

# Open Book Documentation

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Board Setup</b>	<b>6</b>
<b>3</b>	<b>Soldering Strategies</b>	<b>7</b>
<b>4</b>	<b>Assembling the board</b>	<b>10</b>
<b>5</b>	<b>Required Support Software</b>	<b>13</b>
<b>6</b>	<b>Flashing the Bootloader</b>	<b>14</b>
<b>7</b>	<b>Testing the Hardware</b>	<b>17</b>
<b>8</b>	<b>Burning Font Data</b>	<b>20</b>
<b>9</b>	<b>Installing the E-Paper Display</b>	<b>21</b>
<b>10</b>	<b>Next Steps</b>	<b>23</b>



# 1 Introduction

Hi there, and congrats on taking the plunge and deciding to build your own Open Book! The fact that you've sought out this kit and brought it into your life probably signals that you understand the gist of its purpose, but to recap: this is an open hardware device for reading that you are going to build yourself and understand from bottom to top.

For the moment, this guide is broken up into three parts, and this booklet only covers part 1, the board setup. Parts 2 and 3 (a deep dive into each of the Open Book's subsystems, and a section with notes and ideas for future work) are available on The Open Book wiki<sup>1</sup>.

## 1.1 What You Will Need

It's tough to know exactly what to put in this list, because there are many different ways to put this thing together. Having said that, there are some broad requirements, and some specific items that can fulfill those requirements, so I'll outline them here and then go into more specifics in the sections where we need those things.

- All of the parts on the Open Book Bill of Materials (BOM), as well as the Open Book printed circuit board (PCB).

---

<sup>1</sup><https://github.com/joeycastillo/The-Open-Book/wiki/Open-Book-Documentation>

- Tools necessary for soldering the parts on to the board. If you are going to seek out tools to build this, I suggest reading section 3, “Soldering Strategies”, for a breakdown of the options available to you.
- A multimeter<sup>2</sup>. The linked one on Adafruit’s site is quite nice, but you can also pick one up for cheap at your local hardware store<sup>3</sup>.
- A device for burning the bootloader. In the ‘Flashing the Bootloader’ section, I mention a few pro-level debuggers that can do the job. But this guide offers specific instructions for using a SAM D21 dev board to burn the bootloader, so I suggest you get a Feather M0 Express<sup>4</sup> (\$19.95) or a Trinket M0<sup>5</sup> (\$8.95); they’re awesome boards, and you can use them for other projects afterwards.
- Some optional items that enable functionality on the book: a MicroSD card<sup>6</sup> for flashing the global font data, a Lithium-Polymer battery<sup>7</sup> for making the device portable, and an external 1/8” headset<sup>8</sup> if you want to use the Open Book’s audio features.

---

<sup>2</sup><https://adafru.it/2034>

<sup>3</sup><https://www.harborfreight.com/7-function-digital-multimeter-63759.html>

<sup>4</sup><https://adafru.it/3403>

<sup>5</sup><https://adafru.it/3500>

<sup>6</sup><https://adafru.it/1294>

<sup>7</sup><https://adafru.it/3898>

<sup>8</sup><https://adafru.it/1966>

## 1.2 BOM Notes

With the exception of the e-paper display, you can for the most part just go to the Open Book's Kitspace page<sup>9</sup> and 1-click order the parts you need from Digikey. But there are a couple of parts that you can swap out to either save a few bucks or gain a bit of extra functionality:

- The MicroSD slot in the 1-click BOM lacks card detect pins, so you won't get that functionality from it. Adafruit sells a better one<sup>10</sup> that uses the same footprint and does have card detect. It's also slightly cheaper.
- The Feather connectors on the BOM are rather ludicrously expensive; one of them alone costs more than the SAM D51 microcontroller on the board! I have taken to ordering two of these 2x20 surface mount headers<sup>11</sup> and cutting them down to size, which saves \$4 per board.
- If the listed part numbers for some passive components are out of stock, or you find alternatives that are cheaper, you can swap in parts with the same values. Just be extra careful with the 4.7 $\mu$ F capacitor and the 1 $\mu$ F capacitors in the e-paper display block; for the most part, they need to be rated for 25 volts.
- When shopping for the e-paper display, you

---

<sup>9</sup><https://kitspace.org/boards/github.com/joeycastillo/the-open-book/>

<sup>10</sup><https://adafru.it/1660>

<sup>11</sup><https://adafru.it/2187>

may be tempted by Good Display's 4.2" tri-color EPD screens, which add either red or yellow to the standard black and white. **I suggest you don't do it.** The long refresh time (10-15 seconds) required for tri-color display modes makes this sort of display poorly suited for book reading.

- Finally, the 24-pin flex connector. The part on the BOM does work, but in my experience, Good Display tends to include a very high quality 24-pin connector for each display you order from them. I would recommend using that connector over the one in the BOM, but it's there just in case.

## 2 Board Setup

The Open Book comes to you as a bare PCB. Coaxing life into this inanimate sheet of copper and fiberglass will require that you solder dozens of small parts onto the board. The vast majority of those parts are surface mount technology, or SMT. If you have assembled surface mount boards before, no problem! Use whatever method makes the most sense for you. If you have not, there are several strategies that I have used to assemble the Open Book; I'll outline them here, so you can choose one that you're most comfortable with.

## 3 Soldering Strategies

First off, you can solder this board together using a fine-tipped soldering iron and either solder paste or thin solder. Collin Cunningham of Adafruit has an excellent video on the topic<sup>12</sup>; that video covers soldering all of the kinds of devices you'll find on the Open Book board. You'll need the following tools on hand:

- Soldering iron<sup>13</sup> with a fine tip<sup>14</sup>
- Fine-tipped tweezers<sup>15</sup> (I prefer the curved tip, but the straight tip<sup>16</sup> works too)
- Flux<sup>17</sup> or a flux pen<sup>18</sup>
- Solder wick<sup>19</sup>
- Solder sucker<sup>20</sup> (also available in extra fancy<sup>21</sup>)
- Thin solder<sup>22</sup> (that's lead free, but there's also leaded solder<sup>23</sup>)
- Magnifier loupe<sup>24</sup> or microscope<sup>25</sup>.

---

<sup>12</sup><https://youtu.be/QzoPxvIM2qE>

<sup>13</sup><https://adafru.it/180>

<sup>14</sup><https://adafru.it/1249>

<sup>15</sup><https://adafru.it/422>

<sup>16</sup><https://adafru.it/421>

<sup>17</sup><https://adafru.it/2667>

<sup>18</sup><https://adafru.it/3468>

<sup>19</sup><https://adafru.it/4456>

<sup>20</sup><https://adafru.it/148>

<sup>21</sup><https://adafru.it/1597>

<sup>22</sup><https://adafru.it/1930>

<sup>23</sup><https://adafru.it/1886>

<sup>24</sup><https://www.digikey.com/short/zhvph4>

<sup>25</sup><https://adafru.it/636>

I've also found that some plain old high magnification reading glasses, like you might find at the drug-store, help immensely here as well.

While you can assemble the board completely using this method, I've found that using hot air for some parts makes life much easier. Specifically, I'm talking about the microSD slot; it's possible but difficult to maneuver a soldering iron in the little slotted area, but I feel more comfortable putting down some solder paste and hitting it with hot air, as depicted in this video by Antti Kupila<sup>26</sup>.

To do this hot air reflow, you would also need:

- A hot air soldering station<sup>27</sup>, and
- Solder paste<sup>28</sup>. Personally, I use this solder paste myself<sup>29</sup>, but that's a big syringe; the small tube of Maker Paste should be plenty.

The hot air soldering station is also very useful if you need to rework, i.e. removing a chip that was placed in the wrong orientation: you can blow hot air at the component and remove it with tweezers.

Speaking of hot air, the final method for soldering a board like the Open Book is the one I prefer the most: the reflow oven. This method involves placing all the solder paste and all the components all at once, and

---

<sup>26</sup><https://youtu.be/vzoMEBmCNQQ>

<sup>27</sup><https://adafru.it/1869>

<sup>28</sup><https://adafru.it/3217>

<sup>29</sup><https://www.digikey.com/short/zh1qmq>



then baking the board at high temperature to set everything in its place. I inherited one of these<sup>30</sup> from a colleague, but you can also DIY one, as shown in this tutorial from Dan Cogliano<sup>31</sup>. I've even seen folks reflow on a skillet<sup>32</sup> or with a spoon and a lighter<sup>33</sup>!

All of these (with the possible exception of that last one), are valid strategies for assembling the Open Book. Whichever one you choose, there are a few things to be aware of:

- Be aware of the polarity of diodes and LEDs. Diodes have a gray line on their plastic body that should face the same direction as the line in the diode symbol on the silkscreen. LEDs have a marking, usually green, indicating the cathode; this should match with the dotted marking on the silkscreen.
- Be aware of the different types of diodes and MOSFETs used in this design. Q1 and Q2 look nearly identical, but have very different electrical properties.
- Similarly, the Zener diodes in the 'Extra Ports' block are easy to mix up with the Schottky diodes in the 'E-Paper Display' block. Make sure to put the correct diodes in the correct places.
- When placing IC's, be conscious of the pin one

---

<sup>30</sup>[https://www.ebay.com/sch/?\\_nkw=T-962+reflow+oven](https://www.ebay.com/sch/?_nkw=T-962+reflow+oven)

<sup>31</sup><https://learn.adafruit.com/ez-make-oven/>

<sup>32</sup><https://www.sparkfun.com/tutorials/59>

<sup>33</sup>[https://twitter.com/\\_mg\\_/status/1152317329646088192](https://twitter.com/_mg_/status/1152317329646088192)

alignment; in most cases, there is either a dot or a notch on each chip that should match with the dot or notch on the board's silkscreen.

- The shift register may be a little different: the silkscreen on the board depicts a notch instead of a pin 1 dot. The pin 1 dot should be on the same side as the notch, the right side.

## 4 Assembling the board

Using whichever method you have chosen, place and solder all of the surface mount components, but stop just short of placing any through-hole components or the e-paper screen.

Inspect all the parts for bridged pins, and use a multimeter to check for short circuits between the power and ground nets. To do this, set your multimeter to continuity mode, and touch one probe to GND in the Feather header. Then touch the other probe to each of VBAT, VBUS and 3V3, also in the Feather header. You shouldn't hear any beeps. If you do hear a beep, that signals a short circuit; you should not power up the board until you address it.

- If 3V3 is shorted to ground, use your magnifying device to look for solder bridges on the fine-pitched pins of the flex connector and the SAM D51; there are a handful of spots where a power pin neighbors a ground pin. Also check the two bottom right pins of R13. If you find a bridged

pin, remove any excess solder with wick and / or flux.

- If either VBAT or VBUS are shorted to ground, inspect the right side of R6 and the right side of U2. Remove any excess solder.

At this point it's wise to take one last look at the USB connector to make sure that none of these pads are bridged. In particular, use your multimeter to make sure that there are no short circuits between the three pins on the right. If any of these pins have a short, **do not power up the board until you address it**. This space is admittedly tricky. You may be able to get in there with solder wick or flux, but personally I've found it easier to use hot air to remove the port entirely and re-place it.

Finally — very important — **double check that you have soldered the two through holes on either side of the Micro-USB connector!** This is important to the structural stability of the USB port; I once ripped a USB port, pads and all, completely off of the PCB after having forgotten to secure these posts. This kills the board.

Once all the power nets are in order and the USB port is secure, plug the board in to a USB power brick, and use your multimeter to measure 3V3 and VBUS: in the appropriate DC voltage measurement mode, touch the black probe to GND in the Feather header, and the red probe to each of VBUS and 3V3. You should see 3.3 volts on 3V3, and 5 volts on VBUS.

### 4.1 Troubleshooting

- If you see 5 volts on VBUS and significantly more than 3.3 volts on 3V3, **unplug the device immediately**, and check again for solder bridges on the USB port's pads. A bridged pad from VBUS to D- could cause this condition. Unfortunately it could also fry the microcontroller's USB interface. (you'll know this for sure if, later in this guide, you are able to flash the bootloader but can't connect to the board over USB, in which case you'll have to desolder and replace the SAM D51)
- If you see 5 volts on VBUS but less than 3.3 volts on 3V3, double check the orientation of diode D1: the gray line indicating polarity should be facing left, away from the USB port. If it is backwards, desolder it and flip it around.
- If you do not see 5 volts on VBUS, jiggle the cable and see if it helps. Either way, unplug the board and double check that all of the pins of the Micro-USB connector are soldered fully, re-working if need be.

At this point, you can solder the through-hole buttons (they go in from the top, facing toward the screen) and the Enable switch (which goes in from the bottom, with the metal case exposed on the backside).

Now we can move on to the software portion of this section.

## 5 Required Support Software

We will be working with the Arduino IDE<sup>34</sup> for this portion of the guide; if you do not already have Arduino on your machine, download and install it now. You will also need to have both Adafruit and Oddly Specific SAMD board support installed. Follow the instructions here<sup>35</sup> to get this set up. The two board manager URLs are:

- [https://adafruit.github.io/arduino-board-index/package\\_adafruit\\_index.json](https://adafruit.github.io/arduino-board-index/package_adafruit_index.json)
- [https://www.oddlyspecific.org/boardsupport/package\\_oddlyspecific\\_index.json](https://www.oddlyspecific.org/boardsupport/package_oddlyspecific_index.json)

If you have trouble finding the Oddly Specific board support package, search for ‘Open Book’ in the Boards Manager search box; it should be the only result.

In addition to the board support package, you will need several libraries installed. Most of them, we will install from the Arduino Library Manager:

- **Adafruit DAP Library**
- **Adafruit BusIO**
- **Adafruit GFX Library**
- **Adafruit EPD**
- **Adafruit SPIFlash**
- **SDFat - Adafruit Fork**

---

<sup>34</sup><https://www.arduino.cc>

<sup>35</sup><https://learn.adafruit.com/add-boards-arduino-v164/setup>

- **Adafruit NeoPixel**
- **Adafruit MP3**
- **arduino-menusystem** (optional; needed for MVBook example in Part 2)

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

- **The Open Book**<sup>36</sup>
- **Babel**<sup>37</sup>

You can either use git to clone these repositories in your Arduino libraries folder, or download them as 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.

## 6 Flashing the Bootloader

The SAM D51 you soldered onto this board is, as you read this, blank; there’s no user code running on it. Eventually, we want to be able to plug it in with USB and get code on it that way, but there’s one stop on

---

<sup>36</sup><https://github.com/joeycastillo/The-Open-Book>

<sup>37</sup><https://github.com/joeycastillo/babel>

the way to that, and that's flashing (aka burning) the bootloader. The bootloader is a small program that resides right at the beginning of the SAM D51's Flash memory; it's the first piece of code that runs when powered up.

We are going to install a program called the UF2 bootloader, from Adafruit and Microsoft. It uses the SAM D51's native USB interface to offer two different ways to get code onto the chip: USB serial emulation, which lets you upload code from Arduino, and USB mass storage, which lets you write a UF2 file by dragging it onto the board like a thumb drive. But since there's no bootloader on there now, we will need to flash it using the Serial Wire Debug, or SWD interface.

You can access this interface using a SWD cable and the 2x5 header in the "Debug Port" block, if you populated it; if not, you can use five pogo pins<sup>38</sup> and a steady hand on the five debug pads underneath it (this has been my preferred method). If nothing else, you can solder wires to these pads, and connect them to the device you're going to use for programming.

Speaking of which: you're going to need a second device to burn the bootloader! If you have a J-Link, Atmel-ICE, or other SWD-capable debugging tool, you can use it and Atmel Studio<sup>39</sup> to burn the

---

<sup>38</sup><https://adafru.it/2429>

<sup>39</sup><https://learn.adafruit.com/how-to-program-samd-bootloaders/programming-the-bootloader-with-atmel-studio>

bootloader. But those gadgets are expensive, so instead, we're going to walk through using another dev board to burn the bootloader.

In the Arduino IDE, go to File -> Examples -> Open Book, and find the sketch called BurnBootloader. This sketch is not designed to run on the Open Book, but rather to run on a SAM D21 board like the Feather M0, which we'll call the "host board".

1. Connect the Open Book's RESET, SWCLK and SWDIO to pins 10, 11 and 12 of the host board (leave 3V and GND alone).
2. Plug the Open Book in to power, and the host board in to your computer
3. Run the sketch on the host board, and open the serial monitor; you should see it erase the Open Book's flash chip, and then burn the bootloader!

You'll know it worked when you double tap 'Reset' on the Open Book: the red LED should begin to pulse, and the Neopixel should glow red or green.

### 6.1 Troubleshooting

- If you open the serial monitor and find that the BurnBootloader program cannot connect to the Open Book, double check the Open Book's Enable switch; it should be switched toward the inside of the board. If it is switched the other



way, toward the mounting hole, the board is powered down; switch it in the other direction and try again.

- If you still cannot connect, double check that that pin 10 is connected to RESET, 11 to SWCLK and 12 to SWDIO; if any of these wires are swapped, it will not connect.
- If you are using any host board other than a Feather M0, try again using a Feather M0. In theory this sketch should work with other host boards, but I experienced issues using an M4 board, and these instructions were tested specifically with a Feather M0.

Once you see the BurnBootloader sketch succeed, you're done with the debug interface! From now on we can talk directly to the Open Book via its own USB port.

## 7 Testing the Hardware

Note that this step requires a few external parts, if you want to test functionality that depends on them:

- A MicroSD card<sup>40</sup>, just for the sake of testing that the card slot is seated properly.
- An external 1/8" headset<sup>41</sup> with microphone and inline button

---

<sup>40</sup><https://adafru.it/1294>

<sup>41</sup><https://adafru.it/1966>

- A Lithium-Polymer battery matching Adafruit's battery polarity (I suggest this one<sup>42</sup>)

If you don't have one or more of these items on hand, no worries! You can choose to skip any test.

1. In the Arduino IDE, go to File -> Examples -> Open Book, and find the sketch called Open-Book\_SystemTest.
2. Plug the battery in to the Open Book, and double tap the Reset button; you should see the red LED pulse.
3. Plug the Open Book in to your computer, make sure its serial interface is selected, and run the sketch.
4. Open the serial monitor; this is where we'll execute the various phases of the test. Follow the prompts, testing each of the subsystems. If there are any tests you wish to skip, send the letter 'S' out over the serial monitor and it will move on.

After the board test is done, you will get a report about what passed and failed. If everything you expected to pass, passed? You're almost done!

### 7.1 Troubleshooting

Depending on what failed its test, some suggestions:

---

<sup>42</sup><https://adafru.it/3898>

- In general, double check all the pins of the SAM D51; make sure none are either bridged to a neighboring pin, or not connected with solder. If you see a joint that is not connected, unplug both USB and battery, re-solder the pin and try again. Otherwise:
- Buttons: double check the placement of the shift register U5. You should have a beveled edge along the top of the chip and a dot at the top right indicating pin 1.
- SD Card: If you get the message “Card failed, or not present”, double check the connections of the MicroSD socket.
- SD Card Detect: Some MicroSD card slots lack card detect contacts. If this test fails, it’s not the end of the world; it’s a nice-to-have feature but far from a hard requirement.
- Babel or QSPI Flash Chip: If either fails to be recognized, check the placement of U6 or U4, respectively. In both cases, the dot indicating pin 1 position should be at the top right. If it is not, you can remove this chip this using hot air and tweezers, and re-place it correctly
- Mic Preamp: double check the placement of U7. The dot on this one is tiny, but it should be on the top right. Also make sure that none of the pins are bridged; this component has a finer pitch than some of the others.

## 8 Burning Font Data

This step requires the MicroSD card from the last step, as well as a binary blob called `babel.bin`; it's the language support file that contains all the glyphs and Unicode data that the Open Book will use to display texts. You can find it in the `babelconvert` subfolder of the Babel library, or download it here<sup>43</sup>.

1. Use your computer to copy the `babel.bin` file to your MicroSD card, and eject the drive
2. Plug the MicroSD card in to the MicroSD socket.
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 as the sketch writes pages of data to the Flash chip. Wait until they finish scrolling; you will see "Done!" appear when the process is complete.
3. Finally, enter `V` and press 'Enter'. This will ver-

---

<sup>43</sup><https://github.com/joeycastillo/babel/raw/master/babelconvert/babel.bin>

ify 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.

### 8.1 Troubleshooting

- If the verification fails a few pages in, return to step 1 (enter 'E') and try the process again.
- 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 Babel verification 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.

## 9 Installing the E-Paper Display

Finally! For this last test, we are going to install the sketch before attaching the screen. In the Arduino IDE, go to `File -> Examples -> Open Book`, and find the sketch called `OpenBook_ScreenTest`.

1. Plug the Open Book in to your computer, make sure its serial interface is selected, and run the sketch.

2. Unplug the device.
3. Carefully take the e-paper display out of any protective packaging it's in. Remember, it's made of very thin glass!
4. Thread the screen's flex cable through the hole up top, and secure it in the flex connector.
5. Taking care to avoid damaging the flex connector, turn the device over so you can see the screen, and power up the device by plugging it in to power.

If you see the screen refresh, you're done! Secure the screen to the front side of the Open Book with double sided tape.

### 9.1 Troubleshooting

- If you do not see the screen refresh, 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.

## 10 Next Steps

In this section, we walked through the basic steps required to assemble the board and bring it to life for the first time. In the process of assembling and testing the board, you probably gleaned some sense of how the board works in broad strokes. For more details, check out the Open Book wiki<sup>44</sup>:

- In part 2, we’ll delve deeper into each section of the board: the purpose of each subsystem, the components that comprise them, their underlying workings and how they interact with each other.
- In Part 3, you’ll find notes on building an enclosure for the Open Book, information about custom waveforms for the e-paper display, and details about the data on the Babel Flash chip.

Congrats on building your book!

---

<sup>44</sup><https://github.com/joeycastillo/The-Open-Book/wiki/Open-Book-Documentation>

## **Open Book Documentation**

*CC-BY-SA 4.0, Joey Castillo*

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.  
To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to  
Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.