HalloWing Badge Image Viewer

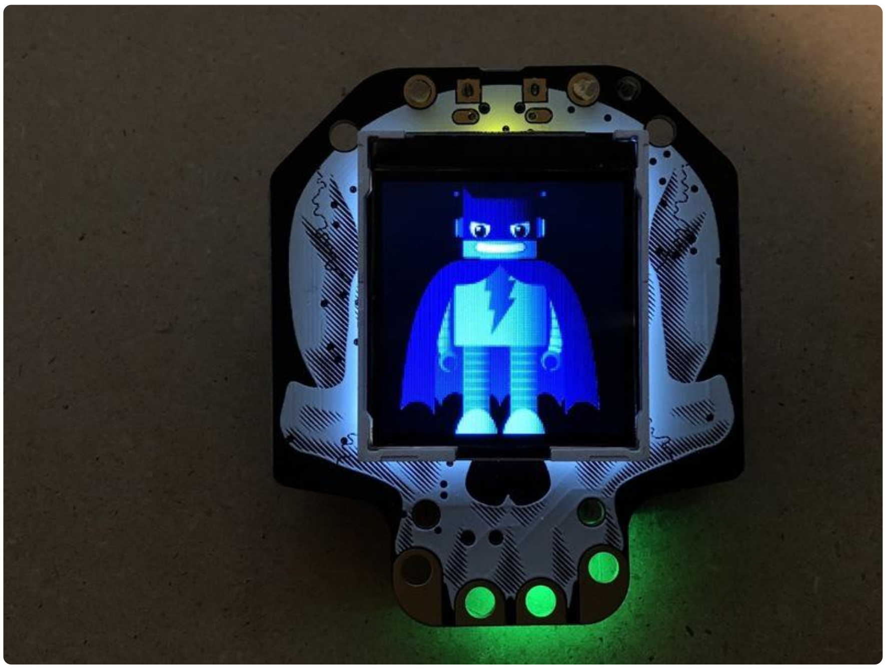
Created by John Park

https://learn.adafruit.com/hallowing-badge

Last updated on 2023-08-29 03:51:49 PM EDT

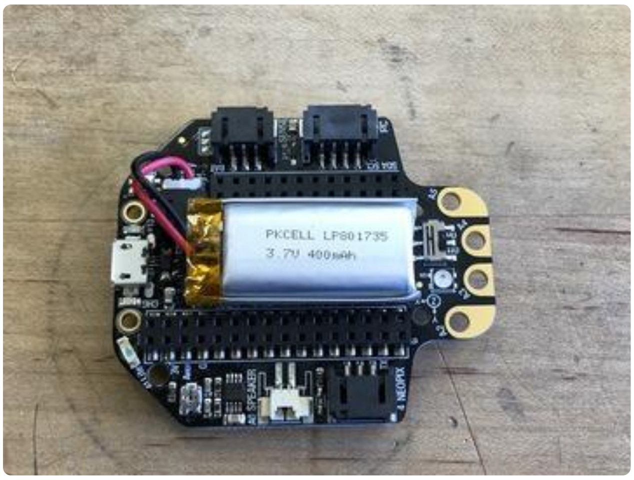
Table of Contents

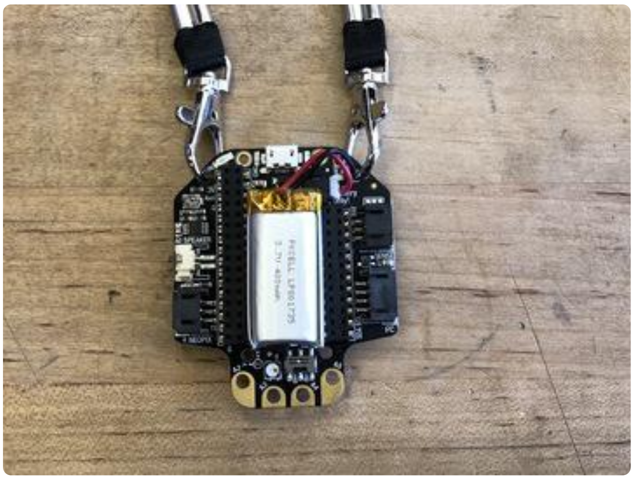
HalloWing Badge Image Viewer



This mini guide will show you how to display graphics on the HalloWing's TFT screen using CircuitPython. This can be a lot of fun when paired with the lanyard if you'd like to make your own personalized conference badge or "HELLO MY NAME IS:" sign!

Or, use the HalloWing as a little photo wallet to show off pictures of your kids/pets/cosplay!





Player Prep

Plug a small LiPoly battery into the battery JST port on the HalloWing.

You can use a small bit of double-stick foam tape to secure the battery to the HalloWing.

Attach a lanyard to two of the mounting holes on the HalloWing.

If you want some fun Halloween-themed Adafruit characters to use, download this .zip file, uncompress it, and then drag the .bmp files to your HalloWing!

Adafruit_Halloween_BMPs.zip

Tip: Images inside of a folder on your HalloWing won't be seen by this image player code below, only files at the root level of the HalloWing are seen. So, you can swap files in and out of folders to quickly change your image set!

Image Player CircuitPython Firmware

Download the latest version of CircuitPython for the HalloWing specifically for displaying graphics files easily by clicking the green button below. Adafruit recommends the version 4.0 or higher.

To load CircuitPython onto the HalloWing, plug in the HalloWing over USB to your computer and then double-click the reset button on the HalloWing. This will put it into bootloader mode, allowing you to change the firmware. You'll see a USB drive appear called HALLOWBOOT.

Drag the .uf2 file you downloaded onto the HALLOWBOOT drive. Once it copies over, HallowWing will automatically restart and appear as a flash drive named CIRCUITPY, and will be ready for you to load a CircuitPython program that can take advantage of the display capabilities.

Download the latest version of CircuitPython for this board via CircuitPython.org

File Format

To prepare your graphics for the HalloWing, you'll need to size them to 128 x 128 pixels. You can do this in nearly any graphics program, such as Photoshop, GIMP, Preview on osx, ImageMagik, MS Paint, you name it!

Save your file as a 16-bit .bmp file. Here you can see some example graphics.

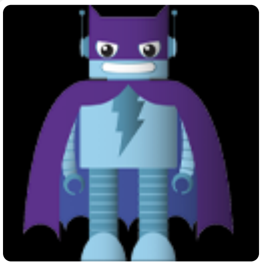
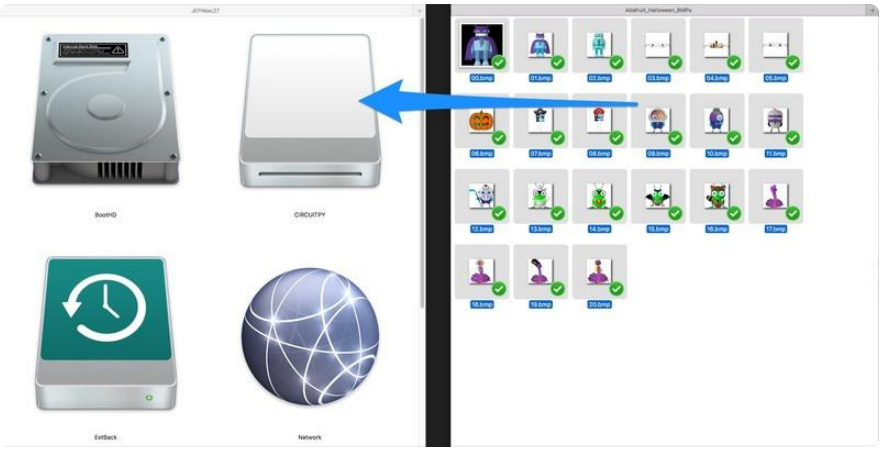


Image File Storage

Simply plug in your HalloWing over USB to your computer, and it will show up as a drive named CIRCUITPY.

Drag your .bmp image files onto the CIRCUITPY drive.



This is what the contents of your HalloWing's CIRCUITPY drive will look like once the images have been copied over.

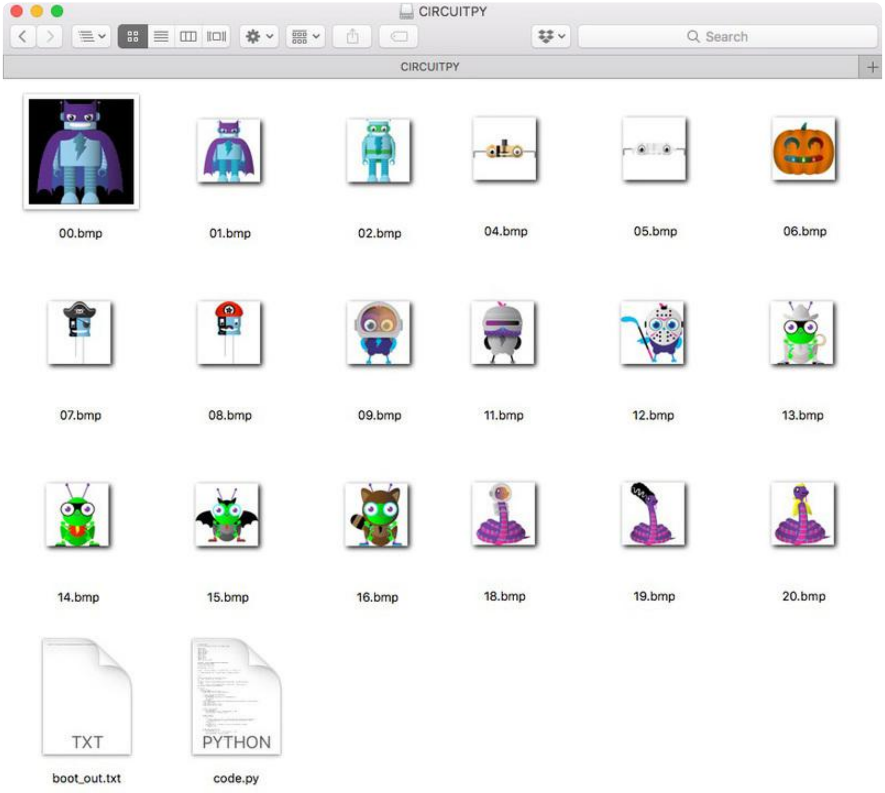


Image Player Code

This program lets you use the HalloWing as a simple image viewer. It will open the first image it finds on the flash storage, fading up its brightness.

Adafruit really likes using the Mu editor to edit the CircuitPython code. See this guide on loading and using Mu ().

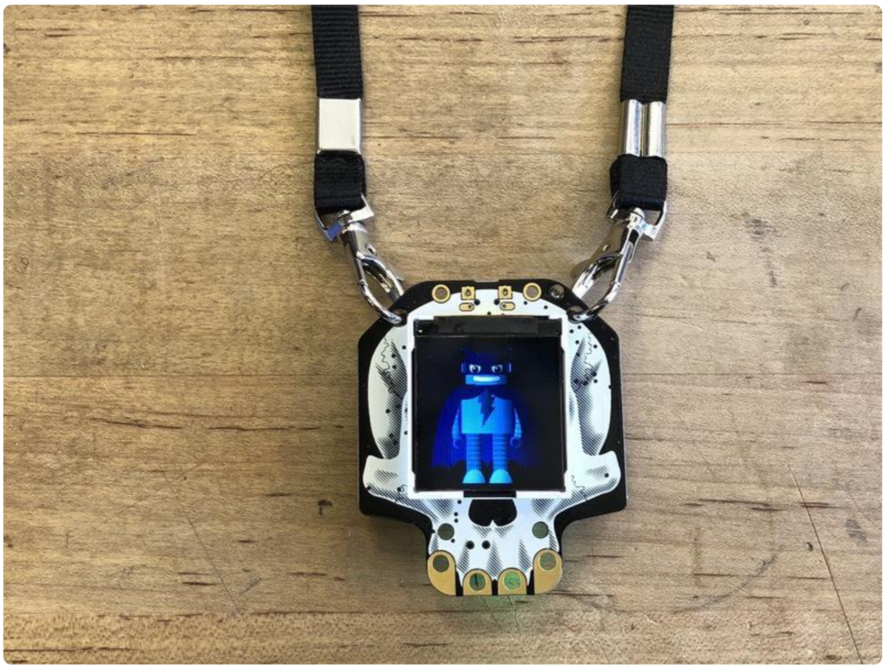
You can then advance to the next image or go back to the previous image by pressing the outer two "teeth" of the HalloWing. These are capacitive touch pads TOUCH4 and TOUCH1.

You can also adjust the brightness down and up respectively with TOUCH3 and TOUCH2.

Copy the code from below and paste it into Mu. Then, save it at code.py to your HalloWing.

Backlight code is temporarily disabled while some adjustments are being made.

```
import board
import displayio
import time
import pulseio
import touchio
import os

forward_button = touchio.TouchIn(board.TOUCH4)
back_button = touchio.TouchIn(board.TOUCH1)

brightness_up = touchio.TouchIn(board.TOUCH3)
brightness_down = touchio.TouchIn(board.TOUCH2)

#backlight = pulseio.PWMOut(board.TFT_BACKLIGHT)
splash = displayio.Group()
board.DISPLAY.show(splash)

max_brightness = 2 ** 15

images = list(filter(lambda x: x.endswith("bmp"), os.listdir("/")))
i = 0
while True:
    forward = False
    backward = False

    with open(images[i], "rb") as f:
        try:
            odb = displayio.OnDiskBitmap(f)
        except ValueError:
            print("Image unsupported {}".format(images[i]))
            del images[i]
            continue
        face = displayio.TileGrid(odb, pixel_shader=displayio.ColorConverter(),
x=0, y=0)
        splash.append(face)
        # Wait for the image to load.
        board.DISPLAY.wait_for_frame()

        # Fade up the backlight
        for b in range(100):
            #backlight.duty_cycle = b * max_brightness // 100
            time.sleep(0.01)

        # Wait forever
        while not forward and not backward:
            forward = forward_button.value
            backward = back_button.value
            if brightness_up.value:
                max_brightness += 16
            elif brightness_down.value:
                max_brightness -= 16
            if max_brightness &lt; 0:
                max_brightness = 0
            elif max_brightness &gt;= 2 ** 16:
                max_brightness = 2 ** 16 - 1
            #backlight.duty_cycle = max_brightness

        # Fade down the backlight
        for b in range(50, -1, -1):
            #backlight.duty_cycle = b * max_brightness // 100
            time.sleep(0.005)

        splash.pop()

        if forward:
            i += 1
        elif backward:
            i -= 1
        i %= len(images)
```



Now, you can try your hand at adjusting the code to do other things! How about a random image player for use with the lanyard as a badge? What about four images each assigned to a different touch button? You can do all sorts of nifty things with the HalloWing display!

Want to push the board even harder with image display? Check out the Spirit Board guide https://learn.adafruit.com/hallowing-spirit-board