FFX8RS-NN | **QUICK SETUP GUIDE**

RS-232 Switch

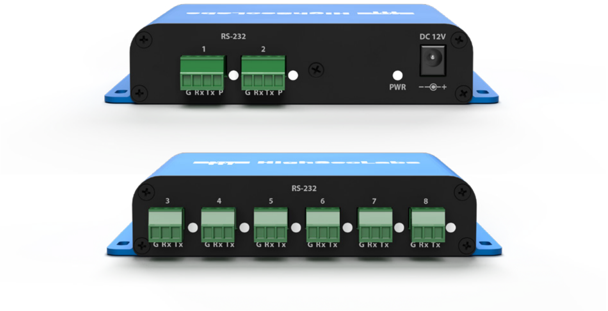
TABLE OF CONTENTS

INTRODUCTION

RS-232 Switch

The RS-232 Switch allows control of multiple RS-232 devices from a single or dual control point, acting as a gateway for all the connected devices.

This guide provides instructions on how to install and operate an RS-232 Switch.



INSTALLATION

Before Installation

Ensure all devices are powered OFF before installing.

Connect the Controlling Devices

- Connect controlling devices to the RS-232 Switch via the 2 4-pin serial blocks labeled #1 and #2.

Note: While Ports #1 and #2 are the only ports that provide power, the Switch can send commands from any of the RS-232 ports.

Connect the Receiving Devices

- Connect up to 6 receiving devices to the Switch via the 6 3-pin serial blocks labeled #3-#8.

Power ON the RS-232 Switch

- Power ON the RS-232 devices.
- Power ON the RS-232 Switch by connecting a power supply to the 12V DC port.

Operation

The RS-232 Switch receives commands from the connected controller(s), then sends them to the selected receiving devices via a dedicated cable for each device. These commands can be sent from any RS-232 compatible device, such as an HSL Remote Control Unit, and can be sent to one, multiple, or all the connected devices (including the sending device itself).

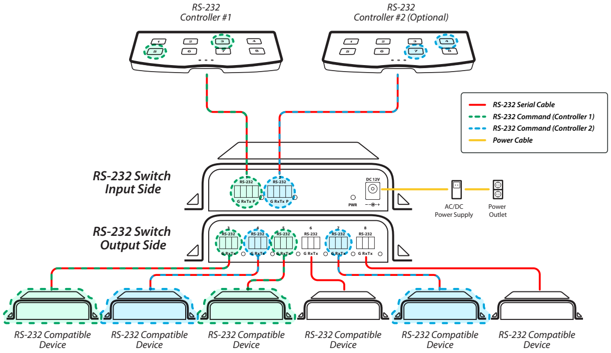


***Figure 1**: Controller 1 sends a command to devices connected to Ports 3 and 5; Controller 2 sends a command to devices connected to Ports 4 and 7.*

Chaining Multiple Switches

The RS-232 Switch can also chain to other Switches, then send commands to devices connected to the chained Switches.

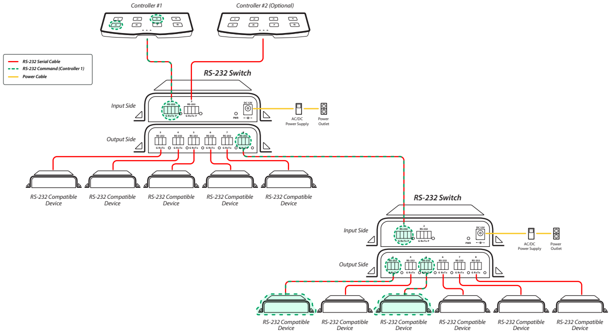


***Figure 2**: Controller 1 sends a command to devices on Ports 4 and 5 of a Switch chained to Port 8.*

A: RS-232 Command Structure:

The RS-232 Switch can send commands from one device to any or all other connected devices, including the sending device itself. This requires a specific command structure to address and route commands properly. The command structure is as follows:

$1<Header><Message data>$D$2

- **Start <$1>:** Marks the beginning of the command
- **Header:** Defines the command's attributes and destination channels
- **Message Data:** The command itself
- **Carriage Return <$D>:** Marks the end of the Message Data
- **Tail <$2>:** Marks the end of the entire command

Byte Stuffing:

The RS-232 interface uses **<$**> as the control character to begin and end commands. When creating an RS-232 command, the header cannot contain **<$>**, and any instances of **<$>** in the message data must be written instead as **<$3>**. For example, a command with the characters **<C2$$d>** must be written as **<C2$3$3d>**.

Header Structure:

Message Attributes:

The command can be sent directly to a connected device or retransmitted through another connected RS-232 Switch.

- **<space><n>:** Transmit the command directly to a connected device. **<n>**= the destination channel(s).
- **R<n><space>:** Retransmit the command to another RS-232 Switch. **<n>**= the destination channel(s) of the chained Switch(es).

Note: Commands can be retransmitted through multiple levels of Switches that are chained together, but a command can only be sent to devices that are on the same level of chaining. For example, a command cannot be sent to a device directly connected to the Switch and also retransmitted through a chained Switch to a different device.

Destination Channels:

The destination channel(s) are represented as bits in an 8-bit byte, then entered in hexadecimal.

(See Appendix B for complete instructions on finding the values for destination channels)

- Example: Channels 3 and 5 = hex 14.

Header/Tail Stripping

When a command is sent into the RS-232 Switch, it strips the header and tail before sending it to the destination device. If multiple Switches are chained together, each Switch only strips the first header in the command.

Examples:

- **Direct Connection:** To send the command **GO** to devices directly connected to Channels 3 and 5:
  $1 14GO$D$2
  - The Switch strips the header **<$1 14>** and tail **<$2>** from the command before sending it to the connected devices, so the devices receive the message **GO.**
- **Chained Switch:** To retransmit the command **GO** to devices connected to Channels 3 and 5 of a chained Switch (Switch 2), which is connected to Channel 4 of the transmitting Switch (Switch 1):
  $1R08 14GO$D$2
  - Switch 1 strips the header **<R08>** from the command before sending it to Switch 2, so Switch 2 receives the command **$1 14GO$D$2.**
  - Switch 2 strips the header **<$1 14>** and the tail **<$2>** from the command before sending it to the connected devices, so the devices receive the message **GO.**
- **Multiple Levels of Chaining:** To retransmit the command **GO** to devices connected to Channels 3 and 5 of a chained Switch (Switch 3), which is connected to Channel 4 of a chained Switch (Switch 2), which is connected to Channel 1 of the transmitting Switch (Switch 1):
  $1R01R08 14GO$D$2
  - Switch 1 strips the header **<R01>** from the command before sending it to Switch 2, so Switch 2 receives the command **$R08 14GO$D$2.**
  - Switch 2 strips the header **<R08>** from the command before sending it to Switch 3, so Switch 3 receives the command **$1 14GO$D$2.**
  - Switch 3 strips the header **<$1 14>** and the tail **<$2>** from the command before sending it to the connected devices, so the devices receive the message **GO.**

B: Determining Destination Channel Values

When creating the header for an RS-232 command, the destination channel is expressed in a 2-digit hexadecimal value.

To find this value, read the channels from right to left, with Channel 1 being the bit furthest to the right and Channel 8 being the bit furthest to the left, expressing them as either **1** for an active channel or **0** for an inactive channel. All 8 channels are represented together as an 8-bit byte.

Example:

For a command being sent to Channels 3 and 5, the 3rd and 5th bit in the byte would be active (**1**), and all other bits would be inactive (**0**). Therefore, expressing the values from right to left, the 8-bit byte would be **00010100**, expressed in hexadecimal as **14**. Following the syntax described in Appendix A, the complete structure to send the command **GO** to Channels 3 and 5 would be:

$1 14GO$D$2

Highseclabs.com

For more information about HSL's solutions, please contact:

HighSecLabs Inc.
905 James Record Road STE A,
Huntsville AL, 35824

HSL Support
256-203-3036
https://highseclabs.com/contact/

Sales
Sales@highseclabs.com