

Simple Pendant

Difficulty level Easy - Very few parts and no soldering

Time required 2 hours approx. depending on embellishments.

Description This project uses the basic Circuit-Playground board, a plug-in battery, decorations, and some sample code. It does not require soldering or other electrical modifications. However, you can change your mind and add other items if decorations have not yet been glued to the board. When this project is complete, it can be used as is, or the programming can also be modified to change the color pattern, or to add features such as changing the display with sound, motion, or temperature. These directions only shows how to build the basic project. If you choose to make later additions or modifications, please check out the other project instructions, or look for additional project information on the Adafruit website.

Step one – secure the materials.

To build this project you will need a Circuit Playground or Gemma board, rechargeable Lithium Ion battery and battery charger, and a decorative pendent, or other item to be illuminated by the circuit. For example, some versions of this project use a sea shell to attach in front of the circuit in place of the pendant which can help add interest to the project and diffuse the light. All of these items should have been included in a package with these instructions. Please be sure that your kit came with all the items needed to build this project.

Step Two – Plug in the battery to test the circuit.

The rechargeable battery should have been charged before you received the kit, however you should test the battery and the circuit board by plugging it into the circuit playground board to see if it will light up. The Circuit Playground boards should have a default program loaded in them to control the color changing LEDs on the board. Make sure that the board lights up when you plug in the battery and that you are happy with the pattern of lights that you get. If the board does not light up or does not give you the effect that you expected, the problem can probably be corrected by downloading a new program as seen in the next step.

By the way. If you are running the default program you can toggle between a few different patterns by pushing one of the built in buttons on the circuit board. The other built in button can be used to toggle through different brightness settings.

Step Three – Download a new program (optional)

1. Download the code repository [here](<https://github.com/Team997Coders/wearable-workshop>)
 1. Click on **code** on the top right of the screen and select **Download ZIP**.
 2. Navigate to your downloads folder and extract the ZIP file.
 3. Navigate to the newly created folder and find pendant-playground-express/code.py file.
2. Download code to the pybadger
 1. Plug in the Circuit Playground Express using the USB cable into your computer. This should show up as a USB flashdrive.
 2. Navigate to the USB flashdrive file folder.

3. Copy the pendant-playground-express/code.py file onto the flashdrive. You will need to overwrite the previous code.py.
4. The Circuit Playground Express after a few seconds should start running the new code.py!

Step Four – Add the cover.

In order to hide the electronics and make your project more interesting you can add a top layer by gluing the pendant or another item like a shell to the top of the circuit. But be sure that you do not want to make any other changes before you do this because it would be difficult to solder new electrical parts to the board later if you wish to enhance your project further. If you have not made up your mind yet you can use a limited amount of glue to make it easier to take apart later if you do want to make electrical changes. You can also choose to tie or tape the parts on for now, or just hold them in place to get an idea of how the finished project will look. But if you are sure, the front of the pendant that you have chosen can be secured using hot glue. It is best to use several small dabs around the edges to produce the most secure mounting.

Step Five – Make your own program changes

If you want to further personalize this project the best way to accomplish that is by making changes to the software that control it. This can be as simple as changing a few numbers in the code that control how fast the LEDs blink or change colors. Look for lines with `time.sleep` and experiment with different values to see how that changes the appearance of the display. You might also experiment to see if you can change the colors or intensity. There are also sensors on the Circuit playground that detect motion, temperature, sound, etc. It is possible to change the program to have the lights change based on background music, movement, or other things. Controlling the lights using these other sensors is complex, so you might want to look up examples on Adafruit.com for more ideas, or ask a mentor if you get stuck.

If you want to try finding a new program to run, it would be best to start with one intended to light up the LEDs on the Circuit Playground. You can try using a program designed to control LEDs on a different board. However, you might need to make minor changes to get a program designed for different hardware to work on Circuit Playground. The most common problem is that the circuit playground has the LEDs connected to pin 8, and programs designed for different hardware might use a different I/O.

Step Six – Make electrical changes or using the circuit for something else.

The Circuit Playground used in this project is actually a powerful micro-controller. If you ever get tired of using it as a pendant, there are a lot more things that you can do with it. Any of the projects designed for the Circuit Playground can be run on this board. You might need to remove the pendant cover to connect your new project parts, but this circuit is capable of doing a lot more than just blinking a few lights. You could even use it to run many programs that were not intended for the Circuit Playground at all. Micro-controllers are quite adaptable, and you may find many other uses for this circuit in the future with a bit of imagination.