

DJBB

# MIDI LOOPSTER

## User Manual & Assembly Guide

Last Updated: 3/24/2024



The DJBB Midi Loopster is a an open source MIDI controller with a simple looper feature that makes for some fun & quick experimental jams. It plays a bit like a guitar pedal looper, but with the flexibility of MIDI.

## Features:

- USB and Mini TRS (Type A) MIDI Out
- MIDI Looper - record and overdub loops of midi notes
- Satisfyingly clicky buttons that aren't too loud.
- Scale select
- Encoder Play Mode - play notes by turning encoder
- Set velocity of pads independently
- Velocity map a single pad across all pads - nice for hi-hats
- Open Source, relatively easy to "hack"
- "Chord Mode" - record a chord or sequence to a specific pad

## Resources

**Design process:** <https://www.djbajablast.com/post/djbb-midi-loopster>

**Github:** <https://github.com/derrickthomin/DJBB-Midi-Loopster>

**STL Files:** <https://www.printables.com/model/661953-djbb-midi-loopster-3d-printed-parts>

**Submit Issue / Idea:** <https://github.com/derrickthomin/DJBB-Midi-Loopster/issues>

# Table of Contents

<b>User Manual</b>	<b>4</b>
<i>Play Screen</i>	4
<i>Scale Screen</i>	7
<i>Looper Screen</i>	7
<i>Tips &amp; Tricks</i>	8
<i>Change Default Startup Settings</i>	8
<i>Installing Software / Updating</i>	9
<b>Assembly Guide</b>	<b>12</b>
<i>Reference Diagrams</i>	12
<i>Build Steps</i>	12

# User Manual

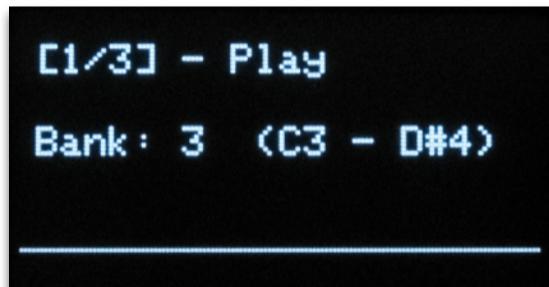
To change settings on the current screen, turn the encoder. To move between screens click the encoder, then twist. The  icon on the bottom of the screen indicates that turning the encoder switches screens.



*In nav mode - turning encoder changes screens*

## Play Screen

This is the default startup screen. Here you can Change MIDI banks, set note velocities, enable single note mode, and toggle between standard / encoder play modes.

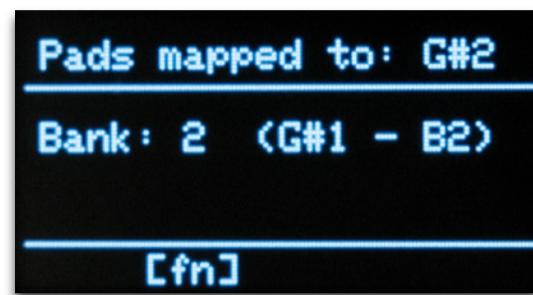


*Play mode screen. Turn encoder to change banks.*

- To **Play Notes**, press the 16 pad buttons.
- To **Change Banks**, Turn the encoder.
- To **Set pad velocity**, hold pad(s) and turn the encoder.

- To enter **Single Note Mode**, hold **[fn]** (top left button) and click a pad.

The pad's note will be mapped to all 16 pads with a gradient of velocities from 1-127. To undo, hold **[fn]** and click any pad.



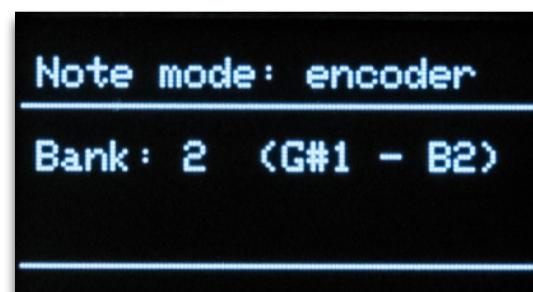
*Hold [fn] and press pad to velocity map single note across all pads.*

- To switch between **Standard**, **Encoder**, and **Chord** play modes, double click **[fn]**.

## ENCODER MODE

Double click **[fn]** to switch from standard to encoder mode.

- In Encoder Play Mode, you can still play the pads as usual, but now if you hold a pad and turn the encoder, it sends notes.
- An off note will only be sent if you release the pad, or if you turn the encoder counterclockwise, which immediately sends off notes for all pads.

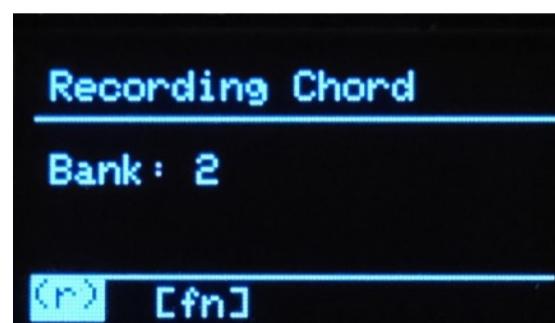


*Encoder Mode - Hold note and turn encoder to play notes.*

## CHORD MODE

Double click **[fn]** to switch from encoder mode to **Chord Mode**.

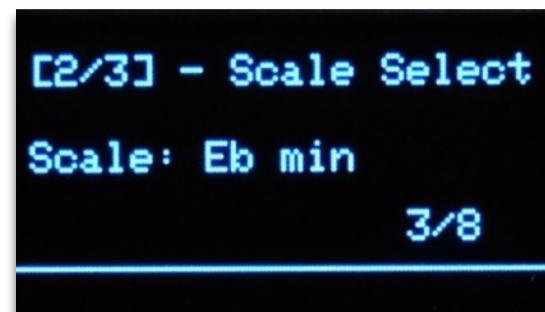
- Hold **[fn]** and select a pad to begin recording a chord or sequence of notes. Press **[fn]** to stop recording. Silence will automatically be trimmed from the beginning.
- Now when you play the selected pad, the chord or sequence plays.
- Hold **[fn]** and click the pad again to clear the recording.



*Hold **[fn]** and click a pad to  
begin recording.*

## Scale Screen

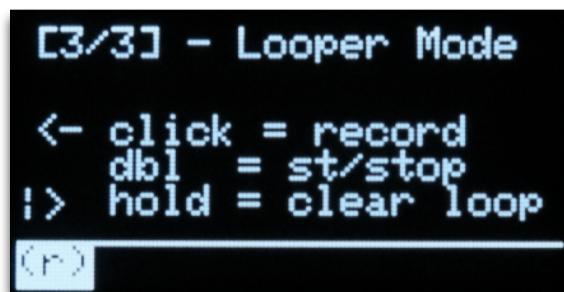
Use the **[fn]** button and encoder to change the key and scale. Chromatic mode is the default, which includes all notes.



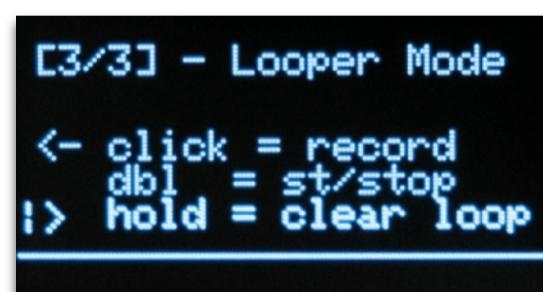
Scale screen

## Looper Screen

- To **Begin Recording** or **Stop Recording**, click **[fn]**
- To **Start** or **Stop** loop, double click **[fn]**
- To **Overdub** click **[fn]** while the loop is playing. Click again to disarm recording.
- To **Delete The Loop**, hold **[fn]**
- To **Delete a Random Loop Note**, turn encoder counterclockwise
- Only 50 notes can exist at a time



Looper currently recording



Looper playing but not recording

## Tips & Tricks

- Updates to the display mess with MIDI timing - an issue when a recorded loop is playing. I think it's fun, but If this is an issue for you, consider setting PERFORMANCE\_MODE to True in `loopster_settings.json`. Note that the timing goes back to normal on the next loop around.
- Iterate to find wonky loops - create a loop, use the [Delete a Random Loop Note feature](#) (turn encoder CCW) to randomly delete a note from the loop. Then, you can re-record on top of this. Repeat. Since random notes are removed, you can end up with unexpected results.
- Try [Single Note Mode](#) (Play Screen) with hi-hats.
- The encoder is a standard size - most knobs you find on Amazon / wherever will fit if you want to replace the default one.

## Change Default Startup Settings

Some settings can be customized via the **loopster\_settings.json** file. JSON files are just structured text, so open the file in any text editor. Each parameter has an associated help text property that you can reference.

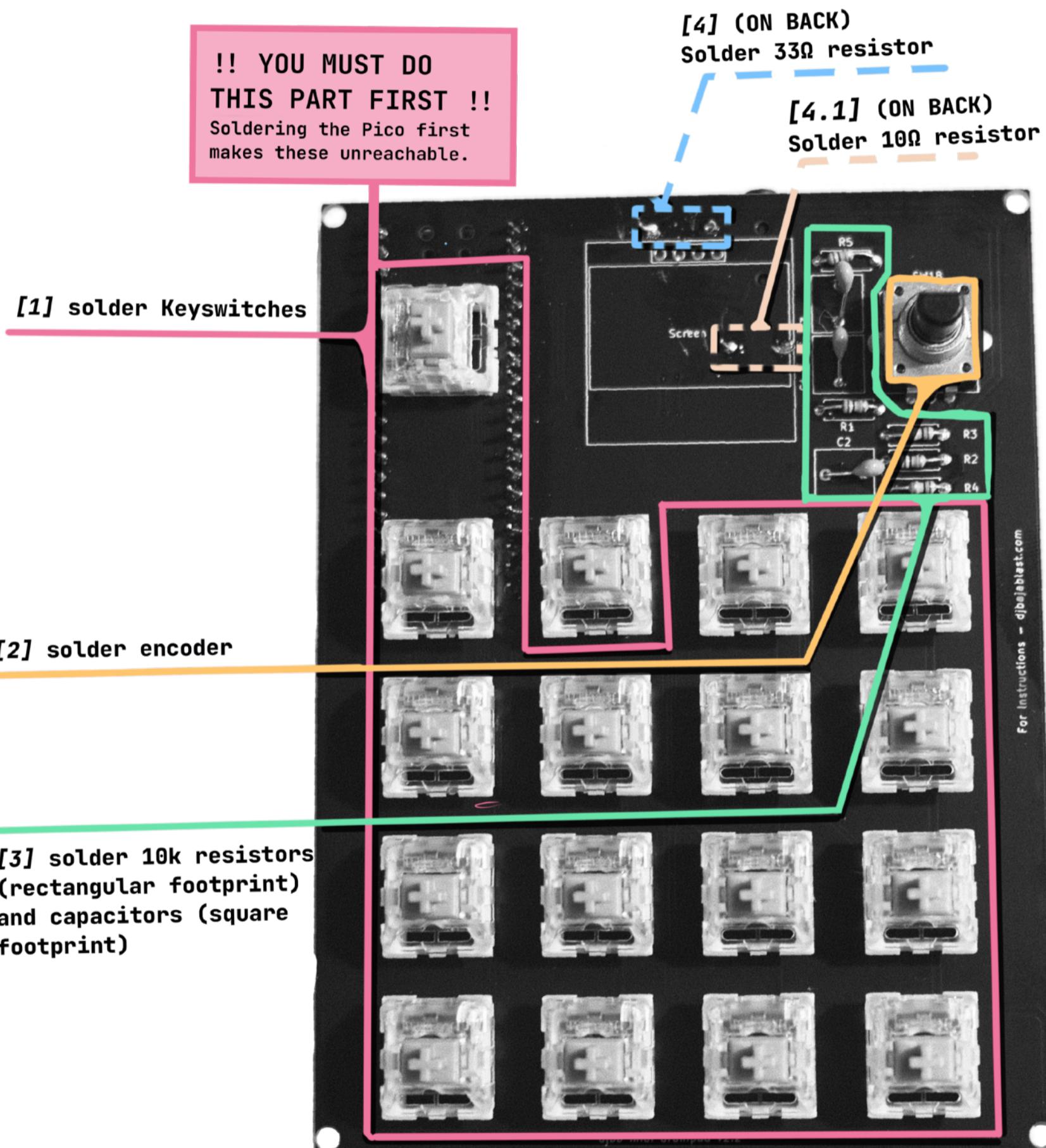
**Caution** - there is no data validation, so be sure the values you enter make sense. For example, don't set the default velocity to 200 (outside of the 0-127 MIDI range).

If you mess it up, no worries. Just copy and paste from the file in the [Github Repo](#).

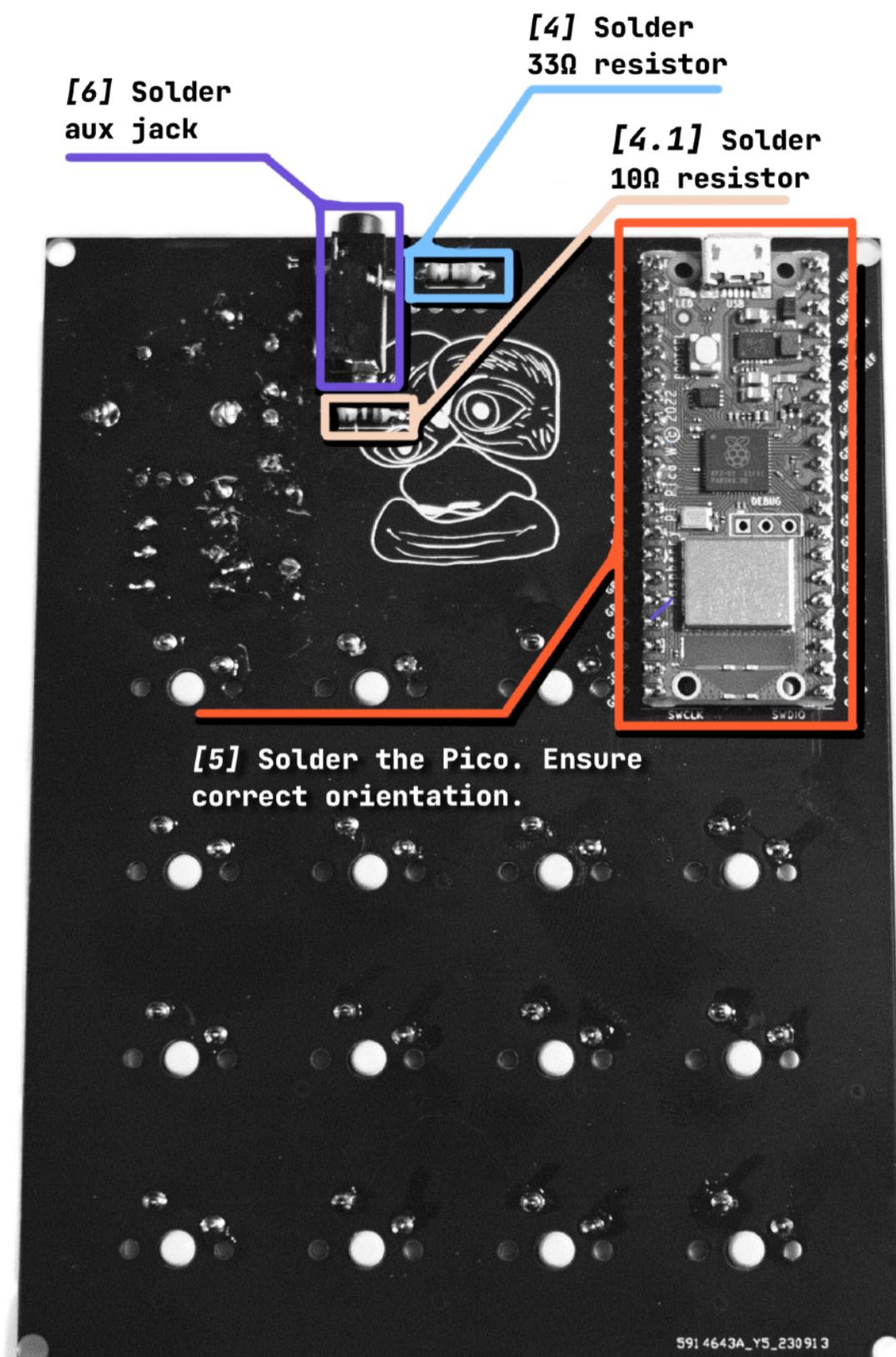
Are there more things you wish you could customize? Submit an issue [here on GitHub](#).

# Installing Software / Updating

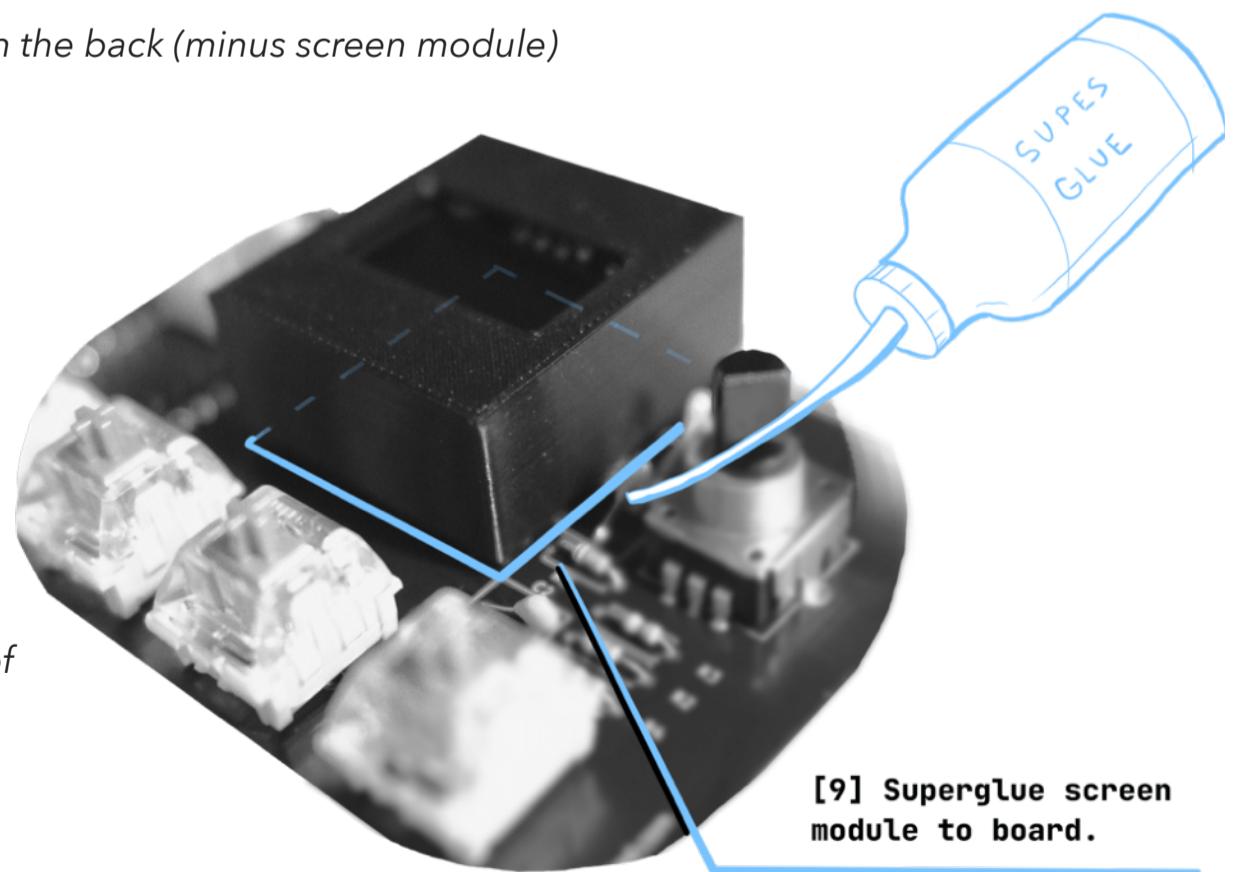
If you are just *updating* the loopster software, skip to step 4.



Assembled Loopster PCB from the Front (minus screen module)



Assembled Loopster PCB from the back (minus screen module)



Use superglue to glue the bottom of the screen module to the board ->

[9] Superglue screen module to board.

**For kit buyers -** the software should already be loaded onto the Pico and this can *all* be skipped.

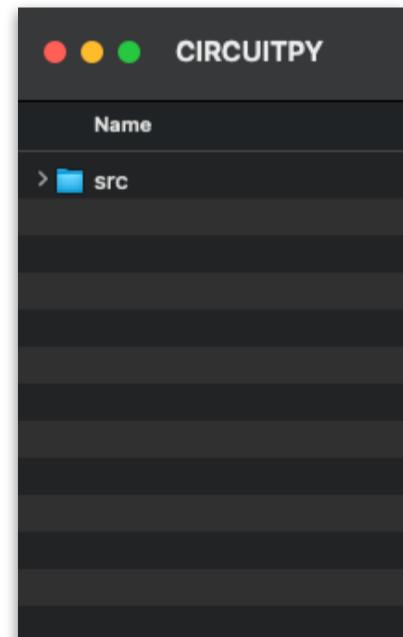
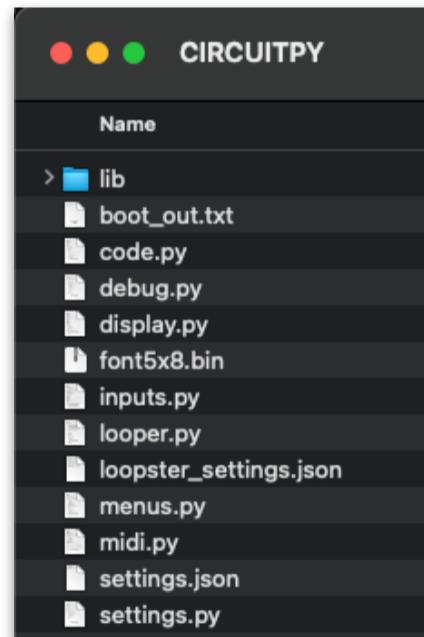
Before installing the Loopster code, you must install CircuitPython<sup>1</sup>, a lightweight version of the Python programming language. This is the “operating system”, if you will.

1. Download the latest .uf2 file from [https://circuitpython.org/board/raspberry\\_pi\\_pico\\_w/](https://circuitpython.org/board/raspberry_pi_pico_w/)
2. Hold the Boot button on the Pico and plug in the micro USB cable while still holding the button. You can release the button once a new device pops up in Finder / Explorer (likely has RPI in the name).
3. Drag the .uf2 file to the new device. it should reboot and reappear as a new device called CIRCUITPY.
4. Counterintuitive.. but delete everything from the CIRCUITPY device. Don’t worry, it doesn’t undo the previous step.

**YES**

**NO**

5. Go to the [Loopster GitHub Repo](#) and download it as a ZIP file.
6. Open the **src** folder and drag all files/folders to the empty CIRCUITPY device. The next time it reboots, it should be named “LOOPSTER” in Finder / Explorer.



Copy **contents** of **src**  
folder...

... not the whole folder

---

<sup>1</sup>You can read more about installing CircuitPython on the Pico here: <https://learn.adafruit.com/getting-started-with-raspberry-pi-pico-circuitpython/circuitpython>

# Assembly Guide

See the video version [here](#).

## Reference Diagrams

## Build Steps

**YOU MUST DO STEP [1] FIRST. DO NOT SOLDER ON THE RASPBERRY PI PICO BEFORE STEP 1 IS COMPLETE.**

If you're 3D printing your own parts, [download the STL files](#) and print them.

**[1]** Solder the keyswitches to the front of the board, but leave out the key directly to the right of the Pico footprint (see diagram - the only button included in the pink circle). This makes soldering the Pico easier in a later step.

**Check the solder joints for the [fn] button (left of screen footprint) and the button directly below it.** Once the Pico is soldered on, you can't access these.

**[2]** Solder the encoder to the front of the board.

**[3]** Solder the  $10k\ \Omega$  resistors and the capacitors to the front of the board. These are on the top right of the board if looking at the front. The capacitors have square footprints on the board while the resistors are rectangles.

Flip the board over

**[4]** Solder the  $10\Omega$  and  $33\Omega$  resistors to the back of the board. The  $33\Omega$  resistor is closest to the top edge of the board.

**[5]** Solder the Raspberry Pi Pico W headers on if you haven't yet. Then solder the Pico to the back of the board. **Make sure the USB port is near the top edge of the board.**

Snip all protruding leads from resistors and capacitors from the board.

**[6]** Solder the aux jack to the back of the board.

**[7]** Flash the Loopster firmware using the “Updating Firmware” section below.

**For Kit Buyers -** skip this step.

If you 3D printed your own parts - insert the screen into the bottom of the screen module, and secure it by screwing in (4) m3 x 3-5 mm screws, directly into the plastic.

**[8]** Place the screen headers onto the screen module pins (4 pin female header), and place it on the board. Solder the headers in while holding the screen in place.

**For Kit Buyers -** the top part of the screen module may not be fully pressed on - don't press it on yet.

Do an initial test - plug the loopster into power and make sure the screen turns on, and that turning the encoder does something.

OPTIONAL - Unplug the screen and clean excess flux from the PCB with isopropyl alcohol. Be careful not to get too much into the keyswitches, it can dissolve some of the lube inside. Not the biggest mistake - they will still work, just changes the feel.

**[9]** Plug in the screen module and **make sure the screen turns on one final time before you fix anything into place.** use super glue to fix the bottom of the module to the board.

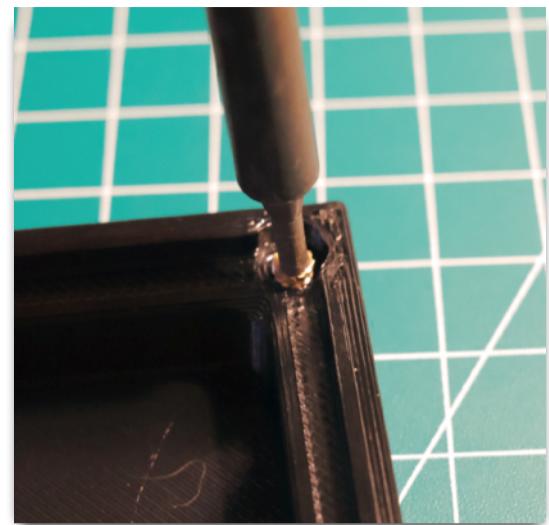
If you 3D printed your own parts - place the top of the screen module onto the bottom.

Gently press the top of the screen module down. See the reference diagram for this step - it is fully down in that picture.

If you 3D printed your own case - Use a soldering iron to melt a m3 heat-set brass insert into each corner of the case (4 total). Place the assembled PCB in and attach with m3 x 3mm screws.



You "Geek on" now.



*Use soldering iron to melt  
heat insets into the case.*