

This document is also available on the Open Book Wiki:

<https://github.com/joeycastillo/The-Open-Book/wiki/E-Book-Wing-Setup-Guide>

E-Book Wing Setup Guide

This is a guide to setting up the E-Book Feather Wing for the first time. It assumes that you have populated the E-Book Wing board and soldered all the components into place. You should have the following items on hand:

- E-Book Feather Wing
- A Feather M4 Express
- A MicroSD Card
- A MicroSD Card Reader
- Solder and a Soldering Iron

This page will walk through some tasks required to set up your development environment, as well as some tasks required to set up the hardware itself.

Setting up your development environment

We will be working with the Arduino IDE (<https://www.arduino.cc>) for this project; if you do not already have Arduino on your machine, download and install it now.

You will also need to have Adafruit SAM board support installed. If you do not already have this board support package installed, you can follow these instructions to get this set up:

<https://learn.adafruit.com/add-boards-arduino-v164/setup>

You will need several libraries installed. Most we will install from the Arduino Library Manager.

- **Adafruit GFX Library**
- **Adafruit BusIO**
- **Adafruit MCP23008 Library**
- **Adafruit EPD**
- **Adafruit SPIFlash**
- **SDFat - Adafruit Fork**
- **arduino-menusystem** (optional, needed for MVBook at the end)

Finally, you will need to install the following two libraries by copying them to your Arduino libraries folder:

- **The Open Book:** <https://github.com/joeycastillo/The-Open-Book>
- **Babel:** <https://github.com/joeycastillo/babel>

You can either use git to clone these repositories in your Arduino libraries folder, or download the zip files and move the unzipped folders to the libraries folder. Note that the zip files may unzip as folders named "The-Open-Book-master" or "babel-master"; rename them to remove the "-master", so they are named "The-Open-Book" and "babel", and copy them to the libraries folder.

When you have all of these libraries installed, restart the Arduino IDE.

Testing the Screen

The first sketch will validate that the screen is working. Before beginning this test, double check the fine-pitched flex connector at the top right of the board. Inspect it for any bridged pins. If you see any bridged pins, separate them by applying flux and dragging your (clean) soldering iron away from the bridged spot.

When you are ready to test the screen:

1. Go to File -> Examples -> Open Book and load the `Open_Book_Screen_Test` sketch.
2. Plug the Feather M4 in to the back of the E-Book Wing board using the headers. Make sure that the Feather is oriented so that long and the short row of headers match up with the long and short socket header on the board.
3. Plug the assembled device in to your computer.
4. Select the Feather M4 Express from the list of boards in Tools -> Board. In Tools -> Ports, ensure that your Feather M4 is selected. If you do not see it in the list of ports, double tap the Reset button; this tends to fix a lot of issues with the board not appearing or responding.
5. Once you have selected the Feather M4, run the sketch!

You should see the screen flash to life, and display the Open Book Project logo.

Troubleshooting

- If you do not see the Feather show up in the Arduino IDE, check the Enable switch. If it is switched toward the center of the board, it is in the OFF position, and the Feather is not

receiving power. Switch it toward the outside of the board to turn it on.

- If you do not see the screen refresh, double check that the Feather is mounted correctly, and that the pins are correctly aligned. You should also double check that the screen's flex cable is seated correctly; re-seat the cable if necessary.
- If the screen image looks weak, gray or shows lots of noise or static, double check the orientation and type of the diodes in the E-Paper Display block. All three diodes should bear a tiny marking "B3" (not "CM") on the top of their cases, and the gray line which indicates polarity should be facing upward, toward the top of the board.
 - If one or more of the diodes is facing the wrong direction, desolder it and flip it around.
 - If one or more of the diodes bears the marking "CM", you've placed one of the Zener diodes here instead of the Schottky diodes that belong in this spot. Find the Schottky diode (again, look for the B3 marking). You may have put it in the "Extra Ports" block. Desolder and swap them out.

Burning the Babel Flash chip

The secondary Flash chip on the E-Book Wing board is dedicated to language support, but it currently does not have any data on it. To use it for languages, you will need to burn the glyph and font data on to that chip.

By default, the Babel Flash chip is not connected to any pin on the Feather, to make it easy for people to choose a pin that they want to use for it. In the case of the Feather M4 though, we have a spare pin on D4, so we'll use that one.

1. Unplug the device from your computer and remove the Feather from the Wing.
2. Locate the BCS solder jumper (the solder mask might be hard to read; consult the board diagram included with your board, or look for it right next to the "ER" in "FEATHER", near the socket header).
3. Using your soldering iron, bridge the two pads of the BCS solder jumper. This connects BCS to pin D4, which is what the Babel library will expect for a Feather M4.
4. Re-seat the Feather in the socket header, and plug the device back in.

That's the hardware setup. Next, you will need to get your card reader. The Babel library reads data from a 2 megabyte blob containing Unicode character data, Unifont glyphs and tables of localized case mapping and form shaping data. We are going to copy that blob to the MicroSD, and then run a sketch to burn that image to the Flash chip.

The file you will burn is called `babel.bin` ; it's located in the `babelconvert` folder inside the babel library you downloaded.

1. Copy the `babel.bin` file to your MicroSD card, and eject the drive
2. Plug the MicroSD card in to the socket on the e-book wing.
3. Go to File -> Examples -> Babel Universal Character Library and load the `BurnBabelBurn` sketch.
4. Run the sketch, and open the Serial Monitor (Tools -> Serial Monitor)

At this point you will see a menu of options for actions you can take.

1. First, enter `E` and press 'Enter'. This will erase the Flash chip.
2. Next, enter `W` and press 'Enter'. You will see a series of messages indicating the writing of a page to Flash. Wait until they finish scrolling; you will see "Done!" appear when the process is complete.
3. Finally, enter `V` and press 'Enter'. This will verify that the burned data matches the blob on the SD card.

If the image verifies, you're done! You can delete the `babel.bin` file from the SD card; the Flash chip will retain this data indefinitely.

Troubleshooting

- If the verification fails a few pages in, return to step 1 and try the whole process again. Also, if you have just gotten done soldering the chip, wait until it cools down completely; I've experienced errors when I tried to flash the Babel image right after pulling the board from the oven.
- If the process fails more than three times, try using a different brand of MicroSD card; I have had issues with off-brand cards, but no problems at all with Sandisk cards.
- If the Flash size appears as 0, and writing fails, double check the placement of the Flash chip. The dot indicating pin 1 position should be at the bottom left. If it is not, you can remove this chip this using hot air and tweezers, and re-place it correctly.
- If you get the message "Card failed, or not present", double check the connections of the MicroSD socket.

Running the MVBook Software

The MVBook software is a simple menu system for displaying a few works, stored in a specific format, and allowing the user to select and page through them. It's not as full-featured as I hope it someday will be, but it is my hope that as we build more of a community around this hardware, we can advance better software support together.

The MVBook example folder includes a zip file called `books.zip` with a handful of public domain works you can copy to an SD card. The MVBook example folder also includes a Python script called `bookbinder.py` that formats plain text files to function with MVBook. Ideally, the canonical format for reading will be plain text files, but for the moment, some of the work of pagination and embedding metadata needs to happen on a computer.

1. Copy the works you want to read with MVBook to an SD card. Insert it into the E-Book Wing.
2. Go to File -> Examples -> Open Book and load the `Open_Book_MVBook` sketch.
3. Run the sketch!

You should see the screen refresh and display a listing of the books on the MicroSD.

Controls

- Use the direction pad's "Up" and "Down" arrow keys to navigate the list of works.
- Use the direction pad's "Select" button in the middle to select a work in the list and enter reading mode.
- When reading a work, use the "Previous Page" and "Next page" buttons on the far left and right to navigate the work.
- To return to the main menu while reading a work, press the center "Select" button.

Troubleshooting

- If the MVBook sketch does not run, ensure that you completed the steps in "Burning the Babel Flash chip" above, as MVBook depends on Babel for text rendering.
- If the interface is not responding, especially after you have left the device alone for a while, ensure that the board has power by plugging it in or pressing the reset button. The e-paper display will retain whatever text was on the screen last even after the battery dies, which can be confusing.

Ideas for Future Projects

The Feather M4 has a DAC on A0 that the E-Book Wing uses to drive a headphone jack; you can use the [Adafruit Talkie library](#) for synthesized speech, or the [Adafruit MP3 library](#) to play back audiobooks or podcasts.

MVBook only makes use of a handful of the available buttons; you could put together a much more interesting user interface by making use of the full D-Pad, or configuring the Lock button to emit an interrupt for low power sleep/wake operation.

The STEMMA ports along the left side offer a lot of flexibility! The A1 port (on the bottom) also has a DAC; you could add [a STEMMA speaker](#) if you don't want to use headphones, or add [a Neopixel strip](#) to either port to make a reading light! These ports also work to add a [microphone](#) for voice control (there's a demo of this, but may need some tweaking; look for

`Open_Book_Heart_Of_Darkness_TensorFlow`). The STEMMA I2C port (left side center) lets you plug in all manner of sensors; for accessibility, you could add a [gesture sensor](#) to turn pages with a flick of the hand.

If you have a Feather with stacking headers, you can also add [an AirLift Feather Wing](#) to add WiFi to your book! The wing leaves pins 11, 12 and 13 free as well as the UART (pins 0 and 1), so it stacks with any wings that can use those pins or the I2C pins, such as (among others): the [DS3231 Precision RTC FeatherWing](#) (for telling time), [Ultimate GPS FeatherWing](#) (maybe a navigation device?), [ADXL343 + ADT7410 Sensor FeatherWing](#) (weather station?), or [INA219 FeatherWing](#) (workbench friend).

In terms of book-reading software, my goals are to get on-device pagination working, and have a basic system for YAML-style front-matter for title and author metadata. This will probably be a separate project from the Arduino MVBook project. For updates on this effort, or if you have questions or comments about either the E-Book Wing or the Open Book Project, visit the GitHub repository: <https://github.com/joeycastillo/The-Open-Book/>