The 'Data Shunt Box'
An easy, inexpensive way to transfer text files between computers

G. S. Herman-Giddens, J. J. Shifflette, M. S. Spach
Duke University, Durham, N. C. 27706

ABSTRACT

A simple, low-cost mechanical switch is used in transfering text files between computers of widely varying size, architecture, manufacturer, and geographic location utilizing standard serial interfaces, typically at 300-1200 baud. No special software is required since the standard utility programs of each computer are used to transmit and receive the text. The only requirement is that the effective rate of character transmission must be slow enough to allow the receiving system to process a character before the next one arrives since no handshaking interlock is provided. With the data shunt box, direct 'media-free' data transfer between different systems is within the budget of any user of small systems.

INTRODUCTION

Users of small system often find it necessary to transfer text, such as programs and data, between computer systems. With the bewildering variety of peripherals available, often it is not possible to find common recording media between two systems. Since peripherals are generally expensive relative to the cost of the computer itself, purchase of special peripherals for media compatibility is not usually practical for users of small systems.

Several years ago our laboratory found itself in the position of many users of small systems - we had access to several systems of differing capability and media and we wanted to transfer source programs and data between the systems. We didn't have the money to buy common peripherals and we didn't want to devote any more scarce programming hours than absolutely necessary to a project. Even a simple program not only requires time to conceive, plan, test, and debug it, but also requires time for documentation, maintenance, and the training of others in how to use it.

To solve the problem of text transfer between systems, we built a simple, inexpensive mechanical switch box to enable us to transfer characters between widely differing computer systems using each system's standard serial interface to a terminal. The serial interface can be accessed directly via a cable, or can be accessed via phone lines using, for example, and acoustic coupler.

CONCEPT

We decided that the simplest and easiest way to transfer text between computers was to let each computer treat the other as a human sitting at the terminal. This approach eliminates the need for special software since most systems already have many utility programs for listing files at a terminal or accepting text from a terminal operator. We established four modes of interconnection between two systems, A and B, and a terminal as shown in figure 1. The transmit and receive signals of each component in each mode is shown by arrows.

Modes I and II use the normal connections between a terminal and the respective system - Mode I for the terminal with system A, and Mode II for the terminal with system B. Mode III allows system A to send

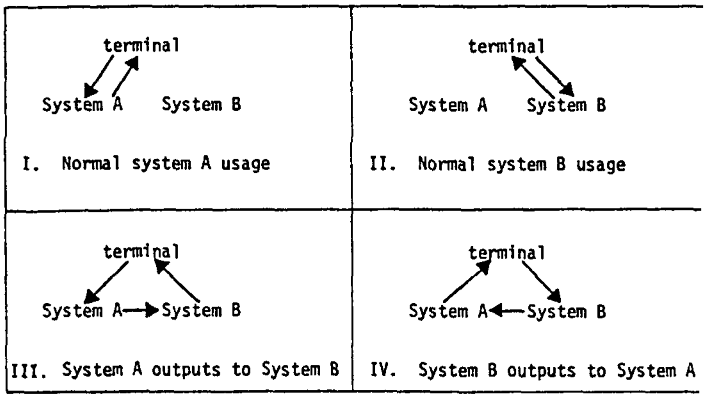


Figure 1. The four modes of the data shunt box with the transmit and receive signals shown by arrows.

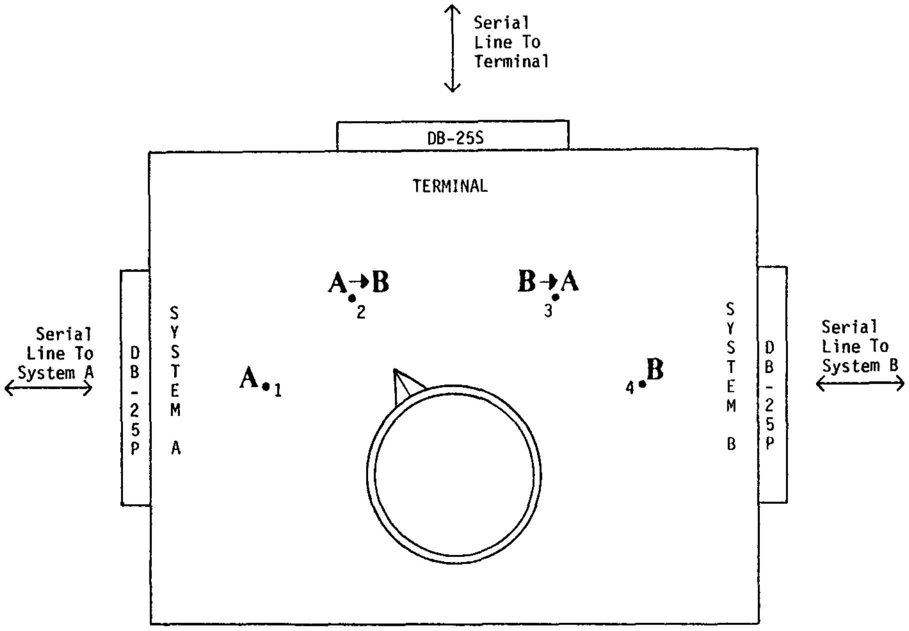


Figure 2. Drawing of data shunt box. Clockwise, the four switch positions each connect the system as follows: 1) A - terminal with system A, 2) A→B - system A to system B, system B to terminal, terminal to system A, 3) B→A - system B to system A, system A to terminal, terminal to system B, 4) B - terminal with system B.

characters to system B instead of the terminal. The terminal operator enters characters at the keyboard which are sent to system A. System B sends characters to the terminal operator. If one wants to transfer text from system A to system B one enters the appropriate text collection commands on system B, and then enters the appropriate text lising commands on system A so that the listed text goes to system B instead of the terminal, and, if system B is set up to echo input, then system B echoes to the terminal operator who can then monitor the transmission of the text from A to B on the screen of the terminal.

The data shunt box consists of three standard RS-232-C connectors and a 4-position multi-pole rotary switch. A drawing of the data shunt box is shown in figure 2. One of the connectors is attached to a standard computer terminal. The other two connectors are attached to each of the two computers using a serial interface and modem, if necessary. The 4-position switch may be set for each of the four possible modes. At positions 1 and 2 the terminal is connected normally to the indicated computer. At positions 3 and 4, each computer, in essence, sees the other as a human entering text from the keyboard of the terminal. Specifically the keyboard of the terminal is connected to the input of one computer while the output of that computer is connected to the input of the second computer. The output of the second computer goes to the screen or printer of the terminal so that the operator can monitor the data transfer.

EXAMPLES OF DATA SHUNT BOX USE

The general operational sequence necessary to transfer files is shown in table 1. Note that it is important that the receiving system be set up first and that the end-of-line character not be entered until the data shunt box switch has been set for system to system transfer. Also, of course, all three ends of the connection, system A, system B, and the terminal, must be set for the same baud rate and terminal characteristics. In actual operation, this sequence is simple since the switch box is adjacent to the terminal and it is convenient to switch control back and forth between systems. In our lab, we use an expanded box which allows us to easily switch between three systems.

Example 1: Sending from a DEC PDP-11/20 (system A) to a DEC LSI-11/2 (system B) using the DEC utility program PIP (Peripheral Interchange Program). Both systems are running the RT-11 operating system.

1. Set the switch to B only operation and set up PIP on the LSI-11/2 to receive text from the terminal by entering:

```
RU PIP          (run PIP program)
FILE1.TXT=TT:   (send to FILE1.TXT from the
                 terminal, TT:)
```

   The LSI-11/2 is now waiting for input.

2. Set the switch to A only operation and set up PIP on the PDP-11/20 to send text to the terminal by entering:

OPERATOR CONTROL SEQUENCE TO TRANSFER A FILE FROM SYSTEM A TO SYSTEM B

1) Set the Data Shunt Box switch to the system B only operation.

2) Using system B standard utility programs, set up system B to receive text from the terminal and to write it to a file on system B. Usually edit or copy utility programs can do this easily.

3) Set the Data Shunt Box switch to the system A only operation.

4) Using system A standard utility programs, set up system A to send text to the terminal from a file, but do not type the end-of-line character which would cause the operation to begin.

5) Set the Data Shunt Box switch to the A B operation.

6) Enter the end-of-line character (typically 'RETURN') on the terminal to cause the transfer of text to begin. Characters are now sent from system A to system B, and are echoed to the terminal (assuming that system B echoes back what it receives).

7) When transmission as seen on the terminal stops, set the Data Shunt Box switch back to the B system only operation and enter the proper end-of-file character or sequence on the terminal for system B.

Table 1. Operator control sequence to transfer a file from system A to system B.

```
RU PIP          (run the PIP program)
TT:=FILE2.TXT   (send to terminal, TT: from
                 FILE2.TXT)
```

Do not enter the line terminating character 'RETURN' at the end of this line which would cause the transfer to begin.

3. Set the switch to A→B operation and then enter the line terminating character:

   'RETURN'

   which causes the transfer to begin.

4. When transmission ends, as seen by observing the system B characters echoed on the terminal, set the switch to B only operation and enter the end-of-file character:

   'CTRL Z'

   which terminates the input to the file FILE1. TXT on the B system (LSI-11/2).

Example 2: Sending a source program from a remote IBM 370 165-2 running TSO (system A) via an acoustic coupler to a local LSI-11/2 (system B).

1. Set the switch to B only operation and set up the LSI-11/2 to receive characters from the terminal:

```
RU PIP          (run the PIP program)
FILE1.TXT=TT:   (send to FILE1.TXT from term-
                 inal, TT:)
```

   LSI-11/2 is now waiting for input.

2. Set the switch to A only operation, establish the phone connection with TSO and run the editor program, QED:

```
QED FILE1 TEXT   (open file FILE1 for editing)
LIST             (list file on terminal)
```

   Do not enter the end-of-line terminating character 'RETURN' after typing LIST.

3. Set the switch to A→B operation and then enter the line terminating character:

   'RETURN'

   which causes the transfer to begin.

4. When transmission ends, as seen by observing system B characters echoed, set the switch to B only operation and enter the end-of-file character:

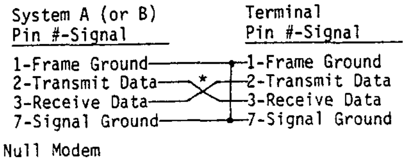
   'CTRL Z'

   which terminates the input to the file FILE1.TXT on the B system (LSI-11/2).

IMPLEMENTATION

A typical RS-232-C signal connection of terminal to computer (1,2) is shown in figure 3. The data shunt box signal connection for sending from system A to system B (Mode III in figure 1) is shown in

STANDARD CONNECTION FOR TERMINAL TO SYSTEM A (or B)



Note: Additional signals are normally found on connectors of the system or the terminal. In order to insure the proper opperation of the data shunt box, it is necessary to pass these signals through the box or as we preferred to do , bypass the signals locally (ie. connecting pin 5 'Clear to Send' to pin 20 'Data Terminal Ready' at the terminal connector in the box).

Figure 3. Typical RS-232-C signal connections for attaching a terminal to a system.

CONNECTIONS FOR SENDING FROM SYSTEM A TO SYSTEM B VIA THE DATA SHUNT BOX

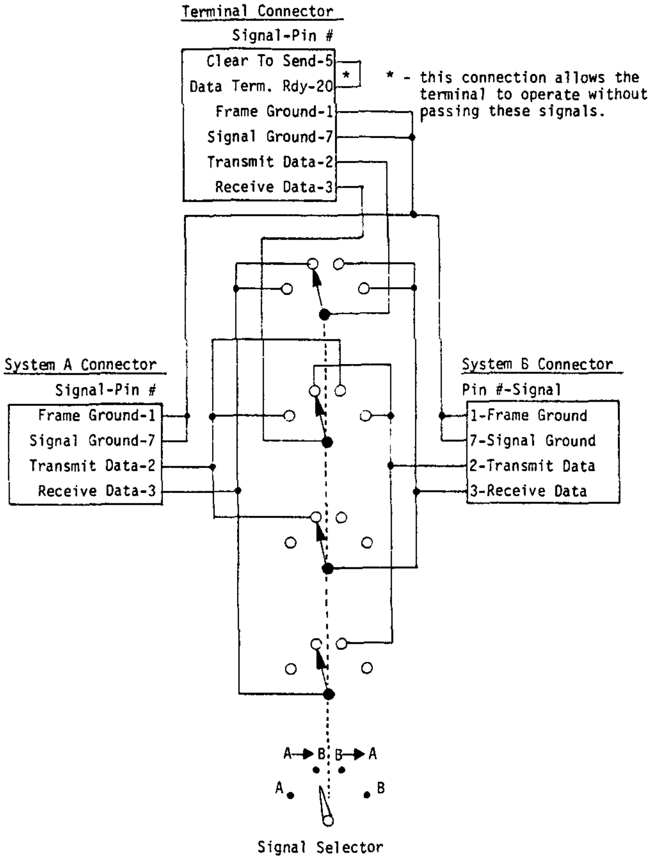
System A Pin #-Signal	Terminal Pin #-Signal	System B Pin #-Signal
1-Frame Ground	1-Frame Ground	1-Frame Ground
7-Signal Ground	7-Signal Ground	7-Signal Ground
2-Transmit Data		3-Receive Data
3-Receive Data	2-Transmit Data	
	3-Receive Data	2-Transmit Data

Note: If System B does not echo the data it receives, the terminal will not monitor the transmission.

Figure 4. Data shunt box signal connections for sending from system A to system B.

figure 4. A simple wiring diagram for the rotary switch in the box and the RS-232-C connections is shown in figure 5. Note that some terminal manufacturers occasionally use signals in addition to those listed in figure 3. If additional signals are necessary, they can be included in the rotary switch network or can be made to bypass the switch, as was done for the terminal's "clear-to-send" and "data-terminal-ready" signals in figure 5.

The data shunt box requires few parts. Table 2 shows the parts needed, a possible supplier for each part, and a current price for each part. It should be possible to make your own box for around $25 (not including the few hours labor). The most complicated part of the construction is the signal interconnection.



* - this connection allows the terminal to operate without passing these signals.

Figure 5. Simplified wiring diagram for rotary switch in data shunt box.

SPECIAL CONSIDERATIONS

Speed: Since the sending and receiving are not synchronized, it is possible to overload the receiving system and miss characters if the baud rate or effective sending rate is too high for the receiving system. The long-term effective sending rate is less than the baud rate in many systems since buffer filling and emptying and operating system response must be taken into account. We have found that 50-120 characters per second is about the maximum possible with a small dedicated system (RT-11) (due to buffering operation), and several larger systems (due to the sometimes degraded response due to other concurrent tasks). If higher speeds are required, then special programs for receiving using larger buffers and/or double buffering, or special programs for sending using stepped down transmission rates can be written. This generally is not worth the effort for casual transfers, but may be worthwhile for transmitting text on a regular basis. We use such a special program for one transfer operation and can run at about 240 characters per second from a DEC PDP-11/45 RSX system to an LSI-11/2 RT-11 system.

DATA SHUNT BOX PARTS LIST

Quanity	Description (a possible source)	Unit Price	Total Price
1	4 Pole 4 Position Rotary Switch (centralab # pa1013)	3.94	3.94
1	Aluminium Box 5"x4"x3" (bud # cu-3005a)	2.30	2.30
1	Sub D Connector - 25 Pin Socket (trw-cinch # db-25s)	5.01	5.01
2	Sub D Connector - 25 Pin Plug (trw-cinch # db-25p)	3.21	6.42
1	Knob (harry davies # 2300)	.20	.20
10'	Wire 22 guage, stranded	1.00	1.00
	Total Cost (current prices)		$ 19.07

Table 2. Data shunt box parts list with a possible supplier and a typical retail price.

Echoing: The data shunt box works well when the receiving system echoes each character providing a visual check of the progress of the transmission. This, however, requires more processing by the receiving system and may cause an overload at the higher baud rates. Most systems allow echoing to be disabled. This can be done if faster response is desired. Operation of terminals in 'batch mode' does not lend itself to using the data shunt box. Likewise, systems (such as TSO) which depend upon 'half-duplex' operation with local echoing by the terminal are not quite so convenient for receiving text.

End-of-file: Since the receiving system is taking characters from an 'operator,' it must be signaled at the end of the text file. Most edit or copy utility programs have a character or character sequence which indicates the end-of-file. This end-of-file sequence usually must be entered on the receiving system by the operator after the file is sent although it is possible to append the sequence to the file on the sending system before transmission.

Text files only: Since the receiving system is expecting characters which are entered by the operator, special characters which might be sent by a binary file could cause problems on the receiving system. Thus transmission should be limited to text files unless one is sure that no special characters could be sent. Sometimes the interaction of special characters between two systems causes problems. For example, the end-of-input sequence for the TSO QED editor is a line containing only a 'RETURN' character. Thus TSO QED cannot receive any file which contains a 'RETURN' as the only character on the line. If such a transfer were necessary, the file being transmitted could be edited first to substitute, for example, a 'space' 'RETURN' sequence for the 'RETURN' character, then the edited file could be sent. Also, it is possible that an extra blank line may be received at both the beginning and the end of the file, depending upon what characters are echoed from the receiving system. These blank lines can be edited out after the file has been received.

SUMMARY

The data shunt box provides a low-cost, simple method of transfering text between different systems, using each system's standard software and a single terminal. For larger volume or higher speed transfers, other methods, such as parallel, synchronized inter-processor transfers (3) or serial network protocols are necessary. Where low volume or occasional transfers are suitable, this switch box works well, requires no software investment, and can usually be made to work quickly on completely foreign systems. With the data shunt box, 'media-free' data transfers between systems is available to any user of small systems.

REFERENCES

1. EIA Standard RS-232-C, Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, Engineering Dept., Electronic Industries Assoc., 2001 Eye St., N.W., Washington, D.C. 20006, August 1969.

2. Industrial Electronics Bulletin No. 9, Application notes for EIA Standard RS-232-C, Electronics Industries Assoc., 2001 Eye St., N.W., Washington, D.C. 20006, May 1971.

3. G. S. Herman-Giddens, J. J. Shifflette, M. S. Spach, "An inter-processor link using DR-11's," to appear in Proc. of the Spring 80 U.S. DECUS Symposium, Chicago, Illinois, April 22-25, 1980.