Serial Ports Still Bash the Bits

Jon Titus, Senior Technical Editor

Like many inexpensive desktop PCs, my Dell Dimension C521 lacks a serial port. But, don't write off serial communications which will continue to play important roles in industrial controls, point-of-sale terminals and other equipment. USB and Ethernet have a place, but simple serial I/O still solves a lot of problems.

Microcontrollers (MCUs) usually include at least one universal asynchronous receiver/transmitter (UART) that can transmit eight data bits serially. But, unlike USB and Ethernet ports, a UART does not require a complex software stack and an API. Although UART operations require some set-up of control registers and configuration of I/O pins, simple software handles these tasks. MCU vendors may offer a wizard or basic UART library that makes on-chip UARTs easy to use. (The first UART chips, circa 1971, came in 40-pin ceramic DIPs.)

Serial ports usually communicate via a standard that defines signal levels, pin configurations and signals originally designated to control modems. Engineers commonly refer to this standard as "RS-232," but as of 1997, the proper designation is ANSI/EIA/TIA-232F. Early modems used a 25-pin DB-25 connector that supplied many unneeded control signals. But for many years, board and industrial computer vendors have used a standard 9-pin DE-9 connector to offer users and developers a serial-port connection. Basic two-way RS-232 communications use three pins: transmitted data, received data and ground. Control signals use the other six pins. Signal voltages must comply with the RS-232 standard's specifications.

Figure 1. This DIN485+I converter lets an RS-232 device communicate with one RS-422 device or multiple devices in an RS-485 network. Courtesy of Sealevel Systems.

The RS-232 standard covers a single-ended communication scheme that usually involves one transmitter and one receiver (point-to-point). You can implement single-transmitter/many-receiver (multi-point) connections, but you are on your own because the standard does not define them.
Most RS-232 links do not extend much beyond 100´, although I have used a 1,300-foot low bit-rate link without trouble. Like all single-ended communication paths, though, an RS-232 circuit can run into difficulties in an electrically noisy environment.

To increase noise immunity, engineers might adopt the ANSI/EIA/TIA-422-B (RS-422) standard that defines differential signals, and line drivers and receivers for serial communications. Unfortunately, the RS-422 standard does not specify a connector type or pinouts. So, engineers adopt their own connector which may not easily connect to third-party equipment that also uses RS-422 signals.

As an alternative to using RS-422 differential driver and receiver ICs for on-board serial ports, engineers can opt to add an external adapter that converts to and from single-ended RS-232 and differential RS-422 signals. Thus, equipment designers maintain standard RS-232 ports in their equipment, but they can still extend communications over longer distances and enhance RFI/EMI immunity. Keep in mind that if you adopt RS-422 signaling, the RS-232 standard still defines control-signal operations. Also, RS-422 signaling requires five wires -- one for ground and two each for transmitted data and received data.

You can use RS-422 communications in a point-to-point or multi-point network configuration. However, some communications require a multi-drop arrangement that includes several transmitters and receivers. A multi-drop arrangement works much like the CAN bus described in our December Embedded Systems column. One pair of wires, properly terminated, carries all communications. So information can travel between any transmitter and any receiver, but only in one direction at a time.

The ANSI/EIA/TIA-485 (RS-485) standard defines this type of differential communication. Suppliers such as Maxim, Texas Instruments, National Semiconductor and Linear Technology offer interface ICs that work with either RS-422 or RS-485 links. You find RS-485 communications mainly in industrial and measurement equipment that must operate in electrically noisy areas. Single-board computers and serial-I/O boards often let developers select either RS-232, -422 or -485 signals for communications. The latter two signal types may require small add-on boards that provided RS-422 and -485 transmitter/receiver ICs.

The RS-485 standard does not specify a standard connector, although many vendors have adopted the same 9-pin DE-9 connector they use for RS-232 ports. This 9-pin RS-485 connector supplies differential received-data and transmitted-data lines, differential Clear-to-Send (CTS) and Ready-to-Send (RTS) control lines and a ground connection.

Serial I/O networks that involve more than two receivers or transmitters require a protocol that addresses devices and controls the flow of data. You can adopt the standard UART protocol -- start-bit, eight data bits and stop bits -- but if you use the

data bits for information and addresses, things can get tricky. Some UARTs can provide a 9th bit that you can use to distinguish address from data transmissions. In some cases, you can use, say, three bits to address your linked RS-485 devices and the remaining five address bits to control operations at the addressed device.

Years ago, Motorola produced an addressable asynchronous receiver/transmitter (AART) chip, the MC14469, which responded to a seven-bit address. If you want a multi-point RS-232 network, the datasheet for the now-obsolete MC14469 will help you understand how to implement similar functions in an MCU.

For further reading . . .

Axelson, Jan, "Serial Port Complete," Lakeview Research. www.lvr.com. [1]
Find the Motorola MC14469 data sheet at: http://preview.tinyurl.com/28wzyc [2]
"RS-422 and RS-485 Application Note," B&B Electronics.
http://preview.tinyurl.com/7woca [3]
"Z8 Encore! 9-Bit UART Implementation," AN 0146, Zilog.
www.zilog.com/docs/z8encore/appnotes/an0146.pdf [4]

Source URL (retrieved on *06/19/2013 - 4:07am*):
http://www.ecnmag.com/articles/2007/12/serial-ports-still-bash-bits?qt-most_popular=0

Links:

[1] http://www.lvr.com.%3c/t1
[2] http://preview.tinyurl.com/28wzyc%3C/t1
[3] http://preview.tinyurl.com/7woca%3C/t1
[4] http://www.zilog.com/docs/z8encore/appnotes/an0146.pdf%3C/t1