The RS-232-C interface was developed for a single purpose, unambiguously stated by its title:

"Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange."

Every word in the title is significant: it describes the interface between a terminal (DTE) to a modem (DCE) for the transfer of serial data.

This document consists of the following issues:

Data Transfer Modes

Logical data in microcomputers is represented as bits (binary digits). Bits are customarily explained through tables that illustrate each bit's contribution to some overall logical scheme. Although the bit is an intellectual construction, it is, nevertheless, physically a voltage whose magnitude gives the bit his value (i.e. 1 or 0).

When bits must be moved about within the computer itself, they are transmitted along wires. If the data to be transmitted is in 8-bits format bytes, then eight separate, discrete wires must simultaneously carry the eight representative electrical voltages between the two points. This simultaneous transmission of the eight bit-voltages that constitute a byte is referred to as ***"parallel transfer"***. Parallel transfer, then, is done byte-by-byte. Since all eight bits arrive at their destination at the same instant, parallel data transfer can be accomplished at extremely high speeds. These qualities make it the preferred method of data transfer whenever possible.

Data transfer, especially high-speed data transfer, demands a tightly controlled environment. The internal temperature of the computer must be regulated and the electrical properties of resistance, capacitance, and inductance carefully pre-calculated. As long as data is being moved about inside a computer, this environment is stable and predictable. But a great deal of computer data must be transported to the outside world. Microcomputers communicate with peripheral devices such as printers, terminals, modems, print buffers, etc. These processes are known collectively as input/output, or simply I/O.

The Interface

An interface is the point of contact between dissimilar environments; between the computer's circuitry and external devices. Since an interface is a sort of "door" to the computer's world, it is sometimes called an ***I/O port,*** or just a port.

The primary objective of any interface is to provide a medium for the transfer of data. Further more, self-protection and usability are also important goals for any interface. Once such an interface has been established, the transfer of data to external environments is possible.

When considering parallel transfer for the interface, two major problems arise. The first is the wire itself. At least nine wires - eight for the data bits, one for circuit common ("ground") - are needed. Still more wires are usually required to control the flow of data across the interface. Another problem lies in the very nature of the bits/voltages themselves. When a bit/voltage changes state from a one to a zero, or vice versa, it does so very rapidly -in the order of nanoseconds (one billionth of a second).This abruptness is itself an essential part of the process of data transfer. Slow changes between zero and one are not even recognized as data. As a cable gets longer, its electrical properties (capacitance & inductance) restrict the abruptness with which a bit can change between zero and one, and data corruption or loss becomes likely. Because of this, the speed inherent in parallel data transfers makes transmission over long cables problematic.

Therefore, its use is restricted to a few peripheral devices (such as printers) that are likely to be used in close proximity to the computer, or that must be operating at very high speeds.

The obvious alternative to sending all bits simultaneously on multiple wires is to send them singly, one after the other. At the receiving end, the process is reversed and the individual bits are reassembled into the original byte. With just one bit to transmit at a time, data can be transferred with a simple electrical circuit consisting of only two wires. This scheme - known as ***"Serial Transfer"*** - reduces the bulk and much of the expense of the parallel technique.

This saving is offset by a decrease in efficiency: it takes at least eight times longer to transmit eight individual bits one after the other than to transmit them all simultaneously in parallel. This speed limit is insignificant for many typical applications. Serial peripheral devices are slow, at least in comparison to the internal speed of microprocessors. Each involves some time-consuming, sometimes mechanical process that greatly limits its speed: printers are limited by the speed of their print-heads, modems by the frequency restrictions of the telephone lines, and disk drives by their slow rotational speed. So the speed inherent in the process of parallel data transfer is largely wasted on such peripheral devices. The serial method, therefore, can afford to sacrifice some speed while still adequately servicing the peripheral devices. In such cases, the sacrifice in speed is inconsequential in comparison to the increased reliability and transmission range.

Standard Interfaces

There are always several ways to design any circuit "correctly", any number of perfectly functional interfaces for an application are possible. In this diversity lies a problem fundamental to all interface circuitry: compatibility with other interfaces.

In the late 1960's a need surfaced for remote access to mainframe computers. It became desirable for the end-users to access computers from remote locations. Short distances- a few hundred feet, perhaps within the same building- could be spanned by the addition of

extra wires. For truly distant remote access, telephone lines were considered. For many reasons computer data cannot be injected directly into the telephone network. A translating device - the Modem - is required.

When computerized telecommunications was in its infancy, the Bell System supplied most of the data equipment to its lines. Bell naturally exercised strict control over the modem interface. But as activity in the telecommunications field increased, and more and different kinds of equipment began to appear, Bell surveyed the hodgepodge of equipment that the computer industry was threatening to connect to its lines. It saw little that it liked and much that it felt would compromise and complicate the delivery of communications service to the public. The telephone companies predictably prohibited the connection of most of these devices.

Interfacing Basics

In its simplest form, the RS-232-C interface consists of only two wires-one to carry data, plus a ***"circuit common"***. The circuit common is the absolute voltage reference for all the interface circuitry, the point in the circuit from which all voltages are measured.

A typical DTE device is an ordinary video terminal with a keyboard and a video display. Data on pin 2 of the DTE is transmitted, while the same data on pin 2 of a DCE (modem) is received data.

Bidirectional Data

Terminals and modems are not usually one-way devices- each may also perform the opposite function. For example, modems usually fetch characters from the telephone line and output them to the terminal. Similarity, the terminal receives the characters output from the modem and displays them on the video screen. Bidirectional interchange between the two devices is directly analogous to the connection of two telephones. The differences between the DTE and DCE is: DTEs transmit on pin 2 and receive on pin 3. DCEs transmit on pin 3 and receive on pin 2.

Handshaking

There remains only the straightforward matter of interactive device control, i.e. ***handshaking***. Handshaking is the way in which the data flow across the interface is regulated and controlled. Two distinct kinds of handshaking are described in Software Handshaking and Hardware Handshaking.

An important distinction between the kinds of signals of the interface is between ***data signals*** and ***control signals***. Data signals are simply the pins which actually transmit and receive the characters, while control signals are everything else. If a modem can automatically answer the telephone, for example, it must be able to *report an incoming*

call to the computer and not start transferring data to the computer without first *receiving an "OK, I'm ready to receive now" confirmation* from the computer.

There are generally two or three such inquire-confirm pairs on an interface that allow one device to "talk" to the other. There is in practice no guarantee that a modem and/or terminal will implement any or all of these handshaking features. The manufacturers of equipment may arbitrarily decide to apply some of the standard handshaking, no handshaking at all, or to invent a scheme of their own.

The RS-232-C Interface Standard

RS-232-C interface was developed for a single purpose, unambiguously stated by its title:

"Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange."

The interface standard document consists of four parts:

- **Electrical Signal Characteristics**
- **Interface Mechanical Characteristics (Connectors)**
- **Functional Description of Interchange Circuits**
- **Standard Interfaces for Selected Communications System Configurations**

RS-232-C equipment "Compatibility"

While some of the signals on the RS-232-C interface are implemented almost universally on microcomputers, others are applied liberally without regard to any established practice. What can be expected from any device claiming to be "RS-232-C compatible"?

Areas of RS-232-C Compatibility:

- *The prescribed electrical characteristics (voltage, etc) of the interface are, by necessity, closely observed.* If a device claims to be "RS-232-C compatible" it means that you can connect it to another such "compatible" device without damaging either. This guarantees that they will match well enough electrically to permit the exchange of data.
- *The voltage levels assigned for zero and one will correspond to those described in the standard.*
- *A few pins on the connector are absolutely predictable:* * pin 2 & pin 3 are transmitted/received data * pin 7 is Circuit Common.
- *A terminal is a DTE.* When the standard was written, terminals were usually printing terminals; there were no video displays like those in use today. Instead, the computer responded to all commands by printing them. Printer interfaces therefore are traditionally configured DTE.

- *A modem is a DCE.* Because the RS-232-C standard was intended to standardize this interface, modems are nearly always DCE; however a few modem manufacturers - mindful that computer manufacturers can't decide if their serial ports should be DTE or DCE - have begun to include switches inside their equipment to permit the user to rearrange the traditional DCE pin assignments to DTE. Thus, even the holy distinction that the modem is, by definition, Data Communication Equipment, is beginning to blur.

HISTORIC OVERVIEW

The RS-232-C was originally set to standardize the interconnections of terminals and host computers through public telephone networks. Modems were used to translate the digital data signals from the computer equipment to analog audio signals suitable for transmission on the telephone network, and back to digital signals at the receiving end.

In the mid- to late 1960's, nearly all serial links for remote access to computers were through a telephone line. Remote access to the large mainframes of the time was accomplished almost exclusively by using the telephone network.

At that time, each manufacturer of equipment used a different configuration for interfacing a DTE (Data Terminal Equipment) with a DCE (Data Communications Equipment). Cables, connectors and voltage levels were different and incompatible, thus the interconnection of two pieces of equipment made by two different companies required the use of voltage level converters, and the manufacturing of special cables and connectors.

In 1969, EIA with Bell Laboratories and other parties established a recommended standard for interfacing terminals and data communications equipment. The object of this standard was to simplify the interconnection of equipment manufactured by different firms.

The standard defines electrical, mechanical, and functional characteristics. The electrical characteristics include parameters such as voltage levels and cable impedance. The mechanical section describes the pin number assignments and plug. The connector itself, however, is not specified. The functional description defines the functions of the different electrical signals to be used.

This standard shortly became RS-232-C (Recommended Standard number 232, revision C from the Electronic Industry Association), and a similar standard was available in Europe, developed by the CCITT (Comite Consultatif Internatinale de Telegraphie et Telephonie), and known as V.24 (functional description) and V.28 (electrical specifications). RS-232-C was widely adopted by manufacturers of terminals and computer equipment.

In the 1980's, the rapidly growing microcomputer industry found the RS-232-C standard cheap (compared to parallel connections) and suitable for connecting peripheral

equipment to microcomputers. RS-232-C quickly became a standard for connecting microcomputers to printers, plotters, backup tape devices, terminals, programmed equipment and other microcomputers.

Since the standard only supported transfer rates up to 20 kbps (Kilobits per second), and distances of up to around 16 meters, new standards were adopted by EIA. The RS449 (mechanical) and RS423 (electrical) is upward-compatible with RS-232-C and can operate at data rates up to 10Mbps and distances of up to 1200 meters. Changing to a new standard, though, is a costly and long process. The RS-232-C is so widely available that it is certain to stay with us for some time to come.

RS-232 Signals Functional Description

General: The first letter of the EIA signal name categorizes the signal into one of five groups, each representing a different "circuit":

- *A - Ground*
- *B - Data*
- *C - Control*
- *D - Timing*
- *S - Secondary channel*

- **1 Protective Ground**
  - ***Name: AA***
  - ***Direction: -***
  - ***CCITT: 101***

  This pin is usually connected to the frame of one of the devices, either the DCE or the DTE, which is properly grounded. The sole purpose of this connection is to protect against accidental electric shock and usually this pin should not be tied to Signal Ground.

  This pin should connect the chassis (shields) of the two devices, but this connection is made only when connection of chassis grounds is safe (see ground loops below) and it is considered optional.

  Ground loops are low impedance closed electric loops composed from ground conductors. When two grounded devices are connected together, say by a RS-232 cable, the alternating current on the lines in the cable induces an electric potential across the ends of the grounding line (either Protective Ground or Signal Ground), and an electric current will flow across this line and through the ground.

Since the loops impedance is low, this current can be quite high and easily burn out electric components. Electrical storms could also cause a burst of destructive current across such a loop. Therefore, connection of the Protective Ground pin is potentially hazardous. Furthermore, not all signal grounds are necessarily isolated from the chassis ground, and using a RS-232 interface, especially across a long distance, is unreliable and could be hazardous. 30 meters is considered the maximum distance at which the grounding signals can be connected safely.

- **2 Transmit Data**
  - ***Name: BA***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 103***

Serial data (primary) is sent on this line from the DTE to the DCE. The DTE holds this line at logic 1 when no data are being transmitted. A "On" (logic 0) condition must be present on the following signals, where implemented, before data can be transmitted on this line : CA, CB,CC and CD (Request To Send, Clear To Send,Data Set Ready, Data Terminal Ready).

- **3 Receive Data**
  - ***Name: BB***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 104***

Serial data (primary) is sent on this line from the DCE to the DTE. This pin is held at logic 1 (Mark) when no data are being transmitted, and is held "Off" for a brief interval after an "On" to "Off" transition on the Request To Send line, in order to allow the transmission to complete.

- **4 Request To Send**
  - ***Name: CA***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 105***

Enables transmission circuits. The DTE uses this signal when it wants to transmit to the DCE. This signal, in combination with the Clear To Send signal, coordinates data transmission between the DTE and the DCE.

A logic 0 on this line keeps the DCE in transmit mode. The DCE will receive data from the DTE and transmit it on to the communication link.

The Request To Send and Clear To Send signals relate to a half- duplex telephone line. A half duplex line is capable of carrying signals on both directions but only one at a time. When the DTE has data to send, it raises Request To Send, and then waits until the DCE changes from receive to transmit mode. This "On" to "Off" transition instructs the DCE to move to "transmit" mode, and when a transmission is possible, the DCE sets Clear To Send and transmission can begin.

On a full duplex line, like a hard-wired connection, where transmission and reception can occur simultaneously, the Clear To Send and Request To Send signals are held to a constant "On" level.

An "On" to "Off" transition on this line instructs the DCE to complete the transmission of data that is in progress, and to move to a "receive" (or "no transmission") mode.

- **5 Clear To Send**
  - ***Name: CB***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 106***

An answer signal to the DTE. When this signal is active, it tells the DTE that it can now start transmitting (on Transmitted Data line). When this signal is "On" and the Request To Send,Data Set Ready, and Data Terminal Ready are all "On", the DTE is assured that its data will be sent to the communications link. When "Off", it is an indication to the DTE that the DCE is not ready, and therefore data should not be sent.

When the Data Set Ready and Data Terminal Ready signals are not implemented, in a local connection which does not involve the telephone network, the Clear To Send and Request To Send signals are sufficient to control data transmission.

- **6 Data Set Ready**
  - ***Name: CC***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 107***

On this line the DCE tells the DTE that the communication channel is available (i.e., in an automatic calling system, the DCE (modem) is not in the dial, test or talk modes and therefore is available for transmission and reception). It reflects

the status of the local data set, and does not indicate that an actual link has been established with any remote data equipment.

- **7 Signal Ground**
  - ***Name: AB***
  - ***Direction: -***
  - ***CCITT: 102***

  This pin is the reference ground for all the other signals, data and control.

- **8 Receive Line Signal Detect or Data Carrier Detect**
  - ***Name: CF***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 109***

  The DCE uses this line to signal the DTE that a good signal is being received (a "good signal" means a good analog carrier, which can ensure demodulation of received data).

- **9 +P**

  This pin is held at +12 volts DC for test purposes.

- **10 -P**

  This pin is held at -12 volts DC for test purposes.

- **12 Secondary Receive Line Signal Detect**
  - ***Name: SCF***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 122***

  This signal is active when the secondary communication channel is receiving a good analog carrier (same function as the Receive Line Signal Detect signal).

- **13 Secondary Clear To Send**
  - ***Name: SCB***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 121***

  An answer signal to the DTE. When this signal is active, it tells the DTE that it can now start transmitting on the secondary channel (on the Secondary Transmitted Data line).

- **14 Secondary Transmitted Data**
  - ***Name: SBA***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 118***

  Serial data (secondary channel) is sent on this line from the DTE to the DCE. This signal is equivalent to the Transmitted Data line except that it is used to transmit data on the secondary channel.

- **15 Transmission Signal Element Timing**
  - ***Name: DB***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 114***

  The DCE sends the DTE a clock signal on this line. This enables the DTE to clock its output circuitry which transmits serial data on the Transmitted Data line.

  The clock signal frequency is the same as the bit rate of the Transmitted Data line. An "On" to "Off" transition should mark the center of each signal element (bit) on the Transmitted Data line.

- **16 Secondary Receive Data**
  - ***Name: SBB***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 119***

  Serial data (secondary channel) is received on this line from the DCE to the DTE. When the secondary channel is being used only for diagnostic purposes or to interrupt the flow of data in the primary channel, this signal is normally not provided.

- **17 Receiver Signal Element Timing**
  - ***Name: DD***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 115***

The DCE sends the DTE a clock signal on this line. This clocks the reception circuitry of the DTE which receives serial data on the Received Data line.

The clock signal frequency is the same as the bit rate of the Received Data line (BB). The "On" to "Off" transition should indicate the center of each signal element (bit) on the Received Data line.

- **19 Secondary Request To Send**
  - ***Name: SCA***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 120***

The DTE uses this signal to request transmission from the DCE on the secondary channel. It is equivalent to the Request To Send signal.

When the secondary channel is only used for diagnostic purposes or to interrupt the flow of data in the primary channel, this signal should turn "On" the secondary channel un-modulated carrier.

- **20 Data Terminal Ready**
  - ***Name: CD***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 108.2***

When on, tells the DCE that the DTE is available for receiving. This signal must be "On" before the DCE can turn Data Set Ready "On", thereby indicating that it is connected to the communications link.

The Data Terminal Ready and Data Set Ready signals deal with the readiness of the equipment, as opposed to the Clear To Send and Request To Send signals that deal with the readiness of the communication channel.

When "Off", it causes the DCE to finish any transmission in progress and to be removed from the communication channel.

- **21 Signal Quality Detector**
  - ***Name: CG***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 110***

  This line is used by the DCE to indicate whether or not there is a high probability of an error in the received data. When there is a high probability of an error, it is set to "Off", and is "On" at all other times.

- **22 Ring Indicator**
  - ***Name: CE***
  - ***Direction: DTE <- DCE***
  - ***CCITT: 125***

  On this line the DCE signals the DTE that there is an incoming call. This signal is maintained "Off" at all times except when the DCE receives a ringing signal.

- **23 Data Signal Rate Selector**
  - ***Name: CH/CI***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 111/112***

  The DTE uses this line to select the transmission bit rate of the DCE. The selection is between two rates in the case of a dual rate synchronous connection, or between two ranges of data rates in the case of an asynchronous connection.

  Typically, when this signal is "On", it tells the DCE (modem) that the receive speed is greater than 600 baud.

- **24 Transmitter Signal Element Timing**
  - ***Name: DA***
  - ***Direction: DTE -> DCE***
  - ***CCITT: 113***

  The DTE sends the DCE a transmit clock on this line. This is only when the master clock is in the DTE.

An "On" to "Off" transition should indicate the center of each signal element (bit) on the Transmitted Data line.

- ### A note on signal travel direction

  The pin names are the same for the DCE and DTE. The Transmit Data (pin number 2) is a transmit line on the DTE and a receive line on the DCE, Data Set Ready (pin number 6) is a receive line on the DTE and a transmit line on the DCE, and so forth.

- ## Electrical Signal Characteristics

  - ### Voltage levels defined in the standard

Data signals	"0","Space"	"1","Mark"	
Driver (Required)	5 - 15	-5 - -15	Volts
Terminator (expected)	3 - 25	-3 - -25	Volts
Control signals	"Off"	"On"	
Driver (Required)	-5 - -15	5 - 15	Volts
Terminator (expected)	-3 - -25	3 - 25	Volts

- ### The Noise Margin Issue

  Note that terminator (receiving end) voltages are not the same as driver required voltages. This voltage level definition compensates for voltage losses across the cable.

  Signals traveling along the cable are attenuated and distorted as they pass. Attenuation increases as the length of the cable increases. This effect is largely due to the electrical capacitance of the cable.

  The maximum load capacitance is specified as 2500pf (Pico farad) by the standard. The capacitance of one meter of cable is typically around 130pf, thus the maximum cable length is limited to around 17 meters. However, This is a nominal length defined by the standard, and it is possible to use longer cables up

to 30 meters, with low-capacitance cables, or with slow data rates and a proper error correction mechanism.

- ## Interface Mechanical Characteristics

  The connection of the DCE and the DTE is done with a pluggable connector. The female connector should be associated with the DCE. The following table lists the pin assignments defined by the standard. The type of connector to be used is not mentioned in the standard, but the DB-25 (or on IBM-AT's, a minimal DB-9) connectors are almost always used.

- ## Pin designation for the 25-pin and 9-pin DB connector

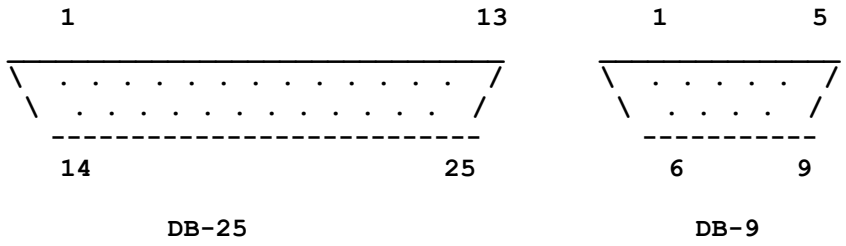
  includes equivalent CCITT V.24 identification, and signal direction

DB-25 Pin #	DB-9 Pin #	Common Name	EIA Name	CCITT	DTE-DCE	Formal Name
1		FG	AA	101	-	Frame Ground
2	3	TD	BA	103	--->	Transmitted Data, TxD
3	2	RD	BB	104	<---	Received Data, RxD
4	7	RTS	CA	105	--->	Request To Send
5	8	CTS	CB	106	<---	Clear To Send
6	6	DSR	CC	107	<---	Data Set Ready
7	5	SG	AB	102	----	Signal Ground, GND
8	1	DCD	CF	109	<---	Data Carrier Detect
9		--	--	-	-	+P
10		--	--	-	-	-P
11		--	--	-	-	unassigned
12		SDCD	SCF	122	<---	Secondary Data Carrier Detect
13		SCTS	SCB	121	<---	Secondary Clear To Send
14		STD	SBA	118	--->	Secondary Transmitted Data
15		TC	DB	114	<---	Transmission Signal Element Timing
16		SRD	SBB	119	<---	Secondary Received Data
17		RC	DD	115	--->	Receiver Signal Element Timing
18		--	--	-	-	unassigned
19		SRTS	SCA	120	--->	Secondary Request To Send
20	4	DTR	CD	108.2	--->	Data Terminal Ready

```
  21            SQ      CG      110   <---    Signal Quality
Detector
  22     9      RI      CE      125   <---    Ring Indicator
  23            --      CH/CI 111/112 --->    Data Signal Rate
Selector
  24            --      DA      113   <---    Transmitter Signal
Element Timing
  25            --      --       -      -     unassigned
```

- **Diagram of the DB-25 and DB-9 connectors**

  male connectors , front view



Introduction

The EIA standard has left some unspecified areas regarding what constitutes a compliant cable-connector implementation. One area is specifying the connector itself, while another area is defining for the 21 circuits in the standard which is optional and which is required. This was done on purpose, since this standard cable was intended for simple local terminal interface through a multiplexed, synchronous, dedicated line that is shared by a cluster of remote terminals and is equipped with automatic dialing units.

Let's look at some sample configurations:

1. Transmit only
2. Transmit only with RTS
3. Receive only
4. Half Duplex
5. Full Duplex
6. Full Duplex With RTS
7. Special

RS-232-C standard configuration

```
RS-232-C interchange circuit    (1)    (2)   (3)   (4)   (5)   (6)  (7)
-------------------------------------------------------------------------
```

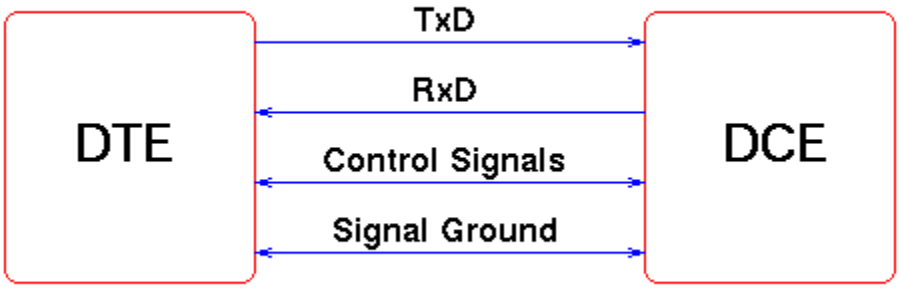
1	Protective Ground	-	-	-	-	-	-	o
7	Signal Ground	X	X	X	X	X	X	X
2	Transmitted Data	X	X		X	X	X	o
3	Received Data			X	X	X	X	o
4	Request to Send		X		X		X	o
5	Clear to Send	X	X		X	X	X	o
6	Data Set Ready	X	X	X	X	X	X	o
20	Data Terminal Ready	S	S	S	S	S	S	o
22	Ring Indicator	S	S	S	S	S	S	o
8	Received Line Signal Detector			X	X	X	X	o

X = required for any configuration
S = required for using PSDN (public switched telephone network)
o = specified by cable designer

Notice that only *one* circuit is *absolute requirement* for any such cable, this is the Signal Ground on Pin 7. That means that as long as Signal Ground is included on pin 7 this cable is **"RS-232-C compliant"**".

Although microcomputer system have one-way devices such as transmit-only joysticks, or receive only printers, the most common situation is a full-duplex, two-way communication. The classical application for the above is a send/receive terminal where characters are transmitted from the keyboard to a microcomputer and echoed back to a display screen. The data is traveling in both directions from the DTE (keyboard and screen), to the DCE (computer serial I/O port). There are two configurations for full-duplex, one with and one without the Request to Send line implemented. A "safe" strategy is to always include it.

The figure below shows a schematic full-duplex configuration



RS-232-C Standard Full Duplex Cable

- The Signals present in a standard full-duplex RS-232-C cable are :
  - Signal Ground
  - Transmitted Data
  - Received Data
  - Request to Send
  - Clear to Send
  - Data Set Ready
  - Received Line Signal Detector
- For a modem over a switched telephone network we must add two more signals
  - Data Terminal Ready
  - Ring Indicator

To understand the function of each signal lets look at the events taking place to eventually exchange data. Each event will cause a transition from state to state from the idle state through the data exchange and communicating state to finally back to idle state.

The events and transitions are grouped in phases:

- Alerting
- Equipment Readiness
- Circuit Assurance
- Channel Readiness

Alerting

In the Alerting phase, the call originating station dials the phone number of the call-answering station. The remote telephone begins to ring, and the Ring Indicator Signal of the answering DCE make an OFF to ON transition. This ends this phase.

Equipment Readiness

The Equipment readiness phase is involving events associated with turning Data Set Ready and Data Terminal Ready to ON, and thus completing this phase.

Circuit Assurance

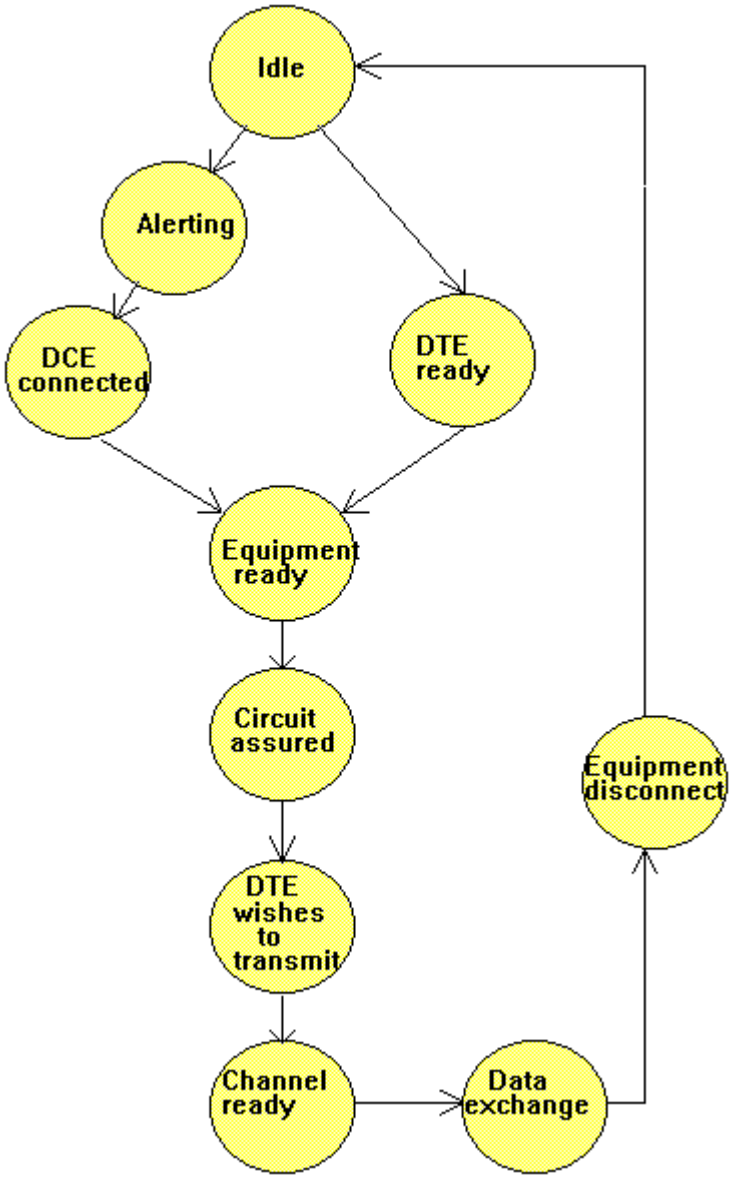
The Circuit Assurance consists of events associated with turning ON the Received Line Signal Detector signals at both communicating stations.

Channel Readiness

The Channel Readiness Phase uses Request to Send and Clear to Send handshaking to arrive at the target state of active data-exchange state. All these events result in the

equipment being disconnected from the telephone network and getting back to original idle state.

The following diagram illustrates the above:

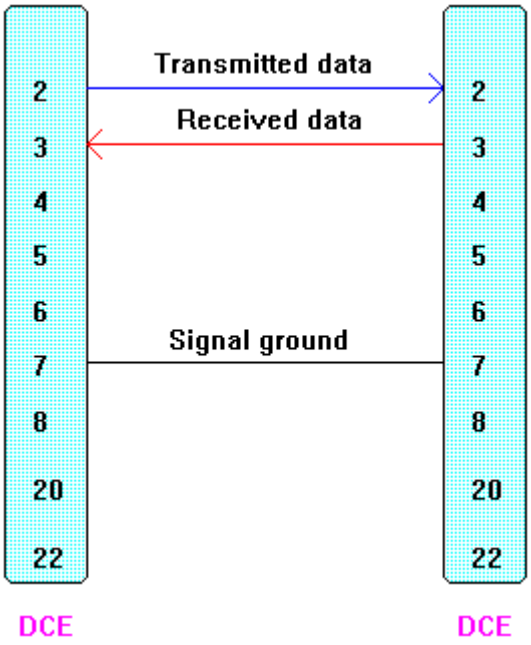


Other (Non Standard) Common Configurations

Three-Wire Economy Model

This model involves only minimum number of circuits for full-duplex communications. The circuits present are Transmitted Data on pin 2, Received Data on pin 3, and Signal Ground on pin 7. There are configurations for which this cable is entirely adequate, but many common Microsystems components use the RTS and CTS circuits. This equipment will not transmit, until it received an asserted CTS signal. For this we use the next model to trick the USART-based I/O ports into transmission.

The following diagram illustrates the above:



Three-Wire with Luxury Loop-Back

This cable has the following loop-back jumpers:

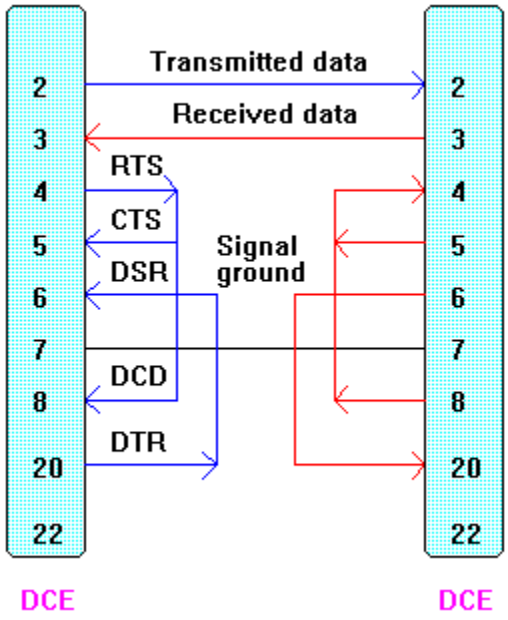
- Request to Send -> (jumpered to) -> Clear to Send
- Request to Send -> (jumpered to) -> Received Line Signal Detector
- Data Terminal Ready -> (jumpered to) -> Data Set Ready

By jumpering Data Set Ready, the Equipment Readiness phase is completed as soon as the DTE asserts its Data Terminal Ready line. This is achieved when the DTE is powered-up. Also, when the DTE is powered-up the Request to Send is asserted and the Circuit Assurance phase is completed, since the Request to Send is jumpered to the

Received Line Signal Detector. Since Request to Send is jumpered to Clear to send, it is also implies the completion of the channel readiness phase. The bottom line is that Data Terminal Ready and Request to Send are the only two events required to achieve the target data-exchange state.

Notice that this implementation omits some features from the full implementation. Most of them concerning preventing overrun errors.

The following diagram illustrates the above:



Null Modem with Luxury Loop-Back and the Null Modem with Double-Cross

This model is designed to answer the requirement to trick two DTEs into communicating over a strictly *local* RS-232-C interface, with no modems or DCEs. This concept is identified as the **"crossover technique"** The word *luxury* is used since there are some modest models in which some loop-backs are omitted. When none of the loop-backs is present it is simply the three-wire economy model. Notice that this model is exactly the three-wire with luxury loop-back model with null modem.

The double Cross variant involves a crossover between the following two pairs of control signals:

- Request to Send <-> Received Line Signal Detector
- Data Terminal Ready <-> Ring Indicator

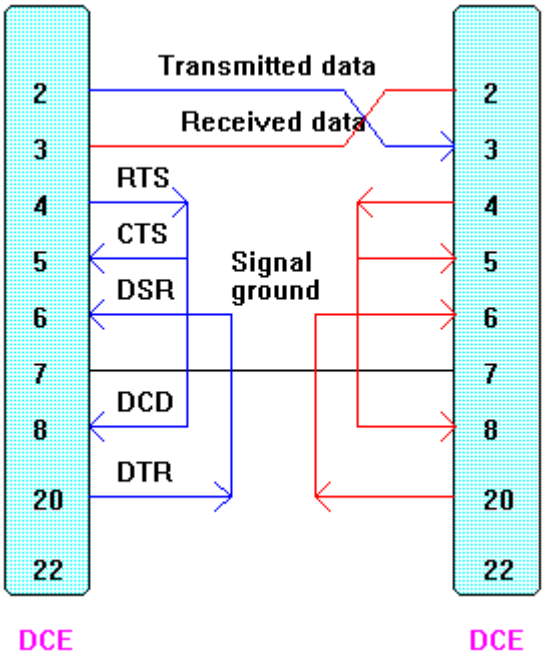
The following loop-back is included:

- Request to Send -> (jumpered to) -> Clear to Send
- Ring Indicator -> (jumpered to) -> Data Set Ready

And finally a crossover for null-modem function:

- Transmitted Data <-> Received Data

The following diagram illustrates the null modem with luxury loop-back:



EXAMPLES

DTE connected to DCE

When a PC wants to send data it sets the data terminal ready line. This DTR signal goes into the DTR line of the DCE. The DCE recognizes that the DTE is requesting a connection. if an open phone line exist for the DCE, it sets the DSR and data carried detected. When the PC sees the 2 signals on its DCD and DSR input lines, it sets the request to send line - which says that the PC has data to send to the DCE. if the DCE is clear to accept data, it sets the clear to send line - which tells the PC that the DCE is free to receive, and the PC begins transmitting data over the TxD line where it is received on the corresponding line on the DCE.

The problem with the standard is that it assumes that once the communication link was established, it would not be broken and that there was no speed mismatch with the two partners. It is true when working with terminals - but not with computers. Since a printer

is much slower that the PC, we have a problem here and handshake would solve it. Since it is not standard - it won't succeed.

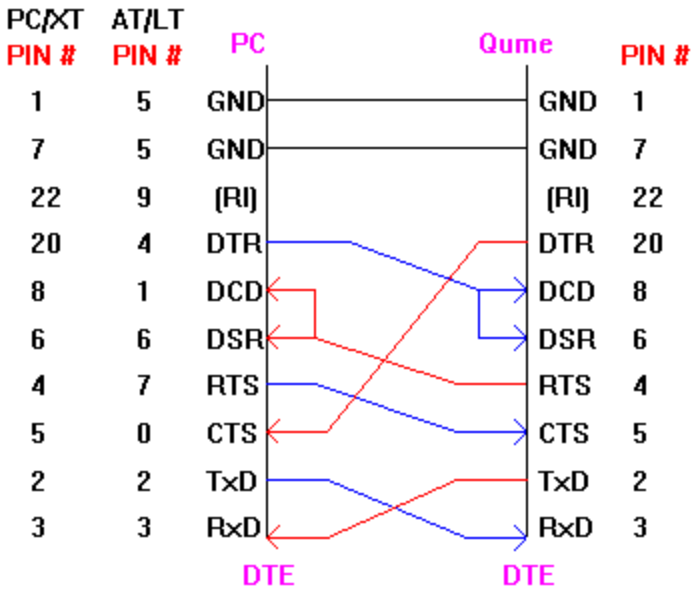
DTE connected to DTE

Here we have a PC connected to Qume Sprint serial printer. The printer is much slower than the PC and the printer manufacturer implemented a hardware handshake. When the printer's buffer becomes nearly full, its lowers the DTR line - tells that it is not ready. When the PC recognizes this, it stops sending until the buffer is empty and the DTR is set again.

At first we match the send lines with the receive lines on the other device by connecting the RxD on one device with the TxD wires on the other. (This is the minimum request for DTE-DTE).

The PC and the printer come up with their DTR lines set. Consider the signals as seen by the printers. The PC's DTR line satisfies the printer's DCD and DSR input line requirements. The printer responds by setting RTS, which in turn satisfies the PC's need for a DCD and DSR signal. The PC also requires its CTS send line to be set before it can transmit, but notice that the printer's DTR is wired to the PC's CTS, so this requirement is satisfied. The PC can now transmit and receive and so is the printer. All the standard's rules are satisfied except the order in which the PC's CTS input line was set, but this makes no difference in its function.

The following diagram illustrates the above:



RS-422, RS-423, and RS-449 - A Compatible Improvement of RS-232-C

In 1973 the RS-232-C standard needed a major revision because of the accelerating rate of technological change. Three new standards incorporate new interface technology designed to overcome the shortcoming of RS-232-C.

The old standard had some problems:

- The data transmission rate was limited to 20 kbps.
- The distance for transmission is limited to 50 ft.
- The standard does not specify a connector, which led to some 25-pin designs not compatible with each other.
- Only one conductor per circuit is used, with only one signal return for both directions of transmission.
- The interface uses unbalanced transmitters and receivers (unbalanced circuit is less desirable (relative to performance) than a balanced circuit).
- The interface can generate considerable crosstalk among its component signals.
- The overall interface design is for discrete component technology.

The new standards had to achieve the following goals:

- Maintain compatibility with the old RS-232-C.
- Support a higher signaling rate bandwidth over longer distances than the RS-232-C.
- Add interface circuit to perform such functions as loop-back testing.
- Resolve the mechanical interface problems that were caused from the lack of connector specifications in the old standard.
- Improve the electrical characteristics of the interface by providing for balanced circuits.

The RS-422 standard defines a double-ended electrical interface module that can signal at rates well in excess of the 20 kbps limitation of RS-232-C. The mechanical connections for this interface are provided by the RS-449 and covered in its standard.

The single-ended electrical standard RS-423 is intended to reach the compatibility with RS-232-C, while simultaneously conforming to both RS-422-type electrical conventions and the RS-449 mechanical standard. The RS-423 is designed to connect to RS-232-C and RS-422 and its electrical specification is almost identical to RS-232-C. The RS-422 has balanced transmission and it is more reliable and has high-speed signaling rates.

TEST MODE is a new signal for RS-449, which in conjunction with other optional signals provides a means for testing the communications equipment.

From the standpoint of performance, the new standards are a distinct improvement, but considerations of cost and convenience, and general resistance of the user community to change, mean that some time will pass before theses standards become widespread.