CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-001
Date: October 31, 1986
Subject: Systech Patch Panel
Submitted By: Dick Baker

NOTE: This replaces current Comm-001.

Systech patch panel compatibility. The patch panel for the Sytech async controller model 800/1600 will not work on the controller model 850/1650 and vice versa. The connectors on the end of the patch panel signal cables are different for the two models. This prevents connecting the wrong model patch panel. So if you have to switch models the controller to bulkhead signal cable will also have to be replaced. The small controller to bulkhead DC cable will work on either model .

Here are the part numbers involved.

Model 800/1600	
Controller	221-000001-202
Patch Panel	500-000133-200 (not D shell connector)
Signal Cable	601-500002-200 (not D shell connector)
Model 850/1650	
Controller	221-000002-200
Patch Panel	500-000161-200 (D shell connector)
Signal Cable	601-500009-200 (D shell connector)

CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-002
Date: May 30, 1986 (Revised 3/1/87)
Subject: SPU Modem
Submitted By: Bill Georgia

SPU MODEM PART NUMBER 211-000100-200 ONLY

Problem: SPU Modem powers up in "Voice Mode".

For those who have installed FM0033 and the SPU Modem still powers up in "Voice Mode", rather than "Data Mode", there is now an answer. Buck Treherne has reported that doubling the size of the capacitor (to .02 MFD) does the trick. Probably the easiest way would be to parallel the .01 MFD with another .01 MFD. Tried it myself and it worked for me, too.

NOTE: This Tech Tip is **not** applicable to SPU Modem Part Number 211-000105-200.

MCU-FS

CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-003
Date: March 1, 1987 (Rev. 07/15/87)
Subject: Communications Cabling
Submitted By: Brad Jones

The purpose of this Tech Tip is to identify and describe the various cables that can be used to connect Terminals, Modems, etc. to the C-1.

- **C-1 ↔ Asynchronous Local Terminal:**

C-1		TERMINAL	SIGNAL
1	↔	1	Frame Ground
2	←	2	Transmit Data
3	→	3	Receive Data
7	↔	7	Signal Ground
20	←	20	Data Terminal Ready

PART NUMBERS

604-100002-00x (Terminal End is Female)
604-100003-00x (Terminal End is Male)
P/N suffix defines length: -001=5 Feet; -002=10 Feet; -003=25 Feet; -004=50 Feet; and -005=70 Feet.

- **C-1 ↔ Modem:**

C-1		MODEM	SIGNAL
1	↔	1	Frame Ground
2	←	3	Receive Data
3	→	2	Transmit Data
4	←	5	Clear to Send
5	→	4	Request to Send
6	→	20	Data Terminal Ready
7	↔	7	Signal Ground
8	←	8	Carrier Detect
20	←	6	Data Set Ready
22	←	22	Ring Indicator

PART NUMBERS

604-100001-00x (Modem End is Male)
604-100007-00x (Modem End is Female)

P/N suffix defines length: -001=5 Feet; -002=10 Feet; -003=25 Feet; -004=50 Feet; and -005=70 Feet.

- **C-1 ↔ Micom® Switch:**

C-1		MICOM®	SIGNAL
1	↔	1	Frame Ground
2	←	3	Receive Data
3	→	2	Transmit Data
6	→	20	Data Terminal Ready
7	↔	7	Signal Ground
20	←	6	Data Set Ready

PART NUMBERS

604-060001-00x

P/N suffix defines length: -001=5 Feet; -002=10 Feet; -003=25 Feet; -004=50 Feet; and -005=70 Feet.

Note: At the C-1 end, jumper **from** pin 4 (RTS) to pin 5 (CTS).

CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-004
Date: March 30, 1987
Subject: Ethernet Part Numbers
Submitted By: Brad Jones

Convex Part Numbers for the various components required to connect two (2) or more C-1 Systems, a C-1 and another system, etc. together via Ethernet are shown below.

PART NUMBER	*DESCRIPTION*	*QUANTITY*
604-010001-202	Cable Assembly, Ethernet, 24 feet (Taps must be **more than** 8 feet apart; so each 24 foot length will support 2 Taps)	1
211-000101-005	Terminator Assembly, 49.9 Ohms (one required per cable end)	2
304-000021-001	Adapter, Ethernet Cable (required when connecting two 24 foot lengths together)	1

Note: To add to **existing** 24, 48, 72, etc. foot lengths of Ethernet Cable, order Convex Part Number 550-000151-204, which will obtain the Part Numbers and Quantities shown above.

- **Basic, Unterminated Ethernet Cable - Manufacturer Part Numbers:**

MANUFACTURER	*PART NUMBER / MARKINGS*
Belden	9880 / AWM Style 1478, Ethernet/IEEE 802.3, Coax NJ
Interlan	PT 704-AAYA-000 / MONCO 9U AWM Style 1478

- **Miscellaneous Convex Part Numbers:**

PART NUMBER	*DESCRIPTION*
211-000101-200	Multibus Ethernet Controller
211-000101-001	Ethernet Transceiver (Line Tap)
211-000101-003	Kit, Transceiver Installation
601-150001-200	Cable, Ethernet Controller to Bulkhead, 5 feet
601-150001-202	Cable, Ethernet Controller to Bulkhead, 3 feet
601-150001-203	Cable, Ethernet Controller to Bulkhead, 15 inches
604-150001-001	Cable, Bulkhead to Ethernet Transceiver (Tap), 20 feet
604-150001-002	Cable, Bulkhead to Ethernet Transceiver (Tap), 50 feet

CONVEX	**Field Support Tech Tip** **Product:** C-1 **Tech Tip Number:** Comm-005 **Date:** April 15, 1987 **Subject:** Excelan Ethernet Messages **Submitted By:** Brad Jones

Typical examples of Excelan Ethernet Transmit and Receive Error messages are shown below:

```
[CCU7@11:00:00] ex: receive error 4
[CCU7@11:00:00] ex: receive error 10
[CCU7@11:00:00] ex: receive error 28
[CCU7@11:00:00] ex: transmit error 40
```

The Transmit and Receive Error Codes are described below.

- **Receive Error Codes -**

*CODE**	*DESCRIPTION*
00	Packet received with no error.
X4	Packet received longer than buffer available; truncated.
X8	Runt packet received (shorter than expected).
1X	Packet received with alignment error.
2X	Packet received with CRC error.
4X	No packet received; buffer supplied was less than 64 bytes.
0A1	Reception failed; EXOS 201 was not in controller mode.

- **Transmit Error Codes -**

*CODE**	*DESCRIPTION*
00	Successful transmission; no retries required.
X1	Successful transmission; one (1) retry required.
X2	Successful transmission; more than one (1) retry required.
X8	(Version 2.0 Transceivers only) Indicates the absence of the signal "SQE TEST" during the Inerframe Spacing Interval. This code can be ORed with all others ***except*** "40" and "0A1". (NOTE: Normal Convex configuration of the Excelan Ethernet Controller, Part Number 211-000101-200, is to **disable** this check)
1X	Transmission failed due to excessive collisions.
2X	No Carrier Sense signal detected during transmission.
4X	Transmission failed; transmit length not in range.
0A1	Transmission failed; EXOS 201 was not in controller mode.

* = All codes are in hexadecimal. Except where stated otherwise, all codes can be ORed with other codes. In the codes shown above, "X" is *don't care*.

...continued on next page

...continued from previous page

Excelan Ethernet Error Codes *other than* Transmit and Receive are described in the following table (see previous page for Transmit and Receive Error Codes).

CODE	OPERATION	DESCRIPTION
A0	Configuration	Invalid address for configuration message.
A1	UPLOAD EXECUTE DOWNLOAD	Invalid request.
A2	EXECUTE	Invalid starting address.
A3	UPLOAD DOWNLOAD	Specified memory does not exist; no copy done. Destination memory block overlaps reserved memory; no copy done.
A4	Configuration	Invalid operation mode parameter.
A5	Configuration	Invalid host data format test pattern (i.e., the NX 200 cannot find any reasonable conversion to derive the expected data pattern from that supplied in the test pattern).
A7	Configuration	Invalid configuration message format. This may occur if reserved fields contain an improper value.
A8	Configuration	Invalid movable data block address parameter.
A9	Configuration	Invalid number of processes parameter.
AA	Configuration	Invalid number of mailboxes parameter.
AB	Configuration	Invalid number of address slots parameter.
AC	Configuration	Invalid number of hosts parameter.
AD	Configuration	Invalid host message queue parameter. The NX 200 returns this code upon detection of any inconsistency in the message queue (i.e., bad interrupt type, invalid segment address, bad link, etc.).
AE	Configuration	Insufficient memory for movable data block.
AF	Configuration	Net boot failed.
B0	Self-Test	Checksum on NX 200 EPROM failed.
B1	Self-Test	Memory test failed for 0 to 128 K.
B2	Self-Test	Memory test failed for 128 K to top-of-memory.
B3	Self-Test	Counter test failed.
B4	Self-Test	Interrupts test failed.
B5	Self-Test	Transmission test failed.
B6	Self-Test	Receive test failed.
B7	Self-Test	Local loopback data path test failed.
B8	Self-Test	CRC test failed.

...continued on next page

...continued from previous page

CODE	OPERATION	DESCRIPTION
B9	Self-Test	Checksum on physical address EPROM failed.
BA	Self-Test	Bus timeout.
BB	Self-Test	Ethernet chip initialization failed.
BC	Self-Test	Ethernet chip self-test failed.
BD	Self-Test	Ethernet chip resource counter failed.
BE	Self-Test	Ethernet loop-back test alignment error.
BF	Self-Test	iSBX board not in place.
C0	Operational	Specified time exhausted.
C1	Operational	Host memory read/write test failed.
C8	Operational	Parity hardware logic failed.
C9	Operational	NMI interrupt for bus timeout failed.
CA	Operational	Host interrupt test failed.
CB	Operational	Command unit test failed.
CC	Operational	Divide error exception.
CD	Operational	Undefined interrupt type.
CE	Operational	Command not executed by the CU of the 82586.
CF	Operational	Command block sync failed between hardware and software.
D1	NET_RECV NET_ADDRS	The specified slot does not exist or access is prohibited.
D2	NET_RECV	The specified slot was empty.
D3	NET_ADDRS	Improper address. Multicast slots can only take multicast addresses and the physical slot can only take a physical address. Attempting to write the broadcast slot (number 255) results in this error.
0A1	NET_RECV NET_ADDRS NET_STSTCS	Failed. The EXOS 201 is not in controller mode.

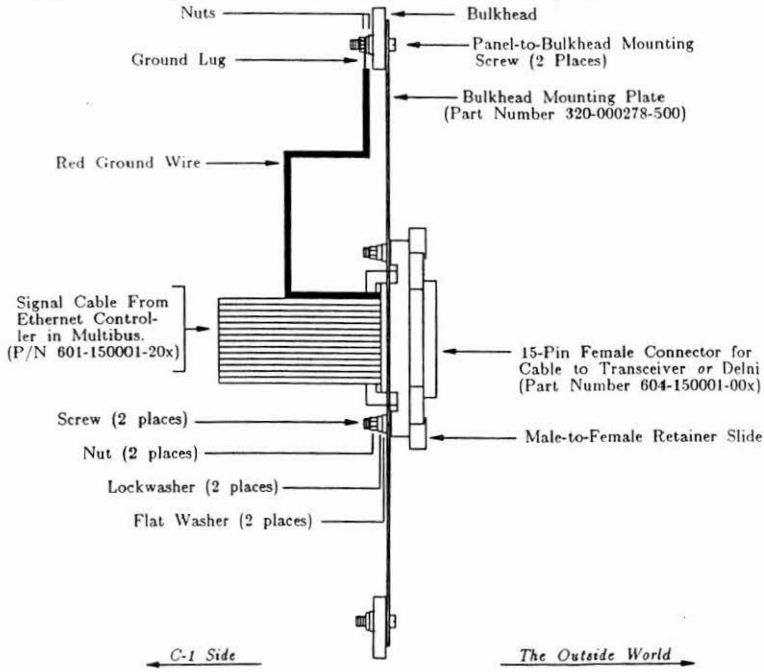
CONVEX

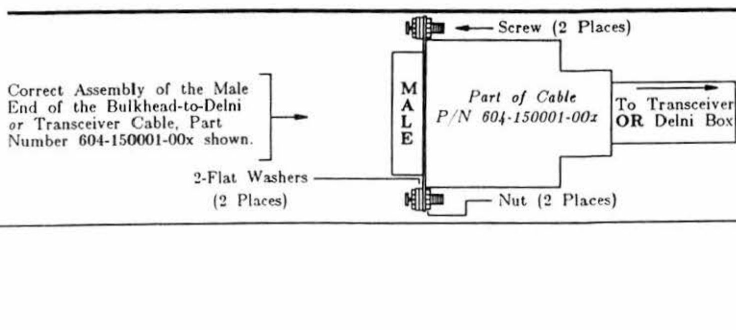
Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-006
Date: September 15, 1987
Subject: Ethernet Bulkhead Connector
Submitted By: Brad Jones

Two problems have been seen from time-to-time that can contribute to Ethernet Transmit and Receive errors: (1) Mounting the Bulkhead Connector of Cable Part Number 601-150001-20x on the wrong side of the Bulkhead Mounting Plate and, (2) incorrect placement of the flat washers on the male connector of Cable Part Number 604-150001-00x. Either problem alone or both together will result in unreliable mating of the Ethernet cable to the C-1 Bulkhead connector. The following items must be verified and, if necessary, corrected as soon as possible on installed systems. Additionally, these items must *always* be verified during future installations.

- **Proper Mounting of the Bulkhead Connector** (*Shown Actual Size*)





CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-007
Date: December 15, 1987
Subject: Excelan Ethernet Firmware
Submitted By: Brad Jones

The **original** Tech Tip Number "Comm-004" (March 15, 1987) attributed some problems to Revision 4.4 of the Excelan Ethernet Firmware. This proved to be an unfounded decision and, as a result, the original version of the Tech Tip was **voided** on March 30, 1987 and replaced with the current Tech Tip Number "Comm-004", *Ethernet Part Numbers.*

At this time the Excelan Ethernet Firmware situation is as follows:

Excelan Ethernet Firmware Revision 5.3 or later is required **only** if a system is using *DECnet* or *Convex COVUEnet* software. In all other applications, Excelan Ethernet Firmware Revision 5.3 and earlier is acceptable and useable.

CONVEX

Field Support Tech Tip

Product: C-1
Tech Tip Number: Comm-008
Date: February 6, 1988
Subject: Terminal/Modem Conversion
Submitted By: Rick Miller

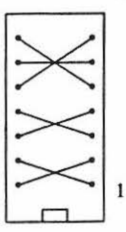
There are two ways to convert a port on a systech patch panel from a terminal to a modem port:

METHOD 1: If you use modem cable #604-000001-00x there's no need to mess with the systech panel that's setup with terminal header plugs. This is the easiest method. (You can troubleshoot a modem port with a terminal and it's cable with this type of setup.)

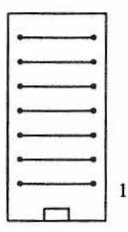
METHOD 2: Or you could strap the patch panel internally via a jumper plug. (For a modem, use #221-000001-203, a Convex number.) This is installed next to the USART for the port being changed. It's handy if you've plenty of terminal cables and wish to avoid ordering or running modem cables.

NOTE: If time is short... you can quickly manufacture an IC header (modem or terminal plug) by using the drawings below.

Modem Plug



Terminal Plug



CONVEX

Field Support Tech Tip
Product: C-1
Tech Tip Number: Comm-009
Date: January 31, 1989
Subject: RS-232 Explanation
Submitted By: AL Haddix

RS-232 DEFINITION

The purpose of RS-232 is to provide a standard interface configuration for the transmission of data between DATA TERMINAL EQUIPMENT and DATA CIRCUIT-TERMINATOR EQUIPMENT, employing serial binary data interchange.

This will serve as an explanation of this standard as well as the CONVEX implementation, and to supply basic definition of terms and simple explanation of standard application techniques. We will concern ourselves primarily with full-duplex asynchronous operation and ignore for this discussion the use of synchronous applications.

The following terms are used extensively when conversing on RS-232 and are frequently misunderstood:

1) Full-Duplex refers to the transmission of data in both directions, over the same lines simultaneously. In other words while transmitting over the TX line the same unit can receive data over its RX line.

2) Asynchronous transmission is sometimes referred to as start/stop format because of the use of start and stop bits to bracket the incoming data. Data is transmitted using standard ASCII code and usually an odd or even parity bit, forming an 8 bit character.

3) Data Terminal Equipment (DTE) Refers to equipment such as computers terminals and printers and is generally responsible for origination of local signals, such as, Data Terminal Ready and Request to Send.

4) Data Circuit-Terminator Equipment (DCE) is generally a modem or equipment that is emulating a modem. DCE is responsible for the establishment and maintenance of the connection. This is accomplished through the DCE signals: Clear to Send, Data Set Ready and Carrier Detect.

...continued from previous page

RS-232 OVERVIEW

RS-232 signals can be divided into four groups: Data, Ground, Control and Timing. As synchronous operation will not be discussed in this text, timing will be ignored.

The ground leads are by far the simplest to understand and consist of the frame ground on pin 1 and the signal ground on pin 7. The frame ground is a simple power ground to keep the connections at the same potential much as the ground on any household appliance functions. The signal ground is used as a reference for all other signals.

The Data lines are referred to as transmit and receive and are located on pins 2 and 3 repectively. These lines perform exactly as their name would indicate with transmit being considered a DTE signal and receive a DCE signal.

There are five (5) standard control lines associated with RS-232 and these are:

1) Data Terminal Ready (DTR) indicates the DTE is on line and ready to connect with the local DCE.

2) Request to Send (RTS) is a signal that is initiated by the remote DTE indicating that it wishes to transmit.

3) Data Set Ready (DSR) indicates the DCE is available for transmission.

4) Clear to Send (CTS) is initiated by the DCE to indicate that it is clear to initiate a transmission. This would be in response to a Request to Send.

5) Data Carrier Detect (DCD) indicates that carrier is present and a transmission is currently in progress.

As reported earlier in this text, these signals are used to establish and maintain the connection of the data lines. No other reason exists for these lines, except to insure the proper delivery of data.

...continued from previous page

This is the configuration of the RS-232 connector:

FUNCTION	PIN	NAME	DIRECTION
DATA	2	TX	From DTE
DATA	3	RX	From DCE
CONTROL	4	RTS	From DTE
CONTROL	5	CTS	From DCE
CONTROL	6	DSR	From DCE
CONTROL	8	DCD	From DCE
CONTROL	20	DTR	From DTE
CONTROL	22	RI	From DCE
GROUND	1	FG	N/A
GROUND	7	SG	N/A

RS-232 OPERATION

Operation of RS-232 is relatively easy remembering that the primary considertion is that for every signal raised from a DCE there must be a corresponding signal from the DTE and vice-versa.

The primary signal used to initiate a connection is Data Terminal Ready (DTR). This signal must be active for the DCE equipment to even respond. For explanation purposes it will be easier to briefly summarize the interaction of these signals using the example of half-duplex mode:

1) RTS is initiated by the far end DTE.

2) DCD is checked by the local DCE to verify RTS has been asserted by the remote DTE.

Field Support Tech Tip

Tech Tip Number: Comm-009

...continued from previous page

3) If the remote end's RTS is active (DCD is on), the modem ignores the request and the DTE retry's. This indicates that there is presently a connection already established and the DCE cannot respond to the new request. If DCD is off (meaning no activity), the local modem responds by asserting CTS to the remote DTE.

4) The DTE then sends data over the TX line and the modem transmits.

5) The local modem receives this data and places it on the RX line, pin 3, for presentation to the local DTE.

6) the originating DTE continues with RTS asserted until all data is transmitted. Then it drops RTS, which, in turn drops DCD at the far end and CTS locally.

7) The line is now idle and either DTE can now obtain control of the line.

This process will be repeated over and over until the communication process is complete. The important thing to remember is that DTR was asserted through all of this and if DTR were to be turned off for any reason, the connection would be immediately terminated.

Data Terminal Ready can be dropped at any time during this process, for any of the following reasons:

1) Hang up the line.

2) Place the DTE in local "mode".

3) Sending a Control-D character, which will be interpreted as a disconnect.

In the full-duplex mode of operation both DTE's would maintain RTS constantly. The DCE's would in turn maintain CTS and also DCD as the DTE would be maintaining RTS. This configuration would maintain a constant connection for full-duplex transmission.

...continued from previous page

When the equipment is in the same room it is not necessary or desired to use modems for the data connections. In this event the lines can be cross-connected in order to emulate DCE signals with the available DTE signals. In this manner either side may emulate DTE or DCE, depending on the direction of transmission.

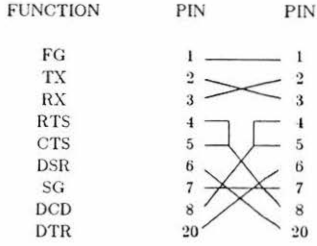
When cross-connecting it is necessary to remember that the DTE signals, DTR and RTS, must be used to create the DCE signals, DSR, CTS and DCD. Also remember that outputs must be routed to inputs.

In order to emulate DSR (pin 6), DTR (pin 20) may be crossed over. In other words, you can cross-connect pin 20 at one end to pin 6 at the far end. As CTS is a response to RTS this can be easily emulated by connecting pin 4 to pin 5 on the same side and in order to generate DCD on the far end then this connection can be run to pin 8 on the far end.

The data connections can be similarly handled by crossing pins 2 and 3 of both ends, so that the xmit line of one end is run to the receive line of the far end. In this manner we will be able to handle our our full-duplex operation.

The grounds would, naturally, be run straight across from one end to the other. The ring indicator on pin 22 would not be necessary and so could be omitted.

The final generic connection would appear as below:



...continued from previous page

FLOW CONTROL

There are some devices in the world that do not transmit and are much slower than the standard DTE and these require a different connection format. The primary example of such a device would be a line printer.

For these devices it will be necessary to implement flow control. This involves the regulation of data transmitted between two devices and the operation of one is significantly slower than the other. Printers fall into this category because of the mechanical operation required. So it would be necessary to suspend data flow while the printer returns to the home position, or the paper is being changed as well as numerous other functions unique to a line printer.

There are two standard methods of flow control and these are XON/XOFF and hardware flow control.

XON/XOFF is primarily a software implementation where a control character is returned to the system to indicate a pause in the transmission and then another control character will reinitiate the transmission.

The hardware method is very similar to the cross-connetion discussed previously, requiring the printer to yank a predetermined line to indicate a pause and then release it when ready. This line will vary between manufacturers and can be determined by examination of the operations manual. The CTS signal is used by many printers to handle this hardware flow control, but other printers use other facilities and will be indicated in the manual.

CONVEX IMPLEMENTATION

The RS-232 standard actually defines several possible interfaces between DTE and DCE. In any application, devices use only a subset of the complete standard depending on capabilities and requirements. The Systech async controller addresses all of the signals that have been previously discussed. But every device that could be interfaced to this controller will vary as to functionallity and implementation. Keeping this in mind it then becomes obvious that there are numerous possibilties for cable configurat ions.

...continued from previous page

In some instances it is only necessary to supply a four wire connection. This includes only the two grounds and the transmit and receive lines. While this fills the basic requirements it is not advisable as it does not allow for noise and fluctuations involved with any computer application. In addition the systech controller does not regulate the number of interrupts per port, so with this configuration and a noisy or unterminated line it is possible to flood the CCU and the JP with unnecessary interrupts thus reducing the overall performance of the system.

Using the basic control lines in the configuration insures a greater degree of control and safety. There will be fewer garbled or lost characters and devices will behave more closely to their designed specs.

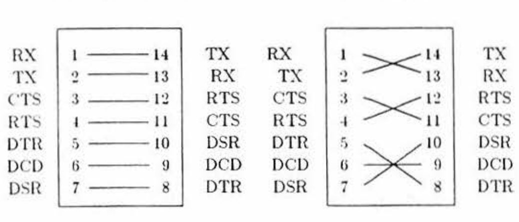
It is also important to note that there are two methods, with the Systech controller, of changing the signal cross-connections:

METHOD 1

Systech has supplied jumper configurations on the individual ports to change the hardware handshaking. This in essence changes the characteristics of the port from a DTE to a DCE configuration. The 2 standard jumpers in use are labeled modem and terminal. The modem jumper gives the Convex a DTE emulation, while the terminal plug maintains the DCE characteristics.

The following are wiring diagrams for jumper plugs on systech ports:

FOR TERMINAL　　　　FOR MODEM



* NOTE: The system is shipped initially with the terminal jumper installed, but can be changed on request.

...continued from previous page

It is obvious from the above diagrams that the purpose of these jumpers is to accomplish the cross connections instead of relying on a cable solution. This would allow the cable for any device to be wired in a straight through manner.

METHOD 2

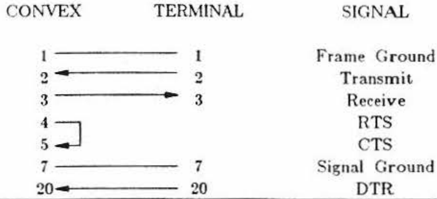
The second method of handling the handshake is to leave the terminal jumper installed and change the physical configuration of the cable. Using this method would enable the use of a straight through cable when connecting a modem to the host system. It is important to keep in mind that these two methods can conflict and so it is advisable to be aware of the type of jumper installed on each port, so as not to create problems with the cable configuration.

Again, it is important to understand, that all manufacturer's use different RS232 signals to represent various functions. But understanding the configuration at the Convex end should make it relatively easy to cross the appropriate leads.

So far, the discussion has dealt, primarily, with generic functions and explanations. Unfortunately, the implementation varies with almost any device. This is because of different interpretations as well as application and function of the various RS232 devices available. So, the discussion will now turn to a more specific implementation of RS232.

The following examples will deal directly with RS232 devices connected to a port on the Systech async controller installed on a CONVEX system. In each case the specific implementation will be a minimal recommendation. There will be other alternatives for each example, so please, do not understand this to be the only possibility.

The first example will be an explanation of the cross-connection necessary for a C. Itoh terminal. In each case the example will assume that a terminal jumper is installed in Systech async port.



...continued from previous page

As can be seen from the above configuration, this will require a minimum of 5 conductors to accomplish this. While it is common practice to use only three conductors, this is not recommended for the reasons detailed in earlier discussions. It should also be evident that the cross connections are actually made by way of the terminal jumper installed on the port.

In this configuration the connections for pins 1,2,3 and 7 are obvious, but the remaining connections could use a little explanation. The loop back connections between 4 and 5 are only to insure that everytime a request is made that there is a corresponding "clear to send" so transmission can take place. In this way the DCE signal "CTS" can be emulated and ensure the data can transfer will always be successful.

The connection for pin 20 will be routed by the jumper to pin 8. In this way DSR will be asserted any time the terminal is on-line. This, again, will emulate the interaction between DTR and DSR. Remember, that for any transfer to be accomplished DTR must be active from the terminal and recognized by the CONVEX port. For further illustration of this, see figure 1 below:

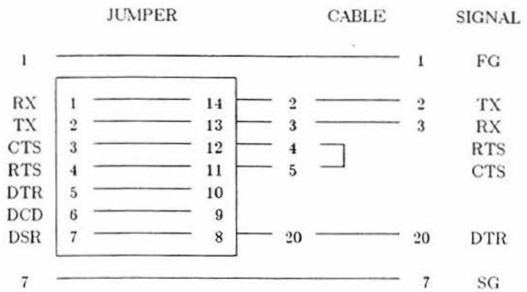


FIGURE 1

The next example will show the proper configuration for a modem and, again, will assume a terminal jumper installed.

...continued from previous page

CONVEX		MODEM	SIGNAL
1	————	1	FG
2	————	3	TX
3	————	2	RX
4	————	5	CTS
5	————	4	RTS
6	————	20	DTR
7	————	7	SG
8	————	8	DCD
20	————	6	DSR
22	————	22	RI

This configuration is a little more involved, as there are many more connections required for the modem to communicate with the CONVEX port.

First, remember the earlier discussion on the DTE to DCE interaction. This configuration is identical and will be for any standard modem. Remember that when using the modem jumper that this cable would be configured with all connections staight through. In other words, all pins on the system side would connect directly to the corresponding pin on the modem side.

Second, it must be understood that with the terminal jumper installed this configuration becomes a straight pin to pin layout as can be seen in figure 2 below:

	JUMPER			CABLE		SIGNAL
1				————	1	FG
RX	1	————	14	2 ————	3	RX
TX	2	————	13	3 ————	2	TX
CTS	3	————	12	4 ————	5	CTS
RTS	4	————	11	5 ————	4	RTS
DTR	5	————	10	6 ————	20	DTR
DCD	6	————	9	8 ————	8	DCD
DSR	7	————	8	20 ————	6	DSR
7				————	7	SG
22				————	22	RI

FIGURE 2

...continued from previous page

The operation here is somewhat simpler than previous discussions, because this is a full duplex connection. As discussed previously, full-duplex transmission is nothing more than simultaneous activity in both directions. This means that, unlike half-duplex, after the link is completed, the control lines will maintain their state throughout the connection. In other words, once RTS is active and CTS is granted then DCD and DSR will also remain active for the duration of the link. In addition DTR will be act ive anytime the system is on-line.

This makes things fairly simple to understand now, because the only lines that should show a change in level would be the transmit and receive lines on pin 2 and 3 respectively. This is due to the change between one and zero states of the data.

As previously discussed, there exists another means of manipulating the signals for a modem. This is by using the modem jumper on the particular port that the modem is attached. When using this modem jumper the system would be converted to a standard DTE configuration, allowing a common straight through modem cable to be used. This cable along with the jumper is diplayed below in figure 3.

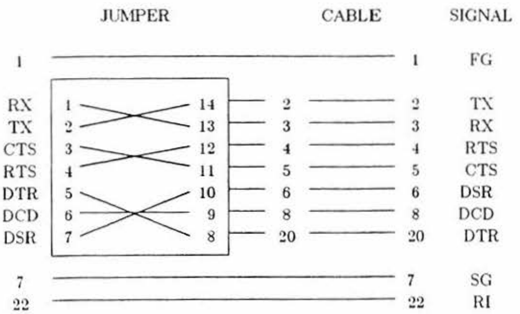


FIGURE 3

...continued from previous page

The final example will involve that of a printer to a serial port. This discussion will undertake to illustrate both flow control and hardware handshaking as would be faced with a specific situation. For the purpose of this discussion a Fujitsu DX2300 printer will be used.

The DX2300 is very simple in its operation and requires an equally simple cable configuration:

CONVEX		MODEM	SIGNAL
1	———	1	FG
2	———	2	TX
3	———	3	RX
4	┐ (looped to 5)		RTS
5	┘		CTS
7	———	7	SG
20	———	20	DTR

A communications protocol must be used with this printer to prevent a buffer overflow condition when print data is received faster than the printer can empty the print buffer. There a two (2) protocols available with the DX2300 and these are: XON/XOFF or READY/BUSY.

Using XON/XOFF protocol relies on the unix terminal driver to handle control characters form the external device. The DX2300 will send a DC3 control code to the system when the print buffer is full. The host system will then cease transmitting and wait for the printer to transmit a DC1 control code. This indicates that the buffer is empty and the printer is, again, ready to receive print data.

READY/BUSY protocol is the simple change of the DTR signal to indicate a buffer full condition. When the printer is unavailable the DTR signal is forced low indicating the printer is off line to the host system.

When the printer is first turned on a DC1 will be sent to the host, if in XON/XOFF mode, or DTR is true if in READY/BUSY mode.

This is a very simple example, but one which illustrates a couple of different methods of manipulating the printer. Other printers will use other methods and this can easily be determined by examination of the printer operation manual.

...continued from previous page

UNIX

What this means to unix is that for a command or program to open a port, DTR must be active. Otherwise the program will block in the open statement and therefore not gain control of the port.

When a logout or hangup is detected, unix sends SIGHUP to all processes and then drops DSR for 2 seconds except when there are backgound jobs running on the port. In this case the logout is executed but DSR is not dropped. If the port is attached to a port selector or modem which depends on DSR dropping a handler will have to be written that allows unix to ignore SIGHUP and does an IOCTL (DROPDTR);SLEEP ().

If a printer is being attached using printcap it will be necessary to read the tty (4) man page where the tty flags are defined. The printcap entries will allowthese flags to be set or unset on eac print job.

As can be seen from this discussion, RS232 is really a very simple and straight forward protocol for serial communication. All that is necessary is to determine the signals that each end of the connection need satisfied. After this is determined it will be a fairly simple task to make a cable cross-connection scheme that will satisfy both ends requirements.