RS-485 - A Proud Legacy

Author: Advantech
E-mail: eainfo@advantech.com
“What hath God wrought?” This message, one of the first serial digital communications in history, was sent in American Morse Code on May 24, 1844 by Samuel F. B. Morse to open the first long distance telegraph line (from Baltimore to Washington DC). Since then, serial digital communications have been one of the strongest backbones of technology, from computer backplanes to industrial controls. And they show no sign of being replaced soon.

In the early modern computing era (the 1960s and early 1970s) serial communications was selected by the vast majority of vendors for device and IC integration. This came about for several reasons, including simplicity, easier interconnection and higher clocking speed. Since then, parallel communications buses (like the Centronics printer bus) have practically died out in favor of serial communications, both inside computers and as interconnecting buses like USB and SATA.

The Serial Communication Concept

From its beginning with Morse code, a serial communication consists in transmitting a single bit of data at a time, sequentially, forming a serial data stream. When error-checking bits are included, it is possible to easily reassemble the message at the receiving end.

After Morse code, the most common early serial communication standard was RS-232, developed in 1962 for interconnecting mainframe terminals, modems and teletypewriters. It was extended,
amended, revised and used for applications its original designers could never have even visualized. It is still being used today.

One of the earliest uses for RS-232 (now known correctly as TIA-232-F) was as an output bus for early personal computers. Most early personal computers, especially the IBM PC and its clones, came with an “RS-232 compatible” output connector. These have now been superseded by USB and other high speed serial bus communications.

**Industrial Device Communications**

RS-232 invaded the industrial space first as a way to connect things like check-weighers to PCs and other devices. When the PLC took over factory floor control in the 1970s, at first they tended to be standalone devices…what used to be called the “Islands of Automation.” As control systems became more integrated, though, it became necessary to find ways to inter-communicate between PLCs and the human-machine-interfaces (HMIs) that were beginning to be used on the factory floor. Serial data communications was selected because it was robust, relatively immune to EMI, RFI and crosstalk, and could be connected with a very few (as few as four) connection wires.

**Flavors of Serial Communications**

There are three most commonly used versions of serial data communications standards. They are the Recommended Standards (RS) from the early 1960s: RS-232, RS-422 and RS-485. Each of them is still in use in industrial and laboratory data communications.

RS-232 is a single-ended, unbalanced full-duplex point-to-point communications methodology, with a maximum data rate of 19.2 kilo baud and a maximum physical range of about 15 meters (roughly 50 feet), and a single driver and receiver.

**RS-232:**

**Unbalanced Line Drivers**

---

**RS-232 Interface Circuit**

- Driver Loaded Output Voltage Range = ± (5 to 15 volts)
- Receiver Range
  - Low: +3 volts
  - High: +5 volts
  - Range:
  - SPACE or logic 1
  - MARK or logic 0
An early variation on RS-232 was RS-422 which allowed burst communication to up to 10 receivers simultaneously from a single driver. RS-422 was differential balanced, full duplex point-to-point, had a range of 1200 meters (about 4000 feet), and could operate at a much higher maximum data rate than RS-232, about 10 Mbaud.

The first attempt at a distributed networking communications standard was another variation on RS-232 called RS-485. RS-485 was differential balanced like RS-422, but where the -232 and -422 variants had a single driver; RS-485 was capable of up to 32 drivers and up to 32 receivers in a half-duplex multidrop configuration. It, too, could reach up to 1200 meters (4000 feet) and also operated at 10 Mbaud. It’s fast, efficient, robust and is still in very common use today.

RS-485/422: Balanced Line Drivers
The multi-drop capability of RS-485 made it possible for the first time to integrate communications into the PLC-based automation products on the factory floor.

Why Have RS-232, -422 and -485 Survived Ethernet?

Ethernet, (IEEE 802.3) the most commonly used networking topology, is also a serial communications standard. Since it is used so ubiquitously in networking, one must ask why it hasn’t replaced all the RS-232 variants entirely, with “Ethernet to each device,” as Cisco famously intends.

Comparing RS-485 with Ethernet, we find both advantages and disadvantages to each. The major disadvantage of RS-485 is communication speed. The maximum data transfer rate for RS-485 is, as we have said, 10 Mbaud. But that can only be sustained to the same range as Ethernet, 100 meters (328 feet). Typically, RS-485 links are operated at much reduced speeds, such as 9600 baud or 19.2 kbaud. This data transfer speed reduction allows operation up to the 1200 meters (4000 feet) that we claimed for it.

RS-485 is designed to operate as a half-duplex single master with 31 slaves (32 receivers includes the master). The master polls each slave, waits for the response, and then polls the next slave. Collisions of data packets are therefore averted.
Ethernet, which could be described as a sort of full-duplex high speed version of RS-485, has no intrinsic method of avoiding data packet collisions. In applications like process control, the lower speed of RS-485 is usually more than adequate.

RS-485, when operating at the reduced speeds under 115 kbaud, has a range of 1200 meters (4000 feet) as we've said. Ethernet, however, must have a repeater every 100 meters (328 feet) to amplify the signal.

Typically, RS-485 uses a single, twisted pair of “copper” conductors. Twisted pair allows the balanced differential RS-485 signal to reinforce itself and resist interference and crosstalk. Ethernet also uses copper twisted pair wiring (as well as fiber) but since RS-485 is operating at considerably lower speed, and is also half-duplex, RS-485 tends to be more resilient to EMI (electromagnetic interference), RFI (radio frequency interference) and noise.

RS-485, as well as Ethernet, can be propagated wirelessly, often using the IEEE 802.11x or WiFi standard. Both electrical standards can be installed using fiber optic cable.

**Protocols versus Standards**

Many automation professionals confuse an electrical standard like RS-485 or Ethernet (IEEE 802.3) with a protocol like Modbus or Profibus.

Modbus was originally developed at Modicon, now Schneider Electric, in order to communicate between Modicon PLCs and HMIs on the plant floor. Hence the name MODiconBUS, or Modbus.

Modbus was the first industrial communication protocol developed specifically for PLCs and other controls, and it is still extremely widely used. There are several forms of the protocol, including Modbus RTU and Modbus TCP/IP. Modbus RTU is usually transmitted over an RS-485 communication connection. Modbus RTU can be communicated over wireless, fiber, or copper, and can be used in RS-485, RS-422 or RS-232 standard formats. Modbus can even be communicated over undifferentiated Ethernet, although it is likely that a better solution would be Modbus TCP/IP, which was designed specifically to communicate Modbus data over Ethernet.

Some protocols, such as the legacy A-B Data Highway from Allen-Bradley (now Rockwell Automation) and Profibus, require specialized variations of RS-485 technology. This is not unusual. From the beginning, with the prefix “RS” standing for “Recommended Standard,” vendors have created many variations of the RS-485 standard. This often contributes to interoperability issues and other design problems. RS-232, RS-422 and RS-485 have been around for enough time for IC vendors to have produced special VLSI chipsets for the standards. That is, when you build your product, you use a standard chipset that you purchase to provide the RS-485 output. This would be great, except that the “standard” chipset may very well have proprietary features that render compatibility and interoperability problematic. For example, there are RS-485 chips that support more than the “standard” 32 transmitter/receivers (1 host, 31 slaves).

In addition, while the “standard” implementation of RS-485 is half-duplex, it is not unusual to see full-duplex implementations of the standard. Full-duplex requires careful management of the communication, using additional software, especially if multiple masters are implemented or the potential for multiple slave devices to respond immediately exists. RS-485, being a serial polling standard, has difficulty with interpreting multiple simultaneous responses, and the data stream can become corrupted if proper attention is not paid at the design level.
Unlike RS-232 and RS-422, an RS-485 device must receive an instruction to shift from low impedance Transmit mode to high impedance Receive mode. This must be done so the next device in the polling chain can itself shift from Receive to Transmit mode. This is critical, since if two devices are in low impedance Transmit mode at the same time, this prevents the communication signal from being transmitted clearly—the serial data bits can become garbled and the message unreadable.

**Timing diagram of RTS**

In RS-485 implementations, therefore, timing of the physical state of the device is critical. Hardware vendors manage this differently but equivalently. Thus, it requires software or protocol coordination to fully manage this timing issue effectively.

For instance, in the popular Advantech ADAM-4520 series of RS-232 to RS-485 converters, a firmware method called “Auto Flow Control” is used.

This control resides in the hardware, and it automatically shifts the converter’s RS-485 component from transmit to receive when the data stream stops changing for about two bytes. Advantech says this feature is baud rate configurable. Auto Flow Control works well, but the software or protocol must follow the common RS-485 design—do not transmit another command to the slaves until a reasonable time-out period has passed, which accommodates the responses of the slave devices.
Even though RS-485 networks can reach up to 1200 meters (4000 feet) without any additional amplification simply by lowering the baud rate of the data transfer, there are times where long range and high baud rate are required, or a custom configuration of devices, such as a star or tree topology is needed. Doing this also extends the number of devices from 1 master and 31 slaves (32 transmitter/receivers) to multiples of 32. For example, Advantech accomplishes this with its ADAM-4510 series of RS-485 Repeaters.

RS-485, as can be seen from the ADAM-4510 Series, can be used to extend the signal from RS-232 and RS-422 devices. This is important in interoperating with legacy devices on the plant floor.

RS-485 Is Not Going Away!

Formerly, it was believed that networking technologies, as well as industrial control elements, were going to follow a conventional lifecycle. As it has turned out, control systems are lasting far beyond their design life expectancies. It is not uncommon to find PLCs in plants that have been installed for two or three decades. This means that the communication protocols those PLCs and other control devices were designed to use are also lasting beyond their own design life expectancies.

Ethernet, because of determinism issues, did not become a trusted method of communication in automation applications until the mid 1990s with the invention of the Ethernet switch. Its acceptance has taken time, and many devices on the plant floor continue to have only RS-485
communications. There are even current implementations of RS-485 by hardware vendors making new products using the almost 50 year old standard.

When RS-485 was developed, the perceived need for communications security on the plant floor was much less than today. Even during the development of Ethernet, a far lower premium was placed on data security than robustness and throughput. Once the Internet came along, and Ethernet connections to the outside world became ubiquitous, there have been created a whole new level of security concerns. Since RS-485 is not normally connected to the Internet, it continues to provide an additional "air gap" that reduces system vulnerability. No system is ever completely invulnerable to attack, and RS-485 systems, especially those running Modbus TCP/IP, are sometimes connected to the Internet, but RS-485 can provide an additional level of security.

RS-485 is typically easily understood and applied, even by personnel with limited experience in networking and communications. This makes it a very important method of networking for small automation applications with smaller budgets. Since it often employs relatively open and mostly free architectures like Modbus RTU or user designed ASCII protocols, RS-485 is easier for overworked staffs to implement than other standard-based communications methods. Far from being antiquated, because of its ubiquity, RS-485 has been updated and easily interfaces to common application development environments like .NET.

“Works fine, lasts a long time!” could be the motto of the serial data communications family of RS-232, RS-422 and, of course RS-485. A proud legacy indeed.