ProtoPCB: Reclaiming Printed Circuit Board E-waste as Prototyping Material

Jasmine Lu
University of Chicago
Chicago, Illinois, USA
jasminelu@uchicago.edu

Sai Rishitha Boddu
University of Chicago
Chicago, Illinois, USA
sairishitha.boddu@gmail.com

Pedro Lopes
University of Chicago
Chicago, Illinois, USA
pedrolopes@uchicago.edu

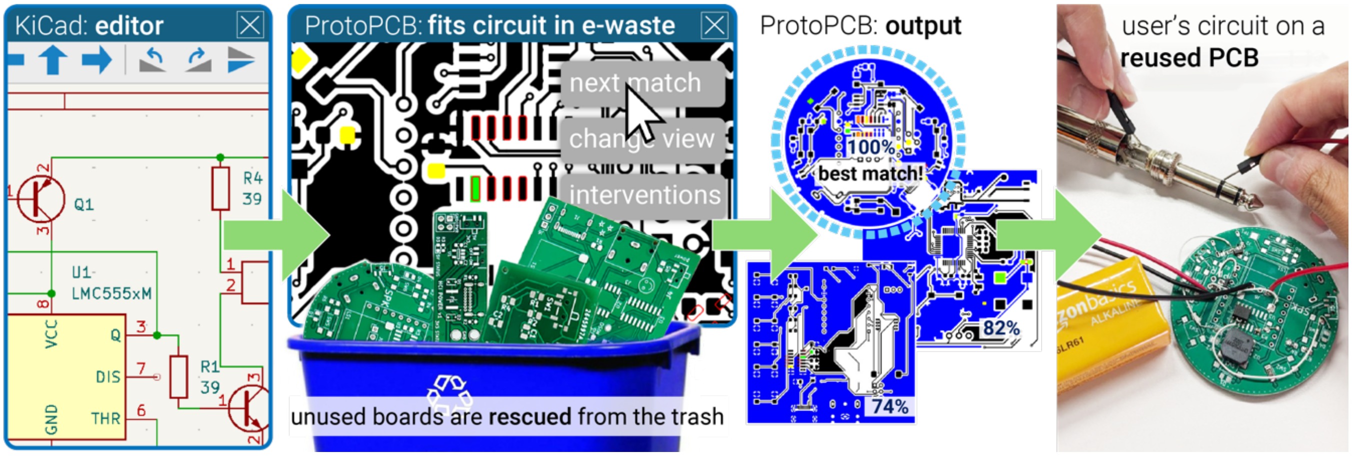

Figure 1: We propose an approach to reusing existing printed circuit boards (PCBs) rather than discarding them as e-waste. We realize this via our interactive tool, which analyzes discarded PCBs and finds ways to physically fit a user's desired circuit on them.

Abstract

We propose an interactive tool that enables reusing printed circuit boards (PCB) as prototyping materials to implement new circuits—this extends the utility of PCBs rather than discards them as e-waste. To enable this, our tool takes a user's desired circuit schematic and analyzes its components and connections to find methods of creating the user's circuit on discarded PCBs (e.g., e-waste, old prototypes). In our technical evaluation, we utilized our tool across a diverse set of PCBs and input circuits to characterize how often circuits could be implemented on a different board, implemented with minor interventions (trace-cutting or bodge-wiring), or implemented on a combination of multiple boards—demonstrating how our tool assists with exhaustive matching tasks that a user would not likely perform manually. We believe our tool offers: (1) a new approach to prototyping with electronics beyond the limitations of breadboards and (2) a new approach to reducing e-waste during electronics prototyping.

CCS Concepts

- **Hardware** → Printed circuit boards; PCB design and layout.

Keywords

Printed circuit boards, electronics prototyping, electronic waste, reuse

ACM Reference Format:
Jasmine Lu, Sai Rishitha Boddu, and Pedro Lopes. 2025. ProtoPCB: Reclaiming Printed Circuit Board E-waste as Prototyping Material. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3706598.3714095

1 Introduction

Electronics prototyping using printed circuit boards (PCBs) has become commonplace not only in industry but even for students/hobbyists. The rise of fabrication houses that produce small batches of PCBs at affordable costs, the advancement of PCB design tools, and even personal fabrication made designing PCBs much easier.

Despite these advancements, manufacturing a PCB is still *slow*, *expensive*, and, importantly, *wasteful*. This is further exacerbated given that prototyping typically involves iterating on a design, requiring a new PCB to be produced with every design iteration—creating even more cost, time, and e-waste. More critically, while PCBs offer advantages for prototyping electronics (e.g., enabling the use of both surface-mounted components and through-hole components, being smaller and more compact than breadboards, etc.), they are difficult to use during *rapid* prototyping—quickly iterating through prototypes becomes virtually impossible when designing a PCB.

This work is licensed under a Creative Commons Attribution 4.0 International License.
CHI '25, Yokohama, Japan
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1394-1/2025/04
https://doi.org/10.1145/3706598.3714095

Instead, we explore the idea that electronic design tools could *reuse* printed circuit boards (PCBs). Whereas previous works have explored how interactive tools can support reuse of *electronic components* from e-waste, we extend this work into the domain of *PCB* reuse. PCBs can be sourced from older prototypes, broken appliances, electronic waste, and more—essentially any modern device contains a PCB. To enable PCB reuse, we engineered a novel computational tool, *ProtoPCB* (Figure 1), that identifies how a desired circuit could be implemented on an existing PCB. ProtoPCB does this by: (1) leveraging computer vision to locate exposed solder pads, vias, and traces; (2) employing our search technique to find possible locations to *physically* fit components from the new design onto the discarded PCB; and (3) identifying methods of modifying the PCB board to enable more reuse (e.g., cutting a trace or adding a wire).

As we found in our technical evaluation, all these approaches enable our tool to find ways to retrofit a user's target circuit into existing PCBs. We found that across nine PCBs with varying degrees of electronic complexity, it was possible to fit ~82% of the desired components on other PCBs. With our tool, engineers no longer need to waste their old PCBs but, instead, can reuse them in their prototyping process. This offers two benefits: (1) a new recycling strategy for e-waste by recycling PCBs; and (2), rapid prototyping with surface-mounted components (without buying breakout boards or making a new PCB at each iteration).

2 Background and Related Work

To situate this work, we provide an overview of key areas that it builds on: (1) interactive tools for electronics design, (2) electronics prototyping, and finally, (3) sustainable HCI efforts.

2.1 Interactive tools for electronics design

While every step of the electronics design process has specific tools (e.g., circuit simulation, bill of materials generation), most relevant to our work are interactive editors for circuit design and PCB design.

Given the importance of these tools, many researchers have extended their interactive functionality to better assist users. For instance, Lin et al. explores how such editors can be reimagined by treating individual components as abstracted modules to highlight their most important electrical characteristics [13]. In SVG-PCB, authors explored a parametric approach to designing PCBs, allowing designs to be easily edited and remixed [17]. Pinpoint presents a workflow for developing PCBs and test rigs to enable techniques to debug and test a PCB [20]. ARDW also enhances the PCB testing experience by employing augmented reality projections that highlight relevant board features during the process [4]. Other work focuses on enabling electronics prototyping in new domains such as 3D-printed surfaces or curved breadboards [22, 30]. In our work, we are similarly interested in extending electronics tools and prototyping practices, but we are focused on how such tools could enable *reusing PCBs*.

2.2 Prototyping with electronics

While the focus of our work is on engineering a computational method to extend the life of PCBs, one of its most promising applications is for electronics prototyping. Thus, we discuss common approaches in this area.

There are many techniques employed in electronics prototyping. Notably, the HCI community has been at the forefront of new electronics prototyping toolkits and strategies to make prototyping more accessible [2, 11, 18]. We overview four common techniques: (1) breadboards, (2) protoboards, (3) breakout-boards, and (4) PCBs.

Breadboards are beginner-friendly, fast, and accessible as they do not require soldering. Breadboards work by inserting components with pins, called *through-hole components* (THT), in dedicated sockets, which connect it to other components in the same row. Breadboards are typically only used for the early stages of prototyping since they are bulky, use many wires, and are not resistant to mechanical stress.

Protoboards (or perfboards) mimic some of the breadboard's functionality but require soldering. These enable a more compact implementation with similar tradeoffs (only suited for through-hole components). However, modern components are manufactured in smaller packages, called *surface-mounted components* (SMD), without the long pins that can fit in breadboards/protoboards; instead, these SMDs are meant to be soldered in printed circuit boards (PCBs).

Breakout boards are pre-made PCBs that fit a specific SMD component footprint (i.e., a standard size, such as SOIC-8, SOT-23, etc.). They feature pads to solder the SMD and traces that extend to pins (for use with breadboard/protoboard). While these allow using SMDs, engineers often require a different breakout board for every SMD footprint (or, in the best case, breakout boards can fit a wider range of sizes within one standard footprint, e.g., a TQFP32 breakout that can fit ICs with variable sizes as long as they are of this class). As such, users are left with few options to rapidly prototype with SMDs, since they would need to manufacture a PCB. While there are advances in speeding up PCB printing (e.g., inkjet-printing [9, 21], conductive ink machines [19, 24], or milling machines [6]), *these will further exacerbate the e-waste issue* by creating more e-waste.

Combining with existing techniques. We see ProtoPCB as a new prototyping method that allows engineers to reuse *already manufactured PCBs* when prototyping circuits. As we will detail later (see Section *5*) we envision many ways that engineers can use ProtoPCB. While in its most extreme form, a user might implement an entirely new circuit in a reused PCB using ProtoPCB (no breadboards or breakout boards), the most likely way is to use ProtoPCB in concert with the aforementioned, established approaches. In fact, we believe that ProtoPCB is especially synergetic with breakout boards. One such example is how ProtoPCB can be leveraged to rapidly "fabricate" breakout boards—without the need to buy one anew—by fitting a new circuit or even a single component into an existing PCB, one can also think of ProtoPCB as a companion to breakout boards.

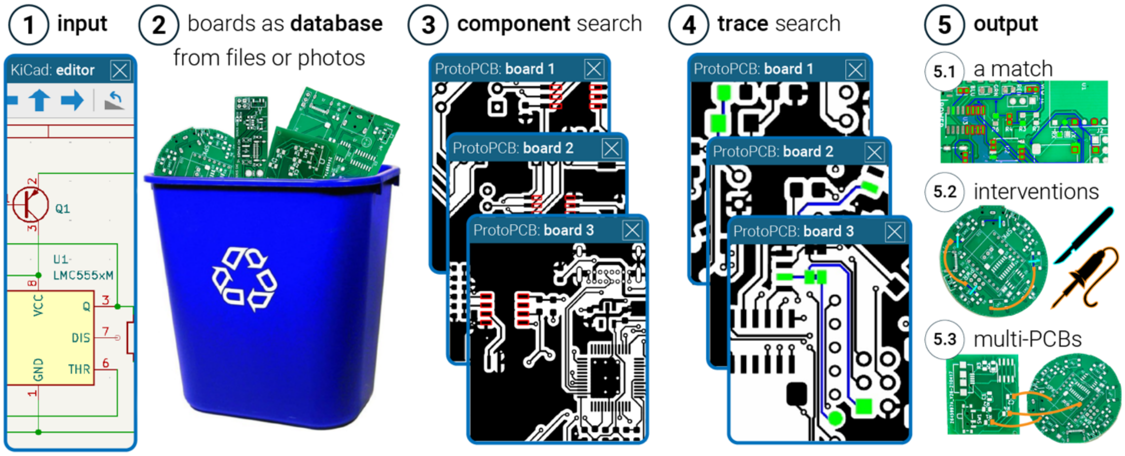

Figure 2: Our approach and its general workflow: fit a user's desired circuit into an old PCB—enabling PCB reuse.

2.3 Efforts in Sustainable HCI for reducing e-waste

Last, our work is informed by growing efforts in sustainable interaction design [3] and sustainable computing [7]. Such "calls for action" have emphasized the need for more circular ecosystems (including manufacturing, maintenance, and recycling). Various HCI researchers have responded to these by proposing new techniques for fabrication or electronics.

Waste reduction and reuse in fabrication. Making multiple iterations of designs often generates significant waste [25, 28]. One example is prototyping with 3D printers, which generates waste with every iteration or failed print. Researchers mitigate this by reclaiming filament [16], using waste as infill [17] or as supports [18], or even printing only wireframes [15]. Similar ideas have been explored in other domains (e.g., laser-cutting [1], sewing [12], etc.).

Waste reduction and reuse for electronics. E-waste is the *largest growing waste stream in the world* with upwards of 60 million tons annually [5], which has motivated researchers to explore new alternatives. For instance, *FabricatINK* leverages recycled E-ink to create bespoke displays [8]. Similarly, Mandel and Ju discuss the potential of sourcing e-waste across several case studies (e.g., reusing motors from discarded hoverboards) [15, 16]. Others have explored how the materials in electronics themselves can be made more environmentally friendly, such as by using biodegradable materials [1, 10, 23]. Last, recent works explored how this can be done during electronics prototyping. For example, *DeltaLCA* explores how PCB designs can be profiled for their carbon footprint [29]. *SolderlessPCB*, explore how components can be secured to PCB using screws rather than soldering [26] and *PCB Renewal* explores how to create PCBs with modifiable traces [27]. Finally, *ecoEDA* provides suggestions of which components users can reuse from their e-waste [14]. In ProtoPCB, we turn our focus to PCBs (rather than components like in *ecoEDA*) and explore computational strategies that can be used to elevate their reuse potential.

3 Our approach: reuse a PCB to implement a (completely different) circuit

We introduce a new concept in which discarded printed circuit boards (PCBs) are re-used as materials to create new circuits. However, this idea is *not* something that a user can be expected to perform manually, since a typical PCB has hundreds of electrical connections (pads, traces, vias)—making the task of fitting a completely new design on an old PCB daunting if done by hand or manually in a PCB editor (as shown in our technical evaluation, the electrical connections on an Arduino PCB is on the order of thousands).

Workflow. Our tool uses the workflow depicted in Figure 2 (1) takes a user's schematic as input (e.g., a KiCad file); (2) takes any number of PCB boards as its database (these can be screenshots, KiCad files, or even photos); (3) analyzes each PCB to find if the user's components can be fit on it; (4) recursively analyzes each connection in the user's circuit and searches for an equivalent trace on the PCBs; (5.1) if a PCB completely implements the user's circuit, which would be less common, the process ends; or likely, (5.2) a PCB could implement the circuit with some interventions (e.g., cutting a trace or adding a wire), or (5.3) by stitching several PCBs together using wires.

How does this work? Our tool is able to retrofit a circuit onto a PCB even if it is very different from the PCB's original circuit because: (1) it abstracts away from the semantic level, e.g., our tool does not need to fit a resistor where a resistor was originally located—instead, it finds a new copper location that physically fits the resistor's footprint; (2), it *exhaustively* find ways to retrofit without worrying about the boundaries of components – for example, instead of seeing two resistors side by side, our tool sees four copper locations which can fit a voltage regulator; (3) it explores more than just traditional component pads, it considers any exposed copper as a site to solder a new component—this includes any test pads, touch pads, pads for connectors, and much more; finally, (4) while modern PCBs appear to be specific to a single implementation, many aspects of PCBs are also standardized - e.g., many components share the same footprint allowing our tool to find matches if the desired circuits also used these footprints.

PCBs as input data. ProtoPCB can take input from different data formats, including (1) design files of former prototypes (KiCad files), (2) PCBs found online (as screenshots), or (3) photos of PCBs found in e-waste. All the PCBs in the Section 5 were taken from the e-waste of an undergraduate-level electronics class (we had no control over the designs of these PCBs, they were done by others, none of the authors). Finally, examples in our evaluation are from PCB boards found in the *Open-Source Hardware* certified repository.

Contribution & Benefits. Our main contribution is providing a new method of prototyping where printed-circuit-boards (PCBs) are reused for prototyping instead of discarded as e-waste. We instantiate our concept by developing an interactive tool (ProtoPCB) that recursively matches a user's circuit to a database of PCBs. ProtoPCB is designed for users with experience with electronics prototyping, namely with soldering and PCB design; this could encompass not only engineers & researchers but also advanced makers.

Our approach provides key benefits: (1) it abstracts PCB designs away from their specific components, footprints, and file formats, instead our approach focuses on the geometry of the *copper* on the PCB—allowing to increase the chance of reuse; (2) we introduce a method that computationally solves the search of a circuit on a target PCB—being able to traverse a sizeable search space (pads × traces × components × possible locations × possible interventions)—achieving this manually would be frustrating and time-consuming; finally, (3) it reduces the *waste* and costs of prototyping.

4 Implementation

ProtoPCB is implemented in *Python* using *OpenCV*. It was engineered to read data from *KiCad* design files and/or images. All code, datasets, and evaluation data are open source[1]. Next, we offer an overview of the algorithmic implementation of our tool.

4.1 Obtaining PCBs for the database from images or design files

To add a PCB to the ProtoPCB database, users can provide either **(1) a design file** (e.g., a KiCad *.pcb* file), which ProtoPCB renders to an image using information on the PCB layers that contain copper (*F.Cu, B.Cu*) superimposed with mask layers (*F.Mask, B.Mask*)—this is done via *cairosvg* in conjunction with *KiCad command line tools*; or, **(2) a photo (or screenshot) of a PCB**, which requires an additional step of pre-processing to binarize the photo—using OpenCV's *Canny* edge detection, followed by *cv.findContours*, and *cv.threshold*, which finds lighter areas that represent the copper (i.e. pads, vias, and traces). Additionally, and optionally, once a PCB's photo is displayed on ProtoPCB's GUI, the user can inspect detected areas (e.g., pads/traces) and indicate if they want to ignore these (e.g., ignoring silkscreen labels or drawings). Finally, the processed representation of the board is added to ProtoPCB's database—i.e., the user can now use ProtoPCB to fit new circuits into this board (as depicted in Figure 3).

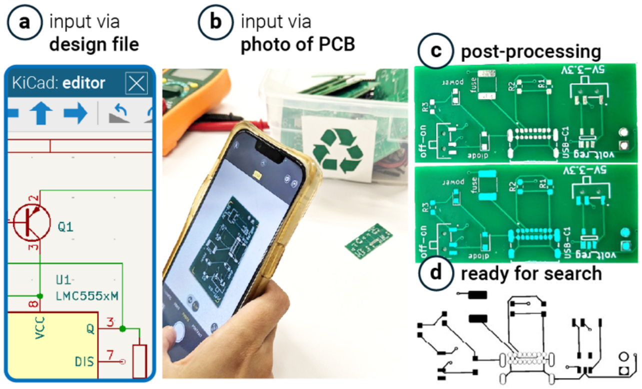

Figure 3: Two possible input formats for the PCBs in the database: from (a) design files or (b) screenshots/photos of PCBs.

4.2 ProtoPCB algorithmic search

Iterating over the user's desired circuit. ProtoPCB iterates over each component in the user's desired circuit—a schematic file (*.sch* in KiCad). To improve matching of the hardest case, we always start with the component with the most pins (e.g., a microcontroller). For each component, we extract its footprint as an image—using the previous method we described to load *.pcb* files (using *cairosvg* on the footprint) and pass it to our component matching algorithm.

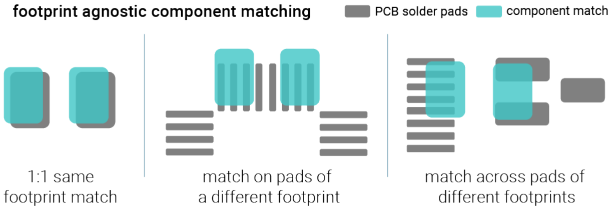

Figure 4: Using a footprint-agnostic component matching approach, ProtoPCB does not need to match components to their exact footprint but, instead, can find diverse placement options that maximize the solderable pad area for component pins.

Component matching. We use the image representation of a component's footprint as a mask (*cv.findContours*) and move it across a target PCB's image, at steps of 1/16 of the minimum dimension of the footprint. As we pan, ProtoPCB intersects (using *cv.bitwise_and*) the mask with the target PCB's copper, saving any location where all pins overlap with PCB by more than 50% (a user-defined heuristic via our advanced settings in case users prefer more or less solder coverage). After the mask is panned across the whole PCB, ProtoPCB repeats this for different rotations of the component (0, 45, 90, . . ., 315 degrees), maximizing the possible ways a component can be placed. Using this approach, component placement matches will include placements along the same footprint but also on different footprints and combinations of footprints, as long as most of each pad is covered as seen in Figure 4.

---

[1] Code & datasets (e.g., scripts, PCB database, and output of ProtoPCB on each test case): https://lab.plopes.org/#ProtoPCB

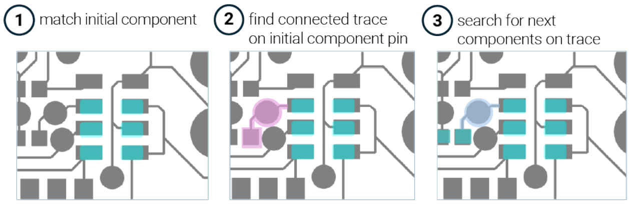

Figure 5: ProtoPCB finds ways of implementing a circuit net by (1) identifying a placement match for an initial component, (2) looking at the connected trace of the relevant pin on that component, and (3) searching for component matches that touch pads on that trace. Using this strategy, ProtoPCB maximizes the utility of existing traces on the PCB board.

Matching PCB traces to connections. For each location on a PCB that can fit one of the user's desired components, ProtoPCB will search for traces that can accommodate the electrical connections needed to implement the user's schematic. We follow the traces that connect to each previously intersected pad (using *cv.findContours*) but also follow connections that go through the back side of the PCB (through holes called *vias,* which we locate using its KiCad *.drl* drill file and *contour hierarchies*). Figure 5 shows an example of how we expand from (1) one pad of the component (in turquoise); (2) find the connected trace and pad (in pink); and, finally, (3) explore matches on that pad for the next component (recursively, ProtoPCB will execute this for all pads).

Recursive search. This is realized with a *greedy* search approach, starting from an initial component (most pins), finding initial component matches, finding all nets that touch the initial component, and then continuing with the next component (and its nets) with the largest number of pins that have not been covered by the search. This process continues until the full circuit is realized. A circuit is only considered completed if: (1) shared components are matched on the same location, (2) different components do not use the same pad (i.e., are not placed on top of each other), and (3) different traces are used for each net to ensure no short circuits (results that fail these tests are discarded and never shown to the user).

Adding interventions. If a full match is not found, the search continues to find *matches with interventions*. Matching continues as before but whenever a component pin cannot be matched on the board (e.g., due to a conflicting trace or inability to find component matches on its traces), ProtoPCB sees if a **match can be made by adding a wire** (placing the component on a different trace, but adding a new connection) or **by cutting a trace** (isolating pads from a trace, if connecting to the pad would cause an unwanted connection). Adding a wire is done by searching on pads outside of the trace, verifying that the addition of this component placement does not conflict with existing nets, and then storing the added component as if it was another node connected on the trace but with a note that an intervention would be needed. Cutting a trace on the other hand requires changes to the board itself. To do this, nodes are isolated by drawing a ring around impacted pads and identifying the necessary cuts to surrounding traces (using OpenCV draw functions and *findContours*). Once the cuts have been applied to the board, the board's map of traces and pads gets re-created using *findContours* so that new traces created by trace cuts are treated as separate. If one of these interventions is found, the search continues (greedily and recursively) until the full circuit is realized.

4.3 Additional search options

In addition to the standard search method, it is also possible to activate modes to "allow partial matches" and/or "use multiple boards" that expand the search parameters.

Partial matches. ProtoPCB also includes a feature to access the "partial matches" which showcases an incomplete circuit match that may be missing an electrical connection (no wire could realize it) or missing a component. Through this feature, users can identify how much of their circuit they can implement on a given PCB board and how much of their circuit they may have to find additional boards for or connect via protoboard, breakout board, or breadboard.

Stitching multiple boards. Analogously, users can use the tool to split up their input circuit into subcircuits and match those subcircuits on separate boards. Such a strategy allows for extremely diverse types of PCBs *and* components to be used as users can stitch together various boards that satisfy different parts of the circuit. These will need to be manually connected by the user soldering wires.

5 Examples of Prototyping with ProtoPCB

To showcase our tool, we describe examples of how it can be applied for electronics prototyping: (1) to create breakout boards, (2) to create circuits rapidly on PCBs, and (3) to evaluate a PCB's reusability during design.

5.1 Creating Breakout Boards

If a user is prototyping with breadboards and wants to use an SMD component, they typically resort to a breakout board (a pre-made PCB that fits the target component). This requires ordering it, manufacturing it, or having it in ample quantities around. Figure 6 illustrates how to use ProtoPCB to rapidly make a breakout board for an SMD headphone jack, which has an unusual footprint while prototyping an Arduino-based synthesizer. In Figure 6, (a) the user loads this component's footprint into ProtoPCB (which takes photos or KiCad files); (b) ProtoPCB searches through a library of discarded PCBs, and finds one PCB that can fit the target component in 39 different locations; (c) shows the user the best location (i.e., best overlap of copper under the pads); and, finally, (d) our user solders four cables and connects it to their breadboard, continuing to refine it. Note that our user could stop working and order this breakout board (*Sparkfun* sells a specific breakout board just for this component [31]), yet this would not only be *wasteful* but also antithetical to the act of *rapid* prototyping.

While we think this process is especially useful for creating breakout boards for components with unusual footprints (e.g., SMD audio Jacks, MEMS microphones, etc.), it also allows for producing breakout boards for standardized footprints (e.g., create a TQFP-32 breakout for the popular ATMEGA328) and reducing waste.

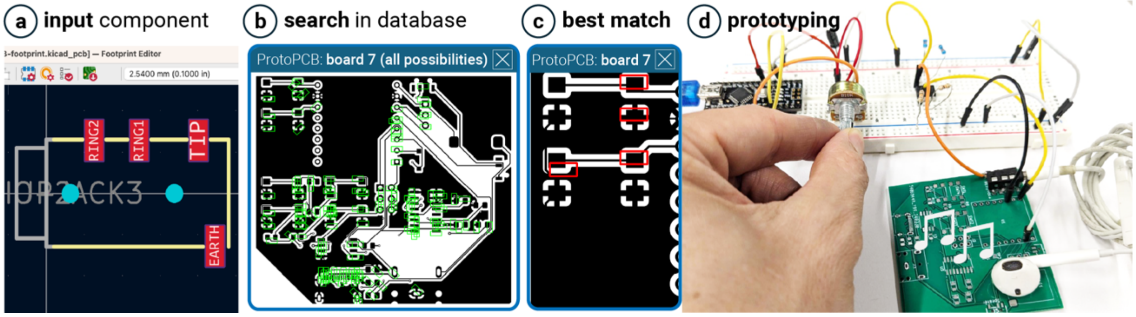

Figure 6: Creating a breakout board for an SMD headphone jack with an unusual footprint in just a few minutes via ProtoPCB.

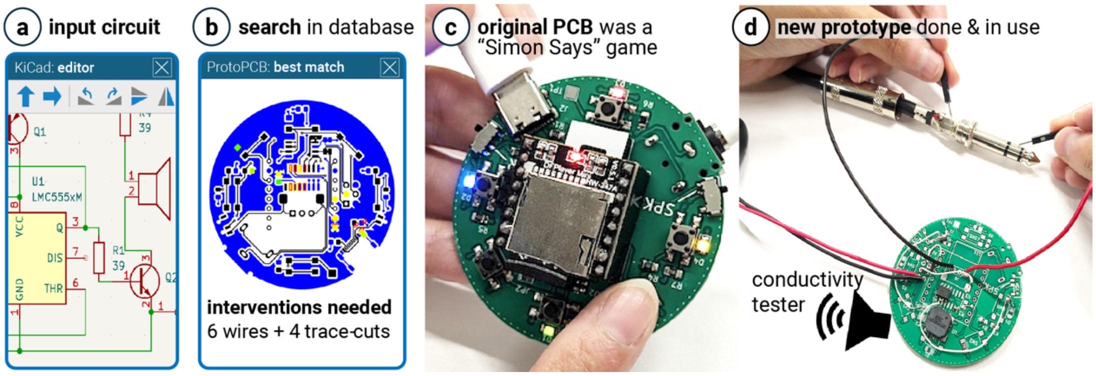

Figure 7: This user implements their conductivity tester circuit *directly* and *rapidly* in a PCB, by reusing one of their previous PCBs.

5.2 Prototyping *with* PCBs

While PCBs are not typically seen as prototyping materials, they excel in other qualities such as their ability to support both SMD and THT components, their robustness due to soldering, and even lower noise. In Figure 7, we illustrate an example of prototyping with ProtoPCB: (a) the user loads a schematic for a conductivity tester (a device that makes a sound if an electrical connection is present—designed using transistors and an LM555); (b) ProtoPCB searches the database of discarded PCBs and identifies one that could be modified (using six wires and four trace cuts) to realize this circuit immediately; (c) this PCB was left over from the design of a "Simon Says" game that the user has previously implemented (this is a functional device made by someone else, not by the authors); (d) finally, after soldering the additional wires and cutting traces (with a scalpel, a typical PCB technique)—the user now has a first iteration of this idea, already in a PCB. In fact, as any engineers used to working with PCBs, we grew accustomed to waiting week(s) to receive a new PCB from the factory. While the method shown here might appear slow, one can add wires/cuts in minutes meanwhile no tool to date can produce a PCB reliably in this short amount of time.

5.3 ProtoPCB to estimate *future* re-uses

Finally, while our previous examples depicted the usefulness of our tool for rapid prototyping, we turn to a conceptually unique way to use ProtoPCB: as a tool to evaluate how amenable a PCB is for *future* reuses (depicted in Figure 8). To this end, users can run ProtoPCB on a PCB they are currently designing to estimate how much it can implement other input circuits. We believe this can used by future end-users as a kind of reuse-*benchmarking* for their own PCBs.

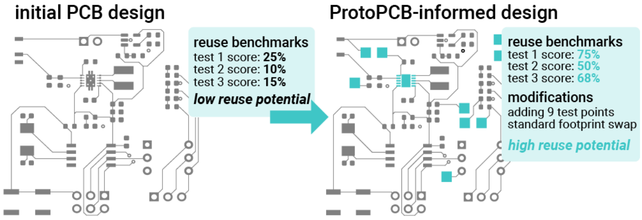

Figure 8: ProtoPCB can be used to benchmark PCB designs for reusability by testing how many other input circuits (as part of a benchmarking suite, for example) could be implemented on this version of the PCB. In this example, after running ProtoPCB on a benchmarking suite, our user sees low scores indicating a low reuse potential and makes simple modifications (e.g., adding test points or using a more standard footprint). Now, this new design scores on this reusability benchmark.

Table 1: Overview of electrical features and complexity of boards used in evaluation.

PCB Number	PCB 1	PCB 2	PCB 3	PCB 4	PCB 5	PCB 6	PCB 7	PCB 8	PCB 9
Board Name	LM4040 Breakout	ACS72x Breakout	VEML6070 Breakout	BioAmp EXG Pill	Motor Driver	LSM9DS1 Breakout	Circuit Playground	CANtact	Arduino UNO
No. Components	7	7	7	26	19	20	74	90	111
Number of Nets	4	6	6	17	26	22	56	52	62
Number of Pins	18	19	21	67	78	99	250	348	298
Complexity[a]	504	798	882	29614	38532	43560	1036000	1628640	2050836

[a] Complexity calculated by multiplying number of components, nets, and pins.

6 Technical evaluation

To validate our tool, we took nine PCBs from the *Open Hardware Association*'s repository [32] (open-source hardware certified, both schematic and board files are open-sourced). These PCBs were selected randomly but curated to ensure diversity across three dimensions: pin count, number of components, and number of nets as shown in Table 1. We group them as low, medium, or high complexity by multiplying those dimensions to estimate complexity. As shown in Figure 9, these boards span a wide variety of uses, from simple breakout boards to EMG amplifiers, or even microcontrollers (e.g., the popular Arduino UNO).

Method. We ran ProtoPCB with each board as input (desired schematic) and all other boards as database—excluding: (1) the input board from the database and (2) the boards with lower pin count than the desired circuit—**this yielded 32 test evaluations** (i.e., ProtoPCB was asked to fit every desired circuit into a PCB of higher complexity). The goal was to determine if ProtoPCB could match components, traces, and find interventions to realize these circuits.

6.1 Overall results

Figure 10 details the percentage of components that were matched per input circuit for all PCB boards (calculated as the ratio of: number of matched components / number of components in the input circuit). We observed that across all 32 test-cases, ProtoPCB was found matches for an average of 82% (STD: 6.9%) of the components in the desired circuit.

As expected, no circuit fits perfectly in another circuit—this is understandable in that these served very distinct purposes, were created using different sizes (a bigger board can match more circuits by virtue of its size alone), components, and so forth. First, ProtoPCB was able to find five PCBs (~14% of the cases) that realized the entire circuit with interventions (we will detail these in the next section by analyzing two examples in detail). As expected, these full matches with interventions were cases where a lower complexity circuit (e.g., a breakout board for the LM4040 or a breakout board for the VEML6070) was matched against the most complex ones (e.g., the Arduino UNO). Next, ProtoPCB was able to realize the majority of circuits (~72%) at >80%-99% of matching. Finally, a smaller fraction of the circuits (~28%), were matched at 50%-80% of completeness. The complete dataset of our evaluation (i.e., ProtoPCB's output for all the 32 tests depicted in Figure 10) will be made available[1], to enable future researchers not just to build on ProtoPCB's codebase, but also to leverage these tests as benchmarks for new developments.

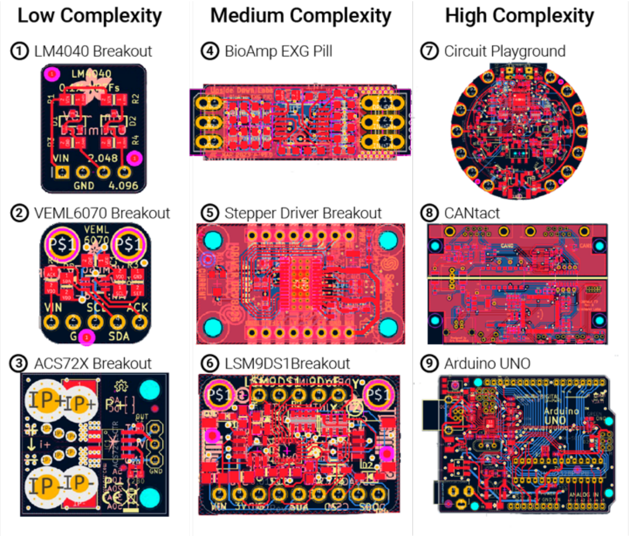

Figure 9: We use nine boards of varying complexity to evaluate our approach. These boards were grouped into categories of low, medium, and high complexity. Notably, these boards are diverse in terms of their complexity as well as their functionality.

6.2 Examining interventions in two exemplary cases

For the sake of visual clarity, we do not present the output for all 32 cases, these can be found in our repository[1]. We present two examples that were *matched with interventions* in detail, illustrating how ProtoPCB modified these PCBs to enable reuse

Utilizing the PCB's existing traces. For the case of matching the VEML6070 Breakout circuit on to the BioAmp EXG Pill PCB, almost all the input components were able to be matched except U1 (the UV sensor), and JP1, the connector pads. All remaining connections and components were fully implemented through existing underlying traces. A selection of those connections is visualized in Figure 11. Notably, ProtoPCB searches for ways to maximize

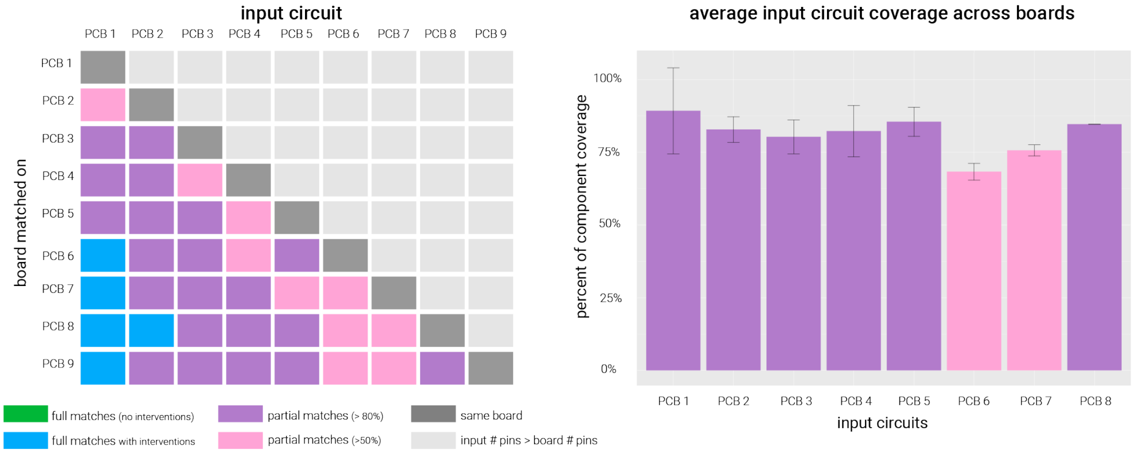

Figure 10: On the left, we visualize the degree of completeness of ProtoPCB's matches between input circuit and possible PCBs. On the right, we visualize the average percentage of component coverage of input circuits.

circuit implementation without needing to consider what the net was originally. For instance, on the top, the input-circuit's VDD net is realized by using the GND net (a large ground "pour" net). In the ProtoPCB identified match, components are placed to utilize trace connections optimally.

Identifying interventions. We now turn to our second example to examine some examples of interventions. In Figure 12, we illustrate the interventions needed when matching the LM4040 breakout as input circuit against the LSM9DS1 PCB. Notably, most of the circuit is still implemented via existing PCB traces (highlighted in blue), but certain points needed intervention. In this case, ProtoPCB found that with 5 added wires (represented in orange) the input circuit could be fully realized on this database PCB.

7 Discussion

From our technical evaluation, we found that ProtoPCB is a promising approach to extending the life of discarded PCBs, by matching them with input circuits. While our evaluation could not possibly explore all the variability within PCB design, it used boards that were chosen due to their mainstream availability (all these boards are already popular with engineers & makers) and represented a fair spectrum of complexity (e.g., with ~100 components). While, on average, circuits were matched against other PCBs with 82% (STD: 6.9%) completeness, we did not observe perfect matches. We foresee that such perfect matches are only likely to happen in a smaller subset of simpler input-circuits (comprised of only simple standardized components, e.g., LEDs, in low numbers). Nonetheless, the promise of our tool comes not only in perfect matches but also matches with interventions (fully realizing a circuit by adding wires or cutting traces) and even partial matches (e.g.., rapidly creating breakout boards, leaving some components out to test later, etc.).

Limitations. Our approach is not without limitations. First, our speed is still suboptimal (usually minute/s), which stems from our greedy search process and no parallelism. Additionally, our tool's usefulness is limited by the PCBs a user has, so it is most appropriate for users who often create boards (e.g., engineers & advanced makers). While our tool can also be used on PCBs with soldered components, reusing such PCBs would require desoldering which can be done through multiple approaches – soldering iron, hot air gun, hot plate, or reflow oven (for melting solder and displacing all components on the board). Additionally, ProtoPCB reads two-layer PCBs and does not account for connections on invisible inner layers; this current limitation restricts some of its ability to operate on 'found' boards where users do not have access to design files. In some cases, users may be able to reverse engineer such board via electrical testing, but this could be a challenging process. As shown by our technical evaluation, ProtoPCB may not always be able to find a match on a PCB for all the desired components. In such a case, the user would need to either adjust their design, search for methods of combining multiple boards, or opt to not reuse PCBs for their project. Moreover, our approach is best suited for cases in which the input circuit is of lower complexity than the e-waste PCBs—otherwise, it is unlikely to find useful matches (see *Technical Evaluation*). Finally, ProtoPCB works best for circuits without challenging signal integrity requirements.

Our approach to improving the sustainability of electronics prototyping focuses on remediating the e-waste problem with a new pathway for PCB reuse other than material-recycling. We see this as a worthwhile approach provided how most PCBs (with or without components soldered onto them) are recycled in bulk without regard for the modular functionality of the boards vs. components. ProtoPCB demonstrates that the PCBs themselves can continue to be useful beyond being shredded and mined for their raw materials. While this approach to sustainability is not a one-size-fits-all, as it still depends on the production of PCBs and does not address the perhaps excessive production of PCBs, ProtoPCB does extend the lifetime and utility of PCBs.

Future work. As a new approach, additional work should follow to explore the boundaries of PCB-reuse; however, we see ProtoPCB as the first step to computationally reclaiming e-waste PCBs. As next steps, we believe there are significant performance gains to

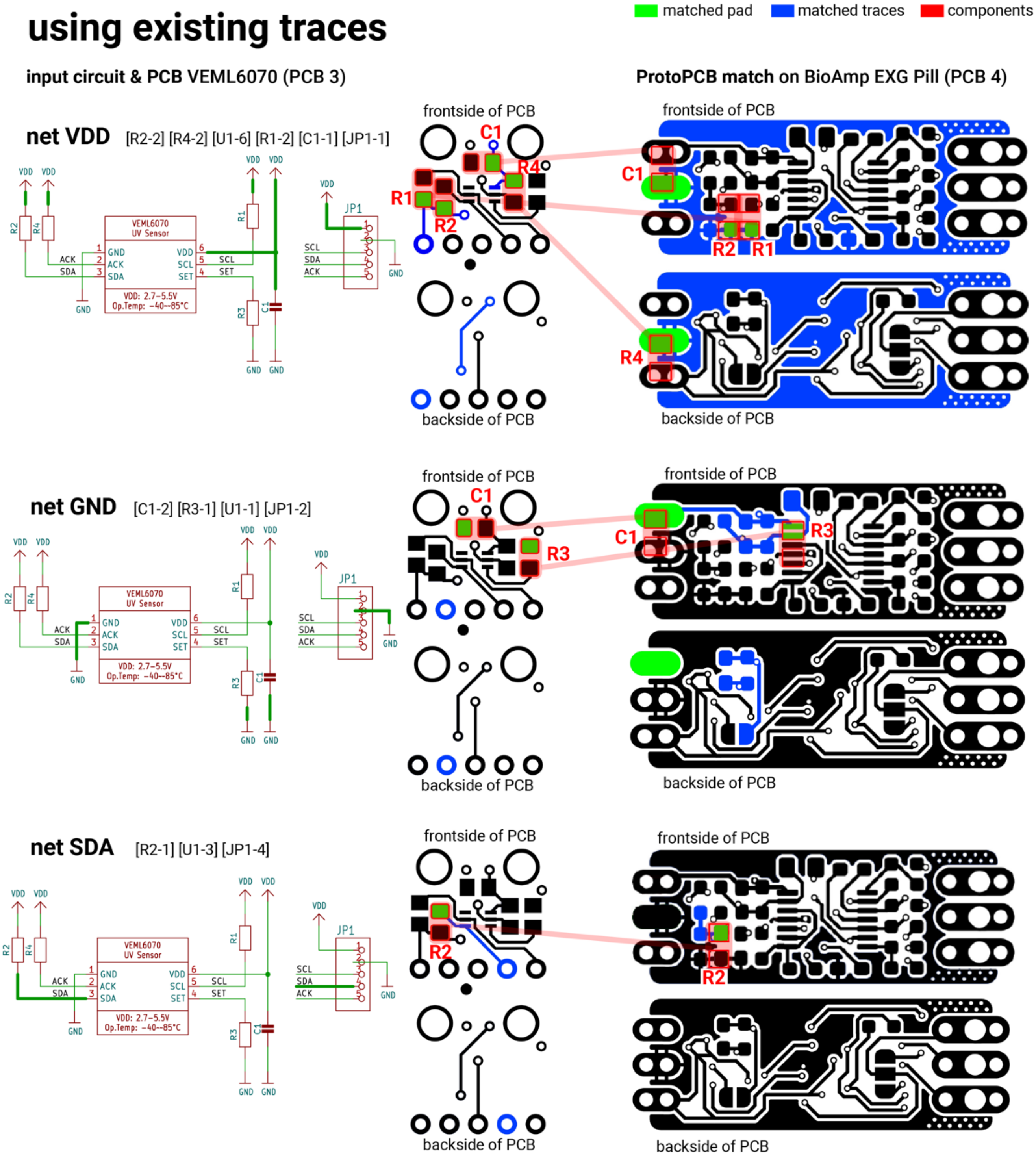

Figure 11: In the case of matching the VEML6070 Breakout circuit to the BioAmp EXG Pill PCB, most of the circuit connections could be realized by using existing traces on the PCB. We highlight these cases with the VDD, GND, and SDA nets.

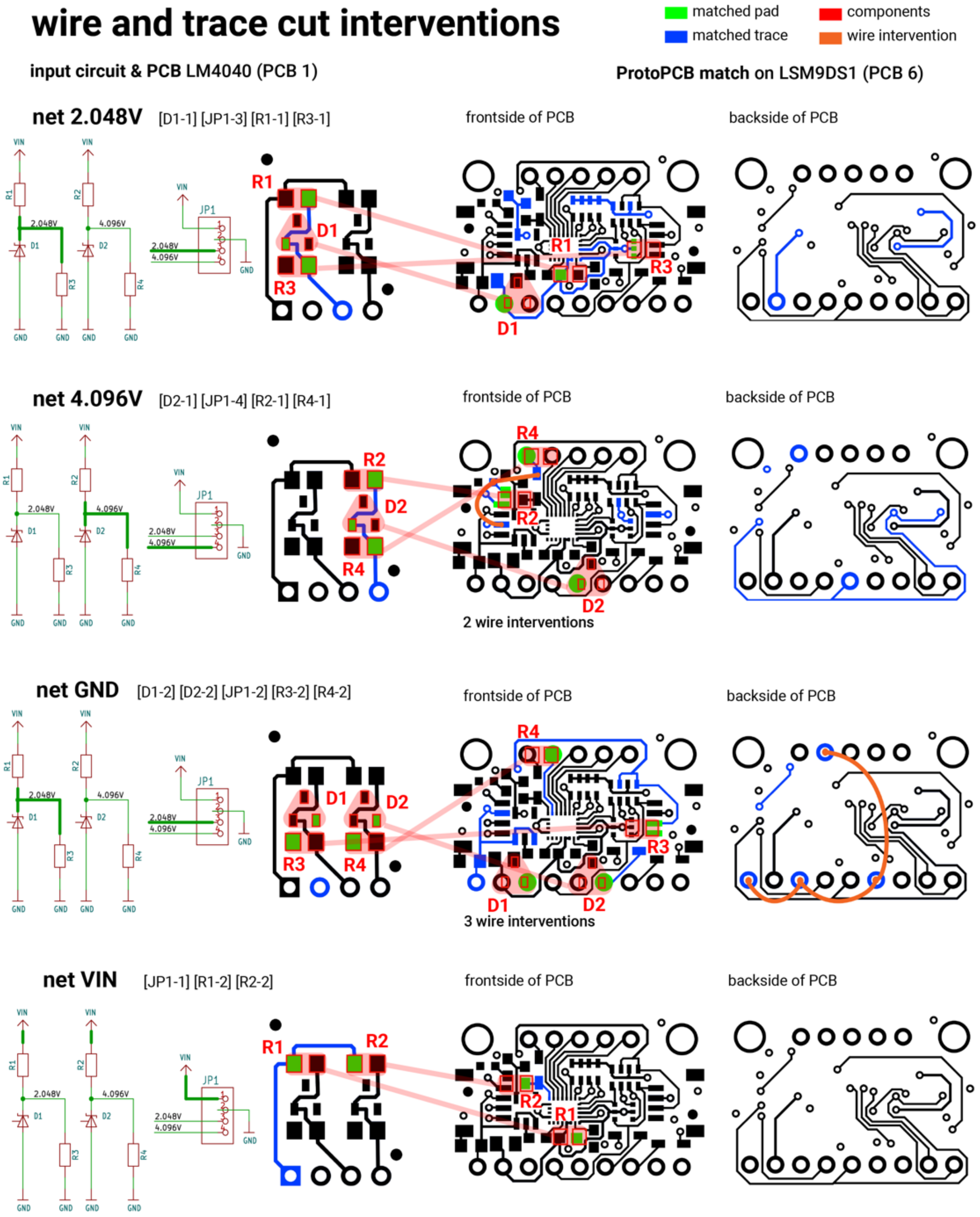

Figure 12: We illustrate the example of matching the LM4040 breakout board onto the LSM9DS1 breakout board. To implement the circuit on the board, ProtoPCB identified a combination of using existing traces, adding wires, and cutting traces that are needed. We highlight the interventions needed along each net.

be had by parallelizing our algorithmic search strategy. Moreover, it may be worthwhile to incorporate additional heuristics based on common patterns in electronic design. Additionally, there are even more additional intervention techniques beyond wire/trace-cutting that can be exploited (e.g., combinations with fabrication machines like laser cutters or PCB mills). Finally, while ProtoPCB is primarily a tool to promote the reuse of e-waste PCBs, it might also provide additional benefits to users as a prototyping tool, each of these aspects depicts fertile ground for future explorations: **(1) Supporting the reuse of complex PCBs**—while our approach works best with simple circuits, it can also be useful when trying to reuse complex circuits as is the case of contemporary consumer electronics with extremely small components (i.e., small soldering pads, usually hard to reuse); in these cases, ProtoPCB's search strategy can combine multiple pads to make a single unified solder point, allowing users to reuse even complex PCBs. **(2) Supporting the reuse of non-standard components**—reusing non-standard SMD components (as featured prominently in *ecoEDA* [14]) is difficult without a PCB to solder them onto, but ProtoPCB makes this more accessible. **(3) Creating Stencils**—ProtoPCB can also be used as a way to produce *soldering stencil designs*. Stencil designs created with ProtoPCB would not copy the original PCB stencil but would be customized for the circuit components that the user is intending to solder on this PCB (e.g., only the necessary pad geometries to solder are exposed).

8 Conclusions

With ProtoPCB, we explored a novel interactive tool to assist users with identifying methods of reusing their previously discarded PCBs to implement new circuit designs. Our technical evaluation shows that across a variety of PCB designs, it is possible to implement circuits on PCB boards manufactured for entirely different purposes. As such, we believe our tool enables electronics engineers to integrate a new approach to prototyping with PCBs that centers around reuse rather than manufacturing a new board with every design iteration. As an approach to recycling electronics waste that capitalizes on the existing utility of the PCB, we hope our work can inspire similar approaches to computationally analyzing e-waste to identify how it can be reused in new projects.

Acknowledgments

This work is supported by a University of Chicago Climate Institute Seed Fund. We are also grateful to the many students across the years in Pedro Lopes' PCB Engineering class that generated a surplus of e-waste boards that we reused for our research. Additionally, we would like to thank the reviewers for their valuable feedback.

References

[1] Vicente Arroyos, Maria L K Viitaniemi, Nicholas Keehn, Vaidehi Oruganti, Winston Saunders, Karin Strauss, Vikram Iyer, and Bichlien H Nguyen. 2022. A Tale of Two Mice: Sustainable Electronics Design and Prototyping. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 1–10. https://doi.org/10.1145/3491101.3519823

[2] Thomas Ball, Peli De Halleux, James Devine, Steve Hodges, and MichałMoskal. 2024. Jacdac: Service-Based Prototyping of Embedded Systems. *Proceedings of the ACM on Programming Languages* 8, PLDI: 692–715. https://doi.org/10.1145/3656405

[3] Eli Blevis. 2007. Sustainable interaction design: invention & disposal, renewal & reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 503–512. https://doi.org/10.1145/1240624.1240705

[4] Ishan Chatterjee, Tadeusz Pforte, Aspen Tng, Farshid Salemi Parizi, Chaoran Chen, and Shwetak Patel. 2022. ARDW: An Augmented Reality Workbench for Printed Circuit Board Debugging. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–16. https://doi.org/10.1145/3526113.3545684

[5] Cornelis P. Baldé, Ruediger Kuehr, Tales Yamamoto, Rosie McDonald, Elena D'Angelo, Shahana Althaf, Garam Bel, Otmar Deubzer, Elena Fernandez-Cubillo, Vanessa Forti, Vanessa Gray, Sunil Herat, Shunichi Honda, Giulia Iattoni, Deepali S. Khetriwal, Vittoria Luda di Cortemiglia, Yuliya Lobuntsova, Innocent Nnorom, Noémie Pralat, and Michelle Wagner. 2024. *The Global E-waste Monitor 2024.* International Telecommunication Union (ITU) and United Nations Institute for Training and Research (UNITAR), Geneva/Bonn. Retrieved September 5, 2024 from https://ewastemonitor.info/the-global-e-waste-monitor-2024/

[6] Frikk H. Fossdal, Jens Dyvik, Jakob Anders Nilsson, Jon Nordby, Torbjørn Nordvik Helgesen, Rogardt Heldal, and Nadya Peek. 2020. Fabricatable Machines: A Toolkit for Building Digital Fabrication Machines. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 411–422. https://doi.org/10.1145/3374920.3374929

[7] Roch Guérin, Amy McGovern, and Klara Nahrstedt. *Report on the NSF Workshop on Sustainable Computing for Sustainability (NSF WSCS 2024).*

[8] Ollie Hanton, Zichao Shen, Mike Fraser, and Anne Roudaut. 2022. FabricatINK: Personal Fabrication of Bespoke Displays Using Electronic Ink from Upcycled E Readers. In *CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3491102.3501844

[9] 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* (UbiComp '13), 363–372. https://doi.org/10.1145/2493432.2493486

[10] Marion Koelle, Madalina Nicolae, Aditya Shekhar Nittala, Marc Teyssier, and Jürgen Steimle. 2022. Prototyping Soft Devices with Interactive Bioplastics. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (UIST '22). https://doi.org/10.1145/3526113.3545623

[11] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2: 1–24. https://doi.org/10.1145/3463523

[12] Mackenzie Leake, Kathryn Jin, Abe Davis, and Stefanie Mueller. 2023. InStitches: Augmenting Sewing Patterns with Personalized Material-Efficient Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3544548.3581499

[13] Richard Lin, Rohit Ramesh, Antonio Iannopollo, Alberto Sangiovanni Vincentelli, Prabal Dutta, Elad Alon, and Björn Hartmann. 2019. Beyond Schematic Capture: Meaningful Abstractions for Better Electronics Design Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300513

[14] Jasmine Lu, Beza Desta, K. D. Wu, Romain Nith, Joyce E Passananti, and Pedro Lopes. 2023. EcoEDA: Recycling E-Waste During Electronics Design. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (UIST '23). https://doi.org/10.1145/3586183.3606745

[15] Ilan Mandel and Wendy Ju. 2023. Recapturing Product as Material Supply: Hoverboards as Garbatrage. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (DIS '23), 564–579. https://doi.org/10.1145/3563657.3596128

[16] Ilan Mandel and Wendy Ju. 2024. Designing with What Remains. In *Designing Interactive Systems Conference*, 3002–3015. https://doi.org/10.1145/3643834.3661628

[17] Leo McElroy, Quentin Bolsée, Nadya Peek, and Neil Gershenfeld. 2022. SVG-PCB: a web-based bidirectional electronics board editor. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, 1–9. https://doi.org/10.1145/3559400.3562004

[18] David A. Mellis, Leah Buechley, Mitchel Resnick, and Björn Hartmann. 2016. Engaging Amateurs in the Design, Fabrication, and Assembly of Electronic Devices. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (DIS '16), 1270–1281. https://doi.org/10.1145/2901790.2901833

[19] Martin Nisser, Christina Chen Liao, Yuchen Chai, Aradhana Adhikari, Steve Hodges, and Stefanie Mueller. 2021. LaserFactory: A Laser Cutter-based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3411764.3445692

[20] Evan Strasnick, Sean Follmer, and Maneesh Agrawala. 2019. Pinpoint: A PCB Debugging Pipeline Using Interruptible Routing and Instrumentation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–11. https://doi.org/10.1145/3290605.3300278

[21] Daniel Tobjörk and Ronald Österbacka. 2011. Paper Electronics. *Advanced Materials* 23, 17: 1935–1961. https://doi.org/10.1002/adma.201004692

[22] Nobuyuki Umetani and Ryan Schmidt. 2017. SurfCuit: Surface-Mounted Circuits on 3D Prints. *IEEE Computer Graphics and Applications* 37, 3: 52–60. https://doi.org/10.1109/MCG.2017.40

[23] Eldy S. Lazaro Vasquez and Katia Vega. 2019. Myco-accessories: sustainable wearables with biodegradable materials. In *Proceedings of the 23rd International Symposium on Wearable Computers* (ISWC '19), 306–311. https://doi.org/10.1145/3341163.3346938

[24] Sahira Vasquez, Mattia Petrelli, Martina Costa Angeli, Julio Costa, Enrico Avancini, Giuseppe Cantarella, Niko Münzenrieder, Paolo Lugli, and Luisa Petti. 2021. Cost-effective, mask-less, and high-throughput prototyping of flexible hybrid electronic devices using dispense printing and conductive silver ink. In *2021 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*, 1–3. https://doi.org/10.1109/EDTM50988.2021.9420858

[25] Zeyu Yan, Tingyu Cheng, Jasmine Lu, Pedro Lopes, and Huaishu Peng. 2023. Future Paradigms for Sustainable Making. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–3. https://doi.org/10.1145/3586182.3617433

[26] Zeyu Yan, Jiasheng Li, Zining Zhang, and Huaishu Peng. 2024. SolderlessPCB: Reusing Electronic Components in PCB Prototyping through Detachable 3D Printed Housings. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–17. https://doi.org/10.1145/3613904.3642765

[27] Zeyu Yan, Advait Vartak, Jiasheng Li, Zining Zhang, and Huaishu Peng. PCB Renewal: Iterative Reuse of PCB Substrates for SustainableElectronic Making. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3706598.3714276

[28] Zeyu Yan, Mrunal Dhagude, and Huaishu Peng. Make Making Sustainable: Exploring Sustainability Practices, Challenges, and Opportunities in Making Activities. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3706598.3713665

[29] Zhihan Zhang, Felix Hähnlein, Yuxuan Mei, Zachary Englhardt, Shwetak Patel, Adriana Schulz, and Vikram Iyer. 2024. DeltaLCA: Comparative Life-Cycle Assessment for Electronics Design. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 1: 1–29. https://doi.org/10.1145/3643561

[30] Junyi Zhu, Yunyi Zhu, Jiaming Cui, Leon Cheng, Jackson Snowden, Mark Chounlakone, Michael Wessely, and Stefanie Mueller. 2020. MorphSensor: A 3D Electronic Design Tool for Reforming Sensor Modules. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 541–553. https://doi.org/10.1145/3379337.3415898

[31] SparkFun TRRS 3.5mm Jack Breakout - BOB-11570 - SparkFun Electronics. Retrieved September 13, 2024 from https://www.sparkfun.com/products/11570

[32] OSHWA Certified Projects List. Retrieved December 15, 2023 from https://certification.oshwa.org/list.html