RS-232 TRAINING

HISTORY

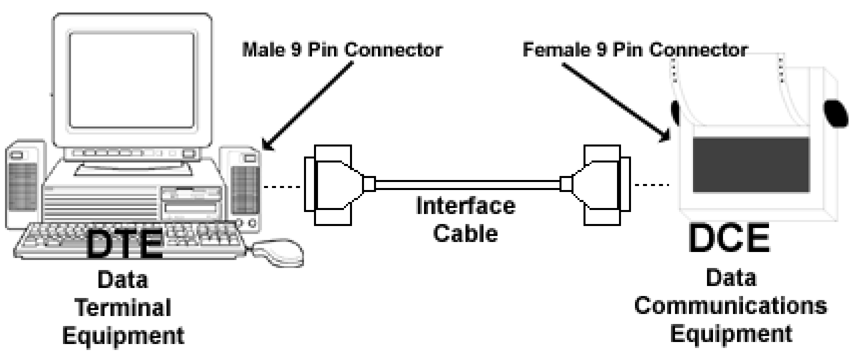
RS-232 was originally adopted in 1960 by the Electronic Industries Association (EIA). Over the 40+ years since this standard was developed, the Electronic Industries Association published three modifications, the most recent being the EIA232E standard introduced in 1991. In this document we will see several parts of the original RS-232C standard and mostly the ones used in the PC world. This document will describe the use and configuration of serial ports using RS-232.

EQUIPMENT DEVICE TYPE

RS-232 was created for one purpose, to interface between Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) employing serial binary data interchange. A PC is a DTE device, while most other devices that connect to PC serial ports are usually DCE devices. Since most DCE devices are meant to be connected to a PC, they will have a DB-9 pin female connector. All PC's have a DB-9 or DB-25 male connector for the serial port. This means that a standard cable would be female to male for either DB-9 or DB-25 pin ports.

DTE devices supply the brains for the operation – a PC requests data via bi-directional ASCII communications from a 7000XL (or any Doran indicator) – making the 7000XL a DCE device. It is also common to have a 7000XL print directly to a dumb printer or a label printer. In both of these cases, the 7000XL is the brains of the operation (DTE) as it directs the printer (DCE) what to print and when.

Since a 7000XL can be either a DCE or DTE device, the polarity of the DB-9 or DB-25 pin connectors must be specified. For connection to a PC, select a female connector and for most printers select a male connector.



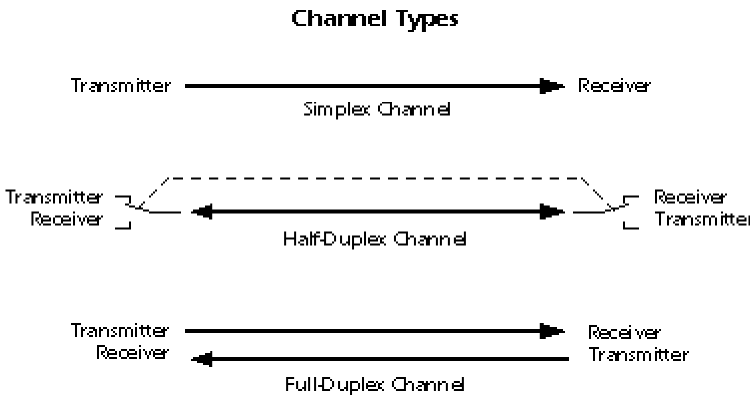
CHANNEL TYPES

A channel whose direction of transmission is unchanging is referred to as a simplex channel. For example, a radio station is a simplex channel because it always transmits the signal to its listeners and never allows them to transmit back.

A half-duplex channel is a single physical channel in which the direction may be reversed. Messages may flow in two directions, but never at the same time, in a half-duplex system. In a telephone call, one party speaks while the other listens. After a pause, the other party speaks and the first party listens. Speaking simultaneously results in garbled sound that cannot be understood. This channel requires handshaking to regulate the single line of transmission to prevent data loss.

A full-duplex channel allows simultaneous message exchange in both directions. It really consists of two simplex channels, a forward channel and a reverse channel, linking the same points. Full-duplex RS-232 communications only requires three wires.

Figure 1: Channel Types



SERIAL COMMUNICATIONS

The concept behind serial communications is as follows, data is transferred from sender to receiver one bit at a time through a single line or circuit. The serial port takes 8, 16 or 32 parallel bits from your computer bus and converts it as an 8, 16 or 32 bit serial stream. The name serial communications comes from this fact; each bit of information is transferred in series from one location to another.

The serial port on your PC is a full-duplex device meaning that it can send and receive data at the same time. In order to be able to do this, it uses separate lines for transmitting and receiving data. Some types of serial devices support only one-way communications and therefore use only two wires in the cable - the transmit line and the signal ground.

SERIAL TRANSMISSION METHODS

In the real world, some bits in the serial data transmission can be corrupted. If one bit is missing at the receiving end, all succeeding bits are shifted resulting in incorrect data when converted back to a parallel signal. So to establish reliable serial communications you must overcome these bit errors that can emerge in many different forms.

Asynchronous Communication

Two serial transmission methods are used that correct serial bit errors. The serial ports on IBM-style PCs are asynchronous devices and therefore only support asynchronous serial communications. This method is known as asynchronous communication because the sending and receiving end of the communication are not precisely synchronized by the means of a signal line. Since the DTE and DCE are not synchronized, the baud rate (speed of data transmission) and the start, stop and data bits must be configured identically on both devices. This is the most common problem when trying to set up communications between two RS-232 devices.

The asynchronous method adds markers around the data bits to help track each data frame (see figure 2). By introducing a start bit which indicates the start of a data stream, the position of each bit can be determined by timing the bits at regular intervals. By sending start bits in front of each data bit stream, the two systems don't have to be synchronized by a clock signal, the only important issue is that both systems must be set at the same port speed. When the receiving end of the communication receives the start bit it starts a short-term timer. By keeping streams short, there's not enough time for the transmitter and receiver timers to get out of sync.

An asynchronous line that is idle is identified with a value of 1, (also called a mark state). By using this value to indicate that no data is currently being sent, the devices are able to distinguish between an idle state and a disconnected line. When a character is about to be transmitted, a start bit is sent. A start bit has a value of 0, (also called a space state). Thus, when the line switches from a value of 1 to a value of 0, the receiver is alerted that a data character is about to come down the line.

In the PC environment, each stream of data bits are broken up into 5, 6, 7 or 8 bit words. Both receiver and the transmitter must agree on the number of data bits, as well as the baud rate. Almost all devices transmit data using either 7 or 8 data bits.

Seven bit words are used to accommodate all upper and lower case text characters in ASCII codes (ASCII characters 0-127). Likewise, using 5 data bits limits the highest possible value to 31. Eight bit words are used to exactly correspond to one byte. By convention, the least significant bit of the word is sent first and the most significant bit is sent last. When communicating, the sender encodes each data word by adding a start bit in front. After the data bits have been transmitted, a stop bit is sent. A stop bit has a binary value of 1 - or a mark state - and it can be detected correctly even if the previous data bit also had a value of 1. This is accomplished by the stop bit's duration. Stop bits can be 1, 1.5, or 2 bit periods in length.

Parity

Besides the synchronization provided by the use of start and stop bits, a parity bit can be inserted between the last bit of the word and the first stop bit for data integrity. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose either even parity, odd parity, mark parity, space parity or none at all.

When even or odd parity is being used, the number of marks (logical 1 bits) in each data word are counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bits just sent is even or odd. With odd parity, the parity bit is logical 0 when the number of mark bits in the preceding word is an odd number. Even parity will set the parity bit to logical 1 when the number of mark bits in the preceding word is an even number. Parity error checking is very rudimentary. While it will tell you if there is a single bit error in the character, it doesn't show which bit was received in error. Also, if an even number of bits are in error then the parity bit would not reflect any error at all. However, a statistical analysis of data communication errors has shown that a single-bit error is much more probable than a multiple bit error in the presence of random noise. Thus, parity is a reliable method of error detection.

Mark parity means that the parity bit is always set to the mark signal condition and likewise space parity always sends the parity bit in the space signal condition. Since these two parity options serve no useful purpose whatsoever, they are almost never used.

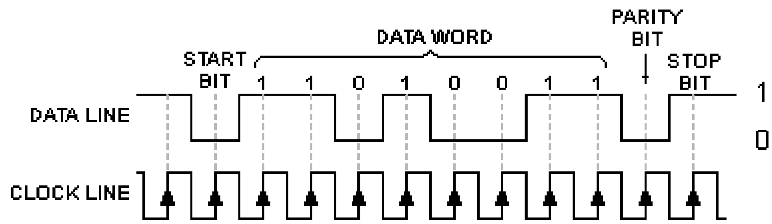


Figure 2: Asynchronous Serial Data Frame (8E1)

In the example above you can see how the data frame is composed of and synchronized with the clock signal. This example uses an eight data bit word with even parity and one stop bit also referred to as an 8E1 setting.

Baud Rate

Another important part of every asynchronous serial signal is the baud rate. The baud rate refers to the signaling rate at which data is sent through a channel and is measured in electrical transitions per second. Therefore, a rate of 9600 baud corresponds to a transfer of 9600 bits per second with a bit period of 104 microseconds (time = 1/9600). The rate at which the data is sent is based on the minimum speed of 300 bps. Faster speeds are all based on the 300 bps rate, you merely double the preceding rate, so the rates are as follows, 600, 1200, 2400, 4800, 9600, 19200, 38400.

Handshaking

In some cases the DCE or receiving unit is a dumb printer with limited memory. In these cases, more data may be sent than the printer can use, causing data to be lost. To prevent this loss of data, Handshaking is used. When the receiving unit is busy or incapable of receiving further data it activates the handshaking, which tells the DTE or sending unit to stop transmitting. When the receiver is ready for more data, it deactivates the handshaking and data transmission continues.

Doran indicators offer both hardware and software handshaking. Hardware handshaking makes the use of CTS or clear to send. With hardware handshaking, the DCE unit controls the flow of data. When this signal is a logical 1, the DTE is permitted to transmit. When the DCE is busy, the CTS line is a logical 0 and the DTE stops sending data.

Software handshaking relies on bi-directional communications to send the XON (^Q) and XOFF (^S) flow characters. When XOFF is received by the DTE or DCE, the transmission is halted until a XON is received.

Synchronous Communication

The second transmission method is synchronous communication, the sending and receiving ends of the communication are synchronized using a clock that precisely times the period separating each bit. By checking the clock, the receiving end can determine if a bit is missing or if an extra bit (usually electrically induced) has been introduced in the stream. One important aspect of this method is that if either end of the communication loses it's clock signal, the communication is terminated. This communications method is not used by PC's and is therefore, not commonly seen.

CABLE LENGTHS

The RS-232C standard imposes a cable length limit of 50 feet. The external environment has a large effect on lengths for unshielded cables. In electrically noisy environments, even very short cables can pick up stray signals. The following chart offers some reasonable guidelines for 24 gauge low capacitance wire under typical conditions. You can greatly extend the cable length by using additional devices like optical isolators and signal boosters. Optical isolators use LEDs and Photo Diodes to isolate each line in a serial cable including the signal ground. Any electrical noise affects all lines in the optically isolated cable equally - including the signal ground line. This causes the voltages on the signal lines relative to the signal ground line to reflect the true voltage of the signal and thus canceling out the effect of any noise signals.

Baud Rate	Shielded Cable Length (feet)	Unshielded Cable Length (feet)
300	4000	1000
1200	3000	500
2400	2000	500
4800	500	250
9600	250	100

NULL MODEM CABLES AND ADAPTORS

If you connect two DTE devices (or two DCE devices) using a straight RS232 cable, then the transmit line on each device will be connected to the transmit line on the other device and the receive lines will likewise be connected to each other. A Null Modem cable or Null Modem adapter simply crosses the receive and transmit lines so that transmit on one end is connected to receive on the other end and vice versa. In addition to transmit and receive, DTR & DSR, as well as RTS & CTS are also crossed in a Null modem connection.

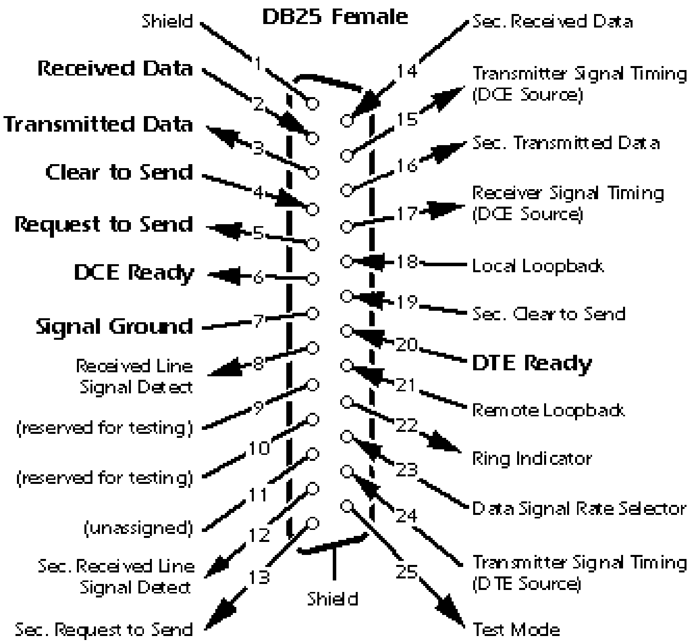
PLUGS AND PINOUTS

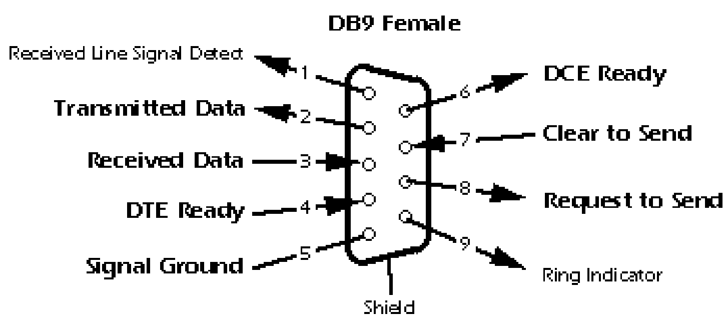
Most equipment using RS-232 serial ports use a DB-9 connector since all you need in asynchronous mode is 9 signals. But take note that the document does specify the amount of pins and their assignment, 20 affected to different signals, three are reserved and two are not affected. Normally the male connector is on the DTE side and the female connector is on the DCE side but, this is not always the case.

RS-232 stands for Recommend Standard number 232 and C is the latest revision of the standard. The serial ports on most computers use a subset of the RS-232C standard. The full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins.

This is a list of all signals specified in the RS232C standard. Each signal is identified by its letters, pin number on a DB-25 and DB-9 connector and its signal name.

Looking Into the DCE Device Connector

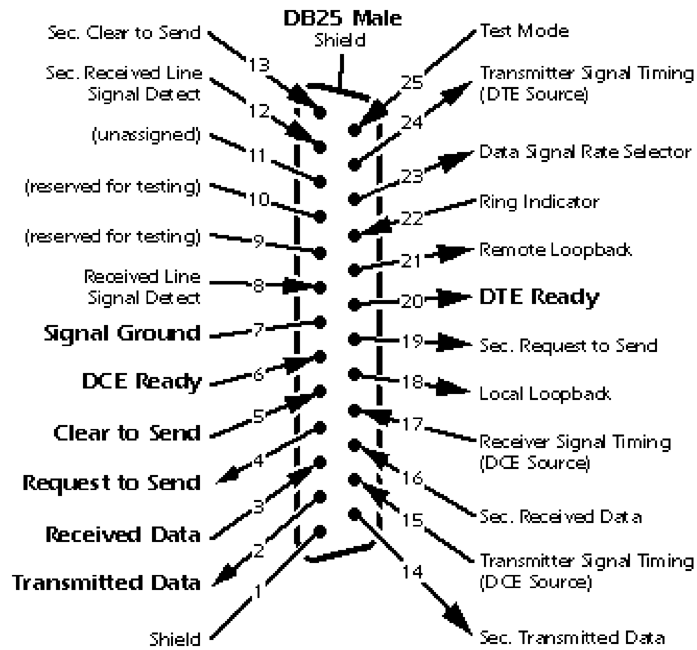


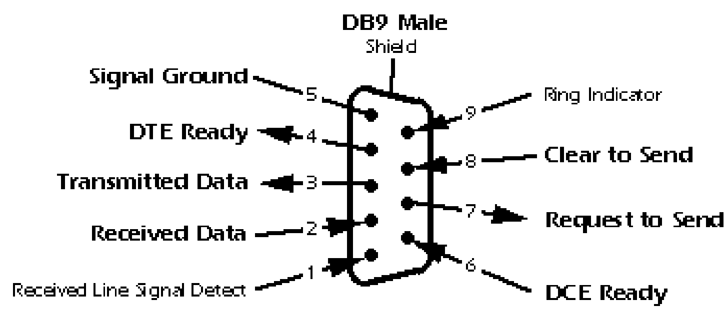


Received by DTE Device

Transmitted from DTE Device

Looking Into the DTE Device Connector





Received by DTE Device

Transmitted from DTE Device

		DCE Device		DTE Device	
SIGNAL NAME	ABBR.	DB-25 Pin #	DB-9 Pin #	DB-25 Pin #	DB-9 Pin #
Shield		1		1	
Received Data	RXD	2	3	3	2
Transmitted Data	TXD	3	2	2	3
Clear To Send	CTS	4	7	5	8
Request To Send	RTS	5	8	4	7
Data Set Ready or DCE Ready	DSR	6	6	6	6
Signal Ground	GND	7	5	7	5
Data Carrier Detect	DCD	8	1	8	1
Test		9		9	
Test		10		10	
Unassigned		11		11	
Secondary Carrier Detect		12		12	
Secondary Request To Send		13		13	
Secondary Received Data		14		16	
Transmitter Signal Timing		15		15	
Secondary Transmitted Data		16		14	
Receiver Signal Timing		17		17	
Local Loopback		18		18	
Secondary Clear To Send		19		19	
Data Terminal Ready or DTE Ready	DTR	20	4	20	4
Remote Loopback		21		21	9
Ring Indicator		22	9	22	
Data Signal Rate Selector		23		23	
Transmitter Signal Timing		24		24	
Test		25		25	

SIGNAL DESCRIPTIONS

Signal Ground
This is the logical ground which is used as a point of reference for all signals received or transmitted. This signal is very important and must be present for all communications.

Transmitted Data
This line is used to transmit data from the DTE to the DCE. It is maintained at a logical 1 state when nothing is transmitted. According to the RS-232 standard, the DTE will start to transmit when a logical 1 is present on all of the following lines:

- Clear To Send
- Data Terminal Ready
- Data Set Ready
- Data Carrier Detect

Please note that many PCs and all Doran indicators do not require these lines to be a logical 1 before transmitting.

Received Data

This circuit is used to receive data from the DCE to the DTE. According to the RS-232 standard, the terminal will start to transmit when a logical 1 is present on all of the following lines:

- Request To Send
- Data Terminal Ready
- Data Set Ready
- Data Carrier Detect

Please note that many PCs and all Doran indicators do not require these lines to be a logical 1 before transmitting.

The standard specifies the output levels as being **-5 to -15 Volts for logical 1** and **+5 to +15 Volts for logical 0**, and the input levels as being **-3 to -15 Volts for logical 1** and **+3 to +15 Volts for logical 0**. This ensures data bits to be read correctly even at maximum lengths between the DTE and DCE, which is specified as 50 feet although you could probably go to much greater distances without any problems (see Cable Lengths above). As you may have noticed, logical 1 are represented by a negative tension and vice versa. There's no particularly good reason for the inversion except that it's the way things have always been done.

Request To Send

On this line, the DTE will send a signal when it wants to receive data from the DCE.

Clear To Send

Here the DCE will send a signal when it's ready to receive data from the DTE.

Data Set Ready or DCE Ready

At a logical level of 1, this line indicates to the DTE that the DCE is ready to send data.

Data Terminal Ready or DTE Ready

When a logical level 1 is sent from the DTE, the DCE can start to send and receive data. When this line passes to logical level 0 the DCE will stop all communications.

Data Carrier Detect

On this line the DCE indicates to the DTE that it has established a carrier with a remote device.

Shield

This line is connected to the power ground of the serial adapter. It should not be used as signal ground. By connecting this line on both sides you make sure that no large currents flow through the signal ground in case of an insulation defect or other defect on either side. On the other side, when two devices are separated by great distances you may not wish to use this signal, because of different ground potential and it is possible that it may carry a substantial current as a ground loop. If the current is strong enough, it will cause electrical interference.

Ring Indicator
This line is used mostly by communications software when the modem is not in "auto answer" mode and will indicate to the software that a remote device is calling. This is signal is optional when not using software that will answer a phone call automatically.

Remote Loopback or Signal quality
Although rarely used, this line serves to indicate to the DTE that the quality of the signal is poor or just not good enough to keep a good connection.

Data Signal Rate Selector
In the case where a modem able of multiple connection rates, the DTE could choose the speed at which it is connected. Usually this line is kept a logical level 0 which selects the highest speed.

Data Signal Rate Selector
This signal is the same as CH but in this case the modem selects the speed at which the DTE communicates.

Timing Circuits
In synchronous mode, it is necessary to have some way to exchange clock signals, here are three timing circuits used in the RS-232 protocol.

> **Transmitter Signal Timing**
> DTE towards DCE (clock part of the DTE)
> DCE towards DTE (clock part of the DCE)
>
> These two circuits are used to synchronize the flow of data. Timing is given by the DTE or DCE but never from both at the same time. Usually data is transmitted to the modem or it's own clock control on the DB circuit.
>
> **Receiver Signal Timing**
> DCE towards DTE (clock part of the DCE)
>
> This circuit is used to synchronize data received from the DTE. The clock signal received on this line indicates to the DTE at which instant to sample the received data on the BB line.

Secondary Lines
All secondary lines serve the same function as primary lines, but serve as error correction for modems and do not apply to RS-232 communications for Doran products.

CONNECTING DORAN INDICATORS VIA RS-232

Doran offers four types of serial cables as standard options: female DB-9 or DB-25 pin with hardware or software handshaking. All standard options have female connectors so connection to a PC will work every time, but connection to a printer depending on manufacturer and style, may require a gender changer. Doran does offer a DB-9 printer cable option called DLP-3, which offers a male connector for use with Eltron™ or other printer styles.

The standard wire color code for printer cables for Doran is as follows:

Wire Color	Doran Function
Red	TXD
White	RXD (CTS in hardware handshaking mode)
Black	GND

Doran indicators do not support any additional control lines for RS-232, which is why we rely upon jumpers in the Doran manufactured serial cables to comply with the RS-232 standard.

DORAN INDICATOR (DCE) TO PC (DTE)

The standard configuration for hardware handshaking for Doran indicators is as follows:

25 Pin Female Connector to DTE Hardware Handshaking			
DTE Function	Pin Number	Doran Wire Color	Doran Function
RXD	3	Red	TXD
GND	7	Black	GND
RTS	4	White	CTS

9 Pin Female Connector to DTE Hardware Handshaking			
DTE Function	Pin Number	Doran Wire Color	Doran Function
RXD	2	Red	TXD
GND	5	Black	GND
RTS	7	White	CTS

Hardware Handshaking

The hardware handshaking mode allows only transmission of data to a PC or printer. The RXD line on the Doran indicator is transformed into CTS when hardware handshaking is selected in the indicator setup menu. Therefore, when in CTS mode, Doran indicators are not able to receive data through the serial port and communications are not bi-directional. Hardware handshaking is necessary, as some computers and printers need to halt transmissions when the input buffer is full to prevent loss of data. When the buffer is full, the PC or printer changes RTS to logic 0, which tells the indicator to stop transmitting data. When Doran indicators are in hardware handshaking mode, CTS will check the RTS circuit for logic 1 before sending data. Most PCs and printers today will not have a problem keeping up with the small amount of data a scale can download. Since it is rare to have the need for a tape printer to provide a printout of every scale reading, (continuous print mode) hardware handshaking is not often used.

25 Pin Female Connector to DTE Software Handshaking			
DTE Function	**Pin Number**	**Doran Wire Color**	**Doran Function**
RXD	3	Red	TXD
TXD	2	White	RXD
GND	7	Black	GND

9 Pin Female Connector to DTE Software Handshaking			
DTE Function	**Pin Number**	**Doran Wire Color**	**Doran Function**
RXD	2	Red	TXD
TXD	3	White	RXD
GND	5	Black	GND

Software Handshaking
Software handshaking allows for full duplex bi-directional communications and is the typical configuration for any Doran serial cable option. In software handshaking output can be disabled by sending XOFF (^S) and can be resumed by sending XON (^Q). The indicator can still receive remote commands when the output is disabled by and XOFF. When output is resumed with the XON command, the indicator will immediately dump any print strings held in the print buffer while the output was disabled. The buffer size is different for each indicator and is usually the size of one print string.

No Handshaking
These cables can also be used with the Handshaking parameter set to off. This will disable bi-directional communications and only allow the scale to transmit and not receive remote commands.

DORAN INDICATOR (DTE) TO PRINTER (DCE)

This is the standard configuration for the DLP-3 and DLP-3A serial cables:

9 Pin Male Connector to DCE Software Handshaking			
DCE Function	**Pin Number**	**Doran Wire Color**	**Doran Function**
RXD	3	Red	TXD
GND	5	Black	GND

This cable represents how simple RS-232 communications can be. Since some printers do not send data to the indicator, it is only necessary for the indicator to transmit data and provide a signal ground.

FIBER OPTIC OPTION

The Guardian XL indicator has a fiber optic option. Fiber optics transmit serial communications through the presence or absence of light. Traditionally with fiber optic systems the presence of light is logic level 1 and the absence of light is logic level 0. This causes a problem for battery powered units as when the communications are idle, by standard, the transmit signal from the scale is logic 1 requiring power to keep the light present almost constantly. To preserve battery life we have inverted the traditional standard where the light is absent most of the time, preserving battery power. Therefore, if a customer wishes to send data via fiber optic cables to a PC, they must purchase Doran's fiber optic to RS-232 conversion box.

STANDARD DORAN REMOTE COMMANDS

All Doran indicators with an RS-232 port allow for remote commands. Remote commands will only operate when the Handshaking parameter is set to Software (see handshaking above). Advanced indicators such as the 8600, 4200 and GuardianXL allow for many more commands than are listed here – refer to their respective technical manuals for further information. The standard remote command set is as follows:

Command	Scale Response
U	Scroll to next unit (lb, kg, g, oz or lb-oz). Note that not all units are always active. The *Convert Select* parameter determines what units will be active. This remote command can be deactivated by disabling UNITS in the *Push Button* parameter menu without deactivating the W and Z remote commands.
W	Print request. This instructs the indicator to print. The scale will not print while in motion. If the *Print on Demand* parameter is set to on the scale will print when the scale becomes stable. If set to off, the print request will be discarded. Remote print requests will operate even when the Data Output parameter is set to an autoprint selection.
Z	Zeros the scale. Scales will not zero while in motion. If the *Zero on Demand* parameter is set to on the scale will print when the scale becomes stable. If set to off, the print request will be discarded.

STANDARD DORAN RS-232 PARAMETERS

The following parameters control the RS-232 port functionality in most Doran scale indicators. Refer to your technical manual for a complete list of these parameters.

Parameter	Description
Data Output (d.o.)	Determines when serial data will be sent. Common settings are: • Transmit on Demand – Scale prints when the PRINT button is pressed. See Print on Demand parameter below. • Continuous Print –In some applications, real time weights are required and the weight is typically sent to a PC or PLC for system controls. In this instance, the continuous print feature is recommended and is the only instance where printing while in motion is allowed by a Doran scale. Each display update is printed, so to increase the number of printed weight samples per second, simply increase the speed of the display's digital filter. In continuous print mode, when the scale is in motion and is printing, the print string will indicate the scale was in motion by typically inserting a MOT into the print string. In legal for trade mode, printing while in motion is disabled. • Auto Print 1 – Print every time the scale becomes stable. This feature must be used with caution as any stable reading will be output the RS-232 port including a zero weight reading. • Auto Print 2 – The fastest and most commonly used way to output stable weight data is to use the Auto Print 2 feature. The Auto Print 2 feature prints the first stable weight and will not print again until the scale is within the zero band. The zero band value is based upon the last pushbutton (displayed) zero +/-1% of capacity. Therefore, if the scale was calibrated to • 25 x 0.005 lb the zero band would be between –0.250 and 0.250 lb around the last pushbutton zero. The zero band allows for the fastest possible weighing when printing as the scale does not have to reach exactly 0.000 lb before another print is allowed execute. The zero band also allows for environments where vibration and air currents are present.
Print on Demand (Pod)	This feature is only relevant when Data Output is set to Transmit on Demand. • When off, the scale will only execute the Print function if the scale is stable, otherwise the Print function will be discarded. • When on, this feature saves the Print function while the scale is in motion. When the scale becomes stable the Print function is then executed.
Output Format (For.)	Defines the string sent when a print is requested. All indicators have a different selection of print strings – see technical manuals for further information.

Parameter	Description
Baud Rate (br.)	Determines baud rate of serial transmission. 9600 is the most common setting.
Data Bits and Parity (d.b.P)	Selects data word length and parity selection. Not all Doran indicators have this parameter available. For indicators without this parameter the setting defaults to the most common serial setting: 8 bits, no parity and 1 stop bit (8n1)
Handshaking Setup (HS)	Selects type of handshaking used by serial communications. See description of handshaking above.

Note: the abbreviation within parens is what the scale will display when this parameter is selected in scale setup mode.

TROUBLESHOOTING

Most calls you will receive will be from dealers setting up their scales for the first time. Configuring a printer or PC system can be frustrating to the end user so be patient. Offer to configure the unit here to save customer irritation (we will do this for free if we have experience with the printer). We will hook up the scale to the printer here, get it working and send it back. We do not know how to configure any label printer but the Eltron™ line. All printers must be sent here with the manual.

If the customer wants to configure their system over the phone try the following sequence of questions:

1. Is the cable manufactured and installed by Doran? If not, the cable is usually the problem in this instance. Every cable that is manufactured at Doran is tested and is good. Open up the connector and double check the wire colors and pin positions. Compare these colors with the other end inside the indicator.
2. Some older cables manufactured by Doran contain jumpers that controlled communications when most devices had little buffer memory. Modern devices have large buffers available and the receiving device never needs to halt communications. Have the customer uninstall the jumpers if they are present.
3. Verify baud rate and parity selections are the same for both devices. A PC requires com port, baud rate and parity settings to be set properly in communications software such as Hyperterminal – double check the com port is set correctly. Dumb printers such as Epson TMU-200 require dip switches to configure bits, baud rate and parity – this will require the printer's manual! Smart printers such as Eltron™ label printers are configured via the RS-232 port, using the proprietary printer label software. Almost all printers have a data dump mode where the current configuration of the baud rate and parity is printed out. Try this first as I have found the dumb printers to be quirky – the dip switches could be set right but the printer does not reflect the correct settings.
4. Verify signal levels are coming out of the indicator properly. Please note that it is extremely rare for Doran indicators to have problems with RS-232 ports. The transmit signal (TXD) from the indicator, when not transmitting, will have a voltage of –7VDC to -10VDC in comparison to the signal ground (GND). The connector housing must be opened to take the proper measurements, as probes do not fit in the female openings of connectors. If the TXD and GND pins fail to have the correct voltage, open the scale and measure the TXD and GND from the scale motherboard. When proper voltages are measured from the motherboard, examine why the cable did not have the proper voltages.
5. Offer to have the customer send the equipment back to Doran for configuration.

ASCII CHARACTER SET

The first 32 values are non-printing **control characters**, such as return and line feed. You generate these characters on the keyboard by holding down the Control key while you strike another key. For example, Bell is value 7, Control plus G, often shown in documents as ^G. Notice that 7 is 64 less than the value of G (71); the Control key subtracts 64 from the value of the keys that it modifies. The control symbols are viewable on some terminal programs. Most terminal programs such as Windows standard Hyperterminal do not display these characters and merely print a space in the symbol's place.

Control Characters

Description	Abbr.	Sym	Oct	Dec	Hex	Control-Key
Null character	NUL		0	0	0	^@
Start of heading	SOH	☺	1	1	1	^A
Start of text	STX	☻	2	2	2	^B
End of text	ETX	♥	3	3	3	^C
End of transmission	EOT	♦	4	4	4	^D
Enquiry	ENQ	♣	5	5	5	^E
Acknowledge	ACK	♠	6	6	6	^F
Bell	BEL	•	7	7	7	^G
Backspace	BS	◘	10	8	8	^H
Horizontal tab	HT	○	11	9	9	^I
Line Feed	LF	◙	12	10	a	^J
Vertical tab	VT	♂	13	11	b	^K
Form Feed	FF	♀	14	12	c	^L
Carriage Return	CR	♪	15	13	d	^M
Shift Out	SO	♫	16	14	e	^N
Shift In	SI	☼	17	15	f	^O
Data link escape	DLE	►	20	16	10	^P
XON	DC1	◄	21	17	11	^Q
Device control 2	DC2	↕	22	18	12	^R
XOFF	DC3	‼	23	19	13	^S
Device control 4	DC4	¶	24	20	14	^T
Negative acknowledge	NAK	§	25	21	15	^U
Synchronous idle	SYN	-	26	22	16	^V
End transmission block	ETB	↨	27	23	17	^W
Cancel line	CAN	↑	30	24	17	^X
End of medium	EM	↓	31	25	19	^Y
Substitute	SUB	→	32	26	1a	^Z
Escape	ESC	←	33	27	1b	^[
File separator	FS	∟	34	28	1c	^\
Group separator	GS	↔	35	29	1d	^]
Record separator	RS	▲	36	30	1e	^^
Unit separator	US	▼	37	31	1f	^_

Char	Oct	Dec	Hex	Description
SP	40	32	20	Space
!	41	33	21	Exclamation mark
"	42	34	22	Quotation mark
#	43	35	23	Cross hatch
$	44	36	24	Dollar sign
%	45	37	25	Percent sign
&	46	38	26	Ampersand
`	47	39	27	Closing single quote
(	50	40	28	Opening parentheses
)	51	41	29	Closing parentheses
*	52	42	2a	Asterisk
+	53	43	2b	Plus
,	54	44	2c	Comma
-	55	45	2d	Hyphen, dash, minus
.	56	46	2e	Period
/	57	47	2f	Slant
0	60	48	30	Zero
1	61	49	31	One
2	62	50	32	Two
3	63	51	33	Three
4	64	52	34	Four
5	65	53	35	Five
6	66	54	36	Six
7	67	55	37	Seven
8	70	56	38	Eight
9	71	57	39	Nine
:	72	58	3a	Colon
;	73	59	3b	Semicolon
<	74	60	3c	Less than sign
=	75	61	3d	Equals sign
>	76	62	3e	Greater than sign
?	77	63	3f	Question mark
@	100	64	40	At-sign
A	101	65	41	Uppercase A
B	102	66	42	Uppercase B
C	103	67	43	Uppercase C
D	104	68	44	Uppercase D
E	105	69	45	Uppercase E
F	106	70	46	Uppercase F
G	107	71	47	Uppercase G
H	110	72	48	Uppercase H
I	111	73	49	Uppercase I
J	112	74	4a	Uppercase J
K	113	75	4b	Uppercase K
L	114	76	4c	Uppercase L
M	115	77	4d	Uppercase M
N	116	78	4e	Uppercase N
O	117	79	4f	Uppercase O

Char	Oct	Dec	Hex	Description
P	120	80	50	Uppercase P
Q	121	81	51	Uppercase Q
R	122	82	52	Uppercase R
S	123	83	53	Uppercase S
T	124	84	54	Uppercase T
U	125	85	55	Uppercase U
V	126	86	56	Uppercase V
W	127	87	57	Uppercase W
X	130	88	58	Uppercase X
Y	131	89	59	Uppercase Y
Z	132	90	5a	Uppercase Z
[	133	91	5b	Opening square bracket
\	134	92	5c	Reverse slant
]	135	93	5d	Closing square bracket
^	136	94	5e	Caret
_	137	95	5f	Underscore
`	140	96	60	Opening single quote
a	141	97	61	Lowercase a
b	142	98	62	Lowercase b
c	143	99	63	Lowercase c
d	144	100	64	Lowercase d
e	145	101	65	Lowercase e
f	146	102	66	Lowercase f
g	147	103	67	Lowercase g
h	150	104	68	Lowercase h
i	151	105	69	Lowercase i
j	152	106	6a	Lowercase j
k	153	107	6b	Lowercase k
l	154	108	6c	Lowercase l
m	155	109	6d	Lowercase m
n	156	110	6e	Lowercase n
o	157	111	6f	Lowercase o
p	160	112	70	Lowercase p
q	161	113	71	Lowercase q
r	162	114	72	Lowercase r
s	163	115	73	Lowercase s
t	164	116	74	Lowercase t
u	165	117	75	Lowercase u
v	166	118	76	Lowercase v
w	167	119	77	Lowercase w
x	170	120	78	Lowercase x
y	171	121	79	Lowercase y
z	172	122	7a	Lowercase z
{	173	123	7b	Opening curly brace
|	174	124	7c	Vertical line
}	175	125	7d	Closing curly brace
~	176	126	7e	Tilde
DEL	177	127	7f	Delete, cross-hatch box