor-4000 Accessed: 2021-04-30.

[13] Arduino Shields 2021. Arduino Playground: Arduino-Compatible Hardware. https://playground.arduino.cc/Main/SimilarBoards Accessed: 2021-04-30.

[14] Jonny Austin, Howard Baker, Thomas Ball, James Devine, Joe Finney, Peli De Halleux, Steve Hodges, Michał Moskal, and Gareth Stockdale. 2020. The BBC micro:bit: from the U.K. to the world. *Commun. ACM* 63, 3 (2020), 62–69. https://doi.org/10.1145/3368856

[15] Massimo Banzi. 2008. *Getting Started with Arduino* (ill ed.). Make Books - Imprint of: O'Reilly Media, California, CA, USA.

[16] BASIC Stamp 2021. Parallax Inc BASIC Stamp products. https://www.parallax.com/product-category/basic-stamp/ Accessed: 2021-04-30.

[17] Ayah Bdeir. 2009. Electronics as material: littleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (Cambridge, United Kingdom, 2009-02-16) *(TEI '09)*. Association for Computing Machinery, New York, NY, USA, 397–400. https://doi.org/10.1145/1517664.1517743

[18] BeagleBone 2021. BeagleBoard - community supported open hardware computers for making. https://beagleboard.org/ Accessed: 2021-04-30.

[19] Bitalino 2021. BITalino - Biomedical Equipment | Low-Cost Toolkit. https://bitalino.com/ Accessed: 2021-04-30.

[20] Paulo Blikstein. 2013. Digital fabrication and 'making' in education: The democratization of invention. *FabLabs: Of machines, makers and inventors* 4, 1 (2013), 1–21.

[21] Paulo Blikstein. 2013. Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. In *Proceedings of the 12th international conference on interaction design and children.* 173–182.

[22] Paulo Blikstein et al. 2015. Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Found. Trends Hum. Comput. Interact.* 9, 1 (2015), 1–68.

[23] Leah Buechley. 2006. A Construction Kit for Electronic Textiles. In *2006 10th IEEE International Symposium on Wearable Computers* (2006-10). IEEE Institute of Electrical and Electronics Engineers, New York, NY, USA, 83–90. https://doi.org/10.1109/ISWC.2006.286348 ISSN: 2376-8541.

[24] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, 2008-04-06) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 423–432. https://doi.org/10.1145/1357054.1357123

[25] 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)* (2005-12-28). IEEE Institute of Electrical and Electronics Engineers, New York, NY, USA, 3 pp. https://doi.org/10.1109/WMTE.2005.55

[26] Filipe Calegario, Marcelo Wanderley, Stéphane Huot, Giordano Cabral, and Geber Ramalho. 2017. A Method and Toolkit for Digital Musical Instruments: Generating Ideas and Prototypes. *IEEE MultiMedia* 24 (2017), 63–71. https://doi.org/10.1109/MMUL.2017.18

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

[28] Kathy Charmaz. 2014. *Constructing grounded theory*. sage.

[29] Circuit Playground 2021. Introducing Circuit Playground – Adafruit Learning. https://learn.adafruit.com/introducing-circuit-playground/overview Accessed: 2020-09-17.

[30] Click 2021. Mikroelektronika Click Boards. http://www.mikroe.com/click Accessed: 2021-04-30.

[31] Beginner's Mind Collective and David Shaw. 2012. Makey Makey: improvising tangible and nature-based user interfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12* (Kingston, Ontario, Canada). ACM Press, New York, NY, USA, 367. https://doi.org/10.1145/2148131.2148219

[32] Coral 2021. Coral homepage. https://coral.ai/ Accessed: 2021-04-30.

[33] Angelo Corsaro, Douglas C Schmidt, Raymond Klefstad, and Carlos O'Ryan. 2002. Virtual component: a design pattern for memory-constrained embedded applications. In *Proceedings of the 9th Annual Conference on the Pattern Languages of Programs*. Citeseer, 1–13.

[34] Cubelets 2021. Modular Robotics Cubelets robot blocks. https://www.modrobotics.com/ Accessed: 2021-04-30.

[35] Cubetto 2021. Meet Cubetto - Primo Toys Cubetto: A toy robot teaching kids code & computer programming. https://www.primotoys.com/ Accessed: 2021-04-30.

[36] Cubit 2021. Cubit Steam Education 2020 homepage. https://cubit.cc/ Accessed: 2021-04-30.

[37] Hugo Placido da Silva, Ana Fred, and Raul Martins. 2014. Biosignals for Everyone. *IEEE Pervasive Computing* 13, 4 (2014), 64–71. https://doi.org/10.1109/MPRV.2014.61

[38] DeltaMaker 2021. DeltaMaker: An Elegant 3D Printer. https://www.deltamaker.com/ Accessed: 2021-04-30.

[39] James Devine, Joe Finney, Peli de Halleux, Michał Moskal, Thomas Ball, and Steve Hodges. 2018. MakeCode and CODAL: intuitive and efficient embedded systems programming for education. *ACM SIGPLAN Notices* 53, 6 (2018), 19–30.

[40] Daniel Drew, Julie L. Newcomb, William McGrath, Filip Maksimovic, David Mellis, and Björn Hartmann. 2016. The Toastboard: Ubiquitous Instrumentation and Automated Checking of Breadboarded Circuits. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo Japan, 2016-10-16). ACM, New York, NY, USA, 677–686. https://doi.org/10.1145/2984511.2984566

[41] Eagle Autodesk 2021. https://www.autodesk.com/products/eagle Accessed: 2021-04-30.

[42] Michael Eisenberg, Ann Eisenberg, Mark Gross, Khomkrit Kaowthumrong, Nathaniel Lee, and Will Lovett. 2002. Computationally-enhanced construction kits for children: Prototype and principles. In *Proceedings of the Fifth International Conference of the Learning Sciences*. 23–26.

[43] ESLOV 2021. ESLOV IoT Invention Kit (Canceled). https://www.kickstarter.com/projects/iot-invention-kit/eslov-iot-invention-kit Accessed: 2021-04-30.

[44] ESP32 2021. A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications. https://www.espressif.com/en/products/socs/esp32 Accessed: 2021-04-30.

[45] ESP8266 2021. A cost-effective and highly integrated Wi-Fi MCU for IoT applications. https://www.espressif.com/en/products/socs/esp8266 Accessed: 2021-04-30.

[46] Espressif Modules 2021. Espressif offers a wide range of fully-certified Wi-Fi & Bluetooth modules powered by our own advanced SoCs. https://www.espressif.com/en/products/modules Accessed: 2021-04-30.

[47] 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* (Ann Arbor, Michigan, 2011-06-20) *(IDC '11)*. Association for Computing Machinery, New York, NY, USA, 238–241. https://doi.org/10.1145/1999030.1999071

[48] Fritzing 2021. Fritzing. http://fritzing.org/ Accessed: 2021-04-30.

[49] Geppetto Gumstix 2021. Welcome to Geppetto. https://geppetto.gumstix.com/ Accessed: 2021-04-30.

[50] 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* (Orlando, Florida, 2001-11-11) *(UIST '01)*. Association for Computing Machinery, New York, NY, USA, 209–218. https://doi.org/10.1145/502348.502388

[51] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S Reynolds, and Joshua R Smith. 2017. Finding common ground: A survey of capacitive sensing in human-computer interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3293–3315.

[52] Grove 2021. Grove Beginner Kit for Arduino(EOL) - Seeed Wiki. https://wiki.seeedstudio.com/Grove_Beginner_Kit_for_Arduino/ Accessed: 2021-04-30.

[53] 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* (Montreal QC, Canada, 2018-04-19) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173656

[54] 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* (Montreux, Switzerland). ACM Press, New York, NY, USA, 299. https://doi.org/10.1145/1166253.1166300

[55] Steve Hodges. 2020. *Democratizing the Production of Interactive Hardware*. Association for Computing Machinery, New York, NY, USA, 5–6. https://doi.org/10.1145/3379337.3422877

[56] Steve Hodges and Nicholas Chen. 2020. Long Tail Hardware: Turning Device Concepts Into Viable Low Volume Products. *IEEE Pervasive Computing* 18 (2020), 51–59. https://doi.org/10.1109/MPRV.2019.2947966

[57] 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* (Denver, Colorado, USA). ACM Press, New York, NY, USA, 391. https://doi.org/10.1145/2445196.2445315

[58] Steve Hodges, Sue Sentance, Joe Finney, and Thomas Ball. 2020. Physical computing: A key element of modern computer science education. *Computer* 53, 4 (2020), 20–30.

[59] 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* (Toronto, Ontario, Canada, 2014-04-26) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1743–1746. https://doi.org/10.1145/2556288.2557150

[60] H. J. Kamps. 2021. Hardware is Hard: Getting a Kickstarter project out the door. https://medium.com/triggertrap-playbook/hardware-is-hard-getting-a-kickstarter-project-shipped-59c9596bdd7f Accessed: 2021-04-30.

[61] Mitchell Karchemsky, J.D. Zamfirescu-Pereira, Kuan-Ju Wu, François Guimbretière, and Bjoern Hartmann. 2019. Heimdall: A Remotely Controlled Inspection Workbench For Debugging Microcontroller Projects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19* (Glasgow, Scotland Uk). ACM Press, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300728

[62] Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (Zurich, Switzerland, 2013-09-08) *(UbiComp '13)*. Association for Computing Machinery, New York, NY, USA, 363–372. https://doi.org/10.1145/2493432.2493486

[63] Majeed Kazemitabaar, Liang He, Katie Wang, Chloe Aloimonos, Tony Cheng, and Jon E. Froehlich. 2016. ReWear: Early Explorations of a Modular Wearable Construction Kit for Young Children. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA, 2016-05-07) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 2072–2080. https://doi.org/10.1145/2851581.2892525

[64] 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* (Denver, Colorado, USA, 2017-05-02) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 133–145. https://doi.org/10.1145/3025453.3025887

[65] Rushil Khurana and Steve Hodges. 2020. Beyond the Prototype: Understanding the Challenge of Scaling Hardware Device Production. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA, 2020-04-21) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376761

[66] André Knörig, Reto Wettach, and Jonathan Cohen. 2009. Fritzing: a tool for advancing electronic prototyping for designers. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction - TEI '09* (Cambridge, United Kingdom). ACM Press, New York, NY, USA, 351. https://doi.org/10.1145/1517664.1517735

[67] Kristof Van Laerhoven, Albrecht Schmidt, and Hans-Werner Gellersen. 2002. Pin&Play: Networking Objects through Pins. In *Proceedings of the 4th international conference on Ubiquitous Computing* (Göteborg, Sweden, 2002-09-29) *(UbiComp '02)*. Springer-Verlag, New York, NY, USA, 219–228.

[68] Mannu Lambrichts, Jose Maria Tijerina, and Raf Ramakers. 2020. SoftMod: A Soft Modular Plug-and-Play Kit for Prototyping Electronic Systems. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Sydney NSW, Australia, 2020-02-09) *(TEI '20)*. Association for Computing Machinery, New York, NY, USA, 287–298. https://doi.org/10.1145/3374920.3374950

[69] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–17.

[70] Johnny C. Lee, Daniel Avrahami, Scott E. Hudson, Jodi Forlizzi, Paul H. Dietz, and Darren Leigh. 2004. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques* (Cambridge, MA, USA, 2004-08-01) *(DIS '04)*. Association for Computing Machinery, New York, NY, USA, 167–175. https://doi.org/10.1145/1013115.1013139

[71] 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* (Québec City, QC, Canada, 2017-10-20) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 471–479. https://doi.org/10.1145/3126594.3126643

[72] Lego Mindstorms 2021. Lego Mindstoms EV3. https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313 Accessed: 2021-04-30.

[73] David I. Lehn, Craig W. Neely, Kevin Schoonover, Thomas L. Martin, and Mark T. Jones. 2004. e-TAGs: e-Textile Attached Gadgets. (2004), 12. https://vtechworks.lib.vt.edu/handle/10919/80538

[74] Joshua Lifton, Deva Seetharam, Michael Broxton, and Joseph Paradiso. 2002. Pushpin Computing System Overview: A Platform for Distributed, Embedded, Ubiquitous Sensor Networks. In *Pervasive Comp Lect Notes Comput Sci (LNCS)* (2002-08-21), Vol. 2414. 605–614. https://doi.org/10.1007/3-540-45866-2_12

[75] Loxone Smart Home 2021. Loxone Smart Home & Commercial Projects Create Automation. https://www.loxone.com/enus/ Accessed: 2021-04-30.

[76] m5stack 2021. M5STACK - Modular Rapid ESP32 IoT Development Board. https://m5stack.com/ Accessed: 2020-09-17.

[77] Make Magazine 2021. Make: celebrates your right to tweak, hack, and bend any technology to your will. https://makezine.com Accessed: 2020-09-17.

[78] Make Magazine 2021. Makers' Guide to Boards. https://makezine.com/comparison/boards/ Accessed: 2021-04-30.

[79] Mbed 2021. Free open source IoT OS and development tools from Arm Mbed. https://os.mbed.com/ Accessed: 2021-04-30.

[80] mBot 2021. Makeblock mBot Entry-level Educational Robot Kit. https://www.makeblock.com/mbot Accessed: 2021-04-30.

[81] Will McGrath, Daniel Drew, Jeremy Warner, Majeed Kazemitabaar, Mitchell Karchemsky, David Mellis, and Björn Hartmann. 2017. Bifröst: Visualizing and Checking Behavior of Embedded Systems across Hardware and Software. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City QC Canada, 2017-10-20). ACM, New York, NY, USA, 299–310. https://doi.org/10.1145/3126594.3126658

[82] William McGrath, Jeremy Warner, Mitchell Karchemsky, Andrew Head, Daniel Drew, and Bjoern Hartmann. 2018. WiFröst: Bridging the Information Gap for Debugging of Networked Embedded Systems. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18* (Berlin, Germany). ACM Press, New York, NY, USA, 447–455. https://doi.org/10.1145/3242587.3242668

[83] 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*. 83–90.

[84] David Merrill, Emily Sun, and Jeevan Kalanithi. 2012. Sifteo cubes. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (Austin, Texas, USA, 2012-05-05) *(CHI EA '12)*. Association for Computing Machinery, New York, NY, USA, 1015–1018. https://doi.org/10.1145/2212776.2212374

[85] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An empirical characterization of IFTTT: ecosystem, usage, and performance. In *Proceedings of the 2017 Internet Measurement Conference* (London United Kingdom, 2017-11). ACM, New York, NY, USA, 398–404. https://doi.org/10.1145/3131365.3131369

[86] micro:bit 2021. Micro:bit Educational Foundation. https://microbit.org/ Accessed: 2021-04-30.

[87] Brad Myers, Scott E Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 1 (2000), 3–28.

[88] 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* (Montreal QC, Canada, 2018-04-19) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173762

[89] Koya Narumi, Steve Hodges, and Yoshihiro Kawahara. 2015. ConductAR: an augmented reality based tool for iterative design of conductive ink circuits. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka,

Japan, 2015-09-07) *(UbiComp '15)*. Association for Computing Machinery, New York, NY, USA, 791–800. https://doi.org/10.1145/2750858.2804267
[90] Netduino 2021. So long, Netduino; may you long live in our hearts. https://www.wildernesslabs.co/netduino Accessed: 2021-04-30.
[91] Neuron 2021. Makeblock Neuron Makeblock - Global STEAM Education Solution Provider. https://www.makeblock.com/steam-kits/neuron Accessed: 2021-04-30.
[92] Grace Ngai, Stephen C.F. Chan, Hong Va Leong, and Vincent T.Y. Ng. 2013. Designing i*CATch: A multipurpose, education-friendly construction kit for physical and wearable computing. *ACM Transactions on Computing Education* 13, 2 (2013), 7:1–7:30. https://doi.org/10.1145/2483710.2483712
[93] Niko Home Control 2021. Niko Home Control. https://www.niko.eu/en/products/niko-home-control Accessed: 2021-04-30.
[94] NRF52840DK 2021. Bluetooth Low Energy, Bluetooth mesh, NFC, Thread and Zigbee development kit for the nRF52840 SoC. https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK Accessed: 2021-04-30.
[95] Particle 2021. Particle Company News and Updates. https://www.particle.io/ Accessed: 2021-04-30.
[96] Florian Perteneder, Kathrin Probst, Joanne Leong, Sebastian Gassler, Christian Rendl, Patrick Parzer, Katharina Fluch, Sophie Gahleitner, Sean Follmer, Hideki Koike, and Michael Haller. 2020. Foxels: Build Your Own Smart Furniture. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Sydney NSW, Australia, 2020-02-09) *(TEI '20)*. Association for Computing Machinery, New York, NY, USA, 111–122. https://doi.org/10.1145/3374920.3374935
[97] Pine 2021. A Small Form-Factor Single Board Computer Capable of 4K Playback. https://www.pine64.org/devices/single-board-computers/rock64/ Accessed: 2021-04-30.
[98] Piper 2021. Piper Computer Kit. https://www.playpiper.com/ Accessed: 2021-04-30.
[99] Pmod 2021. Digilent Pmod Modules and Connectors - Interface with Development Boards. https://store.digilentinc.com/pmod-modules-connectors/ Accessed: 2021-04-30.
[100] Printoo 2021. Printoo: Paper-Thin, Flexible Arduino-Compatible modules! https://www.kickstarter.com/projects/1030661323/printoo-paper-thin-flexible-arduinotm-compatible-m Accessed: 2021-04-30.
[101] Project Bloks 2021. Project Bloks - Project Bloks research into tangible programming is complete. https://projectbloks.withgoogle.com/ Accessed: 2021-04-30.
[102] Propeller 2021. Propeller Parallax Inc. https://www.parallax.com/catalog/microcontrollers/propeller Accessed: 2021-04-30.
[103] 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* (Seoul, Republic of Korea, 2015-04-18) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2457–2466. https://doi.org/10.1145/2702123.2702487
[104] Raspberry Pi 2021. Compute Module 3+. https://www.raspberrypi.org/products/compute-module-3-plus/ Accessed: 2021-04-30.
[105] Raspberry Pi 2021. Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. https://www.raspberrypi.org/ Accessed: 2021-04-30.
[106] Raspberry Pi Pico 2021. The new flexible $4 microcontroller board from Raspberry Pi. https://www.raspberrypi.org/products/raspberry-pi-pico/ Accessed: 2021-04-30.
[107] Mitchel Resnick and Brian Silverman. 2005. Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children*. 117–122.
[108] Joel Sadler, Kevin Durfee, Lauren Shluzas, and Paulo Blikstein. 2015. Bloctopus: A Novice Modular Sensor System for Playful Prototyping. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (Stanford, California, USA, 2015-01-15) *(TEI '15)*. Association for Computing Machinery, New York, NY, USA, 347–354. https://doi.org/10.1145/2677199.2680581
[109] SAM Labs modules 2021. SAM Labs homepage. https://samlabs.com/us/ Accessed: 2021-04-30.
[110] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (Austin, Texas, USA). ACM Press, New York, NY, USA, 483. https://doi.org/10.1145/2207676.2207743
[111] 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* (Charlotte, NC, USA, 2015-11-05) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 103–108. https://doi.org/10.1145/2807442.2807508
[112] 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* (Banff, Alberta, Canada, 2006-11-02) *(ICMI '06)*. Association for Computing Machinery, New York, NY, USA, 72–75. https://doi.org/10.1145/1180995.1181010
[113] James Scott, A.J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. 2011. PreHeat: controlling home heating using occupancy prediction. In *Proceedings of the 13th international conference on Ubiquitous computing* (Beijing, China, 2011-09-17) *(UbiComp '11)*. Association for Computing Machinery, New York, NY, USA, 281–290. https://doi.org/10.1145/2030112.2030151
[114] Seeed Studio 2021. Seeed Studio Bazaar, The IoT Hardware enabler. https://www.seeedstudio.com/ Accessed: 2020-09-17.
[115] Snap Circuits 2021. Educational STEM Toys: Snap Circuits. https://www.elenco.com/ Accessed: 2021-04-30.
[116] SparkFun 2021. SparkFun Electronics homepage. https://www.sparkfun.com/ Accessed: 2020-09-17.

[117] SparkFun Qwiic 2021. Prototyping with I2C has never been easier. https://www.sparkfun.com/qwiic Accessed: 2021-04-30.
[118] Sphero 2021. STEM Kits & Robotics for Kids Inspire STEM Education with Sphero. https://sphero.com/ Accessed: 2021-04-30.
[119] STM32 Discovery Kits 2021. STMicroelectronics STM32 Discovery Kits. https://www.st.com/en/evaluation-tools/stm32-discovery-kits.html Accessed: 2021-04-30.
[120] Evan Strasnick, Maneesh Agrawala, and Sean Follmer. 2017. Scanalog: Interactive Design and Debugging of Analog Circuits with Programmable Hardware. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada, 2017-10-20) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 321–330. https://doi.org/10.1145/3126594.3126618
[121] Teensy 2021. Teensy USB Development Board. https://www.pjrc.com/teensy/ Accessed: 2021-04-30.
[122] TI Launchpad 2021. Hardware Kits & Boards Design Resources. http://www.ti.com/design-resources/embedded-development/hardware-kits-boards.html Accessed: 2021-04-30.
[123] TinkerForge 2021. Tinkerforge homepage. https://www.tinkerforge.com/en/ Accessed: 2021-04-30.
[124] Nicolas Villar and Hans Gellersen. 2007. A malleable control structure for softwired user interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07* (Baton Rouge, Louisiana). ACM Press, New York, NY, USA, 49. https://doi.org/10.1145/1226969.1226980
[125] 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* (Seoul, Republic of Korea, 2015-04-18) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2991–3000. https://doi.org/10.1145/2702123.2702391
[126] Peta Wyeth and Gordon Wyeth. 2001. Electronic Blocks: Tangible Programming Elements for Preschoolers. *Proceedings of the Eighth IFIP TC13 Conference on Human-Computer Interaction* (2001), 8.

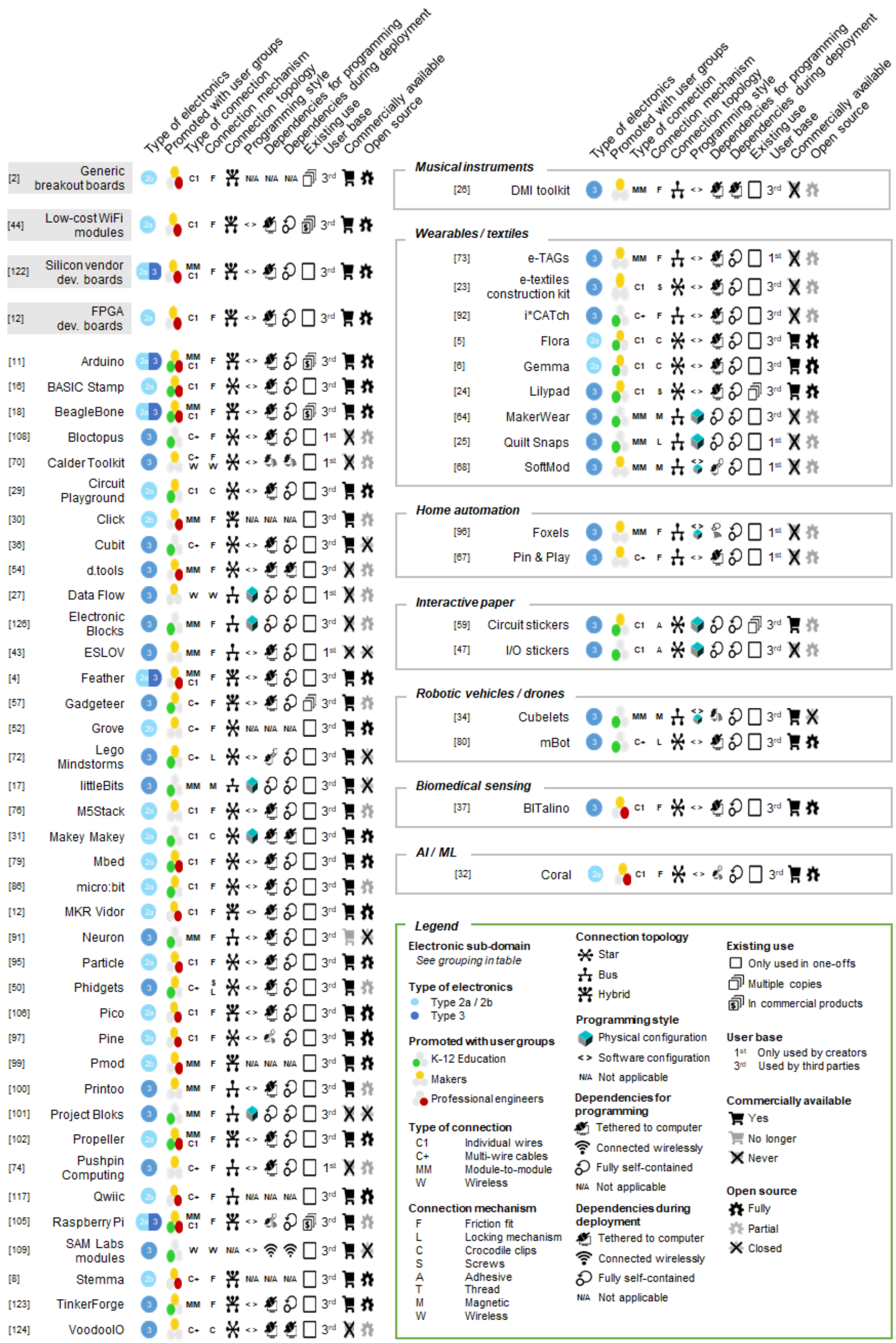

Fig. 8. A visualization of the full dataset. The four clusters, representing more than a single toolkit, have a gray background.