Chapter 5

Serial Data Communications
- Hardware Application

A Summary...

Asynchronous communications techniques. The RS-232C System. Definitions and shortcomings. How to physically link devices on the RS-232C standard. Extending the horizons of serial communications through optic fibres. The balanced electrical circuit of RS-449. The 20mA Current Loop System. Cable considerations and precautions.

Read This Chapter If...

- You want to learn about point to point asynchronous serial links and how to apply them in practice
- You want to learn how to link RS-232C devices
- You think you already understand the RS-232C system!

5.1 The RS-232C Standard

It is somewhat ironic to use the word "standard" in conjunction with this section heading. We will shortly discover that in practice, there is little that remains as standard in the common application of the Electronic Industries Association (EIA) RS-232C specification for asynchronous serial communications links.

The RS-232C (commonly abbreviated RS-232) standard is also identified as one of the "V" series of specifications from the Commité Consultatif Internationale de Telegraphie et Telephonie (CCITT). As a result of this, RS-232 is sometimes referred to as the V.24 standard. The system was originally introduced as a specification for the connection of Data Terminal Equipment (DTE) to a Post Telephone and Telecommunications (PTT) modem. This is shown schematically in Figure 5.1.

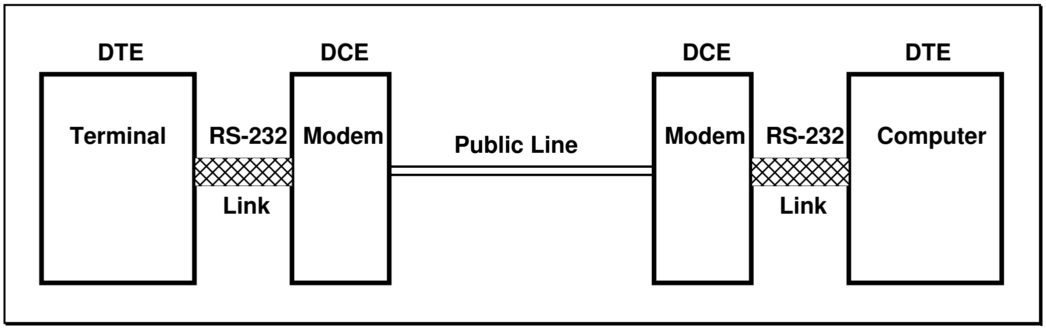


Figure 5.1 - Original Application of RS-232 C (V.24)

The RS-232 specification was designed to fulfil the needs of the environment shown in Figure 5.1. Its primary function application was in short-distance, low-bit-rate links between devices (computers and modems) in clean-room environments. The problems associated with RS-232 have come about largely because we have extended its use to a vast range of different applications, for which it was never intended.

The RS-232 port on each of the devices shown in Figure 5.1 is driven by a Universal Asynchronous Receiver Transmitter (UART). The UART performs conversion of outgoing data from parallel form to serial form and incoming data from serial to parallel form. In addition, it is also equipped with special inputs and outputs, which are used to co-ordinate the flow of data with an external device. These special purpose inputs and outputs are referred to as the hardware hand-shaking lines.

Figure 5.2 shows a schematic of the UARTs in DCE and DTE devices and the way in which data transfer and hand-shaking occurs in the RS-232 system. Hand shaking inputs and outputs in Figure 5.2 are signified by a circle containing the input or output number (on the UART). An inequality sign (< or >) inside the circle denotes the normal direction of information flow with respect to the local device.

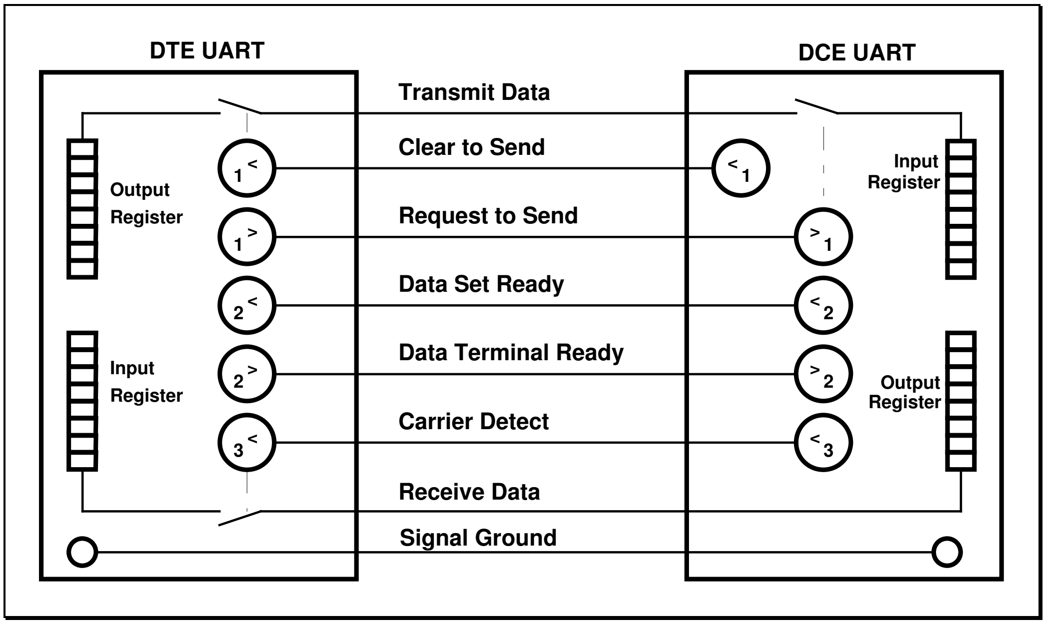


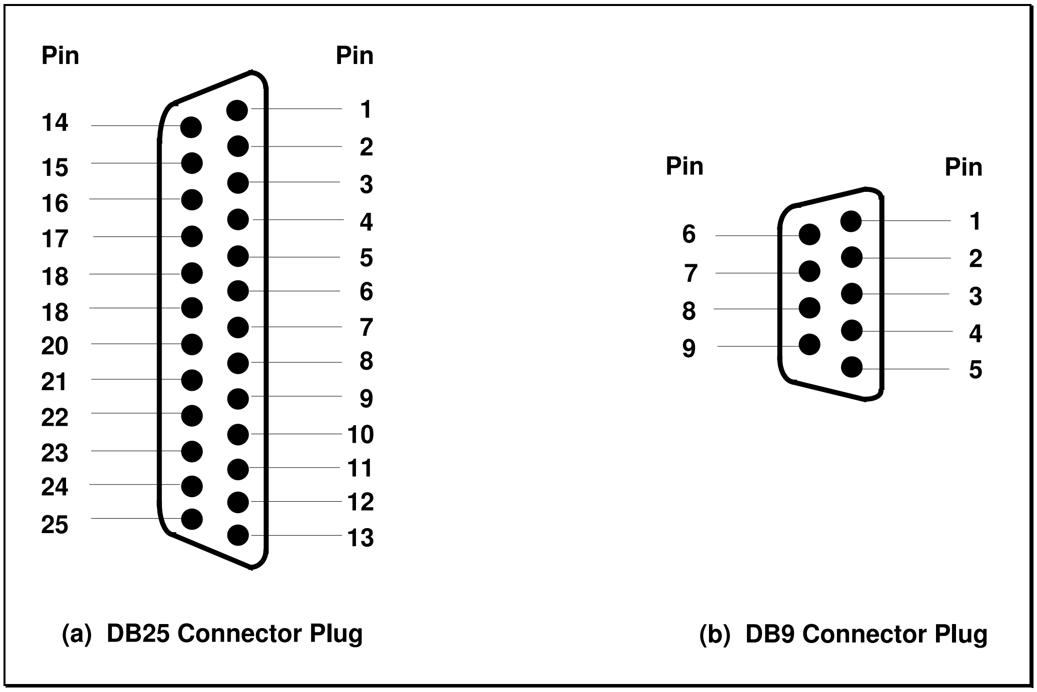
Figure 5.2 - Schematic showing DCE and DTE UARTs and Hardware Hand-Shaking on an RS-232C Link

In Figure 5.2, the "Request to Send", "Clear to Send" and "Carrier Detect" lines are directly responsible for switching data transmission on and off. The "Data Set Ready" and "Data Terminal Ready" lines are used so that DCE and DTE devices, respectively, can indicate whether they are powered up and ready for communication. We will look at these lines in more detail as we proceed through this chapter, but for the moment we only need observe the structure of the hand-shaking in relation to the UARTs.

We also need to recognise at this early stage that RS-232 is one of the most important of all the communications specifications associated with manufacturing, if for no other reason than because it is the most widely used and misunderstood system. For all the sophisticated communications networks that are available for manufacturing, few cause as much frustration and irritation as this one standard. Most computer controlled devices in manufacturing have, what are referred to, as "RS-232 compatible" communications ports. As we progress through this chapter, we shall see just how much (or how little) this actually means to the system designer.

5.2 The RS-232C Connectors

There are two, very common connectors (plugs) used for RS-232 communications. These are "D" shaped connectors that come in either a 25 pin (called DB25) or 9 pin (called DB9), male or female form. These are shown in Figure 5.3. As with most elements of the specification, "D" plug configurations are not universally and some equipment manufacturers choose to use a totally different connector. For example, round "DIN" plugs are sometimes used in an industrial environment because they are more rugged.



(a) DB25 Connector Plug

(b) DB9 Connector Plug

Figure 5.3 - Common RS-232 Connector Plugs

Table 5.1 is provided for reference and shows the side-by-side pin functionality for both the DB9 and DB25 RS-232 connectors. Again, these pin configurations are not universally adopted, so we need to check manufacturers' specifications for each application. We also note that the 9 pin connectors obviously cannot accommodate the same number of signal lines as the 25 pin connectors. Therefore how can the 9 pin connectors be used? As it happens, modern RS-232 applications require only the 9 pins of the DB9 connector in order to function. Nowadays, the extra pins on the DB25 connector are rarely used in equipment connections.

Function	***Abbreviation***	***Normal Source of Signal***	***DB25 Pin***	***DB9 Pin***
Protective Ground	GND		1	
Transmit Data	TXD	DTE	2	3
Receive Data	RXD	DCE	3	2
Request to Send	RTS	DTE	4	7
Clear to Send	CTS	DCE	5	8
Data Set Ready	DSR	DCE	6	6
Common Signal Ground	COM		7	5
Carrier Detect	CD	DCE	8	1
Negative Voltage Rail	-V		9	
Positive Voltage Rail	+V		10	
			11	
Secondary Received Line Signal Detector		DCE	12	
Secondary Clear to Send		DCE	13	
Secondary Transmitted Data		DTE	14	
DCE Transmitter Signal Element Timing		DCE	15	
Secondary Received Data		DCE	16	
Receiver Signal Element Timing		DCE	17	
			18	
Secondary Request to Send		DTE	19	
Data Terminal Ready	DTR	DTE	20	4
Signal Quality Detector	SQD	DCE	21	
Ring Indicator	RI	DCE	22	9
Data Signal Rate Selector	DSRS		23	
DTE Transmitter Signal Element Timing		DTE	24	
			25	

Table 5.1 - Pin Numbers and Functionality of RS-232C Plugs Based Upon DB25 or DB9 Connectors

When we talk of data communications, we sometimes refer to the gender of a device in terms of whether that device is Data Terminal Equipment (DTE) or Data Communications Equipment (DCE). In terms of the original intention of RS-232, these terms had a good deal of significance. Unfortunately, since the RS-232 system is now used in many different areas, the practical applications of the specification have muddled the original meanings of the gender. For this reason, some modern computerised devices turn out to be DCE whilst others are DTE, and still others can be switched from DCE to DTE. The only way to be certain is by electrical testing of the connectors, and we will look at techniques for this in subsequent sections.

When we talk of RS-232 communications, we also need to talk about the gender of hardware connectors and whether they are male or female. If the original RS-232C definitions were strictly adhered to, then all devices with male connectors could be identified as DTE and all devices with female connectors could be identified as DCE. However, this is yet another specification that is not upheld in practice. It is therefore possible for DTE devices to have female connectors and DCE devices to have male connectors.

5.3 Basic RS-232 Connections

For a few brief moments let us forget the ugly real world and concentrate on the all too rare situation, where two RS-232 devices are compatible and can be directly connected to one another. From this point onwards we shall only use the pin numbers as they are defined for a DB25 connector in Table 5.1. The RS-232 connection is shown in Figure 5.4, where a DTE device is connected to a DCE device through a 9 wire cable, with 25 pin connectors at either end. Note the terminology that has been used to define the direction of signals with respect to each device, the ">" and "<" symbols act as arrowheads indicating the direction of flow.

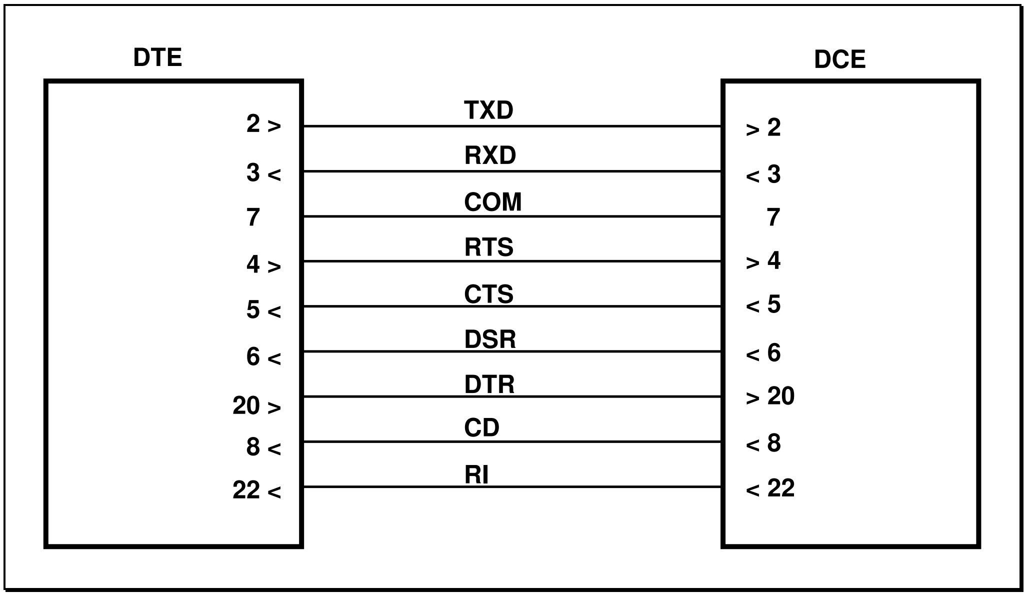


Figure 5.4 - Basic 9 Wire RS-232 Connection of DTE to DCE

It seems to go against the entire purpose of serial communications to use 9 lines to transfer data. In fact only three lines are essential for full-duplex RS-232 communications. These are the TXD, RXD and COM (Signal Ground) lines. The other hardware hand-shaking lines can be omitted by selective design in many applications.

Hardware hand-shaking on RS-232 links is extremely undesirable, purely because of the variations that exist in non-standard links (that is, when we use RS-232 for connecting devices other than computers and modems together). If one has influence over the design of a non-standard link then it should be avoided altogether. Unfortunately, understanding the hardware hand-shaking is a necessary evil because a large number of devices (particularly printers) already use it extensively and must be accommodated accordingly.

Before we proceed any further, we need to examine how binary data is represented on the RS-232 link. The voltage levels for RS-232 logic levels are shown in Figure 5.5. Note well the inverted logic and also the error margin between outputs and inputs. This margin is designed to cater for attenuation in the lines.

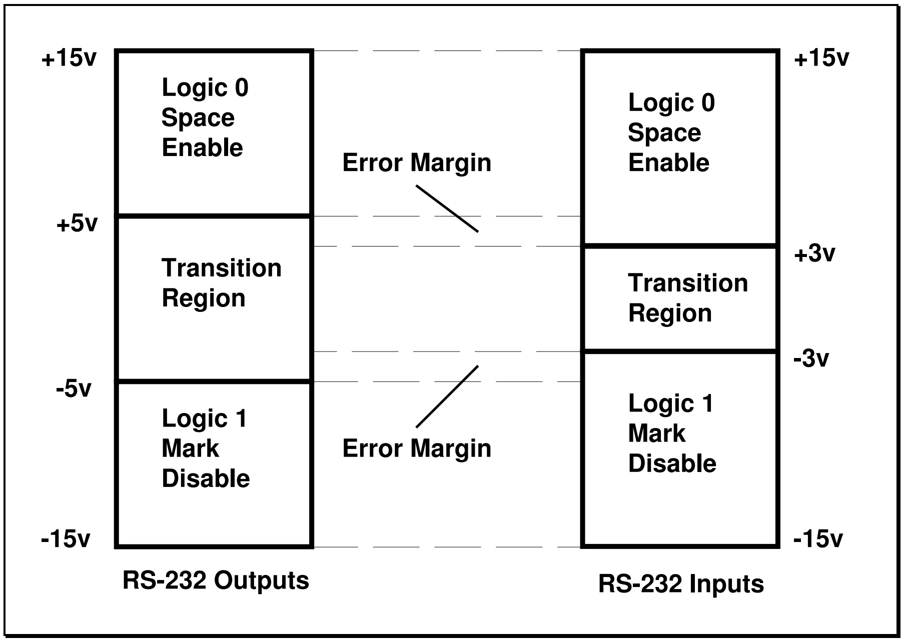


Figure 5.5 - Logic Voltage Levels for RS-232

Fortunately for users, the RS-232 voltage ranges shown in Figure 5.5 are generally adhered to in most implementations of the specification. Another feature of RS-232 that is universally adhered to, is that all the outputs and inputs are "electronically buffered". In other words, it is possible to short-circuit any two pins on an RS-232 port without causing damage to the circuitry. In light of the amount of experimentation that is often required to establish an RS-232 hardware link, this is an absolutely vital feature of the system.

Now that we have established the correlation between voltage levels and binary logic levels, we can examine what should theoretically happen, in terms of hardware hand-shaking, on an operational link. We use the model shown in Figure 5.6 as the basis of our examination and assume that the entire 9 wire connection is in place for the RS-232 link. As fate would have it, the link of Figure 5.6 can be used in either half-duplex or full-duplex mode and the hardware hand-shaking differs depending upon which mode is active. The explanation which follows attempts to define both the half and full-duplex hand-shaking arrangements.

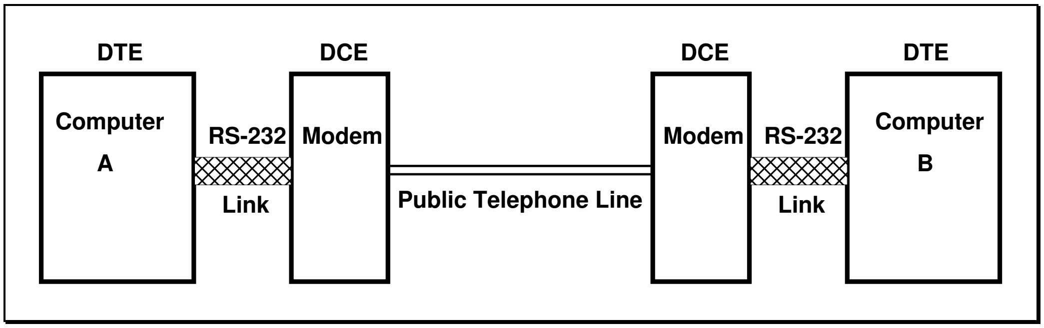


Figure 5.6 - Basic Model for RS-232 Hand shaking

Let us assume that Computer A wishes to access Computer B by dialling it on the telephone network. In the normal course of events, once Computer A dials, Computer B should respond with an answering tone, then a brief log-on message (prompt) to Computer A. The normal communications exchange then begins. The following hardware hand-shaking sequence occurs:

(a) If Computers A and B are both ready for communications then they each enable their Data Terminal Ready (DTR) lines.

(b) If Modems A and B are both ready for communications then they each enable their Data Set Ready (DSR) lines.

(c) After Computer A has dialled the number, Modem B enables its Ring Indicator (RI) to tell Computer B that it has been called.

(d) Computer B responds to the Ring Indicator by enabling its Request to Send (RTS) line.

(e) When Modem B receives an RTS from Computer B it transmits a carrier tone across the phone line to Modem A.

(f) When Modem A receives the carrier signal from Modem B, it enables its Carrier Detect (CD) line. In a half-duplex link, this tells Computer A that it should not send data because computer B wishes to transmit. In a full-duplex link, an enabled CD tells Computer A that the link is active.

(g) After a short delay (which gives the A side time to get ready to receive information) Modem B gives Computer B a Clear to Send (CTS) signal indicating that it should now proceed with data transmission.

(h) Computer B transmits its message to Modem B, which then modulates the binary information into data tones. In a half-duplex link, when Computer B has finished transmission it disables its RTS line. In a full duplex link both DTEs keep their RTS lines enabled.

(i) In a half-duplex link, when Modem B notes that the RTS line is disabled, it stops transmitting the carrier to Modem A. Modem A responds by disabling its Carrier Detect. In the full-duplex link the carrier always remains on.

(j) In the half-duplex arrangement, when Computer A notes that the Carrier Detect is disabled, it enables its Request to Send line in order to send a response message.

(k) After a short delay (which gives the B side time to get ready to receive information) Modem A responds to the Request to Send signal from Computer A with a Clear to Send signal. Computer A can then transmit its data.

This two-way process continues until neither of the devices has any more data to transmit. The process terminates and the devices disconnected from the phone lines. The entire process is shown schematically in Figure 5.7. Note the convention where enabled signals are qualified by a "↑" and disabled signals are qualified with a "↓".

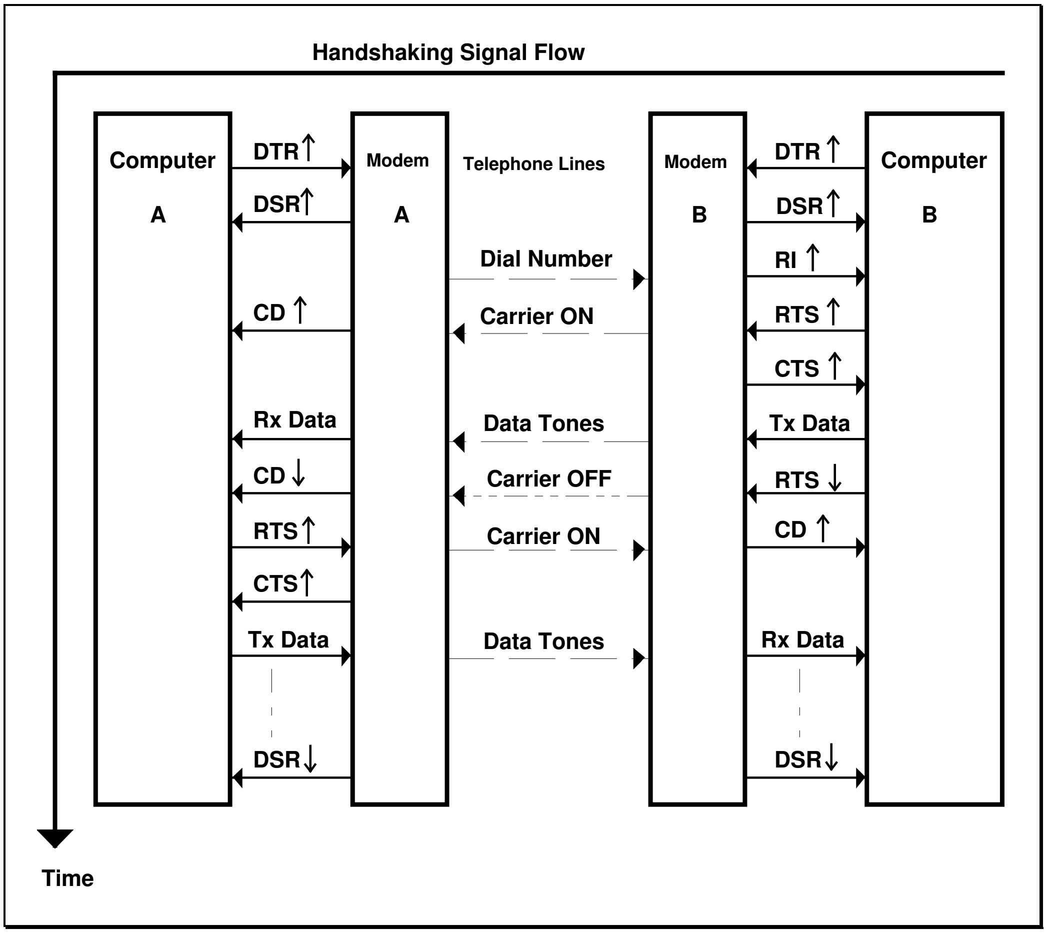


Figure 5.7 - Basic RS-232 Hardware Hand shaking Model for a Half-Duplex Communications Link

5.4 Complex RS-232 Connections

Following the discussions of section 5.3 a number of questions need to be raised:

- How do these computer to modem interfaces relate to other RS-232 implementations?
- Do all RS-232 ports respond with the same hand-shaking sequences noted in 5.3?
- How can we tell which device is DCE or DTE?
- What happens if we want to interface two DCE or two DTE devices to one another?

The answer to the first question is perhaps the simplest. RS-232 was designed for the environment and hand-shaking described in section 5.3. That is, for computer to modem links. However, because such a large proportion of computer equipment was fitted with an RS-232 port, countless other communications applications have since developed. These include computer to printer or plotter, computer to computer and computer to machine communications. Although all the RS-232 hand-shaking lines, described in section 5.3, are labelled in terms of their use with modems, there is no reason why they cannot be used for other applications. Unfortunately, as these new applications developed, the hand-shaking sequences, described in 5.3, were of less and less relevance and manufacturers strayed away from the original standard.

The net result is that we use RS-232 links for an enormous range of applications for which it was never intended. We are left with hand-shaking signals that have little or no relevance to their application and to make matters worse, equipment manufacturers vary the operation of hand-shaking lines to suit their own needs. So in answer to the second question we can say no - not all RS-232 ports respond to hand-shaking in exactly the same way.

We now move on to the problem of determining the gender (DCE or DTE) of a particular RS-232 device. We can, as a first step, examine the manufacturer's handbook and see what they believe the gender of their device to be. This is not always reliable, since some manufacturers do not appear to understand the definitions of DCE and DTE themselves. There is however, a simple series of tests that can be used to determine the gender of an RS-232 device. These require the use of any simple voltmeter.

When an RS-232 device is idle, its "output data" line (pin) will be marking time (binary 1 or negative voltage between -5v and -15 v). For a DTE, the output data line is labelled as the TRANSMIT DATA (TXD) line and the input data line is labelled as RECEIVE DATA (RXD). The labelling convention for data flow is relative to the DTE device. Therefore the output data line for a DCE is actually RXD and the input line is TXD.

By testing the voltage of both the Transmit and Receive pins on an RS-232 port, with respect to the signal ground (COM) pin, we can decide which of the two is marking time. The true "input data" line to the port will be floating. The true output data line will be marking time with a negative voltage.

It is important however, that before testing or using a connector, one is relatively certain that it is in fact RS-232. Connectors such as the DB25 and DB9 are not unique to RS-232 - they are just general purpose connectors. If one mistakenly assumes that a particular connector is RS-232 and then uses it without further testing, then a potentially damaging (or even dangerous) situation could arise. The following test steps can therefore be used to ascertain gender or to help us confirm that a device is RS-232 compatible:

(a) Check the manufacturer's description of the pin-configuration for the RS-232 port

(b) Ensure that the serial port is active (open) - some computers leave their serial ports inactive unless instructed otherwise by an operating system command

(c) Connect the positive side of a d.c. Voltmeter to what the manufacturer describes as the Transmit pin and negative side to the Signal Ground pin.

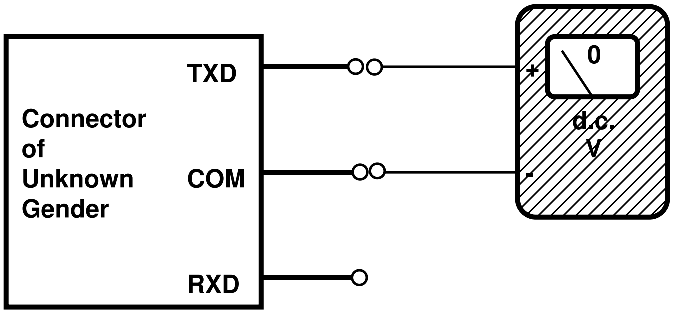
(d) If the voltage is negative (-5v to -15v) then the device is DTE.

(e) If the voltage is not negative then move the positive terminal of the Voltmeter to the pin described by the manufacturer as the Receive pin. If this voltage is negative (-5v to -15v) then the device is DCE.

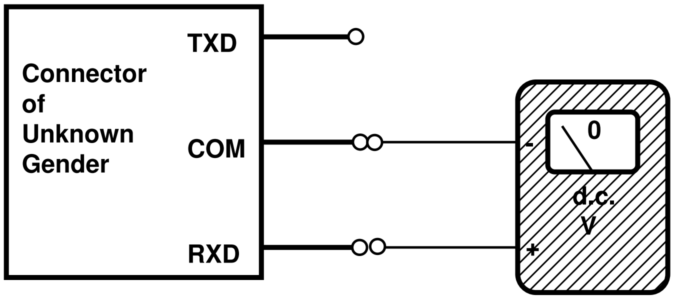
(f) If neither of the voltages are negative then there are a a number of possibilities (other than the rather obvious one of the port not working):

    - the serial port is not active (open)
    - the device is a receive only device
    - the port is not RS-232 compatible at all

This test procedure is shown schematically in Figure 5.8.



Step (c) - A negative reading indicates that the device is probably DTI



Step (e) - A negative reading indicates that the device is probably DCE

Figure 5.8 - Testing the Gender of an Unknown RS-232 Port

Once we have ascertained whether a device is DCE or DTE we still have a number of problems that may need to be resolved. The most common is the problem of what to do when we want to interconnect two devices that are both the same sex. In the simplest scenario, where the manufacturers have been wise enough not to use hardware hand-shaking in the link, it is possible to construct a special 3-wire cable to form the DCE/DCE or DTE/DTE connection as shown in Figure 5.9. This cable configuration is referred to as a 3-wire "null-modem" cable, since it makes the RS-232 system operate as if the modems were transparent.

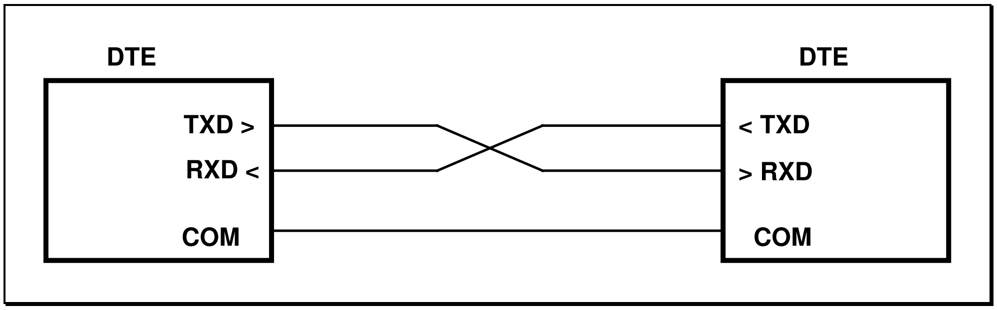


Figure 5.9 - Simple, 3-wire "Null Modem" Interface for Devices of the Same Gender DTE to DTE (or DCE to DCE)

The real problems with RS-232 systems arise when hardware hand-shaking is involved. We could simply try to cross all the hand-shaking lines as we have with the data lines. The connection is shown in Figure 5.10. This works for some signals, but fails when we don't have complementary hand-shaking lines. DCE and DTE devices are clearly not symmetrical. For example, the Carrier Detect and Ring Indicator lines are both outputs from a DCE and both inputs to a DTE - there is no purpose in crossing over two pairs of inputs. RTS and CTS are complementary, but crossing these two lines over does not necessarily achieve a sensible handshaking scheme.

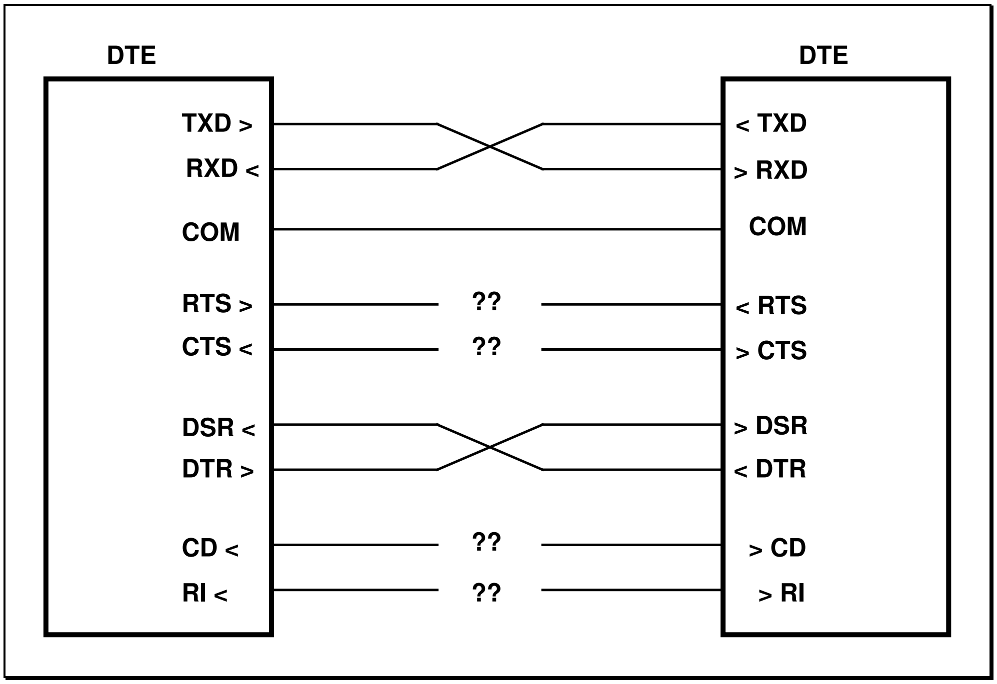


Figure 5.10 - Making a Null Modem with Hand-shaking Lines

The problem illustrated is generic and a similar situation will arise if we try to interconnect two DCE devices that both expect full hardware hand-shaking. You should now appreciate why hardware hand-shaking is so undesirable. The same type of problem also occurs when one device is equipped with hardware hand-shaking whilst the other is not. Some devices will not function at all until their hardware hand-shaking lines are enabled. For example, if the CD is not enabled on some DTE devices they will assume that the transmission link is inactive and therefore will not transmit. To make matters worse, many devices only input or output some of the 6 standard hand-shaking lines.

We therefore need to be able to trick devices into thinking that hand-shaking has been fulfilled. For example, with the UARTs of Figure 5.10, the Data Terminal Ready line on each device will be enabled whenever that device is ready for transmission. We can short-circuit the DTR line to the CD line, thereby causing each device to think that a carrier is always present. We can also short-circuit the RTS and CTS lines on each device so that whenever an RTS is asserted, the CTS is automatically received. This connection is shown in Figure 5.11.

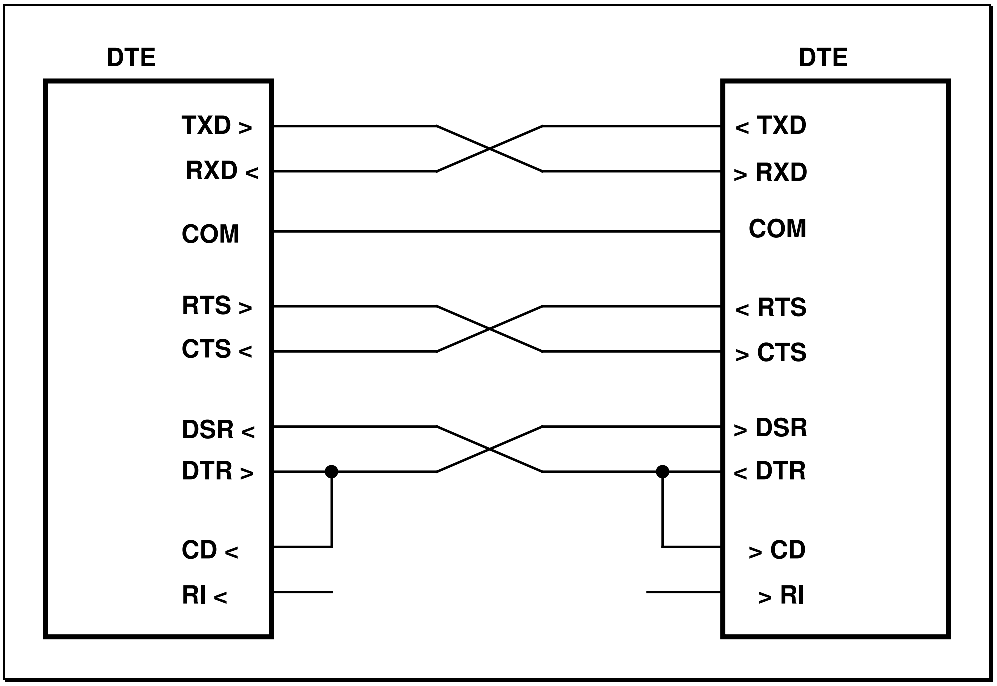


Figure 5.11 - Rectifying Hand-shaking Problems on RS-232

A more complex problem that can arise is where one RS-232 device requires a hand-shaking pin enabled, but there are neither complementary pins on the other device nor other enabled pins on the local device. In this case, we sometimes have to borrow a known "enabled" signal from the remote device by using an extra line.

Another common problem with RS-232 hardware hand-shaking occurs when a remote DCE device is unable to supply a CTS signal to a DTE. In this case we can short-circuit between the DTR and the CTS pins on the local DTE.

It is also possible, as a last resort, to by-pass the hardware hand-shaking on both DTE and DCE devices using the connections shown in Figure 5.12.

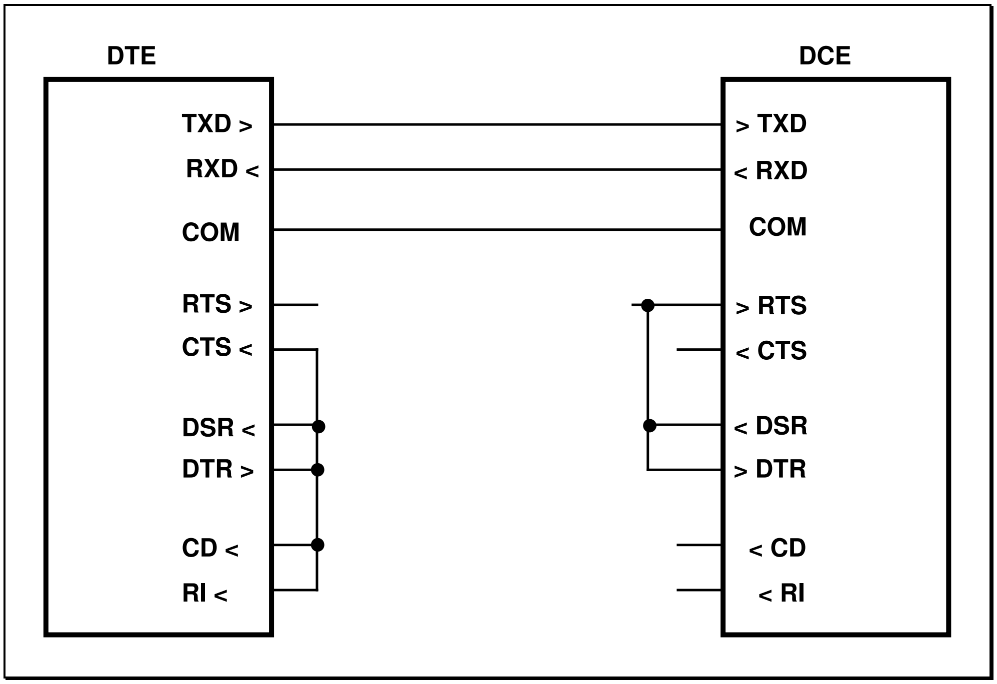


Figure 5.12 - By-passing Hardware Hand-shaking

Figure 5.12 would seem to indicate that it is possible to drive an RS-232 link entirely without hardware hand-shaking. This is only true provided that both the devices are aware that hardware hand-shaking is inoperative. The connection shown in Figure 5.12 would fail in an environment where the hardware hand-shaking was used extensively to co-ordinate the flow of data. For example, in Figure 5.12, as far as the hardware hand-shaking goes, both devices are free to transmit continuously. If this situation is acceptable (and in some cases it is) then the connection can be used - otherwise it can cause other problems.

The other general problem with by-passing the hand-shaking process is that one needs to understand exactly which signals a device needs enabled before it will allow transmission to occur. For example, if DTE UART is set to work in a half-duplex mode and the Carrier Detect (CD) line is enabled, then the device probably won't transmit because the CD is designed to indicate that a remote device is already transmitting. If a DTE UART is set to work in a full-duplex mode, then it probably won't transmit unless the CD is enabled, because in this instance the CD indicates that the link is active.

The result of all this discussion is that every RS-232 link has to be specially tailored, bearing in mind the gender and availability of hardware hand-shaking. There are a few off-the-shelf cable configurations such as the simple 3-wire straight through TXD, RXD and COM) and corresponding null-modem connection (in Figure 5.9) which do not use hand-shaking. DB25 → DB9, DB9 → DB25 and male to female converters are also common items. However, for the majority of applications the link designer must become familiar with the hand-shaking problems and solutions thus far described in order to establish effective links.

5.5 A Summary of Points Related to RS-232 Links

(i) There are no connectors that are universally adopted for RS-232, although the DB25 and DB9 are in common usage. DB connectors are general-purpose electrical connectors and it is very dangerous to assume that a plug is RS-232 compatible just because it uses one of these connectors. Conversely, it is also possible that a plug, which does not utilise one of the DB connectors is RS-232 compatible.

(ii) The common pin configurations for the DB25 and DB9 connectors are outlined in Table 5.1, but are not universally adopted. Always compare with the manufacturer's specification for the connector.

(iii) The male or female gender of a connector is in practice not a reliable means of assessing whether a device is DCE or DTE.

(iv) If one is sure that a connector is RS-232, then it is possible to short-circuit any number of pins (on that connector) together without damaging the port. The RS-232 voltage levels (-5 -> -15v and +5 -> +15v) corresponding to 1 and 0 are normally adhered to in practice.

(v) Before connecting two devices of unknown genders (DCE/DTE), it is wise to first ascertain their true genders from electrical testing of transmit and receive pins - a manufacturer's specification is not necessarily reliable.

(vi) The hardware hand-shaking on RS-232 ports varies dramatically from one device to another. Some devices have a number of inactive hand-shaking lines whilst others use the complete set.

(vii) Establishing RS-232 hardware hand-shaking often requires tricking devices into thinking that signals are present. This is achieved by short-circuiting the required hand-shaking pins to known "enabled" pins.

(viii) There are no "standard" RS-232 connector cables. Although there are some which are commonly used, generally, each cable is tailored for a specific application.

(ix) Hardware hand-shaking is undesirable and software hand-shaking should always be used wherever feasible. That is, where possible, RS-232 links should preferably be designed as 3-wire (TXD, RXD and COM) links.

5.6 Devices to Assist in Establishing Serial Links

After reading the previous section, the reader may be left to wonder whether there are any devices available to make the task of the RS-232 link designer an easier one. As it turns out, there are many such devices.

One of the biggest practical problems in making test links is that RS-232 connectors and twisted-pair cables are very awkward to handle and difficult solder. Without some special tools, experimenting with hand-shaking can be very time consuming and unproductive.

The simplest RS-232 design tool is ribbon cable. A ribbon cable is nothing more than a number of insulated, parallel conductors joined together to form a flat ribbon. This is an ideal mechanism for making short test links. A good starting point for RS-232 testing is to try transmitting via a straight-through connection using ribbon cable. A 25 conductor ribbon cable and special crimp connectors on DB25 plugs make it possible to form "straight through" test leads very quickly and without the use of solder. It should be noted however, that the conductors in ribbon cable are not shielded from electro-magnetic interference and are subject to cross-talk if the ribbon cable is too long. It is therefore imperative that the ribbon cable is only used over short distances (several metres maximum) and only in an electro-magnetically clean environment. Once the pin-connection diagram for the cable has been determined, the ribbon cable should be replaced with shielded, twisted-pair cable.

Another useful device for deciphering the hardware hand-shaking on RS-232 links is referred to as a "break-out" box. A typical device is shown in Figure 5.13. The basic purpose of a break-out box is simply to bring the 25 pins from each of two RS-232 devices (which are ultimately to be linked) to a common central point. One can then experiment with hand-shaking by making connections at a convenient location. Most break-out boxes also feature a bank of switches which can connect or disconnect corresponding pins. Many also feature switches which can cross connect complementary lines for "null modem" experimentation. Another common feature of break-out boxes is that they contain light emitting diodes to show which lines are currently enabled or disabled.

A number of low-cost link design software packages are also available to assist with hand-shaking. Some systems allow users to manually enter the results of break-out box testing for computer analysis. Other systems work by connecting an unfamiliar RS-232 device to the Personal Computer running the analysis software. The computer then analyses the hand-shaking requirements of the unfamiliar device. The only problem with such a system is that it relies upon the computer having the traditional RS-232 port structure in order to operate in the first instance.

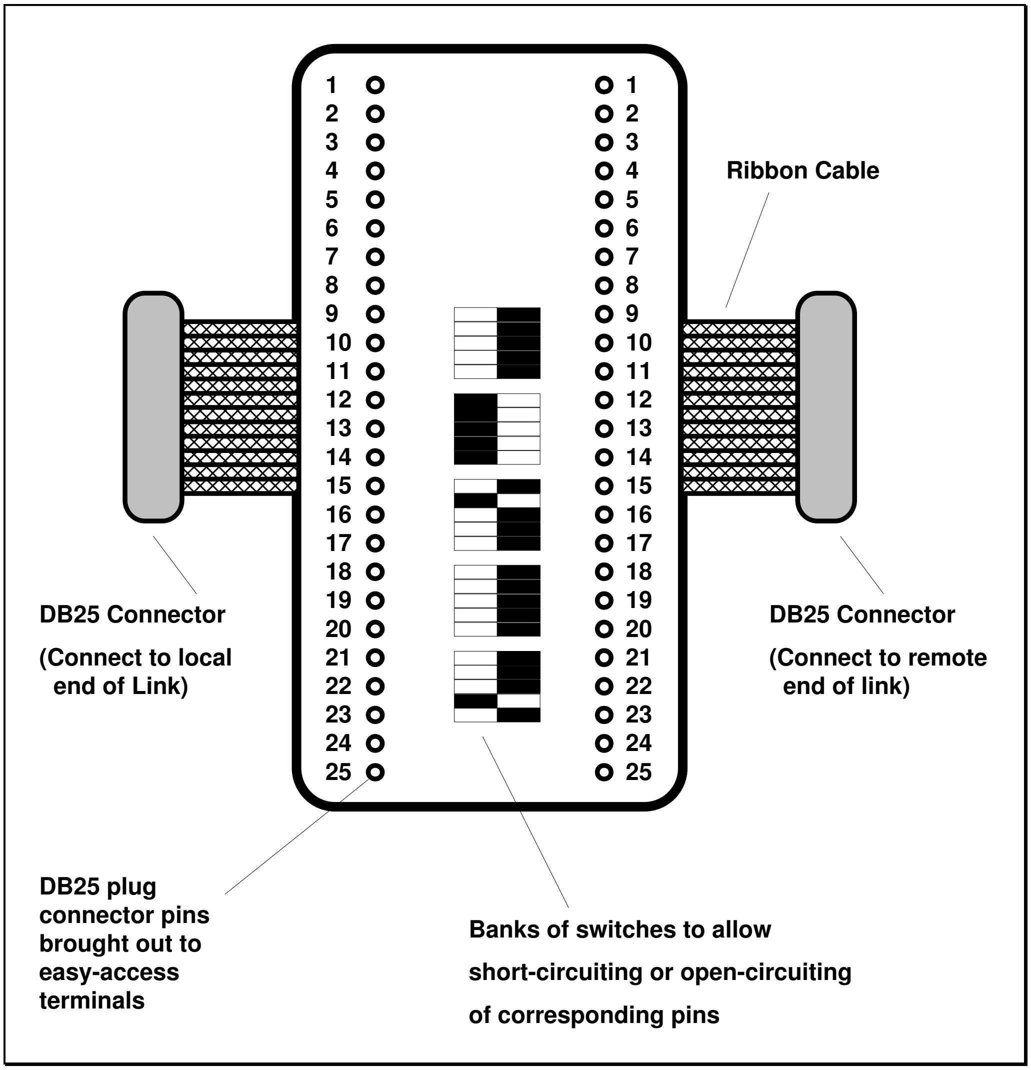


Figure 5.13 - Typical Break-out Box

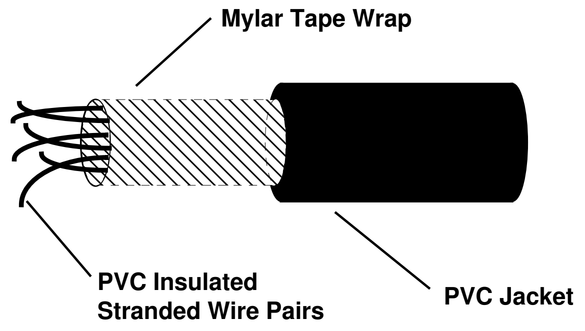
A far more sophisticated link designer's aid is the "Line-Analyser". Typically this is a micro-processor based system capable of monitoring all data and handshaking activity on a serial link for the purpose of detecting and correcting any number of faults and anomalies. This is however an expensive tool which is not normally available in manufacturing organisations.

5.7 Selecting RS-232 Cables and Line-Drivers

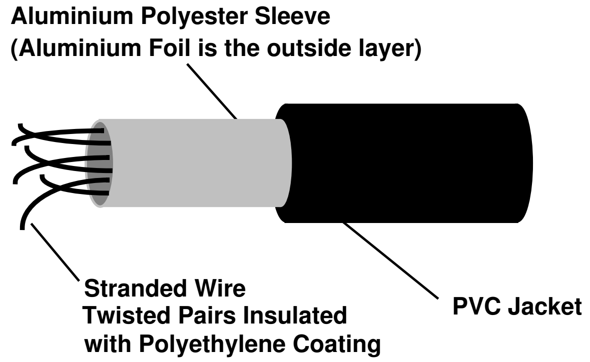
The cables that are used for RS-232 communications are neither unique nor do they have any outstanding qualities. The physical RS-232 link can be achieved by any number of general purpose communications cables. For an RS-232 link designer with an unlimited expense account, it may seem that selecting an appropriate cable is as easy as looking up an equipment catalogue and choosing the most expensive variety. To some extent this is true, but it is most important to appreciate the different varieties of cable available and their performance and limitations before making a selection.

General purpose communications cables are composed of a cluster of insulated, twisted-pairs of conductors enclosed in some form of plastic sheath. In twisted-pair systems it is possible to buy cables with differing numbers of internal conductors - 4, 7, 12, 16 and 25 conductor cables are common. The cost per metre of a 25 conductor cable is typically triple that of the equivalent 4 conductor cable. Consequently one only buys a cable with the minimum number of conductors required to achieve an operable link. Therefore, the first step in the cable selection process is always to determine the type of hand-shaking involved in the RS-232 link and the actual number of conductors required to maintain a reliable transfer of data. We can then begin to look at a range of cables that will supply at least this number of conductors.

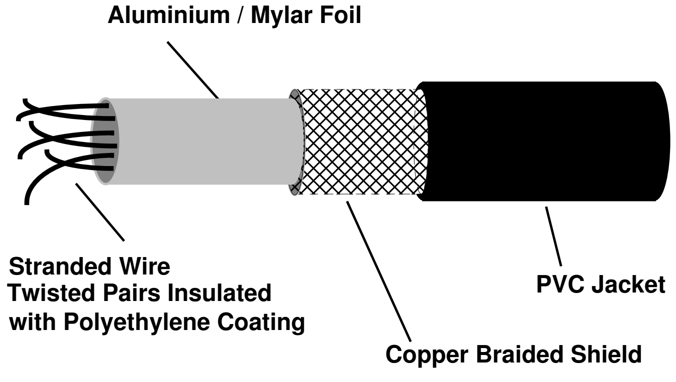
General purpose communications cables are marketed in a range of different quality levels. The quality of the cable depends upon the way in which the various cable parameters have been optimised. Shunt capacitance, series resistance and susceptibility to electro-magnetic fields are the three parameters that define the quality of a general-purpose, twisted-pair cable. The lower the series resistance and shunt capacitance (per metre) of the cable, the less the data signals will be distorted and attenuated and therefore the higher the possible transmission distance. The better the cables are shielded from electro-magnetic interference the less likely it is for data to be corrupted. Shielding from electro-magnetic interference is achieved by surrounding the data conductors with conducting materials. Some low-cost, twisted-pair cables have no shielding, whilst others use an aluminium/mylar foil or braided copper cage. High quality cables use both the aluminium foil and braided copper for shielding. A range of twisted-pair cable grades are illustrated in Figure 5.14 to highlight the differences in cable composition.



(a) Typical "Standard" Grade Twisted-Pair Cable



(b) Typical "Medium" Grade Twisted-Pair Cable Arrangement



(c) Typical "High" Grade Twisted-Pair Cable Arrangement

Figure 5.14 - Compositions of Commercially Produced Twisted-Pair Cables

In addition to the composition of the cable, the quality of the conductors must obviously affect the total performance of the cable. Higher grade cables use better conductors than are normally used in standard grade cable (Typically 24-AWG in high grade cables and 22-AWG in standard grade). A standard grade cable, such as that shown in Figure 5.14 (a) would typically be rated with a maximum transmission distance of 15 metres using RS-232 signals. The higher grades of cables, shown in Figure 5.14 (b) and (c) would typically be rated at 150 metres - a factor of 10 difference.

Other criteria that must be considered when selecting cables for communication, within the industrial environment, are their resistance to industrial gases and liquids and more importantly their resistance to fire. Cables in which insulation failures arise from normal (industrial) use can be the source of data corruption and considerable expense in terms of fault-finding. In situations where data links are of vital importance, the cost of fire resistant cable may be well justified. Teflon or Halon insulated cables are fire resistant cables that can be used in instances where, say, emergency shut-down signals need to be sent to production machines after the eruption of a factory fire.

Shielding can isolate data conductors from electro-magnetic interference whilst they are in the cable, but what about the connectors and terminators? Connectors are often overlooked, but in an industrial environment an inappropriate connector can allow electro-magnetic interference to corrupt data signals just as they would be corrupted in unshielded cable. Metallic (conducting) casings provide the same type of shielding for conductors inside connectors as do copper braiding and foil on cables. From a safety point of view it is also preferable that such conducting casings are insulated in an non-conducting sleeve.

We have already noted that improving the quality of a cable (optimising its parameters) increases the distance over which we can transmit using RS-232. However, no matter how high the cable quality there are always distance limits to transmission. These can be increased through the use of line-drivers (boosters) and modems. Line boosters are powered amplifiers that are placed at strategic lengths along a transmission line and act as relay stations. They are transparent to the operation of the data link itself. Similarly, modems and other signal drivers can be used to transmit signals over long distances through varying media.

Figure 5.15 (a) shows two computers (A and B) linked by a short RS-232 link that contains hardware hand-shaking tailored for the specific application. If we wish to separate computers A and B by a long distance, then we can use line-drivers at each end of the long distance line. This is shown in Figure 5.15 (b). However, in doing so we now find that we have to satisfy the hand-shaking between each device and its local line-driver rather than that of the distant computer. It is yet another reason why hardware hand-shaking is so undesirable.

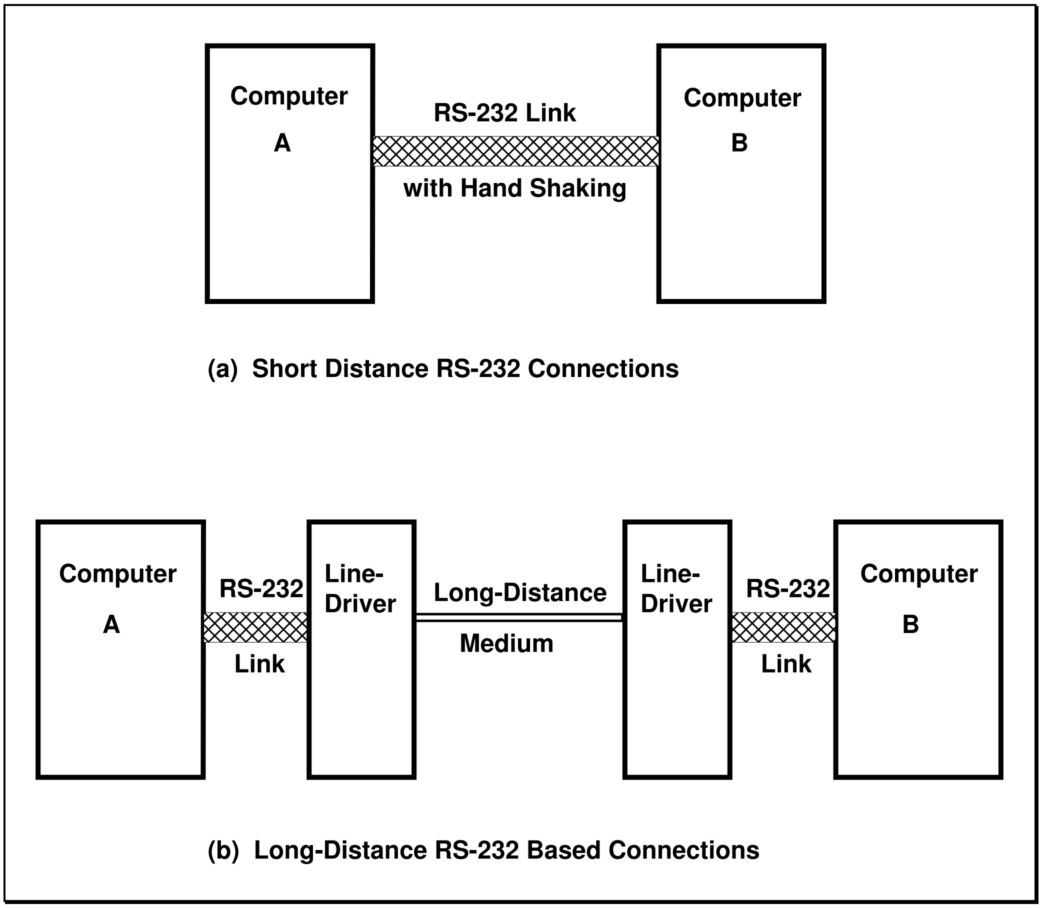


Figure 5.15 - Extending the Range of the RS-232 System

The long-distance transmission medium shown in Figure 5.15 can be one of a number of different forms. It can be twisted-pair, co-axial or optic fibre cable or for that matter just plain air, with data transmission occurring through electro-magnetic propagation at radio frequencies.

Optic fibre cables convey digital signals by transmission of light pulses in a glass-cored cable and are therefore impervious to electro-magnetic interference. They are therefore ideal for the industrial environment. Optic fibre links are commonplace in industry, particularly in point to point links where connectors and interfacing circuits (which convert from RS-232 to light pulses and vice-versa) are readily available at reasonable cost.

The transmission of data through air at high frequencies also has its advantages because equipment can be moved around without having to reorganise cables. However in order to minimise the effects of electro-magnetic interference from industrial equipment, it is often necessary to transmit at very high frequencies. This is referred to as "millimetre wave" transmission. The problem with millimetre wave transmission is that the transmitter and receiver need to be on a direct line of sight in order for correct data transmission to occur. In other words, physical obstructions between devices can cause problems.

Overall, extending the range of RS-232 communications to long distances, which are in the order of kilometres, is not a difficult proposition, provided that the two remote devices are not relying upon one another for hardware hand-shaking. There are many commercially available devices that will convert and/or modulate RS-232 signals into a form suitable for transmission over such extended distances.

For all the short-comings of RS-232 it is fortuitous that widespread use of the system has led to the proliferation of an enormous range of accessories, interfaces and line-drivers. These accessories are not only widely available but also of a relatively low cost per link.

5.8 Configuring UART Parameters

Once a "point to point" asynchronous (RS-232) serial link has been physically realised, by constructing a cable to interface the devices at either end, there is still some hardware configuration to be performed before any sensible data transfer can occur.

The UART on each device is driven by a clock signal, whose frequency can be varied to alter the number of bits per second at which data is fed from the output register to the transmit line. The clock signal also determines the rate at which incoming data is fed from the "receive" line to the input register. It is therefore essential that the devices on either end of an RS-232 link have UARTs set with the same clock speed so that data will be correctly interpreted.

The UART control circuits on two communicating devices must also agree on the number of bits used to represent data. Since both 7 and 8 bit data formats are common on RS-232 links it is also important that devices are both set to use the same convention. The number of stop bits on RS-232 links also varies with differing applications. Both 1 and 2 stop bit formats are common and again it is important that both devices are set to the same format in order for data to be framed correctly. The same applies to parity checking bits. As long as both devices agree on one of odd, even, mark, space or no parity, data can be correctly framed and interpreted by the UARTs on each device.

A complete RS-232 frame is shown in Figure 5.16, incorporating a start bit, 8 data bits, a parity check bit and 2 stop bits.

In any user-programmable system, UART parameters are configured or altered through the CPU (microprocessor), which can load binary strings into UART control registers, via the data bus. The UART control registers can be addressed by the CPU in the same way as other memory locations in the system. In order for this transfer of data to occur, a simple piece of code has to execute on the CPU. In some computer systems, the Operating System has built-in commands that allow a user to configure a serial port through its UART command registers. Software developers however, often need to write their own sub-programs to set up the UART parameters before feeding information into or out of a serial port.

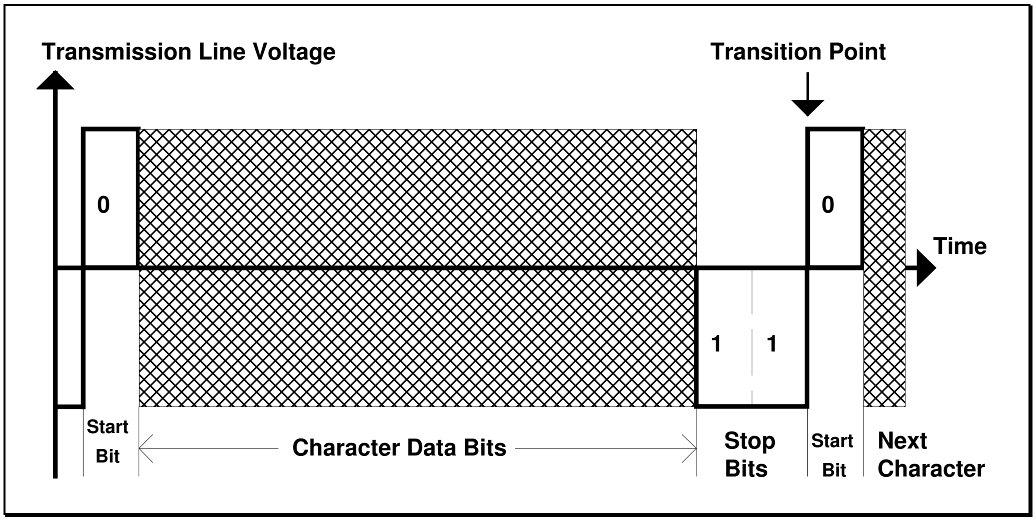


Figure 5.16 - RS-232 Signal Transmission (Framing)

In less intelligent devices (such as serial printers) UARTs have preset parameters that cannot be changed by user software programming. However, some of these devices allow the user to "hardware program" the UART parameters through the use of hardware switches that physically set bits in UART control registers to appropriate states. In the unlikely situation where neither of two devices can be programmed to achieve "matched" UART parameters, then clearly no sensible RS-232 communication can take place.

In situations where both devices can be programmed, the following priorities should be assigned:

- Maximum bit rate (ultimately governed by the quality of the communications cable, but typically up to 19200 bps)
- 1 stop bit (2 stop bits increase overheads)
- No parity check bit (if software hand-shaking is used)

Note that unless the software, running on the linked devices, is capable of flagging parity errors or taking corrective measures, then there is no value in using parity. It only increases overheads.

The minimum number of bits that must be transmitted for every single transmitted character: is

1 start bit ⎤
7 data bits ⎥ ⇒ 22% overhead
1 stop bit ⎦

The maximum number is:

1 start bit ⎤
8 data bits ⎥ ⇒ 33% overhead
1 parity bit ⎥
2 stop bits ⎦

In as much as the maximum bit rate of the link is sometimes outside the control of the end user, it is therefore wise to minimise the unnecessary bit overheads in the system in order to increase performance.

5.9 Bit Rates and Baud Rates

At this point it is appropriate to introduce a term used in conjunction with the speed of communications links - that is, the "Baud Rate". Up until now, its use has been avoided and link speeds have been specified in "bits per second" (bps) because this is a precisely understood quantity.

The term Baud rate is derived from the French communications pioneer "Baudot" and is a quantity describing the "signalling" rate. The signalling rate defines the length of the shortest signal divided into one second. In simple systems, such as straight RS-232, where no modulation is used, the shortest signal is the bit and therefore in this situation the Baud rate equals the number of bits per second.

<u>However</u>, when modulation is used the signal on the transmission line, at any instant in time, can contain 2 or more bits of information (multiple channels). In this case the bit rate and the Baud rate are not the same and clearly the Baud rate is only a fraction of the bit rate.

We therefore note that the "Baud rate" is (in general) not the number of bits per second at which data is transmitted. It is the number of signal units per second on the transmission line. Each signal unit may contain multiple bits of data.

5.10 RS-422 and RS-449 Hardware Links

You should now be aware that in the RS-232 system, any spurious voltage that is induced on the transmit or receive lines is liable to corrupt the interpretation of data. This is because the receiving device at each end of the link decodes the binary value of a signal based upon its voltage with respect to the signal ground line. The RS-422 signalling standard, which is also referred to by the CCITT specification V.11, attempts to address this problem of noise induction along a transmission line.

In the RS-422 system, the transmit and receive lines are paired "differential" circuits. This is shown schematically in Figure 5.17, where two transmission lines are used for each signal. This means that a minimum of 5 conductors are required for a full-duplex link.

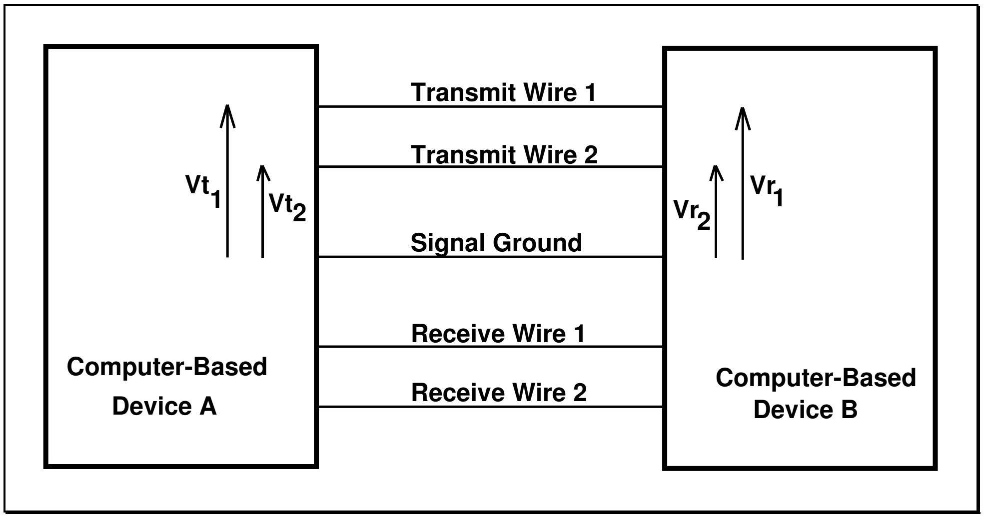


Figure 5.17 - Transmission Using Paired Differential Circuits

In differential-voltage communications systems, each transmitter produces two voltage signals of equal and opposite polarity to represent every binary digit. The receiver in such a system looks only at the difference between the two voltages in the pair of lines to determine whether the signal represents binary 1 or 0. Any noise that is induced on the link will be induced equally on the pair of transmit conductors. However since the receiver only looks at the difference between the two, the noise component is ignored because it is the common component. This is referred to as common mode rejection. Typical differential signal voltage waveforms for the transmit wire pairs are shown in Figure 5.18.

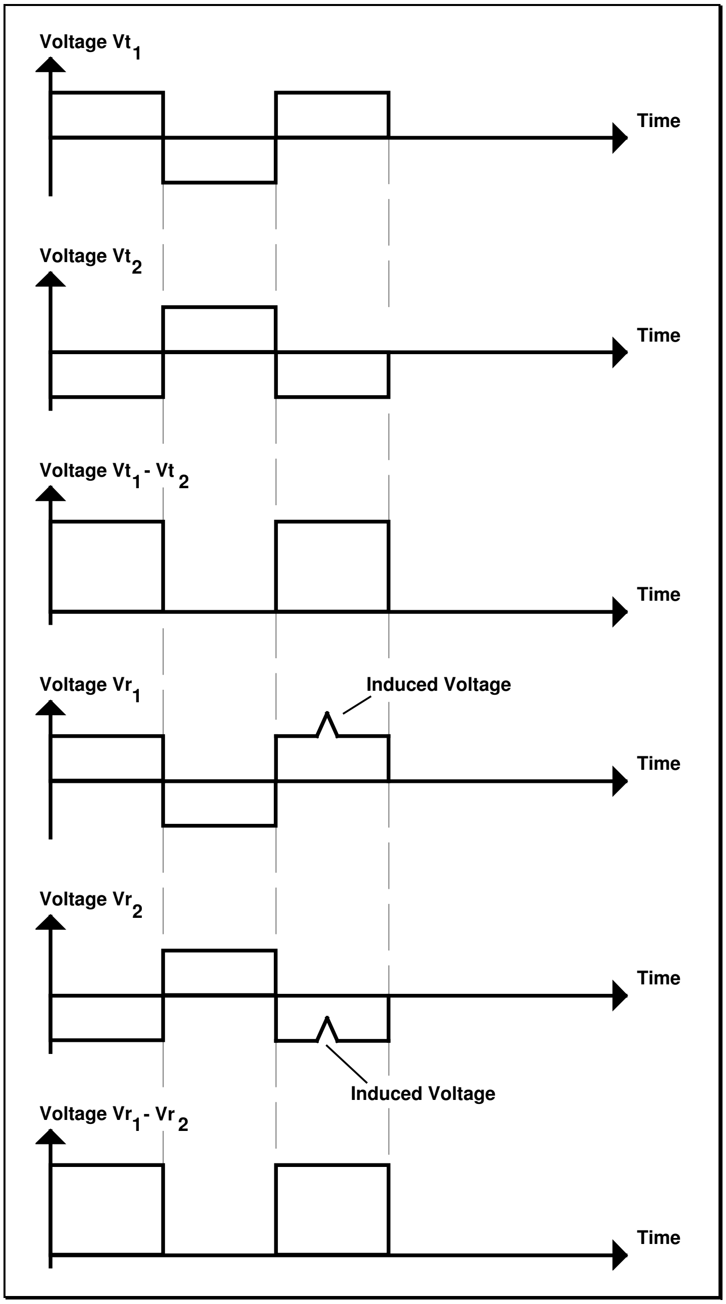


Figure 5.18 - Differential Signal Properties

A modified version of RS-422 has also been defined and is referred to as RS-423. RS-423 circuits allow RS-232 outputs (which are single-ended) to be input to a differential receiver.

The RS-449 system is the communication standard that uses RS-422 signalling techniques. It is also referred to, using CCITT terminology, as the V.35 standard. RS-449 is analogous to the RS-232 system in that it has similar hand-shaking lines and a receive and transmit circuit. The major difference is that all signals are differential and not single ended. RS-449 connectors generally come in the form of a 37 pin "D" shaped plug. This is shown in Figure 5.19.

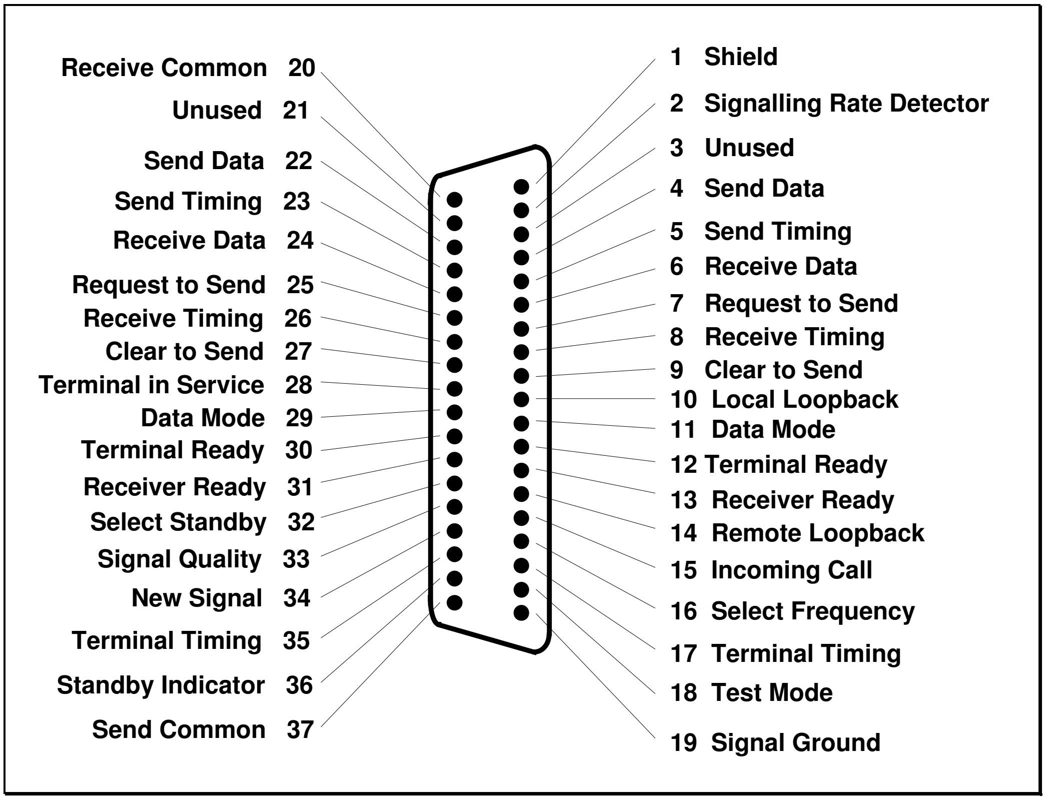


Figure 5.19 - RS-449 Connector Plug

Figure 5.19 shows some hand-shaking lines that clearly have similar functions to those in RS-232. For example, in RS-449 terminology, the Data Mode and Receiver Ready lines correspond to the Data Set Ready and Data Terminal Ready lines of the RS-232 system. A feature of the RS-449 system that is not found in RS-232 is the Test Mode line. When a DTE enables this line, the DCE automatically loops back (echoes) its incoming signal back through to the Receive lines of the DTE. This allows a DTE device to test for faults in the DCE.

The balanced circuits of RS-449 allow it to operate at much higher bit rates and over longer distances than RS-232. Typically it is possible to transmit at a rate of one megabit per second over twisted-pair links of 100 metres. The separation can be even greater at lower bit rates.

Although RS-449 is inherently a better system for data transfer than RS-232 it is not widely used in point to point links within the manufacturing environment. Few industrial equipment manufacturers provide an RS-449 port as standard equipment. This is another example of how volume generally dictates technology in computing and data communication environments. Many original equipment manufacturers design their systems with the most prolific communications interfaces and not necessarily those that are the best suited for the task. Since personal computers and workstations are the most prolific "computing" devices, one can often gauge trends for industrial computer controllers from the events that take place in the office environment - whether these are directly relevant technologies or not. In this instance we find that RS-232 interfaces are provided on computer controlled machines because the specification is very common on personal computers.

A number of data communications equipment suppliers provide low cost interfaces and line-drivers which facilitate conversion between RS-232 and RS-449 signalling. This provides another technique for extending the range of communications on RS-232 devices. However, the usefulness of RS-232 to RS-449 converters has been diminished since the introduction of RS-232 to optic fibre converters, which provide far greater noise immunity.

5.11 The 20mA Current Loop

Another alternative to the RS-232 specification for serial data communication is the 20mA current loop. This system allows for longer transmission distances than RS-232 but at similar bit rates. Transmission lines of a kilometre in length are feasible with the 20mA loop system.

As its name implies, binary information is represented by the current flowing from the transmitter to the receiver (and back again). A current of 20mA signifies a binary 1 and an open circuit (zero current) signifies a binary 0. The 20mA loop system is shown schematically in Figure 5.20.

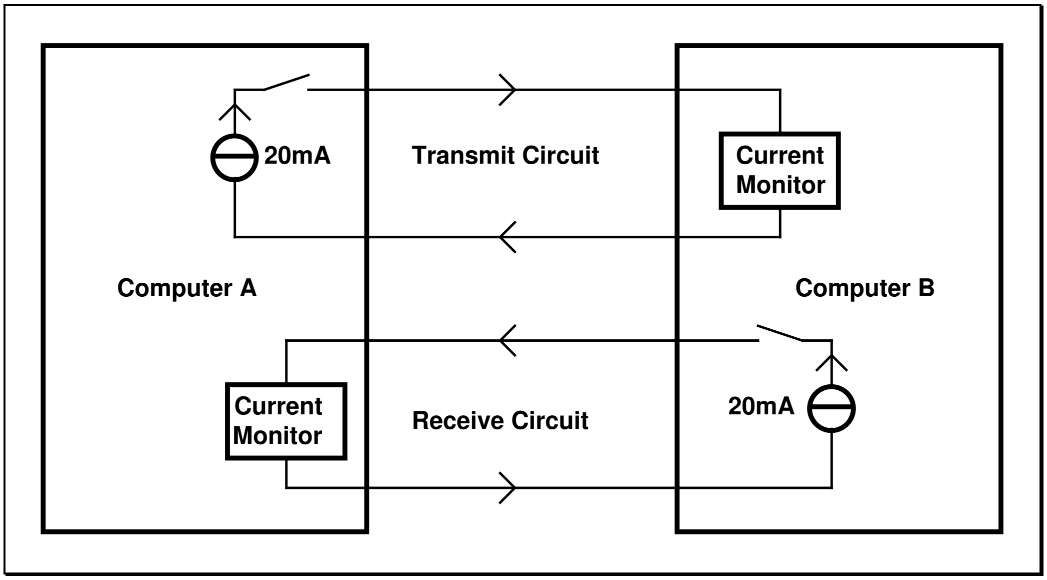


Figure 5.20 - Schematic of 20mA Current Loop System

As with the RS-449 system, a pair of wires is used for data transmission and hence the Common Mode Rejection properties of the 20mA loop system are good. However, as with the RS-449 system, use of the 20mA loop system within manufacturing is very limited.

5.12 Summary of Key Factors in Point to Point Serial Links

The following points summarise the discussions contained with this chapter in terms of establishing a serial point to point link between two devices. They are schematically shown in Figure 5.21.

(i) Determine the standard to be used for communications between devices (ie: RS-232, RS-449, etc.)

(ii) Determine the gender of devices (DCE/DTE)

(iii) Satisfy the hardware hand-shaking requirements of devices, remembering to avoid hardware hand-shaking if at all possible

(iv) Try to achieve an operational link over a short distance, using ribbon cables and break-out boxes

(v) Determine the actual required separation of the devices

(vi) Select an appropriate transmission medium - (ie: cable and line-drivers or modems) with due regard to electro-magnetic interference and susceptibility to chemical and fire exposure.

(vii) Set the serial transmission parameters - (ie: baud rate, parity, stop bits, etc.)

All the above points are only the basic hardware considerations - or more aptly descriptions of "physical" requirements. We have not yet considered the software requirements that need to be fulfilled in order to transfer data between two devices once the physical link is functional. This will be covered in subsequent chapters.

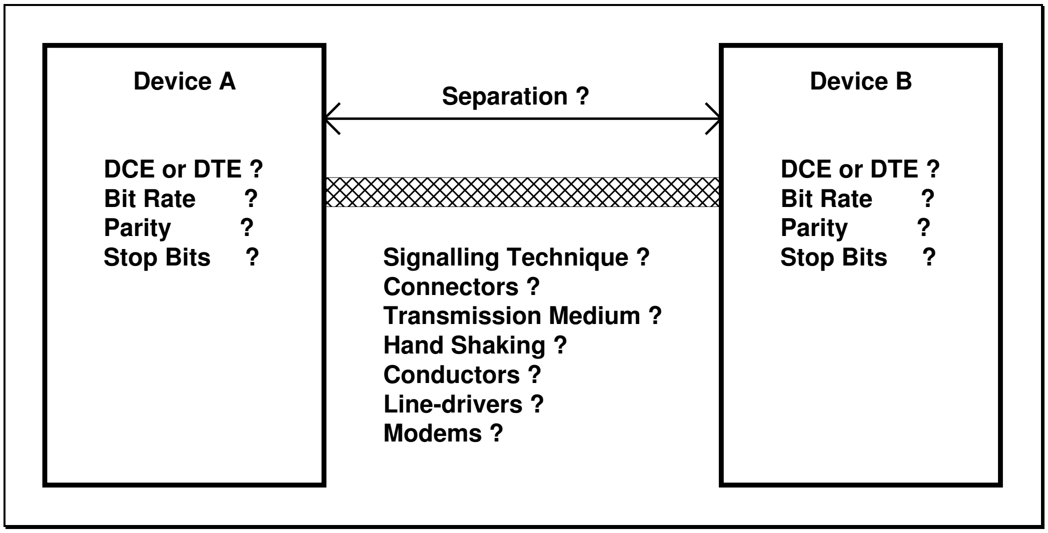


Figure 5.21 - Hardware Considerations in Serial Links