



Synchronized Eyes with Two HalloWings

Created by Phillip Burgess



<https://learn.adafruit.com/synchronized-eyes-with-two-hallowings>

Last updated on 2024-06-03 02:31:56 PM EDT

Table of Contents

Overview	3
Wiring	3
• One More Jumper	
Software — Easy Recipes	6
• Installing	
• If it doesn't work:	
• Attach HalloWing Boards to Silhouette	
• Cat	
• Owl	
• Nauga	
• Deer	
Software — Customizing	11
• More Control	

Overview



The Adafruit HalloWing board ships with a spooky eyeball demo pre-installed by default. But what if you want two eyeballs peeking around?

Some prior projects have you soldering up two graphic displays and either a [microcontroller \(https://adafru.it/j6D\)](https://adafru.it/j6D) or [Raspberry Pi \(https://adafru.it/zrC\)](https://adafru.it/zrC)...but neither was “HalloWing simple.”

Now we’ve cooked up a way for two HalloWing eyes to sync up and **move together...** and it requires **no soldering at all!** And with our ready-made designs, you might not even need to touch a code editor.

The eyes are great for things like Halloween, as part of a jack-o’-lantern or a window decoration (we even have some templates you can print and use)...or for cosplay, where they might go on a hat or be part of a companion prop.

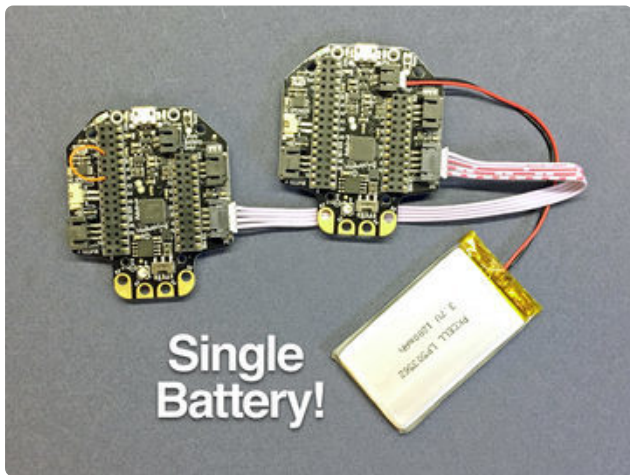
Wiring

There’s a couple ways to physically link the two boards. One’s quite tidy, the other may be easier with stuff you already have around.



This 4-pin JST STEMMA cable (<http://adafru.it/3568>) connects between the I2C sockets on both HalloWing boards. Click, click, done!

If you go this route, both boards will **share one power source**...you only need a battery or USB cable to **ONE** of the HalloWing boards, the other will receive power through the STEMMA cable. In fact, **use of two batteries is a bad idea in this setup**, since they would be connected and the voltages may be slightly different.



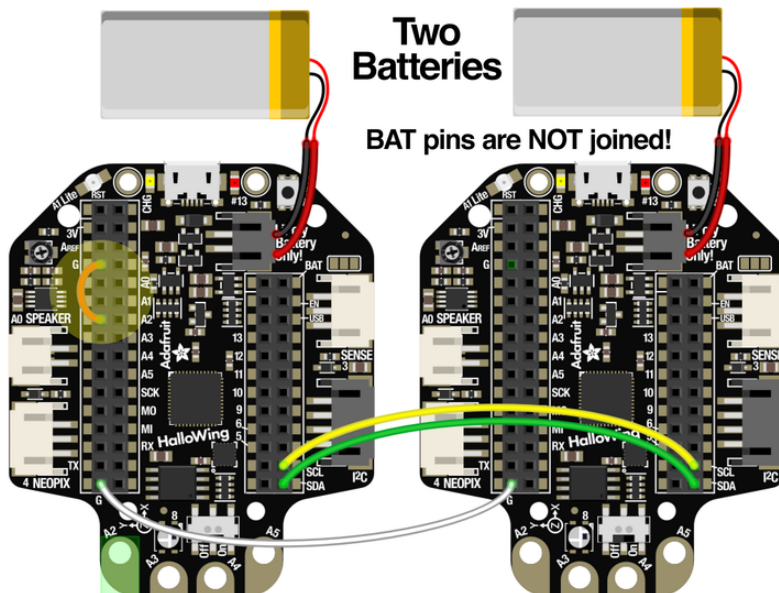
But two boards means half the runtime from the single battery...you may want to switch to something bigger than the 400 mAh battery included in Adabox 009, if that's what you've got. Or leave off the battery and use a **USB cable** to connect one of the HalloWings to a **USB battery bank**, the kind used for charging phones. That should run for **hours**!

The **alternative** is to use a few **jumper wires** to connect the two boards. These can be inserted into the FeatherWing socket pins on the back of the board.

Connect the **same** pins across **both** boards:

- GND (there are two GND pins, either one will work)
- SDA
- SCL

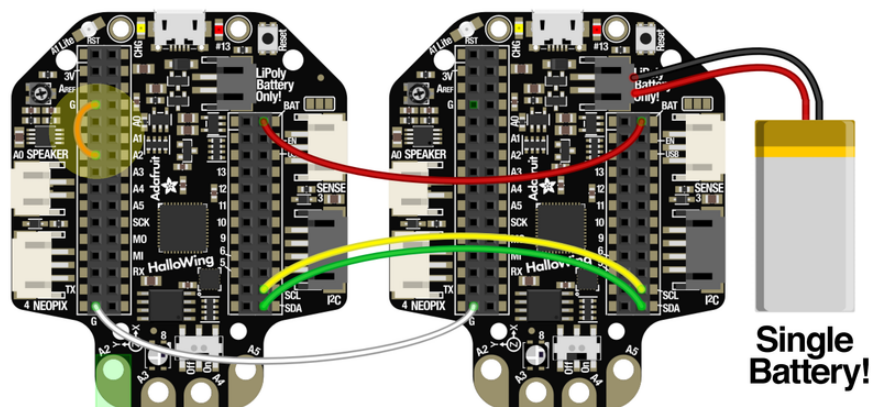
Connect SDA on one board to SDA on the other board, SCL to SCL and so forth. You should **not** cross the wires; this is not a “null modem” connection.



Wired this way, each board can safely use its own separate battery. But if you'd prefer a single power source as with the STEMMA cable technique mentioned above, add one more jumper wire between the two boards:

- BAT

Do NOT make this connection if using two batteries!



Remember with the single-battery approach that run time is halved. Consider using a USB power bank!

One More Jumper

NOW: in either case, whether using the STEMMA cable or jumper wires, ONE of the two HalloWing boards needs to control the show (while the other follows along).

Connect a jumper wire between a **GND** pin (there are two, use either one) and **A2**. Do this on only **one** of the boards. This one will now track the other when **both** are switched on.

In the diagrams above, we've added the wire to the left board (it's highlighted in yellow)...but it can go on either one. But not both!

Software — Easy Recipes

We've put together a few drag-and-drop “recipes” for this project that **don't require any programming** know-how. More experienced users can see the Customizing page for insights about getting into the code and **fine-tuning various details**.

Also we've created some Halloween-themed **cut-out shapes** you can use. Of course you can just **freehand** a design, or there's tons of freely-available **clip art** elsewhere on the web...but since these are our own, we can legitimately **share** them! You can print them out and trace onto cardboard or foam-core, then cut out with a hobby knife...or maybe you have one of those fancy vinyl cutters that also handles construction paper. Use whatever tools and materials you're comfortable with, it's all good! When finished, the silhouettes can be taped in a window, or use a hot glue gun to add a “kickstand” in the back, like a photo frame, or a dowel or tomato stake to poke it in the ground.

If designing your own character: the eye holes should be circles about **26 mm** in diameter (a smidgen over 1 inch), spaced about **56 mm** (2¼") apart center-to-center.

These particular examples **do not use the light sensor**, since the HalloWing board will most likely be situated just behind an opaque layer of cardboard or similar. But if you poke a hole in your design to keep the light sensor exposed, and if you make the necessary changes to the code, it's an option. Otherwise the pupils do their own thing.

Installing

Plug one **HalloWing** board into your computer with a **USB** cable. Make sure the power switch is set to the “**on**” position, then **double-click the RESET button** on the back of the board.

After a moment, a small flash drive called **HALLOBOOT** should appear on your system. Drag-and-drop one of the **.UF2 files** (downloadable below) on to this flash

drive. There will be a few seconds of LED flashing, then the HALLOBOOT drive will be ejected.

Unplug the first HalloWing board and connect the USB cable to the **second** HalloWing, repeating the process: power switch on, double-click reset, wait for the HALLOBOOT drive and drag-and-drop a .UF2 file there.

With the same software on both boards now...if they're properly wired and powered on, you should see a pair of blinky eyes!

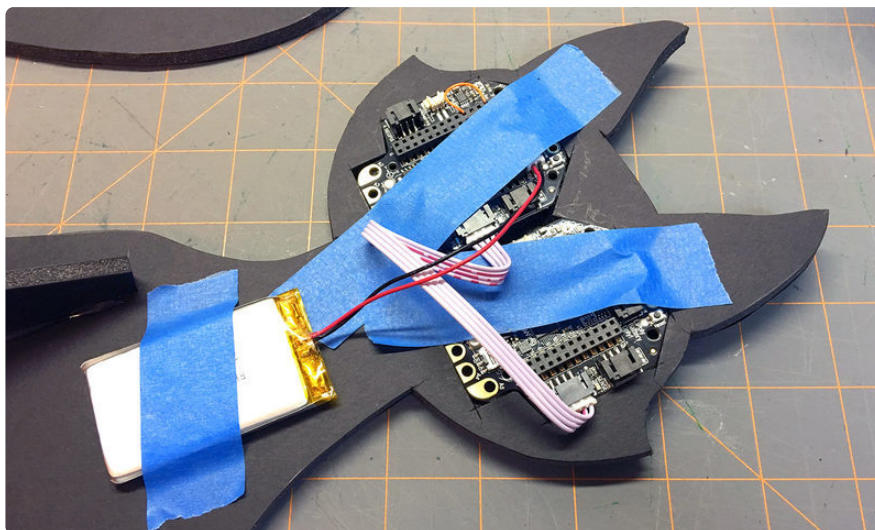
If it doesn't work:

- Confirm BOTH boards are switched on.
- Check your wiring against the diagrams on the prior page.
- Make sure ONE board has a jumper between A2 and GND.

(The candle effect seen in the animations below is not part of this project...those are separate Circuit Playground Express boards running code from [our Jack-o'-Lantern guide](https://adafruit.it/CpQ) (<https://adafruit.it/CpQ>).)

Attach HalloWing Boards to Silhouette

Like a Hollywood building façade, nobody's gonna see this side of things and it's okay to take short cuts. Line up the eyes with the eye cutouts and hold the boards in place with some masking tape. For something a little more durable, a few dabs of hot glue can work, and is easily removed later using a cotton swab dipped in rubbing alcohol.



Cat



A Halloween classic! Two bright yellow eyes with slit pupils. Unlike the original default eye which is a bit too creepy and real, this instead uses a 2D “flat” color, like a cartoon.

2EYES-CAT.UF2

<https://adafru.it/CRX>

The cut-out for this one (PDF file) is a larger design that won't fit on a single letter-size page...so maybe you can “tile” it across two pages and tape them together.

Silhouette-Cat.pdf

<https://adafru.it/CRY>

Owl



The other Halloween classic! Again, this one uses a 2D, cartoon-like look, with blue irises in case you'd like to make a [Minerva](https://adafru.it/CRZ) (https://adafru.it/CRZ) owl rather than a generic Halloween owl.

2EYES-OWL.UF2

<https://adafru.it/CR->

This one's a simple one-page PDF. You can dress up the silhouette with a paint pen (as shown above), or cut and glue shapes from colored construction paper or fabric scraps to give it more variety.

Silhouette-Owl.pdf

<https://adafru.it/CS0>

Nauga



Wait, what? Okay, so Naugas have nothing to do with Halloween...but resemble something Tim Burton would stick in his films.

The Nauga is a mythical species from the 1960s that served as a mascot for Uniroyal's Naugahyde material (a brand of artificial leather).

Simplest eyes of all...white with round black pupil, no iris. Just like the original Nauga.

2EYES-NAUGA.UF2

<https://adafru.it/CS1>

Another one-page PDF. This might be just a tiny bit too wide for some printers...the edges of the arms might be clipped off...but it's very little, and you can just freehand those lines.

As with the owl, this can be dressed up with paint pens, or get all fancy with different-colored paper cutouts, or suitably vintage tacky fabric scraps.

Silhouette-Nauga.pdf

<https://adafru.it/CS2>

Deer

Something a bit more Christmas-themed. Cartoon deer eyes with huge specular highlights:

2EYES-DEER.UF2

<https://adafru.it/Db6>

Software — Customizing

Customizing the eye features isn't as drag-and-drop simple...you'll need some familiarity editing code in the Arduino IDE, installing libraries, compiling and uploading software to the HalloWing board and so forth.

Our “[Electronic Animated Eyes using Teensy 3.1/3.2 \(https://adafru.it/j6B\)](https://adafru.it/j6B)” guide has a link to the source code and some insights on modifying the configuration. It also explains prerequisite libraries. Here we're focusing just on how to get two HalloWing boards talking to each other (which is not enabled in that code by default)...

In the file config.h, you'll see a section that looks like this (starting around line 50):

```
// DISPLAY HARDWARE SETTINGS (screen type & connections) -----
#ifdef ADAFRUIT_HALLOWING
  #include <Adafruit_ST7735.h> // TFT display library
  #define DISPLAY_DC      38 // Display data/command pin
  #define DISPLAY_RESET   37 // Display reset pin
  #define DISPLAY_BACKLIGHT 7
  #define BACKLIGHT_MAX   128
  // #define SYNCPIN      A2 // I2C sync if set, GND this pin on receiver
  // #define SYNCADDR     0x08 // I2C address of receiver
  // (Try disabling SYMMETRICAL_EYELID then)
#else
  // Enable ONE of these #includes to specify the display type being used
  #include <Adafruit_SSD1351.h> // OLED display library -OR-
  // #include <Adafruit_ST7735.h> // TFT display library (enable one only)
  #define DISPLAY_DC      7 // Data/command pin for ALL displays
  #define DISPLAY_RESET   8 // Reset pin for ALL displays
#endif
```

Un-comment the two SYNC lines (remove the leading “//”), like so:

```
#define SYNCPIN      A2 // I2C sync if set, GND this pin on receiver
#define SYNCADDR     0x08 // I2C address of receiver
```

Also, if using one of the cardboard cutouts and you haven't made an opening for the light sensor, comment out the **LIGHT_PIN** line so the pupils will change size on their own:

```
//#define LIGHT_PIN      A1 // Hallowing light sensor pin
```

OPTIONAL: since we'll have distinct left and right eyes now, you might want to enable the part of the code that gives the eyelids a more interesting shape. That's near the top of the code, starting around line 6:

```
// GRAPHICS SETTINGS (appearance of eye) -----  
  
// If using a SINGLE EYE, you might want this next line enabled, which  
// uses a simpler "football-shaped" eye that's left/right symmetrical.  
// Default shape includes the caruncle, creating distinct left/right eyes.  
#ifdef ADAFRUIT HALLOWING // Hallowing, with one eye, does this by default  
  #define SYMMETRICAL_EYELID  
#else  
  // Otherwise your choice, standard is asymmetrical  
  //#define SYMMETRICAL_EYELID  
#endif
```

Comment out the first `SYMMETRICAL_EYELID` definition like this:

```
//#define SYMMETRICAL_EYELID
```

While you're in there, you can select one of the alternate eye designs if you like (the `#include` lines near the top).

Now compile and upload the code to **both boards**.

More Control

If you want to create completely custom eyes...not just tweaking some settings like we've done here...the [Electronic Animated Eyes \(https://adafru.it/j6B\)](https://adafru.it/j6B) guide explains how to delve deeper into this project and use your own images. This requires some additional skills including image editing and running Python scripts from the command line.