



RoverMux User Guide

1 Introduction

Take control of your self-driving rover in emergencies—or any time you like—with the 3-channel RoverMux.

2 Assembly Instructions

Assembly is easy. And, you can learn how to solder at the same time. Review [Sparkfun's Soldering Tutorial](https://learn.sparkfun.com/tutorials/how-to-solder---through-hole-soldering) if you need to. Here's a helpful info-graphic from the tutorial:

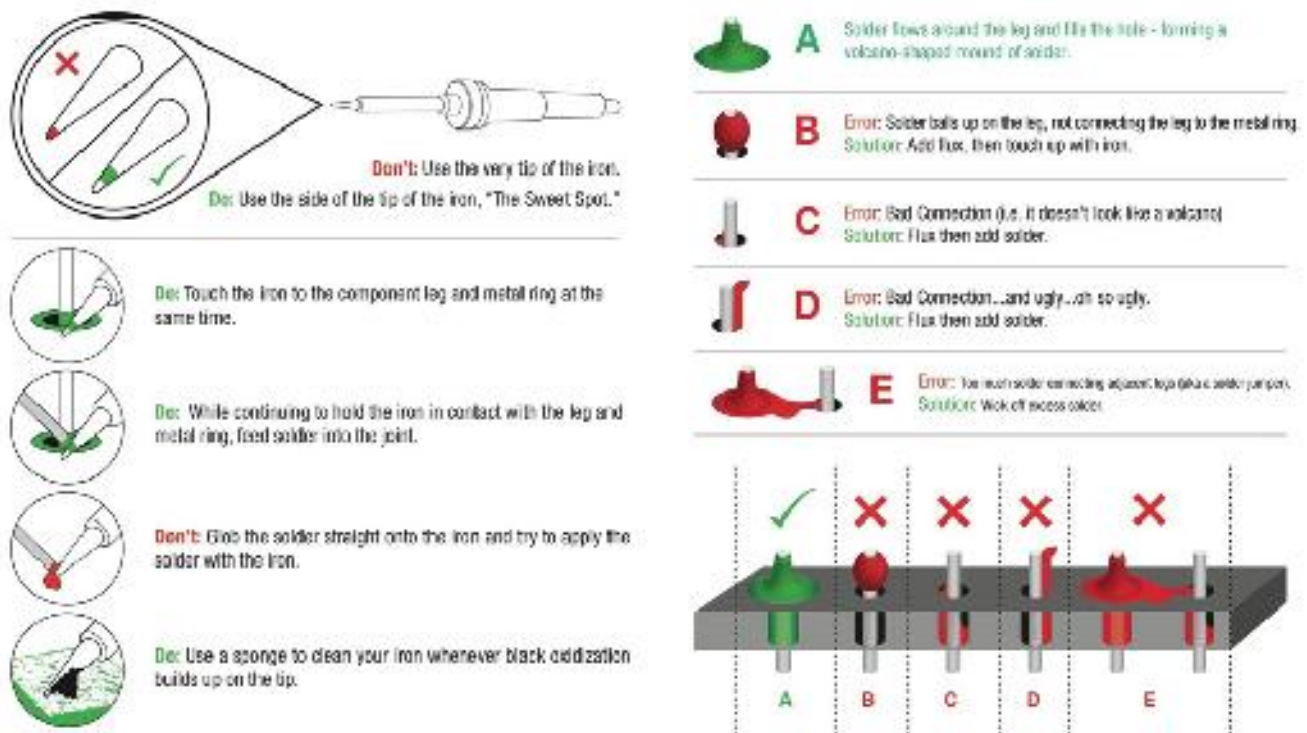


Illustration 1: <https://learn.sparkfun.com/tutorials/how-to-solder---through-hole-soldering>

2.1 You'll need

- Soldering iron, 40W
- Sponge to clean the iron (I recommend a brass sponge)
- Workbench with plenty of light
- Ventilation since breathing flux fumes is irritating
- Soldering surface (e.g., marble tile sample)
- Rosin core solder 0.022" or 0.032" diameter
- Kester #2331-ZX flux pen (optional)

2.2 Pin Headers

Apply flux pen or flux paste to the three rows of pin pads, top and bottom.

Install the long side of the black, red, and yellow/white headers in a breadboard.

Place the RoverMux upside down onto the pin headers so that the black headers are at the edge of the RoverMux board.

Double-check that the yellow headers are in the middle of the board and black headers are at the edge of the board.

Hold down the RoverMux and solder a pin for each header. The ground header will take longer to melt the solder.

Now, solder the remaining pin headers. Again, the ground pins will take a little longer to heat up at first until the ground plane gets sufficiently hot.

2.3 ISP Header

Optionally you can solder on a 2x3 AVR ISP header if you want to flash your RoverMux.

Install the ISP header onto the RoverMux board on the top side.

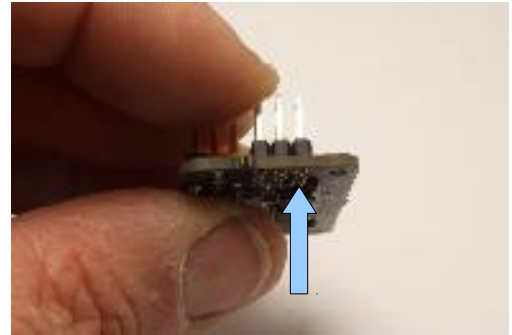


Pick which pin you're going to solder first (blue arrow, right).

Carefully hold the header in place without touching the pin you're about to solder (because it will get really, really hot).

Carefully solder the one pin. Make sure the header is aligned vertically. If not, re-solder/adjust as needed.

You can let go of the pin headers. Continue to solder the remaining pins.



2.4 Cleanup

You'll want to remove the rosin and flux



I usually just use isopropyl alcohol and an old toothbrush

You can also buy chemicals specifically for removing flux and rosin.



3 How RoverMux Works

The following table shows how you can use your R/C Transmitter to switch between manual control (R/C) and autonomous (MCU) control.

Mode	RX CH3 plugged in?	Solder Jumper?	R/C Transmitter	
			ON	OFF
3-Channel	YES		CH3 switches between MCU control and R/C control	MCU in control
2-Channel	NO		R/C in control	

3.1 2-Channel Mode

The simplest mode is 2-Channel. In this mode, when the R/C Transmitter is ON, it is in control of your rover. When the R/C Transmitter is OFF, the MCU is in control of your rover.

3.2 3-Channel Mode

To switch between MCU and R/C control using CH3 of a 3-Channel Transmitter, you must first:

- Connect CH3 on your Receiver (RX) to RoverMux 3R/C
- Close the Solder Jumper on top of the RoverMux board (green arrow, right)



When the Transmitter is ON, the CH3 switch (blue arrow, right) toggles between MCU control and R/C control. The example picture shows a FlySky FS-GT3B transmitter.

The MCU is always in control when the Transmitter is OFF.



3.3 Adjusting Channel 3

You may need to adjust the CH3 Reverse, Endpoint, and Trim settings on your transmitter for RoverMux to recognize the two CH3 modes.

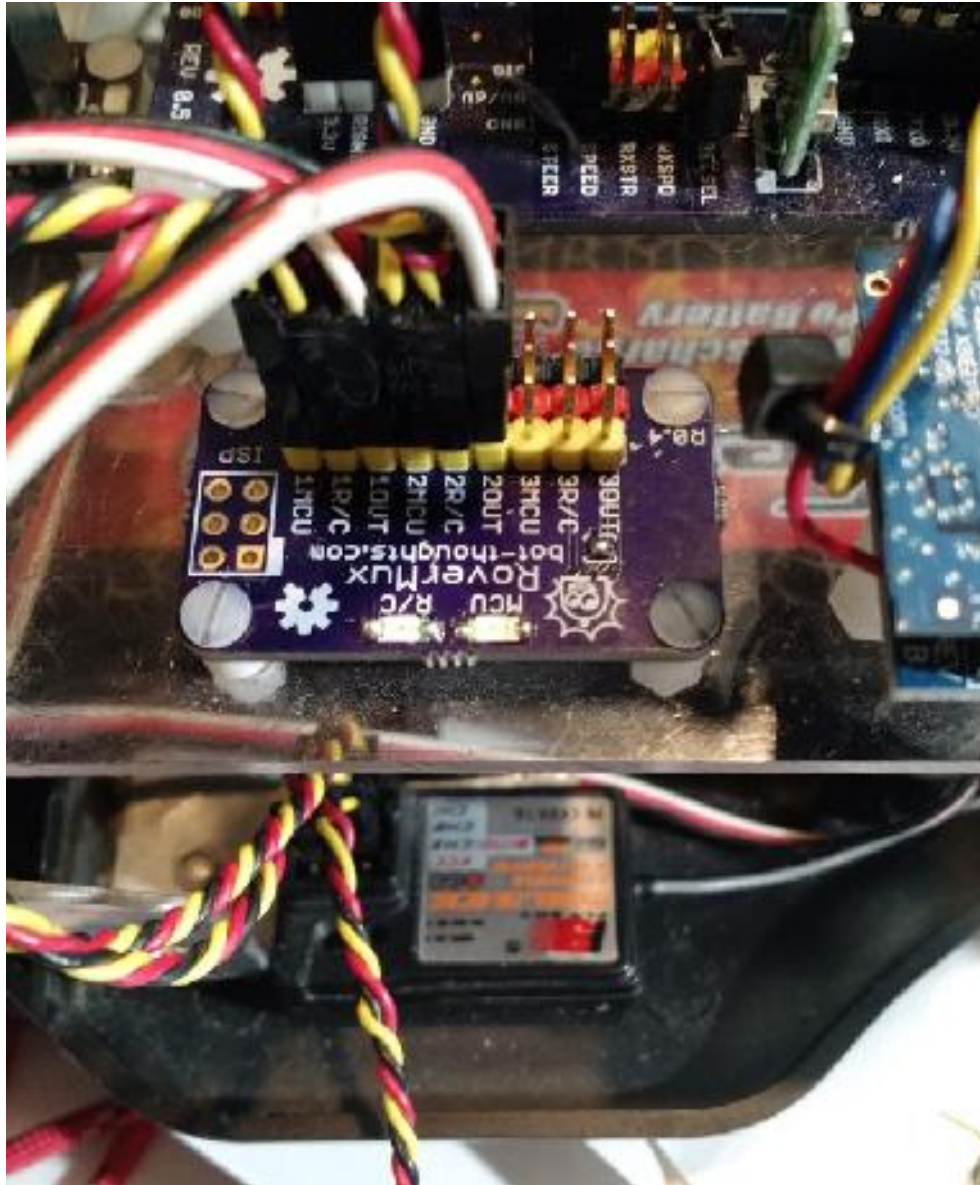
See the pictures below as an example of adjusting these settings on a FlySky FS-GT3B.



4 Installation

4.1 Securing RoverMux On Your Rover

Install the RoverMux onto your rover's electronics platform (e.g., polycarbonate, acrylic, plywood) using 2mm or #2 machine screws, nuts, and standoffs. I prefer polycarbonate; it's lightest, strongest, easiest to drill nice holes, and it looks clean and cool.



4.2 Plugging In The Cables

The following tables shows you how to connect the 3-pin servo cables to RoverMux, your Rover, and your microcontroller (MCU) aka autopilot hardware.

RoverMux Pin	Connect To...
1R/C	Receiver CH1
2R/C	Receiver CH2
3R/C	Receiver CH3 (<i>Optional</i>)
1MCU	Microcontroller STEERING control
2MCU	Microcontroller THROTTLE control
3MCU	<i>Optional</i>
1OUT	Rover STEERING Servo
2OUT	Rover ESC or THROTTLE Servo
3OUT	<i>Optional</i>



5 Hacking and Reflashing

If you want to hack the firmware, you probably already know something about AVR programming. This isn't intended to be an exhaustive guide to doing so but only aims to provide the general information needed, assuming you have done some AVR programming already.

5.1 The MCU

The RoverMux either uses an ATtiny13, 13A or 25. You'll have to check the label on the 8-pin MCU on the bottom of the board to be sure.

5.2 Setting Up

- First, install the AVR ISP header onto RoverMux (see Section 2.3).
- The source code is located in the *firmware/* directory.
- Clone the git repository from <https://github.com/shimniok/RoverMux>

5.3 Programmer

You can use a standard AVR programmer to flash the MCU, such as:



Pololu AVR
Programmer
recommended



AVR Dragon
recommended



AVRISP MkII



JTAG ICE

5.4 Windows

- Install the AVR Toolchain.
- Use AVR Studio to edit the files, compile, and program the MCU.
- Use Atmel's documentation and online resource for more information.

5.5 Linux

- On Linux, install the Atmel AVR gcc toolchain for Linux.
- Use any text editor to edit source such as Gedit, Geany, GNU Emacs, vim, etc.
- Compile using make
- Edit the Makefile and configure it to use the correct *avrdude* settings for your programmer
- After connecting the programmer to the RoverMux AVR ISP header, use *make install* to flash the chip with the code you compiled.

5.6 Mac

- Download the [CrossPack for AVR Development](#), which is packaged as a *.dmg* file.
- Open the *.dmg* file, and double-click on *CrossPack-AVR.pkg* to install the AVR toolchain.
- Ensure you have *make* installed (you may need to install Developer Tools or install make using Homebrew, etc).
- Use vi, GNU Emacs, Eclipse, XCode, etc., to edit the source code.
- Compile using make from the command line while in the RoverMux firmware directory
- Edit the Makefile and configure it to use the correct *avrdude* settings for your programmer
- After connecting the programmer to the RoverMux AVR ISP header, use *make install* to flash the chip with the code you compiled.