T
N
W
Corporation
5924 QUIET SLOPE DRIVE
SAN DIEGO, CA 92120

This users' manual for the TNW-2000 Serial Interface presents the information you need to use the TNW-2000, accompanied by background material describing the RS-232 and IEEE-488 data transmission standards.

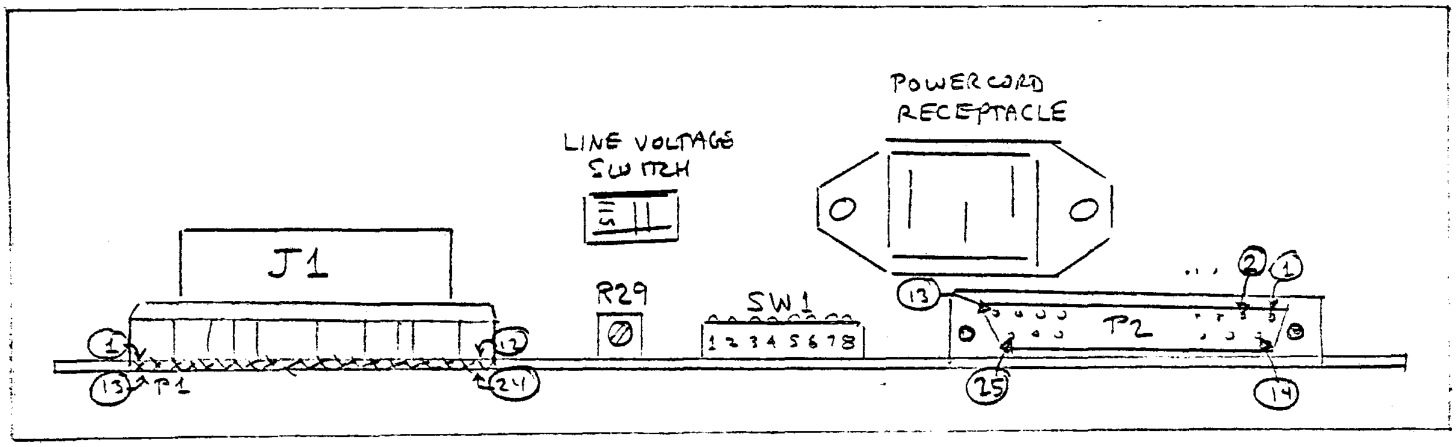
The sections of the manual cover:

We welcome your comments on, suggestions for, and criticisms of our Serial Interface Module and this documentation.

Doug Gage (714) 223-1176
Gary Huckell (714) 583-2652

READ THIS FIRST!

CAUTION: BEFORE PLUGGING POWERCORD INTO OUTLET BE SURE 115/230 VOLT POWER SWITCH IS IN CORRECT POSITION FOR VOLTAGE IN YOUR AREA. USE SMALL SCREWDRIVER TO OPERATE LINE VOLTAGE SWITCH.



REAR VIEW OF TNW-2000

NOTE: THE TNW-2000 HAS NO POWER ON/OFF SWITCH, POWER-ON INDICATOR LIGHT, OR FUSE. TOTAL POWER CONSUMPTION IS LESS THAN 5 WATTS, AND THE UNIT CAN (AND SHOULD) BE LEFT ON AT ALL TIMES.

REAR PANEL COMPONENTS

P1: IEEE BOARD-EDGE CONNECTOR IDENTICAL TO IEEE CONNECTOR ON REAR OF COMMODORE PET COMPUTER.

J1: IEEE STANDARD CONNECTOR. THIS CONNECTOR IS WIRED PIN-TO-PIN IN PARALLEL WITH P1 TO ENABLE CHAINING OF PERIPHERAL EQUIPMENT ON THE IEEE BUS.

R29: RS-232 BAUD RATE ADJUSTMENT. THIS IS A MULTITURN POT. TURNING CLOCKWISE INCREASES THE BAUD RATE (USE A SMALL SCREWDRIVER TO MAKE THE ADJUSTMENT).

SW1: 8 POSITION DIP SWITCH USED TO SELECT IEEE ADDRESS, PET OR STANDARD ASCII, PARITY, AND BAUD RATE RANGE.

P2: STANDARD RS-232 FEMALE CONNECTOR. MANY RS-232 PERIPHERALS CAN BE PLUGGED IN DIRECTLY.

LINE VOLTAGE SWITCH: SELECTS 115 OR 230 VOLT LINE POWER. CAUTION: APPLYING 230 VOLTS TO THE TNW-2000 WITH THE SWITCH SET TO 115 VOLTS WILL CAUSE DAMAGE TO THE UNIT.

POWERCORD RECEPTACLE: PLUG FOR STANDARD POWERCORD, WHICH IS SUPPLIED WITH THE TNW-2000.

CONNECTION OF THE TNW-2000 TO PET OR OTHER COMPUTER

THE 24 CONDUCTOR FLAT CABLE SUPPLIED WITH THE TNW-2000 CAN BE USED IN EITHER OF TWO WAYS. NOTE THAT ONE END OF THE CABLE HAS A STANDARD IEEE CONNECTOR WHILE THE OTHER END HAS A BOARD-EDGE CONNECTOR COMPATIBLE WITH THE PET. IF THE IEEE END IS PLUGGED INTO J1 OF THE TNW-2000, THE OTHER END CAN BE PLUGGED INTO A PET (IT IS KEYED TO PREVENT INCORRECT INSERTION). CONNECTED IN THIS MANNER, A SECOND IEEE PERIPHERAL (USING A DIFFERENT IEEE BUS ADDRESS) CAN BE CONNECTED TO THE PET BY PLUGGING IT INTO P1 OF THE TNW-2000. THIS METHOD OF CONNECTING MULTIPLE DEVICES IS CALLED CHAINING.

IF THE COMPUTER (OR PERIPHERAL, IF CHAINING) USES A STANDARD IEEE CONNNECTOR, RATHER THAN THE PET STYLE BOARD-EDGE CONNECTOR, CONNECT THE BOARD EDGE CONNECTOR OF THE CABLE TO P1 OF THE TNW-2000. THE STANDARD IEEE CONNECTOR ON THE CABLE CAN BE CONNECTED TO THE COMPUTER OR PERIPHERAL CHAIN. CONNECTOR J1 OF THE TNW-2000 CAN THEN BE USED TO CONTINUE THE CHAIN.

CONNECTION OF TNW-2000 TO PERIPHERAL DEVICES

THE TNW-2000 IS CONFIGURED AS A DATA COMMUNICATIONS EQUIPMENT (DCE), AND THE UNIT PROVIDES A FEMALE RS-232 (EIA) CONNECTOR FOR YOUR DEVICE TO CONNECT TO. THIS IS THE NORMAL REQUIREMENT FOR MOST DEVICES, APART FROM MODEMS AND OTHER COMMUNICATIONS EQUIPMENT. THE FOLLOWING TABLE LISTS THE SIGNALS THE TNW-2000 CONNECTS TO.

P2 PIN	SIGNAL NAME	FUNCTION
1	PROTECTIVE GROUND	TIED TO CHASSIS AND POWER LINE GROUND WIRE IN TNW-2000. FLOATING FROM SIGNAL GROUND, PIN 7.
2	RECEIVED DATA	INPUT: SERIAL DATA BEING RECEIVED BY TNW-2000.
3	TRANSMITTED DATA	OUTPUT: SERIAL DATA BEING TRANSMITTED BY TNW-2000.
4	REQUEST TO SEND	INPUT: MUST BE HIGH (OR FLOATING) OR TNW-2000 WILL NOT TRANSMIT DATA.
5	CLEAR TO SEND	OUTPUT: GOES HIGH WHENEVER POWER IS ON.
6	DATA SET READY	OUTPUT: GOES HIGH WHENEVER POWER IS ON.
7	SIGNAL GROUND	GROUND REFERENCE FOR ALL OTHER INTERFACE SIGNALS.
8	RECVD SIGNAL DETECTOR	OUTPUT: GOES HIGH WHENEVER POWER IS ON.
20	DATA TERMINAL READY	INPUT: MUST BE HIGH (OR FLOATING) OR TNW-2000 WILL NOT TRANSMIT DATA.

RS-232 INTERFACE SIGNALS

IN ORDER TO INTERFACE THE TNW-2000 TO A MODEM (A DCE), A SPECIAL CABLE CAN BE USED TO MAKE THE TNW-2000 APPEAR TO BE A DTE. THIS CABLE HAS A MALE EIA CONNECTOR ON EACH END; PINS 1 AND 7 ARE WIRED STRAIGHT THROUGH, WHILE PINS 2/3, 4/5, AND 6/20 ARE CROSSED. THIS CABLE IS AVAILABLE FROM TNW CORPORATION AS OPTION TNW-2000MM.

IF YOU ARE USING THE TNW-2000 TO INTERFACE TO A PRINTER AND THE PRINTER HAS A MATCHING CONNECTOR YOU SHOULD HAVE NO

PROBLEM. HOWEVER, RS-232 IS NOT ALWAYS IMPLEMENTED THE SAME BY DIFFERENT MANUFACTURERS. POSSIBLE PROBLEMS ARE ENDING UP WITH TWO FEMALE CONNECTORS THAT MUST BE JOINED OR HAVING BOTH THE TNW-2000 AND THE PERIPHERAL DEVICE TRANSMITTING ON PIN 3 AND RECEIVING ON PIN 2. BOTH PROBLEMS CAN BE CORRECTED USING A SPECIAL CABLE.

A MORE DIFFICULT PROBLEM OCCURS WHEN THE PERIPHERAL DEVICE SIGNALS "BUSY" BY SETTING ITS "BUSY CONTROL" HIGH RATHER THAN LOW. THE BUSY SIGNAL MAY ALSO BE PRESENTED ON A PIN OTHER THAN PIN 4 OR PIN 20, WHICH THE TNW-2000 WILL RESPOND TO. READ THE DOCUMENTATION FOR YOUR DEVICE TO LEARN HOW THE MANUFACTURER HAS IMPLEMENTED THE RS-232 INTERFACE.

IF YOU HAVE A PROBLEM USING YOUR INTERFACE OR IF YOU HAVE SOLVED A PROBLEM THAT YOU FEEL OTHERS MAY ENCOUNTER, PLEASE WRITE TO US AT THE ADDRESS AT THE FRONT OF THIS MANUAL.

TNW-2000 STRAPPING (IEEE ADDRESS, PARITY, PET ASCII, BAUD RATE)

STRAPPING IS ACCOMPLISHED BY SETTING THE INDIVIDUAL SWITCHES OF SW1 LOCATED ON THE REAR OF THE TNW-2000. TO OPERATE A SWITCH, PRESS DIRECTLY ON THE INDENTATION WITH THE POINT OF A PENCIL OR OTHER INSTRUMENT.

IEEE ADDRESS STRAPPING (SWITCHES 5, 6, 7, 8)

THE LOWER 4 BITS OF THE TNW-2000'S IEEE BUS ADDRESS ARE SET BY SWITCH POSITIONS 5, 6, 7, AND 8. POSSIBLE ADDRESSES ARE THEREFORE 0 THROUGH 15; HOWEVER, ADDRESSES 0 - 3 ARE ALREADY USED INTERNALLY BY THE PET. IEEE ADDRESSES 16 - 31 (INSTEAD OF 0 - 15) CAN BE SELECTED BY MOVING A STRAP ON THE TNW-2000 PRINTED CIRCUIT BOARD INSIDE THE CABINET. THE IEEE ADDRESS IS SET TO 4 AT THE FACTORY.

(OP = SWITCH OPEN, CL = SWITCH CLOSED)

ADDRESS	5	6	7	8
0*	CL	CL	CL	CL
1*	CL	CL	CL	OP
2*	CL	CL	OP	CL
3*	CL	CL	OP	OP
4	CL	OP	CL	CL
5	CL	OP	CL	OP
6	CL	OP	OP	CL
7	CL	OP	OP	OP
8	OP	CL	CL	CL
9	OP	CL	CL	OP
10	OP	CL	OP	CL
11	OP	CL	OP	OP
12	OP	OP	CL	CL
13	OP	OP	CL	OP
14	OP	OP	OP	CL
15	OP	OP	OP	OP

* INDICATES ADDRESSES THAT MUST NOT BE USED WITH THE PET.

PARITY STRAPPING (SWITCHES 2,3)

TWO STRAPS, PARITY INHIBIT (SWITCH 3) AND EVEN PARITY (SWITCH 2) CONTROL PARITY GENERATION AND CHECKING. THE LENGTH OF THE DATA CHARACTER IS AUTOMATICALLY ADJUSTED, WHEN PARITY IS SELECTED, TO BE EITHER 7 BITS WHEN PARITY IS PRESENT OR 8 BITS WITH NO PARITY. THE NUMBER OF STOP BITS GENERATED IS ALWAYS 2. THE POSITION OF THE EVEN PARITY SWITCH HAS NO EFFECT WHEN PARITY IS INHIBITED. PARITY IS SET ENABLED AND EVEN AT THE FACTORY.

PARITY	2	3
INHIBITED	-	OP
EVEN	OP	CL
ODD	CL	CL

PET/ASCII CONVERSION STRAPPING (SWITCH 4)

NOTE: THIS DISCUSSION APPLIES TO THE STANDARD 4K/8K PET. THE 16K/32K PETS USE A DIFFERENT CHARACTER SET WITH UPPER AND LOWER CASE ALPHABETIC CHARACTERS INTERCHANGED WHEN DISPLAYED ON THE SCREEN (THIS WAS DONE SO THAT SHIFTING WOULD YIELD UPPERCASE, INSTEAD OF LOWERCASE). SOFTWARE CHARACTER CONVERSION IS REQUIRED FOR USE OF THE TNW-2000WITH 16K/32K PETS.

LOWER CASE ALPHABETIC CHARACTERS IN THE PET COMPUTER DO NOT CONFORM TO THE ASCII STANDARD. THIS SWITCH OPTION WHEN ENABLED WILL CAUSE THE TNW-2000 TO AUTOMATICALLY CONVERT TRANSMITTED CHARACTERS TO STANDARD ASCII AND RECEIVED CHARACTERS TO NON-STANDARD PET HALF-ASCII. THIS IS THE STANDARD STRAPPING TO USE WHEN THE TNW-2000 IS USED BETWEEN A PET AND PRINTER. THIS MODE SHOULD BE DISABLED WHEN USING THE INTERFACE WITH A PROCESSOR OTHER THAN THE PET, OR INTERFACING TO A PERIPHERAL THAT REQUIRES A CHARACTER LENGTH OF 8 DATA BITS. PET/ASCII CONVERSION IS SET ENABLED AT THE FACTORY (SWITCH 4 OPEN).

BAUD RATE SELECTION (SWITCH 1)

THE BAUD RATE IS SET BY FIRST SETTING SWITCH 1 TO EITHER HIGH OR LOW RANGE AND THEN ADJUSTING POT R29 TO THE DESIRED BAUD RATE. THIS CAN BE DONE EITHER BY USING AN OSCILLOSCOPE OR WITH THE PROGRAM LISTED BELOW. THE BAUD RATE IS SET TO 300 BAUD AT THE FACTORY. LOW RANGE (SWITCH 1 CLOSED SHOULD BE USED FOR 110 TO 1200 BAUD, HIGH RANGE (SWITCH 1 OPEN, FOR 1200 TO 9600 BAUD (USE EITHER FOR 1200).

The program below can be used to set the baud rate. Disconnect any device from the RS-232 connector. Connect the IEEE interface to the PET. Set the baud rate switch to the desired range. Adjust R29 to vary the baud rate. Baud rate increases as the pot is turned clockwise. The program outputs one or more blocks of characters, measures the time required to complete the transfer and then calculates the baud rate. As the baud rate is raised, the amount of data sent is increased to insure a measurement accuracy of better than 1%. After each new calculation (approx 10 seconds) the "PASS" number is incremented. One problem with the program is that the PET can not be stopped in the middle of the output of a character string. If the TNW-2000 baud rate is adjusted to a lower value while the program is running it can take a long time to complete the current pass. It may be better to stop the program and re-run it. To stop the program hold the stop key down until the PET stops or temporarily disconnect the IEEE cable. Remember that the baud rate only has to be set within a few percent (but always try for 1%). It may not be possible to set it so that the program reads the exact value desired. As an example, trying to display 2400 baud exactly requires setting the rate to within .05%. The program can be modified to set a particular baud rate, but don't try changing the loop by sending single characters in place of the long string used. The time used by the processor between characters will cause a large error in calculated rate. The program assumes that the TNW 2000's IEEE bus address has been set to 4.

```
10 OPEN 1,4,1: J=1: K=1
20 FOR I=1 TO 255: A$=A$+"A": NEXT
30 T=TI: FOR I=1 TO J: PRINT#1, A$;: NEXT: T=TI-T
40 PRINT "[CLR]BAUD RATE IS"; INT(168300*J/T),"PASS"; K
50 K=K+1: J=J-SGN(T-200): IF J=0 THEN J=1
60 GOTO 30
```

Remember to type out "PRINT#"; do not use "?" as an abbreviation for PRINT. The character [CLR] is the character you get by shifting the [HOME] key.

OPERATION OF TNW-2000 WITH PET

Start by reading the Commodore booklet "PET COMMUNICATION WITH THE OUTSIDE WORLD", or the corresponding section of the PET (or CBM) manual. Connect the TNW-2000 IEEE cable to the PET, set the option switches as detailed above, adjust the baud rate, and connect the peripheral device to the TNW-2000 RS-232 connector. All program examples shown assume the TNW-2000 has been left set to IEEE address 4.

TO TRANSMIT (I.E., PRINT).

TO PRINT, YOUR PROGRAM MUST FIRST OPEN AN OUTPUT FILE. UNLESS YOU NEED MORE THEN TEN FILES OPEN SIMULTANEOUSLY, IT IS NOT NECESSARY TO CLOSE THE TNW-2000 FILES. PRINTING THROUGH THE TNW-2000 IS VERY SIMILAR TO PRINTING TO THE SCREEN. THE PET WILL TRANSMIT A CARRIAGE RETURN - LINE FEED AT THE END OF EACH LINE. HOWEVER, NOT ALL SCREEN FUNCTIONS WORK DIRECTLY. FOR EXAMPLE, USING COMMA SEPARATORS IN A PRINT STATEMENT WILL NOT RESULT IN A NEATLY SPACED PRINTED OUTPUT. PRINT FORMATTING HAS TO BE DONE IN SOFTWARE. CONTROL CHARACTERS SUCH AS DEL, RVS, AND CURSOR MOVEMENT DO NOT OPERATE. IN FACT, THESE NON-ASCII CHARACTERS CAN HAVE PECULIAR EFFECTS ON SOME PRINTERS THAT CAN BE CONTROLLED OVER THE RS-232 CHANNEL THROUGH THE USE OF ASCII CONTROL CHARACTERS.

EXAMPLE:

```
10 OPEN 1,4,1
20 PRINT#1, CHR$(12);"THIS IS A TEST"
30 A=10: B=20
40 PRINT#1, A;"     ";B
50 STOP
```

THIS PROGRAM TRANSMITS AN ASCII "CONTROL L" OR "TOP OF FORM" CHARACTER, FOLLOWED BY:

```
THIS IS A TEST
10     20
```

THE CHARACTER CHR$(12) IS ONE OF THE MANY ASCII CHARACTERS THAT ARE NOT REPRESENTED IN THE STANDARD PET CHARACTER SET. THEY CAN BE SENT BY USING THE CHR$ REPRESENTATION OF THEIR DECIMAL VALUE. DON'T FORGET THE SEMICOLON IN THE PRINT# COMMAND, OR AN EXTRA END-OF-LINE WILL BE PRINTED.

TO LIST A PROGRAM

TO LIST A PET PROGRAM ON A PRINTER OR OTHER PERIPHERAL DEVICE, ENTER THE FOLLOWING SEQUENCE OF COMMANDS:

```
OPEN 1,4,1:  CMD 1:  LIST
```

THE PROGRAM WILL THEN BE LISTED. SPECIAL SCREEN CONTROL CHARACTERS AND GRAPHIC CHARACTERS WILL NOT PRINT CORRECTLY. GRAPHIC CHARACTERS WILL PRINT AS THEIR LOWERCASE EQUIVALENTS. THE ABOVE COMMAND LEAVES THE PET IN THE CMD MODE. ALL LISTS WILL WILL NOW GO TO THE INTERFACE. TO RETURN TO NORMAL, FORCE A SYNTAX ERROR BY TYPING AN ILLEGAL COMMAND TO THE PET (E.G. "ABAB [RETURN]").

IF THE PET "HANGS UP" AFTER THE CMD INSTRUCTION OR AFTER FORCING A SYNTAX ERROR, THE REASON IS THAT THE DEVICE CONNECTED TO THE TNW-2000 IS OFF-LINE AND IS REFUSING TO ACCEPT OUTPUT BY HOLDING ITS RTS OR DTR CONTROL LINE LOW. THIS "HANG-UP" CAN BE CLEARED BY PLACING THE DEVICE ON-LINE, BY DISCONNECTING EITHER THE RS-232 OR IEEE CABLE, OR BY RESETTING THE PET (POWER OFF/ON).

TO RECEIVE (INPUT)

FIRST, IT IS NOT POSSIBLE TO USE THE INPUT# COMMAND. THIS IS FOR A VARIETY OF REASONS. FIRST, THE PET HAS A TIME OUT FUNCTION BUILT INTO ITS IEEE INPUT SO IT WILL NOT WAIT FOR THE OPERATOR TO ENTER DATA. SECOND, IT APPEARS THAT THE INPUT COMMAND HAS A SOFTWARE BUG, BUT THIS IS HARD TO INVESTIGATE BECAUSE OF THE TIME OUT. THIRD, THE TNW-2000 CAN ONLY BRING IN A SINGLE CHARACTER AT A TIME DUE TO A PET IEEE SOFTWARE ANOMALY. TO INPUT, YOU MUST USE THE GET# COMMAND. USING THIS COMMAND WITH THE TNW-2000 IS SIMILAR TO USING IT WITH THE PET KEYBOARD. THE FOLLOWING PROGRAM EXAMPLE WILL INPUT CHARACTERS FROM A DEVICE UNTIL A CARRIAGE RETURN IS RECEIVED AND THEN OUTPUT THE STRING TO THE DEVICE.

```
10 OPEN 1,4,1: OPEN 2,4,0
20 A$=""
30 GET#2, C$: IF C$="" GOTO 30
40 A$=A$+C$: IF ASC(A$)<>13 GOTO 30
50 PRINT#1, A$: GOTO 20
```

THE NUMBER 13 IS THE DECIMAL VALUE FOR AN ASCII CARRIAGE RETURN. IF PARITY IS DISABLED FOR THE TNW-2000, BUT IS USED BY THE PERIPHERAL DEVICE, THE DECIMAL VALUES OF RECEIVED CHARACTERS WILL NOT MATCH THE ASCII TABLES. THE STANDARD METHOD IS TO MASK THE PARITY BIT USING ASC(A$) AND 127. DON'T TRY TO CONVERT A CHARACTER TO ITS ASCII VALUE WITHOUT FIRST TESTING FOR NULL("") SINCE ASC("") PRODUCES AN ERROR.

A FEW LAST REMINDERS

SELECTING THE BAUD RATE OF A PRINTER/KEYBOARD: THE BAUD RATE OF THE TNW-2000 MUST BE SET TO MATCH THAT OF THE PERIPHERAL DEVICE TO WHICH IT IS CONNECTED. IF THE PRINTER HAS A SIGNAL (DTR OR RTS) TO STOP THE TNW-2000 WHEN ITS (THE PRINTER'S) BUFFER FILLS, THEN THE BAUD RATE CAN BE SET AT THE HIGHEST RATE THE PRINTER CAN SUPPORT. IF THE PRINTER CAN'T "THROTTLE" THE TNW-2000'S OUTPUT, THEN THE BAUD RATE MUST BE SLOW ENOUGH TO KEEP FROM OVERFLOWING THE PRINTER'S BUFFER. WHEN INPUTTING DATA TO THE PET, KEEP IN MIND THAT THE PET IS ONLY FAST ENOUGH TO ACCEPT ABOUT 30 CHARACTERS PER SECOND (300 BAUD) WHEN RUNNING A BASIC PROGRAM. USING A MACHINE LANGUAGE LOOP, DATA CAN BE ACCEPTED AT THE FASTEST RATE OF THE TNW-2000.

INCOMPATIBLE REQUEST TO SEND/DATA TERMINAL READY SIGNALS: IF THE DEVICE INTERFACED TO THE TNW-2000 DOES NOT HAVE A METHOD TO THROTTLE OUTPUT, OR IF ITS METHOD IS NOT COMPATIBLE WITH THE TNW-2000, LEAVE THE SIGNALS FLOATING AND USE THE DEVICE AT A RATE FOR WHICH THE BUFFER DOESN'T OVERFLOW. GROUNDING PIN 4 OR 20 OF CONNECCTOR P2 (OR APPLYING A NEGATIVE VOLTAGE OF LESS THAN 25 VOLTS) WILL HALT OUTPUT FROM THE TNW-2000.

INTERFACE LEVELS: THE TNW-2000 INTERFACES TO RS-232 DEVICES WITH SIGNAL LEVELS UP TO +- 25 VOLTS. IT IS ALSO TTL COMPATIBLE, BUT WITH LOW NOISE MARGIN. YOU MUST USE AN ADAPTOR TO USE THIS INTERFACE WITH A CURRENT LOOP DEVICE.

OVERVIEW

RS-232C

RS-232

EIA (Electronic Industries Association) Standard RS-232-C is entitled "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange". The standard itself is a 29 page document whose major sections are titled "Electrical Signal Characteristics", "Interface Mechanical Characteristics", "Functional Description of Interchange Circuits", and "Standard Interfaces for Selected Communication System Configurations".

What this translates into is a convention for data transmission over short distances and at bit rates of up to about 20 kbps, which was originally designed to interface terminals to telephone modems, but which is now used very widely for the interconnection of computers and peripherals of all sorts, including terminals, printers, and plotters, as well as modems.

Background: Serial, Asynchronous

The transfer of digital data is often accomplished by sending the data in what is called serial form: a bit at a time over a single wire (as opposed to parallel, in which different bits travel over different wires at the same time) and using one of two general transmission techniques: synchronous or asynchronous. Synchronous data transmission requires a clock (usually transmitted on a separate line) to mark the start of each data bit interval. In addition, special synchronization data patterns may be used to allow for the receiver to locate the first bit of the message. With synchronous transmission, each data bit must follow contiguously after the sync word, since one data bit is assumed for each clock period.

With asynchronous transmission, a clock signal is not transmitted with the data and the characters need not be contiguous. This is made possible by adding synchronizing start and stop elements to each character as shown in the figure below.

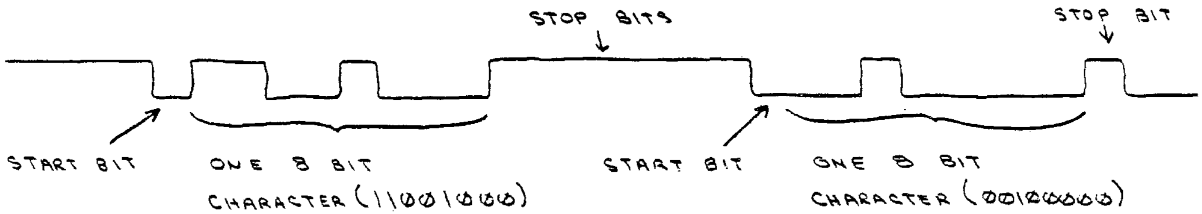


Figure 2-1 ASYNCHRONOUS DATA

The start element is a single logic zero (space) data bit that is added to the front of each character. The stop element is a logic one (mark) that is added to the end of each character. The stop element is maintained until the next data character is ready to be transmitted. (Asynchronous transmission is often refered to as start-stop transmission). Although there is no upper limit to the length of the stop element, there is a lower limit that depends on the data communications equipment characteristics. Typical lower limits are 1.0, 1.42, or 2.0 data bit times. The negative (logic)* going transition (from logic one to logic zero) of the start element defines the location of the data bits in one character. A clock source at the receiver is reset by this transition and is used to locate the center of each data bit, at which time the logic signal voltage level is sampled.

* positive voltage transition

There are several reasons for using asynchronous transmission. The major reason is that transmission is simplified since a clock signal need not be transmitted with the data. Another advantage of asynchronous transmission is that the characters need not be contiguous in time, but may be transmitted as they become available. This feature is valuable when transmitting data from manual entry devices such as keyboard.

Asynchronous transmission over a simple twisted wire pair can be accomplished at moderately high baud rates: 9600 baud or higher depending on the length of the wire, type of line driver, etc. When operating over the telephone network, a modem is required to convert the data pulses to tones that can be transmitted through the network.

While the RS-232-C specification applies to both synchronous and asynchronous data transmission, the TNW-2000 will support only asynchronous communications.

RS-232-C: Mechanical And Electrical

The RS 232 spec calls for a 25 pin connector, with the male part tied to the Data Terminal Equipment (DTE) and the female to the Data Communications Equipment (DCE). Electrically, a logic zero (SPACE) is represented by a voltage between +5 and +15 V, a logic one (MARK) by a voltage between -5 and -15 V. Note that the voltage falls as the logic level rises, and vice versa; this scheme is inverted for MIL STD 188C. The RS232 Standard also prescribes electrical impedences, drive capabilities, and signal voltage rate-of-change limits; consult the spec if you are really interested.

MIL STD 188C And Current Loops: IMPORTANT - Not all serial equipment uses the RS 232 interface signal levels. The two important exceptions are the MIL STD 188C low level interface and the high level current loop interface. MIL STD 188C differs from RS 232 in signal polarity: the voltage levels for MARK and SPACE are reversed, as mentioned above. If a MIL STD 188C equipment is inadvertently connected to a RS 232 interface, no damage will occur.

The high level interface is a more severe problem. IF A HIGH LEVEL EQUIPMENT IS CONNECTED DIRECTLY TO THE TNW-2000, SOMETHING WILL BE DESTROYED!

There are two types of high level interfaces. Standard Teletype equipment requires 120 Volts, 60 milliamperes since the interface signal must drive a high speed relay directly. Later Teletypes, including the very common Model 33, use a solid state high level interface. This is normally speced to work with 120 volts, 20 milliamperes, but since it is solid state and not a relay, it will actually work at a much lower voltage (but with the same current). Many serial equipments have been designed to be compatible with the 20 mill current loop, using a current of 20 milliamps, but at a voltage of 12 volts or less. The TNW-2000 does NOT connect directly to the current interface.

RS-232-C: Interchange Circuits

The signal interchange circuits defined by RS-232 C fall into four groups: ground, data, control, and timing. Since the timing signals (clocks) apply only to synchronous communications (which the TNW-2000 does not support) and ground is an uninteresting signal, we will be concerned here only with the data and control lines.

Embedded within a complete RS 232 interface are two separate bidirectional data channels. The so-called secondary channel is intended (in the context of the spec) to serve as a low speed backward channel or as an auxiliary channel to convey status information.

Associated with each of the two data channels are three control signals, Request to Sent (to DCE), Clear to Send (from DCE), and Received Line Signal Detector (from DCE). Six additional signals are associated with the interface as a whole rather than with either of the data channels in particular: Data Set Ready (from DCE), Data Terminal Ready (to DCE), Ring Indicator (from DCE), Signal Quality Detector (from DCE), and Data Signal Rate Selectors for both DTE and DCE. These control lines serve several major functions:

Operational Status: Data Terminal Ready (pin 20) is set by the DTE to indicate that it is functional (often a power-on indicator). Data Set Ready (pin 6) is the complementary function performed by the DCE.

Initiation of Data Transfer: Request to Send (pin 4) is activated by the DTE when it wishes to transmit data to the DCE; Clear to Send (pin 5) is the signal by which the DCE indicates that it is capable of receiving data from the DTE for transmission (i.e., is generating a carrier tone and receiving the carrier from the remote modem).

Status Checking: Signal Detect (pin 8) is set by the DCE to indicate that a carrier of sufficient amplitude is being received. Signal Quality Detector (pin 21) is set by the DCE to indicate that the quality of the communications line is satisfactory.

Initiation of Link: Ring Indicator (pin 22) is set by the DCE to indicate that an incoming call is being initiated (answer mode for the modem).

The modem orientation of these signal functions is clearly apparent. In the more general case in which the DCE and DTE roles are not being filled by a terminal and a modem, these signals can be used for whatever purposes the user may choose, provided that the interpretation assigned each signal is shared by the two devices.

RS-232-C Connector Pin Assignments - For completeness only, we present in this subsection a complete list of the signal to pin assignments prescribed by the RS-232-C specification (including the equivalent signal designations of the CCITT, the international standards organization corresponding to the American EIA).

Most systems will use a relatively small subset of these signals. In fact, many peripheral devices use no control lines, or, if they are used, they can be jumpered rather than actively manipulated.

#	EIA	CCITT	FROM	NAME
1	AA	101		Protective Ground
2	BA	103	DTE	Transmitted Data
3	BB	104	DCE	Received Data
4	CA	105	DTE	Request to Send
5	CB	106	DCE	Clear to Send
6	CC	107	DCE	Data Set ready
7	AB	102		Signal Ground
8	CF	109	DCE	Received Line Signal Detector
9				(Reserved for Data Set Testing)
10				(Reserved for Data Set Testing)
11				Unassigned
12	SCF	122	DCE	Sec. Rec'd Line Signal Detector
13	SCB	121	DCE	Secondary Clear to Send
14	SBA	118	DTE	Secondary Transmitted data
15	DB	114	DCE	Transmission Signal Element Timing
16	SBB	119	DCE	Secondary Received data
17	DD	115	DCE	Receiver Signal Element Timing
18				Unassigned
19	SCA	120	DTE	Secondary Request to Send
20	CD	108.2	DTE	Data terminal Ready
21	CG	110	DCE	Signal Quality Detector
22	CE	125	DCE	Ring Indicator
23	CH/CI	111/112		Data Signal Rate Selector: DTE/DCE
24	DA	113	DTE	Transmitter Signal Timing Element
25				Unassigned

[The TMW-2000 does not use all of these pins!]

OVERVIEW:

IEEE BUS

The happiest thing about the IEEE bus interface is that the user of the TNW module need know virtually nothing about how it works. Its use with PET, in particular, is almost completely transparent. Physically you just connect the TNW module to your computer with a cable (keyed so it can't fit onto the wrong connector, or upside down). After checking the IEEE bus "Address" of your module as set by DIP switches on the board, all that's left is programming. It is nevertheless recommended that the TNW module user read the remainder of this sectin in order to become familiar with what is actually happening in his machine.

"IEEE 488" is a shorthand designation for "IEEE Standard 488-1975" which is the "IEEE Standard Digital Interface for Programmable Instrumentation." The IEEE (Institute of Electrical and Electronic Engineers) developed this standard (their 488th, and adopted in 1975), so that programmable instruments made by different manufacturers, but all following this Standard , could communicate easily with one another. The Standard has also been adopted by ANSI (the American National Standards Institute) as ANSI MC 1.1-1975.

The actual form of the interface was developed by the Hewlett-Packard Company (who call it the Hewlett-Packard Instrument Bus, or HP-IB), and later adopted by the IEEE and ANSI. The instruments for which it was originally intended include such devices as frequency synthesizers, counter/timers, and the like. The IEEE 488 interface suddenly became very important to the world of personal computers when Commodore designed this interface into their PET.

The IEEE bus allows the transfer of digital data among up to 15 devices connected together by cables whose total length is less than 20 meters, so it is not designed to work over geographical distances. Data is transferrred in parallel format (8 bits, or one "byte", at a time), and in an asynchronous fashion (the speed of each transaction on the bus is set by the slowest device participating in that transaction), with a total maximum data rate on any signal line of 1 Mbps (million bits per second).

The cable that connects devices on the IEEE bus has 24 wires: 8 grounds and 16 signal lines. These signals are functionally divided into 3 groups: 8 data lines (DIO1-DIO8), 3 data byte transfer control lines, and 5 bus management lines.

The 3 data byte transfer control lines are used in what is called a "3 wire handsnake" to ensure that the 8 data bits are properly transferred. These signals are named Not Ready for Data (NRFD), Data Valid (DAV), and Not Data Accepted

(NDAC). The signals are active low, which means that if any one or more devices on the bus pulls the signal level low (sets it), it is low. Otherwise, it is high (released).

Each transfer of data involves a "Talker" or transmitter, and one or more "Listeners" or receivers. (We'll explain later how devices get to be Talkers and Listeners). Essentially the way it works is that the Talker waits until all Listeners have released NRFD (that is, have indicated that they ARE ready for data), then it puts the data onto the data lines and sets DAV, to tell the Listeners that the data can be sampled. As each Listener detects that DAV is set, it sets NRFD, then takes data and releases NDAC. When the final Listener released NDAC, the signal level on tat line goes high and the Talker knows that all Listeners have received the data. The Talker then releases the DAV line to indicate that data is no longer valid and available on the line. When all the Listeners are ready to receive the next byte, the NRFD line will rise again and the Talker can put the next byte on the data lines (if it has another byte), and so on.

The dialogue, then, sounds something like this:

> Listeners: "All of us are ready for you to give us the next byte."
>
> Talker: "Here is the next byte for you to copy."
>
> Listeners: "Now we are not ready for the NEXT byte because we are now working on copying the current one."
>
> Listeners: "We have all read the current byte."
>
> Talker: "I have now removed the byte from the data lines."
>
> Listeners: "All of us are ready for you to give us the next byte."

And so on until all the data has been transferred.

Devices get to be the Talker or Listener through the use of the 5 bus management lines. One and only one device on the bus performs the function of a Controller, who (alone) sets and releases a signal called Attention (ATN). Some devices (Like the TNW 488/103 and the TNW 488/232) are not capable of performing the Controller function. Others (like the PET) assume that they are the Controller, and are incapable of relinquishing control (this is why 2 PETs cannot be connected directly together via the IEEE bus: they would both try to be the Controller, with very bad results). Finally, some devices can either act as the Controller or relinquish control to another device.

When the Controller sets ATN and transmits a data byte, every device on the bus becomes a temporary Listener, and interprets the data in a particular way. Every device on the bus has an "Address," a number between 0 and 30, assigned to it. If the data byte has a value of between 32 and 62, the device whose address is 32 less than the value of the byte makes itself a Listener. If the value is 63, all Listeners stop Listening) this is called "Unlisten"). If the value of the byte is between 64 and 94, the device whose address is 64 less than that value becomes a Talker; 95 is the "Untalk" command. Some other values (less than 32 or greater than 95) also have specific meanings when ATN is set, but it is not important to describe them here.

It is not necessary, although it is usual, that a device should have the same talk address as Listen address. Some devices , including some other TNW interface modules, use multiple addresses for both Talk and Listen, in order to simplify the distinction between data and command/status information. The Commodore PET has dedicated the device numbers (addresses) 0 through 3 for internal peripherals (screen, tape drives, and keyboard), so these numbers are not available for addresses for devices to function with the PET. Other restrictions may apply for use with other computers.

The other 4 bus management signals besides ATN are Service Request, (SRQ), Interface Clear (IFC), Remote Enable (REN), and End or Identify (EOI). SRQ is used by a device on the bus to signals to the Controller that it requires some attention. The Controller then executes what is called a Serial Poll in order to see which device is doing the requesting. The PET doesn't do anything with SRQ. IFC is used to initialize all devices on the bus; PET performs this function only when it is powered up. REN is relevant only to instruments that can function either under program control (over the IEEE bus) or through physical controls on the instrument front panel. Finally, EOI is set by a Talker to indicate that the data byte being transferred is the last of the sequence. Again, the user of the TNW module with PET need not be concerned with any of these details, they are included here for information only.

ASCII

Bits b7					0	0	0	0	1	1	1	1
b6					0	0	1	1	0	0	1	1
b5					0	1	0	1	0	1	0	1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	COLUMN → / ROW ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL ØØ	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
0	0	0	1	1	SOH Ø1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0	0	1	0	2	STX Ø2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0	0	1	1	3	ETX Ø3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0	1	0	0	4	EOT Ø4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0	1	0	1	5	ENQ Ø5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0	1	1	0	6	ACK Ø6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0	1	1	1	7	BEL Ø7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	0	0	0	8	BS Ø8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	0	0	1	9	HT Ø9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	0	1	0	10	LF 1Ø	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	0	1	1	11	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	0	0	12	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	¦ 124
1	1	0	1	13	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	0	14	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1	1	1	1	15	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127

NUMBERS NEXT TO CHARACTERS ARE THE ASCII DECIMAL VALUE.

NUL — Null
SOH — Start of Heading
STX — Start of Text
ETX — End of Text
EOT — End of Transmission
ENQ — Enquiry
ACK — Acknowledge
BEL — Bell
BS — Backspace
HT — Horizontal Tab
LF — Line Feed
VT — Vertical Tab
FF — Form Feed
CR — Carriage Return
SO — Shift Out
SI — Shift In
DLE — Data Link Escape
DC-1 to 4 — Device Control
NAK — Negative Ack.
SYN — Synchronous Idle
ETB — End of Trans Block
CAN — Cancel
EM — End of Medium
SUB — Substitute
ESC — Escape
FS — File Separator
GS — Group Separator
RS — Record Separator
US — Unit Separator
DEL — Delete (Rubout)

CONTROL CHARACTERS

ASCII

b_7 b_6 b_5					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	COLUMN / ROW	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL 00	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
0	0	0	1	1	SOH 01	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0	0	1	0	2	STX 02	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0	0	1	1	3	ETX 03	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0	1	0	0	4	EOT 04	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0	1	0	1	5	ENQ 05	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0	1	1	0	6	ACK 06	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0	1	1	1	7	BEL 07	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	0	0	0	8	BS 08	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	0	0	1	9	HT 09	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	0	1	0	10	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	0	1	1	11	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	0	0	12	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	¦ 124
1	1	0	1	13	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	0	14	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1	1	1	1	15	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127

NUMBERS NEXT TO CHARACTERS ARE THE ASCII DECIMAL VALUE.

NUL — Null
SOH — Start of Heading
STX — Start of Text
ETX — End of Text
EOT — End of Transmission
ENQ — Enquiry
ACK — Acknowledge
BEL — Bell
BS — Backspace
HT — Horizontal Tab
LF — Line Feed
VT — Vertical Tab
FF — Form Feed
CR — Carriage Return
SO — Shift Out
SI — Shift In
DLE — Data Link Escape
DC-1 to 4 — Device Control
NAK — Negative Ack.
SYN — Synchronous Idle
ETB — End of Trans Block
CAN — Cancel
EM — End of Medium
SUB — Substitute
ESC — Escape
FS — File Separator
GS — Group Separator
RS — Record Separator
US — Unit Separator
DEL — Delete (Rubout)

CONTROL CHARACTERS

[4K / 8K] PET CHARACTER SET

As mentioned in the text, the PET character set is a sort of half-ASCII.

Keyboard Encoding

Two keys on the PET keyboard perform special functions: holding SHIFT down sets the most significant bit (bit 7) of any other key pressed to 1; otherwise that bit is 0; the STOP key causes a break in a running BASIC program; shifted, this key (RUN) performs the equivalent of LOAD followed by RUN. The operating system interprets all other keys as character input: GETC$:C=ASC(C$) yields the following values:

RVS	18
HOME	19
CRSR down	17
CRSR right	29
DEL	20

All other keys produce their ASCII equivalents: RETURN yields 13, and the other keys are all the ASCII values from 32 through 95 inclusive.

Screen Encoding

C$=CHR$(C):PRINTC$ produces the following effects (numbers not shown have null effect):

13	RETURN (same as ASCII 13)
17	CRSR down
18	RVS (begin reverse field printing
19	HOME
20	DELETE delete character
29	CRSR right
32-95	same as ASCII
96-127	repeat of ASCII 32-63
141	RETURN
145	CRSR up
146	OFF (end reverse field of characters)
147	CLR (clear screen)
148	INST insert character
157	CRSR left
160-255	PET graphics charcters. Execution of POKE 59468,14 replaces the graphics characters 193-218 with lower case a-z.

"High Level" Machine Language Hooks
for PET IEEE 488 bus....

FFC6 LDX with the file number of an open input file, then JSR FFC6 makes the device a talker (then use JSR FFE4 to get characters)

FFC9 LDX with the file number of an open output file, then JSR FFC9 makes the device a listener (then use JSR FFD2 to print characters)

FFCC JSR FFCC issues UNLISTEN and UNTALK commands on the IEEE 488 bus, restores I/O to "from keyboard" and "to screen" (clobbers the accumulator A)

FFD2 LDA with a byte, then JSR FFD2 to print the char to the current output file

FFE4 JSR FFE4 gets a character from the current input file, leaves it in A

FFCF JSR FFCF inputs a character from the current input file (harder to use?)