

# Instructions for the AIDE9 DE-9 RS232 Tester/Null-Modem Switch

## \*\*\* WARNING \*\*\*

This device for use by trained personnel only. Misuse may cause injury or death.

**Do not touch the device while it is connected to any circuit.**

Do not touch the components on the device after use; they may be hot.

Do not lick or eat the device.

Not for use in life-support systems.

## \*\*\* WARNING \*\*\*

All pin numbers refer to the top connector. Normally the female connector.

The switch selects the mode for pins 2 & 3 (RXD & TXD):

Both "A" sides down = straight through.

Both "B" sides down = null-modem

Installing jumpers:

Jumpers installed across = straight through.

Jumpers vertical = null-modem

Installing a jumper across the 2 or 3 headers with the switch set to null-modem will short pins 2 & 3 on both connectors, making a quick loopback.

Jumper Layout:

|            |   |     |        |
|------------|---|-----|--------|
| Top ( )    | 2 | ( ) | Bottom |
| Bottom ( ) | 3 | ( ) | Top    |
| Top ( )    | 4 | ( ) | Bottom |
| Bottom ( ) | 6 | ( ) | Top    |
| Top ( )    | 7 | ( ) | Bottom |
| Bottom ( ) | 8 | ( ) | Top    |
| Top ( )    | 1 | ( ) | Bottom |
| Bottom ( ) | 9 | ( ) | Top    |

Normally there should be no jumpers on 2 and 3, the switch quickly takes care of selecting straight through or null-modem for pins 2 and 3.

For straight-thru use, install jumpers across to connect matching pins on top and bottom connectors.

For most null-modem applications install vertical jumpers between 4&6 and 7&8 to cross those pins between the connectors.

Pins 1 and 9 are RI and DCD, often not used, and since both originate from the "modem" side, I would expect them to usually be straight-thru. (Maybe.

If everyone always used the same pins in the same way, you wouldn't even need this tester, now would you?)

Of course, you can also use twists of wrap wire or flying jumpers to connect the pins any way you want.

(over)

Other switch settings:

1A and 2B down = loopback (Pins 2,3) on bottom connector + pin 2 on top

1B and 2A down = loopback (Pins 2,3) on bottom connector + pin 3 on top

LEDs:

Red indicates a negative voltage, Green a positive.

The negative is normally the idle state, so a solid red LED on RXD or TXD means no data. Flashes of green means bursts of data, yellow means constant data, solid green means a "break" condition.

The LEDs will start to light at 3 volts, which is the absolute minimum RS-232 receiver specification.

Maximum safe continuous voltage on any pin is 18 volts.

RS-232 specs do allow swings as high as +/- 25V, a few seconds exposure to 25V should not damage the tester, but longer periods **will** cause damage.

Each LED will load the corresponding signal line at 1mA for a 3V signal, 16mA for a 18V signal.

Obviously this tester is primarily designed to indicate whether the pins are properly connected. It will not troubleshoot errors in data rates or formats.

With normal signal levels of +/-5V to +/-12V, it should be possible to see a LED flash for a single ASCII character at 4800 baud and below. At 9600 a single character will be at best a very faint flash. Continuous data will show yellow at any data rate.

Typical RS-232 Pinout:

| Pin | Function                |
|-----|-------------------------|
| 1   | DCD Data carrier Detect |
| 2   | RXD Data In             |
| 3   | TXD Data Out            |
| 4   | DTR Data Terminal Ready |
| 5   | GND                     |
| 6   | DSR Data Set Ready      |
| 7   | RTS Ready To Send       |
| 8   | CTS Clear to Send       |
| 9   | RI Ring Indicator       |

FAQs:

Why the 18V signal limit?

The design is a tradeoff between having the LEDs light at 3V, having some visible indication of signal level, and cost.

Why does it not have a pulse-stretcher to make short fast data visible?

A pulse-stretcher would either require a battery or at least a charge collector, and a lot more complexity. We decided to keep this tester simple and affordable.

Why does it show pin numbers but not functions?

We positively know which pin is which, but we can't guarantee which function is on which pin.