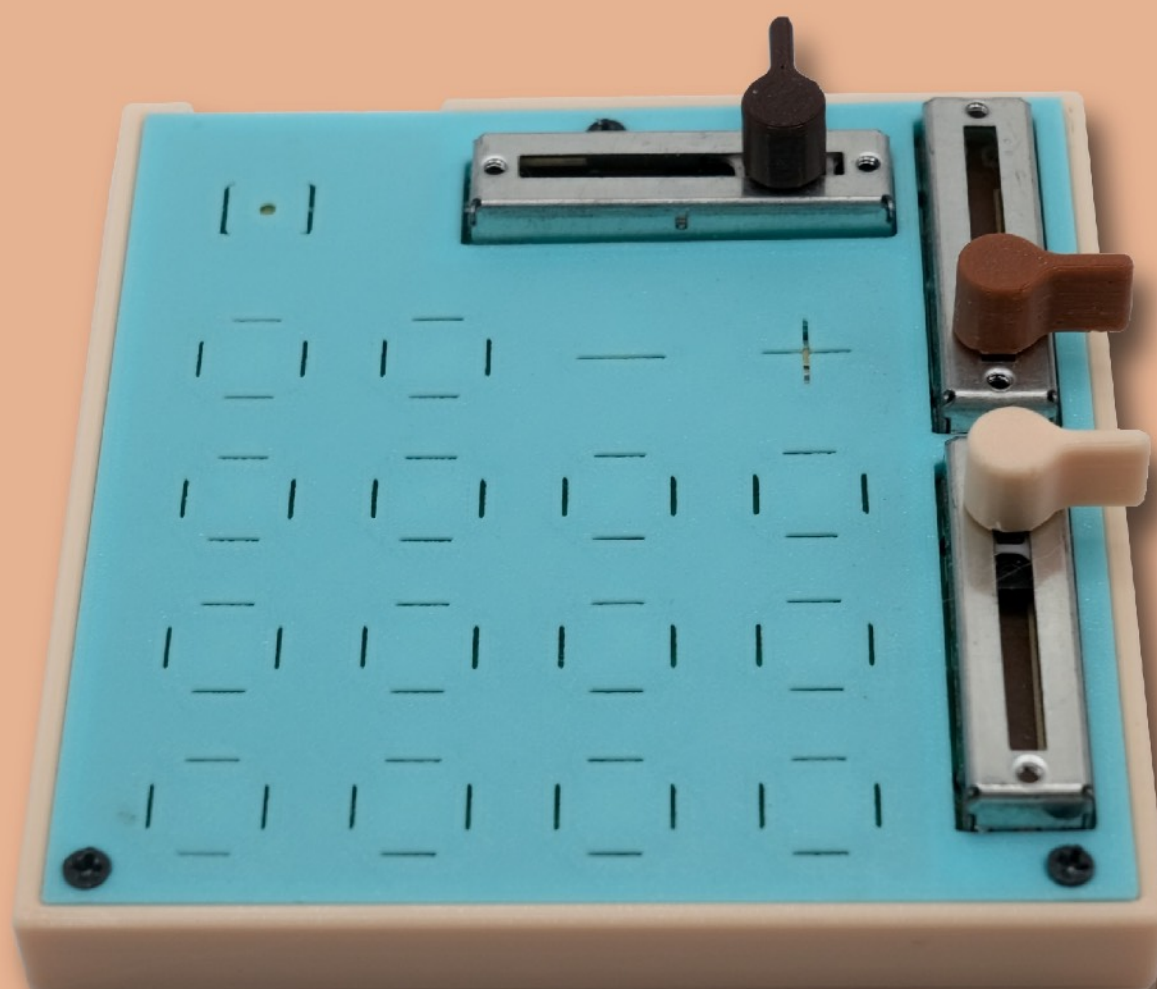


DJBB

MINI MIDI SLIDER 51

User Manual

v1.1 Last Updated: 8/17/2024



The MMS51 is a portable USB MIDI control surface featuring up to 51 mappable sliders.

Features:

- 16 x Drum Pads w/ RGB LEDs (membrane style keys)
- Can latch multiple to send CC messages for each
- Control up to 51 parameters with sliders (3 global, 48 pad-specific - 3 per)
- Small - great for travel
- CC Mode - Latch buttons to send pad-specific CC messages with sliders

Resources

Videos: <https://www.youtube.com/watch?v=U5nADOcyAg4>

<https://www.youtube.com/watch?v=O2LY8Df0sLk>

Github: https://github.com/derrickthomin/micro_midi_slider_pico

STL Files: <https://www.printables.com/model/945561-djbb-mini-midi-slider-51-case>



User Manual

See next page for reference diagram / one pager

Note Mode

- Send Note = press PAD button
- Send global CC messages = move SLIDERS
- Send pad-specific CC messages = hold PAD and move SLIDERS
 - Pad light turns blue to indicate that CC messages will send for that pad by moving sliders
- Change MIDI Bank = Hold FN button and press -/+ keys (top right two pads).

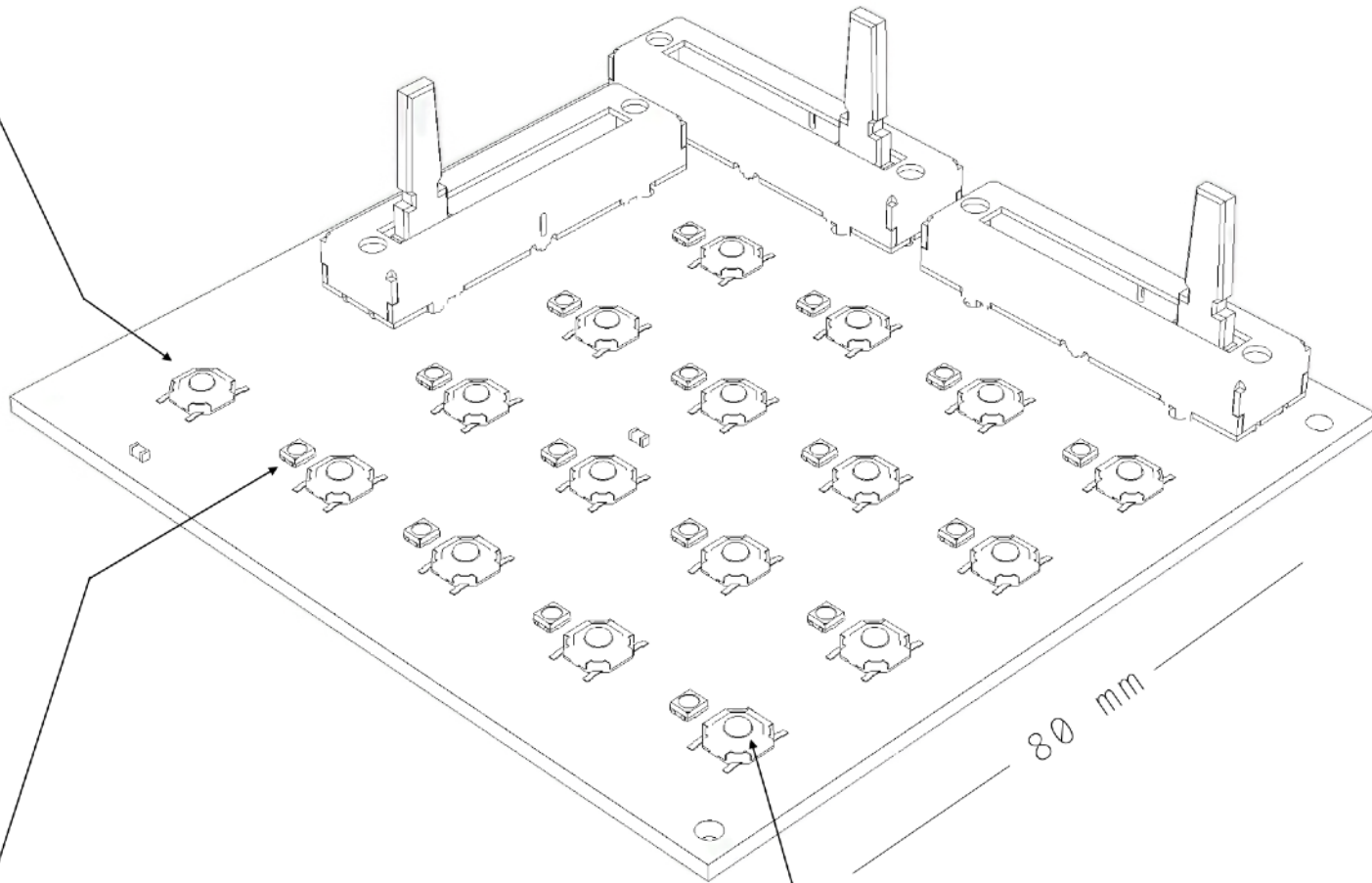
CC Mode

- Enter / Exit CC mode = double click FN button
- Latch / Unlatch buttons to send CC messages for = Click PAD buttons
- Send global CC messages = move sliders when NO pad is latched.
- Send pad-specific CC messages = move sliders while PAD buttons are latched (blue light above)
- Each slider sends a different CC message per pad latched. If 4 pads are latched and slider 1 is moved, 4 CC messages are sent at once.

DJBB MINI MIDI SLIDER 51

- Double click to switch between note / CC modes
- Hold and click **+**/**-** button to change banks

FN



PAD X16

NOTE MODE

- Click to send note
- Hold and move sliders to send pad-specific CC messages

IN CC MODE

- Click button to latch / unlatch
- Move sliders to send pad-specific CC messages for each latched button

LED X16

Yellow = Sending Note

Blue = Send pad-specific CC message for this pad when sliders are moved

Technical

Customizing Settings

You can customize MIDI channel, and which messages are sent by editing **cc_vals.json** in the root directory of the device. Plug it in, locate it in Finder / Explorer, and open cc_vals.json in any text editor. The default settings are shown below.

cc_vals.json default values:

```
{
  "SLIDER_CC_VALS_GLOBAL":[3, 9, 85],
  "SLIDER_CC_VALS_HELD": [
    [10, 11, 12], [13, 14, 15], [16, 17, 18], [19, 20, 21],
    [22, 23, 24], [25, 26, 27], [28, 29, 30], [31, 32, 33],
    [34, 35, 36], [37, 38, 39], [40, 41, 42], [43, 44, 45],
    [46, 47, 48], [49, 50, 51], [52, 53, 54], [55, 56, 57]
  ],
  "MIDI_CHANNEL": 1
}
```

SLIDER_CC_VALS_GLOBAL Determines CC when no button is held

SLIDER_CC_VALS_HELD Determines CC for each button when held / latched.

MIDI_CHANNEL Output MIDI Channel (1-16)

Unplug and re-plug the device in to see the changes.

WARNING: No validation is done on the values entered here. They must be valid CC values, and the format must stay exactly the same (don't delete any values completely, remove commas, etc. If you mess it up, just copy and paste the above back into cc_vals.json and restart the device.

Loading New Firmware

To load the newest firmware, download the GitHub repository (see links at top of this document), unzip it, and copy everything from the "src" folder to your device. Overwrite existing files if asked. For convenience, you can download just the src folder [via this link](#).

Code

To modify the code, plug it into your computer and locate it in Finder / Explorer. The root folder contains all of the code. code.py is what runs when the device is plugged in - start there to see how it works! It's programmed in CircuitPython which you can find many guides for online - it's a very beginner friendly programming language.

Troubleshooting

If it lights up but your computer cannot detect it, ensure your micro usb cable is not power-only.

FAQ

Q: Membrane style keys... why??

A: The tiny buttons are hard to press without some sort of keys on top. I also wanted to diffuse the LED light so it wasn't so harsh. PETG membrane keyboard was the only thing that I could come up with that I could produce myself.

Q: Is the 3D printed keypad really going to hold up?

A: Yes - it's made from PETG which is flexible, durable, and heat resistant (up to 85C / 185F before risk of warping)

Q: I have an idea for a feature... how do I tell you

A: [Submit a new issue here](#)