

# 20-Port Serial Switch



## User's Guide

## Product Features

- 20 ports of relay isolated serial interfaces.
- On-board  $\mu$ P with EEPROM.
- Manual or batch file operation via simple text menus.
- Ability to broadcast through multiple ports simultaneously.
- Support for any baud rate or data format after connection.

## Device Requirements

- Any three-wire RS-232 source and target may be connected.
- Baud rates of 9600, 19,200, 38400, and 115,200 are supported for communications with the menu system. Any baud rate (ie. 110 baud to 2Mbaud) may be used to communicate with attached devices.

## Crosspoint Switch Capabilities and Limitations

A crosspoint data switch provides a link from one host device (such as a PC or modem) to several attached devices (such as routers, radios, or sensors). Use this switch to communicate serial management ports without the use of a PC, or to control several serial devices from a single port. The switch provides a cost-effective alternative to multi-port serial cards when you only need to communicate with one device at a time.

Although this switch can –send- data to multiple devices simultaneously, it can't –receive- from more than one device at a time. There is no buffering or memory in a crosspoint switch. Characters transmitted when a port is turned off are ignored. Only data on pins 2 & 3 of the DB-9 is switched. Handshaking isn't switched.

## Hardware Setup

A 5V wall adapter provides regulated DC power to the switch. The maximum current drain of 600mA occurs when all ports are switched on. Normal power consumption is under 1W.

Two types of ports are provided, console and device. The control (console) port can connect with a PC via a straight-through cable, or to a modem via a crossover cable. The control port pinout is:

Pin 2 = Receive Data (from devices and from the switch to the control port)

Pin 3 = Send Data (to switch and devices from the control port)

Pin 5 = Ground

Handshaking lines are strapped for no flow control (always on).

Pin 7 RTS & 8 CTS are strapped together.

Device ports connect to a managed device via a straight-through cable or to a PC via a crossover cable. The 20 device ports are labeled A-T. The device port pinout is:

Pin 2 = Transmit Data (from devices)

Pin 3 = Receive Data (from devices)

Pin 5 = Ground

Handshaking lines are strapped for no flow control (always on).

Pin 7 RTS & 8 CTS are strapped together.

Connect a PC or terminal to the control port at 9600, N,8,1 when first connecting to the switch. If you are using Windows Hyperterminal, set flow control to “none”.

## Menu Operation

Press ~ (tilde), wait at least ½ a second, and press ~ again to enter the menu. Once the menu is entered, communication with attached devices is disabled. The following message appears:

```
Serial Switch Unit Name: Serial Switch
Enter Command (? for help):
```

Valid commands are:

**A-T Toggle the specified port**

This command enables (connect) or disables (disconnect) a single device port 1-20 (1=A, 2=B...20=T). Port switching is performed after exiting the menu with the “X” or “ESC”.

**Z Enable all 20 ports for broadcast**

This enables simultaneous transmission from the control port to all attached devices. This is useful when configuring a number of identical devices. Port switching is performed after exiting the menu with the “X” or “ESC”.

**W Disable all ports.**

Any enabled ports will be disabled after exiting the menu with the “X” or “ESC”.

**U Set baud rate.**

Note that the baud rate of the switch may be different than the baud rates of the attached devices. This baud rate is used only by the menus. You may transmit at any other baud rate or send non standard bit streams to the attached devices after switching. Baud rates of 9600, 19,200, 38400, and 115,200 are supported. All baud rates operate at 8 bits, no parity and no handshaking. Be sure to write down the baud rate you select. If you forget the setting, you’ll need to try all four baud rates.

**Y Change Unit or Port Name.**

It can be useful to label ports and label the switch. For example, Port 1 might be “Cisco Router”. These settings are limited to 16 characters and are saved immediately in non-volatile EEPROM.

- V**                    **Show port settings**  
This command displays the settings (on or off) of each port, along with the unit names.
- X or Esc**        **Escape**  
Either an ASCII “X” 0x58 or ESC 0x1B may be used to exit the menu at which time the crosspoint relays are enabled.

## Batch File Operation

Manual and batch file control function and syntax is identical. There are three important considerations when using batch files:

1. The switch doesn’t react instantly to commands. It takes approximately 1/20<sup>th</sup> of a second for the crosspoint switch to complete a switching command. To compensate for this, either use a delay command or send a few null characters. Nulls are normally ignored by RS-232 devices. At 9600 baud, 40 nulls are required. Be sure to insert a delay between the ~ attention characters to enable the switch. You can disable this delay to simplify your batch file and speed up switching. Do this only if you’re certain these two characters will not be inadvertently sent in the data stream.
2. The exact syntax of a command is important. For example, “Echo ~>COM1” and “ECHO ~ >COM1” are different. Windows will send the space character 0x20 after the ~ in the second example. From within a program, it’s sometimes desirable to use write(port, str, strlen(str)) instead of fprintf() which will buffer the output string.
3. Again, only one port can “listen” at a time, but feel free to transmit to as many as you like.

## Communication from Within Programs

The simplest method of setting up a port is to use the mode command in a system() function call:

```
system( "mode com1: baud=9600 parity=n data=8 stop=1" )
```

Type mode /? in a command line console for full details. Once a port is set up, a com port can be opened as a stream I/O device using stdio:

```
char menuresponse[128] ;
FILE* portfp = fopen( "com1:", "w+" ) ;
fprintf( portfp, "~" ) ;          /*send tilde */
Delay (500);                     /* wait 500ms */
fprintf( portfp, "~" ) ;          /*send second tilde */
fgets( menuresponse, sizeof( response ), portfp ) ; /* you're
in to the menu, send commands next */
printf( menuresponse ) ;
```

You can also use C++ fstream I/O. The Win32 API has comprehensive low level functions for

serial I/O, but the simpler stdio function works for many applications. For information on W32 serial communications, see:

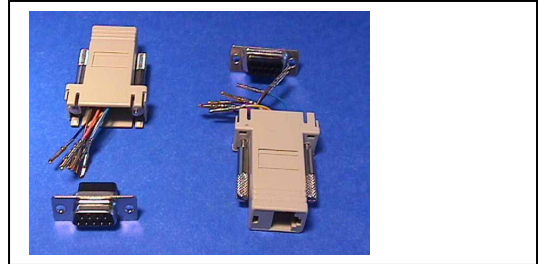
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnfiles/html/msdn\\_serial.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnfiles/html/msdn_serial.asp)

For information on Linux serial communications, see:

<http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html>

## Accessories

DLI stocks RJ-45 to DB-9 adapters, as well as cable. These adapters allow you to insert pins in any order, so you can create crossover or straight-through cables easily. Flat phone-type satin cord makes a clean installation, particularly with Cisco equipment. Call (408) 330-5599 for connectors or cables.



## Support

Please visit [www.digital-loggers.com](http://www.digital-loggers.com) for manuals and accessories. If we haven't answered your questions here, please call (408) 330-5599 or send an email to [support@digital-loggers.com](mailto:support@digital-loggers.com). We'll be glad to help.

