



User Manual

This notice includes information related to your rosco_m68k keyboard (revision one) product. Please read this sheet carefully and retain for your reference.

1. Thank you For Your Purchase!

Thank you for purchasing the rosco_m68k keyboard!

We're sure you'll be delighted with your new retro-and-homebrew-focused fully mechanical keyboard!

This document contains important information about how to build and use your new keyboard – please read it carefully before you build and keep it handy so you can refer back to it later.

You can always find the latest version of this document online at <https://rosco-m68k.com>

2. Switches & Stabilizers

If you purchased your keyboard PCB-only, this section contains some specifications on the additional hardware you will need to source.

The keyboard is designed for use with Cherry MX-compatible switches - we recommend Gateron KS-9-CAP 5-pin switches. We also recommend that 3-pin switches are avoided unless a plate is to be used for strength and stability.

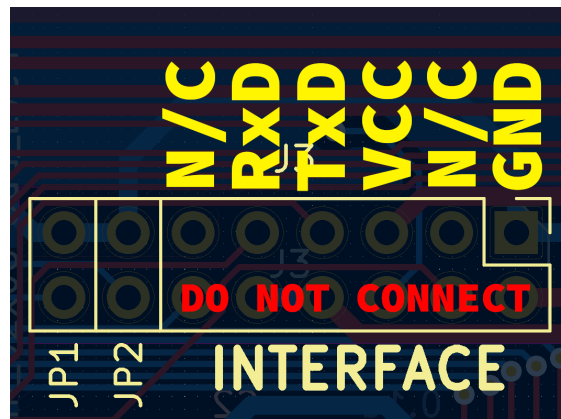
The board directly supports Cherry-compatible screw-in PCB mount stabilizers – we test with genuine Cherry hardware, which is commonly available online. You will require:

- 1 x 6.25u Space bar wire
- 3 x 2u modifier wires
- 8 x Stabiliser housings and stems
- 8 x Screws

Any commodity Cherry-compatible keycap set should work fine, assuming it contains the right number of keys of the appropriate size. We've found that 104-key PC sets work well, and are available from Amazon – though you may find you need to sub a 1u key for one of the 1.25u (in which case we tend to swap out right-command).

3. Connection to rosco_m68k

To connect the keyboard to your rosco_m68k, the following pins on the INTERFACE connector are used:



This pinout matches the pinout on the rosco_m68k main board, and also on commonly-available UART adaptors. We recommend mounting a right-angle 01x06 male header on the keyboard and using a straight six-way cable with DuPont connectors.

Connect to UART B on the rosco_m68k. Firmware 2.42 required for autodetection.

4. Jumper Settings

The board has several jumpers for advanced use-cases:

JP1 – Speed Selection

Open	115,200bps
Short	9600bps

JP2 – Startup Mode Selection

Open	UART Mode
Short	Scancode Mode

JP3 – Passthrough Mode (See below)

Open	No Passthrough
Short	Passthrough Enabled

JP4 – Do Not Short

This jumper is used for testing at the factory, may cause malfunction if shorted.

JP5 – Do Not Short

This jumper is used for testing at the factory, may cause malfunction if shorted.

JP6 – Do Not Short

This jumper is used for testing at the factory, will cause malfunction if shorted.

5. Pass-through mode

In advanced usage, you may wish to connect the keyboard to a device that only supports a single serial channel, and where that channel is also used for terminal output / uploading programs and so on.

Passthrough mode is designed for this purpose.

To use Passthrough mode, short JP3, and connect the keyboard to your computer as normal.

Then connect your usual terminal equipment (or USB-Serial converter) to J6. Once this connection is made, the keyboard will act as a transparent device, unless keys are being sent.

Obviously when transferring files etc using this feature, you should not press keys during the transfer as the keyboard is not disabled – and the keypresses will be inserted into the data stream which could cause the transfer to fail.

In pass-through mode, commands are disabled – any commands sent will be simply passed through without being processed by the keyboard.

Note that the TX/RX pins should be swapped when connecting to USB-Serial adapters.

Note also that the pass-through serial will operate at the same bitrate as the main interface – either 9600 or 115200 depending on JP1.

6. Scan-code Format

In scan-code mode (activated via an appropriate command, or at startup by shorting JP2) the keyboard simply sends make/break codes for individual keys as the matrix scan progresses. In this mode, there are no limitations on which keys can be pressed and held at the same time. All codes are 8-bit unsigned integers with the following format:

*	Down?	Row				Column			
*	-----	-----				-----			
*	1	1	1	1	1	1	1	1	1
*	-----	-----				-----			
*	MSB								LSB

Row and column are 1-based in this scheme.

7. Key Mappings in UART mode

No CAPS LOCK, Unshifted

```
27, '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '-', '=', 8, 0,
'\t', 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', '[', ']', '\\', 0,
0, 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l', ';', '\'', 0, '\r', 0,
0, '\'', 'z', 'x', 'c', 'v', 'b', 'n', 'm', ',', '.', '/', 0, 0, 0,
0, 0, 0, 0, 0, 0, ' ', 0, 0, 0, 0, 0, 0, 0, 0,
```

No CAPS LOCK, Shifted

```
27, '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '_', '+', 8, 0,
'\t', 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', '{', '}', '|', 0,
0, 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', ':', '"', 0, '\r', 0,
0, '~', 'Z', 'X', 'C', 'V', 'B', 'N', 'M', '<', '>', '?', 0, 0, 0,
0, 0, 0, 0, 0, 0, ' ', 0, 0, 0, 0, 0, 0, 0, 0,
```

CAPS LOCK, Unshifted

```
27, '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '-', '=', 8, 0,
'\t', 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', '[', ']', '\\', 0,
0, 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', ';', '\'', 0, '\r', 0,
0, '\'', 'Z', 'X', 'C', 'V', 'B', 'N', 'M', ',', '.', '/', 0, 0, 0,
0, 0, 0, 0, 0, 0, ' ', 0, 0, 0, 0, 0, 0, 0, 0,
```

CAPS LOCK, Shifted

```
27, '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '_', '+', 8, 0,
'\t', 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', '{', '}', '|', 0,
0, 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l', ':', '"', 0, '\r', 0,
0, '~', 'z', 'x', 'c', 'v', 'b', 'n', 'm', '<', '>', '?', 0, 0, 0,
0, 0, 0, 0, 0, 0, ' ', 0, 0, 0, 0, 0, 0, 0, 0,
```

CONTROL (disregards shift / caps modifiers)

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 17, 23, 5, 18, 20, 25, 21, 9, 15, 16, 27, 0, '\r', 0,
0, 0, 1, 19, 4, 6, 7, 8, 10, 11, 12, 0, 0, 0, 0,
0, 0, 0, 26, 24, 3, 22, 2, 14, 13, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

8. Command Reference

Please refer to the online documentation at <https://rosco-m68k.com> for a complete up-to-date version of the command reference.

We do not include it here to save paper.

9. Compliance Notices

All information contained in the product documentation (herein and online) and any additional information and documentation (including this notice) is correct as far as possible at the time of writing. Errors & omissions exempt.

To achieve compliance with local regulations regarding electro-magnetic interference (both transmission and receipt) the product may need to be operated in a suitable grounded enclosure with appropriate application-specific shielding. The Really Old-School Company Limited neither specify nor supply such enclosures and recommend that expert guidance be sought where an enclosure is to be used.

The Really Old-School Company Limited does not authorize the use of any of its products in safety critical or life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the safety critical or life support system or to significantly affect its safety or effectiveness. This includes, but is not limited to, human life support, nuclear safety and control, air-traffic control, and vehicular control.

Products are not authorized for use in such applications under any circumstances.

All PCBs and components we supply are compliant with Restriction of Hazardous Substances in Electrical and Electronic Equipment (RoHS) regulations. Components that you source may not be compliant with the modern regulation, and compliance in finished kits you build will also depend on your choice of solder when building your board. Please dispose of any waste in accordance with relevant Waste Electrical and Electronic Equipment recycling (WEEE) regulations in your jurisdiction.