

Due to its relative simplicity and low hardware overhead (as compared to parallel interfacing), serial communications is used extensively within the electronics industry. Today, the most popular serial communications standard in use is certainly the EIA/TIA–232–E specification. This standard, which has been developed by the Electronic Industry Association and the Telecommunications Industry Association (EIA/TIA), is more popularly referred to simply as “RS–232” where “RS” stands for “recommended standard”. In recent years, this suffix has been replaced with “EIA/TIA” to help identify the source of the standard. This paper will use the common notation of “RS–232” in its discussion of the topic.

The official name of the EIA/TIA–232–E standard is “Interface Between Data Terminal Equipment and Data Circuit–Termination Equipment Employing Serial Binary Data Interchange”. Although the name may sound intimidating, the standard is simply concerned with serial data communication between a host system (Data Terminal Equipment, or “DTE”) and a peripheral system (Data Circuit–Terminating Equipment, or “DCE”).

The EIA/TIA–232–E standard which was introduced in 1962 has been updated four times since its introduction in order to better meet the needs of serial communication applications. The letter “E” in the standard’s name indicates that this is the fifth revision of the standard.

RS–232 SPECIFICATIONS

RS–232 is a “complete” standard. This means that the standard sets out to ensure compatibility between the host and peripheral systems by specifying 1) common voltage and signal levels, 2) common pin wiring configurations, and 3) a minimal amount of control information between the host and peripheral systems. Unlike many standards which simply specify the electrical characteristics of a given interface, RS–232 specifies electrical, functional, and mechanical characteristics in order to meet the above three criteria. Each of these aspects of the RS–232 standard is discussed below.

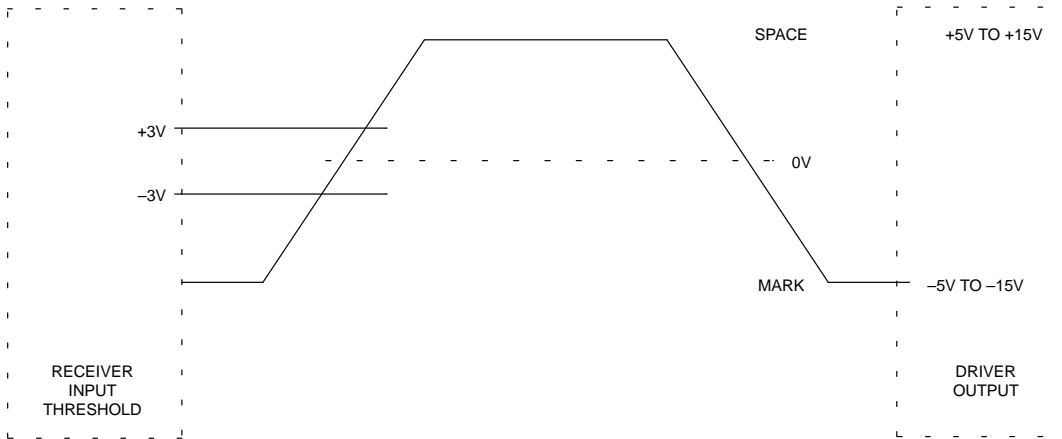
ELECTRICAL CHARACTERISTICS

The electrical characteristics section of the RS–232 standard includes specifications on voltage levels, rate of change of signal levels, and line impedance.

The original RS–232 standard was defined in 1962. As this was before the days of TTL logic, it should not be surprising that the standard does not use 5 volt and ground logic levels. Instead, a high level for the driver output is defined as being +5 to +15 volts and a low level for the driver output is defined as being between –5 and –15 volts. The receiver logic levels were defined to provide a 2 volt noise margin. As such, a high level for the receiver is defined as +3 to +15 volts and a low level is –3 to –15 volts. Figure 1 illustrates the logic levels defined by the RS–232 standard. It is necessary to note that, for RS–232 communication, a low level (–3 to –15 volts) is defined as a logic 1 and is historically referred to as “marking”. Likewise a high level (+3 to +15 volts) is defined as a logic 0 and is referred to as “spacing”.

The RS–232 standard also limits the maximum slew rate at the driver output. This limitation was included to help reduce the likelihood of cross–talk between adjacent signals. The slower the rise and fall time, the smaller the chance of cross talk. With this in mind, the maximum slew rate allowed is 30 V/μs. Additionally, a maximum data rate of 20k bits/second has been defined by the standard. Again with the purpose of reducing the chance of cross talk.

The impedance of the interface between the driver and receiver has also been defined. The load seen by the driver is specified to be 3kΩ to 7kΩ. For the original RS–232 standard, the cable between the driver and the receiver was also specified to be a maximum of 15 meters in length. This part of the standard was changed in revision “D” (EIA/TIA–232–D). Instead of specifying the maximum length of cable, a maximum capacitive load of 2500 pF was specified which is clearly a more adequate specification. The maximum cable length is determined by the capacitance per unit length of the cable which is provided in the cable specifications.

RS-232 LOGIC LEVEL SPECIFICATIONS Figure 1**FUNCTIONAL CHARACTERISTICS**

Since RS-232 is a "complete" standard, it includes more than just specifications on electrical characteristics. The second aspect of operation that is covered by the standard concerns the functional characteristics of the interface. This essentially means that RS-232 has defined the function of the different signals that are used in the interface. These signals are divided into four different categories: common, data, control, and timing. Table 1 illustrates the signals that are defined by the RS-232 standard. As can be seen from the table there is an overwhelming number of signals defined by the standard. The standard provides an abundance of control signals and supports a primary and secondary communications channel. Fortunately few applications, if any, require all of these defined signals. For example, only eight signals are used for a typical modem. Some simple applications may require only four signals (two for data and two for handshaking) while others may require only data signals with no handshaking. Examples of how the RS-232 standard is used in some "real world" applications are discussed later in this paper. The complete list of defined signals is included here as a reference, but it is beyond the scope of this paper to review the functionality of all of these signals.

MECHANICAL INTERFACE CHARACTERISTICS

The third area covered by RS-232 concerns the mechanical interface. In particular, RS-232 specifies a 25-pin connector. This is the minimum connector size that can accommodate all of the signals defined in the functional portion of the standard. The pin assignment for this connector is shown in Figure 2. The connector for DCE equipment is male for the connector housing and female for the connection pins. Likewise, the DTE connector is a female housing with male connection pins. Although RS-232 specifies a 25-position connector, it should be noted that often this connector is not used. This is due to the fact that most applications do not require all of the defined signals and therefore a 25-pin connector is larger than necessary. This being the case, it is very common for other connector types to be used. Perhaps the most popular is the 9-position DB9S connector which is also illustrated in Figure 2. This connector provides the means to transmit and receive the necessary signals for modem applications, for example. This will be discussed in more detail later.

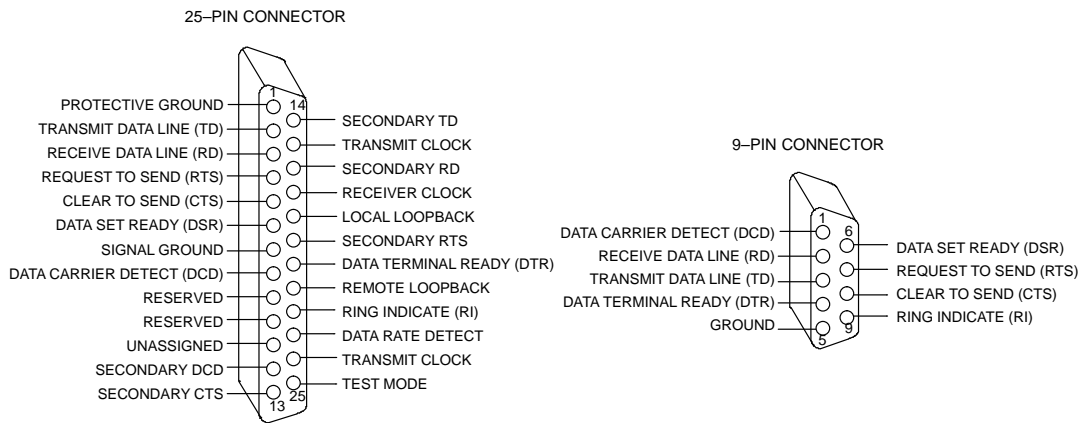
RS-232 DEFINED SIGNALS Table 1

CIRCUIT MNEMONIC	CIRCUIT NAME*	CIRCUIT DIRECTION	CIRCUIT TYPE
AB	Signal Common	–	Common
BA BB	Transmitted Data (TD) Received Data (RD)	To DCE From DCE	Data
CA CB CC CD CE CF CG CH CI CJ RL LL TM	Request to Send (RTS) Clear to Send (CTS) DCE Ready (DSR) DTE Ready (DTR) Ring Indicator (RI) Received Line Signal Detector** (DCD) Signal Quality Detector Data Signal Rate Detector from DTE Data Signal Rate Detector from DCE Ready for Receiving Remote Loopback Local Loopback Test Mode	To DCE From DCE From DCE To DCE From DCE From DCE To DCE From DCE To DCE To DCE To DCE From DCE	Control
DA	Transmitter Signal Element Timing from DTE	To DCE	
DB DD	Transmitter Signal Element Timing from DCE Receiver Signal Element Timing From DCE	From DCE From DCE	Timing
SBA SBB	Secondary Transmitted Data Secondary Received Data	To DCE From DCE	Data
SCA SCB SCF	Secondary Request to Send Secondary Clear to Send Secondary Received Line Signal Detector	To DCE From DCE From DCE	Control

*Signals with abbreviations in parentheses are the eight most commonly used signals.

**This signal is more commonly referred to as Data Carrier Detect (DCD).

RS-232 CONNECTOR PIN ASSIGNMENTS Figure 2



PRACTICAL RS-232 IMPLEMENTATION

Most systems designed today do not operate using RS-232 voltage levels. Since this is the case, level conversion is necessary to implement RS-232 communication. Level conversion is performed by special RS-232 IC's. These IC's typically have line drivers that generate the voltage levels required by RS-232 and line receivers that can receive RS-232 voltage levels without being damaged. These line drivers and receivers typically invert the signal as well since a logic 1 is represented by a low voltage level for RS-232 communication and likewise a logic 0 is represented by a high logic level. Figure 3 illustrates the function of an RS-232 line driver/receiver in a typical modem application. In this particular example, the signals necessary for serial communication are generated and received by the Universal Asynchronous Receiver/Transmitter (UART). The RS-232 line driver/receiver IC performs the level translation necessary between the CMOS/TTL and RS-232 interface.

The UART just mentioned performs the "overhead" tasks necessary for asynchronous serial communication. For example, the asynchronous nature of this type of communication usually requires that start and stop bits be initiated by the host system to indicate to the peripheral system when communication will start and stop. Parity bits are also often employed to ensure that the data sent has not been corrupted. The UART usually generates the start, stop, and parity bits when transmitting data and can detect communication errors upon receiving data. The UART also functions as the intermediary between byte-wide (parallel) and bit-wide (serial) communication; it converts a byte of data into a

serial bit stream for transmitting and converts a serial bit stream into a byte of data when receiving.

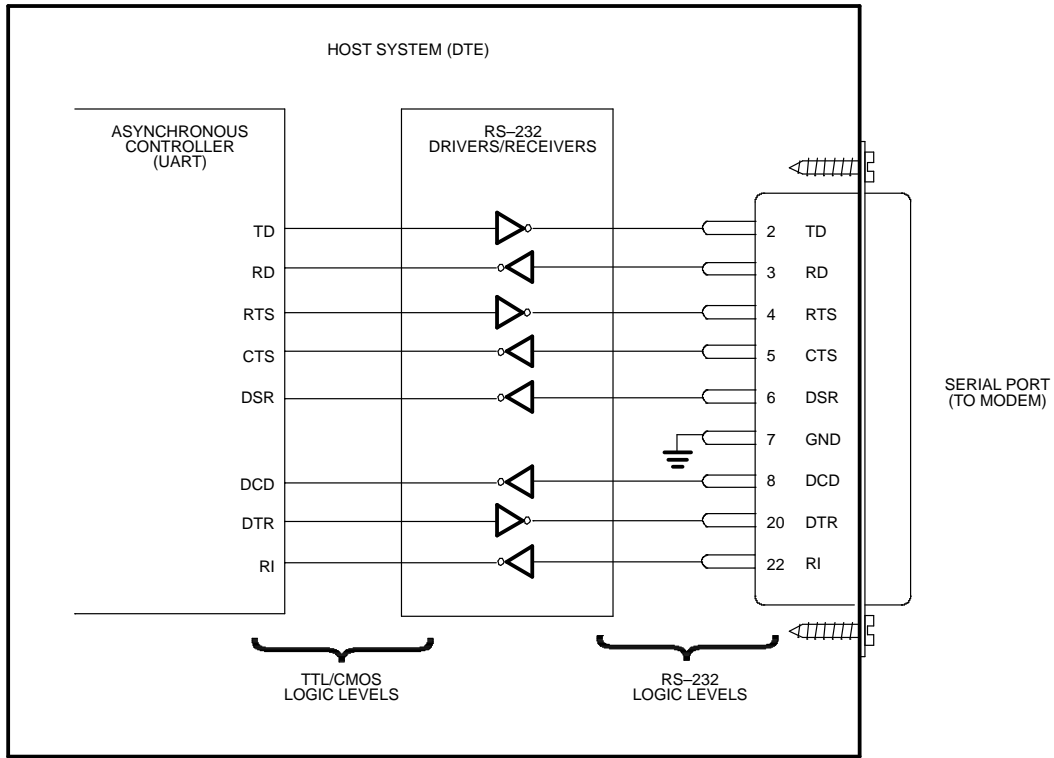
Now that an elementary explanation of the TTL/CMOS to RS-232 interface has been provided we can consider some "real world" RS-232 applications. It has already been noted that RS-232 applications rarely follow the RS-232 standard precisely. Perhaps the most significant reason this is true is due to the fact that many of the defined signals are not necessary for most applications. As such, the unnecessary signals are omitted. Many applications, such as a modem, require only nine signals (two data signals, six control signals, and ground). Other applications may require only five signals (two for data, two for handshaking, and ground), while others may require only data signals with no handshake control. We will begin our investigation of "real world" implementations by first considering the typical modem application.

RS-232 IN MODEM APPLICATIONS

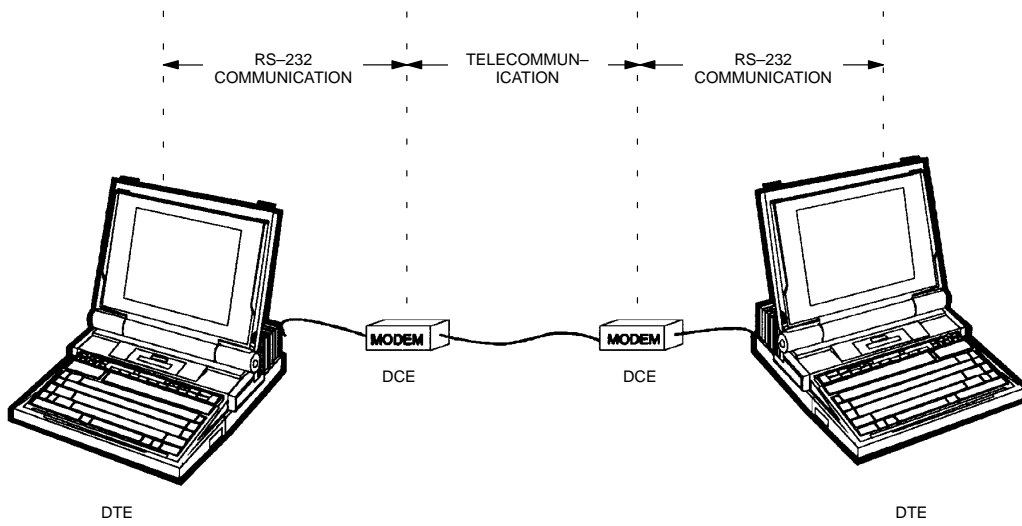
Modem applications are one of the most popular uses for the RS-232 standard. Figure 4 illustrates a typical modem application utilizing the RS-232 interface standard. As can be seen in the diagram, the PC is the DTE and the modem is the DCE. Communication between each PC and its associated modem is accomplished using the RS-232 standard. Communication between the two modems is accomplished via telecommunication. It should be noted that although a microcomputer is usually the DTE in RS-232 applications, this is not mandatory according to a strict interpretation of the standard.



TYPICAL RS-232 MODEM APPLICATION Figure 3



MODEM COMMUNICATION BETWEEN TWO PC'S Figure 4



Many modem applications require only nine signals (including ground). Although some designers choose to use a 25-pin connector, it is not necessary since there are only nine interface signals between the DTE and DCE. With this in mind, many have chosen to use 9- or 15-pin connectors (see Figure 2 for 9-pin connector pin assignment). The “basic nine” signals used in modem communication are illustrated in Figure 3. Note that with respect to the DTE, three RS-232 drivers and five receivers are necessary. The functionality of these signals is described below. Note that for the following signal descriptions, “ON” refers to a high RS-232 voltage level (+5 to +15 volts) and “OFF” refers to a low RS-232 voltage level (–5 to –15 volts). Keep in mind that a high RS-232 voltage level actually represents a logic 0 and a low RS-232 voltage level refers to a logic 1.

Transmitted Data (TD): One of two separate data signals. This signal is generated by the DTE and received by the DCE.

Received Data (RD): The second of two separate data signals. This signal is generated by the DCE and received by the DTE.

Request to Send (RTS): When the host system (DTE) is ready to transmit data to the peripheral system (DCE), RTS is turned ON. In simplex and duplex systems, this condition maintains the DCE in receive mode. In half-duplex systems, this condition maintains the DCE in receive mode and disables transmit mode. The OFF condition maintains the DCE in transmit mode. After RTS is asserted, the DCE must assert CTS before communication can commence.

Clear to Send (CTS): CTS is used along with RTS to provide handshaking between the DTE and the DCE. After the DCE sees an asserted RTS, it turns CTS ON when it is ready to begin communication.

Data Set Ready (DSR): This signal is turned on by the DCE to indicate that it is connected to the telecommunications line.

Data Carrier Detect (DCD): This signal is turned ON when the DCE is receiving a signal from a remote DCE which meets its suitable signal criteria. This signal remains ON as long as the a suitable carrier signal can be detected.

Data Terminal Ready (DTR): DTR indicates the readiness of the DTE. This signal is turned ON by the DTE

when it is ready to transmit or receive data from the DCE. DTR must be ON before the DCE can assert DSR.

Ring Indicator (RI): RI, when asserted, indicates that a ringing signal is being received on the communications channel.

The signals described above form the basis for modem communication. Perhaps the best way to understand how these signals interact is to give a brief step by step example of a modem interfacing with a PC. The following steps describe a transaction in which a remote modem calls a local modem.

1. The local PC monitors the RI (Ring Indicate) signal via software.
2. When the remote modem wants to communicate with the local modem, it generates an RI signal. This signal is transferred by the local modem to the local PC.
3. The local PC responds to the RI signal by asserting the DTR (Data Terminal Ready) signal when it is ready to communicate.
4. After recognizing the asserted DTR signal, the modem responds by asserting DSR (Data Set Ready) after it is connected to the communications line. DSR indicates to the PC that the modem is ready to exchange further control signals with the DTE to commence communication. When DSR is asserted, the PC begins monitoring DCD for indication that data is being sent over the communication line.
5. The modem asserts DCD (Data Carrier Detect) after it has received a carrier signal from the remote modem that meets the suitable signal criteria.
6. At this point data transfer can begin. If the local modem has full-duplex capability, the CTS (Clear to Send) and RTS (Request to Send) signals are held in the asserted state. If the modem has only half-duplex capability, CTS and RTS provide the handshaking necessary for controlling the direction of the data flow. Data is transferred over the RD and TD signals.
7. When the transfer of data has been completed, the PC disables the DTR signal. The modem follows by inhibiting the DSR and DCD signals. At this point the PC and modem are in the original state described in step number 1.

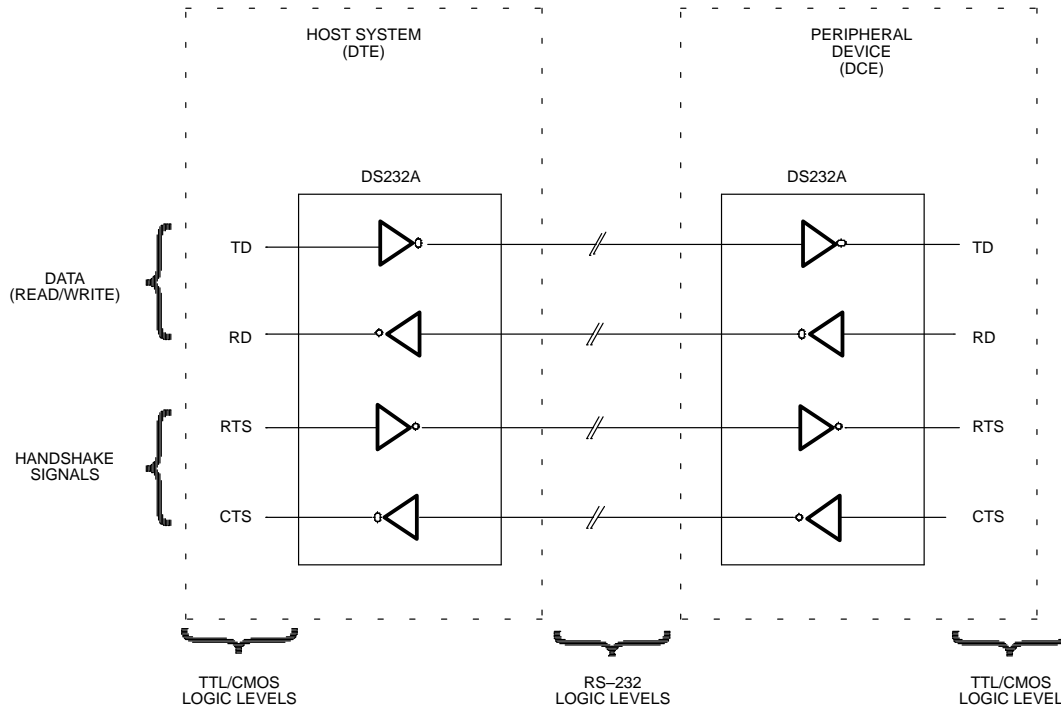
RS-232 IN MINIMAL HANDSHAKE APPLICATIONS

Even though the modem application discussed above is simplified from the RS-232 standard in terms of the number of signals needed, it is still more complex than the requirements of many systems. For many applications, two data lines and two handshake control lines are all that is necessary to establish and control communication between a host system and a peripheral system. For example, an environmental control system may need to interface with a thermostat using a half-duplex communication scheme. At times the control systems may desire to read the temperature from the thermostat and at other times may need to load temperature trip points to the thermostat. In this type of simple

application, five signals may be all that is necessary (two for data, two for handshake control, and ground).

Figure 5 illustrates a simple half-duplex communication interface. As can be seen in this diagram, data is transferred over the TD (Transmit Data) and RD (Receive Data) pins and handshake control is provided by the RTS (Ready to Send) and CTS (Clear to Send) pins. RTS is driven by the DTE to control the direction of data. When it is asserted, the DTE is placed in transmit mode. When RTS is inhibited, the DTE is placed in receive mode. CTS, which is generated by the DCE, controls the flow of data. When asserted, data can flow. However, when CTS is inhibited, the transfer of data is interrupted. The transmission of data is halted until CTS is reasserted.

HALF-DUPLEX COMMUNICATION SCHEME Figure 5



RS-232 APPLICATION LIMITATIONS

As mentioned earlier in this paper, the RS-232 standard was first introduced in 1962. In the more than three decades since, the electronics industry has changed immensely and therefore there are some limitations in the RS-232 standard. One limitation, the fact that over twenty signals have been defined by the standard, has already been addressed – simply do not use all of the signals or the 25-pin connector if they are not necessary. Other limitations in the standard are not necessarily as easy to correct, however.

GENERATION OF RS-232 VOLTAGE LEVELS

As we saw in the section on RS-232 electrical characteristics, RS-232 does not use the conventional 0 and 5 volt levels implemented in TTL and CMOS designs. Drivers have to supply +5 to +15 volts for a logic 0 and –5 to –15 volts for a logic 1. This means that extra power supplies are needed to drive the RS-232 voltage levels. Typically, a +12 volt and a –12 volt power supply are used to drive the RS-232 outputs. This is a great inconvenience for systems that have no other requirements for these power supplies. With this in mind, RS-232 products manufactured by Dallas Semiconductor have on-chip charge-pump circuits that generate the necessary voltage levels for RS-232 communication. The first charge pump essentially doubles the standard +5 volt power supply to provide the voltage level necessary for driving a logic 0. A second charge pump, inverts this voltage and provides the voltage level necessary for driving a logic 1. These two charge pumps allow the RS-232 interface products to operate from a single +5 volt supply.

MAXIMUM DATA RATE

Another limitation in the RS-232 standard is the maximum data rate. The standard defines a maximum data

rate of 20k bits/second. This is unnecessarily slow for many of today's applications. RS-232 products manufactured by Dallas Semiconductor guarantee up to 250k bits/second and typically can communicate up to 350k bits/second. While providing a communication rate at this frequency, the devices still maintain a maximum 30V/μs maximum slew rate to reduce the likelihood of cross-talk between adjacent signals.

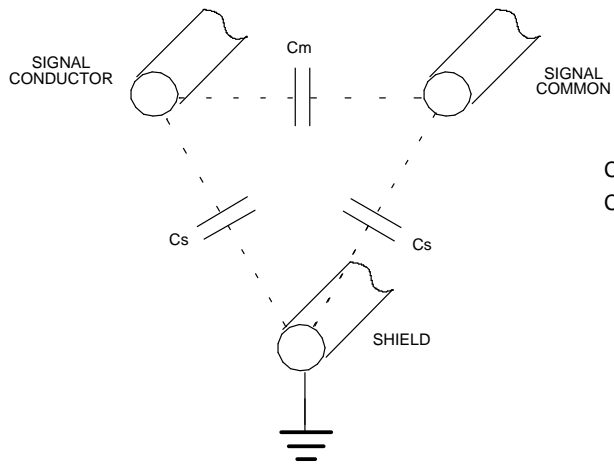
MAXIMUM CABLE LENGTH

A final limitation to discuss concerning RS-232 communication is cable length. As we have already seen, the cable length specification that was once included in the RS-232 standard has been replaced by a maximum load capacitance specification of 2500 pF. To determine the total length of cable allowed, one must determine the total line capacitance. Figure 6 shows a simple approximation for the total line capacitance of a conductor. As can be seen in the diagram, the total capacitance is approximated by the sum of the mutual capacitance between the signal conductors and the conductor to shield capacitance (or stray capacitance in the case of unshielded cable).

As an example, let's assume that the user has decided to use non-shielded cable when interconnecting the equipment. The cable mutual capacitance (C_m) of the cable is found in the cable's specifications to be 20 pF per foot. If we assume that the input capacitance of the receiver is 20 pF, this leaves the user with 2480 pF for the interconnecting cable. From the equation in Figure 6, the total capacitance per foot is found to be 30 pF. Dividing 2480 pF by 30 pF reveals that the maximum cable length is approximately 80 feet. If a longer cable length is required, the user would need to find a cable with a smaller mutual capacitance.



INTERFACE CABLE CAPACITIVE MODEL PER UNIT LENGTH Figure 6



- C_m = Mutual capacitance between conductors.
- C_s = Conductor to interface cable shield capacitance (if shielded cable is used) or stray capacitance to earth (if unshielded cable is used).
 - = $2(C_m)$ for shielded cable
 - = $0.5(C_m)$ for unshielded cable

$C_c = C_m + C_s = \text{Total line capacitance per unit length}$