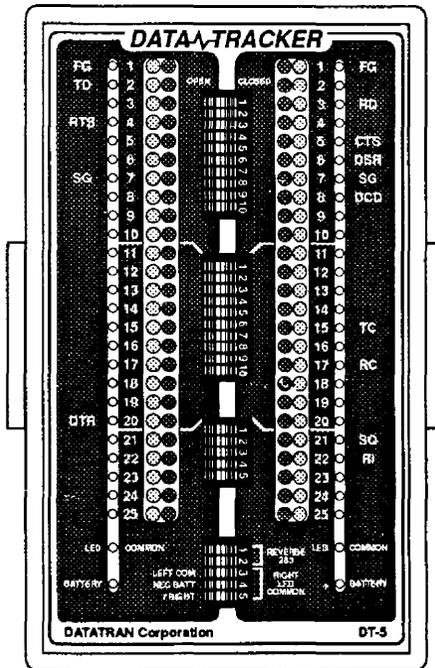


# DATATRACKER DT-5

## OPERATOR'S MANUAL AND INTERFACING GUIDE



Version 4.0  
1-10-90

**DATATRAN Corporation**  
355 Yuma Street  
Denver, Colorado 80223  
(303)778-0870



# Congratulations

You are now the owner of the most advanced RS-232 interfacing device on the market today.

The DATATRACKER™ has been designed to simplify RS-232 interfacing, troubleshooting and cable testing in a dramatically new and time efficient manner. Finally, you will be able to see at a glance the status of every signal in your interface without any special patching. The 100 high efficiency LEDs indicate High, Low, Clocking or Off signals on both sides of the interface for all 50 lines simultaneously. Since there is no longer anything left to the imagination, much of the frustration will be taken out of RS-232.

Datatran Corporation takes great pride in our position as a "market leader" in this fast growing industry. Our reputation for innovative, dependable, quality oriented products continues to grow because your satisfaction is our goal. Your help is needed in achieving this goal however, and we would therefore encourage your comments and suggestions on how we can improve our products. Your input is important to us!

It is our sincere commitment to continue to provide the finest, most reliable products, at the best price, backed with service and support surpassed by no one in this expanding industry. Thank you for selecting Datatran.

## **DATATRAN Corporation**

355 Yuma Street  
Denver, Colorado 80223  
(303) 778-0870

Copyright © 1990 by Datatran Corporation

All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language, in any form or by any means, electronic, magnetic, mechanical, optical, manual or otherwise, without the prior written permission of Datatran Corporation.

Even though Datatran Corporation has tested this product and reviewed the documentation, **Datatran makes no warranty or representation, either express or implied, with respect to its performance, merchantability or fitness for a particular purpose. In no event will Datatran Corporation be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in this product or its documentation,** even if advised of the possibility of such damages. In no case shall Datatran Corporation's liability exceed the cost of this product.

Information contained in this document is subject to change without notice or obligation on the part of Datatran Corporation.

This is version 4.0 of the Instruction Manual for the Datatracker DT-5

IBM is a registered trademark of IBM Corporation.  
MS-DOS is a registered trademark of Microsoft Corporation.  
Centronics is a registered trademark of Genicom Corporation.

Datatracker™ is a trademark of Datatran Corporation.

Printed in the United States of America.  
First Printing: January 1990

Part Number: 621-0015

## **What This Manual Contains**

This manual was designed to be more than just an operator's manual for the Datatracker; it is also an RS-232 Interfacing and Troubleshooting Guide. Included are numerous procedures, comments, tricks and techniques which should assist both the novice as well as the experienced technician in becoming more proficient at understanding RS-232. The information covered in this manual is intended to help you to understand and use the Datatracker to its fullest potential. The following is a summary of what each chapter contains.

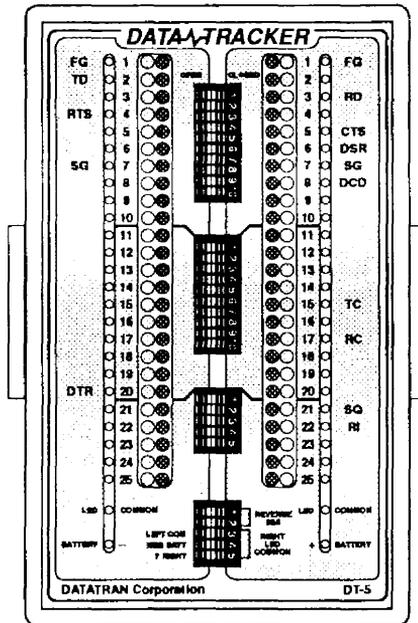
- **Chapter 1** "Introduction" tells you what the Datatracker is and what it can do. This chapter will help you identify and understand the Datatracker's features, parts and accessories.
- **Chapter 2** "Self Test" leads you through the process of testing your Datatracker for proper operation. This procedure will help you gain familiarity with the Datatracker as well as some of the terminology used throughout this manual.
- **Chapter 3** "RS-232 Background Information" reviews the history and the basics of the RS-232 standard. A good understanding of these fundamentals is important in understanding the principles of interfacing.

- **Chapter 4** "Using The Datatracker For Interfacing" details our unique "8 Step Interfacing Method" that can be used on any serial RS-232 data processing devices.
- **Chapter 5** "Cable Testing" will take you through the procedure to test any RS-232 cable in seconds. Also covered is the use of two optional accessories for single ended cable testing.
- **Chapter 6** "Troubleshooting" reviews many of the tricks and techniques that you can use with the Datatracker to find the cause of an interface problem.
- **Appendix** Included are: an ASCII Character Chart, two RS-232 Signal Charts, a condensed version of the "8 Step Interfacing Method", a Schematic Diagram for the Datatracker DT-5, two sample interfaces, and a blank "Interface Documentation Sheet".

We hope that this information, as well as your Datatracker, prove useful in your mastery of RS-232.

## Introduction

A volt meter, ohm meter, ammeter or oscilloscope are all test devices used to provide information about electronic circuits. Whether you are designing, repairing or modifying electronic equipment, these devices may prove invaluable in helping you to accomplish your task. After you become familiar with your Datatracker and its uses, it too may well become an invaluable addition to your other test equipment.



The primary function of an RS-232 break-out box is to "break-out" the lines from a RS-232 interface cable so you can easily connect, or not connect, each individual line to any other line. This function also gives you the ability to isolate a single line of an interface, isolate a group of lines at a time, or isolate all of the lines of one device from the other device.

# Introduction

## Chapter 1

---

It is this ability to selectively connect and disconnect lines when interfacing, and selectively isolate lines when troubleshooting, that makes the break-out box an invaluable piece of test equipment. The reason for using a break-out box is to provide you with the information necessary to aid you in determining how an interface cable should be configured for two devices to communicate properly. Just as with other test equipment, it is up to you to interpret, and then act upon the information provided.

At this point we would like to mention that the Datatracker differs from many of the other RS-232 break-out boxes on the market. By using 100 LEDs, every possible condition and combination of signals is constantly displayed on the Datatracker, leaving nothing to the imagination. By making all of this information available at the same time, the Datatracker is capable of giving you more answers than questions. These are some of the areas in which the Datatracker can assist you:

- Proper output device/port/connector selection
- Proper hardware/software function
- DTE/DCE identification
- Determining required initialization signals
- Determining hardware handshake lines
- Determining software handshake availability
- Verifying proper voltage levels
- Troubleshooting data communications equipment
- Allowing easy access to the interface by other test equipment
- RS-232 cable testing

All break-out boxes do have limitations however. They will not tell you:

- Baud rate
- Number of data bits
- Number of stop bits
- Parity
- Software handshake character

These parameters of the interface are just as important as the

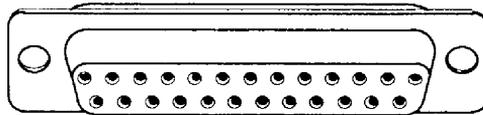
wiring, but need to be determined in some other fashion. You might be able to look them up in the set-up or configure sections of various programs. The Mode command in DOS will allow you to define these parameters but it won't tell you what your system is presently set at. Finding and interpreting DIP switch settings using equipment manuals is also a common ritual. As a final possibility, we would like to suggest using the AutoFox. Unlike the Datatracker, which deals primarily with the wiring aspects of the interface, the AutoFox focuses on the actual format and content of the data. Essentially, the AutoFox is a hand held computer optimized for solving a variety of communications problems quickly and easily.

## Unpacking

Your Datatracker was packed and shipped in a specially designed carrying case. Inside the large zipper compartment you will find the Datatracker itself. The smaller zipper compartment contains two 4 foot extension cables and 4 Mini-Hook clip leads. Also included with the Datatracker is a Warranty Registration Card and this Operator's Manual and Interfacing Guide.

## Input/Output Connections

The Datatracker was designed to connect two RS-232 devices using DB-25 type connectors. On the LEFT side of the Datatracker we provide a "female" DB25S connector and on the RIGHT side a "male" DB25P connector. This configuration allows the Datatracker



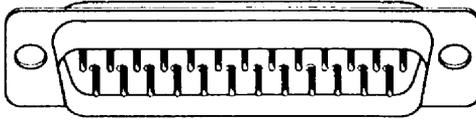
**DB 25S FEMALE CONNECTOR**

to be inserted "in-line" between two RS-232 devices using the two cables supplied with the Datatracker. All 25 lines from the female connector are routed to the corresponding turret pins, LEDs and

# Introduction

## Chapter 1

---



DB 25P MALE CONNECTOR

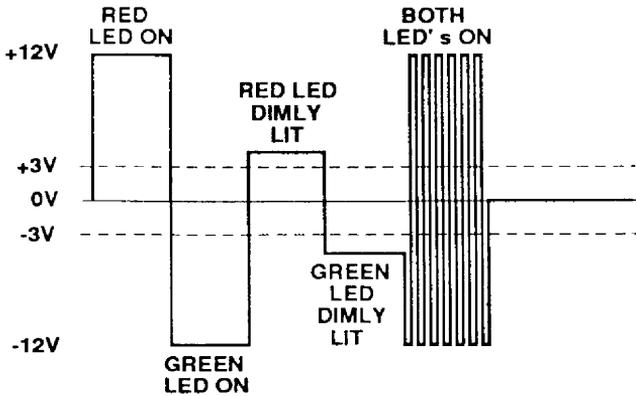
switches on the LEFT side of the Datatracker. The male connector is routed to the RIGHT set of turret pins, LEDs and switches.

### Display

The LEDs (Light Emitting Diodes) on the Datatracker are used to indicate whether a specific line is High, Low, Clocking, or Off (approximately 0 volts). Each of the 50 lines being monitored has this capability. This means that with the Datatracker, all activity on the interface is displayed immediately. There is no need to "patch-in" spare LEDs. This feature alone can save hours of frustration while trying to work with the far more limited display of other types of break-out boxes.

The display on your Datatracker consists of 100 high efficiency LEDs, divided into two groups of 25 red and 25 green LEDs. These LEDs correspond to the DB25 connectors located on either side of the Datatracker. The status of a single line would be indicated by a red LED "on" for a high signal (+3 to +12 volts), or a green LED "on" for a low signal (-3 to -12 volts). The status of a line could also be indicated when both the red and green LEDs are flashing (or seemingly both on) for data, or when both LEDs are "off" for a disconnected line or a line at approximately zero volts.

By using high efficiency LEDs the Datatracker is able to function as a passively designed break-out box. This means that the LEDs derive their power directly from the RS-232 signal lines themselves, without the need for batteries. By virtue of this design, the LED intensity is directly related to the voltage of the signal being monitored. A dim LED will therefore be an indication of a potential problem.



## Input Overload

**CAUTION:** Do not exceed the input limits of your Datatracker. The LED display has been designed to withstand a maximum of  $\pm 50$  Volts. The other components, including the switches and connectors have been designed to handle a maximum current of 250 mA (milli-Ampere). These limitations are well within the specifications for RS-232 and should pose no problem under normal circumstances.

## Switches

There are four groups of slide switches located down the middle of the Datatracker. The top three groups, totaling 25 switches correspond to the 25 RS-232 signal lines. They allow the 2 sides of the Datatracker to be either connected or disconnected on a straight through line-by-line basis. When a switch is positioned to the LEFT, it is an open circuit, or "OFF". Conversely, when a switch is positioned to the RIGHT, it is a closed circuit, or "ON".

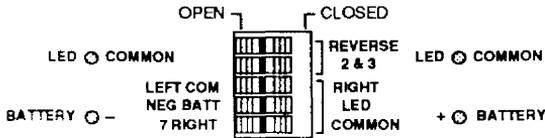
The bottom group of 5 switches are special option switches. The first two option switches are called "REVERSE 2 & 3" and, as the name implies, are used to reverse or cross connect signal lines #2 and #3. The next three option switches all use the "RIGHT LED COMMON" as a common connection point. Option switch #3 ("LEFT COM") will connect the left "LED COMMON" turret pin to the right

# Introduction

## Chapter 1

---

“LED COMMON” turret pin. Option switch #4 (“NEG BATT”) will connect the negative battery turret pin (“BATTERY -”) to the right “LED COMMON” pin. And finally, option switch #5 (“7 RIGHT”) will connect the #7 turret pin (“SG”) on the right side to the right “LED COMMON” pin. The use of these switches will be covered in Chapter 4 of this manual.



### Turret Pins

There are a total of 54 turret pins which have been specifically designed for the Datatracker. They are machined out of a special beryllium copper alloy for durability and then gold plated for corrosion resistance. 50 of these pins provide easy access to each of the 50 RS-232 signal lines. They are divided into two groups of 25 pins each which correspond to the DB25 connectors located on either side of the Datatracker.

The pins just below the 25th line on either side of the Datatracker are marked “LED COMMON”. These pins are connected to the “LED COMMON” bus on each side of the Datatracker and are available for the purpose of connecting the ground reference for the LEDs. Most break-out boxes already have the ground reference selected by hard-wiring their internal “LED COMMON” to line #7 (Signal Ground). The Datatracker’s “LED COMMON”, on the other hand, is not hard-wired to line #7 but is floating and should be connected as needed. Interfacing and cable testing, for instance, will each require a different ground reference.

The bottom two pins are designated “BATTERY -” (Negative) and “BATTERY +” (Positive). These pins allow access to the Datatracker’s internal 9 volt battery. A current limiting resistor is wired in series with the battery in order to minimize the effect of an inadvertent short circuit of the battery turret pins.

## **Battery**

A standard 9 volt battery is provided for tying lines High or Low if the appropriate signal is not already available in the interface. The battery is also used to provide a signal source for cable testing and self diagnostic procedures. The battery compartment is accessed through the latched door on the underside of the unit.

## **Protective Cover**

The Datatracker comes with a detachable protective cover. The inside of this cover contains a condensed version of our 8-Step Interfacing Method which is discussed in detail in Chapter 4. When the cover is not needed it can be snapped onto the back of the Datatracker so it won't get misplaced. Otherwise, the cover should be used whenever possible to protect the turret pins and switches from being damaged.

When new, the cover might seem to be a little difficult to take off. This will become much easier after a break-in period of several dozen removals. In addition, you should find that pulling up on one corner much easier than trying to pull the cover straight off.

## **Extension Cables**

Two 4 foot 25 conductor 'Y' extension cables are included with the Datatracker. One cable has a male connector at one end and both male and female connectors at the other end allowing it to plug into the left side of the Datatracker. The other 'Y' cable is intended to plug into the right side of the Datatracker. We recommend that you use these two 25 conductor cables when doing new interfaces in order to be sure that all 25 lines of both devices are available and monitored.

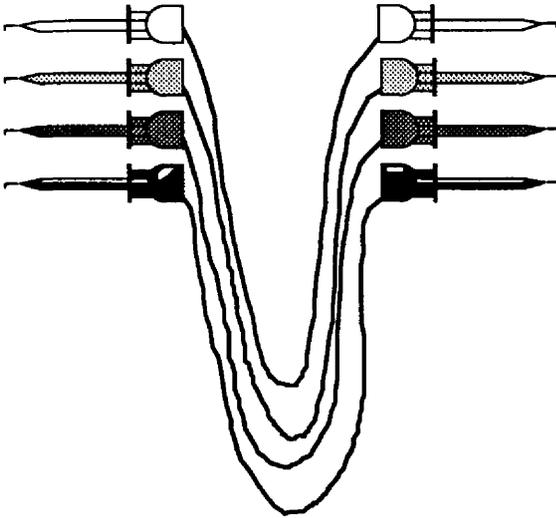
## **Clip Leads**

Also included are 4 Mini-Hook clip leads (in red, black, green & yellow) which have been selected for their ease of use, durability and availability. In conjunction with the turret pins they provide an easy method for cross connecting lines or tying two or more lines together.

# Introduction

## Chapter 1

---



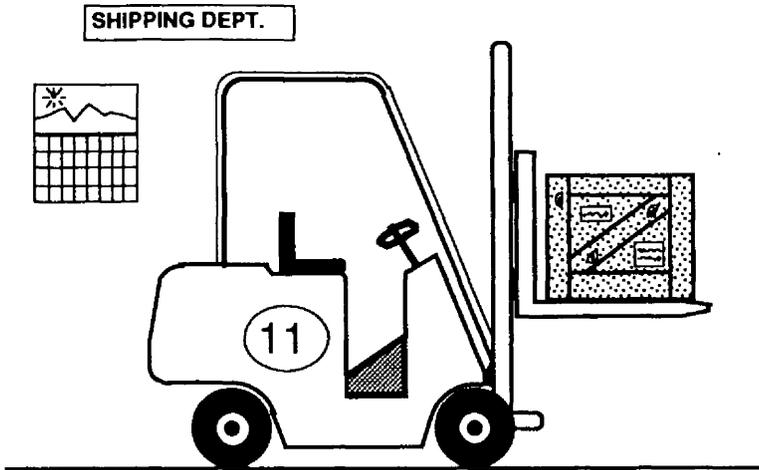
They are normally used during interfacing, cable testing and checking the Datatracker for proper operation.

### Carrying Case

Your Datatracker was supplied with a padded vinyl carrying case. This case was designed to provide maximum portability for the Datatracker as well as to protect the unit from damage. As previously mentioned, the large compartment is used to contain the Datatracker. In order to prevent the turret pins from snagging the material inside the pouch, the Datatracker should always be inserted with its protective cover in place. The smaller compartment was designed to carry the 4 foot extension cables and the Mini-Hook clip leads.

### Problems?

If you should find anything wrong with your shipment, or if the Datatracker was damaged during shipping, please contact your dealer immediately. You might want to retain the packaging material supplied with the Datatracker until you have completed the self



diagnostic procedures listed in the next chapter. If you need to ship your Datatracker anywhere, for any reason, we suggest you use the original shipping container and supplied carrying case. If the original shipping container is not available, please make every effort to supply sufficient packing material and a strong enough container to ensure that no damage is done to the Datatracker.

## **Customer Support**

Thank you for purchasing the Datatracker. We care about our customers and we're very dedicated to providing you with the professional service and support that you deserve. We would therefore like to ask you to HELP US HELP YOU by sending in your WARRANTY REGISTRATION CARD.

There are several good reasons why you should send in your Registration Card. First of all, registered owners are entitled to FREE Technical Support. Our Technical Support people are available Monday thru Friday from 8:00 a.m. to 5:00 p.m. Mountain Time by calling (303) 778-0870. Secondly, we would like to keep you posted on new product developments at Datatran and would very

# Introduction

## Chapter 1

---

much welcome your input in this area. To take advantage of these benefits, just fill out the Registration Card and mail it to us today. We'll even pay the postage. Thank-you!

If you have any questions or problems with your Datatracker that this manual doesn't address, please don't hesitate to contact us. We will be able to help you more quickly however, if you would collect all the information you can regarding the problem. Make notes of any equipment that the Datatracker was connected to as well as the interface configuration.

Many times we can solve your problem while on the phone. If however, it appears that your Datatracker is defective, the Technical Support person will refer you to our Customer Service department for a Return Material Authorization number. Pack your Datatracker in a well padded box along with your name, company, address, telephone number and RMA number and ship prepaid to:

DATATRAN Corporation  
355 Yuma Street  
Denver, Colorado 80223

If the problem with your Datatracker is covered by the warranty, we will either repair it, or at our option, replace it and return it to you free of charge. If the problem is not covered under the warranty, then we will call you regarding the necessary repairs.

If you have moved to a new address or have changed companies, please give our Customer Service department a call with the new information. We'll update our records so that you won't miss out on any important information.

One final note. We're always looking for ways to improve our service, our support or our products. For example, if there is something in the documentation that needs further clarification, please let us know. In addition, if there is a feature or utility that you feel should be added, enhanced or altered, we want to hear from you. Our customers are our most important resource.

## **Datatracker DT-5 Specifications**

- 100 LED's (50 Red & 50 Green)
  - High +3V to +12V (Red on)
  - Low -3V to -12V (Green on)
  - Off +3V to -3V (Both LEDs off)
  - Passive design (1.75 mA/LED @ 10 Volts)
  - Input impedance (5.7 K $\Omega$ /line)
- 54 easy access turret pins
- 30 miniature slide switches
- 2 D-Type 25 pin connectors  
(1 male, 1 female)
- 1 standard 9 volt battery
- Weight: 11 oz. (294 gm) – Datatracker only  
1 lb. 9 oz. (711 gm) – complete kit
- Dimensions: 6" x 3.75" x 1.5" (15 x 9.5 x 4 cm) – Datatracker only

A copy of the Datatracker DT-5 schematic is included in the Appendix for additional information.

# Introduction

## Chapter 1

---

### Optional Accessories

The following list of optional accessories are also available for use with the Datatracker. Please contact either your dealer or our Customer Service department for more information.

- AUTOFOX™ (AF-100)  
hand-held diagnostic device
- MINITRACKER PLUS™ (MT-2)  
cable testing remote read-out
- CABLE TEST ADAPTER™ (CTA-7)  
for single ended cable testing
- PULSE TRACKER™ (PT-1)  
dual pulse trap
- MICROPATCH™ (MP-1, MP-2, MP-3)  
reusable patching adapters
- GENDERMATE® (GM-MM)  
male to male DB25 adapter
- GENDERMATE® (GM-FF)  
female to female DB25 adapter
- Additional Mini-Hook clip leads
- Additional 25 conductor 4 foot 'Y' cable  
(M to M/F) or (F to M/F)
- 25 pin Universal 4 foot cable  
(M/F to M/F)
- 25 pin Universal 8 foot cable  
(M/F to M/F)

## Self Test

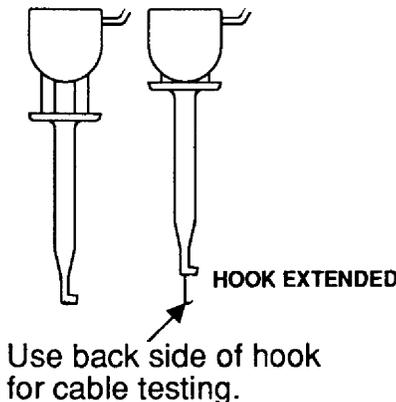
By following this simple procedure, you can completely test the Datatracker for proper operation. This Self Test procedure should be performed when you first receive your Datatracker, since it will help you gain familiarity with it. This procedure may also be performed any time the integrity of the Datatracker is in question.

The following phases should be completed as outlined in order to avoid any confusion since each phase builds upon the successful completion of the previous phases. If a problem is encountered, a volt/ohm meter may be used to test the battery, the clip leads, the cables and the overall continuity of the Datatracker. Feel free to refer to the Datatracker DT-5 schematic located in the Appendix while going through this Self Test procedure.

### Phase 1

#### LED Test, Switches Open

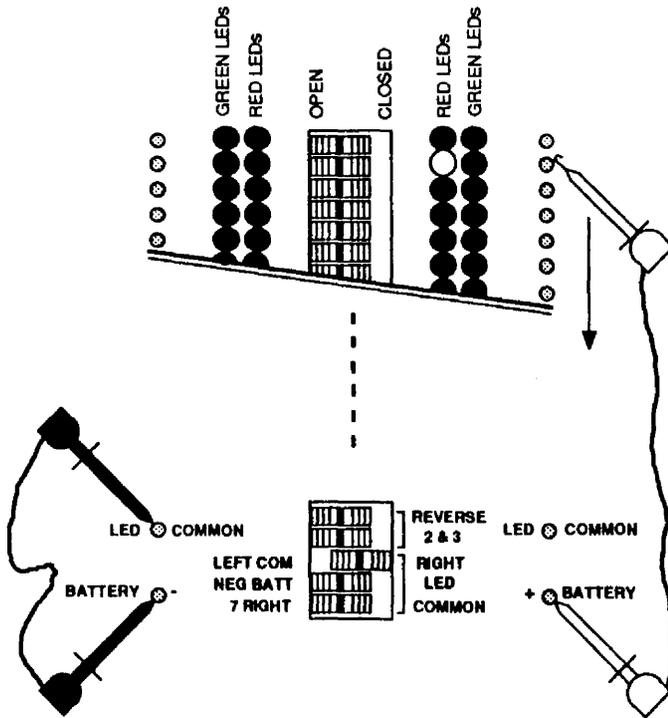
1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open (to the left).
2. Close option switch #3 ("LEFT COM"). This connects the two LED Commons together.
3. Using the black clip lead, connect the left "LED COMMON" turret pin to the negative battery turret pin ("BATTERY -").



# Self Test

## Chapter 2

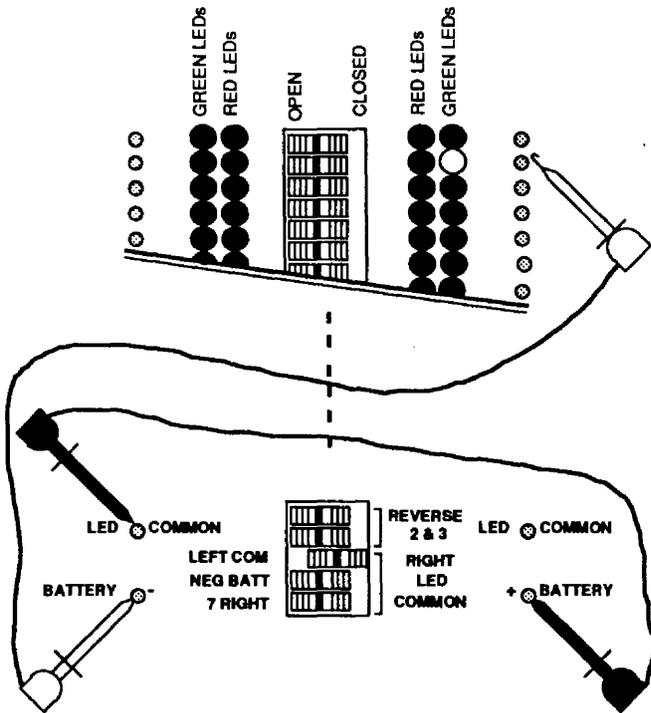
4. Connect one end of the yellow clip lead to the positive battery turret pin ("BATTERY +").



5. Using the free end of the yellow clip lead, extend the brass hook and, one by one, touch each of the 25 numbered turret pins on the right side with the clip lead's hook. It is easier to touch the back side of the hook to the pin to keep it from "hooking" itself onto the pin. As you touch each pin, the corresponding red LED, and only that LED, should light up. This process is called "scanning".
6. When you have finished the right side repeat the process with the left side.

# DATATRACKER

The Professional Breakout Box



7. To check the green LEDs, reverse the yellow and black clip leads on the battery. The black clip lead should now be on the "BATTERY +" pin and the yellow clip lead should be on the "BATTERY -" pin.
8. Repeat steps 5 and 6 making sure that the green LEDs light this time instead of the red.
9. Open the "LEFT COM" switch thereby disconnecting the LED Commons. Scan the pins on the right side again and observe that the LEDs should now fail to light since the right side "LED COMMON" is no longer referenced to the battery.

# Self Test

## Chapter 2

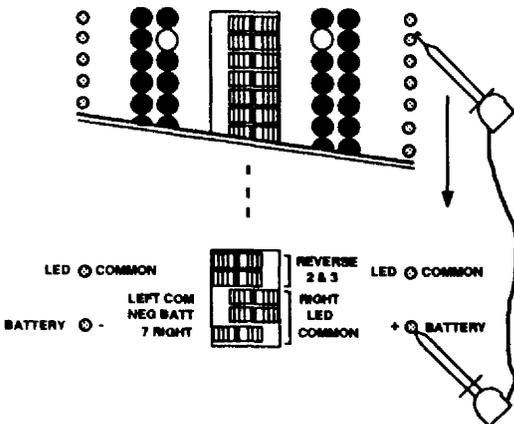
---

**Problems:** If any LED fails to light it is either burned out or an open circuit exists somewhere. If two or more LEDs light at the same time, a short circuit exists. If the LEDs on the right side only fail to light (and the "LEFT COM" switch is fully closed) then either the "LEFT COM" switch is defective or an open circuit exists. If all the LEDs light dimly, then the battery is weak and needs replacement. If none of the LEDs light, then either the battery is 'dead', a clip lead is broken, or an open circuit exists.

This completes the first phase of the Self Test procedure. You have just tested each of the red and green LEDs, each turret pin connection, each DB25 line switch in the OFF position for 'short' circuits, the "LEFT COM" switch for continuity, the yellow and black clip leads and the battery.

### Phase 2 Switch Continuity Test

1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Close the "LEFT COM" option switch as well as option switch #4 ("NEG BAT"). This connects the negative battery terminal to both LED commons.
3. Close all 25 RS-232 line switches.



4. Connect one end of the yellow clip lead to the positive battery turret pin.
5. Using the free end of the yellow clip lead, scan each of the 25 numbered turret pins on the right side. As you do so, the corresponding red LED on the right side and the corresponding red LED on the left side should light up. No other LEDs should be on at this time.
6. Open the "NEG BATT" switch thereby disconnecting the battery reference for both LED Commons. Scan the pins on the right side again and observe that no LEDs should now light.

**Problems:** If the red LED on the left side fails to light (and the corresponding switch is fully closed), then either the switch is defective or an open circuit exists somewhere. If neither LED lights (and the "NEG BATT" switch is fully closed) then either the "NEG BATT" switch is defective or an open circuit exists.

There is no need to scan the other side or the green LEDs since we are only checking for switch continuity in this phase. This completes the second phase of the Self Test procedure.

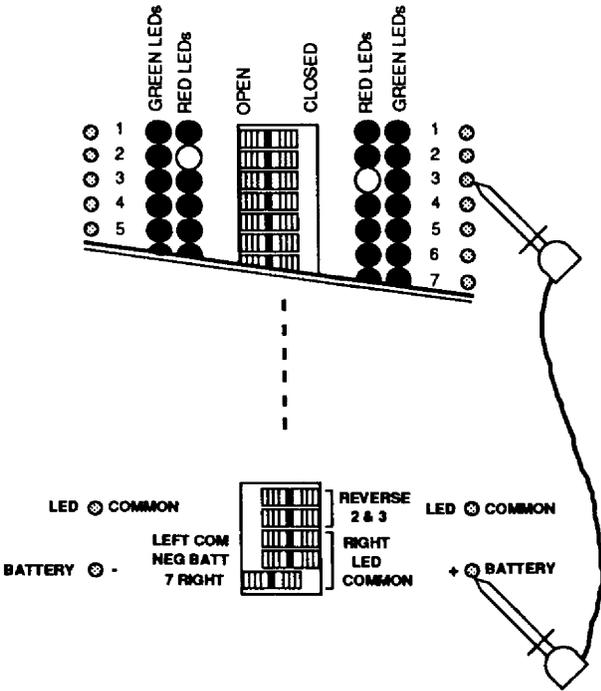
### **Phase 3**

#### **The "REVERSE 2 & 3" Switches**

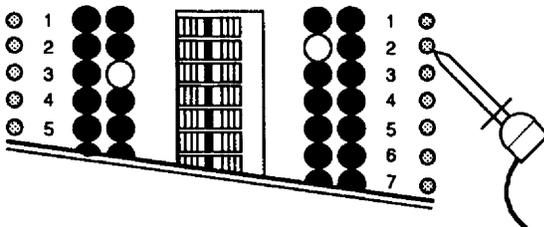
1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Close the "LEFT COM" and "NEG BATT" switches.
3. Connect one end of the yellow clip lead to the positive battery pin.
4. Close option switch #1 and option switch #2 ("REVERSE 2 & 3").
5. Connect the free end of the yellow clip lead to pin #3 on the right side of the Datatracker. Observe that the red LED corresponding to line #3 on the right side and the red LED corresponding to line #2 on the left side should be on.
6. Open option switch #1 and you should see the left #2 red LED go off.

# Self Test

## Chapter 2



7. Disconnect the clip lead from pin #3 and re-connect it to pin #2 on the right side. The #2 red LED on the right side and the #3 red LED on the left side should now be on.



8. Open option switch #2 and observe that the #3 red LED on the left side should now be off.

Problems: If the #2 red LED on the left side failed to light (and option switch #1 is fully closed) then either option switch #1 is defective or an open circuit exists somewhere. The same would hold true for the #3 red LED on the left side and option switch #2.

This completes the procedure for testing the "REVERSE 2 & 3" switches.

## **Phase 4**

### **The "7 RIGHT" Switch**

1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Close the "NEG BATT" switch.
3. Use a clip lead to connect pin #7 ("SG") on the right side to the positive battery pin. The #7 red LED on the right side should now be on.
4. Close option switch #5 ("7 RIGHT") and observe that the #7 red LED should go off. **Note:** Do not leave the Datatracker in this configuration for long since it will quickly drain the battery.

Problems: If the #7 red LED failed to go off (and option #5 is fully closed) then either the "7 RIGHT" switch is defective or an open circuit exists somewhere.

This completes the procedure for testing the "7 RIGHT" switch.

## **Phase 5**

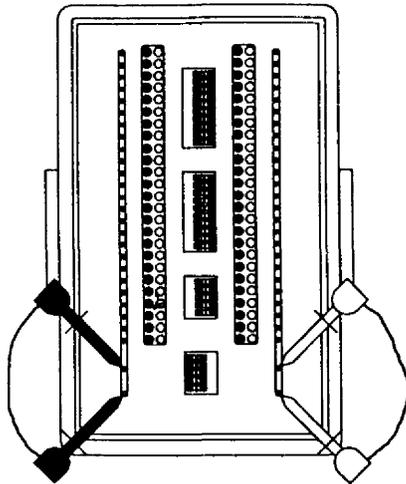
### **Relative LED Intensity Check**

1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Close all 25 RS-232 line switches.
3. Using the black clip lead connect the left "LED COMMON" pin to the negative battery pin ("BATTERY -").
4. Using the yellow clip lead connect the right "LED COMMON" pin

# Self Test

## Chapter 2

---



to the positive battery pin.

5. Observe that all 25 green LEDs on the right side and all 25 red LEDs on the left side should now be on at the same time. Do not be concerned that they appear noticeably dimmer than in previous tests. This is normal. The important thing to observe here is that they should appear relatively equal in intensity. It is very difficult for a marginally defective LED to go unnoticed at this low intensity level.
6. To test the other 50 LEDs, reverse the yellow and black clip leads on the battery.
7. Now observe that the 25 red LEDs on the right side and the 25 green LEDs on the left side should be on.

**Problems:** If any of the LEDs appear significantly dimmer than the rest, then it is defective. Over-voltage transients can cause this type of defect.

This completes this phase of the Self Test and demonstrates a quick way to test 50 LEDs at once for relative intensity.

## **Phase 6**

### **Extension Cable Test**

1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Plug the 'male' end of the male 'Y' cable (supplied with the Datatracker) into the DB25S connector on the left side.
3. Plug the 'female' end of the female 'Y' cable (also supplied) into the right side of the Datatracker.
4. Connect the 2 'Y' cables together using one set of male and female connectors (leave the remaining set open).
5. Close the "LEFT COM" and "NEG BATT" switches.
6. Connect one end of the yellow clip lead to the positive battery pin.
7. Using the free end of the yellow clip lead, scan each of the 25 numbered turret pins on the right side. As you do so, the corresponding red LED on the right side and the corresponding red LED on the left side should light up. No other LEDs should be on at this time.
8. Disconnect the 2 'Y' cables from each other and re-connect them using the other male and female pair.
9. Repeat step 7.

**Problems:** If any LEDs on the left side fail to light then either one of the cables is defective or an open circuit exists somewhere. If any LEDs other than the two corresponding LEDs light then a short circuit or crossed wiring exists somewhere. Try testing the cables individually in an effort to isolate the problem.

Since all 25 line switches are open in this phase, continuity from the right side to the left side is provided by the extension cables. You have just tested the extension cables for open wires, shorted wires and crossed connections, as well as the two DB25 connectors on either side of the Datatracker.

# Self Test

## Chapter 2

---

**This completes the final phase of the Self Test procedure.**

After you have successfully completed all phases of the Self Test, the Datatracker should be ready for use as a diagnostic tool. As you can see, this Self Test procedure appears longer than it really is. We would encourage you to go through it any time you suspect the Datatracker to be defective. If a problem is encountered, please double check the battery, the clip leads and switch settings before contacting your dealer. One final word about slide switches. Unlike toggle switches, slide switches can be positioned to be open, closed or somewhere (unknown) in between. Always be sure, whether during the Self Test or in actual use, that the slide switches are positioned either fully to the left or fully to the right.

## **RS-232 Background Information**

RS-232 is the name of a standard or protocol which defines a method by which computers and various peripherals should communicate with each other. The original application was intended to be used by large mainframe and minicomputers in a central location servicing hundreds (or even thousands) of users with terminal and printers. Since it was physically impossible to locate all of these users in close proximity to the computer (ie. in the same room), a means of connecting them via telephone lines and modems was devised. This is where RS-232 comes in, since its purpose was to define the hardware and electrical aspects of these computer/modem and modem/terminal connections. All went reasonably well until personal computers came along.

With the advent of PCs it became possible to have the computer (with a dedicated terminal built in) and the printer both on the same desk. This not only eliminated the need for the telephone lines but also the modems at each end. This, in turn, should have made everything simple, but unfortunately the RS-232 standard doesn't address direct computer to printer connections. But people soon figured out that the computers and the printers could be fooled or tricked into believing that they're working through modems and telephone lines and RS-232 survived. Even if a new standard had been developed, RS-232 would still be quite viable today since more people than ever are using modems (even with their PCs) to communicate over long distances.

In spite of its relatively common usage today, RS-232 remains one of the most misapplied and misunderstood interface standards currently in use. In this chapter we will cover some background information as well as a basic overview of what RS-232 is and how it works. For a more detailed understanding we would recommend any of the excellent books on the subject listed at the end of this chapter.

### **History**

In 1969 the Electronic Industries Association (EIA) in conjunction

# RS-232 Background

## Chapter 3

---

with various manufacturers of computers and communications equipment adopted the serial interface standard EIA RS-232-C. Its official name is "Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange". As the name implies, this standard defines an interface between a DTE device (ie. computer, printer, terminal, etc.) and a DCE device (ie. modem) for the transfer of serial data.

The standard was updated in late 1986 and is now officially referred to as EIA-232-D. The changes between the more familiar RS-232-C and the new RS-232-D (EIA-232-D) are relatively minor and will be noted as we go through this chapter. This standard can be divided into 5 areas of information defined as follows:

1. Electrical Signal Characteristics.

This portion covers the voltage, current and impedance specifications, how these signals relate to logical 1's and 0's and the maximum transmission speed (20,000 bits per second).

2. Interface Mechanical Characteristics.

This portion covers the assignment of 24 circuits to the numbered connector pins of a 25 pin D-type subminiature connector (RS-232-C did not specifically call out any connector, only that it should have 25 contacts) and the proper use of male and female connectors (DCE devices should be female). RS-232-C specifically limited cable lengths to no more than fifty feet (2500 picofarads of capacitance) but this limitation is now excluded from the RS-232-D standard.

3. Functional Description of Interchange Circuits.

This portion defines the names and functions of the electrical circuits used. Three new signals were added to the RS-232-C standard as well as redefinition of the Frame (or Protective) Ground.

4. Standard Interfaces for Selected Communications Systems.

This portion defines 15 generic interface types which is probably of little use to most people. By contrast, the older RS-232-C standard included more than a dozen specific modem to terminal

interfaces which are rarely in use today.

5. Notes, Recommendations and a Glossary.

Suggestions to product design engineers about what to do and what not to do in order to implement RS-232-D correctly are now included in the published standard. This is augmented by a glossary listing definitions of some of the terms used in the standard.

RS-232 is intended to apply to both synchronous as well as asynchronous serial communications. This standard is also intended to work over dial-up, leased or private line service employing either a 2-wire or 4-wire system. One purpose for the RS-232-D update was to bring it in line with its international counterparts known as CCITT V.24 and V.28 as well as ISO IS2110.

Although it may seem that these specifications are fairly comprehensive, they are actually far from it. Aspects not addressed by the standard include the type of data coding (ASCII, EBCDIC, Baudot, etc.), character length (5,6,7 or 8 bits), the number or type of start and stop bits or characters and the type of parity or error checking scheme. You should add to this the fact that no provision was made for direct DTE to DTE interfaces which are quite common today. These issues taken together with several other confusing points are largely the reasons why RS-232 is still so confusing.

### **Electrical Characteristics of RS-232**

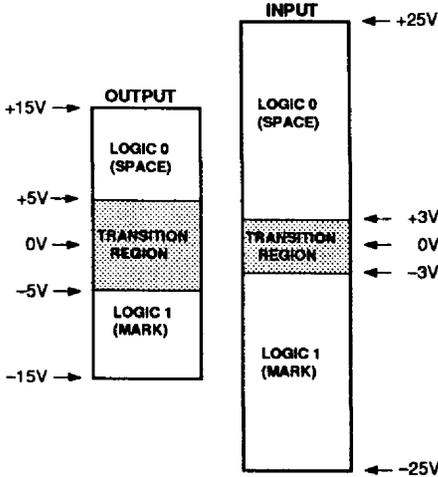
The standard defines the electrical specifications for the transmitting and receiving circuitry. To begin with, transmitters must be allowed to withstand indefinite short circuits either to each other or to the signal ground with no permanent damage. Receivers must meet the following input impedance requirements: their resistive load shall be between 3000 and 7000 ohms and their capacitive load shall not exceed 2500 picofarads. Also spelled out are the maximum and minimum rise and fall times for all RS-232 signals. Beyond this, things get a little confusing.

For instance, all the voltage levels involved do not correspond to conventional TTL logic levels for 1's and 0's. RS-232 outputs are

# RS-232 Background

## Chapter 3

defined as follows: +5V to +15V is a logic "0" (Space); -5V to -15V is a logic "1" (Mark); and +5V to -5V is the transition region. RS-232 inputs are defined in a similar fashion but with a narrower transition region and able to tolerate wider voltage swings: +3V to +25V is a logic "0"; -3V to -25V is a logic "1"; and +3V to -3V is the transition region. As you can see, RS-232 is still a binary logic system with only 2 allowable voltage ranges. The transition region (approximately "0" volts) is not allowed and having either an output or an input in this condition would indicate a problem.



To further complicate matters is the confusing use of some terminology concerning these levels. Terms such as logic "1", HI, Negative, True, Off and Mark are all used interchangeably. The following chart lists some of the more commonly used "synonyms".

| BINARY           | 0        | 1        |
|------------------|----------|----------|
| TELEGRAPH        | SPACE    | MARK     |
| CURRENT FLOW     | OFF      | ON       |
| FUNCTION         | ON       | OFF      |
| LOGIC            | FALSE    | TRUE     |
| COMPUTER "SLANG" | LO       | HI       |
| VOLTAGE          | POSITIVE | NEGATIVE |

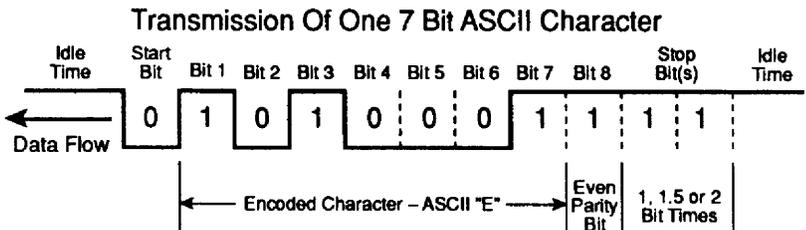
One item to note here is that the actual signal voltage represents an inverted logic. That is to say, a positive voltage is the same as a binary "0" and a negative voltage is the same as a binary "1". This inversion is already built into the RS-232 output and input ICs (1488s and 1489s respectively).

## Data Coding and Character Length

Transmission codes such as 5 level Baudot (named after Emile Baudot), 6 level BCD (Binary Coded Decimal) and 8 level EBCDIC (Extended Binary Coded Decimal Interchange Code) have all served (and are still serving) their places in history. In today's small computers however, as well as with most peripherals, the most widely accepted coding standard is 7 level ASCII (American Standard Code for Information Interchange). ASCII characters are encoded in a format that is 7 bits long with an 8th bit available for parity. This 7 level code defines a total of 128 ( $2^7$ ) characters which includes upper and lower case alphabet, numerals, punctuation marks, special characters and control characters. An ASCII Character Chart is included in the Appendix for your reference.

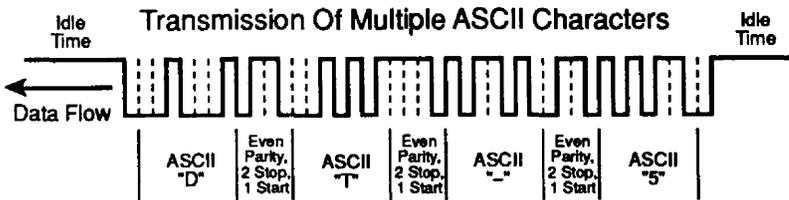
## Asynchronous Character Format

In asynchronous communications, each individual character to be transmitted is encoded and then outputted as a sequential series of pulses. These pulses represent the bit pattern of the character as well as start and stop bits used to synchronize the receiver and a parity bit used for error detection. The bit pattern for the character is arranged in a "backwards" order where the least significant bit is transmitted first and the most significant bit is transmitted last. In order to transmit a 7 level ASCII character with 1 start bit, 1 stop bit



# RS-232 Background

## Chapter 3



and 1 parity bit, a total of 10 bits need to be sent. Although this represents a transmission overhead of 30% (3 bits out of every 10 convey no data) asynchronous transmission nevertheless remains a popular means of communication. Its advantages include the ability to transmit just one character at a time at odd intervals and the ability to synchronize the receiver to the transmitter on a character by character basis. Asynchronous lends itself well to transmitting small amounts of data (such as keyboard data entry) or data at low speeds (to slow printers for example).

### Synchronous Character Format

When it comes to transmitting large amounts of data as quickly as possible synchronous transmission becomes a more effective alternative. In synchronous transmission a character is first encoded and then grouped together with a number of other characters to form a block. This block is preceded by several synchronization characters and a start of transmission character, and then followed by an end of transmission character and an error check character. This block is then sent out as a continuous strings of pulses with no interruptions. Although there is still an overhead associated with synchronous communications, it is generally much less than that of asynchronous. The benefit here is that when long files take less time to transmit, less cost is involved. There are many different synchro-

### Synchronous Transmission Of 16 Data Characters



SYN = Synchronization Character  
STX = Start Of Text

ETX = End Of Text  
EOT = End Of Transmission  
BCC = Block Check Character

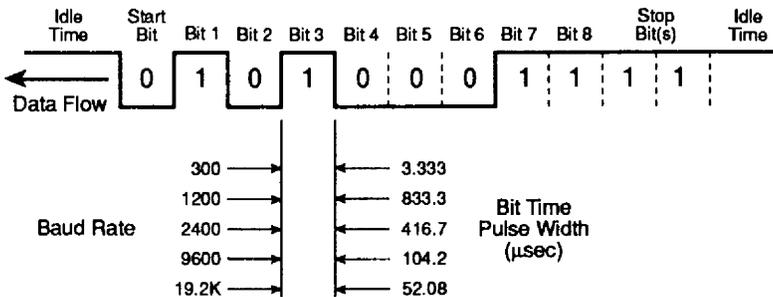
nous transmission protocols in use today such as BISYNC, SDLC, HDLC, etc. and a discussion of these is best left to the appropriate texts on these subjects.

## Transmission Speed and Cable Length

Baud rate, bits per second (BPS) and characters per second (CPS) are different yet related measures of transmission speed. Bits per second, as the expression implies, is a measure of the number of bits (binary digits) transmitted per second. As previously mentioned, the upper limit on transmission speed as defined by the RS-232 standard is 20,000 BPS.

Baud rate, which is often confused with BPS, is actually a measure of signal transitions per second. The distinction here is that by using sophisticated data encoding techniques it is possible to transmit 2 or even 4 bits per signal transition. In this case the BPS rate would be either 2 or 4 times the baud rate. These techniques are commonly used by high speed modems hoping to move more data across a telephone line than the bandwidth would normally allow. In all other applications however, just one bit is encoded per signal transition. In these cases the baud rate is the same as the number of bits per second. Commonly used baud rates are: 50, 75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600 and 19.2K baud.

Characters per second is a measure of the number of characters transmitted per second. To determine the CPS rate simply divide the BPS rate by the number of bits per character (ie. 1200 BPS divided by 10 bits/character equals 120 CPS).

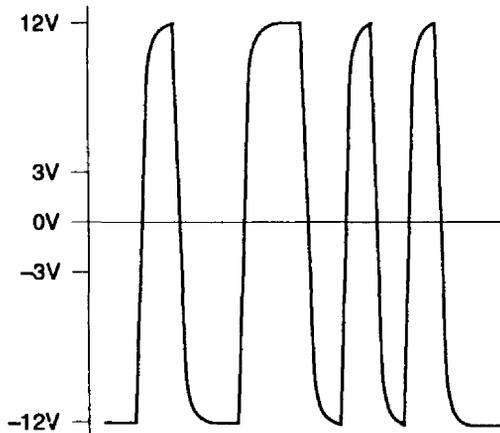
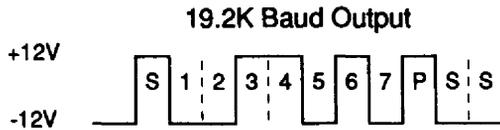


# RS-232 Background

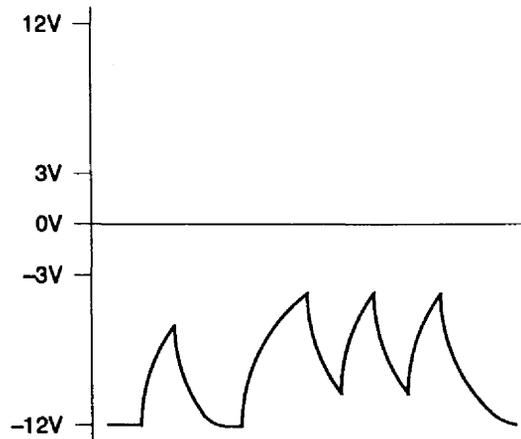
## Chapter 3

It naturally follows that the faster you wish to transmit data, the more BPS need to be sent and the shorter the time interval per bit becomes. The shorter these pulses become, the more difficult it is for them to overcome the electrical properties of the communications channel. Even a short length of wire will exhibit a certain amount of resistance and capacitance which will cause signal distortion and attenuation. In effect, the higher the transmission speed and the longer the cable, the more difficult it will be for the receiver to accurately detect the transmitted pulses.

The older RS-232-C standard suggested that cables should not exceed 50 feet in length. This was not meant to say that transmission would cease at 51 feet however. Indeed,

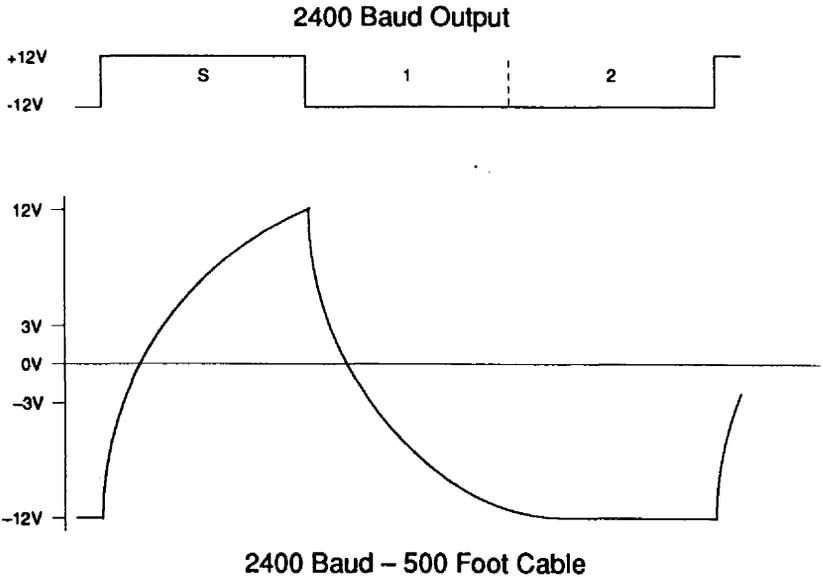


19.2K Baud - 50 Foot Cable



19.2K Baud - 500 Foot Cable

many people have successfully implemented RS-232 communications well in excess of 50 feet, sometimes even several thousand feet. The tradeoff that is usually made for extended distance is reduced transmission speed. A 19.2K baud interface that works at 50 feet probably will not work at 500 feet. This high speed signal would be subject to too much signal distortion and attenuation. By reducing the baud rate to 2400 baud the signal pulses will become much wider and will exhibit less distortion and attenuation. Although we don't recommend this practice, it is nevertheless commonly used in the industry and worth being aware of.



Just for the record, the RS-232-D standard specifically excludes any limitation on the length of cables as long as all the other electrical specifications are met. Suffice it to say that the old specification limiting cables to no more than 50 feet is still a pretty good rule of thumb.

# RS-232 Background

## Chapter 3

---

### RS-232 Signal Definitions

There are a total of 24 circuits defined by the RS-232-D standard. These circuits provide for the following five basic functions: ground, data, control, timing and secondary operations. A complete RS-232-D Signal Chart can be found in the Appendix and should be referred to as we go through these definitions. It is important to note that the following signal names are defined from the DTE point of view.

#### Pins #1: Frame Ground (FG)

Frame Ground (sometimes called Protective Ground) is electrically connected to the equipment frame and may be used as the connection point for the shield if shielded cable is being used to reduce EMI/RFI noise radiation. The shield should only be connected on the DTE side of the interface.

#### Pin #7: Signal Ground (SG)

Signal Ground is the common ground reference point for all RS-232 signal lines including data, control, timing and secondary operations. Without pin #7 none of these signals could function.

#### Pins #2 and #3: Transmitted Data (TD) and Received Data (RD)

These are the pins through which the DTE device transmits and receives data. Conversely, the DCE device transmits on pin #3 (RD) and receives on pin #2 (TD). Both of these lines are held in the marking condition (negative voltage) when not active or between characters.

#### Pins #4: Request To Send (RTS)

The DTE device may use the Request To Send line in order to prepare the modem for transmission. Depending on the transmission mode (full duplex or half duplex) the RTS signal will place the modem in a receive only or transmit only mode.

#### Pin #5: Clear To Send (CTS)

The Clear To Send signal is the corresponding (when used with RTS) signal from the modem back to the DTE device that it is ready to transmit.

**Pin #6: Data Set Ready (DSR)**

Data Set Ready is a signal from the modem to the DTE and is used to indicate that it is connected, powered up and otherwise ready (ie. not in a talk, test or dial mode of operation). This is often used in conjunction with the DTR signal.

**Pin #8: Received Line Signal Detector (DCD)**

Commonly called "Data Carrier Detect", a modem uses this line to inform the DTE that it is receiving a signal on the telephone line of sufficient amplitude and at the proper frequency for transmission.

**Pin #11: Not Assigned**

Although this pin is technically unassigned by the RS-232-D standard it has found many uses in various devices. The most common application has been by a number of printers which use pin #11 as a flow control or hardware handshake line.

**Pin #20: Data Terminal Ready (DTR)**

Data Terminal Ready is used by many DTE devices to either place the modem on-line or take it off-line. In effect, the DTE can tell the modem whether or not it should accept an incoming call or initiate an outgoing call.

**Pin #22: Ring Indicator (RI)**

The Ring Indicator signal is used by an auto-answer modem to inform the DTE that a ring signal is present on the telephone line. The DTE may then instruct the modem to answer the call (take the line off-hook) by using the DTR signal.

These 11 pins define the most commonly used signals in asynchronous RS-232 communications. As you've no doubt noticed, these signals were defined for typical DTE to modem interfaces. What should you do when you need to directly connect two DTE devices such as a computer and a printer without using modems? This is where it gets confusing since you will need to fool the two DTE devices into thinking that they are connected to modems. To make matters worse, all these signals may not be implemented by the DTE devices or may even be used for other purposes. The next

# RS-232 Background

## Chapter 3

---

chapter will present our 8-Step Interfacing Method in order to help you overcome these problems.

It should also be noted that in 1984 IBM Corporation introduced a PC with a scaled down version of the normal DB25 connector RS-232 interface. This RS-232 port used a DB9P connector and was intended for asynchronous serial communications only. The following lists which signals were included: Signal Ground, Transmitted Data, Received Data, Request To Send, Clear To Send, Data Set Ready, Data Carrier Detect, Data Terminal Ready and Ring indicator. Although these signals have the same name and purpose as in the RS-232 standard, 8 of the 9 signals are on different pin numbers. An 'RS-232 Signal Subset' chart is located in the Appendix for your reference.

Before we attempt any interfacing however, we would like to briefly cover the remaining RS-232-D signals. These signals are mainly used in synchronous communications and for various testing purposes.

### **Pins #15, #17 and #24: Timing Signals**

These signals are used to control the timing of the transmitted and received data over synchronous modems. Pin #15 (Transmitter Signal Element Timing-DCE Source), commonly known as Transmitter Clock, is used to provide the DTE with transmitted data timing. Pin #17 (Receiver Signal Element Timing-DCE Source), commonly known as Receiver Clock, is used to provide the DTE with received data timing. Both of these signals are generated by the transmitting modem and both DTE devices are intended to synchronize on this signal. Pin #24 (Transmitter Signal Element Timing-DTE Source), commonly known as External Transmitter Clock, is used when the transmitting DTE device generates the synchronizing signal to the DCE.

### **Pin #21: Signal Quality Detector**

This signal is used by some modems to indicate if there is a high probability of error in the received data.

## Pin #23: Data Signal Rate Selector

Depending on the direction, this is actually two different signals and is used to select the speed of transmission when using a two speed modem.

## Pins #12, #13, #14, #16 and #19: Secondary Channels

Also known as auxiliary circuits, these 5 signals (Secondary Received Line Signal Detector, Secondary Clear To Send, Secondary Transmitted Data, Secondary Received Data and Secondary Request To Send) are equivalent in operation to their primary counterparts. Some modems incorporate these secondary channels and use them for various purposes.

## Pins #9 and #10: Reserved For Testing

These two pins are reserved for data set testing and are sometimes used to check for proper voltage levels. Pin #9 is usually a positive DC test voltage while Pin #10 is usually a negative DC test voltage.

## Pins #18 and #21: Local Loopback and Remote Loopback

Pin #18 is a newly defined signal in the RS-232-D standard (it was unassigned in RS-232-C) and is intended to control the local DCE and place it in an analog loopback mode for testing purposes. Pin #21 also has a new designation (if it isn't being used as Signal Quality Detect) in the RS-232-D standard. This signal will allow the local DCE to place the remote DCE into a digital loopback mode.

## Pin #25: Test Mode

This is the third new signal that was defined by RS-232-D (it was unassigned in RS-232-C). This signal is used to tell the DTE that the modem (or modems) is currently in the Local Loopback (or Remote Loopback) mode and that testing may proceed.

At this point you should have a basic understanding of what RS-232 is and how it was intended to be used. If you should feel the need to learn more about RS-232-C, RS-232-D or communications in general we would recommend any of the books on the following page.

# RS-232 Background

## Chapter 3

---

**EIA Standard: EIA-232-D**

**Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange**

**by Electronic Industries Association**

**Data Communications: A Comprehensive Approach**

**by Gilbert Held and Ray Sarch**

**McGraw-Hill Publications Co.**

**Understanding Data Communications**

**by George E. Friend, John L. Fike, H. Charles Baker and John C. Bellamy**

**Texas Instruments**

**RS-232 Made Easy: Connecting Computers, Printers, Terminals and Modems**

**by Martin D. Seyer**

**Prentice Hall Inc.**

**Computer Communication Techniques**

**by E. G. Brooner and Phil Wells**

**Howard W. Sams & Co., Inc.**

**The RS-232 Solution**

**by Joe Campbell**

**Sybex Computer Books**

### Using The Datatracker For Interfacing

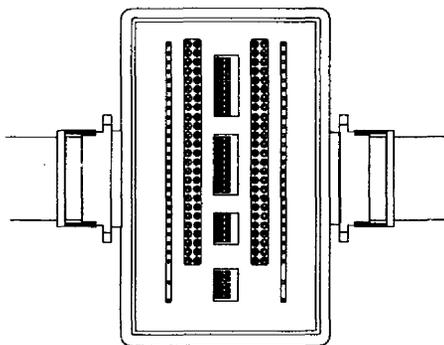
In this Chapter we will describe an "8 Step Interfacing Method" that we recommend as a quick, effective and easy procedure for interfacing RS-232 devices. This method will be described in detail with numerous notes, comments and suggestions to assist you. A one page condensed version of the "8 Step Interfacing Method" is included in the Appendix for your reference. This condensed format has also been reproduced on the inside of the Datatracker's cover.

We would strongly encourage you to read through this chapter in its entirety before attempting an interface. As you will notice, there are many facets to completing an interface properly, and the better that you understand this procedure and all of the pieces of the interfacing puzzle, the greater your chances for success will be.

#### Step #1

##### Initialize the Datatracker

In this step you must prepare and connect the Datatracker for interfacing. Start by removing any clip leads and opening all the switches (left position). Next, connect the Datatracker to the two devices to be interfaced by using the two 25 conductor 'Y' cables supplied with the Datatracker. Using cables with less wires, crossed wires or single-sided connections will, at best, confuse you and at



All Switches open  
Two 25 conductor 'Y' cables

# Interfacing

## Chapter 4

worst, make it impossible to complete the interface. Your ability to monitor all 25 lines of both devices simultaneously is effectively defeated by using the wrong cables. This point cannot be stressed enough: use two 25 conductor straight through cables.

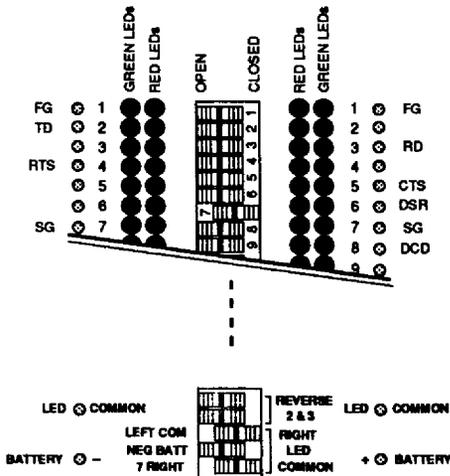
### Step #2

#### Select Ground Reference

In this step you must select the proper ground references for the two devices you are interfacing as well as for the Datatracker.

The first ground reference is called "Frame Ground" ("FG" on line #1). Switch #1 ("FG") should normally be left open since this line is only used for connecting to the shield when using shielded cable. If you close switch #1 ("FG"), you will tie the chassis of both devices to the same earth ground potential. Note: Problems with the AC wiring, either in the devices or the building wiring itself, might warrant leaving this switch open and the two chassis electrically separate.

The next ground reference is called "Signal Ground" ("SG" on line #7). By closing switch #7 ("SG"), you will connect the logic grounds of both devices together. This is a necessary connection in all interfaces since all the other signals are referenced to Signal Ground.



So much for the two devices, now you must also make sure that the Datracker is properly referenced. You do this by closing option switch #3 ("LEFT COM") and option switch #5 ("7 RIGHT"). Closing the "LEFT COM" switch is just like taking a jumper and connecting the left side "LED COMMON" turret pin to the right side "LED COMMON" turret pin. We designed the LED Commons to be separate and floating to allow you more versatility when using the Datracker in other applications. With the "LEFT COM" switch closed you have combined these two points into one node but it is still floating. This is where the "7 RIGHT" option switch comes in. Closing this switch is just like connecting a jumper to the "Signal Ground" (line #7) on the right side of the Datracker to the "LED COMMON" also on the right side. This references the LED Commons to the Signal Ground.

A word of advice is in order here. Unless these last three switches ("SG", "LEFT COM", "7 RIGHT") are closed, you may very likely see some strange and/or dim LEDs turn on. This is due to internal feed-back through the LEDs and the resistor network within the Datracker itself. Proper ground referencing is essential to proper operation of the Datracker.

### **Step #3**

#### **Set Software Parameters**

The next step is to establish and match up the software parameters relating to the RS-232 protocol. These parameters include:

- Baud Rate
- Number of Data Bits
- Number of Stop Bits
- Type of Parity
- Type of Handshake

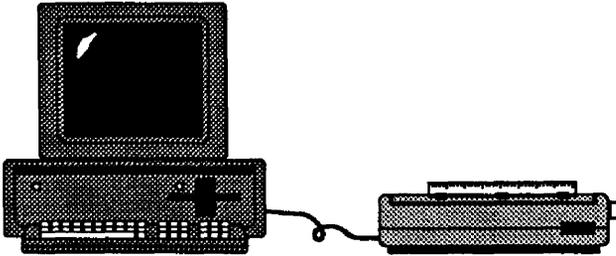
On many computer systems these parameters can be changed in the set-up or configure utilities of some applications programs. On MS-DOS compatible computers, the 'Mode' program will allow you to configure these parameters but not read the current settings. In most printers these parameters are changed by either using a front panel keypad or by resetting some internal switches or jumpers. And

# Interfacing

## Chapter 4

---

finally, in some peripherals, such as some modems, these parameters are either fixed or the device has the ability to automatically adjust some of these parameters to match those of the other device.



Baud Rate = Baud Rate  
Parity = Parity  
Number of Data Bits = Number of Data Bits  
Number of Stop Bits = Number of Stop Bits  
Handshake Type = Handshake Type

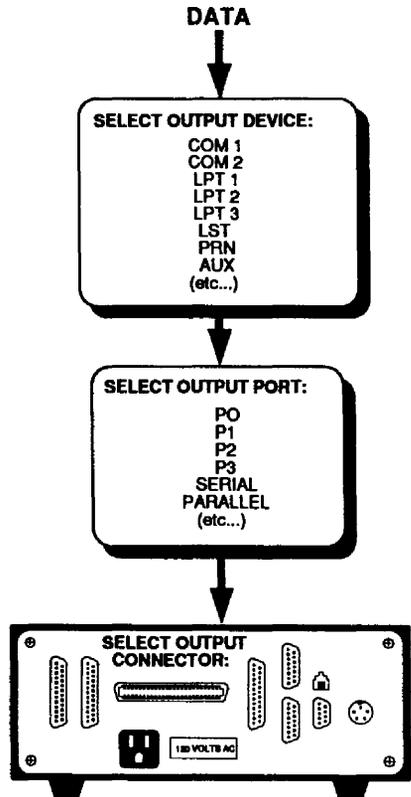
Sometimes it will be relatively easy to change these parameters on both sides of the interface; many times it will not be. Finding the proper utility program (and learning how to use it), locating the appropriate switches and discovering the default (or variable default) settings can easily become the most challenging part of RS-232 interfacing.

As a general rule we would suggest that you go with as high a Baud Rate as both devices will allow and make use of a hardware (versus software) Handshake protocol to control transmission. Your choice of the number of Data Bits, generally 7 or 8, should make no difference in text only applications (graphics, however will require 8), provided both devices match. The number of Stop Bits that you choose, typically 1 or 2, also doesn't make much difference provided both devices match. And as for Parity, either even or odd, it's useful for detecting data transmission errors and is worth implementing provided the equipment can do something useful with this information. If, however, you would prefer that the equipment ignore any

transmission errors (to prevent the system from locking up for instance), there is usually also the possibility of disabling the Parity (selecting "No", "None" or "Off"). Your choice of the number of Data Bits, the number of Stop Bits and the type of Parity should be determined by both the equipment that you're trying to interface as well as the application involved.

No matter what parameter values you decide to use in your particular situation, make sure that both devices are set to match. If they don't, and it's OK if you're not sure, you'll find out later on in the interfacing session. If all else fails, there is nothing wrong with resorting to some trial and error to complete this step.

One final comment is in order here. Included in some computer utility programs is the ability to select the output device/port you wish to communicate through. Needless to say, unless you select the proper port, you won't get very far. Also, be advised that logical port assignment is different from, and is sometimes confused with, physical connector selection. Once again, some trial and error might be in order here. The important thing to remember when resorting to trial and error is to be sure to try all possible combinations, starting with the most likely or "logical", until you get it working. Don't overlook any combination just because it doesn't make sense.



### Step #4

#### Initialize the Transmitting Device

The next step is to determine if the transmitting device needs any kind of initialization. If it does, it will need to receive some kind of input signal through the RS-232 connector before it will transmit any data.

To begin, simply try to transmit a small amount of data, a dozen or so lines should do, and watch the LEDs on either line #2 or #3 for that device. If the LEDs indicate data transmission (several seconds of flashing red and green lights), then no initialization is needed. You now know that you are plugged into the right connector and that this device apparently doesn't need any special signals in order to allow it to transmit. You should proceed on to Step #5.

If on the other hand you saw absolutely no indication of data transmitting then either one or a combination of up to three things could be wrong.

The first possibility is that you may have selected the wrong output device/port/connector in Step #3. In this case the data is probably being directed to some place other than where you want it. You should either go back to the equipment manuals to double check your selection or experiment with some other combinations. The goal here is to observe data transmission taking place on either line #2 or #3.

The second possibility is that the device needs to receive some kind of initialization or control signal before it will transmit any data. The most commonly used lines for this purpose are:

- Request To Send ("RTS" on line #4)
- Clear To Send ("CTS" on line #5)
- Data Set Ready ("DSR" on line #6)
- Data Carrier Detect ("DCD" on line #8)
- Data Terminal Ready ("DTR" on line #20)

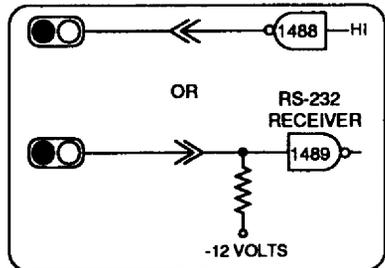
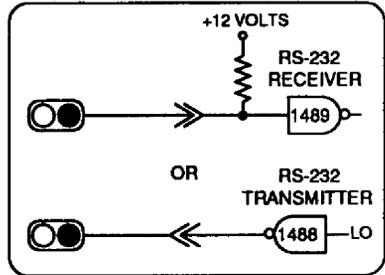
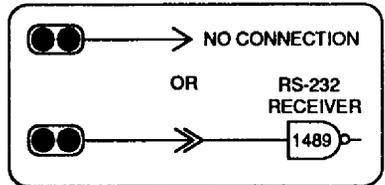
Equipment manufacturers use these lines in different ways for different purposes so you must be aware of all of them. Be advised

that Request To Send and Clear To Send are commonly used together. For instance, if a computer is sourcing a signal on Request To Send, then it is very likely looking for a response on the Clear To Send line. In a similar fashion, Data Terminal Ready and Data Set Ready often work together.

The key here is to determine which line (or lines) the transmitting device is monitoring for a control signal input and what polarity it needs to be. Finding the appropriate input(s) can be done either by consulting the equipment manual and/or schematic for the device or by trial and error. You should also feel free to refer to the RS-232 signal charts in the Appendix for an indication of which lines are typically inputs and which are typically outputs.

An input line will appear on the Datatracker in one of two ways. If the input is floating, no voltage will be on that line and consequently both LEDs (red & green) will be off. An open or unused line will also look exactly like this so be aware of this possibility for confusion. If the input is not floating it will have a pull-up (or pull-down) resistor tied to it inside the equipment and will appear as a red (or green) LED. This is exactly what an output or RS-232 transmitter will look like so you must be aware of this possibility for confusion as well.

Essentially what you must do is tie either one or several of these input lines high (likely) or low (unlikely) while trying to transmit data. Using the jumpers, connect one or more of these 5



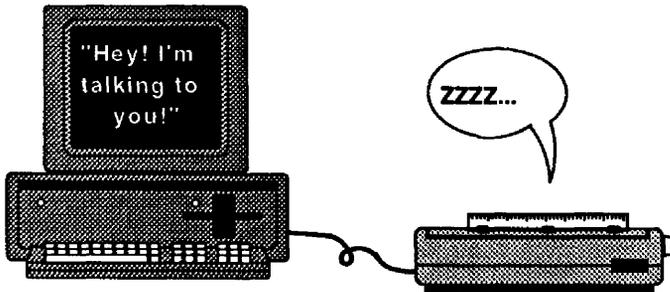
# Interfacing

## Chapter 4

---

lines to any signal in the interface that is already high (red). If this doesn't work, try tying one or more of these lines low. Should you happen to tie a high and a low signal together expecting both of them to end up high or both low but instead you see both LEDs go out (or very nearly so), be advised that you have just connected two outputs (transmitters) together. Although this will not harm the equipment, it will also not do any good, and you may as well just disconnect them without bothering to test this combination.

When you do find the proper line(s), data transmission will be enabled and you should be able to see the results on line #2 or #3. The reason that these control lines are used in some equipment is to indicate the status of one device to another. In a computer/printer interface, for instance, it would be senseless for the computer to transmit data unless the printer were turned on, plugged in (RS-232), on line and loaded with paper. The printer should be able to signal back to the computer on just one or possibly two control lines that all of these criteria have been met. Assured that its efforts will not be in vain, the computer may then proceed to transmit data.



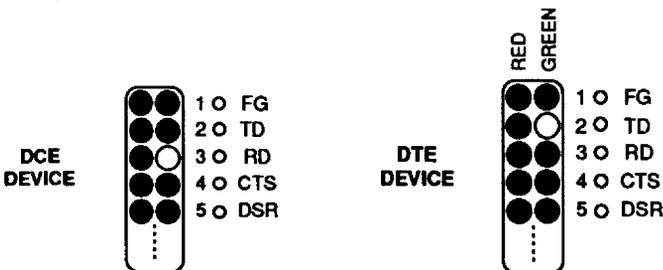
Having accomplished this phase of the interface you may now proceed on to Step #5.

In the, hopefully, unlikely event that you still haven't been able to get data flowing, you might be running into the third possible problem: hardware or software malfunction. If this appears to be the

problem we would refer you to the chapter on Troubleshooting. There is no sense in trying to complete an interface with defective equipment. The defect should first be isolated and repaired.

## Step #5 Data Transmission

In this step you need to determine the proper data connection for the two devices in question. There are only two possibilities here. You may see a green light on line #2 and no light on line #3 (on the same side), which would indicate that the device transmits on line #2 ("TD") and receives on line #3 ("RD"). This is quite common and would indicate that the device is a "DTE" or Data Terminal Equipment type of device (ie. computer, CRT, printer, etc.). The other possibility is that you may see a green light on line #3 and no light on line #2, which would indicate that the device transmits on line #3 ("RD") and receives on line #2 ("TD"). Although this may sound backwards, it is also quite common and would indicate that the device is a "DCE" or Data Communications Equipment type of device (ie. modem).



RS-232 was originally designed to connect DTE devices to DCE devices, such as a computer to a modem and then another modem to a printer or terminal. In this case (DTE to DCE) line #2 and line #3 should be connected straight across.

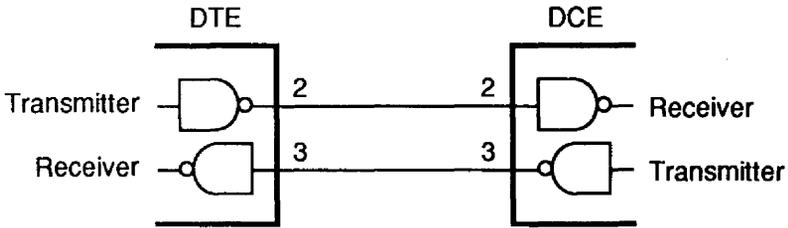
With the advent of today's PCs and other devices, it is quite possible that you may want to connect a computer directly to a printer without the need for modems. In this situation, as in many others, you would be trying to interface two DTE devices. When

# Interfacing

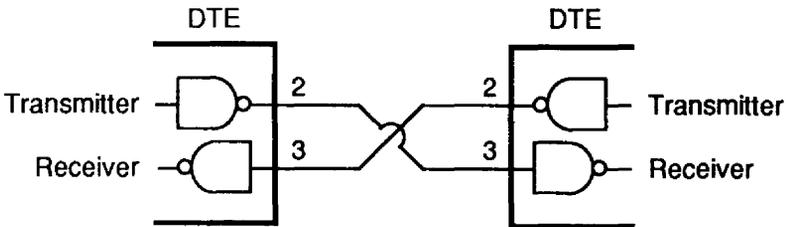
## Chapter 4

---

making this connection (DTE to DTE) lines #2 and #3 must be cross connected or reversed (line #2 to line #3 and line #3 to line #2). This is commonly called a null modem connection.



The general rule here is simple. Connect the transmitter of one device to the receiver of the other device. If the two green LEDs are on different lines ("TD" and "RD" as in a DTE and DCE) use switch #2 and switch #3 to connect these lines straight across. If the two green LEDs are on the same line (generally "TD") then use the first two option switches called "REVERSE 2 & 3" to cross connect these lines. In either case you should end up with all four LEDs green (lines #2 and #3 on both sides).



Caution: The two pairs of switches must be used in an either/or manner. For example, if you need to reverse #2 and #3, do not also connect them straight across. Closing all four switches will tie all four lines together which will not allow the interface to work.

One final note here. Should you run across a device which does not match the description for either a DTE or a DCE (ie. neither LED

on, two LEDs on or a red LED on) then you probably have a hardware problem with that device. Check the equipment manual or schematic for additional information or check the RS-232 driver I.C. (#1488) for proper operation.

## **Step #6**

### **Initialize the Receiving Device**

The next step is to determine if the receiving device needs any kind of initialization. Just as with the transmitting device, it is quite possible that the receiving device needs some kind of input before it will act upon and/or process the incoming data.

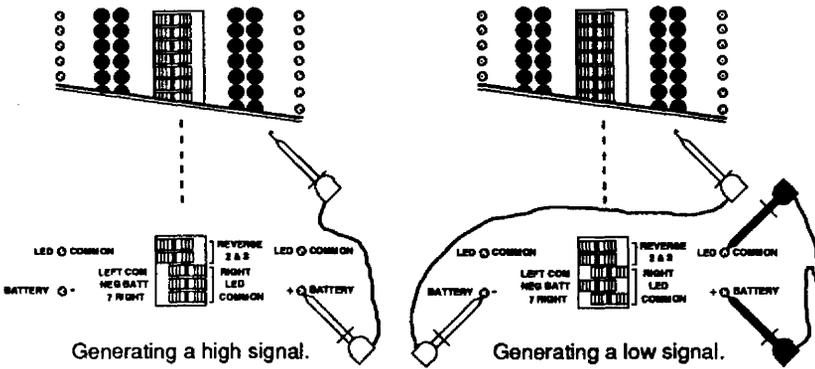
To proceed, simply try to transmit a small amount of data (watch line #2 or #3 for activity) to the receiving device. If it responds to the incoming data (don't worry about the quality of the output yet) then the device probably doesn't need any special signals to allow it to receive data. You may therefore proceed on to Step #7.

If on the other hand the device just sits there and apparently ignores the incoming data, then it probably needs some kind of initialization (or it could also be defective). The procedure that you need to go through, and for that matter the lines involved, is exactly the same as outlined in Step #4. The goal here is to get the receiving device to respond to the incoming data, regardless of the quality of the output. Data quality will be dealt with in Step #7.

One technique in using the Datatracker involves tying lines either high or low. If you need to tie suspected input lines either high or low your first choice should always be to use another line in the interface as a source which is already the correct polarity. In the event no suitable signal exists, you may use the Datatracker to simulate these signals. To generate a high signal, close option switch #4 ("NEG BATT") which will connect the negative battery turret pin to the right "LED COMMON" and, by virtue of the "7 RIGHT" switch, to the Signal Ground in the interface. Using a clip lead that is connected to the positive battery turret pin ("BATTERY +") you can now force selected lines in the interface high. To generate a low signal, the "NEG BATT" switch must be open and a clip lead should connect the

# Interfacing

## Chapter 4



"BATTERY +" pin to either "LED COMMON" pin. By using another clip lead that is connected to the "BATTERY -" pin you will be able to force selected lines low.

Assuming you have been successful in convincing the receiving device to respond to incoming data you might feel that you are done with the interface (particularly if the quality of the output looks OK). Be advised that you aren't finished until you address one more vital element: Handshake.

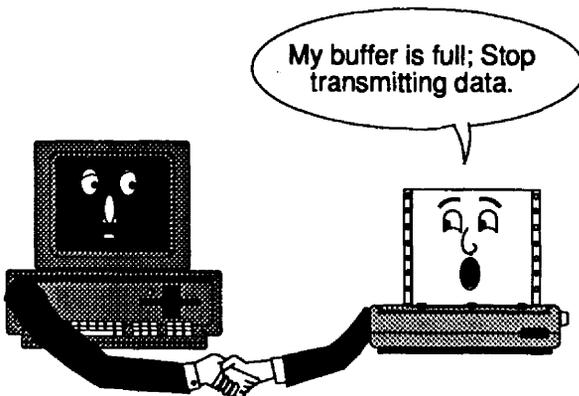
### Step #7 Handshake

The purpose of this step is to determine the necessary handshake or flow control lines between the two devices. The reason for this is that most receiving devices need to have the ability to stop and then re-start the data transmission in accordance with their ability to process the incoming data. For instance, a computer can transmit data much faster than most printers can print (or process) it. The result, without a handshake signal, will be lost or unprocessed data. The amount of data that gets lost depends upon the data transmission rate (Baud Rate), the speed with which the receiving device can process the data, the size of the receiving device's input (or data) buffer and the total amount of data involved.

To begin with, check the quality of the data output following Step

#6. If the output starts out totally meaningless, (that is to say, from the first character on it bears no resemblance to the incoming data) then you probably have a mis-match in one or more of your software parameters (ie. Baud Rate, Data Bits, Stop Bits or Parity). If the receiving device is a printer, for instance, you might see things like: multiple form feeds, multiple lines feeds, no line feeds with multiple overstrikes, seemingly random characters or possibly control or graphics characters you never thought the printer capable of printing. Go back to Step #3 and, one by one, try changing the parameters that might be mis-matched.

The other possibility is that the output starts out meaningful and then after a certain amount of data (whether several lines or several pages) it turns into random characters or blocks of data begin disappearing. These are normal indications for missing handshake or flow control signals. The distinction between this situation and the one above is significant. The fact that the beginning data output is correct would indicate that the software parameters (Baud Rate, Data Bits, Stop Bits and Parity) are properly matched and should not be changed. The point at which the output turns into "garbage" is the point at which the receiving device's buffer fills up and additional data starts getting lost. While all this is going on the receiving device is no doubt working as fast as it can to process the data already in its buffer thereby making room for some more data, filling it's buffer again and then losing even more data.



# Interfacing

## Chapter 4

---

The situation that we just described is exactly the one you need to create in order to find out which handshake line(s) the receiving device is using. What you need to do is transmit enough data fast enough so as to overflow the receiving device's buffer. When this happens you should notice one or more lines on the receiving device's side change state in its vain attempt to halt the transmission. This is its handshake line and you should see it toggle from one color to the other as its buffer becomes full and then back again as more space is made available. Be advised that some devices (most notably some printers) have rather large input buffers (2K, 8K or more) and it may take some time, not to mention data, before the handshake line shows itself. If this is the situation, you must be prepared to transmit either a large file or write a short program to continuously send data to the receiving device until the buffer overflows.

### **GWBASIC Program For Continuous Text**

```
10 FOR A=48 TO 126
20 LPRINT CHR$(A) ;
30 NEXT A
40 LPRINT
50 GO TO 10
```

The most commonly used lines involved in handshaking are the same lines as those used in initialization, namely, line #4, line #5, line #6, line #8 and line #20. To this we should add two more likely candidates: line #11 (technically unassigned by the RS-232 standard and yet a common printer handshake line) and line #19 ("Secondary Request to Send").

Having discovered the handshake line for the receiving device you must now find the corresponding line for the transmitting device. To do this, begin transmitting data again (ignore or turn off the receiving device for the moment) and one by one tie the possible handshake lines first low and then high to see which one stops and

then starts the data transmission. This can be seen by monitoring the appropriate data line (either line #2 or #3) on the transmitting side. You may have already encountered the appropriate line while selecting a hardware handshake protocol in Step #3. For example, if you selected an option that read something like "DSR BUSY" then it would be a fairly safe bet that line #6 is the one you're looking for. Another possibility is that you may have encountered the handshake line while having to initialize the device in Step #4.

In any event, note the polarity of the two handshake lines. If the handshake line of the transmitting device needs to be high in order to transmit and the receiving device's handshake line is normally high until its buffer fills up, then they match. Simply connect them with a jumper and observe the results. Prior to transmission the receiving device's buffer should be empty so its handshake line will be high (for instance) which will enable transmission. When transmission begins, the receiving device begins processing the data as its buffer fills up. When the buffer is full (but has not yet overflowed) you should notice it's handshake line go low thereby telling the transmitting device to stop (observe lines #2 and/or #3). The receiving device will continue to process the data in its buffer thereby making some room in it. As soon as a certain amount of space is available its handshake line will go back high, once again enabling transmission (watch lines #2 and/or #3). This process should continue until all the data has been transmitted and processed.

Should you discover two devices whose handshake lines don't match in polarity (rare), you will have a problem. First see if there are any other handshake lines available on either side of the interface. Failing this, you may have to change the polarity (if possible) of the handshake line of one device so that the handshake will work properly (consult the equipment manuals for additional information). You might also consider using a software handshake protocol.

Another situation that you might encounter is a receiving device that apparently doesn't need to use a handshake line since it can process the incoming data at least as fast as the transmitter can send it. For example, a 200 CPS printer should be able to easily

# Interfacing

## Chapter 4

---

keep up with data from a 1200 baud modem. Be very careful about not connecting the handshake line since other benefits, not immediately obvious, may be lost. Many printers combine their "On Line/Off Line" switch and/or their "Paper Out" sensor with their handshake line making it well worth the exercise to find and connect the handshake line. While you're testing the handshake line you should also be checking the effect of the "On Line/Off Line" switch as well as any other conditions which would effect the receiving device's status.

The above example is also an excellent reason why you shouldn't "cheat" by simply reducing the baud rate to below the receiving device's processing speed. Aside from not maximizing the system's data throughput (and therefore not getting your, or someone else's, money's worth) this short-cut could backfire on you in terms of the other problems mentioned.

As a final note on handshaking, be aware that several software protocols exist as well (ie. "X-ON/X-OFF", "ETX/ACK", etc.). These protocols essentially work on the same principle as a hardware handshake except, instead of using a voltage level on a dedicated handshake line the receiving device sends out special "Control Characters" on its Transmit Data line. One Control Character would tell the transmitting device to stop transmission when the receiving device's buffer is full. Then, when there is room in the buffer again, a second (different) Control Character is sent to the transmitter in order to resume transmission. This can be seen on the Datatracker as a short burst of activity on the appropriate line (either #2 or #3).

As mentioned earlier, we would recommend using a hardware handshake protocol whenever possible. It has been our experience that a significant number of devices do not allow for software handshake and that even when devices do, they may not match in terms of protocol. It is also much more difficult to troubleshoot a short "blip" of data than the relatively obvious voltage level changes on a dedicated line.

## Step #8 Simplify, Test and Document

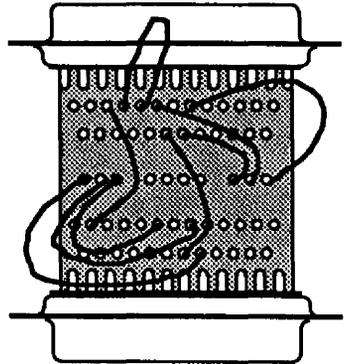
In this, the final step, you can take comfort from the fact that the hard part is over. But, as we all know, the job is never done until the paperwork is complete.

By simplify, we mean take a look at the interface that you patched together on the Datatracker and see if all those connections that you've made are really necessary. It's not very difficult in Step #4 or #6 to end up with more jumpers than you actually need. In the Appendix we have included sample interface configurations just to give you an idea of how things might look. Be sure to make a note of your current configuration before you begin disconnecting things – just in case.

Another thing you might want to consider is the possibility of making the cable reversible if both cable connectors are the same gender. If it is possible to make the connections on both sides of the Datatracker symmetrical, it is generally worth doing. This might prevent a service call should someone inadvertently plug the cable in backwards.

The next step is to actually make up an interface cable. This can be done by either soldering a cable and connectors together, using crimp type insertion pins and sockets or by using an interface adapter such as the Micropatch with a standard cable. Once complete and installed, the interface should be thoroughly tested in order to check out the cable. Marking both ends as to where they go will also be a tremendous aid to those who follow in your footsteps. Remember, they may not have a Datatracker.

The final phase of this last step is to document your accomplishment. In the Appendix you will find a blank



# Interfacing

## Chapter 4

Interface Documentation Sheet which should help you organize the appropriate information. The LED designation by the way, is what you should see after Step #3 (before the interface is connected). Should you ever run into similar equipment or should you ever have to troubleshoot these two devices, you'll be glad you took an extra minute or two to do this.

Please feel free to refer to the Appendix (page A4) or the inside of the Datatracker's cover for a condensed version of this 8 Step Interfacing Method.

### Helpful Hints

Before you try your hand at interfacing we would like to mention several common interfacing problems and suggestions. If you've had a lot of interfacing experience then you've no doubt run into at least one, if not more, of these problems before.

As we mentioned in Step #3, many computer systems use a type of set-up or configure utility program to select the software parameters and possibly the device/port/connector of your choice. We would suggest that before making any changes you should write down all of your starting parameters. The reason for this is that it isn't very difficult to become confused with all of the possibilities. If you make too many changes for you to remember, it might be helpful, if not necessary, to be able to get back to where you started from.

Another common problem is after having made your selection, it is not uncommon that the device may apparently "lock up" on you



**INTERFACE DOCUMENTATION SHEET**

EQUIPMENT MANUFACTURER \_\_\_\_\_

MODEL NUMBER \_\_\_\_\_

TYPE OF EQUIPMENT \_\_\_\_\_

M F EQUIPMENT CONNECTOR M F \_\_\_\_\_

M F CABLE CONNECTOR M F \_\_\_\_\_

| ADDRESS                    |    |             |     |        |     |      |      |       |      |       |       |
|----------------------------|----|-------------|-----|--------|-----|------|------|-------|------|-------|-------|
| 50                         | 75 | 110         | 180 | 300    | 500 | 1000 | 2400 | 4000  | 8000 | 16 2K | 32 4K |
| # OF SERIALS:              |    | # OF PORTS: |     | SERIES |     |      |      |       |      |       |       |
| 8                          | 6  | 7           | 8   | 1      | 1   | 2    | Name | Event | Cmd  | Mark  | Space |
| Hardware TYPE _____        |    |             |     |        |     |      |      |       |      |       |       |
| Hardware I/O: / X-O: _____ |    |             |     |        |     |      |      |       |      |       |       |

| Comments | LED | Pin | Pin | LED | Comments |
|----------|-----|-----|-----|-----|----------|
| _____    | --- | 1   | 1   | --- | _____    |
| _____    | --- | 2   | 2   | --- | _____    |
| _____    | --- | 3   | 3   | --- | _____    |
| _____    | --- | 4   | 4   | --- | _____    |
| _____    | --- | 5   | 5   | --- | _____    |
| _____    | --- | 6   | 6   | --- | _____    |
| _____    | --- | 7   | 7   | --- | _____    |
| _____    | --- | 8   | 8   | --- | _____    |
| _____    | --- | 9   | 9   | --- | _____    |
| _____    | --- | 10  | 10  | --- | _____    |
| _____    | --- | 11  | 11  | --- | _____    |
| _____    | --- | 12  | 12  | --- | _____    |
| _____    | --- | 13  | 13  | --- | _____    |
| _____    | --- | 14  | 14  | --- | _____    |
| _____    | --- | 15  | 15  | --- | _____    |
| _____    | --- | 16  | 16  | --- | _____    |
| _____    | --- | 17  | 17  | --- | _____    |
| _____    | --- | 18  | 18  | --- | _____    |
| _____    | --- | 19  | 19  | --- | _____    |
| _____    | --- | 20  | 20  | --- | _____    |
| _____    | --- | 21  | 21  | --- | _____    |
| _____    | --- | 22  | 22  | --- | _____    |
| _____    | --- | 23  | 23  | --- | _____    |
| _____    | --- | 24  | 24  | --- | _____    |
| _____    | --- | 25  | 25  | --- | _____    |

Dev: \_\_\_\_\_ Cable ID: \_\_\_\_\_

808 Main Street • Denver, Colorado 80202 • 303-776-0570

while trying to get through Step #4. The reason for this is that the device is probably looking for some kind of initialization on either the connector you're currently plugged into or some other port/connector. Most likely you will have to Reset the system in order to regain control of it so that you can continue your trial and error procedures. Be advised that unless you saved your current software parameters they will probably default back to the parameters you started with each time you Reset.

It should also be mentioned that different versions of applications programs and even Operating Systems don't always communicate the same way. As an example, MS-DOS's Mode program in DOS 3.3 is much different, as well as larger, than in the older DOS 3.1. As a result, a perfectly functional interface that works under 3.1 might not work at all (and will need to be redone) when the computer's operating system is updated to DOS 3.3.

We also mentioned in Step #3 that many printers use switches to establish their software parameters. Be advised that some printers are designed to read these switches only upon initial power-up and otherwise ignore them. Therefore, changing these settings while the printer is on, without turning it off and then back on again, may very well be fruitless. You should consult the printer's manual to verify the need for this procedure.

## **Parallel Ports**

As a final note under interfacing we should discuss parallel ports. When the Datatracker was originally designed, RS-232 ports all used DB25 connectors and Centronics parallel ports all used 36 pin connectors. A number of years ago IBM Corporation decided that Centronics parallel ports really didn't need all those 36 lines that the rest of the industry was using. As a result, they introduced a DB25 version of a parallel port which confused quite a few people. No longer would you be able to look at the back of a computer and see a DB25 connector and feel safe in saying that it must be an RS-232 port.

Although the Datatracker wasn't designed to work with parallel

# Interfacing

## Chapter 4

---

ports it won't hurt anything if you plug it into one. What you will see after Step #1 (before proceeding to Step #2) is a number of red and green LEDs all on but very dim. Most notable will be LEDs #2 thru #9 which will typically be a very dim green. This is a sure sign that you're plugged into a DB25 parallel port. If you're attempting to connect a Centronics parallel port to an RS-232 serial port, STOP RIGHT NOW, do not close any switches or connect any jumpers. Connecting RS-232 drivers with TTL parallel port drivers and receivers might damage the parallel port.

If, on the other hand, you just want to see which parallel port signals are active, you must correctly reference the Datatracker's LED Common to the parallel port ground. Pins 19 thru 25 (pin #18 has on occasion been used for +5V) are usually ground for the DB25 version of the Centronics parallel port. Simply connect a jumper between pin #25 and the LED Common of that side. This should make all the green LEDs go off and brighten the red LEDs somewhat. Since a parallel port uses TTL signals (+3 to +5V for high and 0V for low) only the red LEDs will be used. Due to the lower voltages of parallel ports, the red LEDs will also be noticeably dimmer than you are probably accustomed to with RS-232 signals. Although some people have used the Datatracker to do parallel port interfacing and, to a lesser extent, troubleshooting, we can not recommend it for this purpose.

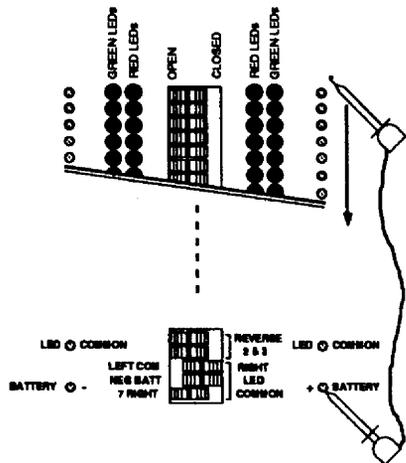
## Cable Testing

Interface cables are an integral part of any data communications/processing system. Due to the difficulties in testing this relatively simple part, the interface cable is frequently overlooked as a possible problem source. The Datatracker, however, has changed all this. Because of its ability to directly monitor all 25 lines of both sides of an interface, the Datatracker also has the ability to completely test all DB25 type interface cables. Unlike using an ohm meter, cable testing with the Datatracker is a very quick and thorough procedure. It will graphically show you exactly where each line goes and leaves nothing to the imagination.

### Normal Cable Testing

In many situations, interface cables are used to connect two devices in the same room and are usually easy to disconnect from the hardware. The procedure for testing these cables is very similar to the procedure used to test the Extension Cables (Phase 6) in the Self-Test Chapter.

1. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
2. Plug both ends of the cable to be tested into the Datatracker using one of the 'Y' cables if necessary to correct any mismatched connectors.
3. Close the "LEFT COM" and "NEG BATT" switches.
4. Connect one end of a clip lead to the positive battery pin.
5. Using the free end of the clip lead, scan each of the 25 numbered turret pins on both sides of the Datatracker.



# Cable Testing

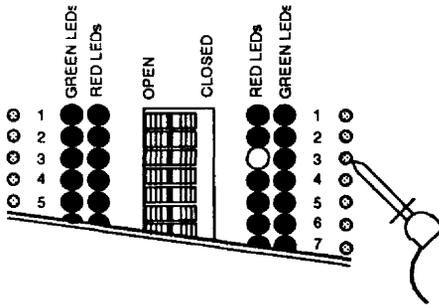
## Chapter 5

---

There are five possible indications that the Datatracker may provide when testing cables. They are:

1. No Connection.

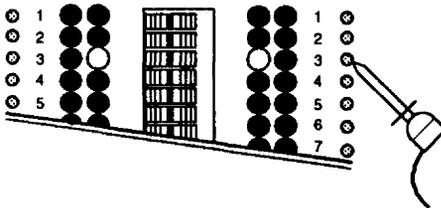
If just one LED lights up (the one on the line that you're scanning) it means that this line is not connected to any other line. If this is



not what you expected to see, then you may have a broken wire, bad connector contact, or a bad solder or crimp joint.

2. Straight Through Connection.

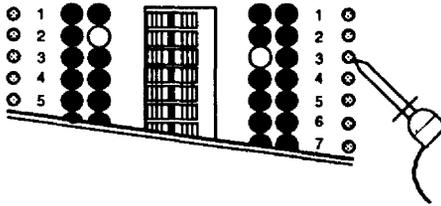
If both the right side and the left side LEDs for the same line light up (the LED on the line you're scanning as well as the LED



straight across from it) then this line is connected straight across only. This is the same indication you received while testing the Extension Cables.

### 3. Crossed Connection.

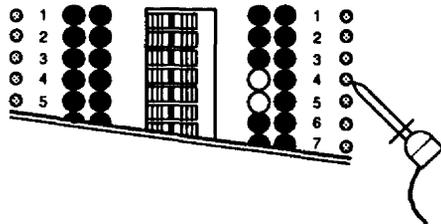
If LEDs on both the right side and the left side light up for different lines (the LED on the line that you're scanning as well as an LED on the opposite side other than the one straight across from it)



then this line is cross connected and does not go straight across. A common example of this is a null modem connection (line #2 connected to line #3 and line #3 connected to line #2).

### 4. Single Sided Connections.

If two or more LEDs light up all on the same side (the side being scanned) then those lines form a single sided connection and do



not go across. You must scan both sides of the Datatracker in order to find all single sided connections in any cable.

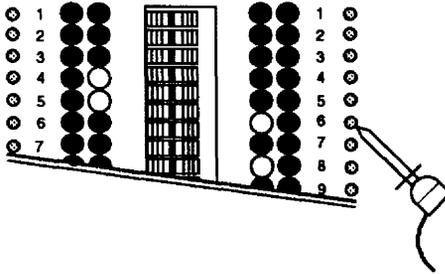
### 5. Any Combination of #2, #3 and #4.

If a total of three or more LEDs light on both sides of the Datatracker then you have a combination of several of the previous indications.

# Cable Testing

## Chapter 5

---



No matter what the situation is, the Datatracker can immediately show you how any line is connected. If you were using an ohm meter, all 49 possibilities would need to be examined one by one – for each line.

The Datatracker can also be used to indicate intermittent connections. If you suspect that a line is intermittent, simply connect the clip lead to the appropriate turret pin. You must observe all of the LEDs on the Datatracker as you start wiggling the cable. If you notice any LEDs going on and off as you push, pull or twist the cable then you have an intermittent connection on that line.

Cable testing can also be called cable identification. Custom cables, existing interface cables and especially unknown “junk box” cables can now be quickly and easily identified and documented. For this purpose we would suggest that you use copies of the Interface Documentation Sheet located in the Appendix. Since cable testing is now relatively simple, it should be done as a routine part of any interface troubleshooting procedure.

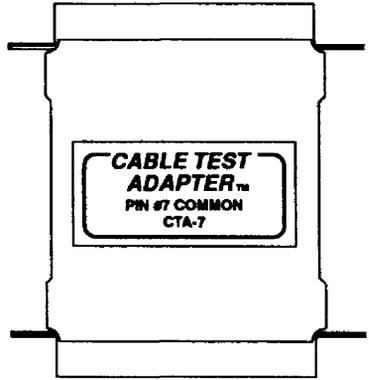
### Single Ended Cable Testing

Not all applications, however, will allow simultaneous access to both ends of the interface cable. It is not uncommon to run cables through walls, over ceilings or under floors. In these situations you may still use the Datatracker, in conjunction with either of two optional accessories, to perform what is called single ended cable testing. By using the Datatracker to scan the cable at one end and using a remote loopback or readout device at the other end, previ-

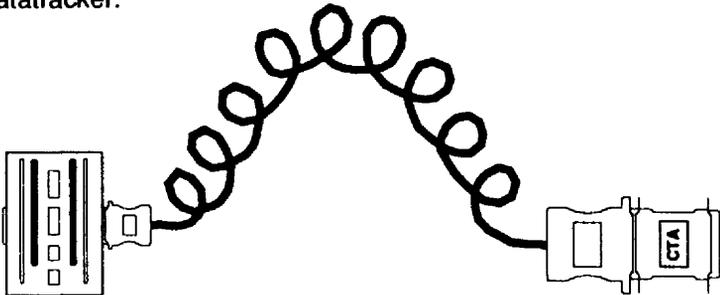
ously installed cables can also be easily tested.

## Cable Test Adapter

When used with the Datatracker, the Cable Test Adapter (CTA-7) will allow you to verify continuity from one end of a cable to the other. The CTA-7 is no larger than a Gendermate, has both male and female DB25 connectors for easy connection and contains a diode matrix to loop signals back to the Datatracker on line #7. It will allow you to quickly test for short circuits, open circuits and intermittent connections. Another application would be to identify one cable out of many in a computer room or a patch panel situation. The procedure for single ended cable testing using a CTA-7 is very similar to normal cable testing.



1. Plug the CTA-7 into the "far" end of the cable to be tested.
2. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
3. Plug the other end of the cable into the appropriate side of the Datatracker.



4. Close the "LEFT COM" and "NEG BATT" switches.

# Cable Testing

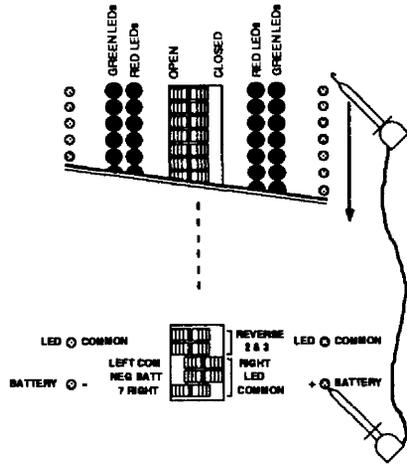
## Chapter 5

5. Connect one end of a clip lead to the positive battery pin.
6. Using the free end of the clip lead, scan each of the 25 numbered turret pins on the side of the Datatracker that the cable is plugged into.

There are four possible indications that the Datatracker may provide when using a CTA-7. They are:

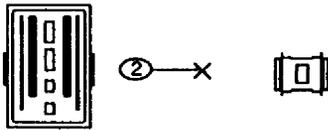
1. No Connection.  
If just one LED lights up (on the line that you're scanning) it means that this line is not connected to the far end or to any other line.
2. Continuity.  
If the LED on line #7 lights up in addition to the LED on the line that you're scanning then you have continuity. This means that the line that you're scanning is connected to the far end.
3. Single Sided Connections.  
If two or more LEDs light up, but not line #7, then those lines form a single sided connection on the Datatracker end of the cable and do not go across to the far end.
4. A Combination of #2 and #3.  
If three or more LEDs light up including line #7, then you have both a single sided connection as well as continuity to the far end. A short circuit between two lines would be indicated in this way.

An important thing to note here is that an indication of continuity tells you that a line is connected from one end of the cable to the other but not how it is connected. For instance, when testing a cable you notice that continuity exists for both line #2 (TD) and line #3 (RD). You now know that these two lines are connected to the CTA-7 end of the cable and that they are not shorted to any other lines on

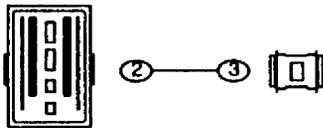
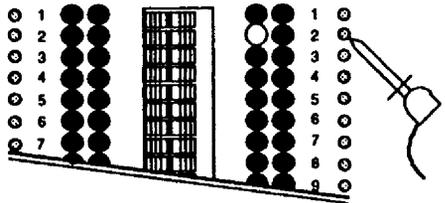


# DATATRACKER

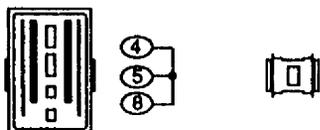
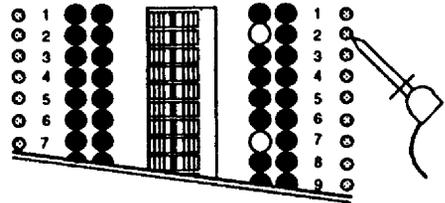
*The Professional Breakout Box*



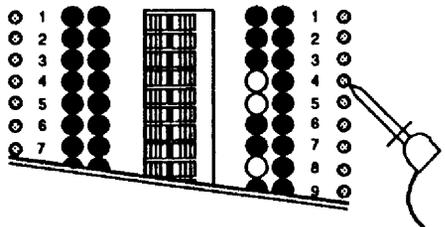
No Connection



Continuity



Single Sided Connection



the Datatracker side. What you will not know is if they go straight across (ie. #2 to #2 and #3 to #3) or if they are reversed as in a null modem cable. For this reason the CTA-7 is best used as a quick way of verifying a cable whose wiring configuration is already known.

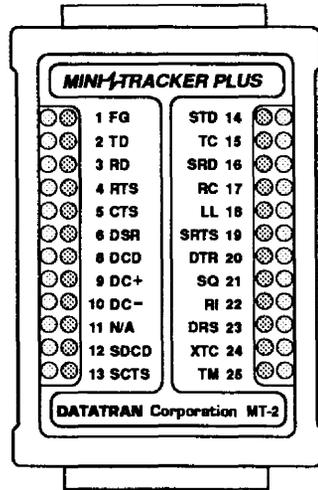
One final comment that we should make about using the CTA-7 is that both shorted as well as single sided connections can only be detected on the Datatracker end of the cable. For this reason you should exchange the Datatracker with the CTA-7 and test the cable again for these conditions from the other end.

# Cable Testing

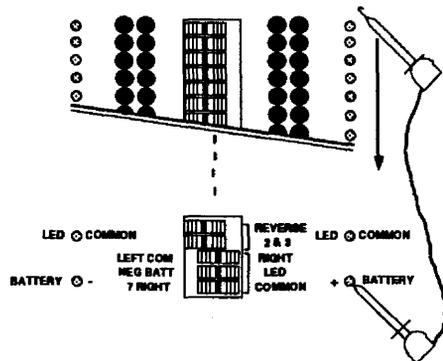
## Chapter 5

### Minitracker Plus

Another optional accessory that may be used with the Datatracker is the Minitracker Plus. The Minitracker Plus was designed to be an in-line signal monitor capable of indicating the status or activity of 24 of the RS-232 lines (1-6 and 8-25) simultaneously. It uses the same high efficiency red and green LEDs as the Datatracker on each of these 24 lines. When used in conjunction with the Datatracker however, the Minitracker Plus becomes an effective remote readout device when doing single ended cable testing. The procedure for using the Minitracker Plus is very similar to that of the CTA-7.



1. Plug the Minitracker Plus into the far end of the cable.
2. Disconnect all cables and clip leads from the Datatracker and ensure that all switches are open.
3. Plug the other end of the cable into the Datatracker.
4. Close the "LEFT COM", "NEG BATT" and "7 RIGHT" switches.
5. Connect one end of a clip lead to the positive battery pin.
6. Using the free end of the clip lead scan each of the 24 (omit line #7) numbered turret pins on the side of the Datatracker that the cable is plugged into.



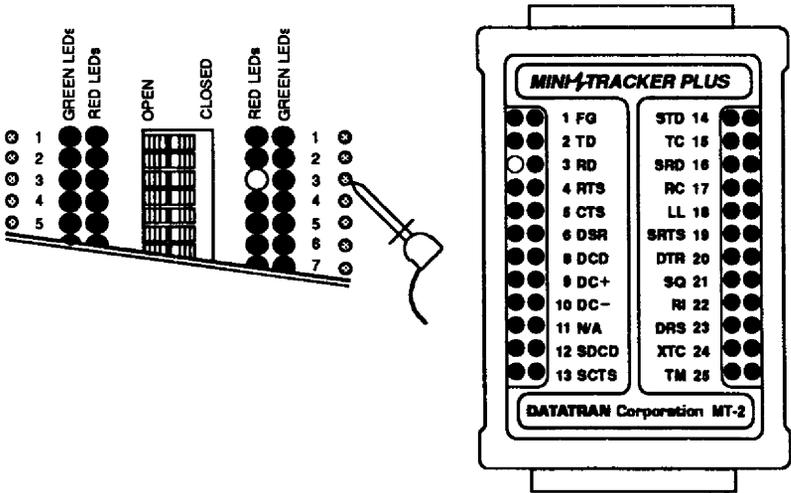
There are five possible indications that the Datatracker and the Minitracker Plus may provide. They are:

**1. No Connection.**

If just one LED lights up (on the line that you're scanning) and no LEDs light up on the Minitracker Plus, it means that this line is not connected to the far end or any other line.

**2. Straight Through Connection.**

If the same number LED lights on both the Datatracker and the Minitracker Plus then this line is connected straight across and nowhere else.



**3. Crossed Connection.**

If different numbered LEDs light on both the Datatracker and the Minitracker Plus then this line is cross connected and does not go straight across.

**4. Single Sided Connections.**

If two or more LEDs light up on the Datatracker and no LEDs light on the Minitracker Plus then those lines form a single sided connection and do not go across.

# Cable Testing

## Chapter 5

---

### 5. Any Combination of #2, #3 and #4.

If a total of three or more LEDs light on both the Datatracker and the Minitracker Plus then you have a combination of several of the previous indications.

As you've no doubt noticed, these five indications are essentially identical to those of normal cable testing using the Datatracker alone. By using the Minitracker Plus as a remote readout you are in fact simulating the other side of the Datatracker at the far end of the cable. The only functional difference here is the special handling of line #7. The Minitracker Plus uses line #7 (Signal Ground) as its LED common and therefore doesn't provide any LEDs for this line. Line #7 is also used as the return path when doing single ended cable testing. Therefore, if all the LEDs fail to light on the Minitracker Plus, line #7 may be open.

In the process of writing this documentation, it was discovered that the Minitracker Plus exhibited two additional features. The first feature was noticed when the "7 RIGHT" switch was opened. In this mode the Minitracker Plus will function just like a CTA-7. The #7 red LED will light whenever a line is scanned that has continuity. The #7 LED will appear to be about half brightness due to the additional resistance presented by the Minitracker Plus. In spite of this we felt it would be worth mentioning as it might save someone many trips back and forth trying to determine how a cable is wired.

The second feature was noticed when line #7 was scanned (also with the "7 RIGHT" switch open). All lines showing continuity will light up in red on the Datatracker and in green on the Minitracker Plus simultaneously. These LEDs will also be at about half intensity due to the doubled resistance back to the negative battery terminal. Once again we felt that this would be worth mentioning since this provides a quick and simple indication of all the wiring connections (single sided connections excepted) in the cable.

## **Troubleshooting**

Troubleshooting with the Datracker is much like interfacing. In fact, it is a good idea to develop a troubleshooting method or procedure like the one described in the interfacing section. By following a systematic approach, you can step through all aspects of the interface until you find the cause of any problem. The key to troubleshooting is 'Isolation'. You should attempt to isolate the problem down to its most basic level. When troubleshooting data communications problems, this isolation process begins by dividing the interface into three sections; device #1, device #2 and the interface cable.

### **Isolation**

Start off by testing the interface cable with the Datracker and compare your results with what a known good cable would look like. If the cable tests bad (or intermittent) then either repair or replace it and test the system again. If on the other hand the cable tests good, or you don't know what a good cable should look like, then just set it aside for now.

The next step is to connect the Datracker between the two devices and attempt to do a normal interface. For instance, let us assume that you're working with a computer and a printer. If you are unable to get the computer to transmit data then you're probably safe in assuming that the printer and the cable are alright and you should concentrate your energies on the computer. Conversely, if the printer is receiving data but fails to respond to it then the computer and the cable are probably alright and you can focus on the printer. If you are successful in interfacing the two devices, then compare your interface configuration with the interface cable (paying particular attention to possible intermittent connections).

Assuming that you've isolated the problem down to a particular device, you should continue your troubleshooting process hoping to pinpoint the problem down to a particular area or even a particular component in the system. Let's return for a minute to the computer that doesn't transmit. The problem could be in several places. You

# Chapter 6

## Troubleshooting

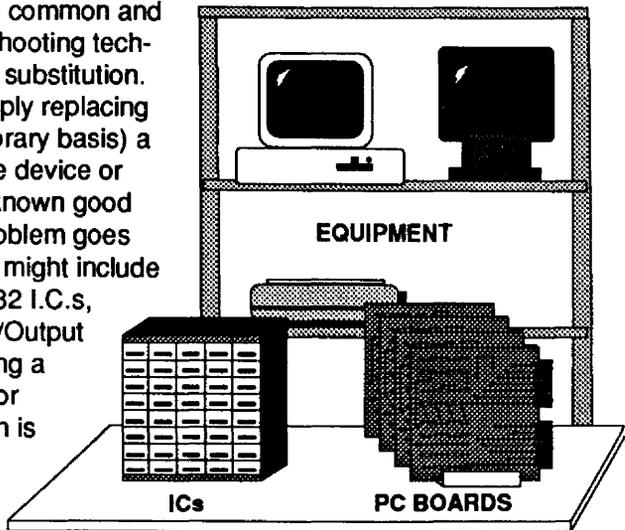
---

may have a bug (a permanent error in the program) in the applications program, a glitch (a temporary error in the program) or possibly the wrong selection for the output device. Likewise, you may also have a bug, a glitch or possibly the wrong port or interface parameters selected in the Operating System. You may have to do some detective work here to find out what recent changes have been made to the system (or what other problems have been recently fixed) for some additional clues. A third possible problem is a hardware malfunction. Assuming that the rest of the computer seems to be working properly, then a good place to start looking would be the actual RS-232 circuitry itself. This is what we mean by isolation, keep zeroing in on the problem until you have it identified.

When you have located the source of the problem, you should then proceed with normal repairs. After repairs have been made, be sure to return to the appropriate step, (the step in which you first observed the problem), and continue with your check-out procedures. This is just good practice since you may uncover more than one problem.

### Substitution

One of the more common and successful troubleshooting techniques is known as substitution. This is done by simply replacing (usually on a temporary basis) a suspected defective device or component with a known good one to see if the problem goes away. Substitution might include replacing the RS-232 I.C.s, swapping out Input/Output boards or even trying a different computer or printer. Substitution is not only limited to hardware. If you

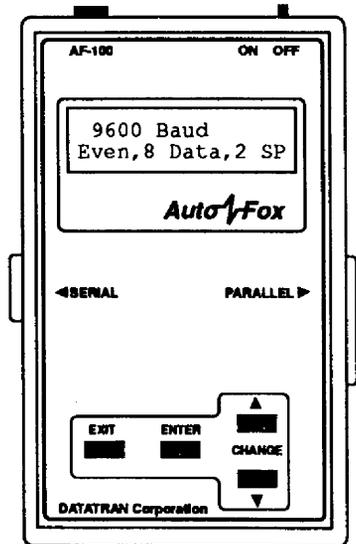


suspect that the problem is related to the applications program, try re-loading it or try running a different program entirely. If you have different ports through which the device may communicate, select a different one and see if that works. The level at which you employ substitution is a function of how far down you have managed to isolate the problem and what "spares" are available.

## Other Test Equipment

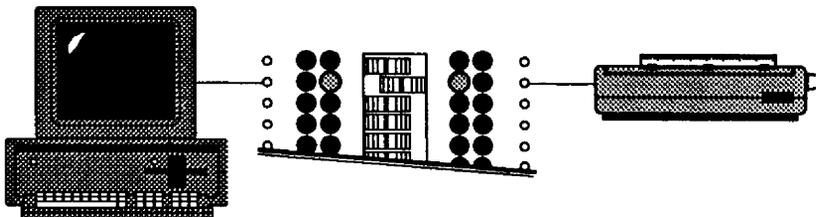
Test equipment such as an oscilloscope or a multi-meter may also prove invaluable when trying to determine an exact frequency or a voltage level in an interface. A logic probe with memory or a Pulse Tracker can aid in "bit-trapping", which is useful in finding short glitches or software handshake protocols. The AutoFox is also a powerful diagnostic and substitution device.

The Datatracker was designed to facilitate the use of other test equipment by making every line of the interface very accessible. Each turret pin has a little cap on top of the pin so that test probes will not slide off during use.



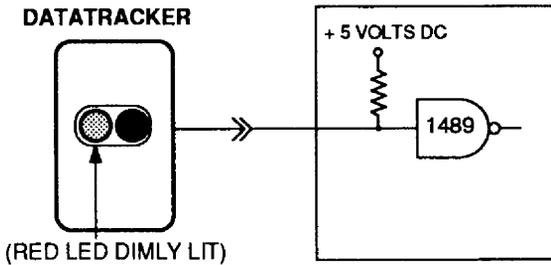
## Dim LEDs

The Datatracker itself can help in identifying a faulty RS-232 transmitter or receiver. By noting the relative intensity of each LED, you can easily identify one that may be dimmer than the rest.



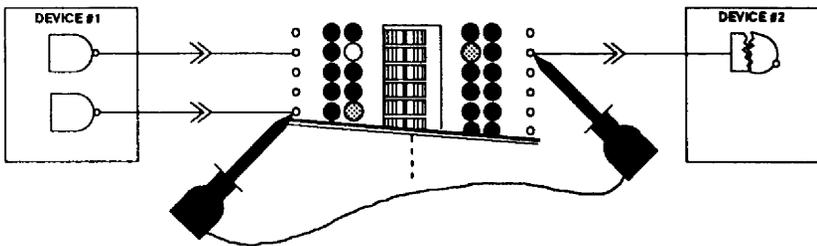
# Chapter 6

## Troubleshooting



There are three possible causes for a dim LED. The first reason would be if a pull up resistor on the input of a receiver was tied to +5 volts. The LED would show this as a dim red light and should not be a cause for alarm in this case. If you have any doubt about the existence of a pull up resistor you should always consult the equipment schematics for that device.

The second cause for a dim LED does pose a problem. If a faulty RS-232 receiver (such as I.C. #1489) is present, it can load down a perfectly normal transmitter to a point where the LED will be very dim. One way of checking for this is to disconnect that receiver from

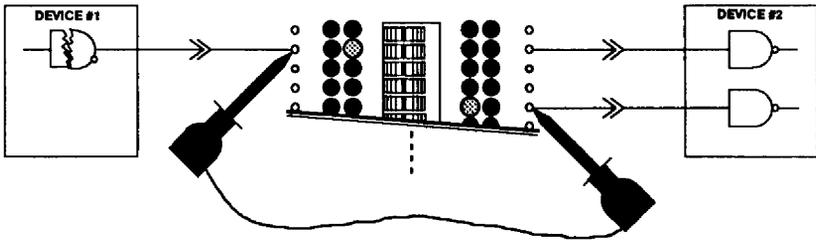


Defective receiver circuitry is verified by noting dimly lit LEDs after cross patching receiver to a second transmitter.

the transmitter and patch it to another line. If the original transmitter now shows a bright LED and the new transmitter LED glows dim, you can assume that the problem has something to do with that

receiver or the corresponding circuitry. It is quite possible that the receiver might even act as a direct short to the transmitter in which case the LED would go out entirely. In any event, replacement of the #1489 receiver I.C. would be recommended.

The third possibility would again present a problem. This would be when a faulty transmitter (such as I.C. #1488) causes an LED to be dim. The load placed on the line by a good receiver (or possibly even the Datatracker) may be too great for a faulty transmitter to overcome. This condition can be tested for in much the same way



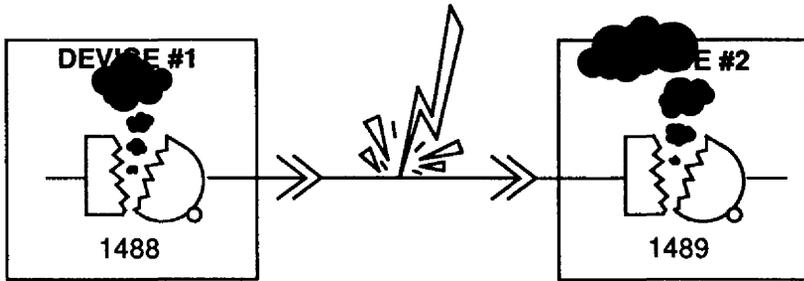
Defective transmitter circuitry is verified by noting dimly lit LEDs after cross patching transmitter to a second receiver.

as a faulty receiver. If no LED comes on at the (open circuit) output of a transmitter, then you probably have a defective #1488 transmitter I.C. which should be replaced. If you notice a dim or even a bright LED at the (open circuit) output of a transmitter and that LED becomes considerably dimmer (or goes out entirely) when connected to a good receiver, then you probably have a marginally defective transmitter which should be replaced as well.

A little bit of trial and error and possibly some substitution may be necessary in order to differentiate between a bad transmitter and a bad receiver. It is entirely possible that both may have gone out at the same time. RS-232 transmitters and receivers are any device's first line of defense against dangerous voltages coming in through the serial port. Induced voltages in RS-232 cables during nearby lightning activity as well as static discharges into the cables can

# Chapter 6

## Troubleshooting

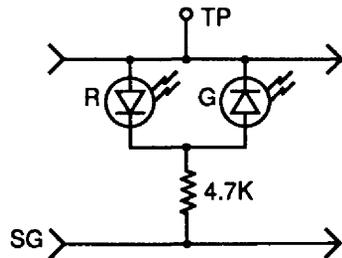


damage these I.C.s. To make matters worse, they don't have to fail completely but may be only marginally bad which may cause intermittent operation.

### Passive Design

One more thing that you should be aware of is that the Dataracker was designed as a passive device which means that it also presents a load to the RS-232 line. Under normal conditions this load is approximately 1.75 mA for most circuits which represents an input impedance of approximately 5.5 K $\Omega$  to 6.0 k $\Omega$  per line. This is within the normal 3.0 K $\Omega$  to 7.0 K $\Omega$  input impedance specification for an RS-232 receiver. The benefit in this is that the Dataracker will act as a test load equivalent to an RS-232 input. If a transmitter fails to light an LED on the Dataracker then it most certainly will have trouble driving a receiver properly.

When designing the Dataracker, it was felt that the advantages provided by a passive monitoring system far outweighed those of any active or buffered design. The ability to see the relative signal strength of all the active signals in an interface simultaneously will allow you to pinpoint marginal I.C. problems without wasting any time. And as we all know, time is money.



**ASCII Character Chart**

| Bit Positions | b8  | 0   | 0   | 0     | 0 | 0 | 0 | 0 | 0   |
|---------------|-----|-----|-----|-------|---|---|---|---|-----|
|               | b7  | 0   | 0   | 0     | 0 | 1 | 1 | 1 | 1   |
|               | b6  | 0   | 0   | 1     | 1 | 0 | 0 | 1 | 1   |
|               | b5  | 0   | 1   | 0     | 1 | 0 | 1 | 0 | 1   |
| b4 b3 b2 b1   | Hex | 0   | 1   | 2     | 3 | 4 | 5 | 6 | 7   |
| 0 0 0 0       | 0   | NUL | DLE | Space | 0 | @ | P | ` | p   |
| 0 0 0 1       | 1   | SOH | DC1 | !     | 1 | A | Q | a | q   |
| 0 0 1 0       | 2   | STX | DC2 | "     | 2 | B | R | b | r   |
| 0 0 1 1       | 3   | ETX | DC3 | #     | 3 | C | S | c | s   |
| 0 1 0 0       | 4   | EOT | DC4 | \$    | 4 | D | T | d | t   |
| 0 1 0 1       | 5   | ENQ | NAK | %     | 5 | E | U | e | u   |
| 0 1 1 0       | 6   | ACK | SYN | &     | 6 | F | V | f | v   |
| 0 1 1 1       | 7   | BEL | ETB | '     | 7 | G | W | g | w   |
| 1 0 0 0       | 8   | BS  | CAN | (     | 8 | H | X | h | x   |
| 1 0 0 1       | 9   | HT  | EM  | )     | 9 | I | Y | i | y   |
| 1 0 1 0       | A   | LF  | SUB | *     | : | J | Z | j | z   |
| 1 0 1 1       | B   | VT  | ESC | +     | ; | K | [ | k | {   |
| 1 1 0 0       | C   | FF  | FS  | ,     | < | L | \ | l |     |
| 1 1 0 1       | D   | CR  | GS  | -     | = | M | ] | m | }   |
| 1 1 1 0       | E   | SO  | RS  | .     | > | N | ^ | n | ~   |
| 1 1 1 1       | F   | SI  | US  | /     | ? | O | _ | o | Del |

# Appendix

## RS-232 Signal Subset DB-9 Connector

| PIN | NAME | DTE<br>DCE | FUNCTION            | CCITT | EIA |
|-----|------|------------|---------------------|-------|-----|
| 1   | DCD  | ◀          | Data Carrier Detect | 109   | CF  |
| 2   | RD   | ▲          | Received Data       | 104   | BB  |
| 3   | TD   | ▶          | Transmitted Data    | 103   | BA  |
| 4   | DTR  | ▶          | Data Terminal Ready | 108.2 | CD  |
| 5   | SG   |            | Signal Ground       | 102   | AB  |
| 6   | DSR  | ◀          | Data Set Ready      | 107   | CC  |
| 7   | RTS  | ▶          | Request To Send     | 105   | CA  |
| 8   | CTS  | ◀          | Clear To Send       | 106   | CB  |
| 9   | RI   | ◀          | Ring Indicator      | 125   | CE  |

## RS-232-D Signal Chart DB-25 Connector

| PIN | NAME   | DTE<br>DCE | FUNCTION                      | CCITT | EIA |
|-----|--------|------------|-------------------------------|-------|-----|
| 1   | FG     |            | Frame Ground                  | 101   | AA  |
| 2   | TD     | ▶          | Transmitted Data              | 103   | BA  |
| 3   | RD     | ▲          | Received Data                 | 104   | BB  |
| 4   | RTS    | ▶          | Request To Send               | 105   | CA  |
| 5   | CTS    | ▲          | Clear To Send                 | 106   | CB  |
| 6   | DSR    | ▲          | Data Set Ready                | 107   | CC  |
| 7   | SG     |            | Signal Ground                 | 102   | AB  |
| 8   | DCD    | ▲          | Data Carrier Detect           | 109   | CF  |
| 9   |        | ▲          | Positive DC Test Voltage      |       |     |
| 10  |        | ▲          | Negative DC Test Voltage      |       |     |
| 11  |        |            | Not Assigned                  |       |     |
| 12  | (S)DCD | ▲          | Secondary Data Carrier Detect | 122   | SCF |
| 13  | (S)CTS | ▲          | Secondary Clear To Send       | 121   | SCB |
| 14  | (S)TD  | ▶          | Secondary Transmitted Data    | 118   | SBA |
| 15  | TC     | ▲          | Transmitter Clock             | 114   | DB  |
| 16  | (S)RD  | ▲          | Secondary Received Data       | 119   | SBB |
| 17  | RC     | ▲          | Receiver Clock                | 115   | DD  |
| 18  | LL     | ▶          | Local Loopback                | 141   | LL  |
| 19  | (S)RTS | ▶          | Secondary Request To Send     | 120   | SCA |
| 20  | DTR    | ▶          | Data Terminal Ready           | 108.2 | CD  |
| 21  | SQ     | ▲          | Signal Quality Detect         | 110   | CG  |
| 22  | RI     | ▲          | Ring Indicator                | 125   | CE  |
| 23  | DRS    | ▶          | Data Rate Selector            | 111   | CH  |
|     |        | ▲          | Data Rate Selector            | 112   | CI  |
| 24  | XTC    | ▶          | External Transmitter Clock    | 113   | DA  |
| 25  | TM     | ▲          | Test Mode                     | 142   | TM  |

## 8 Step Interfacing Method

### Step #1: Initialize The Datatracker

Disconnect all cables and clip leads and be sure that all the switches are open (left). Then connect the Datatracker to the equipment using the 2 'Y' cables supplied.

### Step #2: Select Ground Reference

Close the following switches: switch #7 (Signal Ground), option switch #3 (LEFT COM), and option switch #5 (7 RIGHT).

### Step #3: Set Software Parameters

Match up, as much as possible and in accordance with the equipment and the application, the following parameters: Baud Rate, Data Bits, Stop Bits, Parity and Handshake. Also verify any Device/Port/Connector selections.

### Step #4: Initialize The Transmitting Device

Determine if the transmitting device needs any initialization. Likely lines are: Request To Send (RTS on line #4), Clear To Send (CTS on line #5), Data Set Ready (DSR on line #6), Data Carrier Detect (DCD on line #8) and Data Terminal Ready (DTR on line #20). Also check for proper Device/Port/Connector selection.

### Step #5: Data Transmission

Connect the transmitter of one device to the receiver of the other device. If the two green LEDs are on different lines (line #2 for one device and line #3 for the other device) then close switch #2 and switch #3 to connect them straight across. If the two green LEDs are on the same line (ie. line #2 for both devices) then close the first two option switches (REVERSE 2 & 3) to cross connect these lines.

### Step #6: Initialize The Receiving Device

Determine if the receiving device needs any initialization. Employ the same procedure and group of lines as in Step #4.

### Step #7: Handshake

Evaluate the receiving device's output; if it starts out as "garbage", go back to Step #3 and double check the software parameters. If the output begins correctly, then proceed to overflow the buffer of the receiving device in order to determine its Handshake line. Next, scan the possible lines (same as in Step #4) of the transmitting device (while its transmitting) to determine its Handshake line. If their polarities match, connect them together and test.

### Step #8: Simplify, Test & Document

Verify the need for all your connections. Make the interface symmetrical if possible. Wire up a final interface cable or adapter, test it and label both ends. Fill out an Interface Documentation Sheet with all the pertinent data.



**INTERFACE DOCUMENTATION SHEET**

\_\_\_\_\_ EQUIPMENT MANUFACTURER \_\_\_\_\_

\_\_\_\_\_ MODEL NUMBER \_\_\_\_\_

\_\_\_\_\_ TYPE OF EQUIPMENT \_\_\_\_\_

\_\_\_\_\_ M F EQUIPMENT CONNECTOR M F \_\_\_\_\_

\_\_\_\_\_ M F CABLE CONNECTOR M F \_\_\_\_\_

|                 |    |              |                 |     |     |         |      |      |      |                    |       |       |      |
|-----------------|----|--------------|-----------------|-----|-----|---------|------|------|------|--------------------|-------|-------|------|
| BAUD RATE:      |    |              |                 |     |     |         |      |      |      |                    |       |       |      |
| 50              | 75 | 110          | 150             | 300 | 600 | 1200    | 2400 | 4800 | 9600 | 19.2K              | 38.4K |       |      |
| # OF DATA BITS: |    |              | # OF STOP BITS: |     |     | PARITY: |      |      |      | TRANSMISSION MODE: |       |       |      |
| 5               | 6  | 7            | 8               | 1   | 1.5 | 2       | None | Even | Odd  | Mark               | Space | Async | Sync |
| HANDSHAKE TYPE: |    |              |                 |     |     |         |      |      |      |                    |       |       |      |
| Hardware        |    | X-On / X-Off |                 |     |     |         |      |      |      |                    |       |       |      |

| Comments | LED   | Pin | Pin | LED   | Comments |
|----------|-------|-----|-----|-------|----------|
| _____    | _____ | 1   | 1   | _____ | _____    |
| _____    | _____ | 2   | 2   | _____ | _____    |
| _____    | _____ | 3   | 3   | _____ | _____    |
| _____    | _____ | 4   | 4   | _____ | _____    |
| _____    | _____ | 5   | 5   | _____ | _____    |
| _____    | _____ | 6   | 6   | _____ | _____    |
| _____    | _____ | 7   | 7   | _____ | _____    |
| _____    | _____ | 8   | 8   | _____ | _____    |
| _____    | _____ | 9   | 9   | _____ | _____    |
| _____    | _____ | 10  | 10  | _____ | _____    |
| _____    | _____ | 11  | 11  | _____ | _____    |
| _____    | _____ | 12  | 12  | _____ | _____    |
| _____    | _____ | 13  | 13  | _____ | _____    |
| _____    | _____ | 14  | 14  | _____ | _____    |
| _____    | _____ | 15  | 15  | _____ | _____    |
| _____    | _____ | 16  | 16  | _____ | _____    |
| _____    | _____ | 17  | 17  | _____ | _____    |
| _____    | _____ | 18  | 18  | _____ | _____    |
| _____    | _____ | 19  | 19  | _____ | _____    |
| _____    | _____ | 20  | 20  | _____ | _____    |
| _____    | _____ | 21  | 21  | _____ | _____    |
| _____    | _____ | 22  | 22  | _____ | _____    |
| _____    | _____ | 23  | 23  | _____ | _____    |
| _____    | _____ | 24  | 24  | _____ | _____    |
| _____    | _____ | 25  | 25  | _____ | _____    |

Date: \_\_\_\_\_

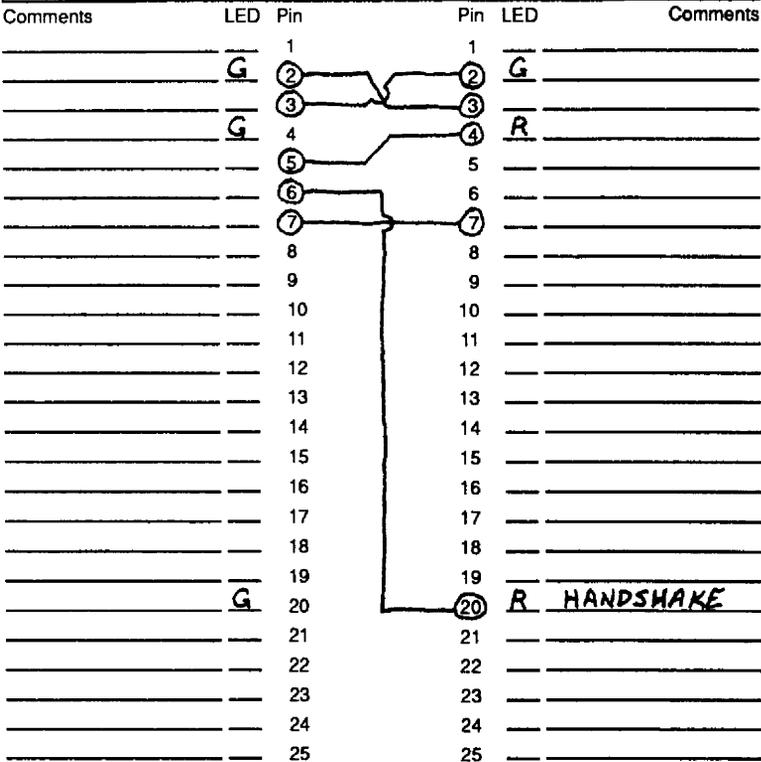
Cable ID: \_\_\_\_\_



INTERFACE DOCUMENTATION SHEET

IBM EQUIPMENT MANUFACTURER H-P  
PS-2 MOD 80 MODEL NUMBER LASERJET II  
COMPUTER TYPE OF EQUIPMENT PRINTER  
DB25  M F EQUIPMENT CONNECTOR  F DB25  
 \_\_\_\_\_ M F CABLE CONNECTOR  M  F \_\_\_\_\_

|                              |    |                 |          |          |     |         |             |                    |             |       |       |              |      |
|------------------------------|----|-----------------|----------|----------|-----|---------|-------------|--------------------|-------------|-------|-------|--------------|------|
| BAUD RATE:                   |    |                 |          |          |     |         |             |                    |             |       |       |              |      |
| 50                           | 75 | 110             | 150      | 300      | 600 | 1200    | 2400        | 4800               | <u>9600</u> | 19.2K | 38.4K |              |      |
| # OF DATA BITS:              |    | # OF STOP BITS: |          | PARITY:  |     |         |             | TRANSMISSION MODE: |             |       |       |              |      |
| 5                            | 6  | 7               | <u>8</u> | <u>1</u> | 1.5 | 2       | <u>None</u> | Even               | Odd         | Mark  | Space | <u>Async</u> | Sync |
| HANDSHAKE TYPE:              |    |                 |          |          |     | DOS 3.3 |             |                    |             |       |       |              |      |
| <u>Hardware</u> X-On / X-Off |    |                 |          |          |     |         |             |                    |             |       |       |              |      |



Date: 1-19-90

Cable ID: \_\_\_\_\_



**INTERFACE DOCUMENTATION SHEET**

\_\_\_\_\_ EQUIPMENT MANUFACTURER \_\_\_\_\_

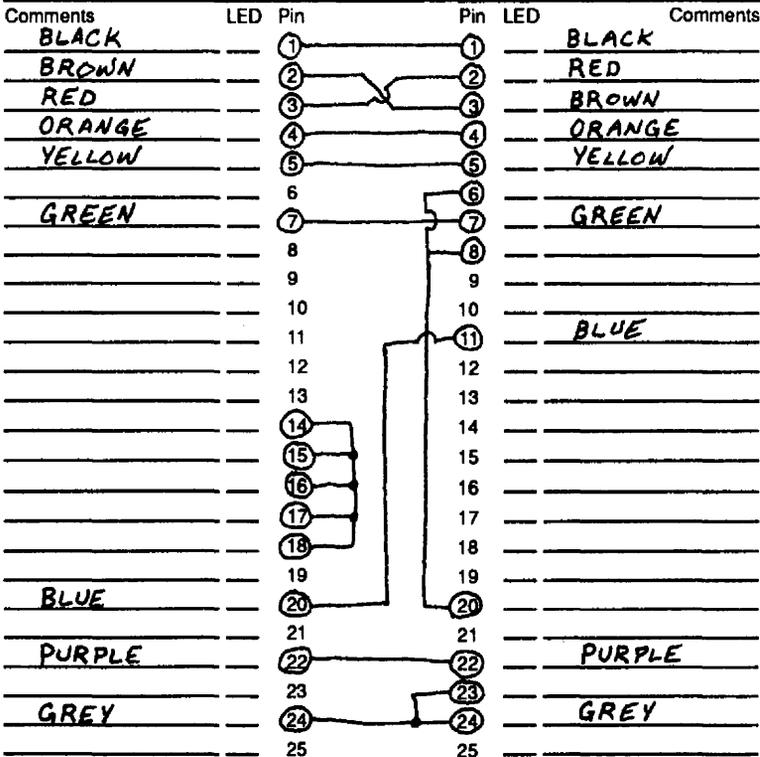
\_\_\_\_\_ MODEL NUMBER \_\_\_\_\_

\_\_\_\_\_ TYPE OF EQUIPMENT \_\_\_\_\_

\_\_\_\_\_ M F EQUIPMENT CONNECTOR M F \_\_\_\_\_

DB25 (M) F CABLE CONNECTOR M (F) DB25

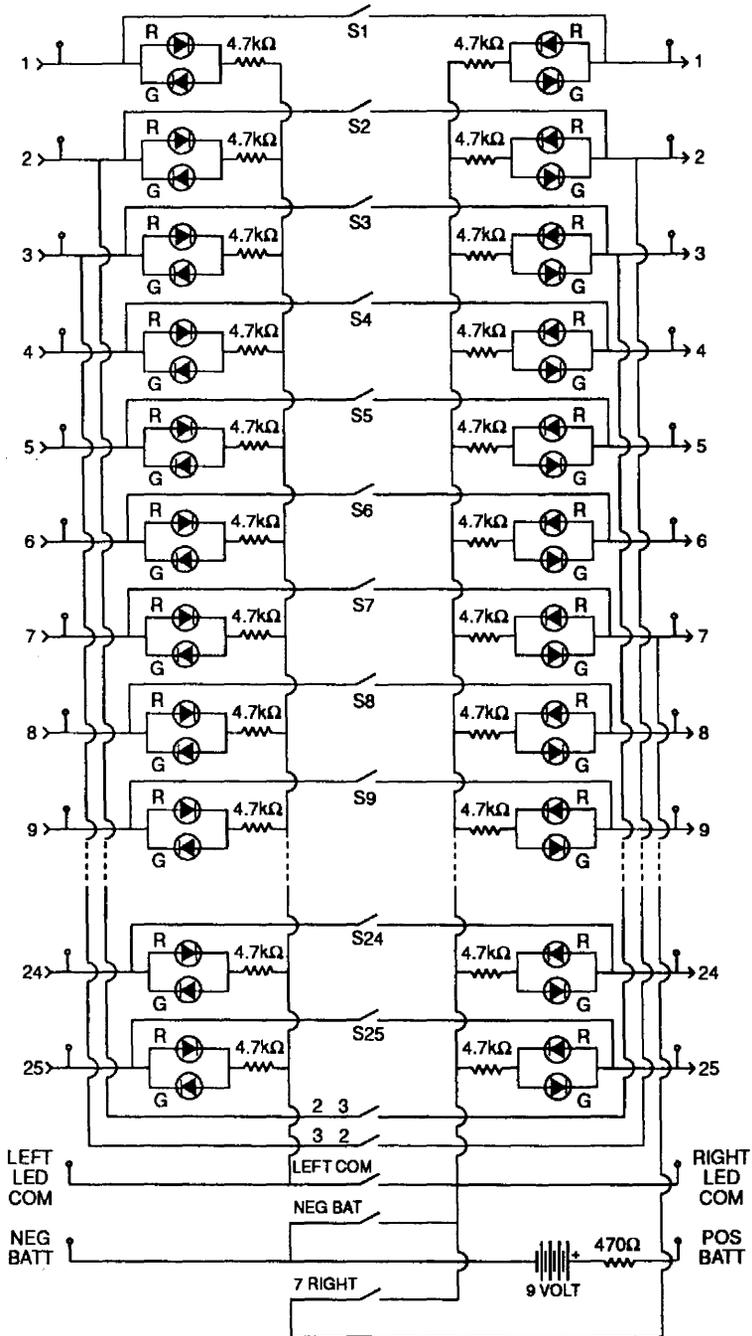
|                 |       |                 |   |
|-----------------|-------|-----------------|---|
| BAUD RATE:      |       |                 |   |
| 50              | 75    | 110             | 150 300 600 1200 2400 4800 9600 19.2K 38.4K |
| # OF DATA BITS: |       | # OF STOP BITS: |   |
| 5               | 6 7 8 | 1               | 1.5 2                                       |
| PARITY:         |       |                 | TRANSMISSION MODE:                          |
| None            | Even  | Odd             | Mark Space                                  |
| HANDSHAKE TYPE: |       |                 | Async Sync                                  |
| Hardware        |       | X-On / X-Off    |   |



Date: 5-27-83

Cable ID: DEMO CABLE

# DATATRACKER DT-5 Schematic



## HELP US HELP YOU

We'd like to know more about you so that we can better serve you and our other customers in the future. Please take a few minutes to fill out both sections below - one to keep as a record of your purchase and the other, postage-paid card, to return to us. When your registration card arrives at DATATRAN, your name will be placed on our mailing list so that we can notify you of product enhancements or new products as they become available.

By having your warranty registration card on file with us, it will also facilitate (should it become necessary) the repair or replacement of your product in accordance with DATATRAN's warranty policy. We want to make sure that you get the maximum performance from your DATATRAN product. Please feel free to give us a call if you have any technical questions or problems. We'll be glad to help.

Thank you for letting us get acquainted with you.

## OWNER'S RECORD

*Please retain this card with your purchase receipt for your records.*

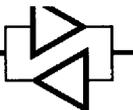
Product \_\_\_\_\_ Model No. \_\_\_\_\_ Serial No. \_\_\_\_\_  
Name \_\_\_\_\_ Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone (\_\_\_\_\_) \_\_\_\_\_ Date \_\_\_\_\_  
Purchased From \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone (\_\_\_\_\_) \_\_\_\_\_ Date \_\_\_\_\_  
Notes \_\_\_\_\_

## WARRANTY REGISTRATION CARD

*To register your warranty coverage with us, please complete and return this limited lifetime warranty card within 30 days of purchase.*

Product \_\_\_\_\_ Model No. \_\_\_\_\_ Serial No. \_\_\_\_\_  
Name \_\_\_\_\_ Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone (\_\_\_\_\_) \_\_\_\_\_ Date \_\_\_\_\_  
Purchased From \_\_\_\_\_  
Comments \_\_\_\_\_

I WOULD LIKE ADDITIONAL INFORMATION ON YOUR FULL PRODUCT LINE



**DATATRAN**  
CORPORATION

**We are a U.S. manufacturer of computer and data communications related test equipment and interconnect accessories.**

**Please address all inquiries to:**

**DATATRAN Corporation  
355 Yuma Street  
Denver, Colorado 80223  
(303) 778-0870**

## **LIMITED LIFETIME WARRANTY**

DATATRAN Corporation (DATATRAN) warrants this product to be free from defects in workmanship and material under normal use and conditions for as long as you use this product. This warranty does not cover damage caused by negligence, misuse, abuse, accidents or unauthorized modifications or repairs outside our factory.

Should service be required by reason of any defect or malfunction under this warranty, please contact DATATRAN for a return merchandise authorization number. Then package the product in a well padded container along with your name and address and ship, postage prepaid, to DATATRAN. Upon verification of the defect or malfunction, DATATRAN will repair, or at its option replace, the product without charge for parts, labor or return postage.

Except as provided herein, DATATRAN makes no warranties, express or implied, including warranties of merchantability or fitness for a particular purpose. Some states do not permit limitation or exclusion of implied warranties; consequently the above mentioned limitation may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

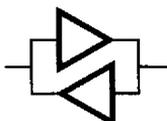


NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS MAIL PERMIT NO. 822 DENVER, CO

POSTAGE WILL BE PAID BY ADDRESSEE



**DATATRAN**  
CORPORATION

**Post Office Box 18623  
Denver, CO 80218-9972**



# ***ATTENTION***

## **IMPORTANT REMINDER**

We'd like to ask you to please take 30 seconds to fill out the enclosed postage-paid Warranty Registration Card. This is our only means of staying in touch with you, our very important customer!

Returning your completely filled out warranty card allows us to keep you informed of product updates, improvements, revisions and optional accessories as they are introduced.

Since your total satisfaction is our #1 goal, please Help Us Help You.

Thank you very much.

*DATATRAN Corporation*