Introduction

Title	Hack Your Pack
Overview	This Learning Activity is designed for flexibility: • Part I can stand alone (allowing for further independent exploration) • Parts I & II can be completed sequentially in one longer or two shorter block(s) of time Students will add LEDs and a power source to their backpack and explore basic circuitry during Part I, including the differences between circuits wired in parallel and ones wired in series. During Part II (see Part II), students will dive into the world of programmable wearables. They'll change the physical logic of the circuits they created using a soft switch, run a "sketch" that has been preloaded onto their hardware, then experiment with changing the sketch to make the lights blink and flash in patterns of their own design. The workflow for each part is based on a typical software development lifecycle (SDLC).

Prerequisites	Familiarity with basic sewing is helpful, though not required. Students should be old enough to handle sharp objects and small things (the LED sequins are tiny).
Learning Objectives	After successfully completing Part I of this activity, the learner should be able to: 1. List at least one kind of constraint commonly found in projects 2. Draft a plan for a simple circuit with 2 or more polarized components in parallel 3. Identify the proper orientation of polarized components relative to a power supply 4. Tell whether a switched circuit will work based on the position of the switch.
Process Skills Practiced	▪ Assessment ▪ Communication ▪ Critical Thinking ▪ Information Processing ▪ Management ▪ Teamwork

Background

Computing and electronics are being incorporated into every facet of our lives. From refrigerators that inventory their contents, to fitness trackers, to physical buttons that order from Amazon when pushed, the intersection of electrical engineering and computer science is growing ever closer.

Using an eTextile workshop to introduce students to the basic principles of physical computing offers many benefits:

- serving as an attractive entry point to computer science for girls and young women[1];
- allowing students to express their own creativity;
- creating a sense of empowerment by enabling students to modify their own environment;
- providing the basis for further exploration of wearable tech and small circuitry; and
- taking full advantage of the array of benefits that hands-on learning provides.

Reading

- https://sciencing.com/advantages-disadvantages-series-parallel-circuits-6306911.html - advantages and disadvantages of series and parallel circuits

- https://www.engineersedge.com/battery/battery_series_parallel_connections.htm - power in series and parallel circuits
- http://physics.bu.edu/py106/notes/Circuits.html - series and parallel circuits
- Soft Circuits: Improving Attitudes Toward Circuits through Crafternoons (http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1002&context=phys_capstoneproject) - research on using eTextiles to attract young women and girls to computing
- https://learn.sparkfun.com/tutorials/lilypad-basics-e-sewing – basic info about how to create a circuit and more tips for working with LED sequins and conductive thread

Directions

The approach we're taking follows one example of a Software Development Life Cycle (SDLC).

> **Instructor:** *Strongly suggest reading https://learn.sparkfun.com/tutorials/lilypad-basics-e-sewing – basic info about how to create a circuit and more tips for working with LED sequins and conductive thread*
>
> *Before going over the instructions for the project itself –* ***and before handing out thread*** *-- cover the *Intro to Circuits & eTextiles.**

Intro to Circuits & eTextiles

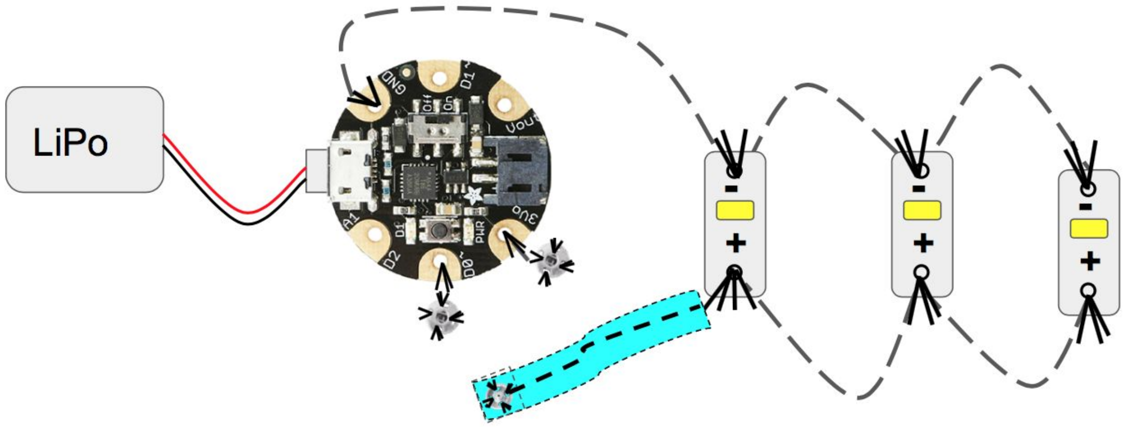
These things are so important that we're putting them here at the top rather than embedding them below.

- What makes a circuit?

> **Instructor:** *A good explanation can be found here: https://learn.sparkfun.com/tutorials/what-is-a-circuit You'll need to tailor the depth of your explanation to your students. Topics you may want to cover include:*
> - *Components of a Circuit*
> - *How electricity "moves" through a circuit*
> - *Conductivity*
> - *Short Circuits*
> - *Open Circuits*
> - *Voltage and Resistance*
> - *Circuit diagrams*
> - *Connecting Components in Parallel and Series*

- What is the circuit we'll be making?

> **Instructor:** *Draw the circuit on the whiteboard as you go through the next se ction, pointing out the pieces.*



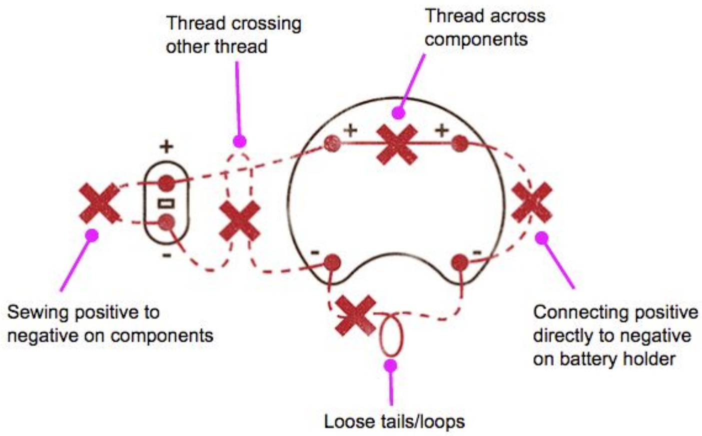
The circuit we'll be making uses conductive thread (which conducts electricity like wire) to connect a Gemma, which is a microcontroller board that can also serve as a "dumb" power pass-through, with three LED sequins, which are designed to be sewn onto fabric. A rechargable Lithium Polymer (LiPo) battery supplies the power.

We're going to create a "soft switch" (using felt, conductive thread and snaps) in order to change the function of the circuit. When the switch is closed in the 3v0 position, power passes straight from the battery through the Gemma to the LED sequins, then back to GND.   When the switch is closed in the D0 position, the Gemma acts as a microcontroller and must have a "sketch" (or instructions or program) loaded in order to work. The three LEDs (which act as if they are a single light source) will be controlled based on the instructions in the sketch.

The LEDs will be connected in parallel, which means that if one LED burns out there are other parallel paths that the electricity can flow through. So even if one of your LEDs is broken, the other two should still light up.

> **Instructor:** *If you have time, you can work through ways that the connection could be sewn that would allow each LED to be controlled individually. (Note that D0, D1, and D2 are similar in function.)*

- Most importantly: **Do Not Cross The Streams!** Basically, do not let the positive **(+)** trace (trace = thread = leg of circuit) ever cross the negative **(-)** trace. If you do, the circuit will short and things won't work! What does that mean? Don't do this:



- Use a relatively short length of thread – no more than an arm's length. Any more than this and everything will get all tangled up.
- Aim to run out of thread at a component, as it makes it easy to knot and start a new piece of thread. If that's not possible, you can just start a new length of thread by knotting it around the old one – remember each trace must be continuous.

- All the + must be connected. All the - must be connected. Even the LEDs have + and - which makes things easy for you!
- You can't cross a gap – like an a zippered opening – in between two LEDs or between the Gemma and the first LED.
- Any metal can short your circuit – this includes a metal button, a rivet, etc. So avoid those when you are sewing.
- Each time you attach a new component, you'll want to do at least three loops through the component and the fabric
- Don't pull your thread too hard! It is easy to snap.

Requirements gathering

1. We often think of functional requirements in terms of User Stories.
    - User Stories: describe what users want the system to do
        1. Example: The person carrying the backpack will want to be able to replace or recharge the power supply.

- **List at least one more User Story on the index cards provided.**

2. Another part of the Requirements Gathering Process involves listing the assumptions and constraints for the project.
   - Assumptions are things that are true for the purposes of this project.
     i. Example Assumption: We're going to assume that you can be trusted with a sewing needle.
   - Constraints are limitations that must be taken into account – often constraints reflect resource availability like time or money.
     i. Example Constraint: All components (except the power supply) should be washable.
   - **List at least one Assumption or Constraint on your card and indicate which it is.**

> **Instructor suggestion: if there's time, have students share and discuss their Requirements, Assumptions and Constraints. Once complete they can become the basis for the Design process.**

Design

1. Using a sheet of paper, plan the placement of your LEDs & Gemma. Draw each LED (marking + and -) as well as the Gemma.

   Overall Design considerations:
   - Visibility of the LEDs – where will they be seen or be useful?
   - Length of conductive thread (you have ~4' to work with for each "trace", or leg of the circuit)
   - The Gemma and battery can go on the inside of the pack, but the LEDs need to go on the outside in order to be seen

   LED placement
   - The closer together the LEDs are, the less sewing there is. We recommend placing all three LEDs in an area smaller than the diameter of a baseball, especially if you are not an experienced sewist.

> **Polarity**
>
> Electrical polarity (positive and negative) describes the direction of current flow in an circuit. Current flows from the positive pole to the negative pole. In a direct current (DC) circuit, like that found in batteries, one pole is always negative, while the other pole is always positive, and the electrons flow in one direction only.

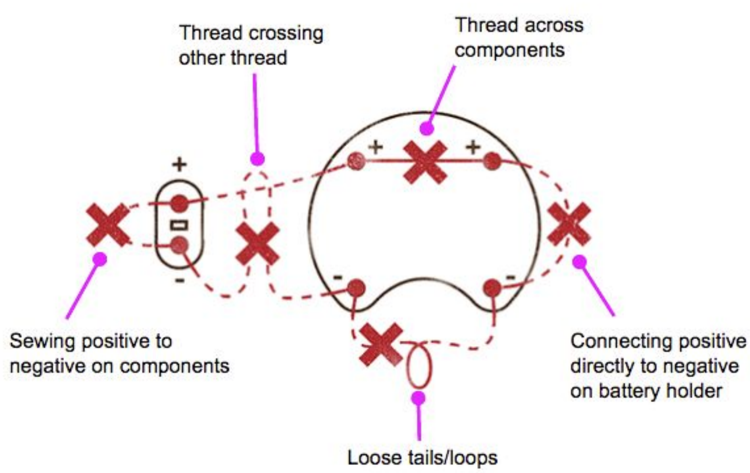
Electrical components can be either be non-polarized (or symmetric) or polarized, meaning it has a correct orientation.

A **non-polarized** (or symmetric) component can be connected into a circuit facing either direction and it'll still work.

A **polarized** component – a part *with* polarity – can only be connected to a circuit in one direction. A polarized component might have two, twenty, or even two-hundred pins (or connections), and each one has a unique function and/or position. If a polarized component were connected to a circuit incorrectly, at best it wouldn't work as intended. At worst, an incorrectly connected polarized component will smoke, spark, and be one very dead part.

- Because all the components we're using have polarity, it's important to ensure that each of the components is connected in the correct direction. In terms of our circuit, this means that:
  i. All the positive terminals on the LEDs must be connected to each other – as well as to one of the correct terminals on the Gemma..
  ii. All the negative terminals on the LEDs must be connected to each other – as well as to the GND terminal on the Gemma.
- Any metal can short your circuit – this includes a metal button, a rivet, etc. So avoid those when you are sewing.
- You can't cross a gap – like an a pocket opening – in between two LEDs or between the Gemma and the first LED).
- You can't cross the positive connections with the negative ones.

- **In a nutshell, don't do any of these things:**



Gemma & battery pack

- Figure out where you want the Gemma to be placed. You want to be able to easily reach the switch to turn your lights on and off.
- Figure out where the LiPo battery pack should go. You want to be able to reach the LiPo battery so that you can remove it to replace it.

These are washable – but you have to remove the battery from the battery pack first.

2. Once you've finished your design, ask one of the instructors to review it with you before proceeding
3. After an instructor has OK'd your plan, use the tailor's chalk or the fabric marker to transfer the design to your bag. Mark and check your traces.
4. Affix your components.

> **Instructor suggestion: Using a hot glue gun and a tiny dab of glue to affix each component in place on the pack minimizes the chance that things will go wrong between the making and executing of the plan.**

Implementation

> **Instructor suggestion: Create supply "kits" and set up the room in advance. However, do not pass out thread until after reviewing the circuit instructions above and the sewing instructions below .**

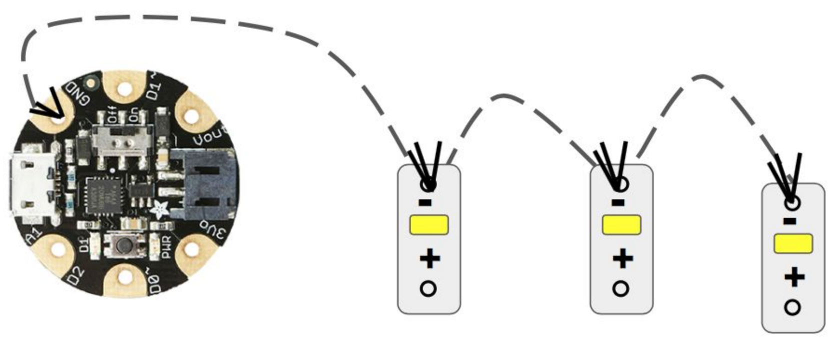
Prepping your needle & thread

1. Thread the needle. If you're having trouble, try moistening the eye of the needle (not the end of the thread) – surface tension of the water will help "pull" the thread through the needle. Also, try holding the thread still and moving the needle towards it.
2. Do **not** double the thread – leave a tail of 3-4"
3. Make a small knot in the end of the thread opposite the needle.

Connecting the LEDs

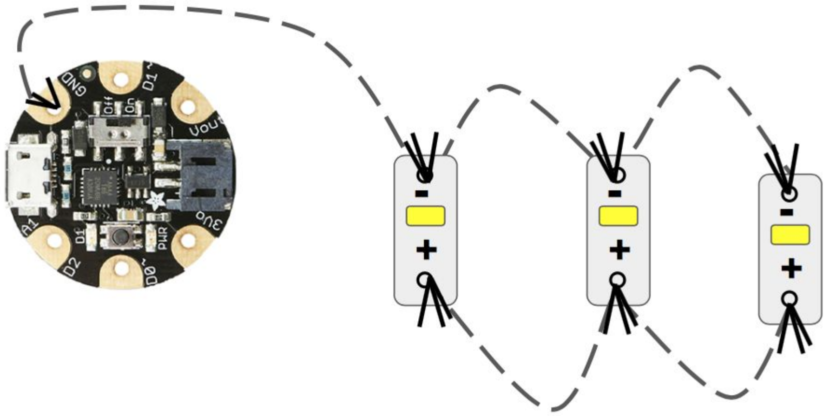
4. Start with a small stitch through the fabric of your clothing to anchor your thread, then pass your needle up through the GND terminal of the Gemma.
5. Go through the terminal and the fabric at least three times, like this:

6. Your first “trace” (or leg) of the circuit starts with the GND terminal on the Gemma then goes to the negative terminal on each of your three LEDs, as shown. Each time you connect an LED, make sure you use three “loops” through the terminal and the fabric beneath.
    - Use a running stitch (where you just go in and out of the fabric and you have approximately the same amount of stitching on the “top” and “bottom” of the fabric) to sew over to your first LED.
    - Make sure to keep your stitches short and even – we recommend they be no longer that ~.25 (about the width of a ballpoint pen).
    - Go around the negative terminal of the first LED and through the fabric (thereby attaching the LED to your fabric) at least three times
    - Keep sewing towards the next LED, which you’ll attach to the backpack using at least three loops around the terminal again.



7. Lather, rinse, repeat, until you get to the last LED. After attaching the last LED, knot your thread off, and trim your tail (it can cause a short if you leave a tail dangling!).

8. Tie a new knot in your thread and, starting at the LED that is furthest away from the Gemma, connect the positive terminals of the three LEDs in just the same way as you did the negative trace. When you reach the LED closest to the Gemma, knot the thread and trim the tail. You will have this when you’re done:



Making the Soft Switch

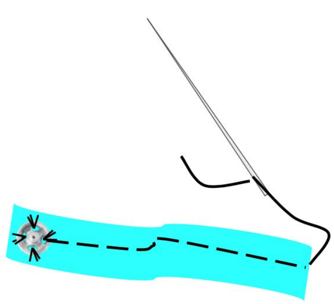
9. Now we're going to make the "soft switch" – a felt bridge that will allow you to change the way your circuit is connected to the Gemma. Cut a strip of felt that's roughly .5" longer than needed to reach from the positive terminal on the LED closest to your Gemma to the D0 terminal on the Gemma and roughly .5" wide.

10. At one end of the felt strip, use your conductive thread to attach the male part of a snap. As with other components, make sure to go through each hole in the snap at least 2 (and preferably 3) times.

Your felt strip should look something like the image to the left after you've attached the snap.

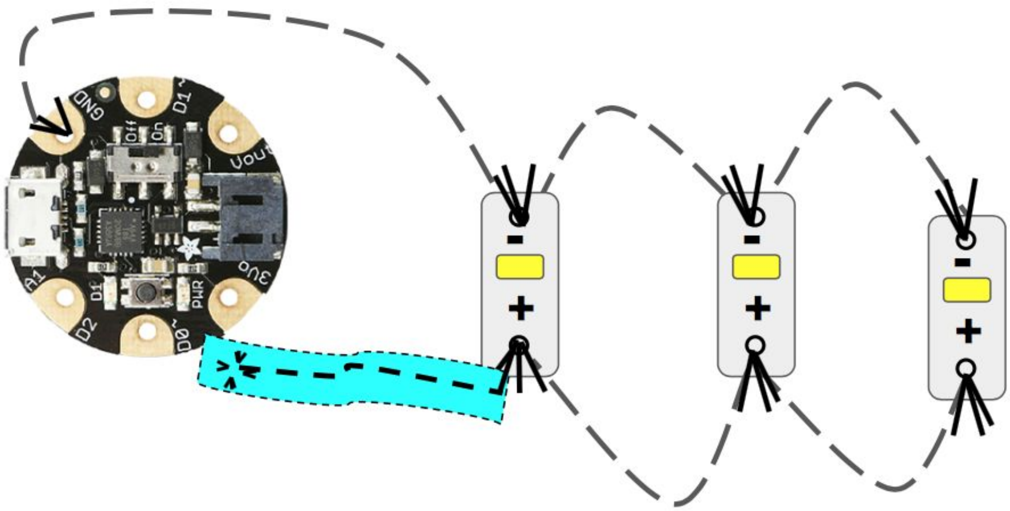
11. Once the snap is firmly attached, continue your line of stitching down to end of the felt strip opposite the snap using a straight running stitch. Remember to keep your stitches no longer than ~¼" or the width of a pencil.

You should end up with this:

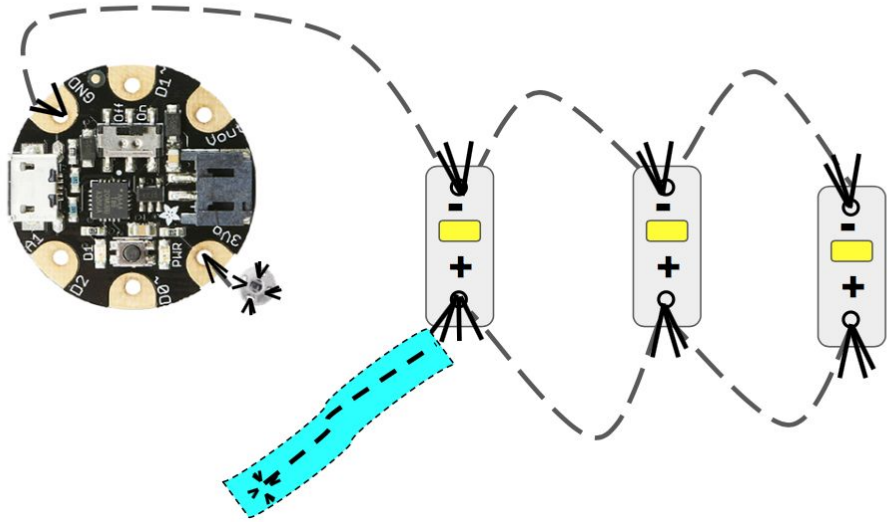


15. Leaving the needle attached, place the felt strip so the end where you've just exited the felt with with your needle is close to the positive terminal of the LED closest to the Gemma, like this:

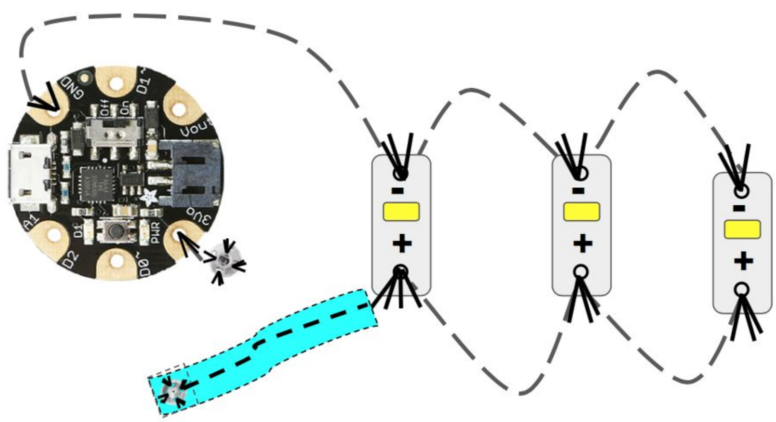
16. Attach the felt strip to the LED in the same way as you've attached the other components. Use at least 2 stitches, because you want to ensure there's a good connection between the LED and your "soft switch."



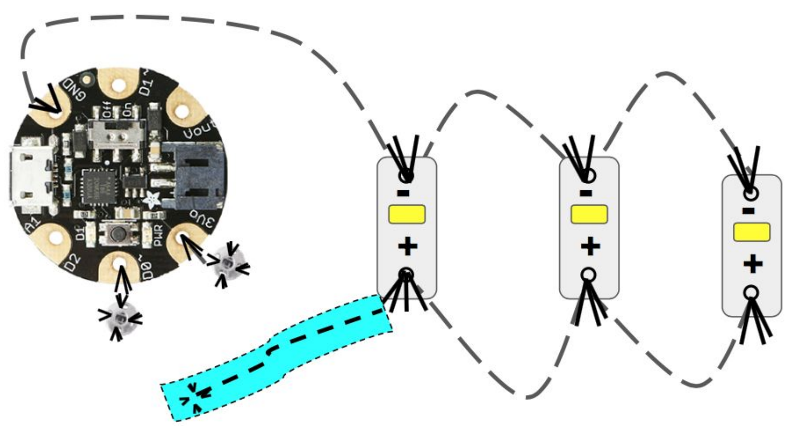
17. Attach the female part of a snap to the backpack very near the 3v0 terminal, which is used to transmit power only. Like you did for the snap on the felt, make sure you go through each hole at least 2 times (and preferably three). You'll have:



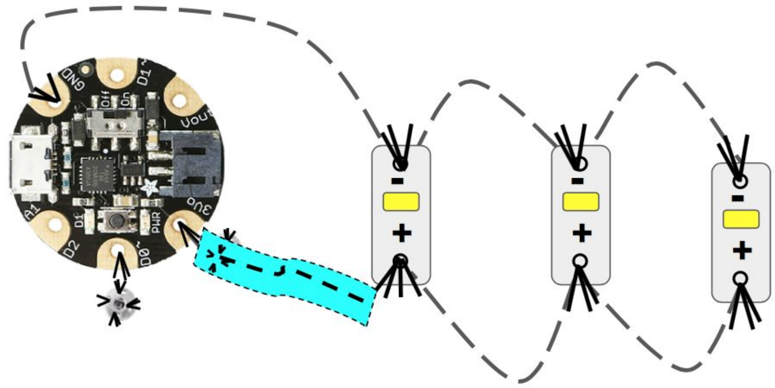
18. Connect the snap you just added to the 3v0 terminal of the Gemma (taking a few short stitches in between if necessary), and once again making sure you take at least three stitches around the connection point on the Gemma.:



19. Repeat this process with another female snap half, this time putting the snap near the D0 terminal of the Gemma, which is a terminal that can send data as well as power.

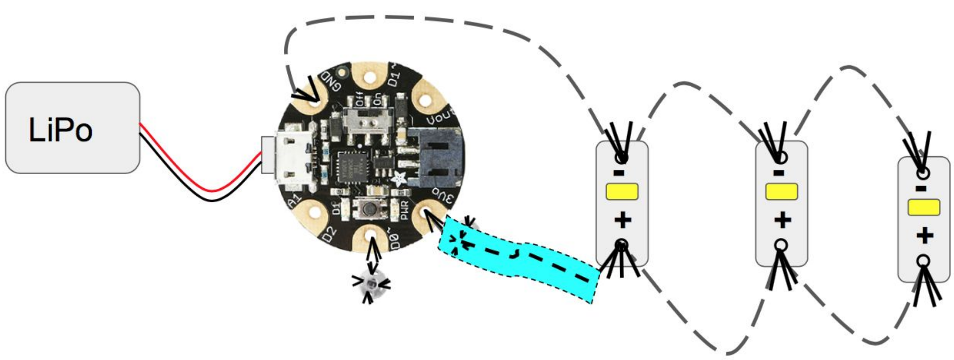


20. Snap the soft switch to the first snap you added, completing the circuit. This connection, the 3v0 terminal, does not require any “instructions” to be loaded on the Gemma (and in fact, it can’t send any – it is only a power pass-through).



Testing

- Time to power your circuit up!
- Make sure the power switch is turned to “off”
- Plug the LiPo battery in (it will only go in one way):



- Flip the power switch “on”
- Lights?
    - If so -- yah!!!
    - If not -- here are some things to troubleshoot
        - **Power - is your battery charged?** Test this by borrowing a “known good” battery.

"Known good" – a piece of equipment, like a bulb or a battery, that you know works. By replacing possibly faulty components with "known good" ones you can eliminate variables in your testing.

- **Do your LEDs work?** You can test this by using a pair of micro-clip leads to connect a CR2032 battery holder (with a battery in it) to each LED in sequence.
    1. Connect one end of one lead to the positive terminal on the battery holder and the other end of the same lead to the positive terminal of the LED.
    2. Then connect the negative terminal of the battery holder of the battery holder to the negative terminal on the LED, using opposite ends of the second LED.
    3. Turn the battery switch on.
    4. You should be able to light each LED individually this way (though there are occasionally duds)
- Do *some* of your LEDs work but each LED can be lit via a 2032 battery pack? Or do your lights work sporadically (they flicker unpredictably)? This usually indicates a short in between the lights that work and the lights that don't or a sporadic short. To look for a short, examine the stitching for:
    1. flapping thread tails
    2. loose stitches which may be flexing and coming into contact with other parts of the circuit
    3. Metal in the garment that might be touching the thread: zipper pulls, buttons, grommets, etc. are all likely candidates.
    4. Loose connections at components – you want three tight little loops attaching each component to the fabric.
- If no lights work at all, make sure that you haven't "crossed the streams" – that there's nowhere in the circuit – even at a component - where the positive leg of the circuit can touch the negative one.

Deployment (Final Clean up and launch)

Since you've tested the circuit and it is working there are just a few more things to do:

- Make sure the battery pack is secure.

- If necessary, you can make pocket to support the battery weight with a small square of felt attached to the bag with regular thread or a glue gun.
- Trim all loose threads. Seal any wonky knots with clear nail polish.
- If there is a lot of stitching showing on the outside of the bag, you can use puffy paint to “seal” the stitching lines, protecting them from abrasion and moisture.

Maintenance

This bag requires very little maintenance, though you will need to recharge the LiPo battery periodically!

Before washing, please remove the LiPo battery.

Deliverables

(short notes to be fleshed out)

- card/note with assumption or requirement
- Circuit diagram with plan
- Hacked Pack

Assessment

Criteria	Level 1 (fail)	Level 2 (pass)	Level 3 (good)	Level 4 (exceptional)

Comments

ACM Body of Knowledge **Area & Unit(s)**	)
ACM Topic(s)	Development Methods, Verification and Validation (Defect tracking)
Level of Difficulty	Medium (requires some understanding of the intended functionality of the software, ability to use bug tracking software, and critical thinking skills).
Estimated Completion Time	Not counting background reading, students should be able to find and "groom" a bug during an hour-long class.
Environment / Materials	**Each participant will need (see purchasing recommendations at end of doc):** • 1 LiPo Battery • 1 Gemma • 1 LiPo Charger • USB 2.0 A-Male to Micro B Cable: • 2 sets of metal snaps (they'll only use 1 male and two female) **And (from the Hack Your Hoodie supplies in Westford, so no need to purchase)** • 3 LED sequins • ~ 10' of conductive thread • Needle **For the entire class you'll need:** • 1 sheet of craft felt, cut into strips that are ~ 3.5" long and ~.5" wide (one 8.5"x11" sheet of craft felt should yield 48 strips) • Tools (scissors etc.) included in the Hack Your Hoodie Go Bag • 2 or 3 index cards for each student

Author(s)	Gina Likins
Source	14 Ways to Contribute to Open Source without Being a Programming Genius or a Rock Star
License	This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

Appendix:

Plan (pedagogy not yet inserted. Still working on instructions)

Students will add LEDs and a power source to their backpack and explore basic circuitry in the morning. In the afternoon, students will dive into the world of programmable wearables. They'll change the logic of their circuits, see the results of running the "sketch" that is preloaded onto their hardware, then experiment with changing the sketch to make the lights blink and flash in patterns of their own design.

Needed:
Gemmas
LiPo Battery and Charger
USB connector
Felt strips
Snaps (2 sets of m-f per student)

- + Regular Hack your Hoodie Supplies

Part I

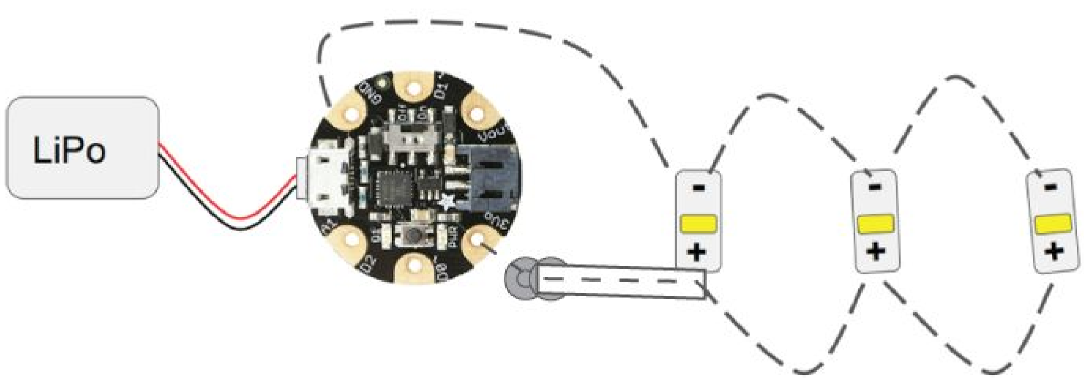
Have students start with the "normal" Hack your Hoodie plan (battery + 3 LED sequins in parallel) but substitute a Gemma (or, preferably the SparkFun equivalent of a Gemma) for the battery pack.

Use a strip of felt and a snap to make a connector piece. Attach one end of the connector to the nearest + on the LEDs. Attach the female end of the snap (facing down) to the other end of the connector.

Then run a very short line of stitches out from 3v0 to the male end of the snap (which faces up).

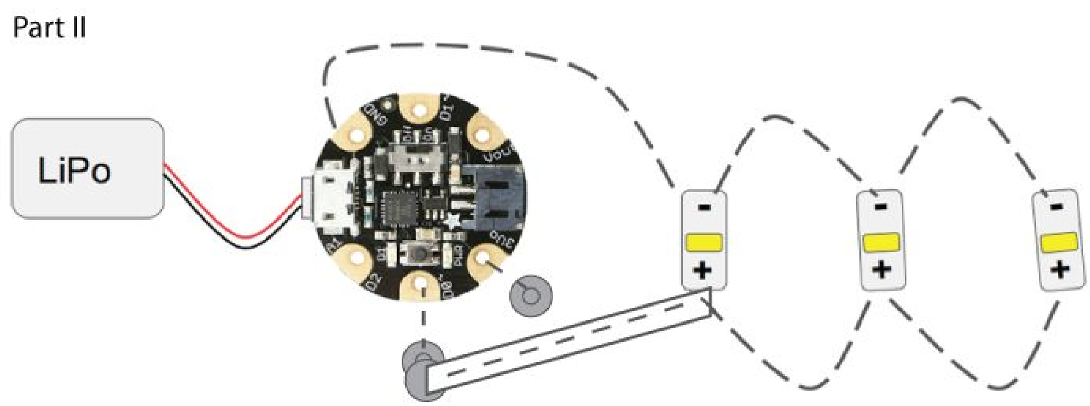
Now when the snap is connected, the LEDs get power and will light up. (The Gemma does not require a sketch to function... it can act as a “dumb” battery holder.)

Part I



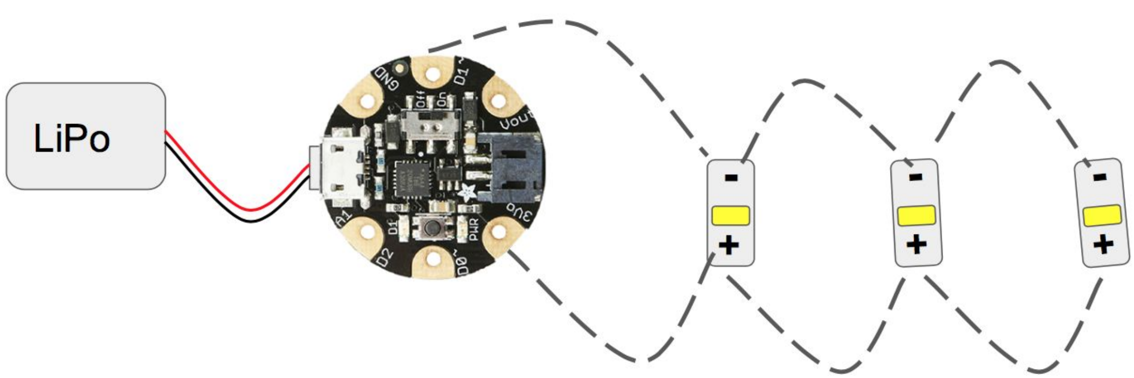
Part II
Then, if another short line of stitches is run from the D0 to a second male snap (facing up again), the connector can be switched from the 3v0 connection (“dumb”) to the D0 connection (which is a signalling connection, so the LEDs won’t light up until the Gemma has a script.

Now the students can switch between “just plain works” and “programmable”. Any extra time can be spent writing sketches for the Gemma.



Option 2: Small Amount of Rework

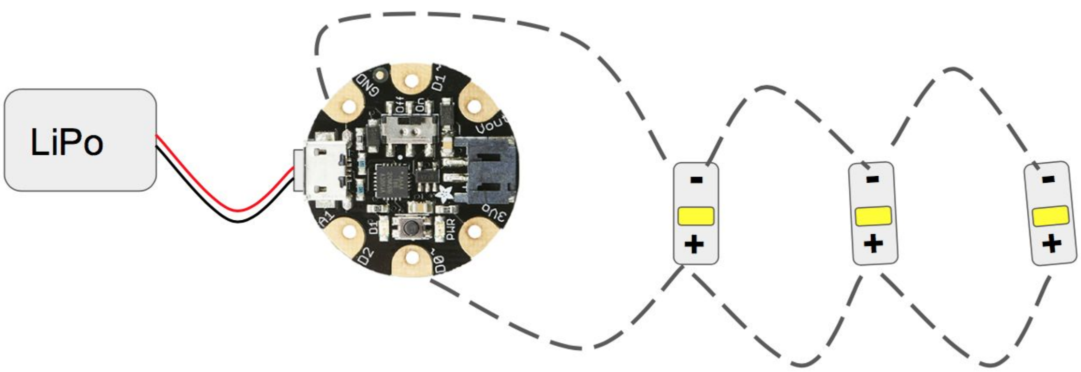
Part I
Have students start with the "normal" Hack your Hoodie plan (battery + 3 LED sequins in parallel) but substitute a Gemma (or, preferably the SparkFun equivalent of a Gemma) for the battery pack. The 3vo would go to the + leg and the GND gets connected to the - leg.



If this works the way I think it should (waiting for answers), at the end of this students will have lights that can be turned off and on with the switch on the Gemma. The battery is rechargeable (which is a nicer set up that I use for hack your hoodie).   Everything can stop here and things will work. Furthermore, we can send them home with instructions on how to move forward with Part II on their own, if they want. (And maybe I can get one of the folks I train up in Westford interested in learning this next part to help them?)

Part II

If it is still early in the day and they've completed Part I successfully, the "physical" aspect of Part II is cutting set of stitches that run from the 3vo to the + of the first sequin, then running a new line of stitches between D0 and the + of the first sequin, ending up with:



At this point the LEDs will be powered from a "smart" connection (D0) therefore, nothing will light up without a sketch loaded onto the Gemma.

If so, this would be perfect since then they could spend the rest of the time programming the LEDs to turn off and on and Fade and whatnot. (This part can take as long as they have -- messing around with the Arduino sketches to make the lights do things is fun)

BTW: In this scenario the 3 LEDs are acting as one unit from a control POV: if a "Fade Up" is sent, all 3 LEDs should fade up. This is the way it has to be unless we're willing to do a substantially more complicated reword between Parts I and II , which would have me worrying that we are sending students home with a non-functioning project which is Not OK.

On Wed, Jan 17, 2018 at 10:25 AM, Kristina Bowen <kbowen@redhat.com> wrote:
Hi Matt and Gina,

Thank you again for your support of the Boston STEM event in February. We met with the city of Boston and the school system last week and shared our thoughts regarding Red Hat activities (Lego + Hack your Pack). They were very excited.

Notes (ignore this)

Even though you can program Gemma using the Arduino IDE, it's not a fully 100% Arduino-compatible. There are some things you trade off for such a small and low cost microcontroller!

- Gemma does not have a Serial port connection for debugging so the serial port monitor will not be able to send/receive data
- Some computers' USB v3 ports don't recognize the Gemma's bootloader. Simply use a USB v2 port or a USB hub in between
-
-

Even though you can program GEMMA using the Arduino IDE, it's not a fully 100% Arduino-compatible. There are some things you trade off for such a small and low cost microcontroller! GEMMA does not have a Serial port connection for debugging so the serial port monitor will not be able to send/receive data Some computers' USB v3 ports don't recognize the GEMMAs bootloader. Simply use a USB v2 port or a USB hub in between We do not have full Windows 8 driver compatibility tested. At this time we only have it working with Mac, Linux or Windows 7/XP

Purchasing Recommendations

LiPo Batteries: 1 ea per student and instructor + 2 spares

- https://www.amazon.com/ADAFRUIT-INDUSTRIES-BATTERY-150MAH-pieces/dp/B011OA6C60/ref=sr_1_72?srs=5424951011&ie=UTF8&qid=1516642233&sr=8-72 This is a set of 10 LiPo batteries.
- https://www.amazon.com/oneself-Lithium-Polymer-rechargeable-keychain/dp/B00M6L9MCS - single LiPo battery

Gemmas: 1 ea per student and instructor + 2 spares

- https://www.amazon.com/Adafruit-Gemma-Miniature-Arduino-like-Electronic/dp/B00GEDHO1U/ref=sr_1_1?ie=UTF8&qid=1516642370&sr=8-1&keywords=gemma
- Or from Adafruit directly: https://www.adafruit.com/product/1222 (at QTY 10+ this drops to $8.96; however, I don't know how quickly they can ship)

LiPo chargers: 1 ea per student and instructor + 2 spares
https://www.amazon.com/Adafruit-Micro-Lipo-MicroUSB-Jack/dp/B00MJ0HOKS/ref=sr_1_98?srs=5424951011&ie=UTF8&qid=1516642294&sr=8-98

USB 2.0 A-Male to Micro B Cable: 1 ea per student and instructor + 2 spares

- (don't know which of these kinds of cable is the best value… these look reasonable to me)
- https://www.amazon.com/UGREEN-Adapter-Samsung-Controller-Android/dp/B00N9S9Z0G/ref=sr_1_2_sspa?ie=UTF8&qid=1516680126&sr=8-2-spons&keywords=micro+b+to+usb+cable&psc=1 (2-pack of short ones)
- https://www.amazon.com/AmazonBasics-Male-Micro-Cable-Black/dp/B072J1BSV6/ref=sr_1_17?ie=UTF8&qid=1516680126&sr=8-17&keywords=micro+b+to+usb+cable (3-pack, 3')

Snaps: each student needs 2 pair. Must be metal and sew-on (not prong) style. .25" or 7mm diameter or so would be easiest to work with

- https://www.amazon.com/Round-Snaps-Nickle-Corchetes-Silver/dp/B078VXPYXS/ref=sr_1_2?s=arts-crafts&ie=UTF8&qid=1516742879&sr=1-2&keywords=Snaps+metal+1%2F4%22 – not sure this is the best set, but to give you an idea

- https://www.amazon.com/Sets-High-Quality-Snaps-Silver/dp/B00S41RA32/ref=sr_1_2?s=arts-crafts&ie=UTF8&qid=1516743144&sr=1-2&keywords=Snaps+metal+7mm – these are good.