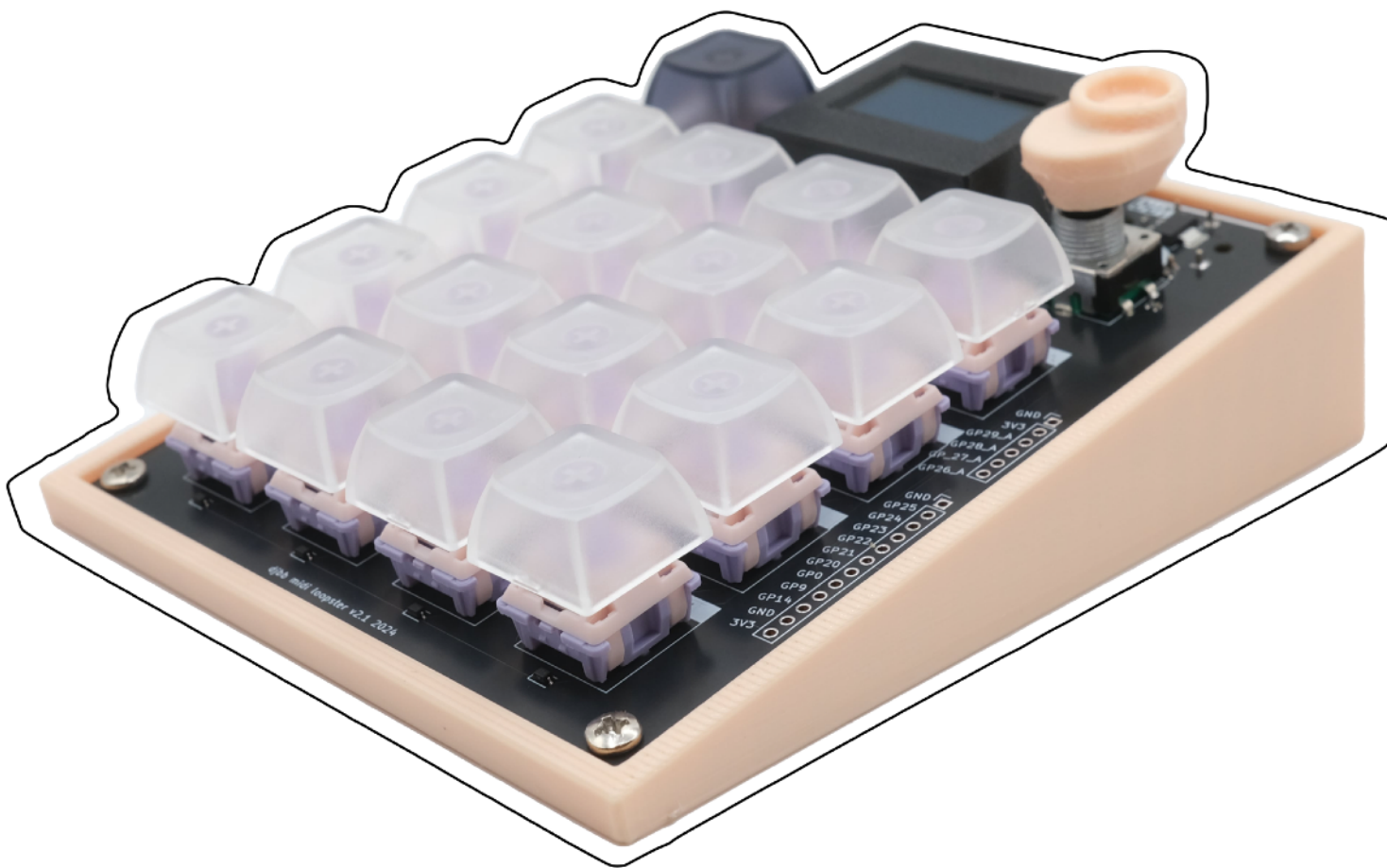


DJBB

MIDI LOOPSTER 2

User Manual

Last Updated: 6/19/2025



The Loopster 2 is a midi controller / recorder / looper. It features a compact design with full-sized MIDI I/O + USB MIDI, mechanical keyswitches, and per pad LED indicators.

Features:

- Full-sized MIDI In and Out & USB MIDI (micro usb)
- MIDI Looper - record loops and one shots of notes, cc messages, or both
- Mechanical keyswitches (*not velocity sensitive*)
- Scale filters, velocity map mode
- Encoder Arpeggiator - hold down pads, cycle through recorded notes & CC messages
- Presets - Save and load settings & recorded pad loops
- Per-Loop Midi Channels
- MIDI Passthrough Mode
- Open Source & easy to extend via extra GPIOs, Python

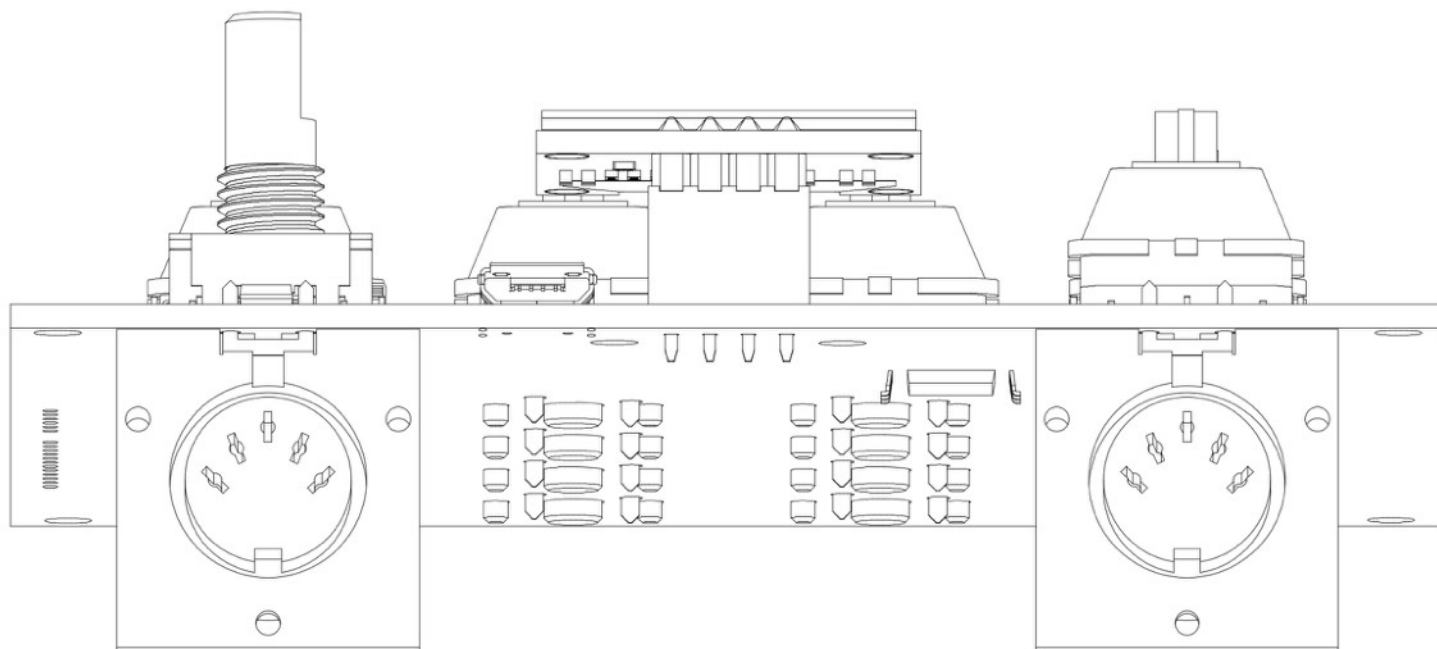


Table of Contents

Links & Resources	4
Quick Reference	5
Using the Loopster	6
1) Play Screen	6
2) Scale Screen	9
3) MIDI Settings and 4) Other Settings	9
5) Load Preset	9
6) Save Preset	9
Updating Firmware	10
Feedback / Reporting Issues	11
Technical	12
Modifying Code	12
Extending the Loopster 2	12
Appendix A: List of Settings	15
Appendix B: Recorded Loop Data Structure	17

Links & Resources

Github: <https://github.com/derrickthomin/DJBB-Loopster-2.0>

STL Files: <https://www.printables.com/model/1066847-djbb-midi-loopster-20-case-and-screen>

Submit Issue / Idea: <https://github.com/derrickthomin/DJBB-Loopster-2.0/issues>

Quick Reference

Play Mode	Desired Action	Buttons
All	Switch Play Modes	Double Click FN Button
(CHD)	Record events to a pad	Hold [fn] and press a pad to begin recording midi events. Click [fn] to stop recording.
(CHD)	Change from loop to 1-shot	Hold pad with recorded midi events and turn the encoder.
(CHD)	Change record quantization settings	Hold [fn] and turn encoder to change value Hold encoder button and turn to change amount.
(VEL)	Set velocity of individual pad(s)	Hold pad(s) and turn encoder
(VEL)	Map one note across 16 velocities	Hold [fn] and click a pad. Do the same to undo.
(ARP)	Cycle through arp notes / ccs	Hold pads and turn the encoder. Pads with loops recorded will add all midi events in the loop to the arp sequence
(ARP)	Change arpeggiator settings	Hold [fn] and turn encoder to change arp type Hold encoder button and turn to change length (gate)

Pad Color	Meaning
Yellow	Pads: Note Playing
Blue	CC Playing
Purple	Loop Exists on Pad
Red	Recording
Green (Solid)	Loop Playing
Green (Blinking)	Loop Queued for Playback (MIDI Sync is ON but Loopster has not received a play message yet)
Yellow (encoder LED)	External MIDI Input Received (Notes)
Blue (encoder LED)	External MIDI Input Received (CCs)

Using the Loopster

To change settings on the current screen, turn the encoder. To move between screens click the encoder once to enter NAV mode, then twist to navigate.

To lock the encoder so that you don't accidentally change something, double click the encoder button. This can be useful when using the arpeggiator and other functions.

1) Play Screen

This is the default startup screen. Here you can change MIDI banks, play modes, and configure quantization and arpeggiator settings. By default, the Loopster 2 starts in Chord Mode. You can see the current play mode by looking at the bottom of the screen.

Note that some modes have additional settings displayed on the screen, like the arpeggiator type and gate length shown below. To change these settings directly from the play screen, hold either the [fn] button or the encoder button, and twist the encoder.



Play mode screen with the ARP play mode.

Play Modes

Double click the [fn] (top left) button to change play modes. The change will be reflected on the bottom of the screen

(CHD) Chord Mode

This is the default mode. In this mode you can record MIDI loops to a specific pad for playback. You can record up to 500 notes and 1500 CCs per pad. At full CC resolution, this comes out to about 50 full twists of a pot at medium speed. The overall limit for all recorded events is 5000.

Hold [fn] and click a pad to begin recording. To stop recording, click [fn] again. To change record quantization time, hold [fn] and turn the encoder.

To change quantization amount, hold the encoder button and turn the encoder. See additional recording settings in the **MIDI Settings** section.

To change the recorded chord or sequence from a loop to a one-shot, hold the pad and turn the encoder. By default it will record to a loop. The default otherwise is whatever mode was set last.

To start/stop playback, click the pad with recorded notes / ccs. Buttons with recorded MIDI will be purple, and green when currently playing. If MIDI sync is on, a blinking pad means queued for playback (once transport from an external MIDI source is sent eg. pressing play in Ableton).

In oneshot mode, cc events will play back immediately, regardless of other settings. So for each cc number with a recorded value, the last recorded value for that number is sent immediately when playback starts. Think of it as a snapshot of the last position of the knobs you were turning before recording was stopped.

(VEL) Velocity Mode

To change the velocity for a specific pad, hold that pad (or multiple) and turn the encoder.

Holding the [fn] button and pressing a pad will map that MIDI note across 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.*

(ARP) Arpeggiator Mode

Hold a pad and turn the encoder to cycle through the arpeggio. If you hold a pad with anything recorded to it, all of the midi events (notes / ccs) recorded are added to the arpeggio.

To change the arpeggiator type, hold [fn] and turn the encoder. To change the length of each arp note (gate), hold the encoder button and turn the encoder.

Note that when switching to the arpeggiator mode, the encoder is locked. Holding pads and turning it will still play notes, but turning it on its own will do nothing. This is to prevent accidental bank/screen changes when using ARP mode. Double click the encoder button to unlock, or change to another play mode.



*Play screen with arpeggiator
mode selected*

2) Scale Screen

Turn the encoder to change the scale. Click the [fn] button, or hold [fn] and turn the encoder to change the root note.



Scale screen

3) MIDI Settings and 4) Other Settings

Turn the encoder to change the value of the setting. Hold **[fn]** and turn the encoder to scroll through settings.

While the Midi Settings screen is selected, hold a pad and turn the encoder to change the assigned midi channel for midi events recorded to that pad.

See [Appendix A](#) for the full list of settings

5) Load Preset

Use the encoder to select a preset. Press [fn] to load it.

6) Save Preset

Use the encoder to select a preset, or select *NEW*. Press [fn] to overwrite/save. Any recorded loops on pads will be saved as well. You cannot change the name of these presets on the Loopster, however you can edit the presets.json file to change the name if you wish.

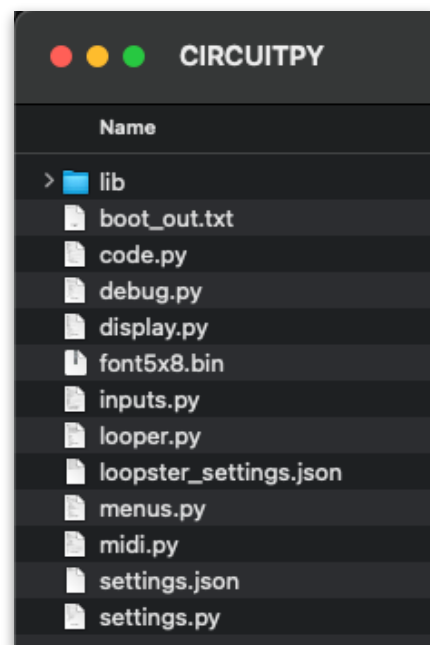
Midi Settings, Other Settings, and midi events recorded to pads will be saved with the preset.

Updating Firmware

The Loopster 2 runs on CircuitPython¹, a lightweight version of the Python programming language. The “operating system”, if you will.

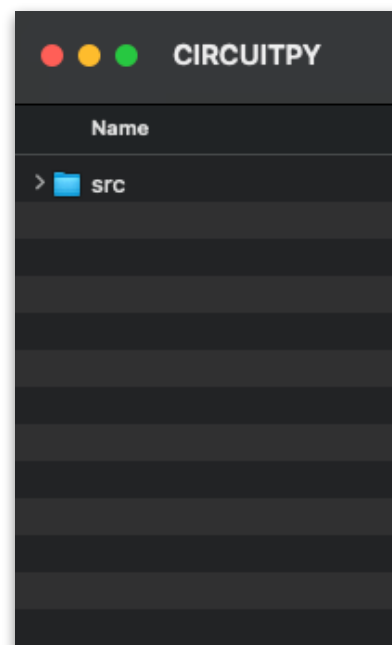
1. Go to the Loopster [2 GitHub repo](#) and download it as a ZIP file.
2. Hold the [fn] button and continue holding it while plugging in the Loopster 2. It should show up just like a USB drive on your computer. If not, make sure the USB cable is not power only.
3. Open the **src** folder and drag all files/folders to the Loopster 2 root directory. If prompted to overwrite files, select yes.

YES



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

NO



... not the whole folder

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

Feedback / Reporting Issues

If you find any bugs or issues, please report them via the GitHub link below. If you have any enhancement / new feature ideas, I'd also love to hear!

Report Issues and Ideas Here: [https://github.com/derrickthomin/DJBB-Loopster-2.0/
issues](https://github.com/derrickthomin/DJBB-Loopster-2.0/issues)

Technical

Only read on if you are interested in modifying the Loopster and digging into the code. Otherwise, just go have fun with it.

Modifying Code

To be able to save updates to the loopster, it must be in “read only” mode. More info can be [found here](#). Note that it’s a bit confusing - read only in this context means read only to the *device* modifying it’s own data (such as saving a preset);

To start the Loopster 2 in read-only mode, hold the [fn] button, and plug it into your computer while continuing to hold. Once it boots up, you should be able to move files to/from the Loopster 2 just like a USB drive. Note that in read only mode, presets cannot be saved (again, see the link listed above).

Extending the Loopster 2

[Here is a YouTube video with some examples of what you can do.](#) Code snippets for each of these add ons can be found in `useraddons_examples.py` in the GitHub src folder. If you’d like, you can 3D print the breadboard edition of the case - [find the STL file here](#).

The Loopster 2 has 13 (including 4 analog) GPIO pins broken out on the right side of the board. These can be used to add your own peripherals to the Loopster, like additional knobs or buttons, LEDs, and pretty much whatever you want. The Loopster is built on the RP2040 MCU by the Raspberry Pi foundation. Anything you can do with that in CircuitPython, you could add onto the Loopster 2.

Available Digital Pins	Available Analog Pins
GP0, GP9, GP14, GP20, GP21, GP22, GP23, GP24, GP25	GP26, GP27, GP28, GP29



Examples are available in `useraddons_examples.py` for the following:

Neopixels
XY Joystick
Button for Shifting Octaves
Potentiometer for changing MIDI velocities
Encoder for changing MIDI channels
Photoresistor for changing MIDI velocities
Control a Motor using PWM
Play sounds with a piezo via PWM
Trigger relay switches to power high powered externals
PIR sensor to shift octaves
Accelerometer / Gyroscope to control modulation (send CC messages)
Update a 7 segment display
Use a DC motor as an input

To add your own externals, write a function in `useraddons_examples.py`, and then call that function in one of the “hooks” below. Each of these “hook” functions are already called in the main Loopster 2 code, so you don’t need to modify any other file to run your custom code. Again, see `useraddons_examples.py` for a bunch of examples.

Here are the 4 “hooks” where you can call your custom code

Hook	Description	Example Use Cases
<code>check_addons_slow()</code>	Less frequent calls for non-urgent tasks.	- check a pot to change global velocity - Update an external display
<code>check_addons_fast()</code>	ore frequent calls for time-sensitive tasks.	- shift all midi notes when a button is pressed - change MIDI channel quickly (to get wacky with loops)
<code>handle_new_notes_on(noteval, velocity, padidx)</code>	Triggered when a new note is played. Your function does not need to use them, but these parameters are available for use: noteval: midi value (0-127) velocity: midi velocity (0-127) padidx: index of pad (0-15)	- light up neopixels when a note plays - Play note on piezo buzzer
<code>handle_new_notes_of(noteval, velocity, padidx)</code>	Same as above, but on note off	Same as above, but on note off

Appendix A: List of Settings

Where	Setting	Options	Description
MIDI Settings	Default BPM	60–200 (integer)	Main tempo for loops and clock.
MIDI Settings	CC Resolution	1, 2, 5, 8, 16, 32, 64	Sensitivity for recording CC changes.
MIDI Settings	MIDI Clock Source	AUTO, USB, AUX	Which MIDI port to listen to for clock messages
MIDI Settings	Default Note Velocity	1–127 (integer)	Default note velocity.
Hold pad(s) & turn encoder in MIDI Settings	Pad MIDI Channel	1–16 (per pad) or global	Assign each pad a MIDI output channel for loops.
MIDI Settings	MIDI Input Channel	1–16	Input channel for receiving MIDI.
MIDI Settings	MIDI Output Channel	1–16	Output channel for sending MIDI.
MIDI Settings	MIDI Sync Enabled	On/Off	When enabled, respects start, stop, and clock (for BPM) messages from external device
MIDI Settings	MIDI Passthrough	On/Off	Forward incoming MIDI to DIN output.
MIDI Settings	MIDI Port Type	USB, AUX, ALL	Send/receive MIDI via USB, DIN, or both.
MIDI Settings	Record CC	On/Off	Record MIDI CC (knobs/faders) in loops/chords.
MIDI Settings	DIN MIDI Direction	both, in, out	Set DIN MIDI ports to send, receive, or both.
MIDI Settings	USB MIDI Direction	both, in, out	Set USB MIDI ports to send, receive, or both.
Other Settings	Arpeggiator Note Duration	1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64	How long arpeggiator notes play. Also known as gate.
Other Settings	Arpeggiator Polyphonic	On/Off	If true, previous note is turned off before next arp note plays.

Other Settings	Arpeggiator Pattern	up, down, random, random octave up, random octave down, random start up, random start down	Random octave randomly transposes each arp note up or down an octave (or not at all).
Other Settings	Encoder Steps per Arp Note	1-10	Encoder turns needed to advance arpeggiator.
Other Settings	One-shot Chord: Instant Mode	On/Off	If on, all notes in one-shot chords play at once. Note length is respected - even though all note on messages are sent at once, note off messages are sent according to recorded note length.
Other Settings	LED Brightness	0.0-1.0 (10% steps)	Pad LED brightness.
Other Settings	Default Chord Loop Type	chordloop, oneshot	Default type for new chord recordings. If a chord type is changed in the play menu, that becomes the new default (until device reset)
Other Settings	Quantize (%)	0-100	Quantization strength for recorded notes/CC.
Other Settings	Quantize Division	none, 1/4, 1/8, 1/16, 1/32, 1/64	Subdivision grid for quantizing.
Other Settings	Quantize CC	On/Off	Whether to quantize recorded CC.
Other Settings	Quantize Loop Length	none, 1, 1/2, 1/4, 1/8	Quantizes the loop length.
Other Settings	Startup Menu	1-6 (shows first menu on boot)	Sets which menu shows when device powers up.
Other Settings	Trim Silence	start, end, none, both	Removes silence at loop start/end.

Appendix B: Recorded Loop Data Structure

Perhaps you want to programmatically create pad loops, or modify them in some way after recording. If so, review how things are linked up and loaded below.

When a preset is saved, a CSV file is saved to the "chords" folder in the root directory of the Loopster. This file is then referenced in the preset in presets.json.

Events (notes and CCs) for each pad are listed under their respective ##PADX## header. This is how the loopster knows where to load the events.

For Note on and Note Off, each note has the structure:

<Midi value>, <velocity>, <note pad index>, <ticks>

For CCs:

<CC Num>, <CC Val>, 0

On the next page is an example chord.csv file. It contains data for 2 pads, where the first pad has notes and CCs, and the second has just CCs.

Example chord.csv file

```
##PAD0##  
#METADATA#  
loop_type,oneshot  
ticks,209  
time,5.464  
bpm,120.0  
#NOTES_ON#  
79,120,4,0  
81,120,5,56  
#NOTES_OFF#  
79,127,4,4  
81,127,5,60  
#CC#  
102,125,0  
##PAD1##  
#METADATA#  
loop_type,oneshot  
ticks,68  
time,2.767  
bpm,120.0  
#NOTES_ON#  
#NOTES_OFF#  
#CC#  
102,124,0  
102,69,0
```