Global Knowledge™

Expert Reference Series of White Papers

RS-232: A Key To Understanding Data Communications

RS-232: A Key To Understanding Data Communications

George Mays, CISSP, CCNA, A+, Network+, Security+, INet+

Introduction

In all disciplines, a few solutions seem to stand the test of time – things that were important decades ago are important today and have value in the future as new things come along that are built upon past successes. RS-232 data communications falls into this category.

That "COM" port on the back of your PC is, in fact, an RS-232 interface. These days, it is commonly used in the initial configuration of network devices like switches and routers. It may also be used to interface with cash drawers, bar code scanners, receipt printers, lab equipment, and myriad other devices. In the not-too-distant past your COM port may have been used for an external modem, a mouse, or even a printer.

Many readers work with network devices like access points, managed hubs, switches, and routers. The majority of these have a "console" port, which is, again, an RS-232 port. This port is used for configuration of the device and, sometimes, for troubleshooting after the equipment is installed on the network.
RS-232 is a popular solution even now, more than forty years after its birth.

History

RS-232 dates back to the early 1960s. The Electronics Industries Alliance (EIA) originated the standard and continues to oversee it through their companion organization, the Telecommunications Industry Association (TIA).

The "RS" in the name of the standard is widely considered to mean Recommended Standard. However, you will very likely see RS-232 referred to as EIA/TIA-232, or simply TIA-232, in recent times.

I first encountered an RS-232 connection in 1970 – one year after the extremely popular RS-232C standard was published. The interface was used to connect a Teletype (yes, a real Teletype) to a Bell 103 modem. The speed was blinding – 110 bits per second – that's right, 10 characters per second. Throughout the 70s, RS-232 ruled the world of data communications between mainframes, minicomputers, and remote terminals. It was widely used to connect asynchronous "dumb" terminals to computers. But you also found it in synchronous applications employing high-speed modems; for example, a 9600 baud link that might exist between the computing facilities at a university's main campus and its medical school. RS-232 serial ports were also a staple of the earliest personal computers to emerge in the market.

In the 80s, with the amazing success of the IBM PC and similar products that followed, RS-232 found new applications. The "COM" port, as we have come to call it, interfaced with everything from printers, to modems, to mice.

The Internet blossomed in our collective consciousness in the 90s, and there was our tried-and-true solution, RS-232, to interface with our newer, higher-speed modems and to allow us to configure the hubs, switches, and routers that proliferated. In the new century, we often find these interfaces on our wireless access points as well.

You don't think that those home routers and wireless access points have RS-232 interfaces? Guess again. On many of the internal circuit boards there are headers, or attachment points, for RS-232 devices – presumably vestigial of the engineering, development, and testing processes.

It seems that whenever an inexpensive, well-understood communications interface is required, especially where speed is not terribly important, RS-232 remains a very common solution.

Typical Application

Figure 1.

The RS-232 standard refers to two classes of devices: Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE). Classically, the DTE component is a computer or a terminal. These days, a router is often the DTE piece of the puzzle. In the past, the DCE component was usually a modem connected to the public, switched telephone network, but now you are likely to find a CSU/DSU instead that connects to a digital leased line (like a T1, for instance) or maybe a Terminal Adapter (TA, for ISDN). See Figure 1 for a simple example.

RS-232 is often used to directly connect two DTE devices. Consider the connection between the COM port on a laptop (DTE) and the console port on a router (also DTE). Of course, this gives rise to some special cabling and/or configuration concerns. We naturally want the transmitted signal from one device to appear on the receive side of the other device. See Figure 2.

Figure 2.

Signals

There are basically two kinds of RS-232 signals that appear on the cable: Control Signals and Data Signals.

Signal	Direction	DB25	DB9	Descripiton
RTS	DTE --> DCR	4	7	Request To Send
CTS	DCE -->DTE	5	8	Clear To Send
DSR	DCE -->DTE	6	6	Data Set Ready
DCD	DCE --> DTE	8	1	Data Carrier Detect
DTR	DTE --> DCE	20	4	Data Terminal Ready
RI	DCE --> DTE	22	9	Ring Indicator

Figure 3. Commonly Used Control Signals

Controls signals, like Control

Signals, like Request To Send (RTS) and Data Set Ready (DSR), are used to inform the device on the other end of the wire of a change in the state of the device or of a requested action. DSR, classically, means that the modem is ready to communicate with the data terminal equipment. Figure 3 summarizes the control signals.

If you were to observe these control signals with an oscilloscope or a DVM, you would see that the signal manifests as a positive voltage, with respect to Signal Ground, to indicate a TRUE condition. Likewise, you would see a negative voltage to represent a FALSE condition.

The actual voltage levels can vary considerably with RS-232. The standard says the amplitude of a signal originating from an RS-232 device must be at least ±5 volts and may range as high as 25 volts. The receiver is supposed to be somewhat more forgiving and accept an input signal of ±3 volts minimum. The range of values between -3 volts and +3 volts is considered to be a transition zone. Usually you see voltages of around 11 volts, give or take 3 volts.

So, for example, Data Terminal Ready (DTR) off might show up as -10 volts on the DTR line. Similarly, DSR on might appear as +10 volts on the DSR line.

Data Signals

Data signaling appears primarily on the Transmit Data (TD) and Receive Data (RD) lines. There are two active states you will observe: MARK and SPACE.

MARK equates to a "1" bit and will be seen as a negative 3 to 25 volts.

SPACE equates to a "0" bit and will manifest as a positive 3 to 25 volts.

Each symbol transmitted is preceded by a START bit, a "0" bit, and ends with one (usually) STOP bit, a "1" bit. There may also be a parity bit inserted before the STOP bit, but this is very uncommon these days.

Consider an example. Let's say we transmit the ASCII character "e" (lower-case). The ASCII code for "e" is 0x65, which is 01100101 in binary. You would see this on the wire:

```
0 10100110 1
```

The first "0" bit is the START bit, and the last "1" bit is the STOP bit. Notice that the data bits are serialized onto the wire from right to left (low-order bit first), so 01100101 shows up as the mirror image 10100110. See Figure 4. This depicts the letter "e" captured by the oscilloscope. You might like to compare that with Figure 5, which exhibits what an ASCII "r" (0x72, binary 01110010) looks like.

Notice that when no data is being transmitted the line is in a constant MARKing state, essentially a constant stream of "1" bits. So when a "0" bit (the START bit) does come in, it indicates the start of a new character.
Other Signals

A couple of other lines are important. There are two ground leads. One is Protective (or Chassis) Ground and the other is Signal Ground. Signal levels are determined relative to Signal Ground, but it is very common to tie the two ground leads together.

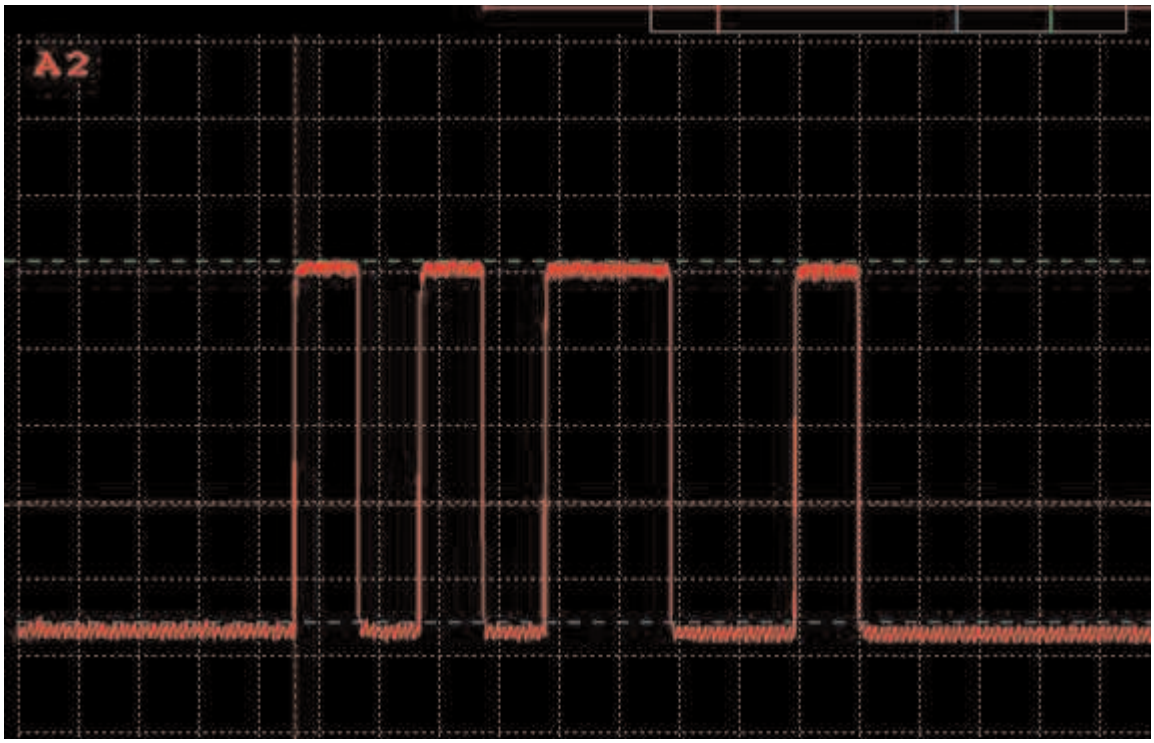

Figure 4. Lower Case "e" Bit Train

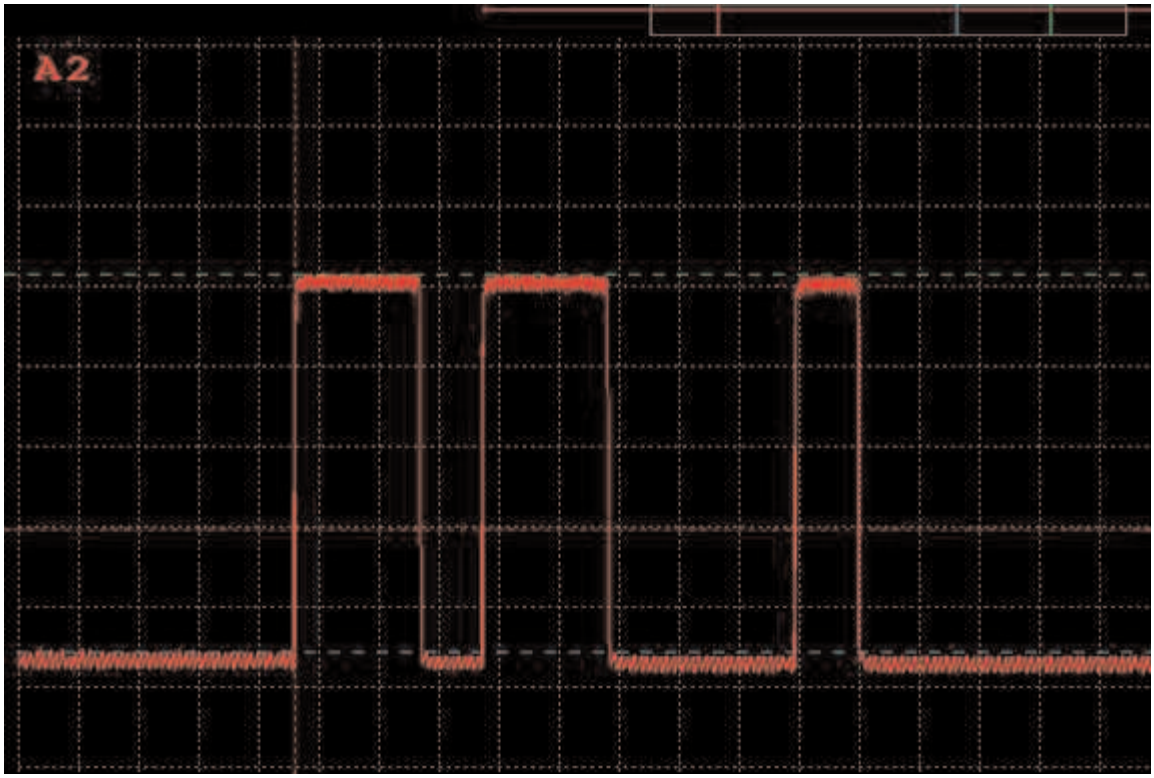

Figure 5. Lower Case "r" Bit Train

Request To Send (RTS) and Clear To Send (CTS) deserve a little special attention. These signals can be used in two ways. First, they are used as a form of hardware flow control, whereby dropping CTS you inform the other device that it is NOT clear to send, so it should wait until CTS comes back up before resuming transmission. Second, they can be used as a means of turning the line around on a half-duplex link.

These days we are largely spoiled; we are used to full-duplex communications in which we can send and receive simultaneously. That wasn't always the case. Some links used to be half-duplex – one channel that is shared by both modems. One person talks while the other listens, then you turn the line around, then the other person gets to talk. RTS can be used as the signal to the modem to turn the line around so that we can send.

Also of interest are the Transmit Clock and Receive Clock leads. These are used to provide timing signals for synchronous links. In a synchronous application the DCE device commonly provides clocking information to the DTE device so that the DTE device knows the rate at which to sample the data being received.

Be aware that we rely largely upon asynchronous RS-232 communications, in which case, these timing signals are not provided. Instead, in an asynchronous system, both the sending and receiving devices must by pre-configured to operate at the same speed and with the same symbol framing characteristics. This is why you are commonly asked to configure your COM port to 9600 bits per second, 8 data bits, no parity, and 1 stop bit - that's because those settings are the default for the console port on your switch or router.

How Control Signals Are Used

The role of the control signals is not apparent unless you have written software or firmware to control these kinds of devices. Allow me to outline for you a typical series of events involving a computer connected to a modem via an RS-232 interface.

- DCE (modem) initializes and asserts DSR, Data Set Ready. (Historical note: modems used to be called "data sets" many moons ago.)
- An incoming phone call is detected by the modem. The modem asserts RI, Ring Indicator, to the computer.
- The computer realizes that the phone is ringing so it brings up DTR, Data Terminal Ready, at which point the modem answers the incoming phone call.
- The modems on the two ends of the phone line coordinate various options with one another. This is when you hear the screeching and warbling on the phone line.
- When the modem is satisfied that it has a stable connection with the other modem, it brings up DCD, Data Carrier Detect, sometimes called RLSD, Receive Line Signal Detect, to inform the computer that the path is open for transmitting and receiving data.
- Data goes back and forth across the modem link.
- When the computer no longer needs the link, it drops DTR, signaling the modem to hang up the call.

Connectors and Pin Assignments

There are three connector types commonly used with RS-232: the 25-pin D-connector, the 9-pin D-connector, and the RJ45 connector. See Figures 6a, 6b, and 6c.
As many as 20 pins could be used conceivably, but many of the signals are obscure and have fallen almost completely into disuse. The important ones are:

Figure 6a. DB-25 Male

Figure 6b. D8-9 Female

Figure 6c. RJ45

Quite frankly, Ring Indicator can be easily dispensed of, because initializing the DTE with DTR asserted the modem will automatically answer an incoming call. Then DTE can simply key on CD to know when it's okay to pass data.

So we are down to 10 or so pins that are actually of importance to us. Maybe we don't need that big 25-pin connector. Alas, that's why you see the 9-pin DB-9 and 8-pin RJ45 so widely used.

On the Internet, diagrams abound that illustrate the various ways these connector types can be wired together. A couple of handy tables are good to have though. See Figures 7 and 8 for the pin assignments from the DTE perspective. They summarize the pinouts for all three type of connector.

Direction	DB-25	D8-9	Description
	1		Protective Ground
DTE --> DCE	2	3	Transmit Data
DCE --> DTE	2	2	Receive Data
DET --> DCE	4	7	Request to SEnd
DCE --> DTE	5	8	Clear to Send
DCE --> DTE	6	6	Data Set REady
	7	5	Signal Ground
DCE --> DTE	8	1	Data Carrier Detect
DTE --> DCE	20	4	Data Terminal Ready
DCE --> DTE	22	9	Ring Indicator

Figure 7. Commonly Used Signals and Pin Assignments

DTE			Adapter	
Signal	RJ45 Pin	RJ45 Pin	D8-9 Pin	Signal
RTS	1	8	8	CTS
DRT	2	7	6	DSR
ExD	3	6	2	RxD
6ND	4	5	5	GND
6ND	5	4	5	GND
RxD	6	3	3	TxD
DSR	7	2	4	DTR
CTS	8	1	7	RTS

Figure 8. RJ45 Rollover Cable Pin Assignments

You may be curious about the two genders of D-connectors. First, be aware that the world is not neat and orderly in regard to the use of the genders. The standard suggests that the DTE device should have a male connector, and the DCE device should have a female connector. And by-and-large, that is what you will see. However, don't be surprised to find that a particular situation calls for a male-male or female-female cable – it happens.

Also, take a moment to consider the difference in the pin assignments between the male and female connectors. Think about it. When you plug the male connector into the female connector, the pin at the top left of the male connector will mate with the hole at the top right of the female connector. So the pin designations are the mirror image of one another for the male and female connectors. In short, when viewing the connectors head-on with the wide side of the trapezoid at the top, pin 1 is at the top left of the male and at the top right for the female.

Handy Tools

If you work with RS-232 now and again, then you will certainly benefit from having a few handy tools that can make your life a bit easier.

First and foremost you need to have a trusty known-good cable. For me that is usually a six-foot cable with DB-9 male and female connectors. I also keep a Cisco rollover cable handy as well that has a DB-9 female and an RJ45.

Those cables won't suffice in all cases. You may encounter a device with a 25-pin connector. Or, you may find that you have a gender conflict. (I don't mean you personally.) This is why you need an assortment of adapters like those illustrated in Figure 9.

Figure 9. Various Adapters

Another common situation that one encounters is the need to connect two DTE devices together. They both expect the other side of the connection to behave as DCE – and they'll be disappointed. This is where a Null Modem adapter is very handy.

There are various ways that Null Modem cables or adapters can be constructed. Here is an example that I used to use to also cut down on the number of conductors that I needed in the cable; see Figure 10.

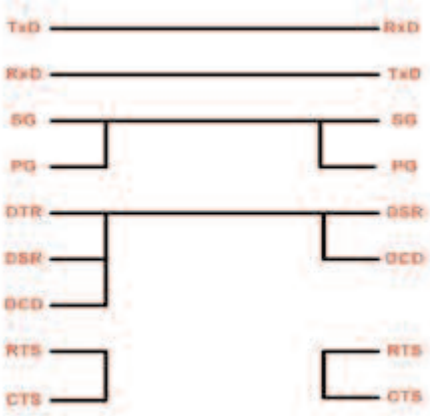

Figure 10.

Obviously, if you want two devices to talk directly with each other, you have to connect the Transmit Data lead from one device to the Receive Data lead on the other device; and vice-versa. Tying the two ground leads together is also a common thing, so we connect Signal Ground and Protective Ground (Chassis Ground) on both ends together. The last trick is quite simple. When Data Terminal Ready comes up on the left side of the cable, Data Set Ready and Carrier Detect are brought up on both sides of the cable. And finally, when Request To Send is asserted on either side of the cable, Clear To Send is immediately supplied to the device.

Figure 11. Bit Error Rate Tester

You will find a variety of ways that folks have come up with to make such cables and adapters. The Internet is littered with diagrams. Some are simpler than others. Some are more complicated. And finally, some work better in certain circumstances than others. You get the idea; it's not a one-size-fits-all world out there. So when you get a good adapter or cable that works well for you, then put your name on it and take care of it.

If you have a need to test an RS-232 interface, then Bit Error Rate Tester (or BERT) is the tool for you. See Figure 11.

If you simply want to observe the state of the control signals, that's easy. You need a Breakout Box (or BOB). See Figure 12.

Figure 12. Breakout Box

Perhaps you have occasion to observe the data that is being sent back and forth across an RS-232 link. For that you need a Line Monitor, which is sometimes called a Data Line Analyzer or a Data Scope. A Line Monitor can be a standalone device that you wire in-between the two devices you wish to examine. Typically, such a tool has a CRT display upon which you can watch the data going back and forth and a built-in breakout display that shows you the control signals. A line monitor can also be implemented using software for your PC. In this case however, you have to have multiple RS-232 adapters in your PC that are supported by the software vendor. There is a tangle of special cables that go with these products as well.

Both kinds of line monitors allow you to record the traffic on the communications link. This includes the control signals, not just the data. Certain tricky problems, especially those having to do with the timing of events, require a line monitor. It is common to find older equipment of this kind for sale on the Internet at very low prices.

Last, but not least, is an oscilloscope. This is probably the most basic of test equipment on your electronics workbench. The oscilloscope allows you observe how the RS-232 signals change over time. You can see when a signal goes high. You can see when it goes low. You can observe the voltage levels. You can see the timing of the data bits as they fly by on the wire. (See Figure 13 showing the name "fred" traveling down the wire.) It is truly the best way to really understand the behavior of an RS-232 interface.

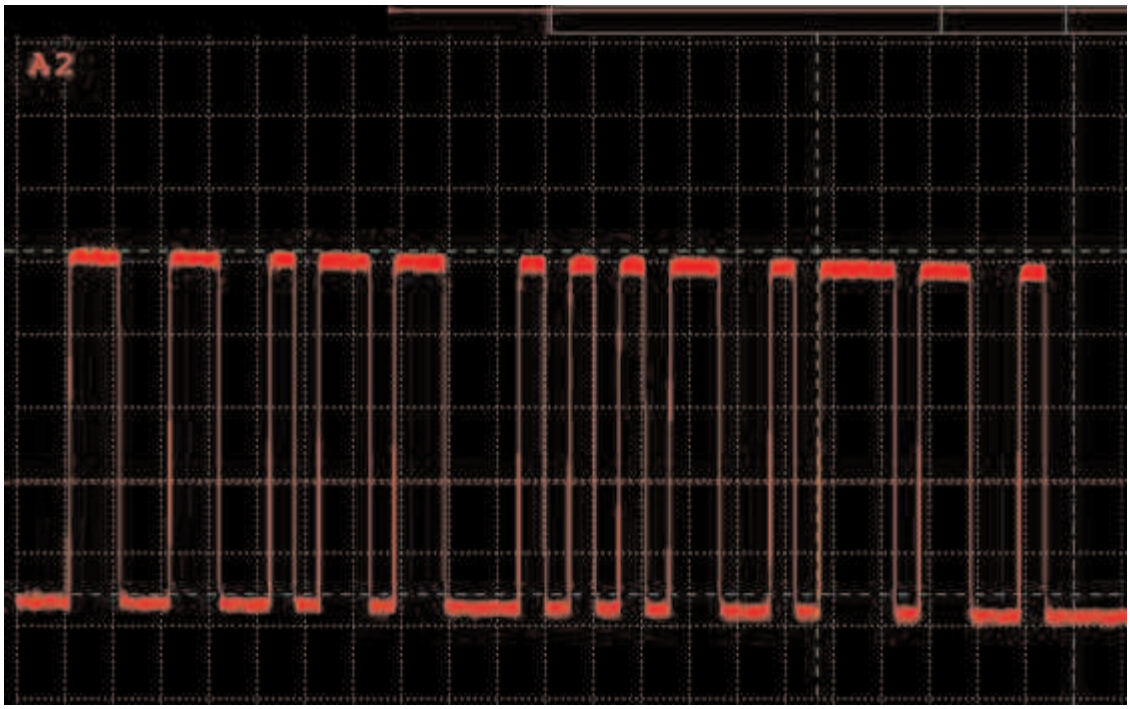

Figure 13. Oscilloscope Display of "fred"

One further note: if you want to observe all of the signals at the same time, then a Logic Analyzer is the tool for that. These can be found on the Internet at affordable prices, too. But beware, make sure that you get all of the probes and wiring harnesses if you decide to buy a used analyzer – those things will prove to be difficult and/or costly to replace if they are not included with your purchase.

Data, Parity, and Stop Bits

The symbols transmitted over an RS-232 link can vary somewhat in how they are organized or formatted. The variables are:

- **Data Bits –** the number of data bits in each symbol transmitted. This may be set to a value between 5 and 8. The old Baudot code used by Teletypes was a five-bit code. ASCII is traditionally considered to be a seven-bit code. Now, more commonly, we employ the eight-bit ASCII variants that emerged with the popularity of the PC.

- **Parity Bit –** an error checking option. Possible values are EVEN, ODD, NONE, MARK, and SPACE. The NONE option is most common these days. But there was a time when the EVEN parity was most common. This adds an additional bit to each symbol transmitted. Here's how that bit is set:

  - **EVEN –** if the number of Data Bits set to "1" is an odd value, the parity bit should be set to "1" so that there is an even number of "1" bits. If there is already an even number of "1" bits, then the parity bit is set to "0".

    Example: ASCII "r" (0x72) 0 01001110 0 1

- **ODD –** if the number of Data Bits set to "1" is an even value, then the parity bit is set to "1" so that there is an odd number of "1" bits. If there is already an odd number of "1" bits, then the parity bit is set to "0".
  Example: ASCII "r" (0x72) 0 01001110 1 1

  NONE – no Parity Bit is transmitted.
  Example: ASCII "r" (0x72) 0 01001110 1

  MARK – the parity bit is always a "1" bit.
  SPACE – the parity bit is always a "0" bit.

- **Stop Bits –** the number of Stop Bits at the end of the symbol. Recall that the Start Bit is always a "0" bit and that the Stop Bit is always a "1" bit. The possible values are 1, 1.5, and 2.

Doing a little arithmetic, it becomes apparent that symbols transmitted via the RS-232 interface could be a bit train as short as 7 bits (1 Start, 5 Data, no Parity, 1 Stop). Likewise, it could be as long as 12 bits (1 Start, 8 Data, 1 Parity, 2 Stop).

As you might imagine, it is absolutely imperative that both sides of an RS-232 connection be configured in the same way if they are to successfully communicate.

Far and away the most common default settings for RS-232 devices is 9600 bits per second, 8 data bits, no parity, and 1 stop bit. You may have noticed the little "8N1" in the status window of HyerTerminal – that reflects the symbol framing.

Conclusion

Experts have been predicting the death of RS-232 for decades, and yet it continues to serve us well. It is a highly proven technology that is inexpensive and easy to implement. And it is well understood and widely accepted in IT and related endeavors.

It will be interesting to reread this paper in 2018….

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge.
Network+ Boot Camp
Understanding Networking Fundamentals
Internet and Network Communications

For more information or to register, visit www.globalknowledge.com or call 1-800-COURSES to speak with a sales training advisor.

Through expert instruction, you will understand key concepts and how to apply them to your specific work situation. Choose from more than 700 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT and business training needs.

About the Author

George Mays has over 35 years of experience in computing, data communications, and networking, including mainframe systems programmer, Fortune 500 DBA, management of systems programming, data communications, IT operations, engineering, software development, and networking. He is also the author and course director for Global Knowledge's Network+ Boot Camp and has contributed to several hacking and security books. He is an instructor in TCP/IP, Troubleshooting, Network Protocols, Network Fundamentals, A+, Security+m and CISSP. George holds various industry certifications including CISSP, CCNA, A+, Network+, Security+, INet+, and he acts as a consultant in the fields of general networking and security.