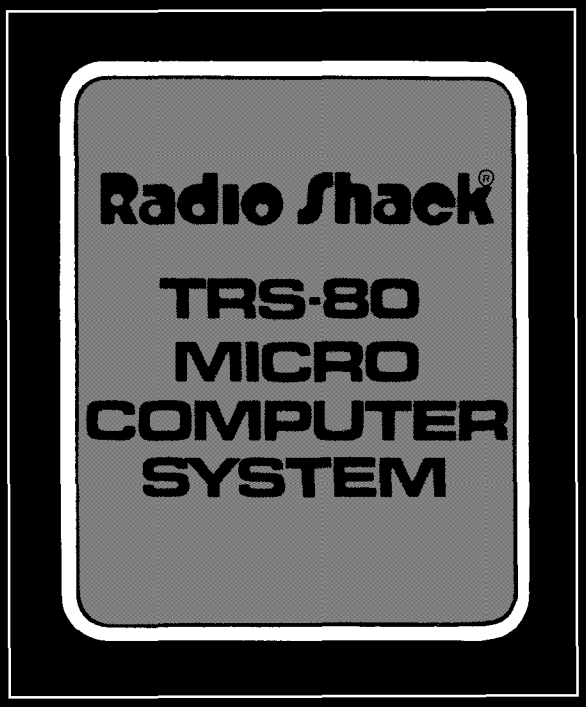
TRS-80 RS-232-C Interface

Catalog Number 26-1145

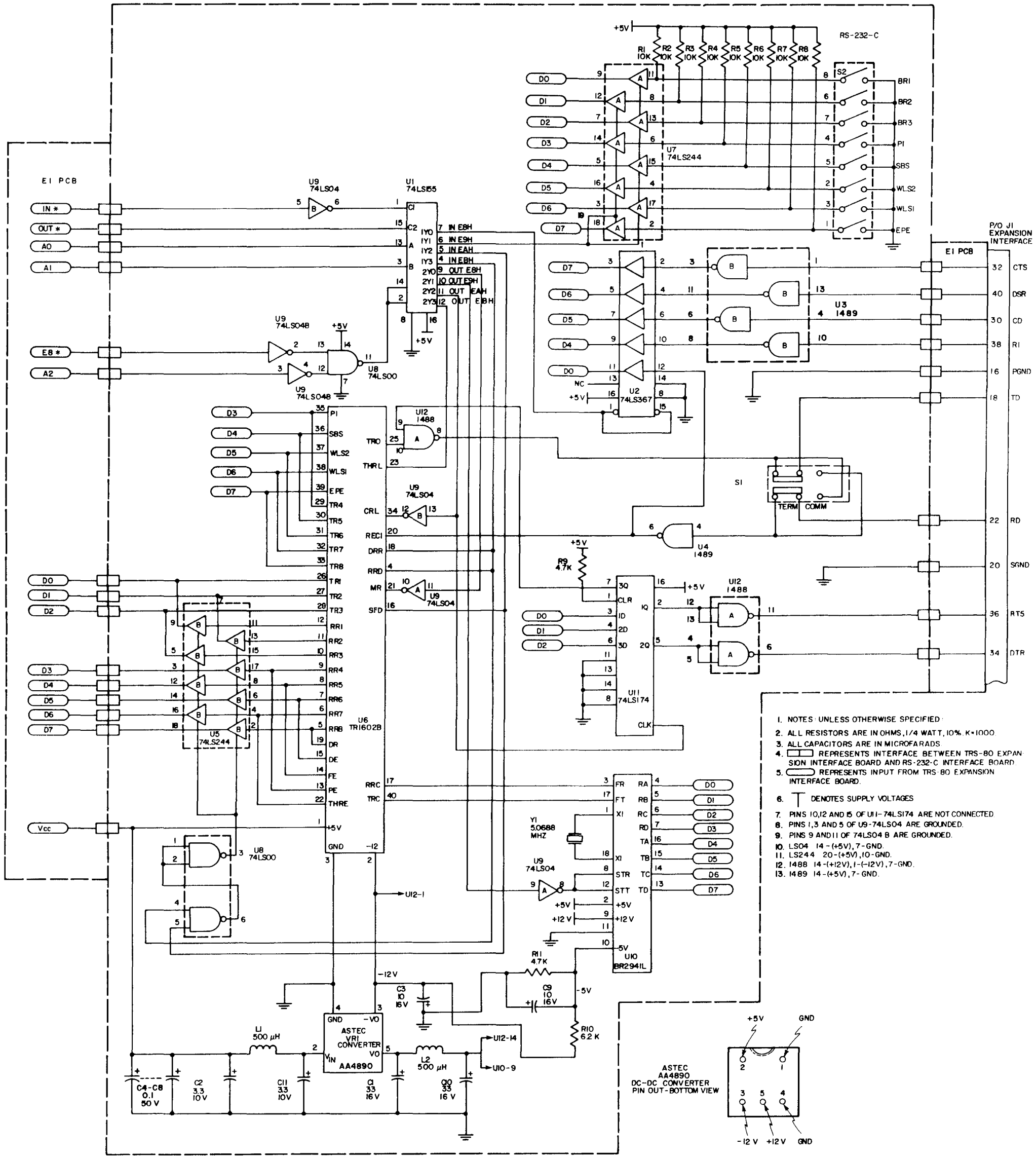


HARDWARE

INTRODUCTION and Discussion of Digital Signals
INSTALLING and Checkout
THEORY OF OPERATION
TERMinal Program and Listing
DECWRITER Application Program

Required Equipment:
TRS-80, Level II with 16K RAM
Expansion Interface with
RS-232-C Installed

CUSTOM MANUFACTURED IN THE USA FOR RADIO SHACK A DIVISION OF TANDY CORPORATION



1. NOTES: UNLESS OTHERWISE SPECIFIED:
2. ALL RESISTORS ARE IN OHMS, 1/4 WATT, 10%. K=1000.
3. ALL CAPACITORS ARE IN MICROFARADS.
4. ▭ REPRESENTS INTERFACE BETWEEN TRS-80 EXPANSION INTERFACE BOARD AND RS-232-C INTERFACE BOARD.
5. ⬭ REPRESENTS INPUT FROM TRS-80 EXPANSION INTERFACE BOARD.
6. ⊤ DENOTES SUPPLY VOLTAGES.
7. PINS 10,12 AND 15 OF U11-74LS174 ARE NOT CONNECTED.
8. PINS 1,3 AND 5 OF U9-74LS04 ARE GROUNDED.
9. PINS 9 AND 11 OF 74LS04 B ARE GROUNDED.
10. LS04 14-(+5V), 7-GND.
11. LS244 20-(+5V), 10-GND.
12. 1488 14-(+12V), 1-(-12V), 7-GND.
13. 1489 14-(+5V), 7-GND.

RS-232-C SCHEMATIC DIAGRAM

To Our Customers

What is an Interface?
It's a generalized means of communication between your TRS-80 and some external device, providing the necessary conventions regarding data-identification, transmission rates, send-receive sequences, error-checking techniques, etc. However, an Interface does **not** provide the programming necessary to **use** any particular TRS-80/ external device system.

For example, having the Interface installed does **not** automatically enable you to send BASIC programs from one TRS-80 to another; to output to a line printer via the Interface; etc. Such applications require "driver programs" which must be custom-designed for the equipment you intend to use.

In this manual you'll find listings for two such "driver programs". One allows your TRS-80 to function as a terminal, the other allows your TRS-80 to output to a popular serial interface line printer, the DECWRITER.

The Radio Shack RS-232-C Interface is designed to meet the EIA standards. However, we cannot guarantee that it will work with all so-called "RS-232-C compatible" devices. Nor do we commit ourselves to provide engineering and programming support for such applications, or other special custom-use situations.

We do, however, guarantee that our Interface will function correctly with all our own RS-232-C equipment.

Introduction

The term RS-232-C refers to a specific EIA (Electronics Industries Association) standard which defines a widely accepted method for interfacing data terminal equipment with data communications equipment. The RS-232-C Interface is by far the most universally used standard for interfacing data processing equipment. Most video terminals, modems, card readers, line printers, mini-microcomputers, etc., utilize the RS-232-C standard for data interchange between devices.

The addition of the RS-232-C to the TRS-80 Expansion Interface opens up a whole new world of compatibility. The RS-232-C Interface is designed to mount inside the TRS-80 Expansion Interface. For example, the external equipment could be a serial line printer, a Radio Shack Telephone Interface, or a video terminal, etc. The point is this, now you can interface with any equipment which is RS-232-C compatible.

A most useful application program of the RS-232-C Interface will permit you to connect your TRS-80 via a Telephone Interface to time sharing computer system. This TERM program is supplied (on cassette tape) with the RS-232-C Interface.

Many TRS-80 owners would like to be able to use their serial printers with the LPRINT command in LEVEL II BASIC. With the RS-232-C Interface installed in the Expansion Interface, you can do that. A programming example and listing for this application is provided in this Manual.

Transmission of Digital Data

The transfer of digital data over relatively long distances is generally accomplished by sending data in serial form using a single twisted wire pair to connect the transmitting and receiving devices. One of two general transmission techniques is commonly used, asynchronous or synchronous. The transmission technique used in the Radio Shack system is asynchronous-bit-serial. Since we don't use the synchronous technique, we'll not mention it again. Asynchronous transmission does not require a synchronizing clock to be transmitted with the data and, the characters need not be contiguous. This means that gaps of varying lengths may be present between transmission of individual characters.

The bits which comprise a data character (generally from five to eight bits in length) and synchronizing start and stop elements are added to each character as shown in Figure 1. The start element is a single logic zero (0) data bit that is added to the front of each character. The stop element is a logic one (1) that is assed to the end of each character. The stop element is maintained until the start element of the next character is transmitted. There is no upper limit to the length of the stop element. However, there is a lower limit that depends on system characteristics. Typical lower limits are 1.0, 1.42 or 2.0 data-bit intervals (although most modern systems use 1.0 or 2.0 stop bits). The negative-going transition of the start element defines the location of the data bits in the character being transmitted. A clock source at the receiver is reset by this transition and is used to locate the center of each data bit.

There are several good reasons for using the asynchronous data transmission system. A clock signal does not need to be transmitted with the data, thus, equipment is simpler. Also, the characters don't need to be sent all at one time; they can be transmitted as they become available. This is particularly useful when transmitting data from manual-entry input devices (e.g. a keyboard). The major disadvantage of asynchronous transmission is that it requires a significant portion of the communications bandwidth for start and stop elements.

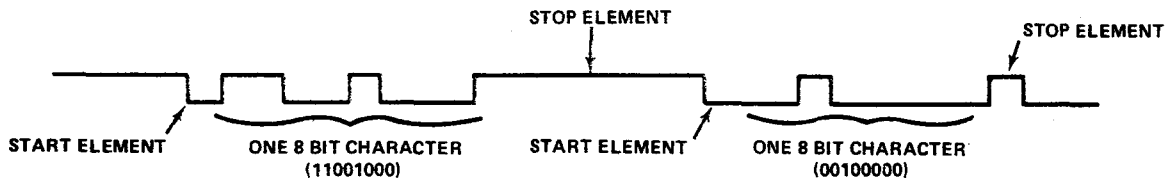


FIGURE 1. ASYNCHRONOUS DATA

The rate at which asynchronous data is transmitted is defined as the **baud rate**. Baud rate is the inverse of the time duration of the shortest signal element. Normally, this is one data bit interval. The baud rate is equal to the bit rate if one stop bit is used; but for systems which use more than one stop bit, the baud rate does not equal the bit rate.

Asynchronous transmission over a simple twisted wire pair can be accomplished at moderately high baud rates (10K baud or higher, depending on the length of wire, type of drivers, etc.). Transmission over the telephone network is generally limited to approximately 2K baud and a modem is required to convert the data pulses to tones that can be transmitted through the telephone network. Radio Shack's Telephone Interface is the ideal modem for this RS-232-C Interface.

Signal Conventions

The E.I.A. RS-232-C electrical specification defines voltage levels and corresponding logic conventions associated with data and control information transmitted between equipment. For data interchange, the signal is considered in the **marking** condition when the voltage measured at the interface point is more negative than –3 Volts(with respect to signal ground). The signal is considered in the **spacing** condition when the voltage is more positive than +3 Volts(with respect to signal ground). The marking condition corresponds to a logic one (1) and the space condition corresponds to a logic zero (0). For timing and control interchange circuits, the function is considered to be"on"when the voltage on the interchange circuit is more positive than +3 Volts(with respect to signal ground); and is considered to be"off"when the voltage is more negative than –3 Volts(with respect to signal ground). The "on"condition corresponds to a logic zero (0) and the"off"condition corresponds to a logic one (1). Table 1 summarizes this information.

NOTATION	INTERCHANGE VOLTAGE	
	Negative	Positive
Binary State	1	0
Signal Condition	Marking	Spacing
Function	OFF	ON

TABLE 1. ON/OFF CONDITION

Pin Designations and Signal Descriptions

The mechanical specification of the RS-232-C requires a 25-pin connector (called a DB-25). Table 2 specifies the pin assignments and signal descriptions as they apply to the Radio Shack RS-232-C Interface.

Pin Number	Abbreviation	Description
1	PGND	Protective Ground
2	TD	Transmit Data
3	RD	Receive Data
4	RTS	Request-to-Send
5	CTS	Clear-to-Send
6	DSR	Data Set Ready
7	SGND	Signal Ground
8	CD	Carrier Detect
20	DTR	Data Terminal Ready
22	RI	Ring Indicator

TABLE 2. PIN DESIGNATIONS AND SIGNAL DESCRIPTION

Protective Ground: This must be bonded to the chassis or equipment frame. It may also be connected to Signal Ground.

Transmit Data: Direction-to data communication equipment. Signals on this circuit are generated by the data terminal equipment for transmission of data to remote equipment. This signal should be held in the marking condition during intervals between characters and at all times when no data is being transmitted.

Received Data: Direction-from data communication equipment. Signals on this circuit are received from remote equipment which transmits data to the terminal. This signal should be held in the marking condition during intervals between characters and at all times when no data is being received.

Request-to-send: Direction-to data communication equipment. This signal is required by the terminal equipment to control the direction of data transmission by the data communication equipment. On one-way or duplex channels, the "on" condition maintains the data communication equipment in the transmit mode. The "off" condition maintains the data communication equipment in the non-transmit mode.

On a half duplex channel, the "on" condition maintains the data communication equipment in the transmit mode and inhibits the receive mode. The "off" condition maintains the data communication equipment in the receive mode.

Clear-to-Send: Direction-from data communication equipment. This signal is generated by the data communication equipment and indicates whether or not the data set (modem) is ready to transmit data. The "on" condition is an indication to the data terminal equipment that the data set can accept data on the Transmit Data circuit. The "off" condition is an indication to the data terminal equipment that it should not transfer data to the data set.

Data Set Ready: Direction-from data communication equipment. This signal indicates the status of the local data set to the data terminal equipment. The "on" condition of this circuit indicates that the data communication equipment is not in test, talk or dial mode and has completed any timing functions required to complete call establishment (answer tone, etc.). The "off" condition will appear at all other times and indicates that the data terminal should accept only Ring Indicator signals and ignore all other signals (appearing on any other interchange circuit).

Data Terminal Ready: Direction-to data communication equipment. This signal is used to control the switching of the data communication equipment to the communications channel. The "on" condition indicates to data communication equipment that it should connect to the communications channel and that it should maintain the connection as long as the "on" condition is present. The "off" condition causes the data communication equipment to be removed from the communications channel following any in-process transmission of data.

Ring Indicator: Direction-from communication equipment. The "on" condition of the circuit indicates that a ringing signal is being received on the communications channel. In general, this means that the data set is being polled and that data communication is desired by the polling device. The "off" condition is held during the off segment of the ringing cycle (between actual rings) and at all other times when ringing is not being received.

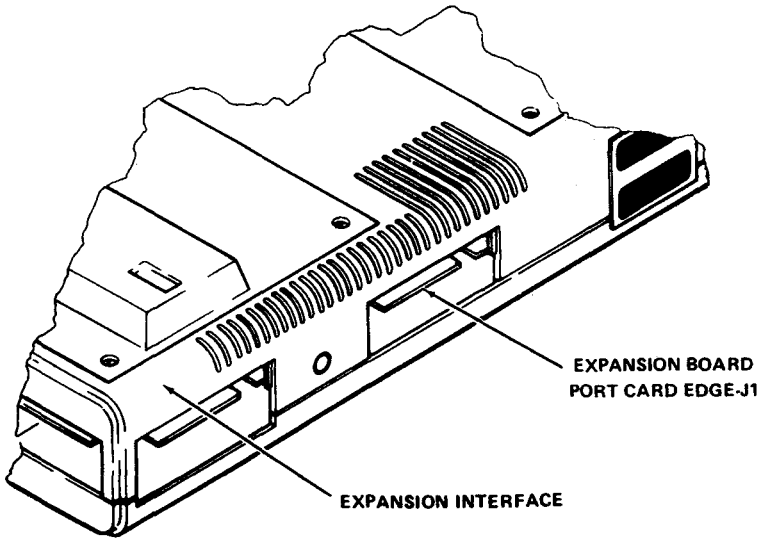
Carrier Detect (Receive Line Signal Detector): Direction-from data communication equipment. When "on", this signal indicates that the data set is receiving a carrier from a remote data set via the communications channel. The "off" condition indicates that no carrier is being received or that the signal quality is unsuitable for data demodulation.

Installing The RS-232-C

Note: Some of the first TRS-80 Expansion Interface units were manufactured with a 42-pin connector (required for the RS-232-C Interface). If your Expansion Interface does not incorporate this connector, you must return it for proper installation of this connector and the RS-232-C. Any attempt to install this connector on your own will automatically void all warranty.

If your Expansion Interface incorporates the necessary 42-pin connector, install the RS-232-C Interface as follows:

1. Position the TRS-80 Expansion Interface as shown in Figure 2 and remove the Expansion Door.
2. Remove two machine screws and washers from the 42-pin connector.
3. Position the RS-232-C Interface as shown. Line up the 42-pin connector mounting holes with the screw holds in the RS-232-C Interface Printed Circuit Board. Fasten with two washers and screws.
4. Position the RS-232-C as shown below (front view). Remove cover from the center of the Expansion Interface. Carefully press the 40-pin connector of the cable supplied with the RS-232-C over the edge connector which will be exposed; **the flat cable must extend downward.** Connect the other end of the cable (DB-25) to the desired Modem or Line Printer, etc.



FRONT VIEW OF EXPANSION INTERFACE

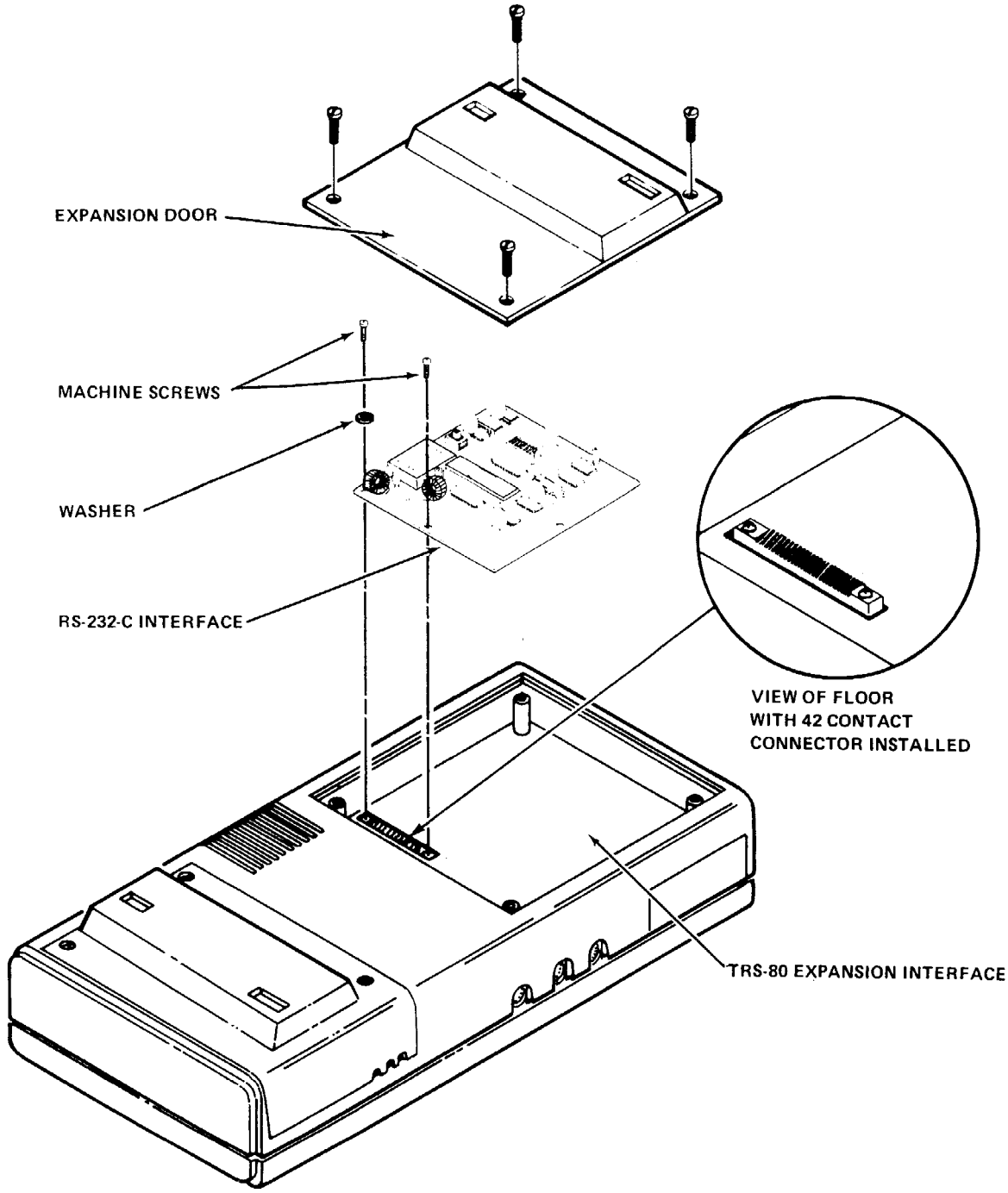


FIGURE 2. RS-232-C INSTALLATION

Checkout

1. Set the RS-232-C Sense Switches (listed in Table 3 on Page 13) for the desired Baud Rate, Parity, etc., as shown in the following example:

Switch	S8	S7	S6	S5	S4	S3	S2	S1
Baud Rate = 300	CLOSED	CLOSED	OPEN					
One stop bit and parity enabled				CLOSED	CLOSED			
Word length = 7 bits of data plus one parity bit						CLOSED	OPEN	
Even parity								OPEN

Also, set TERM/COMM Switch to TERM (assuming you are connecting to a Modem).

2. Load the cassette tape via the Level II BASIC "System" command using "TERM" as the label. Type a slash symbol (/) and press **ENTER** to activate the program (which is located at 5000-Hex).

3. Correct activation of the TERM tape will be indicated by a clearing of the Video Display Screen; the cursor will be at the upper left corner of the Video Display screen.

Note: If you connect together pins 2 and 3 of the 25-pin connector [(DB-25) on the RS-232-C cable], the keyboard output will be "echoed" back to the Video Display Screen, indicating correct operation of the Interface and TERM program.

4. Replace the Expansion Door.

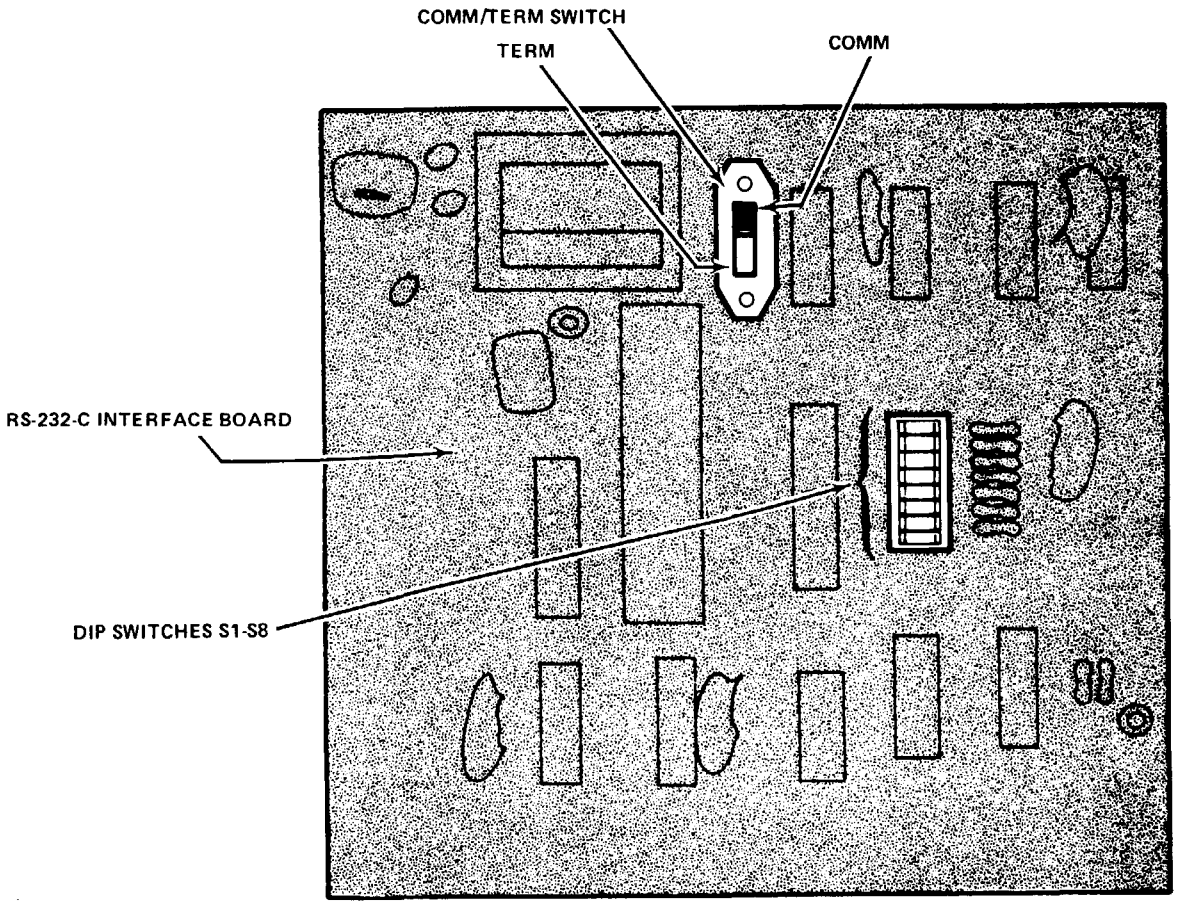


FIGURE 3. RS-232-C SWITCHES

Theory of Operation

The Radio Shack RS-232 Interface is a versatile programmable interface which will permit you to interface with almost any equipment which is RS-232-C compatible. There are essentially two different categories of RS-232-C equipment, namely data terminal equipment and data-comm (data communications) equipment. The Radio Shack RS-232-C Interface is configured from the data terminal's point of view, for interfacing with data-comm equipment.

You can control the configuration of the Interface two ways, by sense switch selection or by direct selection using software control.

Universal Asynchronous Receiver-Transmitter

Figure 4 is a block diagram of the RS-232-C Interface. The heart of the Interface is the UART (Universal Asynchronous Receiver-Transmitter). This MOS LSI device (TR1602A) contains most of the hardware needed to receive and transmit serial data. The UART is an industry standard, general purpose, programmable device for interfacing an asynchronous serial data channel to the parallel data structure of a microprocessor. The transmitter section of the UART converts parallel data from the microprocessor into a serial word which contains the data along with start, parity and stop bits. The receiver section converts a serial word (with start, data, parity and stop bits) into parallel data, and verifies proper data transmission by checking parity and receipt of a valid stop bit.

The operation of the UART can be programmed as follows: The word length can be 5, 6, 7 or 8 bits; parity generation and checking can be inhibited, the parity may be even or odd and the number of stop bits may be either one or two, with one-and-one-half when transmitting a 5-bit code.

The UART contains several internal registers which will permit you to configure the serial data channel, monitor the UART status, load data to be transmitted, and read data that is received on the serial data channel. Later we will discuss the UART registers in more detail.

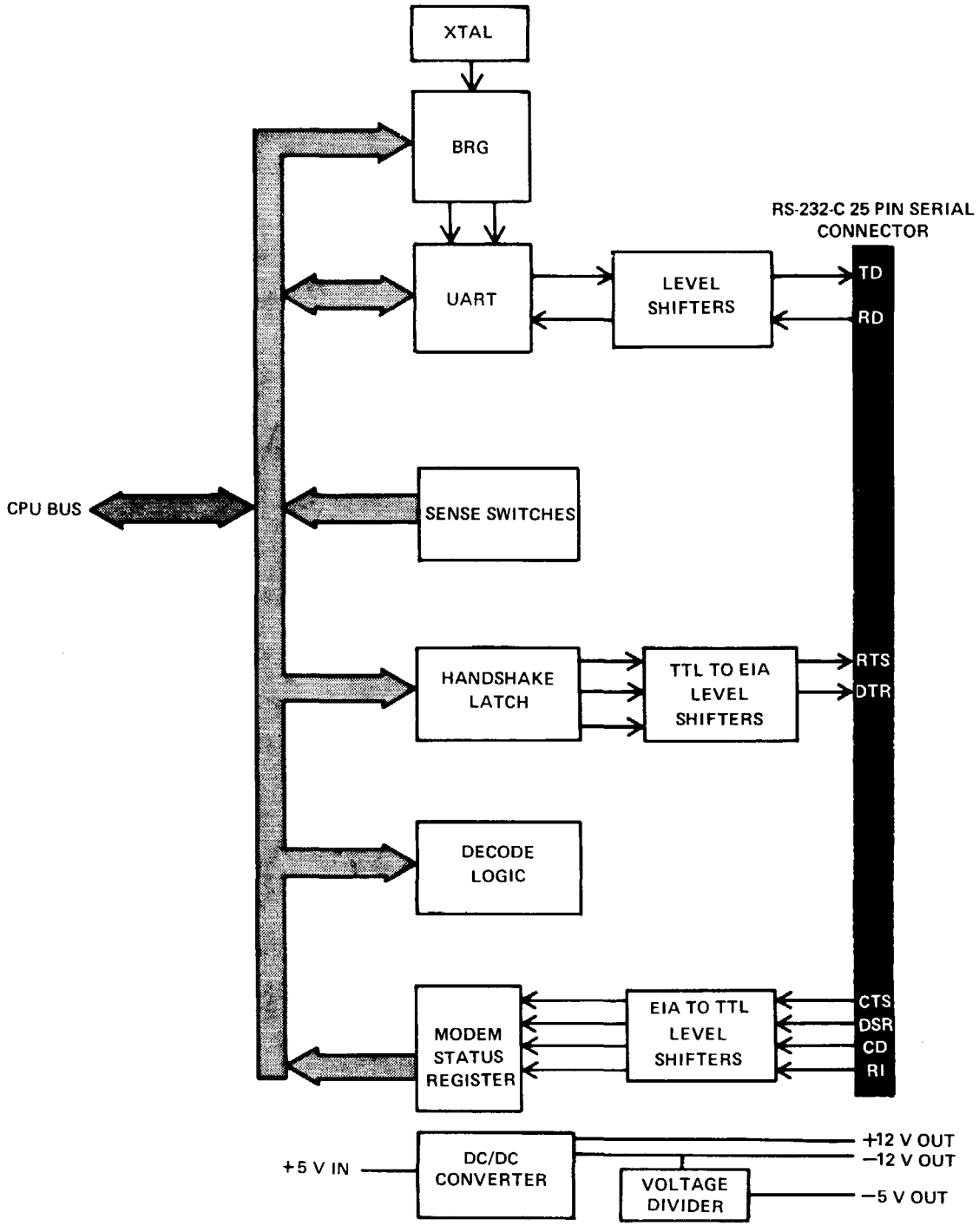


FIGURE 4. RS-232-C BLOCK DIAGRAM

Baud Rate Generator

Directly above the UART in Figure 4 is a block labeled BRG (Baud Rate Generator). This section of the interface outputs two clock signals essential to the proper operation of the UART. The two clock signals (TRANSMIT FREQ & RECEIVE FREQ) determine the transmit and receive baud rates of the serial channel. A 5.0688 MHz crystal is used as a timing reference for the BRG. The BRG can be programmed to divide this reference frequency down to the frequencies needed by the UART. For both transmit and receive operations, the UART requires a clock at 16 times the baud rate desired. The BRG transmit and receive frequency outputs can be independently programmed by loading the appropriate constants into its internal registers. This feature of the BRG makes it possible to transmit and receive data at different baud rates.

Sense Switches

Directly below the UART in Figure 4 is a block labeled SENSE SWITCHES. This section is made up of eight single-pole single-throw switches (S1-S8) which you use to select the baud rate, bits per word, parity (odd or even), stop bits (one, one-and-one-half, or two), and enable or disable parity generation. If you want, you can ignore these switches and under software control, directly configure the Interface for non-standard baud rates, different receive and transmit frequencies, etc. Table 3 summarizes the operation of the sense switches.

Handshake Latch

The Radio Shack RS-232-C Interface can control (with proper software) the logic state of two of the control signals on the Interface (Request-To-Send and Data Terminal Ready). This is accomplished by loading the Handshake Latch with the appropriate bit pattern. Four of the interface signals can be sensed by the CPU (Clear-To-Send, Data Set Ready, Carrier Detect, and Ring Indicator) by reading the Modem Status Register, shown in Figure 4. The Handshake Latch and Modem Status Register, with the appropriate software, allows a handshake dialog to occur between the CPU and the equipment connected to the RS-232-C Interface.

BAUD RATE	S6	S7	S8
110	Closed	Closed	Closed
150	Closed	Closed	Open
300	Open	Closed	Closed
600	Open	Closed	Open
1200	Closed	Open	Closed
2400	Closed	Open	Open
4800	Open	Open	Closed
9600	Open	Open	Open

PARITY ENABLE	S4	STOP BITS	S5
Parity Enable	Closed	One Stop Bit	Closed
Parity Disabled	Open	Two Stop Bits	Open

WORD LENGTH (Excluding parity bit)	S2	S3
5 Bit Word	Closed	Closed
6 Bit Word	Closed	Open
7 Bit Word	Open	Closed
8 Bit Word	Open	Open

PARITY SELECT	S1
Odd Parity	Closed
Even Parity	Open

TABLE 3. OPERATION OF SENSE SWITCHES

Logic Conventions

The logic internal to the RS-232-C operates with TTL logic levels (3.5V or more = Logic 1, and 0.8V or less = Logic 0). The logic convention used for interfacing two RS-232-C devices is EIA levels (−3V or less = Logic one, +3V or more = Logic 0). The logic levels must therefore be converted from one convention to the other when interfacing two devices. The blocks in Figure 4 labeled EIA To TTL and TTL To EIA provide this level conversion.

Port Addresses

The Radio Shack RS-232-C Interface is an I/O-mapped device which uses four port addresses (E8H, E9H, EAH and EBH) for communication with the CPU. I/O-mapped devices in a Z-80 system use the lower address bits (A0-A7) in conjunction with the IN* and OUT* signals to address the desired ports. Figure 5 shows the timing associated with this addressing scheme.

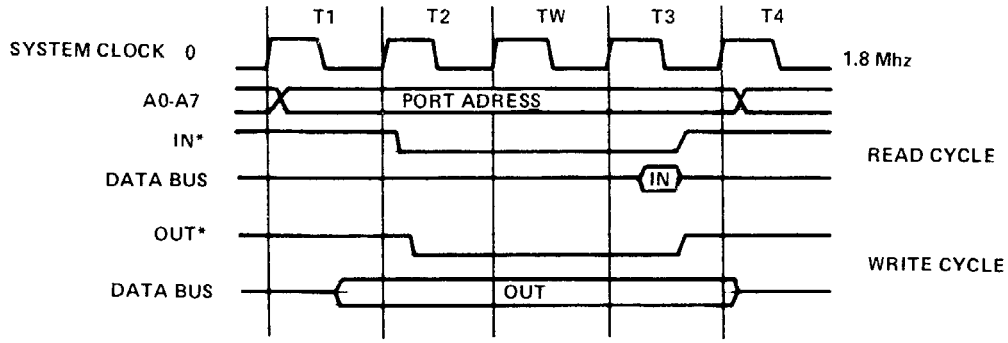


FIGURE 5. I/O TIMING DIAGRAM

Decode Logic

The block labeled DECODE LOGIC in Figure 4 provides eight low-going strobes that are used by the RS-232-C. These strobes allow the CPU to select any of the registers and latches in the RS-232-C Interface for an input or output operation. Table 4 summarizes the I/O port address allocation as implemented in the RS-232-C Interface and the function performed.

ADDRESS	OUT (I/O PORT WRITE)	IN (I/O PORT READ)
E8H	Master Reset (Any Data)	Modem Status Register
E9H	Baud Rate Select	Configuration Sense Switches
EAH	UART Control Register and Handshake Latch	UART Status Register
EBH	Transmit Data Register	Received Data Register

TABLE 4. I/O MAPPED MEMORY ALLOCATION

Each of the above registers performs a specific function related to the RS-232-C Interface. An output to the Master Reset Port (E8H) resets the UART into a known state. This operation should be performed once before attempting to load or read any of the UART's registers. This operation is performed by executing the following Z80 instruction:

```
OUT (ØE8H), A ; OUTPUT TO MASTER RESET LOCATION
```

The contents of A are not important and do not relate in any way to the result of the operation. An input operation from this port address (E8H) causes the contents of the Modem Status Register to be loaded into a register of the CPU. This is accomplished by executing the following Z80 instruction:

```
IN A, (ØE8H) ; READ MODEM STATUS REGISTER
```

Execution of this instruction loads the contents of the Modem Status Register into A. The information now in A represents information relevant to the status of external equipment connected to the Interface.

Any output operation to Port E9H loads the Baud Rate Generator with a constant which determines the receive and transmit baud rates of the serial channel. This is accomplished by executing the following Z80 instructions:

```
LD A, 22H        ; LOAD CONSTANT IN A
OUT (ØE9H), H    ; LOAD BRG WITH CONSTANT
```

Execution of this sequence results in the serial channel being configured to transmit and receive data at 110 baud. Table 5 summarizes the relationship between the constants loaded into the BRG and the resultant baud rates selected. The high-order nibble (D7-D4) loaded into the BRG determines the transmit baud rate, while the low-order nibble (D3-D0) determines the receive baud rate of the serial channel.

Summary of BRG Programming

Radio Shack software supports the baud rates which have a yes in the Switch Selectable column of Table 5. You can program non-standard baud rates by loading the BRG with the appropriate constants. An input operation from port E9H loads a register in the CPU with the bit pattern programmed by the Configuration Sense Switches. As we noted before, the Sense Switch Block in Figure 4 consists of eight single-pole single-throw switches. When a switch is open, it pulls up (+5V) the input of an octal tri-state buffer; when a switch is closed, it pulls an input low.

NIBBLE LOADED	TRANSMIT OR RECEIVE BAUD RATE	10x CLOCK FREQ.	% ERROR	SWITCH SELECTABLE ?
0H	50	0.8 kHz	0	No
1H	75	1.2 kHz	0	No
2H	110	1.76 kHz	0	Yes
3H	134.5	2.1523 kHz	0.016	No
4H	150	2.4 kHz	0	Yes
5H	130	4.8 kHz	0	Yes
6H	600	9.6 kHz	0	Yes
7H	1200	19.2 kHz	0	Yes
8H	1800	28.8 kHz	0	No
9H	2000	32.081 kHz	0.253	No
AH	2400	38.4 kHz	0	Yes
BH	3600	57.6 kHz	0	No.
CH	4800	76.8 kHz	0	Yes
DH	7200	115.2 kHz	0	No
EH	9600	113.6 kHz	0	Yes
FH	19,200	316.8 kHz	3.125	No

TABLE 5. SUMMARY OF BRG PROGRAMMING

The decode logic, in response to an input operation from port E9H, provides a low-going strobe which is connected to the enable input of the tri-state buffer. This strobe, when low, allows the buffer to present the logic state of its inputs to the data bus for loading into the CPU. This operation is performed by executing the following instructions:

```
IN A, (ØE9H) ; LOAD SWITCH SELECTIONS
```

After execution of this instruction, the A register on the CPU will contain the bit pattern which corresponds to the switch settings. This information must now be interpreted by the software and the proper sequence of commands issued by the CPU to configure the Interface.

An output to the Port Address EAH loads the UART Control Register and Handshake Latch. These two functions are separate and distinct, with each using different bits of the register for control. The upper five bits (D7 - D3) of this Port Address determine the word length, stop bits, and parity convention of the serial channel. The lower three bits (D2 - D0) are latched control outputs, one of which disables Transmit Data when low, and the remaining two control Data Terminal Ready and Request-To-Send. A certain degree of caution should be used when outputting to this Port Address to avoid an unwanted interaction. Table 6 summarizes the registers and bit allocations of each for the RS-232-C Interface.

DATA BIT	MODEM STATUS REGISTER	CONFIGURATION SENSE SWITCHES	UART CONTROL REGISTER AND HANDSHAKE LATCH	UART STATUS REGISTER
D7	Clear to send Pin 5 DB-25	Even Parity Enable 1 = even, 0 = odd	Even Parity Enable 1 = even, 0 = odd	Data Received 1 = Condition true
D6	Data Set Ready Pin 6 DB-25	Word length Select 1	Word length Select 1	Transmitter Holding Register Empty 1 = Condition true
D5	Carrier Detect Pin 8 DB-25	Word length Select 2	Word length Select 2	Overrun Error 1 = Condition true
D4	Ring Indicator Pin 22 DB-25	Stop Bit Select 1 = 2 bits, 0 = 1 bit	Stop Bit Select 1 = 2 bits, 0 = 1 bit	Framing Error 1 = Condition true
D3	Unused	Parity Inhibit 1 disables parity	Parity Inhibit 1 disables parity	Parity Error 1 = Condition true
D2	Unused	Baud Rate 3	Break 0 Disables Transmit data	Unused
D1	Receiver Input UART P20	Baud Rate 1	Request to send Pin 4 DB-25	Unused
D0	Unused	Baud Rate 2	Data Terminal Ready Pin 20 DB-25	Unused
	IN ØE8H	IN ØE9H	OUT ØEAH	IN ØEAH

TABLE 6. BIT ALLOCATIONS FOR REGISTERS AND LATCHES

The following example, in Z80 assembly language, shows how to read the Sense Switches. Combine this with the latch bits desired, load the UART Control Register and Handshake Latch, and save the current state of these control bits for later updates if necessary.

```
IN A, (ØE9H)      ; READ SENSE SWITCHES
AND ØF8H          ; ZERO LOWER 3 BITS
OR Ø5H            ; SETS REQUEST TO SEND, RESET DATA
                    TERM READY, AND SETS BREAK
OUT (ØEAH)        ; LOADS UART CONTROL REGISTER &
                    HANDSHAKE LATCH
LD (IMAGE), A     ; SAVES BIT PATTERN FOR UPDATES
```

The label IMAGE in the last line above refers to memory location which is used to save the current state of the UART Control Register and the Handshake Latch. Suppose you want to change the logic state of Request-to-Send. The following example shows how to do this without changing the UART Control Register or the other Handshake bits.

```
LD A, (IMAGE)     ; LOAD CURRENT STATE OF UART
                    CONTROL REGISTER AND HANDSHAKE
                    LATCH
RES Ø,A           ; RESETS BIT ZERO IN A (REQUEST
                    TO SEND)
OUT (ØEAH),A      ; LOADS NEW BIT PATTERN
LD (IMAGE),A      ; SAVE NEW PATTERN FOR UPDATES
```

An input operation to Port Address EAH reads the UART Status Register. Bits D7-D3 convey information about the current status of the UART (Receive Data ready for input, Transmitter Holding Register empty, overrun error, framing error, and parity error). The remaining bits D2-D0 are unused. You can read this register by executing the following Assembly Language instruction.

```
IN A, (ØEAH)      ; READ UART STATUS REGISTER
```

Execution of this instruction loads the "A" register with the contents of the UART Status Registers. Additional software instructions can interpret this status information and determine if a new character can be loaded for transmission, if a new character has been received or if received errors have occurred.

An output operation to Port Address EBH loads the UART Transmitter Holding Register with a new character to be transmitted by the UART. You should not do this until the holding register is found to be empty (meaning that the previous character has been transferred to the transmitter register for transmission). Do this by reading the UART Status Register and checking the proper bit for a logic one (1). The following Assembly Language instructions illustrate how this is done.

```
                        ; OUTPUT CHARACTER TO UART
                          FOR TRANSMISSION
                        ; A SHOULD CONTAIN CHARACTER
                          TO BE TRANSMITTED
        PUSH A          ; SAVE CHARACTER TO BE
                          TRANSMITTED
STATIN  IN A, (ØEAH)    ; LOAD UART STATUS REGISTER
        BIT 6#A         ; TEST TRANSMITTER HOLDING
                          REGISTER FOR A HIGH
        JR Z, STATIN    ; TRY AGAIN IF NOT
        POP AF          ; RESTORE CHARACTER TO BE
                          TRANSMITTED
        OUT (ØEBH),A    ; LOAD HOLDING REGISTER
                          WITH CHARACTER
```

An input operation from Port Address EBH reads the UART Receiver Holding Register and resets the Received Data bit in the UART Status Register. You should not do this until the Received Data bit is set in the UART Status Register (meaning that a complete character has been received and transferred to the Receiver Holding Register). The following sequence of Assembly Language instructions illustrate how this is done.

```
                        ; INPUT A CHARACTER FROM
                          THE UART
                        ; CHARACTER RECEIVED WILL BE
                          IN A
INCHAR  IN A, (ØEAH)    ; LOAD UART STATUS
        BIT 7, A        ; TEST DATA RECEIVED FOR HIGH
        JR Z , INCHAR   ; TRY AGAIN IF NOT
        IN A, (ØEBH)    ; LOAD RECEIVED CHARACTER
```

Execution of this sequence will load the character received into register A and the UART Control Register Received Data bit will be reset.

Earlier we told you that there are two categories of RS-232-C equipment, data terminal equipment and data communications. The two categories differ in which pins on the DB-25 connector are used for the Transmit Data and Receive Data functions. Data terminal equipment transmits data on pin 2 and receives data on pin 3. Equipment which interfaces to data terminal equipment must therefore receive data on pin 2 and transmit data on pin 3 (refer to Table 2). This means that it is sometimes necessary to reverse the operation of these two signals, depending on the specific interface requirements. A double-pole double-throw switch is provided in the Radio Shack RS-232-C Interface for this purpose (refer to Schematic Diagram). One position is labeled TERM (data terminal) and other COMM (communications equipment).

The RS-232-C Interface requires four power supply voltages (+5, −5, +12V and −12V). The Expansion Interface provides the +5V supply from which the other voltages are derived by using a DC-to-DC converter. The DC-to-DC converter used in the Radio Shack RS-232-C Interface inputs +5V and outputs +12V and −12V. A simple resistive voltage divider is used with the −12V to produce the −5V supply (refer to Schematic Diagram). DC-to-DC converters are inherently noisy and must be properly decoupled from the input supply and the outputs must be properly filtered. L1, L2, C1, C10, C11, C3 and C9 perform this decoupling and filtering function.

Terminal Program

The Following Hardware is Required:

* TRS-80 LEVEL II with 16K RAM
* Expansion Interface with RS-232-C installed.

The "TERM"inal program will cause the TRS-80/RS-232-C to function as a full duplex terminal which displays ASCII encoded text inputted from the RS-232-C port, and which outputs keyboard input to the RS-232-C port. (A complete listing of the TERM program is provided later on in this Manual.)

RS-232-C Input to Display

Activation of program clears the screen and the cursor will be positioned at the upper left. As text is input, it is displayed and the cursor increments to the right. Line overflow goes onto the next line (is not truncated). As the screen is filled, incoming text causes it to roll-up (scroll) with the new text directed to the bottom of the screen.

Back space (hex Ø8) and *return* (hex ØD) are recognized and executed by the display program. Other ASCII control codes are ignored. Delay periods are required (null [hex ØØ]) after *return* at high baud rates (1200, 2400, etc.) to allow the display to scroll as indicated in Table 7.

BAUD Rate	#Nulls
110	None
150	None
300	None
600	None
1200	2
2400	4
4800	8
9600	16

TABLE 7. NULLS REQUIRED AFTER *return*

Lower case ASCII will be converted to upper case for display (the TRS-80 cannot display lower case without hardware modification [not supported by Radio Shack]).

Keyboard Output to RS-232-C

The keyboard is continually scanned for new key input signals. New key strobes are converted to ASCII and outputted to the RS-232-C Interface for transmission.

A keyboard entry will not cause an input to the display unless the RS-232-C output is echoed back to the input section of the RS-232-C port. This is called full duplex or FDX.

If you short together pins 2 and 3 on the RS-232-C connector, the terminal can act in a "local" mode and keyboard action will directly display on the CRT. Notice also that such keyboard activity is not stored in memory for later transmission, as is sometimes an option with certain expensive CRT terminals.

The Radio Shack Telephone Modem has a switch that can be set to "half duplex" (or HDX) to allow communication over a telephone line and provide (within the modem coupler) the echo function which allows the TERM program to be used in HDX situations. Most other low speed (300 Baud) modems will also allow this.

Special provision has been made to allow four 7-bit codes (selected by the user) to be output from the keyboard. Refer to Table 8.

Keys pressed	Code at Address	Default Code
shift↓ , A	50 B9H	Ø3H = EOT
shift↓ , B	50 BAH	1BH = ESC
shift↓ , C	50 BBH	FCH = 1
shift↓ , D	50 BCH	FFH = DEL

TABLE 8. KEYBOARD OUTPUT CODES

To generate any of these codes it is necessary to:

(1) hold down the 'shift' key,

(2) hold down the "↓" key,

(3) press A, B, C or D.

The keys may be released in any order.

Other standard control codes can be generated in the same way, with E through O generating Ø5H through ØFH and P through Z generating 1OH through 1SH (you can test this by generating Ø8H shift↓ ,H for a back space).

If the default special codes are not appropriate, load TBUG LEVEL II via the SYSTEM command in BASIC (use the Reset push button) and enter the desired codes into 5ØB9 through 5ØBCH. Use TBUG to re-enter TERM by a jump to 5ØØØH. Note that TERM is located at 5ØØØH through 5ØBC so that it can be resident in RAM along with TBUG.

HINTS and NOTES

1. Shorting pins 2 and 3 of the RS-232-C connector is a good way to test programs and hardware.
2. In data communications, problems will readily occur if you have incorrectly set the even/odd parity, parity enable, baud rate, terminal/data set, (input/output correspondence with pins 2 and 3, etc.).
3. The TERM program does not check for the various error status flags provided with the RS-232-C hardware such as parity error, overflow, etc. If you are a skilled programmer, you will be pleased to know it is possible to use TBUG to make use of this ignored information in various applications.

Program Listing of TERM

The program listing for TERM has been modified (for cassette tape) to include the following "patch" which displays a vertical bar on the Display screen whenever any of the following serial input faults occur:

1. Parity fault
2. Over-run (i.e. UART received another character before the previous one was read in by the TERM program)
3. Framing (i.e. a spacing level was detected by the UART when the "stop bit" period should have been occurring)

You can disable the "patch" by changing:
at 5018, 5019, 501A hex, replace C3, C0, 50 with E6, 7F, FE as shown in the listing.

The patch is:

1. At 5Ø18, 5Ø19, 5Ø1A put C3, CØ, 5Ø
2. At 5ØCØ hex insert:

```
5ØCØ F5          PATCH PUSH AF
5ØC1 3A665Ø             LD A,(5Ø66H)    ; GET UART STATUS
5ØC4 E6 38              AND 38          ; FROM UCB
5ØC6 28 Ø5              JR 7, NOFLT     ; OV, FE, PE?
5ØC8 3E AA              LD A,ØAAH
5ØCA CD 33 ØØ           CALL DSP$       ; DISPLAY BAR
5ØCD F1          NOFLT POP AF
5ØCE E6 7F              AND 7FH
5Ø DØ FE GO             CP GOH
5ØD2 C3 1C 5Ø           JP OUT-OF-PATCH
```

Following is the complete listing of the TERM program.

```
00001                   ;  CASSETTE PROGRAM TO BE SYSTEM LOADED INTO
00002                   ;  LEVEL II TRS-80 WITH RS-232-C INTERFACE WHICH
00003                   ;  CAUSES SYSTEM TO BEHAVE AS A CRT TERMINAL.

00005                   ;  PROG. BY RKUBALA, REVISION THUR 20 JULY 1978.

00007       0033        DSP$    EQU 0033H
00008       002B        KBD$    EQU 002BH
00009       0046        CIO$    EQU 0046H

00011       5000    >           ORG 5000H

00013 5000 3E1C         TERM    LD A,1CH                ;HOME CURSOR
00014 5002 CD3300               CALL DSP$
00015 5005 3E1F                 LD A,1FH                ;CLEAR SCREEN
00016 5007 CD3300               CALL DSP$
00017 500A 3E0E                 LD A,0EH                ;TURN CURSOR ON
00018 500C CD3300               CALL DSP$
00019 500F CD5850   >           CALL MRUART

00021                   ;  MAIN PROGRAM - JUGGLES RS-232-C, KB, & CRT.
```

Term Program Listing (Continued)

```
00023 5012 CD4F50  >  INS       CALL SIN                    ;SEE IF NEW UART INPUT.
00024 5015 B7                   OR A
00025 5016 2812                 JR Z,OUTS                   ;IF NOT, LOOK AT KB.
00026 5018 E67F                 AND 7FH                     ;STRIP PARITY FROM ASCII.
00027 501A FE60                 CP 60H
00028 501C FA2150  >            JP M,$+5
00029 501F E65F                 AND 5FH                     ;CONV. LOWER TO UPPER CASE
00030 5021 FE0A                 CP 0AH
00031 5023 28ED                 JR Z,INS                    ;IGNORE "LF".
00032 5025 CD3300               CALL DSP$                   ;DISPLAY DISPLAYABLE CHAR ON CRT
00033 5028 18E8                 JR INS

00035 502A CD2B00       OUTS    CALL KBD$
00036 502D B7                   OR A
00037 502E 28E2                 JR Z,INS
00038 5030 FE05                 CP 05H
00039 5032 F23D50  >            JP P,NOSPEC
00040 5035 21B850  >            LD HL,SPECTB-1              ;SPECIAL CODES TABLE
00041 5038 4F                   LD C,A
00042 5039 0600                 LD B,0
00043 503B 09                   ADD HL,BC                   ;HL => SELECTED SPEC. CODE
00044 503C 7E                   LD A,(HL)                   ;GET SPECIAL CODE.
00045 503D FE1A         NOSPEC  CP 1AH
00046 503F 28D1                 JR Z,INS                    ;IGNORE SHIFT DOWN-ARROW
00047 5041 CD4650  >            CALL SOUT                   ;OUTPUT KB INPUT TO RS-232-C
00048 5044 18CC                 JR INS
00050                   ;  RS-232-C DRIVER CALL SEQUENCES

00052 5046 114150  >  SOUT      LD DE,SUCB
00053 5049 C5                   PUSH BC
00054 504A 0620                 LD B,20H                    ;OUTPUT BYTE.
00055 504C C34600               JP CIO$

00057 504F 114150  >  SIN       LD DE,SUCB
00058 5052 C5                   PUSH BC
00059 5053 0640                 LD B,40H                    ;INPUT BYTE. IF ANY.
00060 5055 C34600               JP CIO$

00062 5058 114150  >  MRUART    LD DE,SUCB
00063 505B C5                   PUSH BC
00064 505C 0680                 LD B,80H                    ;RESET UART.
00065 505E C34600               JP CIO$

00067                   ;  RS-232-C UNIT CONTROL BLOCK.

00069 5061 E0           SUCB    BYTE 0E0H                   ;FUNCTIONS MASK
00070 5062 6950  >              WORD RS232                  ;DRIVER ADDRESS
00071 5064 00                   BYTE 0                      ;TERM SWITCH CONFIG
00072 5065 00                   BYTE 0                      ;BAUD RATE CODE
00073 5066 00                   BYTE 0                      ;UART STATUS
00074 5067 00                   BYTE 0                      ;MODEM STATUS
00075 5068 00                   BYTE 0                      ;CTRL REG IMAGE
00077                   ;  RS-232-C DRIVER
00078                   ;
00079                   ;    ENTRY:  IX => UCB+0
00080                   ;            C  =  PARAMETER
00081                   ;            B  =  FCT. CODE
00082                   ;
00083                   ;    EXIT:   A  =  STATUS OR DATA

00085 5069 78           RS232   LD A,B                      ;EXAMINE FCT. POST
00086 506A 17                   RLA
00087 506B 3821                 JR C,IUART
00088 506D 17                   RLA
00089 506E 3805                 JR C,RSRD
00090 5070 17                   RLA
00091 5071 380E                 JR C,RSWR
00092 5073 AF           RSX     XOR A
00093 5074 C9                   RET

00095 5075 DBEA         RSRD    IN A,(CTRL)                 ;GET UART STATUS REG
00096 5077 DD7705               LD (IX+5),A                 ;IMAGE TO UCB
00097 507A CB7F                 BIT 7,A                     ;IS RCVD BYTE AVAIL?
```

```
00098 507C 28F5                 JR Z,RSX                ;IF NOT
00099 507E DBEB                 IN A,(DATA)             ;GET DATUM.  DO NOT STRIP PARITY POSN
00100 5080 C9                   RET

00102 5081 DBEA         RSWR    IN A,(CTRL)             ;GET UART STATUS REG  CONTENTS
00103 5083 DD7705               LD (IX+5),A             ;IMAGE TO NCB.
00104 5086 CB77                 BIT 6,A                 ;IS XMIT HOLDING REG EMPTY?
00105 5088 28F7                 JR Z,RSWR               ;IF NOT, WAIT.
00106 508A 79                   LD A,C
00107 508B D3EB                 OUT (DATA),A            ;OUTPUT BYTE
00108 508D C9                   RET
00110       00E8        MR      EQU 0E8H
00111       00E8        MODEM   EQU 0E8H
00112       00E9        CONFIG  EQU 0E9H
00113       00EA        CTRL    EQU 0EAH
00114       00EB        DATA    EQU 0EBH

00116                   ;   INITIALIZE RS-232-C HARDWARE USING CONFIG. SWITCHES

00118 508E D3E8         IUART   OUT (MR),A              ;RESET UART WITH ANY OUT DATA
00119 5090 DBE9                 IN A,(CONFIG)           ;GET TERM CONFIG. SWITCHES STATE
00120 5092 DD7703               LD (IX+3),A             ;SAVE IMAGE.
00121 5095 E6F8                 AND 0F8H                ;MASK OFF BAUD RATE INFO.
00122 5097 F605                 OR 05H                  ;SET BRK, RESET DTR, SET RTS
00123 5099 DD7707               LD (IX+7),A             ;SAVE IMAGE OF CTRL. REG
00124 509C D3EA                 OUT (CTRL),A            ;AND PUT IN CTRL. REG

00126 509E DBE9         IBRG    IN A,(CONFIG)           ;GET BAUD RATE SWITCH STATE
00127 50A0 E607                 AND 07H                 ;BAUD RATE BITS ONLY
00128 50A2 21B150               LD HL,BAUDTB
00129 50A5 0600                 LD B,0
00130 50A7 4F                   LD C,A
00131 50A8 09                   ADD HL,BC               ;HL => BAUD RATE CODE.
00132 50A9 7E                   LD A,(HL)               ;GET BAUD RATE CODE.
00133 50AA DD7704               LD (IX+4),A             ;SAVE IMAGE IN NCB
00134 50AD D3E9                 OUT (CONFIG),A          ;LOAD BAUD RATE GEN.
00135 50AF AF                   XOR A
00136 50B0 C9                   RET

00138                   ;   BAUD RATE CODE TABLE

00140 50B1 22           BAUDTB  BYTE 22H                ;110 BAUD
00141 50B2 44                   BYTE 44H                ;150 BAUD
00142 50B3 55                   BYTE 55H                ;300 BAUD
00143 50B4 66                   BYTE 66H                ;600 BAUD
00144 50B5 77                   BYTE 77H                ;1200
00145 50B6 AA                   BYTE 0AAH               ;2400
00146 50B7 CC                   BYTE 0CCH               ;4800
00147 50B8 EE                   BYTE 0EEH               ;9600

00149                   ;   TABLE OF SPECIAL CODES.

00151 50B9 03           SPECTB  BYTE 03H                ;DEFAULT:  EOT - CTRL"A"  (ALSO "ATTN")
00152 50BA 1B                   BYTE 1BH                ;DEFAULT:  ESC - CNTRL"B"
00153 50BB 7C                   BYTE 7CH                ;DEFAULT:  VERT BAR - CNTRL"C"
00154 50BC 7F                   BYTE 7FH                ;DEFAULT:  DEL - CTRL"D"

SCALARS

   CIO$ --- 0046            CONFIG - 00E9           CTRL --- 00EA           DATA --- 00EB
   KBD$ --- 002B            MODEM -- 00E8           MR ----- 00E8

%TO (DEFAULT) SECTION (50BD)

   BAUDTB - 50B1            IBRG --- 509E           INS ---- 5012           IUART -- 508E
   NOSPEC - 503D            OUTS --- 502A           RS232 -- 5069           RSRD --- 5075
   RSX ---- 5073            SIN ---- 504F           SOUT --- 5046           SPECTB - 50B9
   TERM --- 5000


  156 SOURCE LINES      156 ASSEMBLED LINES
```

Example Applications Program for Line Printer

A common application for the Radio Shack RS-232-C is the output of serial data to a line printer.

Before continuing with the specifics of the software, an overview of the I/O driver organization of Level II BASIC is in order. Shortly after power on in Level II BASIC, a set of addresses relating to the I/O drivers is written into the lower part of RAM. These addresses direct the BASIC software to each of the I/O drivers as needed (i.e. LPRINT, PRINT, CLOAD, CSAVE, LIST, LLIST, etc.).

These addresses remain in low memory as long as power is supplied to the computer or unless they are written over with new information. In general, these addresses are not written over in normal operation and are purposely located in an area of memory which is not disturbed by Level II BASIC. We can, however, purposely write over the addresses in order to direct BASIC to an I/O driver of our own design. The addresses noted above, along with other I/O specific information, are referred to as "device control blocks" or DCB's. The DCB for the Line Printer begins at memory address 4Ø25H and is organized as shown below.

MEMORY ADDRESS	MEMORY CONTENTS
4025H	DCB TYPE (Input, or output operation)
4026H	DRIVER ADDRESS (Low order Byte)
4027H	DRIVER ADDRESS (High order Byte)

The line printer driver normally referred to by the DCB is located in the Level II ROMS and expects a Centronics parallel type interface to be utilized. If we purposely alter this DCB to point to an address in RAM and place our own I/O driver at this address, we can utilize the Level II commands which refer to the line printer (LPRINT, LLIST) with the serial line printer.

For this example let's assume that we wish to interface a Level II TRS-80 to a serial line printer operating at 300 baud. The line printer expects a seven-bit word, one stop bit, and even parity. Let us also assume that the printer has one output line which can be tested to determine if the printer can accept data. When status line is low it means that the printer is not busy and can accept data from the Interface; when it is high, it means that the printer is busy and cannot accept data for the time being.

This status information from the printer can be tested by the TRS-80 CPU if we connect it to one of the four inputs available on the RS-232-C Interface (CTS, DSR, DC, RI). A jumper wire from pin 6 of the printer connector (assuming the standard DB-25 is used on the printer) to the "status output" of the printer should be the only modification needed.

The printer should now be ready to print if we can provide the proper software to interface with it. Before we can actually output data, the printer and the RS-232-C Interface must be properly initialized as to the baud rate, word length, parity convention and stop bits required by the printer. This is one of the tasks which must be taken care of by the driver software. The assembly language listing on pages 27 and 28 provides Software to drive a DECWRITER. Lines 180 through 440 of the listing comprise the initialization section of the driver routine. This section of code tests a flag in memory to see if the Interface has been initialized, and, if so, branches around the rest of the initialization section and goes directly to the actual output section of the code (lines 500 through 620). The initialization occurs the first time the driver program is called by BASIC and a flag is set indicating this fact. This section of code is bypassed on successive calls to the driver routine.

Assembly Language DECWRITER Driver Program

```
00E8          00100 RESURT  EQU 0E8H ; AN OUT TO THIS LOC RESETS THE UART, AN IN READS THE RS232 CONTROL BITS
00E9          00110 SWITCH  EQU 0E9H ; AN OUT TO THIS LOC LOADS THE BAUD RATE GENERATOR, AN IN READS THE SENSE SWITCHES
00EA          00120 CNTREG  EQU 0EAH ; AN OUT TO THIS LOC LOADS THE UART CONTROL REGISTER, AN IN READS THE UART STATUS REG.
00EB          00130 DTAREG  EQU 0EBH ; AN OUT TO THIS LOC LOADS TH UART XMIT HOLDING REG., AN IN READS THE RECEIVED DATA
7F00          00140         ORG 7F00H       ; 32512D ORG FOR DRIVER
              00150 ; RS232C OUTPUT DRIVER TO BE USED WITH THE LPRINT COMMAND IN LEVEL II BASIC.
              00160 ; THE DRIVER IS POKED INTO HIGH MEMORY (16K MACHINE) AND THE DEVICE CONTROL BLOCK
              00170 ; CHANGED TO VECTOR THE LPRINT COMMAND TO THE RS232C DRIVER WITH A SHORT BASIC PROGRAM.
7F00 E5       00180 INIT    PUSH HL         ; SAVE REG. USED
7F01 C5       00190         PUSH BC
7F02 F5       00200         PUSH AF
              00210 ; THIS SECTION OF CODE IS USED TO INITIALIZE THE RS232C INTERFACE TO CORRESPOND TO
              00220 ; THE OPTIONS SPECIFIED BY THE SENSE SWITCHES IN THE INTERFACE (BAUD RATE, STOP BITS, BITS/CHAR. ETC.)
7F03 3A487F   00230         LD A,(FLAG)     ; CHECK FLAG TO SEE IF UART AND BRG HAVE BEEN INIT.
7F06 FE01     00240         CP 01H
7F08 2820     00250         JR Z,RESTOR     ; RESTORE REG. AND OUTPUT CHAR IF SO
7F0A 3E01     00260         LD A,01H
7F0C 32487F   00270         LD (FLAG),A     ; SET FLAG TO INICATE INIT.
7F0F D3E8     00280         OUT (RESURT),A  ; READ 37DCH TO RESET UART
7F11 DBE9     00290         IN A,(SWITCH)   ; READ SENSE SWITHES
7F13 E6F8     00300         AND 0F8H        ; LOP OFF LOWER 3 BITS
7F15 F604     00310         OR 04H          ; RESETS RTS,RESETS DTR,SETS BRK IN HANDSHAKE LATCH
7F17 32477F   00320         LD (SWTIMG),A   ; LOAD SWTIMG W/IMAGE OF LATCH BITS
7F1A D3EA     00330         OUT (CNTREG),A  ; LOAD UART W/SWITCH IMAGE
7F1C DBE9     00340 BAUDST  IN A,(SWITCH)   ; SET BAUD RATE ACCORDING TO SWITCH SELECTION
7F1E E607     00350         AND 07H         ; LOP OFF UPPER 5 BITS
7F20 213F7F   00360         LD HL,BDTABL    ; POINT TO FIRST LOC IN BAUD TBL
7F23 0600     00370         LD B,00H        ; ZERO B REG
```

```
7F25 4F      00380         LD C,A          ; PUT OFFSET IN C
7F26 09      00390         ADD HL,BC       ; ADD OFFSET TO HL
7F27 7E      00400         LD A,(HL)       ; LOAD POINTED VALUE
7F28 D3E9    00410         OUT (SWITCH),A  ; LOAD BRG W/TABLE VALUE
7F2A F1      00420 RESTOR  POP AF
7F2B C1      00430         POP BC
7F2C E1      00440         POP HL          ; RESTORE REG.
             00450 ; THIS SECTION OF CODE DOES THE ACTUAL OUTPUT OF THE CHAR TO THE UART FOR SERIAL XMIT.
             00452 ; IT FIRST CHECKS THE UART TO SEE IF IT'S HOLDING REGISTER IS EMPTY, LOOPS UNTIL IT IS
             00453 ; AND LOADS THE CHARACTER  TO BE TRANSMITTED TO THE HOLDING REGISTER.  IF THE CHAR WAS A CARRIAGE
             00454 ; RETURN A LINE FEED IS ALSO OUTPUT.
7F2D DBEA    00490 STATIN  IN A,(CNTREG)   ; LOAD UART STATUS
7F2F CB77    00500         BIT 6,A         ; TEST THRE FOR HIGH
7F31 28FA    00510         JR Z,STATIN     ; LOOP IF NOT
7F33 79      00520         LD A,C          ; LOAD A W/CHAR TO BE OUTPUT
7F34 D3EB    00530         OUT (DTAREG),A  ; LOAD HOLDING REG W/CHAR
7F36 FE0D    00540         CP 0DH          ; IS IT A CARRIAGE RET ?
7F38 2004    00550         JR NZ,RETRN     ; RETURN IF NOT
7F3A 0E0A    00560         LD C,0AH        ; IF SO OUTPUT A LINE FEED ALSO
7F3C 18EF    00570         JR STATIN       ; OUTPUT TO UART
7F3E C9      00580 RETRN   RET             ; RETURN TO CALLING CODE
             00590 ; THE FOLLOWING TABLE DEFINES THE BAUD RATE SELECTED BY THE SENSE SWITCHES IN THE INTERFACE
7F3F 22      00600 BDTABL  DEFB 22H        ; 110 BAUD
7F40 44      00610         DEFB 44H        ; 150 BAUD
7F41 55      00620         DEFB 55H        ; 300 BAUD
7F42 66      00630         DEFB 66H        ; 600 BAUD
7F43 77      00640         DEFB 77H        ; 1200 BAUD
7F44 AA      00650         DEFB 0AAH       ; 2400 BAUD
7F45 CC      00660         DEFB 0CCH       ; 4800 BAUD
7F46 EE      00670         DEFB 0EEH       ; 9600 BAUD
7F47 00      00680 SWTIMG  DEFB 00H        ; IMAGE OF HANDSHAKE LATCH
7F48 00      00690 FLAG    DEFB 00H        ; FLAG TO INDICATE INITIALIZATION
0000         00700         END
00000 TOTAL ERRORS
RETRN   7F3E
STATIN  7F2D
BDTABL  7F3F
BAUDST  7F1C
SWTIMG  7F47
RESTOR  7F2A
FLAG    7F48
INIT    7F00
DTAREG  00EB
CNTREG  00EA
SWITCH  00E9
RESURT  00E8
```

BASIC Program for DECWRITER

```
10 REM ** POKE NEW DCB TYPE AND ADDRESS IN RAM (4025H) **
20 POKE 16421,2:POKE 16422,0:POKE 16423,127
30 REM ** POKE RS-232-C I/O DRIVER INTO HIGH MEM (7F00H) **
40 FOR X=32512 TO 32584
50 READ Y
60 POKE X,Y
70 NEXT X
75 END
80 DATA 229,197,245,58,72,127,254,1,40
90 DATA 32,62,1,50,72,127,211,232,219,233
100 DATA 230,248,246,4,50,71,127,211,234
110 DATA 219,233,230,7,33,63,127,6,0,79,9
120 DATA 126,211,233,241,193,225,219,234
130 DATA 203,119,40,250,121,211,235,254
140 DATA 13,32,4,14,10,24,239,201,34,68
150 DATA 85,102,119,170,204,238,0,0
```

The initialization code must reset the UART, read the configuration sense switches, load the UART with the control information specified by the switches (stop bits, word length, and parity convention), load the BRG with the correct code for the baud rate specified by the switches, and set the initialization flag in memory.

Lines 500 through 620 accomplish the actual output of data to the printer. The software first checks to see if Data Set Ready is low and loop-tests each time if not. When found low, it checks the UART's transmitter holding register status-bit for a high (indicating the UART can accept a new character for transmission), loops if not, or loads the UART holding register with the next character if so. The character just loaded is then tested to see if it was a carriage return (ØDH) and if so, a line feed (ØAH) is transmitted also. This line feed may or may not be necessary depending upon the specific printer being interfaced. (This part of the code can be deleted if desired.) Control is now transferred back to BASIC.

Following the assembly language driver on page 28 is a BASIC program listing. This BASIC program accomplishes two tasks, writing a new address in the line printer DCB (line 20) and writing the actual printer driver code into high memory (lines 40 through 120). Lines 80 through 150 are data statements which contain the decimal equivalents of each byte of the driver object code shown in the second column of the assembly language listing on pages 27 and 28.

To use this program, see page 30.

Assuming the specific interface requirements are identical to this example, now all you do is type in the BASIC program listed on page 28, run it and now you are ready to use a serial line printer with BASIC. A step-by-step illustration of this procedure is shown below.

1. Power up TRS-80 and Expansion Interface.
   MEMORY SIZE?
   should be displayed on the Video Monitor.
   You should respond with:
   32511
   and press **ENTER** (Decimal equivalent of 7EFFH).
   This response reserves the top 256 bytes of a 16K TRS-80 for the printer driver.

2. Set configuration switches for desired baud rate, stop bits, word length and parity as required. See esample on page x.

3. Type in the BASIC program listed on page xx with appropriate modifications for specific application or load from cassette a previously saved version of the above.

4. Run the BASIC program.

5. Connect the Line Printer to the RS-232-C with the cable provided (or an appropriately modified cable).

6. Now you can use a serial line printer with the Level II BASIC LPRINT and LLIST commands.

Any BASIC programs which utilize this technique must include the BASIC program on page 29 as a part of the coding and this section of code must be executed prior to using the serial line printer. If you do not want to locate the object code for the line printer driver at 7FØØH (as in this example), you can reassemble the source code on page 27 with a different origin (line 140) using the TRS-80 Editor Assembler and modify the BASIC program to correspond to this new location. The response to the memory size question must also be modified to be the decimal equivalent of one byte less than the beginning address of the line printer driver object code. The data statements in the BASIC program on page xx must be altered to reflect any change in the object code due to relocation or modification. The last two poke statements on line 20 on page 28 must be changed to correspond to the decimal equivalent of the new beginning address of the driver object code (argument of second poke statement is the low-order byte of the new address, and the argument of the third poke statement is the high-order byte of the new address).

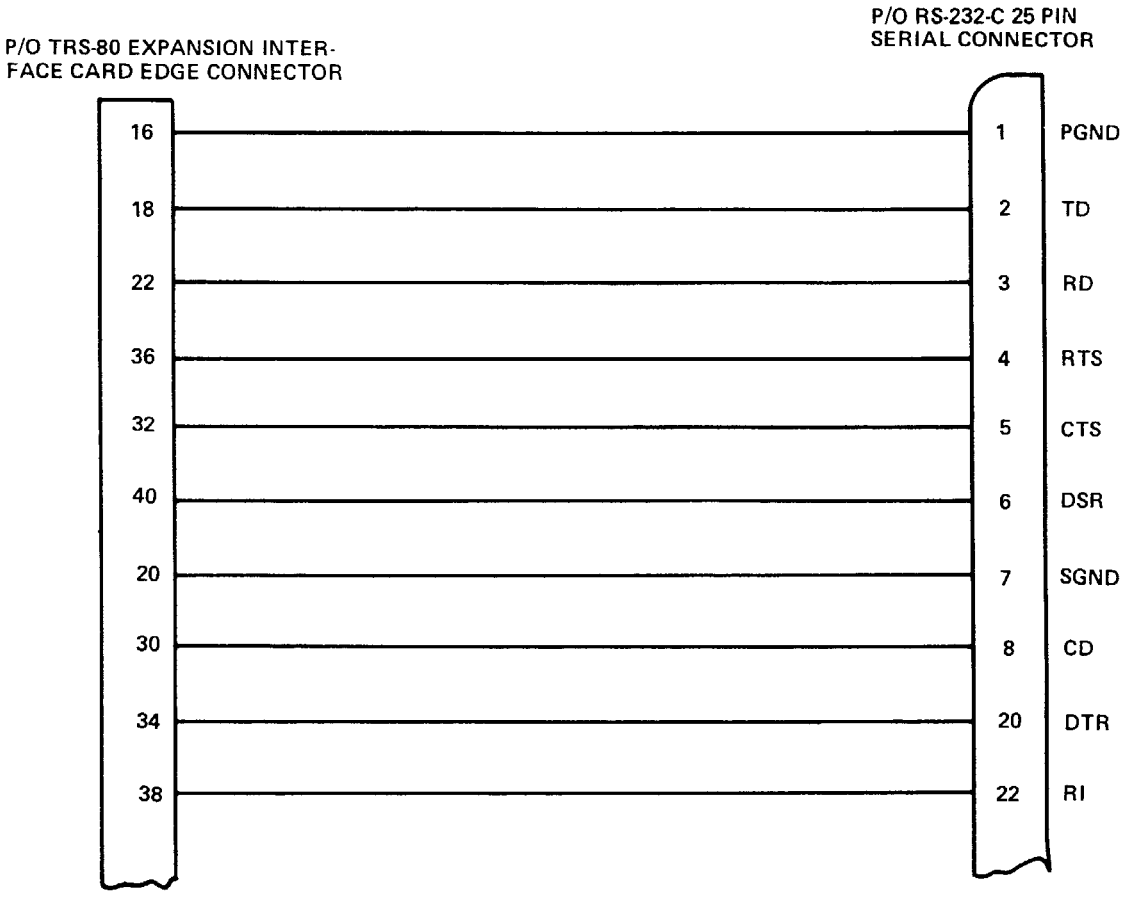


FIGURE 6. RS-232-C CABLE (40-CONDUCTOR TO 25-PIN CONNECTOR)

LIMITED WARRANTY

Radio Shack warrants for a period of 90 days from the date of delivery to customer that the computer hardware described herein shall be free from defects in material and workmanship under normal use and service. This warranty shall be void if this unit's case or cabinet is opened or if the unit is altered or modified. During this period, if a defect should occur, the product must be returned to a Radio Shack store or dealer for repair. Customer's sole and exclusive remedy in the event of defect is expressly limited to the correction of the defect by adjustment, repair or replacement at Radio Shack's election and sole expense, except there shall be no obligation to replace or repair items which by their nature are expendable. No representation or other affirmation of fact, including but not limited to statements regarding capacity, suitability for use, or performance of the equipment, shall be or be deemed to be a warranty or representation by Radio Shack, for any purpose, nor give rise to any liability or obligation of Radio Shack whatsoever.

EXCEPT AS SPECIFICALLY PROVIDED IN THIS AGREEMENT, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND IN NO EVENT SHALL RADIO SHACK BE LIABLE FOR LOSS OF PROFITS OR BENEFITS, INDIRECT, SPECIAL, CONSEQUENTIAL OR OTHER SIMILAR DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR OTHERWISE.

RADIO SHACK A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA	BELGIUM	U K
280-316 VICTORIA ROAD	PARC INDUSTRIEL DE NANINNE	BILSTON ROAD WEDNESBURY
RYDALMERE, N.S.W. 2116	5140 NANINNE	WEST MIDLANDS WS10 7JN

PRINTED IN U.S.A.