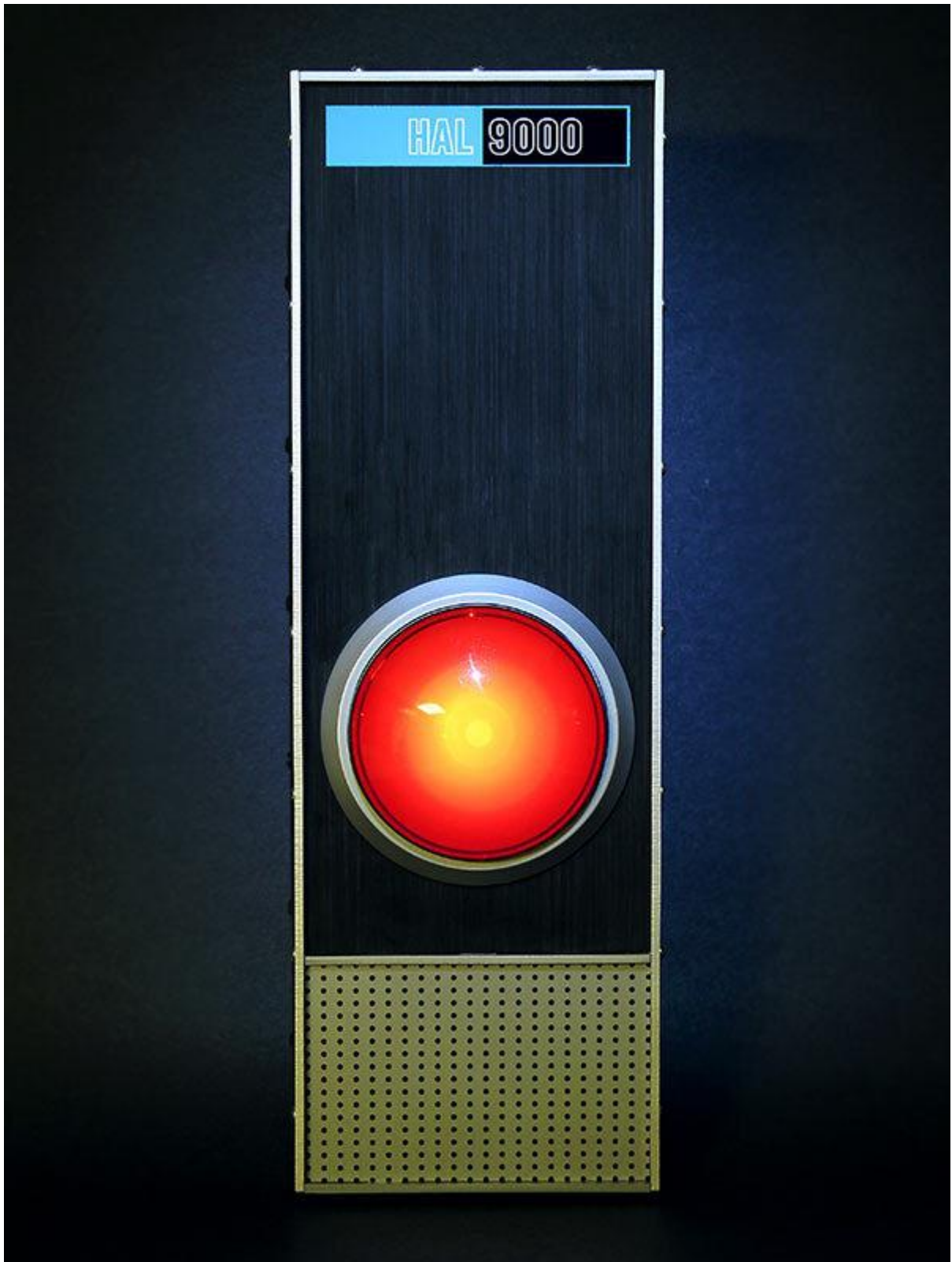




Affordable HAL 9000 Replica

Created by Phillip Burgess



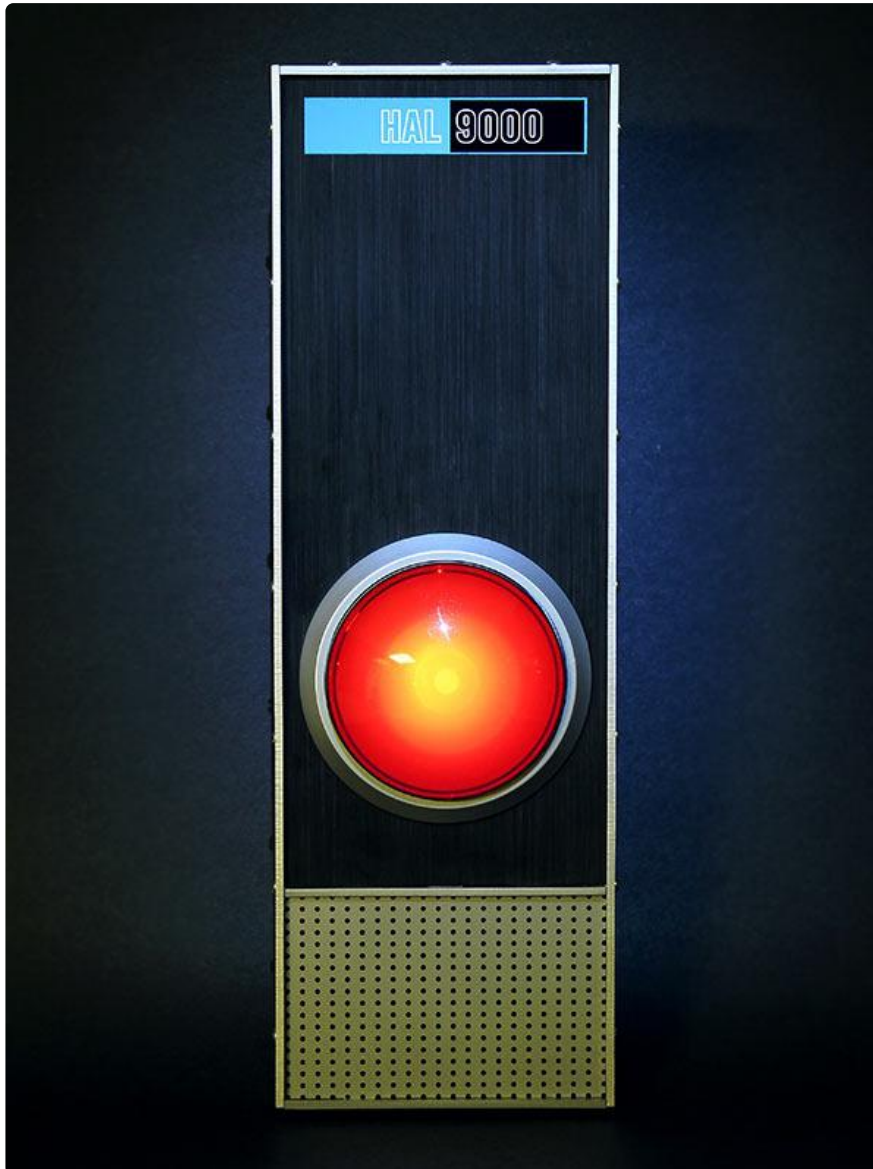
<https://learn.adafruit.com/hal-9000-replica>

Last updated on 2021-11-15 06:00:20 PM EST

Table of Contents

Overview	5
• Before you Begin...	6
It Starts With a Button	6
• Paint Like a Pro:	8
Crafting a Case	10
Assembly	15
Adding Sounds	19
Updating with CRICKIT	23
• Connect button to CRICKIT	26
• Connect speaker to CRICKIT	28
• Connect 5V DC power to CRICKIT	29
• Getting Familiar with Mu	30
• HAL speaks!	33
• Controlling volume	33

Overview



The Pareto Principle — also known as the 80/20 Rule — is the idea (originally from economics, but now applied in many ways) that 80% of results stem from 20% of the effort.

Devoted film fans will spend countless hours and hundreds of dollars (occasionally even thousands) to create flawless replica props for their personal collections. The iconic eye of HAL 9000 from 2001: a Space Odyssey is one such object of desire... popular enough that detailed (and pricey) licensed reproductions exist. This is cool stuff! But if we relax our criteria just a bit, you or I can turn out a pretty decent, recognizable facsimile in a weekend for just a small fraction of the cost. The 80/20 rule in action!

We're not selling a prop or even a kit here...that would raise a big licensing stink, so

please don't ask. What follows are some ideas on creating one yourself. Much like our [not-a-Back-to-the-Future](https://adafru.it/c8G) (<https://adafru.it/c8G>)-clock project (<https://adafru.it/c8G>), the concept came about when customers noted that a component already in our shop resembled an unrelated film item — in this case, our [Massive Red Arcade Button](http://adafru.it/1185) (<http://adafru.it/1185>) and HAL's distinctive lens.

Before you Begin...

This is not so much an electronics project as a craft project. The former would actually be easier to explain...a concrete matter of “connect wire A to point B,” period. The craft element though...what's described here as “affordable” or “simple” might be contingent on a lot of techniques, tools and materials that I've amassed over the years (for example, getting access to a laser cutter may be a tall order). You'll need to adapt the process to suit your skill set and the particular assortment of tools and materials at your disposal.

As described here, this project involves among other things: laser cutting, spray painting, inkjet printing, sanding, soldering (if adding voice) as well as patience, a well-stocked junk drawer and a willingness to improvise. Before buying any materials, read through the guide, see how it matches up with what you have and know.

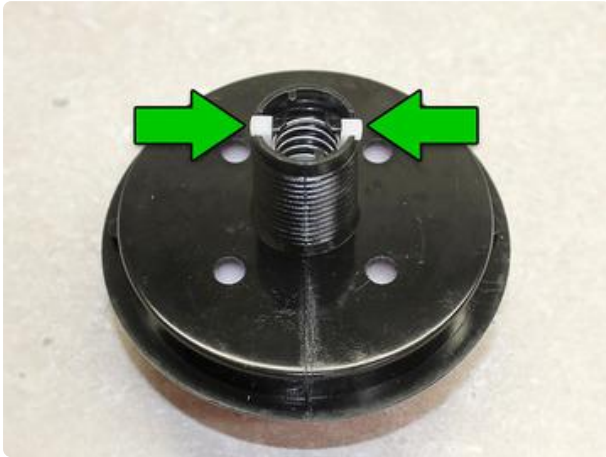
It Starts With a Button



The centerpiece of the project is this nifty LED-lit “massive arcade button” in the Adafruit shop, about \$10. From the moment this product photo was posted, comparisons were being made to HAL 9000. The button comes in other colors too —

you use the blue version to make SAL 9000 (the sister A.I. from 2010: Odyssey Two), or pick a different color and create your own unique 9000 Series character.

Doing a little homework to find actual HAL dimensions, I was surprised to see the button and bezel are within just a couple millimeters of the genuine prop — a Nikkor f/8 fisheye lens. We just need to dress it up a little, repainting the bezel from black to silver...



If you've installed the microswitch in back, remove that first by turning counterclockwise.

Squeeze these two tabs inward to release the button. It's possible but a bit tricky to do with finger pressure alone, so you may want to use needle-nose pliers (gently!).



The button then slides out the front of the bezel. Don't lose that spring!

There may be a slight oily residue on the bezel (mold release from manufacturing, I suspect). This can be cleaned with dish soap and water, but be certain it's absolutely dry before painting!

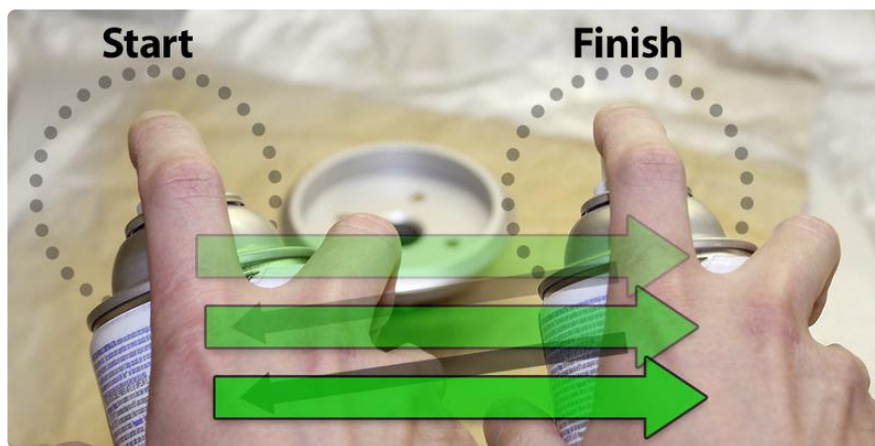


Find a metallic-looking spray paint that's specifically suited to bond to plastic; this saves a priming step. Hobby stores are a good source...not everything in the hardware store is plastic-ready, but I did find this suitable Rust-oleum paint there for about \$4. Aluminum color, perfect!

Paint Like a Pro:

The key to a good paint job is to use several light coats rather than one thick gloppy one. Allow 10-20 minutes between each coat to allow the paint to set up a bit. Make sure the paint is well-mixed prior to starting, and periodically give a few extra shakes while working.

Hold the can about a foot away from the subject, and spray a light mist in three or four overlapping horizontal passes, starting just off one edge, working straight across and past the other edge. Yes, you'll end up with more paint on your dropcloth than on the item, but it's key to getting a flawless finish. Go fold laundry or something, then come back for another coat (perhaps turning the piece for a different angle). Repeat until there's a few solid coats with no black showing through.



Avoid clogs: when done, clear the nozzle by inverting the can and spraying until just air comes out...or some newer nozzle designs work fine just wiping with a rag.

Allow the paint to dry for at least a couple hours (ideally overnight) before handling the piece. You can work ahead on other steps in the interim.



Perfect finish!

(Shown on a clean sheet of paper for contrast.)



Optional: if you want to take an extra step — let's call it fanboy factor 10 — you can color the perimeter of the button black to better resemble the original lens. It's not a perfect solution, but cheap and easy is the goal here.

Model paint and a steady hand is an inexpensive approach. My craft stash happened to include some black vinyl contact paper (shelf liner) from a prior project. I used the clean, factory-cut straight edge toward the front of the lens, and trimmed at the back with a hobby knife.



If you do decide on the black outer ring, you'll probably want the inside edge of the bezel to be free of paint, otherwise it rubs off when the button is pressed (as it inevitably will be). You can either sand this away (as shown here), washing and drying completely afterward, or with forethought could mask off this edge with tape before spray painting.

Once the paint is completely dry, the button can be reassembled. The spring (you still have the spring, right?) is a bit reluctant to go in straight...it may take a few attempts. Just be patient and try again.



Tally so far: \$14.

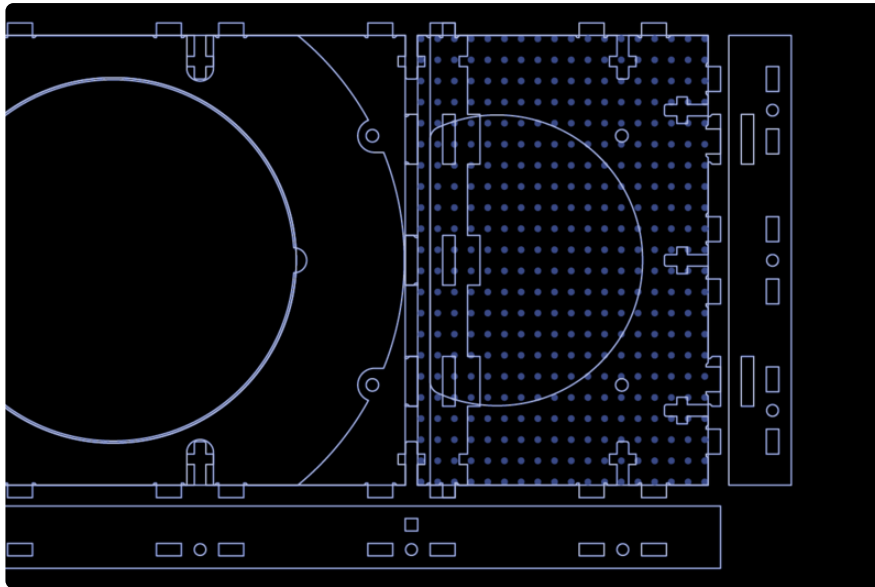
This doesn't include the extra materials for the optional black ring (sandpaper or masking tape, black model paint or contact paper) as I already had suitable items on-hand. Your tally may vary, depending on what steps you take and how you adapt this to your own supplies.

Crafting a Case

The hard work of coming up with a good blueprint was already done by Amadeus Prokopiak of the [Replica Prop Forum \(https://adafru.it/c9M\)](https://adafru.it/c9M), an excellent community of film aficionados. Working from photos of the original piece, and by scrutinizing individual Blu-Ray frames, Amadeus developed (and subsequently shared with the community) complete dimensions for his replica — possibly even better-constructed than what the rest of us saw on screen. If you want to take this project to the next level, [follow his thread \(https://adafru.it/c9O\)](https://adafru.it/c9O)! As for us, for right now, we'll be sticking with the quick bargain basement plan.

Working in Adobe Illustrator, I adapted the design to work with laser-cut acrylic and to fit our button. I chose to use common fasteners rather than glue...maybe not canon, but fast and easy. Acrylic cement has a way of seeping everywhere and ruining paint jobs. If you can't get access to a laser cutter, you might still adapt this file (or Amadeus' original) to your medium of choice.

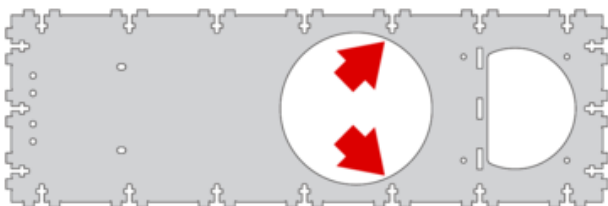
[YOU CAN DOWNLOAD THE GRAPHICS FILES HERE \(ADOBE ILLUSTRATOR 10 FORMAT\) \(https://adafru.it/c9V\)](https://adafru.it/c9V)



Since there wasn't a satisfactory laser-cuttable facsimile of aluminum, and with the metallic spray paint already on hand, I cut the entire case from 1/8" (3mm) black acrylic and painted it. Unfortunately I had to buy virgin plastic for this...the cheap discount scraps from the local plastics shop are all cut down to 8x10 inches, but HAL is nearly 14" tall. Two 15x10" sheets set me back \$15, the largest single expense of the project (unless you decide to add sounds).



I could have bought a bit less material, but having some surplus on hand is a good thing. Consider cutting and painting a few extra parts. This affords the opportunity to "cherry pick" the best-looking parts for your finished prop, and also to recover from goofs. They do happen!



Be extra careful with the large main "spine" piece. The plastic is quite thin between the eye hole and two particular T-slots. It's not a problem once all the supporting pieces are added.

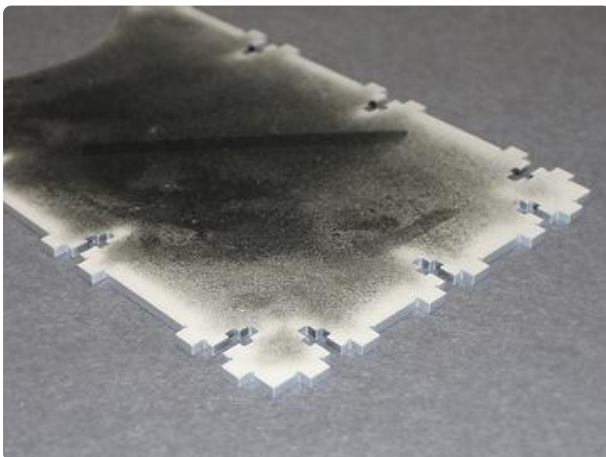
After cutting, the frame pieces will need to be painted as we did with the button bezel; use the previously-described techniques to ensure a smooth coat.

You'll need to paint both sides of these pieces, and the edges as well. Spray at a 45° angle to reach both the face and edge. Allow to dry completely, then flip over and paint the other side.

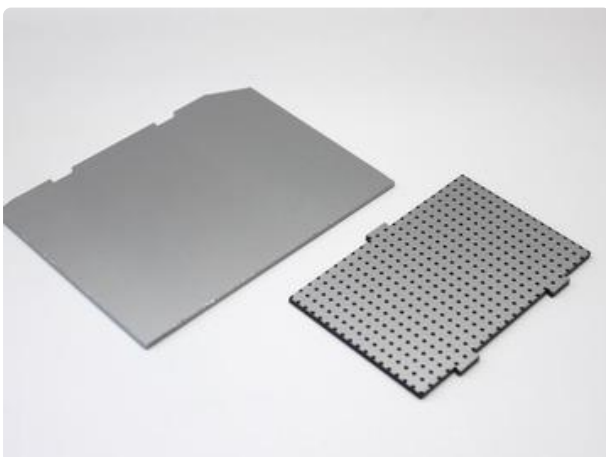


Here are the frame pieces after painting. The ruler is there as a point of reference: notice how “wavy” these pieces are; narrow pieces laser-cut from black acrylic seem particularly susceptible to this. The plethora of screw holes help hold everything flat once assembled.

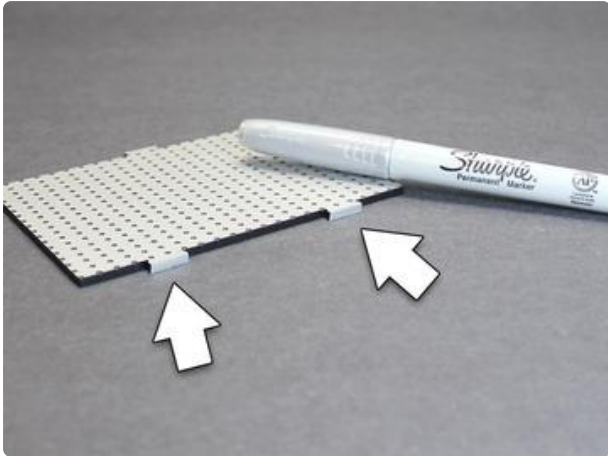
In hindsight, I might cut these pieces out of clear acrylic; it seems less prone to this sort of warping, and it's all getting painted anyway.



Because T-slot construction was used, the edges of the main spine need special attention. They'll be poking through the frame sides, so the edges of this one piece were also painted to match.



The speaker grille required its own set of hoops to jump through: all those tiny holes would be time-consuming and error prone (laser-cut holes tend to stick in place). Instead, a scrap of black acrylic was painted before cutting, then the laser was used to etch the surface and reveal the black plastic underneath those spots before the final perimeter cut.



The edge tabs on the grille needed to be painted too, but spraying would ruin the engraved “holes.” A silver Sharpie marker did the trick...but you could also spray a little paint into a disposable cup and dab it on with a brush or Q-tip.

The front faceplate can be handled a number of ways. It could be cut from the same glossy black acrylic and installed as-is, but this is the least authentic; like the black lens ring, we can step up the fanboy factor with a little extra work if we so choose...

- Black acrylic is available with a matte finish on one side, if you don't mind buying an extra bit of special material for this one part.
- Matte black spray paint can be used to kill the shine. Awful fingerprint magnet though.
- If you did the black contact paper thing around the eye, same material could be applied here.
- The front faceplate could be cut from 1/8" birch plywood and stained/painted black.
- We can rough up the acrylic to vaguely approximate a wood grain or brushed metal appearance...



On the left is a faceplate cut from matte acrylic. On the right, regular gloss acrylic that's been “distressed” to give it a faux texture.



Here's the rig that was used to create the textured finish. A full sheet of 220 grit sandpaper (\$1.25) is taped face-up to the workbench, while a scrap of wood provides a straight edge to work against.

The acrylic piece was dragged back and forth for a few minutes, being exceedingly careful never to move it perpendicular to the "grain." Occasionally the piece was turned around or the sandpaper was repositioned so there wouldn't be just one set of grooves.

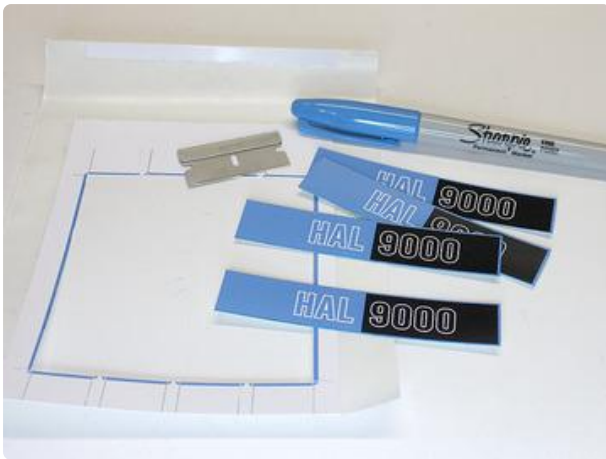
When finished, the piece is quite pale and dusty, but washing it with dish soap and water restores most of the dark appearance. Be sure to let it dry completely before proceeding.

The HAL 9000 name tag was inkjet printed on glossy photo paper. This is another case where a tool I already had around isn't factored into the cost.



Borrowing a trick first used for the [DeLorean Time Circuit \(https://adafru.it/c8G\)](https://adafru.it/c8G) project, this Xyron sticker machine makes short work of adhesive-backing the name tags...but I'm not suggesting you need to run out and buy one. A glue stick, spray adhesive or just some double-stick tape could work just as well.

Or save a step and simply use inkjet sticker paper, if you have it.



The name tags are trimmed with a razor blade, and the white edges cleaned up with a blue Sharpie.

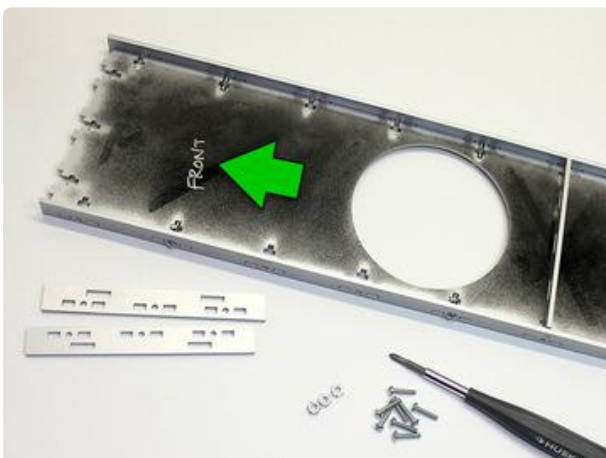
I mentioned earlier the value in producing multiple copies of parts when the material allows; that's why there's four name tags here. The first two were trimmed a little off-center. The other two are usable, but the last I felt was the cleanest of the bunch and is what will go on the finished prop.

I'm ready for my closeup, Mr. Kubrick!



Tally so far: \$30.25. I'm including the sandpaper in that figure, but otherwise I'm again assuming certain tools or materials will be scrounged from around the house.

Assembly



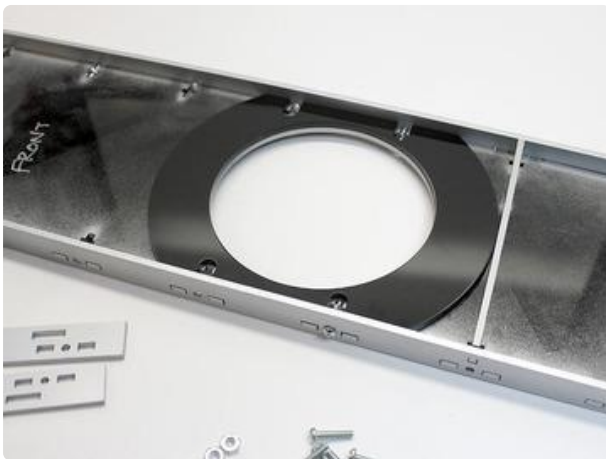
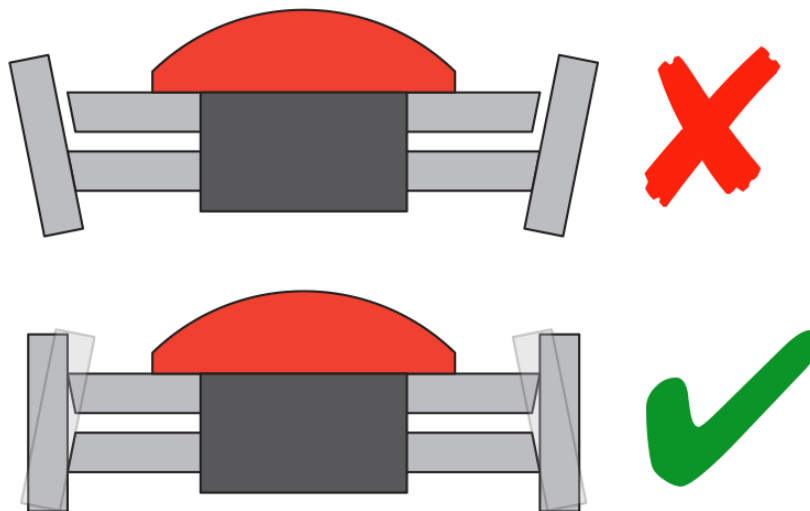
The spine, middle separator and the frame sides are joined using 1/2" #4-40 screws and nuts (20 total). About \$3. The top and bottom edges will be added shortly.

Hey, wait a sec...if the center piece is symmetrical, it should be OK to flip either way. Why then is this face labeled "front"?

The way the beam is focused in a laser cutter means that edges will always have a slight bevel. When building a box, normally we can rely on other planes to mutually

hold things square. But our HAL faceplate is rather thin and has only the single spine piece holding screws.

The trick is to examine the edge carefully...spot the bevel and arrange this piece so the frame parts want to lean “inward” — but will be held square by the front face. Done wrong, the frame pieces will splay outward and the corners won’t match up.

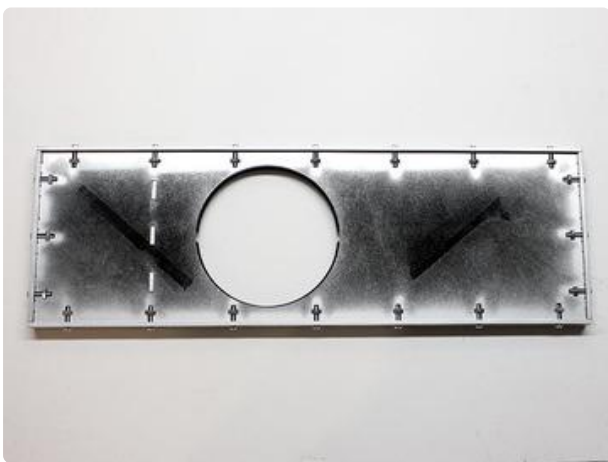


This spacer ring makes the unit feel more solid, so it doesn’t cave in when the button is pressed. Note that it’s not quite symmetrical — there’s a definite “up” and “down” due to the T-slots.

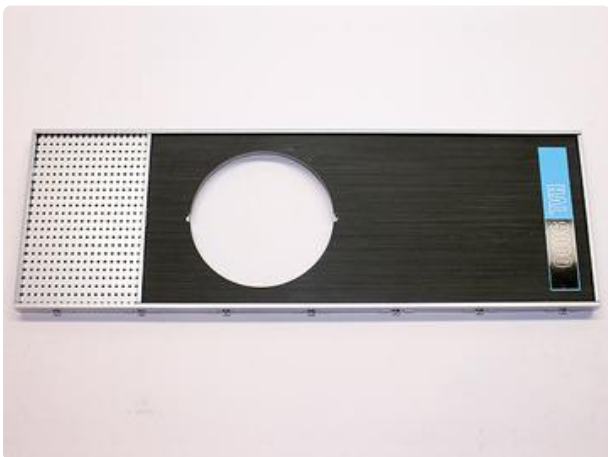


The front face then slots into place, and the top edge of the frame can be installed. Note that this piece only has tabs on the top and bottom, in order to minimize the number of holes along the frame (the button will be helping support it).

Not shown, the speaker grille can be similarly installed now, and the bottom edge of the frame with it.

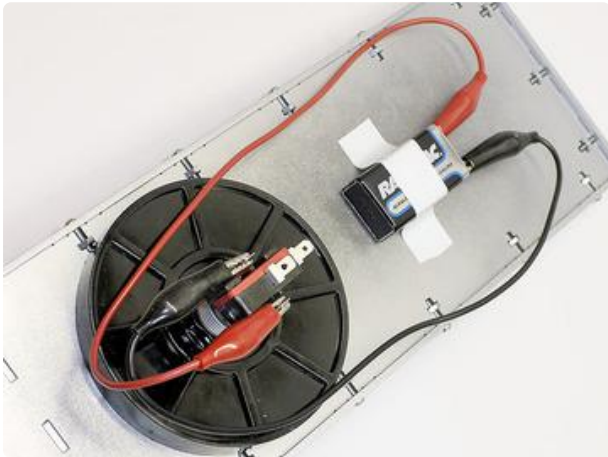


Face down. Confirm all of the edge screws are installed, straight and firm (but don't crank these down too hard — you'll crack the plastic).



Flipped over, a familiar face starts to emerge.

The button is then inserted from the front, secured in back with the included hardware.



Insert the microswitch if you haven't already, and confirm that the button "clicks."

The LED inside the button is already wired with a resistor for use up to 12 Volts. A 9 Volt battery (about \$3) works fine. This connects to the two side terminals next to the switch.

LEDs have a specific polarity. You may need to switch battery terminals to get it to light.

"I'm ready for my first lesson."



Tally: \$36.25 (battery included). The usual disclaimers apply.

So there you have it...a quick and inexpensive faux 9000-series computer. Not a perfect reproduction, but good enough to impress your geek friends, and it's hard to argue at that cost.

You could bolt it to the front of your tower PC gaming rig and call it done. Or with a bit of extra effort and expense we can take this to the next level and add some sound effects...

Adding Sounds

Adding electronics to the mix will boost the cost and complexity a fair bit, but we can still squeak by for just under \$100 — or about 20% the cost of the professionally-made reproductions. As with the prior steps, this figure implies one is already tooled up for soldering...with an iron, some bits of wire and a few additional minor parts and skills.

Our voice box is made around the Arduino Uno (\$30) and the Adafruit Wave Shield (\$22). The latter has [an entire build tutorial of its own \(https://adafru.it/cl8\)](https://adafru.it/cl8) — we won't reiterate that whole thing here — so please begin with that guide and work through the examples to make certain your soldering is good and that the board works. You

can test with headphones, but for the finished piece we're using an 8 Ohm, 1 Watt speaker from the shop (\$2).

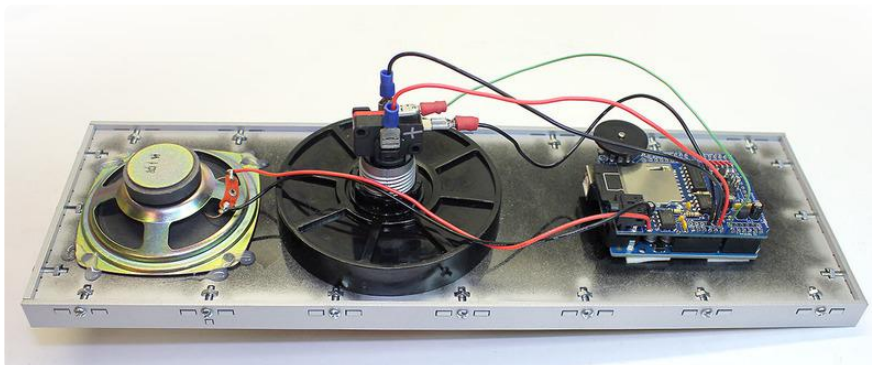
For power you can either use the same 9 Volt battery as before...connecting to the Arduino using a 9V Battery Clip w/5.5mm Plug (\$3)...or plug into the wall with a 9 Volt Switching Power Adapter (\$7).

Do not continue until you have a working Wave Shield. If you have the "Pi Speak" example working, that's a good sign and you can proceed.

We're not including any HAL 9000 sound files here, lest we run afoul of copyright issues. With just a bit of Google [searching](https://adafru.it/c9R) (<https://adafru.it/c9R>) you can find an ample supply already in WAV format. The volume level on some is a bit soft; free software like [Audacity](https://adafru.it/c9T) (<https://adafru.it/c9T>) can be used to boost this. Or you can watch the DVD and record some clips for yourself off the telly, the so called "analog loophole."

The WAV files are stored in the root directory of a FAT-formatted SD card. The card wasn't factored into the cost above, figuring it's likely that you (or someone you know) has spares around...the Wave Shield is a great way to reuse those feeble old 64 MB cards that are probably rattling around your camera bag.

Originally (as shown below) I affixed the speaker and Arduino to the back using hot-melt adhesive and double-stick foam tape, but have since modified the files to include mounting holes for #4-40 screws. If using screws, these should be inserted from the front before the faceplate and grille are installed.

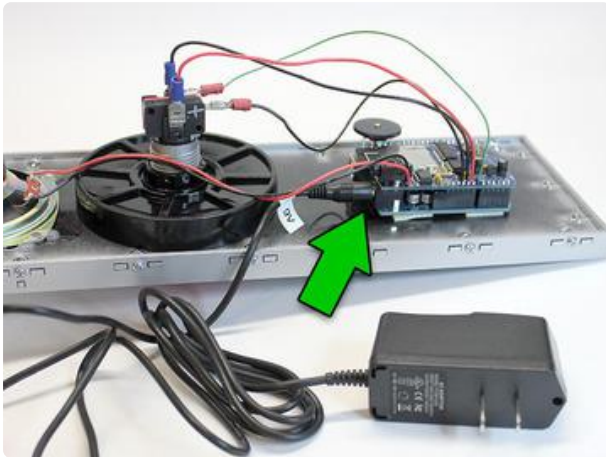


The Wave Shield includes solder holes adjacent to each Arduino header pin, allowing wires to be added. Remove the shield from the Arduino, insert wires from above and solder on the underside. The following connections are made:

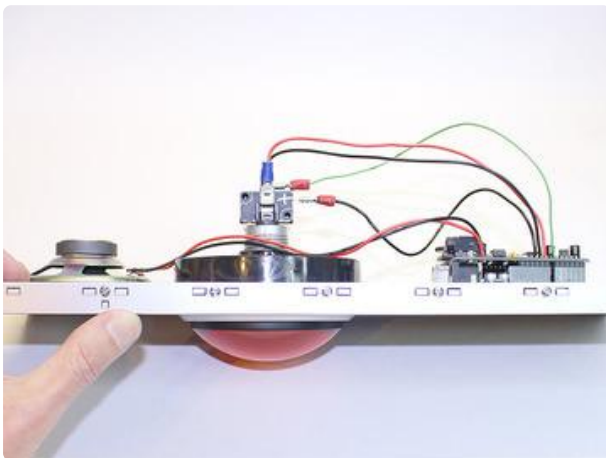
- The + side of the LED (in the button) connects to VIN on the Wave Shield / Arduino.
- The – side of the LED connects to one of the GND connections (there are two right next to VIN).

- One terminal of the switch (either one) connects to the other GND.
- The other switch terminal connects to Analog Pin 0 (along the same edge of the board as VIN, on the next header over).
- The speaker (two wires) connects to the outputs on the Wave Shield board, just behind the headphone jack.

I happened to use spade terminals on the wires, but there's no harm in soldering directly to the switch.



The 9V power source (whether battery or AC adapter) connects to the DC jack on the Arduino board.



If there's a downside to this prop, it would be that it's not exactly svelte. The parts hang almost 3 inches (8 cm) out the back, so you can't just stick it on a wall. It might go on another box (a tower PC gaming rig was previously mentioned)...or you'll be cutting holes in the wall if you plan to install it as a doorbell or such.

Here's the Arduino sketch for speaking random quotes when the "eye" is pressed. Remember, you need to have some WAV files in the root directory of an SD card. You'll want to edit the filename list to match what you have:

```
// Random HAL demo; adapted from PiSpeak sketch. When button
// on A0 is pressed, plays a random WAV file from a list.

#include <WaveHC.h>;
#include <WaveUtil.h>;

// REPLACE THESE WITH YOUR ACTUAL WAVE FILE NAMES:
// These should be at the root level, not in a folder.
static const char PROGMEM
  file00[] = "0.wav",
  file01[] = "1.wav",
  file02[] = "2.wav",
```

```

file03[] = "3.wav",
file04[] = "4.wav",
file05[] = "5.wav",
file06[] = "6.wav",
file07[] = "7.wav",
file08[] = "8.wav",
file09[] = "9.wav";
// If adding files above, include corresponding items here:
static const char * const filename[] PROGMEM = {
  file01, file01, file02, file03, file04,
  file05, file06, file07, file08, file09 };
// Sorry for the silliness, but this is how PROGMEM string
// arrays are handled.

#define error(msg) error_P(PSTR(msg))
SdReader card;
FatVolume vol;
FatReader root;
FatReader file;
WaveHC wave;
uint8_t debounce = 0, // Button debounce counter
prev = 255; // Index of last sound played

void setup() {
  Serial.begin(9600);
  if(!card.init()) error("Card init. failed!");
  if(!vol.init(card)) error("No partition!");
  if(!root.openRoot(vol)) error("Couldn't open dir");
  // PgmPrintln("Files found:");
  // root.ls();

  digitalWrite(A0, HIGH); // Enable pullup on button
  randomSeed(analogRead(A1)); // Randomize first sound
}

void loop() {
  if(digitalRead(A0) == HIGH) { // Button not pressed
    debounce = 0; // Reset debounce counter
    return; // and nothing else
  }

  if(++debounce >= 20) { // Debounced button press
    uint8_t n;
    char name[20];

    do { // Choose a random file...
      n = random(sizeof(filename) / sizeof(filename[0]));
    } while(n == prev); // ...but don't repeat last one

    prev = n; // Save file #
    debounce = 0; // Reset debounce counter
    strcpy_P(name, (char *)pgm_read_word(&filename[n])); // PROGMEM->RAM
    if(wave.isPlaying()) wave.stop(); // Stop WAV if playing

    if(!file.open(root, name)) {
      PgmPrint("Couldn't open file ");
      Serial.print(name);
      return;
    }
    if(!wave.create(file)) {
      PgmPrintln("Not a valid WAV");
      return;
    }

    wave.play(); // Start playing
    while(wave.isPlaying()); // Wait for completion
    sdErrorCheck(); // Check for error during play
    while(digitalRead(A0) == LOW); // Wait for button release
  }
}

```

```
}  
  
void error_P(const char *str) {  
  PgmPrint("Error: ");  
  SerialPrint_P(str);  
  sdErrorCheck();  
  for(;;);  
}  
  
void sdErrorCheck(void) {  
  if(!card.errorCode()) return;  
  PgmPrint("\r\nSD I/O error: ");  
  Serial.print(card.errorCode(), HEX);  
  PgmPrint(", ");  
  Serial.println(card.errorData(), HEX);  
  for(;;);  
}
```

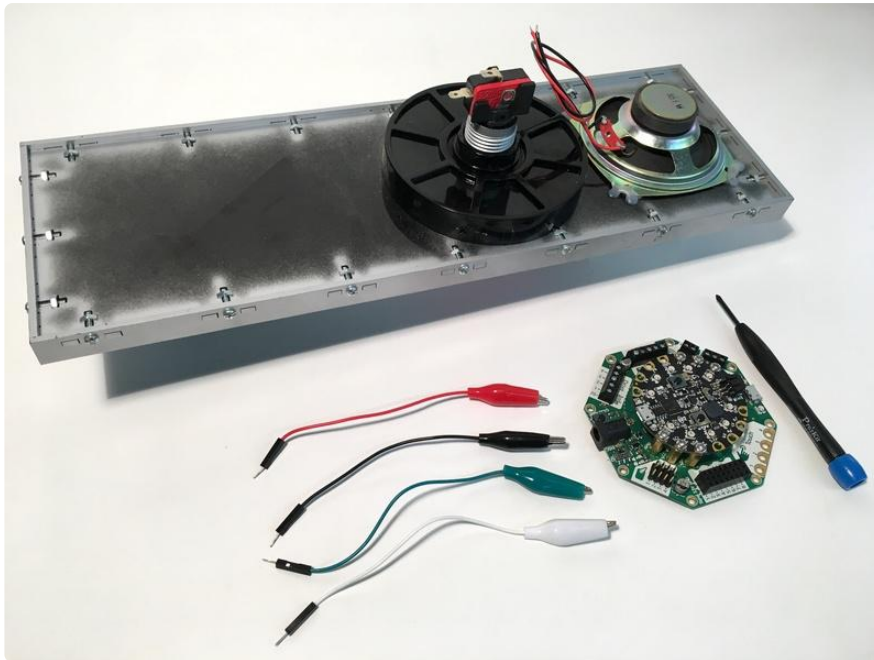
One of the cool bits here vs. a pre-made unit...being based around open source tools and hardware, you can totally hack and customize this. Instead of (or in addition to) the eye button, you could add a PIR motion sensor, or a magnetic switch so HAL speaks when you open the refrigerator door. With an Ethernet Shield you could make an internet-connected HAL that reports on the outside world (pending emails, etc.). The sky's the limit!

Updating with CRICKIT

If you'd prefer to opt for a [CRICKIT \(https://adafru.it/Biy\)](https://adafru.it/Biy) powered 9000 series rather than use the Wave Shield, this page will show you how!

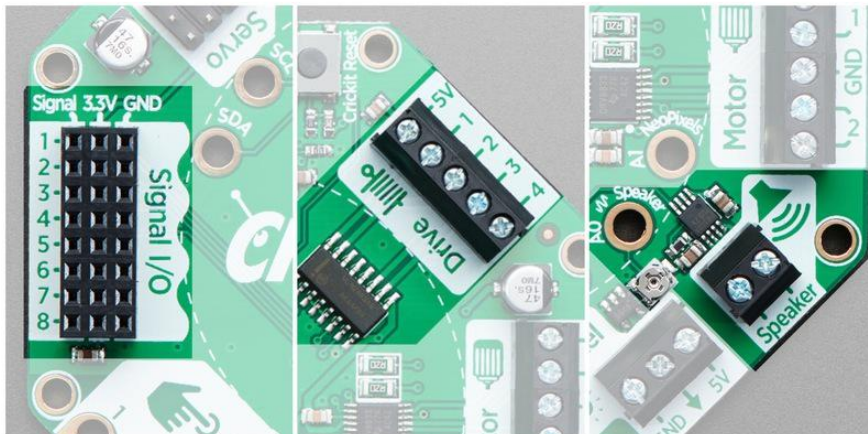
If this is your first time using CRICKIT, [follow this guide to get acquainted \(https://adafru.it/C2K\)](https://adafru.it/C2K).

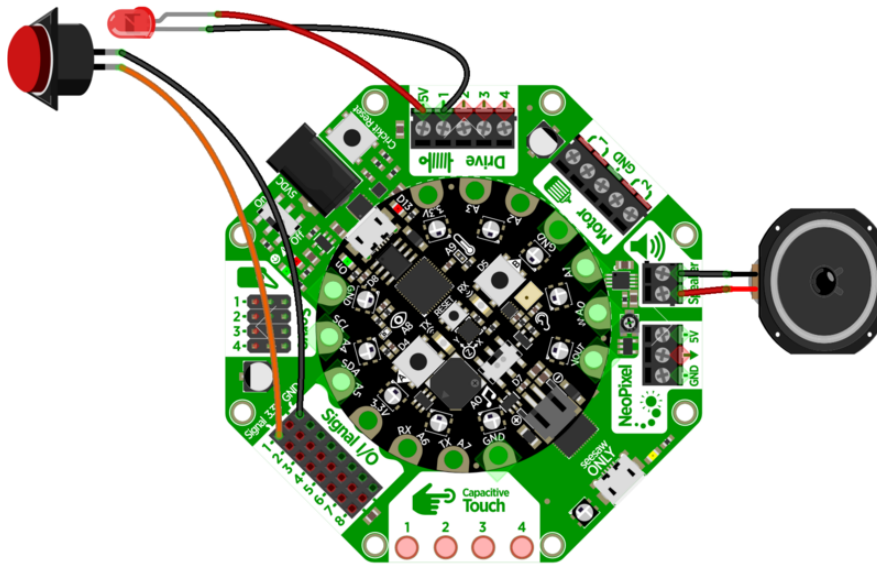
To get your CRICKIT and HAL talking, you will need some [alligator clip to male jumper wire \(https://adafru.it/xAV\)](https://adafru.it/xAV) connectors, a screwdriver, and a [micro USB cable \(https://adafru.it/iaa\)](https://adafru.it/iaa) (to upload code from your computer).



We will be taking advantage of CRICKIT's ability to multitask combined with the ease of CircuitPython.

The Signal port will be used to detect the button press, the Drive block to control the LED, and the Speaker block send audio signal to the speaker.





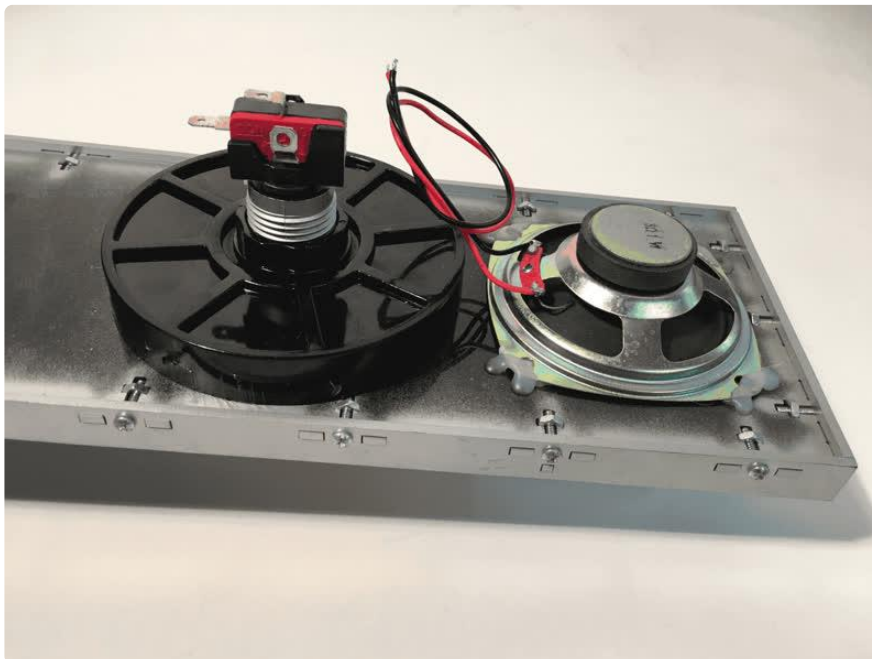
fritzing

HAL9000-crickit.fzz

<https://adafru.it/C32>

Attach the alligator clips to the button terminals as shown.

The two connections extending horizontally outwards detect a button press, while the two vertical terminals on the side are connected to the button's internal LED.

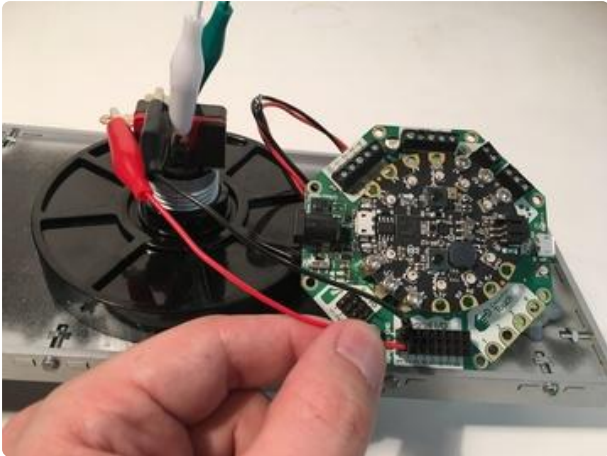




Connect button to CRICKIT

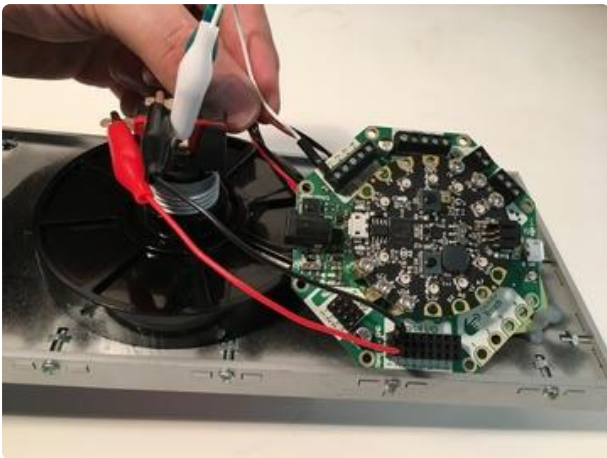
Now it's time to connect everything to your CRICKIT board.

Connect the red and black button wires to the Signal and GND ports.

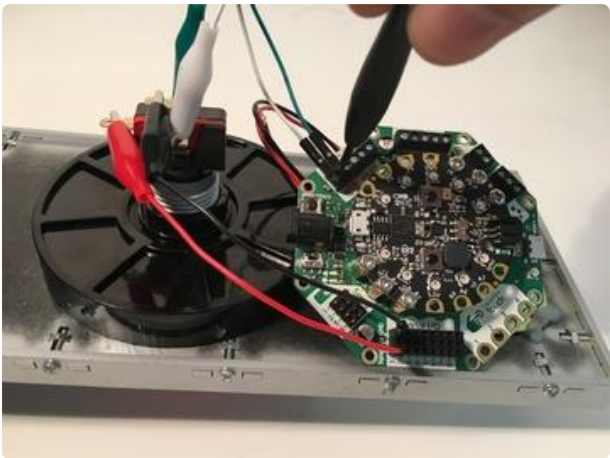
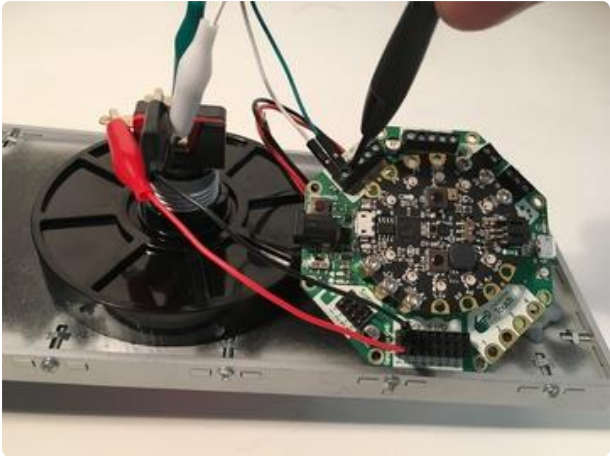
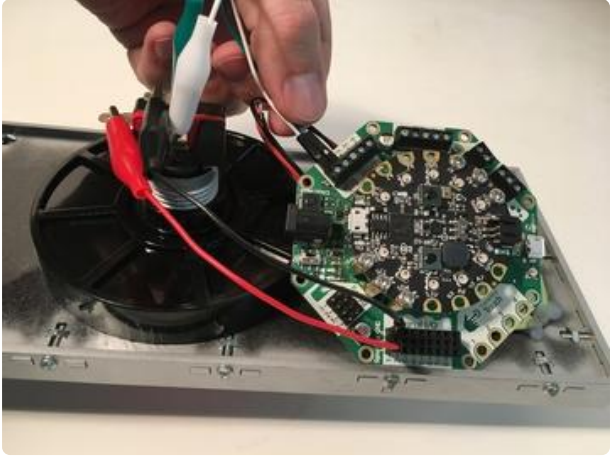


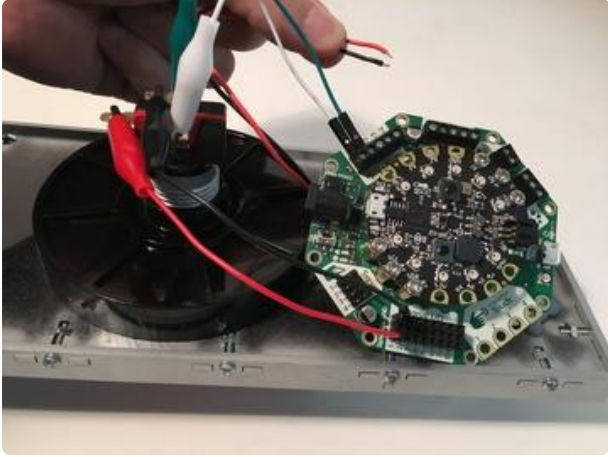
Connect the LED+ wire to Drive 5V

Connect the LED- wire to Drive #1.



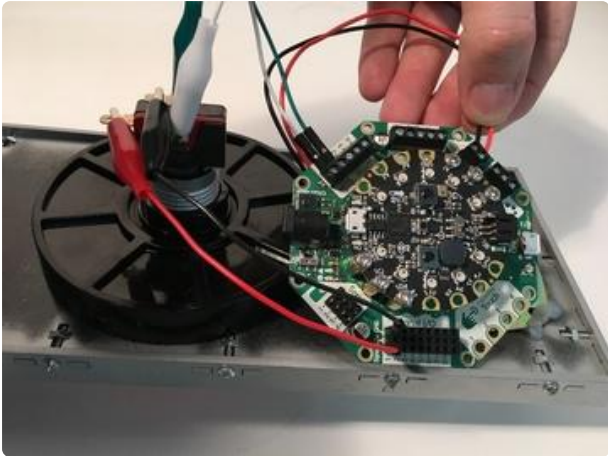
Use a screwdriver to tighten down the connections to the Drive terminal blocks.



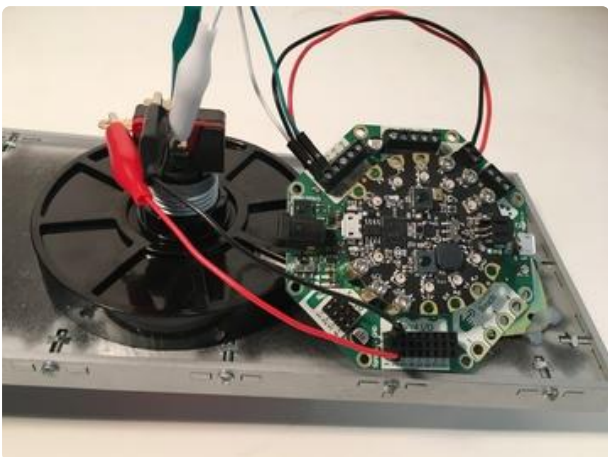
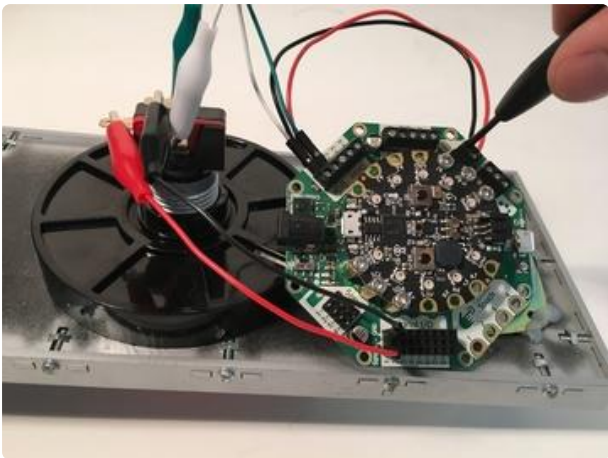


Connect speaker to CRICKIT

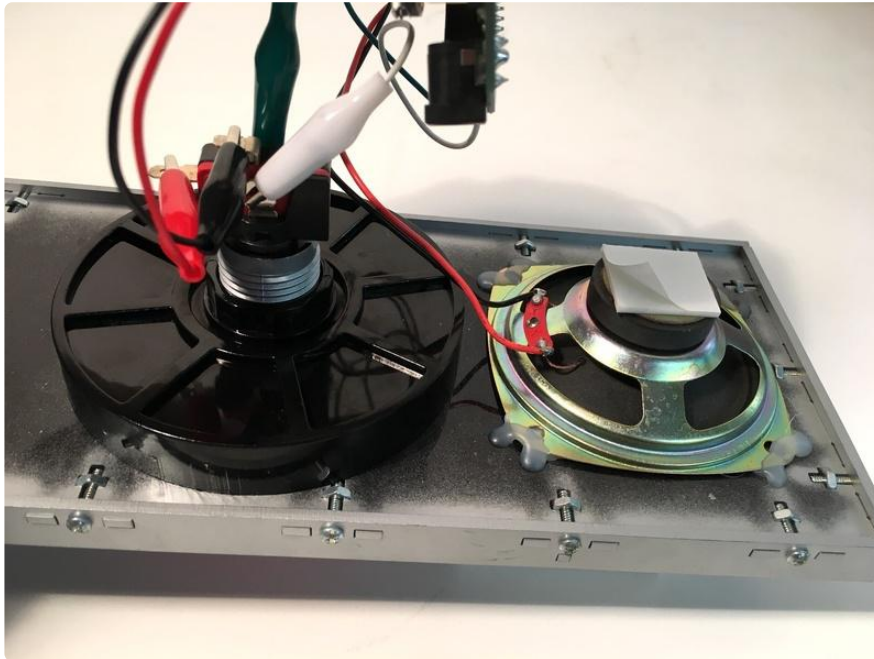
Take the red and black wires coming from the speaker and insert them into the Speaker terminal block.



Tighten both connections with a small screwdriver.



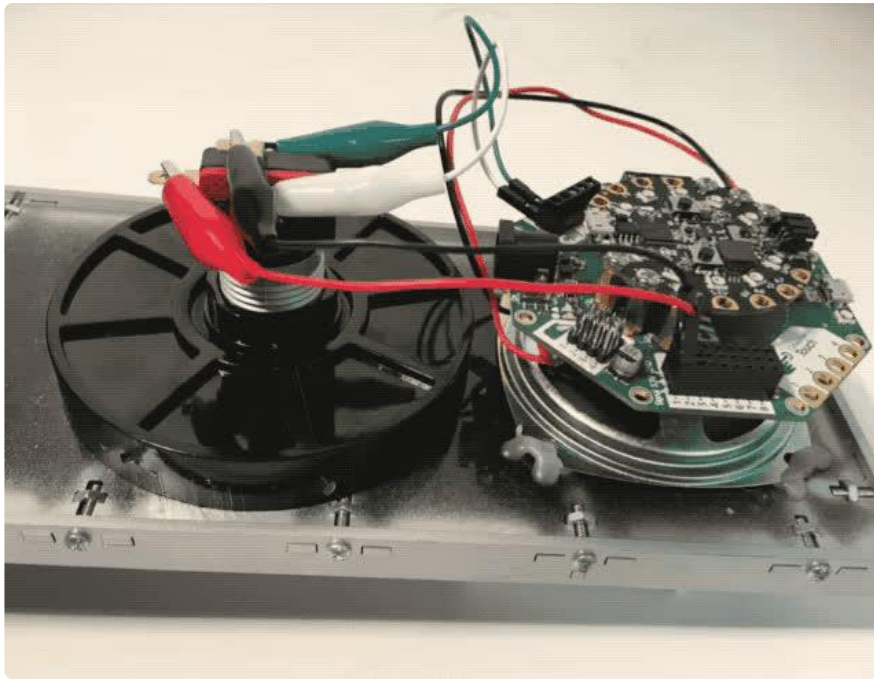
A small square of double sided foam tape will hold your CRICKIT in place on the back of the speaker.



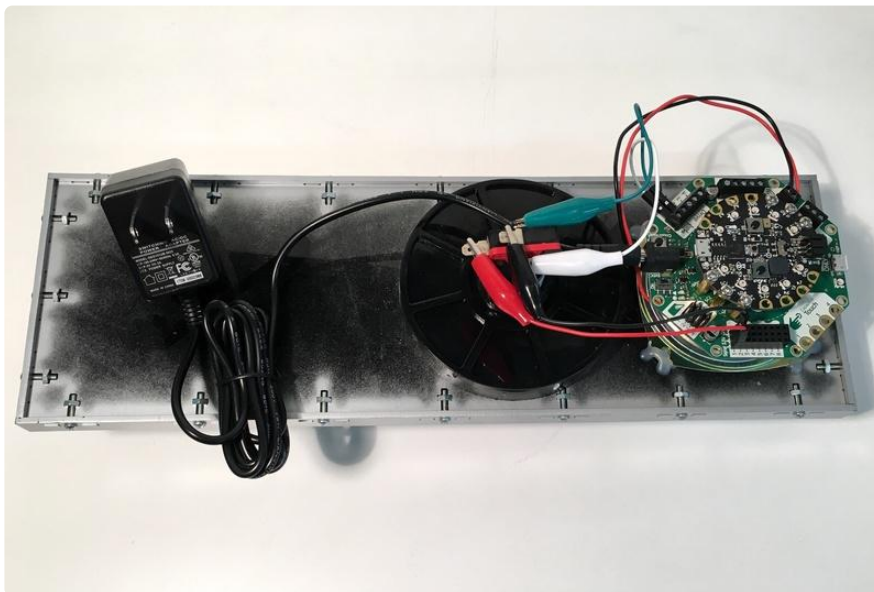
Connect 5V DC power to CRICKIT



Make sure you're using a 5 volt power supply for CRICKIT!



Once everything is connected you're ready for some code!



Getting Familiar with Mu

We will be using CircuitPython code, which is easiest to work with in the Mu Editor. If you haven't previously used Mu, [this guide will get you started \(https://adafru.it/ANO\)](https://adafru.it/ANO).

If you haven't used Circuit Playground Express with CRICKIT before, make sure you've updated it with the latest special 'seesaw' version of the CPX firmware. [This guide will show you how \(https://adafru.it/Bfh\)](https://adafru.it/Bfh).

Copy and paste the code below in your Mu editor window.

```
import os
import random
import time
import board
import audioio
import audiocore
from adafruit_crickit import crickit

# Hal button-and-voice example

# Button connected to Signal pin #1 & ground:
BUTTON = crickit.SIGNAL1
crickit.seesaw.pin_mode(BUTTON, crickit.seesaw.INPUT_PULLUP)

# LED connected to 5V & Drive pin #1:
LED = crickit.drive_1
LED.duty_cycle = 65535

# Find all Wave files in CIRCUITPY storage:
WAVEFILES = [file for file in os.listdir("/")
              if (file.endswith(".wav") and not file.startswith("_."))]
print("Audio files found:", WAVEFILES)

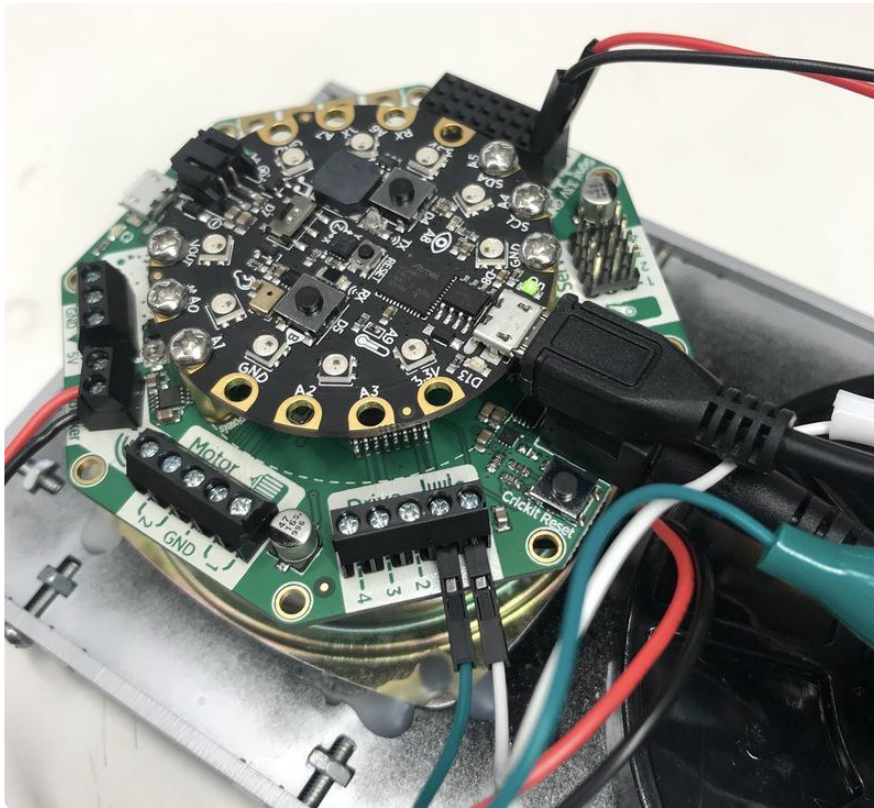
# Audio playback object:
AUDIO = audioio.AudioOut(board.A0)

# Function to play a wave file in its entirety:
def play_file(wavfile):
    print("Playing", wavfile)
    with open(wavfile, "rb") as f:
        wav = audiocore.WaveFile(f)
        AUDIO.play(wav)
        while AUDIO.playing:
            LED.duty_cycle = random.randint(5000, 30000)
            time.sleep(0.1)
    LED.duty_cycle = 65535

while True:
    if not crickit.seesaw.digital_read(BUTTON):
        # Play a random wave file
        play_file(random.choice(WAVEFILES))
        # Then wait for button to be released
        while not crickit.seesaw.digital_read(BUTTON):
            continue
```

Use a micro USB cable to connect to the port on the Circuit Playground Express mounted on top of CRICKIT. You should see "CIRCUITPY" show up as a new device on your computer.

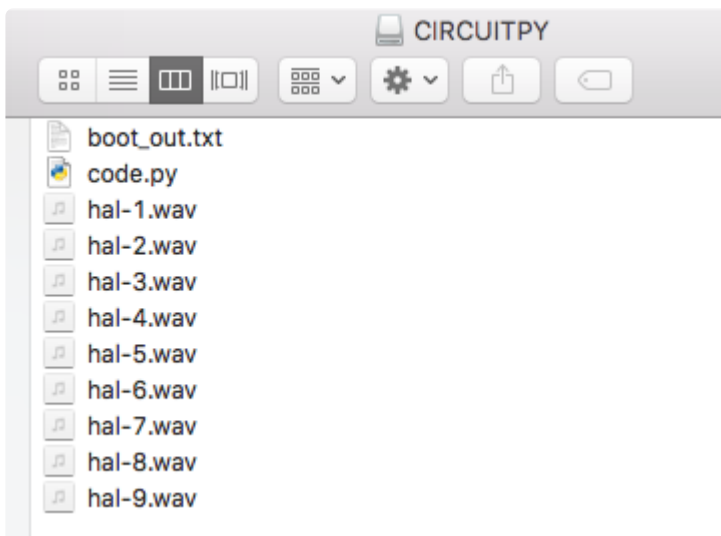
Save this code to CIRCUITPY as a file named "code.py".



Audio files for HAL 9000 are not included here, lest we run afoul of copyright issues, but with just a bit of Google [searching](https://adafru.it/c9R) (https://adafru.it/c9R) you can find an ample supply already in WAV format.

Once you have WAV files you're happy with, you'll want to drag and drop them directly onto your CIRCUITPY drive.

When all is said and done your CIRCUITPY drive should look something like this.



It's OK if the file names for your .wav files are different from those in this image (the code is only looking for a specific file type, the names of the files themselves don't matter).

HAL speaks!

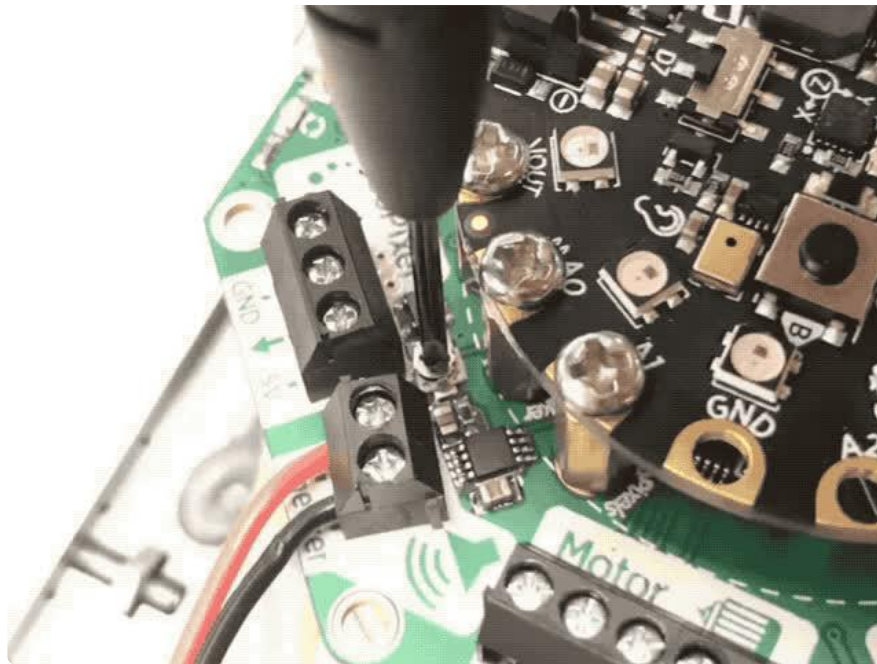
Connect CRICKIT to power and you should see the red button illuminate. Give the button a press and you'll be greeted with a random HAL 9000 quote.



Controlling volume

If you find HAL's voice is too loud or too quiet you can easily adjust this on the fly without the need for changing the code.

Use the small potentiometer on CRICKIT to adjust the speaker's volume up or down.



X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Adafruit Accessories](#) category:

Click to view products by [Adafruit](#) manufacturer:

Other Similar products are found below :

[3209](#) [3561](#) [3560](#) [3562](#) [2503](#) [3011](#) [3048](#) [2973](#) [2868](#) [2958](#) [2836](#) [2968](#) [3610](#) [3568](#) [3551](#) [2858](#) [3353](#) [3584](#) [3484](#) [2865](#) [3556](#) [3559](#) [3262](#)
[3348](#) [2499](#) [2878](#) [2963](#) [3219](#) [3005](#) [2882](#) [PGM1202](#) [02-LDR1](#) [02-LDR12](#) [02-LDR13](#) [02-LDR14](#) [02-LDR15](#) [02-LDR2](#) [02-LDR3](#) [02-LDR4](#)
[2194](#) [862](#) [460](#) [905](#) [02-LDR20](#) [02-LDR21](#) [02-LDR22](#) [02-LDR23](#) [1008](#) [1020](#) [1031](#)