Trinket React Counter

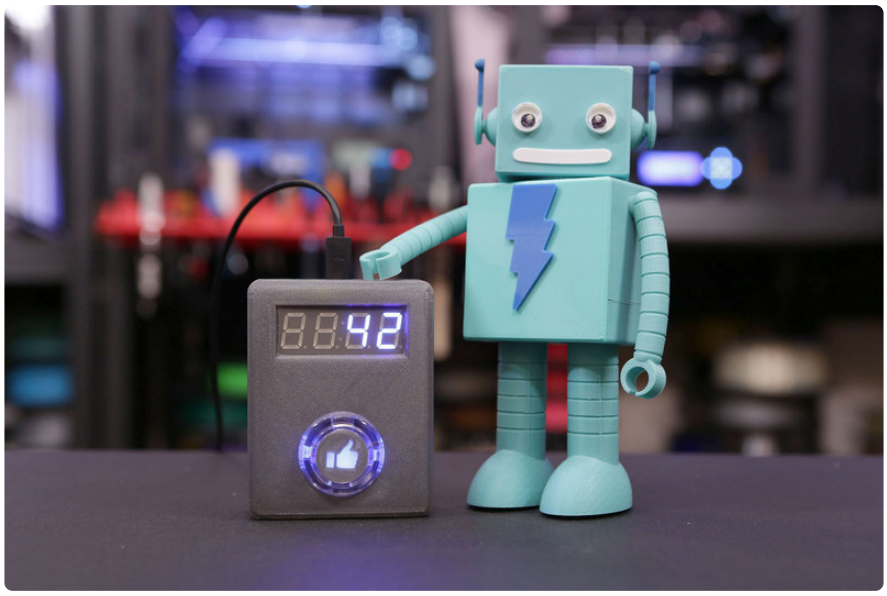
Created by Ruiz Brothers

https://learn.adafruit.com/trinket-react-counter

Last updated on 2024-06-03 01:53:12 PM EDT

Table of Contents

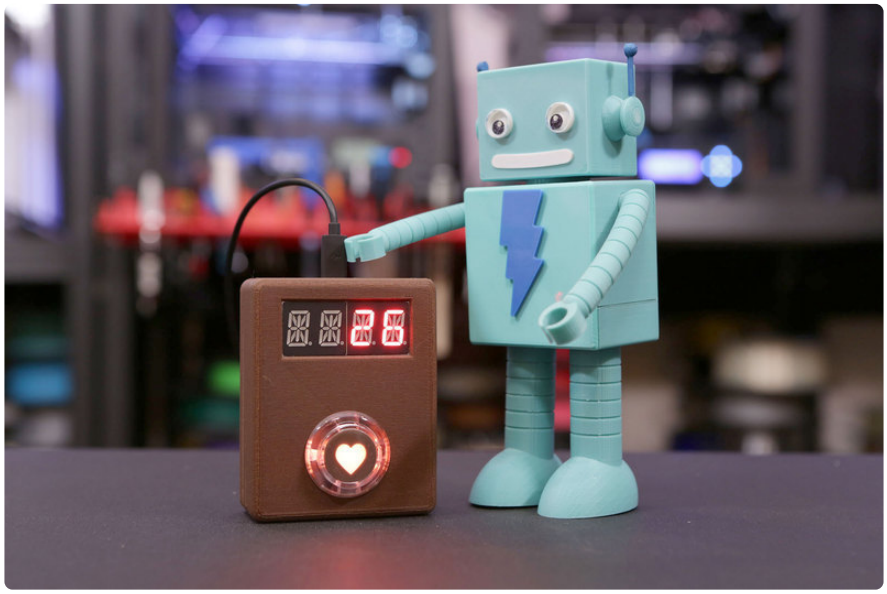
Overview



"Physical" Like Button

Inspired by Facebook "Reactions". You press the button, and the number count gets displayed on the LED screen.

Looks like a novelty product, but this can be used for all sorts of useful stuff like keeping track of attendees at an event, keep track of a score for a game or anything else that needs a tally counter.



Customize it, Make it!

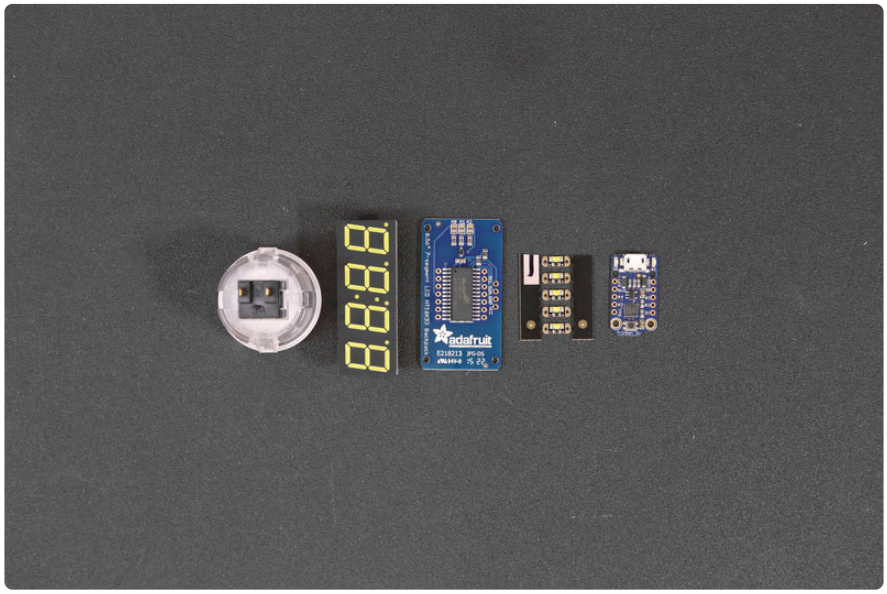
The really cool thing about this is that you can build your own with a just few components and a 3D printed case.

To make your own, you'll need an Adafruit Trinket, a 7-segment LED display and an Arcade button.

Prerequisite Guides

Please take a moment to walk through the following guides to get a basic understanding of the components used in this project.

- Introducing Adafruit Trinket (https://adafru.it/dhx)
- Adafruit LED Backpacks (https://adafru.it/m7f)



Parts

You'll need the parts listed below to complete this project.

- 5V/3.3V Adafruit Trinket (https://adafru.it/m7A)
- 0.56" 7-Segment Display (http://adafru.it/879)
- 30mm Arcade Button (http://adafru.it/471)
- Adafruit LED Sequin (http://adafru.it/1758)

Tools & Supplies

You'll also need the following tools and supplies. If you don't have access to a 3D printer, you can have a local maker ship the parts to you via 3DHubs.com (https://adafru.it/edO)

- Soldering Iron w/ Solder (https://adafru.it/doU)
- 3D Printer (https://adafru.it/diH) + Filament (http://adafru.it/2080)
- 26AWG Silicone Coated Wires (http://adafru.it/1970)
- 4x #4-40 3/8 machine screws
- 4x #2-56 3/8 machine screws
- Super Glue
- MicroUSB cable (http://adafru.it/592) + 5V power supply / Battery Pack
- Helping Third Hands (http://adafru.it/291)
- Panavise Jr. (http://adafru.it/151)

Software

Trinket sketch to build a react or 'like' button that counts button presses and displays them on a 7-segment or 14-segment LED backpack.

You need the following hardware to build this device:

- Adafruit Trinket (https://adafru.it/m7B) (either the 5V Trinket (http://adafru.it/1501) or 3.3V Trinket (http://adafru.it/1500) will work)

- Adafruit 7-segment LED Backpack Display (http://adafru.it/879) or Adafruit 14-segment quad alphanumeric LED Backpack Display (http://adafru.it/1911)
- Momentary push button (http://adafru.it/1009)

Download the Arduino Sketches in a Zip File

https://adafru.it/EMI

Usage

Load the right sketch in Arduino depending on your hardware. The 7-segment display should use the TrinketReactCounter_7segment sketch, and the 14-segment quad alphanumeric display should use the TrinketReactCounter_14segment sketch. Make sure the following libraries are installed too (using the library manager or a manual install):

- Adafruit GFX Library (https://adafru.it/aJa)
- Adafruit BusIO Library (https://adafru.it/GxD)
- Adafruit LED Backpack (https://adafru.it/mau)

```
// SPDX-FileCopyrightText: 2019 Tony DiCola for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// Adafruit Trinket React Counter Sketch - 7-segment display
//
// Use a 7-segment LED backpack to display the number of times a
// button has been pressed.  Great for building a physical 'like'
// or react button.  The value will be stored  in EEPROM so it
// will persist between power down/up.
//
// NOTE: As-is this sketch needs to run on a Trinket because it
// assumes the switch on pin #1 has a pull-down resistor to ground.
// If using another board without this pull-down you can explicitly
// add a ~10kohm resistor from digital #1 to ground.
//
// Author: Tony DiCola
// License: MIT (https://opensource.org/licenses/MIT)
#include <EEPROM.h>
#include <Wire.h>
#include "Adafruit_LEDBackpack.h"
#include "Adafruit_GFX.h"


// Uncomment the line below to reset the counter value in EEPROM to zero.
// After uncommenting reload the sketch and during the setup the counter
// will be reset.  Then comment the line again and reload to start again
// (if you don't comment it out then every time the board powers on it
// will reset back to zero!).
//#define RESET_COUNT

// OR just hold the button for longer than the RESET_HOLD_SECOND below
// to reset the count!
```

```
// Configuration:
#define LED_BACKPACK_ADDRESS   0x70   // I2C address of the backpack display.
                                      // Keep the default 0x70 unless you
                                      // change the backpack's address bridges.

#define COUNT_BUTTON_PIN       1      // Digital input connected to the button that
                                      // will increase the count. This line should
                                      // have a pull-down resistor to ground.  The
                                      // opposite side of the button should be
                                      // connected to a high level like 5V or 3.3V.

#define COUNT_ADDRESS          0      // Address in EEPROM to store the counter.
                                      // This will take 2 bytes (16-bit value).
                                      // You don't need to change this unless you
                                      // want to play with different EEPROM
locations.

#define RESET_HOLD_SECONDS     5      // Number of seconds to hold the button down
to
                                      // force a reset of the count to zero.  Set
to 0
                                      // to disable this functionality.

// 7-segment display
Adafruit_7segment backpack = Adafruit_7segment();
uint32_t holdStart = 0;

void update_display() {
  // Get the count value from EEPROM and print it to the display.
  uint16_t count;
  EEPROM.get(COUNT_ADDRESS, count);
  backpack.print(count, DEC);
  backpack.writeDisplay();
}

void setup() {
  // Setup button inputs.
  pinMode(COUNT_BUTTON_PIN, INPUT);

  // Initialize the LED backpack display.
  backpack.begin(LED_BACKPACK_ADDRESS);

  // Clear the count in EEPROM if desired.
  #ifdef RESET_COUNT
    uint16_t count = 0;
    EEPROM.put(COUNT_ADDRESS, count);
  #endif

  // Update the display with the current count value.
  update_display();

  // Reset the last known time the button wasn't being held.
  holdStart = millis();
}

void loop() {
  // Take a couple button readings with a small delay in between to detect when
  // the signal changes from high to low, i.e. the button was released.
  int firstCount = digitalRead(COUNT_BUTTON_PIN);
  delay(20);
  int secondCount = digitalRead(COUNT_BUTTON_PIN);

  // Check for count button release.
  if (firstCount == HIGH && secondCount == LOW) {
    // Button was released!
    // Increment the count value stored in EEPROM.
    uint16_t count;
    EEPROM.get(COUNT_ADDRESS, count);
```

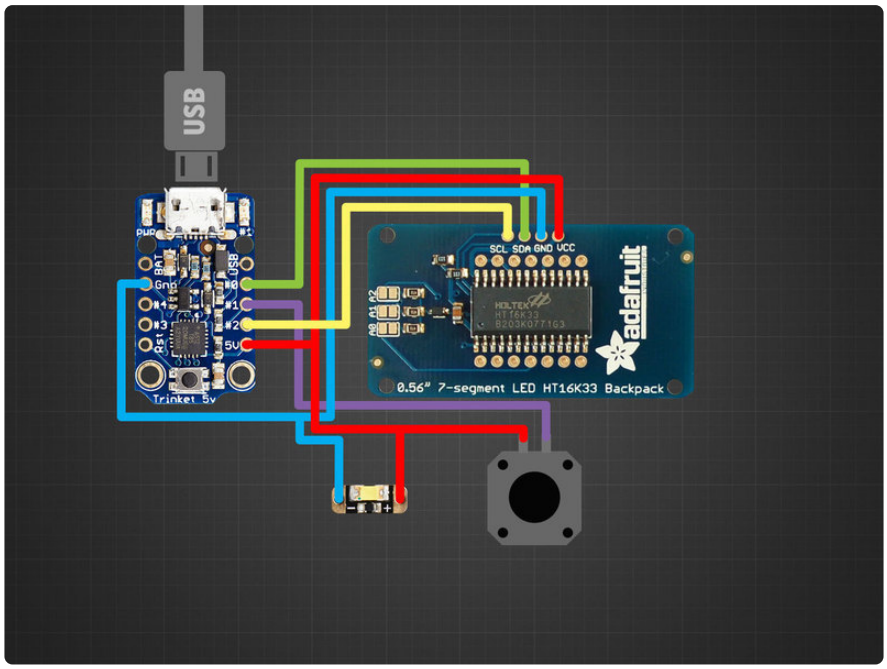
```
    count += 1;
    EEPROM.put(COUNT_ADDRESS, count);
    // Update the display with the latest count value.
    update_display();
  }

  // Reset the hold start if the button wasn't pressed (i.e. first and last were
not both high levels).
  if ((firstCount != HIGH) || (secondCount != HIGH)) {
    holdStart = millis();
  }

  // Check if the button has been held for the amount of reset time.
  if ((RESET_HOLD_SECONDS > 0) && ((millis() - holdStart) >=
(RESET_HOLD_SECONDS*1000))) {
    // Reset to zero and update the display!
    uint16_t count = 0;
    EEPROM.put(COUNT_ADDRESS, count);
    update_display();
    // Reset the hold time to start over again.
    holdStart = millis();
    // Flash the display a few times to give time to remove finger.
    for (int i=0; i<5; ++i) {
      delay(200);
      backpack.clear();
      backpack.writeDisplay();
      delay(200);
      update_display();
    }
  }
}
```

```
// SPDX-FileCopyrightText: 2019 Tony DiCola for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// Adafruit Trinket React Counter Sketch - 14-segment quad alpha display
//
// Use a 14-segment quad alphanumeric LED backpack to display the
// number of times a button has been pressed.  Great for building
// a physical  'like' or react button.  The value will be stored
// in EEPROM so it will persist between power down/up.
//
// NOTE: As-is this sketch needs to run on a Trinket because it
// assumes the switch on pin #1 has a pull-down resistor to ground.
// If using another board without this pull-down you can explicitly
// add a ~10kohm resistor from digital #1 to ground.
//
// Author: Tony DiCola
// License: MIT (https://opensource.org/licenses/MIT)
#include <EEPROM.h>
#include <Wire.h>
#include "Adafruit_LEDBackpack.h"
#include "Adafruit_GFX.h"


// Uncomment the line below to reset the counter value in EEPROM to zero.
// After uncommenting reload the sketch and during the setup the counter
// will be reset.  Then comment the line again and reload to start again
// (if you don't comment it out then every time the board powers on it
// will reset back to zero!).
//#define RESET_COUNT

// OR just hold the button for longer than the RESET_HOLD_SECOND below
// to reset the count!

// Configuration:
```

```
#define LED_BACKPACK_ADDRESS   0x70   // I2C address of the backpack display.
                                      // Keep the default 0x70 unless you
                                      // change the backpack's address bridges.

#define COUNT_BUTTON_PIN       1      // Digital input connected to the button that
                                      // will increase the count. This line should
                                      // have a pull-down resistor to ground.  The
                                      // opposite side of the button should be
                                      // connected to a high level like 5V or 3.3V.

#define COUNT_ADDRESS          0      // Address in EEPROM to store the counter.
                                      // This will take 2 bytes (16-bit value).
                                      // You don't need to change this unless you
                                      // want to play with different EEPROM
locations.

#define RESET_HOLD_SECONDS     5      // Number of seconds to hold the button down
to
                                      // force a reset of the count to zero.  Set
to 0
                                      // to disable this functionality.

// 14-segment quad alphanumeric display
Adafruit_AlphaNum4 backpack = Adafruit_AlphaNum4();
uint32_t holdStart = 0;

void update_display() {
  // Get the count value from EEPROM and print it to the display.
  uint16_t count;
  EEPROM.get(COUNT_ADDRESS, count);
  // Use writeDigitRaw function to write out each digit for the thousands,
  // hundreds, tens, and ones place.
  int thousands = count / 1000;
  int remainder = count % 1000;
  int hundreds = remainder / 100;
  remainder = remainder % 100;
  int tens = remainder / 10;
  remainder = remainder % 10;
  // Check if the value is too large to display and just print dashes.
  if (thousands >= 10) {
    backpack.writeDigitAscii(0, '-');
    backpack.writeDigitAscii(1, '-');
    backpack.writeDigitAscii(2, '-');
    backpack.writeDigitAscii(3, '-');
  }
  else
  {
    // Print out the number starting from the first non-zero digit.
    if (thousands > 0) {
      backpack.writeDigitAscii(0, '0'+thousands);
      backpack.writeDigitAscii(1, '0'+hundreds);
      backpack.writeDigitAscii(2, '0'+tens);
      backpack.writeDigitAscii(3, '0'+remainder);
    }
    else if (hundreds > 0) {
      backpack.writeDigitAscii(1, '0'+hundreds);
      backpack.writeDigitAscii(2, '0'+tens);
      backpack.writeDigitAscii(3, '0'+remainder);
    }
    else if (tens > 0) {
      backpack.writeDigitAscii(2, '0'+tens);
      backpack.writeDigitAscii(3, '0'+remainder);
    }
    else {
      backpack.writeDigitAscii(3, '0'+remainder);
    }
  }
  backpack.writeDisplay();
}
```

```
void setup() {
  // Setup button inputs.
  pinMode(COUNT_BUTTON_PIN, INPUT);

  // Initialize the LED backpack display.
  backpack.begin(LED_BACKPACK_ADDRESS);

  // Clear the count in EEPROM if desired.
  #ifdef RESET_COUNT
    uint16_t count = 0;
    EEPROM.put(COUNT_ADDRESS, count);
  #endif

  // Update the display with the current count value.
  update_display();

  // Reset the last known time the button wasn't being held.
  holdStart = millis();
}

void loop() {
  // Take a couple button readings with a small delay in between to detect when
  // the signal changes from high to low, i.e. the button was released.
  int firstCount = digitalRead(COUNT_BUTTON_PIN);
  delay(20);
  int secondCount = digitalRead(COUNT_BUTTON_PIN);

  // Check for count button release.
  if (firstCount == HIGH && secondCount == LOW) {
    // Button was released!
    // Increment the count value stored in EEPROM.
    uint16_t count;
    EEPROM.get(COUNT_ADDRESS, count);
    count += 1;
    EEPROM.put(COUNT_ADDRESS, count);
    // Update the display with the latest count value.
    update_display();
  }

  // Reset the hold start if the button wasn't pressed (i.e. first and last were
not both high levels).
  if ((firstCount != HIGH) || (secondCount != HIGH)) {
    holdStart = millis();
  }

  // Check if the button has been held for the amount of reset time.
  if ((RESET_HOLD_SECONDS > 0) && ((millis() - holdStart) >=
(RESET_HOLD_SECONDS*1000))) {
    // Reset to zero and update the display!
    uint16_t count = 0;
    EEPROM.put(COUNT_ADDRESS, count);
    update_display();
    // Reset the hold time to start over again.
    holdStart = millis();
    // Flash the display a few times to give time to remove finger.
    for (int i=0; i<5; ++i) {
      delay(200);
      backpack.clear();
      backpack.writeDisplay();
      delay(200);
      update_display();
    }
  }
}
```

Circuit Diagram



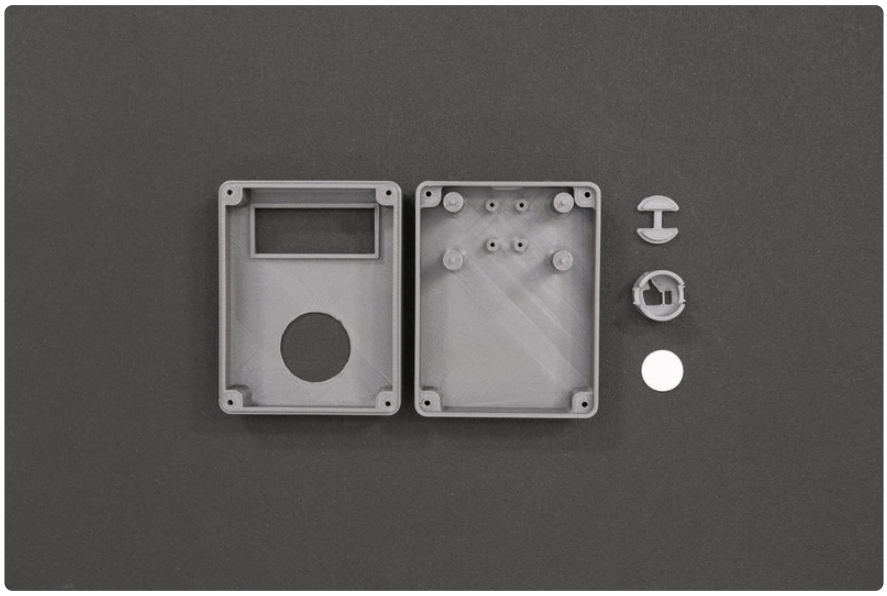
Reference Connections

Use the circuit diagram to reference for connecting the components together. The diagram does not depict exact wire lengths or size of components.

Connect the hardware as follows:

- Trinket 3V/5V power to 7-segment power (+) and one side of the button. If using the 14-segment display connect 3V/5V power to the additional power (+) pin.
- Trinket #0 to 7-segment SDA (D).
- Trinket #1 to opposite side of button.
- Trinket #2 to 7-segment SCL (C).
- Trinket GND/ground to 7-segment ground (-).
- Trinket 3V/5V power to positive (+) on LED sequin
- Trinket GND/ground to negative (-) on LED sequin

3D Printing



Materials

We suggest using PLA material but your free to use ABS, PET or exotic composites like wood, metals and others. The parts are listed in the tablet below.

7seg-case-btm.stl	text	text
7seg-case-top.stl	text	text
ab-cap.stl	text	text
ab-LED-holder.stl	text	text
ab-heart-diff.stl	text	text
ab-thumb-diff.stl	text	text

Slice Settings

Depending on your 3D printers hardware, you'll need to use your prefered slice settings. The parts are oriented to print "as-is" and doesn't require any support materials (very minimal overhangs).

These are the slice settings we used on our Printrbot Play, sliced using Simplify3D

- 220C Extruder (on a non-heated bed)
- 20% Infill
- 2 shells/parameters
- 5 top and bottom layers
- 0.9 Extrusion multiplier
- 0.48 Extrusion width

Customize Design

The enclosure and emjoi cover parts are available to modify and download. Click below to download the source. Add your favorite emjoi!

Download Source

https://adafru.it/m7E

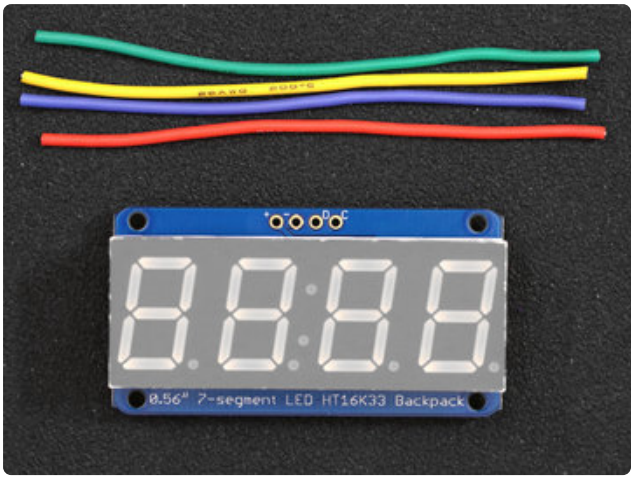
Download STLs

All the 3D printed can be downloaded on our thingiverse post, linked below. If you make one, please post a photo of it via make button!

Download STLS

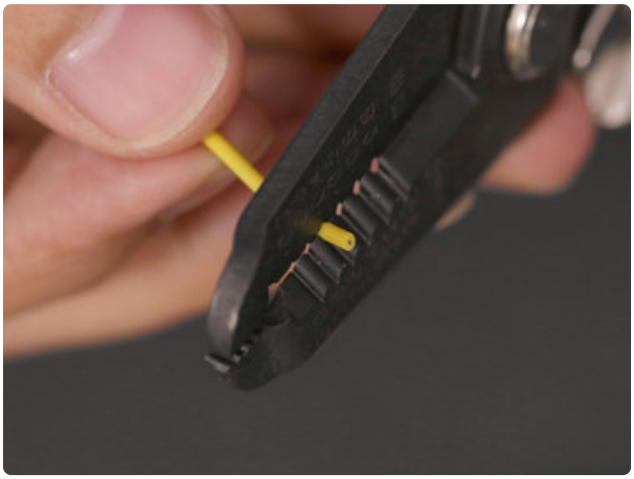
https://adafru.it/m7F

7-Segment Display



Measure & Cut Wires

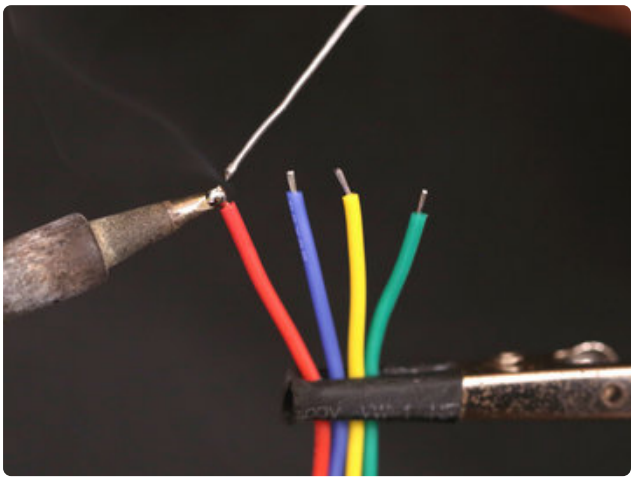
Let's start by getting four pieces of wire. I recommend using 26AWG silicone coated wires. These should be about 75mm in length. They don't have to be multi-colored but it's nice if you have them to better decipher the connections.

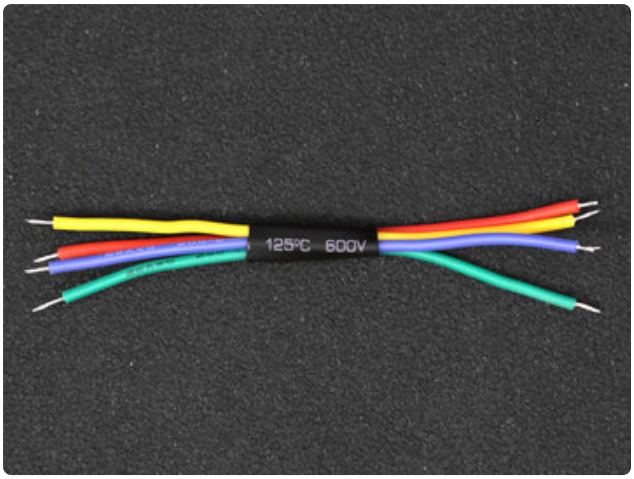


Strip & Tin Wires

It's good practice to tin your wires before soldering them to pins - this will help prevent the strands of wire from fraying.

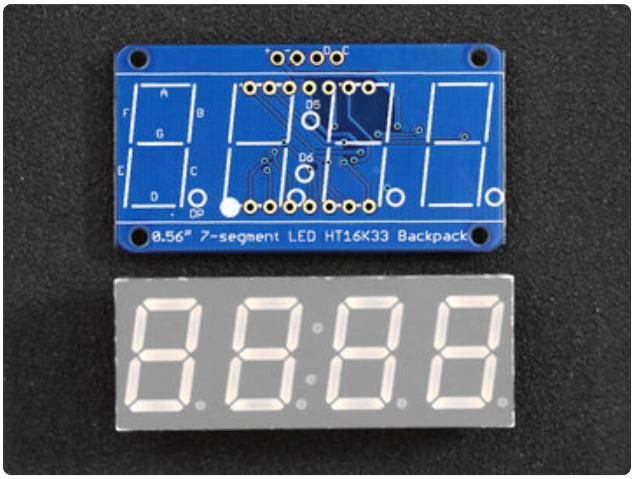
Use a pair of wire stripper to remove 5mm of insulation from the tip of each wire. I like to secure all my wires to a helping third helper and tin them in a group. This is much more convenient than setting up each one.





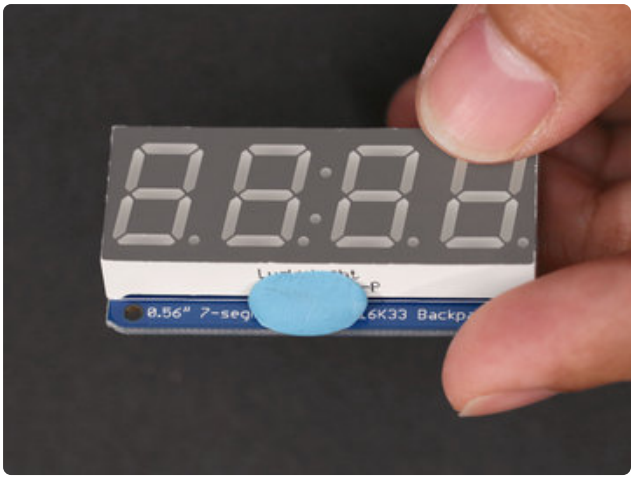
Bundle Wire Set

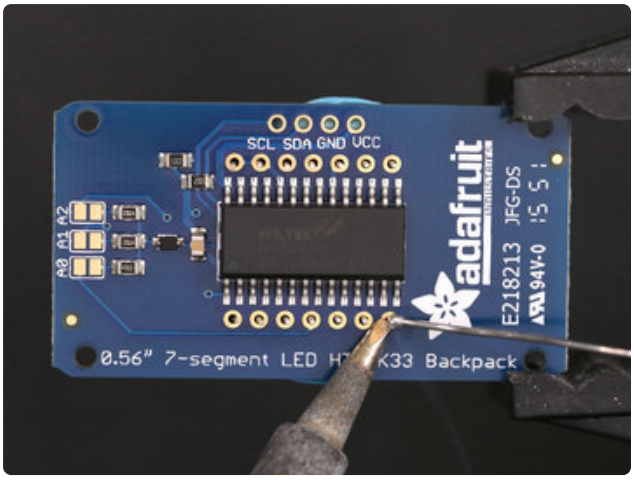
With all four wires stripped and tinned, lets bundle them up using a piece of heat shrink tubing. This stuff is normally used for insulating exposed connections, but I like to use it to group wires togeher. This will keep the wiring nice and tidy.



Install LED Display to Backpack

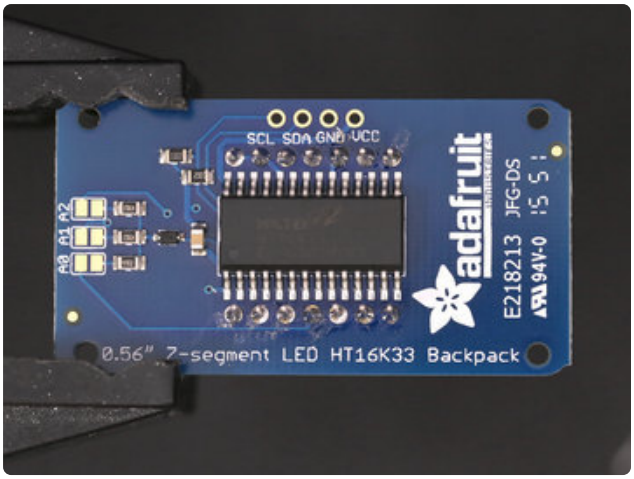
OK, now attach the 7-segment LED display to the backpack PCB. Be sure to insert the display in the correct orientation. The PCB has a outline and markings, use them to determine the correct orientation. The dots on the PCB should match with the dots on the display. Line up the pins on the display and carefully insert them into the pins on the PCB. If the pins don't fit, you may need to slightly bend the pins into place.

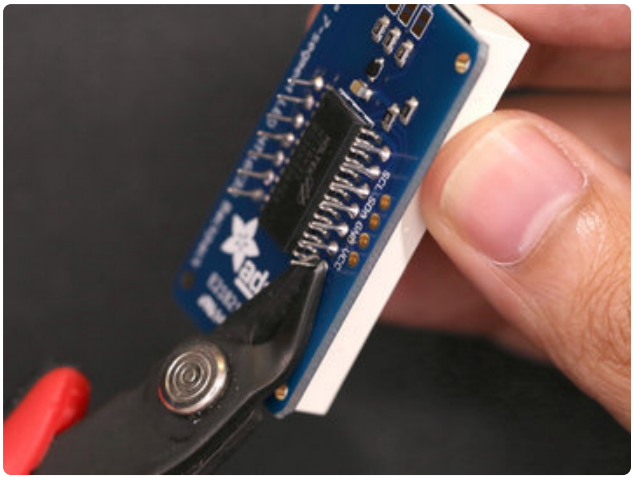




Solder LED Display to Backpack

OK, now it's time to solder the LED display to the backpack PCB. I like to use a piece of mounting tack to keep the display in place while I solder up the pins. Alternatively, a piece of tape is suffice. I suggest using a piar of helping third hands or a panavise jr. to keep the PCB steady while soldering.

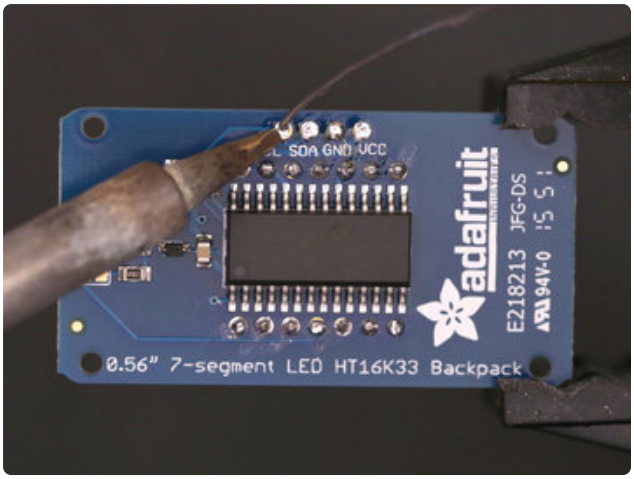


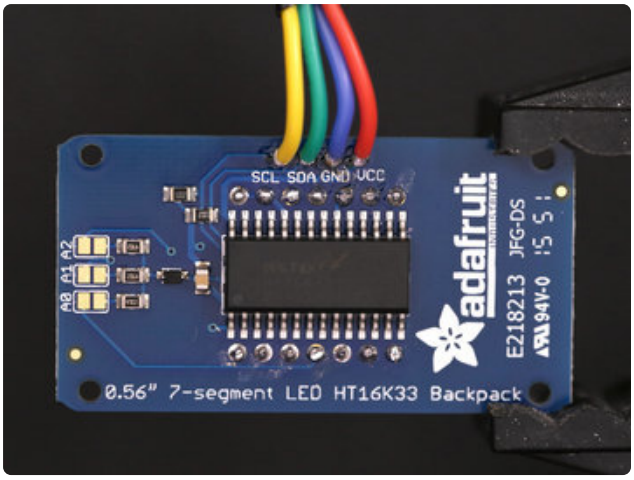


Trim Excess Leads

Inspect your solder joints and ensure you've applied a sufficient amount of solder to the pins. Make sure there are no cold solder joints. They should look like Hershey's Kisses.

Once they're good, you should remove the excess pins from the LED display. Cut all them short using a pair of flush snips.





Solder Wires to LED Display Backpack

Now it's time to connect our four wires to the pins on the Backpack PCB. I suggest tinning the four pins, located on the top of the Backpack PCB first.

The order of the colored wires doesn't matter too much but I like have some sort of sense. Red is positive, Blue is negative. Serial clock (SCL) and serial data (SDA) can be whatever color you'd like. The colors are just to help keep track of the connections.

My method for soldering wires into the pins is to heat up the eage of the tinned pin with the soldering iron while and insert the wire into the pin while the solder is molten, then quickly move the iron away.

Wired 7-segment LED Display

Our LED display is now wired and ready to connect to the Trinket micro-controller. But first, let's setup the arcade button and LED sequin. Set the LED display aside for now and get ready for the next steps. Here's a good spot to take a break!

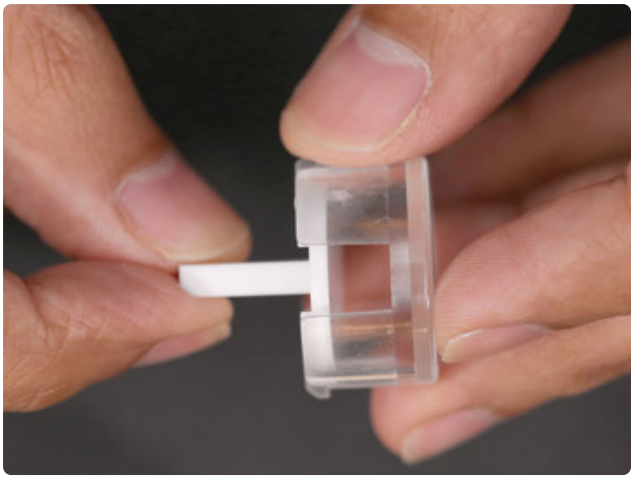
Arcade Button

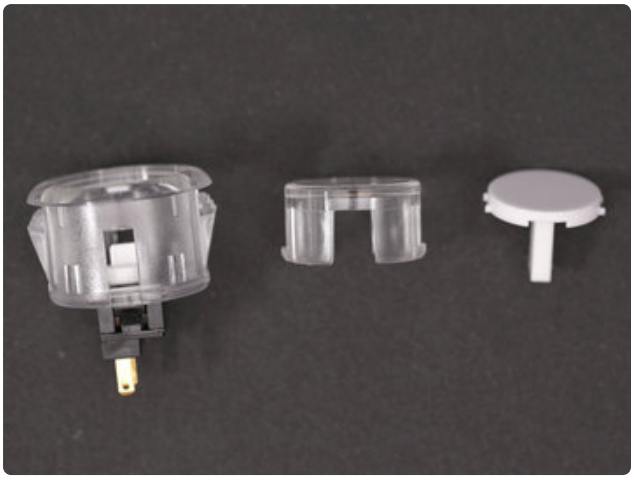
Arcade Button

This is a 30mm arcade button with a low-profile and nice press feel. We'll need to take it apart in order to add our 3D printed emjoi cover and an LED sequin so it can light up!

Remove Actuator

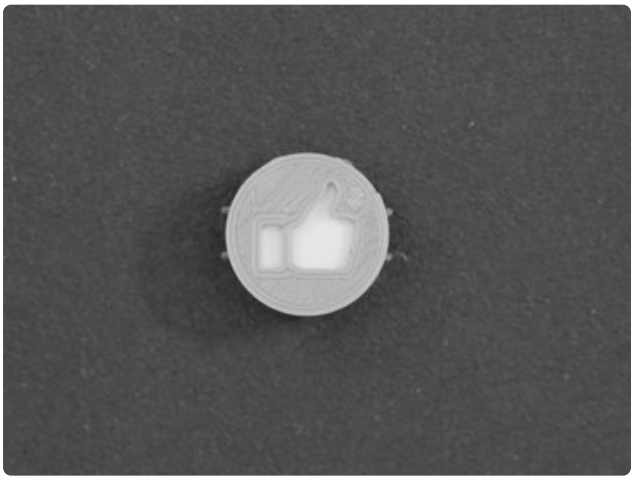
Let's start by removing the actuator/cover from the button housing. There's two clips on the side of the that keep the cover held in place. Use a screwdriver to push those clips inward and move it up. This will make the cover come lose. Then, pull the cover out of the button housing.





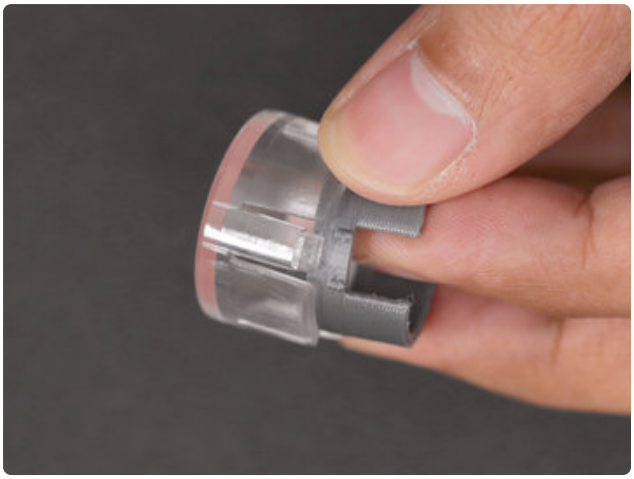
Separate Pieces

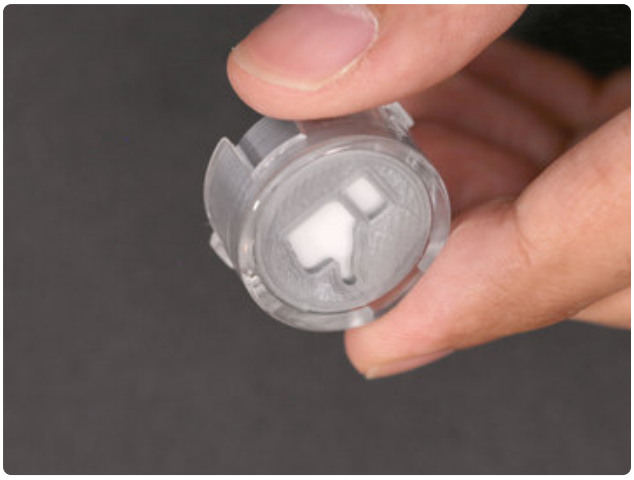
With the cover out of the button housing, pull out the white colored actuator from the cover. You can disgard the actuator, we'll replace it with a 3D printed emjoi actuator. But, keep the transparent cover, we still need it!



Insert Diffuser to Cover

Now we can assemble our emjoi actuator/ cover. It's a two piece thing. The **ab-cap.stl** part should be printed in either translucent or white colored filament while the emjoi cover can be printed in a dark color. This is to allow the LED to diffuse the cap and make the emjoi cutout glow. Press the **ab-cap.stl** part into the emjoi cover. It should have a tight tolerance and snap into place.

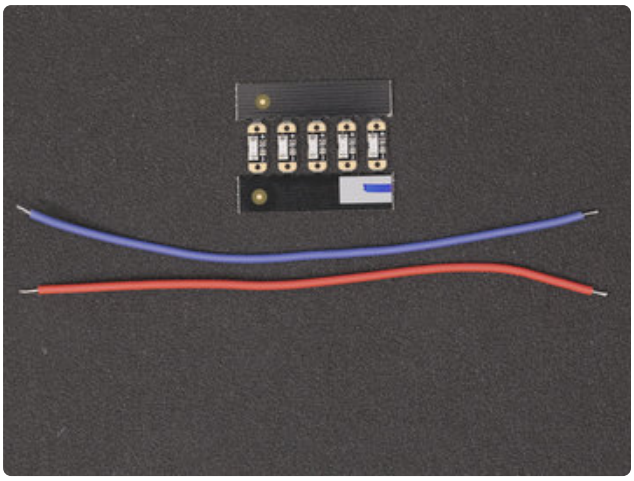




Install Emjoi into Button Cover

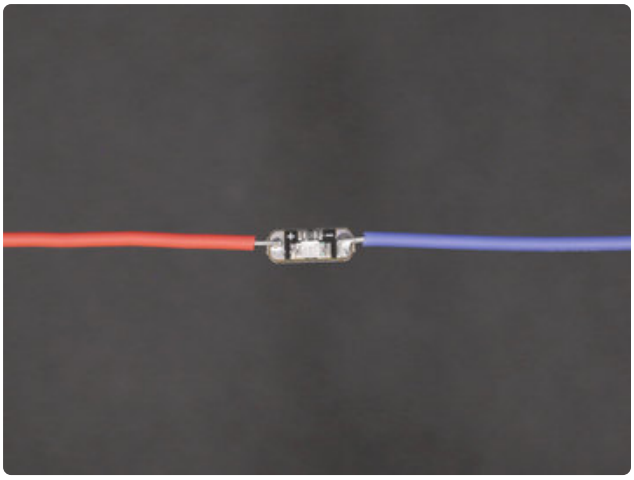
Next, insert the emjoi cover into the transparent cover. There's tiny protrusion on the sides of the cover that should go through the slits on the side of the transparent button cover. Make sure they're lined up when pressing it in. Press it all the way in, the tolerances should be tight enough to hold it into place.

LED Sequin



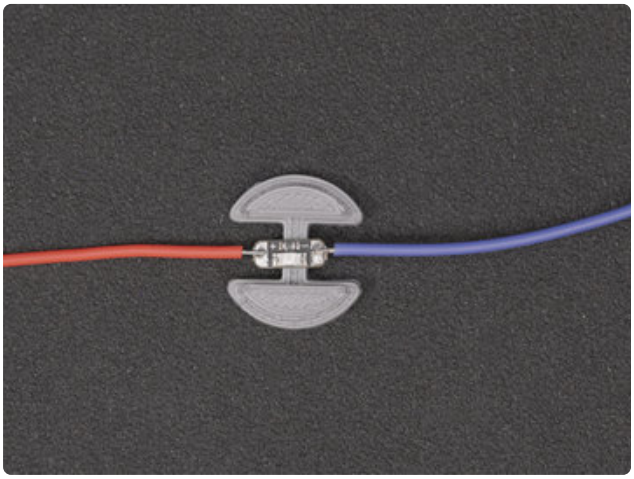
Measure & Cut Wires for LED

The Adafruit LED Sequins are great for this project because they're tiny, really bright and already have a resistor on the PCB. They come in a pack of 5. Break off one piece from the set. Then, measure two extra wires to about 75mm in length. Go ahead and remove the tips from each wire using wire strippers and tin the tips with solder, just like we need for the LED display.



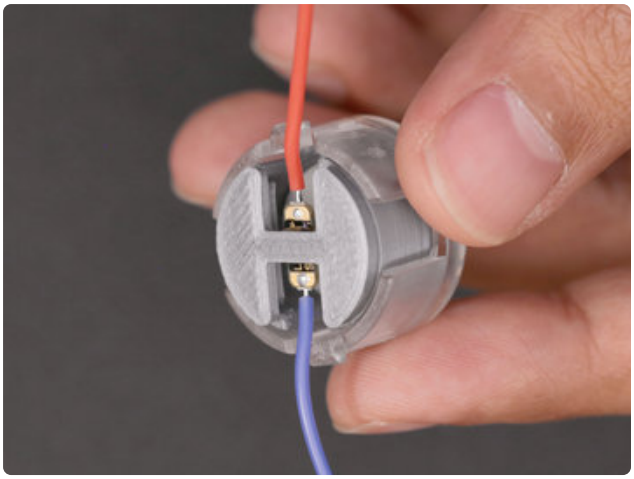
Solder Wires to LED Sequin

Let's add a small amount of solder to the positive and negative pads of the LED sequin. Then, solder one wire to each of the pads. I used a red colored wire for positive and blue for negative. They're marked with a + and – label on the PCB.



Secure LED Sequin to Holder

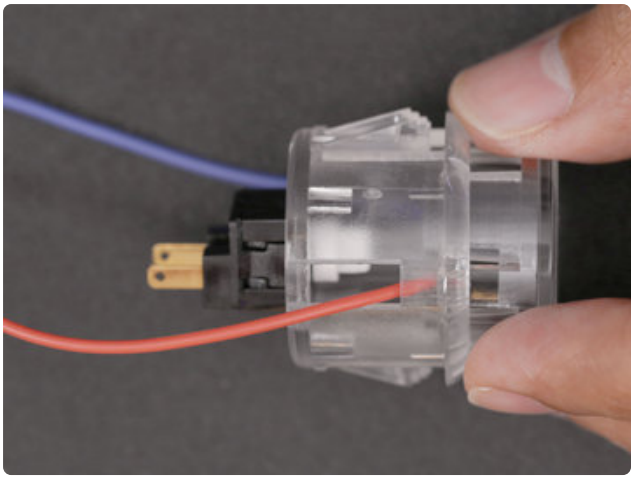
OK, now that our LED sequin is wired up, we can secure it to the **ab-LED-holder.stl** part. I used super glue to keep the LED held in place. Follow the orientation like in the photo. Place the LED sequin in the center of the holder part. Give the adhesives a few minutes to dry before moving onto the next steps.



Install Holder to Actuator Cover

After the glue has set, let's install the LED holder piece to the printed cover. Reference the photo for correct positioning.

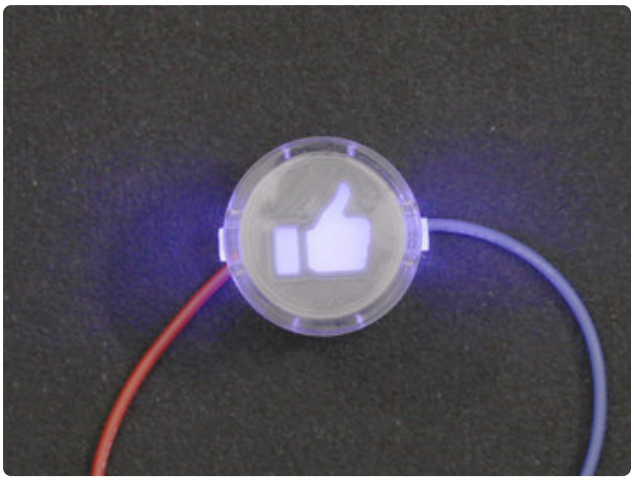
Then, grab the arcade button housing and the two wires from LED. Thread the wires through the openings on the side of the housing. Follow the photo for reference.



Reinstall Actuator to Button

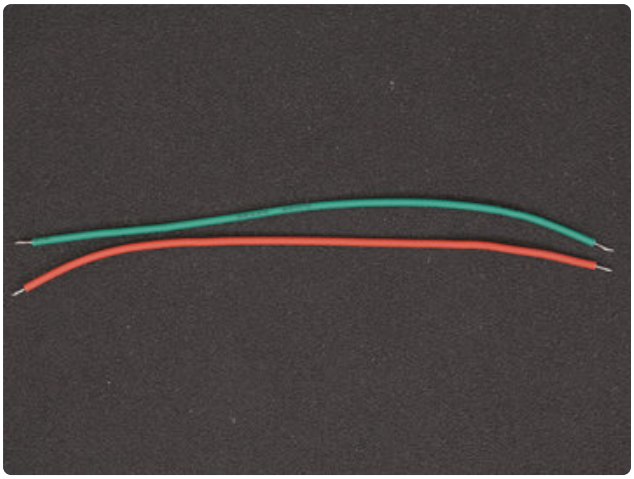
Slowly insert the cover back into the housing of the arcade button. Make sure the wires aren't being kinked. Try to keep the LED holder piece to the printed emjoi cover. Press the pieces together until the clips snap into place. If everyone is correctly in place, you should be able to press the button with ease.

If not, the LED holder may have come loose - remove the cover and try again. This requires a bit of finesse to get it right. If the LED holder keeps coming loose, you could glue it to the bottom of the emjoi cover.



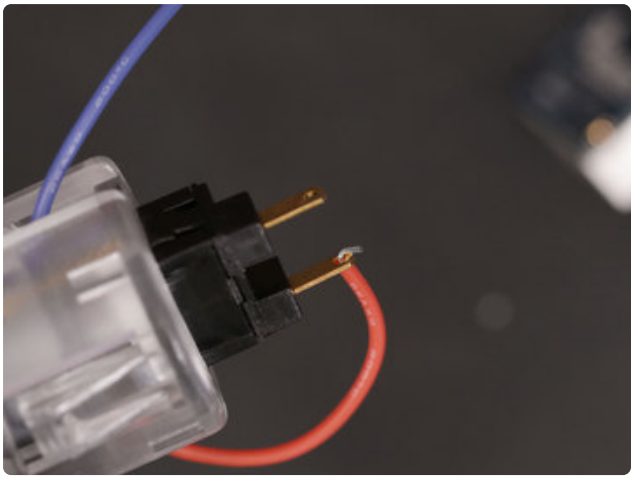
Test LED

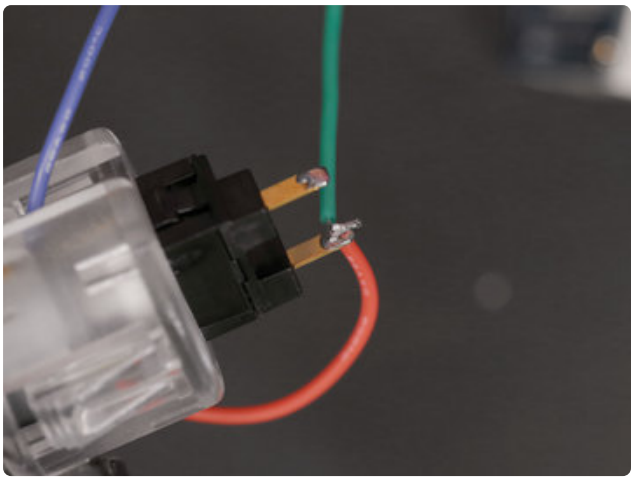
Now is a good time to test the LED. I did this by grabbing a coin cell battery by pressing and holding the wires to the positive and negative spots on the battery. The LED Sequin should light up! ~Thumbs Up~



Arcade Button Wires

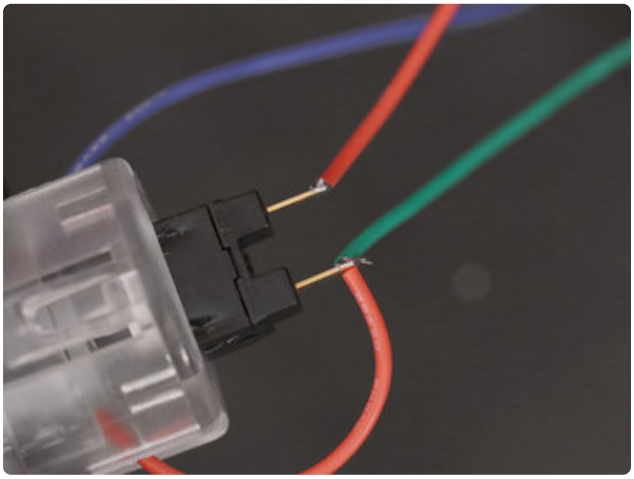
Next, we need to make "yet another" set of wires – these are going to be for the two leads on arcade button. They can be about 75mm in length and do the same "song and dace" for stripping and tinning the tips.





Arcade Button Wiring

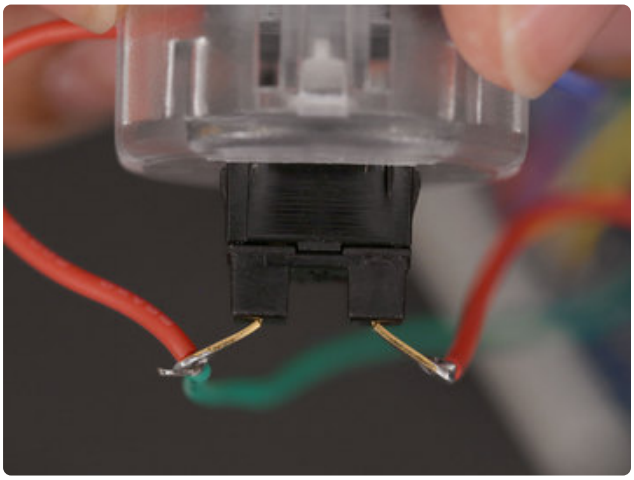
Let's start connecting wires from the arcade button. First up, let's focus on the positive connection from the LED sequin. I found the easiest way to do this is to share "power". Cut the wire short from the LED positive connection, then strip the tip of it (remove about 7mm of insulation) and thread it through the hole of the closest lead on the arcade button. Bend the exposed wire so it's held in place. Now we can solder one of our extra wires to this lead without the other coming loose. This extra wire is going to be connected to the 5V/3V pin of the Trinket. It will effectively "share" power from it. Reference the photo to get a better image of how this works.

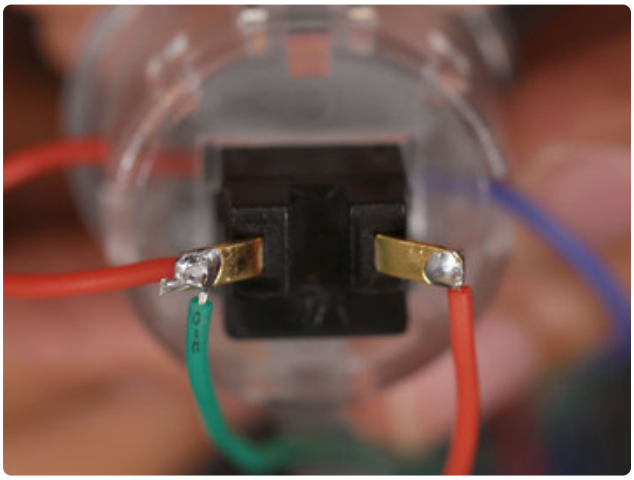


Arcade Button WIring [Continued]

Now we can solder the second "extra" wire to the opposite lead of the arcade button. You can either thread the wire through the hole or just solder it in place. This second wire will be connected to one of the digital pins on the Trinket later.

Hopefully the coloring here isn't confusing. In retrospect, I should have soldered this red colored "extra" wire to the positive connection of the LED sequin, but I digress.



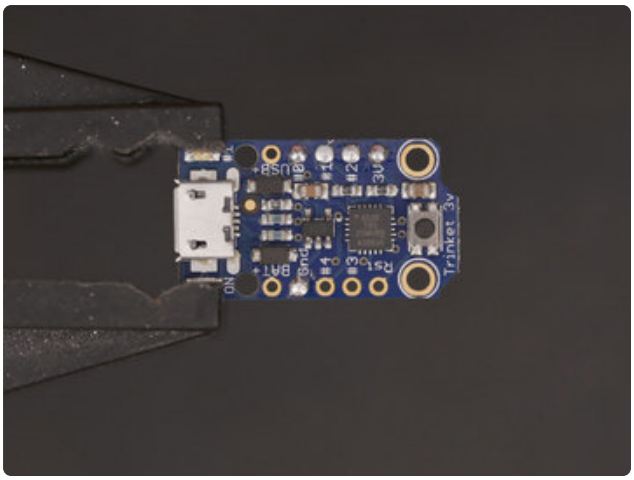


Bend Leads of Arcade Button

To accomodate for the limited space in the enclosure, you have to bend the leads of the arcade button so they're slightly angled. Reference the photo to get a better idea of how much they should be bend. If they aren't bend... the button won't fit inside the case. I recommend using a pair of pliers to do this.

Alrighty, now we have our two "extra" wired hooked up and our positive connection from the LED sequin wired to one of the leads on the arcade button. The negative connection from the LED sequin will be wired into the ground/GND pin on the Trinket later.

Trinket



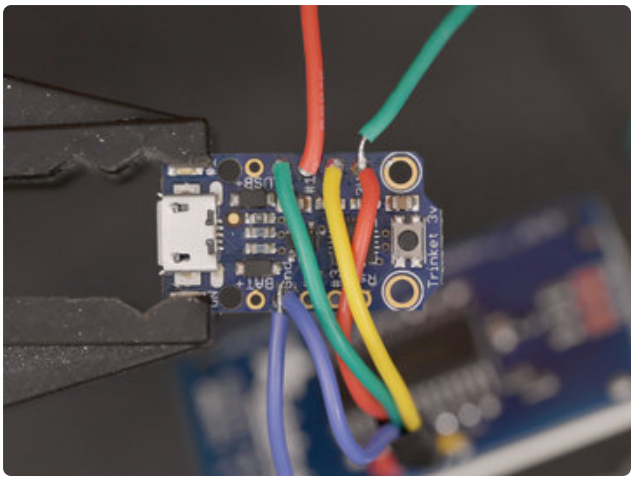
The Adafruit Trinket

OK, now it's time to get our Trinket ready for wiring. I recommend securing it to a pair of third helping hands, or a panavise jr like I have here. Go ahead and tin the following pins on the Trinket.

#0
#1
#2
3V/5V
GND

Trinket Connections

Let's get to wiring our connections to the Trinket. It's pretty straight forward, the only thing I found tricky here are the connections that need to share power and ground.



Our negative connection from the LED sequin goes to the ground/GND pin on the Trinket. But, the negative connection from the 7-segment LED sequin also needs to go into ground/GND on the Trinket.

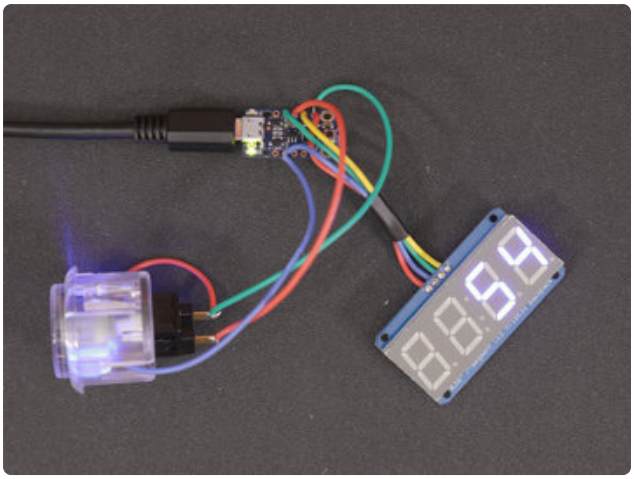
Our positive connection from the LED sequin and arcade button needs to go to the 3V/5V power pin on the Trinket. The positive connection from the 7-segment LED display also needs to go to 3V/5V power pin on the Trinket.

It's a bit crammed here, so be cautious about accidentally bridging any connections or cold solder joints.

Wired Connections

Connect the hardware as follows:

- Trinket 3V/5V power to 7-segment power (+) and one side of the button.
- Trinket 3V/5V power to positive (+) of LED Sequin
- Trinket #0 to 7-segment SDA (D).
- Trinket #1 to opposite side of button.
- Trinket #2 to 7-segment SCL (C).
- Trinket GND/ground to 7-segment ground (-).
- Trinket GND/ground to negative (-) of LED Sequin.



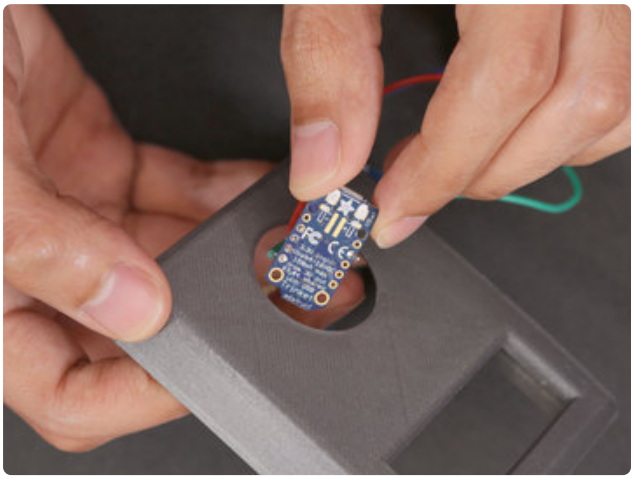
Check Circuit

If everything went well with wiring, we should test our circuit before mouting things to the enclosure. The code should have been uploaded to Trinket before we did any wiring, but if you haven't yet - That's ok, better late than never :-)

Connect a micro USB cable to the Trinket and plug it into the USB port on your computer. Wait for the boot loader to clear and the 7-segment LED display should display "----" or "0" number count. The LED sequin will power on instantly. Try pressing the arcade button, the number on the LED display should increment each time you press it.

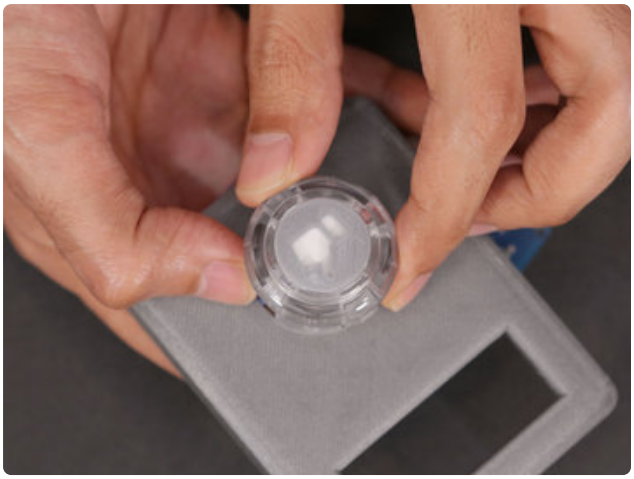
If everything works as except, let's move onto the mounting the components to the 3D pritned enclosure.

Mount Components



Insert Components to Case

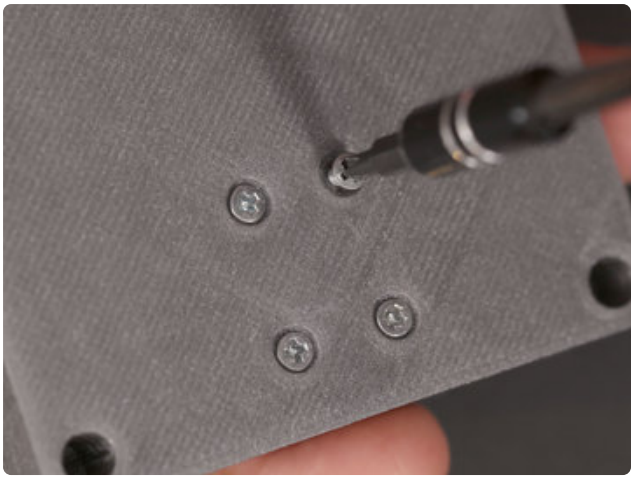
Start with the **7seg-case-top.stl** part with the outside facing up. Insert the 7-segment LED display and Trinket through the circular hole.

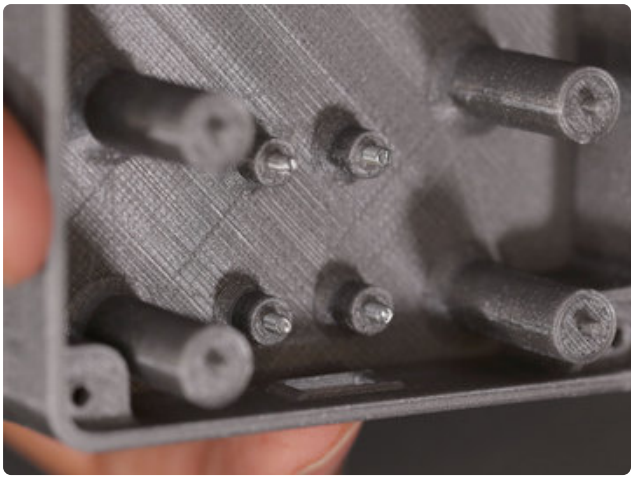


Arcade Button & 7-segment LED Display

The last component to go through the circular hole will be the arcade button. Orient the button so the emjoi is upright in the desired position. Then, press the arcade button through the hole until it snaps into place, fully flush with the case.

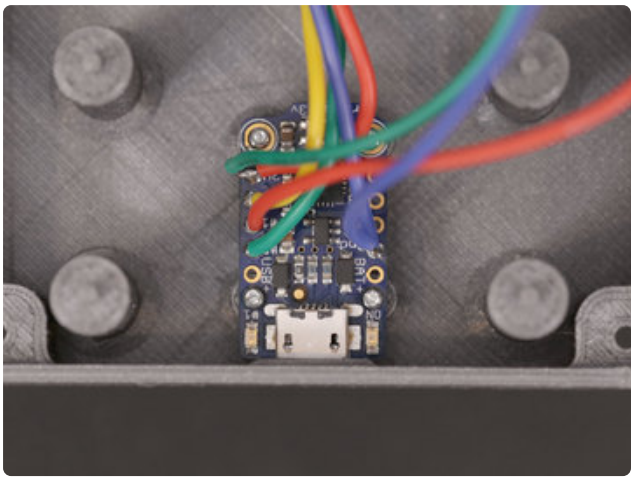
Position the 7-segment LED display over the rectangular cutout and press it through so it's seated into the enclosure.





Add Screws

Grab the **7seg-case-btm.stl part** and fasten four #2-56 sized 3/8 long flat phillips machine screws into the four holes on the bottom. Fasten the all the way until the heads are flush with the surface of the case. Our Trinket will be seated on these four screws.



Install Trinket

Grab the Trinket and orient it so the microUSB port is facing the edge of the case. Insert the Trinket at an angle and position it over the screws so the mounting holes can go over them. Press it down into place and seat the Trinket into the screws.

Case Closed

OK, now we can join the two halves of the case together. The microUSB port should be orientated with the LED display. Press the two halves together to snap them shut. There are screw holes on the bottom of the **7seg-case-btm.stl** part on each corner where you can insert a #4-40 3/8 flat phillips machine screw, but I actually didnt need them since the tolerances are already pretty tight and hold the case together nicely. If you want to add them, it's entirely up to you!

Finished Assembly

Congradulations! Your physical like button is ready for use. It's basically a cool looking digital tally counter. If you made one, please consider taking a photo of your creation and positing a "make" on our Thingiverse page (https://adafru.it/m7F).

Questions, Issues, Concerns?

If you need any assistance with your build, please don't hesistate to post them up on the Adafruit Forums. We have on staff a dedicated support team to help you trouble shoot your project builds!

Get Help - Adafruit Forums

https://adafru.it/dlr