Twinkly Earwarmer Headband
Created by Kathy Ceceri

https://learn.adafruit.com/TwinklyEarwarmer

Last updated on 2021-11-15 07:54:46 PM EST
## Table of Contents

### Overview
- Parts List -- Non-Programmable Version 4
- Parts List -- MakeCode and CircuitPython Versions 5
- Suggested Tools List 6
- Suggested Materials List 7

### Sew the Earwarmer
- Measure and Cut Your Fabric 8
- Mark and Stitch the Back Seam 9

### Easiest Version - Embroider on a Wire Light
- Position the Wire Lights 12
- Attach the Wire to the Fabric 13
- Embroider Over the LEDs 15
- Closing the Top 16

### Intermediate Version - Sew on Gemma and NeoPixels
- Prep Your Circuit 18

### MakeCode Maker Animation 27

### Advanced Version - CircuitPython Chill Sensor
- Loading CircuitPython on Your Gemma M0 30
- Install the Mu Editor 31
- Temperature Sensor Code 31
- Test Your Code 32
Nobody likes hat hair, but that's no reason to let your ears freeze! Earwarmers are the perfect compromise -- cute and warm, but plenty of room for your hair to breathe free.

And by doubling the fabric to create a tube, you've got a handy place to hide a small battery holder -- enough to power a string of festive LEDs or stitch on some programmable NeoPixels!

This super-simple project lets you dress up your winter wear three different ways:

1. Use basic embroidery stitches to add a string of LEDs on a lovely silver wire, no coding required!
2. Go a little fancier with color-changing NeoPixels controlled by a Gemma M0 programmed in easy-to-learn Microsoft MakeCode.
3. For a challenge, weave a little CircuitPython magic into the Gemma to tap into a secret temperature sensor that shows you the chill factor in colored NeoPixels.
Parts List -- Non-Programmable Version
The Wire Light LED Strand comes with a battery holder -- all you need to add are two CR2032 coin cell batteries. And they're easy to work with, because the wires are insulated and won't short out if they touch. They're even waterproof! I used the Cool White LEDs, but you can also get red, blue, green, and warm white.

Wire Light LED Strand - 12 Cool White LEDs + Coin Cell Holder
Add sparkle to your project with these lovely silver wire LED strands. These strands are very interesting, they use two silver wires that are coated so they don't short if they...
https://www.adafruit.com/product/894

CR2032 Lithium Coin Cell Battery
A perfect match for our sew-able coin cell holder. This non-rechargeable coin cell is CR2032 sized: 20mm diameter, 3.2mm thick. It...
https://www.adafruit.com/product/654
Parts List -- MakeCode and CircuitPython Versions
The hardware is the same for both programmable versions. Just the code is different -- so you can even switch back and forth! And don't forget the batteries (above).

Adafruit GEMMA M0 - Miniature wearable electronic platform
The Adafruit Gemma M0 is a super small microcontroller board, with just enough built-in to create many simple projects. It may look small and cute: round, about the...
https://www.adafruit.com/product/3501

Flora RGB Smart NeoPixel version 2 - Pack of 4
What's a wearable project without LEDs? Our favorite part of the Flora platform is these tiny smart pixels. Designed specifically for wearables, these updated Flora NeoPixels have...
https://www.adafruit.com/product/1260
2 x 2032 Coin Cell Battery Holder - 6V output with On/Off switch
This tiny coin cell battery holder is ideal for small portable or wearable projects. It holds two 20mm coin cells (2032 are the most popular size) in series to generate 6V nominal. (If... https://www.adafruit.com/product/783

Stainless Medium Conductive Thread - 3 ply - 18 meter/60 ft
After months of searching, we finally have what we consider to be the ultimate conductive thread. It's thin, strong, smooth, and made completely of 316L stainless steel. Once you... https://www.adafruit.com/product/641

Suggested Tools List

Here are some recommended tools, but feel free to use what you have on hand. Standard crewel (embroidery) needles also work well with conductive thread.

Super Scissors
Save your scissors! Instead of using your nice shears (and dulling them) or wire cutters (not right for the job) - use these super scissors. They're meant for engineering/maker...
https://www.adafruit.com/product/1599
Needle set - 3/9 sizes - 20 needles
Mighty needles, sew like the wind! This needle set is the only one you'll need for any sort of hand sewing, especially using our conductive thread and wearable electronics...
https://www.adafruit.com/product/615

Suggested Materials List

The construction of the earwarmer headband is the same for all three versions. Here's what you will need, in addition to the parts and tools listed above:

- felt or fleece (in one or more colors) -- warm, soft, and easy to sew because they don't fray. The nice thick felt used for the examples in this guide was made by shrinking wool sweaters (Instructions for felting sweaters can be found multiple places online (https://adafruit.it/HFk) or in my book Fabric and Fiber Inventions (https://adafruit.it/HFl))
- regular cotton or polyester thread -- use where you want stitches to be secure but not overly visible
- embroidery floss -- great for decorative stitching that draws attention, and for covering up electronics
- straight pins -- to hold the fabric together while you work
- seam ripper -- for fixing mistakes

Optional, depending on the version you'll be making:

- tape (for holding the wire LEDs in place as you sew)
- adhesive dots (tiny ones to hold the NeoPixels in place as you sew)
- liquid seam sealant like Fray Check, or clear nail polish (for securing the conductive thread)
- press-on fabric adhesive sheets, such as Peel n Stick Fabric Fuse (for attaching felt appliques over NeoPixels and conductive thread stitches)
Measure and Cut Your Fabric

The first step is to cut your piece of fabric to fit your head. Use a cloth tape measure, piece of yarn, or your fabric to measure the width and length you like. This project is very forgiving, so don't worry if something doesn't come out exactly as you meant it to. It will probably still work.

First, decide how wide you want your ear warmer headband to be -- the examples in this guide had a finished width of 4 inches wide. Double that to get the width of the fabric piece (eg, 4 X 2 = 8 inches).

To get the length, wrap a cloth tape measure around your head where you would wear an ear warmer headband. Take the measurement of the front edge -- from the point that is closest to your forehead at the top, to the inside corner of the bottom (closest to your neck), going all the way around. Add about 4 inches to this length. The examples here were about 11 inches long, so the piece of fabric to work with was 15 inches long.
Cut out a strip of fabric using those measurements. If you need help, draw a paper pattern first, or draw on the fabric with tailor's chalk (or blackboard chalk if that's what you have lying around).

Mark and Stitch the Back Seam

To make the headband, fold the fabric strip in half, long edges together. The "right side" of the fabric (the side you want to show) should be facing out.

The folded edge is the front of the headband, closest to your face. The two short ends will be stitched together and go at the back of your neck.
Wrap the folded strip of fabric so the short edges are together at the back of your neck.

Pinch the edges of the fabric together at the top and bottom. Carefully take the headband off, then pin the strip at those points to hold the layers together.

Draw a gently curved line between these two points. This will make the headband fit more snugly around your head.

Take regular thread. Use a secure stitch like a backstitch (https://adafruit.it/HFm) to sew up the seam.

Trim the excess fabric to about a quarter of an inch. The excess seam material will be on the outside of the headband. You will finish it off with embroidery floss.

Leave the back edge of the band open for the time being and go on to the electronics. You will close it all up later.
Easiest Version - Embroider on a Wire Light

For this version of the headband, you'll be sewing a wire light onto the headband. The wire light is all one piece -- all you have to do is stitch around it to hold it in place. And it's waterproof, so it's OK to wear in damp weather (as long as the battery pack is tucked away inside the headband).

Since the stitches will be showing, you might as well make them part of the design! To do that, you will use embroidery floss -- a thick, shiny thread made up of six strands that can be separated or used all together.

And since the lights are pretty bright, you can also use embroidery floss stitches to cover the LEDs and diffuse the light. That reduces the intensity and spreads the light around to cover a slightly larger area.

You don't need to know any official stitches for this project -- it's perfectly OK to just make them up as you go along.

If you need help with the basics, The Hand Sewing Class on Instructables (https://adafru.it/HFn) can help you get started, and give pointers on the stitches mentioned here. The Instructables Embroider Class (https://adafru.it/HFo) also has tips for sewing with embroidery floss.

Be careful not to let sharp objects like needles, pins, or seam rippers damage the electronics!
Position the Wire Lights

Since you're limited by the length of the wire, start by putting the battery holder near the back of the neck.

Open up the headband and on the inside layer, mark and cut a slit a little further away from the bottom of the back seam than the length of the battery pack. Make it wide enough for the battery pack to fit through.

Put the battery pack inside the pouch formed by the headband, about where you want it to sit when you are wearing it. On the outside layer, mark and cut a small slit just big enough for the wire to fit through.
Attach the Wire to the Fabric

Pull out the wire and position it around the outside of the headband.

Make sure to leave enough extra wire to let you take the battery pack when needed.

Hold the wires in place with tape. Now you're ready to sew them on! The technique you will use is called **couching** ([Adafruit](https://adafruit.it/HFp)).

Anchor your thread or floss on the inside of the outer layer (inside the pouch). Wrap the thread around the wire three or four times, just before the first light. Then make tight little stitches around the wire, evenly spaced, to hold it in place against the fabric. Remove each piece of tape as you come to it.

When you get to an LED, make a stitch right before and after it. And make sure the bulb is pointing out!

Keep going to the end of the wire, then anchor the floss just past the last LED.
Embroider Over the LEDs

You can sew any kind of design to cover the LEDs and diffuse the light. Here they are covered with messy stars. Each star starts out with three criss-crossing stitches of thick embroidery floss.

Then go back in and make shorter stitches between the long ones. Turn the light on from time to time to make sure you are covering the glare.

Do the same on all the lights. If you use different color floss, you will get the effect of multi-colored lights.
Closing the Top

The final step is to close the top of the headband. Starting at the bottom of the back seam, sew through all the layers of fabric, gathering them together in the stitch.

When you reach the top edge, go all the way around.

A simple whip stitch works just goes over the top, around and around. It’s good for holding the edges of the opening together.

A blanket stitch ([https://adafruit.it/HFq](https://adafruit.it/HFq)) creates a line of thread or floss across the top of the opening, which gives you a nice finished look.

There are also more elaborate variations. This example uses a stitch called a crossed buttonhole. It takes a little practice to get it right!

If you make a mistake in your embroidery, you can take out the stitches with a seam ripper and start again. Just be careful not to damage the wire or electronics.
Intermediate Version - Sew on Gemma and NeoPixels

Want to add programming? With a Gemma M0 and Flora RGB Smart NeoPixels, you can create a string of lights that's codable in multiple formats. It lets you flash the lights on and off and change color, individually, to create an infinite variety of lighting effects.

All you need are the same sewing skills you just used to construct your earwarmer. If you're new to sewing circuits, there's a detailed guide to working with conductive thread here. (https://adafruit.it/HFr)
Prep Your Circuit

Circuit Basics
The Flora NeoPixels are individual color-changing LEDs on a miniature sewable board.

Look at it closely and you'll see metal pads with holes that let you sew them on.

There are pads for power (+) and ground (-). There are also two holes with pads for the signal from the Gemma M0 controller. They are marked with arrows. The arrows should be pointing away from the Gemma!

You'll be using conductive thread to create the traces, or lines of the circuit that connect the components together.

Here's how to connect the components:

- One trace (shown here as a yellow alligator clip wire) connects the D1 pad on the Gemma to the arrow on the NeoPixel pointing away from the Gemma.
- One trace (shown as the red alligator clip wire) connects the Vout pad on the Gemma to the power (+) pad on the NeoPixel.
- And the third trace (shown in black) connects the GND pad on the Gemma to the ground (-) pad on the NeoPixel.
To connect additional NeoPixels to the circuit, line them up so they are all pointing away from the Gemma. Then connect the arrow on the other side of the NeoPixel board to the arrow on the next NeoPixel.

You must use separate piece of conductive thread for the signal trace between each NeoPixel board.

The thread for the power and ground can each just continue on from one LED to the next.
Lay Out the Circuit
To keep your components in place in preparation for stitching them on, use a tiny adhesive dot to attach it to the fabric.

Make sure the glue doesn't cover the metal pads. The conductive thread must be able to touch the metal pads to make a good connection between the boards.

In this example, the NeoPixels are laid out in a wavy line.

The photo with the black, yellow, and red lines shows where the traces for the ground, signal, and power will be sewn.
Stabilize the Sewing Area

The felted sweater used for this example was still a little stretchy, so a thicker piece of felt was added inside the headband to stabilize it. It's held on by the stitches of the circuit.

Before starting the conductive thread stitches, secure the Gemma M0 board onto the fabric.

On the side not being used for the circuit, use non-conductive thread or embroidery floss to sew some stitches into the holes on the other side of the board in place.

The D0 pad was left empty in case you want to use it later as a touchpad or in other programming.

Conductive thread is made of steel -- it can cause a short circuit, just like an un-insulated wire! Keep checking while you work that you have no loose threads, ends, or split hairs that could touch another part of the circuit.
Conductive Thread Tips

The conductive thread can be stiff and wiry. To help get it onto the needle, here’s a trick:

Fold a piece of regular thread into a loop. Pass the loop through the eye. Next, pass the conductive thread through the regular thread loop. Then pull the tails of the regular thread back through the eye of the needle. The conductive thread should come along with it.

You can never make too many knots when sewing with the conductive thread! However, that creates another problem: Loose thread tails that can cause short circuits. So always anchor the tail under nearby stitches whenever possible. Make sure you are only connecting that trace to the same trace!

If there are a lot of different traces close together near where you want to end a thread, make a few stitches back in the opposite direction. Then knot the thread in the middle of the line of stitching.

If you have a double layer, you can make these stitches through the inner layer only so they don’t show.
Sew the Circuit Traces

Begin with the signal thread traces. Knot the thread around the D1 pad hole to start. Go around several times until it's nice and secure.

Then use a short running (in and out) stitch until you reach the "in" signal pad on the first NeoPixel. Since the hole here is smaller, bring the needle up from the wrong side of the fabric before the NeoPixel. Then you can aim the point of the needle down into the hole.

Again, go around through the hole as many times as you can. Then take a few stitches back away from the NeoPixel. Knot the thread around the stitching and cut it, leaving a short tail. You will secure the loose end later.

Sew the trace between the other arrow pad on this NeoPixel to the proper arrow pad on the next NeoPixel. Continue until they are all connected.

Then sew a trace from the GND pad on the Gemma M0, connecting to the (-) pads on all the NeoPixels. Make sure this trace doesn't touch the first line of stitching, or any other parts of the boards!

Do the same with the trace from the Vout pad on the Gemma M0 to the (+) pad on all the NeoPixels.

Check once more that no loose threads are touching anything they shouldn't.

Then test your circuit by running a program on the Gemma M0. Go to the MakeCode page or the CircuitPython page to program the board, depending on which version you want to use.
Finishing the Circuit

On the back of the circuit, use liquid seam sealer or clear nail polish to glue any loose threads in place. Let dry before you do anything else.

The exposed stitches on the right side of the headband should be protected too. You can use couching stitches (like in the non-programmable Wire LED version), making the stitches close enough to touch.

Or you can stick felt shapes, known as appliques, over the stitching.

Cut out your felt shape, then trace around it on sheet of a peel-and-stick fabric fuse. Peel off the backing paper on one side and press it onto your felt shape. Then peel off the other backing paper to expose the adhesive.

Place the felt shape over the exposed stitches. If any stitches are still sticking out when you are done, you can coat them with a little more seam sealer.
The circuit in this project is protected but it is NOT waterproof! Be careful and avoid wearing your headband in very damp weather.

**Diffusing the LEDs**

To protect and diffuse the lights in your circuit, you can use the same applique technique and stick a piece of fabric right over the LED. Use a whitish color if you want the colors of the LEDs to come through clearly.

For this design, small strips of felt are rolled up to look like roses. To hold the rose together, pass a needle with regular thread through the entire roll, then take a few stitches to hold the thread in place. Go through the whole roll in different directions, then make a few stitches and tie off the end. Tuck the ends inside the rose.

Trace and cut the peel-and-stick fabric fuse the same way as above, and press them over the LEDs.
Covering the Gemma

If you want to cover the Gemma M0 as part of your design, make sure you can still get to the buttons, switches, and plugs.

Here a flower applique, fortified with some decorative blanket edge stitching, is attached to the headband with a few stitches of regular thread in two spots.

That makes it easy to lift a flap from top or bottom to get to all the important parts.

When not in use, the applique can be smoothed flat to protect the Gemma and diffuse the light of the onboard LED, which is programmed to be part of the animation.
The next pages tell you how to program your lights in either MakeCode or CircuitPython!

MakeCode Maker Animation

It's simple to program your Twinkly Earwarmer with the Gemma M0 in Microsoft MakeCode! The version to use with Gemma M0 is called MakeCode Maker (https://adafru.it/C9N). It lets you program the onboard color-changing LED, and add components to the board's pins.

For this project, you need to tell the board where to find a strip of four external NeoPixels. Combine them with Gemma's onboard RGB DotStar LED and you can create an animated light show with five lights!

If you haven't used Gemma M0 with MakeCode before, there's more information about using MakeCode on the Gemma M0 guide (https://adafru.it/Dvk).

To get started right away, you can just open and edit the sample code shown at the top of the page.

Click the button below to open the MakeCode Maker editor, then add to or change the light display as you wish.
To download the file to the board, connect the Gemma M0 to your computer with the USB cable. Press the reset button and it will go into bootloader mode. That means it's ready to receive code.

Click **Download** on the MakeCode page and save the file. It will have a .uf2 extension.

Then find the drive on your computer labeled GEMMABOOT. Drag and drop (or copy and paste) the program into Gemma! You'll know the code was downloaded when the onboard pixel changes to violet.

While the Gemma is still connected to the computer, check how the MakeCode runs on the actual NeoPixel strip so you can make any adjustments.

If it is your first time running MakeCode on your Gemma board, or if you have previously installed Arduino or CircuitPython, you may need to double press the reset button to get your board into bootloader mode.

There are lots of ways to animate your strip of pixels! To create your own animation, you can use the `strip` code blocks, found in the LIGHT category. Then you can preview it in the simulator before you download it to your Gemma.
Here's a detailed explanation (https://adafruit.it/HFt) of how the NeoPixels work and how to make them do interesting things.

You can see and edit the handcoded animation shown above by clicking the button below. Have fun!

Click here to see a handcoded MakeCode animation
https://adafruit.it/HFs

Advanced Version - CircuitPython Chill Sensor

Want more of a challenge? If you code the board in CircuitPython, you can access the hidden temperature sensor and use it to control the colors of your pixels!

Adafruit's Anne Berela has written code that will turn the lights red if the temperature is above 65 degrees and blue if it goes below that.

If you are new to CircuitPython, there are few steps to go through to use this code. Follow the instructions below or go here (https://adafruit.it/BIM) to learn more about installing and programming in CircuitPython.
Loading CircuitPython on Your Gemma M0

This project was tested using CircuitPython 4. If you're not sure if that's what you are running on your Gemma M0, plug in your board, go to CIRCUITPY drive, and check the folder marked boot_out.txt. It will tell you what version is loaded on the board. The latest stable version should be fine.

If you need to install or update CircuitPython, tap the reset button once or twice to enter bootloader mode.

Once successful, the RGB LED on the board will flash red and then stay green. A new drive will show up on your computer called GEMMABOOT.

Then click the button below, which will take you to the download page for the latest version of CircuitPython. Save that file to the GEMMABOOT drive on your computer.

You've now installed CircuitPython on your board! The drive you will be working with from here on is called CIRCUITPY.

Click here to download the latest version of CircuitPython!
https://adafru.it/Eme

Next, you'll need to install some libraries. CircuitPython libraries are separate files designed to work with CircuitPython code. CircuitPython programs require a lot of information to run. CircuitPython is so simple to use because most of this information is processed in the background and stored in libraries. Some libraries are built into CircuitPython. Others are downloaded and stored on your CIRCUITPY drive in a folder called lib.

Click the button below and download the bundle for use with your version of CircuitPython. Save it on your computer, not the board! You'll only need two files from this huge folder.

Look for these two files -- neopixel.mpy and adafruit_dotstar.mpy -- and drag them to the lib folder on the Gemma M0.

Click here to find the CircuitPython library to use with your board!
https://adafru.it/ENC
Install the Mu Editor

Adafruit recommends using the Mu editor on your computer to work with CircuitPython code. Instructions for downloading it to your computer can be found [here](https://adafruit.it/ANO) or the button below.

Once you've got it installed, open it up. You're ready to copy the code to make your temperature sensor lights work!

Click here for instructions on installing the Mu editor!

[https://adafruit.it/ANO](https://adafruit.it/ANO)

Temperature Sensor Code

Copy and paste the code below into your editor or click the download link.

```python
# Gemma M0 Onchip Temperature Sensor
# Project by Kathy Ceceri
# CircuitPython by Anne Barela
# Adafruit Industries, 2019

import time
import board
import microcontroller
import neopixel
import adafruit_dotstar

ROOM_TEMP = 65.0  # Set this to the temp to change from blue to red (F)

# Set up NeoPixel strand
pixels = neopixel.NeoPixel(board.D1, 4, brightness=0.2)  # Change from 0.0 to 1.0

# For the Gemma M0 onboard DotStar LED
dotstar = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)

def deg_f(deg_c):
    # Convert Celcius to Fahrenheit
    return(deg_c * 9 / 5) + 32.0

while True:
    temp = deg_f(microcontroller.cpu.temperature)
    if temp > ROOM_TEMP:
        pixels.fill((255, 0, 0))  # (255,0,0) is red
        dotstar.fill((255, 0, 0))  # Set to red
    else:
        pixels.fill((0, 0, 255))  # (0,0,255) is blue
        dotstar.fill((0, 0, 255))  # Set to blue
    time.sleep(1.0)  # Wait 1 second
```
If you need to make any adjustments, such as changing the "room temperature" cutoff that triggers the colors (ROOM_TEMP), or you have a different number of pixels attached to your Gemma M0, you can type them right into the code.

Then click the Save button at the top of the Mu editor and save this file as code.py on your CIRCUITPY drive.

If everything is working, the onboard LED (and the external NeoPixels if they're connected) will turn red.

Test Your Code
Now disconnect the Gemma from the computer and connect it to the battery holder. Take the board outside (if it’s chilly), place a bag of frozen veggies on the board, or just toss it in the freezer. If it’s working, the lights should go from red to blue and back again as it cools down and warms up.

If it's not, or if it's working but a little wonky, check your conductive thread connections or your batteries.
Like any tech project, it takes some fiddling to make the hardware and software cooperate. Enjoy playing with your temp-sensing headgear!