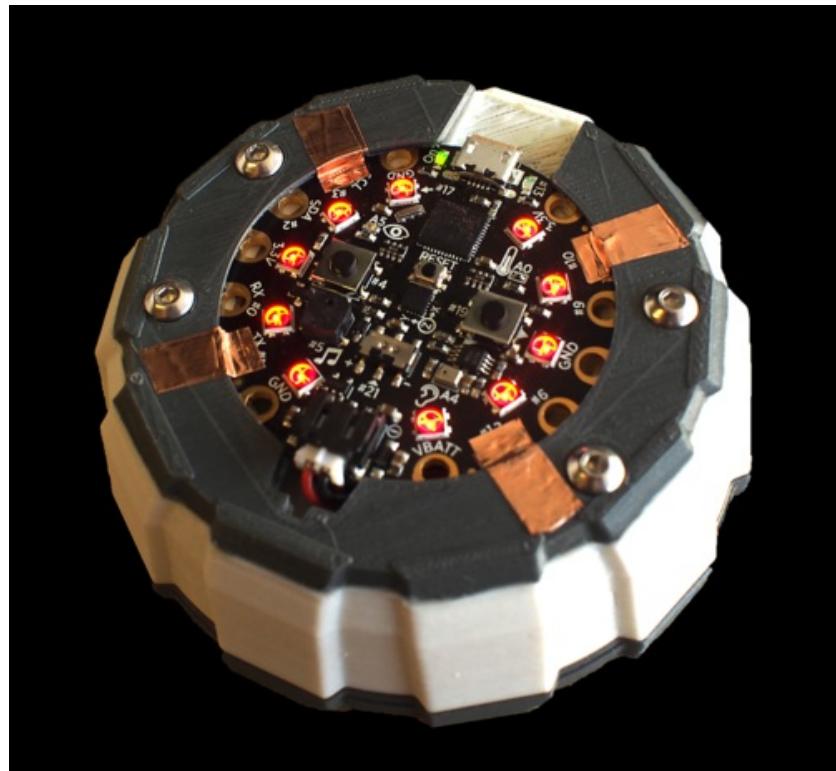


□

Circuit Playground Password Vault

Created by John Park

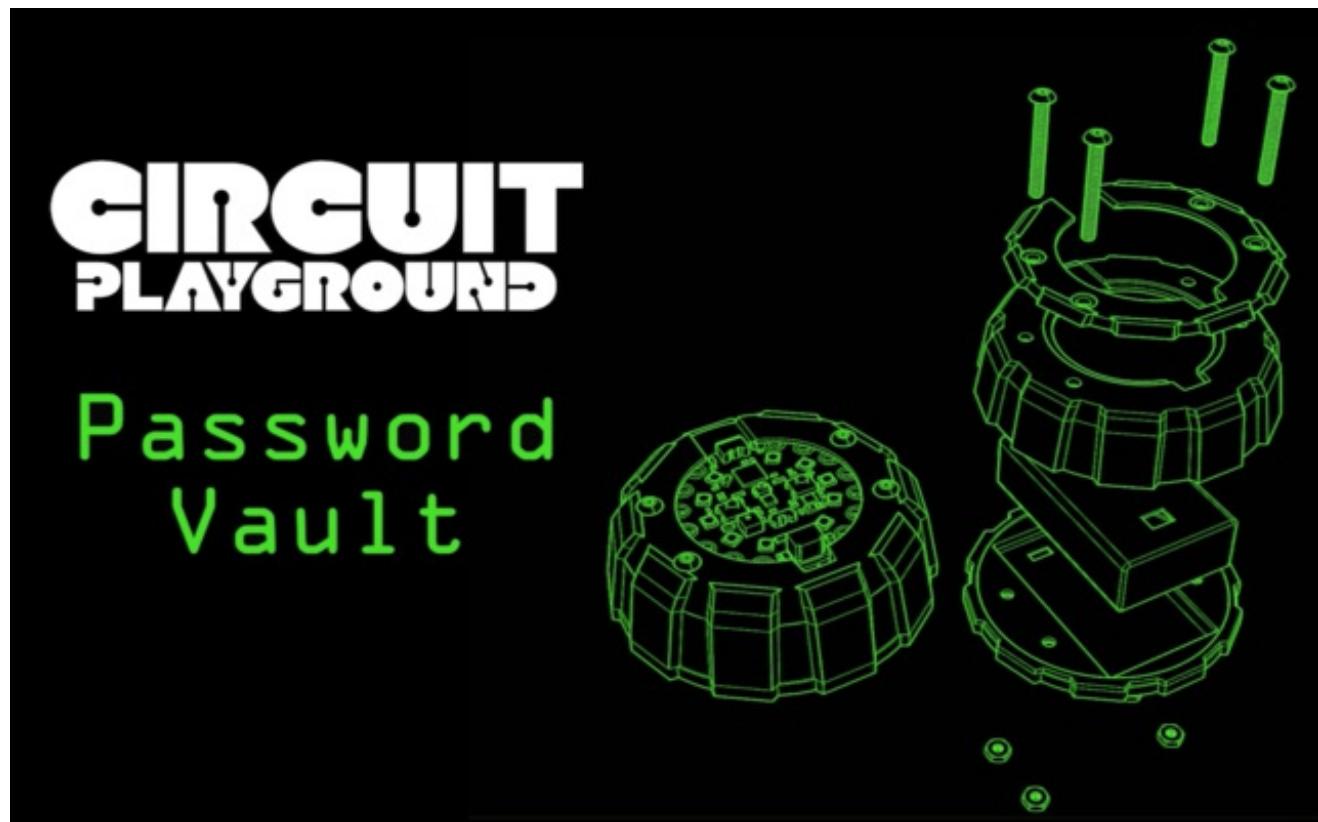


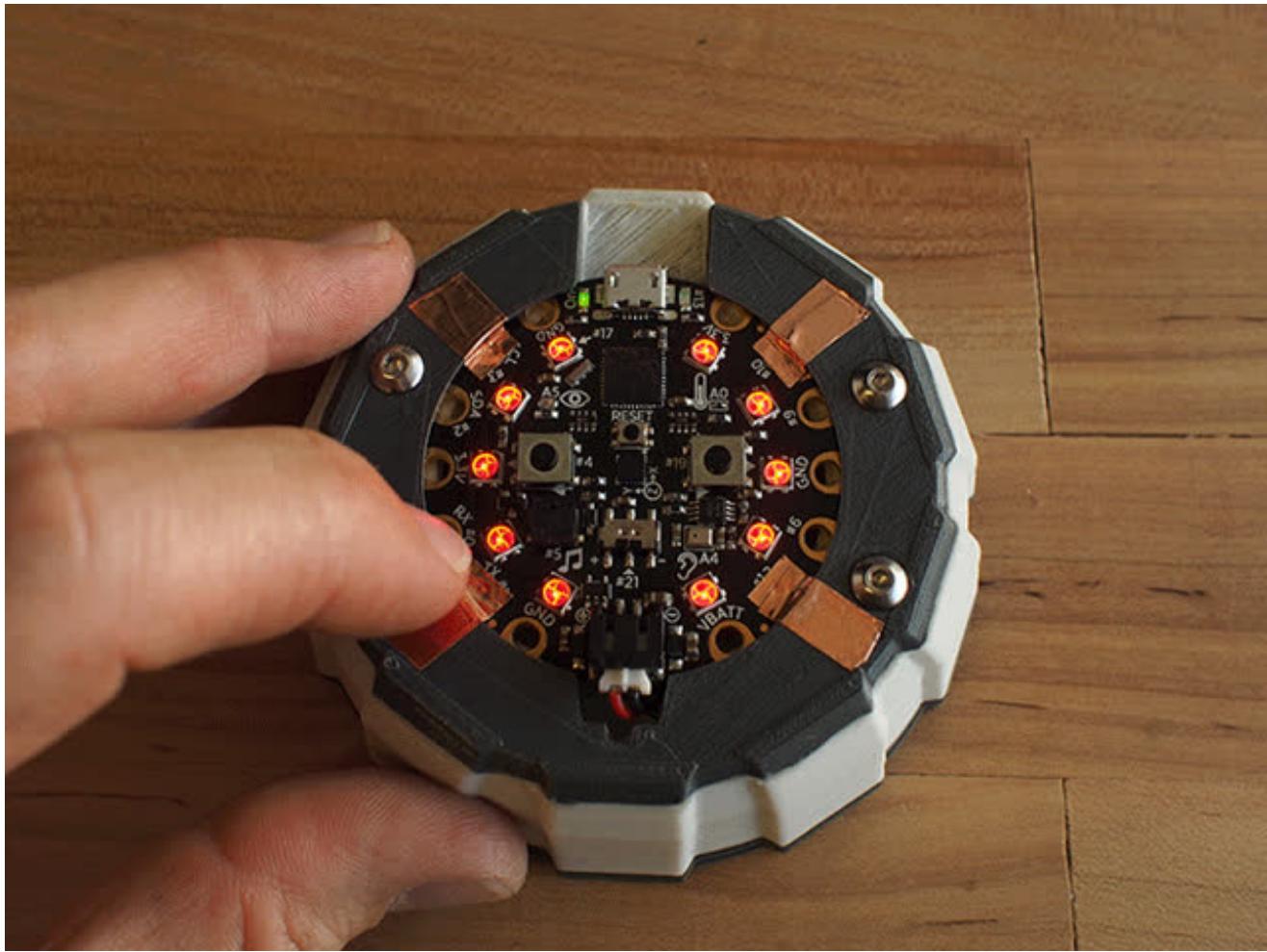
Last updated on 2016-09-26 02:40:52 AM UTC

Guide Contents

Guide Contents	2
Overview	3
Secure, Accessible Passwords	4
Prep the Circuit Playground	5
Required Parts	6
Optional Parts	8
3D Printed Enclosure	9
Alternate Enclosures	10
Password Vault Coding	11
Password Vault Logic	11
Arduino Sketch	11
Download the Code	16
Adjust the code	16
Save It	17
Upload Code to the Board	17
Build the Enclosure	18
Making a Case	18
3D Printed Parts	18
Assemble the Case	19
Capacitive Switches	27
Fasten	34
Finishing	36
Use the Password Vault	40
Startup and Unlock	40
Password Retrieval	42

Overview





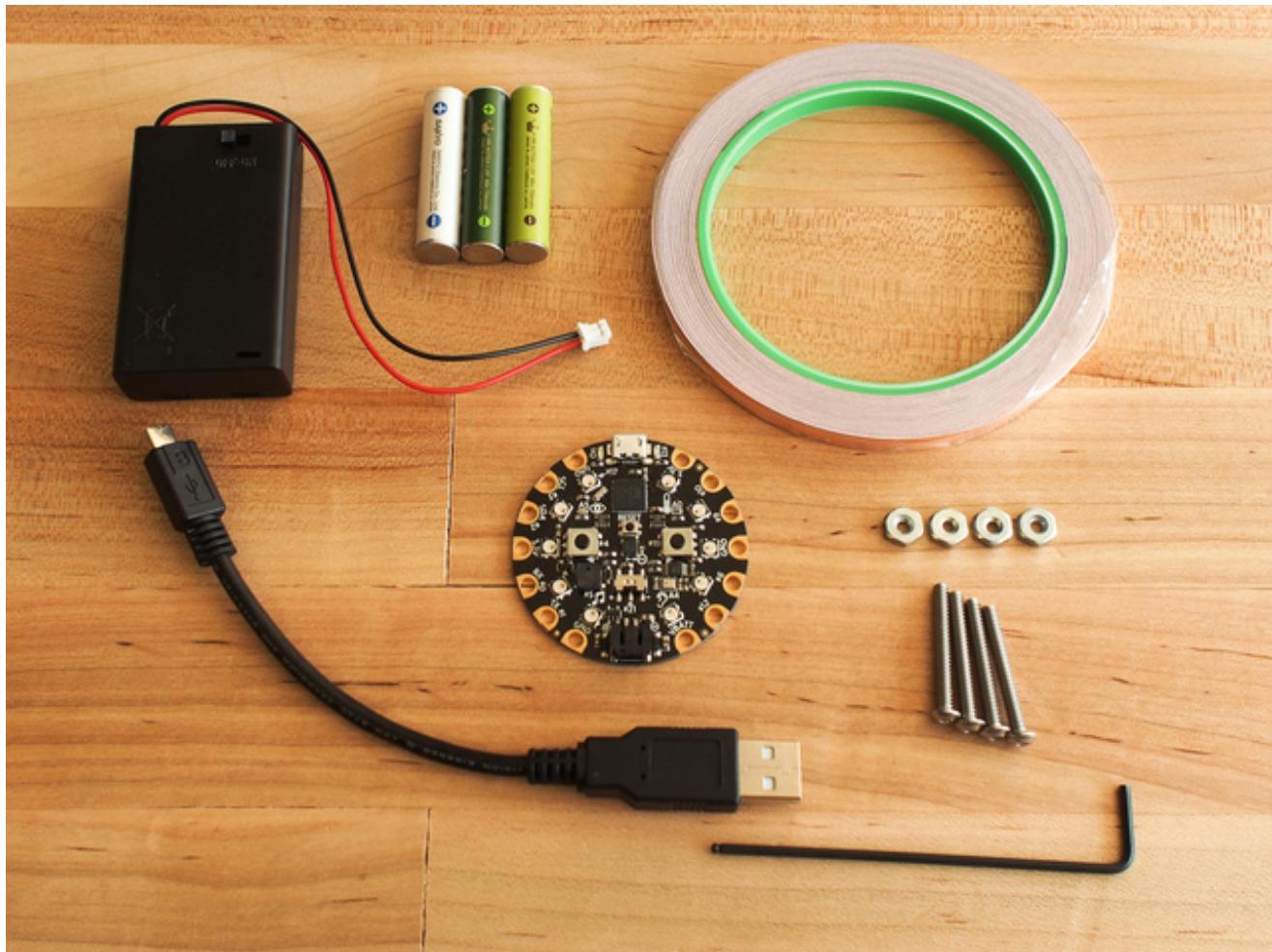
Secure, Accessible Passwords

Long, random passwords -- such as eF@V3p%2R*qhw8 are great because they're so secure when compared to easy ones such as Pencil or 12345, but can be difficult to remember and to type. Once you build this **Circuit Playground Password Vault**, you won't need to remember or type them ever again! *The Circuit Playground can act as USB keyboard* so it can be plugged into any computer and be used to enter password for you.

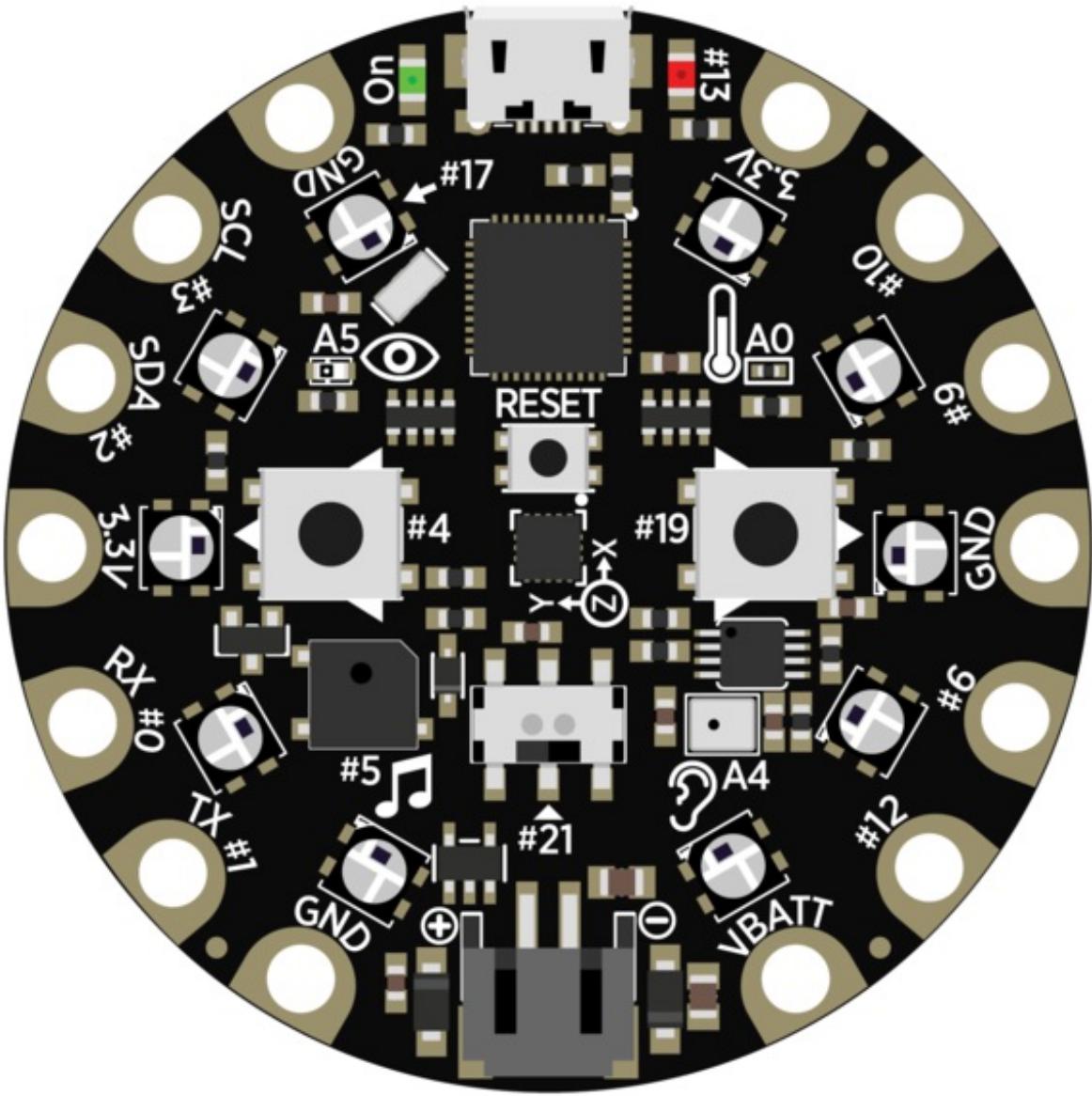
But what if the Password Vault should fall into the hands of an enemy or sibling? Not a problem! You'll create a unique unlock sequence that must be keyed in to unlock the device, so your passwords stay secure.

Once you've built the Password Vault, you can store up to ten passwords and deploy them at the push of a button!

Prep the Circuit Playground

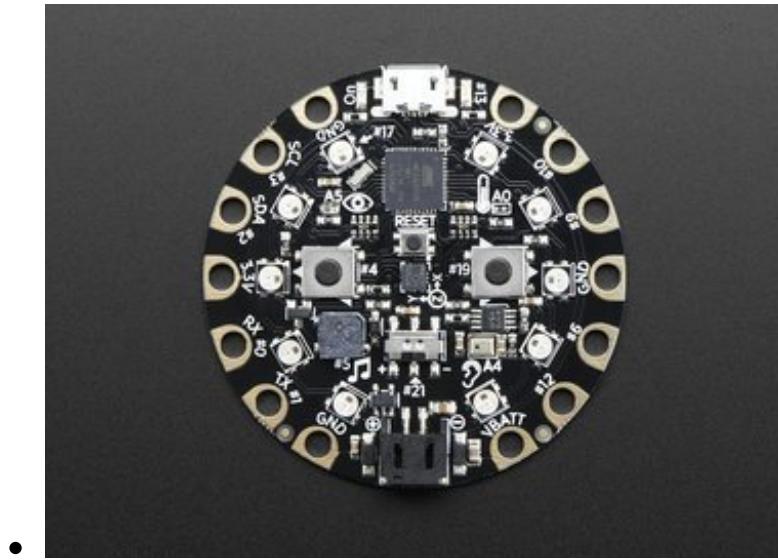


Before you get started, make sure you've been through the basic tutorials on using the Circuit Playground. You'll need to have installed the Arduino IDE, added the Circuit Playground board to Arduino, and added the Circuit Playground library. This excellent guide, [Introducing Circuit Playground](http://adafru.it/pAP) (<http://adafru.it/pAP>) and [Circuit Playground lesson #0](http://adafru.it/r0D) (<http://adafru.it/r0D>) will show you how to do all of that and get you going.



Plug your Circuit Playground into your computer now, and launch the Arduino IDE. Double check that you've selected the board and port as in the Lesson #0 tutorial and are ready to upload code.

Required Parts



A Circuit Playground!

Available at
Adafruit (<http://adafru.it/3000>)



Micro USB Cable any length.
You probably have one of these around, they're the most common USB cable.

Make sure its a data/sync cable!

USB cable available at
Adafruit (<http://adafru.it/592>)

A HUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times students have nearly given up due to a flakey USB cable!

Battery Holder with power switch

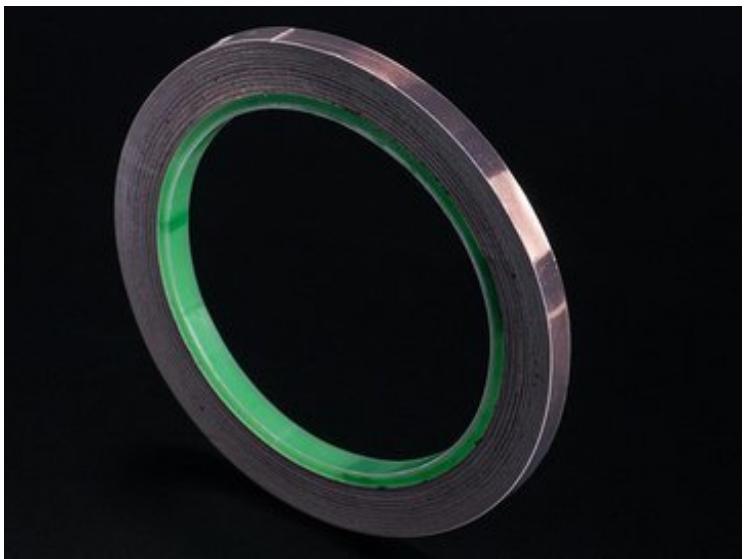
3 x AAA Battery Holder with



[On/Off Switch and 2-Pin JST](http://adafru.it/727) (<http://adafru.it/727>)

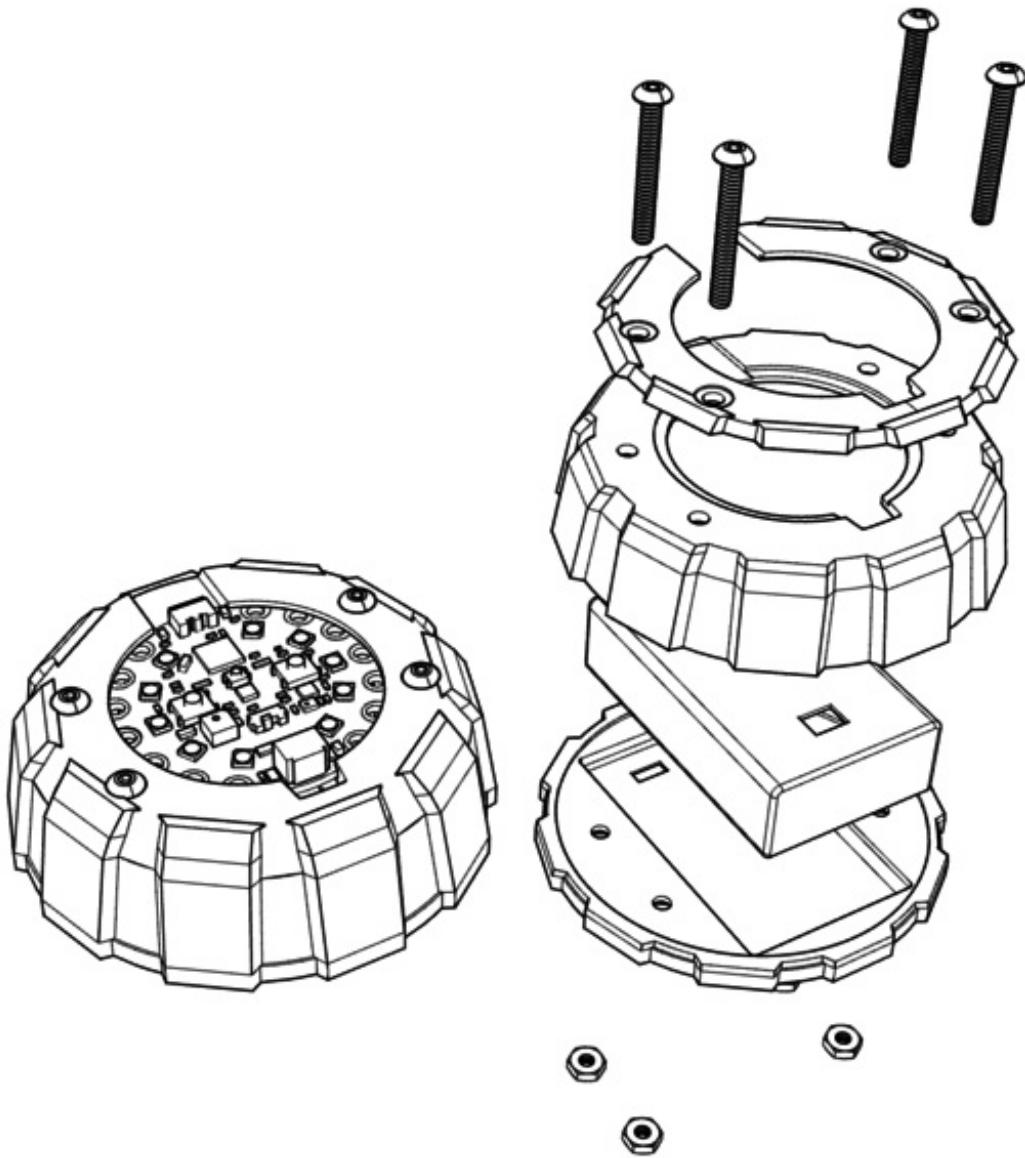
3 AAA batteries

Technically, this project could be done without batteries, since USB supplies power, but having batteries allows you to enter your unlock code and prep the Vault before making the connection.



[Copper Foil Tape with Conductive Adhesive](http://adafru.it/1128) (<http://adafru.it/1128>)

Optional Parts



3D Printed Enclosure

The Password Vault was designed to be enclosed in a 3D printed case that allows access to the battery pack ON/OFF switch as well as the four copper pads for entering the unlock code.

If you like, download and print the case yourself, or send the files to an online 3D printing service.

You'll also need the following hardware to fasten the case:



4x [#6-32 thread, 1-1/8" long button-head cap screws](#) (<http://adafru.it/aLt>)

4x [#6-32 threaded hex nuts](#) (<http://adafru.it/aLt>)

•

Alternate Enclosures

You can also choose to make your own unique enclosure. Build one from PVC pipe, a tennis ball can, or maybe hide it inside a bandage box, it's fun to look around your garage, junk drawer, or the hardware store to see if you can reimagine something as a cool new case for your Password Vault.

Password Vault Coding

Password Vault Logic

The Password Vault sketch does three basic things:

1. When turned on, it remains locked until you enter the unlock code
2. Allows you to select one of your ten saved passwords
3. "Types" your password over USB into a computer

Arduino Sketch

This is the complete Password Vault sketch. You can download it from the link below.

```
// -----
//Circuit Playground Password Vault
// John Park
// for Adafruit Industries
//
//Password Vault stores up to ten passwords, uses a code sequence to unlock,
//can type your long passwords for you!
//
// Plug into any computer with a USB cable
// Flip the slide switch and then tap your secret code sequence on the
// capacitive pads
// If wrong code is entered, buzzer will sound and red LED will blink.
// When code is successfully entered, the NeoPixels will light green.
// Now, you can choose one of up to ten stored passwords with the right
// button, you'll see a corresponding NeoPixel light up for your selection
// Press the left button and your password will be typed by the Circuit
// Playground over USB into the password field (be sure to click there with
// your mouse first so the field has focus)
//
// For use with the Adafruit Circuit Playground.
//
// MIT license, check LICENSE for more information
//
// -----
#include <Adafruit_CircuitPlayground.h>
#include <Wire.h>
#include <SPI.h>
#include <Keyboard.h>
```

```

//*****
//Create your own secret six-digit unlock sequence here.
//The sequence can be any combination of six touches
//on capacitive pads #1, #3, #10, & #12
//*****
//Alter the line below to change the sequence
int codeSeq[] = { 1, 3, 10, 12, 10, 10};

//This array is the placeholder that will change as the code is entered,
//when codeEntry[] matches codeSeq[], sucess
int codeEntry[] = { 0, 0, 0, 0, 0, 0};

//*****
//Enter your passwords here,
//up to ten of them:
//*****

char* PASSWDs[] = {
    ..R45p83rrY..",
    "p3Nc1l.6{fG3$;" ,
    "3Ff0rT@9j2y/&" ,
    "P0intS^[F4f8?" ,
    "DoUBLeA87*+F]" ,
    "?F3ErPsS.C.M.O.D.S." ,
    "mK$Bv7M;?gyFu2" ,
    "6@LKNs(WV[vq6N" ,
    "HorseStapleBattery" ,
    "bLk3y3d4p1R4t3,)" } ;

//-----
//Global Variables
bool codePrompt = 0; //readiness to enter code
bool codeBegun = 0; //code entry has begun
bool codeCompare = 1;//compare code entry and correct code
bool passPrompt = 0; //ready to select password slot
bool passPrep=0; //password entry state prep
int passSlot = 0; //current password slot choice
bool passKeyed = 0; //password has been keyed in over USB
int n = 0; //code entry array position

void setup() {
    Keyboard.begin(); //allows Circuit Playground to act as USB HID device
    CircuitPlayground.begin();
    CircuitPlayground.playTone(932,200); //startup beep (A# just like an
    //Apple IIe)
    int i=0; //flash the red LED
    for (i=0;i<2;i++) {
        CircuitPlayground.redLED(true);
        delay(200);
}

```

```

CircuitPlayground.redLED(false);
delay(200);
}
CircuitPlayground.setBrightness(15); //fairly dim pixel brightness
//setting -- goes up to a blinding 255!
CircuitPlayground.clearPixels(); //clean up if any were previously on
}

void loop() {

//-----
//Slider check.
//read slider to see if it is time to ender code
if(codeBegun==0){
    bool previousSliderState = CircuitPlayground.slideSwitch(); //get the
    //initial state of the slider switch as a reference to compare when it changes
    delay(20); //wait a moment
    bool sliderState      = CircuitPlayground.slideSwitch(); //check slider
    //again

    if ((sliderState == previousSliderState)&&(codeBegun == 0)) {
        return; //no change, do nothing
    }
    else if ((sliderState != previousSliderState)&&(codeBegun == 0)) { //the
        // slider was moved
        //light the ring of NeoPixels
        int i;
        for(i=0;i<10;i++){
            CircuitPlayground.setPixelColor(i, 255, 0, 0);
            delay(100);
        }
        codeBegun = 1;//flip this so code entry will be accepted
        previousSliderState=sliderState; //set these the same again so they
        //can detect future switches
        codePrompt = 1; //flip this so the code prompt will occur
    }
}
//-----
//Code entry.
//Allow unlock code to be entered using capacitive pads
if ((codePrompt == 0)&&(passPrompt==0)) { //do nothing until code prompt flips
    return;
}
else if ((codePrompt == 1)&&(passPrompt==0)) {//check the cap switches for entry
    int cap1=CircuitPlayground.readCap(1);
    int cap3=CircuitPlayground.readCap(3);
    int cap10=CircuitPlayground.readCap(10);
    int cap12=CircuitPlayground.readCap(12);

    //check the cap switches
}

```

```

if (cap1<25 && cap3<25 && cap10<25 && cap12<25) {
    return;
}
else if (cap1>25) {
    codeEntry[n]=1;//add code entry to the array
    //light a pixel yellow
    CircuitPlayground.setPixelColor(n,255,255,0);
    n++;
    delay(400);
}
else if (cap3>25) {
    codeEntry[n]=3;//add code entry to the array
    //light a pixel yellow
    CircuitPlayground.setPixelColor(n,255,255,0);
    n++;
    delay(400);
}
else if (cap10>25) {
    codeEntry[n]=10;//add code entry to the array
    //light a pixel yellow
    CircuitPlayground.setPixelColor(n,255,255,0);
    n++;
    delay(400);
}
else if (cap12>25) {
    codeEntry[n]=12;//add code entry to the array
    //light a pixel yellow
    CircuitPlayground.setPixelColor(n,255,255,0);
    n++;
    delay(400);
}
}

//-----
//Code check.
//Six digits have been entered, test if they're the correct ones
if ((n<6)&&(passPrompt==0) ) {
    return;
}
else if((n>=6)&&(passPrompt==0)) {
    //memcmp compares two arrays
    codeCompare = memcmp(codeSeq, codeEntry, sizeof(codeSeq)); //0 = match
    if (codeCompare==0) {
        CircuitPlayground.playTone(632,100); //beep
        delay(200);
        CircuitPlayground.playTone(632,100); //beep
        //light up green
        int i;
        for (i=0;i<10;i++) {
            CircuitPlayground.setPixelColor(i, 0, 255, 0);
            delay(100);
    }
}
}

```

```

        codePrompt=0;//to stop taking entries
        passPrompt = 1; //ready to pick password slot
    }
}
else {
    CircuitPlayground.playTone(232,400); //beep
    n=0;//reset counter for entry
    //reset pixels to red
    int i;
    for (i=0;i<10;i++) {
        CircuitPlayground.setPixelColor(i, 255, 0, 0);
        delay(100);
    }
}
}

//-----
//Password selection prompt.
//Ready for button input
if (passPrompt == 0) { //not ready to pick password slot
    return;
}
else if ((passPrompt==1)&&(passKeyed==0)&&(passPrep==0)){ //ready to pick a password slot
    if (passPrep==0){
        CircuitPlayground.setPixelColor(passSlot, 255, 255, 255); //light slot 0
        passPrep=1 ; //flip this so we move on
    }
}

else {
//-----
// Check for button presses.
// Detect if the left or right button is released by taking two
// readings with a small delay in between. If the button changes
// state from pressed -> released then we can update indicators.
bool leftFirst = CircuitPlayground.leftButton();
bool rightFirst = CircuitPlayground.rightButton();
delay(10);
bool leftSecond = CircuitPlayground.leftButton();
bool rightSecond = CircuitPlayground.rightButton();

//-----
//Right button selects password slot.
if (rightFirst && !rightSecond) {
    passSlot++; //increments the password slot
    CircuitPlayground.setPixelColor(passSlot-1, 0, 255, 0); //fill in the previous pixel to green again
    CircuitPlayground.setPixelColor(passSlot, 255, 255, 255);

    if (passSlot > 9) { //rolls over from tenth slot to first
        passSlot=0;
        CircuitPlayground.setPixelColor(passSlot,255,255,255);
    }
}
}

```

```

        }
    }

//-----
//Left button prints the password over USB.
if (leftFirst && !leftSecond){
    CircuitPlayground.playTone(432,400);
    delay(100);
    CircuitPlayground.playTone(500,300);
    CircuitPlayground.setPixelColor(passSlot, 0, 0, 255);
    Keyboard.print(PASSWDS[passSlot]) ;
    passKeyed = 1;//password has been printed over USB
}
}

}
}

```

Download the Code

You can press the button below to download the code. Unzip and move the directory *CircuitPlaygroundClassScheduler* to your Arduino sketch directory.

[CircuitPlaygroundPasswordVault.zip](#)

<http://adafru.it/rc1>

Adjust the code

Before uploading the code to your Circuit Playground, you'll want to change a the unlock sequence and the actual passwords you want to store and use. Open the *CircuitPlaygroundPasswordVault.ino* file in the Arduino IDE, and then make changes to these two sections shown below.

```

*****
//Create your own secret six-digit unlock sequence here.
//The sequence can be any combination of six touches
//on capacitive pads #1, #3, #10, & #12
*****
//Alter the line below to change the sequence
int codeSeq[] = { 1, 3, 10, 12, 10, 10};

*****
//Enter your passwords here,
//up to ten of them:
*****


char* PASSWDS[] = {
    ..R45p83rrY..",
    "p3Nc1l.6{fG3$;" ,
    "3Ff0rT@9j2y&" ,

```

```
"P0intS^[F4f8?" ,  
"DoUBLEaA87*+F]" ,  
"?F3ErPsS.C.M.O.D.S." ,  
"mK$Bv7M;?gyFu2" ,  
"6@LKNs(WV[vq6N" ,  
"Tt.fV2Vf.D[8sb" ,  
"bLK3y3d4p1R4t3,)" } ;
```

Save It

Once you've adjusted those sections, save the sketch. In the next section, you'll upload it to the Circuit Playground.

Upload Code to the Board

Plug your Circuit Playground into your computer with the USB cable.

In the **Arduino IDE**, select the **Adafruit Circuit Playground** item in the menu Tools > Board.

Then, select the proper port in **Tools > Ports**.

Press the reset button twice on the Circuit Playground so that the red #13 LED begins pulsing, indicating it is ready to be programmed.

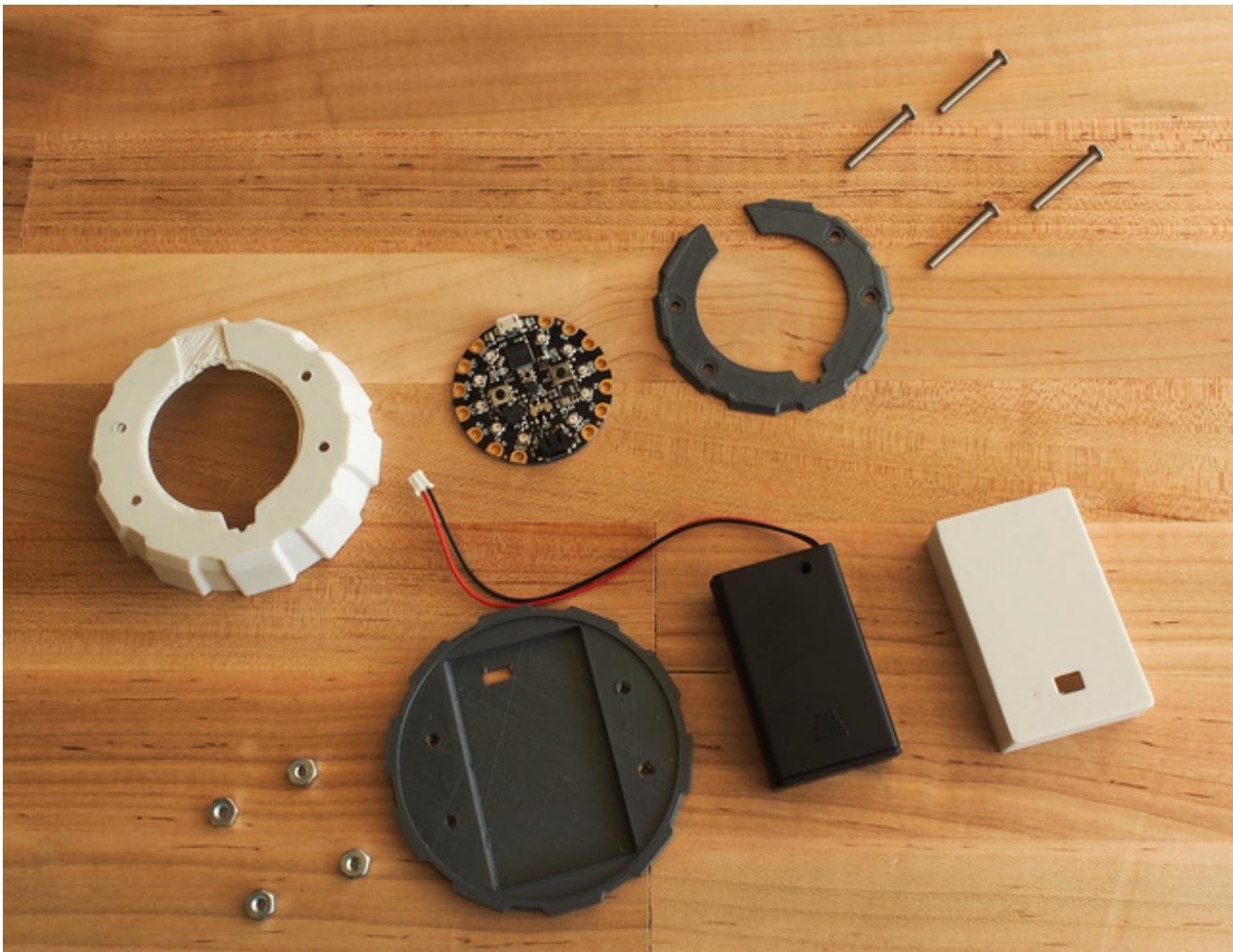
In the **Arduino IDE**, click **Sketch > Upload**.

The board will blink, and after a few second the new code will be uploaded to the board.

The board will blink the red #13 LED twice and beep when the sketch starts.

You'll build the enclosure in the next section.

Build the Enclosure



Making a Case

It's time to build a rugged-looking case for your Password Vault. While you can use your Password Vault without any additional hardware other than the Circuit Playground, it's fun to make a more finished case for regular use in the field.

3D Printed Parts

You'll print four parts for the case -- a base, battery box, middle, and top. You can download the **PasswordVault_files** from the link below. These can be printed with most

any material, such as PLA or ABS plastic. The case pictured here was printed with CPE (co-polyester) filament.

[PasswordVault_files.zip](#)

<http://adafru.it/rc2>

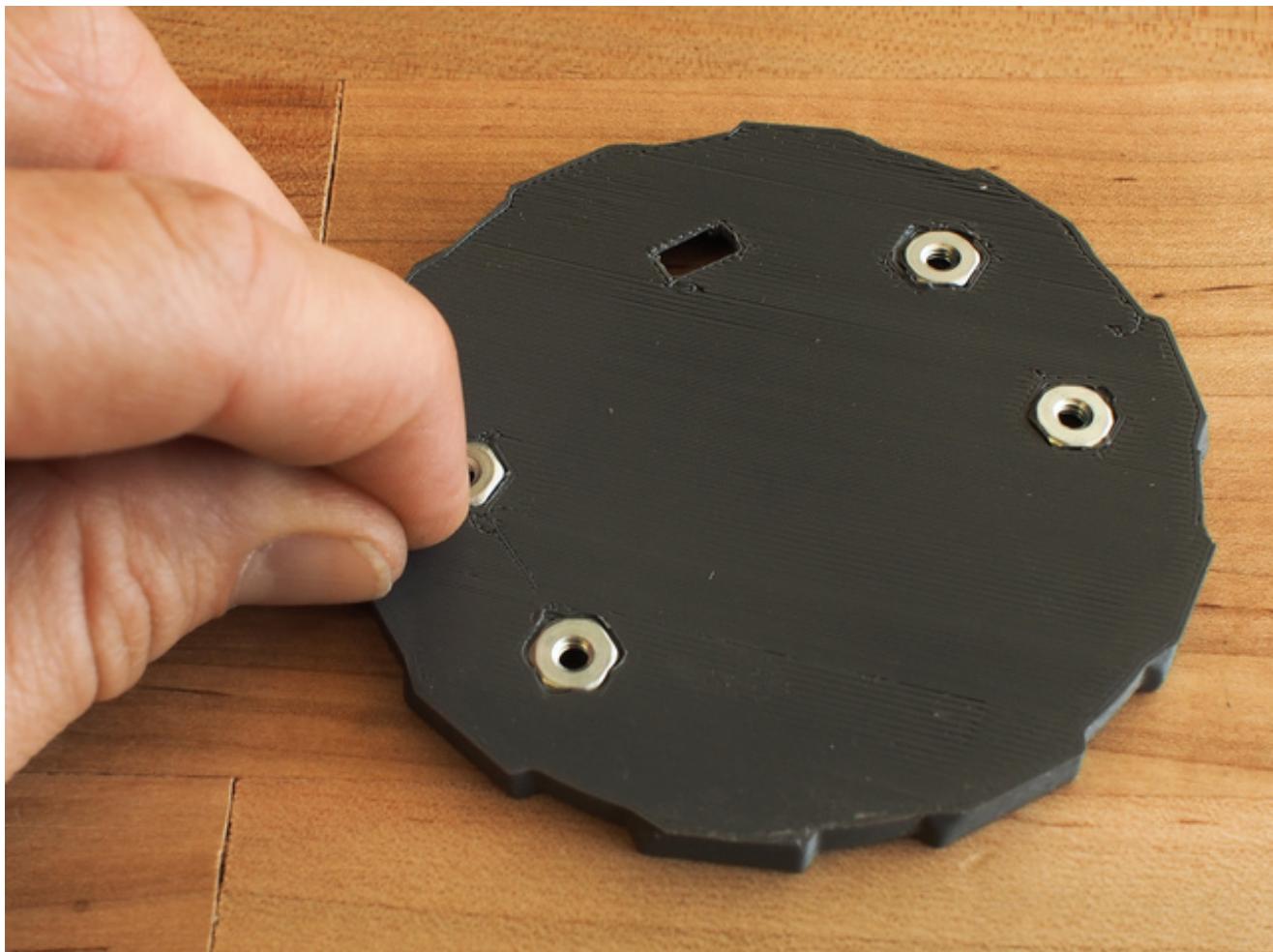
The units of the model files is millimeters. As a reality check when importing to your software or sending out to have printed online, the length of the battery box is 66.5mm.

Once printed, clean up the parts with a hobby knife to remove any excess brim material and clean up the screw and nut holes and the battery box hole.

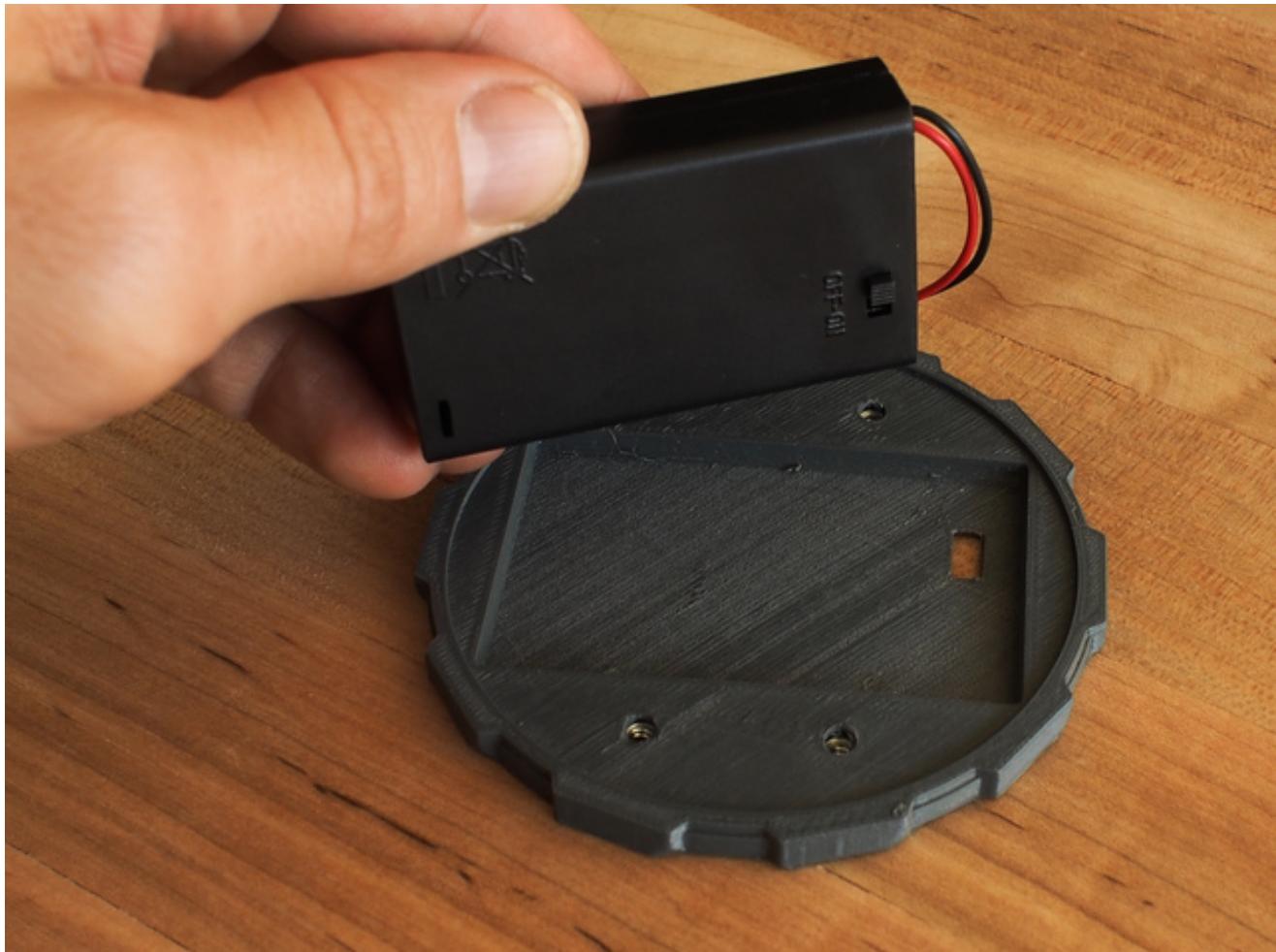


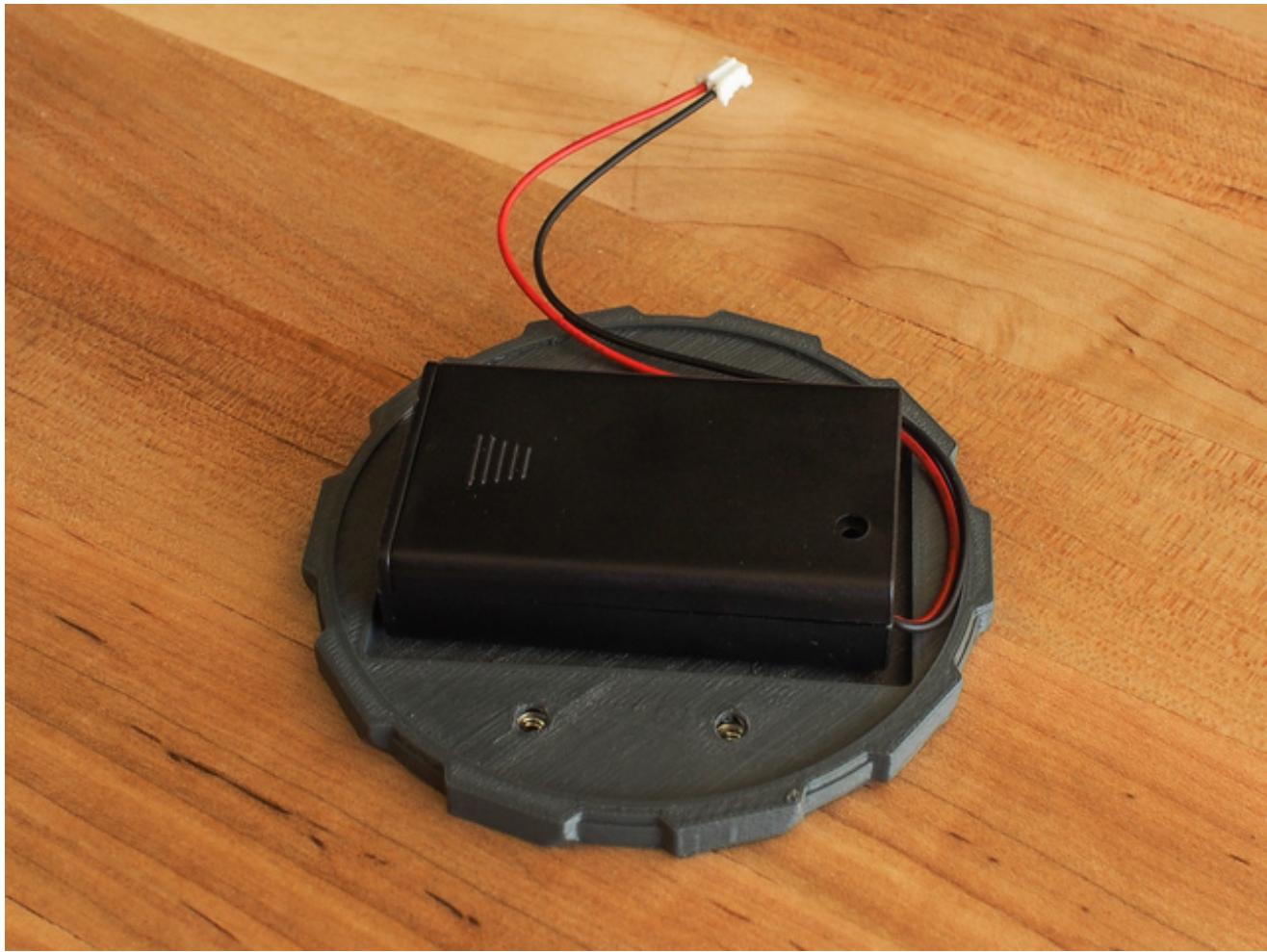
Assemble the Case

The case can be assembled from the bottom up. First, insert the hex nuts in the four cutouts in the 3D printed base bottom. You may need to press down to get them fully seated in place.

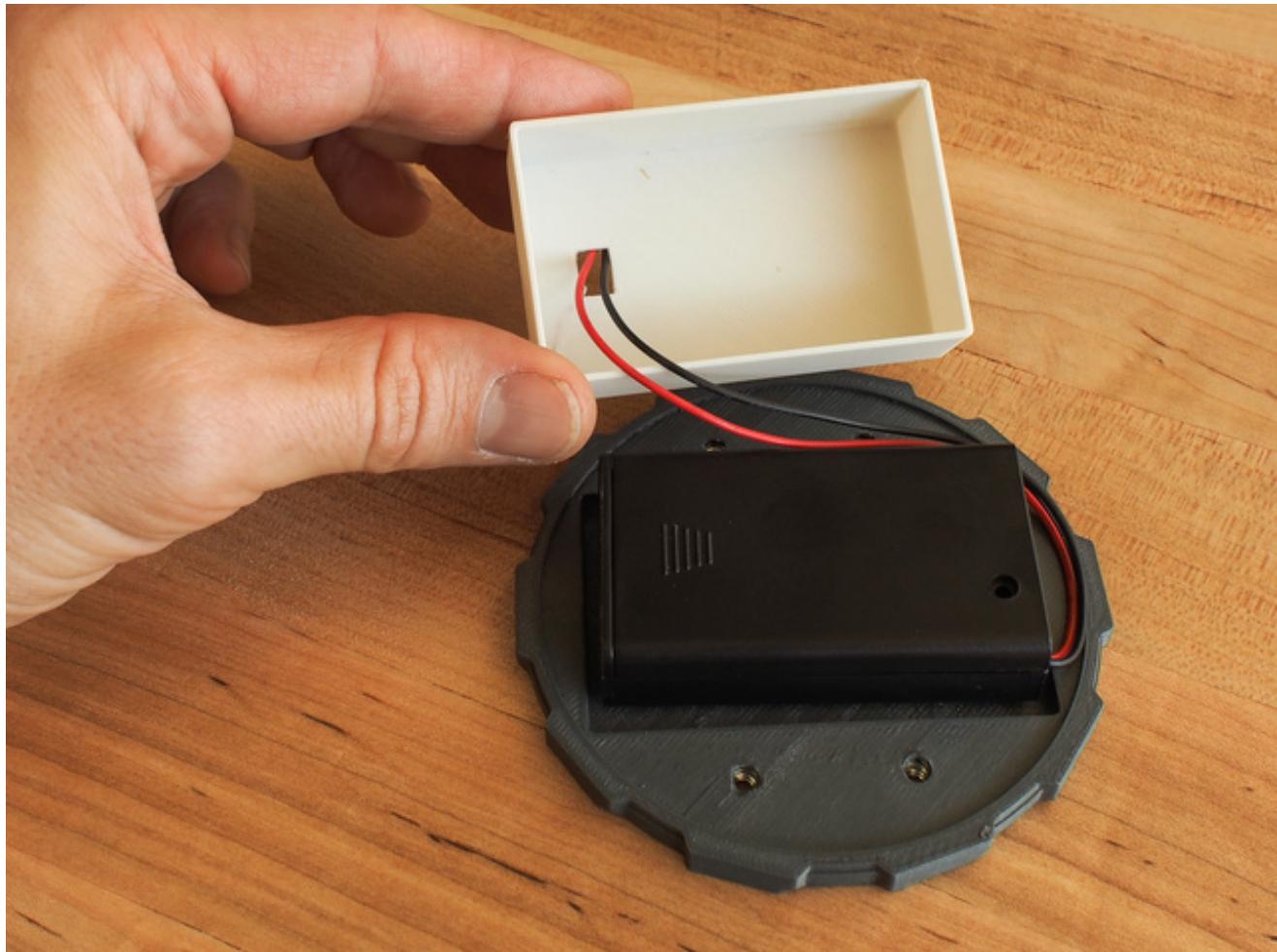


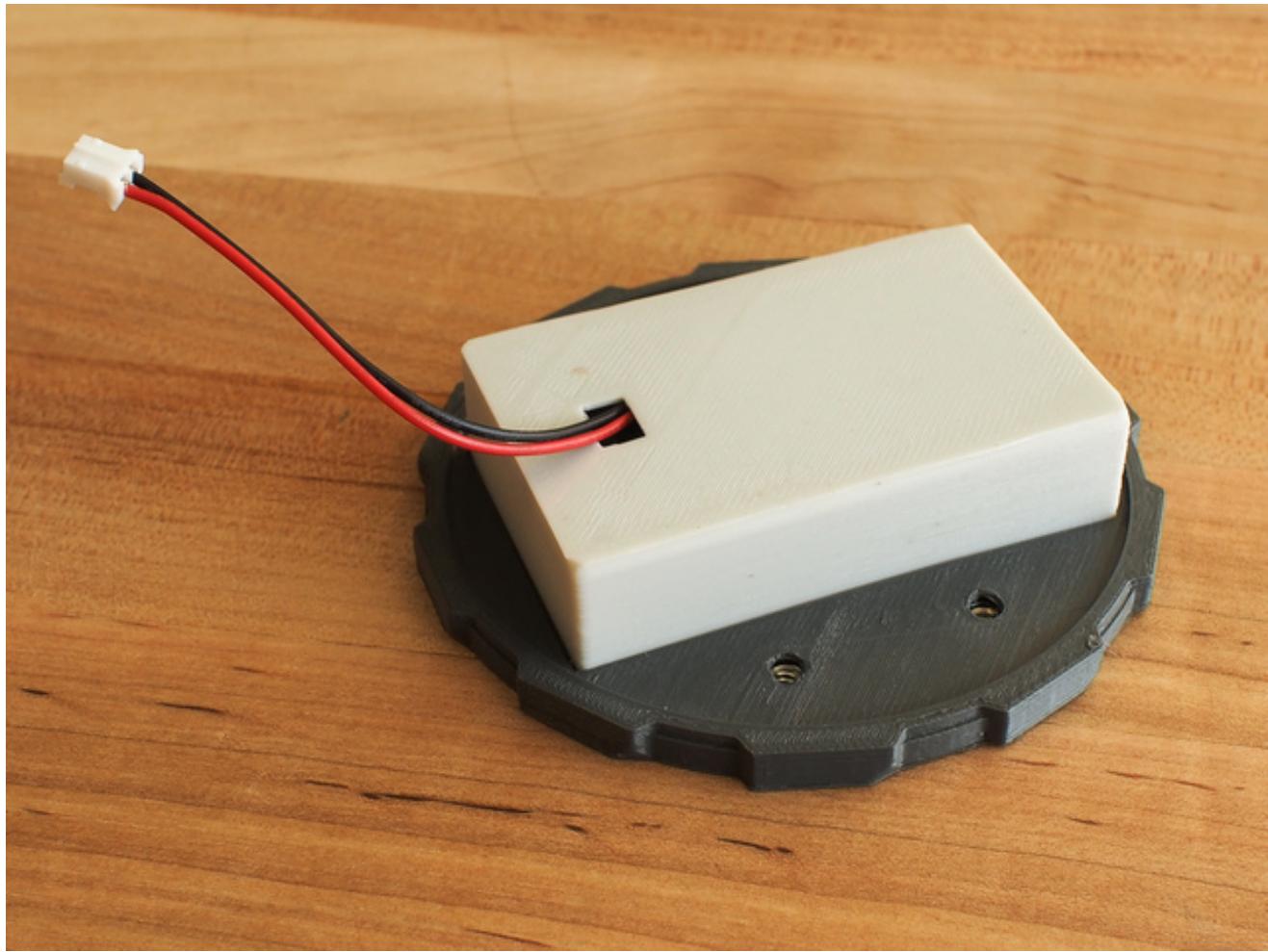
Next, put three AAA batteries in the switched battery case, close it, and then place it face down in the base so the switch is exposed from the bottom.



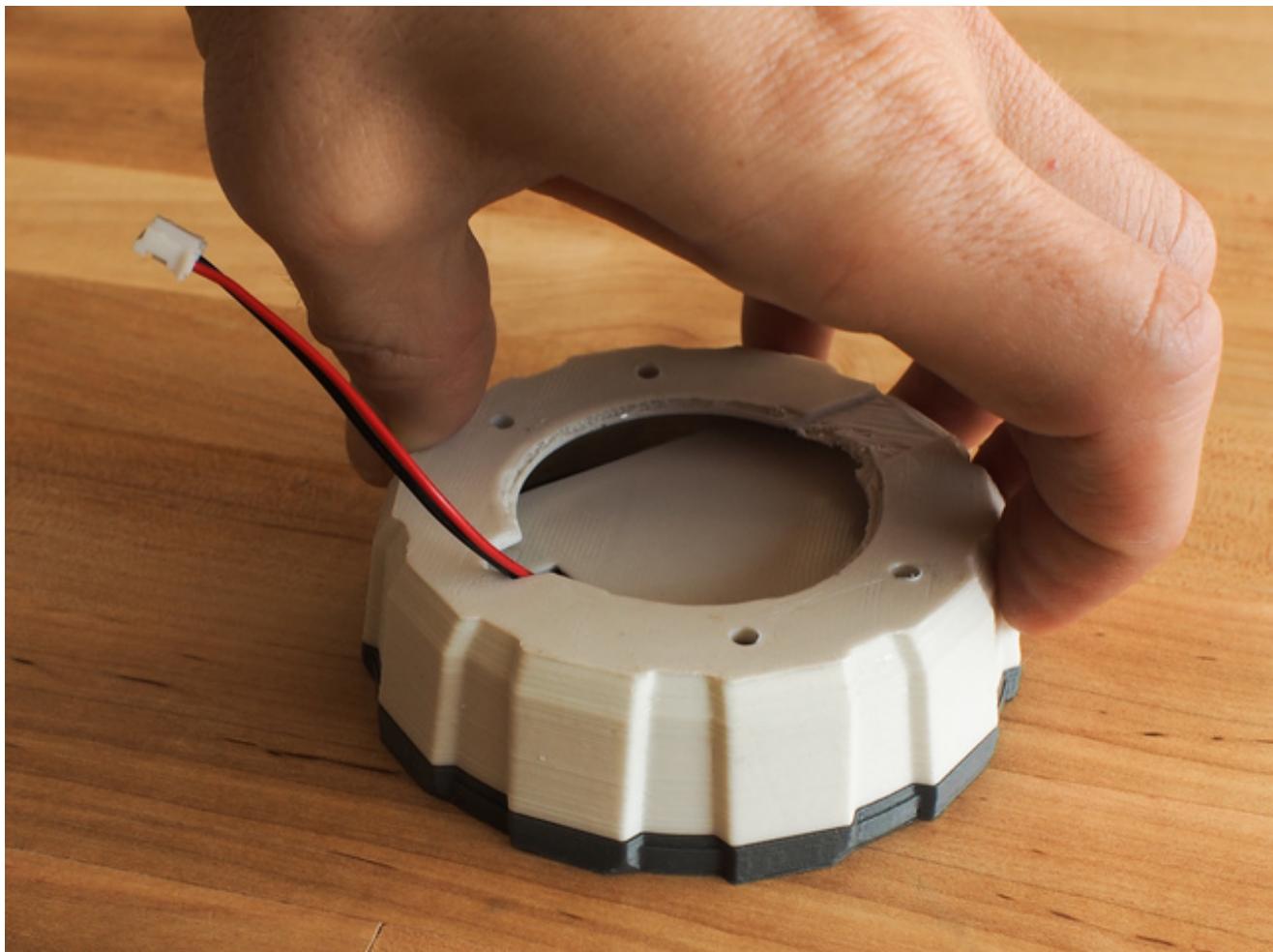


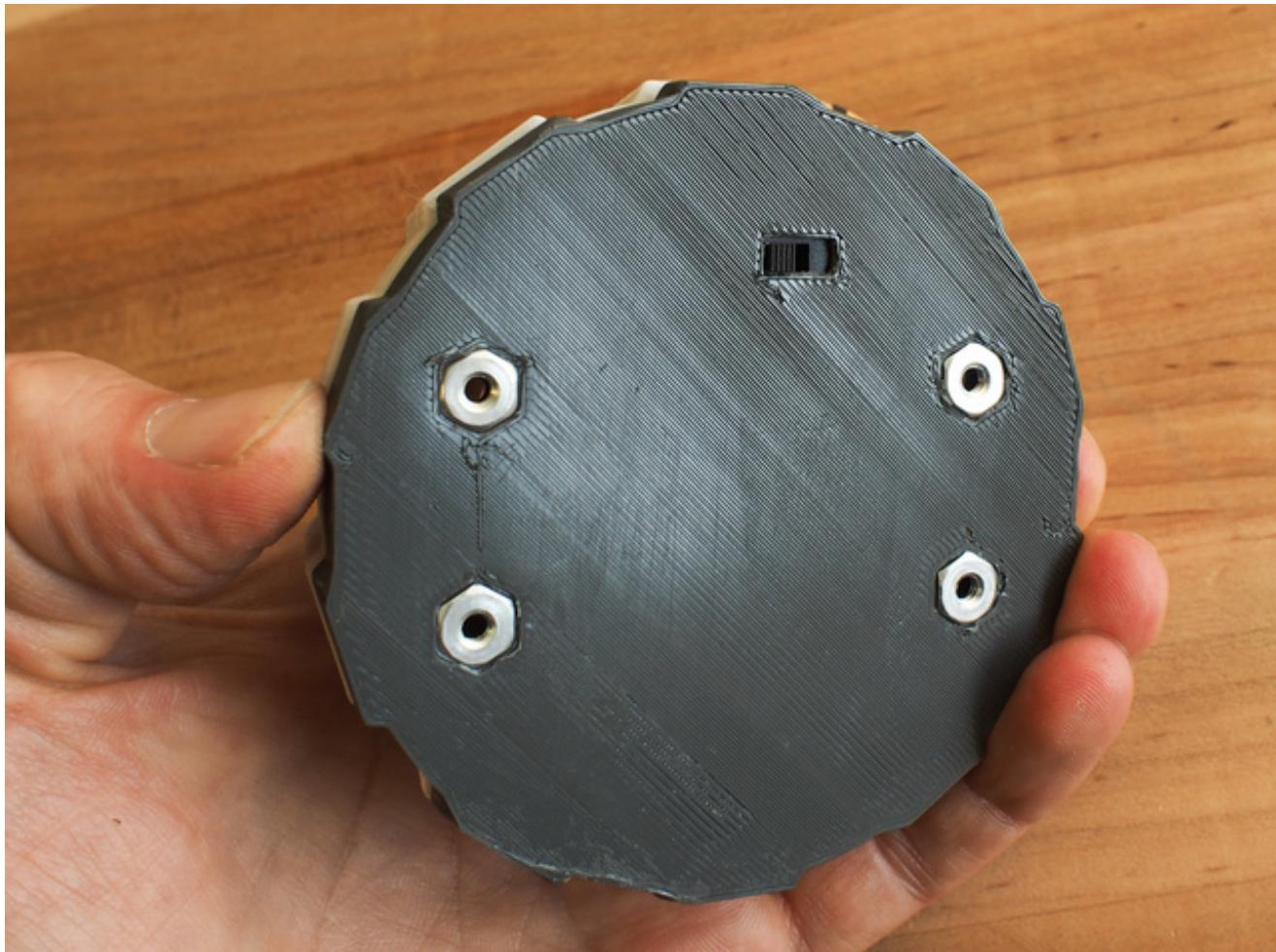
Feed the battery case's JST connector wire through the hole in the 3D printed battery box, then cover the battery case, snapping the box into place on the recessed base cutout.

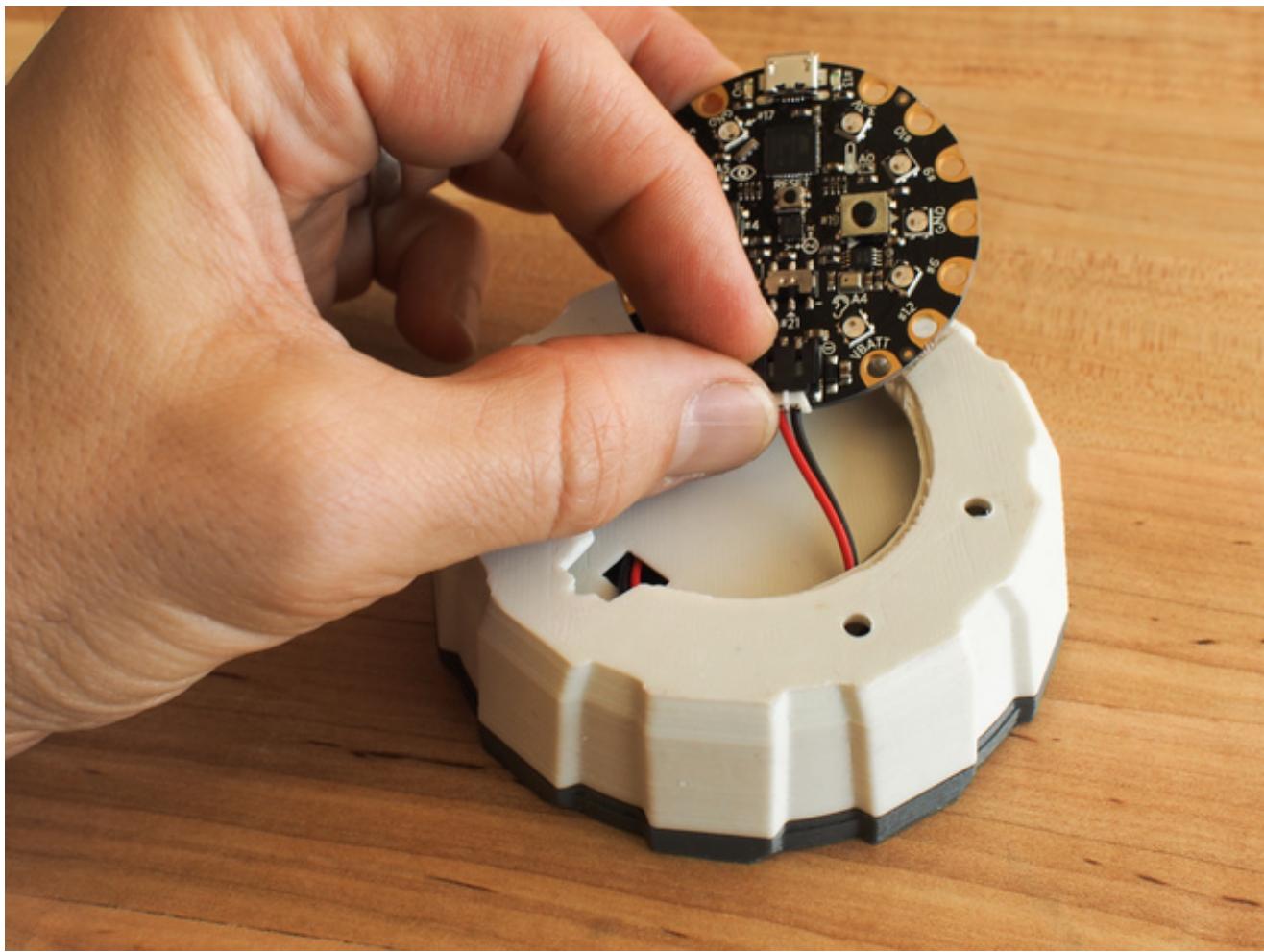




Place the 3D printed middle section over the base.







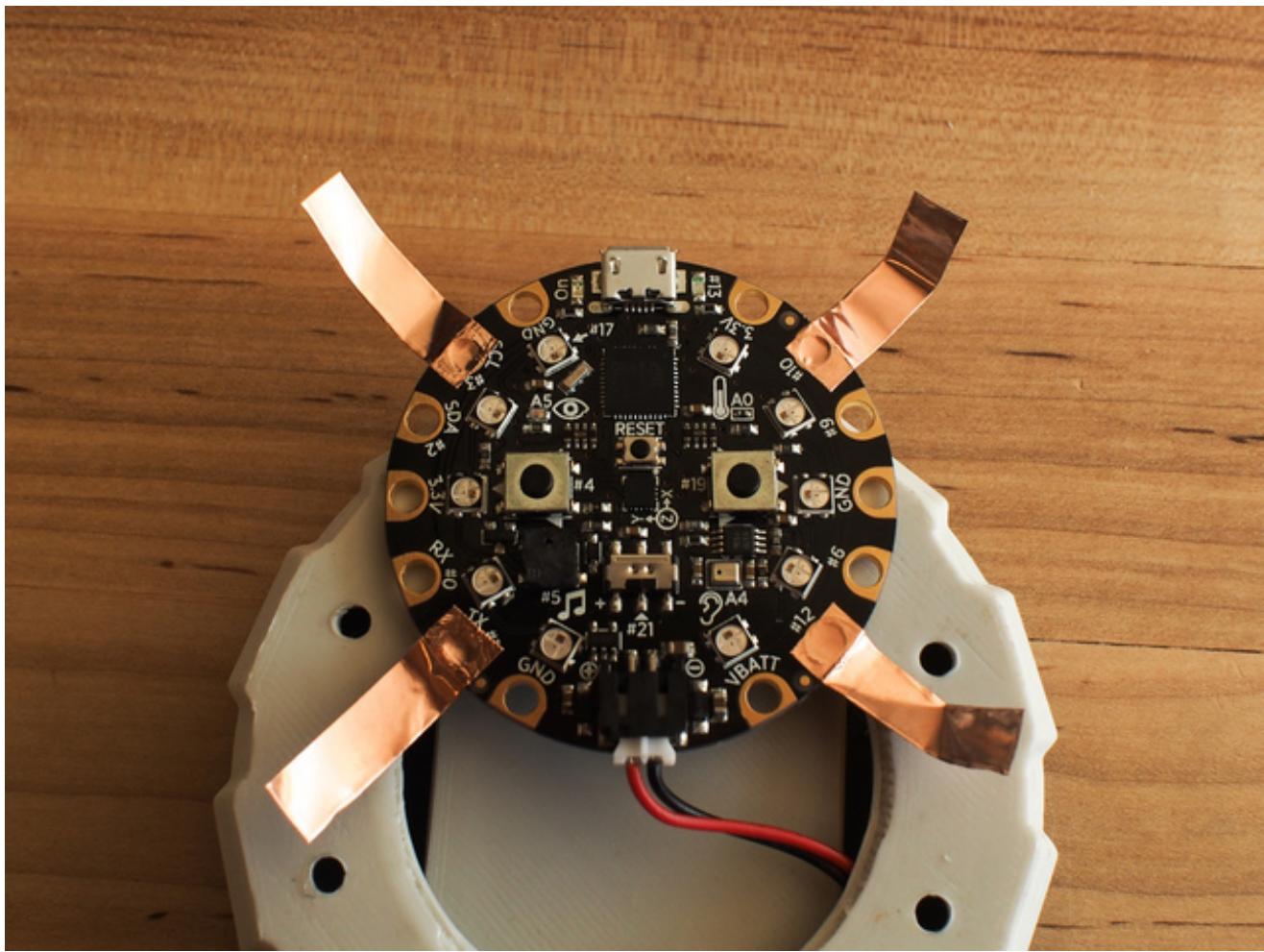
Plug the JST connector into the Circuit Playground and test that it turns on when the battery box switch is flipped. It's a good idea to test power switches and connections along the way as you are assembling enclosures.

Capacitive Switches

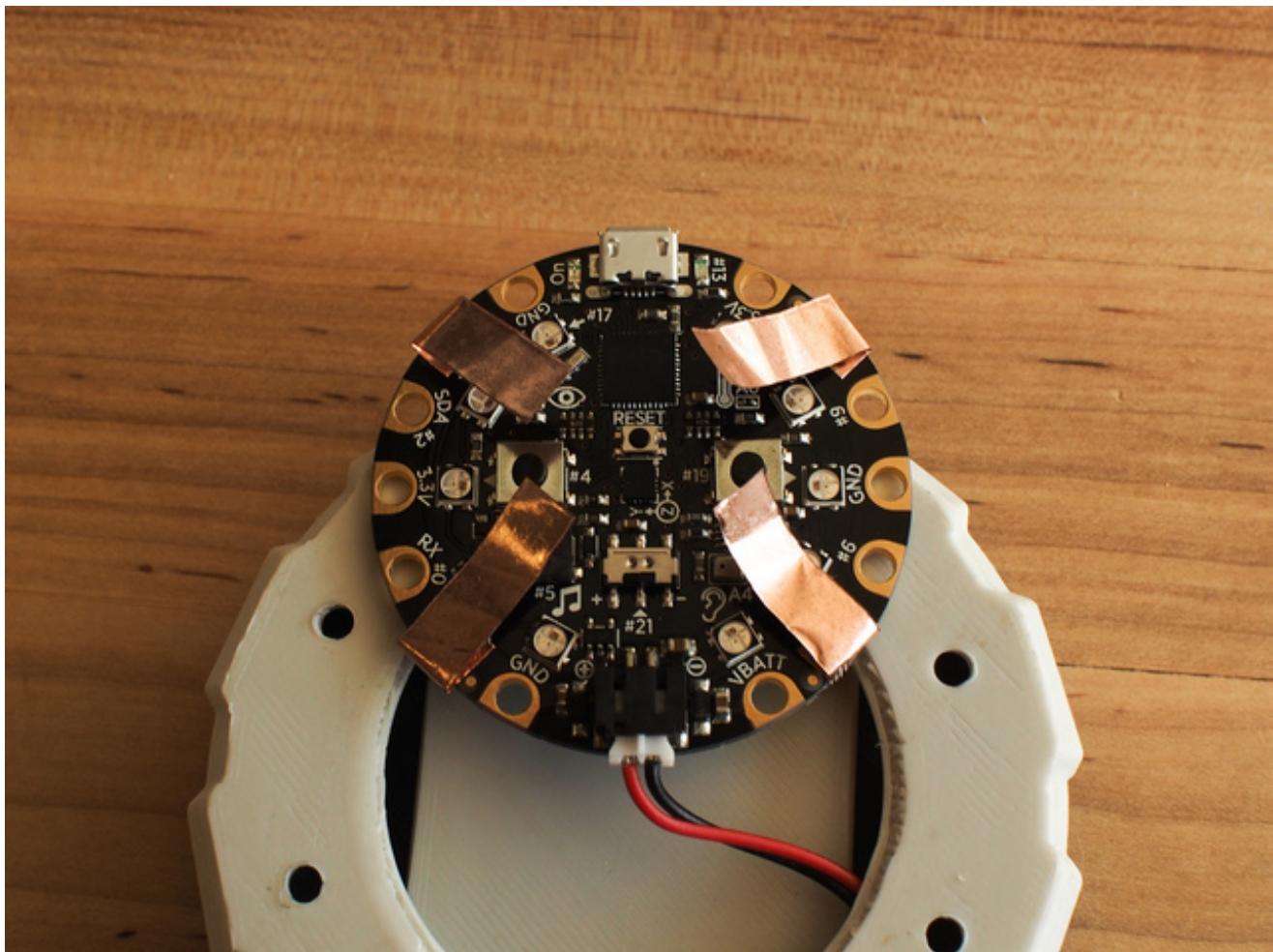
The capacitive pads on the Circuit Playground that you'll use to enter your unlock code will be easier to use when enclosed in the case if they are extended with copper tape. Cut four 1-1/2" long strips of the copper tape, then peel off their backings and press the ends to the top side of pads #1, #3, #10, #12. Be careful not to bridge any other components or connections, such as the small copper test points on the board.

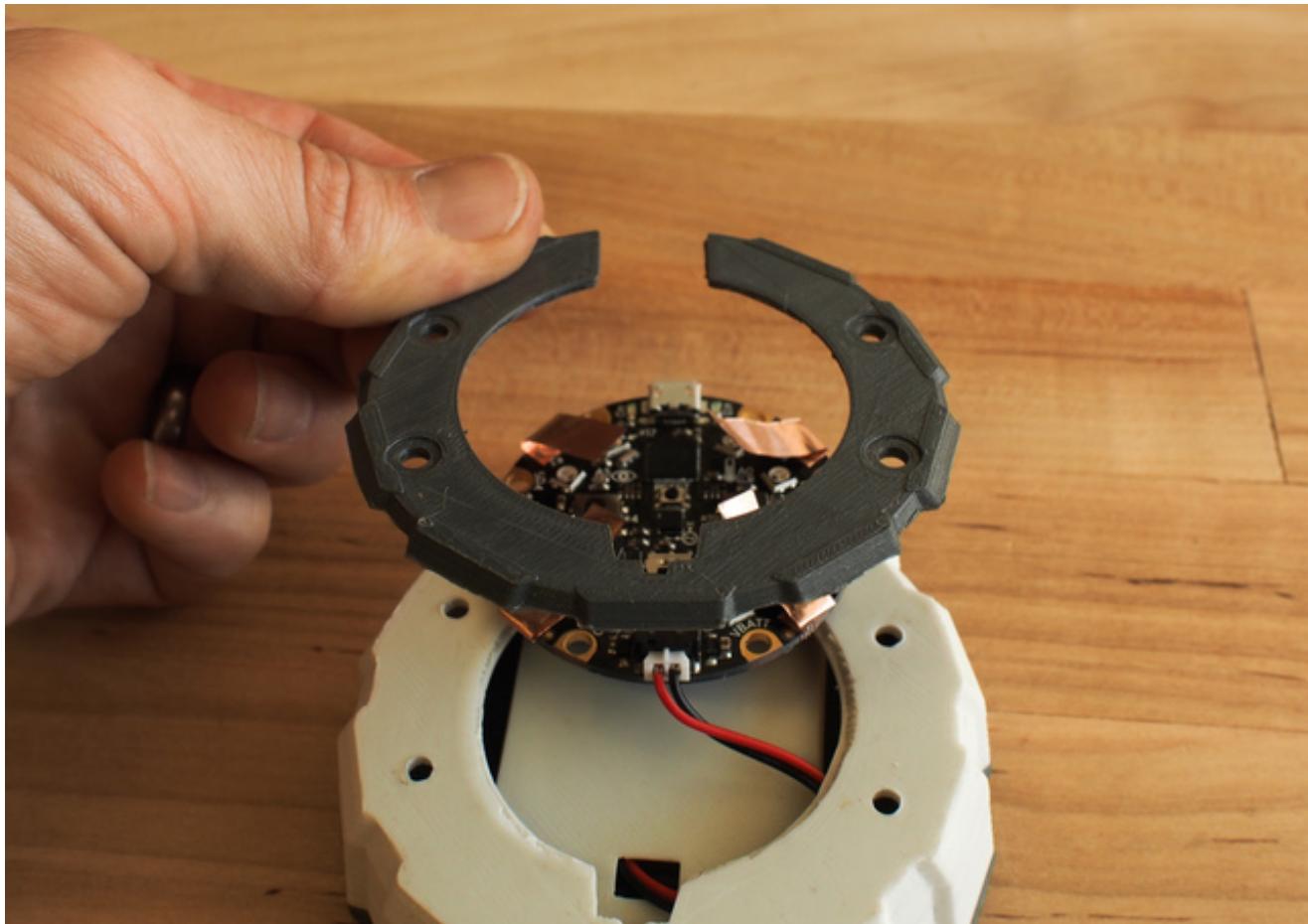
Adafruit's copper tape has conductive adhesive on the back, which is a requirement or the copper pads won't electrically extend the pads. Make sure if you source your own it has conductive glue!

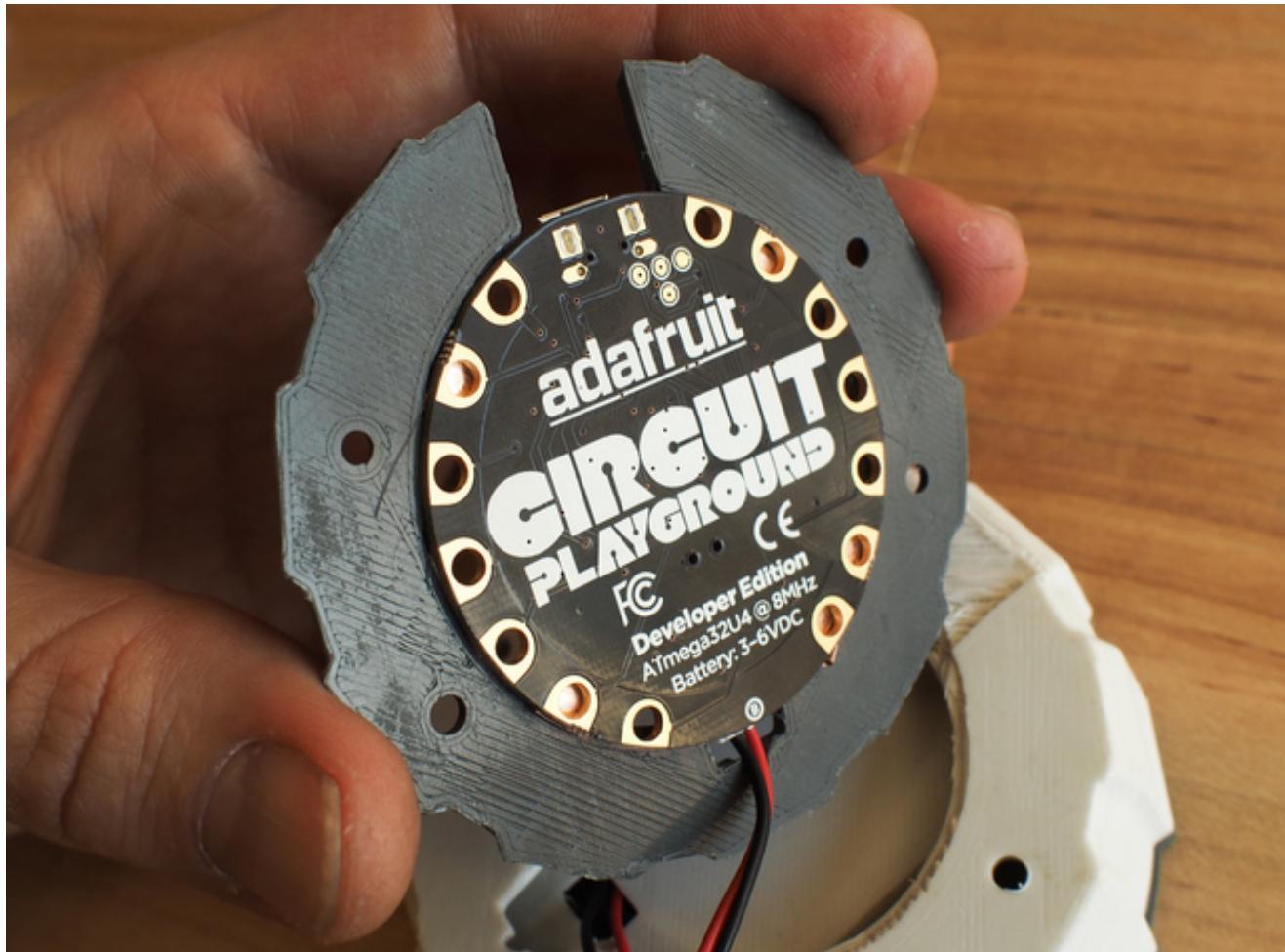


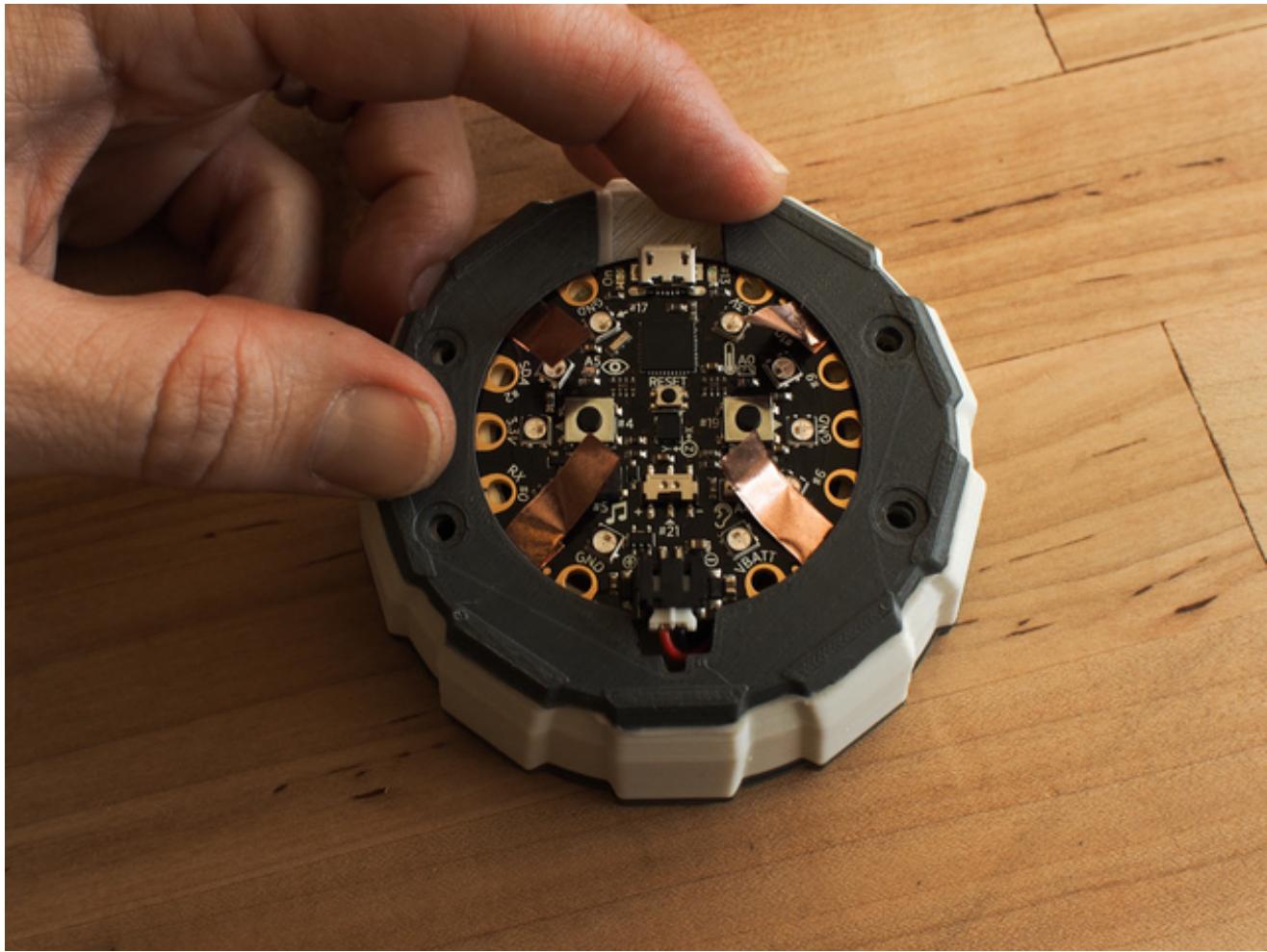


Fold the copper tape strips back over themselves toward the center, and then press the Circuit Playground into the 3D printed case top. It will fit snuggly inside the inner rim.

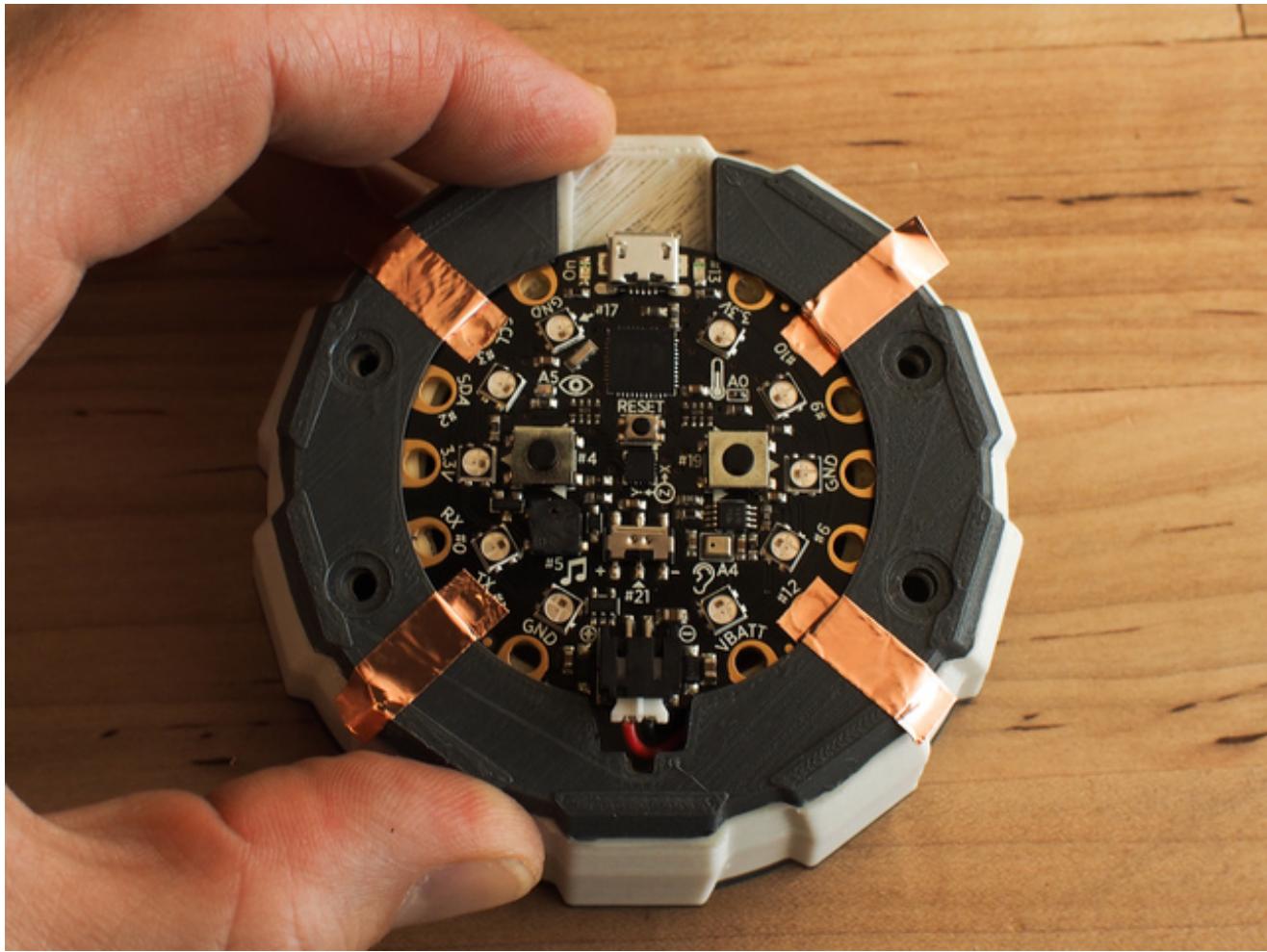






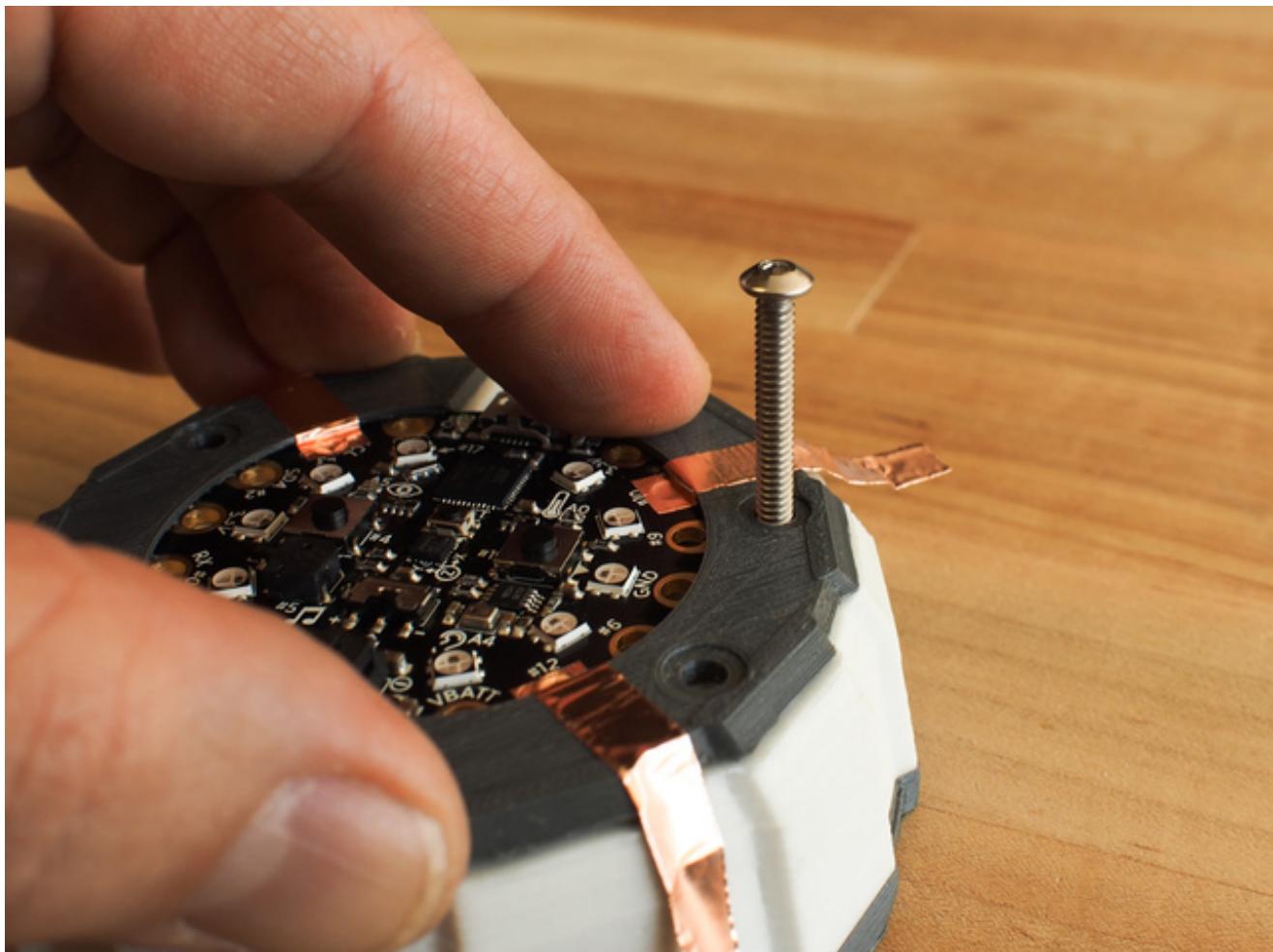


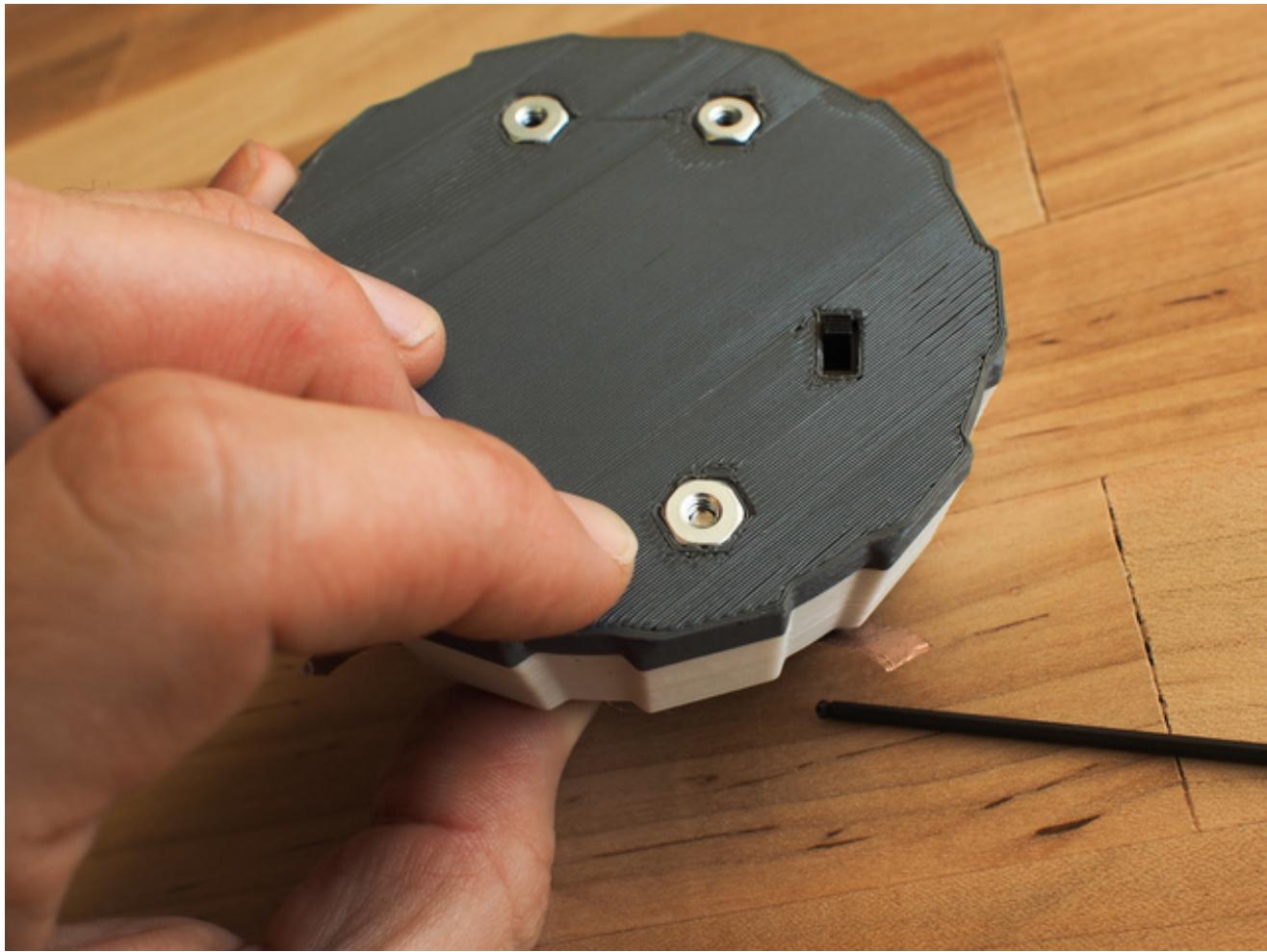
Now, fold the tape strips back over the top and stick them to the 3D printed case top. These will be the touchpads you'll use to key in your code entry.



Fasten

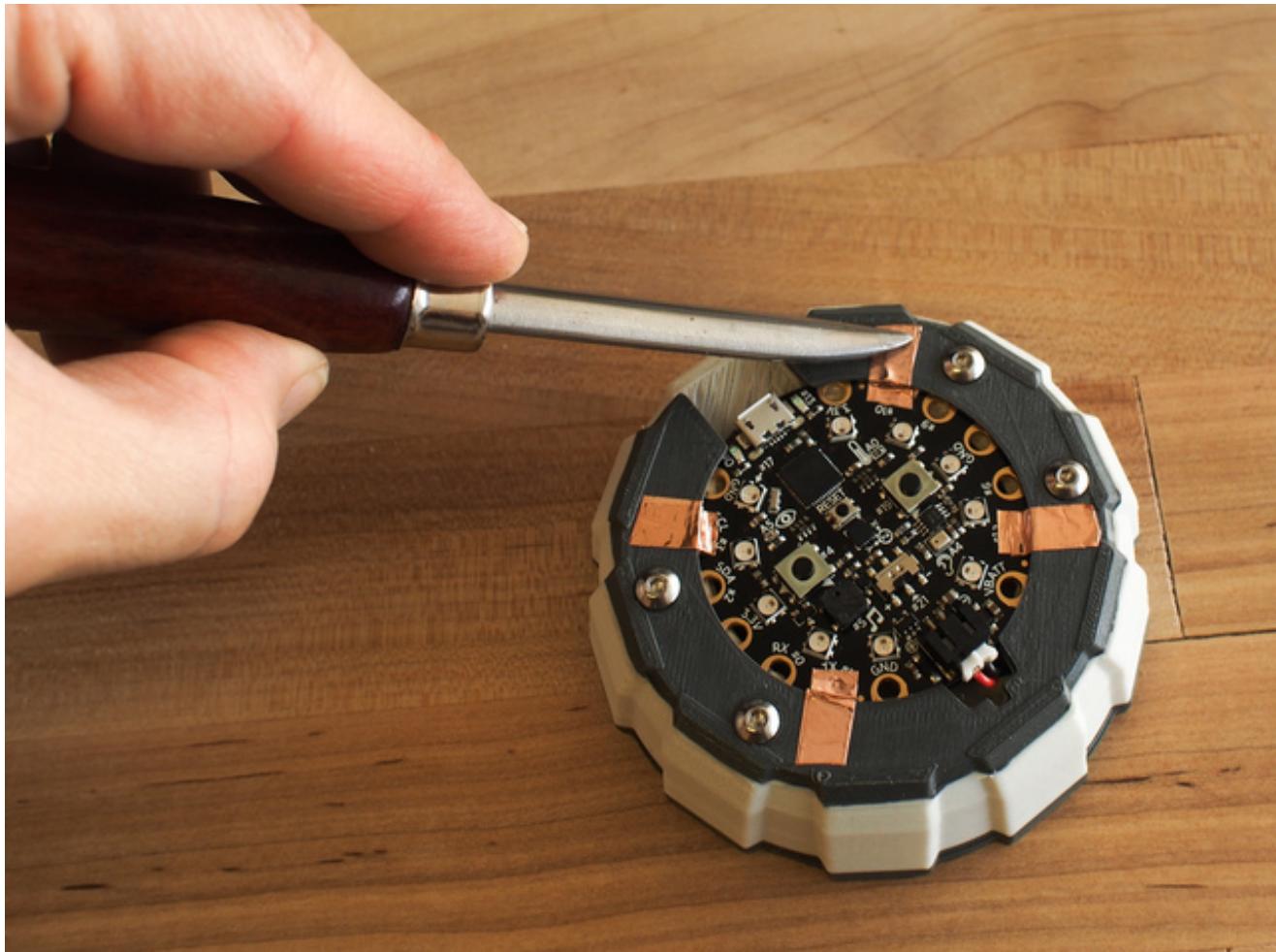
You can now insert and tighten the four screws until they are secured in the hex nuts underneath.

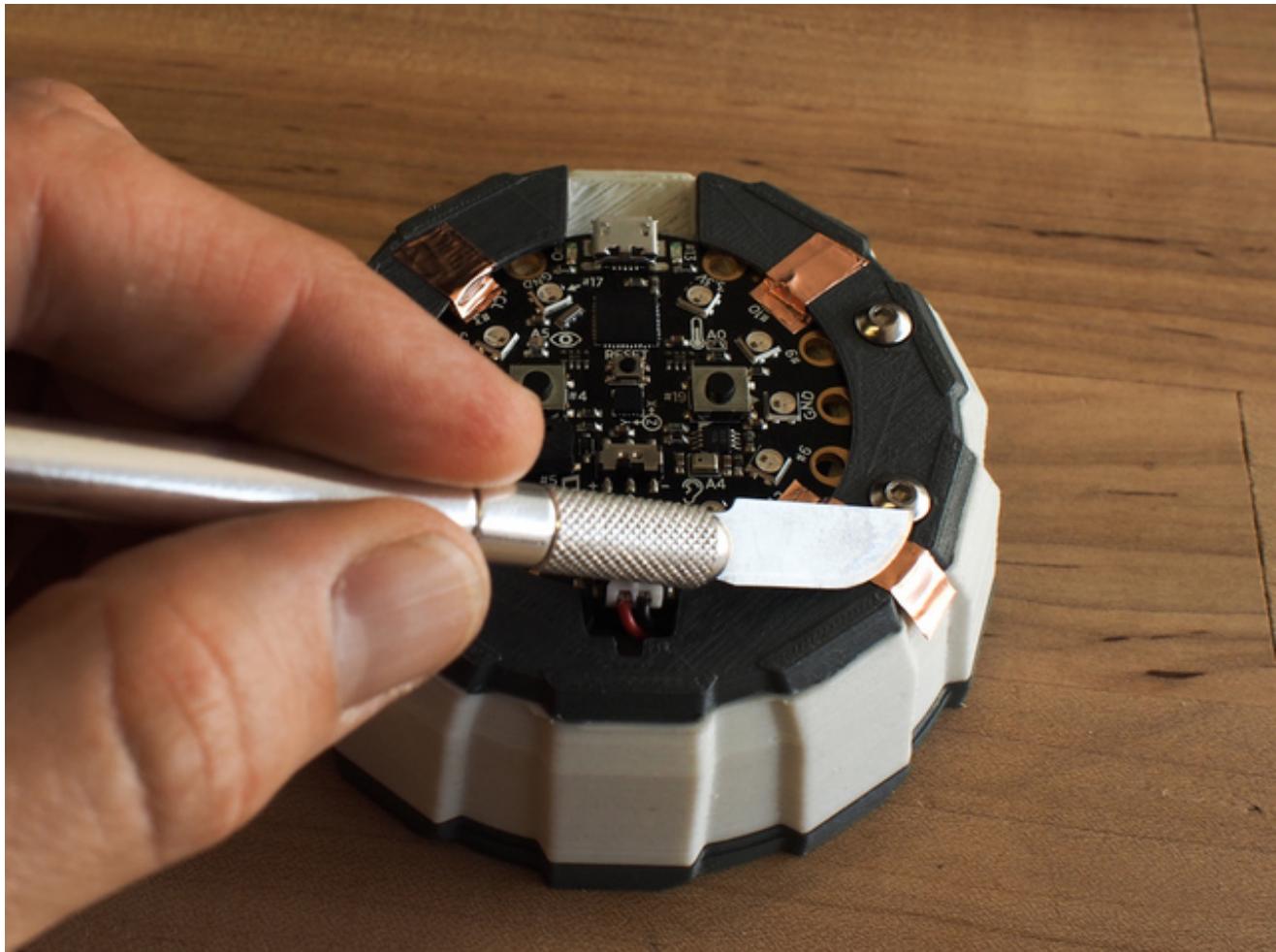


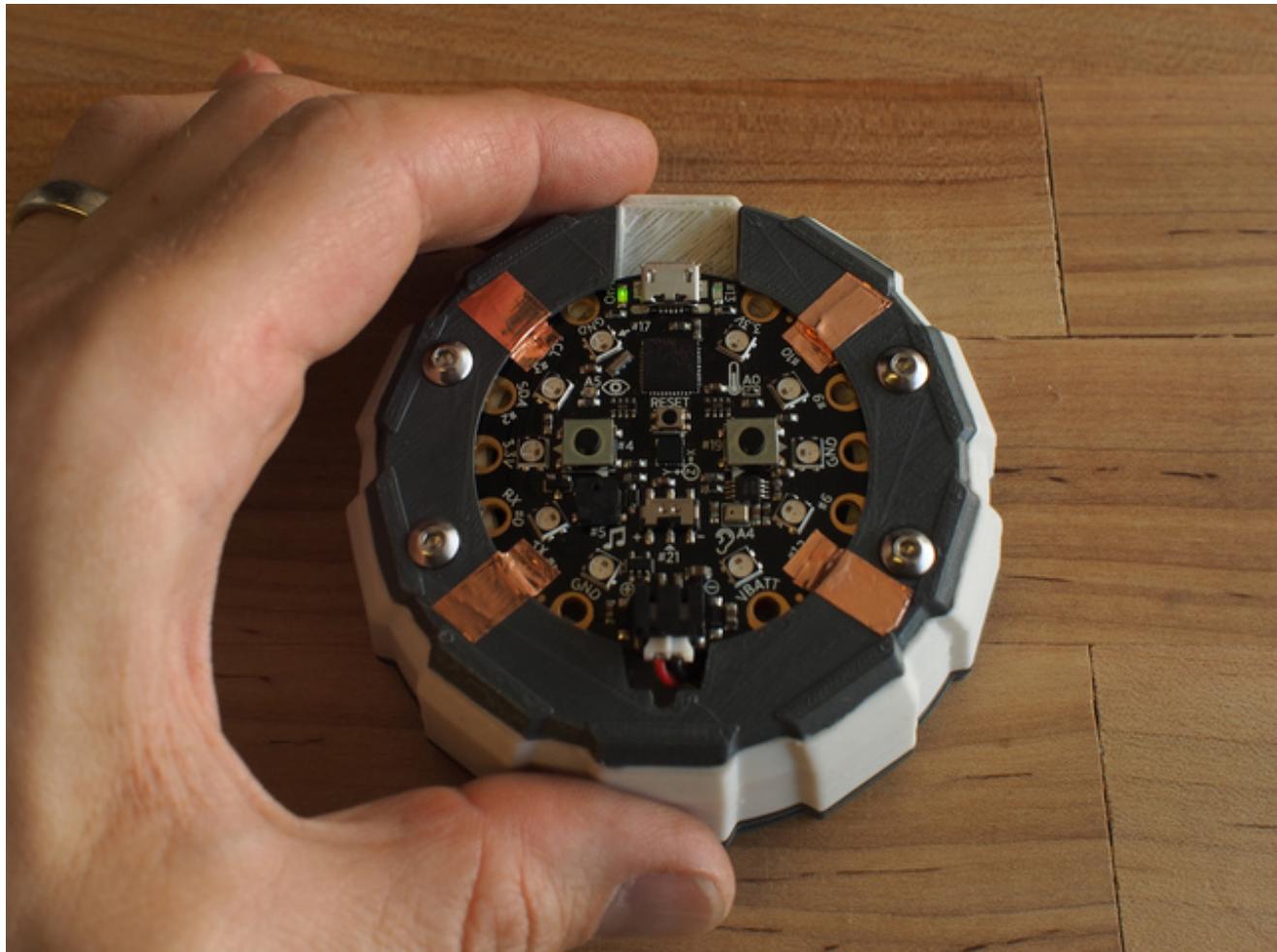


Finishing

Use a burnisher or your fingernail to smooth down the tape, then trim off the excess with a hobby knife.







Next, you'll use the Password Vault to enter a password on your computer.

Use the Password Vault

Startup and Unlock

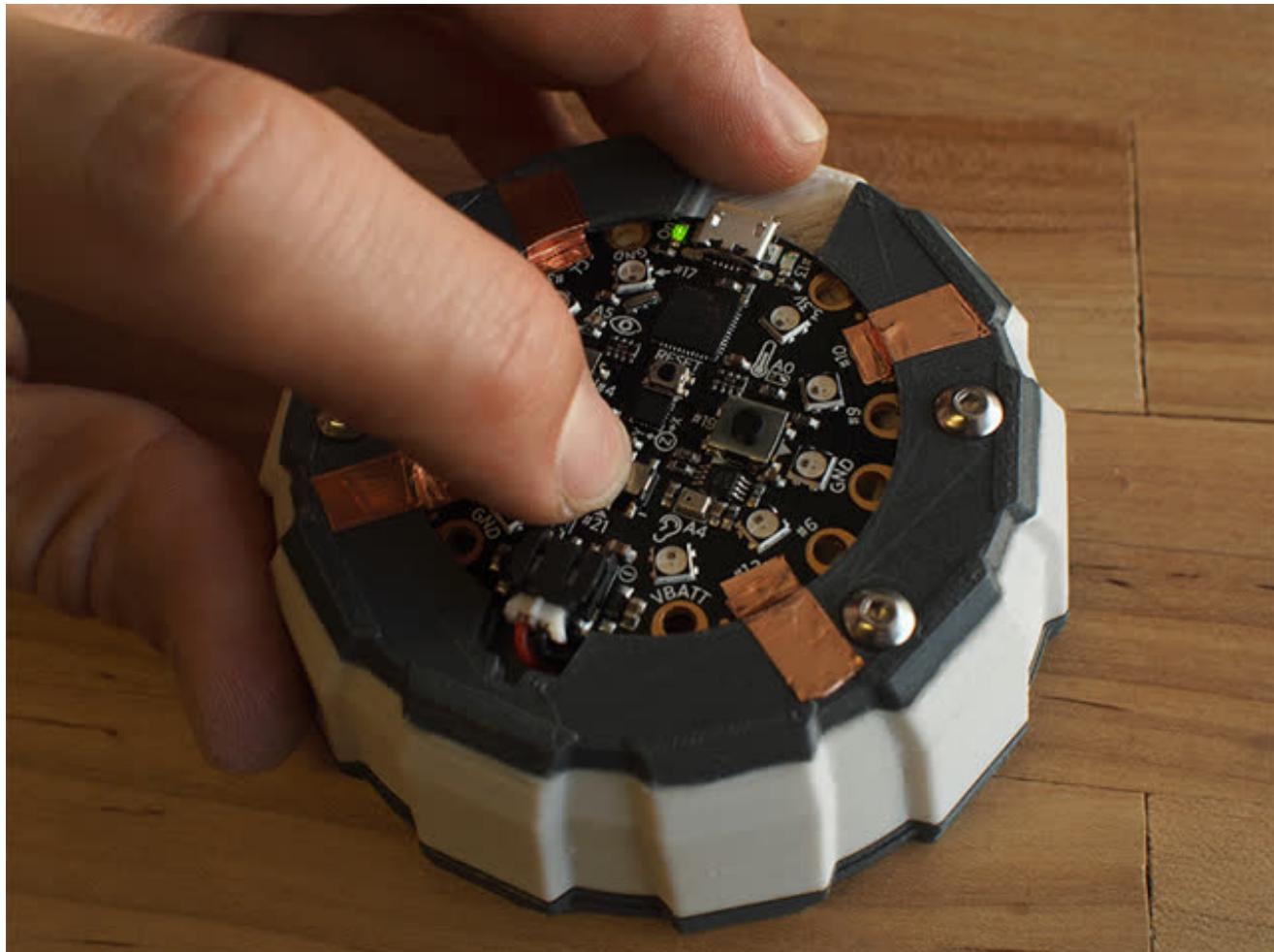
To use your Password Vault, first turn it on by flipping the battery switch under the base. It will beep and flash the red LED twice to indicate that it is on and ready.

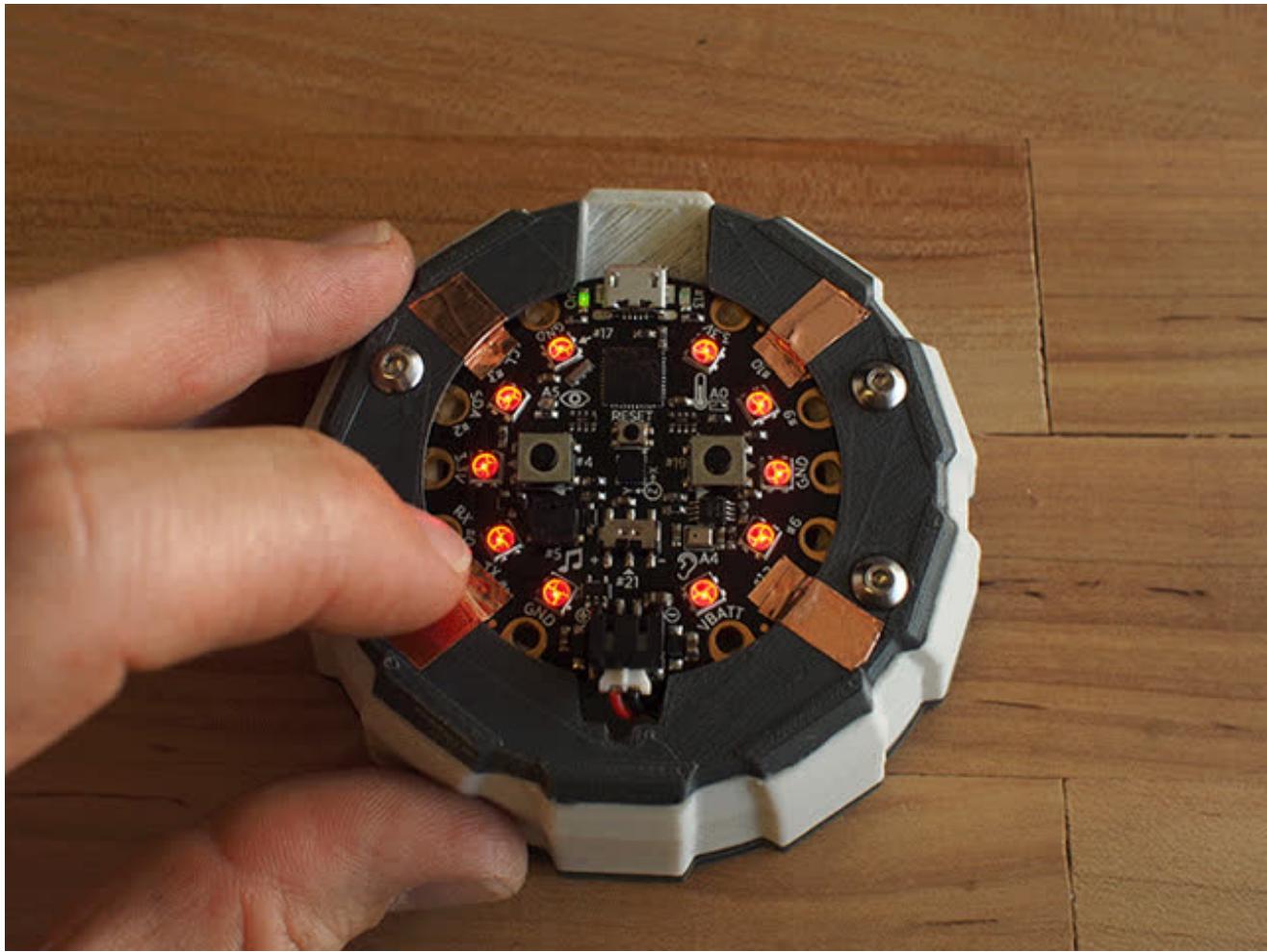
When you're ready to enter your unlock cod, flip the slide switch on the Circuit Playground. The ring of NeoPixels will light up red.

Tap your secret, six-digit access sequence on the four copper tape pads.

As you tap each pad, a NeoPixel will turn yellow in sequence as feedback. When all six have been pressed the speaker will buzz with either a low tone to indicate an incorrect code entry, or a higher pitched double beep if correct. The NeoPixels will light red if failed, or green if sucessful.

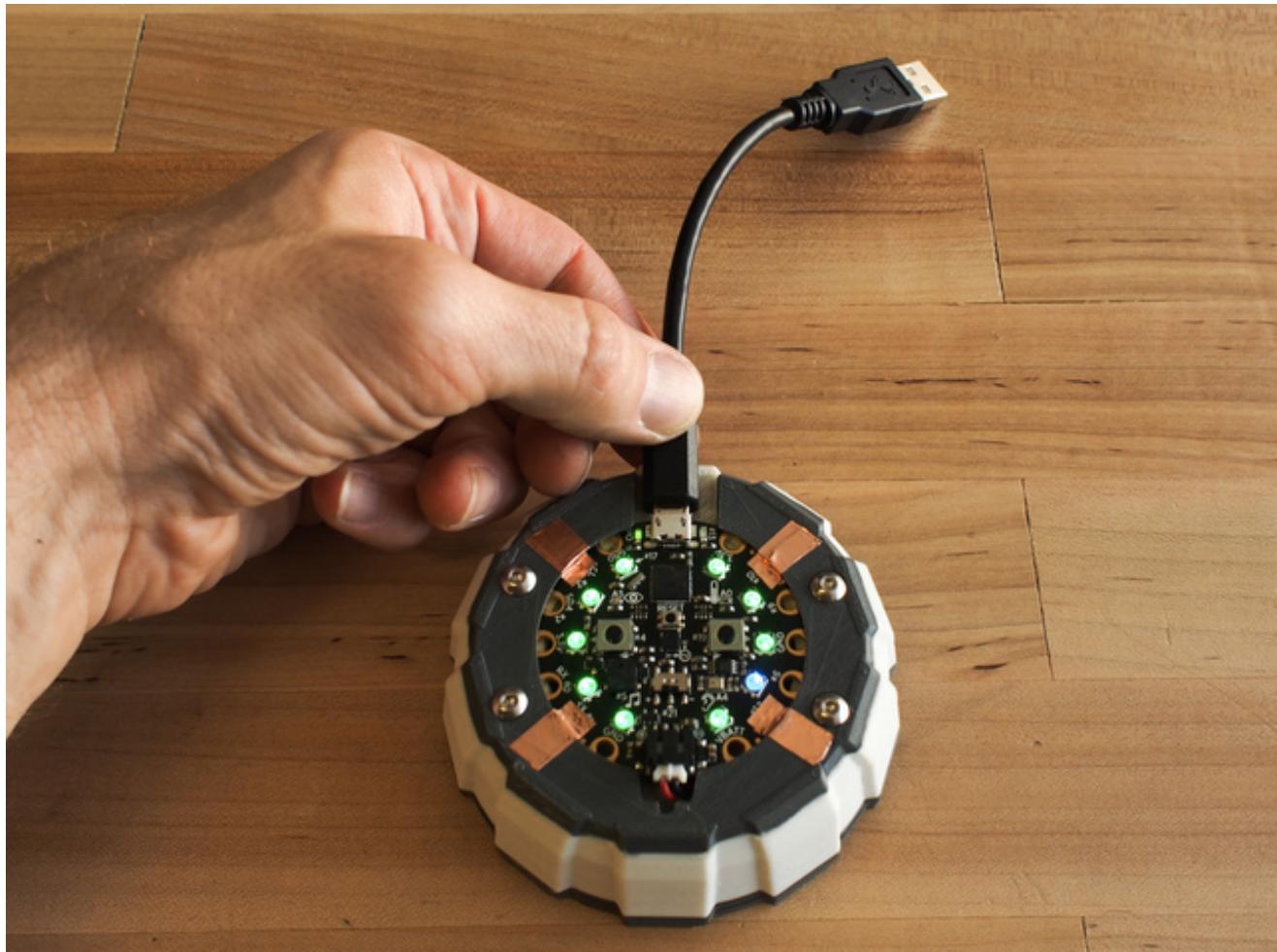
If you've entered the wrong code you can simply try again -- there is no built in self-destruct option! Once you've entered the correct code the Password Vault is unlocked, and the NeoPixels will light up green.





Password Retrieval

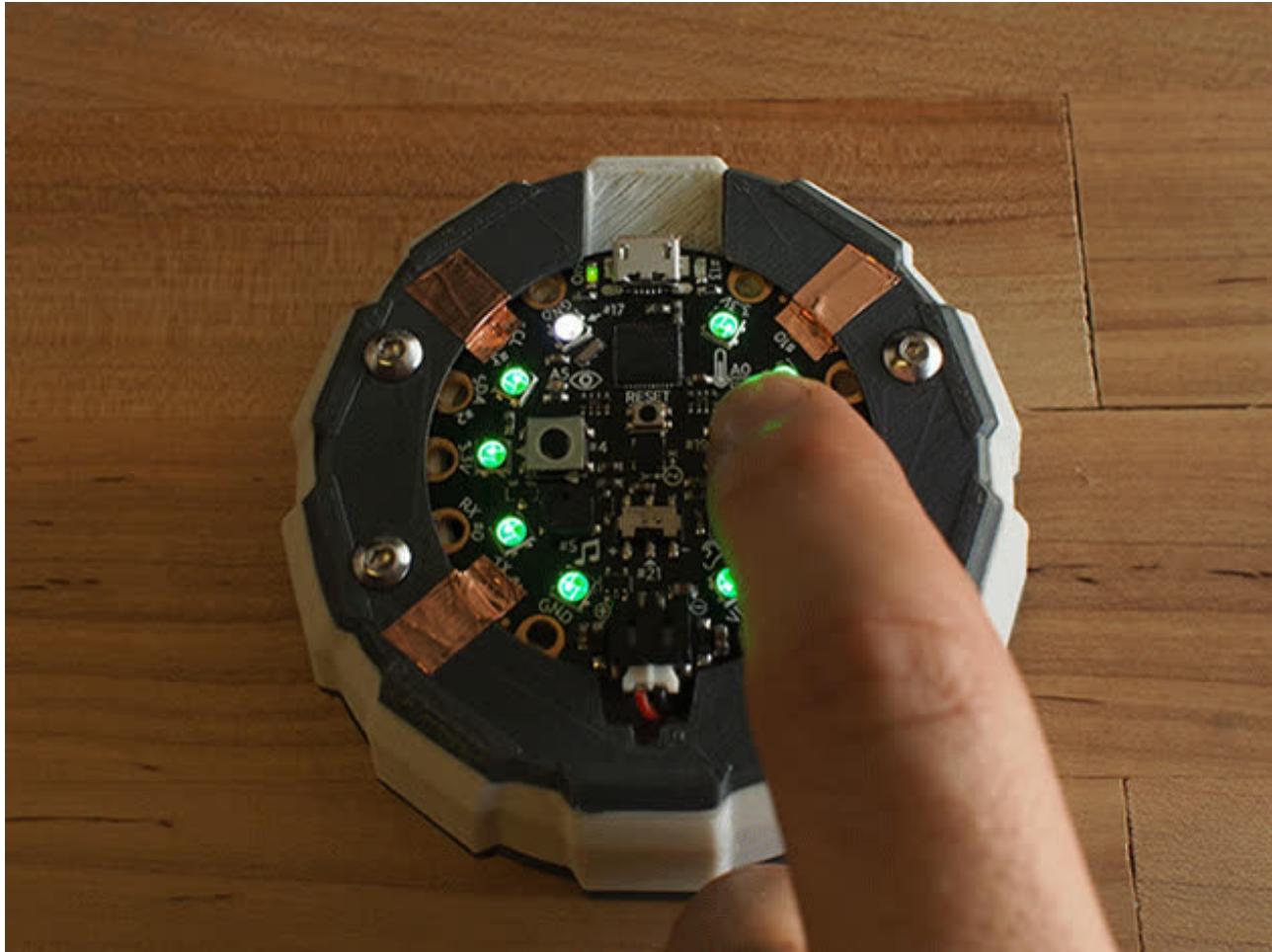
Plug the Password Vault into any computer that needs one of your passwords (this could be any password including the main login, a website, or a game, for example). There is no need to install any drivers -- Circuit Playground is automatically recognized as a USB keyboard.





Since you already unlocked the Password Vault, you may now select which of your ten passwords to use. By default, the first slot is picked -- you can tell by the position of the white NeoPixel.

Press the right button on the Circuit Playground to choose the next slot, cycling through the ten positions.



On the computer, click the mouse into the password field you want to fill -- this could be the login password, a web password, anything really.

You aren't limited to storing passwords in the Password Vault -- anything you'd like to type can be entered. You could, for example, store and retrieve favorite movie quotes, addresses, or even use it as a quick way to paste the Declaration of Independence every time you need it!





Now, press the left button on the Circuit Playground Password Vault and the password will be typed in for you! A dual tone will play on the speaker and the NeoPixel for that password slot will turn blue to indicate that it has been entered.

Your password has been entered, you're in!