Serial Interface Debugging

Overview:

The RS-232 interface offers a great deal of flexibility in terms of supporting a wide range of data formats and protocols. Other more modern, standardized interfaces often include a well defined basic protocol, and sometimes, a degree of “auto-negotiation” built-in. It is RS-232’s inherent flexibility that is the root source of most serial interface problems. Debugging an RS-232 serial interface requires specialized tools and techniques. In the worst case, an oscilloscope may be required to inspect the interface signals directly, evaluating noise, signal-levels, timing and data corruption. However, it is more commonplace to have problems in the interface setup parameters or the message protocol itself. Both of these can be diagnosed with a common PC (ex: laptop), appropriate software, and some inexpensive accessories.

Diagnosing RS-232 serial links with a PC will require these items:

1. PC (or laptop), running Windows (XP or higher), preferably with 2 serial (COM:) ports. An external USB/Serial adapter (or two) are acceptable.
3. Serial gender-adapters (F/F, M/M). Sometimes these are required.
4. Serial-tap cable assembly (usually hand-built).
5. If you suspect there is a signal-corruption or other low-level hardware problem, you may need an oscilloscope or multimeter as well.

Note: If necessary, the “Hyperterm” software provided with some versions of Windows can be used, though it can be difficult or awkward to use for a duplex “serial monitoring” application. Using “REALTERM” is highly recommended.
Building a serial-tap cable assembly:

The serial-tap cable assembly is easily constructed using commonly available components and hookup-wire. You will need the following items:

1. Serial DB9-Male connector, with crimp or solder-terminals. QTY = 1.
2. Serial DB9-Female connector, with crimp or solder-terminals. QTY = 3.
3. 20 to 28 AWG insulated hookup wire, and preferably stranded (not solid) so that it’s flexible and not easily prone to breakage. Any color is acceptable.

Note: You may be able to construct this serial-tap cable using cut-off pieces of “serial extension cables”, or any other DB9 M/F connectors you can scavenge from your junk-box.

Serial-tap cable equivalent schematic:

Note: The PC serial port assignments (COM1: and COM2:) may be different on your specific PC or laptop.

Note: Pin-3 on the PC serial port connectors is not used.

Serial-tap cable wiring:

<table>
<thead>
<tr>
<th>P1 (DB9-F) to DTE device</th>
<th>P2 (DB9-M) to DCE device</th>
<th>P3 (DB9-F) to PC-COM1</th>
<th>P4 (DB9-F) to PC-COM2</th>
<th>Signal name &amp; direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>DTE ↔ DCE data stream</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>DTE → DCE data stream</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>Signal ground</td>
</tr>
</tbody>
</table>
Connecting the serial-tap cable:

After building the serial-tap cable (above), the connectors P1 and P2 are used to “tap into” the serial link you want to monitor.

For example: To diagnose a serial connection between a Mondo switch and an external control system:
1. Disconnect the serial control cable from the Mondo switch.
2. Connect the serial-tap cable P1 to the free end of the cable you just disconnected.
3. Connect the serial-tap cable P2 to the serial control port on the Mondo switch.

Typical cabling before inserting the TAP cable:

Typical cabling after inserting the TAP cable:

In this example, the serial-tap cable is inserted at the Mondo-end of the RS-232 serial link. The PC (or laptop) has two com ports numbered COM1 and COM2. Your setup or requirements may vary. As shown, P3 and P4 connect to the two COM ports on your PC (which is running the RealTerm software). It makes no difference which ports P3/P4 are connected to, and neither does P1/P2 (just connect them so the DB9 genders mate up properly).
Installing the REALTERM software:

After downloading the REALTERM software package, follow the REALTERM website’s directions for installation. There may be no downloadable (PDF) manual, however, the project website has complete help instructions available.

It is often useful to “print to PDF” while viewing the website in a browser. You may need to download a PDF “print-driver” to do this. Here is one possible PDF “print-driver” for Windows: [http://sourceforge.net/projects/pdfcreator/](http://sourceforge.net/projects/pdfcreator/).

Debugging a serial link:

After REALTERM is installed, use the following as a guide to using it for serial-monitoring a full-duplex serial link:

Start the RealTerm software.

On the PORT tab:
- Select the correct baud-rate for your application (9600 is the default for Mondo-III control ports, but your needs may differ).
- Select the COM port you’ve connected to P3. (COM1 in the above Mondo example)
- Click “CHANGE”.

On the ECHO tab:
- Select the same baud-rate as you’ve chosen above.
- Select the COM port you’ve connected to P4. (COM2 in the above Mondo example)
- Enable the “MONITOR” checkbox.
- Click “CHANGE”.

The terminal screen should now show serial traffic in both directions, and in different colors. On the DISPLAY tab it may be useful to play with the ASCII, ANSI or other data display formats. One of them will be more suitable to the kind of traffic you want to monitor. As you interact with the device (Mondo-III chassis in this example), you will see both sides of the conversation in the terminal window.

Note that REALTERM is also a very handy terminal-emulator. Simply use a normal serial cable (not the serial-tap cable) and you’ll be able to interact with the device (Mondo-III chassis in this example) directly using keyboard input.
Glossary of RS-232 Terminology:

**DTE** = Data Terminal Equipment. This is usually a PC or controlling device. A typical DTE device will have a male connector (either DB9-M or DB25-M).

**DCE** = Data Communications Equipment. This is usually a modem or controlled device. A typical DCE device will have a female connector (either DB9-F or DB25-F).

**BAUD** = Speed setting for serial data transfer. Also referred to as “BAUD rate”. Some common baud rates include: 300, 600, 1200, 2400, 4800, 9600, 14.4K, 19.2K, 38.4K, 57.6K, 115.2K and 230.4K. Not all equipment (DTE’s or DCE’s) can support every possible baud-rate setting.

**Start bit** = Each character begins with a single “0” bit. This signals the beginning of a character transmission. This parameter cannot be changed—it is always set to “1”.

**Data bits** = Each character transmitted over a serial link is composed of a number of individual bits. Usually 7 or 8 bits. However, there are some (ancient!) devices that use as few as 5 bits. These very uncommon now. Both ends of the link should have the same data-bits setting.

**Parity bit** = After the data bits, some serial protocols use an extra “check bit”, which is the computed sum of the preceding data bits. The parity setting can be either **even** (the sum of the data bits plus the parity bit = 0), or **odd** (the sum of the data bits plus the parity bit = 1). In most systems parity checking isn’t used, so the parity setting should be **none**. Both ends of the link should have the same parity setting.

**Stop bits** = There is a fixed minimum idle time between the end of one character and the start of the next. The minimum time is specified as “stop bits”. The setting is either 1 or 2. If a device can’t keep up with incoming data, it’s sometimes helpful to switch to 2 stop-bits at the transmitting end.

**Character Pacing** = Similar to “stop bits”, a “character pacing” setting is sometimes available at the transmit end to add additional whole-character periods of idle time between each character.

**Line Pacing** = Similar to “stop bits”, a “character pacing” setting is sometimes available at the transmit end to add additional whole-character periods of idle time between each line of characters. One line of character is usually terminated with a carriage-return (<CR>) character as a message delimiter. This can vary from system to system, and may not apply to interface protocols which do not use a specific message-delimiter character.

**Straight through cable or Extension cable** = Generally used for connecting a DTE and DCE device together, or for simply extending (without changing signal connections) any serial port or existing cable. This cable always has one male and one female connector.

**Null-modem cable** = Generally used for connecting two DTE, or two DCE, devices devices together.

**Gender changer** = A very small back-to-back connector adapter, used to connect two male or two female connectors together. In some cases, a gender-changer can also function as a null-modem—is is marked accordingly.
RS-232 Signal Usage:

RS-232 signal states:
- “SPACE” = +3V to +12V
- “MARK” = -3V to –12V
- A signal voltage between –3V and +3V is “undefined”.

A note on RS-232 cable lengths:
- Using high quality and shielded cable, an RS-232 link can span 50ft or much more.
- The baud-rate has a significant effect on maximum cable length. For example, a 9600 baud link can be successfully transmitted much farther than a 230.4K link over the same type of cable.
- In cases where a very long RS-232 cable absolutely must be used due to installation site requirements, reducing the baud rate can often reduce data-errors and other communication issues.

Carrier Detect (CD):
- Used by modems to indicate a data-carrier signal was detected and recognized on the analog line (phone line).
- For non-modem uses, this is merely another H/W handshaking signal which generally means “DCE device is ready to operate”.
- MARK = not ready. SPACE = ready (or carrier-is-detected).

Receive Data (RXD):
- This is the serial datastream sent from the DCE device to the DTE device.
- When no data is being received (idle line), this signal is in the MARK state.

Transmit Data (TXD):
- This is the serial datastream sent from the DTE device to the DCE device.
- When no data is being transmitted (idle line), this signal is in the MARK state.

Data Terminal Ready (DTR):
- A signal from DTE to DCE, to indicate it is physically connected and ready to operate.
- MARK = not ready. SPACE = ready.

Data Set Ready (DSR):
- A signal from DCE to DTE, to indicate it is physically connected and ready to operate.
- MARK = not ready. SPACE = ready.

Request to Send (RTS):
- A signal from DTE to DCE, to indicate the DTE wants to send data.
- MARK = not requesting to send. SPACE = requesting to send.

Clear to Send (CTS):
- A signal from DCE to DTE, to indicate it is ready to receive, generally in response to RTS.
- MARK = not ready to receive. SPACE = ready to receive.

Ring Indicator (RI):
- A signal from DCE to DTE, to indicate that the analog phone line is “ringing” and should be answered.
- This signal is usually provided by “modem” type DCE devices.
- MARK = phone line not ringing. SPACE = phone line ringing (generally pulses with same cadence as ringing signal).
Standard RS-232 Interface Cable:

This cable would be the typical “all wires” connection between a DTE and DCE device. Note that the interface can be either a 9-pin or 25-pin at each end. In some cases, you will find a mixed system with a 9-pin at one end and a 25-pin at the other.

<table>
<thead>
<tr>
<th>DB9-M (DTE) Pinout</th>
<th>DB25-M (DTE) Pinout</th>
<th>Signal name and direction</th>
<th>DB25-F (DCE) pinout</th>
<th>DB9-F (DCE) pinout</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>Carrier Detect</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Receive Data</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Transmit Data</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>Data Terminal Ready</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>Signal ground</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Data Set Ready</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>Request to Send</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>Clear to Send</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>Ring Indicator</td>
<td>22</td>
<td>9</td>
</tr>
</tbody>
</table>

Special serial cable connections:

DTE/DCE with H/W Handshake Eliminator:

Some DTE’s (computers for example) and DCE’s (devices, such as displays, touch-screens or printers) can be easily interconnected with a straight-through 3-wire (TXD/RXD/Gnd) cable, but one end or the other demands that a control signal be in a particular state to enable communications. It is impossible to detail every possible variant in this document, but here is one example of a “handshake eliminator” cable:

<table>
<thead>
<tr>
<th>DB9-M (DTE) Pinout</th>
<th>DB25-M (DTE) Pinout</th>
<th>Signal name and direction</th>
<th>DB25-F (DCE) pinout</th>
<th>DB9-F (DCE) pinout</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>n/c</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Receive Data</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Transmit Data</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>DSR/DTR loopback</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>Signal ground</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>n/c</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>RTS/CTS loopback</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td></td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>n/c</td>
<td>22</td>
<td>9</td>
</tr>
</tbody>
</table>
Null-Modem (Full):

This connection is generally used when two similar devices need to be connected to each other. For example, two "DTE" devices (computers), or two "DCE" type devices. The pinout of a DTE-to-DTE (female-to-female) null-modem cable is usually:

* Note: Two wires on one pin.

Null-Modem (Quick):

Often all the signals aren’t required for a full hardware handshaking and merely swapping TXD and RXD will suffice. The pinout of a DTE-to-DTE (female-to-female) null-modem (quick) cable is usually:

<table>
<thead>
<tr>
<th>DB9-F (DTE) Pinout</th>
<th>DB25-F (DTE) Pinout</th>
<th>Signal name and direction</th>
<th>DB25-F (DTE) pinout</th>
<th>DB9-F (DTE) pinout</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 8 4 20* 3 7 6</td>
<td>8 3 2 20* 4 5 9</td>
<td>Carrier Detect ← Data Terminal Ready</td>
<td>4 3 2 1</td>
<td></td>
</tr>
<tr>
<td>2 3 2 3 4</td>
<td>8 1 2 7 6</td>
<td>Receive Data ← Transmit Data</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>3 2 3</td>
<td>20 8 7 6</td>
<td>Transmit Data → Receive Data</td>
<td>2 3 4</td>
<td></td>
</tr>
<tr>
<td>4 20 8 1</td>
<td>8 7 6 5</td>
<td>Data Terminal Ready → Carrier Detect</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>5 7 7 5</td>
<td></td>
<td>Signal ground</td>
<td>2 3 4</td>
<td></td>
</tr>
<tr>
<td>6 6 20* 4</td>
<td></td>
<td>Data Set Ready ← Data Terminal Ready</td>
<td>4 3 2</td>
<td></td>
</tr>
<tr>
<td>7 4 5</td>
<td>5 8 7 6</td>
<td>Request to Send → Clear to Send</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>8 5 4</td>
<td>4 7 6 5</td>
<td>Clear to Send ← Request to Send</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>9 22 9</td>
<td>22 9 8 7</td>
<td>N/C</td>
<td>1 2 3</td>
<td></td>
</tr>
</tbody>
</table>

© 2009-2011 All Rights Reserved - Magenta Research, Ltd. *** All information subject to change without prior notice.