Serial Lines and Devices
Frish, Chapter 12

Technical System Administration
CSCI 4980-001 and ISQA 4000-001

Serial vs. Parallel Communication

- Every device in a computer system must be connected to the system somehow. What’s being communicated is either digital data (bits) or control signals.
- There are two basic types of connections between a device and the system:
  - Serial – one bit at a time, one after the other
  - Parallel – more than one bit at a time, with multiple bits traversing the connection at the same time

Parallel Examples

- The most traditional (read “oldest”) method for connecting a printer to a PC is to use a 25-wire parallel cable.
- With the exception of the newest models (SATA), disk drives (IDE, SCSI, etc.) are connected to the system using a parallel (ribbon) cable.
- The processor (CPU), primary memory, and device controllers communicate using parallel connections (the system bus).
- Mainframes used to connect multiple “frames” (separate cabinets) using “bus and tag” cables – in parallel.

Serial Examples

- USB – Universal Serial Bus – although the cable has several wires, the data communication is entirely serial.
- Ethernet – heavily used in local area network communication, one bit at a time is transmitted.
- Fiber optics – although there aren’t wires involved, only one bit at a time is communicated.
- Wireless networking – again, no wire are used, but only one bit at a time is communicated.
- RS-232 and RS-422 – the oldest, but still popular, serial communication techniques. Most machines used to include serial ports are required equipment, but this practice is fading.

RS-232 and RS-422

- In this module, we focus on RS-232 and RS-422, although we will also briefly touch on USB.
- The names RS-232 and RS-422 are standards. (Actually, there is usually a letter suffixed to distinguish versions of the standard, but we’ll ignore that here, except to note that RS-232C is the most common standard used.)
- RS means “Recommended Standard.”
- RS-232 is the older standard, but is still the most popular.
- RS-422 requires different equipment, but offers higher performance than RS-232 systems.

DTE and DCE

- The terminology for serial cables draws heavily from that used for the connection between serial terminals (like a VT-100) and a modem (connected to a telephone line).
- The modem end of the cable (or the modem itself) is called the DCE – Data Communication Equipment.
- The terminal end is called the DTE – Data Circuit Terminating Equipment.
What’s In a Minimal RS-232 Cable?

- The minimal RS-232 cable has three wires:
  - Signal ground – the common level on which all other signal levels are based
  - Transmit Data (TxD) – carries data from the DTE to the DCE
  - Receive Data (RxD) – carries data from the DCE to the DTE

What Else Could There Be?

- DCD – Data Carrier Detect (DCE->DTE); indicates when the modem has detected a connection
- RTS/CTS – Request/Clear To Send; used to provide control over when data can be transmitted
- DSR/DTR – Data Set/Terminal Ready; indicates when the modem (a “data set”) or terminal is ready (powered up and “good to go”)
- RI – Ring Indicator; used only with a computer as DTE; allows computer to control when to answer.
- FG – “Frame” ground; common DC ground.

What is a “tty”

- "tty" is an acronym or shortcut for Teletype®, one (and perhaps the largest) manufacturer of teleprinter devices (another manufacturer was Kleinschmidt in Germany).
- This is still used to identify those serial lines that connect to keyboard/display devices used for communication with computers, even though the teleprinter as a primary interface device has been obsolete for decades.

Serial Device Files

- The files /dev/tty* correspond to serial devices (DTE) in a UNIX/Linux system (with some system variations).
- The file /dev/tty (with no number) is a “shortcut” to the controlling terminal for a process. For example, writing to this device bypasses any redirection of the standard output.
- The files /dev/tty* are “pseudo” ttys. To a program, they look (and are programmed) like ordinary tty devices, but each is paired with another “device” that allows them to handle networked connections. When you use telnet or SSH to connect to a UNIX box, your “terminal” is a pty device. (Other filenames can be used for pseudo ttys.)

Terminals vs. Serial Lines

- The terminal – serial line relationship is so strongly embedded in UNIX history that for many purposes, a terminal IS a serial line.
- You will find both terms used, but they are frequently interchangeable.
- About the only case where careful distinction must be made is when a “real” terminal is connected to a serial line.

Terminal Characteristics

- Speed, measured in Baud. In the old days, things like 110, 300, and 1200 baud were common. Now we use 9600 and 19200 baud frequently. For ptys, a “fake” baud rate is given.
- Special characters – which input characters cause special actions (example: control-S usually stops output, control-Q restarts it, and control-C causes a SIGINT signal to be sent to the controlling process)
- “Real” terminal characteristics (e.g. width)
- Control settings (e.g. ignore RTS/CTS), input (e.g. what happens when the BREAK key is pressed?), output settings (e.g. time for a carriage return), local settings (e.g. what happens on a backspace?), combination settings (e.g. raw or cooked mode?)
What's a Baud?
- Named after Emile Baudot, a Baud (or baud) is a signaling speed equal to one "state change" per second.
- For communication lines, the number of states possible is just the number of "persistent" signal values used.
- Example: The RS-232 standard states that the low, or "mark" state is indicated by a voltage (relative to the signal ground) of \(-3\) to \(-15\) volts (typically \(-12\)). The high, or "space" state is \(+3\) to \(+15\) volts.
- To move the signal from \(-12\) to \(+12\) volts, it must clearly "pass through" all voltages in between. But those "in between" voltages do not represent "persistent" states.

What's a Bit Per Second (bps)?
- The phrase "N bps" refers to N bits being communicated in one second.
- "Bits Per Second" are not identical to "Baud"!!!!
- Example: Suppose we use a technique with 4 signal levels. Since each signal level could represent a two-bit value (00, 01, 10, and 11), a rate of 1 baud is equivalent to 2 BPS.
- A rate of K baud is equivalent to \(\log_2 K\) BPS.
- Warning: be aware that the term "byte per second" (sometimes abbreviated bps) is also in common use. The difference from bit per second is significant (by a factor of 8, of course).

Bits, Bytes, and Character Codes
- Today, we almost universally think of a byte (8 bits) as being equivalent to a character (that is, capable of containing a character code; 0x41 = 'A', etc). This wasn't and isn't always the case.
- For example, many systems now address international communication issues by using Unicode, in which 16 bits are used to contain a character code.
- Earlier systems used 5, 6, or 7 bits to contain a character code (with fewer characters capable of being directly encoded, of course). For example, ASCII (American Standard Code for Information Interchange) is really a 7-bit per character code.

Synchronous Communication
- In a synchronous communication system, there is a continuous stream of bits being sent – all the time (that's what continuous means, of course). The beginning and ending of the signal for each bit is carefully marked, or synchronized, by using a clock signal that is shared by the sender and the receiver.
- If there is no data to communicate at a particular time, the transmitter sends a synchronous idle bit pattern, which asks the receiver to "please hold."
- Synchronous communication predominates in commercial high-speed serial communication.

Asynchronous Communication
- Most low-speed serial communication is asynchronous: if there's nothing to send, then the signal is held in the marking (\(-12\)volt) state. Transmission of a character can start at any time, and no clock signal is used.
- To announce the beginning of a character, the sender changes the state (from mark) to space (+12 volt) for one bit time (the length of time required to send one bit). This is called the start bit.
- There then follow 5, 6, 7, or 8 data bits (taking 5-8 bit times), an optional parity bit (one bit time), and 1, 1.5, or 2 stop bits (taking 1, 1.5, or 2 bit times).

Asynchronous Example
- The letter "B" (66 decimal = 42 hex) transmitted as a start bit, 7 data bits (least significant bit first), an optional parity bit, and a stop bit (or bits).
8n1, 7e2, ...

- The number of data bits, parity selection, and number of stop bits must obviously be agreed to by each end of a serial communication link.
- The traditional way of representing this is as a digit giving the number of data bits, then the letter n/e/o (no, even, or odd parity), and finally the number of stop bits.
- (1.5 stop bits is/was used only with the 134.5 baud IBM 2741, basically a modified Selectric® typewriter.)
- Note that to transmit one byte (8 data bits) it requires at least 10 bits times (1 start + 8 data + 1 stop bit). So 9600 baud really represents 960 characters per second (at best).

Parity

- Parity bits are used for error checking.
- An even parity bit is chosen to make the total number of “1” data bits even.
- An odd parity bit is chosen to make the total number of “1” data bits odd.
- In the previous example, an even parity bit would be “0”, and an odd parity bit would be “1”.
- These bits, if used, are generated by the sender and checked by the receiver.

Framing, and Framing Errors

- Since the receiver knows that there must be a start and stop bit, then the presence of these is checked (usually by sampling the signal state several times during the period when the bit is supposed to be present.)
- Essentially, the start and stop bits “frame” the data and optional parity bits.
- If a stop bit is not found as expected, then a framing error is reported to the software.

Serial Hardware

- Other than the connector, the most common device in a serial interface on a computer is called a UART – a Universal Asynchronous Receiver Transmitter.
- Common chip designations are the 16550, 82450, and 8250. Sometimes these devices stand alone as individual chips, or are included as part of a larger ASIC chip (a chip containing many common devices needed for a PC).
- Single or double port serial “add-on” cards are available for PCs, as well as multiport (4, 8, 16, or even 32 port) cards for applications requiring many serial ports. For example, providers of dial-up Internet connections might have systems with hundreds of serial ports (and modems).

What Terminal Am I Using?

- The command “tty” will simply display the tty file to which the standard input is connected.
- For example:
  $ tty
  /dev/pts/0
- Another example:
  $ tty /dev/null
  not a tty

Showing Terminal Settings

- The command “stty” with no options can be used to display a terminal’s settings.
- Example:
  $ stty
  speed 38400 baud; line = 0;
  erase = ^H;
  -brkint -imaxbel
Modifying Terminal Settings

- "stty" can also modify terminal settings.
- Example – turn off input echo
  $ stty –echo
  $ echo *This is not echoed on input*
  This is not echoed on input
  $ stty echo $ ... input character echoing resumes
- Example – turn off echo on /dev/tty0
  $ stty –echo –F /dev/tty0

What “Real” Terminal am I Using?

- Try this first:
  $ echo $TERM
- This will display the setting of the shell variable
  named TERM, which is a name of a terminal.
- The terminal name (usually matching a “real”
  terminal name, frequently a vt100) references an
  entry in a terminal “database” that specifies how to
  handle input from and output to a “real” terminal.

The Termcap Database

- On Linux, the file /etc/termcap (or perhaps
  /usr/share/termcap) contains the termcap
database. This is just a “flat” ASCII file.
- "Termcap" = Terminal Capabilities
- Each entry is a (logical) line (with backlash end-
of-line used for line continuation).
- The format of each entry is:
  name1|alias1|...|aliasN:cap1:cap2:cap3#num:...
- "cap" is a capability flag or numeric capability
  name.

Example Termcap Entry

- The Lear Siegler ADM3 terminal (obsolete now) entry looks
  like this:
  adm3|lsi adm3:
  :am:bs:
  :co#80:li#24:
  adm3|lsi adm3: - the name and an alias for the terminal
  am – output to last col. wraps cursor to next line
  bs – backspace moves cursor left one column
  co#80:li#24 – 80 columns, 24 rows (lines)
  bl=^G – control-G sounds the bell
  etc… See “info termcap” for more details

Conditionally Setting TERM

- The “tset” command can be used to set the TERM
  entry to a default value, or to another user-
  specified value.
- Example:
  $ TERM=`tset –Q –m ":?vt100"`
  $ export TERM
- In this example, tset sets the terminal type to
  vt100 if the user doesn't specify another setting.
  This is normally include in the .bash_profile (or
  other) initialization file.

Cables

- Luckily, prepared serial cables are available for
  most serial devices (terminals, modems, printers,
  etc.).
- However, you must get the right cable with the
  right connectors, or else communication with the
  device will fail – miserably!
- Most of the time you'll want a DCE to DTE cable,
  but a "null modem" is a DTE to DTE cable that
  allows a connection between two computers.
**Adaptors**

- Also useful are various cable and connector adapters.
- These convert, for example, from DB-9 to DB-25, or from a female connector to a male (or vice versa) – so called “gender changers”.
- Null modems are also handy, and are used to connecting two similar devices (usually DTE to DTE).

**Connectors**

- There are several types of connectors used on serial devices:
  - DB25 – a 25-pin connector, still in common use
  - DB9 – a 9-pin connector, now used on most newer devices and found on PCs
  - mini-DIN – an 8-pin connector common on older Macs
  - RJ-12 – a modular connector (similar to those used on phones or Ethernet cables) used in “modular” systems.
- DB25, DB9, and mini-DIN connectors also have gender: male connectors have pins, female connectors have sockets.

**Breakout Boxes**

- An invaluable aid in diagnosing and solving serial communication problems is a breakout box.
- This is a device that has one male and one female connector (DB-25 or DB-9 are common), LEDs to show various signal states, and jumpers and switches that allow various cabling connections or signal levels to be altered.