Make:

Wearable Electronics

2nd Edition

Design, Prototype, and Wear Your Own Interactive Garments

KATE HARTMAN

CONTENTS:

CONTENTS:

CONTENTS:

Chapter 7: Sensors 154

Chapter 8: Actuators 198

CONTENTS:

Chapter 9: Wearability 240

Preface

Our bodies are our primary interface with the world. Interactive systems that live on the body can be intimate, upfront, and sometimes quite literally in your face. They sit close to your skin, inhabit your clothing, and sometimes even start to feel like part of you. This makes wearable electronics an exciting, challenging, and inspiring area to explore.

On one level, this book is about how to make wearable electronics. It will introduce you to the tools, materials, and techniques necessary to create interactive electronic circuits and embed them in clothing and other things that can be worn. It will take you on a journey that starts with circuit basics and ends with the more sophisticated details of how to make electronics wearable. In between, you'll learn about materials, microcontrollers, sensors, and actuators and how these things fit into the world of wearable electronics. The tools and techniques that are covered can also be applied to textiles, tapestries, toys, and more!

Maxwell Lander

Kate Hartman (far left) with Social Body Lab collaborators Izzie Colpitts-Campbell, Alexis Knipping, and Boris Kourtokov wearing Monarch V2, a muscle-activated kinetic wearable.

On another level, this book is asking you, *"What's next?"*

It invites you to join the conversation about the future of wearable and body-centric technologies. What do we need? What do we want? And what should be avoided?

We're living in a moment where wearable technologies have become a part of our everyday lives. They live on our wrists and heads, tracking our activities and transporting us into virtual worlds and augmented experiences. In the last two decades, stand-alone consumer wearable electronic devices have gone from a novelty to a norm. In the years ahead, we can anticipate technological advancements that more seamlessly integrate into our clothes.

The more permissive we become in allowing technology into our personal spaces, the more carefully we must consider how it is designed and used: What data is collected, and who can access it? How, where, and when are we comfortable with lights, sounds, and sensations being activated in our clothes? What happens to wearable electronic devices when they break or become outdated? What type of wearables can help us, and which might create indirect (or direct) harm?

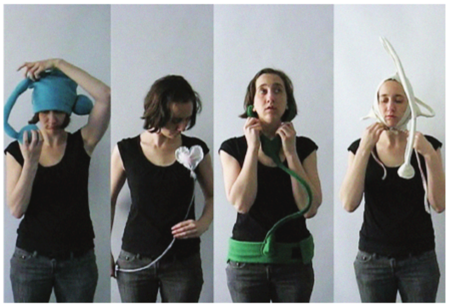
Now is a good time to ask questions, challenge assumptions, and create and hold space for diverse perspectives. Making is a way to creatively and critically engage with technology. This book will, hopefully, help you get started with cocreating the future of wearables.

ABOUT ME

As an artist, I make work to express what I can't in words. I like to try to capture an idea in a thing so I can hand the thing to someone else and see what they make of it.

I started as an image maker—capturing snippets of my experiences and surroundings through analog photography, 16 mm film, and digital video. After a while, the frames containing each medium began to feel claustrophobic. I wanted to extend beyond the frame and immerse others in the work. To explore this, I played with embedding moving images in installation environments and augmenting them with aspects of performance. But it wasn't enough—I wanted the media to respond to the bodies in the space.

So I went to grad school and learned about electronics and programming. I learned how sensor data interpreted by a computer program could allow screen-based media to respond to happenings in the physical world. Along the way, I lost interest in the screen and centered my focus on physical interactivity. Rather than creating interactive spaces that people could pass through, I wanted to create experiences they could inhabit in a more close, personal, and intimate way And that's how I found myself in the world of wearable electronics.



Early imagined wearable projects by Kate Hartman: Muttering Hat, Inflatable Heart, Gut Listener, and Talk to Yourself Hat

The Porcupine Vest, by Kate Hartman, is a wearable constructed from everyday materials that enables the wearer to alter and play with their spatial footprint. It is made with laser-cut cardboard, brass fasteners and washers, nylon webbing, and tri-glide fasteners. In a workshop setting, participants are supplied materials and instructions and left to craft their own creations, which is often when many innovations and deviations occur.

ABOUT YOU

This book is for people who want to roll up their sleeves and get hands-on with wearable electronics. This includes students, researchers, hackers, makers, fashion designers, engineers, industrial designers, developers, costume enthusiasts, artists, and textile mavens. Whatever your discipline is, this book invites you to play, challenge, and create with, in, and about wearable electronics.

There are (at least) two perspectives from which you might approach this book:

- You know some stuff. There's a broad range to this. Maybe, once upon a time, you used an Arduino to blink an LED at a workshop. Or maybe you run a design firm that produces massively robust interactive installations in museums, and now you've got a client who wants you to generate a prototype that's wearable. Either way, you know enough to have a sense of what universe you're in. This book will help you build on what you already know and might even lead you into some areas you didn't expect.

- When it comes to electronics and programming, you're a bit of a n00b. Maybe you're a fashion designer who realizes that interactivity in clothing is something you would like to wrap your head around. Or perhaps you're a sociologist developing a data-collection system that includes sensors that live on the body. Or maybe you're an artist with a newfound interest in self-tracking. In any case, there are likely many things in this book you may not have heard of before. If you're in this category, take this advice: Be brave. It's OK if things are new or you don't understand something on the first go. This book might be your gateway to new things you didn't even realize you wanted to learn. Stick with it—it's interesting stuff!

WHY WEAR IT

When embarking on a wearable-electronics project, the first and most important question you should ask yourself is this:

"Why does this need to be wearable?"

There are many possible answers to this question, but it's important that you have at least one in mind. Here are a few:

- You are sensing the body in such a way that the sensor needs to be placed on the body.
- You have a display or feedback mechanism that needs to stay with a person at all times.
- Your project needs to travel with the user and not stay in one place.
- You want to create a particularly intimate or immersive experience for the wearer.
- Your project is specifically clothing oriented, such as a costume or fashion piece.
- You're interested in the future of wearable electronics and want to use making as a way to think about what's next.

Pinky Linkers, by Kate Hartman, are parametrically-designed devices that allow two people who might not know each other very well to experience a physical bond. They are customized to reflect both the size of each individual's pinky as well as how close each person feels to the other.

- Once you have a reason in mind, you can use it to guide the decisions that follow as you design your project.

TECHNICAL NOTES

This book deliberately avoids replicating existing learning resources. Take note of the references woven into each chapter and the resources provided in appendix A. These breadcrumbs will lead you to tutorials, reference guides, and examples created by smart and talented thinkers, makers, and visionaries.

Circuit diagrams are created using an excellent open-source software called Fritzing. Fritzing's Breadboard layer can create diagrams to reflect how connections are made in the physical circuit, which is helpful for both planning and documenting. I strongly recommend that you donate to the project and use it as a core tool in your wearable-electronics practice.

Code shared in this book can be found here and will be updated as needed: github.com/katehartman/Make-Wearable-Electronics-2E

For battery power, the analog circuits can be powered using CR2032 batteries. Except where noted, the microcontroller circuits can be powered either by a ~1,000 mAh rechargeable lithium polymer (LiPo) battery or a 3x AA or AAA battery pack, or via the microcontroller's USB connection. Battery power is discussed further in appendix B.

Part numbers are not included in this second edition. Electronic parts and materials change quickly! They are increasingly being retired, reworked, and reintroduced as different but similar products. Additionally, suppliers can differ greatly across the globe. This book aims to equip you with the knowledge you need to research and select the best and most readily available parts.

Lastly, some books show you exactly how to build a particular project. Make: Wearable Electronics is not one of them. Instead, this book provides the building blocks that will enable you to bring your own ideas to life. *How-tos* are small sections of specific step-by-step instructions for a particular process or technique. *Experiments* are open-ended prompts sprinkled throughout the book that invite you to play with the concepts, tools, and techniques introduced. *Galleries* are included at the end of each chapter to demonstrate each chapter's topics as applied to real-world projects by artists, designers, makers, and researchers—an excellent source of inspiration!

#122 Engagement Display is a speculative, head-based wearable that visually indicates attention and brain stimulation. The more engagement the device perceives from the user, the more the display mechanisms wriggle and grow. This prototype is part of the Inquisitive Devices project by Social Body Lab (prototype by Valerie Carew).

EXPERIMENT: Imagined Wearable

An experiment in the preface? That's right! The best time for you to start prototyping wearable electronics is right now. Sometimes, it's easier to work through ideas before you even know which technologies you might use to create them.

Imagine something intended to be worn on your body (a garment or accessory) to help you better relate to the world around you. It could be something practical, possible, or desirable. Or it could be something ridiculous, outlandish, annoying, or invasive. The technology your garment utilizes does not have to actually exist and can be one of your own invention.

Once you've imagined your wearable, create a physical, wearable prototype or mock-up that demonstrates what it might look like and how it would work. To make it, you can modify something that already exists (T-shirt, sneakers, top hat, etc.) or create something new from raw materials. It doesn't have to be fancy. Sometimes paper, duct tape, and Sharpies will do just fine.

This is a conceptual prototype—you do not need to implement any technology. Instead, focus on the design of the piece as well as the story behind it. Feel free to be creative, playful, and inventive. Try creating supporting materials such as instructions for use or user scenarios to help develop the story of your wearable. Once you're done, share it with a colleague, classmate, or friend. It's a great way to start a conversation about what they think about the world of wearable electronics!

1

Studio

SURFACE MOUNT TOOLS
HELPING HANDS
BREADBOARDS
PROTOBOARD
GLUE GUN + STICKS
BATTERY GRAVEYARD
PLIERS
SMALL SNIPS
SOLDERING IRON SUPPLIES
DESOLDERING TOOLS
SOLDER
SAFETY GOGGLES

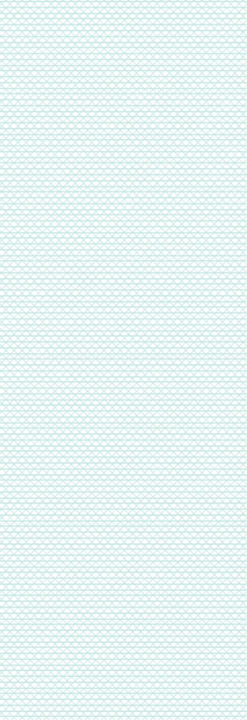
Storage bins for electronics tools and sewing notions at Social Body Lab at OCAD University

Welcome to the world of wearable electronics! Creating a wearable-electronics project is an inherently interdisciplinary activity. When creating a wearable, you might engage in ways of making associated with fashion, textiles, costumes, jewelry, watches, glasses, sportswear, or medical wear. Working with electronics typically involves electricity, circuits, components, and, often, coding. How does one set oneself up for working across such diverse disciplines?

To create a shared home for these practices, we will begin with the idea of creating a **wearable-electronics studio**. We'll use the word *studio* as a placeholder for any space or spaces you might be working in. This could also be a lab, a research space, a makerspace, a hackerspace, a garage, a craft corner, a kitchen table, a picnic table, or somewhere else!

I started making wearables when I was in graduate school, in a work space space shared by 200 students. All space was shared, and we had small, square lockers in which to store our work. Sometimes, I also worked on my wearables projects at home in my bedroom in a shared apartment 145 blocks north of where I was going to school. At that point, it was essential to have small and portable materials and tools. Since then, I've moved on from being a student to becoming, in turn, a researcher, an adjunct instructor, and a professor with a research lab.

Adobe Stock-Lepusinensis



When I joined OCAD University in Toronto, I had the opportunity to establish a wearable-technology research lab, now known as Social Body Lab (SBL). This dedicated space I share with colleagues, collaborators, and student research assistants has gone through numerous iterations, messes, and reorganizations. It has even moved from a small room to a larger one.

But even though I have access to this amazing space, I still maintain a mobile studio as well. During the week, I often teach in multiple buildings, so I need a kit I can work with on the go. Also, I work both at OCAD and at ITP/IMA (the Interactive Telecommunications Program/Interactive Media Arts) at New York University's Tisch School of the Arts, so I have a set of tools that are easy to travel with in a carry-on suitcase. In addition, over the years, I've had the opportunity to teach wearable-electronics workshops in far-flung cities like Buenos Aires, Copenhagen, Shanghai, and Zaragosa (in Spain), so I have tools that work with international plugs and voltages. Through these experiences, I've come to value and prioritize the idea of creating a wearable-electronics studio that is versatile and able to expand and contract according to project and place.

As you are getting started, I suggest you create your wearable-electronics studio with minimal resources and build up from there. Keep it simple.

Use tools and materials you already own—there is no need to run out and buy lots of new things. Borrow where needed from friends, neighbors, family members. Use community resources, perhaps available through makerspaces or hackerspaces, a library, or a school. Depending on where you live, there may be secondhand shops, surplus stores, markets, or corner stores where you can get the necessary materials and tools.

This chapter includes simple guidelines for developing a studio that includes some materials, tools, and methods applicable for wearable-electronics projects. But before we jump in, let's consider the relationship between these terms:

- ***Materials*** are something from which other things can be made. For our purposes, the term can refer to a raw material such as copper or sheep's wool or a manufactured item such as a component or a textile.
- ***Tools*** are devices that allow you to manipulate materials.
- ***Methods*** are ways of making that often pertain to how tools and materials are used together.

Maxwell Lander

Kraft paper, printer paper, gaff tape, paper tape, and painter's tape

With these terms in mind, let's explore what's involved in making wearables, making electronics, and making space for the mingling of these disciplines.

Making Wearables

PAPER & TAPE

Traditional garment design utilizes paper for patternmaking. We can also use paper to prototype wearables. Printer paper is readily available and versatile. When prototyping, I often use the blank back side of to-do lists, bills, and flyers. Paper bags can be cut open for larger dimensions of paper. Rolls of butcher block or kraft paper also come in handy. Tracing paper can be of use when a design needs to be replicated.

Low-tack paper tapes such as painter's tape or *washi* tape work well for connecting paper to itself or for lightly attaching something. Masking tape can be used, but it is more difficult to reposition. Medical tape is perfect for temporarily and safely attaching things to the body—we use it in my Haptics class as a way to quickly explore how vibrating motors feel on different body parts. Gaff tape has a stronger adhesive but can be removed without leaving gunk behind. I generally avoid both electrical tape and duct tape, however, as they are difficult to remove, and they leave a sticky residue.

THREAD & FABRIC

Thread and fabric are key materials used when creating clothing. It is helpful to keep a selection on hand for soft prototyping. If you are comfortable with sewing or clothing design, you likely have certain threads and fabrics you prefer. But if you're new to these areas, here are a few ideas to get you going:

Threads

Fabrics

For thread, simple cotton thread is sufficient to start. I tend to keep black, white, and an accent color in my thread collection. For fabric, basic muslin works just fine, but having a few neat colors and textures in the mix can help jog the imagination and inspire new designs.

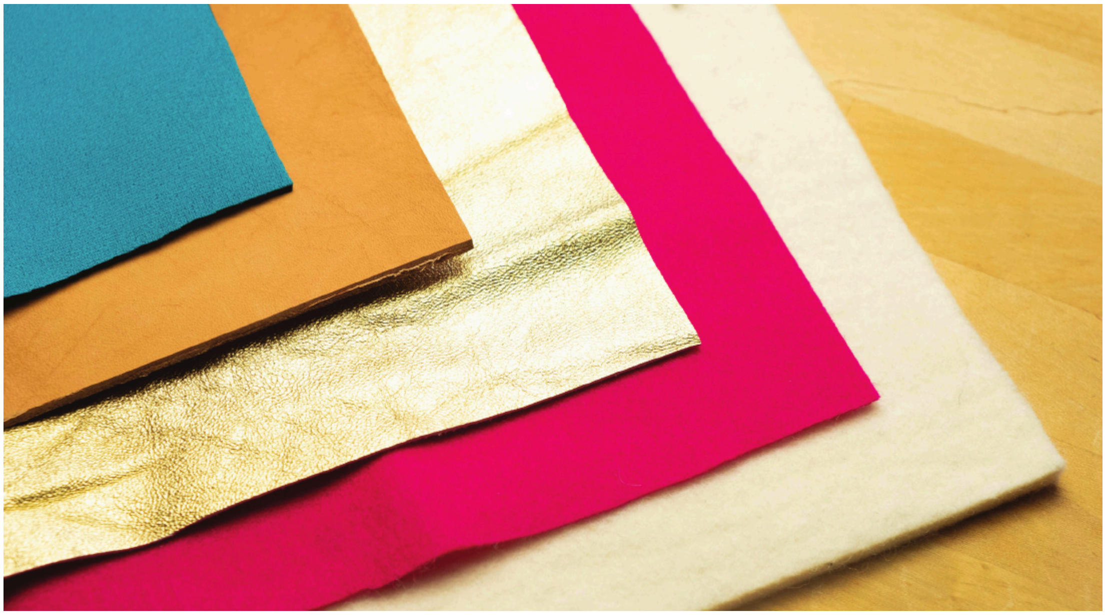
At SBL, we lean toward simple, no-fray materials that are quick to prototype with because they do not require hemming:

- ***Neoprene*** (sometimes known as scuba fabric) is the squishy material used to make wet suits and laptop sleeves. It comes in a variety of thicknesses, is flexible across a wide range of temperatures, and works well as a nonconductive material when creating DIY switches and sensors. Be sure to look for the kind that has a fabric lining on both sides.
- ***Felt*** is used a lot in crafts and works well for prototyping basic wearable accessories. Acrylic felt is very affordable, but wool felt is an excellent upgrade. Both can be cut with a craft cutter or a laser cutter.
- ***Industrial felt*** has become a go-to material at SBL. We often use it in place of neoprene when we want to work with a natural material. It can be slightly harder to find, but it comes in a variety of thicknesses and densities.

MEASURING & CUTTING

Putting care into how you measure and cut materials is a way of treating them with respect. Thoughtfulness in cutting reduces material waste, which is beneficial for both cost-cutting (ha-ha) and environmental considerations.

Tape measures are used to take measurements of the body. Rulers are used to measure the length and width of a material. Calipers can be used to measure thickness.



Nonfraying materials (left to right): neoprene, vegetable tan leather, gold leather, felt, industrial felt

Maxwell Lander

Fabric scissors are a must, particularly when working with conductive fabric. Dedicate a suitable pair of sharp scissors exclusively to fabric—they will stay sharper and last longer. Also, keep a pair of paper or general-use scissors on hand to help you avoid the temptation to use fabric scissors on nonfabric materials. If you'd like to expand your scissors collection, you'll find that detail scissors or thread snips are helpful for cutting and trimming thread. Pinking shears, meanwhile, can be used to cut woven fabrics with a zigzag edge to reduce fraying.

Sometimes it's useful to cut textiles on a surface. A cutting mat, a clear ruler, and a rotary cutter allow for efficient, clean cutting of larger pieces of fabric. This is a popular method for quilters. I often use it when cutting fabric into smaller squares as I prepare kits for a workshop or class.

If you need to cut a complex design, a computer-controlled cutting machine can be a great asset. Craft cutters such as those made by Silhouette, Cricut, and Siser can be set up to cut fabric based on a digital design. Laser cutters can make precise cuts as well—just keep in mind that the edges of the material will be lightly singed or burnt.

Maxwell Lander

Cutting and measuring tools

PINNING

Sewing pins can be used to temporarily hold fabric or even components in place while you are sewing your circuit together. Dressmaker pins or flat-headed pins are made entirely of metal. Round-headed pins have a plastic or glass ball at the end. (The advantage of glass heads is that they won't melt if ironed over.) Many other types of pins are specific to types of fabric or sewing applications. Safety pins can be used for more secure and more comfortable temporary connections and closures.

IRONING & GLUING

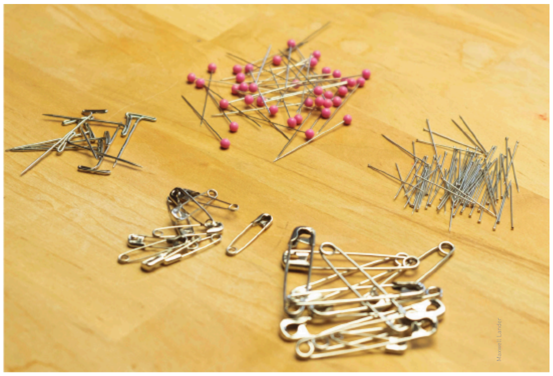
Ironed fabric is much easier to manipulate, cut, and sew. Irons are also useful to melt iron-on adhesives meant to be used to bond fabric.

Many types of irons are available. A basic household iron works well for most purposes. A craft iron can be useful for when you need to iron a small, confined area. However, a travel iron has become my favorite iron in my collection. It's appropriately sized for most of the circuits I work with, and it is dual voltage and packable so I can use it when traveling to other countries. Craft heat presses are also quite nice for securing an iron-on design, but they are a bit more of an investment.

An ironing board, often readily available in domestic settings, is a standard surface to use when ironing. Full ironing boards stand on the floor and are adjustable in height. Mini ironing boards sit on top of a table. However, an ironing mat is slightly more versatile and portable, and it pairs well with a travel iron.

When ironing, always remember that the iron should be placed on its end when not in use. Leaving the iron face down can damage your materials and the ironing surface and may cause smoke and flames. Also, be sure to turn off your iron when walking away, and when packing up, give your iron time to cool down before handling it.

HeatnBond UltraHold is my preferred iron-on adhesive when I want to bind two pieces of fabric together. However, many brands of fabric adhesives are available at fabric and craft stores. Be sure to look at the drying time when purchasing, however. My favorite quick-glue


Maxwell Lander

Types of pins: Round-head pins, straight pins, T pins, and different sizes of safety pins

technique is to use a high-temperature glue gun with fabric glue sticks. The glue dries quickly and makes a secure connection. Just be mindful to let the glue cool before you start handling it.

SEWING

Hand sewing is a common starting place when making soft circuits. Choosing a needle for sewing with conductive thread can be tricky. Be sure to test your needle with all components in your circuit before you begin sewing. It should have an eye that is large enough for your conductive thread to pass through but also small enough to pass through the *sew holes* of your components. For sewing with conductive thread, get sharps with larger eyes, size 7 or similar. Needles meant for embroidery usually work well.

Ironing & gluing tools: Ironing mat, travel iron, roll of HeatnBond, fabric glue, E6000 adhesive, a high-temperature glue gun, and fabric glue sticks

Here are some other accessories that are helpful for hand sewing:

- A needle threader can save time and frustration when working with gnarly conductive thread.
- An embroidery hoop can hold fabric in place when hand sewing. This can make sewing conductive thread circuits a lot easier.
- A seam ripper is the sewing equivalent of the Undo command on your computer. Use this nifty device to remove stitches with ease!

A sewing machine is also an excellent addition to the wearable-electronics studio. Some people love the detailed skill and meditative aspects of hand sewing. I am not one of those people. Even with just a simple straight stitch, I find machine sewing to be a fun and zippy way to stitch together wearables in a flash.

Making Electronics

WORKING WITH WIRE

When working with wire, it's important to have the proper tools for stripping, trimming, and bending it.

Wire strippers are used to remove the insulation at the end of a piece of wire in order to expose the conductive part. Strippers with a 20–30 American Wire Gauge (AWG) range work well for wearable electronics projects. Flat-nose or needle-nose pliers can help with bending, manipulating, and smooshing of wires and the legs of through-hole components. Small snips or flush cutters are useful for cutting and trimming wire or leads of through-hole components, particularly in tight spaces such as close to the surface of a circuit board.

Maxwell Lander

Sewing tools: Seam ripper, embroidery hoop, embroidery needles, needle threader, and assorted needles (left); sewing machine (right)

SOLDERING

Unless you are building circuits that are entirely soft, soldering is an important skill to develop. Soldering irons range from supercheap to quite fancy. Spring for something a little nicer than the cheapest iron you might find online or at a hardware store. At the very least, be sure to get something that has temperature control. A soldering base is used to hold the iron when not in use. It also has a tray for a damp sponge for lightly cleaning the tip of the iron. If a more vigorous cleaning is needed, a brass sponge can be used.

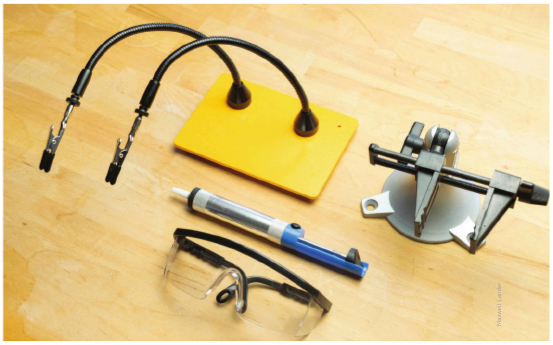
Safety glasses are recommended when soldering or snipping wires. They will protect your eyes if a piece of material flies off in an unexpected direction.

Circuit assembly and soldering often take more than two hands. A helping-hands tool offers a way to hold components in place as you handle the soldering iron and solder. Traditional helping hands usually include a magnifying glass for precise work. Another style of helping hands has flexible arms that attach via a magnet to a metal base. These can be a bit easier to adjust. The PanaVise Jr. is a vise specifically designed for use with printed circuit boards (PCBs). These have adjustable arms with a shallow tray that can accommodate circuit boards of various sizes.

When soldering, sometimes you make mistakes. A solder sucker is a manually activated pump that can suck up solder when it's hot. A solder wick or desoldering braid is finely woven solder that can be used to draw away unwanted solder when needed. Desoldering is an option but is often a bit tedious, so take your time with soldering and try to get it right on the first try!

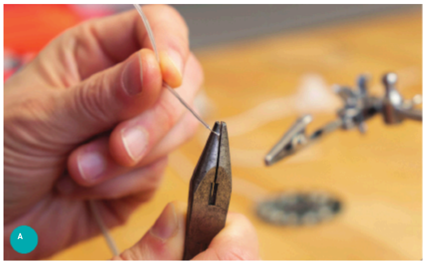
Tools for working with wire: needle-nose pliers, wire strippers, small snips

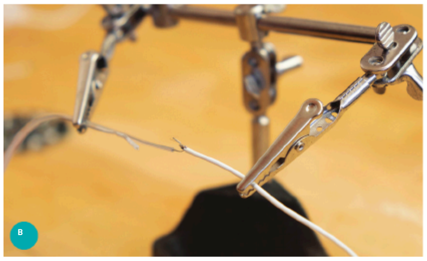
Tools for soldering: solder, fume extractor, helping hands, brass sponge, soldering-iron station with dampened sponge

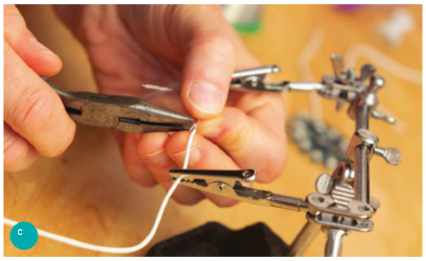


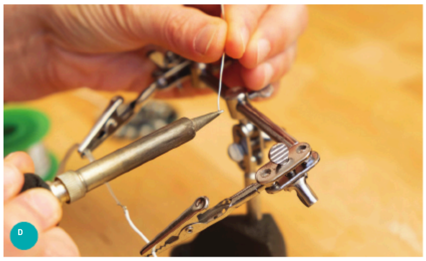
Maxwell Lander

Additional soldering tools: flexible helping hands, circuit board holder, desoldering pump, safety glasses









HOW TO: Soldering Two Wires

While this book won't provide an in-depth look into soldering, here's a small demonstration to get you started. See appendix A for more detailed guides to soldering.

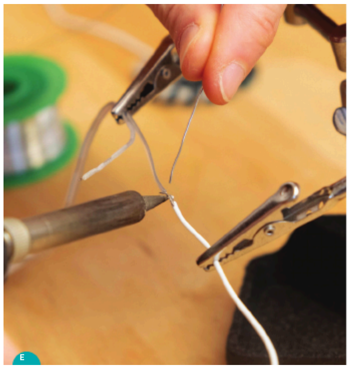
Steps:

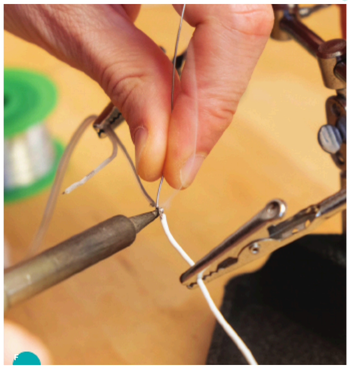
1. Use pliers to create hooks at the stripped ends of two pieces of wire (Figure A).
2. Use the hooks to pair the wires and secure with helping hands (Figure B).
3. Smoosh the hooks with the pliers (Figure C).
4. Tin the tip of the soldering iron (Figure D).
5. Heat the joint before applying the solder (Figure E).
6. Apply the solder to the joint (Figure F).
7. Remove the solder, then remove the tip of the soldering iron (Figure G). Let the joint cool. The connection has been soldered!

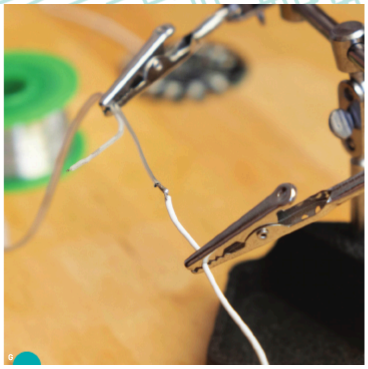
INSULATING

Heat-shrink tubing, which contracts when heat is applied, can be used to insulate exposed electrical connections. It comes in a variety of widths and can be bought on a spool or in a box with a variety of precut lengths. I often prefer to work with clear heat-shrink so I can see what's happening inside.

A heat gun is used to activate heat-shrink tubing. This high-temperature tool runs much hotter than a hairdryer and, when pointed at heat-shrink tubing, will cause it to shrink. Always take care not to point it at yourself, others, or anything







that is heat-sensitive. Working with helping hands on top of a silicone mat is a good way to protect yourself and your work surface.

TESTING

A multimeter is an essential electronics tool for measuring and testing continuity, voltage, current, and resistance. A typical multimeter has a knob that needs to be turned to select what you are measuring as well as in what range. Pocket multimeters are great for when you're on the go. Auto-ranging multimeters are more expensive but a little easier to work with. Different variations of multimeter probes are available. I find probes with alligator clips at the end to be very useful when creating wearable electronics.

Making Space

Developing a studio for your wearable-electronics practice does not always center on a physical space. The space(s) where you create your wearable-electronics projects

Heat gun, helping hands, various sizes of heat-shrink tubing

A multimeter with alligator clip probes

might be vast or cozy, permanent or nomadic. If temporary, they may require your wearable-electronics gear to be repeatedly unpacked and repacked.

Whether you are working in an art studio, a research lab, a hackerspace, a fabrication shop, a classroom, a garage, a bedroom, or a kitchen, develop systems and spaces to support your prototyping work best. Your studio can take the form of an organization system for components and materials, a habit of laying out tools in a particular way, or a dedicated notebook that helps you return to a specific ideation space. However early in or far along you are with your work in wearable electronics, this can help you develop a respect for and commitment to your wearable-electronics practice.

ORGANIZING

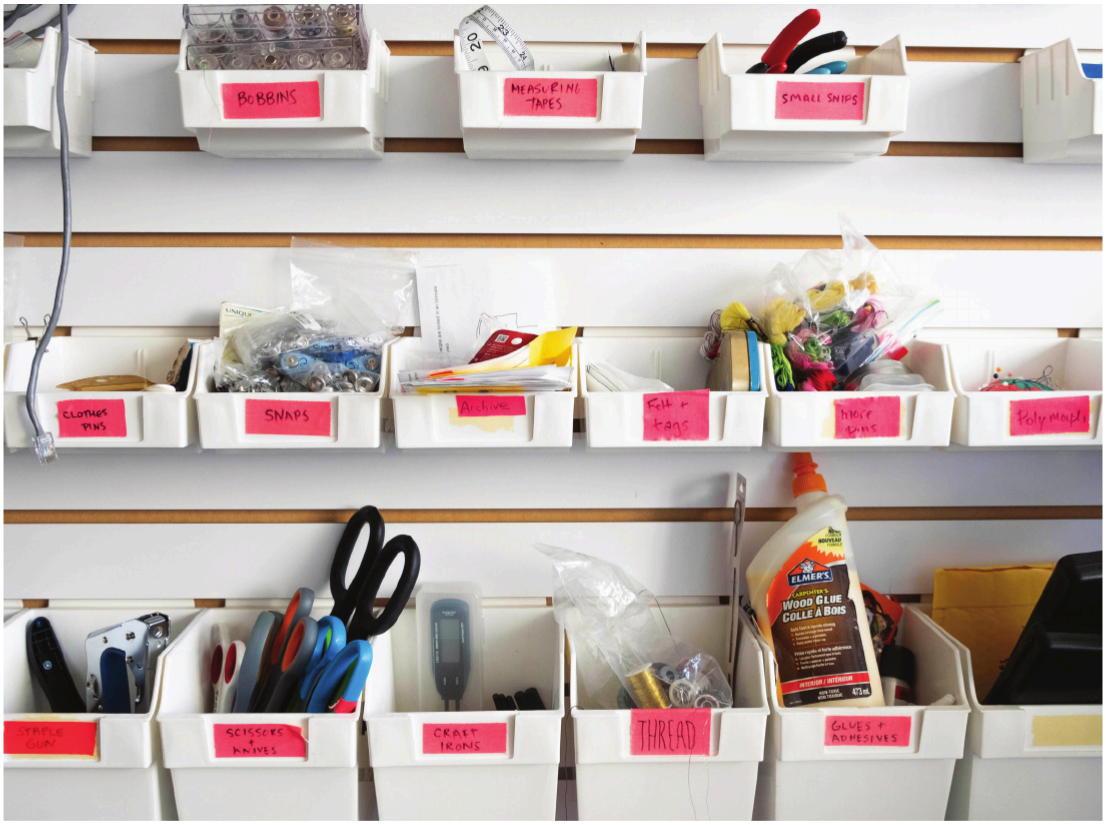
Consider how you might organize the materials and tools associated with making wearables and making electronics:

- Do all your tools and materials fit into a box or boxes?
- How can you keep these items well organized?
- Are there systems or organization tools you could use to make them more readily accessible?

- How can you design elements of your studio to support portability, modularity, and compactness?

Compartment boxes, available at your local hardware or craft store, are a great way to keep parts organized, but for some reason, they don't work well with how I think. I like to be able to rearrange things on the fly. At SBL, we store our work in two sizes of large, open storage bins: active projects on lower shelves, and completed or inactive projects on higher shelves.

Tools are stored in smaller open bins on slatwall, which can be easily rearranged. Very small items such as circuit boards and other components are stored in photo boxes, which can be found at craft stores. I love them because often you can find larger boxes designed to hold photo boxes. This means that everything can be contained in a neat package but can also be rearranged, and you can grab a box or two to take with you on the go.



Open, wall-mounted bins are great for storing tools and materials you need to access regularly.

Maxwell Lander

Neon-pink gaff tape and a black marker are used to label pretty much everything in the lab (and, increasingly, in my home!). For me, using consistent sizes, colors, and labeling of storage containers makes it easier to organize and find things.

Storing components in photo boxes: Closed containers are great for longer-term storage or for packing parts that need to travel with you.

DOCUMENTING

Documenting your work is important, whether it be for class, research, or your own portfolio. It allows you to track your learning, record your decisions, and share what you've made with others. Including documentation tools in your studio and building habits around them allows you to capture your process as you go.

Take a moment to consider: What are your preferred tools for documentation? Depending on the quality of images you seek, your camera can be as simple as a smartphone or as complex as a fancy digital single-lens reflex (DSLR) or mirrorless camera. Be sure to work with a camera that can also take videos so you can capture your light-emitting diodes (LEDs) blinking and motors spinning. Document cameras can also be useful if you are trying to record a technique or demonstration.

Maxwell Lander

A phone stand can be used for project documentation.

A stable base can improve the quality of your images tremendously. Tripods, copy stands, clippable gooseneck phone holders, or even a stack of books can help keep your camera steady.

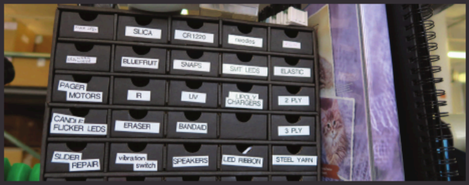
MAKING IT YOURS

Take some time to reflect on how you already work, and consider how that might integrate with your wearable-electronics practice and studio. Remember that creating a wearables studio is an ongoing practice. Your studio will shift and change, depending on where you are in your life, your profession, your studies, your passions, and the world. As you make wearable-electronics projects, keep making and iterating a wearable-electronics studio that works for you!

GALLERY 1:

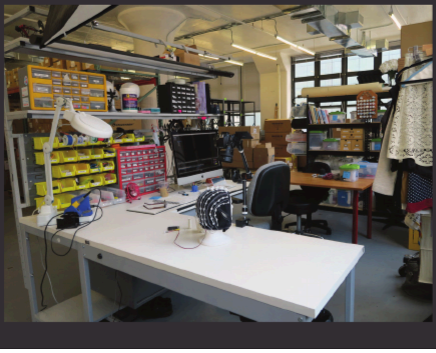
STUDIOS FOR WEARABLES

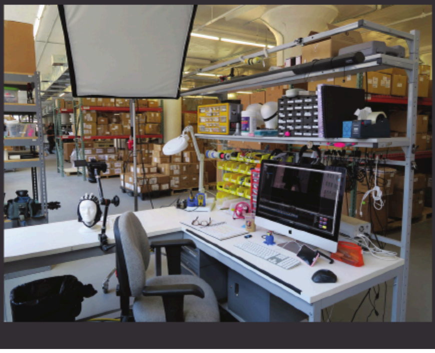
Traditional studios like a wood shop or a painting studio are well-understood spaces. But how do you create a studio that supports working with the hard bits of electronics and softness of textiles? Here are a few inspiring examples of how wearable-electronics colleagues create bespoke studios at home and on the go.

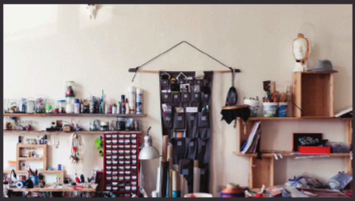


Workspace for Wearables

These photographs show Becky Stern's studio as it appeared while she was director of wearable electronics at Adafruit Industries. The space, designed to support a variety of activities, including sewing and crafting, electronics and soldering, video and photo, and writing and editing, featured clearly labeled bins and drawers that kept supplies within reach, and shelves and a garment rack for storing wearable-electronics prototypes.



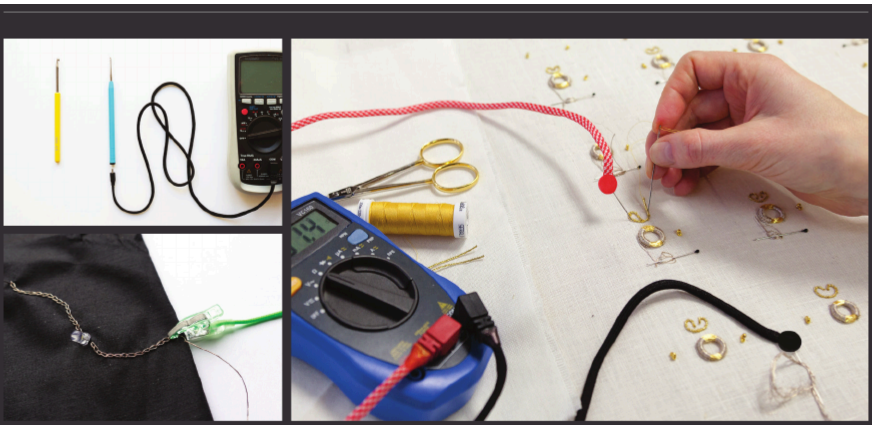




Aimee Shirley for friendsoffriends.com (left & bottom) Irene Posch (top)

KOBA Schneiderei für Elektronische Textilien

KOBA Schneiderei für Elektronische Textilien, by Mika Satomi and Hannah Perner-Wilson (who collaborate as KOBAKANT), was a speculative story realized as a yearlong performance based in a storefront in Kreuzberg, Berlin. (*Koba*, in Japanese, refers to a small family-run business.) KOBA was an electronic-textile tailor shop where anybody could place an order for custom-made wearable-technology garments and accessories. Equipped with physical examples, samples, and catalogs, the shop also served as a platform for discussion, experimentation, and storytelling of alternative realities.

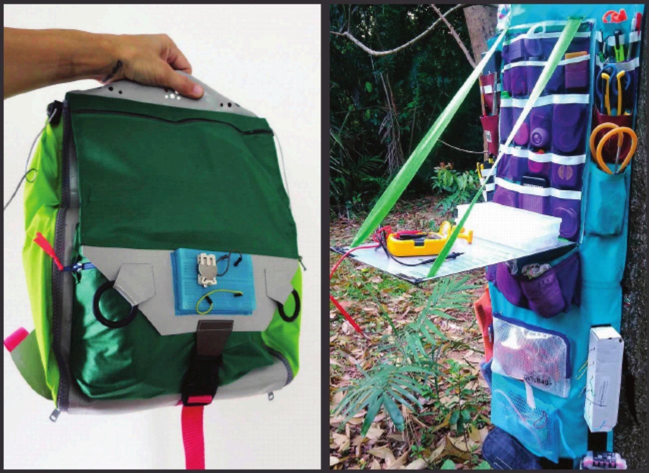


Tools for Electronic-Textile Crafts

Irene Posch adapted needlework tools such as pin probes and clip probes for use in making e-textiles, better enabling a multimeter to be used with conductive threads, yarns, and fabrics. The travel tester, which supports conductivity testing of unexpected materials, is one of the design inspirations for the experiment in chapter 2.

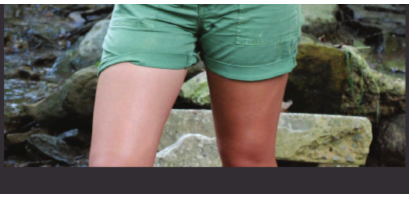
Wearables in the Wild Dinapack

The Wearables in the Wild Dinapack, a transformable backpack-cum-electronics studio created by Hannah Perner-Wilson for Andrew Quitmeyer, is tailored for electronics work while hiking with designers and field biologists around the world.



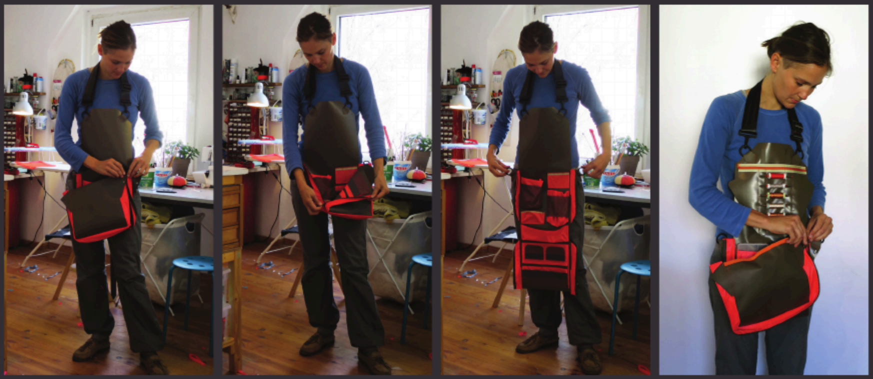
Field Computing: Portable Workspace

The Portable Workspace, designed and fabricated by Jen Liu at the Iowa Lakeside Laboratory during a two-week artist residency, is a wearable device that unfolds into a physical computing studio. The device, inspired by



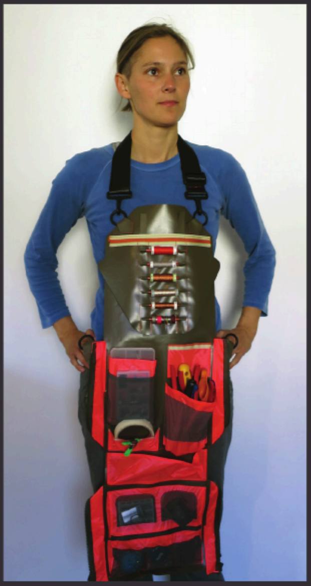
Desi Gonzalez

the Wearable Studio Practice, a collaborative project by Hannah Perner-Wilson and Andrew Quitmeyer, contains tools and a work area for a user to build a project when they are out and about. This project explores changing sites of production for technology as it relates to wearable technology, citizen science, and environmental monitoring.



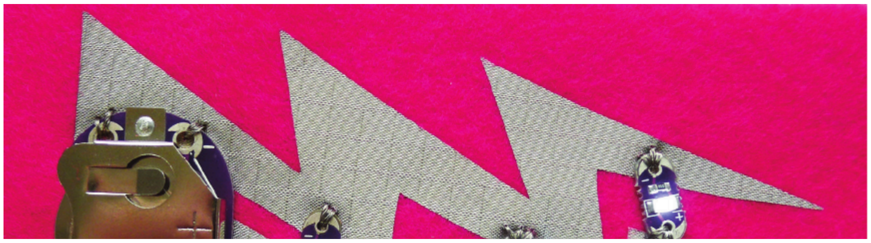
DAYpack

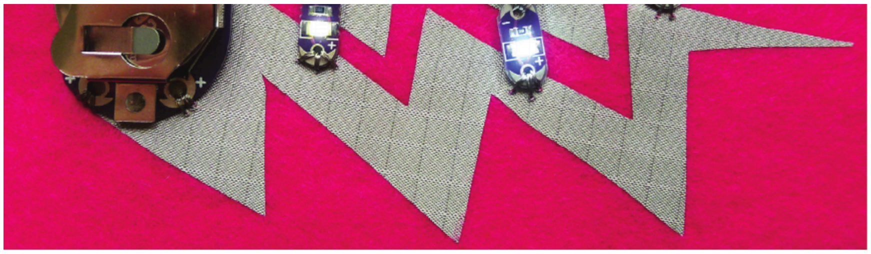
The DAYpack, also designed by Hannah Perner-Wilson, is a shoulder bag, apron, belt, and hanging workstation all in one, which Perner-Wilson made for herself after returning from the Hacking the Wild expedition in Madagascar. It is designed to hold commonly used tools and a selection of materials, as well as food and drink needed for a full day. The DAYpack, which Perner-Wilson used as an everyday bag for roughly a year and half, not only carries supplies but also becomes part of the workspace by offering organization, easy access, and storage.



2

Circuits





Now that you've been introduced to the materials, tools, and methods associated with creating a wearable-electronics studio, we can begin to explore how to make with them. Before diving into designing complex, body-based, interactive projects, however, we will begin by developing an understanding of basic circuits.

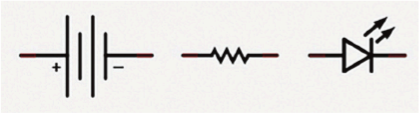
Circuits that are meant to live on the body have a unique set of material needs. When making wearable electronics, you must consider which materials to use in your circuits. Because bodies tend to bend, twist, and shake their booties, the materials used in wearable circuits are subject to a lot of wear and tear. Conductive materials used in wearable circuits need to be durable, flexible, and sometimes even soft.

In this chapter, you will learn about how circuits work as well as how to construct them using a variety of tools and materials.

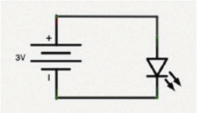
Adobe Stock-Lepusinensis

Circuit Basics

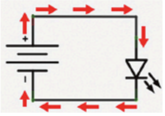
Several key concepts are essential for understanding and constructing circuits. These concepts will help guide circuit design and choice of components:



Circuit symbols for (left to right) a battery, a resistor, and an LED.

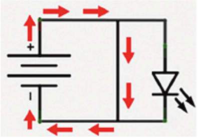


Circuit schematic of a basic circuit with a battery and an LED



The current (indicated by the red arrows) travels from power (+) to ground (–).

When presented with the opportunity, electricity will always follow the path of least resistance; in this circuit, the electricity does not reach the LED, so the LED will not light up.



A ***circuit*** is a closed loop of electricity that contains a power source and a load. Conductors provide pathways for the electricity to travel between components in the loop.

A ***power source*** provides electricity for the circuit. Batteries are a sensible power source for wearable electronics because they are relatively portable and compact. A battery or a battery pack can be used as the power source for all the circuits covered in this book.

A ***load*** is something that makes use of the electricity in the circuit. The examples in this chapter use one or more light-emitting diodes (LEDs) as the load.

A ***conductor*** is a material that permits the flow of electricity. This chapter introduces a variety of conductors with which to create electrical connections in a circuit.

A ***circuit schematic*** uses symbols and lines to visually represent the components and connections in a circuit. It illustrates the electrical connections being made within a circuit but does not represent the physical layout of those connections.

Electricity is thought of as traveling from the point of highest electrical potential (*power*, or +) to the lowest (*ground*, or –). In this circuit, electricity flows from the positive terminal of the battery (marked with +) to the positive terminal of the LED, through the LED, and out the negative terminal of the LED to the negative terminal of the battery (marked with –), thus completing the loop. Along the way, it will travel through the LED and (assuming it meets the power requirements) cause the LED to light.

Electricity follows the path of least resistance. You can think of it as being a bit lazy. If electricity has the option of working to light up an LED or to take a path through a nonresistive material back to the battery, it's going to take the easy road.

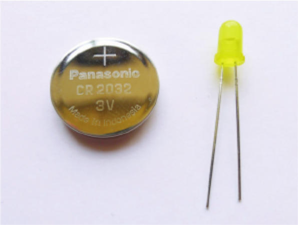
The problem with this alternative path is that it creates a *short circuit*, a closed loop of electricity that has a power source but no load.

If electricity travels from the positive terminal of the battery directly into the negative terminal, it will likely drain the battery. In some situations, the results can be more severe, including smoke, melted wires, and damaged components. At a minimum, the circuit will not function properly. No matter what the circumstance, short circuits are not good and should always be avoided.

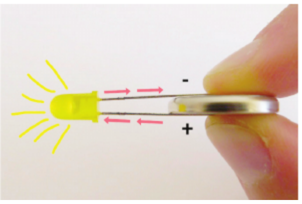
Insulators are materials that do not conduct electricity. They can be used to prevent short circuits.

To quickly build and test this circuit, you can use a 3 V CR2032 coin-cell battery and a through-hole LED.

To create the circuit depicted in the circuit schematic, position the LED so the legs straddle the battery with the positive leg on the positive side of the battery and the negative leg on the negative side of the battery.

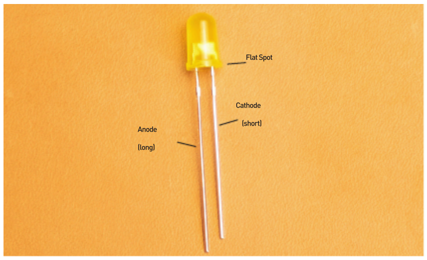


CR2032 3 V battery and 5 mm LED

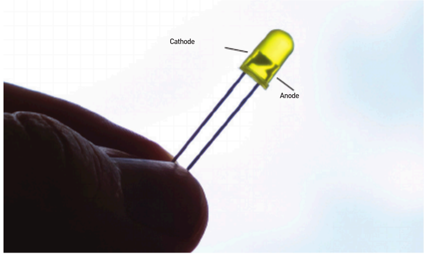


A simple circuit

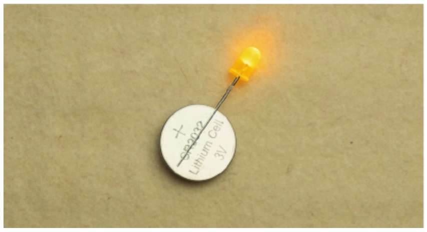
Use your thumb and index finger to hold the legs against the battery, and bam! You have a working circuit! This temporary arrangement of components allows electricity to flow from the battery through the LED and back to the battery, giving the LED the power it needs to light up.



The leg of the anode (positive) side of the LED is usually longer than the cathode (negative). Some manufacturers put a flat spot on the base of the lens of the LED by the cathode leg.



Within the lens of the LED, two pieces extend up from the legs. The piece attached to the cathode leg is the one with the wider top that slants over the shorter portion of the anode.



Maxwell Lander

If all else fails, try connecting an LED to an appropriate power source. If it doesn't light up, it probably means you have it in the wrong way. Flip it and give it another try.

DETERMINING POLARITY

In the previous circuit, you may have noticed that the direction in which you connect the LED determines whether the LED will light up.

LEDs are *polarized*—they have a predetermined *polarity*. Nonpolarized components can be connected in either direction. Polarized components must be connected in a particular direction. If the connections of a polarized component are reversed, the circuit will not function correctly.

The *D* in *LED* refers to a *diode*, a component that allows electricity to pass through it in only one direction. If an LED is connected backward, electricity will not be able to pass through it, which means it will not light up.

For through-hole LEDs, there are several ways to determine the polarity.

Determining the polarity of other components tends to vary, but be on the lookout for a + or – sign, which will indicate the positive or negative side of the component. Sometimes a single dash or stripe on one side of the component is an indicator of the negative side. Red (positive) and black (negative) wires are also a clue.

OHM'S LAW

The temporary circuit we created is a quick-and-dirty way to light up an LED, but there are a few more things to consider when constructing a technically correct circuit. In that circuit, the LED is receiving a bit more than the desirable amount of current. Although the circuit "worked" and the LED lit up, there are longer-term consequences for giving the LED too much current. Excessive current can shorten the LED's life or even burn it out. Because this battery supplies a relatively low amount of current and an LED is not a highly sensitive or expensive component, it's possible to take a more casual approach. But ideally, it's best to design a circuit that respects the needs of its components. To do this, Ohm's law can be used to determine how much resistance is needed.

Ohm's law states that voltage (*V*) is equal to current (*I*) times resistance (*R*). The terms for these three key concepts are defined as follows:

Voltage

The difference in electrical energy between two points. It is measured in *volts* (V).

Current

The quantity or amount of electrical energy passing a particular point. It is measured in *amps* (A) or *milliamps* (mA).

Resistance

The measure of a material's ability to prevent the flow of electricity. Resistance is measured in *ohms*, which is represented by the ohm symbol (*Ω*).

As with any equation, in Ohm's law, two of the three variables can be used to determine the third. All three variations of this equation are helpful when constructing circuits:

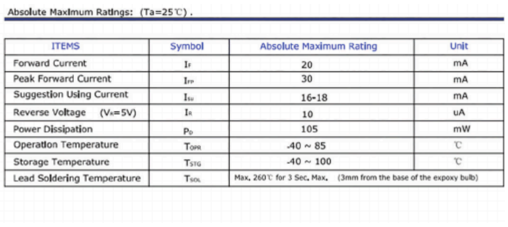
- $V = I \times R$
- $I = V \div R$
- $R = V \div I$

To determine how much resistance is needed in our circuit, let's start with the voltage. The voltage that needs to be used up is equal to the difference between the source voltage (Vs) and the forward voltage (Vf), the voltage used up by the LED. So the equation is actually as follows:

- $R = (Vs - Vf) \div I$

The Vs is that which is supplied by the battery. In this case, the CR2032 battery supplies 3 V.

You can find the Vf and the current required by the LED on the LED's *datasheet*, a document supplied by a component's manufacturer.

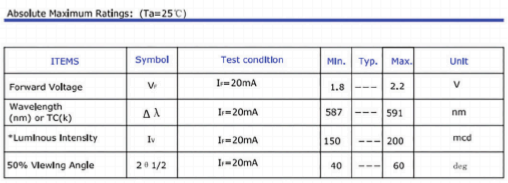


Absolute Maximum Ratings: (Ta=25℃) .

ITEMS	Symbol	Absolute Maximum Rating	Unit
Forward Current	I_F	20	mA
Peak Forward Current	I_{FP}	30	mA
Suggestion Using Current	I_{SU}	16-18	mA
Reverse Voltage (V$_R$=5V)	I_R	10	uA
Power Dissipation	P_D	105	mW
Operation Temperature	T_{OPR}	-40 ~ 85	℃
Storage Temperature	T_{STG}	-40 ~ 100	℃
Lead Soldering Temperature	T_{SOL}	Max. 260℃ for 3 Sec. Max. (3mm from the base of the expoxy bulb)	

This document provides information about the component, including the component's electrical needs and tolerances, mechanical diagrams of its physical packaging, diagrams of any pins or connections, and details on intended use and expected performance.

According to the datasheet, the forward



Absolute Maximum Ratings: (Ta=25℃)

ITEMS	Symbol	Test condition	Min.	Typ.	Max.	Unit
Forward Voltage	V_F	I_F=20mA	1.8	---	2.2	V
Wavelength (nm) or TC(k)	Δ λ	I_F=20mA	587	---	591	nm
*Luminous Intensity	I_V	I_F=20mA	150	---	200	mcd
50% Viewing Angle	2 θ 1/2	I_F=20mA	40	---	60	deg

A detail from the LED's datasheet

voltage of this LED is rated as 1.8–2.2 V—so let's say 2 V. The LED requires 16–18 mA of current, with a maximum of 20 mA. Let's use 17 mA, or 0.017 A, for the calculations.

Now that you have all of the necessary information, the equation will play out as follows:

- *R* = (*Vs* – *Vf*) ÷ *I*
- *R* = (3 V – 2 V) ÷ 0.017 A
- *R* = 58.82 Ω

These calculations tell you that you should ideally add 58.82 Ω of resistance to the circuit.

When you order a component online, there will usually be a link to the datasheet on the web catalog page from where the component was ordered. In rare cases, datasheets will be included in the packaging when you purchase a component. If not, the easiest place to start is the internet: Open your favorite search engine and type in the part number and the word *datasheet* to find it. You will likely find a PDF of the datasheet on the manufacturer's or distributor's website or in a datasheet database, of which there are many on the internet. If you don't know the part number, you can even try a description such as "5 mm yellow LED."

The bonus is that these days, many distributors provide an abundance of additional resources on their parts pages. They include links not only to datasheets but also to tutorials, circuit schematics, circuit diagrams, circuit board design files, sample code, and sometimes even example projects.

UNDERSTANDING RESISTORS

Now you know how much resistance you need. But how do you add resistance to the circuit?

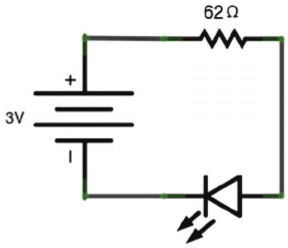
A *resistor* is a component that resists the flow of electricity. It can be included in a circuit to use up extra electricity not needed by the load.

You know from the Ohm's law equations that you need approximately 58.82 Ω resistance for the circuit. However, resistors come in set values, so there may not always be the exact resistor that meets your needs.

If you don't have the exact resistor you're looking for on hand (or if it doesn't exist), you do can one of two things:

- Use the next-largest value. In your circuit, this will be a 62 Ω resistor. However, a more common 68 Ω or even 100 Ω resistor would work just fine.
- Combine two resistors in a row that add up to the correct value (e.g., 56 Ω + 3 Ω = 59 Ω).

When you look at the actual component, you can determine the value of a through-hole resistor by the color bands displayed on it. Each color indicates a value. A 62 Ω resistor is marked



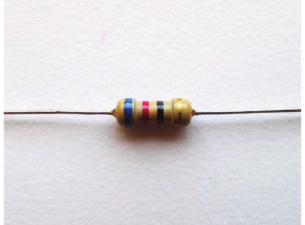
Circuit schematic depicting a 3 V

blue, red, black, and gold.

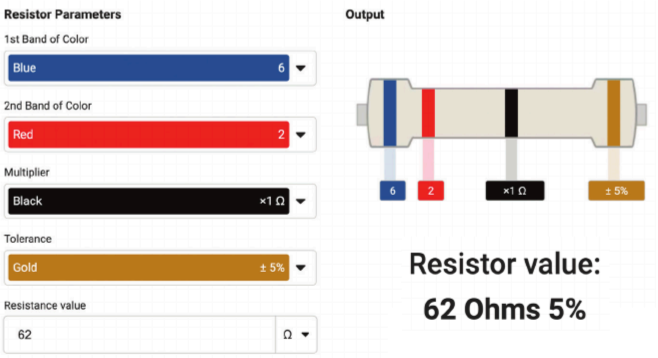
Resistor values can be decoded by consulting a resistor color chart, going to a resistor-calculator website, or downloading a resistor application for your smartphone.

When reading the color bands on the resistor, orient it so that the silver or gold band is on the right, then read the colors from left to right. The first two bands indicate the first two digits of the number, the third band indicates the multiplier for that number, and the fourth band indicates the tolerance—in the case of the resister pictured above, blue (6), red (2), black (1), and gold (±5%).

battery, an LED, and a 62 Ω resistor



A resistor with the stripes blue, red, black, and gold, indicating a value of 62 Ω



Resistor Parameters

1st Band of Color

Blue 6

2nd Band of Color

Red 2

Multiplier

Black ×1 Ω

Tolerance

Gold ± 5%

Resistance value

62 Ω

Output

Resistor value:
62 Ohms 5%

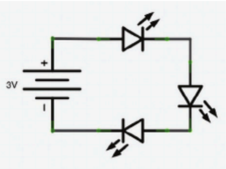
Using a resistor calculator to decode the colors of a resistor

RESISTOR COLOR CODES

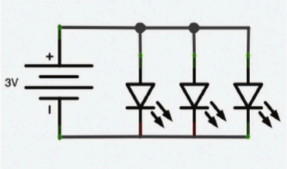
COLOR	VALUE	MULTIPLIER	TOLERANCE	
Black	0	1	—	
Brown	1	10	±1%	
Red	2	100	±2%	
Orange	3	1K	—	
Yellow	4	10K	—	
Green	5	100K	±0.5%	
Blue	6	1M	±0.25%	
Purple	7	10M	±0.1%	
Gray	8	100M	±0.05%	
White	9	1000M	—	
Gold	—	1/10	±5%	
Silver	—	1/100	±10%	
None	—	-	±20%	

SERIES AND PARALLEL

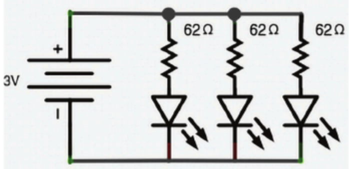
Now you know how to create a circuit with one LED, but how about three? When adding additional components to the circuit, it is important to understand the difference between *series* and *parallel*.



Three LEDs in series A



B Three LEDs in parallel



Three LEDs in parallel with resistors C

Series

In a series, components, such as LEDs, are connected in a row. Electricity flows through one, into the next, and then into the next (Figure A).

Parallel

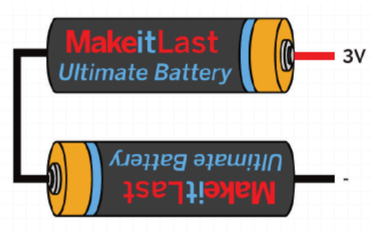
In a parallel configuration, components are connected side by side, each with an independent connection to power and ground.

LEDs can be connected in series, but the power source must supply adequate voltage. The factor that needs to be considered is called *voltage drop*. When electricity passes through and gets used up by a component, the voltage drops before it moves on to the next component.

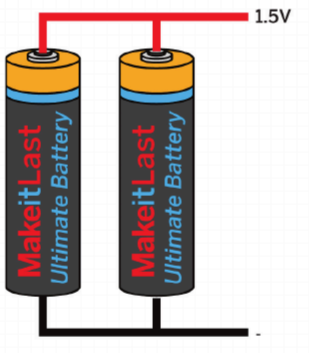
The LEDs used in this chapter are rated for a forward voltage drop of 1.8–2.2 volts. This means that if a 3 V battery is used as the power source, the first LED would get 2 V, the second 1 V, and the last no voltage. This will obviously not yield the desired result. One way to fix this is to use a power source whose voltage could accommodate this voltage drop. Because each of the three LEDs has a voltage drop of around 2 V, you would want a battery pack that provided at least 6 V.

However, there is a preferable way to connect multiple LEDs to a power source—in parallel. In this configuration, each LED receives the same amount of voltage, but the current is divided between them. The only thing missing from this circuit is the resistors (Figure B).

If all the LEDs are the same, they will each use the same resistors (Figure C). However, if you add an LED that requires a different amount of voltage or current, you can use Ohm's law to calculate which resistor it needs.



AAA batteries in series



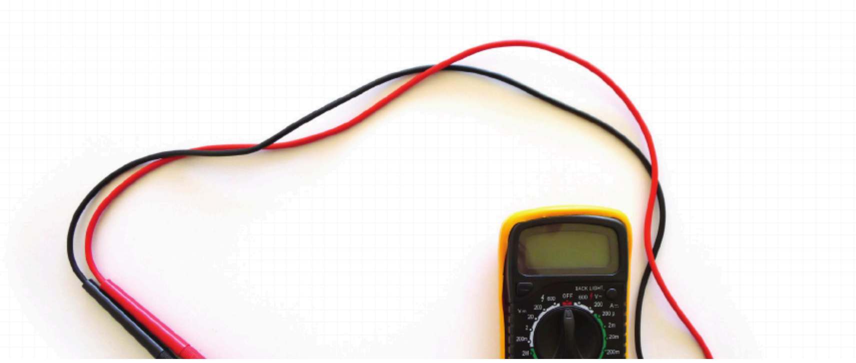
AAA batteries in parallel

Batteries can also be placed in series or in parallel. When batteries are connected in series, the voltage of the two batteries is added together. When they are connected in parallel, the voltage stays the same but their available current is added together.

For instance, AAA batteries usually supply around 1.5 V. If two AAA batteries were placed in series, their voltage would be combined and the resulting battery pack would provide 3 V and the same amount of current as a single AAA battery. If two AAA batteries were placed in parallel, however, the battery pack would supply 1.5 V but twice as much current.

USING A MULTIMETER

Because you cannot see, smell, or hear electricity, you'll need a special tool to detect it. A multimeter, shown below, can be used to check *continuity* (whether current flows unimpeded through two points) as well as to measure voltage, resistance, and current. Multimeters usually have a dial (or buttons) used to select a particular function and probes for making connections with whatever it is you are measuring.



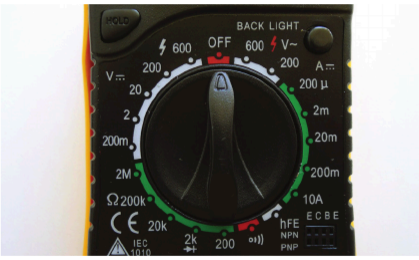
A multimeter

A variety of multimeter tutorials are available in basic electronics books as well as on the internet. Here, we will just cover the basic concept of what a multimeter does and when you might use it when creating circuits.

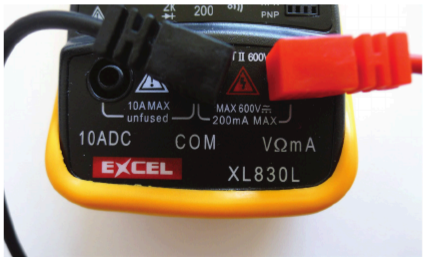
Continuity

The simplest but perhaps most useful function of a multimeter is the *continuity*, or *conductivity*, test. This is most often marked on the dial with a speaker or audio symbol because meters will beep when the test is positive. Once the dial is in position, simply place the probes at two locations across which you'd like to test the continuity or conductivity.

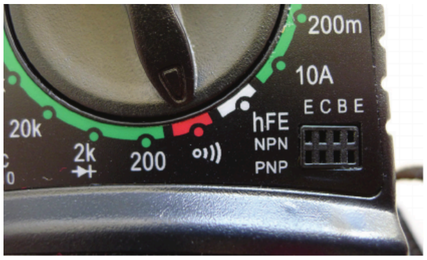
This can be used to check the continuity of a questionable connection: Place the probes on either side of the questionable connection, and if the meter beeps, you're good to go.



The dial of a multimeter is used to select the function.



The connection for the probes; on this meter, you will need to move the red probe to the left if you are measuring current higher than 200 mA.



The knob set for a continuity test

Using a multimeter to test the conductivity of

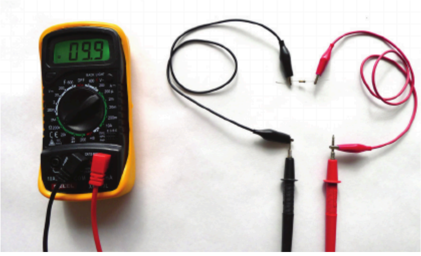
It can also be used to check for short circuits by placing the probes in two places that are *not* supposed to be connected. If the meter beeps, you know you've got a short circuit somewhere.

Finally, it can be used to check the conductivity of a material: Place the probes at two points on a material and see whether you're able to establish a connection across it. This can be especially useful if you're shopping for conductive materials in unusual places like a fabric store. I recommend investing in a pocket multimeter for just these occasions.

If you are testing a material that you *think* is conductive but the meter doesn't beep, the next step is to measure resistance in ohms, just in case it conducts well enough for your needs.

Resistance

To measure resistance, turn the knob to the portion of the dial marked with the ohm sign *conductive fabric*



Measuring the resistance of a fixed resistor

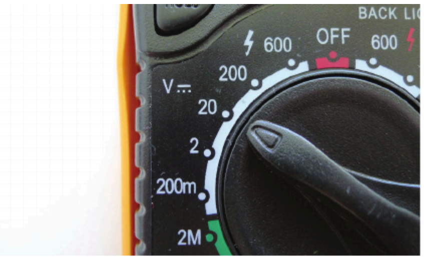
(Ω) and place the probes on either side of the component or material you would like to measure. Alligator clips can be used to securely hold the component during testing.

With this setting, you can check the value of a fixed resistor, monitor the changing resistance of a variable resistor, or determine the resistance of a material like conductive thread. If your multimeter is not auto-ranging, you will have to select a resistance setting that's in the range of what you expect the component to be.

Voltage

Multimeters can also measure voltage. This is helpful for checking the state of a battery or determining whether components of a circuit are receiving the voltage they need. Turn the dial

on the multimeter to the V⎓ setting, set it to the range of voltage you expect to read, and place the probes on either end of whatever you want to measure.



The multimeter knob set to measure voltage

The voltage reading for a fresh CR2032 battery is 3.26 V.

The voltage reading for a used CR2032 battery is 1.37 V.

Current

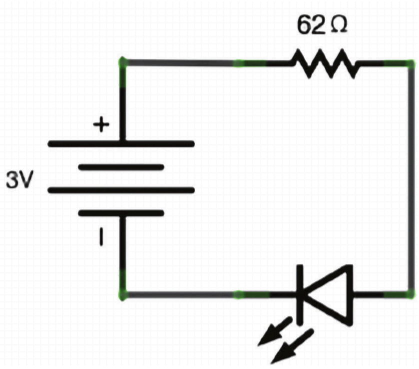
The process for measuring current with a multimeter is a bit different. The meter needs to be *in series* with the circuit in order to determine how much current is being pulled. Turn the dial to the A or mA setting and select the appropriate range. With some meters, you may need to move the probe to another terminal at the bottom of the meter. (Check your multimeter manual for details.) Once you're set up, find a location where you can insert the meter into the circuit and take a reading. Knowing how much current a circuit draws at its peak usage and over time can be extremely helpful in terms of determining which battery to select for your project.

Constructing Circuits

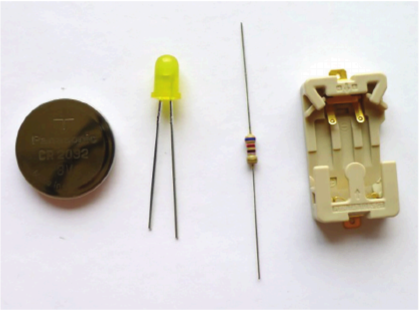
As you learned earlier, conductors, or conductive materials, are those through which electricity can pass. When you construct a circuit, conductive materials provide the pathway for electricity to flow from one component to another.

Now that you have some understanding of how circuits work and how to measure different aspects of them, you can start to think about how to physically construct them. This section will illustrate a variety of ways to bring this basic LED circuit to life.

As you move through different iterations of this circuit, you'll see that through the use of different conductive materials, it can take on many shapes and sizes. The core electronic components you work with will be the same in each circuit. What will differ are the materials and tools you use to create the connections between the components.



A circuit diagram of a 3 V battery, an LED, and a resistor



CR2032 battery, LED, resistor, battery holder

Remember that a circuit schematic shows only the electrical connections. It does not reflect the physical layout or the materials or tools used to create the electrical connections.

The core parts you'll be using are as follows:

- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED

These are all basic and inexpensive components. The LED is a standard one you'd find in any basic electronics kit. The CR2032 battery meets the needs of the circuit in terms of both its voltage and current ratings. It also features a slim profile, which prevents your circuit from getting too bulky. The battery holders you are working with are intended for surface-mount electronics, but you will be modifying them for through-hole and soft-circuit applications. (Note that the minus sign on the base of the holder shows you which terminal of the battery holder is ground.)

Using these components, you will create six versions of this circuit, constructed with alligator clips, wires, a breadboard, a prototyping board (also called a protoboard), conductive fabric, and conductive thread. (Each of these tools and materials is discussed later in detail.) Let's get started!

HOW TO: Alligator Clip Circuit

Alligator clips provide a quick way to prototype simple, temporary circuits. This method is used with many e-textile tool kits.

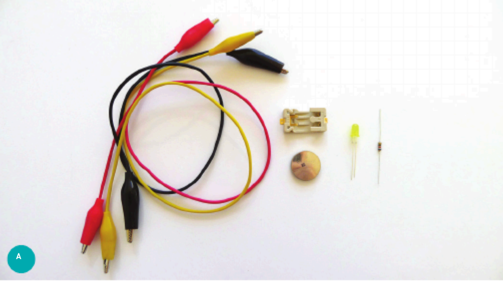
Parts and materials:

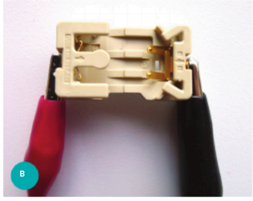
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- (3) alligator clip test leads: red, black, and yellow

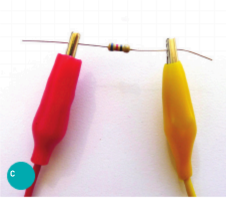
Steps:

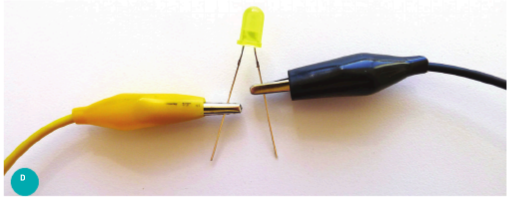
1. Gather the parts for the alligator clip circuit (Figure A).
2. Clip a red cable to the positive terminal (+) of the battery holder and a black cable to the negative (–) (Figure B). (The use of these standardized colors helps you remember what's what.)
3. Clip the other end of the red to the resistor. Clip the yellow alligator clip to the other side of the resistor (Figure C).
4. Connect the other end of the yellow alligator clip to the positive side of the LED. Clip the other end of the black cable to the negative leg of the LED. Spread the legs of the LED a bit to ensure that they don't touch each other (Figure D).
5. Add a battery to light up the LED. The alligator clip circuit is now complete (Figure E)!

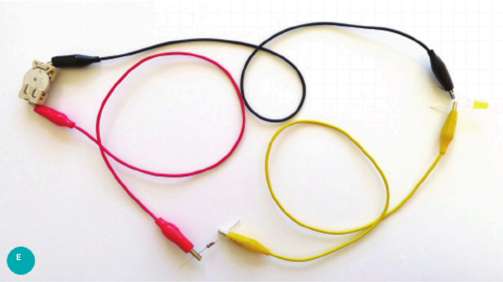
NOTE: Alligator clips grab onto components nicely but tend to slide around a bit, so be sure they do not touch each other so as to prevent short circuits.

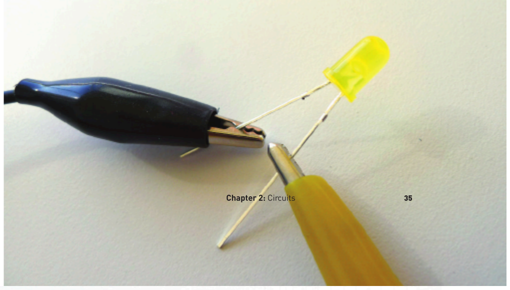


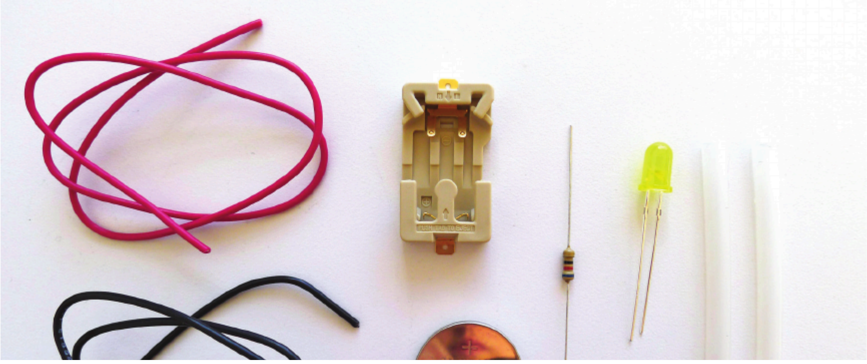


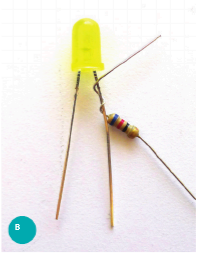


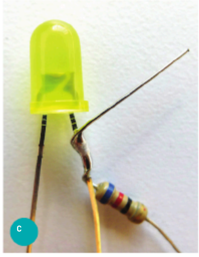












HOW TO: Wire Circuit

Wires can also be used to create connections between components in a circuit. Twisting, bending, or crimping establishes a base physical connection, and soldering those points establishes a secure electrical connection. In the following example, you will use some 22-gauge hookup wire to connect components and heat-shrink tubing to insulate connections.

Parts and materials:

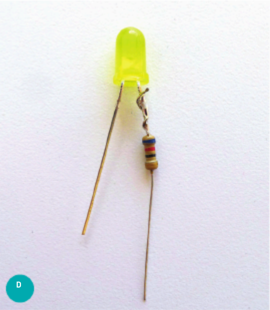
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- 22 AWG solid-core hook-up wire, in red and black
- Heat-shrink tubing

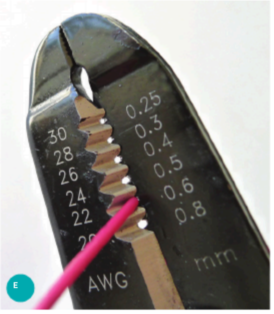
Tools:

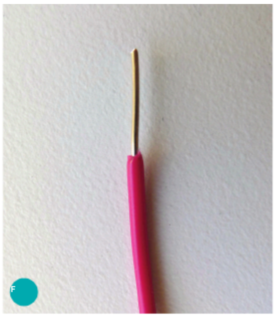
- Wire stripper
- Soldering iron and solder
- Heat gun

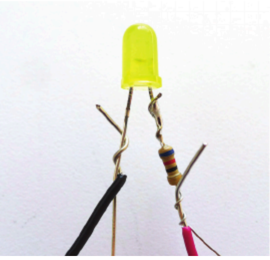
Steps:

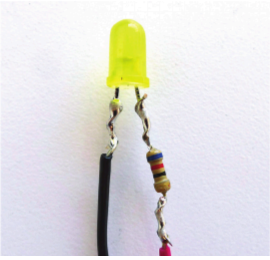
1. Gather the parts for the wire circuit (Figure A).
2. Tightly wrap the leg of a resistor around the positive leg of the LED (Figure B).
3. Solder it in place (Figure C).

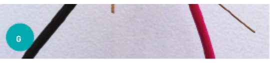


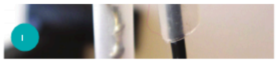




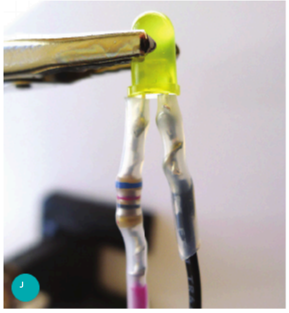


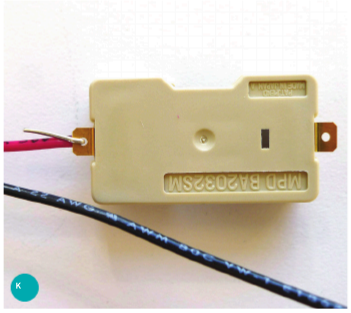


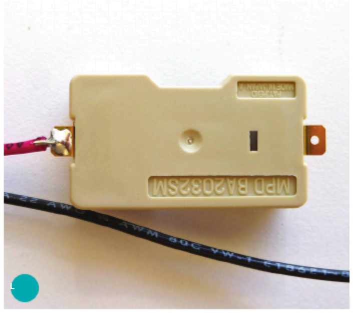


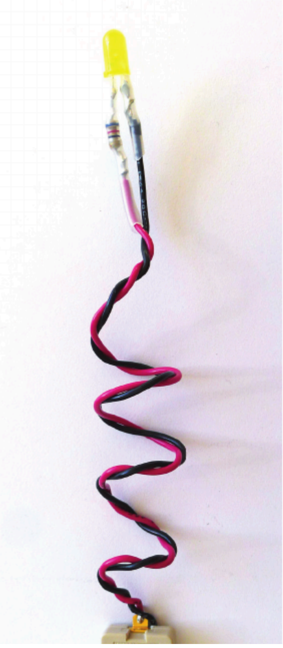


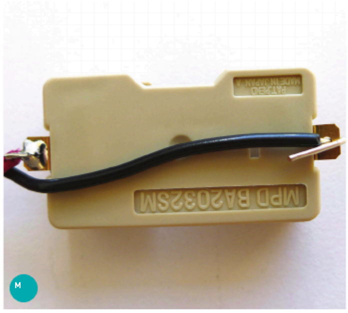
4. Trim any excess (Figure D).
5. Prepare your wires using wire strippers. Cut a short length of the red wire. Place the wire in the slot of wire strippers marked "22 AWG" or "0.6 mm," about a centimeter from one end of the wire (Figure E).
6. Close the wire strippers fully, hold the long end of the wire with your other hand, and pull the strippers gently toward the short end. This will cut and remove the sleeve and expose the wire inside (Figure F).
7. Repeat for the other end of the red wire, then do the same to both ends of your black wire. Your wires are now good to go.
8. Wrap the end of the black wire around the negative leg of the LED. Wrap the end of the red wire around the resistor leg (Figure G).
9. Solder both in place and trim (Figure H).
10. Place some heat-shrink tubing over the exposed connections. Hold everything in place using helping hands (Figure I).

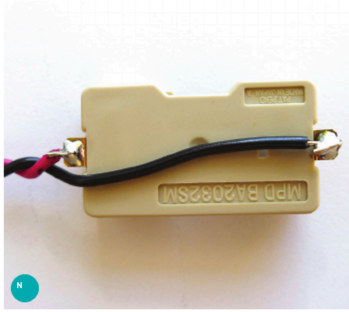












11. Use a heat gun to shrink the heat-shrink so it is snug against the connections. This area of the circuit now has proper insulation (Figure J).
12. Connect the other end of the red wire to the positive terminal of the battery holder (Figure K).
13. Solder the red wire in place to secure the connection (Figure L).
14. Connect the other end of the black wire to the negative terminal of the battery holder (Figure M).
15. Solder in place (Figure N).
16. The free-form wire circuit is now complete! You can create your own variations on this technique using different wire types and by

varying the length of wire (Figure 0).

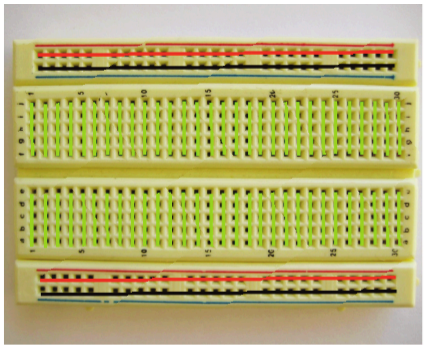
NOTE: The position of the exposed connections creates a high potential for a short circuit. These connections can be insulated by using some heat-shrink tubing.

HOW TO: Breadboard Circuit

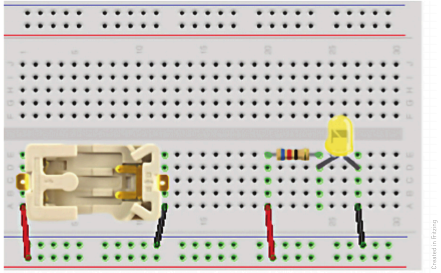
For slightly more complex circuits, breadboards are an excellent solution. They allow you to connect and disconnect through-hole components with ease. Underneath the surface of the breadboard are steel clips that connect wires to conductive traces that run beneath the holes.

On either half of the breadboard, all the holes in each column are connected to one another. The gap in the center separates the halves of the columns, so the only way to make connections to all the holes in a column is to place a jumped wire connecting the two halves. The two top and bottom rows are connected horizontally and are generally used for negative and positive power connections.

When looking at the breadboard, you'll notice that numbers and letters indicate the row and column of each hole. You'll use these markers for reference in your wiring.



Connections beneath a breadboard



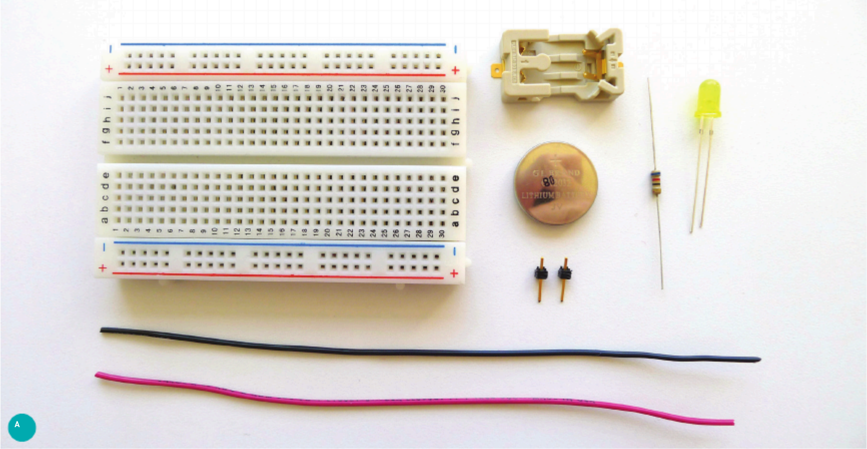
Circuit diagram of a 3 V battery holder, an LED, and a resistor connected on a breadboard

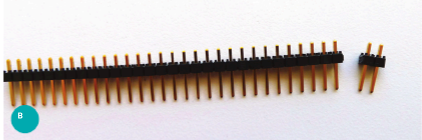
Parts and materials:

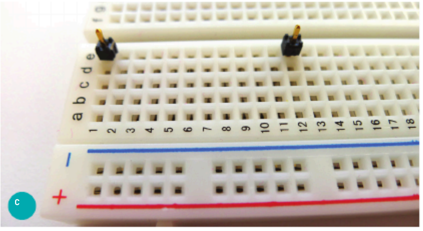
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- 22 AWG solid-core hook up wire, in red and black
- (1) breadboard
- (2) breakaway 0.1-inch (2.54 mm) straight male header pins

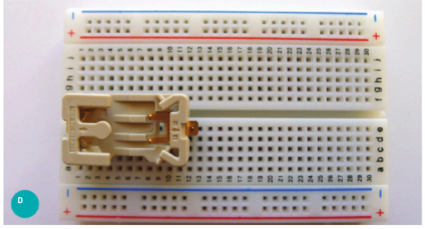
Tools:

- Wire strippers
- Soldering iron and solder


A


B

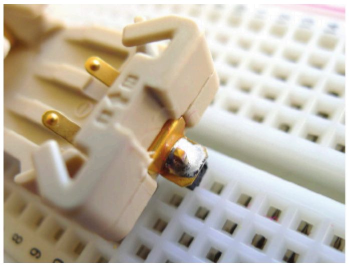

C

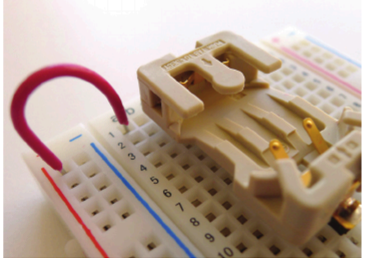

D

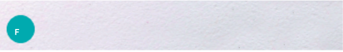
Steps:

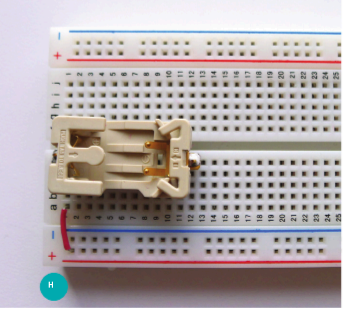
1. Gather parts for the breadboard circuit (Figure A).
2. Snap off two male headers for use with the battery holder (Figure B). This will make it easy to connect to the breadboard.
3. Break apart the two male headers and place a single male header in holes E1 and E12, with the longer end of each header pointing down into the breadboard (Figure C).
4. Place the battery holder on top of the headers so they are inserted into the holes on either side. Orient the battery holder so E1 connects to the positive side of the battery holder (Figure D).

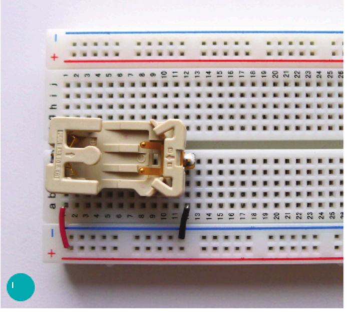
NOTE: While wire of any color will work, using the specified colors will help you to more quickly read which connections are being made when you look at the circuit.

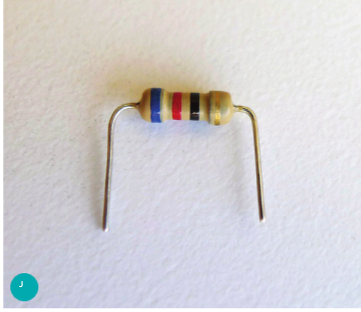












5. Solder the connection between each header and battery terminal (Figure E).

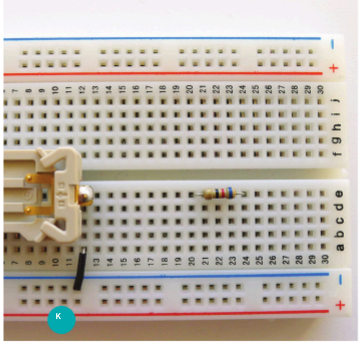
6. Cut a short length of red wire, and strip both sides. The exposed wire should be long enough so that it can be inserted into the breadboard but not so long that it will leave additional length exposed (Figure F).

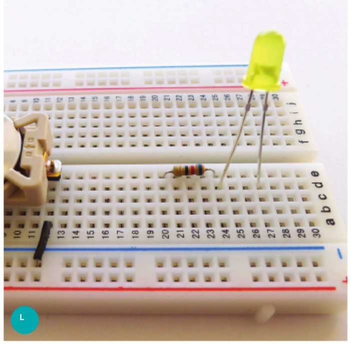
7. Place one end of the red wire into the hole marked A1. Place the other end into a hole in the power bus, marked by the + sign (Figure G).

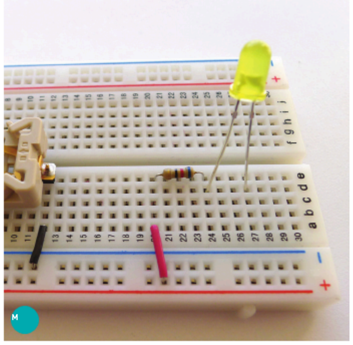
8. (Optional) Trim the wire to a shorter length to make for a tidier circuit (Figure H). Make sure you make the correct connections and the wire is long enough to be fully inserted into both holes. You can leave the wire the length that it is.

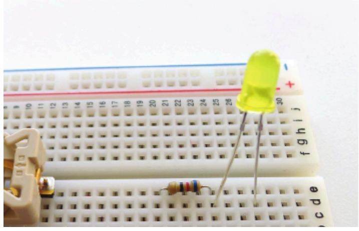
9. Do the same using a black wire to connect A12 and the ground bus, marked by the – sign (Figure I). The power and ground of the battery are now accessible through any holes in the + and – rails proximate to column A.

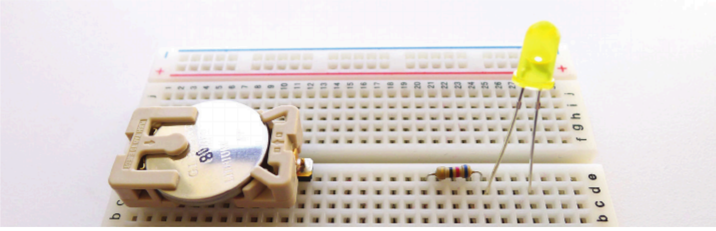
10. Bend the resistor's legs to a right angle. Trim to a length slightly longer than the depth of the breadboard (Figure J).

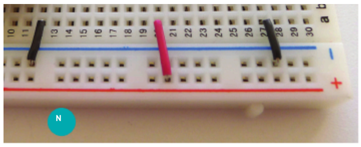


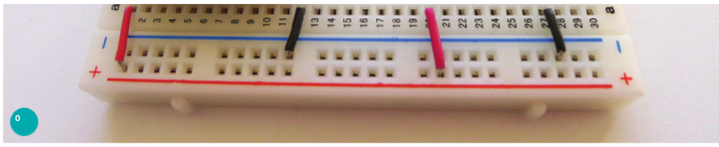









N


O

11. Place the resistor on the breadboard so the legs are inserted in E20 and E24 (Figure K).
12. Place the LED so the positive leg is in D24 and the negative leg is in D27 (Figure L).
13. Use a red wire to connect A20 and + (Figure M).
14. Use a black wire to connect A27 and – (Figure N).
15. The circuit is now complete! Insert a battery in the battery holder, positive side up (+), to light the LED (Figure O).

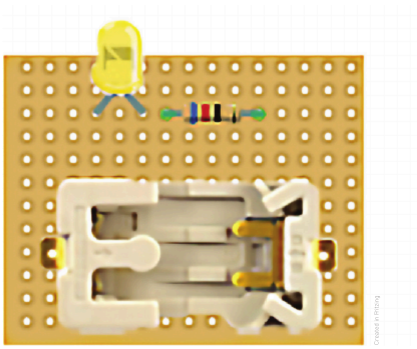
NOTE: Once the headers are soldered, you will be able to remove this battery pack for use in other projects.

HOW TO: Protoboard Circuit

The protoboard is a logical follow-up to a breadboard. It uses the same components and spacing but makes connections that are far more secure and robust than possible on a breadboard.

There are many varieties of protoboard. For this example, you will use the type that has no connections between any of the holes. Other types of protoboard will be reviewed in the following chapter.

The shiny pieces of silver- or copper-colored metal around the holes are called *pads*. If you have pads on only one side, make sure they are on the bottom, where the legs come out.



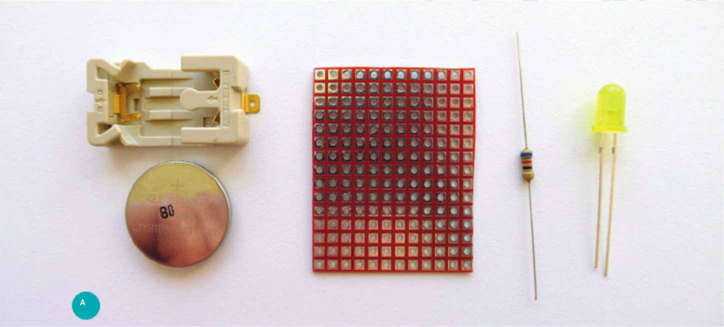
Circuit diagram of a 3 V battery holder, an LED, and a resistor connected on a protoboard

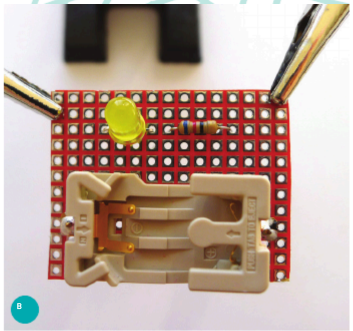
Parts and materials:

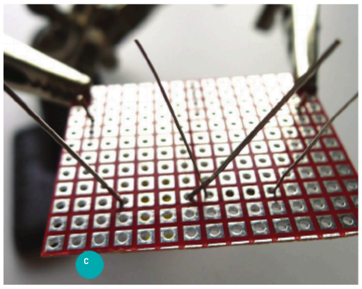
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- (2) breakaway 0.1-inch (2.54 mm) straight male header pins
- (1) small piece of protoboard without any traces connecting the pads, cut to size

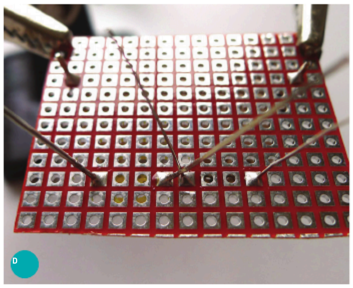
Tools:

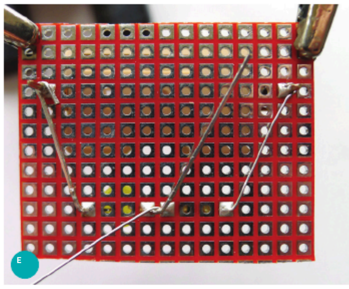
- Soldering iron and solder
- Small snips or flush cutter
- Helping hands

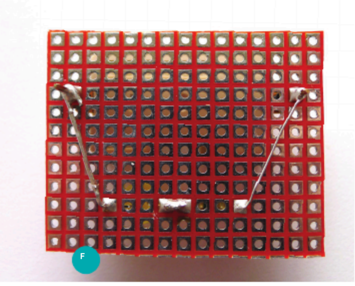


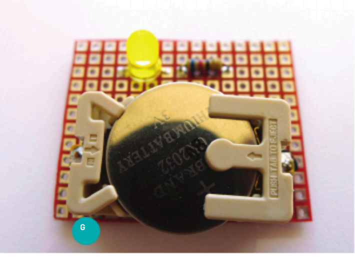












Steps:

1. Gather parts for the protoboard circuit (Figure A).
2. Use some helping hands to stabilize your protoboard. Place your components. For this circuit, you are placing the negative leg of the LED close to the negative terminal of the battery, the positive leg of the LED close to one side of the resistor, and the other side of the resistor close to the positive terminal of the battery. These locations will allow you to create connections with ease (Figure B).
3. Turn the board over. For components like LEDs and resistors, you can bend the legs slightly so they don't fall out when you flip the board (Figure C).
4. Secure the components in place by soldering a connection between the component legs to the pad on the protoboard surrounding them (Figure D).
5. Arrange the legs (and jumper wires, if necessary) so they create the necessary connections to complete the circuit. Solder them in place (Figure E).
6. Snip any excess wire as needed (Figure F).
7. Turn the board over and add the battery to light the LED. The circuit is now complete (Figure G).

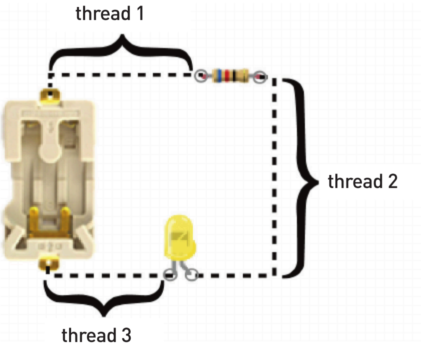
HOW TO: Conductive-Thread Circuit

Working with conductive thread is a first step into the world of soft circuitry. When using conductive thread, electrical connections are sewn rather than soldered. This allows you to create circuits that are soft and pliable.

Before you start assembling the circuit, you should prepare your parts. In order to sew the LED and resistor in place, you will need to modify their shape a bit.

For this technique, you can use any through-hole component that has long legs.

Let's use an LED. You will also need a pair of needle-nose or round-nose pliers.



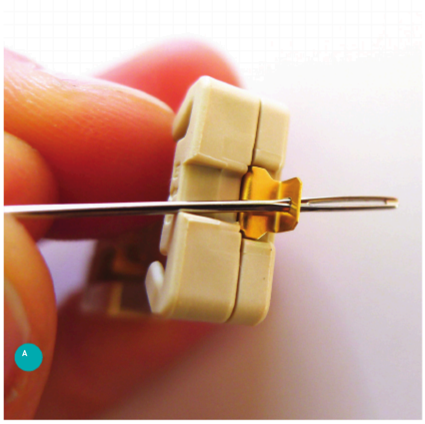
Circuit diagram of a 3 V battery holder, an LED, and a resistor connected with conductive thread

Parts and materials:

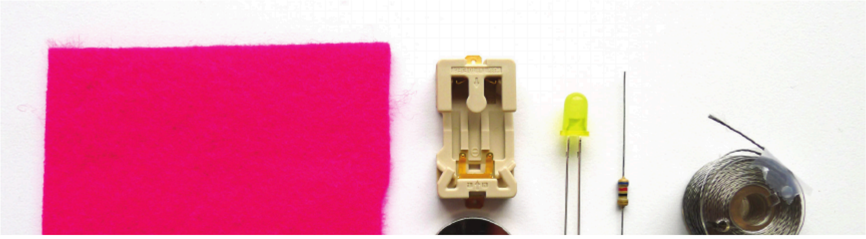
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- Conductive thread
- Fabric
- Fabric glue

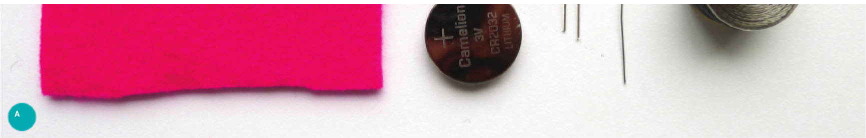
Tools:

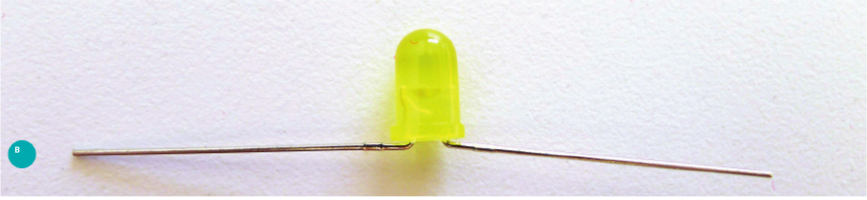
- Needle
- Scissors
- Needle-nose pliers

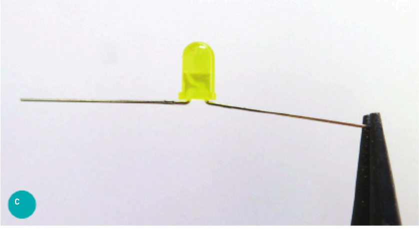


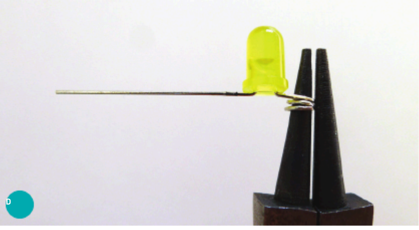
NOTE: When selecting a needle, be sure it can pass fully through the holes in your components. The eye of the needle shown in Figure A is too large to pass through the hole of the battery holder.

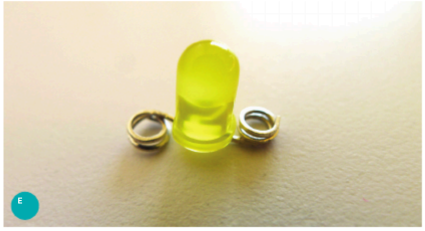


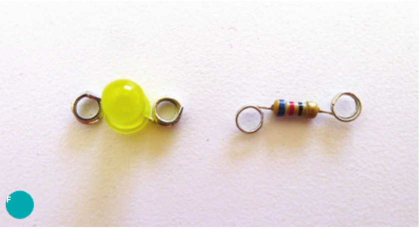






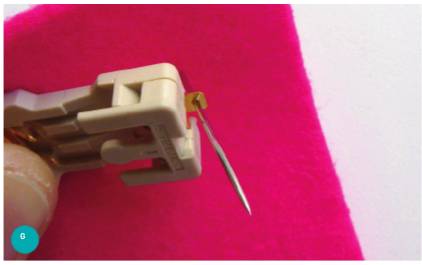


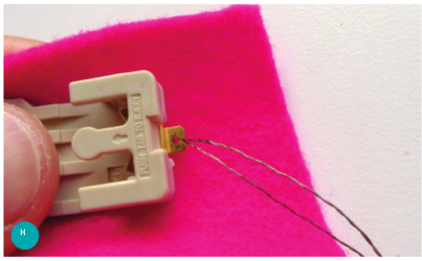




Steps:

1. Gather the parts for the conductive thread circuit (Figure A).
2. Bend the LED's legs so they are parallel to the base, as if the LED were doing a split. This will allow the LED to sit flat on the fabric (Figure B).
3. To prepare the LED to be sewn, grab the end of one leg with the pliers (Figure C).
4. Twist the pliers so the leg of the LED rolls around the pliers to create a loop (Figure D).
5. Repeat on the other side. These loops will later be sewn and secured with stitches of conductive thread (Figure E).
6. Repeat the same technique with the resistor (Figure F).
7. Begin to assemble the circuit by placing the battery holder. Thread the needle with a piece of conductive thread. If you find that the thread is fraying at the end, use a bit of beeswax or moisture to tame the frays.





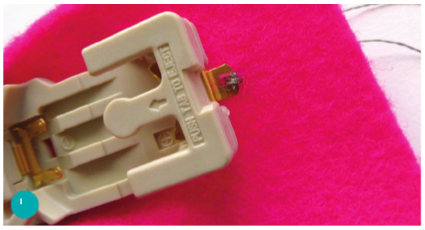
Pull the two ends of the thread so they are equal lengths and knot them together. Pass the needle from the back of the fabric up through the hole of the positive terminal of the battery holder (Figure G).

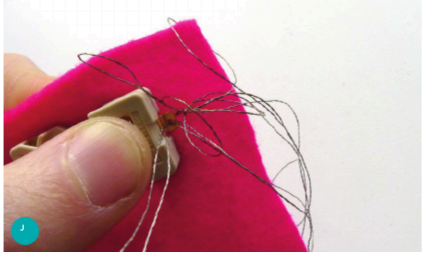
8. Pull the thread through (Figure H).

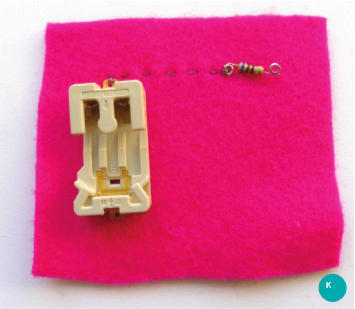
9. Repeat the stitch around the battery holder several times to make a secure connection (Figure I).

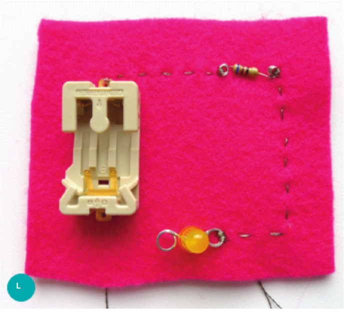
10. Pull the thread slowly, and take care to avoid tangling the conductive thread (Figure J).

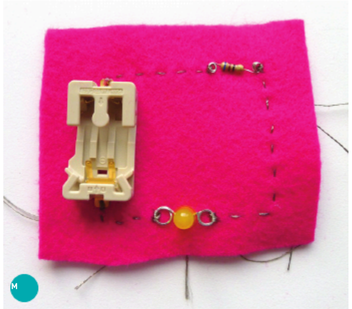
11. Once the terminal of the battery holder is secure, continue to sew in the direction of the resistor. Once you reach the resistor, make several secure stitches around the loop, tie a knot in the back, and snip off the excess thread. As indicated in the circuit diagram, *do*

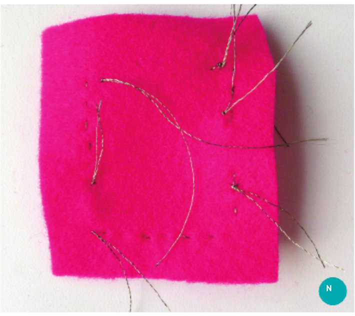


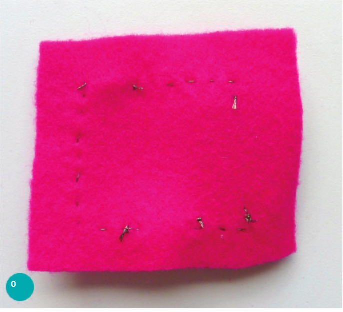












not continue on to the other side of the resistor with the same piece of thread. Doing so will create a short circuit (Figure K).

12. Using a *second* piece of thread, sew the other loop of the resistor, then sew a line to the LED. Stitch around the positive leg of the LED, knot it in back, and trim it (Figure L).

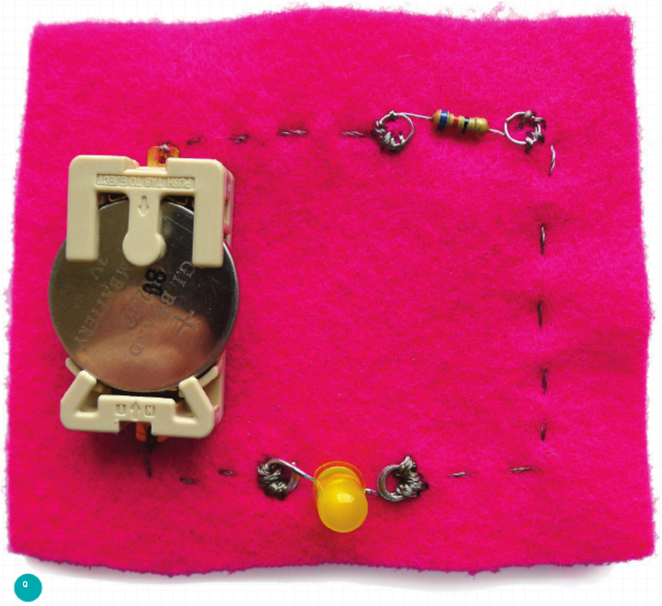
13. Using a *third* piece of thread, sew the negative loop of the LED and connect it to the negative terminal of the battery holder. Tie and trim (Figure M).

14. Review the backside of the circuit. Untrimmed conductive-thread tails create a high probability for short circuits. If a tail accidentally brushes another, it can create an opportunity for the electricity to follow an incorrect path (Figure N).

15. Trim the threads close but not so close that the knots unravel [Figure O].

16. Secure the knots with fabric glue, fray stopper, or nail polish [Figure P].

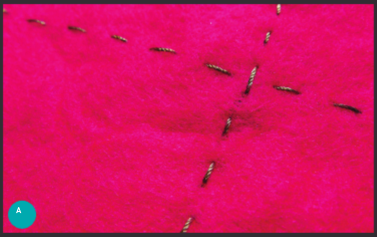
17. Add a battery to complete the circuit [Figure Q].

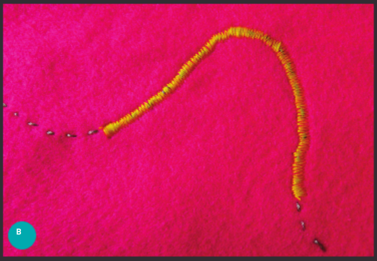


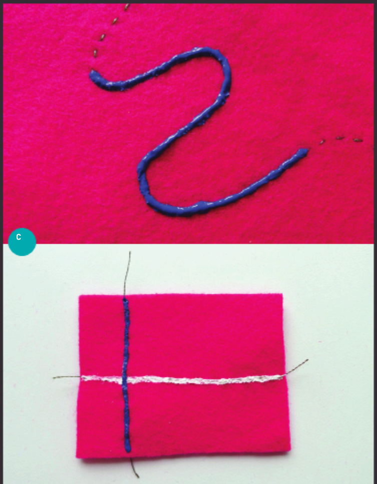
Q

NOTE: Conductive This type of CR2032 battery holder has been adopted by the DIY community because the terminal holes can be used for sewing. Keep in mind, however, that they are not intended for this purpose. The edges of the holes are sharp, and conductive thread sewn very tightly can get cut and fray Conductive thread sewn too loose, on the other hand, will result in an unreliable connection. When you are sewing, try to find the middle ground in the tightness of your stitches.

WARNING: Conductive thread has a strong tendency to tangle. Take your time when sewing, and keep an eye on the thread as you go!







Insulating Conductive Thread

When creating soft circuits, be vigilant about insulation, as many of these materials are by default not insulated.

Your circuit design should, of course, lay out the conductors so they do not touch each other You also need to consider how this layout will perform in the context of the intended use. Clothing is inherently floppy. If you take the garment off and toss it in a pile, could it inadvertently create a short circuit?

Many material approaches can reduce the likelihood of short circuits:

A Layout

The physical layout of the circuit itself can prevent conductors from touching. When crossing conductive thread stitches, going on top of the fabric with one thread and below it with the other enables the nonconductive substrate to act as an insulator between the two conductors.

B Stitching

Nonconductive thread can provide a light layer of insulation in key spots. A satin embroidery stitch or zigzag machine stitch can cover a line of conductive thread and shield it from contact with others.

C Coatings

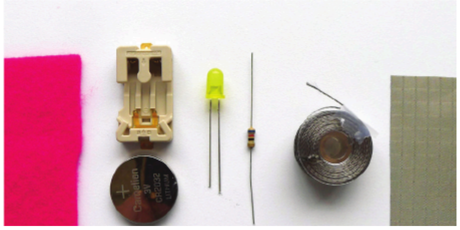
Fabric glue, fabric paint, nail polish, and other coatings can provide spot insulation where it's needed.

Nonconductive fabric can also protect patches or traces of conductive fabrics with an overall layer of insulation. It can be secured with iron-on adhesive or sewn in place. This is helpful for when the garment is folded, crumpled, or handled in an unexpected way. Linings also provide lots of opportunities: They can either hide what is going on underneath, or you can use them as a substrate for your electronics so the circuit is removable from the garment.

HOW TO: Conductive Fabric Circuit

Another approach to soft circuitry is using conductive fabric. Iron-on conductive fabric is particularly exciting because a circuit can be created, cut, and adhered quickly. Also, there is the potential for designs to become visually intricate with limited effort.

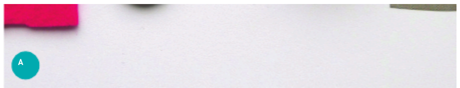
Parts and materials:

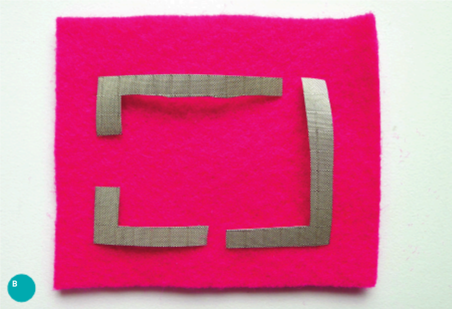


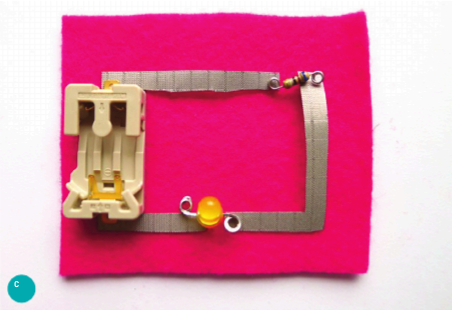
- (1) CR2032 battery
- (1) CR2032 battery holder
- (1) 62 Ω through-hole resistor
- (1) 5 mm through-hole yellow LED
- Conductive thread
- Iron-on conductive fabric
- Fabric
- Fabric glue

Tools:

- Needle
- Scissors
- Iron
- Ironing board
- Needle-nose pliers







Because conductive fabric conducts both electricity and heat, it will be quite hot after you iron it! Be sure to let it cool for a few minutes before touching it.

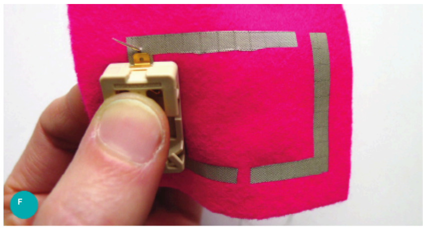
Steps:

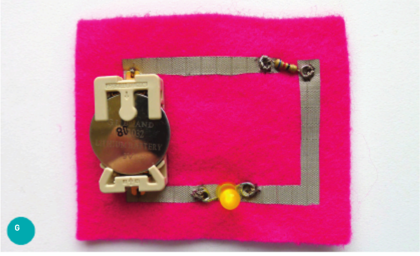
1. Gather the parts for conductive fabric circuit (Figure A).
2. Sketch your circuit design, and cut it out of the conductive fabric. Make sure there are three openings for the battery holder, the resistor, and the LED. Arrange the fabric adhesive side down on the nonconductive fabric. Once your fabric is properly arranged, carefully move it to an ironing board (Figure B).
3. Check the spacing of the components on your conductive fabric by temporarily putting them in place (Figure C).

4. Gently cover the piece to be ironed with a thin piece of cotton or muslin. This will prevent the fabric from getting excessively hot and will protect your iron from getting adhesive gunk on it (Figure D).
5. With the iron set to low heat and the steam turned off, gently iron the circuit until the adhesive melts and the fabric sticks. Lightly press the iron down in different locations rather than sliding it around so as not to shift the conductive fabric (Figure E).
6. Use conductive thread to connect the components to the fabric traces (Figure F). (Remember to use separate pieces of thread for each trace.)
7. Finish the knots in back, and your circuit is complete! Add the battery to light it up (Figure G).

ADVANTAGES AND DISADVANTAGES

The previous section shows how the same circuit can be constructed in six ways with different





conductive materials and building methods. With those techniques in hand, let's look at the advantages and disadvantages of each.

COMPARING CONDUCTIVE MATERIALS FOR CIRCUIT BUILDING

TYPE OF CIRCUIT	ADVANTAGES	DISADVANTAGES
Alligator clip circuit	• Quick • Allows integration of nonstandard components (like DIY sensors) • Plays well with e-textile tool kits; useful for testing components	• Bulky • Not optimal for complex circuits • Unstable • Potential for shorts
Wire circuit	• Extremely flexible • Customizable	• Not practical for complex circuits • Robustness depends on choice of wire and amount of strain placed on circuit
Breadboard circuit	• Quick and efficient • Easy to modify	• Bulky • Not terribly robust • Looks weird on a shirt
Protoboard circuit	• Secure connections • Potentially smaller than a breadboard	• Still a bit bulky • Difficult to integrate with textiles
Conductive-thread circuit	• Flexible • Pliable • Customizable • Fashionable	• Potential for shorts • Some conductive threads have significant resistance • Time-consuming
Conductive-fabric circuit	• Fast(er) approach to soft circuits • Potential for interesting designs	• Subject to shorts • Still requires sewing

Wearable technology is an emerging field. When creating wearable electronics, there is no one right way to build a circuit, and you are not restricted to just one method. Keep these advantages and disadvantages in mind when considering which method to use when creating a circuit.

EXPERIMENT: Wearable Circuits

Now that you know about the various ways to construct a circuit, it is time to play with these techniques in a wearable context. Choose a way in which you'd like to wear this circuit (bracelet, hat, T-shirt), and make it with two types of circuit: one hard and one soft. Once you're done, take your own notes about the strengths and weaknesses of each as well as in what context you could see each being used.

Looking Ahead

This chapter covered what circuits are as well as how to build them using a variety of conductive materials. In the following chapter, you'll develop a more in-depth knowledge of these materials and learn how to decide what to use when.

GALLERY 2:

CIRCUITS IN WEARABLES

Circuits can be built into wearables in wildly different ways. These examples from the field show circuits created through hand and machine sewing, soldering, stapling, cross-stitching, and weaving.

GlowStitch LEDS

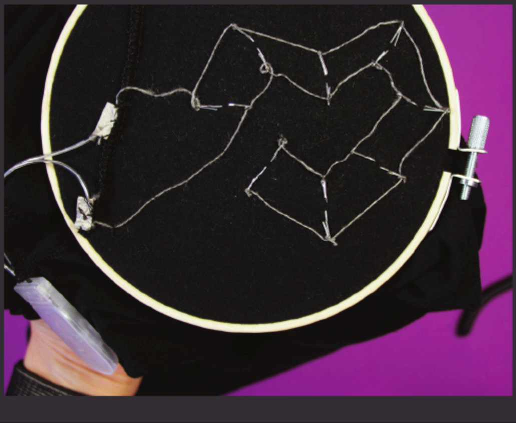
GlowStitch LEDS, by Steph Piper, are chainable micro-LEDs mounted on flexible printed circuit boards that can be included in a machine-sewn circuit.

Bubble Pop Electric

Bubble Pop Electric, by Joanne Jin, uses a machine-embroidery technique to integrate LEDs into a necklace.

Rainbow LED Heart Sewable Circuit

The light-up rainbow LED shirt by Natasha Dzurny is a sewable LED project created as a workshop example. The LEDs are connected using Dzurny's staple method, where the LED leg is folded around the conductive thread.



Sound Impressions

Sound Impressions, by Marlin Velasco, is a performative wearable that uses e-textile sewing to create an experimental listening system. Using a microphone, an amplifier, LEDs, a 555 integrated circuit, and other analog circuitry, Velasco transformed the clothing into a sensitive fabric capable of capturing and generating sound impressions.

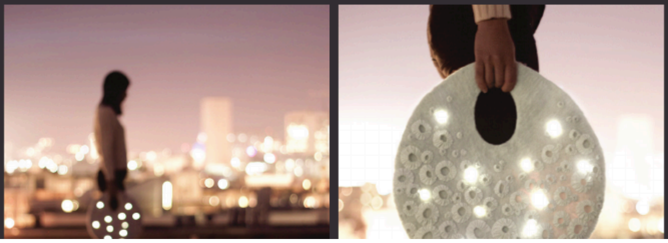
Jeannie Choe

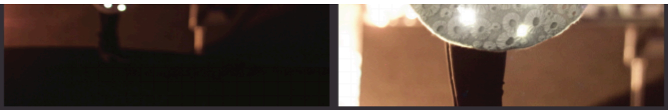
Soft Electric

The Soft Electric, by Grace Kim, reveals the circuit but elegantly incorporates the conductive-thread traces into the aesthetic of the felted cape.

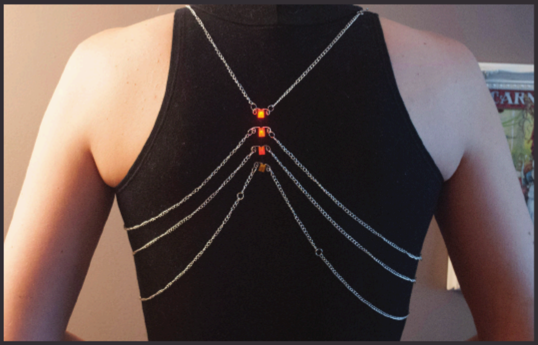
Sessile Handbag

The Sessile Handbag, by Grace Kim, merges technology with natural forms: hand-felted "barnacles" are combined with embroidered LEDs.



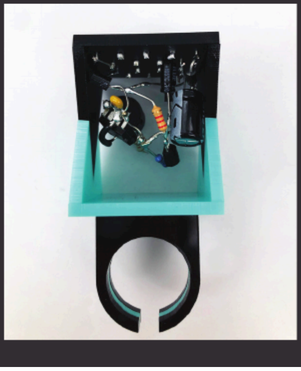


Jeannie Choe



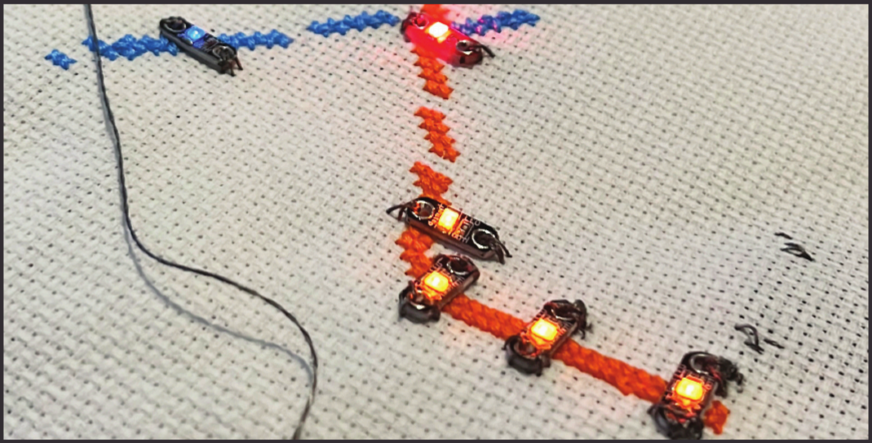
Connection and Motion

Connection and Motion, by Izzie Colpitts-Campbell, is a wearable circuit that uses stainless-steel-coated brass chain to connect LEDs to a battery pack.



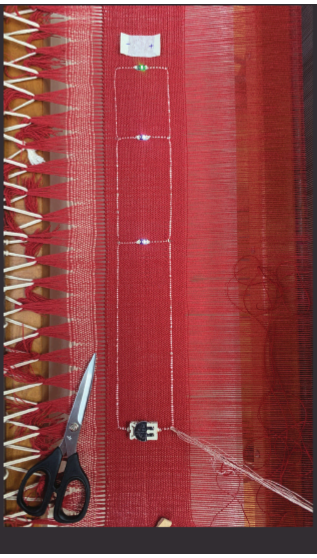
Singing Solar Ring

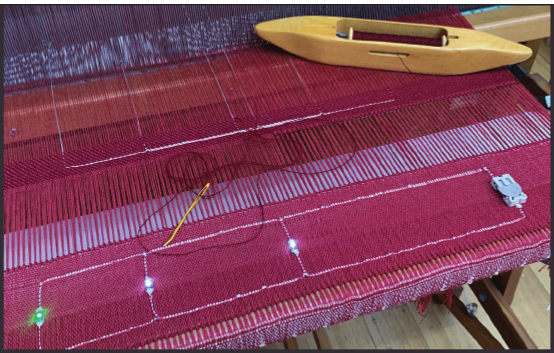
Singing Solar Ring, by Daniel Ryan Johnston, is a laser-cut ring featuring 14 BWP34 solar cells, a voltage detector, a piezo buzzer, and a capacitor to manage power. Once the capacitor reaches a certain voltage, the buzzer sounds, discharging the power that has been collected. The circuit then recharges and the cycle continues.



Stitching Accessibility: An Illuminating Metro Map Embroidery

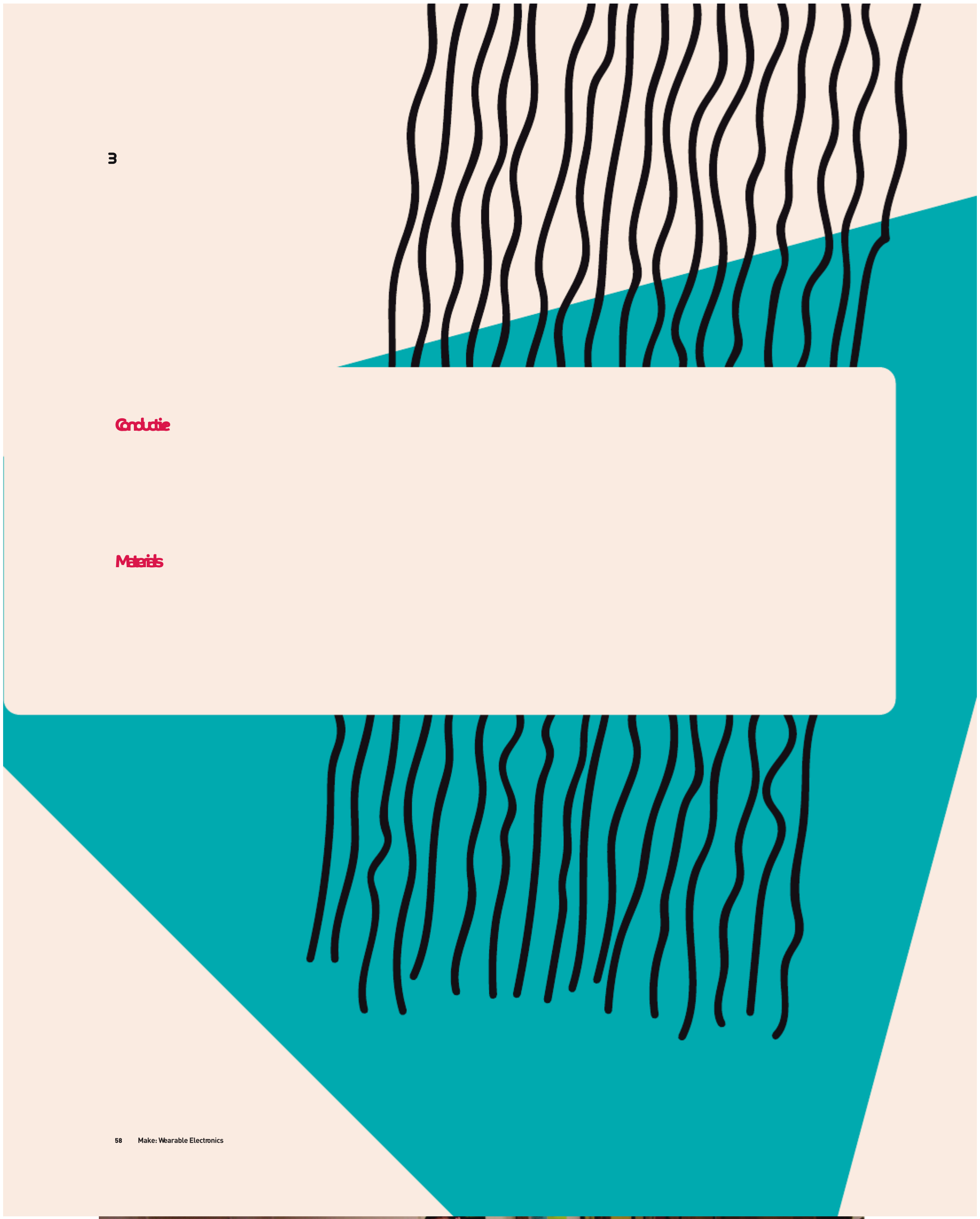
Stitching Accessibility: An Illuminating Metro Map Embroidery, by Hanine El Mir, is a cross-stitched rendering of the STM (Société de transport de Montréal) transit map. Using conductive thread, sewable LEDs are illuminated to indicate which metro stations have elevators.





Weaving Circuits

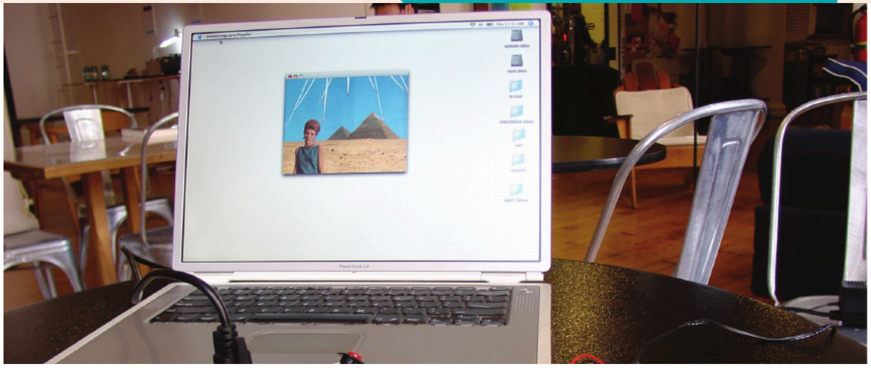
Weaving Circuits, by Victoria Manganiello of Craftwork, cofounded with Nicole Yi Messier, is a parallel circuit woven on a floor loom using sewable LEDs and Karl Grimm silver thread woven into the cloth as supplemental warp and wefts thread. This swatch explores what kinds of circuits are possible with the woven structure and the various challenges and processes necessary for building e-textiles on the loom.

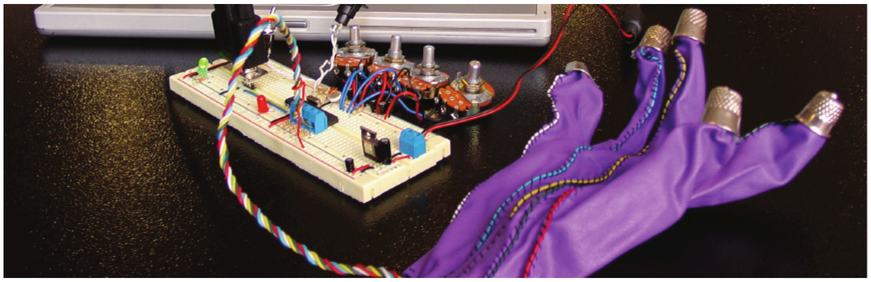


3

Conductie

Materials





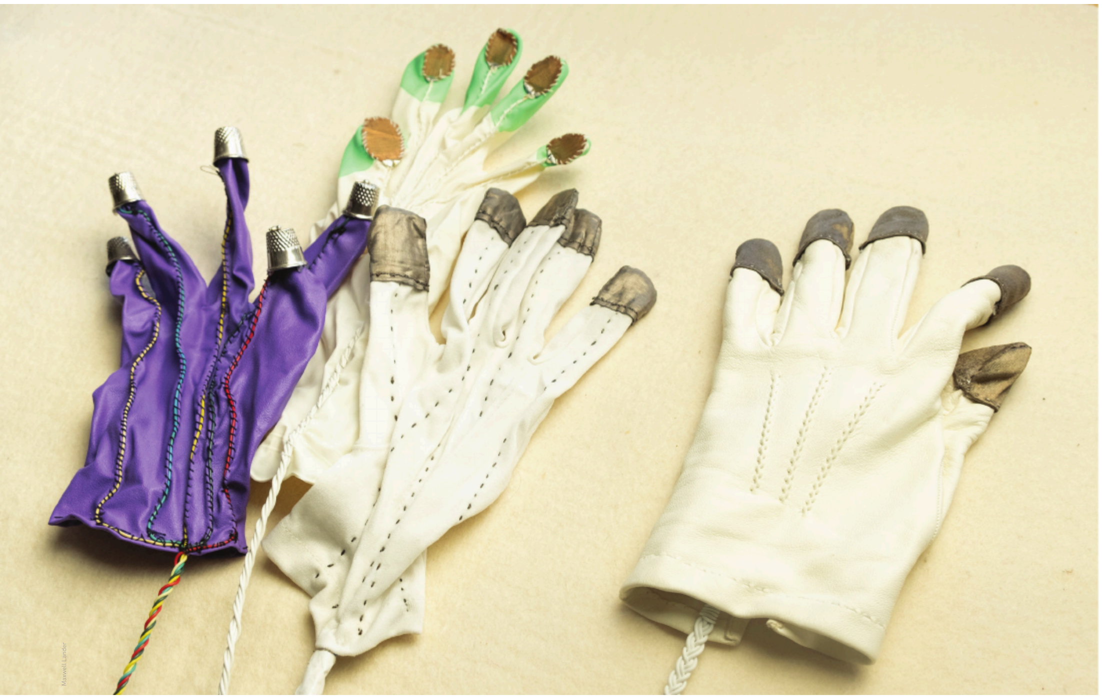
Go-Go Gloves V1, made with dishwashing gloves, solid-core wire, and sewing thimbles

My first wearable-electronics project was a pair of gloves that were used to control the moves of go-go dancers on a screen.

The first prototype consisted of dishwashing gloves, solid-core wire, and a metal thimble for each fingertip, using cotton thread to sew the wires to the glove. When preparing the project for an end-of-term show at the Interactive Telecommunications Program at New York University, I made multiples of this design in hopes of being able to accommodate the variety of hand sizes I expected in attendance, from the small hands of small children to the large hands of adults. I also switched from thimbles to copper mesh for the fingertips.

While the range of sizes worked well, I learned a lot about my material choices. Wear and tear caused the gloves to become sweaty, sticky, and difficult to take on and off. Many of the solid-core wire connections broke after repeat bending along the finger joints. And the copper mesh on the fingertips sometimes poked through the glove material, causing discomfort for the wearer.

Adobe Stock-Lepusinensis



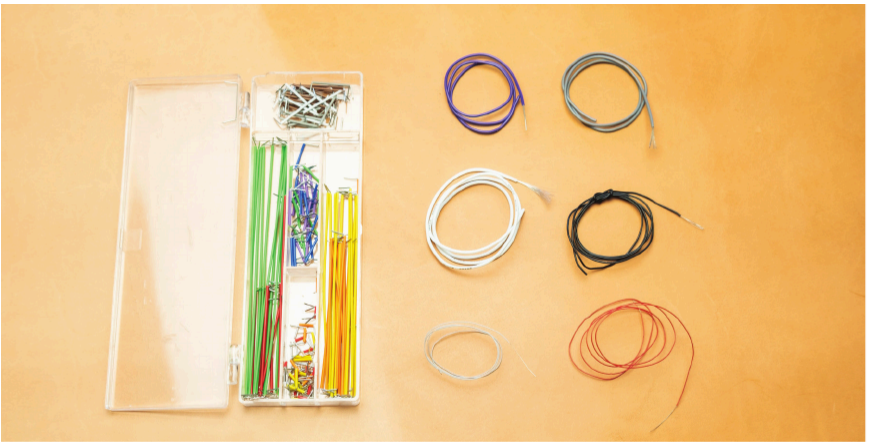
Maxwell Lander

It was around this time that someone in my community introduced me to

Versions 1 through 4 of the Go-Go Gloves

conductive thread and conductive fabric. Still being new to electronics, I had never heard of these materials, and at the time, they were not readily available. However, I was referred to one of the few places conductive thread was available to purchase in small quantities: Lamé Lifesaver, in Victoria, British Columbia. The owner had started the business to make conductive thread more accessible for fencers—those who participate in the sport of sword fighting—who wanted to repair their lamé (jackets worn in competition that sense when an opponent has scored). So I placed an order and waited for my thread to travel from western Canada to New York City. I was also able to source a small amount of conductive fabric.

With these new materials, I was able to create a third prototype of the gloves that were soft, breathable, flexible, and more robust. But these materials also presented their own challenges and limitations. Moving forward, I started to think more carefully about the materials with which I chose to build circuits.



Maxwell Lander

TYPES OF WIRE: *box of hookup wire and (left to right, top to bottom) solid-core wire, stranded wire, 22 AWG silicone wire, 30 AWG silicone wire, and white and red wrap wire*

In the previous chapter, you learned how to assemble circuits using a variety of conductive materials and tools. In this chapter, you will get to know these materials in more detail. You'll also learn about criteria to consider when choosing among them for a wearable-electronics project. Let's start with the conductive materials and tools conventionally used for creating circuits and work our way toward less conventional approaches from there.

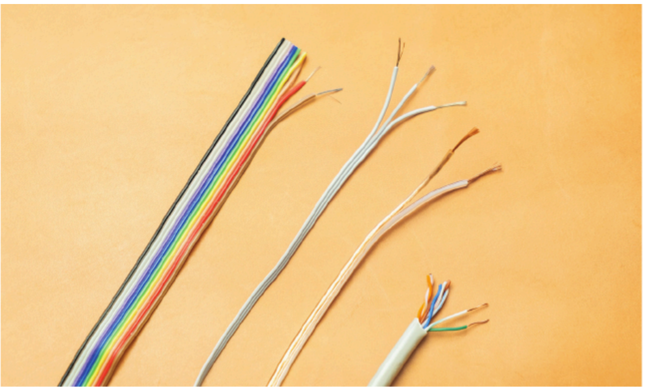
Traditional Conductive Materials

WIRE

Wire is a standard material for making connections within a circuit. Influenced by the excitement of working with new materials, beginners often feel there is an expectation to create wearable-electronic circuits using conductive thread, and may overlook wire. However, conductive thread is not always required—or desirable. Sometimes, wire does exactly what you need it to do, particularly if you're working with the right type. (Note that the unit of measurement for a wire's diameter is called its *gauge*.) Below are descriptions of useful varieties of wire:

Solid-core wire

This consists of a single wire encased in insulation. 22 AWG hookup wire, perfect for prototyping circuits on a breadboard, is sold in spools and in jumper wire sets—precut lengths with stripped ends for easy use. Due to its stiffness, solid-core wire is very good for plugging into things. But because it is stiff, it tends to snap if flexed repeatedly. This makes it a poor choice for wearable projects.



Maxwell Lander

Ribbon cables, speaker cable, telephone cable

Wrap wire

This solid-core wire is thin (30 AWG), relatively flexible, and lightweight and is in some instances suitable for certain wearable applications. It can be discreetly added along the seam of a garment without adding too much extra bulk. But wrap wire is delicate and can break if strain is applied.

Stranded wire

This wire contains multiple conductive strands within an insulating plastic sleeve, which causes it to be very flexible and forgiving. If one strand breaks, the other strands maintain the connection, so it is not likely to cause an interruption in the flow of electricity in a circuit. Because it accommodates repeated bending, stranded wire is a good option if you need to use wire around joints like the elbow or the knee.

Silicone wire

Stranded wire insulated with silicone is particularly flexible, pliable, and floppy, and comes in a variety of gauges. It is well suited for wearables projects.

CABLES

In designing wearable electronics, simplifying circuitry and connections can reduce bulk and aid with troubleshooting. Having several loose wires running along the same path can be uncomfortable and may lead to additional wear and tear.

A cable is a set of insulated wires contained within a casing, jacket, or sheath. Many types of cable can be bought by the spool or repurposed from an e-waste pile or junk shelf.

Ribbon cable

This flat, flexible, and lightweight cable comes with anywhere between four and eighty insulated wires running side by side on a flat plane. You can peel away a select number of conductors you need as a separate chunk.

Speaker cable

This is extremely flexible and widely available, and a simple and clean way to run two connections over a longer distance. The different wires are electrically the same but are marked for polarity, which can be useful when assembling a circuit.

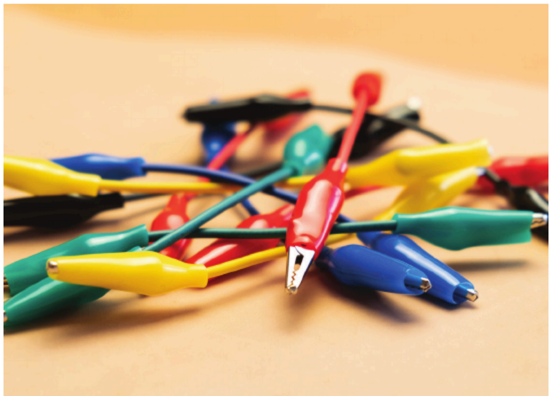
Multiconductor cable

This is a catch-all term for multiple insulated stranded wires collected in a shared jacket. This can include telephone, ethernet, and other common household cables. Though a bit bulkier, these types of cables can offer a spiffy multiconductor solution in a pinch. Just snip off the connector, and you can access the bundle of wires within.

ALLIGATOR CLIPS

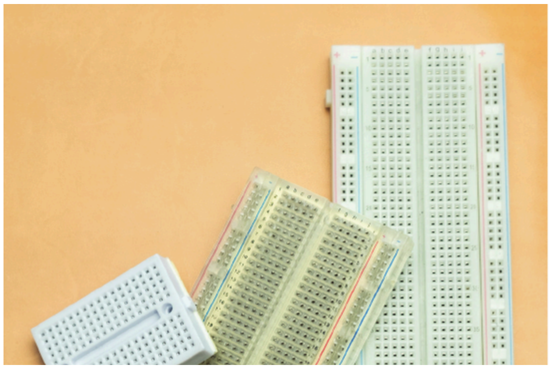
Alligator (or crocodile) clips are a type of prototyping tool that consists of a simple insulated wire with a spring-loaded jaw on either end. This means you can quickly and easily clip together a temporary circuit. Alligator clips are often used to prototype connections for a circuit before you make the final commitment of sewing it together with conductive thread. They are also useful when making temporary connections to a part you are not yet ready to solder.

Alligator clips come in a variety of colors. The standard length allows for flexibility when testing circuits with multiple components.



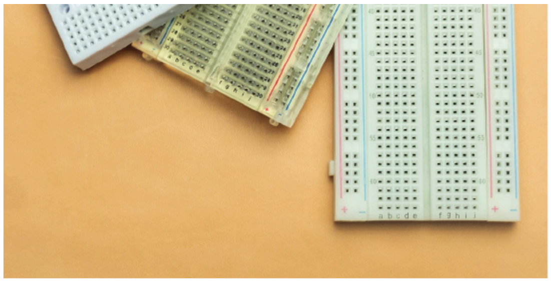
Shorter alligator clips are more compact and can be easier to manage when testing simple circuits.

Maxwell Lander

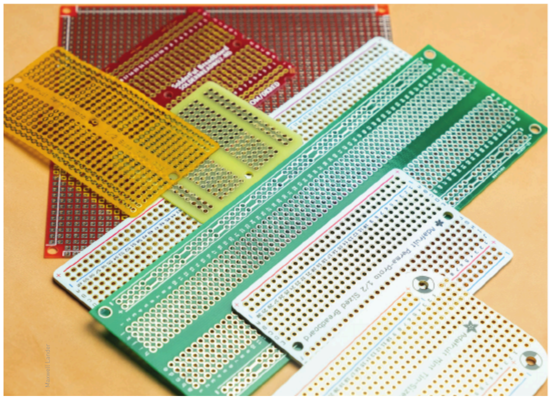


BREADBOARDS

Breadboards are prototyping tools that allow for quick connections to be made by simply plugging wires into holes. Inside the breadboard, the lower layer contains *buses*, or lengths of a conductor that connect multiple holes. On each long side are two buses intended for power (+) and ground (–). Perpendicular to those, in the middle, are rows interrupted by the middle ridge that are



Breadboards: mini, half, and full



A selection of protoboards: snappable with no connections between holes, as well as many variations of breadboard-style protoboards

intended as the canvas on which the majority of the circuit is constructed.

Breadboards offer a fast way to test your circuits but not a robust way to wear them. Protoboard is often a good way to create a more secure version of a circuit prototyped on a breadboard.

PROTOTYPING BOARD

A prototyping board, or protoboard, is a type of circuit board that can serve as a base for more permanent circuits. It contains regularly spaced holes that are lined or ringed with a conductive material such as copper. These conductive areas are called pads. Less expensive protoboards have pads on only one side. Higher-quality protoboards feature through-plated holes, which means the holes are lined with conductive materials, and there are pads on both sides of the boards. To optimize the reliability of your circuit and reduce your frustration, work with a higher-quality protoboard whenever possible.

Components are soldered to the protoboard, which makes for stable and sturdy connections far more secure than those created on a breadboard. Protoboards can also be cut to size, making them easier to fit into small places.

Electrical connections between components are made using solder, component legs, and jumper wires and at times through connections included in the design of the protoboard itself.

Basic protoboards have no connections between the holes, but more sophisticated ones have strips of holes connected by conductive traces, and on others, all the traces are connected. When this is the case, you must cut connections (rather than create them) using a utility knife.

Breadboard-style protoboards, such as Adafruit's Perma-Proto, mimic the layout of connections found on a breadboard. If your circuit prototyping starts on a breadboard, I strongly encourage using a breadboard-layout circuit board. It enables you to copy the same layout for your circuit connections and reduces chances for error.

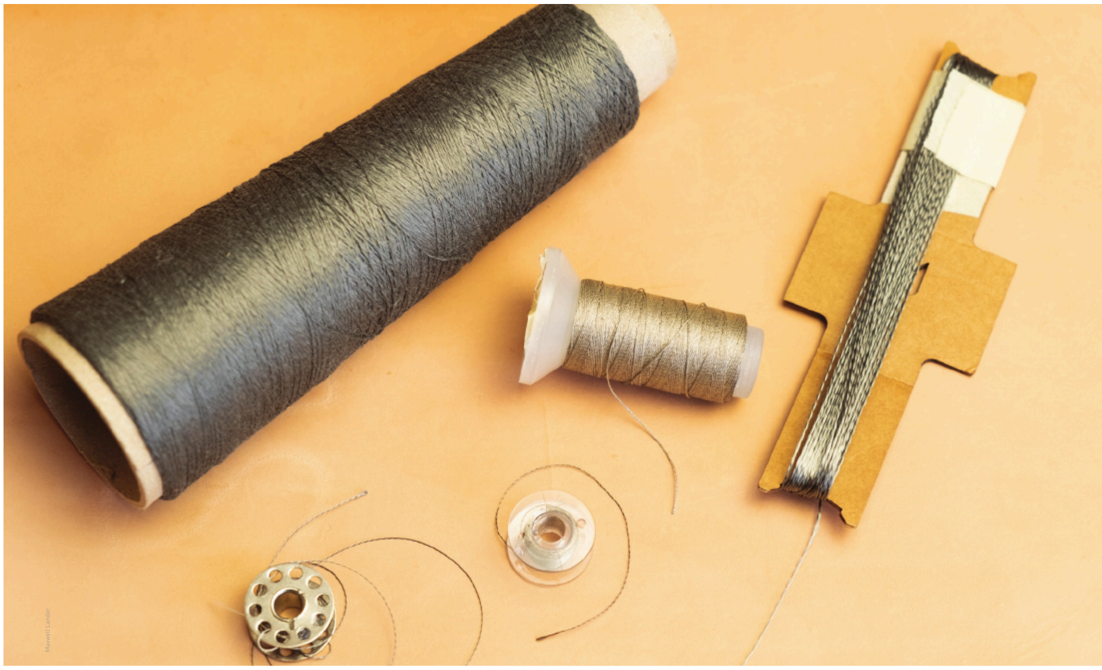
Conductive Thread

With traditional conductors under your belt, you can now explore less conventional conductors! Let's take a look at some softer options.

Conductive thread, which contains conductive metals such as silver or stainless steel, has been widely adopted by makers and artists to make soft electrical connections.

However, conductive thread is a tricky material. When used in the appropriate context with sufficient skill, it can create supple, subtle, and visually stunning circuits. But it also has the potential to be gnarly, knotty, and limiting. I've known it to bring many students under the pressure of an impending deadline to tears and projects to self-destruction by way of a seam ripper. Knowing the properties of your materials and how they align with use cases can save you a whole lot of heartache!

NOTE: If you are new to sewing, take the time to do some stitching with standard cotton thread before moving on to conductive thread. Because of the metallic content, some conductive threads are a bit more difficult to work with. They can be bulkier, quicker to tangle, and slippery, which means knots will sometimes untie.



Maxwell Lander

Conductive thread can be purchased on cones, bobbins, and spools.

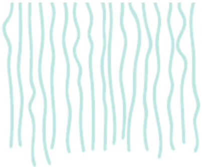
There is an extensive array of conductive threads, most of which are sold in large quantities for industrial purposes. The Comparing Conductive Threads table on page 69 includes a selection of threads more readily available in smaller quantities.

PROPERTIES OF CONDUCTIVE THREAD

Below is a list of the properties you want to consider when choosing a conductive thread.

Thickness

Two-ply? Four-ply? Others? The thickness of the thread affects how easy it



is to sew with and determines which type of needle you'll need and whether you'll be able to use it in a sewing machine. Also, the more plies, the less resistive the thread will be. For instance, four-ply conductive thread is more conductive than the two-ply version of the same material.

Resistance

Resistance is a material's ability to resist the flow of electricity. This is one of the most significant factors to consider when choosing a conductive thread. Some threads have a relatively high resistance, which affects how they can be used in a circuit and which components they can be used with. For instance, motors need lots of current, but conductive thread can deliver only a limited amount, so they are not an ideal match.

Materials

Different threads contain different conductive materials, and as a result, these threads do not all have the same properties. For instance, stainless steel thread is highly conductive and resistant to corrosion, whereas silver-plated nylon thread has a higher resistance but is much softer and more pliable.

Hairiness

While some threads are quite smooth, others have an element of hairiness, which can cause short circuits. This quality should be considered when determining the spacing of lines of conductive thread.

Color

There aren't any choices in this category at the moment, but there should be. Why would the e-embroiderers of the world want palettes limited to a monochrome silver? Let's hope some vibrantly colored conductive threads will be spinning round your bobbins soon.

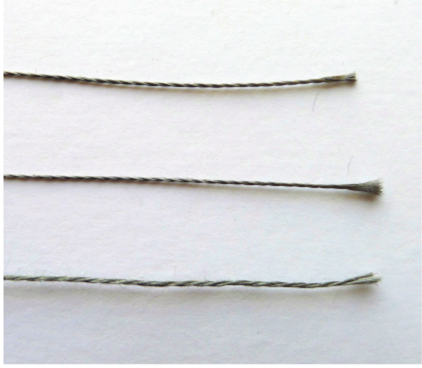
Insulation

Insulation is useful for preventing short circuits. Insulated conductive thread exists but is generally not available in small quantities.

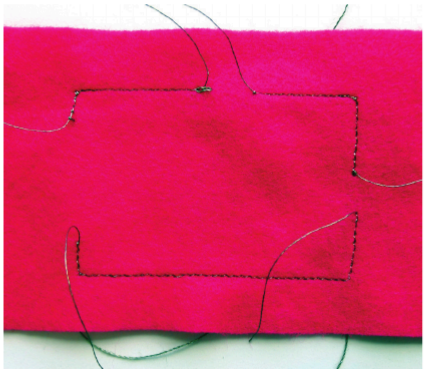
WORKING WITH CONDUCTIVE THREAD

Sewing with conductive thread can be a bit more challenging than regular sewing. The best way to get to know a thread and its quirks is to do some tests before you get started.

With hand sewing, keep in mind that you may need a needle with a larger eye. Also, conductive thread can be slippery, so it is advisable to tie your knots well and at times even reinforce them with a bit of fabric glue.



Conductive threads have differing thickness and material properties, including how hairy, or fuzzy, they are.



Conductive-thread traces created with a sewing machine

With a sewing machine, a good rule of thumb is to use the conductive thread in the bobbin rather than for the top stitch. Some two-ply threads will run through the needle OK, but it depends on the thread and on your machine. Industrial or hardier home sewing machines seem to handle conductive thread better, but it's smart to keep spare needles on hand when trying out new threads in case you end up breaking a needle in the process.

> **NOTE:** Electronic components cannot be easily sewn with a domestic sewing machine. This means that even if you machine-stitch your conductive thread traces, you will need to leave long tails of thread on either end so that you can hand-stitch the connection to the electronic component.

One of the biggest challenges in working with conductive thread is preventing short circuits. Because conductive thread is uninsulated, there are many opportunities for parts of the circuit to touch when they aren't intended to. For instance, if you fold a sewn circuit in half, there's a good chance traces will touch each other temporarily or permanently, preventing its operation. Similarly, if you have long tails on the back of a piece of conductive thread embroidery, they may move around and come into contact. Keep this in mind when sewing with conductive thread: Neat, trim, and well-organized circuits will make you much happier in the long run.

TYPES OF CONDUCTIVE THREAD

The table at the top of the next page compares some conductive threads available in small quantities through electronics-supply companies.

COMPARING CONDUCTIVE THREADS

PRODUCT NAME SOURCE	PART #	PLY # RESISTANCE		(Ω/FT)	MATERIAL NOTES	
Stainless thin conductive thread	Adafruit 640	2	16		stainless steel	stiff
Stainless medium conductive thread	Adafruit 641	3	10		stainless steel	stiff
Stainless thin conductive yarn/thick	Adafruit 603	3	12		stainless steel	furry

conductive thread						
Electro-Fashion conductive thread	Kitronik	2722, 2724, 2727, 2744	?	12	silver	less resistant to oxidization
Conductive thread bobbin—12 m (smooth, stainless steel)	SparkFun	DEV-13814	?	9	stainless steel	smooth
Conductive thread bobbin—30 ft. (stainless steel)	SparkFun	DEV-10867	?	28	stainless steel	hairy
Conductive thread (60 g)	SparkFun	DEV-11791	?	28	stainless steel	hairy

Conductive Fabric

Conductive fabric is a wondrous material. Whereas conductive thread can present issues with resistance, many conductive fabrics do a better job in circuits with higher-current demands. And because this fabric comes in a large sheet, there is more room to play with the visual design of the circuit.

PROPERTIES OF CONDUCTIVE FABRIC

The considerations when working with conductive fabric are slightly different than those of conductive thread.

Type

What type of fabric is it? Woven? Ripstop? Knit? Plated? Does it fray or wrinkle? Can it handle the conditions of your intended use?

Stretch

The stretchiness of a fabric relates to its type but is worth special mention. Bodies are bendy. A stretchy fabric can be particularly helpful for parts of a garment that need to shape-shift or bend frequently. Does the fabric stretch at all? If so, in one direction, or two?

Silver and nickel conductive fabrics

Copper conductive fabrics—ripstop and regular

Substrate

Conductive fabric is generally composed of several layers. What is the base, nonconductive layer? Is it nylon, polyester, or something else? This will ultimately affect the care and comfort of the garment you are creating.

Plating

The plating of the fabric is the conductive part. As with conductive thread, it's worth considering what the metallic content is and how that affects its performance and longevity.

Weight and thickness

Is this fabric thick, thin, heavy, or light? This is important to consider in the context of what you'll be making. If you're working to create something lightweight like a T-shirt, it might not make sense to use a conductive fabric that is thick and heavy.

Surface resistance

While many conductive fabrics have an extremely low surface resistance, there are some exceptions. Be sure to check this before committing to a type of conductive fabric.

Color

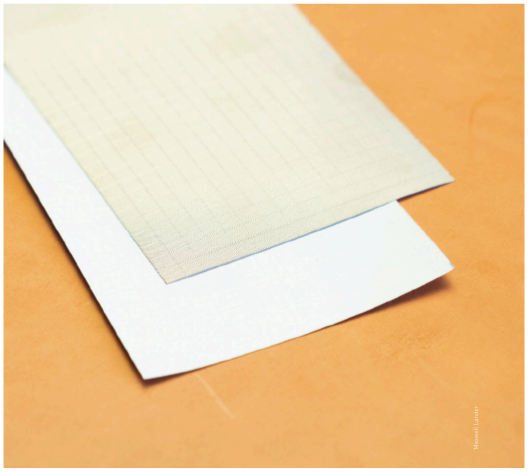
The majority of conductive fabrics types are silver in color, but there are many copper-colored conductive fabrics as well, and a few that are other colors. If another color is better suited for your project, be sure to do some research to see what else is available.

WORKING WITH CONDUCTIVE FABRIC

There are two ways to incorporate conductive fabric: sewing and the use of iron-on adhesive.

Conductive fabric can be sewn just like any other fabric—it's just a question of how you'd like to work it into a circuit. Quilting and appliqué techniques can be used when sewing conductive fabric onto a nonconductive substrate.

The important thing to consider when sewing conductive fabric is that many conductive fabrics tend to fray, and threads from frayed pieces can wander and inadvertently create shorts. This can be mitigated through hemming, serging, or other traditional sewing techniques. You can also aim to work with fabrics that fray less, such as ripstops or nonwovens.



Maxwell Lander

ShieldIt Super Fabric with iron-on adhesive

Another approach is to use iron-on adhesive. Conductive fabric can be purchased with iron-on adhesive already applied (e.g., ShieldIt Super Fabric from LessEMF) or you can apply your own (e.g., HeatnBond or others, available at most fabric stores). Keep in mind that iron-on adhesive is not conductive, so connections between separate pieces need to be bridged with conductive thread.

The advantages of this approach are that it makes it extremely easy to cut out traces for a circuit and it prevents fraying. Conductive fabric can be cut using scissors, a knife, or even a laser cutter. (Laser cutting is a very effective way to create detailed and intricate designs.) With this approach, once circuit traces are cut, they can simply be ironed on!

TYPES OF CONDUCTIVE FABRIC

NAME	SOURCE	PART #	TYPE	STRETCH	COMPOSITION	SURFACE RESISTANCE
Woven conductive fabric	Adafruit	1168	Woven	None	Copper-and-nickel-plated polyester	< 1 Ω/ft.
Knit conductive fabric	Adafruit	1167	Knit	Two-direction	83% nylon, 17% silver	< 1 Ω/ft.
Knit jersey conductive fabric	Adafruit	1364	Knit	Two-direction	63% cotton, 35% silver yarn, and 2% spandex	46 Ω/ft. in stretchier direction, 460 Ω/ft. in less stretchy direction
ShieldIt Super Fabric	LessEMF	11220	Ripstop	None	Copper-and-nickel-plated polyester with nonconductive hot-melt adhesive	< 1 Ω/ft. (one side only)

Ripstop conductive fabric does not stretch and is very easy to sew.

Stretch conductive fabric is slightly more difficult to work with but works well with bendy parts of the body.

Maxwell Lander

Other Conductive Materials

There are many other conductive materials to work with beyond threads and fabrics. Some are more exotic and harder to get, while others are available at your local fabric or hardware store.

CONDUCTIVE-FABRIC TAPE

Conductive-fabric tape comes on a roll with a peel-off backing. What's neat is that the adhesive is often conductive as well—but be sure to test this! If so, you can place one piece of conductive fabric tape on top of another to create a solid electrical connection.

CONDUCTIVE FIBER

Conductive fiber, which can be both felted and spun, is a soft and lightweight conductive material that can be used in a variety of projects. When conductive fiber is felted with nonconductive wool, it can be used to create a tactically pleasing variable resistor or electronic switch!

CONDUCTIVE YARN

Conductive yarn is like conductive thread, except fluffier and a little harder to control. It is excellent for knitting or weaving conductive patches into textiles as well as for creating knitted stretch and pressure sensors.

CONDUCTIVE HOOK AND LOOP

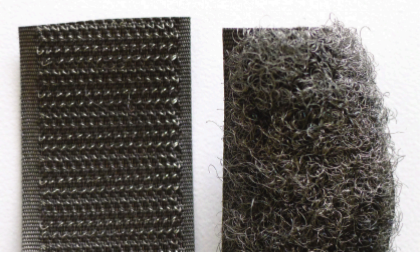
Conductive hook and loop is like a conductive version of Velcro. What's brilliant about it is that it can act as a secure and sensible electronic switch for clothing.

Conductive-fabric tape roll

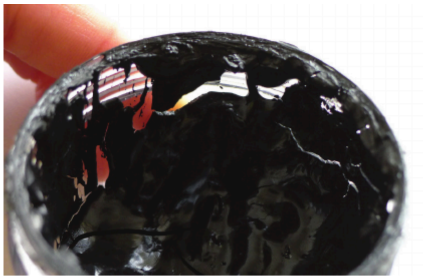
Conductive fiber

Maxwell Lander

Conductive yarn



Conductive hook and loop



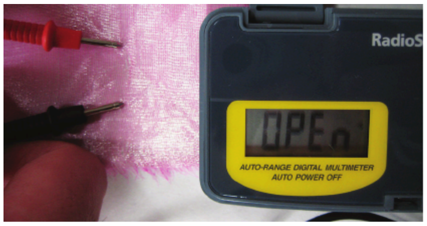
Bare Conductive's Electric Paint

CONDUCTIVE PAINT

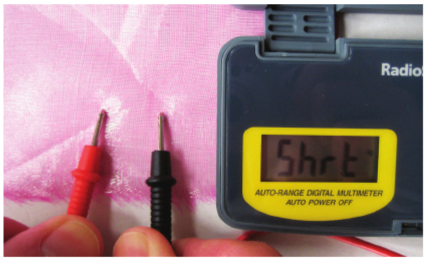
Conductive paints such as Bare Conductive's Electric Paint or LessEMF's CuPro-Cote paint can be used to paint, draw, or silk-screen circuits. It tends to work best on a nonporous substrate.

EVERYDAY STUFF

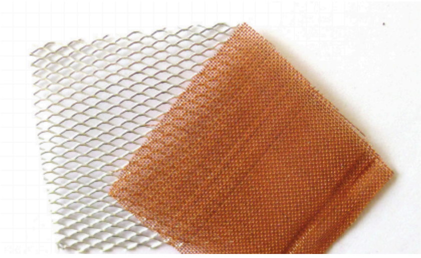
Many commonly available conductive materials



Organza (nonconductive weave)



Organza (conductive weave)



Armature meshes are available at most art supply stores.

can be repurposed in circuits. When shopping for unconventional conductive materials, however, it is helpful to bring a multimeter along so you can test for conductivity. For instance, some organzas have metallic thread that runs in one direction of the weave. A multimeter can be used to test which direction. Sew a line of thread perpendicular to that, and boom! You've got cheap conductive fabric, available from your local fabric store.

There are also nice conductive materials available in the sculpture section of your local art store. For example, some malleable meshes intended for sculpture are also conductive. These are handy because they can be shaped, cut to size, and soldered.

Keep your eyes open when you're out in the world. There are more interesting conductive materials than you'd think!

Selecting Conductive Materials

Now that you are familiar with a variety of conductive materials, the questions will be what to use, and when. Along the way, advantages and disadvantages have been highlighted for each method. Let's review final considerations for selecting conductive materials for your project as well as how to go about sourcing them.

CONSIDERATIONS

Here are some aspects to consider when choosing between types of conductive materials as well as between specific materials within a type.

Ease of use

It is important to consider your own abilities. Which method is most comfortable for you? What do you prototype fastest with? If you've racked up years of experience with a breadboard, it may be more comfortable to do a first prototype that way than to start by whipping out the conductive thread. On the other hand, if you're an expert seamstress, sewing up a quick circuit may seem far less daunting than circuit boards and wires. What is most comfortable for you will probably work best, at least for the first prototype.

Cost

Cost should be considered in relation to what it is you're trying to make. Often, with a prototype, the inclination is to work with what's cheapest and most expendable. But there are times when prototypes turn into showpieces and the choice of materials matters more. You may also be telling a story with your choice of materials, imagining a future time and scenario where they are more ubiquitous and less expensive. On the other hand, you may be working on a prototype for something that is coming to market where the cost of every component and material is crucial. Or perhaps you're just teaching an underfunded workshop where you need to be clever but thrifty with your choice of materials. No matter what, cost is always a factor worth considering.

Insulation

Most wires are insulated. Nearly all soft conductive materials are uninsulated. Will this work for your design? Or are there insulation strategies that could meet your needs? We'll revisit this in future chapters, but it's worth keeping in mind from the get-go.

Resistance

Resistance is a consideration more likely to come up in the realm of softer materials, particularly conductive thread. Low-powered LEDs and thread play nicely together; motors and thread do not. Think about how much current your components require and whether the resistance of your materials will cause any problems in the delivery of the current they need.

Flexibility

Large, inflexible, pointy circuit boards and crevices and curves of the body don't often work well together. What flexibility and form does your circuit need to take?

SOURCING

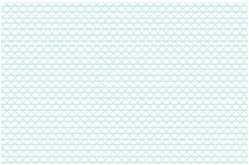
Sourcing the materials discussed in this chapter can be tricky. Although it's always fun to be on the lookout for them, less conventional conductive materials are not typically available at your neighborhood fabric, craft, or hardware store. If you are lucky enough to have a brick-and-mortar electronics store near you (the kind that sells Arduinos), you can check in with them to see whether they've got the e-textile bug and, if not, perhaps encourage them to stock some conductive fabric or thread.

However, in most cases, you will need to purchase these materials from an online source. Depending on the nature of your wearable-electronics project, you may want to select one or work with multiple types of sources for conductive materials.

Trusted sources are the go-to if you are a beginner in this area, or if you are working on a high-stakes or time-sensitive project. In terms of conductive materials, it makes sense to order these materials from suppliers that are using the materials in the same way you intend to use them (wearable electronics, e-textiles, etc.). For me, trusted electronics suppliers and manufacturers meet some or all of the following criteria. They

- make and/or vend exclusively high-quality parts and materials
- provide datasheets and/or clear and precise technical documentation
- create learning materials for their customers
- have strong community engagement
- work responsibly and respectfully with open-source hardware (and, where applicable, software) principles.

Because I'm based in North America, Adafruit and SparkFun, both located in the United States, are at the top of my trusted-sources list, but this can



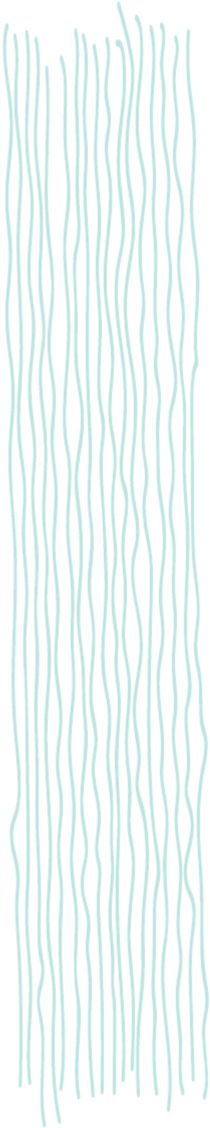
differ globally. When I'm in Canada, I usually order from a global distributor such as DigiKey or Mouser because the shipping is cheaper. This may differ where you are. Do some research and see what makes the most sense in your corner of the world.

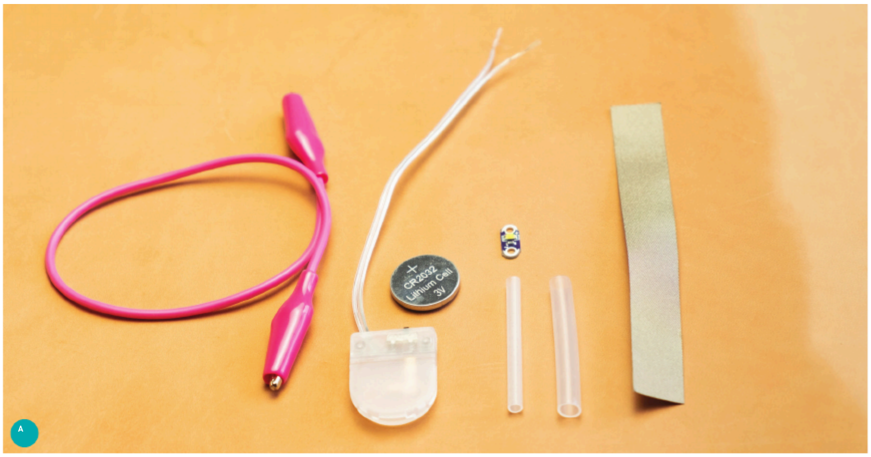
Experimental sources are fun to explore when you have more time to play with new materials. These are suppliers that are lesser known, have less technical documentation, and/or have long or unpredictable shipping times. These days, conductive fabrics and threads are appearing on Amazon, AliExpress, and other online retailers, but often, you need to receive and work with the material to confirm that it is appropriate for your project. These suppliers are best for more exploratory phases of developing a project.

Less conventional sources are suppliers that sell high-quality conductive materials geared toward different markets, and there is even an industry of electromagnetic field (EMF) safety products. LessEMF, located in Albany, New York, sells an extensive array of conductive materials for EMF shielding.

Industrial sources are ones that sell conductive materials in larger quantities. If you find that the materials from the sources above don't meet your needs, keep in mind that there are more options out there—you just need to be a bit more creative in how you look for them. Contact manufacturers for sample requests. Industrial retailers will sometimes sell sample packs, so you can get your hands on a variety to determine which will best suit your project. Similarly, datasheets are available for some of these products and can be consulted when you have questions about their electrical and physical characteristics. When you've identified a material you like, if it's only available in large quantities, coordinate with classmates, people in your hackerspace, or others on the internet to share a bulk order, or suggest to your favorite electronics distributor that it stock the material you want.

Conductive fabrics and threads are manufactured primarily for medical and industrial purposes—such as for their antimicrobial, antistatic, or EMF-shielding properties. The intended applications for these materials provide some indication of why many are not sold in smaller quantities and why aesthetic aspects of the materials have not been well considered. But conductive materials are beginning to appear in more widely available consumer products. Conductive-fabric fingertips are now a common feature found in many gloves for use of touchscreen devices. Similarly, companies that produce heart rate and brain wave monitors use conductive fabric electrodes. As the use of conductive materials becomes more commonplace, hopefully, a wider range of aesthetic options will become available.





A

Maxwell Lander

HOW TO: Conductivity Tester

Multimeters are excellent tools for testing conductive materials, but they are not objects you would typically carry with you. This experiment shows how to build a pocket conductivity tester that is easy to bring to the fabric or store and even to use in your studio.

Parts and materials:

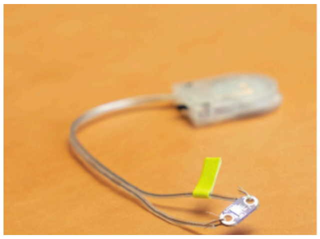
- (1) sewable LED
- (1) alligator clip
- (1) CR2032 battery holder, preferably with on/off switch and wire leads
- (1) CR2032 battery
- Conductive-fabric tape with conductive adhesive
- Clear heat-shrink

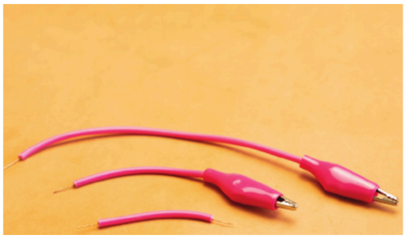
Tools:

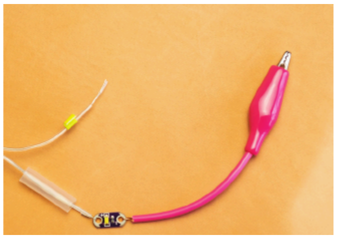
- Wire strippers
- Heat gun

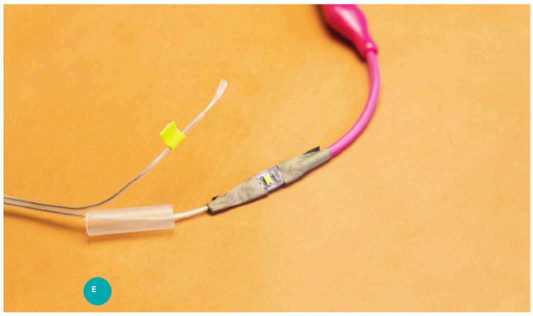
Steps:

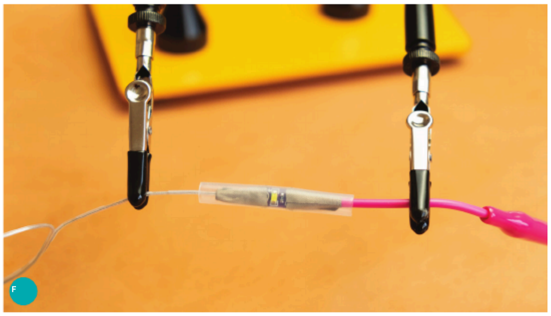
1. Gather your materials (Figure A).
2. Strip the ends of the battery holder's wire leads. Use the sewable LED to identify and mark the negative lead from the battery holder (Figure B).





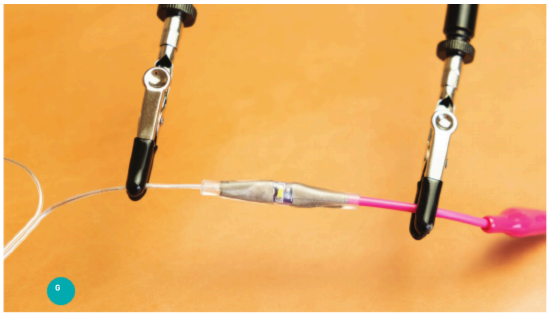


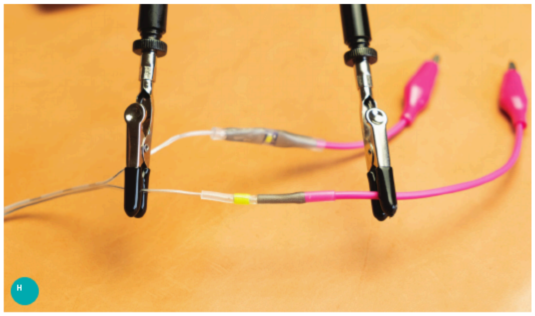


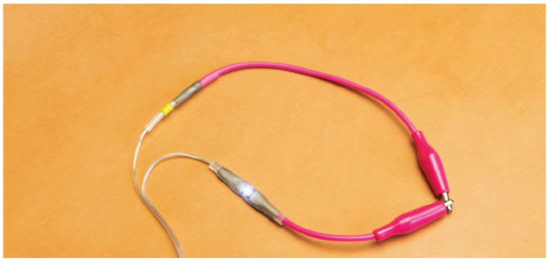


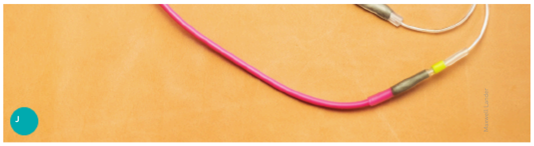
3. Depending on where you want to position the LED, cut the alligator clip in half or in thirds and strip each of the ends. Twist the exposed wires to keep them neat and tidy (Figure C).
4. Slip a piece of heat-shrink large enough to fit over the LED onto the positive lead of the battery holder. Create a hook with the positive lead of the battery holder and hook it around the positive connection on the LED. Hook and connect one half of the alligator clip to the negative side of the LED. Crimp or smoosh the wire hooks to secure them in place (Figure D).
5. Use short pieces of the conductive-fabric tape to keep the hooks in place and improve the electrical connections. Be very careful when placing the tape—it should touch only the exposed wire and sew tab, not other connections on the circuit board (Figure E).
6. Slide the heat-shrink to cover the LED and the tape connections (Figure F).

Maxwell Lander









Maxwell Lander

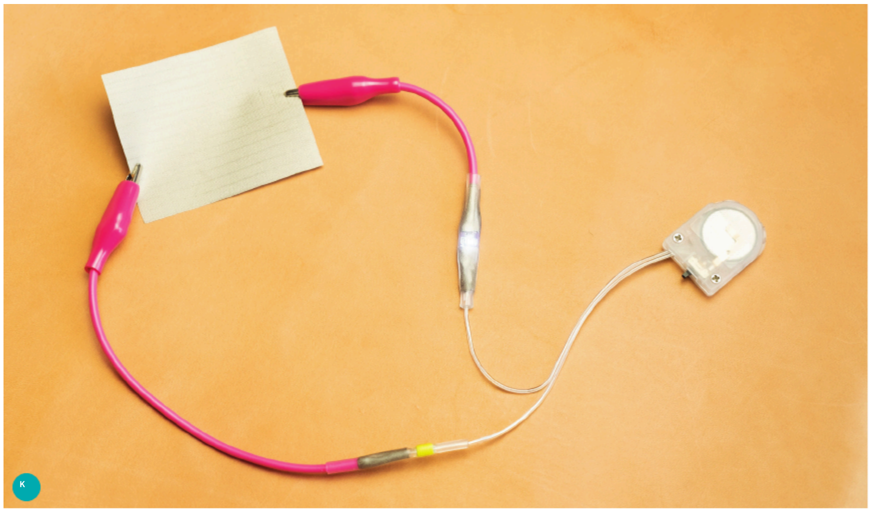
7. Use the heat gun to shrink the tubing until it is snug (Figure G).

8. Repeat this process to connect the other end of the alligator clip to the negative lead of the battery holder (Figure H).

9. Connect the two alligator clips to close the circuit and test your tester. The LED should light (Figure I).

10. Attach both clips to a nonconductive material. The LED should be off (Figure J).

11. Attach both clips to a conductive material. The LED should be on (Figure K).



Maxwell Lander

Troubleshooting tips:

- Check that the LED is connected in the correct direction.
- Make sure the connections contacts are secure.
- Confirm that there in a battery in the battery holder and that it is in the correct orientation.
- If your battery holder has an on/off switch, check that it is in the On position.

From here, use your conductivity tester to test materials and objects to see whether they are conductive!

Looking Ahead

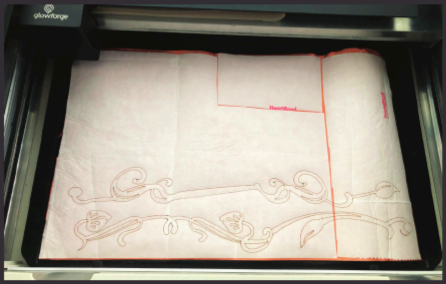
As you can see, many options are available in terms of conductive materials that can be used to build a circuit. In the next chapters, you'll learn about components you can use to create more exciting and complex circuits.

Later, you'll learn more about how to put these components and conductive materials to work when constructing wearable circuits, switches, and sensors.

GALLERY 3:

CONDUCTIVE MATERIALS IN WEARABLES

The results can be quite stunning when using alternative conductive materials in wearables. Sometimes, you want to showcase your circuit rather than hide it. Here are some projects in which conductive fabric, thread, and other materials are incorporated into the visual design of wearable-electronics and e-textile projects.



Conductive Melody

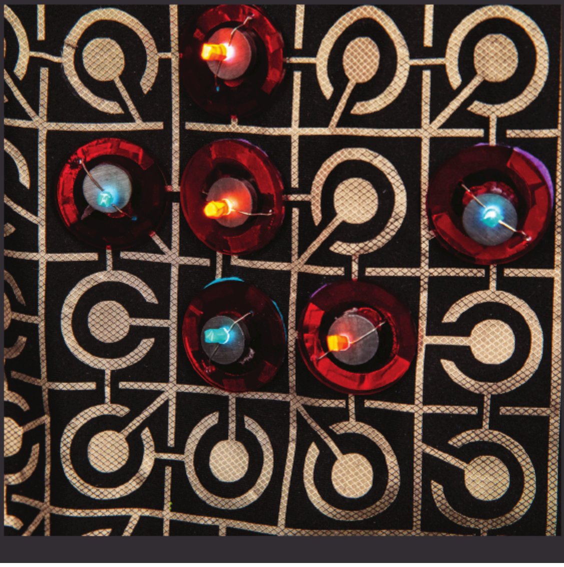
Conductive Melody, by Sahrye Cohen and Hal Rodriguez of Amped Atelier, enables a performer to produce music and light using a sleeve of laser-cut conductive fabric as a capacitive touch interface. Inspired by harp strings, the interface has a vertical organic design. Input from the conductive designs is used by a computer to generate music and is visually represented by the LEDs on the garment. The music can be played from a wearable speaker, a MIDI synthesizer, or a sound system.

IM Blanky

IM Blanky, by Studio (n-1) Architects (Carol Moukheiber and Christos Marcopoulos) and Rodolphe el-Khoury, incorporates conductive fabric into both the visual and the circuit designs.

Embodied RF Ecologies

Embodied RF Ecologies, by Afroditi Psarra, explores the use of an integrated circuit (IC) mixer circuit to make emissions from a National Oceanic and Atmospheric Administration (NOAA) weather satellite audible. The textile antenna is used to detect radio-frequency (RF) transmissions. The circuit traces are made with either conductive thread or fabric wires (handmade using copper taffeta conductive fabric), and the connections are made with snaps in order to be able to be used in a performative context.

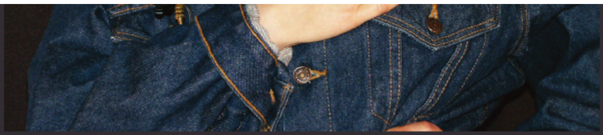


Wearable Tic Tac Toe

Wearable Tic Tac Toe, by Michelle Binyan Xu and Ziye Zhang, is a wearable game board crafted from soft fabric that allows seamless integration with clothing. The circuit base is created with cut conductive fabric and the acrylic game pieces include LEDs and magnets. The rules follow traditional tic-tac-toe; the winner lights up three LEDs of the same color in a row.

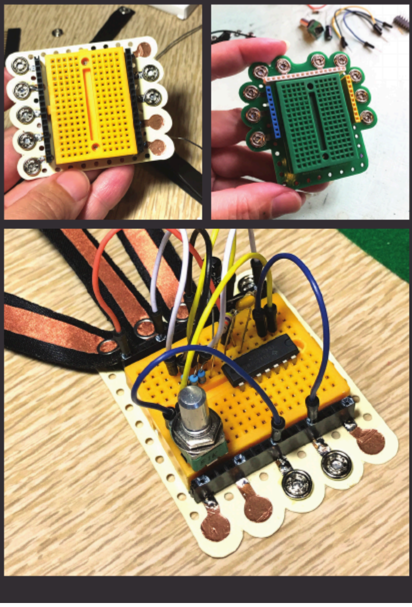
Musical Jacket

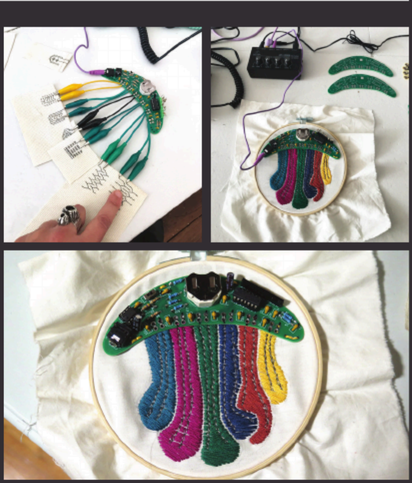
The Musical Jacket, by Maggie Orth, Joshua R. Smith, E. Rehmi Post, Joshua Strickon, and Emily B. Cooper, is an early example of conductive thread in use. It integrates a wearable MIDI synthesizer with an embroidered keypad the wearer can use to play notes and create sounds.



Breadboard Breakout

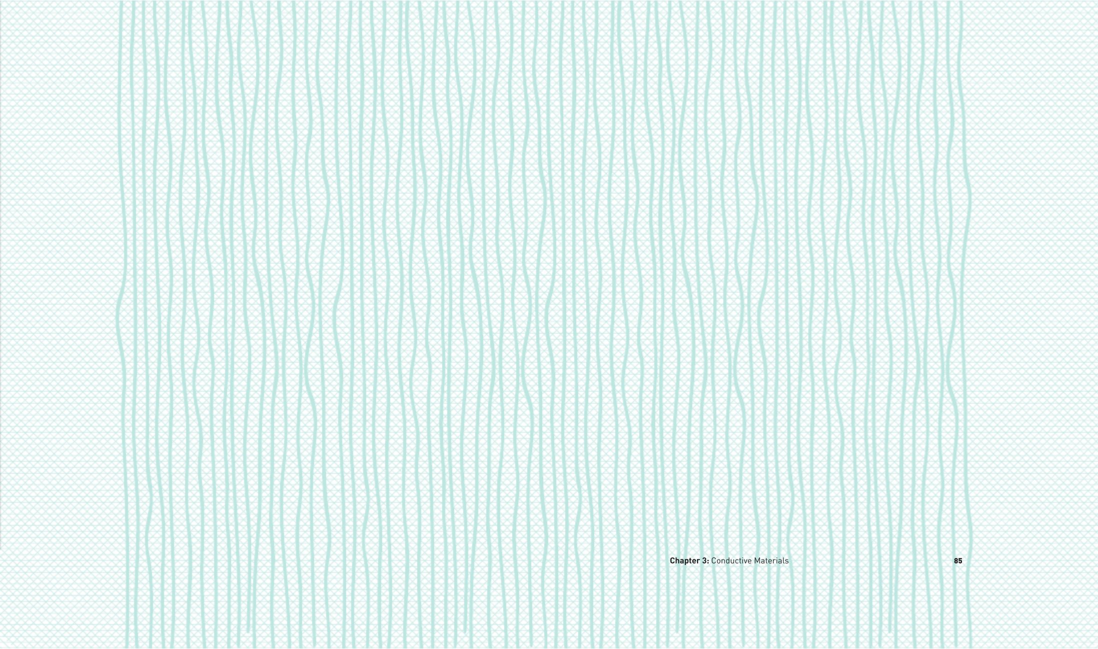
Lara Grant created a prototyping tool intended to integrate a breadboard into e-textile projects. It uses snap fasteners, female headers, and a custom-made printed circuit board.





Stitch Synth

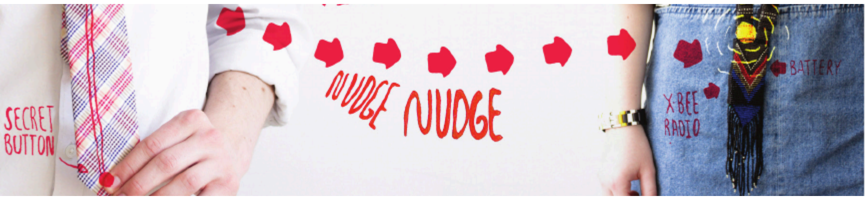
Stitch Synth, by Lara Grant, is a prototype of a kit that translates a well-loved CMOS (complementary metal–oxide–semiconductor) oscillator circuit using a CD40106 chip. It uses all six available oscillators the CD40106 CMOS chip provides and mixes them into one output. When working with this kit, one learns not only how to solder and the function of the circuit but also how to hand-sew touch plates using conductive thread.



4

Switches

BATTERY
X-BEE RADIO
XBEE
SEND A NUDGE TO YOUR PALS!
PRESSING A BUTTON IN ONE NUDGEABLE MAKES THE OTHER VIBRATE
VIBRATING MOTOR
SECRET BUTTON



Nudgeables Accessory Kit, by Social Body Lab (Kate Hartman, Erin Lewis, Jackson McConnell) and Aesthetec Studio

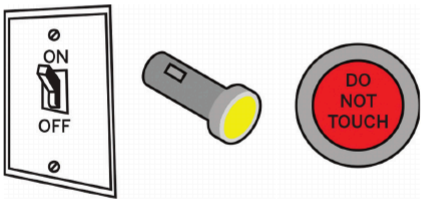
Sometimes we need secret codes to communicate with our friends, partners, or colleagues while in the company of a larger group of people: a nudge, a cough, a scratch of the nose—something that says, "Save me from this conversation," "Let's get out of here," or "I'm thinking of you." What if our clothing could communicate these messages for us?

One of our first projects at Social Body Lab was the Nudgeables Accessory Kit, a tool kit for creating connected garments that can send and receive discreet wireless messages. The kit makes it easy to use simple electronic switches to creatively enable your and your friend/partner/colleague's garments to secretly—but in a tactile way—nudge or notify each other at a distance.

As part of the project, prototypes were commissioned from local fashion, industrial, and jewelry designers who created a variety of bespoke switches. Standard switches were sneakily embedded in ties, necklaces, and bracelets. Conductive materials such as threads and yarns were used to create subtle textile switches integrated into socks, scarves, and headbands.

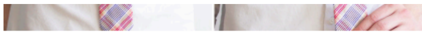
Adobe Stock-Lepusinensis

This T-shirt-and-necklace pair, designed by Jackson McConnell, uses conductive fabric on the back of a pendant to bridge the conductive-fabric pieces on the chest of the T-shirt.

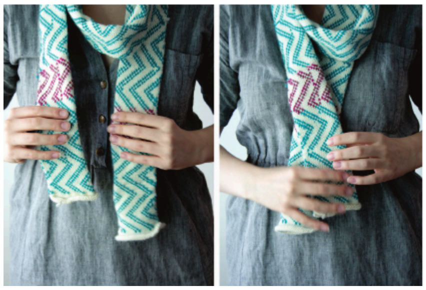


A light switch, a flashlight switch, and a mystery button are all switches that permit or disrupt the flow of electricity.

A switch is something that enables, prevents, or diverts the flow of electricity. It creates or breaks the physical connection of two conductors. Familiar switches include a standard toggle switch that controls the lighting of a room, the slide switch on the barrel of a classic flashlight, or even the blinking red DO NOT TOUCH button on the control panel of a spaceship. And when you put your imagination to use, you can create



This custom necktie by fashion designer Mystica Cooper seamlessly integrates electronics into the design of the tie.



A contact switch created by Erin Lewis integrates conductive thread with knitted yarn: When the pink patches of the scarf are held together, the switch is closed.

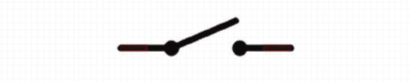
switches in forms you wouldn't expect.

Switches are awesome because they can act as either an intentional input or a passive sensor. A person can activate a circuit by doing something as deliberate as pushing a button on a circuit board or as subtle and intuitive as standing up, blinking their eyes, or even giving someone a hug.

In this chapter, you will learn how switches work, how to use them in simple circuits, what types of switches are available, and how you can make your own.

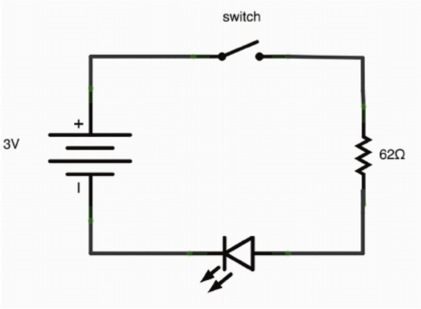
Understanding Switches

The circuit symbol for a basic switch is shown in the image at right.



Circuit symbol for a switch

If you were to integrate a switch into the basic circuit you built in chapter 2, the circuit schematic would look like the image at right.



Circuit schematic for simple LED circuit with a switch

When the switch is *closed*, the two contact points will be connected, and electric current will be able to flow—and the LED will light up.

When the switch is *open*, the two contacts are not connected, so the circuit is interrupted. Electric current is unable to flow through the circuit, so the LED will not light.

POLES AND THROWS

A basic toggle switch or rocker switch is called a *single-pole, single-throw*, or SPST, switch:

Pole

This terms refers to the number of separate circuits controlled by the switch.

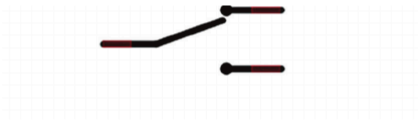
Throw

This terms refers to the number of positions each pole can be connected to.

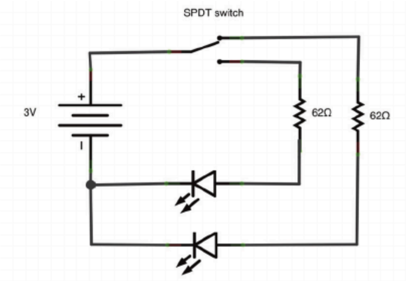
SPST switches have two terminals. When the switch is in the Off position, the connection is open. When it is moved to the On position, it closes the connection, and electricity is able to flow.

An SPST rocker switch

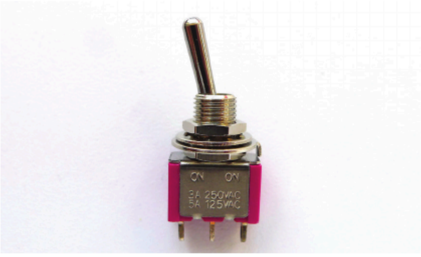
A single-pole, double-throw (SPDT) (*single-pole, double-throw*) switch has three terminals. This type of switch can be used to switch between two circuits. If needed, a SPDT switch can also operate as a SPST—just don't connect anything to the second throw.



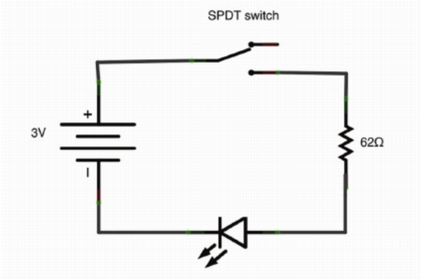
Circuit symbol for a SPDT switch



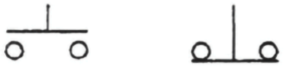
In this circuit, an SPDT switch is used. Only one of the two LEDs will light, depending on the position of the switch.



SPDT toggle switch



In this circuit, the SPDT switch is used as an SPST.



NO and NC momentary switches

In the world of electronic switches, there are far more complex combinations of poles and throws, but it's not likely you'll be using them in basic wearable-electronics projects. The most important thing to understand is what a switch is and how it works in your circuit.

TYPES OF SWITCHES

There are two categories of switches you'll encounter: momentary and maintained.

Momentary switches stay in their state only as long as they are being activated. Once the switch is released, it returns to its previous state, as with the buttons on a remote control. Momentary switches can be normally open or normally closed.

Normally open (NO)

This term refers to a momentary switch whose default state is for the contacts to be open. If it is activated by the user, the switch will close.

Normally closed (NC)

This term refers to a momentary switch whose default state is for the contacts to be closed. When the switch is activated, the switch will open, deactivating the attached circuit.

Maintained switches are the opposite of momentary switches. They will stay in whatever state you put them in. Think of a light switch in a room. When you turn the lights on, the switch doesn't spring back to the Off position when you let go of it.

Off-the-Shelf Switches

Now that you understand some basic switch terminology and how to use a switch in a circuit, you can look at what sort of electronic switches are out there.

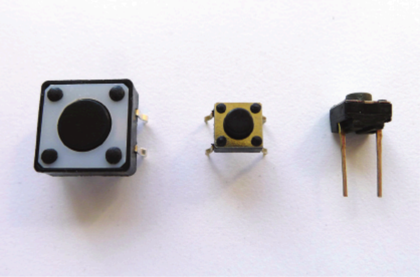
Within the world of electronics, there is a wide range of switches available. Which of these switches are best suited for wearables? For wearables, you generally want either discreet switches to turn things on or off or to change between modes, or switches that work with the body's natural movements. Here are some useful options.

TACTILE BUTTONS

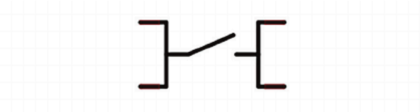
Tactile buttons are momentary buttons that provide light tactile feedback. Because they are momentary, they can be normally open (NO) or normally closed (NC). Their packaging is flat and slim, which enables them to sit well within the profile of garments. For wearables, a broader button face tends to be better than a smaller one, as it provides a larger landing pad for an incoming fingertip. Some tactile switches have four legs. However, it is possible to use a tactile button as a simple SPST switch. Tactile buttons work best when mounted on a circuit board.

LATCHING BUTTONS

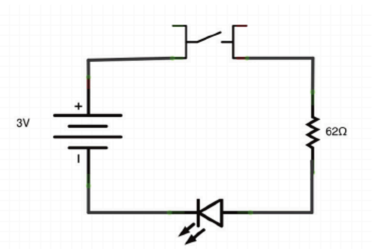
Latching buttons (sometimes called tactile on/off buttons) are maintained switches that respond to being pressed. Press it once and the button will stay closed. Press it again and the button will stay open. They are an excellent option for integrating into the seam of a piece of clothing.



Large and small tactile switches

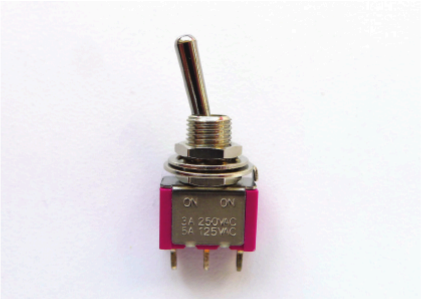


Circuit symbol for tactile switch with four legs



Using a tactile switch as an SPST

This latching button would work well as an on/off button that could be subtly situated in a collar, a cuff, or a sleeve.



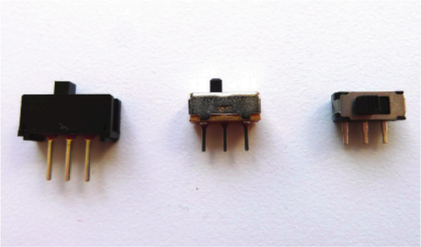
Toggle switch

TOGGLE SWITCHES

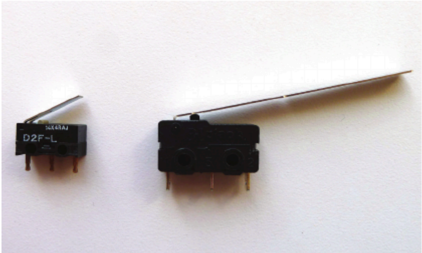
A toggle switch, a maintained switch operated by a lever, is not often used in wearables because the protruding lever is not the most comfortable thing to wear. Toggle switches tend to work best when mounted on a control panel.

SLIDE SWITCHES

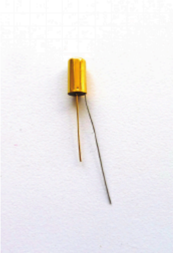
Slide switches, a type of maintained switch



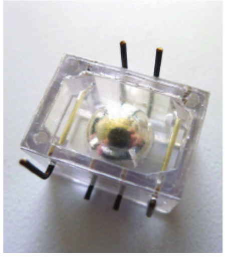
Slide switches



Microswitches with levers



A tilt switch



A multiaxis tilt switch

controlled by moving the switch back and forth in a linear motion, are small and stable, useful for functions like turning a project on and off. Many of these mount well on a circuit board.

MICROSWITCHES

A microswitch is a highly sensitive momentary switch that can be operated by a small movement. A microswitch can come as a wire, a lever, or a roller, which can make it more easily triggered by a certain type of mechanical action. When positioned well, microswitches can be used to detect subtle body movements.

TILT SWITCHES

Tilt switches open or close based on the orientation of the switch. In the past, tilt switches contained mercury. These days, they usually contain a small conductive ball. When the switch is upright, the ball will sit on top of the two contacts, bridging them and closing the circuit. When the switch is inverted, the ball moves and breaks the connection.

Tilt switches work well with wearables because they can disappear into a garment and respond to the movements of the body without the wearer even thinking about it. Raising your hand or touching your toes can suddenly become a way to activate a circuit.

There are also multiaxis tilt switches, which contain multiple sets of contacts. This is useful when you are looking to sense multiple orientations, such as whether you are lying on your front, back, left side, or right side.

DIY Switches

If none of these off-the-shelf switches meet your needs, you can always make your own: Arrange two conductors in such a way that they will sometimes touch and sometimes not, and you've got yourself a switch! Generally, the aim is to come up with a switch that behaves reliably—that is, unless random activation has worked itself into the concept for your project.

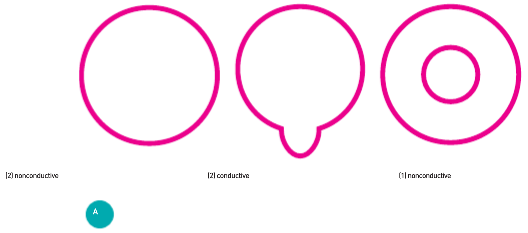
DIY switches can be fully integrated with the materials used in your project so they almost become invisible to the user.

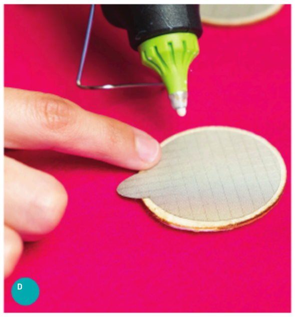
Contact points for switches can be made with any conductive material. These examples use iron-on conductive fabric, but you can substitute conductive thread, yarn, paint, mesh, or any other conductive material of your choosing.

HOW TO: Soft Button

A soft button is the DIY version of a normally open momentary switch: Press it to close the circuit, and release to open it. I sometimes think of this as a sandwich switch because it consists of layers. You could also think of it as a lasagna or a layer cake, but at some point this would turn into a buffet instead of a button-making exercise.

To create a soft button, you need a nonconductive material and a conductive material. It is preferable to use something stiff but squishy, like thick felt, foam, or neoprene, for the nonconductive material—you want it to yield to the touch but spring back afterward. This material will give the switch its physical form and insulate the conductive materials that lie within. For the conductive material, this example uses iron-on conductive fabric, but other conductive materials can be substituted.





Maxwell Lander

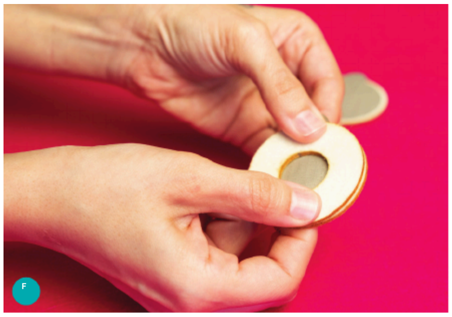
Steps:

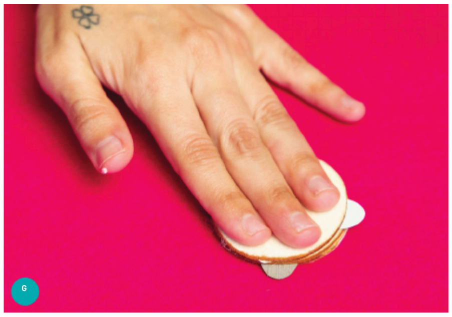
1. Draw and cut paper patterns for the materials—one for the conductive and one for the nonconductive. They can be approximately the same shape, but make sure the conductive-material shape is slightly smaller than the nonconductive-material shape. The conductive-material shape should also include a tab long enough to be clipped with an alligator clip (Figure A).
2. Trace the patterns. Cut three pieces of a nonconductive material in the same shape, one with a hole in the middle. Cut two patches of conductive fabric (Figure B).

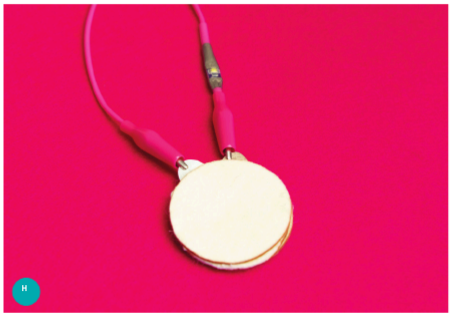
3. Use an iron to adhere the two pieces of conductive fabric in the middle of the two nonconductive pieces. Do not iron the tabs (Figure C).

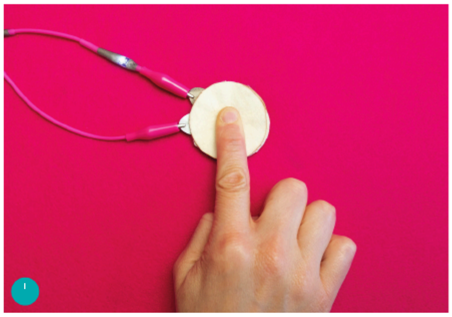
4. Place one piece conductive side up (Figure D).

5. Use a glue gun to create a ring of glue along the perimeter (Figure E).









Maxwell Lander

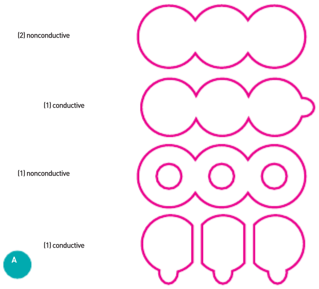
6. Press the piece with the hole on top (Figure F).

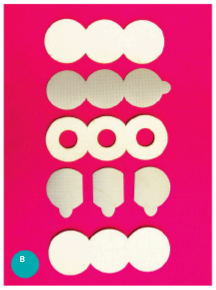
7. Repeat the process to attach the second piece with the conductive fabric facing inward. Position this piece such that the tabs are not too close together (Figure G).

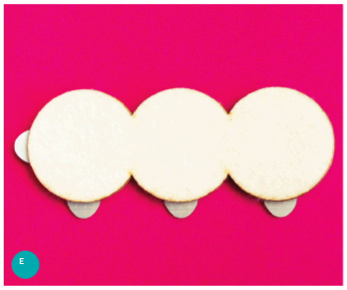
8. Connect your conductivity tester to the two tabs (Figure H).

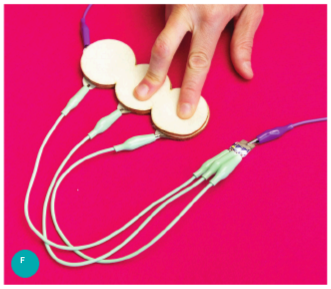
9. Press the button to activate the LED (Figure I).

When creating switches from scratch, it is important to test that they work reliably. If you don't have a conductivity tester, you can use a multimeter. Set the dial of the meter to the continuity setting. Connect the probes to the two contact points of the button with an alligator clip. Press the button and the meter will beep!









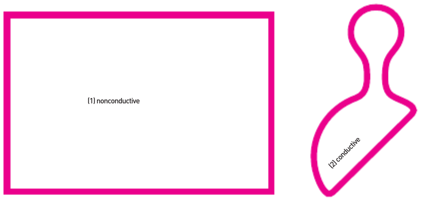
Maxwell Lander

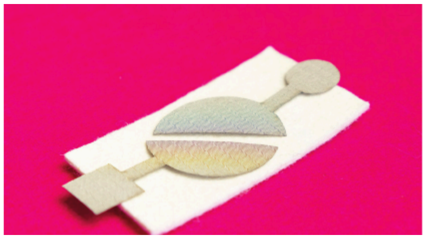
HOW TO: Multiple Soft Buttons

Using the same design principles, we can create a set of soft buttons that include multiple trigger points.

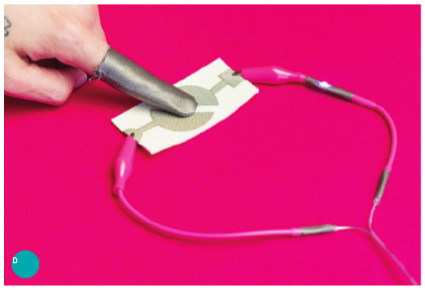
Steps:

1. Draw and cut paper patterns for the materials. Note that one side of the conductive material can be shared across all three buttons (Figure A).
2. Cut the conductive and nonconductive materials (Figure B).
3. Lay out the conductive fabric and iron it to secure it in place (Figure C).
4. Test the arrangement of the pieces (Figure D).
5. Use hot glue to secure the pieces in place. Be sure to glue only around the perimeter of the pieces—not in the middle (Figure E).
6. Test each switch. This testing method uses three sewable LEDs with the ground pins connected with conductive-fabric tape (Figure F).





A



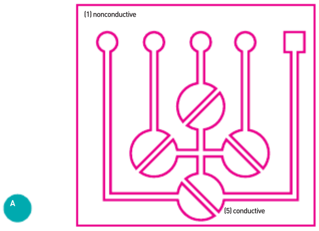
Maxwell Lander

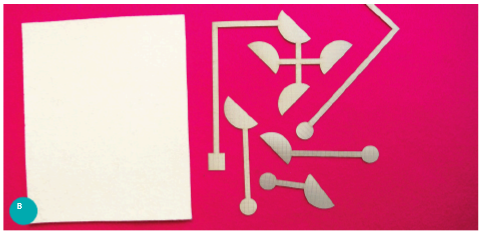
HOW TO: Bridge Switch

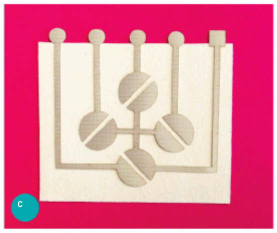
A bridge switch leaves a small break between two pieces of exposed conductive material. When another conductive object or garment is put in contact with the break, it bridges the connection and closes the circuit.

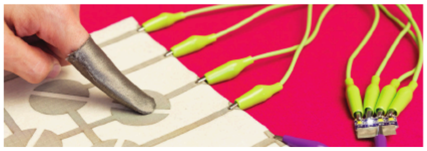
Steps:

1. Draw and cut paper patterns for the two pieces of conductive fabric. Be sure to include tabs long enough to be clipped with an alligator clip (Figure A).
2. Lay out the two pieces of conductive fabric so they are close but not touching (Figure B).
3. Carefully iron the conductive fabric in place (Figure C).
4. Connect a conductivity tester to the two sides of the switch. Cover a fingertip with conductive fabric or copper tape. Test the bridge switch—the LED should light up (Figure D).









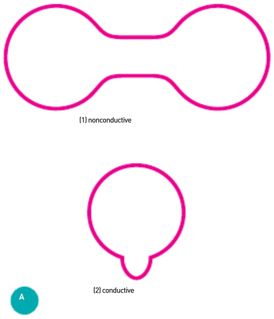
Maxwell Lander

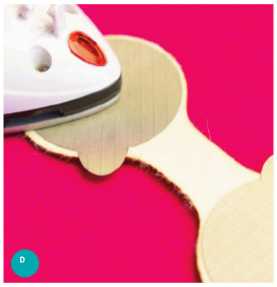
HOW TO: Multiple Bridge Switches

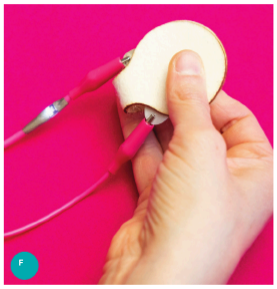
Similar to the multiple-soft-button design, a common contact can be used for one side of the multiple bridge switches if you are clever with a design.

Steps:

1. Draw and cut paper patterns for the conductive fabric. Be sure to include tabs long enough to be clipped with an alligator clip (Figure A).
2. Trace the conductive-fabric pieces and cut them with scissors, a craft cutter, or a laser cutter (Figure B).
3. Arrange the conductive-fabric pieces (Figure C).
4. Iron in place (Figure D).
5. Flip them over, fold the tabs, and iron them in place (Figure E).
6. This creates lovely contact points for your alligator clips (Figure F).
7. Connect some LEDs and test each of the bridge switches (Figure G).







Maxwell Lander

HOW TO: Pinch Switch

A pinch switch leverages the foldability of fabric. Line a stiff strip of material with two pieces of conductive material. When you pinch it together, the two conductive pieces will touch and close the connection.

Steps:

1. Trace the patterns on paper or create your own (Figure A).
2. Use the patterns to cut the required quantities and shapes of each material (Figure B).

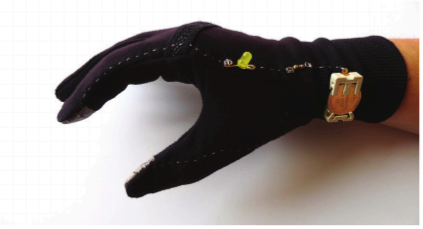
3. Position the conductive fabric. Think about the direction of the tabs and where you would like the wires to fall (Figure C).

4. Iron in place (Figure D).

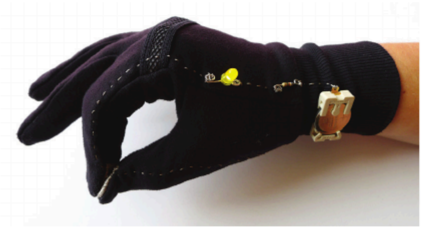
5. Connect the conductivity tester (Figure E).

6. Pinch to close the switch (Figure F).

Gloves with conductive fingertips



Glove with contact switch open



Glove with contact switch closed

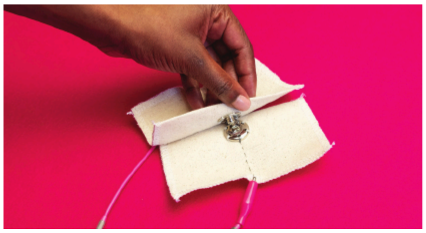
CONTACT SWITCH

A contact switch contains two conductive surfaces that will at some point make contact with each other. This is fun to play with in wearables because you can consider the way the body moves. Different body parts make contact when your arms are by your side, your heels click together, or your head is in your hands.

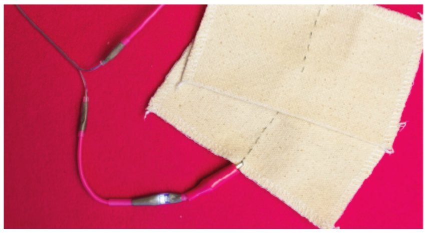
These days, some gloves come with conductive fingertips intended for use with smartphone screens. These can be modified so the fingertips act as a contact switch, lighting up when the index finger is touched to the thumb and turning off when the connection is released.

SNAP SWITCH

Uninsulated metal clothing fasteners can be used as DIY switches. This type of fastener goes by many names around the world—snaps, poppers, press studs, etc. When connected with conductive thread, they can be used to activate or deactivate a circuit.

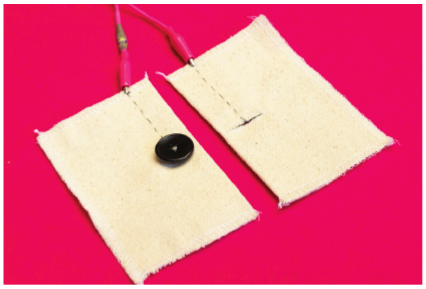


Snap switch open

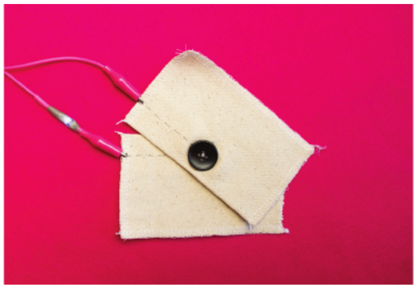


Maxwell Lander

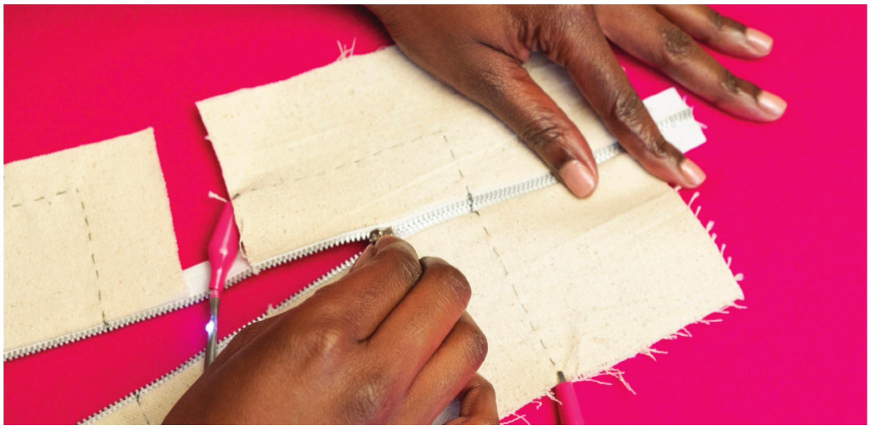
Snap switch closed



Clothing-button switch open



Clothing-button switch closed



Maxwell Lander

A zipper switch when closed

CLOTHING-BUTTON SWITCH

A clothing button can turn into a DIY switch when combined with conductive thread. Use one piece of thread to sew the button and another to line the buttonhole. When buttoned, the switch will be closed.

ZIPPER SWITCH

There are many variants of a zipper switch. This one closes a circuit by bringing together two pieces of conductive thread when the zipper is closed.

EXPERIMENT: High-Five Gloves

When properly planned, a switch can create the opportunity for a social interaction. This experiment demonstrates how to use a simple sewn circuit that includes a bridge switch to create a wearable that responds to a social interaction between two people.

Parts and materials:

- Craft felt

- Iron-on conductive fabric
- (1) sewable CR2032 battery holder
- (1) sewable LED
- Conductive thread
- Nonconductive thread

Tools:

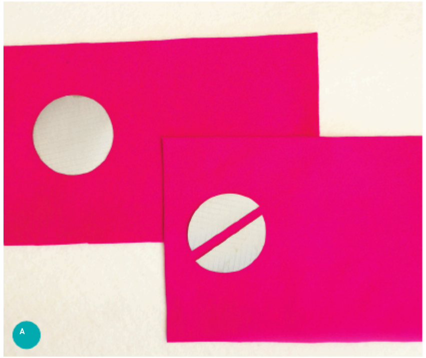
- Sewing needle
- Scissors
- Iron
- Ironing board or mat

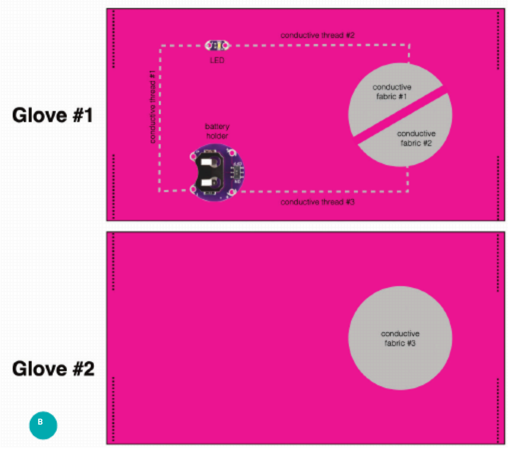
Steps:

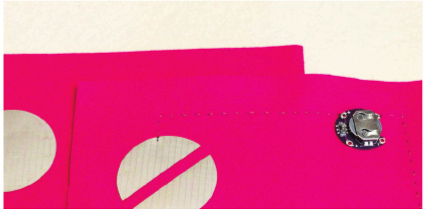
1. Cut two rectangles of felt wide enough to wrap around your hand and long enough to reach from your knuckles to your wrist. Cut two pieces of iron-on conductive fabric roughly the size of your palm. Cut one of those in half. Use an iron to secure the conductive fabric in place (Figure A).
2. Lay out the components as indicated in the diagram (Figure B).
3. Sew the circuit using conductive thread. Be sure to use three pieces of thread for each of the three connections. If the threads are connected, the circuit will not work (Figure C).
4. Fold both pieces of felt in half (Figure D).
5. Sew or glue the sides closed, but leave an open gap for a thumb hole.
6. Put one of the gloves on one of your hands and give the other to a friend (Figure E).
7. Light up with a high-five (Figure F).

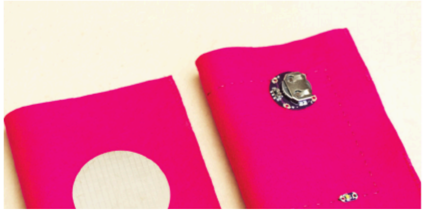
Looking Ahead

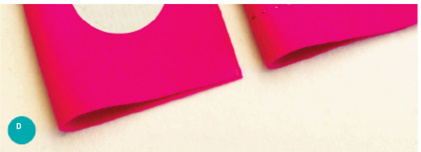
Switches are your first opportunity to create interactive circuits as well as interfaces that live on the body. Later, you will learn about more complex inputs, but even through the use of a simple switch and thoughtful design, it is possible to create highly engaging body-based interactions.









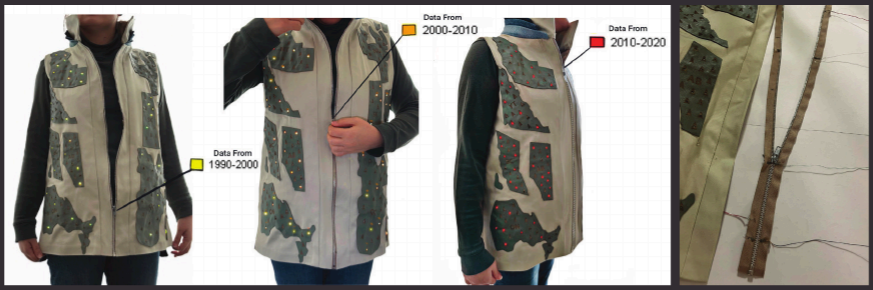


Maxwell Lander

GALLERY 4:

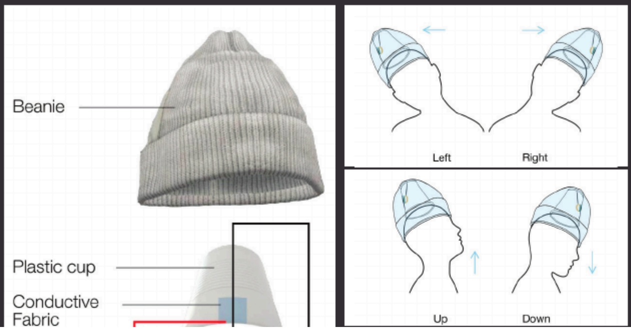
SWITCHES IN WEARABLES

Even though switches often have only two states (on or off), they can turn body movements into actions that control a circuit when worn on the body. Here are some projects by artists and designers that demonstrate how switches can be used in unconventional ways.



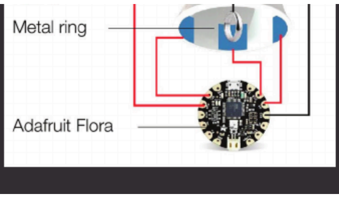
Canada Wildlife Vest

This vest by Maryam Dehghani is a wearable data visualization of Canadian wildfires. Color-changing LEDs are positioned within laser-cut shapes inspired by Canadian provinces. As the wearer moves the zipper switch into different positions, they can activate a visualization of statistical data in various periods.

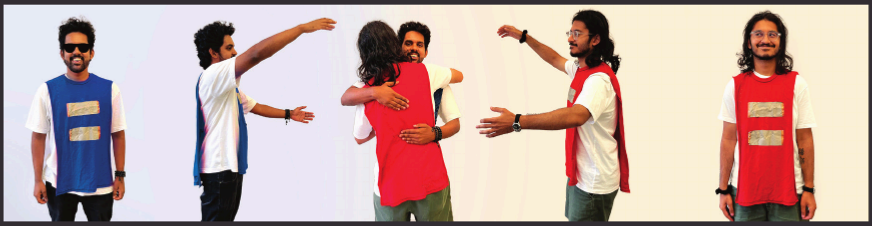


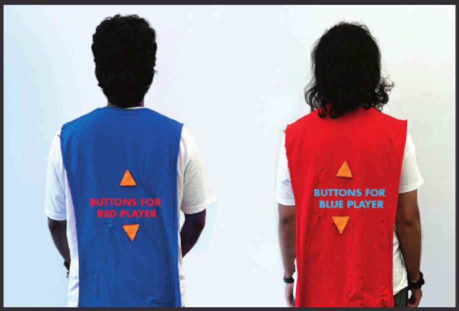
Ctrl Alt Tilt

Ctrl Alt Tilt, by Aishwarya Bhattbhatt, is a wearable game controller. When you tilt your head left, right, forward, or backward, the hanging metal ring comes into contact with one of the conductive-fabric pieces on the sides of the cup. This contact completes the circuit and triggers corresponding actions on the keyboard and the connected game.



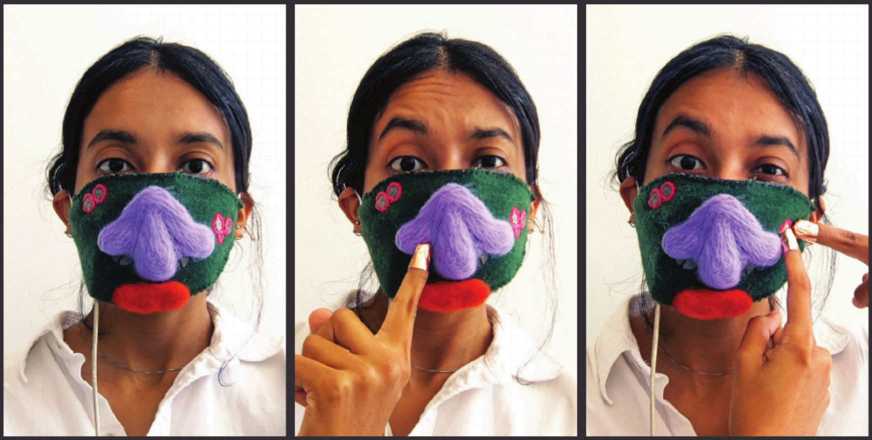
The movement of your head imitates the directional controls you would find on a traditional gaming controller.





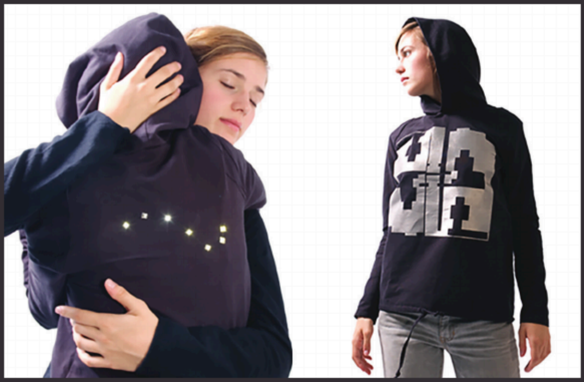
PL.UG

PL.UG, by Nikhil Thomas, is a wearable game controller that thrives on the irony of requiring a hug to play against your opponent: Two horizontal strips of conductive fabric on each vest create a switch that can be closed only when the two players embrace, giving access to their buttons placed on the opponent's back.



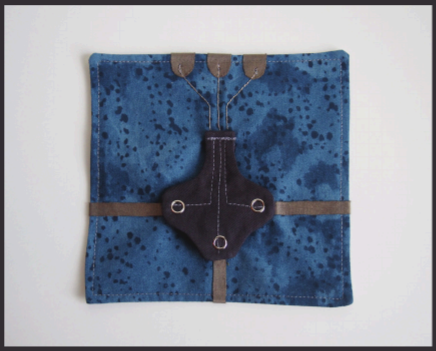
Booger Bonanza

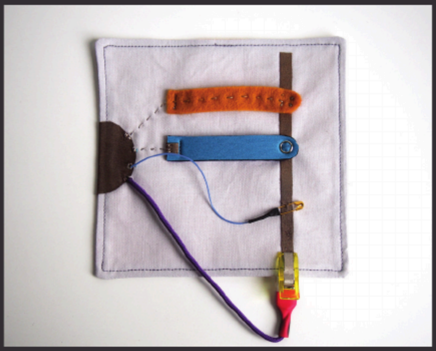
Booger Bonanza, by Abha Patil, is a whimsical game controller that allows you to pick your nose and pop your zits to achieve victory. An Adafruit Flora microcontroller is embedded in the mask, connected to four switches activated by the wearer's fingers.



Embrace Me

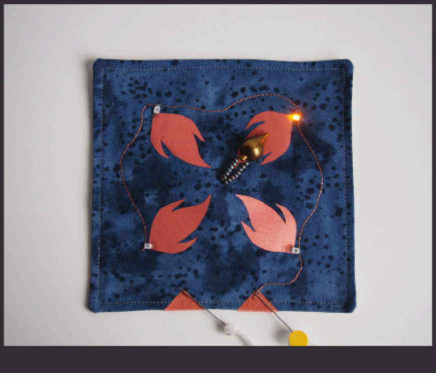
Embrace Me, by Studio 5050, is a pair of hoodies. When two wearers hug, conductive fabric on the chest of each hoodie connects and closes a circuit, causing a pattern on the back of the hoodies to light.

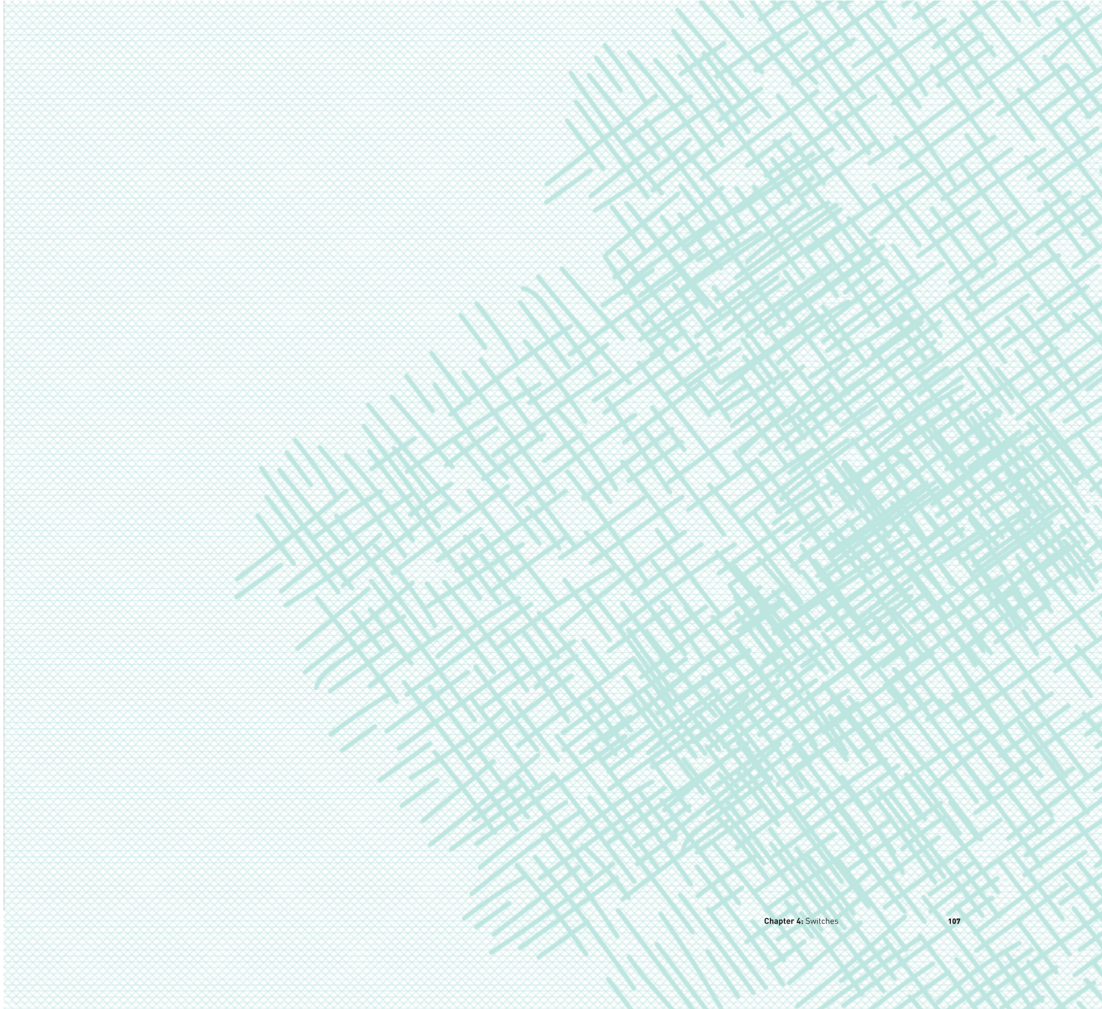




Soft Switches

These swatches were created by Liza Stark for use in teaching and prototyping. The project prioritizes embodiment and tangibility as a central mechanism of learning, acting as a manifestation of material, technical, and social knowledge. Components include snaps, Karl Grimm thread, copper taffeta fabric, Med-tex conductive fabric, surface-mounted-device (SMD) LEDs, conductive and nonconductive beads, and various connectors. These swatches include demonstrations of different kinds of switches, including momentary buttons, toggle switches, and tilt switches.

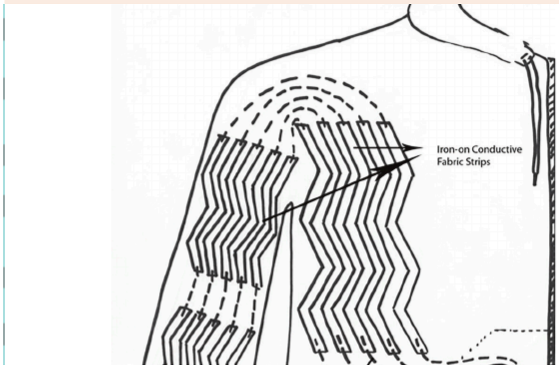


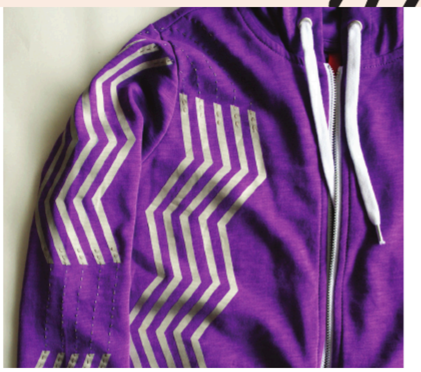


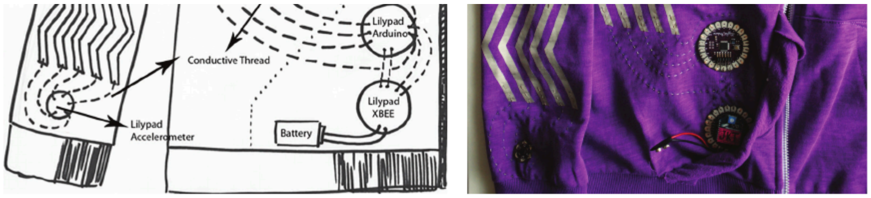
5

[illegible]

Tool Kits





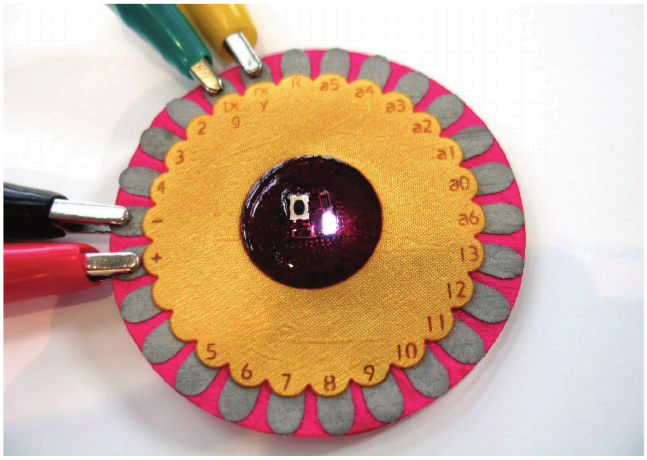


Audience Jacket, by Social Body Lab (Kate Hartman, Ken Leung, Erin Lewis, Oldouz Moslemian)

E-textile tool kits are families of modules designed specifically for use with nontraditional conductive materials such as conductive thread or fabric. For example, SBL's Audience Jacket is created using conductive fabric, thread, and LilyPad modules. Data from a LilyPad accelerometer positioned on the wrist is read by a LilyPad Arduino, which then transmits the data wirelessly via an LilyPad XBee radio to a second XBee radio module attached to a nearby computer running Arduino and Processing. As the wearer raises their arms, a cheer is heard. As the wearer claps, the motion triggers the sound of a crowd clapping. If wearer gives a thumbs-down, a chorus of boos is heard. When the arms are raised, a "Woo!" sound is activated.

In the previous chapters, you learned about the diverse range of conductive materials that can be used to construct circuits. But as you saw when constructing soft circuits in chapter 2, standard electronic components need to be modified if they are to be used with soft conductive materials. In the last chapter, we worked with sewable LEDs. In this chapter, you will learn about a few e-textile toolkits, the modules they contain, and how they can be used within a circuit.

Adobe Stock Lepusinensis



Maxwell Lander

LilyPad early prototype, by Leah Buechley

Because these tool kits are emerging platforms, there is no standard vocabulary for these systems or their parts. Before you get started looking at them in detail, let's define some of the terms we'll be using.

E-textile

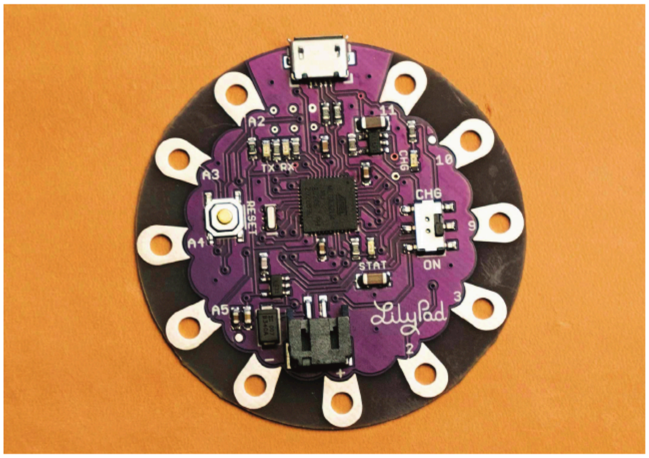
This abbreviation of "electronic textile" refers to a category of electronic parts that can be used in combination with textiles and other soft materials.

Tool kit

This refers to a set of tools that can be used for any number of purposes. This is different from a kit, as a kit's parts are usually meant to be used together to assemble a single thing. Other terms you might run into that mean roughly the same thing are *platform* and *system*.

Module

This refers to a discrete unit that is part of a tool kit. Modules are printed circuit boards that contain electronic components and their necessary connections. Modules can be connected using conductive materials to create a complete circuit.



Maxwell Lander

Lilypad Arduino USB, by Leah Buechley and SparkFun Electronics

The tool kits that will be reviewed in this chapter are mostly designed for use with conductive threads, but they can ultimately be used with whatever conductive materials you like.

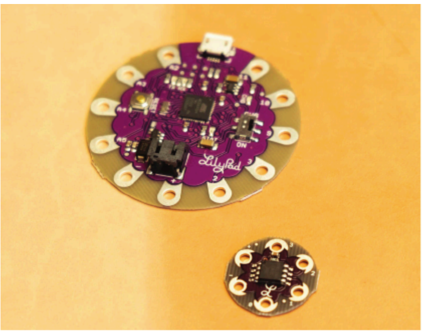
LilyPad

The LilyPad was the first set of widely available electronic components specifically intended for integration with nontraditional conductive materials. First released by SparkFun in 2007, the LilyPad was based on years of research by Leah Buechley during her time at the University of Colorado Boulder.

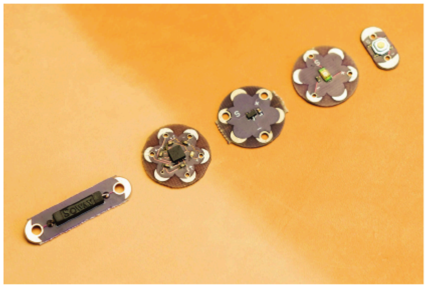
Several characteristics at the time set LilyPad modules apart from more traditional circuit boards.

Sewable

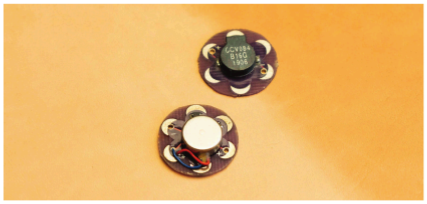
LilyPad modules are designed specifically to enable electrical connections made with hand-sewn conductive thread. Connection points are situated around the perimeter of the boards so they can be easily accessed. These areas are referred to as sew tabs, sew holes, or petals.



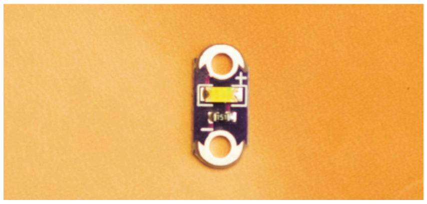
LilyPad Arduino USB and LilyTwinkle



LilyPad Reed Switch, accelerometer, temperature sensor, light sensor, and push button



LilyPad Vibe Board and buzzer



Maxwell Lander

LilyPad LED

Rounded edges

All LilyPad boards have rounded edges. This works well in the context of floppy flexible fabric substrates. If you bend a body part while wearing a LilyPad component, you won't get jabbed by a sharp corner.

Thin

LilyPad boards are a bit thinner than a traditional circuit board, which means they are less bulky and easier to incorporate into a lining, pocket, or seam.

Purple

LilyPad modules are purple. This is meant to make them friendlier and more attractive, and it makes them easier to pick out from a sea of parts.

The LilyPad has set a precedent for many e-textile tool kits that have been developed since. You will see that many of the design choices—such as round boards and sew tabs—are echoed in the tool kits that follow.

There are a variety of modules within the LilyPad tool kit. Some of these are similar to components you used in previous chapters and others you will learn to use in chapters that follow.

Microcontrollers—tiny computers that will live in your jacket lining or your pocket or under your hat—are the brains of your circuit. The LilyPad tool kit contains several options for microcontroller boards. You'll learn more about microcontrollers in the next chapter.

There is also a useful selection of switches and sensors available as LilyPad parts, including a slide switch, a push button, a light sensor, a temperature sensor, and an accelerometer. These modules contain all the necessary resistors and connections so their pins can be connected directly to a microcontroller module without the need for any additional circuitry.

There are several types of actuators in the LilyPad tool kit that can be used for decoration, feedback, or display, including a vibrating motor, a buzzer, and several types of LEDs.

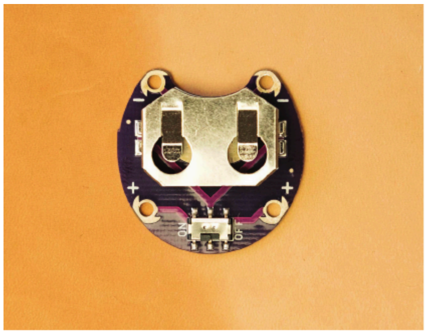
As we've seen in previous chapters, sewable battery holders work well with smaller LED projects.

If you'd like to jump right into programming before building circuitry, the LilyPad ProtoSnap Plus features a LilyPad USB Plus board connected via circuit board traces to a variety of LilyPad sensors and actuators. When you're done programming, you can snap it apart and build the modules into your wearables project.

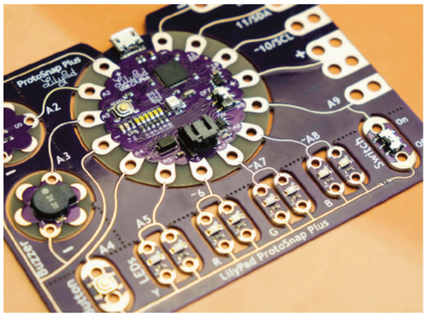
Flora and Gemma

Created by Adafruit Industries in New York, the Flora is a slightly newer e-textiles tool kit. Building on design choices introduced by the LilyPad, the Flora makes some significant leaps forward from an engineering perspective as well as some additional modules.

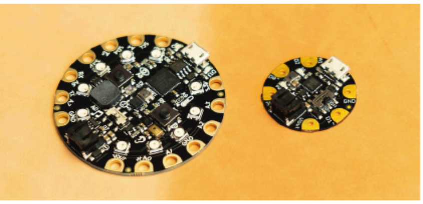
Though not official Arduino boards, the Flora and Gemma microcontroller modules can be programmed with a version of the Arduino Integrated Development Environment (IDE).



LilyPad Coin Cell Battery Holder



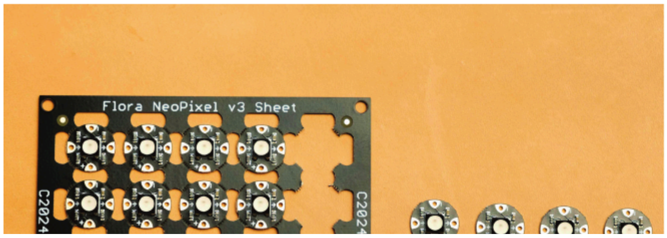
LilyPad ProtoSnap Plus

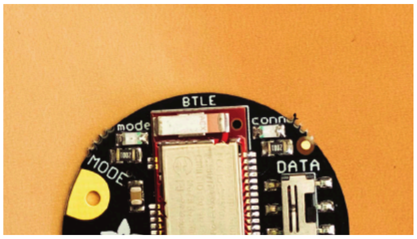


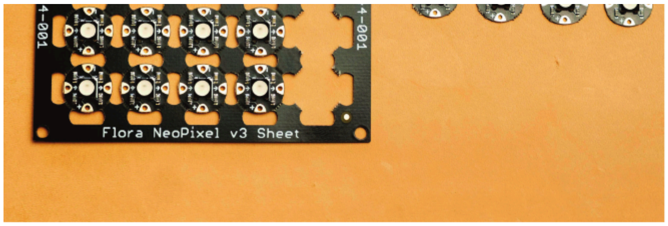
Flora and Gemma

Maxwell Lander

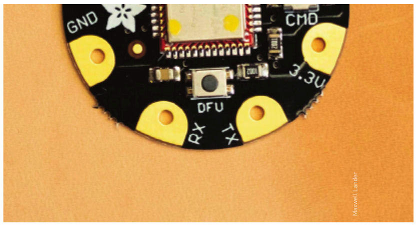
Adafruit LED Sequin







Flora NeoPixels V3



Maxwell Lander

Flora Wearable Ultimate GPS Module

The Flora mainboard is slightly smaller than the LilyPad Arduino USB. It has an extremely robust power system and is designed specifically for use with fabric. We'll look more closely at the Flora in the next chapter.

The Gemma, a tiny cousin of the Flora, is programmed via USB and features three digital pins (two with pulse width modulation, or PWM) and one analog input pin. Because of its small size, it is great for lightweight or tightly spaced applications.

Adafruit LED Sequins are smaller than LilyPad LEDs, and the LEDs are in a different orientation in relation to the sew tabs.

One of the Flora's featured modules is the Smart NeoPixel. These are by far the most heavy-duty LEDs you've seen in the context of e-textiles tool kits. They are very bright, chainable, and individually addressable, which means that if you have a strand of ten pixels, you can set the color and brightness of each one individually.

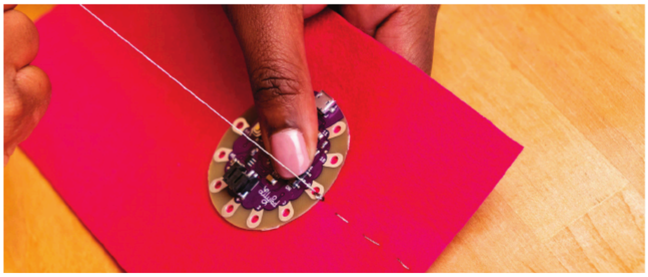
The Flora tool kit also offers a wireless Global Positioning System (GPS) module that is excellent for outdoor, location-aware projects.

Connecting to Sew Tabs

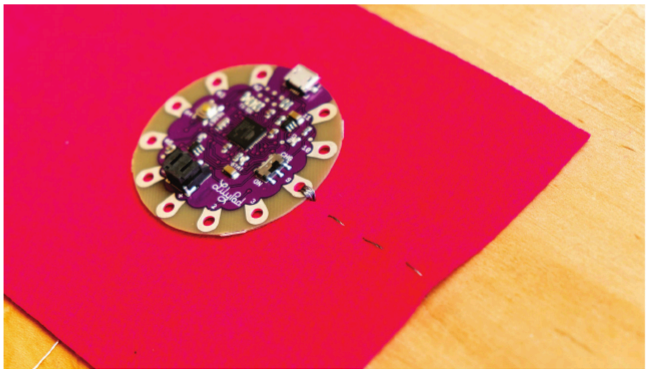
Sew tabs and clippable connections have become much more popular in circuit board design in recent years. Beyond the LilyPad and Flora product lines, they can be seen in Bare Conductive's Touch Board, micro:bits, and Chibitronics. Let's look at some methods for connecting to them.

SEWING SEW TABS

Connections to sew tabs are typically made with either alligator clips (for testing) or conductive thread (for final connections). Let's take a closer look at how to make sewn connections.

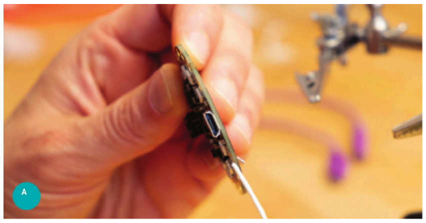


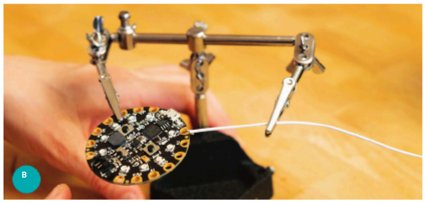
When sewing a sew tab with conductive thread, be sure to pull the thread all the way through.

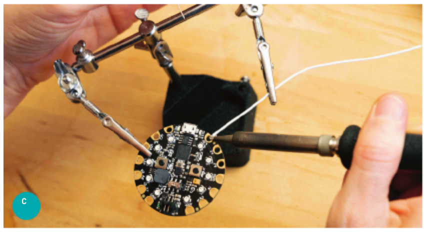


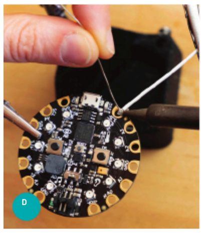
Maxwell Lander

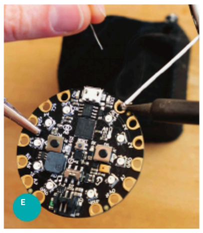
Loop around the sew tab several times to ensure a snug physical and electrical connection. The result should look neat and tidy.











SOLDERING SEW TABS

It is also perfectly fine to make soldered connections if needed. This can be especially helpful for connections that are more sensitive (for example, for certain sensors) or that need a lower resistance or more reliability than conductive thread can provide (for instance, if using a power source with higher voltage or current output than a coin cell). The steps that follow demonstrate how you might do this.

A Strip the end of a piece of wire and make a hook to hook around the sew tab.

B Secure the wire and the circuit board with a pair of helping hands.

C Apply heat at the intersection of the exposed wire and the sew tab.

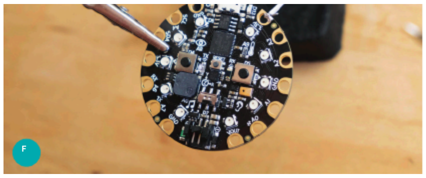
D Apply solder to the sew tab and allow it to melt into that junction.

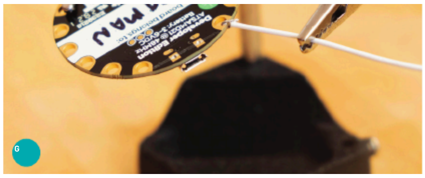
E Remove the solder first.

F Remove the soldering iron's tip second.

G Repeat on the back for extra security.

Note that solder has been used to connect the surface of the sew tab and the wire, but care has been taken to not completely fill in the hole. This will provide a reliable physical connection but should still be relatively easy to desolder if needed.





Maxwell Lander

HOW TO: Twinkle and Tiny

To get comfortable working with e-textile tool kit components, let's make a simple circuit with either a LilyTwinkle or a LilyTiny. These microcontroller modules come preprogrammed with behaviors for LEDs attached to the four output pins. Let's see how these circuits can be assembled.

LILYTWINKLE

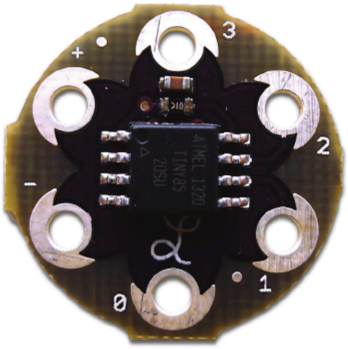
Parts and materials:

- (1) LilyTwinkle
- (1) LilyPad Coin Cell Battery Holder
- (1) CR2032 battery
- (4) LilyPad LEDs
- (7) alligator clip test leads
- Conductive thread (if you would like to sew your circuit)

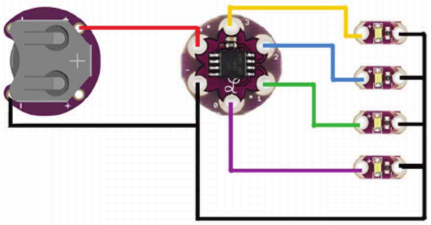
Make the connections shown in the diagram using either alligator clips or conductive thread. If you need a refresher on how to sew circuits with conductive thread, see chapter 2.

Once your circuit is complete, insert the battery in the battery holder. Your LEDs should twinkle away!

The LilyTwinkle program lights LEDs at random, very much like fireflies. Remember that if you have only one LED connected, there may be a long pause before that LED lights up again. If an LED doesn't light immediately, give it time, or connect the other LEDs to make sure the LilyTwinkle is working properly. Also, check the polarity of the LED to make sure it is connected correctly.

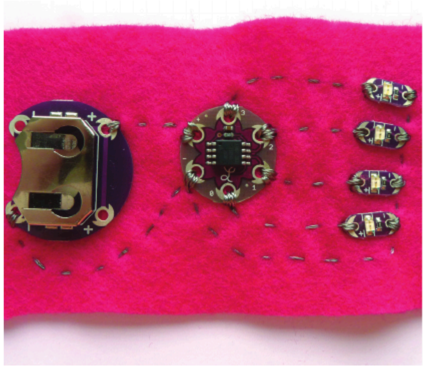


LilyTwinkle

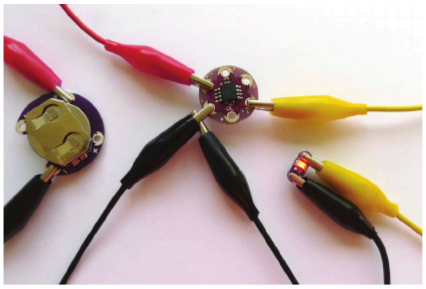


Created in Fritzing

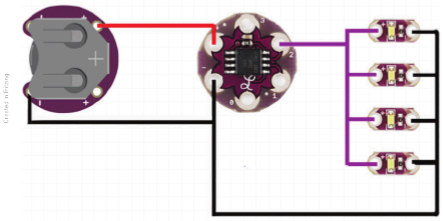
Connections diagram for the LilyTwinkle



LilyTwinkle circuit made with conductive thread



LilyTiny circuit made with alligator clips



LilyTiny LEDs in parallel

LILYTINY

Parts and materials:

- (1) LilyTiny
- (1) LilyPad Coin Cell Battery Holder
- (1) CR2032 battery
- (1–4) LilyPad LEDs
- (4–7) alligator clip test leads
- Conductive thread (if you would like to sew your circuit)

While the LilyTwinkle works well for ambient lights, you may want a more consistent behavior. The LilyTiny has the same hardware as the LilyTwinkle but comes loaded with a different program. It can be used to blink, beat, fade, or "breathe" LEDs attached to allocated pins.

LilyTiny pin functions

Pin	Function
0	Breathing fade
1	Heartbeat pattern
2	Steady blink
3	Random fade

If you're after just one of these behaviors, use just one of the pins.

If you'd like to multiple LEDs to perform the same behavior, you can add them in parallel to the first one. The total number of LEDs you can light will depend on the battery manufacturer, but usually, you can get at least eight or more to work.

As you can see, these circuits are simple and quick to construct. This is an excellent choice for someone who wants more dynamic behavior for LEDs but doesn't want to fuss with programming.

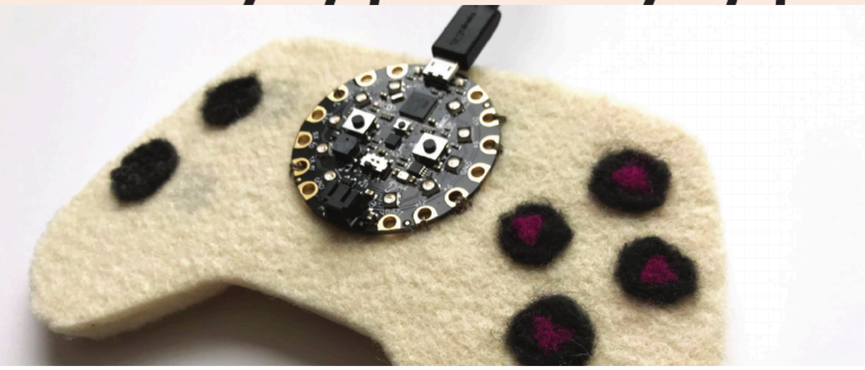
Looking Ahead

E-textile toolkits expand your options for integrating electronics into soft and wearable projects. But it is also important to think beyond these existing tool kits. If you're just getting started, grabbing a tool set like the LilyPad or the Flora can be handy to get a project going, particularly if you start with something like the LilyPad Protosnap Kit.

In the long run, there's probably not a wearables or e-textile tool kit that precisely meets the needs of your vibrantly unique idea. As you work through iterations of a project, don't be afraid to move between platforms and beyond them. The more knowledgeable and adaptable you become with your tools, parts, and materials, the more you will be able to mold your project to fit the curves and nuances of the human form and create entirely novel wearable solutions!

6

Micro-
controllers



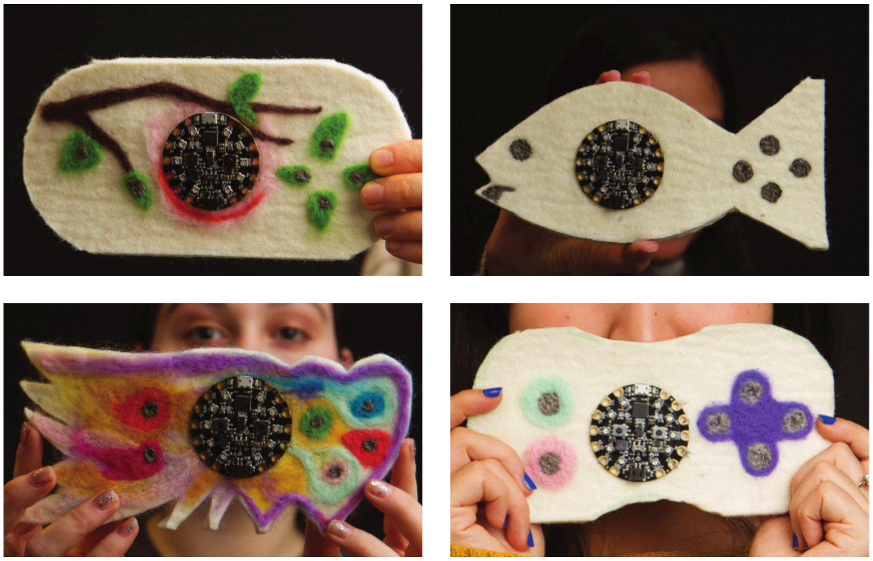
Felted Game Controller, by Social Body Lab (Kate Hartman, Olivia Prior, Yiyi Shao)

Hello, microcontrollers! The Textile Game Controller is an example of a soft interface created with the neat features of Adafruit's Circuit Playground Classic microcontroller board. Using industrial felt as a base, conductive felted sensing pads are connected to capacitive sensing pins, which activate key presses for browser-based games. Games are played by connecting the controller to a computer with a USB cable. The NeoPixels on the Circuit Playground provide local feedback for testing and debugging purposes.

This design originates from the "Fun with Felting" workshop, part of the Textile Game Controllers project—a multiyear collaboration between SBL, game:play Lab, and Dames Making Games. The selection of materials and techniques allowed participants to create game controllers that followed the style of traditional game controllers as well as those imagined in new shapes, colors, layouts, and forms.

In this chapter, you'll begin to explore your options for microcontrollers that can be embedded in clothing and other soft and wearable environments.

Adobe Stock-Lepusinensis



Created by Nimrah Syed

Created by Lucia Wallace

Created by Dandi Feng

Game controllers created with Adafruit's Circuit Playground Classic in the "Fun with Felting"

A microcontroller is basically a tiny computer. Think of it as the brain of your project. You may be more familiar with computers that come in the form of a desktop or a laptop device. Or you may be accustomed to smaller computers such as smartphones and tablets. But what if a computer could live in your clothing or other things you wear on your body?

workshop, hosted by Social Body Lab, game:play Lab, and Dames Making Games

While wearable computing has been an area of research for many years, its entry into the realm of consumer products is relatively recent. In addition, microcontrollers have become much more popular with hobbyists and makers due to their decrease in price and increasing availability and accessibility in the last few decades.

Microcontrollers are computers in their most basic form. This makes them an excellent tool with which to get started exploring how computation can live in the body space.

In this book, you will work with Arduino and Arduino-compatible products to meet your microcontroller needs. Arduino is an open source electronics-prototyping platform intended to be used by artists, designers, educators, hobbyists, and basically anyone who wants to make a physical interactive project but isn't an electrical engineer. The name refers to both the hardware and the software. Let's start with the hardware.

Hardware

ABOUT ARDUINO BOARDS

Arduino and Arduino-compatible boards are printed circuit boards that contain a microcontroller and its related components and circuits. These include pin breakouts, status LEDs, a reset button, and more. They make it easy to get microcontroller circuits up and running quickly without the fuss of building out these nitty-gritty aspects of the circuit yourself.

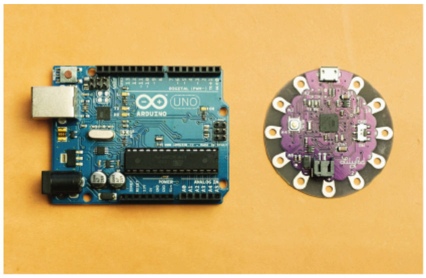
Many Arduino boards are available in a variety of configurations. You can learn more about current boards on Arduino's hardware web page (arduino.cc/en/hardware).

The most common Arduino beginners work with is the Arduino Uno. This is an excellent Arduino board, but from a wearable-electronics perspective, it is quite bulky.

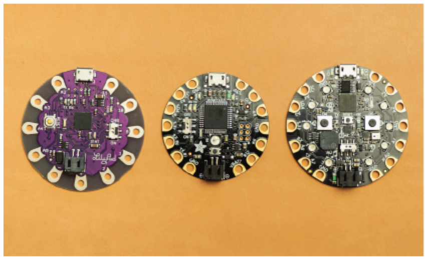
COMPARING MICROCONTROLLERS

In this section we will compare three options for wearable microcontrollers: the LilyPad Arduino USB, the Flora, and the CPX. Here are some features common to all three boards:

- Round shape
- Low-profile (relatively flat)
- Large conductive pads around the perimeter for use with alligator clips or conductive thread
- micro USB connector for programming
- 3.3 V operating voltage
- JST connector for a battery or battery pack
- Built-in LED

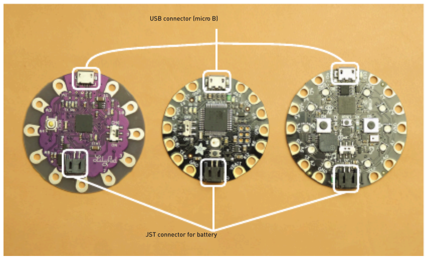


Arduino Uno (left) and LilyPad Arduino USB (right)



Maxwell Lander

Sewable microcontrollers: LilyPad Arduino USB, Flora, and the CPX



Looking at these three boards, you can spot two connectors that are used quite frequently on each one—a micro USB connector and a JST connector.

- Built-in USB support

The following table reveals some differences between these boards.

MICROCONTROLLER COMPARISON

	LILYPAD ARDUINO USB	FLORA	CIRCUIT PLAYGROUND EXPRESS
Source	Sparkfun	Adafruit	Adafruit
Intended Use	e-textiles	wearable electronics	versatile all-in-one board
Diameter	50 mm	45 mm	50 mm
Board Thickness	~7 mm	~7 mm	~8 mm
Weight	~5 g	~4 g	~9 g
Input Voltage	3.8–5 V	3.5–16 V	3.5–6 V
Battery-Charging Circuit	yes	no	yes
Programming Languages	Arduino	Arduino	Arduino, Microsoft MakeCode, JavaScript, CircuitPython
On/Off Switch	yes	yes	no
Built-In Addressable LED	no	yes	yes

Parts and materials for this chapter:

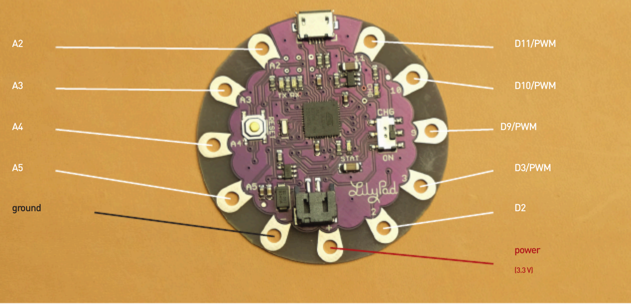
- (1) wearable microcontroller, as defined above
- (1) sewable LED (or through hole LED + 220 Ω resistor)
- (1) push button, manufactured or DIY
- (1) LilyPad light sensor (or photocell +10K Ω resistor)
- (1) USB mini-B cable
- Alligator clip test leads

Pinout diagrams are also an excellent way to learn about the pin layout and features of different microcontroller boards. Always start by looking at the most up-to-date pinout diagram the manufacturer provides. For ease in comparing these boards, below, a similarly styled pinout diagram is shown for each one.

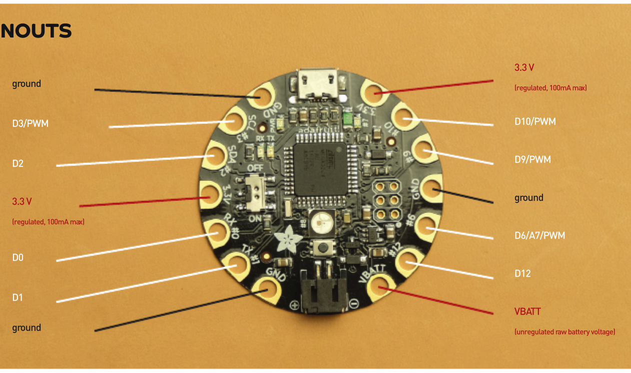
You can see from these pinout diagrams that each board has a slightly different pin selection. As we move through the examples in this chapter, you may need to switch to a pin number that is available on your board. Note that when changing pin numbers in any of these examples, it is essential to use the same number in the code as in the physical circuit.

NOTE: Due to its exciting versatility, the CPX will be used for many of the examples moving forward. It offers an extensive range of built-in features, including on-board sensors (light, temperature, motion, and sound), buttons, a speaker, and addressable LEDs. This means that you can get up and running quite quickly with little to no wiring. However, we are going to ignore these features for the moment and take an approach that is applicable across all three boards.

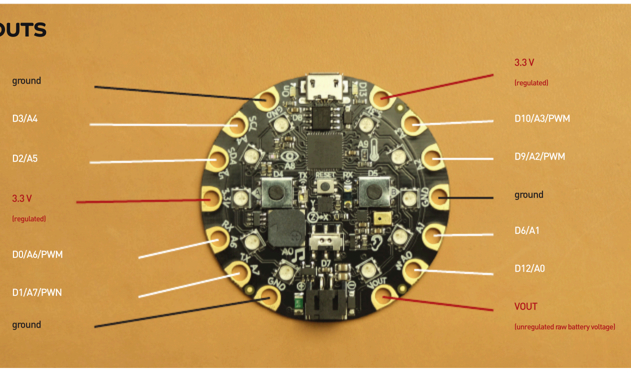
LILYPAD ARDUINO USB PINOUTS

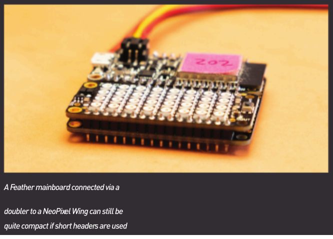


FLORA PINOUTS



CPX PINOUTS





A Feather mainboard connected via a

doubler to a NeoPixel Wing can still be

quite compact if short headers are used

Other Microcontrollers

Not all wearable circuits are made using conductive thread. Beyond the world of e-textile tool kits, there are many other microcontroller options. As of this writing, two families of microcontrollers I find to be quite useful when making wearables are the Arduino Nanos and the Adafruit Feathers.

The Arduino Nano series offers a wide range of functionality.

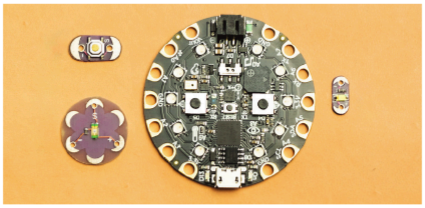
Arduino's Nano 33 BLE Sense board provides Bluetooth communication and many on-board sensors in a compact package.

for assembly.

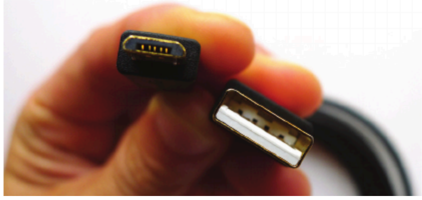
Arduino Nano 33 BLE Sense

The Arduino Nano IoT 33 allows easy access to Wi-Fi, making it easy to respond to or log live data. These boards can be purchased with and without headers. Because they include so many on-board features, without headers is sometimes a preferable format for wearables projects with minimal connections.

Adafruit Feathers are similarly compact. They offer mainboards with a wide range of chipsets as well as an extensive selection of Feather Wings that can be used in combination with mainboards. These boards can be stacked vertically with stacking headers or horizontally with a doubler or tripler breakout board. This creates the opportunity for bespoke functionality in a relatively small and stable form factor.



These parts will be useful for the examples in this chapter: push-button, LilyPad Light Sensor, sewable microcontroller, sewable LED



Micro USB cable

GETTING READY

To prepare for programming, you will need a micro USB cable. It's possible you already own one of these cables, as they come with many digital devices these days. But beware! Some of these cables are intended for *charging only*. For this purpose, you will need a higher-quality data cable. You also may need a dongle to connect the cable to your computer.

Once you've acquired a high-quality cable, connect the small end of your micro USB cable to the microcontroller and the larger end to your computer. Your hardware is ready to be programmed!

Software

DOWNLOADING AND INSTALLING ARDUINO

Now, it's time to get prepare your computer for programming in Arduino and compiling and uploading code to your specific type of microcontroller board. First, you need to download the Arduino software. You can find the version for your operating system on Arduino's software web page (arduino.cc/en/software). Instructions specific to your operating system are available on the help page "Downloading and Installing the Arduino IDE 2" (docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing).

Arduino does provide a cloud (web) editor. However, for the exercises in this book, stick to the downloadable application.

If you have worked with Arduino previously, be sure that you have the most up-to-date version of the software!

INSTALLING ADDITIONAL ITEMS

Depending on which board you are working with, you may need to install some additional items.

Because the LilyPad Arduino is a core Arduino board, it should work with the Arduino software as is.

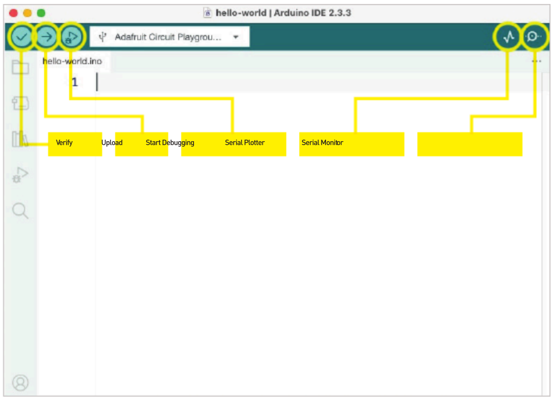
Because the Flora and the CPX are *Arduino-compatible* boards, they will require at least one extra step. Check out the "Getting Started with Flora" (learn.adafruit.com/getting-started-with-flora/overview) and "Adafruit Circuit Playground Express" (learn.adafruit.com/adafruit-circuit-playground-express/overview) tutorials in the Adafruit Learning System for the most up-to-date instructions on any drivers, bootloaders, and board profiles that might need to be installed.

Once you've completed these steps, quit and reopen the Arduino program.

GETTING STARTED IN THE IDE

What's an IDE? The term, which stands for "integrated development environment," refers to the Arduino program in which you write code.

Arduino files are referred to as *sketches*. When you open the Arduino software, you will see a blank sketch.



Blank Arduino sketch

As you mouse over the icons at the top of the window, you will see their various functions. They are as follows.

Verify

This checks the code and indicates whether there are any syntax errors.

Upload

This compiles the code and uploads it to the Arduino board. Once the code is uploaded, it will stay on the Arduino board even when it is unplugged from the computer.

Start Debugging

This allows you to move through the execution of a program, which can help you to better understand how it is working and possibly see ways in which it is not behaving as expected.

Serial Plotter

This opens a pane where you can see a visual representation of data being received.

Serial Monitor

This opens a pane where you can view a text output data being sent and received.

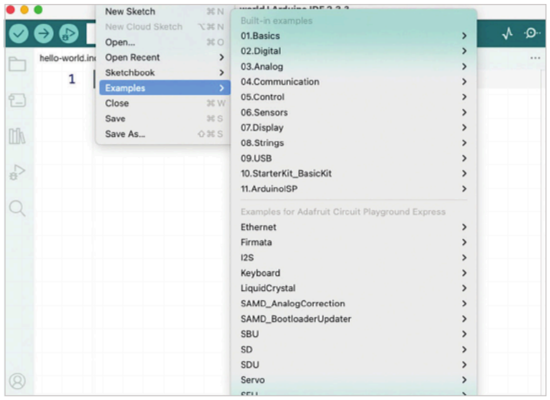
Here are some areas within the window that you will see as you work with the IDE:

- The white area is the text editor in which you will write your code.
- A white box will pop up in the lower right of the screen to provide status updates when your code is uploading.
- When you press Verify or Upload, a black area will open at the bottom

of the screen. This is a console where information about progress and errors will sometimes be displayed.

- The text in the bottom right-hand corner shows the currently configured board and serial port.

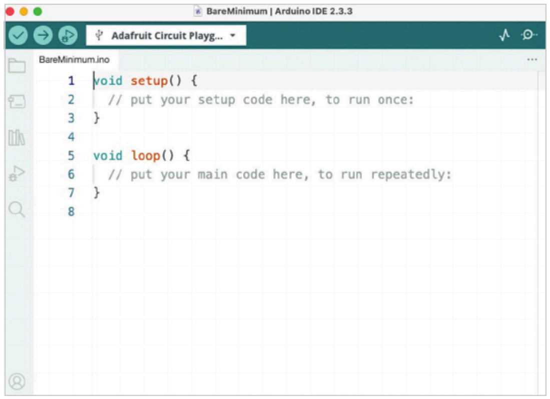
One of the nice things about working in Arduino is that there are many helpful code example sketches. To access them, go to File → Examples. This is where you can find a variety of examples to get you started.



File → Examples menu

In addition to the example code included with Arduino, you can find step-by-step explanations of these examples on the "Built-in Examples" help page (docs.arduino.cc/built-in-examples) on the Arduino website.

To look at the most basic possible Arduino sketch, go to File → Examples → 01.Basics → BareMinimum.



```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

```

BareMinimum example sketch

This is the requisite skeleton of any Arduino program—a great place to start when you are writing a new sketch. Just remember to save it as a different file name so you don't overwrite the example.

Here are some more things to know.

`setup()`

This is where you put commands that are to happen only when the program begins. These happen only once. The setup commands are contained within a set of curly brackets.

`loop()`

This contains commands that will happen over and over again. These are also contained within a set of curly brackets.

Comments

These are lines or blocks of text meant to be read by humans but not by the microcontroller. You will see comments both at the start of the program to provide notes, date, and attribution and throughout the program to explain what is happening along the way. Comments can also be used to remove lines of code that are not in use.

A single-line comment is preceded by two backslashes:

```
// This comment is on one line.
```

A multiline comment falls between a backslash and asterisk and an asterisk and backslash.:

```
/*
This comment
is on
multiple lines!
*/
```

Two great resources for better understanding the Arduino environment and syntax are the "Getting Started with Arduino IDE 2" guide (docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2) and Arduino's Language Reference (docs.arduino.cc/language-reference/).

While this book will review some programming strategies specific to wearables, it will not cover the details of Arduino programming. For more resources, check out appendix A.

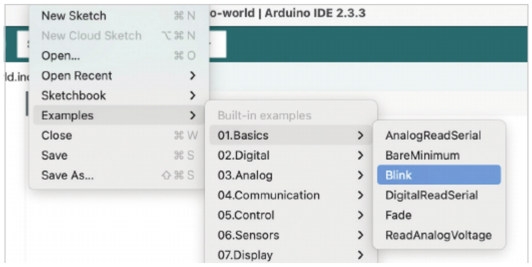
Hello World

"Hello World" is a term used to refer to the simplest possible program that can demonstrate that the system is working. In a typical computer program, this program would write the words "Hello World" to a display device. In Arduino, the equivalent of this is a blinking LED—a little something that lets the Arduino say, "Hey, here I am!" It also lets you know that your hardware and software are configured properly. Let's get your Arduino to say hello.

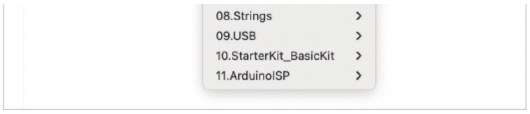
For the circuit, you don't need to do anything. Pretty much every Arduino and Arduino-compatible board has a built-in LED intended to be used expressly for this purpose.

For the code, there is an example sketch that will fit your needs nicely. Go to File → Examples → 01.Basics → Blink. This will open a sketch called "Blink."

Let's take a moment to look at the code example. This code uses a variable



called *LED_BUILTIN*. This refers to pin number 13, which is used for connecting to a built-in LED on many microcontroller boards. For the LilyPad USB and CPX, you can either leave the code as is or replace the three instances of *LED_BUILTIN* with *13*. The Flora, however, has the built-in LED connected to a different pin. If you are using a Flora, replace the three instances of *LED_BUILTIN* with *7*.



Opening the Arduino "Blink" example

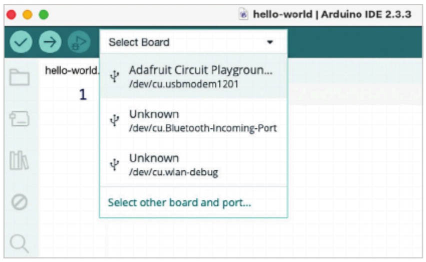
Once the sketch is correct, make sure everything is set up properly for this code to be uploaded to the Arduino. Always check three things before attempting to upload a program to your Arduino board:

- USB connection
- board type
- serial port

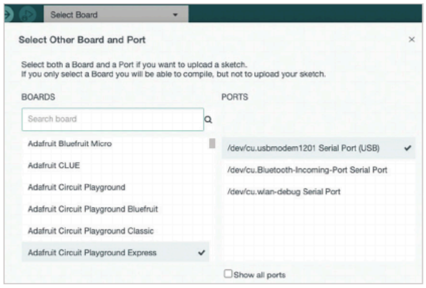
I call these the Magic 3. You should already have your board connected via USB, but it's worth checking. It may sound obvious, but in the midst of programming, you may have forgotten whether you've plugged your board in.

Setting the *board type* enables Arduino to compile the code in such a way that it will work properly on the type of Arduino board you have. If you do not use the correct board type, your code will not compile properly and you will get an error.

Setting the *serial port* tells the Arduino program which USB serial port to send the program to. This prevents the Arduino software from attempting to communicate with your mouse, your Bluetooth headset, or your USB-powered mini-fridge where you keep your emergency stash of Pocari Sweat and Club-Mate. Believe me—that conversation will not go well.


hello-world | Arduino IDE 2.3.3
Select Board
hello-world.
1
Adafruit Circuit Playgroun...
/dev/cu.usbmodem1201
Unknown
/dev/cu.Bluetooth-Incoming-Port
Unknown
/dev/cu.wlan-debug
Select other board and port...

Selecting the board type


Select Board
Select Other Board and Port
×
BOARDS
PORTS
Search board
Adafruit Bluefruit Micro
Adafruit CLUE
Adafruit Circuit Playground
Adafruit Circuit Playground Bluefruit
Adafruit Circuit Playground Classic
Adafruit Circuit Playground Express ✓
/dev/cu.usbmodem1201 Serial Port (USB) ✓
/dev/cu.Bluetooth-Incoming-Port Serial Port
/dev/cu.wlan-debug Serial Port
□ Show all ports

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

Selecting the board type and serial port is done in a drop-down menu in the toolbar, located to the right of the Start Debugging button. Click the Down arrow, click "Select other board and port," and from there, choose your board and port.

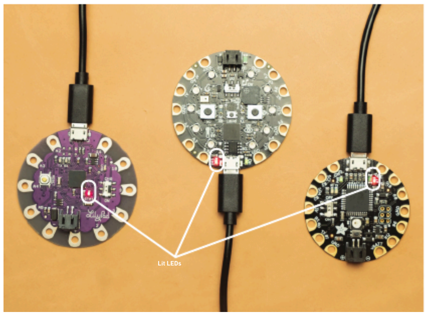
If your board is not showing up, you may need to double-check the Board Manager to make sure the board profile is installed.

If you are having trouble identifying the port, look at the list, unplug the device, watch to see which one disappears, and plug it in again and see if that same port reappears.

Important note: If you are working with the LilyPad Arduino USB, the on/off switch must be in the On position.

Once you've checked the Magic 3, you know you're ready to program the microcontroller board. Let's try to upload the Blink sketch to the

Selecting the serial port



Maxwell Lander

On-board LED lit on LilyPad USB, Flora, and CPX.

Congratulations—you've successfully uploaded your first sketch!

microcontroller board. Here's how:

1. Find the Upload button. It lives in the menu and is marked with an arrow pointing to the right.
2. Press it!
3. Watch the status update in the bottom-righthand side of the screen. It should change from "Compiling sketch" to "Done compiling" to "Uploading" to "Done uploading." If you run into any snags, check out the "Troubleshooting Arduino Sketches" web page (docs.arduino.cc/learn/starting-guide/troubleshooting-sketches).
4. Look at the built-in LED that lives on your microcontroller board. It should be blinking. Hello, Arduino!

EXPERIMENT: Gettin' Blinky

For your "Hello World" exercise, you jumped right in and made no or minimal adjustments to the code. You'll take a close look at digital output in the next section, but in the meantime, dip your toes in the water of code by making a few minor tweaks.

Take a look at the Blink example and find the lines that say this:

```
delay(1000);
```

This is just what it suggests—a function that delays the program for a set amount of time before it performs its next command. The number in parentheses is the length of the delay in milliseconds. (A thousand milliseconds equals one second.) Each delay used in this code is one second long.

Try changing the length of one of the delays. Upload the code to the Arduino board again and take a look at the behavior of the LED. Has the timing of its blink changed?

Spend some more time playing with the delay values and see what results you get!

Digital Output

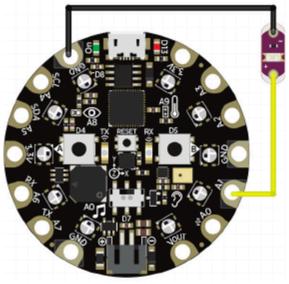
Lighting the built-in LED is enough to make you say, "Yay!" but you'll likely want to experiment with different LEDs soon after. Let's look at ways to add another LED.

THE CIRCUIT

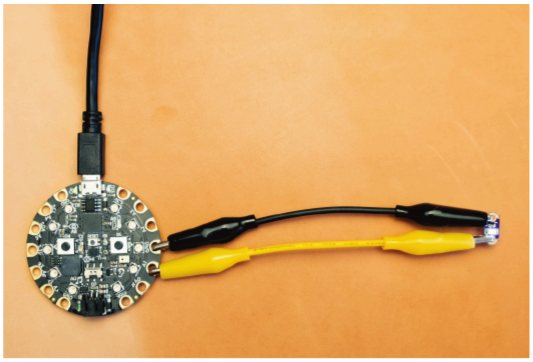
If the pin of the built-in LED pin were physically accessible as a sew tab on the board, so let's work with pin A1 instead. First, try connecting a sewable LED. These are nice because they have everything you need in a compact package. Using alligator clips, make the following connections.

CPX	SEWABLE LED
Pin A1	Power (+)
Ground (–)	Ground (–)

You can also use a through-hole LED.

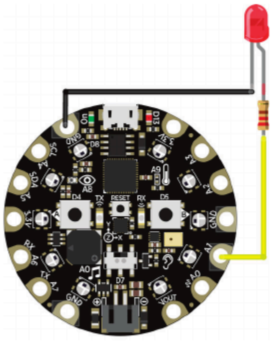


Circuit diagram of CPX with sewable LED on pin A1

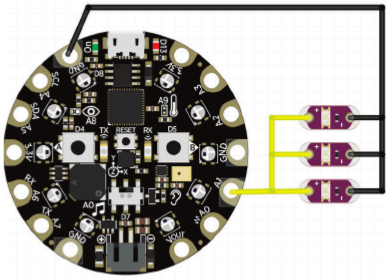


Maxwell Lander

CPX with sewable LED on pin A1



Circuit diagram of CPX with through-hole LED and 220 Ω resistor on pin A1



Created in Fritzing

Circuit diagram of CPX with 3 LEDs in parallel controlled by pin A1

Because the output pins on these boards can often supply more mA of current than a single LED requires, you can typically connect multiple LEDs in parallel. The total number of LEDs that will light in parallel will depend on the type of LED and the total amount of current provided by the output pin. When LEDs are controlled by a single pin, they will have the same behavior. When working with sewable LEDs, the resistor is included, so you don't need to add another.

THE CODE

Let's take a closer look at that Blink example. Here are some helpful things to know.

Variables

These provide a way to name and store values. This could be a changing value, like a reading from a switch or sensor, or a constant value, like a particular pin number you will be using throughout the program. Variables are useful because if you decide to make a change, for example, to which pin you are using for an LED, you have to make the change in only one place in your code (the point at which you define the variable, rather than every time you refer to the LED pin number).

You can read about many variable types in the Arduino Language Reference. In the Blink example, we learned that a variable called *LED_BUILTIN* is used to refer to pin number 13. This is a variable that is included in the Arduino IDE. When we create our own variables, we will need to initiate them; this is covered in the examples that follow:

`pinMode(pin, mode)`

This sets a digital pin as either an input or an output. The two parameters needed are the number of the pin and its mode (i.e., **`INPUT`** or **`OUTPUT`**). This command is included in the setup so that the pin's behavior is determined at the start of the program.

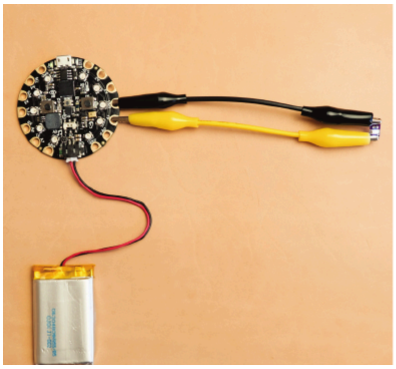
`digitalWrite(pin, value)`

The command is used to control a digital output pin. The first parameter is which pin you would like to address. The second is the value, which can either be **`HIGH`** or **`LOW`**. **`HIGH`** will turn the pin on, sending out V+, in this case 3.3 V. **`LOW`** will turn the pin off.

Now that you have a better understanding of what's going on in the Blink example, we'll create our own variable, called **`LEDpin`**, to store a pin number:

```
int LEDpin = A1; // initialize the variable

void setup() {
  pinMode(LEDpin, OUTPUT); // set the pin mode
}

void loop() {
  digitalWrite(LEDpin, HIGH); // turn LED on
  delay(1000);
  digitalWrite(LEDpin, LOW); // turn LED off
  delay(1000);
}
```

Upload your new code, and voilà! The newly connected LED should light accordingly.



POWER

When you first upload and run these examples, your board will be receiving power from your computer via USB. But what if you unplug it?

One nice attribute of these microcontroller boards is that they include a JST connector for battery connections. A 3.7 V rechargeable battery or a three-battery AA or AAA battery pack is a good option for most of the circuits

CPX circuit powered by battery connected to the JST connector

Maxwell Lander

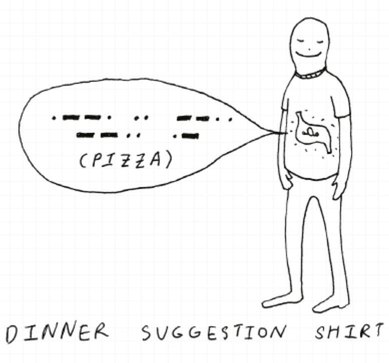


Illustration by Jen Liu

covered in this book.

EXPERIMENT: Morse Code Messages

Even a simple blinking LED can take on great meaning under the right circumstances.

Morse code is a method of transmitting messages with short and long pulses of sound or light. A dash (long pulse) is usually three times the length of a dot (short pulse).

Using the table below, write a program that sends a message by way of the blinks of the LED. Think about what it would be like if you mounted this LED on a piece of your clothing. What would you want it to say?

Morse Code Translation Guide

CHARACTER	CODE	CHARACTER	CODE	CHARACTER	CODE
A	._	J	.___	S	...
B	_...	K	_._	T	_
C	_._.	L	._..	U	.._
D	_..	M	__	V	..._
E	.	N	_.	W	.__
F	.._.	O	___	X	_.._
G	__.	P	.__.	Y	_.__
H	....	Q	__._	Z	__..
I	..	R	._.		

Digital Input

Up to this point, you've been working exclusively with outputs. But in Microcontrollerland, it is important to understand the difference between outputs and inputs.

Outputs are pins where information is delivered *from* the microcontroller in the form of varying voltage. Various types of actuators (e.g., LEDs, motors, and speakers) can be connected to output pins and will use the voltage to perform different actions: LEDs will light up, motors will spin, and buzzers will beep.

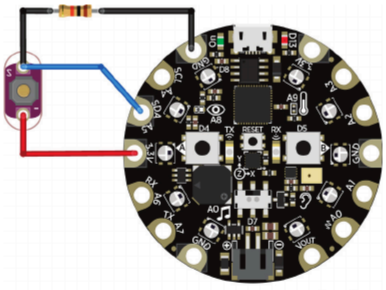
Inputs are pins where you can connect devices that supply information *to* the microcontroller. Such devices include switches and various types of sensors. Information is fed to the microcontroller in the form of varying voltage.

Now that you've gotten some experience with digital outputs, let's give digital inputs a try. Based on what you learned in chapter 4, you have a good idea of what a switch is, how it works, and what types are available to you. But how do you connect them to a microcontroller?

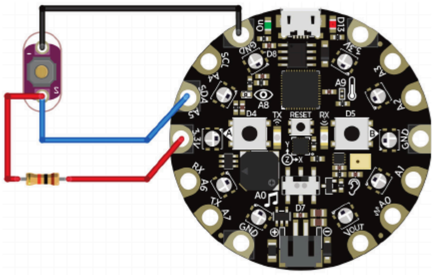
THE CIRCUIT

When connecting a switch to a microcontroller, you can connect it from any digital input pin to either power (+) or ground (–), depending on what kind of logic structure you want to create. If the switch is connected to power, the pin will read "HIGH" when the switch is closed and "LOW" when it's open. If the switch is connected to ground, the logic will be reversed ("LOW" when closed, "HIGH" when open).

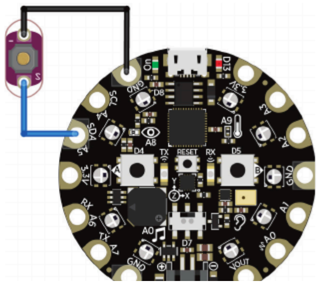
Connecting the switch is not enough to complete your circuit. When the switch is closed, you will have a solid connection to power or ground, depending on how you've wired it. But when the switch is open, the input pin will *float*. A floating pin has no reliable reference and thus can produce erratic values that will likely interfere with the reliability of your program. The way to prevent floating pins is with a *pull-up* or *pull-down* resistor. This resistor is of a large enough value that when the switch is closed, current will follow the path of the switch, but when it is open, it will act as a spring that gently pulls the input back to its resting state.



Circuit diagram of CPX with switch and pull-down resistor



Circuit diagram of CPX with switch and pull-up resistor



Fritzing

If the switch is connected to power, you can use a pull-down resistor connected from the digital input pin to ground. If the switch is connected to ground, use a pull-up resistor connected to power. Within this context, something in the range of a 10K Ω resistor will usually do the trick.

If you want to reduce the amount of wiring you have to do, the LilyPad's ATmega chip has an internal pull-up resistor on the digital pins you can activate with the command **`pinMode(pinNumber, INPUT_PULLUP)`**.

Whichever method you choose, these will all allow the value of the switch to be read by a digital input pin on the microcontroller.

For this example, let's use the circuit with the internal pull-up resistor.

THE CODE

Now that you have a switch connected, how can you write a program that can tell what the switch is doing? Here are a few more Arduino commands that will help you read the value of a digital input:

`pinMode(pin, mode)`

This is something you encountered earlier with digital output. Generally speaking, with digital input, you would set the mode to **`INPUT`**. This will work with circuit examples that use external pull-down or pull-up resistors. However, if you would like to use the *internal* pull-up resistor, then set

Circuit diagram of CPX with switch wired for use with internal pull-up resistor; you can also use two alligator clips without the button and simply connect and disconnect the exposed clips at the loose ends.

the mode to **INPUT_PULLUP**.

digitalRead(pin)

This is the opposite of the **digitalWrite()** command. Rather than controlling a pin by sending voltage out, this allows you to read the voltage coming into a pin. The only parameter you need to provide is the pin number. However, you do need a place to store the information that is read, so this command is usually used in combination with a variable—for example, **buttonState = digitalRead(buttonPin);**.

In order to read values coming into the microcontroller, you need to print it to some sort of display. Because the Arduino has no built-in visual display, you can use USB-serial communication and the Serial Monitor in the Arduino software to view what sort of values you're getting in. The new commands you need to know to accomplish this follow:

Serial.begin(speed)

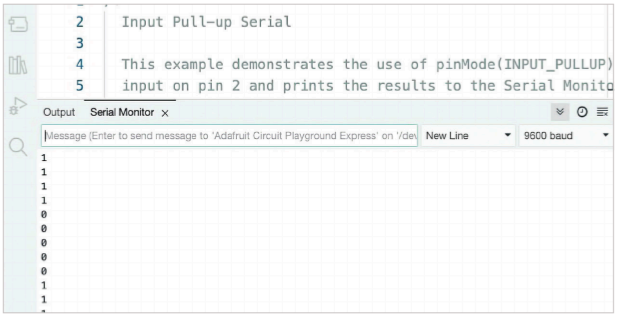
By including this in the **setup()** function, you initializes the serial connection and set the speed of communication. A standard rate you'll often find in examples is 9600 baud. This will work well when communicating between your Arduino and your computer.

Serial.println(val)

This command transmits a value followed by a carriage return (a character sent when you press Enter or Return). In your case, the value will be the switch value.

Now that you understand what's going on, run this code:

```
/*
Make: Wearable Electronics
Digital Input example
 */
// variable for the digital input pin
int buttonPin = A5;
// variable for the reading from the button
int buttonValue = 0;

void setup() {
  // initialize serial communication at 9600 bps
  Serial.begin(9600);
  // set pin as input
  // use internal pull-up resistor
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  // read input pin:
  buttonValue = digitalRead(buttonPin);
  // print button value:
  Serial.println(buttonValue);
  delay(100);
}
```



```
Input Pull-up Serial

This example demonstrates the use of pinMode(INPUT_PULLUP)
input on pin 2 and prints the results to the Serial Monito
```

Output
Serial Monitor ×
Message (Enter to send message to 'Adafruit Circuit Playground Express' on '/dev
New Line
9600 baud

```
1
1
1
1
0
0
0
0
0
1
1
```

Button values as seen in the Arduino Serial Monitor.

Upload the sketch to your Arduino board. Then, while the board is still connected via USB, open the Serial Monitor, and you will see the switch values change as you press and release the button.

When you press the button, the value should change to 0. Otherwise, it will be 1.

See also Arduino's "Digital Read Serial" page (docs.arduino.cc/built-in-examples/basics/DigitalReadSerial).

You've now conquered digital input *and* serial communication in one fell swoop. With that under your belt, let's try a new experiment.

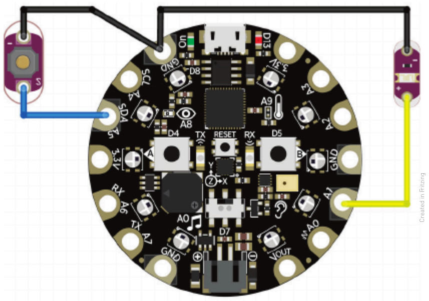
EXPERIMENT: Button as Controller

Now that you know how to work with both inputs and outputs, you can create a relationship between them. Control structures in the Arduino syntax allow you to establish such relationships.

First, let's put together a circuit that includes both an input and an output.

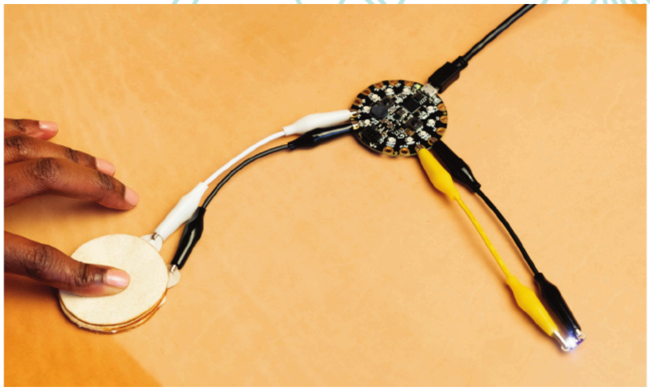
Now, you just need to connect them in the code. The simplest structure to work with is the if statement. It goes something like this:

```
/*
Make: Wearable Electronics
Button as Controller example
*/

// variable for the digital input pin
int buttonPin = A5;
// variable for the reading from the button
int buttonValue = 0;
// variable for the digital output pin
int LEDpin = A1;

void setup() {
 // set pin as input & use internal pull-up resistor
 pinMode(buttonPin, INPUT_PULLUP);
 // set the pin mode
 pinMode(LEDpin, OUTPUT);
}

void loop() {
 // read input pin:
 buttonValue = digitalRead(buttonPin);
 // if button reads high
 if (buttonValue == HIGH) {
   // turn LED on
   digitalWrite(LEDpin, HIGH);
 }
 delay(100);
}
```



Created in Fritzing

CPX with digital input and digital output



Maxwell Lander

The soft button created in chapter 4 can be used as a digital input for the CPX to control the LED connected to the digital output in different ways.

Within this if statement, "HIGH" means 1 and "LOW" means 0. Keep in mind that with a pull-down circuit, the switch will read "HIGH" when it's *closed*. With a pull-up circuit, it will read "HIGH" when it's *open*.

The only problem with this is that once a program starts, if the switch ever reads "HIGH," the LED will stay on forever. There are no instructions if the switch is LOW. In most cases, you will need to use an else clause, which provides instructions for what to do if the initial condition isn't met:

```
// if button reads high
if (buttonValue == HIGH)
```

```
{
// turn LED on
 digitalWrite (LEDpin, HIGH);
// otherwise
} else {
// turn LED off
 digitalWrite (LEDpin, LOW);
}
```

Using this information and the code examples provided in the digital input and output sections, create a program that allows the button to control the lighting of the LED.

Once that's up and running, try switching the logic so the relationship between the button press and the LED light is reversed.

Once that's working, try creating a program that causes the LED to blink while the button is pressed.

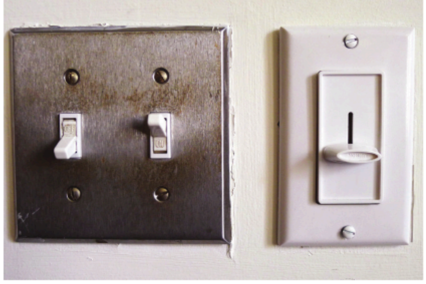
Analog Input

Another key concept when getting to know your microcontroller pins is the difference between digital and analog.

Digital refers to a binary state: On, or off. High, or low. Voltage flowing, or not flowing. 1, or 0. There are only two possible states. There are both digital inputs (such as a switch) and outputs (which could turn an LED on or off).

Analog refers to pins that can accommodate a range of values. With analog inputs, you can connect sensors such as a light sensor that can tell you whether it's light, dark, or somewhere in between. With an analog output, you can accomplish more varied effects, such as an LED that can fade from on to off.

When trying to understand the difference between digital and analog inputs, you can think about the traditional interface devices for home lighting. A regular light switch on a wall is similar to a digital input. It can turn the lights only on or off. But a dimmer is similar to an analog input: It provides enough information so you can tweak the lighting level to create a specific mood.

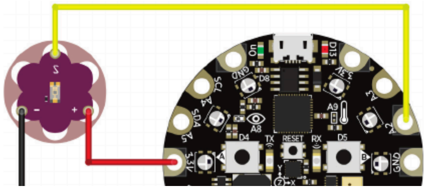


The on/off switches on the left would be considered a digital input; the dimmer on the right would be considered an analog input.

LilyPad Light Sensor

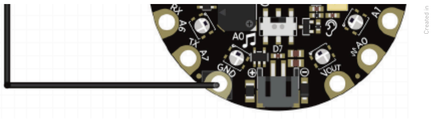
THE CIRCUIT

A common sensor to get started with for analog input is a light sensor such as a phototransistor. It just so happens that there is a LilyPad Light Sensor available. This sensor will output between 0 and V+ depending on the light level it senses,



Fritzing

with 0 V indicating the darkest and V+ indicating the brightest.



CPX with LilyPad Light Sensor on pin A2

If you look closely at the LilyPad Light Sensor, you will see that the pins are marked with +, –, and *S*. This gives you some hints about how to connect your light sensor to your LilyPad Arduino. Make the connections using alligator clips.

On the LilyPad Light Sensor, *S* stands for the signal being produced based on the light levels: between 0 and V+. On the LilyPad Arduino Simple (or any Arduino), when the pin number is proceeded by an *A*, it indicates that the pin is an analog input pin. For this example, you could also use pin A3, A4, or A5, but you would need to adjust the code accordingly.

Once your circuit is complete, connect your microcontroller to your computer, then get yourself ready to program.

THE CODE

The code for reading an analog input is quite similar to that for a digital input, with the exception of this command:

`analogRead(pin)`

This reads the value of a specified analog pin. The pin can be referred to as just the number (for example, 2) or with the "*A*" preceding it (for example, A2).

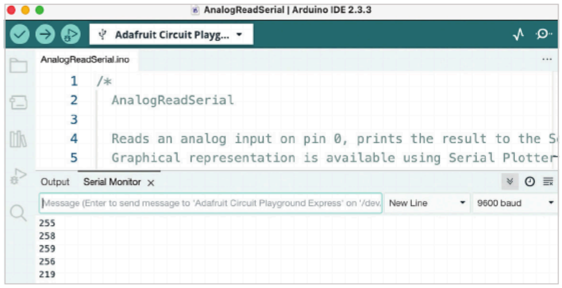
`pinMode()` does not need to be set for an analog input pin.

Here's the code:

```
/*
Make: Wearable Electronics
Analog Input example
 */

// initialize variable for light sensor reading
int lightSensorValue = 0;

// initialize variable for light sensor pin
int lightSensorPin = A2;
```

```
void setup() {
  // initialize serial communication at 9600 bps
  Serial.begin(9600);
}

void loop() {
  // read pin and store value in a variable:
  lightSensorValue = analogRead(lightSensorPin);
  // print the light sensor value:
  Serial.println(lightSensorValue);
  // delay between readings:
  delay(100);
}
```

Check your board type and serial port, and upload the code. Open your Serial Monitor and make sure your baud rate is set to 9600, and you should see sensor values on the screen!


AnalogReadSerial | Arduino IDE 2.3.3
Adafruit Circuit Playg...
AnalogReadSerial.ino
Output
Serial Monitor
Message (Enter to send message to 'Adafruit Circuit Playground Express' on '/dev/
New Line
9600 baud

```
/*
  AnalogReadSerial

  Reads an analog input on pin 0, prints the result to the S
  Graphical representation is available using Serial Plotter
```

```
255
258
259
256
219
```

Light-sensor values as seen in the Serial Monitor

Notice how as you cover and uncover the sensor, the values on screen change. Try moving the circuit toward a very bright light and try covering it completely As the light gets brighter, the values should go up, and as it gets darker, they should go down. Find out the broadest range of values you can observe.

This shows you in a very basic form how to read sensor values with the Arduino. The following chapter goes more in-depth into what sensors are and how to work with them.

See also these Arduino tutorials:

- "Analog Read Serial" (docs.arduino.cc/built-in-examples/basics/AnalogReadSerial)
- "Analog Read Voltage" (docs.arduino.cc/built-in-examples/basics/ReadAnalogVoltage)

- "Analog Input" (docs.arduino.cc/built-in-examples/analog/AnalogInput)

EXPERIMENT: Sensor as a Switch

Sensors can act as switches too. This is this snippet of code you used back in the experiment "Button as Controller" on pages 141–142, to allow a switch to control an LED:

```
if (buttonValue == HIGH) // if button reads high
{
  digitalWrite (LEDpin, HIGH); // turn LED on
} else { // otherwise
  digitalWrite (LEDpin, LOW); // turn LED off
}
```

You can modify this for use with a sensor. For example:

```
if (lightSensorValue > 500)
// if light sensor reads greater than 500
{
  digitalWrite (LEDpin, HIGH); // turn LED on
} else { // otherwise
  digitalWrite (LEDpin, LOW); // turn LED off
}
```

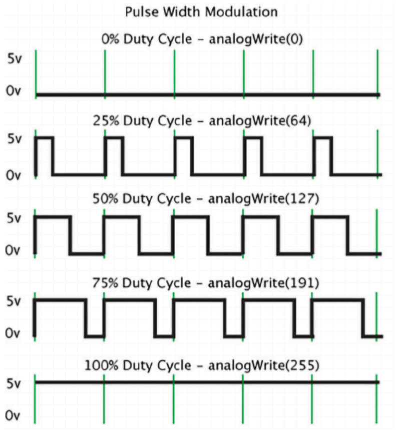
Modify the analog-input example so that changes in light levels will control the LED. Use the values you see in the Serial Monitor to determine the appropriate value to use in your code.

Analog Output

On the output end, analog allows you to provide a range of values rather than simply turning something on or off. This means you can brighten or dim an LED with subtlety or even control the speed of a motor.

But whereas an analog input pin reads a set range of voltages, despite what you might think, an analog output pin does not produce a range of voltages. Instead, it *simulates* a change in voltage to create an analog effect by pulsing 5 V in differing duty cycles. This effect is called pulse width modulation (PWM). If the pin is quickly switched back and forth between 0 V and 5 V, it creates the effect as if it were outputting 2.5 V, and so on.

Arduinos have limited pins that are able to perform PMW. On an Arduino Uno, they are marked with a tilde (~). On some boards, they are marked, but if not, you can always check the


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

Arduino

Pulse width modulation—comparing duty cycles and perceived LED brightness: off (0%), dim (25%), half brightness (50%), full brightness (100%)

pinout diagram, datasheet, or product page.

THE CIRCUIT

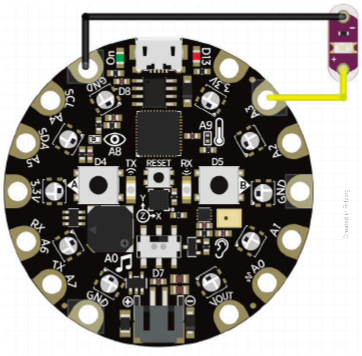
For this circuit, we'll switch to a pin with PWM as indicated by the pinout diagram.

THE CODE

The command you use to control analog output is this:

```
analogWrite(pin, value)
```

The pin is the number of the pin you'd like to control. The value can be between 0 and 255, with 0 being 0 V and 255 being V+. If you would like to do something like brighten and dim an LED, you can incrementally move it through different values.



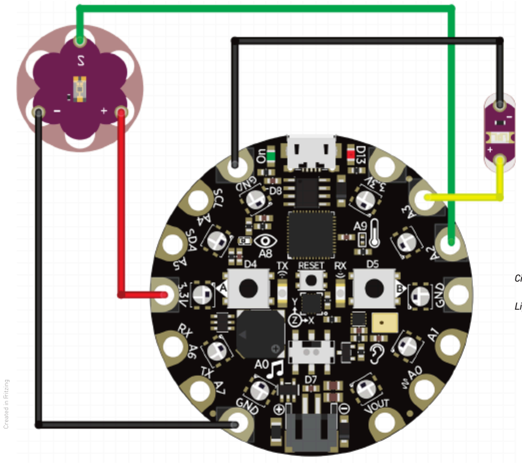
Circuit diagram of CPX with sewable LED on pin A3

Upload this code to see the LED turn on at a variety of brightnesses:

```
/*
Make: Wearable Electronics
Analog Output example
 */

int LEDpin = 10; // LED is connected to pin A3

void setup()  {
  pinMode(LEDpin, OUTPUT); // sets pin as output
}

void loop()  { // LED completely off
  analogWrite(LEDpin, 0);
  delay(100);
  analogWrite(LEDpin, 50);
  delay(100);
  analogWrite(LEDpin, 100);
  delay(100);
  analogWrite(LEDpin, 150);
  delay(100);
  analogWrite(LEDpin, 200);
  delay(100); // LED at full brightness
  analogWrite(LEDpin, 255);
  delay(100);
}
```

This is a very simple way to start out working with analog output. By employing more complex programming methods, you can achieve more sophisticated behaviors.

See also Arduino's "Fading" tutorial (docs.arduino.cc/built-in-examples/analog/Fading/).



Circuit diagram of CPX with the LilyPad Light Sensor and sewable LED

EXPERIMENT: Sensitive System

Many basic interactive projects create a relationship between the values of an analog input and the values of an analog output. Create the circuit in the image above, which includes both a light sensor and an LED.

Using if statements and the **`analogWrite()`** command, use the brightness of an LED to reflect the changes in the values of a light sensor.

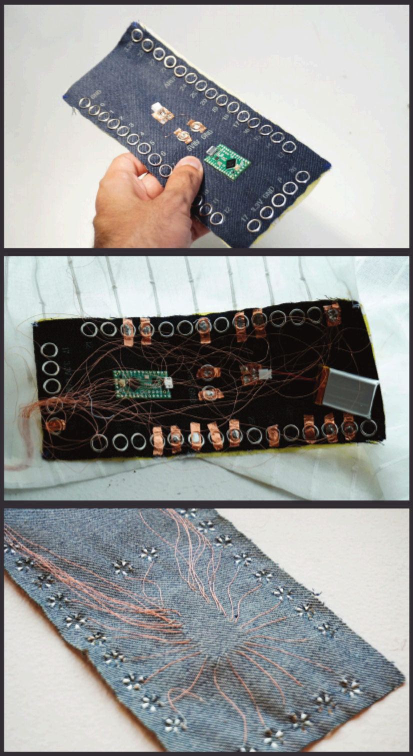
Looking Ahead

This is the most basic of introductions to working with microcontrollers by using the LilyPad Arduino Simple. As you begin to develop projects, keep in mind that there is a broad range of Arduino and Arduino-compatible products out there that might better meet your needs. You will encounter some of them in the coming chapters. You'll also further explore the plethora of sensors and actuators available for use in combination with microcontrollers for your wildly imaginative wearable-electronics projects.

GALLERY 6:

MICROCONTROLLERS IN WEARABLES

Microcontrollers can be worn in many ways. These projects demonstrate e-textile microcontrollers in use, how other microcontrollers can be adapted for use in textiles and wearables, and how on-board features of microcontrollers can be super handy in a wearable context.

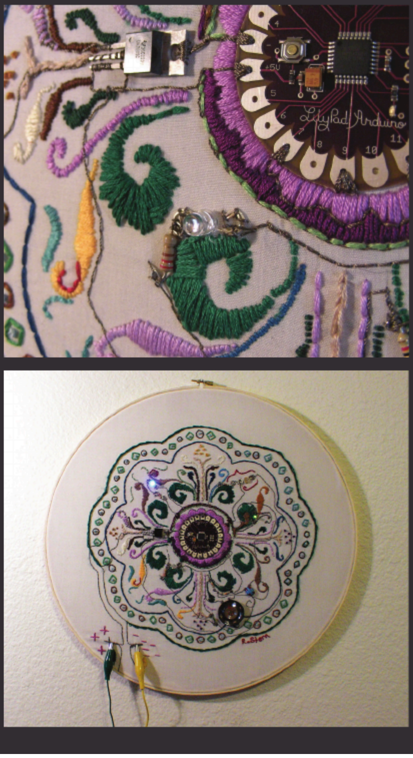


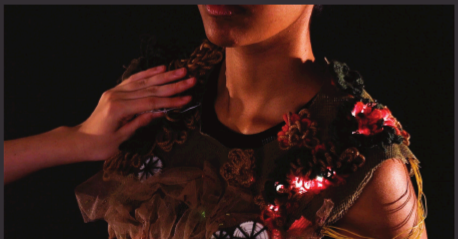
Teensy Soft Breakout Board

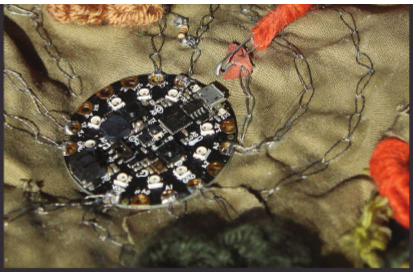
The Teensy Soft Breakout Board, by Salil Parekh, is intended to make it easier to test and debug the small board when using it for e-textiles projects. It's created using fabric scraps and magnet wire. Each snap is connected to a pin, and its larger size allows it to be easily used with crocodile clips while prototyping.

LilyPad Embroidery

LilyPad Embroidery: A Tribute to Leah Buechley, by Becky Stern, is an embroidery sampler that uses traditional floss and techniques combined with conductive thread, a LilyPad Arduino, sensors, and actuators. The amount of light sensed by the sensor changes the speed and pitch of the lights and sounds generated.

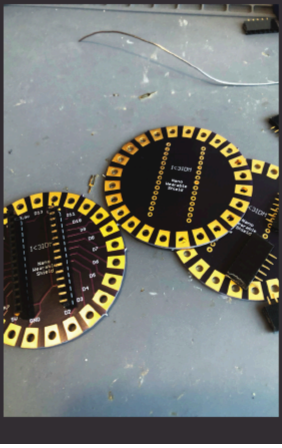


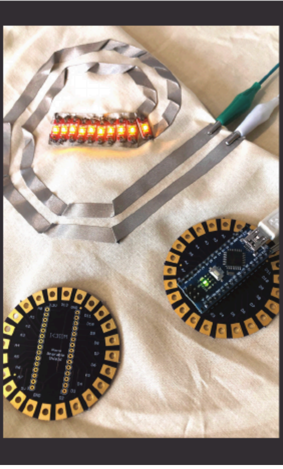




Neo-TCH 1.3

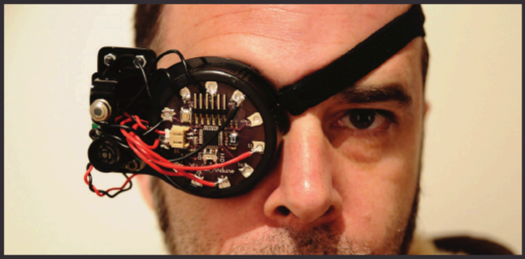
Neo-TCH 1.3, by Anna Pragman, Krithi Nalla, and Tanvi Mishra, is a speculative collection of wearable-tech garments designed to foster nonverbal communication through tactile interaction. The garments feature handmade, soft e-textile sensors, including stroke-and-tap interfaces, which invite playful, intuitive engagement. Utilizing the microcontroller's built-in infrared sensors, the garments encourage connection between the two wearers, facilitating a dynamic, gesture-based dialogue.





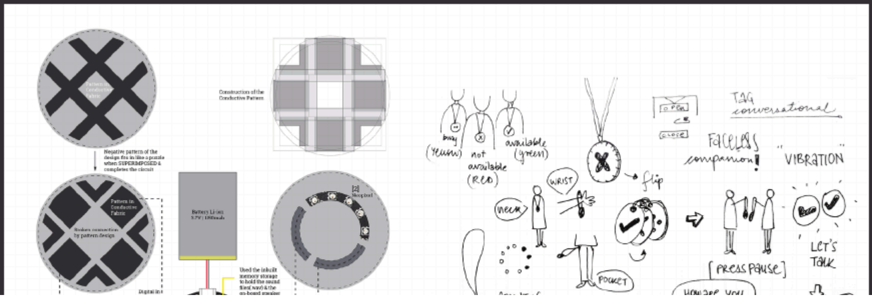
IDM Nano Wearable Shield

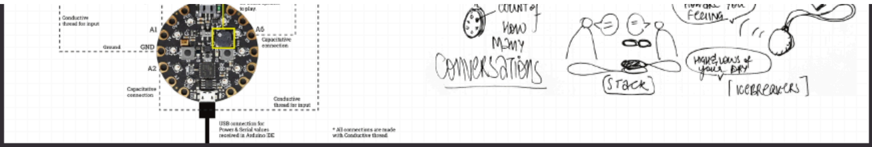
The Nano Wearable Shield, by the Wearables Club in the Integrated Design and Media Department at New York University, is a breakout board for Arduino Nano variants that allows connections to be made using conductive thread or conductive paint. It is an open source hardware project that builds on SparkFun's Photon Wearable Shield design, which itself was inspired by Leah Buechley's Lilypad Arduino design.



Augmented Vision

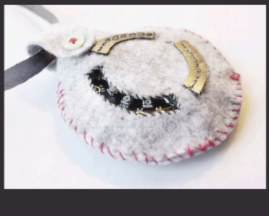
Augmented Vision, by Greg McRoberts, is a wearable seeing aid device that uses flashing RGB (red, blue, green) LEDs to represent fluctuating data gathered by an infrared heat sensor and an ultrasonic distance sensor.

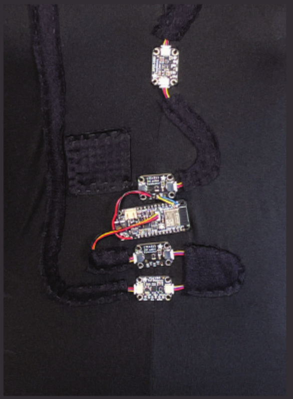

Pattern in Conductive Fabric
Construction of the Conductive Pattern
Negative pattern of the design fits in like a puzzle when SUPERIMPOSED & completes the circuit
Pattern in Conductive Fabric
Broken connection by pattern design
Battery Li-ion 3.7V | 150mah
[2] Neopixel
Used the inbuilt memory storage to hold the sound files(.wav) & the on-board speaker
Digital in
busy (yellow)
not available (RED)
available (green)
open
close
TAG conversational
FACELESS companion!
"VIBRATION"
flip
WRIST
neck
POCKET
[PRESS PAUSE]
LET'S TALK
HOW ARE YOU


to play.
Conductive thread for input
A1
A6
Ground
GND
Capacitative connection
A2
Capacitative connection
Conductive thread for input
USB connection for Power & Serial values received in Arduino IDE
* All connections are made with Conductive thread.
COUNT of How Many Conversations
How are you feeling
STack
Highlights of your Day
[ICEBREAKERS]

TalkTags

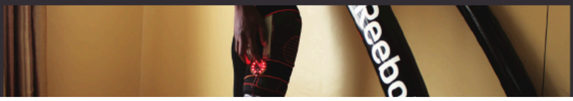
TalkTags, by Manisha Laroia, are social wearables that help people engage in conversations. When the conductive-fabric designs of two badges are connected, users can cycle through different conversation starters to invite their companion to join a real-world chat. The conversation starters are spoken by recorded voice files programmed on the CPX.





Body Theremin

Body Theremin, by Linh My Truong, turns the body into an instrument inspired by the theremin, a musical instrument played without touching it. Wearing a garment with embedded sensors, a participant can manipulate sound through specific body movements and gestures. The main components include an ESP32 Feather V2 microcontroller, accelerometers, a proximity sensor, a time-of-flight distance sensor, I2C extenders, NeoPixels, and Stemma QT cables.

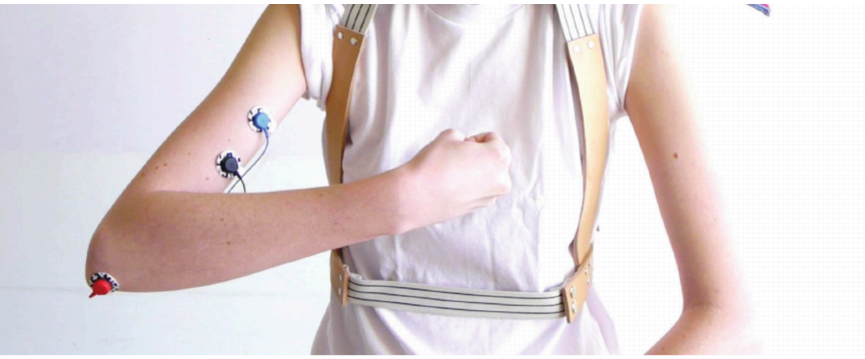


Vibrating Knee Brace

The Vibrating Knee Brace, by Patricia Mwenda, uses a CPX to provide therapeutic vibration during mild exercise like walking and postworkout stretches. The knee brace operates in four distinct modes, indicated by the color of the NeoPixels. The user can switch between modes by pressing the appropriate button on the CPX. In addition, an accelerometer-based mode automatically activates vibration in intervals during motion.

7

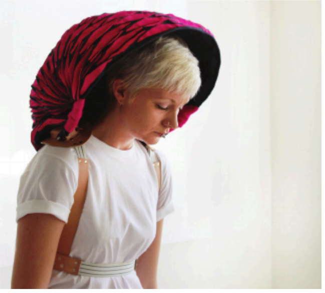
Sensors



Monarch V1 (top) & Nautilus (bottom), by Social Body Lab (Kate Hartman, Boris Kourtoukov, Erin Lewis, Hillary Predko, Jackson McConnell, Erica Charbonneau)

At Social Body Lab, one of our favorite sensors is an electromyography (EMG) sensor. We've used this sensor to create a variety of muscle-activated kinetic textiles.

Monarch is a wearable for personal expression. Pleated textile forms that live on the shoulders expand and contract in response to the movement of the wearer's bicep muscle. These shape-shifting body extensions can be used to express excitement, emphasis, or even aggression.



In contrast, Nautilus allows the wearer to turn inward. A large pleated hood can be raised and lowered in response to the muscle activity of the wearer shrugging their shoulders.

Through the use of sensors like these, existing body language can be augmented in new and creative ways.

Adobe Stock-Lepusinensis

As you get deeper into the realm of interactivity, it's worth considering the wide range of sensors available to you. In this chapter, you'll take a moment to consider how you can best listen to what's happening in, on, and around the body through technological means. You'll encounter an assortment of body-centric sensors and look at some simple ways to work with the data they produce.

Working with Sensors

There are both conceptual and technical factors to consider when working with sensors.

GETTING TO KNOW YOUR SENSOR

Looking at a sensor that is new to you can be intimidating, exciting, or overwhelming. It is easy to look at the name of a newly released sensor, think, *That's exactly what I need!*, order it, get it home, and realize that it is incompatible with your project.

Just as in any good relationship, it's worth sniffing out your prospective sensor before making the big commitment. You can always, at the very least, have a virtual introduction to it through datasheets, product descriptions, reviews, forums, and tutorials. And if you're part of a hacker, maker, or educational community, you may very well know someone who has the sensor you're considering. Ask if you can borrow it and take it out for a spin before taking the plunge and getting one of your own. While electronic sensors are significantly cheaper than they used to be, they're still an investment, so it's worth doing your research.

When encountering a sensor for the first time, whether it be online, in a catalog, or in real life, here is the type of information you should be looking for:

Voltage requirements

Some sensors will work only with 3.3 V and some only with 5 V, while others are able to handle both or something in between. Check the sensor's documentation before connecting power to a new one.

Connector type

Depending on the manufacturer, breakout board, and intended use, sensors will have different connection types. If there are headers, are they male, or female? Do they use breadboard spacing, or something else? Does the sensor have sew tabs because it is intended for use with conductive thread? Is it a standardized plug-and-socket set such Stemma, Stemma QT, Qwiic, Grove, or other flavors of JST connectors? Or is it a proprietary connector specific to the manufacturer? These factors will likely determine what other materials you will need to build your circuit.

Sensitivity

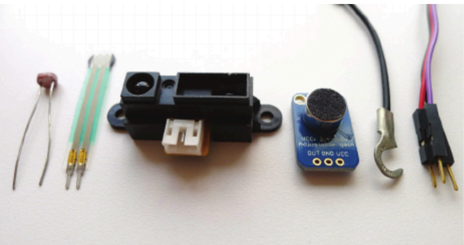
What is the range of values your sensor senses? For instance, smaller force-sensing resistors (FSRs) can sense as little as 2 grams of force, whereas larger ones have a sensing range of 100 grams to 10 kilograms.

Accuracy

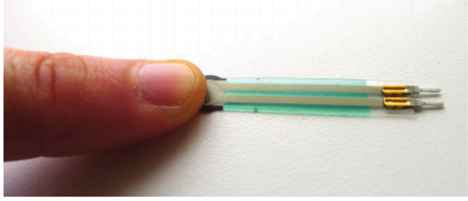
How accurate is your sensor? FSRs are very sensitive but not always accurate. Depending on their range of sensitivity, they provide good answers to questions like "Is this being pressed, or not?" or "Is this being stood on, or not?" but they do not provide the level of accuracy needed if you were building a precise food scale or a body scale.

Shape, size, and weight

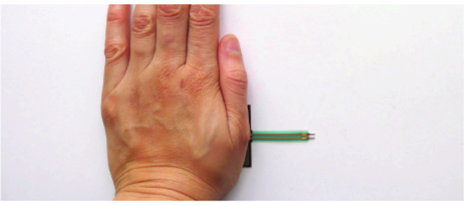
What are the physical characteristics of your sensor? These greatly affect how a component can be worn on the body. Be sure to look at dimensions and technical drawings to determine whether a sensor will fit in with your design.



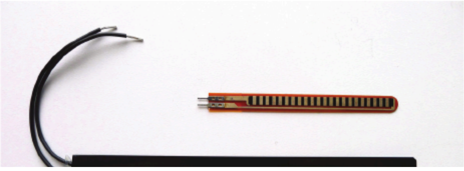
Sensors with different connectors—left to right: legs, terminals, JST connector, pins, hook, male headers

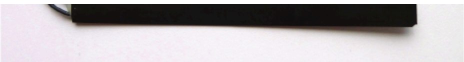


This smaller force-sensing resistor is sensitive to even the lightest touch of a finger.



This larger force-sensing resistor is sensitive to a higher range of pressure, making it appropriate to be pressed firmly, leaned on, or stepped on.





Flex sensors come in different lengths; a shorter flex sensor might be more appropriate for a project with a limited surface area.

Sensor output

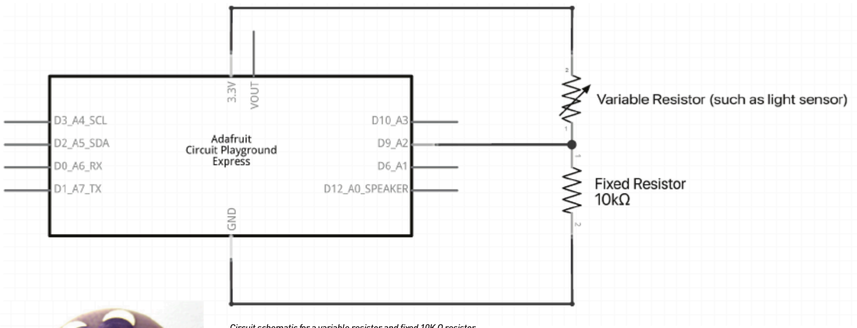
What kind of information does your sensor provide, and how? The sensor's output is what gets read by the microcontroller. You've learned so far that a varying voltage output can be read by the analog input pins on the Arduino. Later in this chapter, you will also learn how to read varying resistance and how a sensor transmits data via serial communication.

VOLTAGE-DIVIDER CIRCUIT

Some sensors are *variable resistors*, components that change resistance in response to a changing condition. Variable resistors you will encounter later in this chapter include flex sensors, stretch sensors, and light sensors.

The problem with trying to read a variable resistor with a microcontroller is that a microcontroller's analog input reads varying *voltage*, not varying *resistance*. Luckily, there is a simple circuit that enables a variable resistor to produce varying voltage: a *voltage-divider* circuit.

This circuit pairs a variable resistor connected from power to the analog input with a fixed resistor connected from ground to the same analog input.

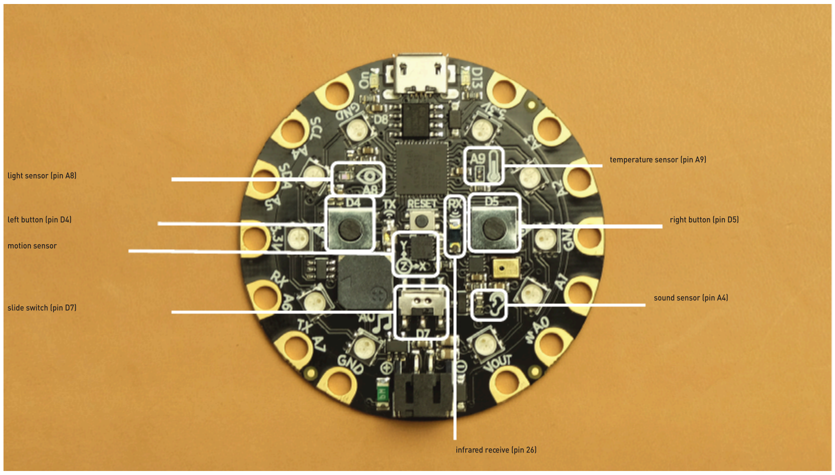


Circuit schematic for a variable resistor and fixed 10K Ω resistor

The fluctuating ratio of their resistances creates a varying voltage between them. For your purposes, the fixed resistor just needs to be in the same order of magnitude as the variable resistor. Many of the variable resistors you encounter in this chapter will work just fine with a fixed 10K Ω resistor.

Closeup of LilyPad Light Sensor; note the resistor on the right as well as how the circuit board traces create a voltage-divider circuit.

You've actually already worked with a variable resistor in chapter 6: the LilyPad Light Sensor. If you look closely at this component, you can see the surface-mount fixed resistor and the traces going to the pads.



CPX on-board sensors

ON-BOARD SENSORS

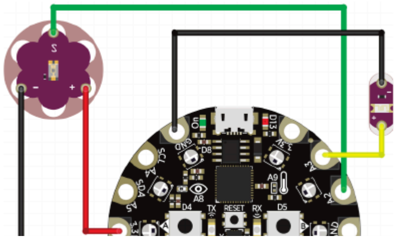
As the cost of components decreases, more microcontroller boards are being released with onboard sensors. This can be very useful for prototyping—you can immediately get started working with sensor data without having to do any wiring or circuit building.

The CPX is an example of a microcontroller board that features built-in sensors. The sensors include light, sound, temperature, and accelerometer. There are also two buttons and a slide switch, as well as a set of capacitive touch pins. To get started with these, check out Adafruit's CPX guide (learn.adafruit.com/adafruit-circuit-playground-express).

Keep in mind that when building wearable-electronics projects, you may not always want your microcontroller and your sensor to be located on the same part of the body. You can always start by prototyping with on-board sensors and later switch to stand-alone sensor boards that you can connect to the microcontroller board using wire, conductive thread, or one of the other approaches introduced in chapter 2.

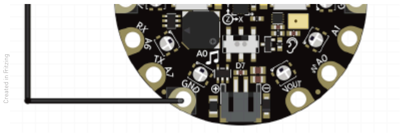
Working with Sensor Data

Once you're able to get access to the sensor data, you then need to figure out what to do with it. In this section, I explore some concepts for making sense of sensor data, including thresholds, mapping, calibration, constraining, and smoothing.

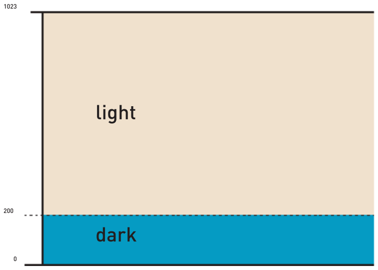


Let's use the light-sensor circuit from the "Sensitive System" experiment in the previous chapter for the following examples. Here are the parts you will need:

- [1] sewable microcontroller
- [1] LilyPad LED
- [1] LilyPad Light Sensor
- [1] USB mini-B cable
- alligator clip test leads



Circuit diagram of CPX with LilyPad Light Sensor as input and sewable LED as output



With a threshold set at 200, all values above are considered light and all below dark.

THRESHOLDS

A *threshold* can be used to set a boundary between one condition and another. Think of it as like a border between two countries or a fence between two yards: Setting the boundary makes it easier to distinguish one thing from the other.

When working with a range of sensor values, sometimes it's helpful to indicate what different ranges within those values mean. Thresholds are a good way to get started.

Say you're working with a LilyPad Light Sensor. It's great to have a bunch of numbers flying by on the Serial Monitor, but how can you use them to make something happen? For instance, if you want an LED to turn on when it's dark and off when it's light, you need to define what dark is.

By looking at the values in the Serial Monitor and exposing the sensor to varying conditions (in this instance, turning the lights on and off), you can select a *threshold* value. This is a number above which you would consider the condition to be light and below which it would be dark.

You can implement a threshold in your code through the use of an if statement:

```
  // if it is "dark"
  if(lightSensorValue<200){
    // Turn LED on
    digitalWrite(LEDpin, HIGH);
  }
  // if it is "light"
  else{
    // Turn LED off
    digitalWrite(LEDpin, LOW);
  }
```

This example will turn an LED connected to pin 10 (A3) on when it is dark and off when it is light. Here is the complete sketch:

```
/*
Make: Wearable Electronics
 Single Threshold example
 */
```

```
//initialize variables
int lightSensorValue = 0;
int lightSensorPin = A2;
int LEDpin = 10;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
void loop() {
  // read the light sensor pin and
  // store value in a variable:
  lightSensorValue = analogRead(lightSensorPin);

  // if it is "dark"
  if(lightSensorValue<200){
    // Turn LED on
```

```
    digitalWrite(LEDpin, HIGH);
  }
  // if it is "light"
  else{
    // Turn LED off
    digitalWrite(LEDpin, LOW);
  }

  // delay between readings:
  delay(100);
}
```

You can also have multiple thresholds and use the else if conditional statement.

The following sketch uses two thresholds and prints out a description of the light level as well as the raw sensor value in the Serial Monitor.

```
/*
Make: Wearable Electronics
 Multiple Threshold example
 */

// initialize variables
int lightSensorValue = 0;
int lightSensorPin = A2;
int LEDpin = 10;
int threshold1 = 500;
int threshold2 = 200;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
```

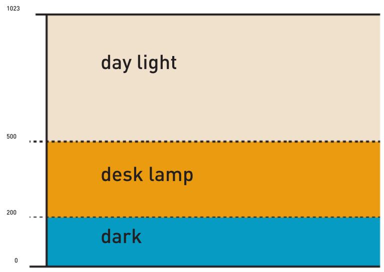
```
void loop() {
  // read the light sensor pin and
  // store value in a variable:
  lightSensorValue =
  analogRead(lightSensorPin);
```

```
  // print the light sensor value
  Serial.print("Light Sensor Value: ");
  Serial.print(lightSensorValue);

  // get ready to print light level
  Serial.print(", Light Level: ");

  // if the value is greater than
  // threshold #1
  if(lightSensorValue>threshold1){
    Serial.println("daylight");
  }
  // if the value is less or equal to
  // threshold #1 and greater than
  // threshold #2
  else if(lightSensorValue>threshold2){
    Serial.println("desklamp");
  }
  // if the value is equal to or less than
  // threshold #2
  else{
    Serial.println("dark");
  }

  // delay between readings:
  delay(100);
}
```

Give this code a try and modify the values so that they better match your environment.

MAPPING

Mapping is a way to translate a value from one range of numbers to another. It can be used to create a direct relationship between an input and an output. For instance, the value provide by a light sensor could control the *brightness* of an LED (as opposed to turning the LED on and off, as you did earlier).



Multiple thresholds set at 200 and 500

To accomplish this, there is a very useful Arduino function called **map()**. It looks like this:

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

In this function, **value** is the **value** you would like to map, **fromLow** and **fromHigh** are the low and high end of the original data set, and **toLow** and **toHigh** are the low and high values of the mapped data set. If you were to map the full range of analog input (0–1023) to the full range of analog output (0 to 255), you'd use the following line of code:

```
map(lightSensorValue, 0, 1023, 0, 255)
```

A complete sketch looks something like this:

```
/*
Make: Wearable Electronics
 Mapping example
 */

// initialize variables
int lightSensorValue = 0;
int lightSensorPin = A2;
int LEDpin = 10;
int mappedLightSensorValue = 0;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
void loop() {
  // read light sensor pin and
  // store value in a variable:
  lightSensorValue =
  analogRead(lightSensorPin);
  // map sensor value
  mappedLightSensorValue =
    map(lightSensorValue, 0, 1023, 0, 255);
  // set analog output accordingly
  analogWrite(LEDpin, mappedLightSensorValue);
```

```
// print the sensor and mapped sensor values:
Serial.print("Light Sensor Value: ");
Serial.print(lightSensorValue);
Serial.print(", Mapped Light Sensor Value: ");
Serial.println(mappedLightSensorValue);

// delay between readings:
delay(100);
```

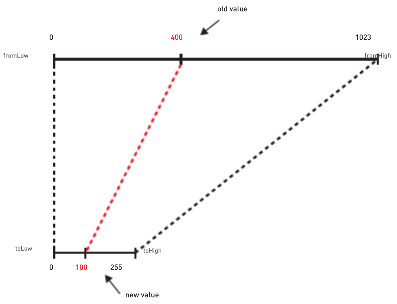
```
}
```

Many sensors don't have values that fully occupy the 0–1023 range. If you have a light sensor whose lowest value is 25 and highest is 940, you can change the *from* values accordingly:

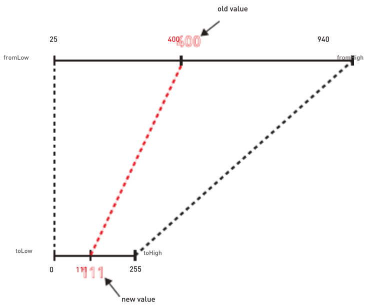
```
map(lightSensorValue, 25, 940, 0, 255)
```

If you are working with a LilyPad Light Sensor as your input and using the mapped value to control an LED on the analog output pin, you would have an LED that brightens and dims in a way that mimics the conditions of the room. If you wanted to inverse the relationship so the LED gets brighter as the room gets darker, you can simply flip the *to* values:

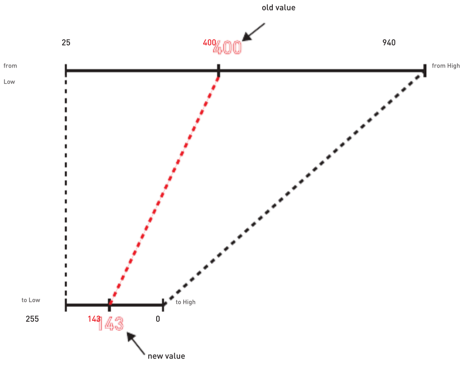
```
map(lightSensorValue, 25, 940, 255, 0)
```



Mapping a value from 0–1023 to 0–255



Mapping a value from 25–940 to 0–255



Mapping a value from 0–1023 to 0–255

See also:

- Arduino **`map()`** reference page (docs.arduino.cc/language-reference/en/functions/math/map)

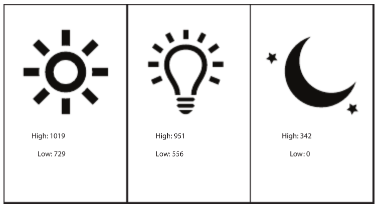
- Arduino In, Out Serial example (docs.arduino.cc/built-in-examples/analog/AnalogInOutSerial)

CALIBRATION

Calibration is a way to fine-tune your code so it is responsive to a specific set of conditions. The range of what a sensor senses will differ based on its environment and context. The amount of light available in your bedroom or studio will differ greatly from that on the street or in a park. A force-sensing resistor will read different values when stepped on by a five-year-old compared to a fifty-year-old. If you know you'll be using your project in varying contexts, it's worth including a calibration routine in your code. You can determine the highest and lowest possible values and configure the rest of your program accordingly.

This is great to include in your setup, but should you need to recalibrate without restarting the Arduino entirely, you can also create a calibration routine triggered by a button.

To see an example of this, in Arduino, go to File → Examples → 03.Analog → Calibration. In the code, change the analog input pin to A2 and the digital output pin to pin 11. Then upload the code to your Arduino board.



The highs and lows of a light-sensor value can differ according to the current conditions; calibration can help with this.

This example looks for the highest and lowest values that occur during the first five seconds the program is running. Once the Arduino is programmed, in order to recalibrate the sensor values, press the Reset button and expose the sensor to the highest and lowest light conditions during the following five seconds.

See also:

- Arduino calibration example (docs.arduino.cc/built-in-examples/analog/Calibration)

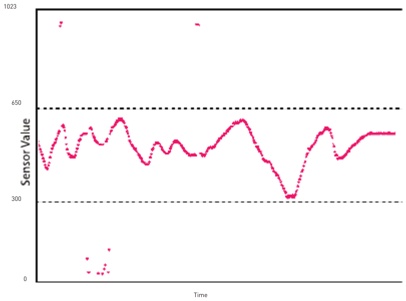
CONSTRAINING

Sometimes your sensor will provide readings that fall outside of your desired range. For these cases, Arduino provides a function called **`constrain()`**. The three parameters needed are the data that is being constrained, the lowest value you would like to keep, and the highest value you would like to keep. If there are any values that are below or above the specified range, the **`constrain()`** function will convert them to the lowest or highest desired values, respectively.

In practice, it might look something like this:

```
/*
Make: Wearable Electronics
 Constrain example
 */
// initialize variables

int lightSensorPin = A2;

int lightSensorValue = 0;

int constrainedLightSensorValue = 0;
```

```
void setup() {

  // initialize serial communication:

  Serial.begin(9600);

}


void loop() {

  // read light sensor pin and store

  // value in a variable:

  lightSensorValue = analogRead(lightSensorPin);

  // constrain the light sensor values from 300 to 650

  constrainedLightSensorValue =

    constrain(lightSensorValue, 300, 650);


  // print the results:

  Serial.print("Light Sensor Value ");

  Serial.print(lightSensorValue);

  Serial.print(", Constrained Light Sensor Value: ");

  Serial.println(constrainedLightSensorValue);


  // delay between readings:

  delay(100);

}
```



*The **`constrain()`** function allows sensor readings to be constrained within a set range.*

See also:

- Arduino **`constrain()`** reference page (docs.arduino.cc/language-reference/en/functions/math/constrain)
- Arduino calibration example (docs.arduino.cc/built-in-examples/analog/Calibration)

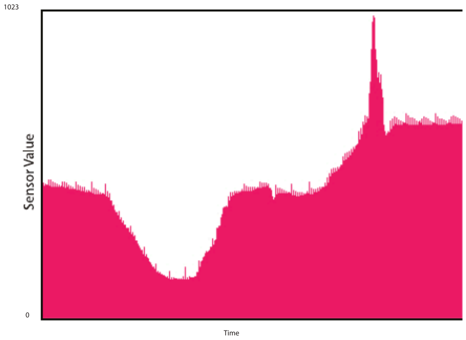
SMOOTHING

While some sensors produce smooth and predictable data, others offer a dataset that's rougher around the edges. Smoothing can help turn an erratic datastream into something cleaner and easier to work with.

To give this a try in Arduino, go to File → Examples → 03.Analog → Smoothing. Once the sketch is open, change the analog input pin to A2 and upload the sketch to your Arduino. Open the Serial Plotter and see what the new, smoothed sensor data looks like!

- Arduino smoothing example (docs.arduino.cc/built-in-examples/analog/Smoothing)

NOTE: Looking at data in the Serial Monitor can be a good place to start, but sometimes it's helpful to see a visual representation of how the data changes over time. Check out the Arduino Graph example (docs.arduino.cc/built-in-examples/communication/Graph) for more information.



Some data requires smoothing.

EXPERIMENT: Wooo! Shirt

Using the light-sensor circuit you've been working with in this section, incorporate the circuit into a shirt with the light sensor positioned in the armpit of a shirt.

Here's some starter code for you to modify. Upload it to your Arduino, and then put the shirt on:

```
/*
Make: Wearable Electronics
 Wooo! Shirt Experiment
 */

// initialize variables
int lightSensorValue = 0;
int lightSensorPin = A2;
int LEDpin = 10;
int wooThreshold = 120;

void setup(){
  // initialize serial communication:
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
```

```
void loop(){
  // read the value from the sensor
  lightSensorValue = analogRead(lightSensorPin);

  // if the arm is up
  if(lightSensorValue>wooThreshold){
    // print Wooo!
    Serial.print("Wooo!");
```

```
      // Turn LED on
      digitalWrite(LEDpin, HIGH);
    }
    // if the arm is down
    else{
      // print boo
      Serial.println("boo ");
      // Turn LED off
      digitalWrite(LEDpin, LOW);
    }

    Serial.print(" Sensor Value: ");
    Serial.println(lightSensorValue);

    delay(100);    // delay for 1/10 of a second
}
```

Illustration by Jen Liu

This code prints a "Wooo!" when it detects light and a "Boo" when it does not. Based on what you've learned about the concepts of thresholds, mapping, calibration, smoothing, and constraining, create a program that reliably prints "Wooo!" when your arm is raised.

Keep in mind that using the Serial Monitor as a feedback mechanism is great for prototyping, but it will keep you tethered to the computer. Later on, you can build a more creative response into your design using LEDs or other possible outputs you'll learn about later in chapter 8 so you can "Wooo!" more effectively in the wild.

What to Sense

It's easy to hear about a cool sensor and decide to do a project with it.

"Oh, there's a really neat X sensor that just came out. I should obviously do an X project!"

But this leads to an interaction that's designed around the technology rather than *technology that's designed around a particular interaction*.

When working with sensors, a good place to start is to think about *what you're trying to sense*. What is the motion, action, or condition? What is the context and environment? What are the important aspects to consider? Then you can ask questions like these:

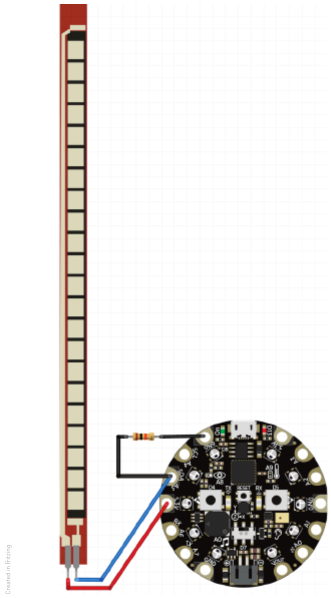
- *Which different sensor (or sensors) could I use?*
- *What do I want to measure (sound, light, pressure, presence etc.)?*
- *Where should the sensors live?*
- *What should I be looking for in the data I am gathering from them?*

In the following section, you'll look at some possibilities for what to sense and a selection of sensors that will fit the bill. The examples that follow focus on sensors *not* included on the microcontroller so you have the flexibility to place the sensor wherever works best for your project. Keep in mind that this selection of sensors is just the tip of the iceberg. Once you have a project idea in mind, go out and research what's available to best help your idea come to life.

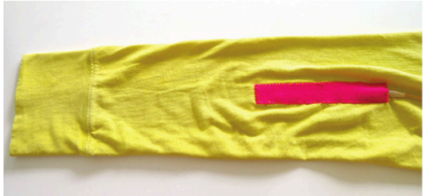
FLEX

Bodies are bendy, and it just so happens that *flex sensors* sense a flex or a bend. They're very good for areas of the body that bend in a broad, round arc. They work well on elbows, knees, fingers, and wrists. They are variable resistors and need to be used in combination with a voltage-divider circuit in order to be read by a microcontroller.

Flex sensor



Circuit diagram of CPX with flex sensor



A fabric pocket is added to the sleeve to hold flex sensor in place.

Here are some factors to consider when choosing a flex sensor:

Length

Flex sensors come in different lengths—usually 2.2 inches or 4.5 inches. Use whichever best fits your application. For instance, the longer ones work well with fingers, but the shorter ones might be more appropriate for toes.

Directionality

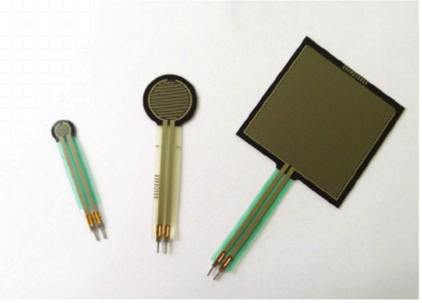
Flex sensors can be single or bidirectional. Bidirectional flex sensors sense flex in both directions, whereas the single-direction ones can sense flex in only one direction. Single direction is fine for many human joints like fingers, elbows, and knees. But bidirectional is useful for joints like the wrist, where the bend can take place in both directions.

Resistance range

Some flex sensors are also available in different resistance ranges. For your purposes, there are no real advantages or disadvantages that come with the options in this category. Just be sure that you're using a resistor of the appropriate size in your voltage-divider circuit.

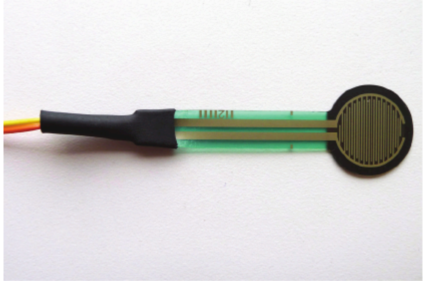
The biggest challenges in working with flex sensors are positioning and protection. In order to get an accurate reading of the flex of your elbow, the sensor needs to be positioned in the same place on your elbow every time. Creating a secure pocket for the sensor can help with this.

Flex sensor on a bent elbow



Force-sensing resistors

A pocket sewn onto the sock helps keep the FSR securely in place.



Heat-shrink tubing is used to protect the delicate solder connections between the sensor and wires.

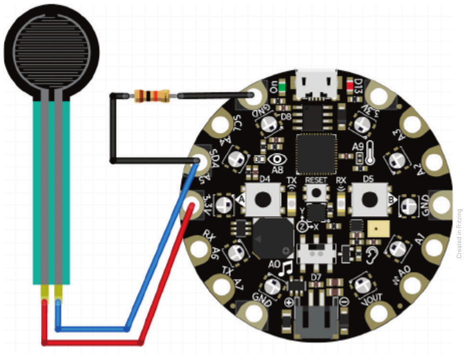
When pressure is applied to the ball of the foot, the microcontroller can read the change in sensor data.

The other thing to consider is that while flexing is a pretty rigorous and strenuous activity, many flex systems are fairly delicate, particularly at their connection terminals. Be sure to protect your connections. Reinforce them with heat-shrink, and protect them with some sort of material.

FORCE

Bodies often touch and get touched. One way to sense touch is through the use of FSRs, mentioned above, which have a makeup that's similar to flex sensors but are configured to be sensitive to pressure rather than bending. They are also variable resistors and have delicate connections similar to those of flex sensors.

They come in various shapes and sizes, and the assorted types are suited for different applications.



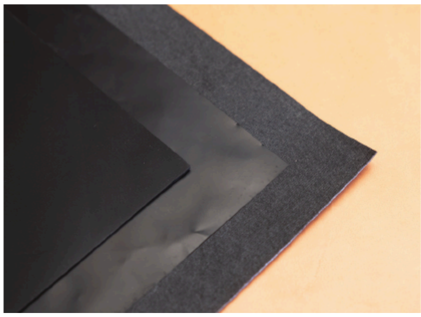
Circuit diagram of CPX with FSR

FSR comparison

TYPE	SENSING AREA	NOTES
Small (round)	0.16" diameter	Very versatile; best for sensing touch at highly specific locations
Medium (round)	0.5" diameter	Excellent for sensing the pressure of a fingertip
Large (square)	1.75" x 1.5"	Sits well on the top of a hand, on a shoulder, on ball of a foot
Long	0.25" x 24"	Great for sensing pressure along the length of an arm or leg

HOW TO: DIY Pressure Sensor

In addition to purchasing manufactured sensors, you can create your own. As you saw earlier, a variable resistor is simply something that changes resistance in response to a changing condition. Think about this from a material perspective, and you can end up with some pretty interesting results.

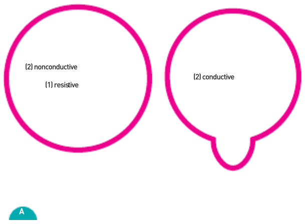


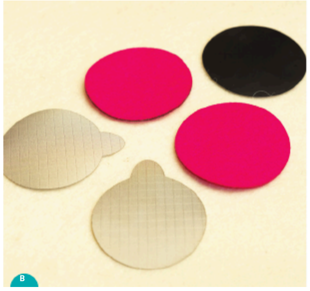
Maxwell Lander

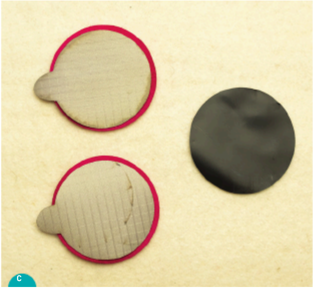
Resistive materials, from left to right: conductive rubber, Velostat/Linqstat, EeonTex fabric

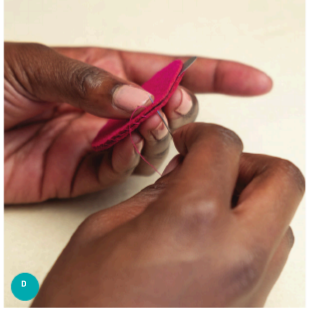
There are many DIY techniques for sandwiching a resistive material between two pieces of conductive material. The resistive material can be a plastic (like Velostat) or a fabric (like some made by Eeonyx), and the conductive material can be conductive fabric, thread, yarn, wire, mesh, or anything else you can dream up. Use a connection to each conductor as the two sides of a variable resistor, and you can monitor the change in values as you apply pressure to—or flex—the sensing sandwich that you've created.

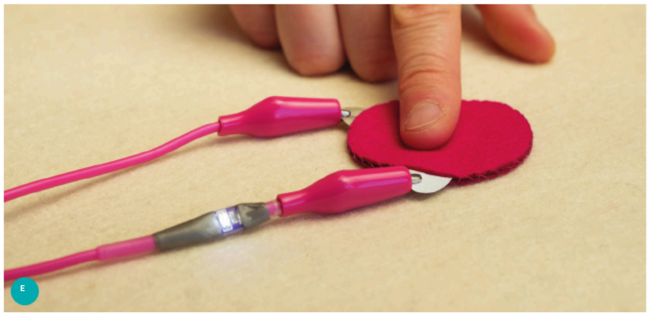
Hannah Perner-Wilson and Mika Satomi maintain a website called How to Get What You Want (kobakant.at/DIY), which is home to a vast repository of DIY electronic-textile and wearable-technology documentation. Check out their sensor section for a helpful collection of tutorials on how to make your own sensors.







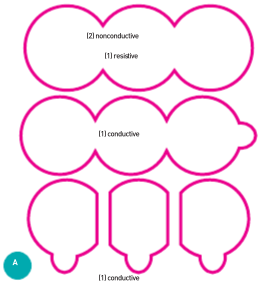


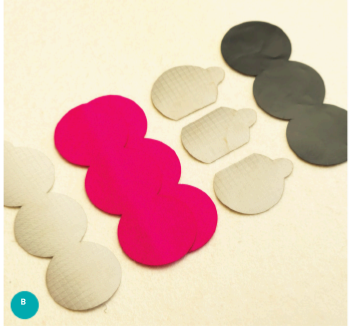


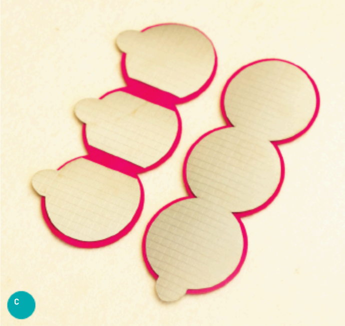
Maxwell Lander

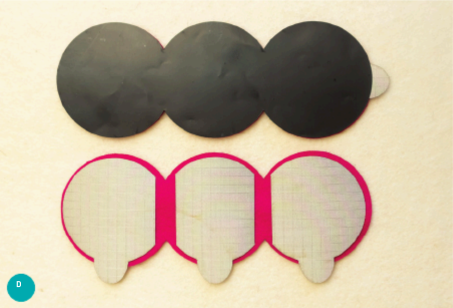
Steps:

1. Trace these patterns or create your own (Figure A).
2. Cut the materials. You can reuse the pattern from the DIY soft button from chapter 4—just replace the third piece of nonconductive material with a hole with one piece of resistive material in the same shape as the nonconductive material (Figure B).
3. Position the conductive fabric on the nonconductive fabric and iron in place (Figure C).
4. Make a sandwich with the conductive fabric facing inward and the resistive material in the middle. Sew or glue the perimeter (Figure D).
5. Test the DIY pressure sensor with a conductivity tester or a multimeter. Look for the brightening and dimming of the LED or the changing resistor values when the sensor is pressed (Figure E).









Maxwell Lander

HOW TO: Multiple DIY Pressure Sensors

This technique can also be used to create multiple pressure sensors.

Steps:

1. Create patterns (Figure A).
2. Cut the materials (Figure B).
3. Arrange the conductive pieces on the nonconductive pieces and iron them down (Figure C).
4. Add the Velostat to the middle and sew or glue the perimeter (Figure D).
5. Test the completed sensors. Each circuit should act as an independent pressure sensor (Figure E).

Try experimenting with different materials. Conductive fabric can be replaced by conductive thread or other conductive materials. You can also try a different resistive material or try to develop your own. Many artists and makers have experimented with felting together

This technique can also be used to create DIY flex sensors.

sheep's wool and conductive fibers (such as steel or copper wool). The addition of the nonconductive sheep's wool to the mix creates electrical resistance. The more you compress the mixed wool, the closer the conductive fibers become to each other—thus lowering the resistance and increasing conductivity.

STRETCH

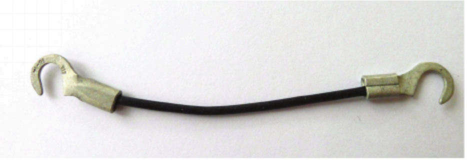
From the bend of a knee to the expansion and contraction of a rib cage with each breath, properly positioned stretch sensors can capture the fluctuating nuances and curves of the human form. A stretch sensor is simply a conductive rubber cord whose resistance decreases the more it gets stretched. This is yet another example of a variable resistor.

Stretch sensors come precut at different lengths with hooks crimped to either end for easy connection, or you can buy one by the meter and cut it to whatever length you need.

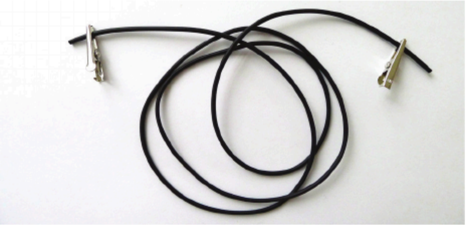
Stretch sensors are a fun material to work with. They can also be elegantly incorporated into textiles through knitting or weaving.

Conductive rubber also comes in sheets, which can be cut into strips. It functions similarly to conductive rubber cord. When connecting to it, the trick is to clamp the material rather than pierce it.

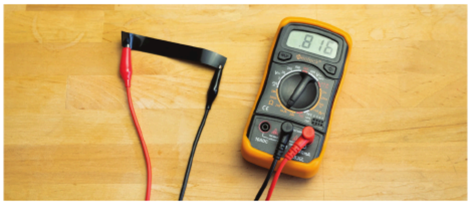
Similar to the pressure-sensor example, DIY



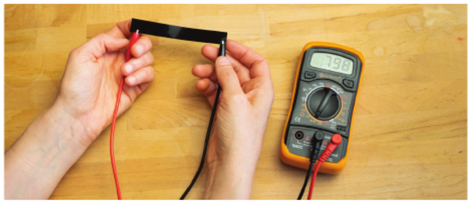
Stretch sensors with hooks attached



Stretch sensor with hardware for customization



A strip of conductive rubber at rest has a resistance of 81.6 Ω.

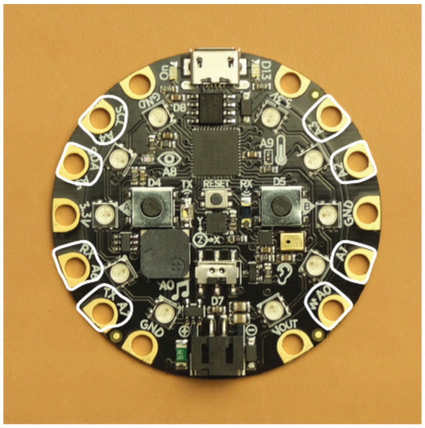


When picked up, it measures at 76.8 Ω.

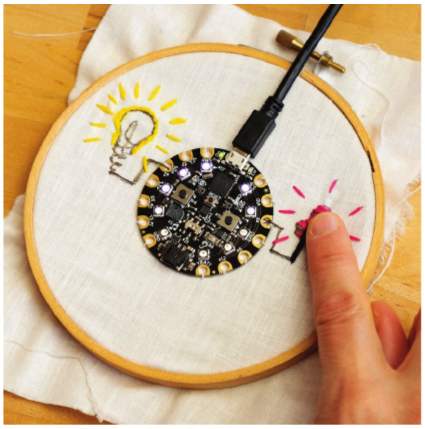
stretch sensors can be created by knitting or crocheting resistive yarns. The more the knit is stretched or pressed, the more highly conductive it becomes. Check out the Gallery at the end of this chapter for this technique in use in project examples.

Maxwell Lander

When it is stretched, the resistance drops to 15.6 Ω.


Maxwell Lander

Capacitive touch pins on CPX



Activating a capacitive touch pin on CPX with embroidered conductive thread (from Textile Game Controllers project by Social Body Lab)

CAPACITIVE TOUCH

Capacitive touch sensors can turn any piece of conductive material into a touch-sensitive interface. This means you can use soft conductive materials such as fabrics and threads to create touch-responsive sensor pads.

Capacitive touch sensors are available as stand-alone breakout boards. These boards can support one or more "keys" or touch pads that can behave as either momentary or toggle switches. Some offer on-board LEDs for a quick indication a sensor is activated. Capacitive sensing capabilities are also integrated into some microcontroller boards such as the CPX.

The tricky bit is that capacitive sensors are very sensitive. The wire or thread that connects the conductive material to the microcontroller or breakout board pin becomes touch-sensitive as well (unless properly shielded). Always be sure to press the Reset button on the board after adding conductive material to a capacitive sensor to make sure it is properly calibrated.

MOVEMENT AND ORIENTATION

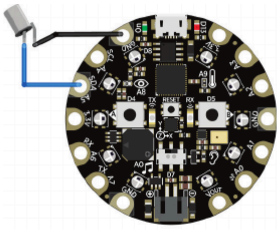
People are active and mobile creatures. They reach for things they want, turn toward loud noises, and crouch down to coax the cat from under the bed. When creating wearables that react to events such as these, it is helpful to be able to sense movement.

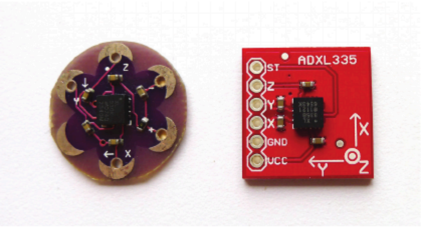
A cheap and easy way to sense movement is through the use of tilt switches.

But there are also far more sophisticated sensors you can use. *Accelerometers* measure acceleration or changes in speed of movement. They can also provide a good measurement of tilt due to the changing relationship to gravity.

Accelerometers have a set number of axes—directions in which they can measure. Most are *three-axis accelerometers*, meaning they can measure acceleration on the *x*, *y*, and *z* planes.

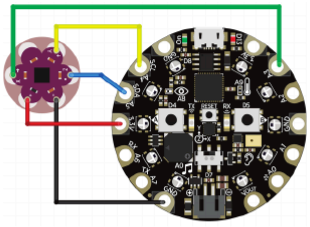
The ADXL335 triple-axis accelerometer is available in a variety of form factors—both on a standard breakout board and on a LilyPad board. These boards contain the same chip but are intended for different uses (conductive-thread circuit versus breadboard circuit). This is an analog accelerometer, meaning it outputs varying voltage for each axis reading. These three outputs can be connected to three analog inputs on the Arduino.

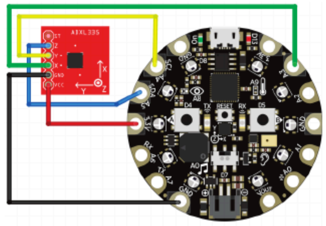

Circuit diagram of CPX with a basic tilt switch connected to a digital input pin



Maxwell Lander

LilyPad Accelerometer and ADXL 335 Accelerometer

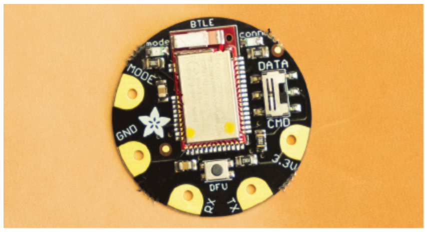

Circuit diagram of CPX with LilyPad Accelerometer



Created in Fritzing

Circuit diagram of CPX with ADXL335 Accelerometer

When working with analog accelerometers, in order for you to get actual acceleration readings, you need to interpret the raw data from the analog input. If need be, see the sensor datasheet for further information on how to do this. If you're just working with relative tilt, observing the changes in sensor data in the Serial Monitor is often good enough to get you started.

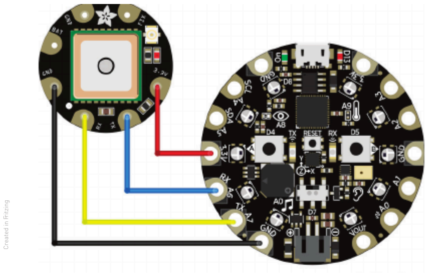


Maxwell Lander

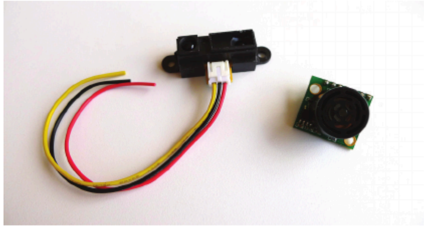
Flora GPS unit

LOCATION

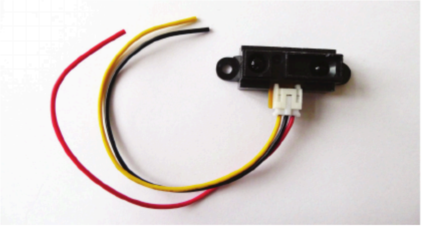
If tilt, motion, and orientation aren't enough, and you want your wearable to know where you are



Circuit diagram of CPX with the Flora GPS module



Proximity sensors



Infrared proximity sensor

on the planet, GPS is the way to go. Just like your car, bike, or phone, your jacket or disco pants can have GPS too. There are a number of Arduino-compatible GPS units available, but the Flora GPS is a compact and sewable option.

PROXIMITY

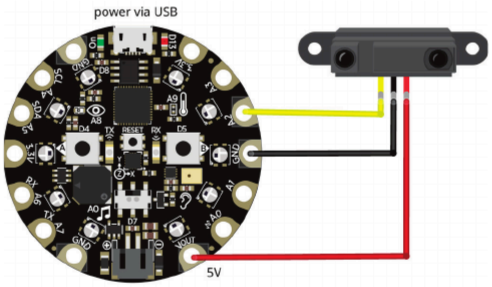
Sometimes you will want to know how close or far away something is from the body. Proximity sensors are useful for detecting nearby objects, walls, or even other people. When selecting a proximity sensor, it is worth considering what your desired range or detecting distance is, as well as what sort of beam width you need to monitor.

There are two types of proximity sensors that are fairly easy to get up and running: infrared and ultrasonic.

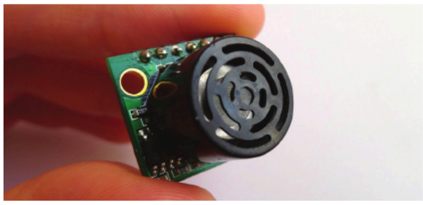
Infrared, or IR, sensors use light to measure proximity. The sensor sends out a beam of infrared light (invisible to the human eye) that bounces off the object in front of it and is read by the sensor. IR sensors are the less expensive option for proximity sensors but are more easily tricked by heat and sunlight. They tend to have shorter, more focused distance ranges, such as 3–30 cm, 10–80 cm, or 20–150 cm.

These sensors can be directly connected to the analog input on any Arduino board.

Ultrasonic sensors work similarly, except that instead of sending out light, they send out ultrasonic sound (which humans can't hear). The sound bounces off whatever is proximate and returns to the sensor. The proximity is determined by the length of time it takes for the sound to return. MaxBotix manufacturers offer a sophisticated line of ultrasonic sensors that meet a range of needs, from the most basic to highly sensitive and rugged. They come in a variety of beam widths, have long sensing ranges (0–150 inches on their most basic model), and even are available with waterproof outdoor housings. Ultrasonic sensors are more expensive and bulkier than IR sensors, but they are more precise and harder to trick.



Circuit diagram of CPX with IR sensor

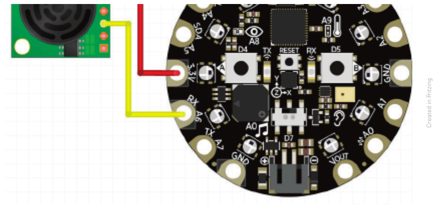


Ultrasonic proximity sensor

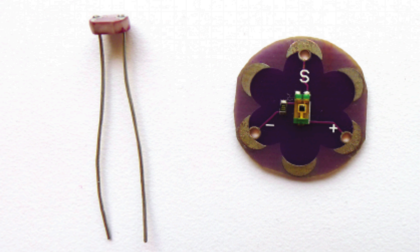
LIGHT

Remember your old friend the light sensor? You used a LilyPad Light Sensor in your first analog input example, but light sensors come in many other forms.

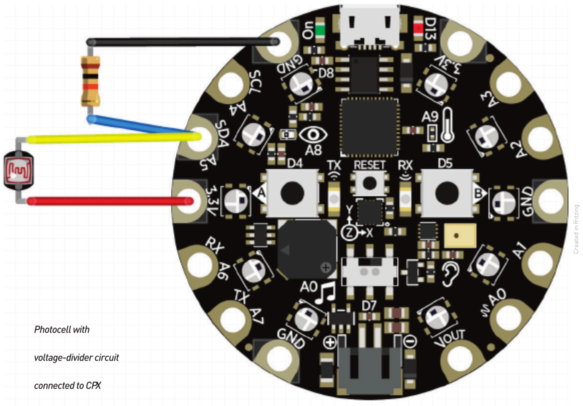
The most basic type of light sensor is the photocell. Its resistance varies based on the level of light it senses. Some have resistance that increases as the light level increases, but others have the reverse relationship. This can be quickly determined by viewing the sensor values in the Serial Monitor.



Circuit diagram of CPX with ultrasonic sensor



Photocell and LilyPad Light Sensor



Photocell with voltage-divider circuit connected to CPX

The photocell is a great sensor to work with because it is small, easy to manipulate, and very inexpensive. It can be used to sense ambient light levels, but it can also be used for less intuitive purposes like determining whether a jacket is open or closed or whether the heel of a shoe is on the ground or in the air.

SOUND

Sounds can provide significant clues about what is going on around you. By detecting sound level, you can create wearables that are more sensitive to their environment, like a scarf that purrs when it is whispered to or a collar that pops up in response to loud noises.

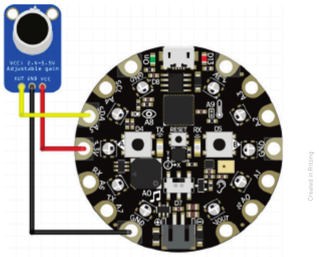
A simple microphone can act as a great sensor for audio-reactive projects. For getting started with reading an audio signal in Arduino, a small *electret microphone* will do the trick. These little guys can't be plugged directly into an analog input—their fluctuating signal is measured in microvolts, which is far too subtle for the ears of your microcontroller. But they are available on breakout boards that feature an amplifier chip and other components that allow them to be directly connected to an Arduino. The Adafruit model features a *trimpot* (a knob that can be adjusted with a screwdriver) on the back that allows you to make adjustments to the gain on the fly.

The microphone signal can be read by an analog input pin, but you need to do further calculations to determine the amplitude (volume) the mic is detecting. Here's some code to get you started:

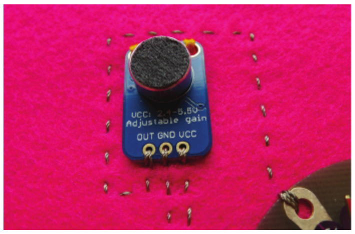
```
/*
Make: Wearable Electronics
 Mic Example
 Based on "Example Sound Level Sketch for the
 Adafruit Microphone Amplifier"
 http://bit.ly/1qlN7hk
 */

int micPin = A5;

// Sample window width in mS (50 mS = 20Hz)
int sampleWindow = 50;

void setup(){
  Serial.begin(9600);
}

void loop() {
    // Start of sample window
  unsigned long startMillis = millis();
  int amplitude;
  int micReading;
  int maxReading = 0;
  int minReading = 1024;

  // collect mic readings and find the
  // max and min
  while (millis() - startMillis < sampleWindow){
    micReading = analogRead(micPin);

    if (micReading > maxReading){
      maxReading = micReading;
      //save the maximum reading
    }
```

MAX4466 electret-microphone amplifier with adjustable gain (front)

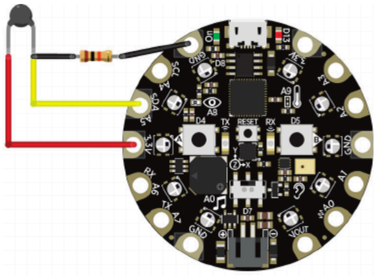
MAX4466 electret microphone amplifier with adjustable gain (back)



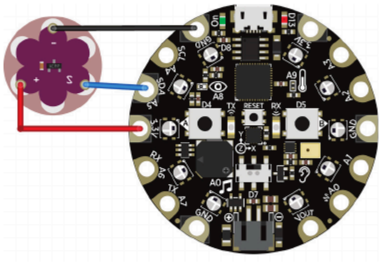
Circuit diagram of CPX with electret-microphone amplifier



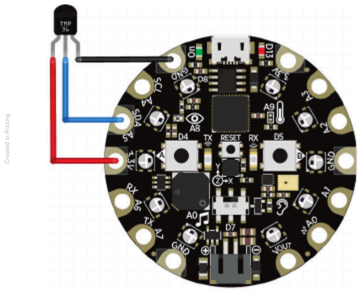
Holes meant for headers can also be used for conductive-thread connections.



Circuit diagram of CPX with thermistor



Circuit diagram of CPX with LilyPad Temperature Sensor



Circuit diagram of CPX with TMP36

```
    else if (micReading < minReading){
      minReading = micReading;
      // save the minimum reading
    }
  }

  // find the amplitude
  amplitude = (maxReading - minReading);
  Serial.println(amplitude);
}
```

TEMPERATURE

Clothing provides warmth and protection. It makes sense that responsive clothing might want to react to temperature. Temperature sensors can be used to sense both environmental conditions and body warmth. Many temperature sensors are available, from analog to digital to high temperature to waterproof to noncontact. They are also sometimes combined with sensors for barometric pressure, humidity, and altitude.

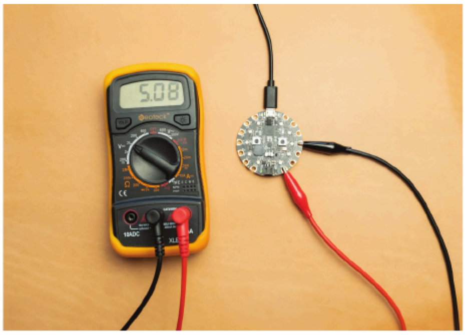
Each of these units works in its own way and requires a bit of research and testing. An easy place to start is with a thermistor (a variable resistor), which can be connected to the Arduino with a simple voltage-divider circuit.

Thermistors are not the most precise temperature sensors, but they are excellent for rough temperature comparison. For example, you can easily use a threshold to create a distinction between what is considered hot and what is considered cold.

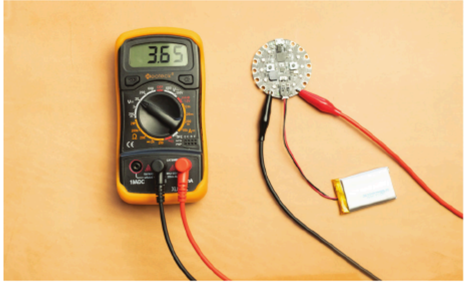
For more precise reading, try working with a LilyPad Temperature Sensor, or a TMP36, a simple analog temperature sensor.

The following code will work with either the LilyPad Temperature Sensor or the TMP 36. Be sure to change the **`supplyVoltage`** variable to whatever voltage you are working with in your circuit.

```
/*
Make: Wearable Electronics
 Temperature Sensor example
 */

// This is a reference voltage for your power
// supply. Measure it with a multimeter when
// running and change to the correct voltage.
float supplyVoltage = 3.3;


int tempSensorPin = A5;

int tempSensorValue;

float tempSensorVoltage;



// the setup routine runs once when you press reset:

void setup() {

  // initialize serial communication at 9600 bits

  per second:

  Serial.begin(9600);

}



void loop() {

  // read the temperature sensor value

  tempSensorValue = analogRead(tempSensorPin);


  // convert the reading to voltage based off

  the reference voltage

  float tempSensorVoltage =

  (tempSensorValue * supplyVoltage)/1024.0;


  // convert the reading to Celsius

  // converting from 10 mv per degree with 500 mV offset

  float temperatureC = (tempSensorVoltage - 0.5) * 100;
```



Measuring voltage from VOut pin as 5.08 V when CPX is powered via USB



Measuring voltage from VOut pin as 3.65 V

```
  // to degrees ((tempSensorVoltage - 500mV)

  times 100)


  // print in Celsius

  Serial.print("Degrees C: ");

  Serial.print(temperatureC);


  // convert to Fahrenheit

  float temperatureF =

  (temperatureC * 9.0 / 5.0) + 32.0;


  // print in Fahrenheit

  Serial.print(", Degrees F: ");

  Serial.println(temperatureF);


  delay(100);

}
```

BIOMETRIC DATA

Your heart beats faster when you're excited, and your skin gets clammy when you're nervous. Besides sensing your environment and your movements, you can also use

when CPX is powered via 3.7 V lithium polymer battery

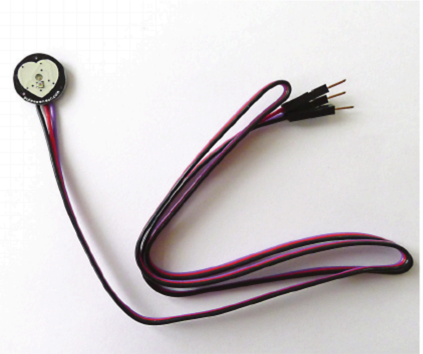
Maxwell Lander

Measuring voltage from 3.3 V pin as 3.30 V when CPX is powered via 3.7 V lithium polymer (LiPo) battery. Note that this voltage passes through a 3.3 V regulator rather than being the raw voltage from the power source.

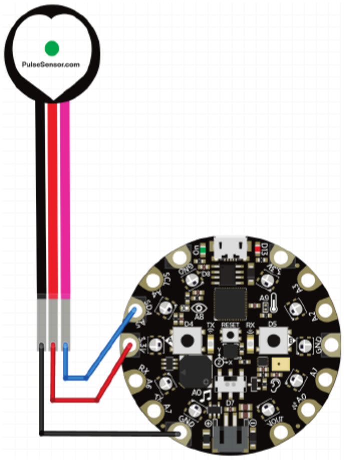
sensors to learn more about what is happening within someone's body. A great place to start sensing these biometrics is pulse, or heart rate.

Optical heart rate sensors, such as the Pulse Sensor Amped, are a small, lower-cost solution for measuring pulse. This type of sensor measures the mechanical flow of blood, usually in a finger or earlobe. It contains an LED that shines light into the capillary tissue and a light sensor that reads what is reflected back. It produces varying analog voltage that can be ready by the analog input on any Arduino.

Sensors that measure the electrical activity of the heart tend to be more expensive but are a more accurate solution for measuring heart rate. They measure the actual electrical frequency of the heart through two conductive electrodes (usually made with conductive fabric or conductive adhesive) that must be pressed firmly against the skin. This single-lead heart rate monitor can either be read as a simple analog output or can be charted as an electrocardiogram (ECG) in a program like Processing.



Pulse sensor



Created in Fritzing

Circuit diagram of CPX with Pulse-sensor

Beyond heart rate, you can measure many other biological signals. A few are listed below:

Electromyography (EMG)

A method of measuring of muscle activity by detecting its electrical potential. Myoware is a company that has released several versions of an accessible EMG sensor that provides varying analog voltage so you can easily read muscle activity with an Arduino analog input pin.

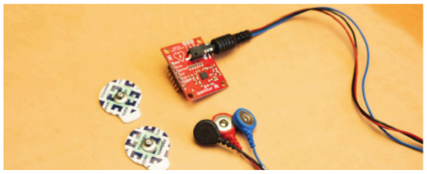
Galvanic skin response (GSR)

A method of measuring the conductivity of the skin. Changes in this conductivity can indicate a response to physical or psychological stimulus. (GSR sensors are used in classic lie detectors.) A GSR sensor can be built with some basic inexpensive electronic components.

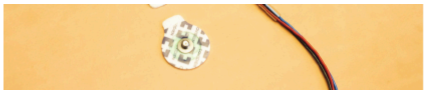
Electroencephalography (EEG)

A method of measuring electrical activity in the scalp. EEG headsets are often used in thought-controlled computing applications.

To learn more about how to work with these



types of sensors, check out the article "Biosensing: Track Your Body's Signals and Brain Waves and Use Them to Control Things," by Sean M. Montgomery and Ira M. Laefsky, in volume 26 of *Make:* magazine.



The single-lead heart rate monitor comes with sticker with pulse sensor

Maxwell Lander

EXPERIMENT: Body Listening

The interfaces you use tend to target specific areas of the body, such as hands, fingers, and feet. But what are other parts or areas of the body that aren't properly considered? For this experiment, create an interface for a part of the body you think is not listened to enough.

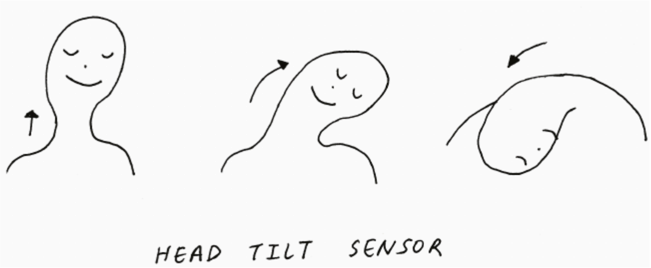


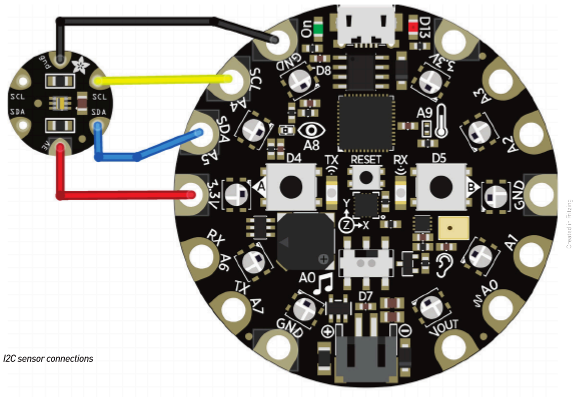
Illustration by Jen Liu

Here's a process to follow:

1. Decide on a body part or area of focus.
2. Make a list of five ways you can sense or listen to that area.
3. Pick one approach you can easily prototype.
4. Prototype it.
5. Try out your invention.
6. Make adjustments to code and hardware as needed.
7. Repeat until you think it listens well.

Working with I2C

Analog sensors provide a varying voltage that can be read by an analog input pin on an Arduino. This is the method used to read the sensors we've looked at so far in this chapter. But many useful sensors communicate sensor values to the microcontroller through different means: I2C (pronounced "I-squared-C" or "I-two-C"), the Inter-Integrated Circuit, a two-wire serial communication protocol. While this book won't provide a comprehensive discussion of I2C, it's still helpful to know the basic details.



I2C sensor connections

The following connections are made between any I2C sensor and the microcontroller:

SCL Serial clock pin—pulses on this pin provide the timing for the communication

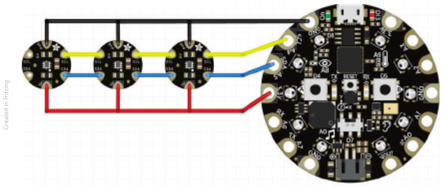
SDA Serial data pin—the wire on which the actual data is sent and received

gnd Share a common ground with the microcontroller

3 V Supply with 3.3 V pin from the microcontroller

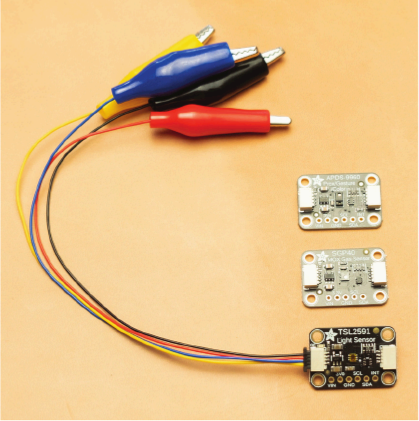
On some boards, SCL and SDA are clearly marked. (For instance, on the CPX, SDA is A4 and SCL is A5.) But if not, be sure to check board's pinout diagram.

A neat aspect of working with I2C devices is that they can be connected together in a chain. This means they don't take up lots of pins on the microcontroller. This also greatly reduces the amount of wiring. I2C sensors (and other devices) usually have a predetermined address, which is used by the microcontroller to speak to a particular device in the chain of I2C devices. This information can be found in the device's datasheet.



Circuit diagram of CPX with a chain of I2C sensors

On the software side, there is an Arduino library called Wire that enables Arduinos to communicate with I2C devices and handles the nitty-gritty details of this protocol so you don't have to. *Libraries* are additional packages of code that can be added in to support different functionalities or tasks. Some (standard libraries) are included with your Arduino download, and others (contributed libraries), you need to



Several sensors that use I2C

Maxwell Lander

download and install yourself. (Wire is a standard library.) In addition, a variety of sensor-specific contributed libraries are available.

While I2C is a more advanced approach to working with sensors and actuators, many sensors have excellent documentation available, so it is fairly straightforward to get up and running quickly. These sensors also often provide more sophisticated functionality, higher accuracy, and more legible data outputs. Examples include sensors that detect light of different wavelengths, color, humidity, air quality, and magnetic north, as well as more advanced options for the categories of sensing that have already been covered in this section. Be sure to check out the latest options from your favorite trusted electronics supplier!

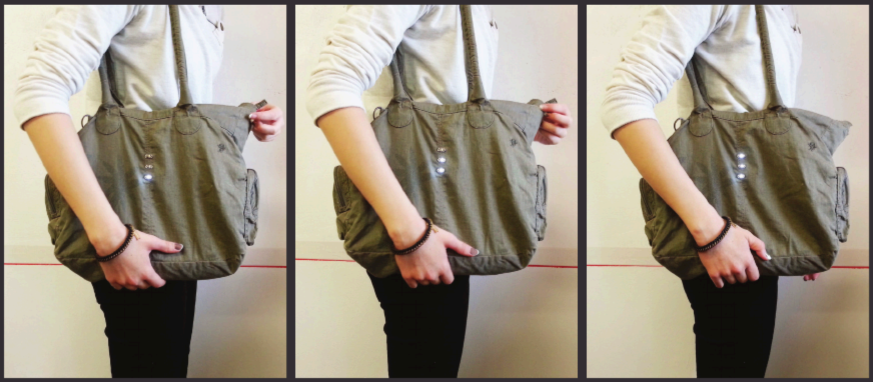
Looking Ahead

This introduction to sensors is really meant as a springboard to launch you into the deep and beautiful pool of sensor possibilities. Remember to start from your concept and work out from there. "Is there a sensor that senses X?" is a great question to bring to a search engine, an online forum, or your neighborhood nerd friend. From there, let the datasheet be your guide, and you'll be on your way to producing smartly sensitive wearable systems.

GALLERY 7:

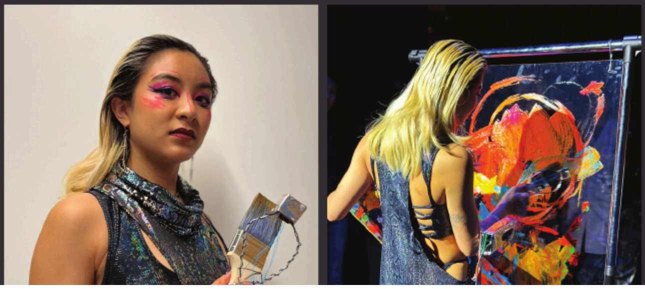
SENSORS IN WEARABLES

Sensors generate data that describes how we touch, feel, and move and the conditions surrounding us. These projects showcase the variety of sensors that artists, designers, and researchers integrate into their wearable creations.



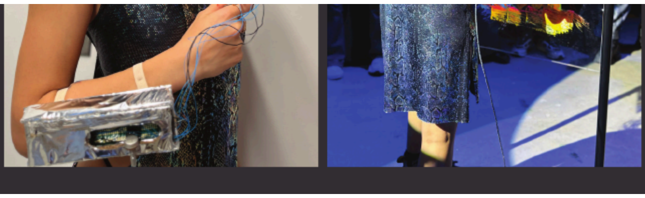
Capacitor Indicator Bag

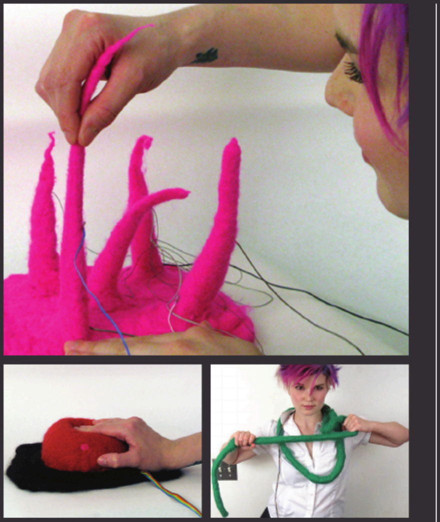
The Capacity Indicator Bag, by Sally Chan, uses a force-sensing resistor and a three-LED display to provide a visual indication of the weight of the bag's contents.



CHROMESTHESIA

CHROMESTHESIA, by Senaida Ng, is a paintbrush instrument that uses an Arduino Nano 33 IoT and Adafruit's RGB Color Sensor TCS34725 to perform improvised live painting composition. The colors are translated into sound using MIDI CC mapped to Ableton Live.



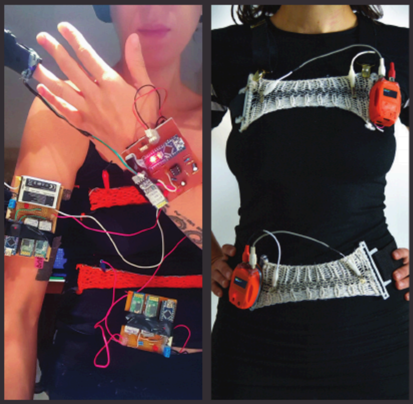


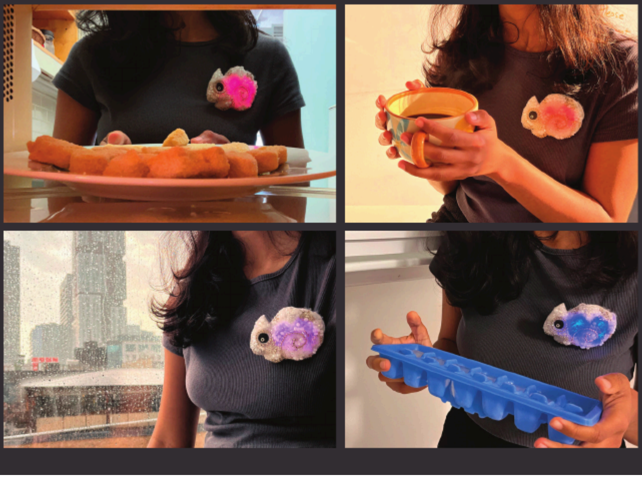
Felt Sensors

Felt Sensors, by Lara Grant, is a collection that includes a stroke sensor (top), a pressure-sensitive button (above left), and a felted stretch sensor (above right).

Pinch to Awaken XR

Pinch to Awaken XR, by Yesica Duarte, uses breath as a means of navigating a virtual space using alternative controllers made of stretch belts knitted with resistive yarn driven by a voltage-divider stretch circuit. They are networked with a ESP32 microcontroller that sends data to Unity via Open Sound Control (OSC).





WeatherWiz

WeatherWiz, by Abha Patil and Nikhil Thomas, is a wearable patch that changes colors in response to the surrounding temperature. The patch is equipped with a Circuit Playground Classic and an 850 mAh battery.

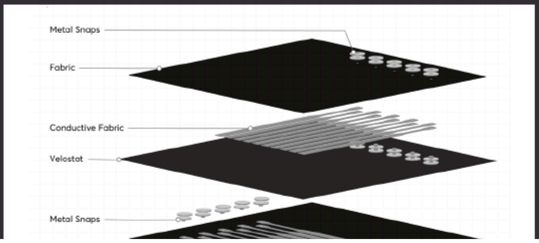
James Prinz

Embroidered Touch 1

Embroidered Touch 1, by Anke Loh, is a dress and skirt embroidered with a colorful free-form grid of both electrically conductive and nonconductive threads that function as touch sensors. The grid-pattern sensor arrangement enables the tracking of both the position and motion of touches. This project was a collaboration with technology artist Christopher Baker and ZSK (Melanie Hoerr, Michaela Kasdorf).

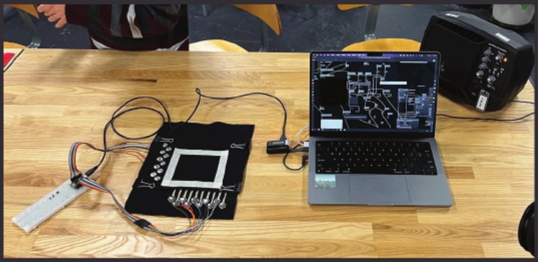
IntelliTex—Wearable Control for Gamified Rehabilitation

IntelliTex, by Yuecheng Peng, Danchang Yan, Haotian Chen, Yue Yang, Ye Tao, Weitao Song, Lingyun Sun, and Guanyun Wang, is a fabrication method that has, as one application, a glove that senses bending actions from three main joints on the hand to capture motion from rehabilitation patients. It can be used as a game controller, offering personalized exercise sequences, making this dull process much more enjoyable. The glove uses a vibration motor to provide haptic feedback to ensure that the patient performs the right action.

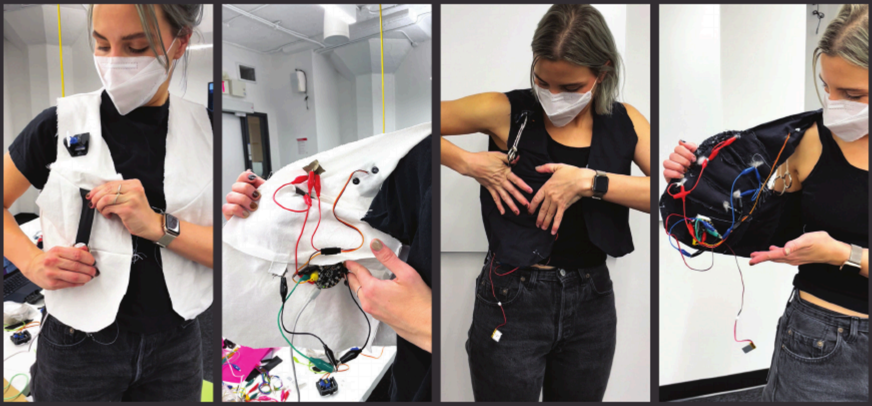


TacTile

TacTile, by Aranya Khurana, is a touch-sensitive fabric matrix and a new interface for musical expression (NIME). The design features two parallel sets of conductive-fabric strips separated by piezoresistive material. Both sets of strips connect to a Teensy 3.2 microcontroller—one set to digital pins and the other set to analog pins, with the latter reading off values. This continuing project for



a master's thesis is based heavily on Maurin Donneaud, Cedric Honnet, and Paul Strohmeier's prior work on the e-textile interface, published in the *Proceedings of the International Conference on New Interfaces for Musical Expression* in 2017.

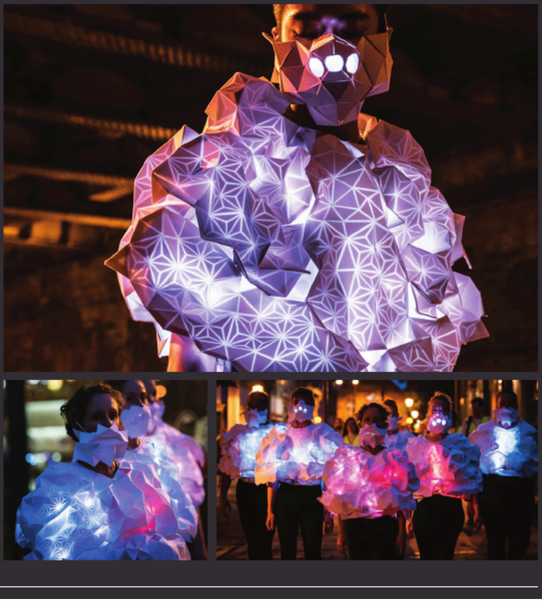


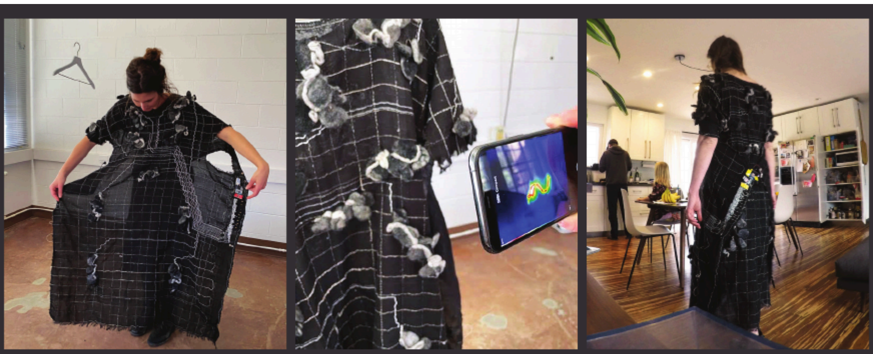
Pull and Push Vests

These vests, created by Olivia Prior, use e-textile sensors to create opposing interactions for controlling a servo motor. The Pull Vest uses stretchy conductive rubber, and the Push Vest contains puffy pockets of conductive wool. Each prototype uses a CPX with the same circuit and the same code.

Human Sensor

Human Sensor, by Kasia Molga, is a futuristic costume that visualizes air pollution in urban habitats. The wearable contains a Raspberry Pi microcontroller, LED strips, and a PM10 air-quality sensor that influences the color of the LEDs—from white (clean) to red (very polluted), with various intermediate shades indicating increasing levels of contamination. The accompanying mask includes a temperature sensor that tracks the rhythm of the wearer's breath, dictating how quickly the LEDs pulse. A GPS tracker is used to record the locations of the sensor readings and transmit the gathered data with pollution levels in real time online.





Wear

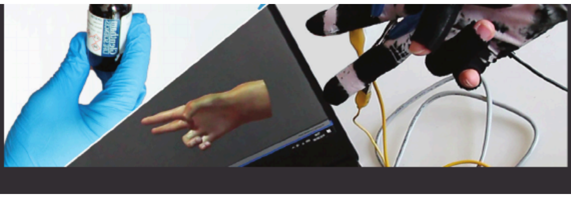
Wear, by Laura Devendorf, is a dress that served as what she calls a "design memoir" that recorded how human and nonhuman bodies exerted pressure on her body during the COVID-19 lockdown. The dress contains twenty-two hand-felted force sensors that recorded the pressures on her body. When she took the dress off, the force interactions were "played back" as heat in the regions where they were captured via resistive heating. The entire garment and circuit were handwoven with 32 AWG silicone-coated wires inlaid during weaving to connect all sensors and actuators.

dB Jacket

The dB Jacket, by Kathleen McDermott, is a custom-built jacket equipped with a decibel meter and an LED display on the back, allowing the wearer to measure environmental noise pollution and display that information to those around them. New York City residents are invited to wear the jacket to help map the sound levels in the city.

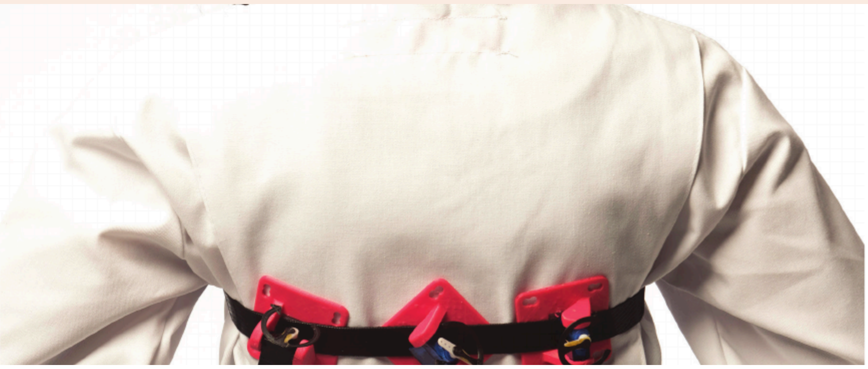
PolySense

PolySense, by Cedric Honnet, Hannah Perner-Wilson, Marc Teyssier, Bruno Fruchard, Jurgen Steimle, Ana C. Baptista, and Paul Strohmeier, is a method for making textiles conductive. These can then be used for creating pressure and deformation sensors. The process of polymerization can be used to add electrical functionalities to everyday materials such as fabrics or feathers to transform them into sensors.



8

Actuators

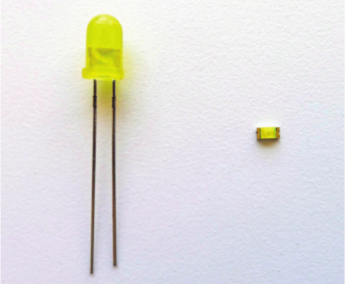


Kinetic Wearables Toolkit, by Social Body Lab (Kate Hartman and Chris Luginbuhl)

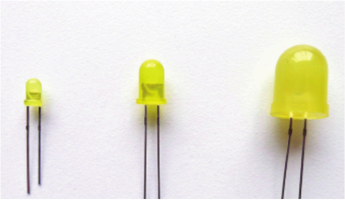
Based on our experiences with making muscle-activated textiles, the SBL team wanted to develop better options for wearing things that move. The Kinetic Wearables Toolkit is a set of 3D-printable mounts for small servo motors, stepper motors, and solenoids that allow them to be more easily worn on the body.

Actuators are the things that go *boom*, *blink*, and *bzzzzt*. They are the things that make things happen. This chapter covers actuators that produce a range of outcomes, including light, sound, movement, and heat. Through the use of these components, you'll be able to produce garments that can glow, shake, and sing.

Adobe Stock-Lepusinensis, Kristy Boyce



LED packaging types: through-hole (left) and surface mount (right)



Light

Whether you're a cyclist or a fashionista, there are times when being seen can make all the difference. Let's review some options for ways to wear light.

BASIC LEDS

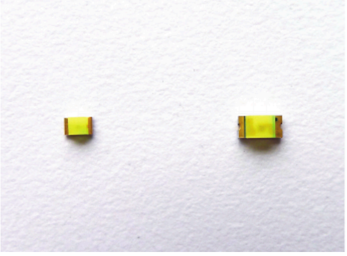
You first encountered LEDs in chapter 2, and you've been using them as a basic output ever since. Let's take a moment to get to know LEDs a little bit better.

First, LEDs, like most electronic components, come in different types of packages. Through-hole LEDs are easy to handle and prototype with, but surface-mount LEDs tend to integrate more delicately with the design of garments.

Each type of package comes in many sizes and sometimes even in various shapes.

LEDs also differ by color, brightness, and viewing angle. Be sure to consult the product description and datasheet of the LEDs you are working with to get the details of how

Through-hole LED sizes: 3 mm, 5 mm, and 10 mm



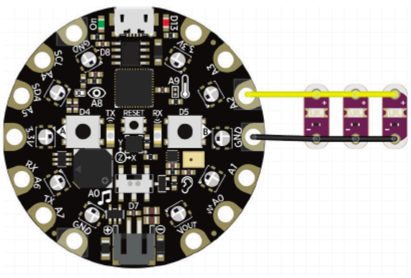
Surface-mount LED sizes: 7805 and 1206 packaging

they'll look and what they need to get glowing.

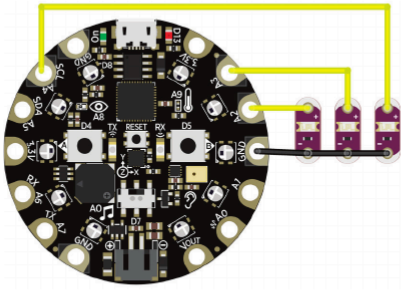
There are many options for controlling LEDs. As you know from the examples in chapter 6, you can use a single pin of a LilyPad Arduino to control three LilyPad LEDs in parallel. These three LEDs will behave in the same way.

If you would like these LEDs to have different behaviors, you would have to use three distinct digital output pins.

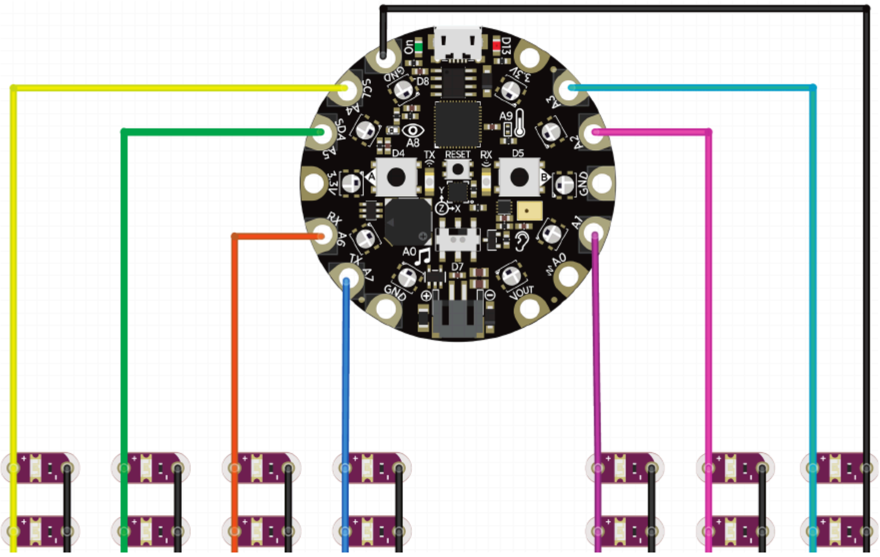
Each output pin can typically power multiple LEDs connected in parallel. The quantity of LEDs that can be lit in this way depends on the microcontroller pin's current output and the current needs of the LEDs. See the datasheet of each part for more information.

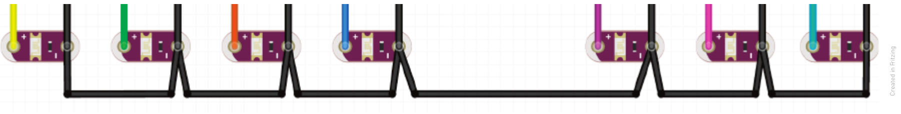


Circuit diagram of CPX with three LilyPad LEDs in parallel controlled by a single pin



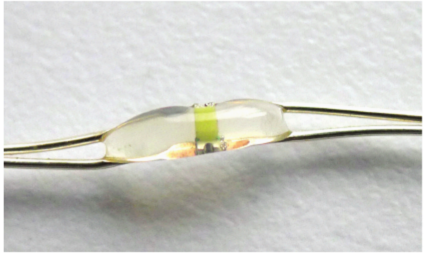
Circuit diagram of CPX with LEDs on three distinct pins, making them individually controllable



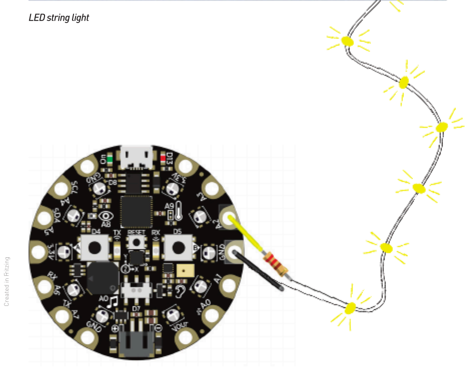


Circuit diagram of CPX with 21 LEDs: Sets of three LEDs in parallel, each controlled by a different pin

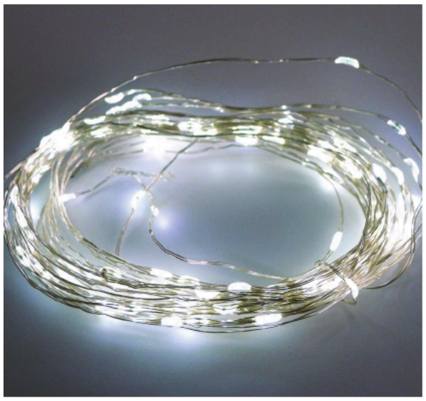
LED string light



LED string light, detail



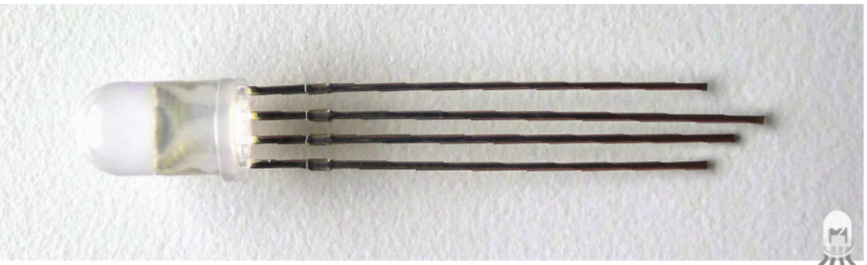
LED string lights connected to CPX



LED string light, lit

Low-voltage LED string lights, such as 3V fairy lights, can be controlled by a single Arduino pin. Simply snip off the battery pack and use a bit of coarse sandpaper to remove the enamel insulation from the wire ends, and then use alligator pins to clip to the CPX. Like all LEDs, those in the string are polarized, so remember to test to see which contact wire connects to the anodes vs. cathodes.

Finally, if you need to control a large number of basic LEDs individually, this can be accomplished through techniques such as *multiplexing or Charlieplexing* or the use of components such as shift registers and PWM extender chips. For more information on these options, see appendix A.

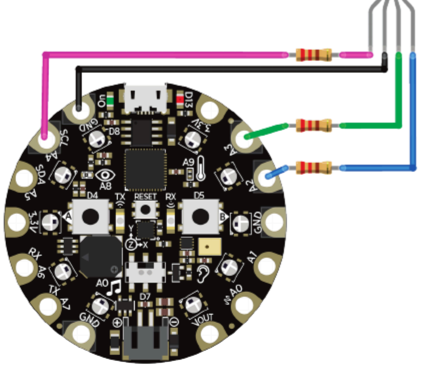


RGB through-hole LED

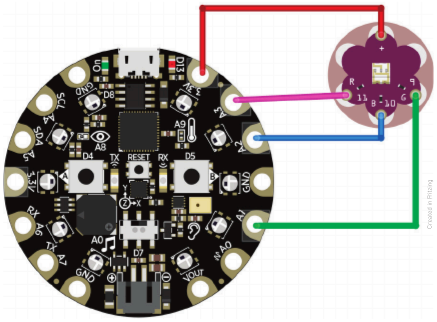
There are also LEDs that can light in multiple colors like an RGB LED. These LEDs have four pins—three that correspond to each color and a fourth that is either a common anode (meant to connect to power) or a common cathode (meant to connect to ground). The color the LED displays depends on the intensity of the PWM signal of each color pin.

There are a variety of ways to wear LEDs, whether for safety, style, or making a statement.

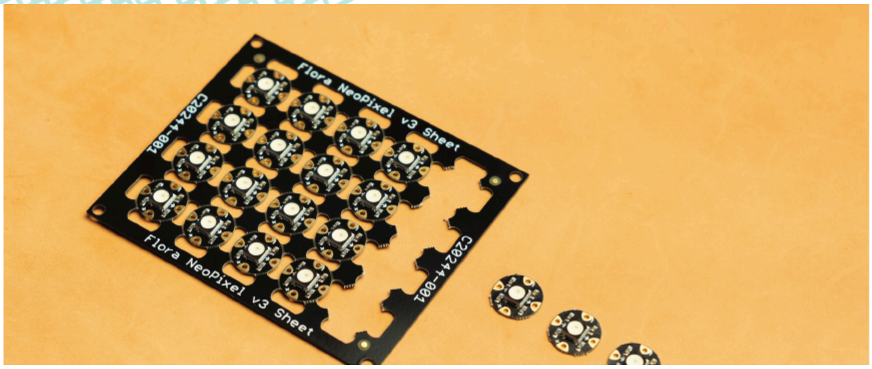
Basic LEDs are just one way to get started with illuminated clothing. In the following sections, I review additional tools that can be used to create wearable light.



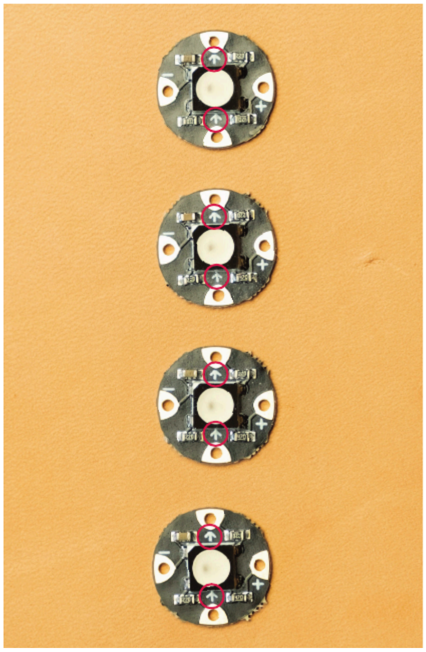
Circuit diagram of CPX with an RGB through-hole LED with a common cathode and resistors



Circuit diagram of CPX with LilyPad Tri-Color LED with a common anode



Flora NeoPixels V3



Maxwell Lander

Note that when you are preparing a circuit, the arrows on the NeoPixels should all face away from the microcontroller.

ADDRESSABLE LEDS

When working with LEDs, you sometimes want to create a visual effect that is bright, bold, and extremely dynamic. Addressable LEDs are easily wired, ultrabright, multicolored, and able to behave independently from the others. NeoPixels, a very popular brand of addressable LED pixels and strips offered by Adafruit, come in a variety of form factors, including wearable and sewable, and are well supported with extensive documentation. What more could you want out of a light-emitting diode?

The Flora NeoPixels, intended for use with the Flora or other sewable microcontroller boards, require three connections—power, ground, and a connection to either a digital output pin (for the first NeoPixel) or the NeoPixel in the chain before it (for the NeoPixels that follow). See the circuit diagrams in the examples that follow to see how these connections are made.

For the software, the NeoPixel library is available for installation via Arduino's Library Manager. Follow the Adafruit NeoPixel Überguide (learn.adafruit.com/adafruit-neopixel-uberguide) for the most up-to-date instructions.

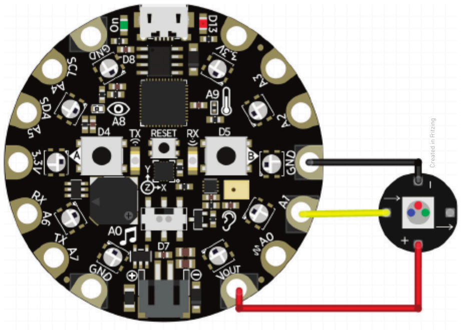
Once your Flora is up and running, you'll be ready to get going with the NeoPixel. Here are some examples.

One Neopixel

To get started, let's light up a single NeoPixel. Once you understand the basics, you can let the fanciness explode.

Start by assembling the circuit with alligator clips. Once your circuit is assembled, program your microcontroller with the following code:

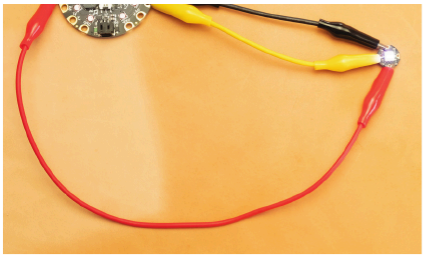
```
/*
Make: Wearable Electronics
```



Created in Fritzing

Circuit diagram of CPX with one external NeoPixel

```
 One-NeoPixel example
 */

#include <Adafruit_NeoPixel.h>

// The digital pin used to control the
pixel strip
int pinNumber = A1;
```



CPX with one external NeoPixel connected with alligator clips

Maxwell Lander

```
// The number of pixels in the strip
int numberOfPixels = 1;

Adafruit_NeoPixel strip =
  Adafruit_NeoPixel(numberOfPixels, pinNumber,
  NEO_GRB + NEO_KHZ800);

void setup() {
  // initialize pixel strip
  strip.begin();
  // set pixels to off to begin
  strip.show();
}

void loop() {
  // set pixel 0 to red
  strip.setPixelColor(0, 255, 0, 0);
  strip.show();
  delay(500);

  // set pixel 0 to green
  strip.setPixelColor(0, 0, 255, 0);
  strip.show();
  delay(500);

  // set pixel 0 to blue
  strip.setPixelColor(0, 0, 0, 255);
  strip.show();
  delay(500);

  // turn pixel 0 off
  strip.setPixelColor(0, 0, 0, 0);
  strip.show();
  delay(1000);
}
```

A few commands in this code are worth explaining:

```
Adafruit_NeoPixel(numberOfPixels, pinNumber,
NEO_GRB + NEO_KHZ800);
```

This command has three parameters: the number of pixels, the pin number, and the pixel-type flag. (Don't change that one.) Be sure to adjust this if you change pins or the number of pixels you are using.

```
strip.setPixelColor(0, 255, 0, 0);
```

This is used to set the pixel color—big surprise! You need to use this command to set each pixel individually—the first parameter is the pixel number (starting with 0) and the second, third, and fourth are the red, green, and blue values.

```
strip.show();
```

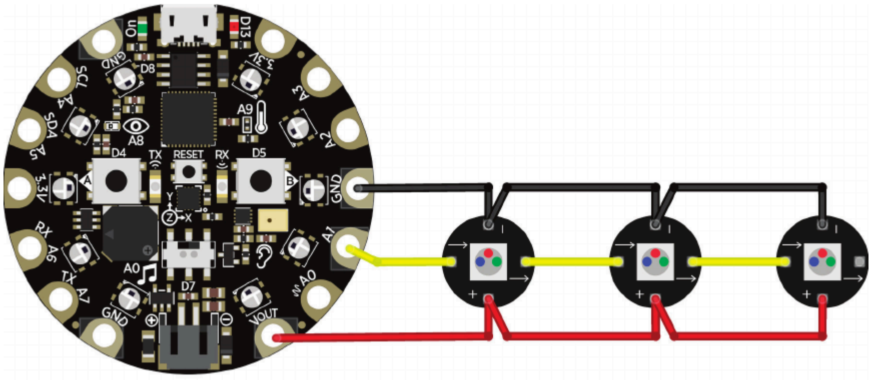
Once you have set all your pixels, this command lights the entire strip with the predetermined colors. Color changes will not appear until the **`strip.show()`** command.

If your pixel does *not* light up, double-check the connections in your circuit and make sure you properly installed the library.

Now that you know how to light up a single pixel, let's try three!

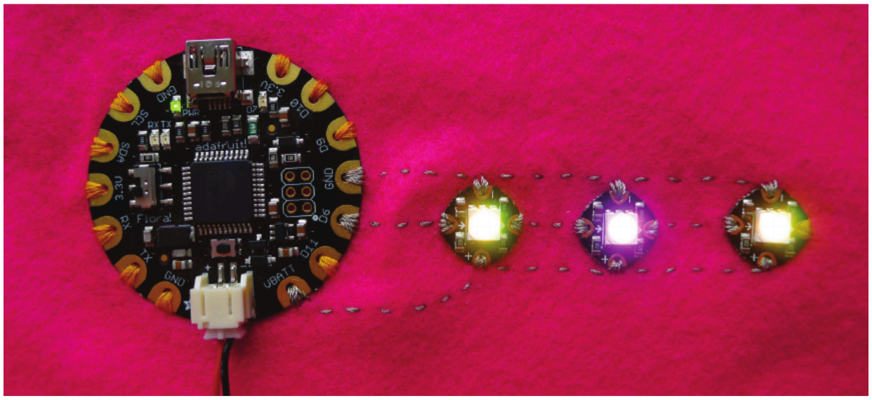
Multiple Neopixels

A nice part of working with NeoPixels is that they chain very easily. Be sure to pay attention to the direction of the arrows on the NeoPixels when assembling this circuit. They should all face away from the Flora board. This circuit can be assembled using alligator clips or sewing with conductive thread.



Circuit diagram of CPX with three sewable NeoPixels

Flora with NeoPixels connected with alligator clips



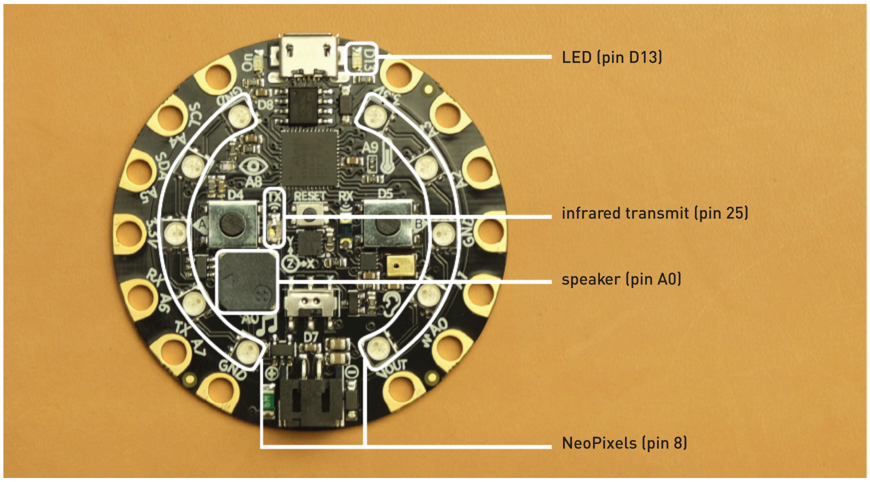
Flora with three NeoPixels sewn with conductive thread

You won't see much change in the code, except that now you are setting the colors of multiple pixels before showing the new configuration of the strip.

Here's the code:

```
/*
Make: Wearable Electronics
 Flora NeoPixel example with 3 pixels
 */

#include <Adafruit_NeoPixel.h>


// The digital pin used to control the

// pixel strip

int pinNumber = 6;


// The number of pixels in the strip

int numberOfPixels = 3;


Adafruit_NeoPixel strip =

  Adafruit_NeoPixel(numberOfPixels, pinNumber,

  NEO_GRB + NEO_KHZ800);


void setup() {

  // initialize pixel strip

  strip.begin();

  // set pixels to off to begin

  strip.show();

}


void loop() {


  // set pixel 0 to yellow

  strip.setPixelColor(0, 255, 255, 0);

  // set pixel 1 to pink

  strip.setPixelColor(1, 255, 51, 153);

  // set pixel 2 to yellow

  strip.setPixelColor(2, 255, 255, 0);

  strip.show();

  delay(1000);
```

```
  // set pixel 0 to pink

  strip.setPixelColor(0, 255, 51, 153);

  // set pixel 1 to yellow

  strip.setPixelColor(1, 255, 255, 0);

  // set pixel 2 to pink

  strip.setPixelColor(2, 255, 51, 153);

  strip.show();

  delay(1000);


  // turn pixel 0 off

  strip.setPixelColor(0, 0, 0, 0);

  // turn pixel 1 off

  strip.setPixelColor(1, 0, 0, 0);

  // turn pixel 2 off

  strip.setPixelColor(2, 0, 0, 0);

  strip.show();

  delay(1000);

}
```



Maxwell Lander

CPX on-board actuators. If you are working with a CPX, you'll see it has a set of eight on-board NeoPixels as well as a few other actuators that will be useful in the sections that follow.

Because these NeoPixels are individually addressable and because it is so easy to quickly add more, the possibilities of what you can do with them are endless. Just use your imagination to explore the lighting effects you would like to create!

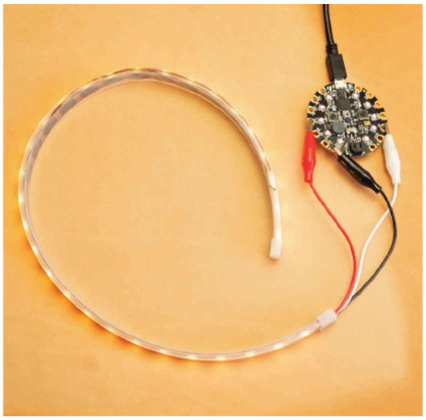
For more complex behaviors, check out Adafruit's Flora RGB Sewable NeoPixels tutorial (learn.adafruit.com/flora-rgb-smart-pixels/overview).

FIBER OPTICS

In addition to components that generate light, there are materials that can transmit light. For a different approach to lighting, let's take a look at fiber optics.

Fiber optics, or optical fibers, are flexible, transparent fibers that can transmit light. They are used for applications that range from sophisticated high-speed communication systems to magical light-up wands you can get at your local summer carnival. Fiber optics come in either *end-glow* or *side-glow* varieties. Apply light to one end of the strands, and you'll see light at the other ends or along the sides.

What's neat about fiber optics is that LEDs are often used as their light source. This is great for you because it makes use of your existing knowledge of LEDs. To look at an example, let's check out my super-awesome fiber-optic headband I got from an electronics surplus site. It includes two LEDs as light sources to

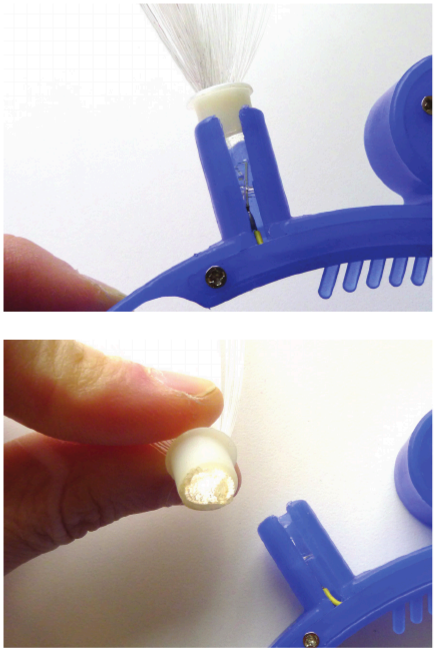


Maxwell Lander

NeoPixel strips and strings can greatly reduce the complexity of the wiring.

illuminate two bundles of fiber optics. If you take a closer look at the images on the following page, you can see how this is assembled.

Fiber-optic headband



In this fiber-optic headband, the LED housing positions the LED so its light is pointed directly into the ends of the fiber optic strands. The plastic ring around the fibers holds them together in a tight bunch.

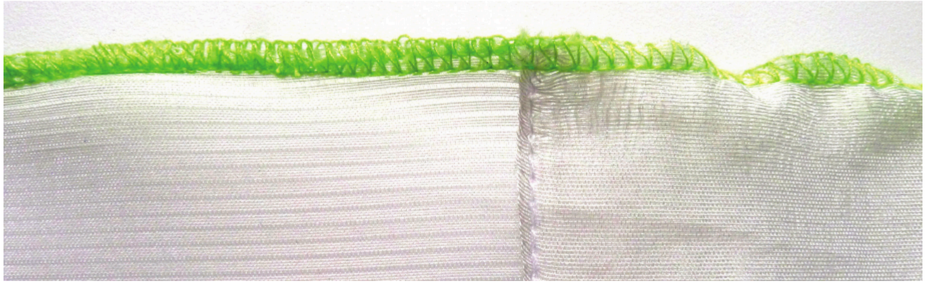
Because of the amazing flexibility and light-transmitting properties of fiber optics, many artists and designers have been incorporating them into their designs, particularly through the practice of weaving. For those not well versed in weaving, manufactured fiber-optic textiles are becoming more widely available.

When handling fiber optics, be careful to not crease them, as this will permanently affect how they transmit light. Be gentle with the material and be sure to roll, not fold, the textile when it is not in use. Bubble wrap is a big help too.

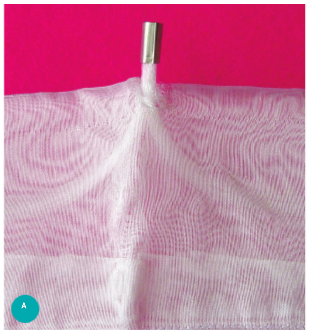
The easiest way to attach an LED to a fiber optic bundle is to use a bit of heat-shrink tubing.

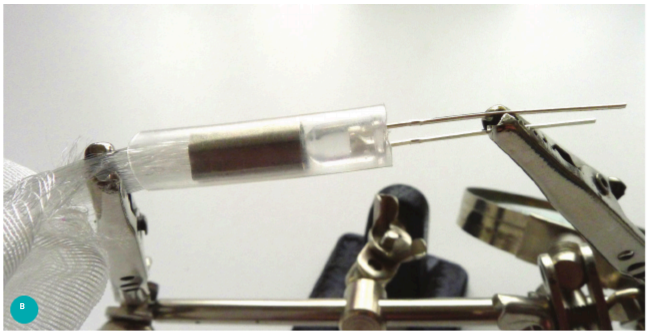
Be sure to aim the heat gun toward the heat-shrink tubing but *away* from the length of the fiber optic filament. Plastic fiber optics can melt when exposed to intense heat.

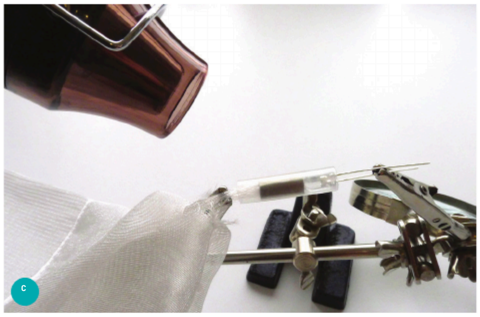
Once your LED is secured to the fiber optic textile, it can be powered or controlled using the standard means you would use for any LED, such as with a digital output pin on a LilyPad Arduino. From there, you can figure out how you might feature this material in a design of your making.

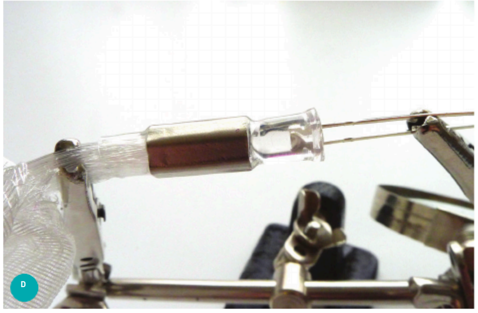


Fiber-optic fabric









HOW TO: Fiber-Optic Textile

To light up a fiber-optic textile, all you need is a single LED (preferably a super-bright one). Let's look at one method for attaching an LED to a fiber-optic bundle.

1. On the textile, look for where the fiber-optic strands are gathered into a bundle (Figure A).
2. Position the LED and the fiber-optic bundle with the helping hands such that the LED points into the strands. Cover the LED and the fiber-optic bundle with an appropriately sized piece of heat-shrink tubing (Figure B).
3. Use a heat gun to shrink the tubing. Be sure to aim the heat gun toward the heat-shrink tubing but away from the length of the fiber-optic filament—plastic fiber optics can melt when exposed to intense heat (Figure C).
4. Inspect the tubing to ensure it is snug on all sides (Figure D).
5. Once your LED is secured, light it with a 3 V battery or the digital-output pin on an Arduino (Figure E).

EXPERIMENT: Be Safe, Be Seen

Using one of the tools you've learned about, incorporate light into a piece of clothing for fashion or utility. Think about when it should be lit and when not, and whether it is the user or the wearable that determines changes in state.

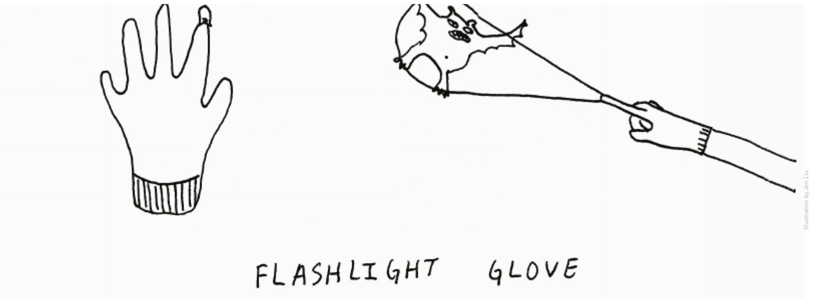


Illustration by Jen Liu

Sound

Sound can be soothing, informative, and even abrasive. How can you get your clothing to speak, sing, or shout? When working with audio for wearables, here are some helpful questions to ask:

- *Would you like to make a simple sound, generate a tone, or play an audio file?*
- *How will the sound be triggered or controlled?*
- *Where will the sound-emitting device live?*
- *How loud should the sound be? Is it intended only for the wearer or also for those who are nearby?*

With those considerations in mind, let's explore your options for embedding audio close to the skin.

BUZZERS

Buzzers are a simple way to provide audio feedback. They are devices that create an audible sound as the result of an electrical signal. There are two types of buzzers you will encounter: electromagnetic and piezoelectric.

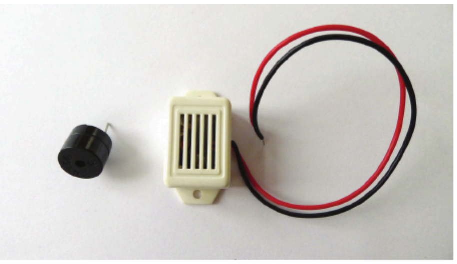
Electromagnetic buzzers create a noise when continuous voltage is applied.

Piezoelectric buzzers require an oscillating signal and can function much like speakers. You'll get to know them in the next section.

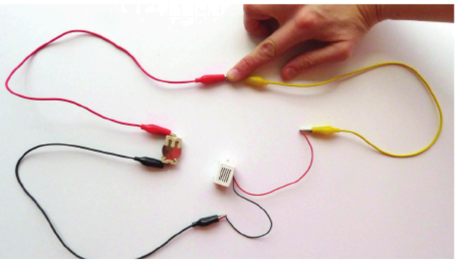
Be sure to look out for the polarity of electromagnetic buzzers. The panel-mount buzzers usually have a + sign to signify the positive side, and with the wired version, you can tell by the colors of the wires (red for positive, black for negative).

Connect the positive side to a digital output pin and connect the negative side to ground. Simply set that digital output pin to High, and the buzzer will sound.

Here is the code:

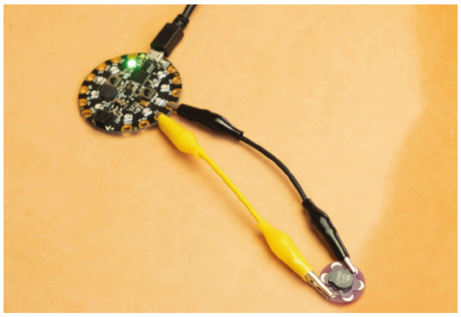


Electromagnetic 3 V buzzers—panel mount and with wires



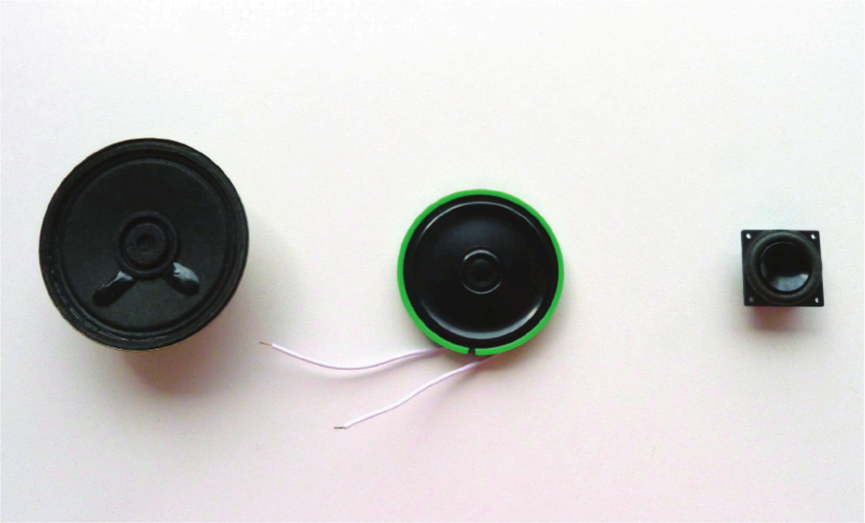
```
/*
Make: Wearable Electronics
Buzzer example
*/

int buzzerPin = 6;

void setup() {
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  digitalWrite(buzzerPin, HIGH);
  delay(500);
  digitalWrite(buzzerPin, LOW);
  delay(3000);
}
```

3 V electromagnetic buzzers are also great stand-alone actuators and can act as an interesting alternative for LEDs when creating simple analog circuits.



CPX with a panel-mount 3 V LilyPad buzzer

Maxwell Lander



Speakers come in many shapes and sizes.

You can sometimes learn a lot about a speaker by looking at the back of it. This is a 2-inch, 0.5 W, 8 Ω speaker.

TONES

The simple buzzers you've looked at so far are great for producing a single, simple tone, but if you want to produce a broader range of sounds, you can also generate specific notes using a microcontroller and a speaker (or a piezoelectric buzzer).

Both speakers and piezoelectric buzzers contain materials that move when voltage is applied. When voltage is applied, the material is in one position, and when it is not, the material is in another position. It is the frequency of switching back and forth between these two positions that moves air in such a way to create different sounds.

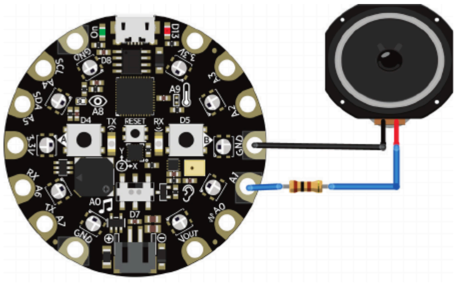
Let's take a look at how you can use the Arduino to produce particular notes.

In order to make a note, you need to use the Arduino to turn the pin on and off at a particular frequency. Luckily there is an Arduino function called **`tone()`** that handles most of this for you. It looks like this:

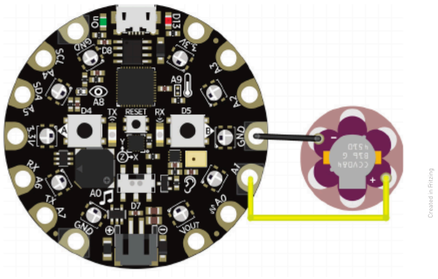
```
tone(pin, frequency, duration)
```

Just provide the pin, frequency in hertz, and parameters for duration (in milliseconds) and the Arduino will generate your desired tone. Try this code as an example:

```
/*
Make: Wearable Electronics
Tone example
*/

int C = 1047;
int D = 1175;
int E = 1319;
int F = 1397;
int G = 1568;
int A = 1760;
int B = 1976;
int c = 2093;

int buzzerPin = 6;

void setup() {
  pinMode (buzzerPin, OUTPUT);
}

void loop() {
  tone(buzzerPin, C, 250);
  delay(300);
  tone(buzzerPin, E, 250);
  delay(300);
  tone(buzzerPin, G, 250);
  delay(300);
```



Circuit diagram of CPX with speaker and 100 Ω resistor



Circuit diagram of CPX with LilyPad Buzzer

```
  tone(buzzerPin, c, 250);
  delay(300);
  tone(buzzerPin, G, 250);
  delay(300);
  tone(buzzerPin, E, 250);
  delay(300);
  tone(buzzerPin, C, 500);
  delay(1000);
}
```

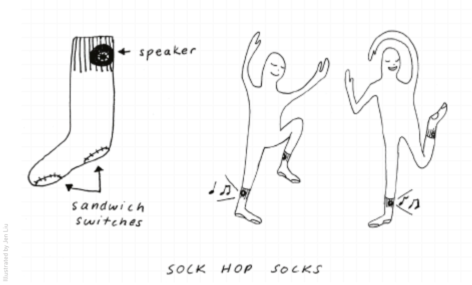
See also these examples:

- Arduino melody tutorial (docs.arduino.cc/built-in-examples/digital/toneMelody)
- Arduino pitch follower using the **`tone()`** function (docs.arduino.cc/built-in-examples/digital/tonePitchFollower)
- Arduino simple keyboard using the **`tone()`** function (docs.arduino.cc/built-in-examples/digital/toneKeyboard)

Note that there is a second tab in these sketches titled pitches.h that defines frequencies for pitches at many octaves.

Despite the simplicity of these tones, they can still be combined to create a variety of melodies, sound effects, and feedback noises.

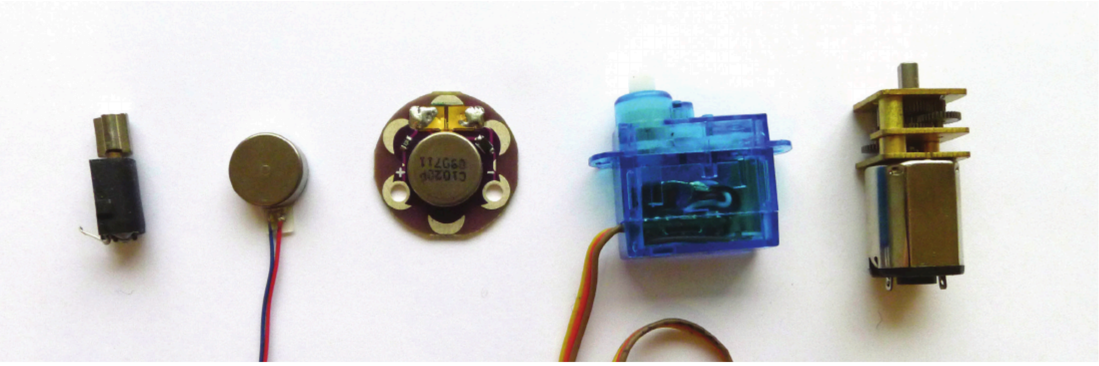
While the ability to produce tones with the Arduino is useful, being able to play digital audio files greatly expands your project's horizons. A variety of Arduino-compatible tools, such as WAV or MP3 shields or even dedicated audio or sound effects boards, enable the playback of digital audio files.



Illustrated by Jen Liu

EXPERIMENT: Wearable Instrument

Now that you know how to generate tones using the Arduino, use your knowledge of sensors and wearables construction techniques to create a body-based instrument with an unusual interface.



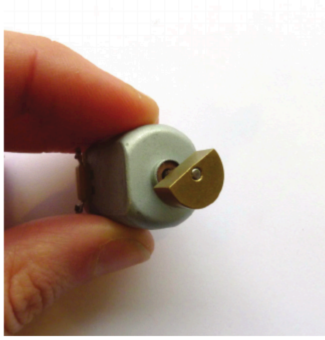
Small motors well suited for wearable applications (left to right: vibration motor [exposed], vibration motor [enclosed], LilyPad Vibe Board, microservo, and a small gearhead motor)

Motion

Making things move can be an enticing prospect. It is also a challenging one in the dynamic arena of the human form. From the tiny buzz of a vibration motor to the sharp and precise movements of a servo to the significant physical transformations created by a gearhead motor, this section covers how to use motors to accomplish a range of movement possibilities.

VIBRATING MOTORS

Vibrational feedback can be powerful, subtle, and even seductive. It can simulate a stroke, a tap, or a tickle. It holds the potential to be perceived only by the wearer and is ideal for situations that warrant privacy and discretion or situations where it is inconvenient or impossible for the wearer to see or hear feedback.



A weighted head causes the DC motor to shake as it spins.

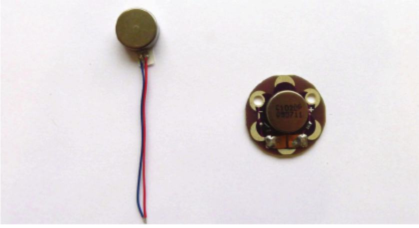
Vibrating motors are basically DC motors with a weighted head attached to the shaft. As the motor spins, the weight spins, thus causing the motor to rock back and forth. Many vibrating motors come with their weighted

head exposed. This can be a bit problematic if you're incorporating the motor into a garment with folds of fabric or other intrusions that can interfere with the spinning of a head. The advantage of small, open vibrating motors (the type often found in cell phones or pagers) is that they are often available at surplus stores for very cheap. When using them, be sure to build in protection so the head can spin freely. The leads also tend to be a bit delicate, so it's worth using heat-shrink tubing to reinforce your connections.

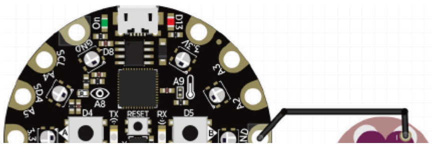
There are also completely enclosed small, flat vibrating motors, sometimes called *pancake motors*. These are well suited for wearable applications and very easy to work with. This is the same kind used on the LilyPad Vibe Board, but you can also purchase the motor on its own and incorporate it into your project as you like.

These motors can be directly connected to either

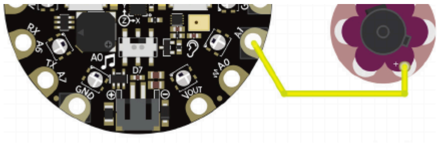
Vibrating motors with exposed heads



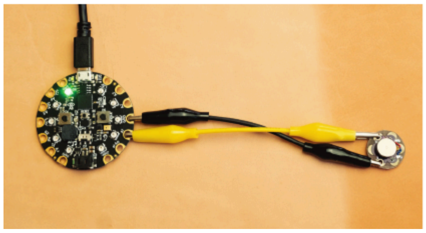
Pancake vibrating motors



Created in Fritzing



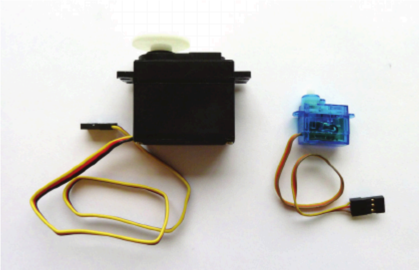
Circuit diagram of CFX with LilyPad Vibe Board



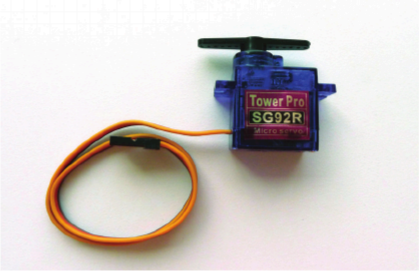
CPX with LilyPad Vibe Board connected with alligator clips

Maxwell Lander

a digital or analog output pin on the Arduino, depending on whether you want to control the intensity of the vibration. Simply connect one end of the motor to the output pin and the other to ground. The 40 mA provided by the Arduino output pin is plenty to get these motors shimmying, but if you'd like a more intense vibrational kick, you just need to supply them with additional current. This is covered later in this chapter.

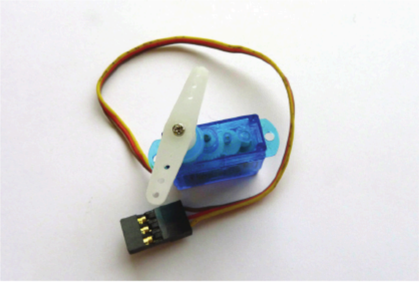


A medium servo and a micro servo



This microservo will run on 3–6 V.

Servo attachments include arms, propellers, and wheels.



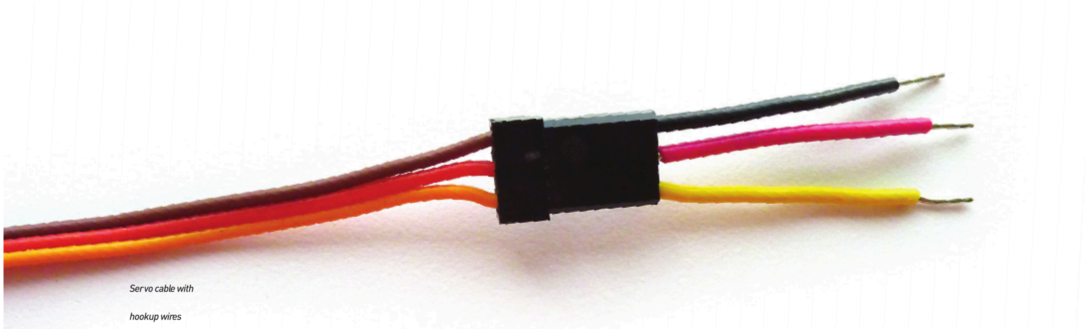
Small servo with propeller attached

SERVO MOTORS

Sometimes you want to use motors to accomplish precise movements. This could be for functional purposes, such as opening and closing a pocket, or for aesthetic purposes, such as the movement of materials to create a dynamically shifting design.

Servo motors are capable of accomplishing small, discreet movements. They are extremely precise in their position and most often have a turning range of 180 degrees, though there are 360-degree models available. A servo motor can be told to turn to any location within its potential range of movement.

Microservos are miniature servo motors useful for wearables because they are small and lightweight. As with any motor, it is important to pay attention to the power requirements of the particular model you are working with. Many servos need 5 V to run, in which case you will need to make sure you have a 5 V power supply included in your circuit.



Servo cable with hookup wires

A few microservos will work with as little as 3 V, which is helpful if you are using a 3.7 V lithium polymer battery.

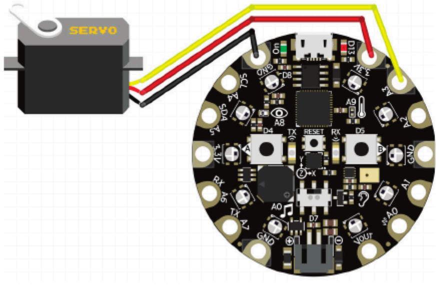
Servos usually come with a number of attachments. These can be screwed directly to the shaft to provide leverage or enable attachment to other materials or for mechanical purposes.

A servo has three connections: power, ground, and *signal*. The servo cable is usually terminated with a female header. You can either insert hookup wire to make temporary connections or snip off the header to access the wires for soldering or sewing.

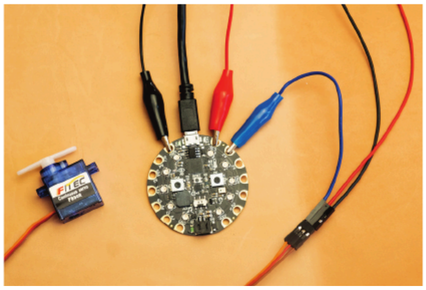
Most Arduinos can control multiple servos simultaneously. Arduino has a built-in servo library. Here are some commands that are useful to know when working with the library:

```
#include <Servo.h>
```

This includes the servo library into your code, so that it's incorporated into your code when it's compiled.



Created in Fritzing



Maxwell Lander

Circuit diagram of CPX with servo motor

Servo circuit with alligator clip connections

Servo mrSpinny

This declares a variable name for the particular servo you are working with. In this case, it is **mrSpinny**. But it could be **myFavoriteServo**, **servo**, or even **Bob**. You'll see in the following commands that it is **mrSpinny** followed by a period followed by the command **mrSpinny** would be replaced with whatever variable you've declared.

mrSpinny.attach(pin)

This declares which pin the servo will be connected to.

mrSpinny.write(angle)

The angle is the position between 0 and 180 you would like the servo to turn to. Keep in mind that it takes time for the servo to turn, so you should always include a delay between **.write** commands so that it has adequate time to turn.

With the circuit complete and this knowledge in hand, you can program the Arduino to control the servo! Here's an example:

```
/*
Make: Wearable Electronics
 Servo example
 */

#include <Servo.h>

// name your servo
Servo mrSpinny;
int servoPin = 10;

void setup()
{
  // set the servo pin
  mrSpinny.attach(servoPin);
}

void loop()
{
```

```
// turn to 0 degree position
mrSpinny.write(0);
// wait 1000 milliseconds
delay(1000);
mrSpinny.write(45);
delay(300);
mrSpinny.write(90);
delay(300);
```

```
    mrSpinny.write(135);
    delay(300);
    mrSpinny.write(180);
    delay(1000);
}
```

See also:

- Arduino sweep and knob tutorials (docs.arduino.cc/learn/electronics/servo-motors)

WHEN LOST IN A CROWD,
ACTIVATE THE HAT!
FRONT
BACK
glove on rod attached to servo motor
"HERE · I · AM!" HAT

Illustrated by Jen Liu

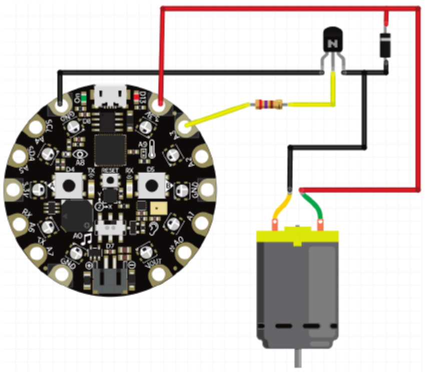
EXPERIMENT: Shake, Spin, or Shimmy

Create a piece of clothing that moves in response to stimuli. Think about where the motor will sit, what and how it will move, and how your material design can best support its movements.

Higher-Current Actuators

Many interesting actuators require more current than the small amount of milliamps an output pin on a microcontroller can provide. Also, keep in mind that the amount of current available from an output pin may vary depending on which microcontroller board you are working with.

A *transistor* is a component that allows a small amount of current to trigger a device that requires a larger amount of current. A transistor circuit can enable a microcontroller's output pin to trigger an actuator connecting to a higher-current power supply.



Created in Fritzing

An example of a DC motor with a transistor circuit

As with most electronic components, there are many types of transistors to choose from. They vary in terms of the amount of current and voltage they can manage and their pin layouts. If you enjoy building circuits, do some research into the variety of transistors available and how you can build a transistor circuit from scratch.

The basic principle is that when a small amount of electricity is applied to the base, a larger amount of electricity flows between the collector and the emitter.

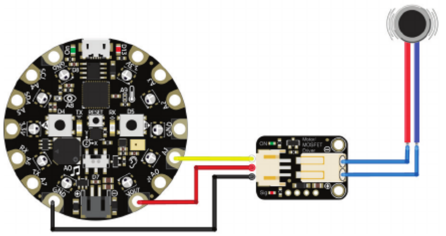
Notice that this circuit uses a component called a *diode*, which allows current to flow in only one direction. You have previously encountered diodes in the form of light-emitting diodes (LEDs). The diode in this circuit is similar, except it does not emit light—it simply limits the flow of the electricity to one direction.

The purpose of this diode is to prevent *blowback voltage*. When voltage is supplied to a motor, it turns. This relationship also works in reverse. If you turn a motor, it can function as a generator and produce voltage. Should that happen accidentally, the diode in this circuit prevents the voltage from traveling back to—and damaging—the microcontroller.

Diode

Building your own transistor circuit provides much opportunity for customization. However, these days, I find that for wearable-electronics projects, it is most efficient in terms of time and space to work with a prebuilt driver board meant to handle higher current loads.

Let's take a look at the Adafruit MOSFET Driver as an example. This board includes the MOSFET transistor, a diode, and some very useful connectors. The result is a tiny board you can use to create quick and reliable connections.



Circuit diagram of CPX with vibe motor and MOSFET Driver board

In the previous section, we were able to get the vibration motor going without any additional accessories. However, it was not buzzing at its full potential. Let's try a modified version of that circuit that includes a MOSFET Driver.

Once your circuit is connected, you can program the microcontroller to control the motor. To simply turn the motor on and off, you can set any pin as a digital output:

```
/*
Make: Wearable Electronics
Motor Digital Example
 */

int motorPin = 6;

void setup(){
  pinMode(motorPin, OUTPUT);
}

void loop (){
  // turn motor on
  digitalWrite(motorPin, HIGH);
  delay(5000);
  // turn motor off
  digitalWrite(motorPin, LOW);
  delay(1000);
}
```

If you'd like a bit more control over the speed, make use of the `analogWrite()` function. Here's an example:

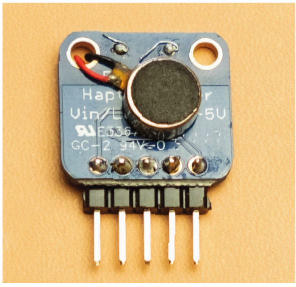
```
/*
Make: Wearable Electronics
Motor Analog Example
 */

int motorPin = 6;

void setup(){
}

void loop (){
  // turn motor off
  analogWrite(motorPin, 0);
  delay(500);
  // spin motor slowly
  analogWrite(motorPin, 100);
  delay(5000);
  // turn motor off
  analogWrite(motorPin, 0);
  delay(500);
  // spin motor at full speed
  analogWrite(motorPin, 255);
  delay(5000);
}
```

If you'd like to develop a more complex behavior for your vibe motor, look into haptic motor driver boards. These boards are controlled over I2C (similar to NeoPixels) and provide a library of haptic effects, including clicks, taps, and ramping up and down.

Let's look at more higher-current actuators that benefit from the use of a transistor.

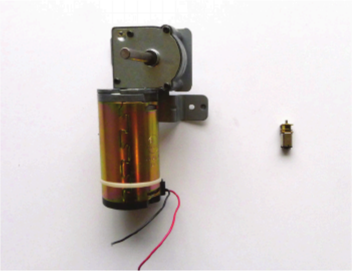
GEARHEAD MOTORS

A DC motor spins freely when voltage is applied. DC motors usually spin quite quickly. A gearhead motor, or gear motor, is



Maxwell Lander

Haptic motor driver board with vibrating motor and headers a DC motor augmented with a set of gears that reduce the number of revolutions per minute (RPMs). DC motors tend to be on the larger side, though smaller ones are nice to work with in wearables. A gearhead motor tends to be much stronger than a typical servo motor.



A very large and a very small gearhead motor

Close up of a very small gearhead motor

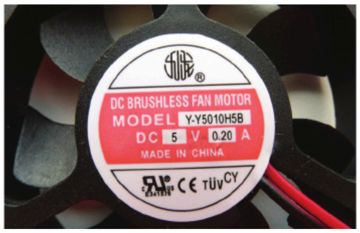
FANS

Because clothing is often used to provide temperature regulation and protection, it's no surprise that wearable-technology designers are often interested in working with actuators that provide a heating and cooling effect.

Thanks to the temperature needs of desktop and laptop computers, a significant number of small 3 V and 5 V fans are readily available for an equally small price.

By taking a closer look at the label on the fan, you can learn about its power needs.

Small fan

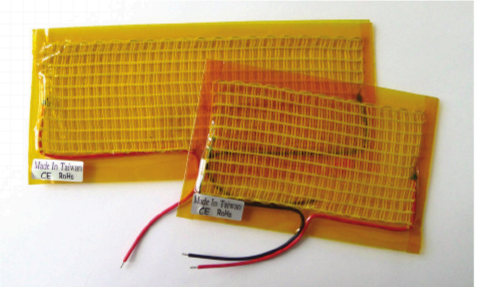


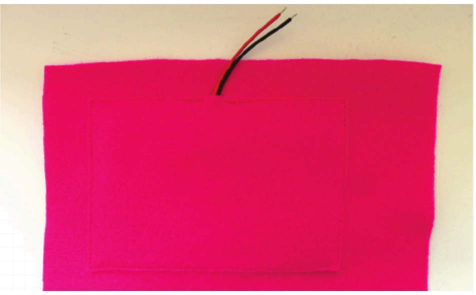
This is a 5 V, 200 mA fan.

200 mA is far beyond the amount of current an Arduino output pin supplies. This is another situation in which you can use a transistor. One way is to use the Mosfet Driver circuit from the previous section and swap out the motor for the fan.

While this type of fan is still a bit bulky to incorporate into clothing, it can have some fun, cooling results.

Keep in mind that when incorporating this type of fan into a garment, it is helpful to use a stiffer, thicker material that can provide proper support for the fan.





A pocket can be a great way to hold a heating pad in place

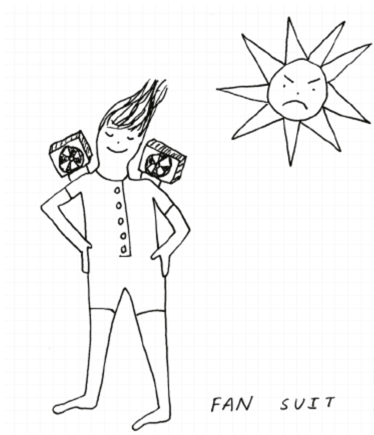
HEAT

From keeping hands toasty on a winter day to providing a slow-growing warming sensation over your heart when someone is thinking of you, heat can provide both physical comfort and even elicit an emotional response when handled in an interesting way. Keep in mind, however, that heating pads are slow and subtle actuators. Your interaction scenario should be designed accordingly.

Electric heating pads are thin and flexible, making them easy to integrate into clothing. Like fans, heating pads are higher-current actuators and will require the use of a transistor circuit. Swap a heating pad into the circuit you used for the fan and the gearhead motor and you'll be good to go!

EXPERIMENT: It's Gettin' Hot in Here

Using a fan, a heating pad, or both, create a climate-controlled wearable that responds to the current temperature. Refer to the previous chapter for more information on how to sense temperature.



Illustrated by Jen Liu

Looking Ahead

As you can see, there's no end to the ways you can use actuators to make things happen when creating wearable electronics. Now that you know how to build a full interactive system that lives on the body, it's time to move beyond the bodysphere and out into the rest of the world. Next up: wearability!

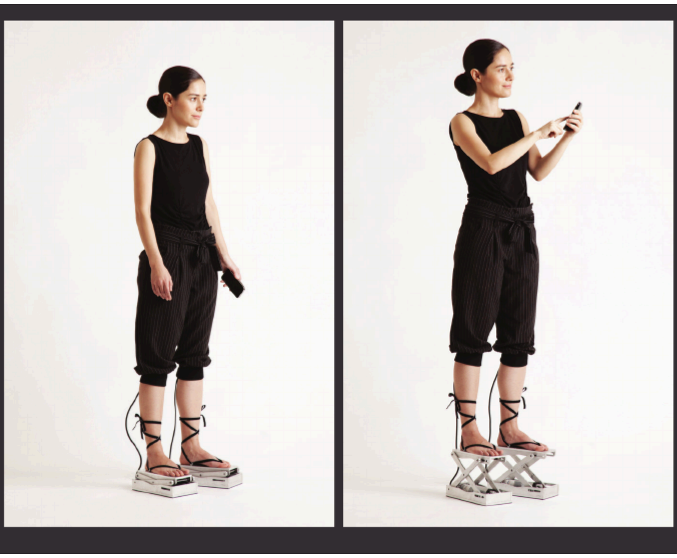
GALLERY 8:

ACTUATORS IN WEARABLES

Through sound, light, movement, and more, wearable-electronics projects come to life in ways that set them apart from our everyday clothes. Check out how these artists, makers, designers, and researchers use actuators in wearables to communicate and relate to the wearer and the world around them.

Light Tissue

Light Tissue, by Sofia Guridi, uses cellulose as optical waveguides to transmit light. They are woven with cotton yarns to create light-emitting fabrics and touch sensors.



Charlie Wan

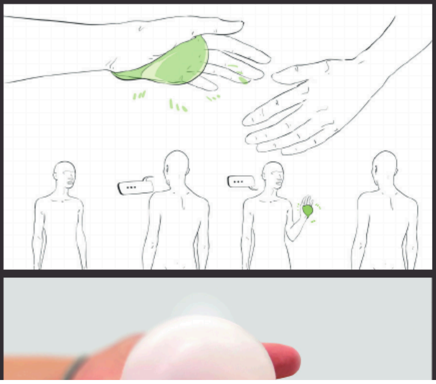
Short++

Short++, by Adi Marom, uses a heavy-duty gearhead motor to physically adjust the height of the wearer.

Elena Kulikova

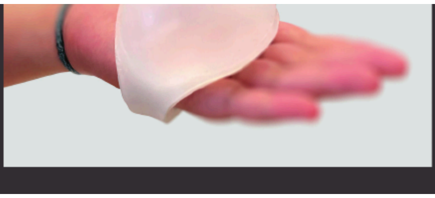
AWElectric: "That Gave Me Goosebumps—Did You Feel It Too?"

AWElectric, by Kristin Neidlinger, is meant to be worn by two people, and designed to make the wearers share a feeling of awe. Electrodermal activity sensors are used to detect heightened feeling from one wearer and then communicate to the other with soft speakers. The speakers were created with an embroidery machine with conductive thread in the bobbin, stitched onto dissolvable fabric. A microcontroller powers electromagnets with 40 Hz to get the wearers' hair to stand up on end.



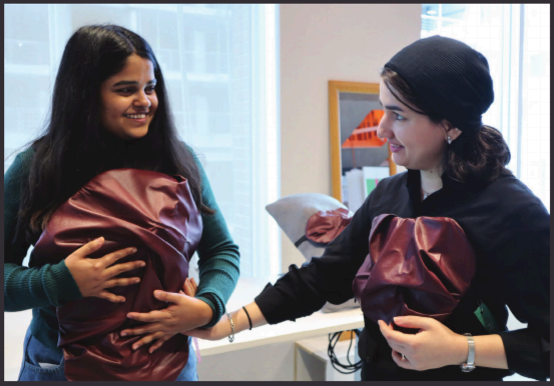
HiPalm: Augmenting Social Interaction

HiPalm, by Mona Safari, is a wearable soft actuator made from cast silicone, designed to augment nonverbal interpersonal communication with or without physical contact. The design addresses the challenge of conveying interpersonal communication as well as emotional states in situations where touch may be restricted—whether due to personal, social, or environmental reasons. HiPalm allows users to express and create interpersonal communication through dynamic, tactile feedback, controlled by proximity detection and heart rate. Worn on the palm, the actuator inflates and deflates based on the user's closeness and physiological state, offering a deeper, more meaningful interaction without the need for direct physical contact.



Bodies of Empathy

Bodies of Empathy, by Mufaro Mukoki, is a set of corsets that map open data to kinetic movements in order to enhance empathy toward the immigrant experience. Each corset represents the average yearly rejection rates for study permits by applicants' continent of origin provided by the IRCC (Immigration, Refugees, and Citizenship Canada) department, and the rejection rate is mapped to the physical size and intensity of movement of the corset. A servo mechanism and resulting sounds contribute to the movement, giving the impression of a heartbeat.



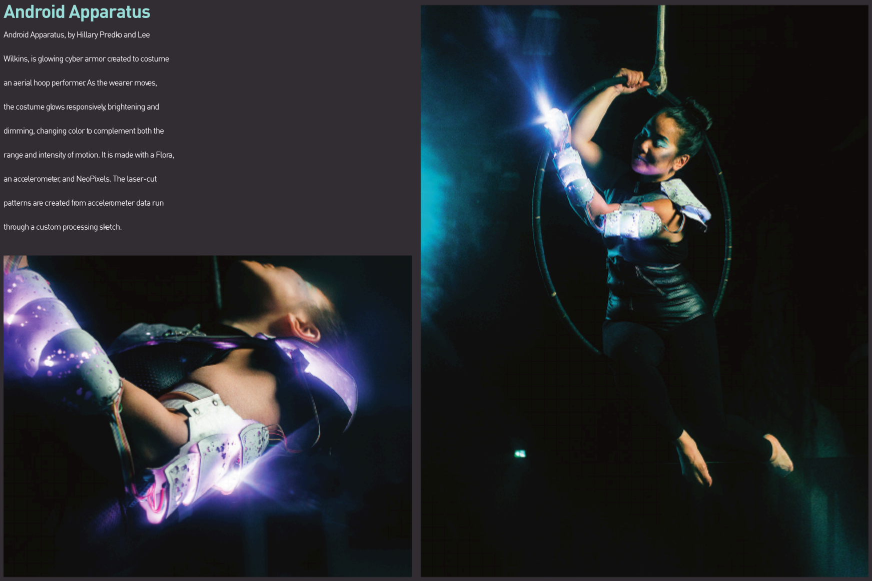
Enza Marcy

feeling distance

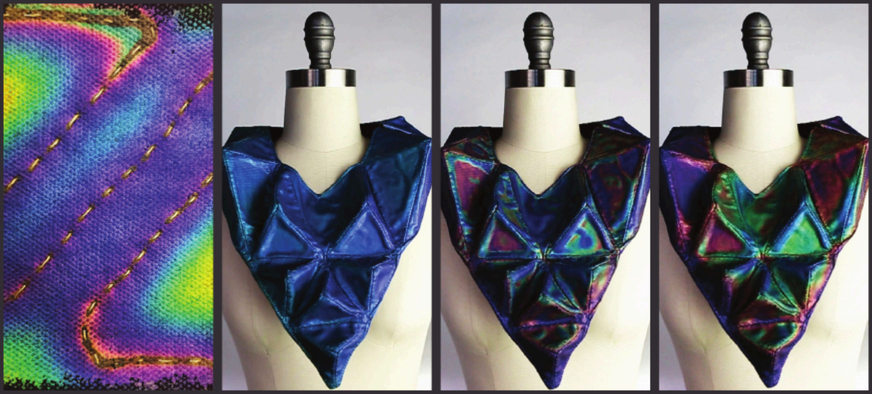
feeling distance, by Kate Sicchio, Tamara Denson, and Taylor Colimore, is an interactive choreographic event performed for an online audience that was able to interact with the dancer's costumes and provide virtual physical touch. When audience members clicked a button on a streaming website, they activated small fans controlled by x-OSC microcontrollers that inflated or deflated the costumes. Dancers: N'dea Harris and Marissa Schroeder.

Android Apparatus

Android Apparatus, by Hillary Predko and Lee Wilkins, is glowing cyber armor created to costume an aerial hoop performer. As the wearer moves, the costume glows responsively, brightening and dimming, changing color to complement both the range and intensity of motion. It is made with a Flora, an accelerometer, and NeoPixels. The laser-cut patterns are created from accelerometer data run through a custom processing sketch.



Maxwell Lander



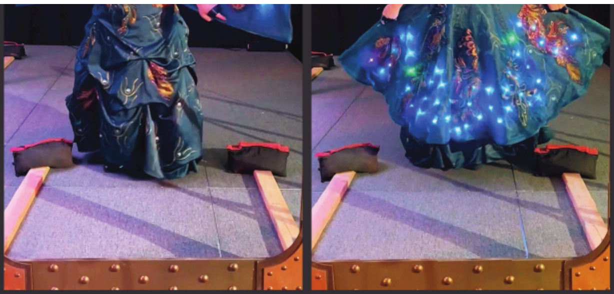
Intimate Space Sympathizer

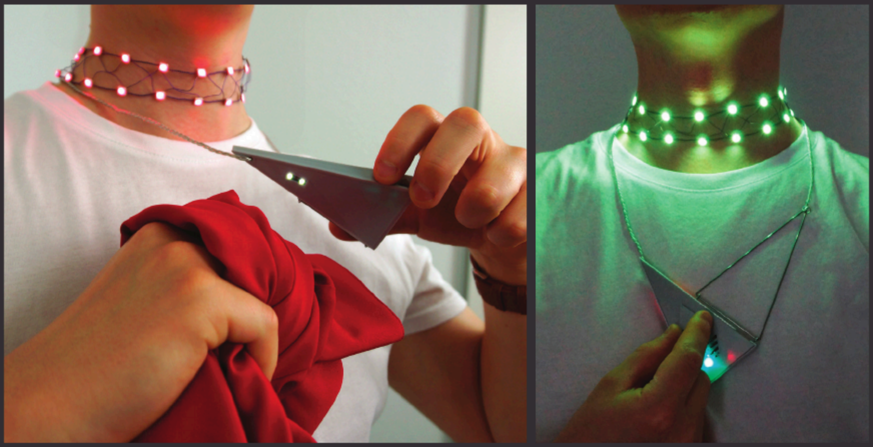
The Intimate Space Sympathizer, by Lee Wilkins and Hillary Predko, uses resistive thread and heating pads to produce heat and signal when the International Space Station passes overhead. The garment is coated in liquid crystal thermochromic pigment that cycles through twelve colors as the heat rises.

Peacock Garden Dress

Peacock Garden Dress, by Katherine Becvar, is embellished with hand-stitched appliqués and hand-painted peacock-feather shapes, paired with a fan-shaped overskirt tail embedded with color-changing NeoPixel LEDs. The CPX microcontroller board is used to sense the tail's orientation

in space so that the colors and patterns of the lights change dramatically when the tail is flipped up or worn down.



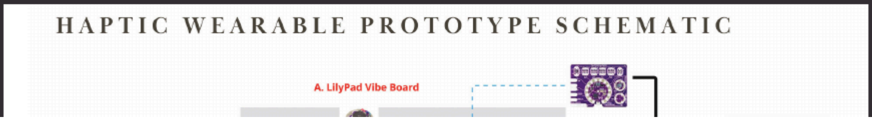


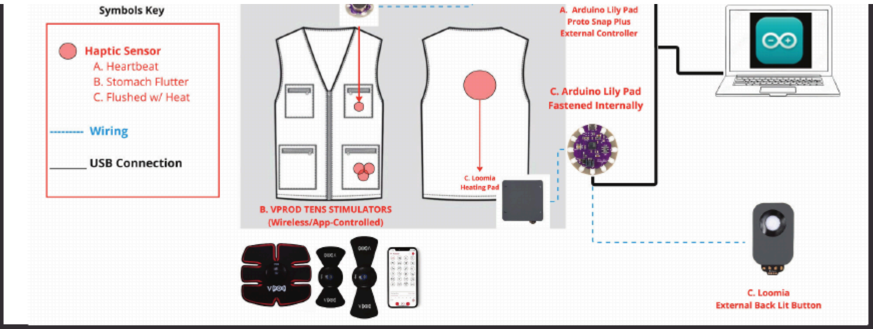
Exploring Interactive Jewelry in Social Context

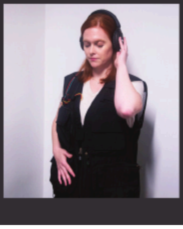
Anna Walczak's research considers topics such as self-expression and social acceptability. To study these areas, she constructed an interactive necklace. It consists of two parts: a decorative collar consisting of twenty RGB LEDs, acting as a decorative display, and a triangular pendant where the circuit is hidden.

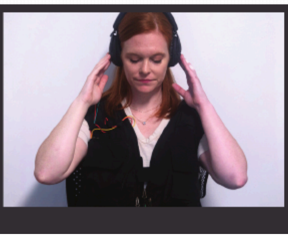
Moving Awareables

Moving Awareables, by Kristi Kuusk and Hsuan-Hsiu Hung, are a set of multisensorial objects that trigger bodily reactions and sensations through integrated tilt sensors and vibrotactile motors. The varied material characteristics (soft, smooth, warm, fluffy, etc.) invite embodied self-awareness.







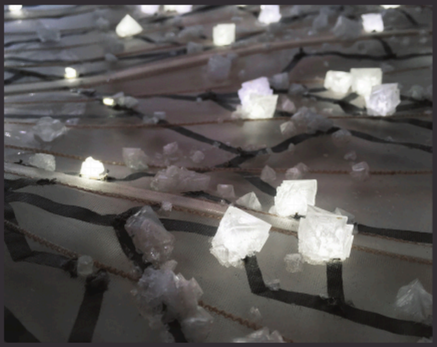


Haptic Memory

Haptic Memory, by Nicole Padilla, is an immersive experience that connects audio narratives with scored haptic reactions created with LilyPad and Loomia components to bring awareness to the impact of our memories' mind/body connection.

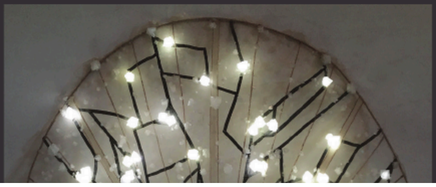
Urban Armor #8: The Public Speaker

Urban Armor #8: The Public Speaker, by Kathleen McDermott, a dress covered in speakers, has been used as a mobile karaoke garment and to trace the boundaries of a local radio transmission.

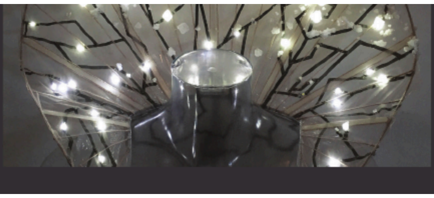


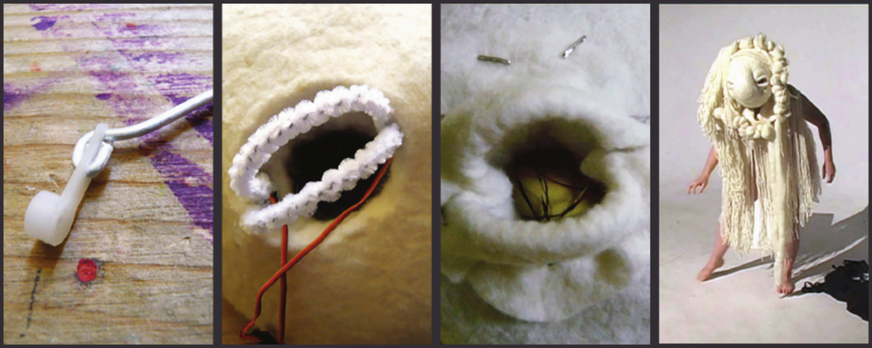
eTextile Crystallography

Melissa Coleman and Rachel Freire's eTextile Crystallography explores the idea of growing luxurious futuristic garments with salt crystals on a washable and reusable electronic base garment. This garment consists of e-textile taffeta traces, customized DotStar miniature LEDs, and resistors. In time the salt crystal garment may need cleaning or will have shed too



many crystals. At this point the garment can be sent back to the crystallographer to be regrown into an entirely new shape.



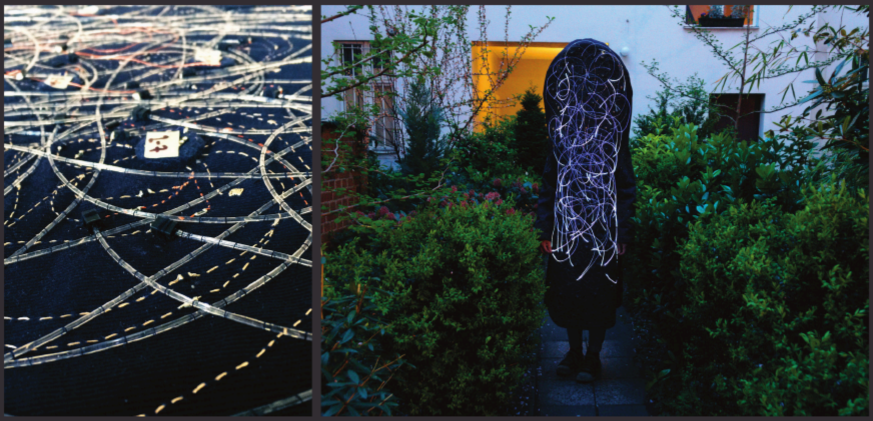


Soft Cyborg

Soft Cyborg, by Rachael Kess, is a mask created for use in performance. Servo motor horns are extended with pipe cleaners and covered with felt to animate the eyelids of a felt mask and make it blink.

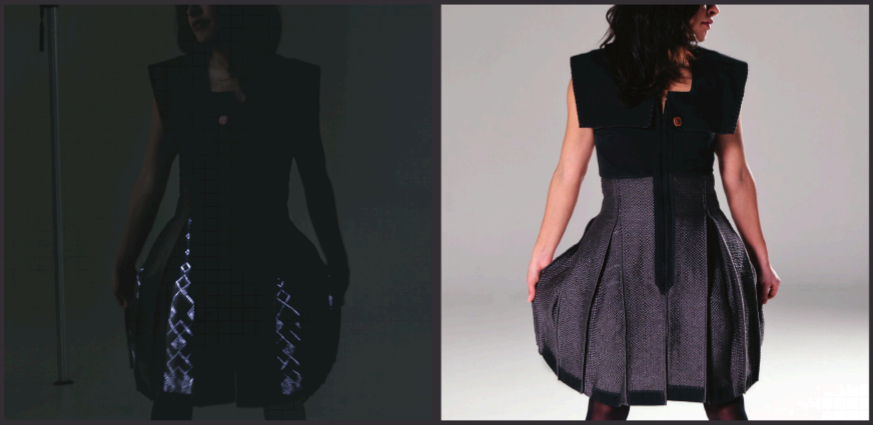
Kids' Distance-Sensitive Jacket

Kids' Distance-Sensitive Jacket, by Niloofar Sanandaji, uses the infrared capability of a CPX microcontroller to maintain a connection between a child and their guardian in crowded places. When the child exceeds a set distance from the person holding the alarm, the jacket will flash, making them easier to find. A pair of CPX microcontrollers were used to program the jacket so infrared messages can be sent between child and guardian.



Artificial Intelligence and Its False Lies

Artificial Intelligence and Its False Lies, by Mika Satomi, is an embroidered artificial neural network that uses an ATtiny85 microcontroller as neurons, fiber optics as synapse/transfer information, and LEDs and photoresistors as ways to activate and receive neural activity. The microcontroller circuits are embroidered with copper threads.



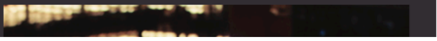
Peter Hoiss

Transformative Textiles

Oldouz Moslemian uses weaving to integrate fiber optics into the custom-designed material for a pleated dress. It is designed such that the fiber optics are present only in the interior of the pleats.

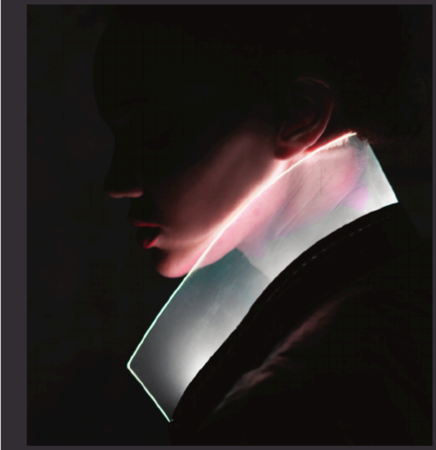
Ocean Dress

The Ocean Dress, by Alice Giordani of DataPaulette, Valérie Marsaudon, and Fang Yang, includes a flexible solar panel, NeoPixels, and a LilyPad Arduino that achieve a self-powered soft, luminous effect. The skirt is made of plastic bags on a fishing net that leads us on the theme of ocean conservation by recovering waste for more ethical fashion.



Mark Lamey

Photograph by Wearable Media; Modeled by Stephanie Berzon



Photograph by Christopher Zapata; Modeled by Esme

Future Species: Sound Interactive Capsule Collection

Future Species: Sound Interactive Capsule Collection, by Hellyn Teng and Yuchen Zhang of Wearable Media, is a sound-reactive fashion line that illuminates in response to sound around the wearer. The transparent PVC collar, easily detached from the garment, has embedded NeoPixel LED strips, a microcontroller, and a mic sensor.

9

Wearability

Primus, by Social Body Lab (Kate Hartman, Nick Puckett, Adam Tindale, Prayag Ichangimath, Sam Kingston), embedded in costumes designed by Joanna Yu

Primus is a set of networked wearable devices created by SBL for the production *R.U.R. A Torrent of Light*, by Tapestry Opera. NeoPixel grids and strips were incorporated into the costumes of the robots as well as other characters. The color and animation pattern of the LEDs provided clues to the audience about the operating system version each robot was running. Singers, dancers, and orchestra members wore variations of Primus, built to withstand wear and tear during repeat performances.

Wearable-electronics projects inhabit an intimate space. Wearables that work well work with you. They start to feel like a part of you. Wearables that don't work well can feel like an invasion of personal space. Moving beyond materials, circuits, and components, we must consider how technology that lives on the body can be used to create an experience.

Now that we know how to work with circuits, conductive materials, microcontrollers, sensors, and actuators, let's revisit the question that is central to a wearable-electronics practice:

What makes something wearable?

Adobe Stock-Lepusinensis, Dahlia Katz

Explore the question of what makes something wearable by wearing the project yourself and/or inviting others to wear what you've created. Wearing a project yourself can provide you with quick impressions as you prototype and give you the benefit of firsthand experience. In addition, having other people wear your wearable is a great way to get feedback from others and identify and resolve your own biases and assumptions that have made their way into your design. With both approaches, you will likely learn things about your project you did not expect.

This chapter introduces an approach to wearability that supports informed decision-making rather than telling you whether you're doing it right or wrong. It presents three possible wearables prototyping stages—versions 0, 1, and 2. If you're into iterative prototyping, you might want to make all three versions. But depending on your level of experience, how you make it, and the needs of your particular wearable project, you might skip ahead to version 2, go straight to version 1, or only do version 0.

Let's look at these possible approaches, some associated wearability criteria, and options to consider when designing electronics that live in clothing. Remember to mix, match, append, and deviate from these approaches to best suit your project's needs.

Version 0: Quick Test(s)

There are many challenges you don't anticipate in designing wearable electronics, which is why it is so important to wear your wearables early and often.

Especially as you are learning, wearing a project can help you challenge your assumptions. I have been making wearable-electronics projects for years, and each time I put on a physical sketch of a new project, I still discover new things I didn't expect.

While it is generally not recommended, you *can* wear a circuit that's a hot mess of alligator clips—just not as a long-term practice. Here are some tips for partially wearing a version 0 prototype.

- Wear only the components you need to wear. This might be a flex sensor on the elbow or LEDs around your collar.
- Use a low-stakes piece of clothing as the base. This can be something you already have around that is not precious.

- Use (very) temporary attachment methods. You can temporarily attach components to clothing with safety pins, Velcro straps, or no-residue or low-tack tapes.
- Prototype directly on the body where needed. This can be done with small, insulated components. For instance, when trying haptic effects, I often use medical tape to adhere small vibrating motors to the skin.
- Umbilical cords are OK. Long wires can create space between the worn components and your microcontroller board.
- Don't worry about integrating a power source into the wearable. Extra-long USB cords are great for powering your circuit, whether it remains plugged into your computer, a USB wall adapter, or a portable battery.
- Always be diligent about preventing short circuits. Quick and messy is fine so long as you won't hurt yourself or someone else—and preferably not your electronic components. Review your circuit to see which areas might be in danger of creating a short circuit with shuffling and movement. Use tape or another means to insulate and secure connections to prevent unnecessary damage.

EXPERIMENT: Auditioning Components for Wearing

Review the sensors and actuators in your circuit. Identify a component for which you have questions about how it might behave in a wearable context, such as the following:

- *Is it the right component for the job?*
- *Where should the component be placed?*
- *How might it feel to wear it on different body parts?*
- *How might it look to someone else?*

To start, make notes beforehand about what you expect to experience. Using the recommendations listed above, do a brief test of what it is like to wear that component. Afterward, make notes about what you experienced. From there, identify criteria for what might make this component more wearable in future

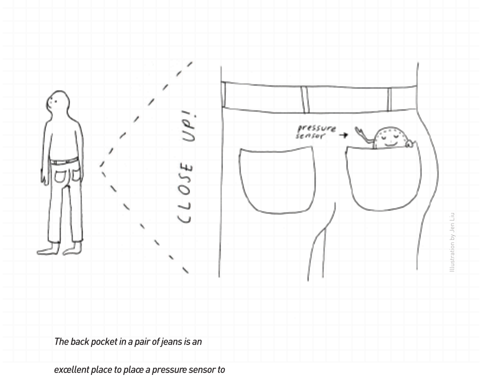


Illustration by Jen Liu

The back pocket in a pair of jeans is an excellent place to place a pressure sensor to test detecting whether the wearer is sitting.

iterations.

Version 1: Functional Prototype

But how do you go from a hot mess of alligator clips to something that can move with you through the world? There are many factors to consider when converting an electronic circuit to a wearable form. What works well with alligator clips, on a breadboard, or in a project box doesn't always translate well to the human body's dynamic, unpredictable, and rugged context.

Version 1 aims to convert the entire circuit into a wearable form, including choosing a base garment, placing components, and designing the circuit layout. To make it fully portable and untethered, you will also want to settle on a power source if you haven't done that already. For more information on battery options, see appendix B.

CHOOSING A BASE

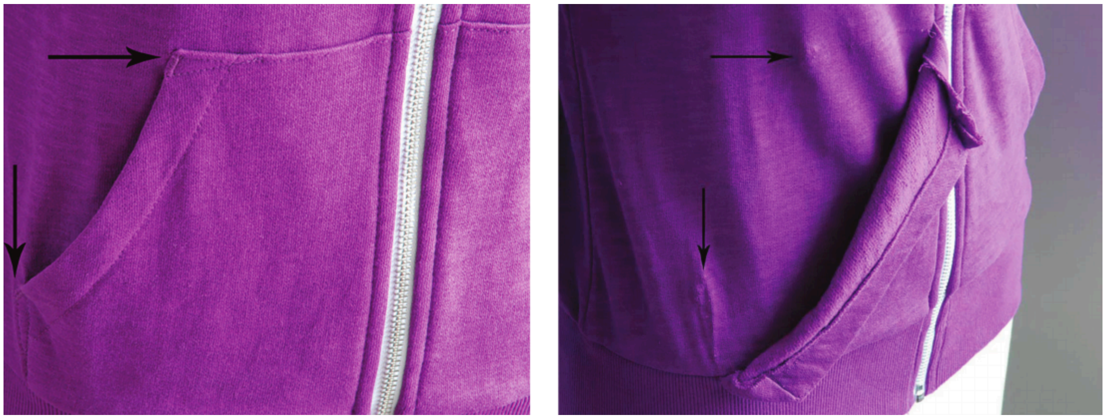
Wearable-electronics projects can take various forms, including jumpsuits, wristbands, gloves, hats, scarves, socks, jewelry, and even singing underpants.

Selecting the base for your electronics is a crucial part of the wearable-electronics design process. It provides the canvas on which you can start to plan your circuit. If you have a circuit you want to wear, you may have a sense of *where* on the body you want to wear it but may not have an understanding of *how*. How clothing is traditionally designed offers many opportunities for wearable technology Take a look at the clothing you are wearing right now. Assess the nooks, crannies, and pieces of real estate that might be available for electronic components.

Hacking, modifying, or augmenting existing clothing and wearable forms is an accessible way to get a prototype up and running. Thrift stores, discount clothing stores, or even giveaway piles can be great places to start when you need a base for the prototype you're working through.

When selecting a base, here are some recommendations:

- Favor stiff over stretchy material. Stretchiness can put a strain on circuit connections.
- Aim for garments with openings and closures, such as zippers or buttons. This allows you to open up and access different areas of the prebuilt garment more easily as you integrate your circuit.



The seams on a hoodie pocket can be opened to provide easy access to the pocket area.

- Look for garments with a lot of real estate, or surface area, for including your circuit.
- Pockets offer excellent support and protection for electronic components. You can either work with existing pockets in a garment or create your own.
- Linings can be used for insulation or for hiding components.

Based on many years of making and teaching wearable electronics, my advice for those who are creating their first wearable-electronics project is this:

Make a hoodie, not a glove.

So many people (myself included!) are drawn toward working with gloves for their first projects because our hands are integral to how we engage with the world around us. But gloves (especially cheap ones) are quite difficult to modify.

A zippered hoodie, however, is one type of garment that is fairly easy to enhance. You can add to it, creating additional pockets or a lining to better accommodate or incorporate circuitry Or you can use a seam ripper to open a seam and elegantly modify the existing design.

For version 1, choose a base that will allow you to make the circuit you are working with wearable without the need to miniaturize or replace all components. It's not yet time to try to shrink your circuit. Work with what you have!

PLACING COMPONENTS

Comfort can be greatly influenced by where components are placed in a wearable. When placing components, consider how the body moves. Are components located in an unobtrusive place, or in a place that exposes them to unnecessary bumping and smooshing? Are they likely to be protected, or subject to constant abuse?

Here are some base guidelines to consider when placing components:

- Consider how to group your components. What needs to live where?

- Keep the microcontroller and the battery close together.
- Items like microcontrollers and batteries can often be hidden in more spacious or discreet areas of the garment.
- Keep heavier items close to the core (i.e., the torso, shoulders, or thighs). If a heavier item needs to go out on a limb (no pun intended), ensure that it is secured directly to the body (like a wristwatch) rather than flapping around on a piece of cloth.
- Items like sensors or actuators may require very specific placement. Sensors need to be placed where they can do their sensing. Actuators are typically used to communicate something to the wearer or those nearby, so they need to be placed somewhere they can be seen, heard, or felt.

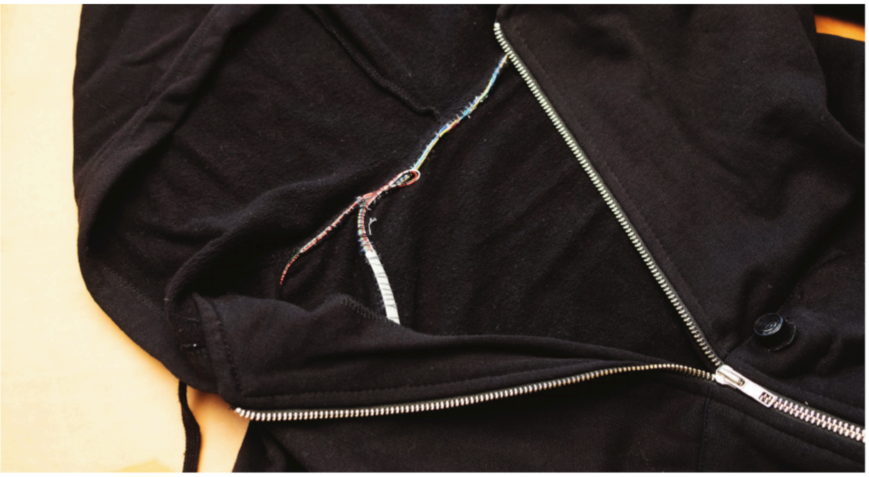
DESIGNING THE CIRCUIT LAYOUT

Designing circuits on a circuit board and designing circuits for wearables are two entirely different practices. With wearables, it is essential to start thinking about circuits in a three-dimensional way. For this reason, it is important to *plan* the layout of your circuit before integrating it into a garment.

Now that you know where you would like your components to live, you can begin to determine how the connections between these components will be made within the garment's structure.

There are a few ways to do this. Choose one medium, and then select a method:

- On paper,
  - trace the microcontroller on a piece of paper and draw out the circuit
  - sketch the full garment and how the components and connections will be organized on it.



Maxwell Lander

Stitching wires along the seam of a garment

- On a screen,
  - take a photo of the garment and digitally draw the components and circuit on top of it
  - digitally sketch the full garment and how the components and connections will be organized on it
  - create a 3D model of the garment with components and connections.

- On the garment,
  - draw the circuit on the garment itself (preferably on a dress form, mannequin, or fellow human) with tailor's chalk or a disappearing fabric marker
  - physically mock up the circuit with Post-it notes or paper cutouts of components with tape or straight pins.

When sketching in 2D, remember that you will definitely need multiple views and will likely need to think about layers of materials.

Additional aspects to consider:

- What types of conductive materials are you are using to create the circuit? Are they rigid, or flexible? Insulated, or uninsulated? Stretchy, or brittle?
- Keep in mind that with soft circuits, you need to consider the layout of your conductive traces to prevent short circuits.
- When possible, run connections along seams and edges. These areas very naturally accommodate some extra material.

EXPERIMENT: One-Hour Wearable

With functional wearables, it's difficult to get it totally right on the first try. The easiest way to learn how wearable something is to wear it.

Based on your notes from version 0, create a next version that is fully wearable.

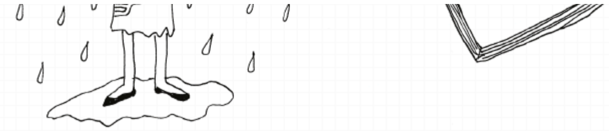
Once the prototype is complete, wear it for an hour. No, seriously:

Wear it for an hour!

Put it on, set an alarm, and then go about doing what you do. Jot down notes while wearing it about how your wearable feels, acts, and performs as you engage in the different contexts of your life. If needed, place yourself in the context where you intend it to be used.

For a wearable-electronics project to be considered truly wearable, it should be comfortable not only to *wear* but also to *use*. When you've completed your one-hour test, ask yourself the following questions:

- *Was the wearable able to be worn for the duration as intended?*
- *Did the electronics and code function as expected?*
- *What was the experience for you as the wearer? How did it make you feel? How did it affect your behavior or interactions with others?*



A one-hour wearable test

Illustration by Jen Liu

Version 2: Refined Prototype

Version 2 is an opportunity to iterate on version 1. While wearing version 1, you may have found that some of your design choices worked well, but it's also likely that there were things that didn't work or that you didn't expect.

Use these observations to inform the revisions you'd like to implement in a second, more refined prototype.

Here are some topics to consider as you create a refined wearable prototype.

ACCESS & REPARABILITY

Don't forget to leave a back door when incorporating a circuit into a garment! When rushing to complete a project, sometimes people will encase the circuit completely without leaving any means of access—for instance, gluing or sewing the material containing the circuit closed. Repeat access is needed to replace the battery, make adjustments, or make repairs. Zippers, buttons, or pockets can keep electronics secure but allow for access to circuitry when needed.

AESTHETICS

How wearables look matters. They are objects that occupy your most intimate spaces. They are an extension of your embodied experience and sense of self. How they look influences how you use them, when you wear them, how you relate to them, and what kind of emotional attachment you have to them.

Think about how you would like your wearable to look and feel. You may want it to be fuzzy and cozy, sleek and fashionable, or techy and sci-fi—the choice is yours.

When embedding electronics in wearables, you are often faced with whether to hide or reveal them. Hiding allows these technogarments to be seamlessly integrated into your existing habits and styles. But revealing has its own stylistic and functional advantages. It reminds both the wearer and the viewer that more is going on. And depending on how the integration is handled, it has the potential to add a "Wow!" factor.

AGENCY

Wearing a wearable-electronics project can feel empowering, stylish, silly, irritating, or deeply embarrassing, depending on how it's behaving. If something suddenly goes wrong, the wearer needs to retain some agency and control during the experience of wearing a garment without having to struggle to take it off.

A simple way to address this is to include an on/off switch in the circuit that the wearer can reach while wearing it. A way to level this up is to include a small control panel for the wearer so they can recalibrate the system, adjust the sensitivity of the sensors, or adjust the behavior of the actuators. These tweaks and controls can help deepen the wearer's sense of embodiment when wearing and using the device.

BESPOKE GARMENTS

If you are a seamstress, fashion designer, leather worker, industrial designer, or jewelry maker, you may be comfortable creating garments or wearable accessories from scratch. This provides an excellent opportunity to incorporate electronics into the design of the wearable itself.

For version 1, you were encouraged to work with an existing garment as a base. With version 2, you might want to move toward creating your bespoke garment or accessory. This can be a modification of an existing pattern, a new design, or an invention of your own. Create bespoke garments only if that speaks to the criteria you prioritize for version 2.

COLLABORATION

One of the most enticing aspects of working with wearable electronics is that it is such an interdisciplinary practice. While you may have a wide range of talents, there is still a strong likelihood that you don't have all the skills you need to produce your ideal wearable-electronics project.

For version 2 or future iterations, you might consider finding someone in your community to work with who has the skills you would like to employ to create your wearable. This could be someone good with textiles, fashion, electronics, coding, laser cutting, 3D printing, jewelry making—who knows! Interdisciplinary teams are the strongest and can significantly contribute to the long-term success of a project. Plus, you'll probably learn a lot along the way!

COMFORT & COMPONENTS

How large, heavy, and bulky are the electronic components in your circuit? How much surface area do they take up? How much do they protrude from the body? Do they conform to the body's natural shape? And are they able to move with or accommodate the movements of the body? These are good questions to consider for your prototype.

When it comes to wearables, relatively small, flat, and smooth components tend to be the easiest to integrate. Small and lightweight packages with curved shapes work best in the body context. This is why many wearables-oriented circuit boards are designed with rounded edges and corners. Curves are more comfortable to wear.

As you refine your prototype, it is sometimes worth reconsidering your choice of microcontrollers. Using a smaller microcontroller board with built-in sensors, actuators, or other capabilities can greatly reduce the overall footprint of a circuit, potentially improving the comfort of wearing it.

It's also important to know your wearer and their scenario for use. Are they a chihuahua, a child, or a CEO? What makes something comfortable to wear also depends on who is wearing it. Always have the needs of your particular wearer in mind when making design decisions about the comfort of a new wearable-electronics project.

CONNECTIONS BETWEEN COMPONENTS

The conductive material (thread, fabric, wire, etc.) used to create connections between components significantly impacts the overall design. Constructing your circuit with hard and soft materials is sometimes useful. Wearables usually introduce a variety of design constraints, and a hybrid approach can help you meet all your needs.

You can see this strategy reflected in various e-textile kits that have been developed. With the LilyPad Arduino, there isn't a sewn connection between the microcontroller and every resistor, capacitor, and LED. But stitching three traces between the LilyPad Arduino board and a LilyPad Light Sensor is a lot more reasonable and makes more sense if the LilyPad Arduino lives on the shoulder and the light sensor is on the cuff of the sleeve.

Revisit chapter 2 to refresh your memory about the advantages and disadvantages of various circuit-construction methods. Hard circuits are excellent for creating small, complex, and robust circuits. Soft materials are advantageous when you need simple, pliable, flexible, and comfortable circuits.

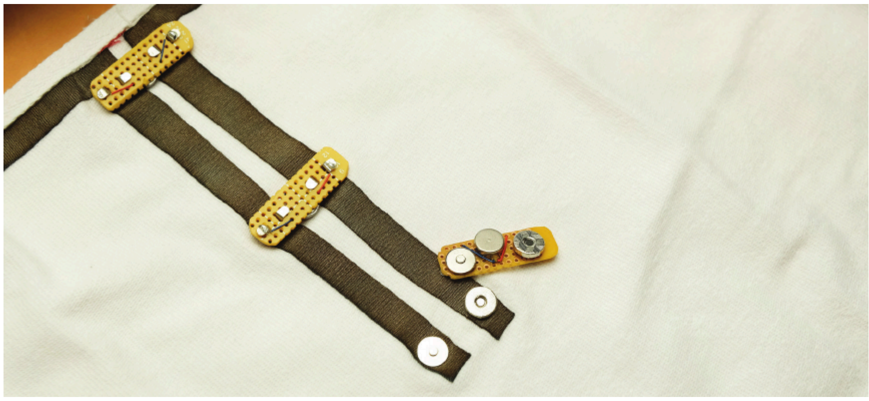
CONNECTORS & MODULARITY

Designing a circuit containing removable modules can help extend the life and increase the durability and practicality of wearable electronics. Being able to replace certain components easily means you can both customize and repair a circuit. The ability to remove sensitive components like batteries also makes it easier for the wearable to be washed.

To make something modular, you need to work with good connectors so that pieces of the circuit can be easily added and removed.

The worlds of electronics and sewing are full of potential connectors. Take some time to pore through the DigiKey catalog, or look up a connector you like on an existing board. Pay attention to standard connectors your favorite electronics manufacturers use. Take your multimeter to the fabric store and spend some time in the closures section looking for conductive connectors.
Depending on what you need connected and where, a variety of options will

likely be available.



Maxwell Lander

The combination of protoboard and magnetic snaps can be used to create components that are easy to remove when the garment needs to be washed.

DURATION OF WEARING

The expected duration of wearing can help to inform decision-making around a more refined prototype. Many aspects of duration can be considered:

- *How long is it meant to be worn for a single wearing?*
- *How many times is it expected to be worn?*
- *How long must it be able to be worn before needing charging, repair, or other upkeep?*

Fitness trackers and smartwatches are good examples to consider. Some must be charged daily, while others need attention only every few days or once a month.

Context of use greatly affects the duration of wearing. The needs of a gown worn once for the Met Gala differ from that of a costume used repeatedly on a concert tour. Meanwhile, a conceptual wearable-electronics project meant to be displayed in a gallery for a month needs to be durable enough to endure repeated use but doesn't require the same consideration around power supply, since it can be simply plugged into an electrical outlet.

FABRICATION METHODS

Digital fabrication methods have also become far more accessible in recent years due to the availability of lower-cost machines, the increase in hacker and maker spaces that enable access to this equipment, and web-based services that will process digital design files and mail back the results.

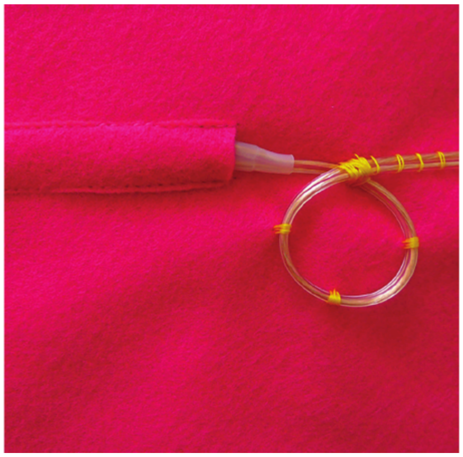
Craft cutters can cut fabric, paper, and other materials using a digital file. They are lower cost and excellent for home use because they do not have any special power or ventilation requirements.

A laser-cutting machine can both cut and etch, allowing you to efficiently cut textiles, leather, paper, wood, and other nontoxic materials. This can be useful when creating complex patterns or when working with multiples.

3D printing enables you to print custom objects, connectors, and cases on demand. There is now a wide range of consumer-grade 3D printers available, and they are popping up in libraries, classrooms, and hackerspaces. In addition, online printing services are available.

Maxwell Lander

Vibe-motor connections protected with heat-shrink tubing



This secured loop of wire provides strain relief for the solder joints of the flex sensor.

Digital fabric printing makes it easy to customize the materials you are working with. Simply create a digital design file and get it printed on a textile! This can be used to create custom patterns or for circuit layout as well.

These tools and processes open up all kinds of new possibilities for wearables. They make it easy to produce bespoke designs. They also enable reproducing designs shared by other people, such as enclosures for particular circuit boards. In addition, they support the creation of multiples, whether it be for wider distribution or for creating a shared experience where everyone is wearing your new, exciting wearable.

PROTECTION

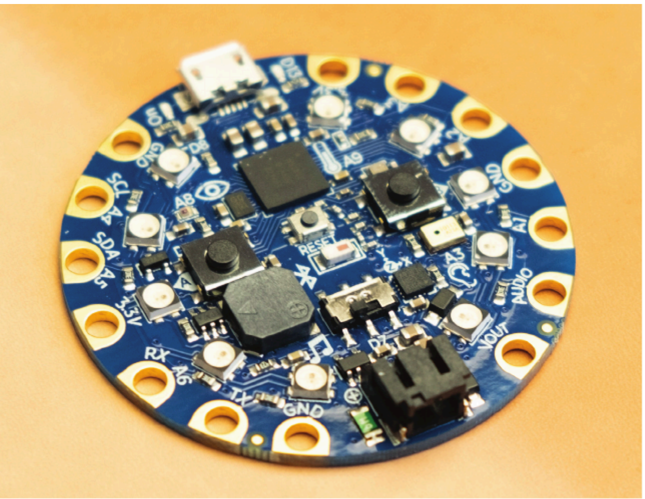
Circuits that live in the body space need to be pretty tough. Bodies bend, squish, bang, tug, and stretch, which is a lot to ask of a circuit. When you make a circuit meant to be worn, you need to ensure that it can withstand wearing, washing, and repairs.

Are there any parts of your circuit that require protection? A layer of foam, batting, or felt could cushion exposed header pins. A waterproof case or coating can shield your circuit from the elements. A lining can protect an exposed circuit from bare skin.

STRAIN RELIEF

A connection that is continually tugged is likely to eventually break. The solution is strain relief.

First, make sure there is ample material to accommodate the full expected range of motion. If you're running a wire along an arm, make sure



Circuit Playground Express Bluefruit is like the CPX we've been working with, except it has Bluetooth capabilities!

there's enough to cover the distance when the arm is flexed, not just when the arm is straight.

You can also take measures to relieve the connection of the strain. With wire, this can be accomplished by making a small loop close to the connection and sewing it in place. This way, any strain is put on the wire, not the solder joint.

WIRELESS COMMUNICATION

What if you want to use gestures, biometric data, or body language to control what happens on a screen? Or log body-generated data to a shared database? Or send a signal from one wearable to another?

So far, you've used various materials, tools, and components to create interactive systems that reside on the body. But what if you want to design wearable devices that communicate beyond the body, connecting wearables to each other or to other systems?

While communication between interactive systems can easily be accomplished with wires, this is not practical in the wearable context. Wires physically tether the wearable to whatever external system it is communicating with. Who wants to get tangled up in wires when they're going for a run, bustin' a move, or just walking around the house?

There are many options for wireless communication, including Wi-Fi, Bluetooth Low Energy, LoRA, and more. When selecting a wireless protocol to work with, here are some questions to consider:

- What are your nodes (points of communication) in your network? Do you want to send information from one wearable to another, to a nearby object, to a screen, or to a cloud service?
- How many nodes are there in your network?
- In which direction(s) do you need to send data? Is the communication unidirectional, or bidirectional?

- What distance does the wireless signal need to travel—a few feet, through walls, across an open field, or across the world?
- Does your network need to be totally independent, as in able to work anywhere? Or is it OK if it relies on the internet or uses a service?
- What is the desired frequency of communication?
- What is the desired accuracy of communication?
- How much power is required for wireless communication?

To get started, look for off-the-shelf examples to get you going, and perhaps even use an app that pairs easily with your board. For example, the colors of the LEDs on the Circuit Playground Express Bluefruit can be controlled using the Adafruit Bluefruit app.

Wireless communication is a wide and deep area of exploration—beyond the scope of this book, but very much worth pursuing and perhaps very relevant to your ideas.

EXPERIMENT: Wearing in the Wild

Each wearable is different, and each wearer is different. As you move through cycles of iteration, you will likely find that the criteria for what you consider to be wearable for a particular project or wearer might shift or change. You will also find that you hit constraints in terms of time or resources.

Depending on the scope of your project, you might also want to make a third iteration. Committing yourself to multiple versions will line you up for a much stronger, well-informed, and robust project in the long run.

Looking Ahead

The field of wearable electronics is on the move. In recent years, some wearable-electronics devices have become ubiquitous and embedded in our daily lives. And yet there are still significant challenges and opportunities for creativity and innovation.

We will hopefully witness significant material developments and the emergence of new connectors that will enable textiles and electronics to better intermingle. Our thinking will mature as to when, where, and how technology can and should (and shouldn't) be worn. We will continue to learn more about how wearables perform in the social context and the long-term effects on our everyday lives and our sense of human connectedness.

In all of this, I encourage you to use the act of making as a way to creatively and critically imagine our possible futures. The fields this book touches on are vast and rich. Many possible journeys extend beyond these last pages. Let the material you've learned act as a springboard to catapult you into exciting and unknown areas of exploration.

GALLERY 9:

WEARABILITY IN WEARABLES

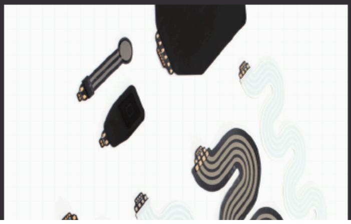
Making wearable electronics actually wearable is perhaps the biggest challenge of them all. This is especially true for individuals or small collaborative teams who need their skills to span multiple ways of making. Yet these artists, designers, and researchers continue to pioneer robust, attractive, and unique ways to integrate electronics into wearables. The projects that follow provide excellent examples. Take the time to research how these and other bespoke wearable-electronics projects are made.

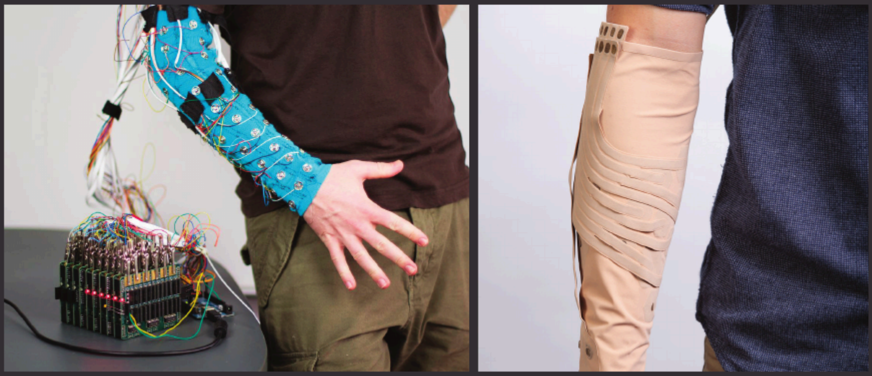
Solar Necklace

Solar Necklace, by Kathleen McDermott and Ash Tiwari, explores the application of mini solar panels. The final piece includes sixty panels sewn into neoprene that make contact with conductive nylon ribbon as a way to pass power through the garment to a charge controller. The artists tested many methods for attaching the panels to fabric, including conductive snaps, soldered wire, and copper tape.

Loomia Electronic Layer

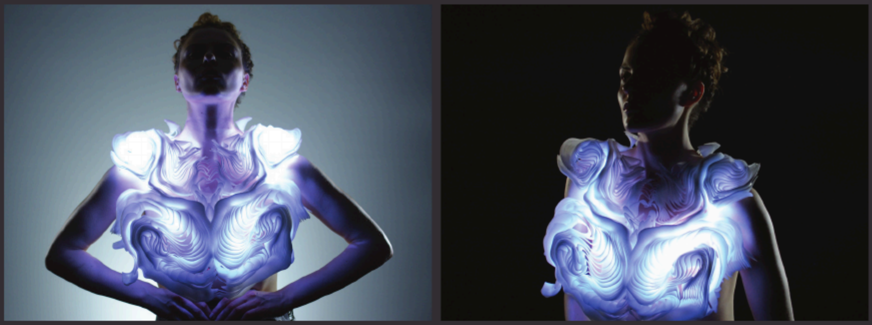
The Loomia Electronic Layer, created by Madison Maxey and Ezgi Ucar, is a soft, flexible circuit that is sewable and bondable. The product uses a nonprinted conductor and thermoplastic polyurethane (TPU) insulation to make components in a wide range of shapes and sizes.





SkillSleeves

SkillSleeves, by Rachel Freire, Jarrod Knibbe, Marion Koelle, and Paul Strohmeier, is a case study of the iterative design of electrode sleeves with a focus on wearability. The study generated new electrode-sleeve designs that begin to make it practicable to take electrical muscle stimulation outside of the lab and new fabrication processes that support rapid production and personalization of this type of sleeve.

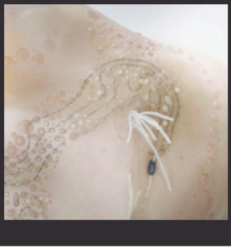


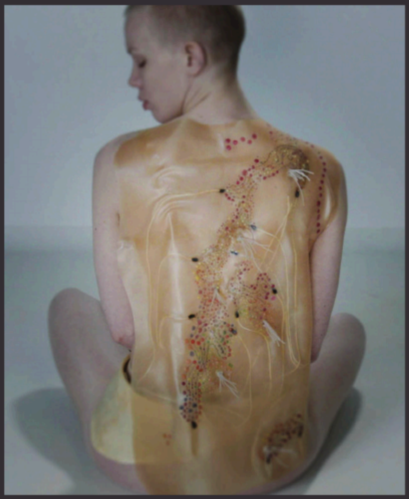
Bodyscape

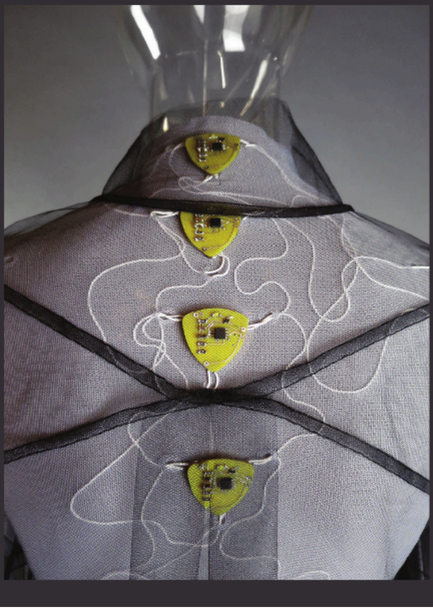
Bodyscape, by Behnaz Farahi, is an interactive 3D-printed fashion piece inspired by the behavior of the human body. As the wearer moves, LED lighting is controlled by a gyroscope that tracks shoulder movements. As a result, various lighting patterns emerge according to the forces being exerted by the body as it dances its way through space.

Coral Love Story

Coral Love Story, by Kasia Molga, is carefully made from naturally decomposing liquid latex. The wearable displays a pattern of the Great Barrier Reef painted using thermochromic ink with conductive threads embedded underneath the pattern. Eight motors are positioned across the piece, marking the location of eight scientific stations on the reef. Raspberry Pi and Feather microcontrollers connect the suit to the internet. They read data that comes from the scientific stations, allowing the wearable to react in real time to coral-bleaching alerts.

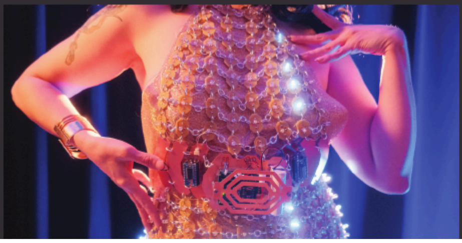


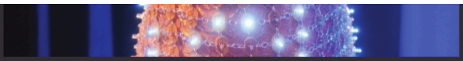


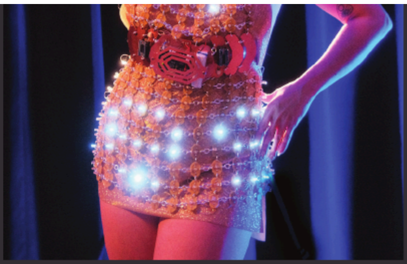


Embodisuit

The Embodisuit, by Rachel Freire and Sophia Brueckner, uses a network of modular vibration motors to create a garment on which information is experienced ambiently across the body. Intended to critique data-harvesting trends, the wearer can tailor which signals the Embodisuit receives from an Internet of Things platform to create a personal, embodied, and private way to experience user data.



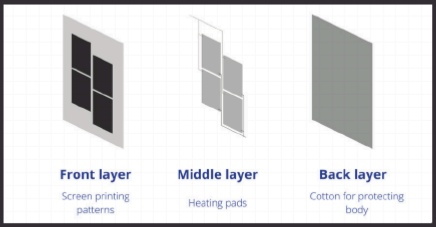


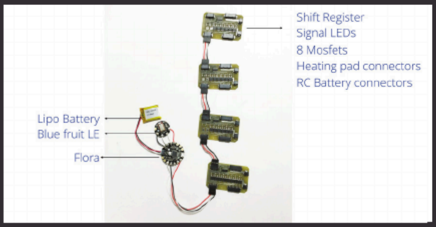


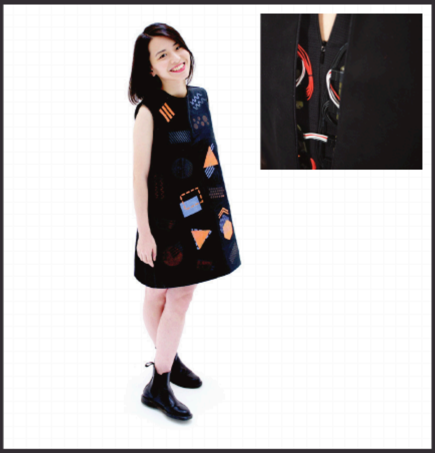
Photograph by D. Peterschmidt

Boundless

Boundless, by Kay Wasil, is a garment that transforms circuit boards and LEDs into wearable art, seeking to highlight the form factors of electronic components rather than embedding or hiding them behind other materials. Components include an Arduino Uno, LEDs, custom-milled circuit boards, nonconductive chain, and a 7 V lithium polymer (LiPo) battery (modeled by Logan Laveau).

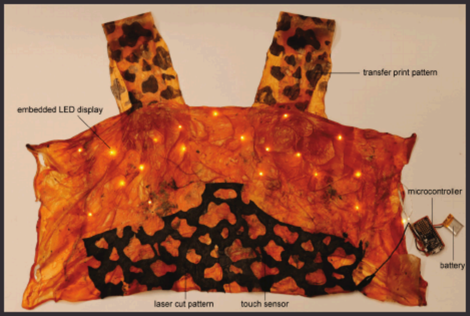






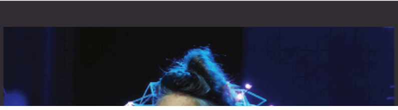
My Heart on My Dress

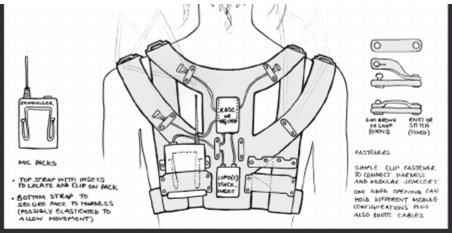
My Heart on My Dress, by Jingwen Zhu, is a bespoke connected garment that visualizes daily experiences and emotions through dynamic thermochromic pigments that create patterns that appear and disappear. The dress is screen-printed with thermochromatic ink and wired with soft circuits and heating pads. Its patterns and colors transform based on data analyzed from a personal-diary app.

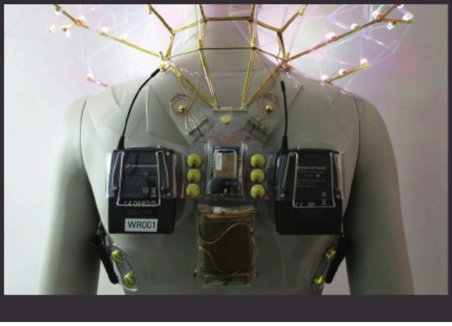


SCOBY Breastplate

The SCOBY Breastplate, by Fiona Bell, Derrek Chow, Hyelin Choi, and Mirela Alistar, is an interactive wearable grown and fabricated from SCOBY (symbiotic culture of bacteria and yeast) biofilm over the course of thirteen weeks. It has embedded sensors (a SCOBY-based capacitive touch sensor), LEDs, and a microcontroller that visualize touch interactions. Once no longer needed, the LEDs and the microcontroller are removed and the rest of the SCOBY Breastplate (including the capacitive touch sensor) biodegrades in about thirty days.




SENNHEISER
MIC PACKS
• TOP STRAP WITH INSETS TO LOCATE AND CLIP ON PACK
• BOTTOM STRAP TO SECURE PACK TO HARNESS (POSSIBLY ELASTICATED TO ALLOW MOVEMENT)
X-OSC OR MIMU
LIPO(S) STACK, INSERT
SAM BROWN OR SNAP (OPENS)
RIVET OR STITCH (FIXED)
FASTENERS
SIMPLE CLIP FASTENER TO CONNECT HARNESS AND MODULAR JEWELRY.
ONE SIDED OPENING CAN HOLD DIFFERENT MODULE CONFIGURATIONS PLUS ALSO ROUTE CABLES

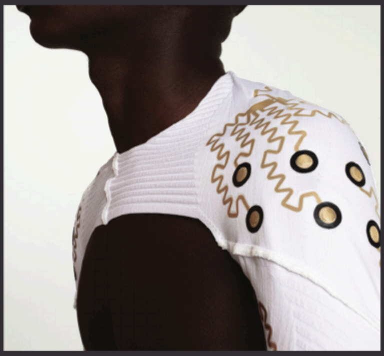


Illuminated Bolero 2.0

The Bolero 2.0, by Melissa Coleman, Rachel Freire, Emma Houston, Rebecca Odedra, Joachim Rotteveel, and Adam Stark, is a modular interactive jewelry piece inspired by networks and designed for musician Imogen Heap to visualize her music as an extension of her body. The garment lets Heap use her MiMU Gloves and her voice to influence the brightness, color, and pattern of the programmable RGB LEDs supported by an x-OSC wireless input-output board, wireless microphones, and LiPo battery packs.

Connextyle: Techstyle for Rehabilitation

Connextyle: Techstyle for Rehabilitation, by Jessica Smarsch with Christian Dils, Paula de Andrés, and Gaia Liesdek, is inspired by the need for data feedback in rehabilitation and an empathetic approach to wearable technologies. The interior of the sleeve uses Fraunhofer IZM's TexPCB technology, specifically designed for collecting and transmitting muscle activity. The snap design of the module, which transmits the data to an app via Bluetooth, allows easy removal for recharging the module and washing the sleeves. The two-part design of the shirt offers patients a stylish complement to the technical sleeves, promoting confidence during the rehabilitation process.



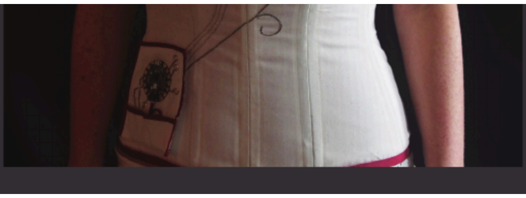
Social Prosthesis

Social Prosthesis, by Morgan Chen, explores the sociality of physical beauty—the change and movement that happens when we alter our appearances in contact with others. It is a moving appendage made up of two headpieces made from rigid and soft structures. The soft skin of the prosthesis curls and contracts when triggered by touch on the face, actuated by shape-memory alloy springs. [Social Prosthesis was made possible through research collaboration and the artist-residency program at the Hybrid Body Lab.]

Data Vows

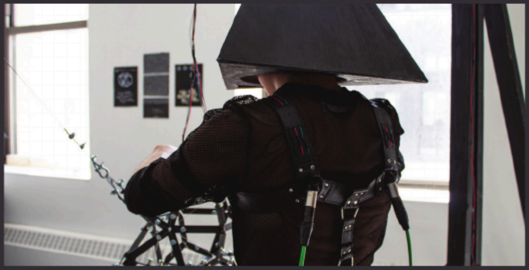
Data Vows, by Liza Stark, is a series of biometric sensing garments designed for near-future commitment ceremonies where these rituals are no longer sworn by words but by displaying and exchanging our most sacred physical information: our heartbeat. The garments include Adafruit's Flora, copper taffeta conductive fabric, Karl

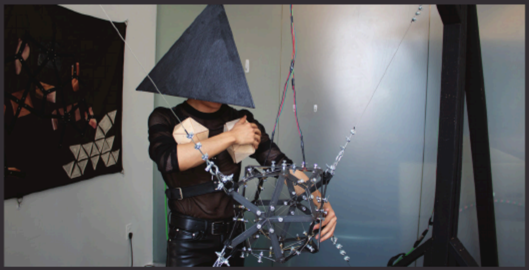
Grimm conductive thread, 1206 SMD LEDs, snaps, and a heart rate transmitter and a heart rate sensor from Polar, all applied to a custom corset.



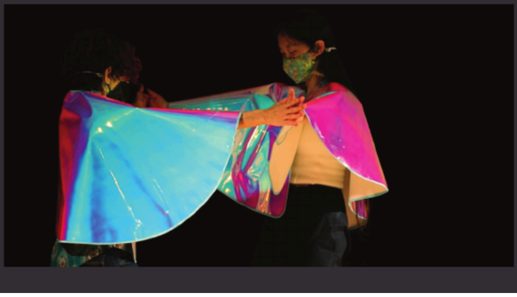
UNVEXED CONFORMAL BODIES

UNVEXED CONFORMAL BODIES, by Ranger Liu, is a devised musical system based on the geometry of the Johnson solids, a set of mathematical polyhedra. It includes metal instruments with pluckable strings and wearable wooden instruments that are played percussively. The instruments are attached to a body harness, which has integrated contact mics to pick up the percussive sounds and send them to XLR outputs at the back of the harness. The sound signals can then be routed to an external mixer or audio interface for amplification.

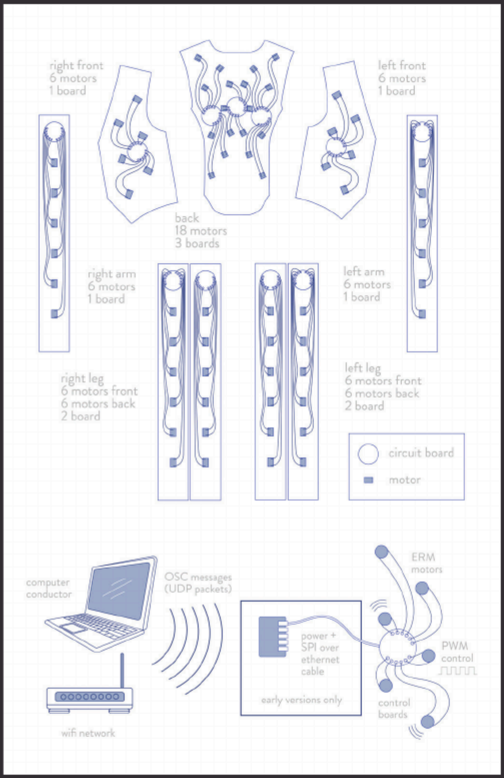


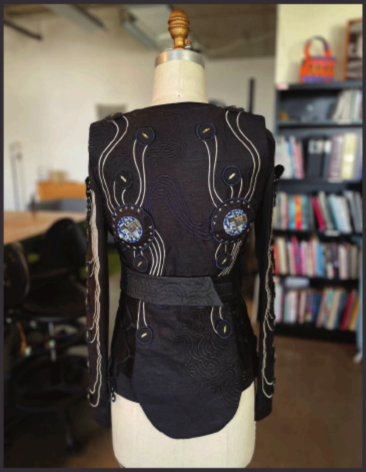


Two Subtle Bodies



Two Subtle Bodies, by Yeseul Song, is an interactive auditory experience that allows two strangers to use interactive capes to experience each other's peripersonal space. As the two bodies move together, they generate sounds that correspond to their movements. Each cape is embedded with a microcontroller (Arduino IoT Nano 33), and the changing signals are interpreted as sound through MIDI and a custom software instrument built on Ableton Live. Garment design collaboration and production by Daniel Ryan Johnston. Sound design collaboration from Jesse Simpson and Greg Halleran. Commissioned and funded by the Korean Cultural Center in Washington, DC, the Embassy of the Republic of Korea, and Sound Scene 2022.



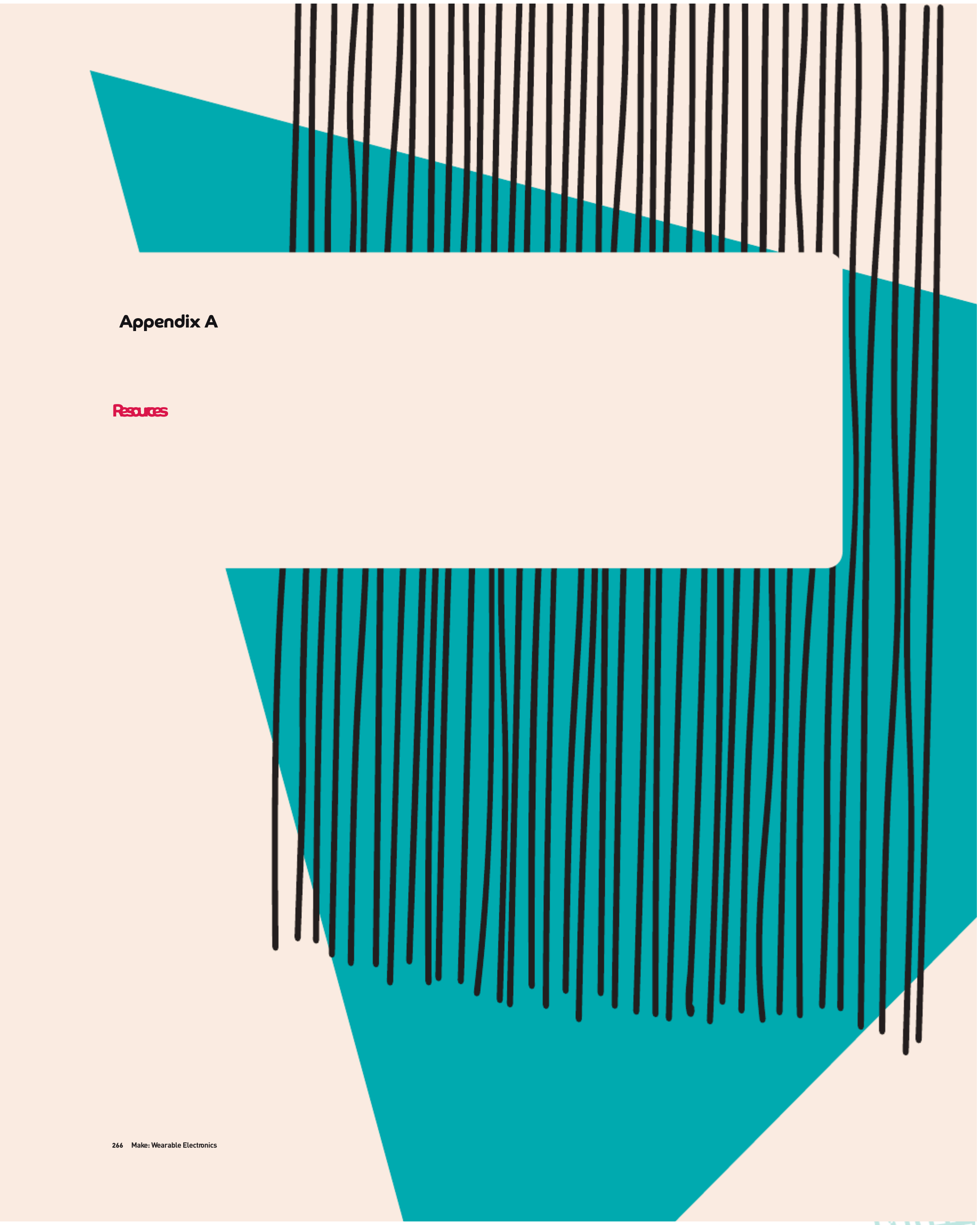


body:suit:score

The body:suit:score is a system of close-fit interactive garments that uses vibration to communicate score instructions and signals directly on a musician's body. The system consists of a software patch that sends and receives control messages over a wireless local-area network to control modules that drive an array of vibrating motors through electronic circuits embroidered directly into the garments. The modular bodysuits hold the motors close to the skin of the musicians so that they can experience the vibrating signals. These garments communicate between the composer/conductor and the performers, resulting in the final musical experience. The body:suit:score was developed by a team that included researchers–creators Alexandra Bachmayer, Audrey-Kristel Barbeau, Sandeep Bhagwati, Joanna Berzowska, Joseph Browne, Isabelle Cossette, Deborah Egloff, Marcello Giordano, Sylvain Payen, Jen Reimer, Julian Stein, John Sullivan, Marcelo M. Wanderly, and Travis West and performers–composers Sarah Albu, Pietro Amato, Erla Axelsdóttir, Adam Basanta, Noam Bierstone, Elinor Frey, Julian Klein, Csenge Kolozsvari, Elizabeth Millar, Felix Del Tredici, Jess Tsang, and Nien-Tzu Weng. These collaborators were from matralab and XS Labs at Concordia University in Montreal and the Input Devices and Music Interaction Library (IDMIL) and the Music Performance and Body Lab (MPBL) at McGill University, also in Montreal.

Appendix A

Resources



Here are some additional resources that will help you learn where to shop, what to read, and where to learn.

Suppliers

ELECTRONICS

- Adafruit Industries: adafruit.com
- DigiKey: digikey.com
- Maker Shed: makershed.com
- Mouser Electronics: mouser.com
- Newark Electronics: newark.com
- SparkFun Electronics: sparkfun.com
- Seeed Studio: seeedstudio.com

Don't forget to support your local businesses! The ones I use are listed below. Check out what there is wherever you live!

- Creatron Inc.: creatroninc.com
- RobotShop: robotshop.com

ELECTRONIC TEXTILES

- Bare Conductive: bareconductive.com
- Chibitronics Inc.: shop.chibitronics.com/collections/connectors-materials
- Kitronik: kitronik.co.uk/collections/e-textiles-conductive-thread/conductive-thread-&-material
- Lamé Lifesaver: conductive-thread.ca
- LessEMF: lessemf.com
- Shop VTT: shopvtechtextiles.com

TEXTILES

In general, I recommend visiting your local fabric shop or craft store for textile-related supplies.

- The examples in this book use both craft felt and industrial felt. Craft felt is easy to find, but my favorite sources for industrial felt is The Felt Store (thefeltstore.com).

Adobe Stock-Lepusinensis

Books

Whether you're looking for instruction or inspiration, these titles will enable you to expand on and deepen your knowledge of the topics covered in this book.

ELECTRONICS

- *Arduino Cookbook*, by Michael Margolis (O'Reilly)
- *Getting Started in Electronics*, by Forrest Mims III (Master Publishing)
- *Learn Electronics with Arduino: An Illustrated Beginner's Guide to Physical Computing*, by Jody Culkin and Eric Hagan (Make: Books)
- *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, by Dustyn Roberts (McGraw-Hill)
- *Make: Electronics*, by Charles Platt (O'Reilly)
- *Physical Computing: Sensing and Controlling the Physical World with Computers*, by Dan O'Sullivan and Tom Igoe (Thomson)
- *Practical Electronics for Inventors*, by Paul Scherz and Simon Monk (McGraw-Hill TAB)

WEARABLES & ELECTRONIC TEXTILES

- *Arduino Wearable Projects: Design, Code, and Build Exciting Wearable Projects Using Arduino Tools*, by Tony Olsson (Packt)

- *Arduino Wearables*, by Tony Olsson (Apress)
- *Crafting Wearables: Blending Technology with Fashion*, by Kate Sicchio, Madeline Gannon, and Sibel Deren Guler (Apress)
- *Fashion & Technology: A Guide to Materials and Applications*, by Aneta Genova and Katherine Moriwaki (Fairchild)
- *Fashion Geek: Clothing, Accessories, Tech*, by Diana Eng (North Light Books)
- *Fashioning Technology: A DIY Intro to Smart Crafting*, by Syuzi Pakhchyan (Make: Books)
- *Make: Getting Started with Adafruit Circuit Playground Express: The Multipurpose Learning and Development Board with Built-In LEDs, Sensors, and Accelerometer*, by Anne Barela (Make: Books)
- *Make: Getting Started with Adafruit Flora: Making Wearables with an Arduino-Compatible Electronics Platform*, by Becky Stern and Tyler Cooper (Make: Books)
- *Make It, Wear It: Wearable Electronics for Makers, Crafters, and Cosplayers*, by Sahrye Cohen and Hal Rodriguez (McGraw Hill TAB)
- *Open Softwear*, by Tony Olsson, David Gaetano, Jonas Odhner, and Samson Wiklund (Blushing Boy Publishing)

- *Playful Wearables: Understanding the Design Space of Wearables for Games and Related Experiences*, by Oğuz Buruk, Ella Dagan, Katherine Isbister, Elena Márquez Segura, and Theresa Jean Tanenbaum (MIT Press)
- *Practical Fashion Tech: Wearable Technologies for Costuming, Cosplay, and Everyday*, by Joan Horvath, Lyn Hoge, and Rich Cameron (Apress)
- *Sew Electric: A Collection of DIY Projects That Combine Fabric, Electronics and Programming*, by Leah Buechley and Kanjun Qiu (HLT Press)
- *Switch Craft: Battery-Powered Crafts to Make and Sew*, by Alison Lewis and Fang-Yu Lin (Potter Craft)
- *Textile Messages: Dispatches From the World of E-Textiles and Education*, by Leah Buechley, Kylie Peppler, Michael Eisenberg, and Yasmin Kafai (Peter Lang)
- *The Ultimate Guide to Informed Wearable Technology*, by Christine Farion (Packt)

Bookmarks

This list suggests resources for learning and ways to connect with others. Add these links to your bookmarks so you can keep up on the latest in wearable electronics!

Learning Resources:

- Adafruit Learning System: learn.adafruit.com
- How to Get What You Want: kobakant.at/DIY
- SparkFun Tutorials: learn.sparkfun.com/tutorials

Communities, Conferences, Events:

- ACM International Conference on Tangible, Embedded, and Embodied Interaction (TEI): tei.acm.org
- Designing Interactive Systems: dis.acm.org
- Electronic Textile Camp: electronictextile.camp
- E-Textile Swatch Exchange (Europe): etextile-summercamp.org/swatch-exchange
- Hackerspaces: wiki.hackerspaces.org
- Maker Faire: makerfaire.com

- Makerspaces: makerspace.com
- Open Source Hardware Association: oshwa.org
- UbiComp/International Symposium on Wearable Computers:

  ubicomp.org

Appendix B

Batteries

Bodies are dynamic, mobile vessels in which we travel through the world. It is because of our transient nature that wearables require a portable power source. This power source most commonly takes the form of batteries.

Here are a few things you need to know about batteries:

- Batteries convert chemical energy to electrical energy.
- There are two types of batteries: primary (single use) and secondary (rechargeable).
- Even within the same battery type, voltage and capacity differ slightly based on manufacturer, chemistry, type, and other factors.
- A stable connection to the power source is essential when creating reliable circuits. Working with the appropriate battery holder or connector greatly improves your chances of making a stable, solid connection.

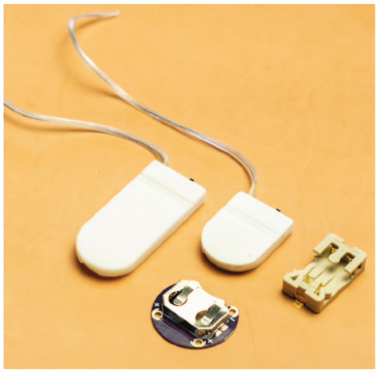
When possible, try to use rechargeable batteries in your projects. They typically have less of an environmental impact and are cheaper in the long run, as you won't have to keep buying new ones.

Coin Cell

A commonly available size of disc-shaped coin-cell batteries, which are often used in e-textile applications, is the CR2032 (20 mm). These small, thin batteries are excellent for low-current wearable applications. (Note, however, that they are not recommended for microcontroller circuits.)

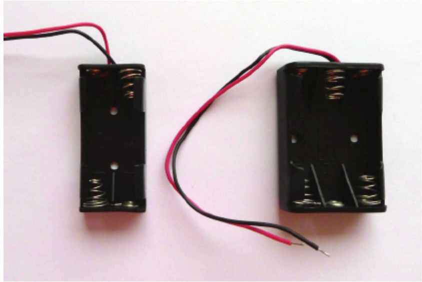
Coin-cell battery holders can either be stand-alone (like the ones we used in chapter 2) or can mounted on a circuit board (like the LilyPad Coin Cell Battery Holder). The latter are more expensive but easier to connect to using conductive thread. In addition, you can also find battery holders that with wire leads, like the ones used to create a conductivity tester.

Adobe Stock-Lepusinensis

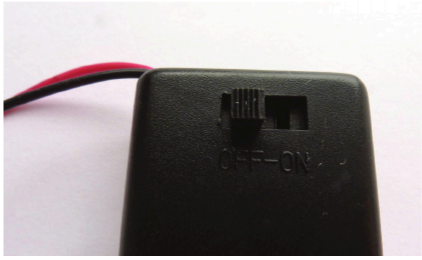


Maxwell Lander

CR2032 holders: case with leads (1x and 2x), LilyPad, and surface mount



2x AAA and 3x AAA battery holders



Round Cell

Cylindrical round-cell batteries usually provide 1.5 V, depending on their chemistry. This category includes AAA, AA, C, and D batteries, and each type is a different size. Usually, the larger the battery is, the greater its capacity. AAA and AA are best suited for wearables. (C and D batteries are too heavy and bulky for most wearable applications.)

AA and AAA battery holders can accommodate anywhere from one to eight or more batteries. Battery holders for two or more batteries usually connect batteries in series, meaning the voltage of the batteries is added together. For example, if you put three alkaline AAA batteries (1.5 V) in a 3x AAA battery holder, that battery pack will provide 4.5 V.

NOTE: Keep in mind that battery voltages differ depending on whether the battery

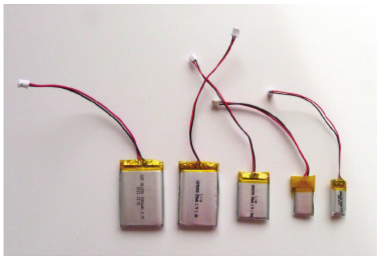
2x AAA battery holder with cover and switch

is primary (disposable) or secondary (rechargeable). For instance, a primary AAA battery provides 1.5 V, whereas a secondary AAA might provide 1.2 V. This doesn't make a huge difference when it's one or two batteries, but at four or more, it can become an issue.

These battery packs will sometimes feature a door or a full enclosure, which can help to protect the batteries, or a switch, which can act as an off/on switch for your project. This is especially useful if your microcontroller does not include an on/off switch.

Lithium-Ion Polymer

Lithium-ion polymer (LiPo) batteries have recently become popular for use in small electronic devices. The ones used in this book are flat, rechargeable, and relatively lightweight, offering 3.7 V and a capacity ranging from 150 to 2000-plus mAh. They are a bit more expensive but ultimately a great investment because they can be used again and again in a variety of projects.



3.7 V lithium-ion polymer rechargeable batteries with JST connectors; these batteries come in a variety of sizes and capacities.

LiPo batteries often feature wires with a JST connector, which is featured on many microcontroller boards. Some of these boards have onboard charging circuits and can recharge the batteries via USB. Otherwise, a stand-alone charging board can be used.

LiPo batteries can be dangerous when not used correctly Here are some important safety tips to consider:

- Always inspect the battery before use. If it is puffy or any part of the wire is exposed, do not use.
- Protect the battery from abrasion and puncturing.
- Charge LiPo batteries with a charger from the same source. Always check datasheets to confirm that the battery and the charger are compatible.
- Try to keep LiPos at room temperature.
- Do not leave charging LiPo batteries unattended.
- Do not charge LiPo batteries on fabric.
- Check regulations when shipping or traveling with LiPo batteries.

Choosing a Power Source

There are many factors to consider when choosing a battery or a battery pack for your wearable electronics projects:

Voltage

Does the voltage supplied by the battery or battery pack fall within the acceptable range for all components you are working with? This includes not only the microcontroller but also sensors and actuators. If not, have you planned to use a voltage regulator or step-up circuit to adjust the voltage accordingly?

Capacity

Manufacturers rate batteries according to ampere hours (Ah): 1 amp-hour equals 1 amp (1000 mA) for 1 hour, 100 mA for 10 hours, and 10 mA for 100 hours. In reality, these figures may differ depending on how much current is drawn.

Size and Weight

Size and weight of the power source are two of the more challenging aspects of building wearable electronics. In additional to their electrical characteristics, it is important to consider batteries from a design perspective: Where will these batteries live in your wearable and on your body? How will they feel? What will they weigh?

Availability

Where will this project be used, and who will be using it? Is it important to be able to replace the batteries easily?

Travel and Shipping Restrictions

Do current restrictions affect the way this type of battery can be shipped or packed for travel on a plane?

Maximum Current Draw

Does the maximum current draw of the battery accommodate the highest expected current draw of your project?

Intended Use

Different batteries are designed for different purposes. Factors to consider include shelf life and whether the use applications will be intermittent or continuous. Consult the battery's datasheet for more information.

BATTERY COMPARISON

TYPE	SIZE (IN MM)	WEIGHT (IN GRAMS)	VOLTAGE	CAPACITY (IN MAH)
Primary (nonrechargeable)				
CR2032	20 x 3.2	~3	3	~250
AAA	45 x 10.5	~11	1.5	~860–1200
AA	50.5 x 13.5	~23	1.5	~1800–2600
9 V	48.5 x 26.5 x 17.5	~35	9	~400–565
Secondary (rechargeable)				
Single-cell lithium-ion polymer (LiPo)	12 x 6 x 5 to 5.8 x 54 x 60	~2–36	3.7	~40–2000

Acknowledgments

The second edition of this book was shaped, wrangled, and supported by amazing collaborators and colleagues, to whom I am grateful for their skills and patience.

Editor Kevin Toyama offered insightful prompts and prods that led me to strengthen aspects of the book that might otherwise have been left untouched. His enthusiasm, humor, and intelligence were core to this project.

Copyeditor Mark Nichol provided thorough reviews and edits that elevated the text and furthered my learning about how to write.

Designer Nate Beale and Make: Creative Director Juliann Brown implemented a complete redesign of the book—above and beyond what might be expected of a second edition. I'm delighted that the vibrant layout and visual materials communicate my excitement about the subject matter.

Outreach coordinator Hillary Predko, photographer Max Lander, and technical reviewer Olivia Prior contributed in ways that extended far beyond their official roles on the second edition. They provided ongoing creative input and made themselves available for numerous consultations. Their playfulness, talent, and generosity were the "make" in the "make or break" of the second edition becoming a reality.

Research assistants Mufaro Mukoki and Mona Safari applied their expertise in fashion and industrial design to creatively contribute to the design and fabrication of prototypes in the second edition's "How To" sections, as evidenced by their hands appearing in the step-by-step photographs.

Over a hundred artists, designers, makers, and researchers worldwide submitted and/or contributed projects to be featured in the end-of-chapter galleries for the second edition. Although we couldn't include all the excellent projects submitted, the review team (myself, Hillary, and Olivia) was beyond inspired. Thank you ALL!

It should also be noted that resources from Adafruit Industries, Arduino, Fritzing, and SparkFun Electronics are referenced throughout the book. I commend these companies for their ongoing creation of well-crafted educational and learning materials.

The second edition builds upon the outstanding work of the first-edition team, including Brian Jepson, Meghan Blanchette, Shawn Wallace, Emma Dvorak, Rob Faludi, Erin Lewis, Pearl Chen, and Lynne Bruning.

The second edition was initiated by Patrick Di Justo, who first invited me to consider this project. His advice and encouragement shaped its early development. Cynthia B. Lauer also provided valuable guidance for the second edition proposal and work plan.

In addition to the official second edition team, Jason Bellenger, Carrie Schulz, and Maria Yablonina always made themselves available for clarifying conversations when I needed to get unstuck. My thinking about making, writing, and scoping was also influenced by brief but meaningful exchanges with Mitchell Akiyama, Kari Love, and Liza Stark.

As a professor, I'm privileged to learn from and with my brilliant colleagues and students at two academic homes: the Digital Futures programs at OCAD University in Toronto and ITP/IMA at New York University in Brooklyn. The contents of this book are heavily influenced by my teaching and research across these two communities.

I wrote this edition alongside many other significant professional endeavors. I benefited greatly from the support of stellar collaborators: the Bodies in Play team (particularly Emma Westecott and Cindy Poremba), ITP Camp's complement of full-time administrators and seasonal staff, and the OCADU School of Graduate Studies academic leadership and staff.

Social Body Lab, which emerged as a character in the second edition, is a shorthand for an inspiring network of people. Established in 2010, the lab's community includes colleagues, collaborators, and many generations of undergraduate, graduate, and post-graduate research assistants. Many of their names are peppered throughout the book alongside projects to which they contributed. A special shout-out goes to Nick Puckett and Adam Tindale, who joined Social Body Lab as co-directors in 2021. I'm excited to see how we'll co-create SBL in the years ahead.

Funding for this project was provided in part by OCADU's Faculty of Arts and Science Dean's Publication Grant and research assistantships supported by the School of Graduate Studies.

Lastly, there is life beyond work: It would be challenging to take on big projects on top of big jobs without my wonderful partner, parents, friends, and family of friends—thank you all for supporting me in all of my (overly?) ambitious endeavors!

About the Author

KATE HARTMAN is an Associate Professor at OCAD University in Toronto, where she is the founding director of Social Body Lab, a research and design team dedicated to exploring body-centric technologies in the social context. Hartman is also an adjunct instructor at the Interactive Telecommunications Program (ITP) and director of ITP Camp at New York University

As an artist and creative technologist interested in the nuances and awkward bits of social interactions, Hartman's research and practice sit at the intersection of design research, participatory art, and human-computer interaction. Through the use of wearable technologies, electronic textiles, and digital fabrication techniques, Hartman explores new possibilities for expressive, tactile, and embodied interactions.