7134/ 0261

DRAFT

RADIO SHACK

TRS-80

RS-232-C INTERFACE

OWNER'S TECHNICAL MANUAL

26-1145

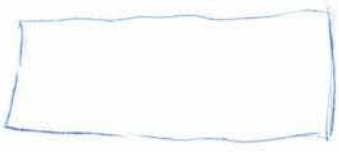


TABLE OF CONTENTS

INTRODUCTION

The term RS-232-C refers to a specific EIA (Electronic 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 to the user. The RS-232-C Interface is designed to mount inside the TRS-80 Expansion Interface. A cable is also provided for connection to external equipment. The external equipment could be, for example, a serial line printer, Radio Shack Telephone Interface, or a video terminal. The list of equipment could continue indefinitely! The point is, the user can now easily interface with any equipment which is RS-232-C compatible.

One application program of the RS-232-C Interface, which will be useful to many TRS-80 and Expansion Interface owners, allows this hardware combination in conjunction with a Telephone Interface to be used as an interactive terminal for communication with a remote time sharing computer system. This program is supplied with the RS-232-C Interface at no additional cost. Many TRS-80 owners would like to be able to use their serial printers with the LPRINT command in Level II BASIC. This is now possible with the RS-232-C Interface installed in the Expansion Interface. 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 - asynchronous or synchronous - is commonly used. The transmission technique used in the Radio Shack implementation is asynchronous-bit-serial. No further reference will be made to synchronous techniques in this document. Asynchronous transmission does not require the use of a synchronizing clock to be transmitted with the data and, the characters need not be contiguous. This implies that indeterminate gaps 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 (Ø) data bit that is added to the front of each character. The stop element is a logic one (1) that is added 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. There is however, 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.

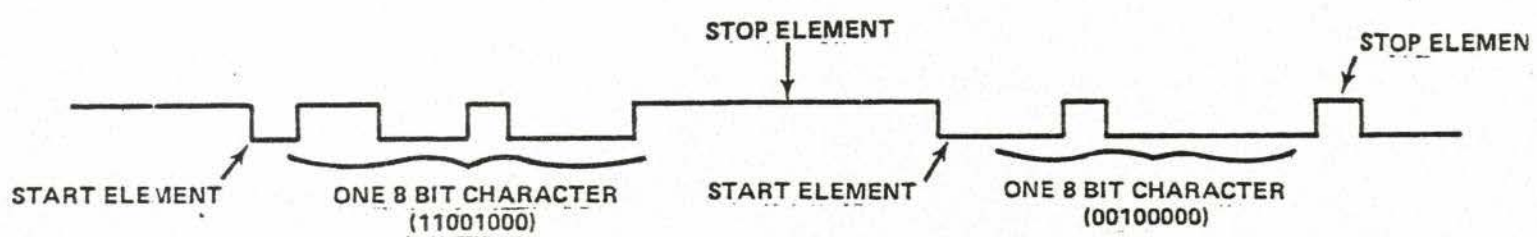


FIGURE 1. ASYNCHRONOUS DATA

There are several good reasons for using asynchronous data transmission. A clock signal does not need to be transmitted with the data, therefore, equipment requirements are greatly simplified. Additionally, the characters need not be contiguous in time and can be transmitted as they become available. This last feature is particularly useful when transmitting data from manual entry devices such as a keyboard. The major disadvantage of asynchronous transmission is that it requires a significant portion of the communications bandwidth for start and stop elements.

The rate at which asynchronous data is transmitted is usually referred to as the baud rate. Baud rate is defined to be the inverse of the time duration of the shortest signal element. In general, 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 similar to the Radio Shack Telephone Interface is required to convert the data pulses to tones that can be transmitted through the telephone network.

Signal Conventions

The RS-232-C specification defines voltage levels and corresponding logic conventions associated with data and control information transmitted between equipment. For data interchange purposes, the signal is considered in the marking condition when the voltage measured at the interface point is more negative than minus three volts with respect to signal ground. The signal is considered in the spacing condition when the voltage is more positive than plus three volts with respect to signal ground. The marking condition corresponds to a logic one and the space condition corresponds to a logic zero. For timing and control interchange circuits, the function is considered to be "on" when the voltage on the interchange circuit is more positive than plus three volts with respect to signal ground, and is considered to be "off" when the voltage is more negative than

minus three volts with respect to signal ground. The "on" condition corresponds to a logic zero and the "off" condition corresponds to a logic one. 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 aspects of the RS-232-C specify a 25 pin connector (generally called a DB-25) which electrically connects the needed signals to each device. 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	Transmitted Data
3	RD	Received 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 Descriptions

Protective Ground: This conductor should be bonded to the chassis or equipment frame. It may also be connected to the 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 communications 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 communications 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 shall appear at all other times and indicates that the data terminal should ignore signals appearing on any other interchange circuit (exception - Ring Indicator).

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 this 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 (Received Line Signal Detector): Direction-from data communication equipment. This signal indicates, when "on", 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.

RS-232-C INSTALLATION AND CHECKOUT

RS-232-C Installation

NOTE: Some of the first TRS-80 Expansion Interface units were manufactured without the 42-contact connector needed to interconnect the Expansion Interface and the RS-232-C Interface. Because of warranty limitations and the need for some technical expertise, we recommend that the owner's of these first units return them to a Radio Shack Service Center for installation of the connector and RS-232-C.

Perform the following steps to install the RS-232-C Interface:

1. Position the TRS-80 Expansion Interface as shown in Figure 2 and remove the expansion door.
2. Remove two machine screws and one washer from the 42-contact connector.
3. Position the RS-232-C Interface as shown. Mount it in the housing so that the 42-contact connector mounting holes align with the screw holes in the RS-232-C Interface printed circuit board.
4. Install the washer and two machine screws in the positions illustrated.
5. Reinstall the expansion door.

Checkout

1. Set the RS-232-C Sense Switches (listed in Table 3) to the desired baud rate, parity, etc. as shown in the following example:

	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

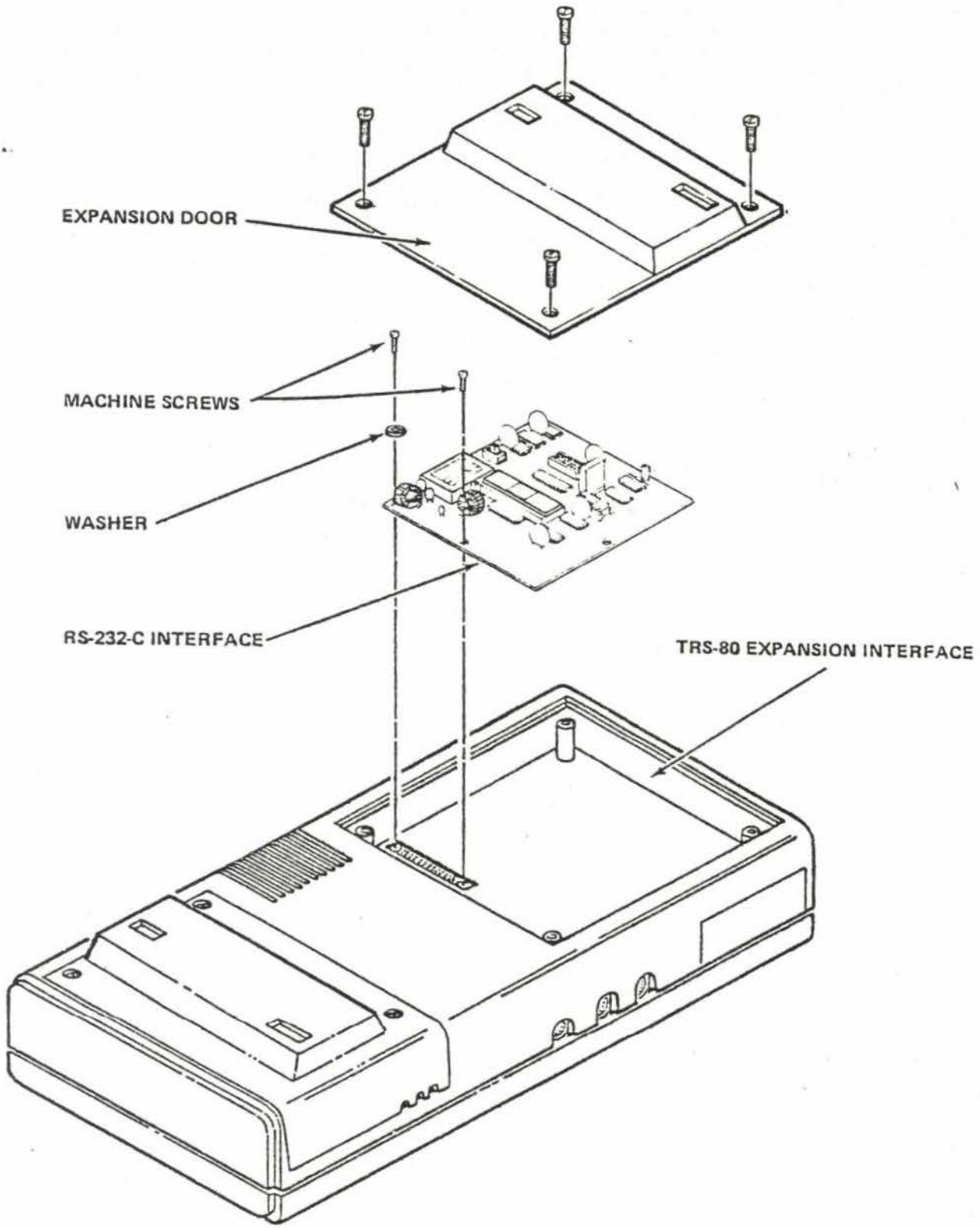


FIGURE 2. RS-232-C INSTALLATION

-- Further, presuming that you are connecting to a modem, the TERM/COMM EQUIP switch should be set to TERM.

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 5ØØØ-Hex).

3. Correct activation of the TERM tape will be indicated by a clearing of the Video Display screen and the positioning of the cursor symbol to the upper left corner of the Video Display screen.

NOTE: If pins 2 and 3 of the 25 pin connector (DB-25) on the RS-232-C cable have been connected together, the keyboard output will be "echoed" back to the Video Display screen indicating correct operation of the Interface and TERM program.

THEORY OF OPERATION

The Radio Shack RS-232-C Interface is a versatile programmable interface which allows the TRS-80 owner to interface with most any equipment which is RS-232-C compatible. There are essentially two different catagories 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 the purpose of interfacing with data-comm equipment.

The user has control of the configuration of the Interface by one of two methods, sense switch selection or direct selection by software control.

Universal Asynchronous Receiver Transmitter

Figure 3 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 may be programmed as follows: The word length can be either 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.

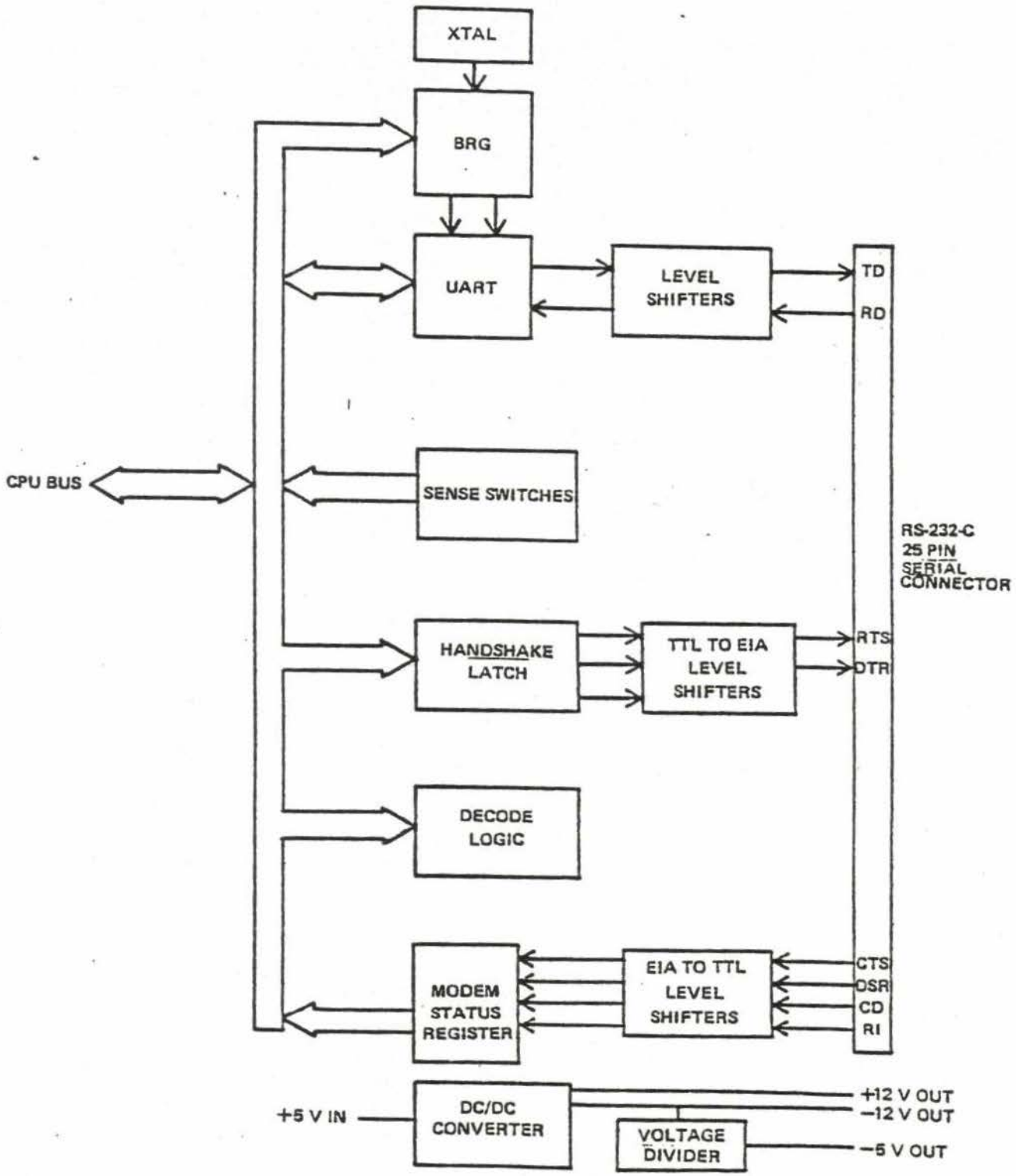


Figure 3. RS-232-C Block Diagram

The UART contains several internal registers which allows the programmer 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. The UART register will be discussed in more detail later.

Baud Rate Generator

Directly above the UART in Figure 3 is a block called the 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. The UART requires a clock at 16 times the baud rate desired, for both transmit and receive operations. 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 3 is a block labeled SENSE SWITCHES. This block is used to communicate to the Interface, the mode of operation desired by the user. Eight single-pole single-throw switches (S1-S8) can 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. The user may elect to ignore these switches and under software control, directly configure the Interface for non-standard baud rates, different receive and transmit frequencies, etc. Table 3, below, summarizes the operation of the sense switches.

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 Bit	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

Handshake Latch

The Radio Shack RS-232-C Interface can, with proper software, control 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, shown in Figure 3, 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 3. 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.

Logic Conventions

The logic internal to the RS-232-C operates with TTL logic levels (3.5V or more = Logic 1, and .8V or less = Logic Ø). The logic convention used for interfacing two RS-232-C devices is EIA levels (-3V or less = Logic one, +3V or more = Logic Ø). The logic levels must therefore be converted from one convention to the other when interfacing two devices. The blocks in Figure 3 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, EBH) for communication with the CPU. I/O mapped devices in a Z-8Ø system use the lower address bits (AØ - A7) in conjunction with the IN* and OUT* signals to address the desired ports. Figure 4 shows the timing associated with this addressing scheme.

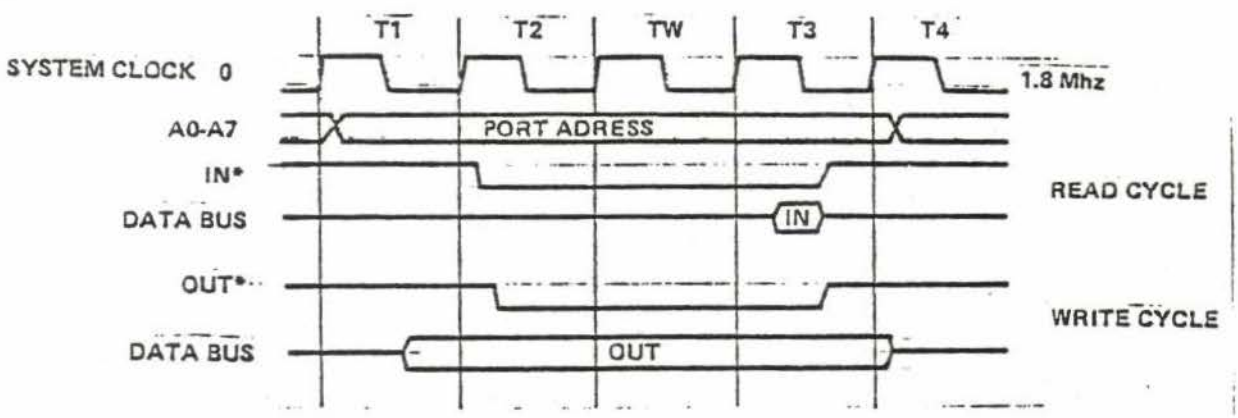


Figure 4. I/O Timing Diagram

Decode Logic

The block labeled DECODE LOGIC in Figure 3 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 perform 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 Z8Ø 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 Z8Ø 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 - DØ) 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. The user may 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. The sense switch block in Figure 3 consists of eight single-pole single-throw switches. Any switch when open, pulls up (+5V) the input of an octal tristate buffer and, when closed, pulls an input low.

NIBBLE LOADED	TRANSMIT OR RECEIVE BAUD RATE	10x CLOCK FREQ.	% ERROR	SWITCH SELECTABLE ?
ØH	5Ø	Ø.8 KHz	Ø	No
1H	75	1.2 KHz	Ø	No
2H	110	1.76 KHz	Ø	Yes
3H	134.5	2.1523 KHz	Ø.Ø16	No
4H	150	2.4 KHz	Ø	Yes
5H	300	4.8 KHz	Ø	Yes
6H	600	9.6 KHz	Ø	Yes
7H	1200	19.2 KHz	Ø	Yes
8H	1800	28.8 KHz	Ø	No
9H	2000	32.Ø81 KHz	Ø.253	No
AH	2400	38.4 KHz	Ø	Yes
BH	3600	57.6 KHz	Ø	No
CH	4800	76.8 KHz	Ø	Yes
DH	7200	115.2 KHz	Ø	No
EH	9600	153.6 KHz	Ø	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 on its inputs to the data bus for loading into the CPU. This operation is performed by executing the following Z8Ø 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 function 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 - DØ) are latched control outputs, one of which disables transmitted 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
D7	Clear to send Pin 5 DB-25	Even Parity Enable 1 = even, Ø = odd	Even Parity Enable 1 = even, Ø = odd
D6	Data Set Ready Pin 6 DB-25	Word length Select 1	Word length Select 1
D5	Carrier Detect Pin 8 DB-25	Word length Select 2	Word length Select 2
D4	Ring Indicator Pin 22 DB-25	Stop Bit Select 1 = 2 bits, Ø = 1 bit	Stop Bit Select 1 = 2 bits, Ø = 1 bit
D3	Unused	Parity Inhibit 1 disables parity	Parity Inhibit 1 disables parity
D2	Unused	Baud Rate 3	Break Ø Disables Transmit data
DØ	Unused	Baud Rate 2	Data Terminal Ready Pin 20 DB-25
D1	Receiver Input UART P20	Baud Rate 1	Request to send Pin 4 DB-25
	IN ØE8H	IN ØE9H	OUT ØEAH

Table 6. BIT ALLOCATIONS FOR REGISTERS AND LATCHES

UART STATUS REGISTER	
D7	DATA RECEIVED 1 = Condition true
D6	TRANSMITTER HOLDING REGISTER EMPTY 1 = Condition true
D5	OVERRUN ERROR 1 = Condition true
D4	FRAMING ERROR 1 = Condition true
D3	PARITY ERROR 1 = Condition true
D2	unused
D1	unused
DØ	unused
	IN ØEAH

Table 6. BIT ALLOCATIONS FOR REGISTERS AND LATCHES (Cont.)

The following example, in Z8Ø 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, RESETS 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 a memory location which is used to save the current state of the UART control register and the handshake latch. Suppose that the programmer wished 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 (received data ready for input, transmitter holding register empty, overrun error, framing error, and parity error). The remaining bits D2-DØ are unused. The programmer 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 instruction 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. In general this should not be attempted until the holding register is found to be empty (meaning that the previous character has been transferred to the transmitter register for transmission). This is done by reading the UART status register and checking the proper bit for a logic one. 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. In general this should not be attempted 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 result in the character received being loaded into register A and the received data bit in the UART control register being reset.

Earlier it was stated that there are two categories of RS-232-C equipment, data terminal equipment and data communications. The two categories of equipment differ as to 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, dependent upon the specific interface requirements. A dual-pole dual-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 EQUIP (communications equipment).

The RS-232-C Interface requires four power supply voltages (+5V, -5V, +12V, -12V). The Expansion Interface provides the +5V supply from which the other voltages are derived with the aid of a DC to DC converter. A DC to DC can be thought of as a black box which inputs one DC voltage and provides one or more different output voltages. 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. Components L1, L2, C1, C1Ø, C11, C3, and C9 perform this decoupling and filtering function.

TERMINAL PROGRAM

Hardware prerequisites: * Level II TRS-80 with 16K RAM

* Expansion Interface with RS-232-C installed.

Program Description:

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.

RS-232-C Input to Display

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

Back space (hex Ø8) and return (hex ØD) are recognized and executed by the display program. Other ASCII control codes are ignored, including the null (hex ØØ) which may be used as fill to allow the display function to scroll after return at high baud rates (12ØØ, 24ØØ, etc.) as indicated in Table 7.

Table 7 - Nulls Required after return

Baud Rate	#Nulls
11Ø	None
15Ø	None
3ØØ	None
6ØØ	None
12ØØ	2
24ØØ	4
48ØØ	8
96ØØ	16

Lower case ASCII input 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 depressions. New key strobes are converted to ASCII and outputted to the RS-232-C Interface for transmission.

A keyboard downstrobe (depression) 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 pins 2 and 3 on the RS-232-C connector are shorted together, the terminal can act as 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 acoustic coupler has a switch that, if set to "half duplex" (or HDX), will allow communication over a telephone line and provide (within the 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 depressed	Code at Address	Default Code
shift ↓, A	5Ø B9H	ØØ3H = EOT
shift ↓, B	5Ø BAH	ØØ1BH = ESC
shift ↓, C	5Ø BBH	ØØFCH = 1
shift ↓, D	5Ø BCH	ØØFFH = 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) strike A, B, C, or D. The keys may be released in any order.

Other, standard, control codes may be generated by the same process with E through O generating Ø5H through ØFH and P through Z generating 1ØH through 1AH (You can test this by generating Ø8H shift ↑,H which is back space).

If the default special codes are not appropriate, TBUG LEVEL II may be loaded via the SYSTEM command in BASIC (use the reset push button) and the desired codes entered 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.

Some Specific Disclaimers

1. The price of the RS-232-C and the organizational structure of Radio Shack do not allow special help such as applications and support beyond the information herin provided.
2. Although an attempt has been made to provide the common RS-232-C modem control signals, use of the RS-232-C and TERMinal program with modems other than the Radio Shack Telephone Interface is not guaranteed and is not supported.
3. The TERMinal program and RS-232-C do not provide a means of loading BASIC LEVEL II data or program material via RAM, cassette, disk, etc.

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 readily occur if the wrong settings for even/odd parity, parity enable, baud rate, terminal/data set,(input/output correspondence with pins 2 and 3, etc.) are made.
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. It is possible for a skillful customer to use TBUG to make use of this ignored information in a particular application.

```
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.

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 116150   >   SOUT    LD DE,SUCB
00053 5049 C5               PUSH BC
00054 504A 0620             LD B,20H        ;OUTPUT BYTE.
00055 504C C34600           JP CIO$

00057 504F 116150   >   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 116150   >   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. REQST.
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 UCB.
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 UCB.
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"

00156                                END
```

```
SCALARS

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

%TERM0 (DEFAULT) SECTION (50BD)

   BAUDTB - 50B1              IBRG --- 509E  INS ---- 5012          IUART -- 508E
   NOSPEC - 503D              OUTS --- 502A  RS232 -- 5089          RSRD --- 5075
   RSX ---- 5073              SIN ---- 504F  SOUT --- 5046          SPECTB - 50B9
   TERM --- 5000


  156 SOURCE LINES     156 ASSEMBLED LINES    12719 BYTES AVAILABLE

          >>> NO ASSEMBLY ERRORS DETECTED <<<
```

EXAMPLE APPLICATIONS PROGRAM

A common application which will interest many Radio Shack RS-232-C owners, 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 (ie. 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 above along with other I/O specific information is 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
4Ø25H	DCB TYPE (Input, or output operation)
4Ø26H	DRIVER ADDRESS (Low order Byte)
4Ø27H	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.

Let us assume for this example 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. This status line, when low means that the printer is not busy and can accept data from the Interface and when high, 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 to 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 36 and 37 illustrates one of many possible approaches to this problem. 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 succesive calls to the driver routine.

Assembly Language Printer 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 3A4E7F     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 324E7F     00270         LD (FLAG),A     ; SET FLAG TO INDICATE 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 324D7F     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 214F7F     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.  IT FIRST
                00451 ; CHECKS TO SEE IF DATA SET  READY IS HIGH, LOOPS IF NOT, OTHERWISE GOES ON TO CHECK THE
                00452 ; UART TO SEE IF IT'S HOLDING REGISTER IS EMPTY, LOOPS UNTIL IT IS AND THEN OUTPUTS THE CHAR
                00453 ; TO BE TRANSMITTED TO THE HOLDING REGISTER.  IF THE CHAR WAS A CARRIAGE RETURN A LINE FEED
                00454 ; IS ALSO OUTPUT WHEN DSR IS HIGH AND THE UART IS EMPTY.
7F2D DBE8       00460 CHKDSR  IN A,(RESURT)   ; READ INTERFACE CONTROL BITS
7F2F CB77       00470         BIT 6,A         ; TEST DSR FOR A HIGH
7F31 28FA       00480         JR Z,CHKDSR     ; TEST AGAIN IF NOT (CAN'T OUTPUT DATA)
7F33 DBEA       00490 STATIN  IN A,(CNTREG)   ; LOAD UART STATUS
7F35 CB77       00500         BIT 6,A         ; TEST THRE FOR HIGH
7F37 28F4       00510         JR Z,CHKDSR     ; LOOP IF NOT
7F39 79         00520         LD A,C          ; LOAD A W/CHAR TO BE OUTPUT
7F3A D3EB       00530         OUT (DTAREG),A  ; LOAD HOLDING REG W/CHAR
7F3C FE0D       00540         CP 0DH          ; IS IT A CARRIAGE RET ?
```

```
7F3E 2004       00550           JR NZ,RETRN     ; RETURN IF NOT
7F40 0E0A       00560           LD C,0AH        ; IF SO OUTPUT A LINE FEED ALSO
7F42 18E9       00570           JR CHKDSR       ; OUTPUT TO UART WHEN DSR IS HIGH
7F44 C9         00580 RETRN     RET             ; RETURN TO CALLING CODE
                00590 ; THE FOLLOWING TABLE DEFINES THE BAUD RATE SELECTED BY THE SENSE SWITCHES IN THE INTERFACE
7F45 22         00600 BDTABL    DEFB 22H        ; 110 BAUD
7F46 44         00610           DEFB 44H        ; 150 BAUD
7F47 55         00620           DEFB 55H        ; 300 BAUD
7F48 66         00630           DEFB 66H        ; 600 BAUD
7F49 77         00640           DEFB 77H        ; 1200 BAUD
7F4A AA         00650           DEFB 0AAH       ; 2400 BAUD
7F4B CC         00660           DEFB 0CCH       ; 4800 BAUD
7F4C EE         00670           DEFB 0EEH       ; 9600 BAUD
7F4D 00         00680 SWTING    DEFB 00H        ; IMAGE OF HANDSHAKE LATCH
7F4E 00         00690 FLAG      DEFB 00H        ; FLAG TO INDICATE INITIALIZATION
0000            00700           END

00000 TOTAL ERRORS
RETRN   7F44
STATIN  7F33
CHKDSR  7F2D
BDTABL  7F45
BAUDST  7F16
SWTING  7F4D
RESTOR  7F2A
FLAG    7F4E
INIT    7F00
DTAREG  00E8
CNTREG  00EA
SWITCH  00E9
RESURT  00E8
```

BASIC Program for Line Printer

```
10 REM ** POKE NEW DCB TYPE AND ADDRESS IN RAM (4025H) **
20 POKE 16421,2 : POKE 16422,0 : POKE 16423,127
30 REM ** POKE RS232C I/O DRIVER INTO HIGH MEM (7F00H) **
40 FOR X=32512 TO 32590
50 READ Y
60 POKE X,Y
70 NEXT X
75 END
80 DATA 229,197,245,58,78,127,254,1,40,32
90 DATA 62,1,50,78,127,211,232,219,233
100 DATA 230,248,246,4,50,77,127,211,234,219
110 DATA 233,230,7,33,69,127,6,0,79,9,126,211
120 DATA 233,241,193,225,219,232,203,119,40,250
130 DATA 219,234,203,119,40,244,121,211,235,254
140 DATA 12,22,4,14,10,24,233,201,34,68,85,102
150 DATA 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 loops testing 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 36 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 36 and 37.

Assuming the specific interface requirements are identical to this example, the user simply types in the BASIC program listed on page 38, runs it and is now ready to use his 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 video monitor.

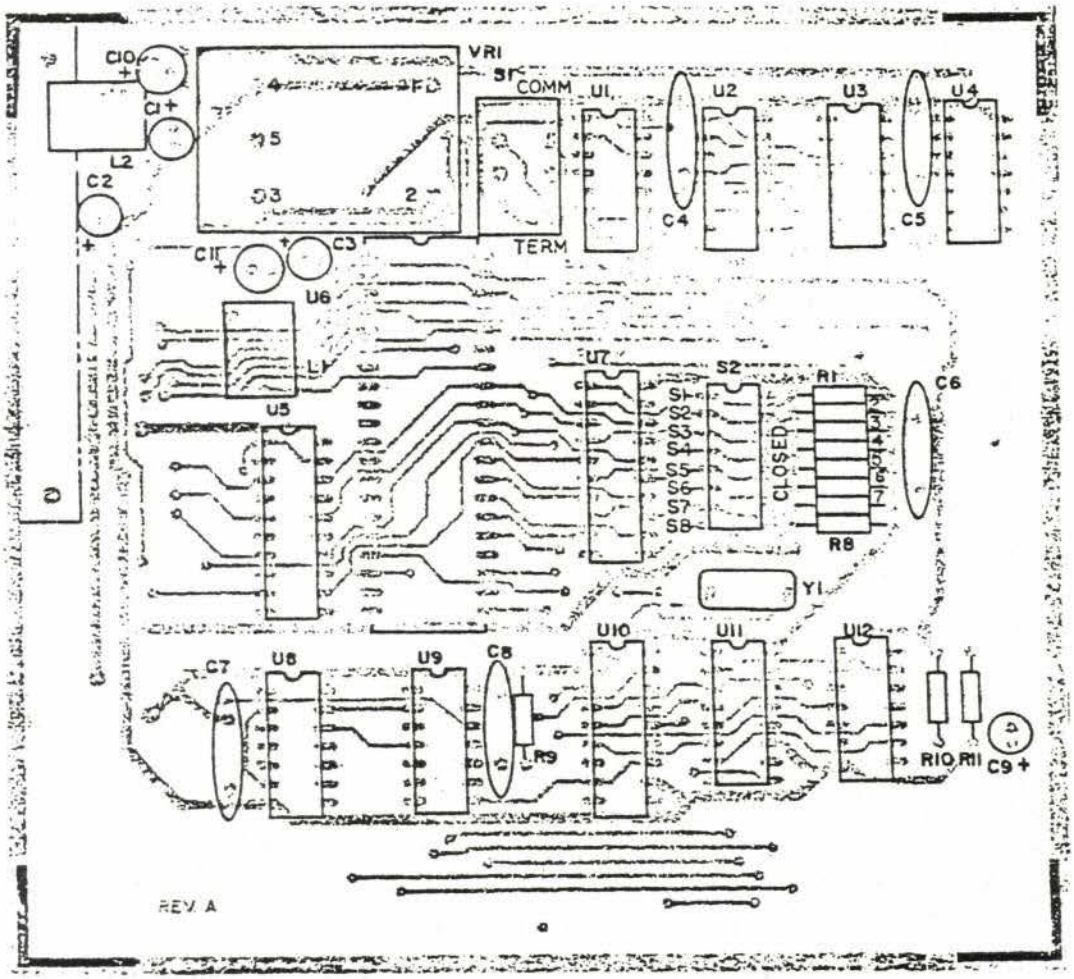
   The operator should respond with:

   32511 AND PRESS ENTER ; Decimal equivalent of 7EFFH.

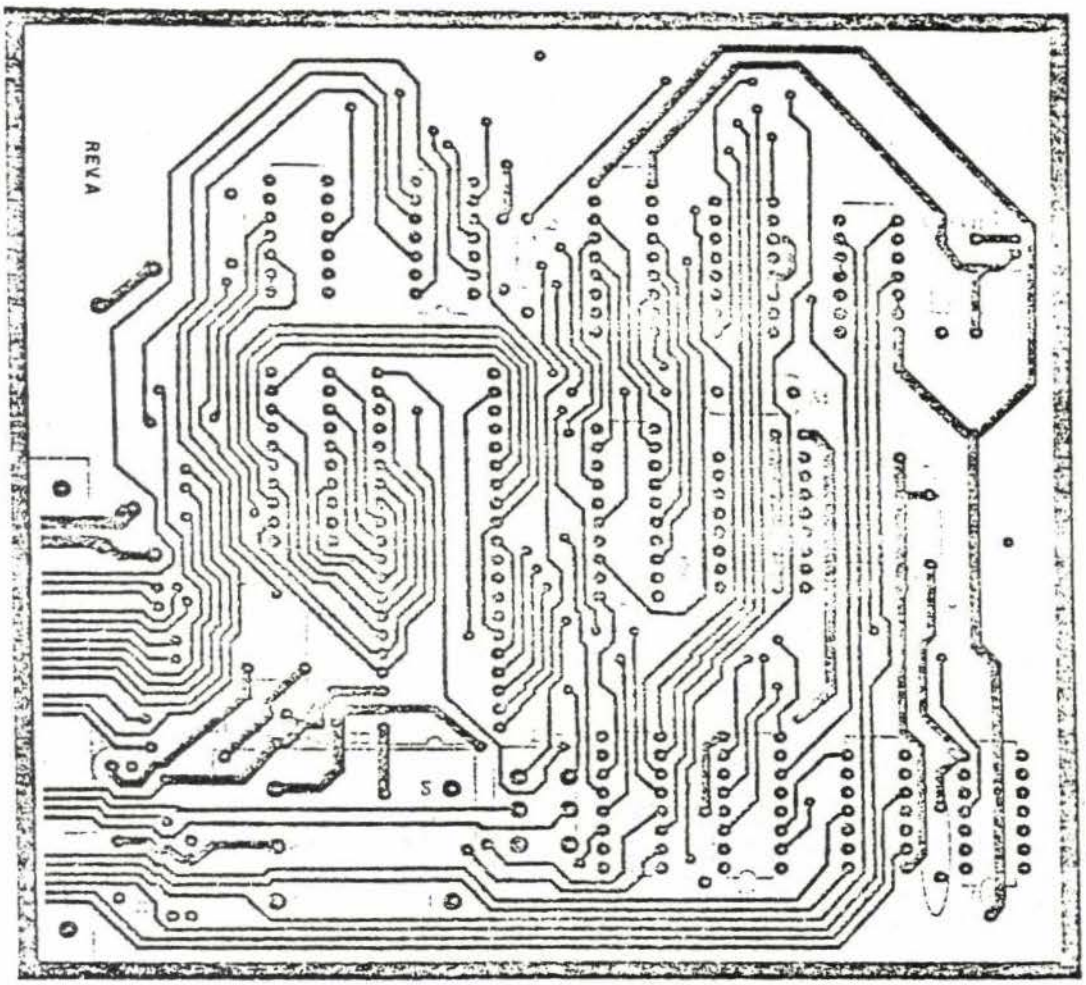
This response reserves the top 256 bytes of a 16K TRS-80 for the printer driver.

2. Set configuration switches for desired baud rate, stop bits, word length and parity convention required. See example on page 8.
3. Type in the BASIC program listed on Page 38 with appropriate modifications for specific application or load from cassette tape a previously saved version of above.
4. Run the BASIC program.
5. Connect the Line Printer to the RS-232-C with the provided or modified cable.
6. We can now utilize the serial line printer with the Level II BASIC commands LPRINT and LLIST.

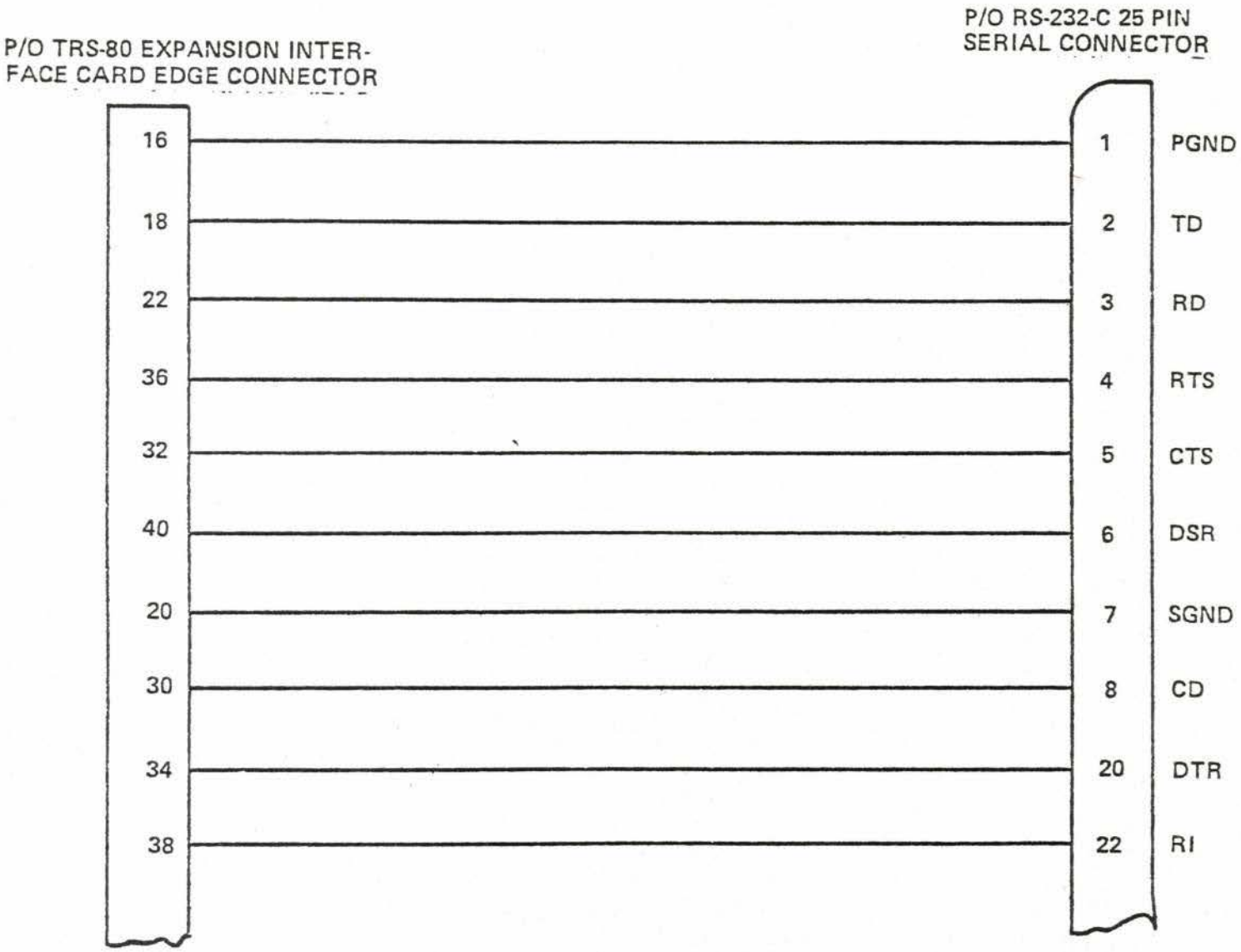
Any BASIC programs which utilize this technique must include the BASIC program on Page 38 as a part of the coding and this section of code must be executed prior to using the serial line printer. If it is undesirable to the user to locate the object code for the line printer driver at 7FØØH (as in this example), the user may reassemble the source code on Page 36 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 38 must be altered to reflect any change in the object code due to relocation or modification. The last two poke statements on line 20 Page 38 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).



Printed Circuit Board (Component Side)



Printed Circuit Board (Circuit Side)



40-Conductor to 25-Conductor
Cable Diagram

PARTS LIST

SYMBOL	DESCRIPTION	PART NUMBER
	Cable	
	RS-232-C, 40 Conductor-to-25 Conductor	
	Capacitors	
C1	33 uF, 16V, Tantalum	1500086
C2	3.3 uF, 10V, Tantalum	1500085
C3	10 uF, 16V, Electrolytic	1500012
C4	0.1 uF, 50V, Ceramic Disc	1500053
C5	0.1 uF, 50V, Ceramic Disc	1500053
C6	0.1 uF, 50V, Ceramic Disc	1500053
C7	0.1 uF, 50V, Ceramic Disc	1500053
C8	0.1 uF, 50V, Ceramic Disc	1500053
C9	10 uF, 16V, Electrolytic	1500012
C10	33 uF, 16V, Tantalum	1500086
C11	3.3 uF, 10V, Tantalum	1500085
	Chokes	
L1	500 uH, 0.1 Ohm	
L2	500 uH, 0.1 Ohm	
	Resistors	
R1	10K, ¼W, 5%, Fixed	4704068
R2	10K, ¼W, 5%, Fixed	4704068
R3	10K, ¼W, 5%, Fixed	4704068

SYMBOL	DESCRIPTION	PART NUMBER
	Resistors (cont)	
R4	10K, ¼W, 5%, Fixed	4704068
R5	10K, ¼W, 5%, Fixed	4704068
R6	10K, ¼W, 5%, Fixed	4704068
R7	10K, ¼W, 5%, Fixed	4704068
R8	10K, ¼W, 5%, Fixed	4704068
R9	4.7K, ¼W, 5%, Fixed	4704061
R10	6.2K, ¼W, 5%, Fixed	4704064
R11	4.7K, ¼W, 5%, Fixed	4704061
	Switches	
S1	Slide, DPDT	5100016
S2	SPST - 8 Position	5106002
	Integrated Circuits	
U1	74LS155, Dual 2-Line to 4-Line Decoder	3102029
U2	74LS367, Tri-state HEX Buffer	3102024
U3	1489, EIA to TTL Converter	3108023
U4	1489, EIA to TTL Converter	3108023
U5	74LS244, Octal Tri-state Buffer	3106008
U6	TR1602, Universal Asynchronous Receiver/Transmitter	3110003
U7	74LS244, Octal Tri-state Buffer	3106008
U8	74LS00, Quad 2-Input NAND Gate	3102006
U9	74LS04, HEX Inverter	3102008
U10	BR2941L, Dual Baud Rate Generator	3102038
U11	74LS174 HEX D Flip-Flop	3102015

SYMBOL	DESCRIPTION	PART NUMBER
	Integrated Circuits (cont)	
U12	1488 TTL to EIA Converter	3106007
	Converter	
VR1	DC-to-DC Converter	4000010
	Crystal	
Y1	5.0688 MHz	

					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	M S B b6 b4
					0	1	2	3	4	5	6	7	
0	0	0	0	0	NUL	DLE	SP	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	10	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	11	VT	ESC	+	;	K	[	k	{	
1	1	0	0	12	FF	FS	,	<	L	\	l	¦	
1	1	0	1	13	CR	GS	–	=	M	]	m	}	
1	1	1	0	14	SO	RS	.	>	N	^	n	~	
1	1	1	1	15	SI	US	/	?	O	_	o	DEL	

L S B
b3 b0

ACK – Acknowledge
BEL – Bell
BS – Back space
CAN – Cancel
CR – Carriage return
DC1 - DC4 – Device controls
DLE – Data link escape
EM – End of media
ENQ – Enquiry
EOT – End of transmission
ESC – Escape
ETB – End of transmission block
ETX – End of text
FF – Form feed
FS – File separator
GS – Group separator
HT – Horizontal tab
LF – Line feed
NAK – Negative acknowledge
NUL – Null
RS – Record separator
SI – Shift in
SO – Shift out
SOH – Start of heading
STX – Start of text
SUB – Substitute
SYN – Synchronize
US – Unit separator
VT – Vertical tab

ASCII CODE

TECHNICAL REFERENCES

1. EIA STANDARD RS-232-C

2. INDUSTRIAL ELECTRONICS BULLETIN NO.12, APPLICATION NOTES ON INTERCONNECTION BETWEEN INTERFACE CIRCUITS USING RS-499 and RS-232-C

3. WESTERN DIGITAL TR1602A/B DATA SHEET

4. WESTERN DIGITAL BR2941L DATA SHEET

5. WESTERN DIGITAL TR1602A/B APPLICATION REPORT #1, ASYNCHRONOUS RECEIVER/TRANSMITTER