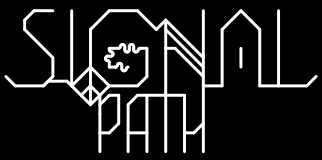


Knobster™

OPERATOR'S
MANUAL

8 KNOBS / 4 BUTTONS / USB MIDI

HELLO

The Knobster is a USB MIDI controller with eight potentiometers and four buttons. It is class-compliant: plug it into any computer and it appears as a MIDI device — no drivers.

It is also yours, all the way down. The circuit board is the product, the firmware is open, and the bootloader will always let you reflash it. Page 10.

IN THE BOX

- Knobster (the black thing with knobs)
- This manual
- Stickers: Signal Path logo + NOT A TOY
- Serial number card (keep it — it identifies your unit for support)

YOU NEED

- A USB-C cable (data, not charge-only)
- A computer with a DAW, synth, or anything that speaks MIDI

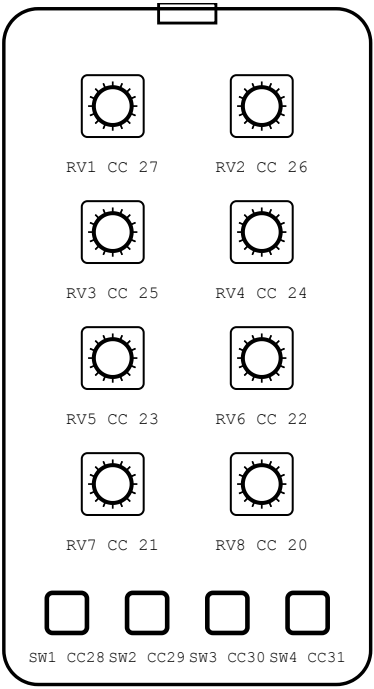
QUICK START

1. Connect the Knobster to your computer with a USB-C cable.
2. It enumerates as a MIDI device named **Knobster**.
3. In your DAW, enable it as a MIDI input.
4. Turn a knob. You are now sending MIDI CC messages on channel 1.
5. Use your DAW or synth's MIDI Learn: wiggle a knob while the target parameter is armed, and it maps.

SERUM TIP

The default CCs 20–27 are chosen to map cleanly onto Serum's eight macros — MIDI-learn each macro once and save it as your default preset.

LAYOUT



Default map, MIDI channel 1. Every CC is reconfigurable.

DEFAULT MIDI MAP

Control	CC	Behavior
Knob 1 (top-L)	27	absolute 0–127
Knob 2 (top-R)	26	absolute 0–127
Knob 3	25	absolute 0–127
Knob 4	24	absolute 0–127
Knob 5	23	absolute 0–127
Knob 6	22	absolute 0–127
Knob 7 (bot-L)	21	absolute 0–127
Knob 8 (bot-R)	20	absolute 0–127
Button 1	28	momentary
Button 2	29	momentary
Button 3	30	momentary
Button 4	31	momentary

Knobs send only when moved, with filtering and hysteresis tuned so values never jitter at rest but no position is unreachable.

THE CONFIGURATOR

Free desktop app for macOS, Windows, and Linux:

signalpath.audio/configurator

Everything lives on the device itself — configure once, use anywhere.

- **MIDI channel** 1–16
- **Any CC per knob/button** (0–127)
- **Button modes** — see next page
- **Knob range trim** (ADC endpoints)
- **Response feel** (filter strength, debounce)
- **Save to flash** — settings persist without the app
- **Import / export JSON** — share setups

BUTTON MODES

MOMENTARY

CC 127 on press, CC 0 on release. Like a sustain pedal: active while held.

TOGGLE

Each press flips between 127 and 0. Like a stomp-box switch: tap on, tap off.

TRIGGER

CC 127 on press, nothing on release. A one-shot — fire a sample, advance a preset, tap tempo.

Set each button’s mode independently in the configurator, then Save to Flash.

GOOD TO KNOW

POWER

USB bus powered, <100 mA. No battery, no power switch, no wall wart. If the cable carries data, it carries enough power.

ON CONNECT

When your DAW opens the device, the Knobster sends every control's current position once — your mix picks up exactly where the knobs physically are.

BARE SHAFTS

No knob caps, on purpose — the knurled shafts grip fine and stay honest. If you want caps, any standard push-on cap for a 6 mm knurled shaft fits.

NO SCREEN, NO LEDS

On purpose. The knob positions are the interface. The values you hear are the truth.

TROUBLESHOOTING

Symptom	Try
Not in MIDI device list	Different cable (must be data-capable); different port; no hubs
Knob moves wrong parameter	Re-run MIDI Learn, or remap the CC in the configurator
Values jump at rest	Raise filter strength (EMA) in the configurator
Knob can’t reach 0 or 127	Widen ADC endpoints in the configurator
Device unresponsive after custom firmware	Recovery, page 11

Still stuck? hello@signalpath.audio — include your device UUID from the configurator’s info panel.

MAKE IT YOURS

The Knobster runs open firmware on an STM32, and you can flash your own code **over USB** — no programmer hardware.

1. Get the SDK: **signalpath.audio/sdk**
2. Write your behavior in C — pitch bend knobs, chord buttons, MIDI notes, whatever you can imagine in 27KB
3. Configurator → Developer → **Reboot to bootloader**
4. Flash:

```
dfu-util -a 0 -s 0x08001000:leave -D your_firmware.bin
```

The protected boot stub and your device's identity page are untouched by this command.

RECOVERY

YOU CANNOT BRICK IT

The first 4KB of flash is a boot stub that always runs first. If your custom firmware misbehaves — or you just want stock back:

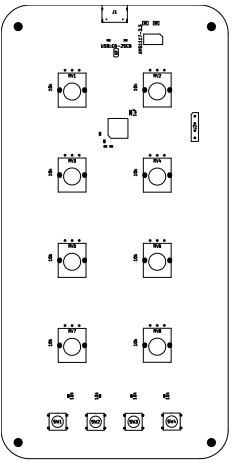
1. Unplug the Knobster.
2. **Hold Button 1 (SW1).**
3. Plug in while holding. The device enumerates as **STM32 BOOTLOADER.**
4. Flash stock firmware (download at signalpath.audio/firmware):

```
dfu-util -a 0 -s 0x08001000:leave -D knobster-stock.bin
```

The boot stub also takes over automatically if the app region is corrupt.

SPECIFICATIONS

Size	160 × 80 mm bare board
Controls	8 × 10k pots (Alps RK09K), 4 buttons
Interface	USB 2.0 FS, class-compliant MIDI
Connector	USB-C
Power	bus powered, <100 mA
Brain	STM32F042, 48 MHz, no crystal
Resolution	12-bit sampling → filtered 7-bit CC
Firmware	open source, user-flashable (USB DFU)



CASE DESIGN

Want an enclosure? Design your own — we publish full 3D models (STEP + STL) and a dimensioned drawing:

signalpath.audio/case

KEY DIMENSIONS (mm)

Board	80 × 160 × 1.6, corners R10
Mounting	4× M3 (Ø3.2), 68 × 146.7 spacing
Shafts	Ø6 × 20 tall, 30×30 grid, cols x=30/60, rows y=15.5+30n
Buttons	y=144.75, pitch 13.4, 7.3 tall
USB-C	top edge, center x=40, 3.3 tall
Under board	leads ≤2.5 — standoffs ≥2.5

Panel holes Ø7 min for shafts. Origin: board top-left corner. Share what you build — we'll feature it.

PROGRAM IT WITH AI

You don't have to write C. The SDK repo is set up so an AI coding agent (like Claude Code) can do it for you — it knows the hardware, the rules, and how to build:

github.com/ryanalexmartin/knobster-sdk

1. Clone the repo, open your agent inside it.
2. Say what you want. For example:

> "Make the four buttons select banks so my 8 knobs control 32 different CCs, with soft takeover. Build it and give me the flash command."

> "Make the buttons send drum-pad notes and let knob 8 set MIDI clock tempo. Verify it fits in flash."

3. Hold SW1, plug in, run the dfu-util command it gives you.

More prompts in **PROMPTS.md**. Remember: you cannot brick it (page 11).

NOT A TOY

designed by
Ryan Alex Martin
Michael Spurlock
Adrian Gariba