Various Cabling and Techniques for:

Serial Data Analysis

Part 1

6 April 2010

Eric Hamilton

Contents

[1] `$Header: d:/Binder4/sda1/RCS/SDA1.tex,v 1.11 2008-06-24 14:00:25-07 Hamilton Exp Hamilton $`

List of Figures

List of Tables

This page intentionally left blank

1 General

1.1 Connecting a PV-130 to Breakout

The Pelco model PV-130 level changing unit requires that pin DB-9-8 (CTS, Clear To Send) be high before it will allow any data to go out of the unit on the DB-9 connector. Fortunately the PV-130 will use that pin to provide power. As a convinent way of providing this power, and eliminating the need for a seperate power supply, make a short cable wired as shown in Table 1, page 5. The DB-25 connector is not normally needed. In my configuration I have DB-9's comming from the Breakout[3] CPU so that I can swap things around easily. On my little sets of lights: RED = +, GREEN = − (MiniTracker). The configuration for monitoring communications data is shown in Figure 7, page 13.

DB-25 Breakout CPU	DB-9-F Breakout CPU	DP-9-M PV-130 End	Use
3	2	2	RD
2	3	3	TD
7	5	5	Ground
4, 5	7, 8	8	RTS jumpered to CTS and connected to CTS
6, 20	4, 6	—	DSR jumpered to DTR
$RCSfile: 130Pins.inc,v $			

Table 1: Enabling a PV-130 with Breakout or FTS data capture software

To make the control line go active type the following into the breakout CPU after each rebooting of the whole computer. (It seems to remember that someone had been writing on the port and leaves it active afterword.):

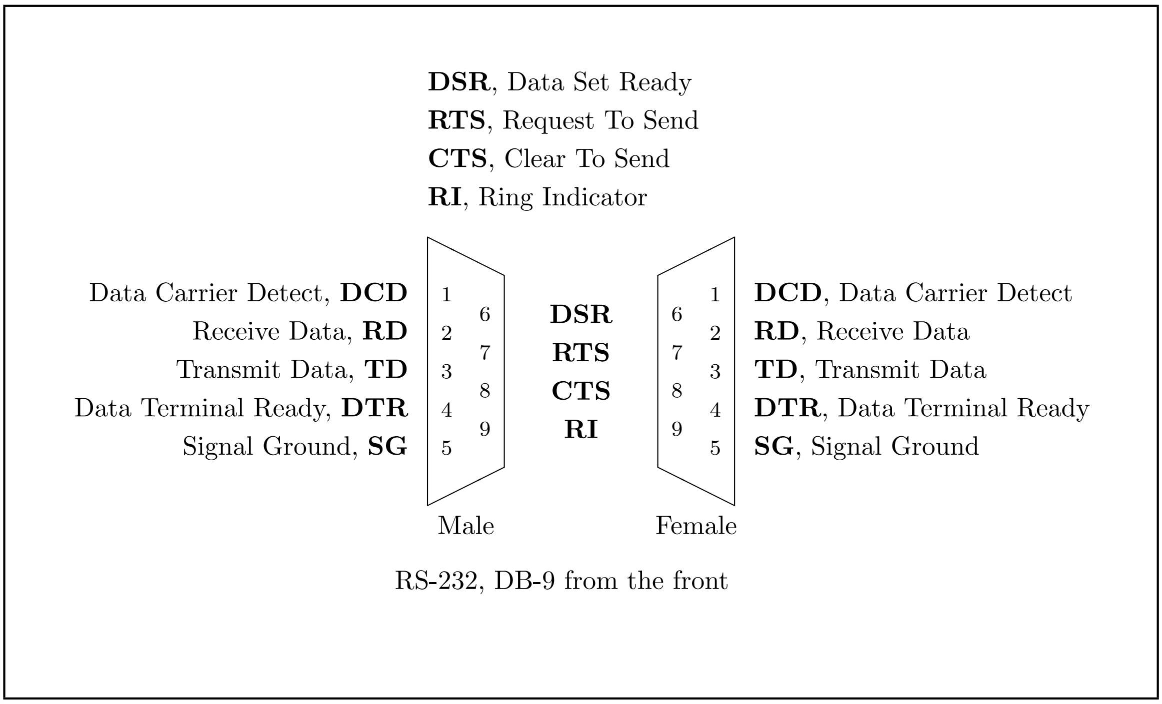
```
copy con con1
abc
ctl-z
```

(Each line is terminated with a CR.) "ctl-z" means control-Z. Repeat for con2 if it is to be used. If an error occurs, either connect the "magic" connector and it will go away, or abort the transfer. Either way the control line is activated. The older software Breakout utilized to two COM ports on a computer. The newer software FTS, uses a special PCMCIA Type II card which gives higher COM numbers.

[2] $Header: d:/Binder4/sda1/RCS/130Pins.inc,v 1.4 2008-04-28 12:25:45-07 Hamilton Exp Hamilton $

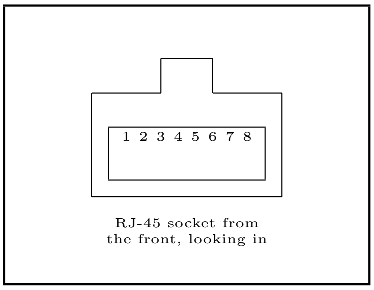
[3] The older software used at Pelco for data capture was named "Breakout". The newer software is called "SerialTest Async" from Frontline Test Systems (FTS). In this document there is no differentiation between the two sets of software. However Breakout never made the transition to Windows gracefully and the FTS software is used exclusively. The saved data formats of the two programs are not compatibale.

[4] $Header: d:/Binder4/sda1/RCS/RS2Piname.inc,v 1.1 2004-10-25 14:18:10-07 Hamilton Exp Hamilton $



$RCSfile: RS2Piname.inc,v $

Figure 1: RS-232 pins outs with names



$RCSfile: RjPins.inc,v $

Figure 2: RJ-45 pin assignments

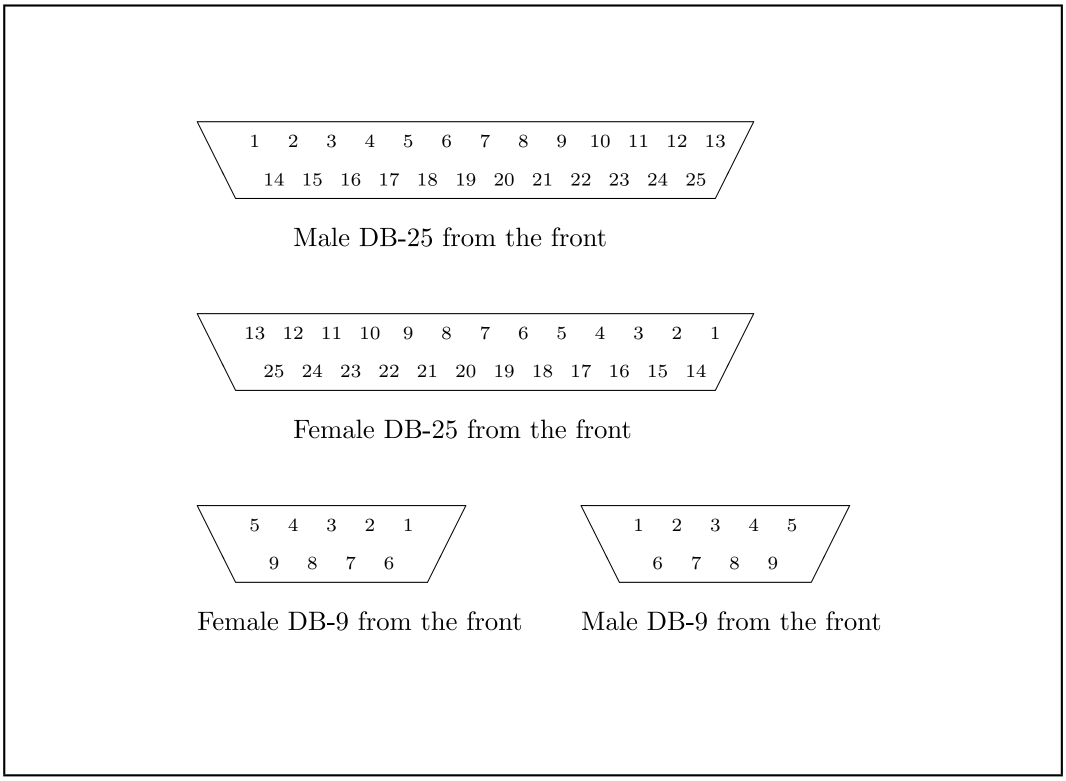
PV-130	
Old	New
TD(A)	TX-
TD(B)	TX+
RD(A)	RX-
RD(B)	RX+
GND	GND
+12V	12V
$RCSfile: 130Pins.inc,v $	

Table 2: PV-130 pin equivalences

PC				**Spectra**	
DB-25-F		**DB-9-F**		**RJ-45-F**	
Tx	2	Tx	3	2	Rx-
Rx	3	Rx	2	7	Tx-
Gnd	7	Gnd	5	8	Tx+
	7		5	1	Rx+
$RCSfile: 130Pins.inc,v $					

Table 3: Sales' idea of a non-PV-130 method of getting from RS-232 to/from RS-422

[5] `$Header: d:/Binder4/sda1/RCS/RjPins.inc,v 1.1 2004-10-25 12:54:53-07 Hamilton Exp Hamilton $`
[6] `$Header: d:/Binder4/sda1/RCS/RSpins.inc,v 1.1 2004-10-25 12:55:01-07 Hamilton Exp Hamilton $`



$RCSfile: RSpins.inc,v $

Figure 3: DB Connector pin outs

2 RS-232 Specific Information

2.1 DB-9 and DB-25 RS-232 Connectors

DB-9	**DB-25**	**Use**
1	8	DCD, Data Carrier Detect
2	3	RD, Recieved Data
3	2	TD, Transmitted Data
4	20	DTR, Data Terminal Ready
5	7	SG, Signal Ground
6	6	DSR, Data Set Ready
7	4	RTS, Request To Send
8	5	CTS, Clear To Send
9	22	RI, Ring Indicator
$RCSfile: Db9to25.inc,v $		

Table 4: DB-9 → DB-25 pin assignments

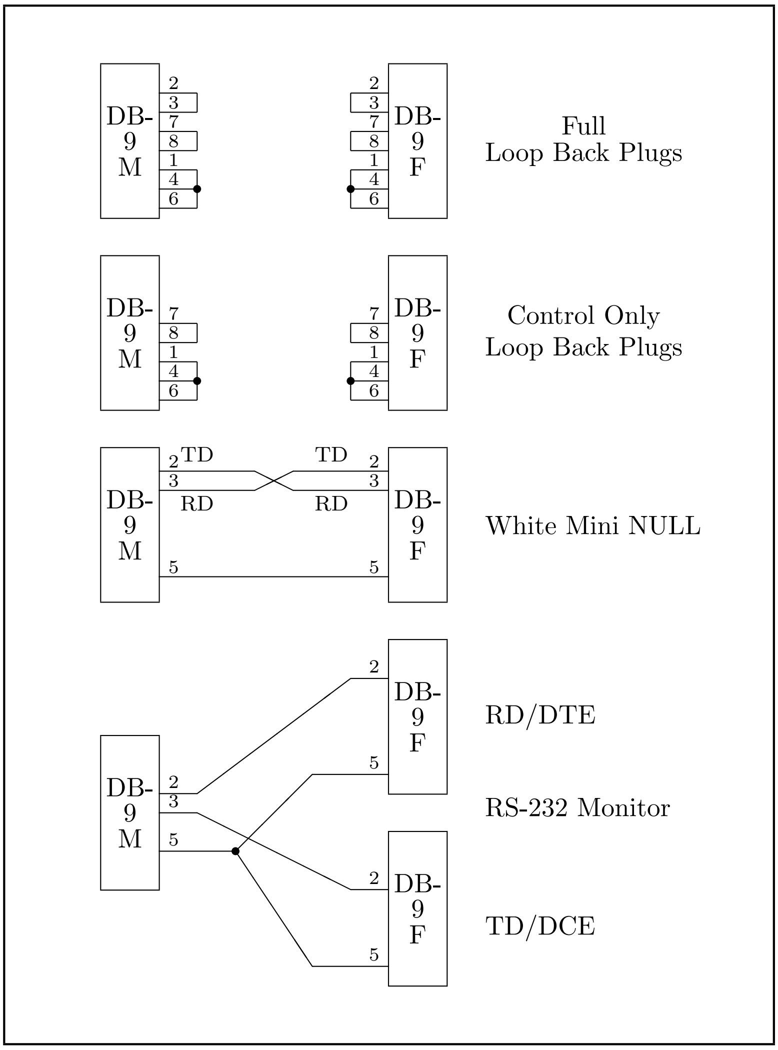
DB-25	**DB-9**	**Use**
2	3	TD, Transmitted Data
3	2	RD, Recieved Data
4	7	RTS, Request To Send
5	8	CTS, Clear To Send
6	6	DSR, Data Set Ready
7	5	SG, Signal Ground
8	1	DCD, Data Carrier Detect
20	4	DTR, Data Terminal Ready
22	9	RI, Ring Indicator
$RCSfile: Db9to25.inc,v $		

Table 5: DB-25 → DB-9 pin assignments

DB-25	DB-9	Use
2, 3	3, 2	TD jumpered to RD
4, 5	7, 8	RTS jumpered to CTS
6, 8, 20	6, 1, 4	DSR, DCD and DTR jumpered together
$RCSfile: Db9to25.inc,v $		

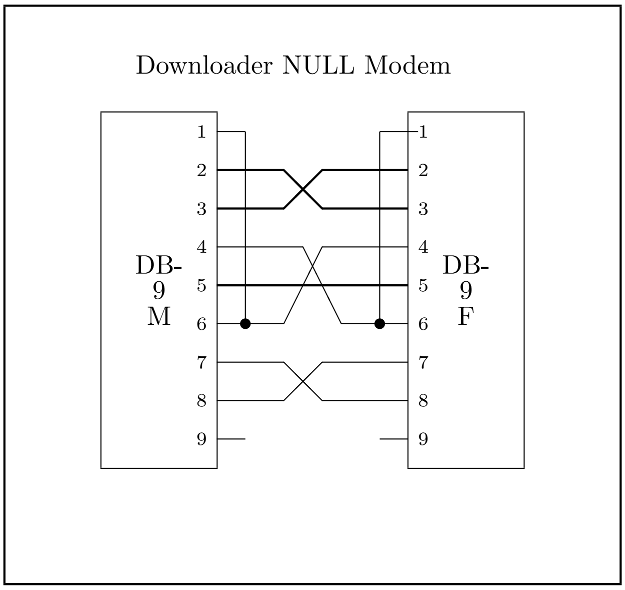
Table 6: Jumpering control signals to fool software

[7] `$Header: d:/Binder4/sda1/RCS/Db9to25.inc,v 1.5 2008-04-28 12:25:48-07 Hamilton Exp Hamilton $`
[8] `$Header: d:/Binder4/sda1/RCS/232Spec.inc,v 1.1 2008-03-25 10:02:40-08 Hamilton Exp Hamilton $`



$RCSfile: 232Spec.inc,v $

Figure 4: Special RS-232 Trick Cables



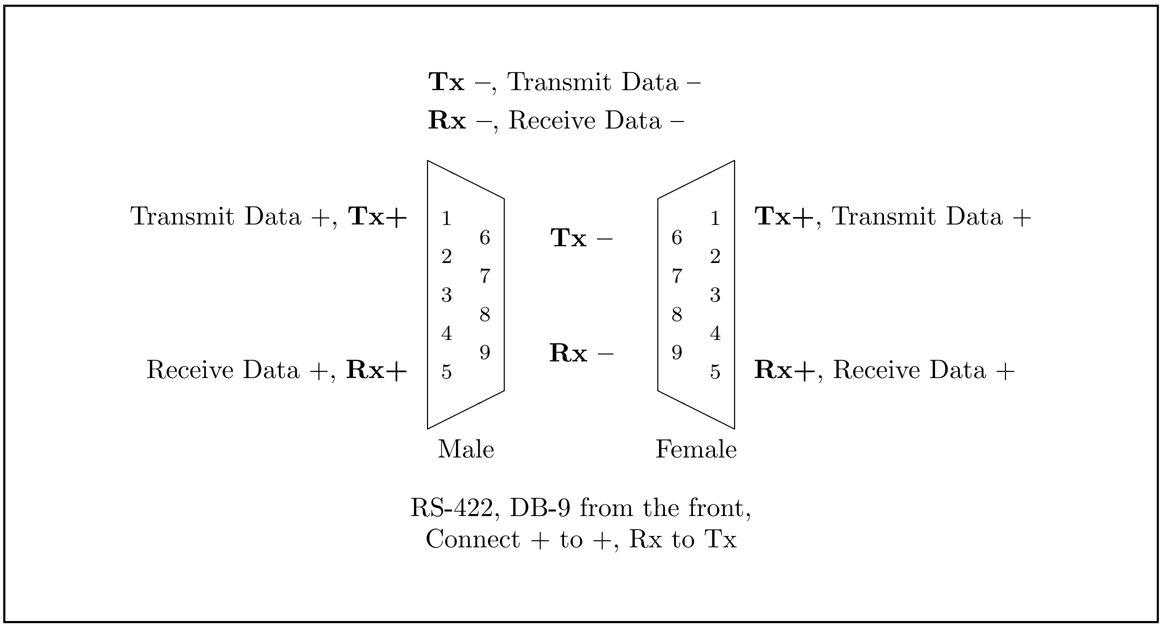
$RCSfile: dots.inc,v $

Figure 5: DB-9 Downloader (Full) Null Modem

3 RS-422/RS-485 Specific Information

RS422				
Esprit Wire Color	DB-9-M Camera		DB-9-F Matrix	
	Use	Pin	Pin	Use
Wh	R+	5	1	T+
Bk	R-	9	6	T-
Re	T+	1	5	R+
Gr	T-	6	9	R-
$RCSfile: 4Pins.inc,v $				

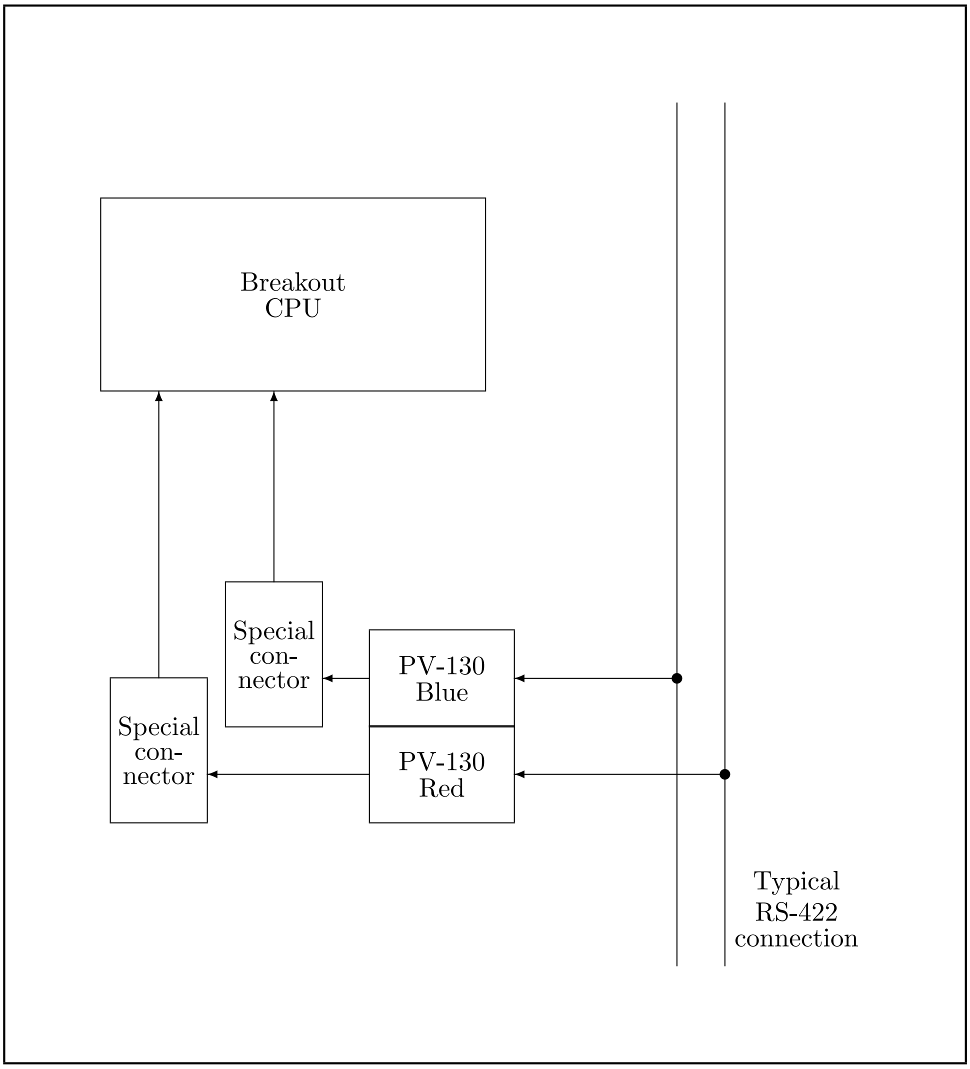
Table 7: Special four pin RS-422 cable DB-9 pinouts



$RCSfile: Rs4PiName.inc,v $

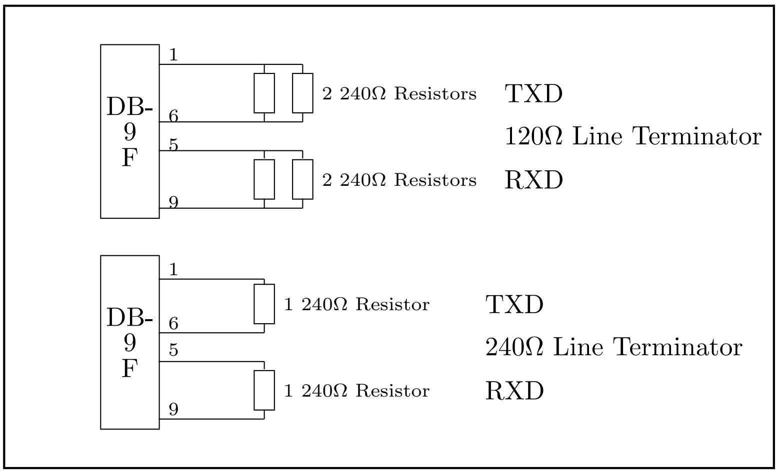
Figure 6: RS-422 pin outs, with names

[9] $Header: d:/Binder4/sda1/RCS/4Pins.inc,v 1.3 2009-11-10 07:12:05-08 Hamilton Exp Hamilton $
[10] $Header: d:/Binder4/sda1/RCS/Rs4PiName.inc,v 1.4 2008-03-17 14:01:08-08 Hamilton Exp Hamilton $
[11] $Header: d:/Binder4/sda1/RCS/MonCnfig.inc,v 1.1 2008-03-17 13:07:58-08 Hamilton Exp Hamilton $



$RCSfile: MonCnfig.inc,v $

Figure 7: Monitoring RS-422 communications with Breakout



$RCSfile: Term.inc,v $

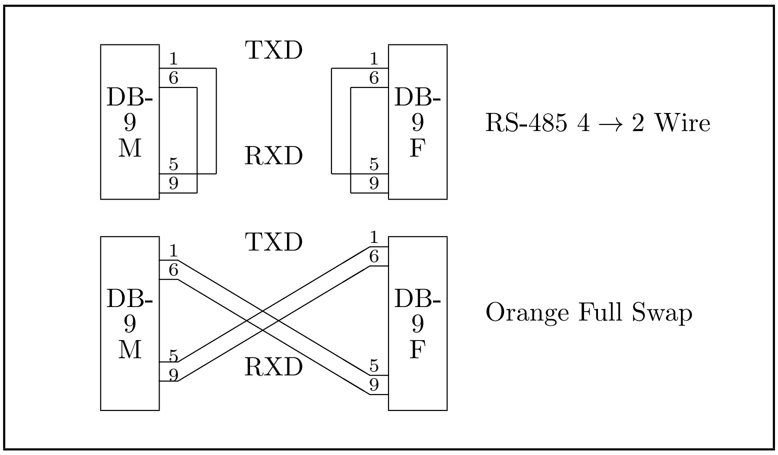
Figure 8: Termination Plugs, 120 and 240 Ω values

3.1 Special DB-9 Connector Tricks

One of the advantages of using the DB-9 connector system diagrammed in Figure 6, page 12, is that connections may be easily made and "problems" with the exact wiring may be easily identified and fixed. To do these "tricks" a set of specially wired Male to Female DB-9 connectors have been assembled. Each on has a specific purpose as outlined below:

1. **Orange/Red**. This special connector has only two wires which are installed between pins 5 and 9, leaving pins 1 and 6 open. This is used when there is a need to **Receive Only**. (Figure 10, page 16)
2. **Orange/Blue**. This special connector has only two wires which are installed between pins 1 and 6, leaving pins 5 and 9 open. This is used when there is a need to **Transmit Only**. (Figure 10, page 16)
3. **Blue**. (Also marked as "RX Flip" on white.) This special connector has a "normal" connection between pins 1 and 6, but reversed connections between pins 5 and 9. This is used to **Invert the Polarity of the Receive Signals**. Sometimes wires will be unexpectedly "flipped" when testing. This connector allows a quick and easy method of identifying the flipped pair, and is also useful for short duration testing without having to repair the wiring. (Figure 10, page 16)
4. **Red**. This special connector has a "normal" connection between pins 5 and 9, but reversed connections between pins 1 and 6. This is used to **Invert the Polarity of the Transmit Signals**. Sometimes wires will be unexpectedly "flipped" when testing. This connector allows a quick and easy method of identifying the flipped pair, and is also useful for short duration testing without having to repair the wiring. (Figure 10, page 16)
5. **Red/Blue**. (Also marked as "Full Flip" on white.) This special connector has "flipped" connection between pins 5 and 9, and between pins 1 and 6. This is used to **Invert the Polarity of the Transmit and Receive Signals**. Sometimes wires will be unexpectedly "flipped" when testing. This connector allows a quick and easy method of identifying the flipped pair, and is also useful for short duration testing without having to repair the wiring. (Figure 10, page 16)

[12] `$Header: d:/Binder4/sda1/RCS/Term.inc,v 1.1 2008-03-18 09:55:43-08 Hamilton Exp Hamilton $`
[13] `$Header: d:/Binder4/sda1/RCS/Db9Spec.inc,v 1.3 2008-03-17 13:09:49-08 Hamilton Exp Hamilton $`



$RCSfile: Special.inc,v $

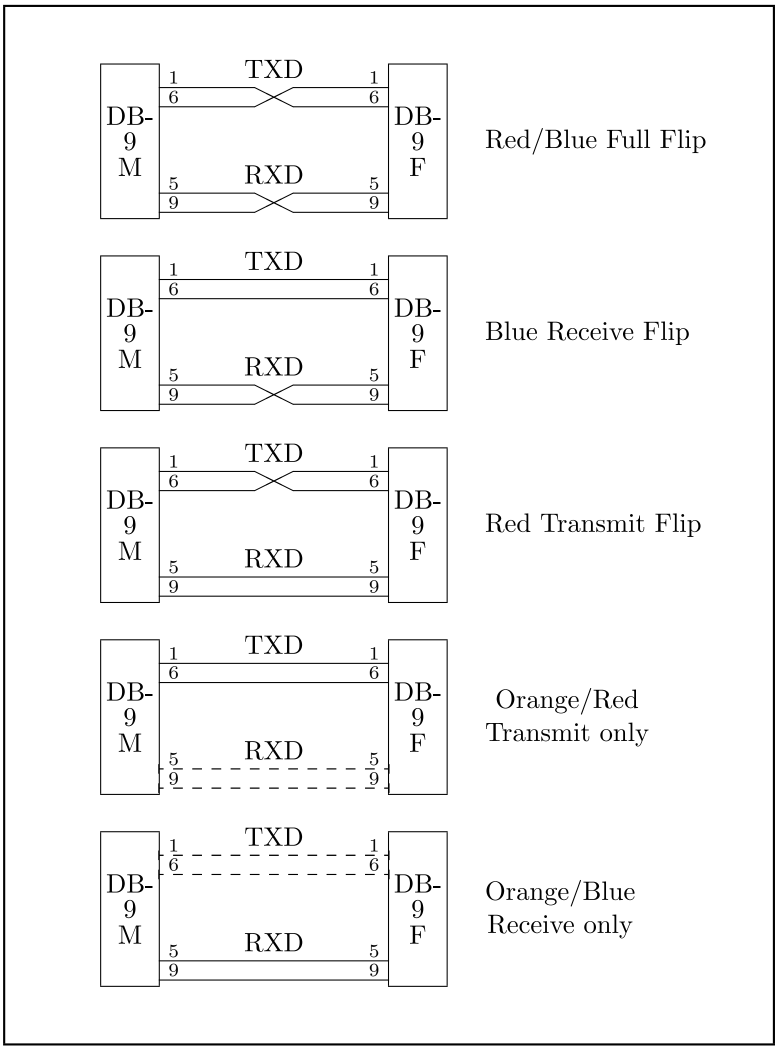
Figure 9: Special Trick Cables

6. **Orange marked TX-RX Swap**. This connector connects pins 1 and 9 to pins 5 and 6, and 5 and 6 to pins 1 and 9. This gives **a full "swap" of receive and transmit lines.** Useful when the wiring has started with an unknown configuration and we find that there is an effort being to have both ends of the line transmit into each other and try to receive from the other receiver. (Figure 9, page 15)

7. **"unmarked"**. There is a pair (one male, one female) of connectors that have pin 1 connected to pin 5 and pin 6 to pin 9. These are used with an "anything cable" to **force/test an RS-485 connection.** (Figure 9, page 15)

8. Several **connectors with LEDs** on them. It is very nice to be able and detect if "anything" is happening on the communications line without having other test equipment connected. When the LEDs flash it is obvious that "something" is going on. The better ones have two LEDs per pair (pins 1/6 and 5/9) so that they turn on readily. LEDs turn on faster than the turn off (or is this related to the persistancy of vision) and a two color LED does not give out as much information as a pair of LEDs do. The exact configuration of the LEDs is not important, other than the polarity be the same for each color on the same connector.

[14] `$Header: d:/Binder4/sda1/RCS/Special.inc,v 1.3 2008-03-17 14:01:08-08 Hamilton Exp Hamilton $`
[15] `$Header: d:/Binder4/sda1/RCS/422Pins.inc,v 1.4 2008-04-28 12:25:46-07 Hamilton Exp Hamilton $`



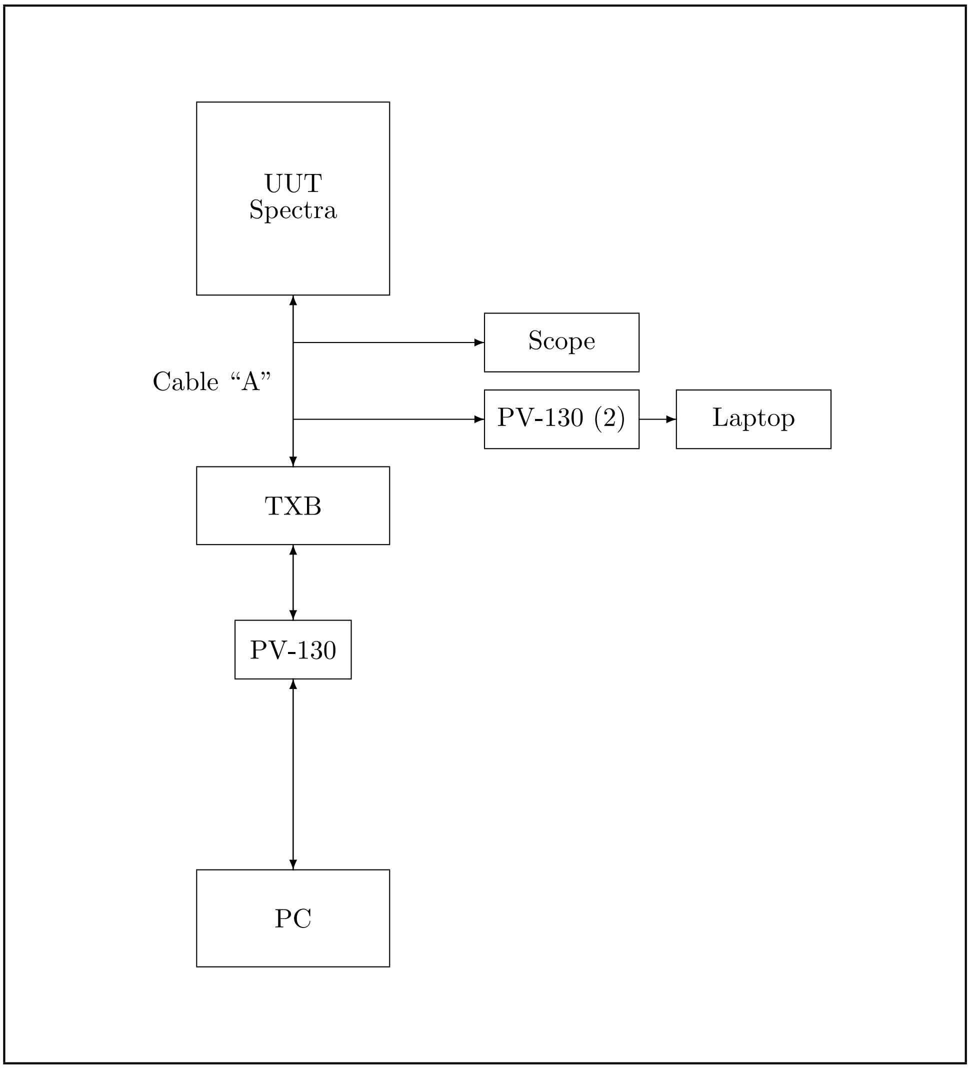
$RCSfile: Special.inc,v $

Figure 10: Trick Red, Blue and Red/Blue Cables

Color/-Marking	Pinning	Use
BL	5, 9 flipped	From keyboard/computer/matrix
RE	1, 6 flipped	From Spectra
WH Rx	1, 6 flipped	From Spectra
RE, BL	5, 9 and 1, 6 flipped	Flip both receive and transmit
WH Full	5, 9 and 1, 6 flipped	Flip both receive and transmit
OR	1, 5 and 6, 9 swapped	Swap transmit and receive pairs
LB	1—6, 5—9	Loop back connector
LEDs	1, 6 and 5, 9	LED. "Long" on 6, 9. "Short" on 1, 5.
$RCSfile: 422Pins.inc,v $		

Table 8: Misc RS-422 connector wirings

3.1.1 Test Configuration A

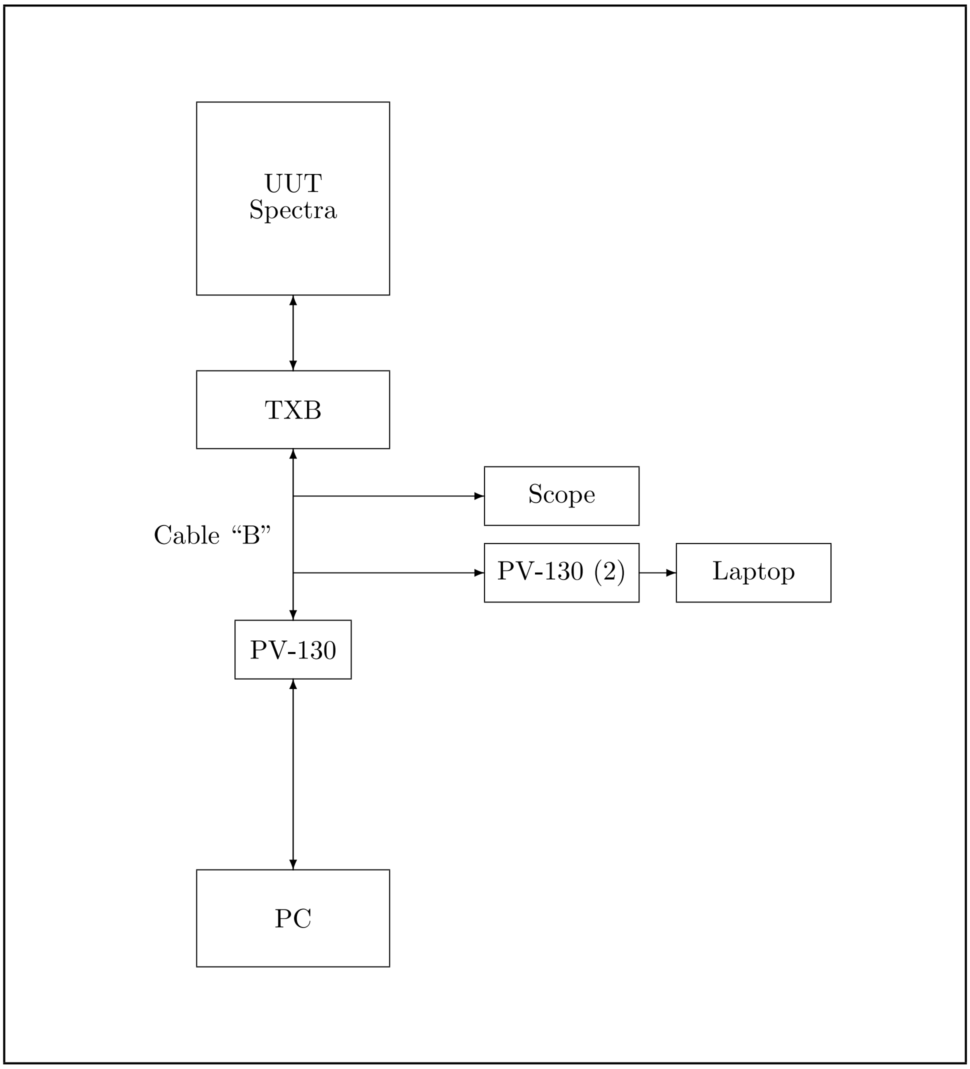


$RCSfile: ConfigA.inc,v $

Figure 11: TXB test configuration “A”, data capture between a TXB and a UUT

[16] $Header: d:/Binder4/sda1/RCS/ConfigA.inc,v 1.6 2008-03-18 09:08:37-08 Hamilton Exp Hamilton $

3.1.2 Test Configuration B

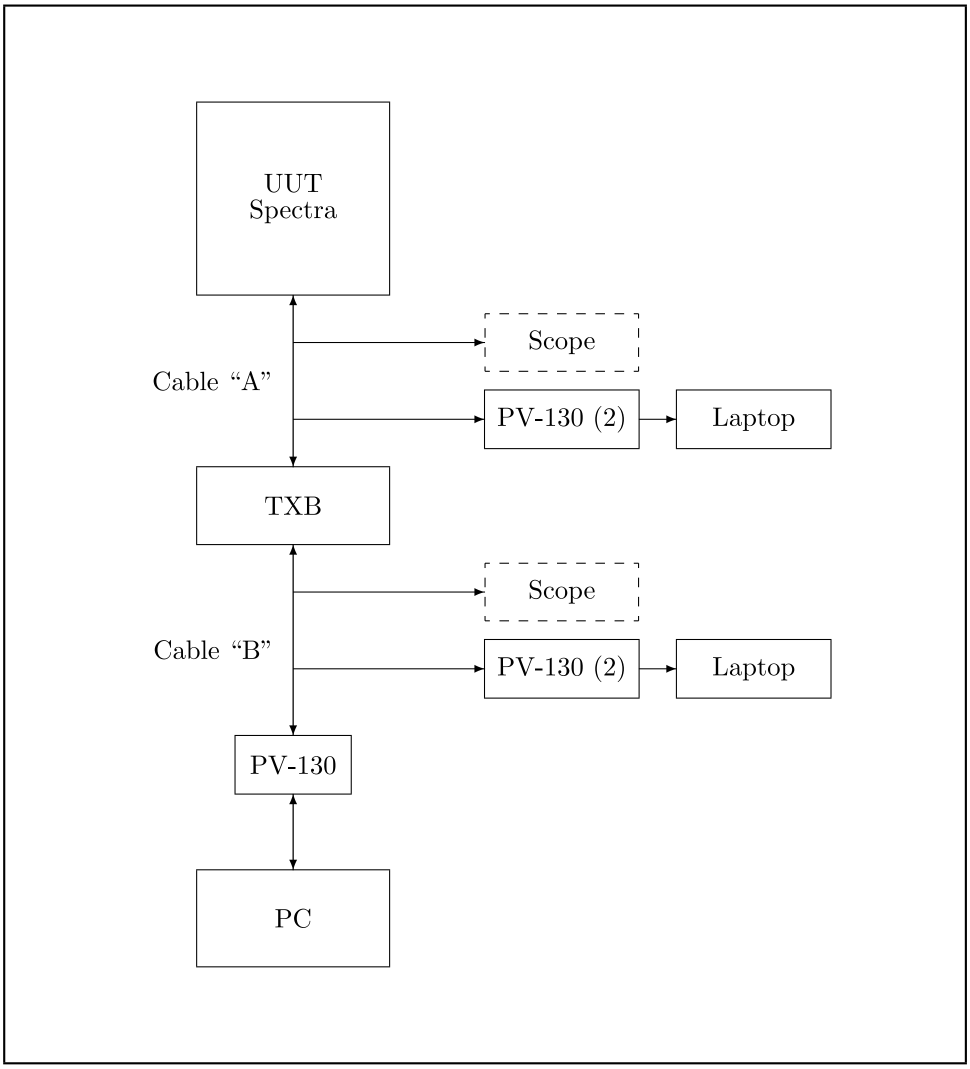


$RCSfile: ConfigB.inc,v $

Figure 12: TXB test configuration "B", data capture between a head end and a TXB

[17] `$Header: d:/Binder4/sda1/RCS/ConfigB.inc,v 1.6 2008-03-18 09:08:37-08 Hamilton Exp Hamilton $`

3.1.3 Test Configuration C



$RCSfile: ConfigC.inc,v $

Figure 13: TXB test configuration "C", data capture between a TXB and a UUT with two capture computers

[18] `$Header: d:/Binder4/sda1/RCS/ConfigC.inc,v 1.3 2008-03-18 09:08:37-08 Hamilton Exp Hamilton $`

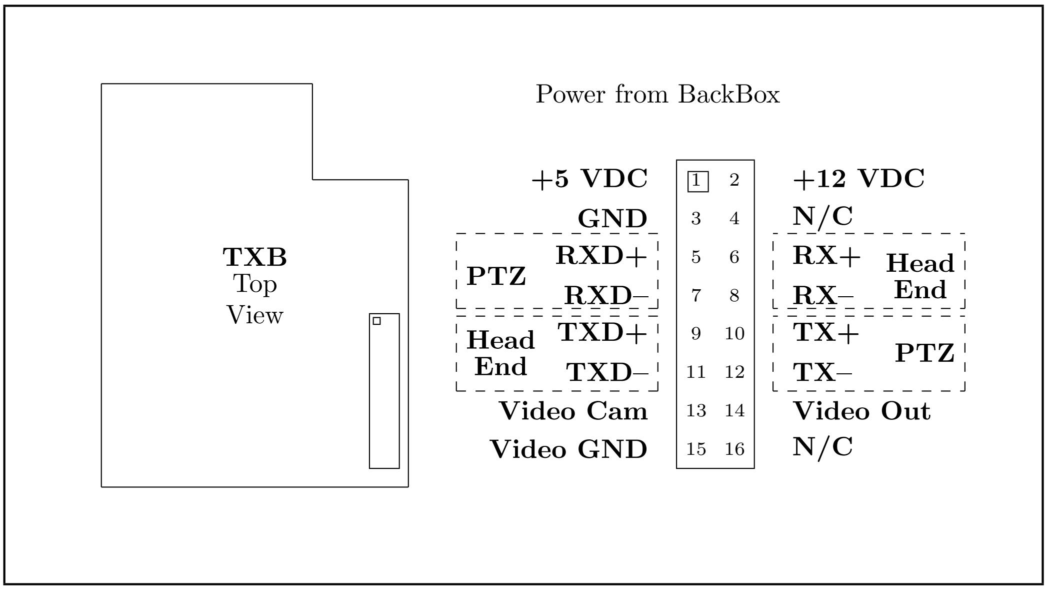
4 TXB Specific Information

J2 Spectra/Esprit connection	
Pin	Use
1	+5 VDC, From an Esprit, open with a Spectra
2	+12 VDC, From a Spectra, open with an Esprit
3	Ground
4	Open on a Spectra, unknown on an Esprit
5	RXD+, To Spectra
6	RX+, From host
7	RXD-, To Spectra
8	RX-, From host
9	TXD+, From Spectra
10	TX-, To host
11	TXD-, From Spectra
12	TX-, To host
13	Video in
14	Video out
15	Ground
16	Open on a Spectra, unknown on an Esprit
The shorting plug that is normally installed in J2 connects the following pins together: 3-4, 5-6, 7-8, 9-10, 11-12, 13-14. And leaves the following pins open: 1, 2, 15, 16.	
$RCSfile: TxbPins.inc,v $	

Table 9: TXB Connector pin assignments on Spectra and Esprit units

Cable	Color	16-pin	Use	DB-9
A	Blue	12	TXB −	6
	Green	10	TXB +	1
	Black	8	RXD −	9
	Red	6	RXD +	5
B	Blue	11	TXB −	6
	Green	9	TXB +	1
	Black	7	RXD −	9
	Red	5	RXD +	5
$RCSfile: CablWire.inc,v $				

Table 10: Cable A & B wiring



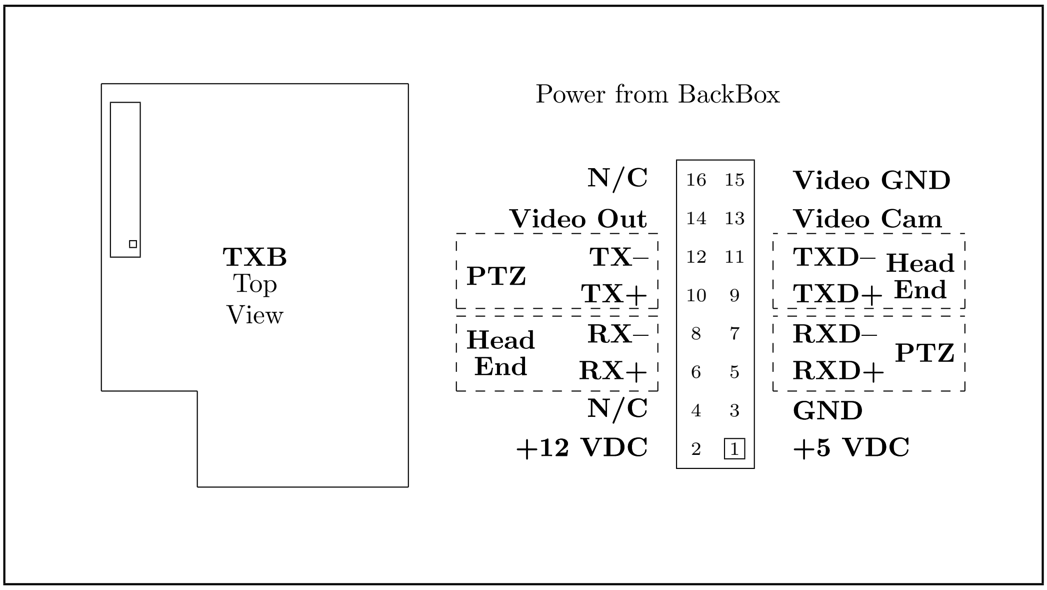
$RCSfile: 16Pins.inc,v $

Figure 14: TXB pin outs

TXB Name		Pin	Spectra Name	
Direction	Name	#	Name	Connector
—	+5V	1	Open	—
—	+12V	2	+12V	—
—	Gnd	3	Gnd	—
—	Open	4	Open	—
To Camera	RXD+	5	RXD+	J6-6
From Control	RX+	6	RX+	P4-4
To Camera	RXD–	7	RXD–	J6-7
From Control	RX–	8	RX–	P4-3
From Control	TDX+	9	TXD+	J6-4
To Camera	TX+	10	TX+	P4-1
From Control	TXD–	11	TXD–	J6-5
To Camera	TX–	12	TX–	P4-2
—	Video	13	Video Cam	J7-10
—	Video	14	Video Out	J1
—	Gnd	15	Gnd	—
—	Open	16	Open	—
$RCSfile: 16Pins.inc,v $				

Table 11: TXB RS-422 signal names *vs.* Spectra signal names, from SH11-0006-00B0 and SH05-0007-00F

[19] `$Header: d:/Binder4/sda1/RCS/16Pins.inc,v 1.6 2008-06-16 08:14:58-07 Hamilton Exp Hamilton $`
[20] `$Header: d:/Binder4/sda1/RCS/16PinsUd.inc,v 1.1 2008-06-16 08:14:18-07 Hamilton Exp Hamilton $`



$RCSfile: 16PinsUd.inc,v $

Figure 15: TXB pin outs upside down

TXB Name		Pin	Spectra Name	
Direction	Name	#	Name	Connector
—	Open	16	Open	—
—	Gnd	15	Gnd	—
—	Video	14	Video Out	J1
—	Video	13	Video Cam	J7-10
To Camera	TX–	12	TX–	P4-2
From Control	TXD–	11	TXD–	J6-5
To Camera	TX+	10	TX+	P4-1
From Control	TDX+	9	TXD+	J6-4
From Control	RX–	8	RX–	P4-3
To Camera	RXD–	7	RXD–	J6-7
From Control	RX+	6	RX+	P4-4
To Camera	RXD+	5	RXD+	J6-6
—	Open	4	Open	—
—	Gnd	3	Gnd	—
—	+12V	2	+12V	—
—	+5V	1	Open	—
$RCSfile: 16PinsUd.inc,v $				

Table 12: TXB RS-422 signal names *vs.* Spectra signal names, from SH11-0006-00B0 and SH05-0007-00F, upside down

4.1 Test Configurations

4.1.1 Cable Wiring

	16-pin			Use
	1			+5 VDC, Esprit
		2		+12 VDC, Spectra
	3			Ground
		4		No Connection
A	5			RXD + to dome
		6	**B**	RX + to head end
A	7			RXD − to dome
		8	**B**	RX − to head end
A	9			TXD + to dome
		10	**B**	TX + to head end
A	11			TXD − to dome
		12	**B**	TX − to head end
	13			Video in, from dome
		14		Video out, to head end, Coaxitron in from head end
	15			Video ground
		16		No Connection
$RCSfile: TestCabl.inc,v $				

Table 13: TXB socket wiring on the Spectra

[21] $Header: d:/Binder4/sda1/RCS/TestCabl.inc,v 1.6 2008-04-28 12:29:42-07 Hamilton Exp Hamilton $

4.1.2 Viewing Data Into and Out of a TXB

The data capture software may only view two directions of communications. The two cables described above, **A** and **B**, will either allow viewing of the data between the TXB and the Pelco PTZ (**A**, Figure 11, page 18) or between the TXB and the head end (**B**, Figure 12, page 19).

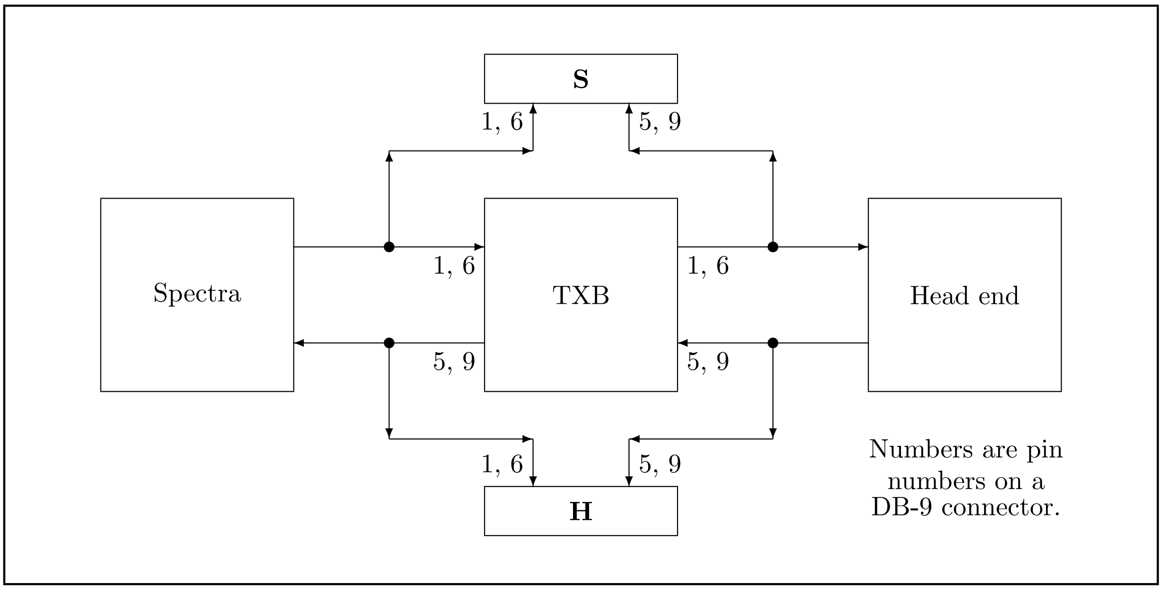
There are times when it is important to know exactly how each input to the TXB[23] is translated and sent out. There is also the reverse problem of knowing how Pelco PTZ replies are translated that come from the PTZ and return to the head end. There are two solutions to this problem:

1. Have two data capture computers with data capture software running on them (Test configuration **C**, Figure 13, page 20).

2. Make a set of special cables:

    2.1 Make a special cable that will allow Head end transmit and TXB transmit data to be viewed simultaneously. This cable will be called the type "**H**" cable.

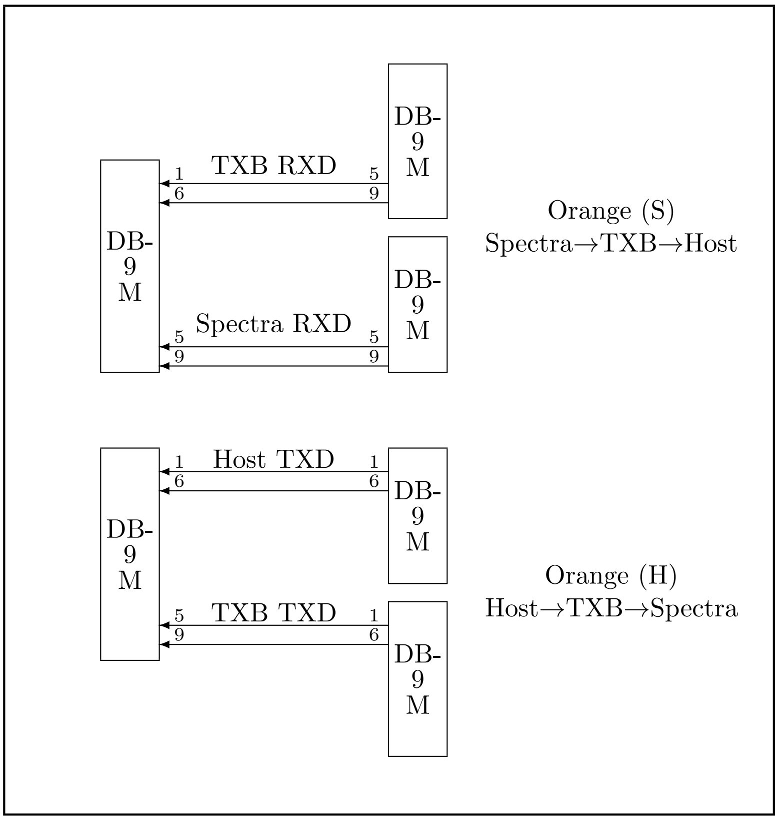
    2.2 Make a special cable that will allow Pelco PTZ equipment transmit and TXB transmit data to be viewed simultaneously. This cable will be called the type "**S**", for Spectra, cable.



$RCSfile: SHCables.inc,v $

Figure 16: Type S and H cables

[22] $Header: d:/Binder4/sda1/RCS/InOuTxb.inc,v 1.2 2008-03-18 09:08:38-08 Hamilton Exp Hamilton $
[23] Also what the exact timing relationships are.
[24] $Header: d:/Binder4/sda1/RCS/SHCables.inc,v 1.3 2008-02-26 13:32:28-08 Hamilton Exp Hamilton $
[25] $Header: d:/Binder4/sda1/RCS/SHWire.inc,v 1.2 2008-03-18 09:08:39-08 Hamilton Exp Hamilton $



$RCSfile: SHWire.inc,v $

Figure 17: Orange S and H Cables

5 Equipment Setups

5.1 Running FTS Software

5.1.1 How to save data with FTS

Utilizing the asynchronous data capture program from FrontLine Test Systems' named "Serialtest Async" requires that it be set up as follows to capture D Protocol data in RS-422 four wire mode at 9600 baud:

This applies to Version 5.1.1.0 of Serialtest Async.

1. `Configuration: Monitor Both 9600,N,8,1 9600,N,8,1 DTE=COM4, DCE=COM5`.

   Configuration parameters must match the equipment being tested/monitored.

2. Under the "`System Settings...`" (Figure 18, page 28) entry of the "`Optons`" pull down menu, be sure to select:

   2.1 Do not check: "`Automatically Save Inported Capture Files in FTS Format`"

   2.2 Do not check: "`Automatically Restart Capturing After 'Clear Capture Buffer' or 'Close Capture File'`"

   2.3 Select a "`Capture Buffer Size (in K)`" of at least `35048`.

   2.4 Select a "`File Size (in K)`" of at least `35048`.

   2.5 "`Capture to this series of files`" with "`Wrap FIles`" being unchecked.

   2.6 "`Name`" "`Capture`"

   2.7 "`Maximum number of files`" "`999`"

   2.8 '`Start new file when the file is full`" Not checked.

3. "`Start up..`" always terminates with a fault, don't waist your time trying to get anything done here.

4. On the "`Advanced...`" sub menu set:

   4.1 Set all ten entries to the maximum value that they will allow. For most of them the maximum value is the same as the default value.

5. For the hardware setup screen (Figure 19, page 29) select:

   5.1 Select "Use FTS Cables"

   5.2 Use this read device: "COM4"

   5.3 and this send device: "COM5"

   5.4 UART FIFO Settings: Select "Use Optimal Settings for Serial Data Analysis"

To actually collect data:

1. Select the "`Go Live`" option under the "`File`" pull down item.

   1.1 Make sure that the baud rate, etc is correct in the sub-menu "`Window`" sub-sub menu item "`Set I/O Configraton`" (Figure 20, page 29).

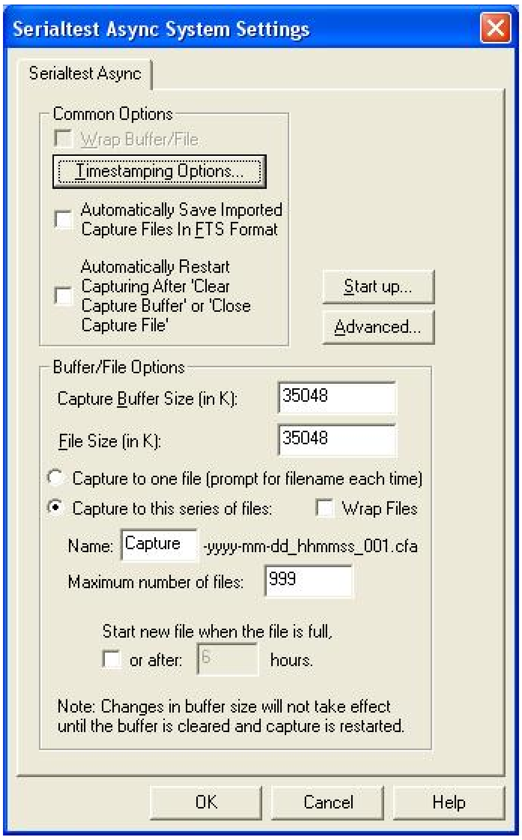


Figure 18: FTS System settings

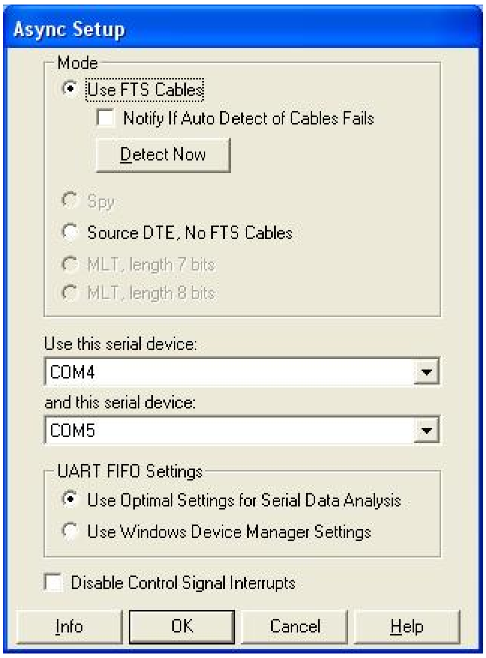


Figure 19: FTS Hardware setup

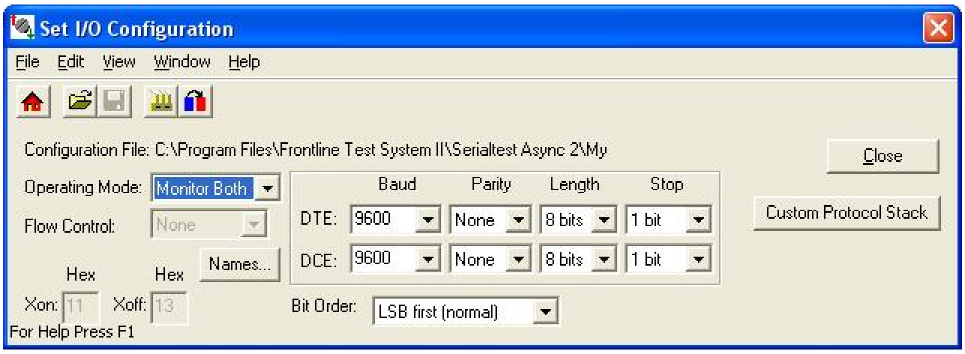


Figure 20: FTS Selections for commnications parameters

2. In the "`Export Capture Buffer`" sub-menu (Figure 21, page 31) select to have, **AND THIS IS CRITICAL** in the "`Displayed Fields`" option box:

    2.1 "`Side`"

    2.2 "`Event Number`"

    2.3 "`Timestamp`"

    2.4 "`Hexadecimal`"

    The order of these fields is essential for the data analysis programs to operate correctly.

3. In the "`As`" window enter what seems to be a unique name. I try to enter the day's date with a single letter as a suffex to indicate which capture of the day it is. It the capture file will end up in: "`\Program Files\Frontline Test System II\Serialtest Async 2\My Capture Files`". These files may be accessed with typing in "`x x`" at the DOS prompt. At this point typing in a third "`X`" will copy all serial capture files to a thumb drive on drive `D:` and move them into a sub-directory called "`CaptureLogs`".

4. For the rest of the items, use these values

    4.1 `Export Events...`

    4.2 `Apply Template: ExportFormat`

    4.3 `Displayed Fields: Side, Event Number, Timestamp, Hexadecimal`

    4.4 `Text Output`

    4.5 `Separate Records With CR/LF`

    4.6 `Align Columns`

    4.7 `Output Header`

    4.8 `Output Field Name Record`

    4.9 `Align Field Names With Data`

    4.10 `Timestamp Format: Native`

    4.11 `Character Set: ASCII`

    4.12 `Signals Characters: 1/0`

    4.13 `Errors Characters: X/<sp>`

    4.14 `Field Delimiter: Space`

    4.15 `Filter Out: Neither Side`

    4.16 `Export: Entire Buffer`

A typical capture's screen should show something similar to the "Event screen" (Figure 22, page 32). This screen is showing a normal D Protocol data capture between the GlassKeyboard and a Spectra IV.

[26]`$Header: d:/Binder4/sda1/RCS/FtsSave.inc,v 1.6 2008-03-18 09:08:38-08 Hamilton Exp Hamilton $`
[27]`$Header: d:/Binder4/sda1/RCS/FtsSave.inc,v 1.6 2008-03-18 09:08:38-08 Hamilton Exp Hamilton $`

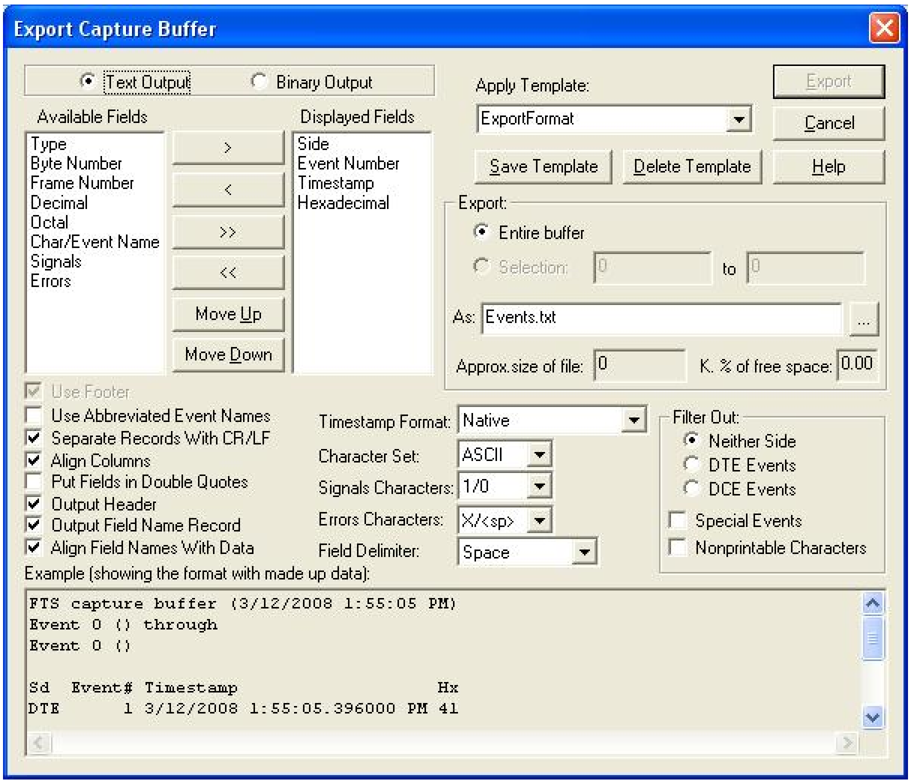


Figure 21: FTS Export screen

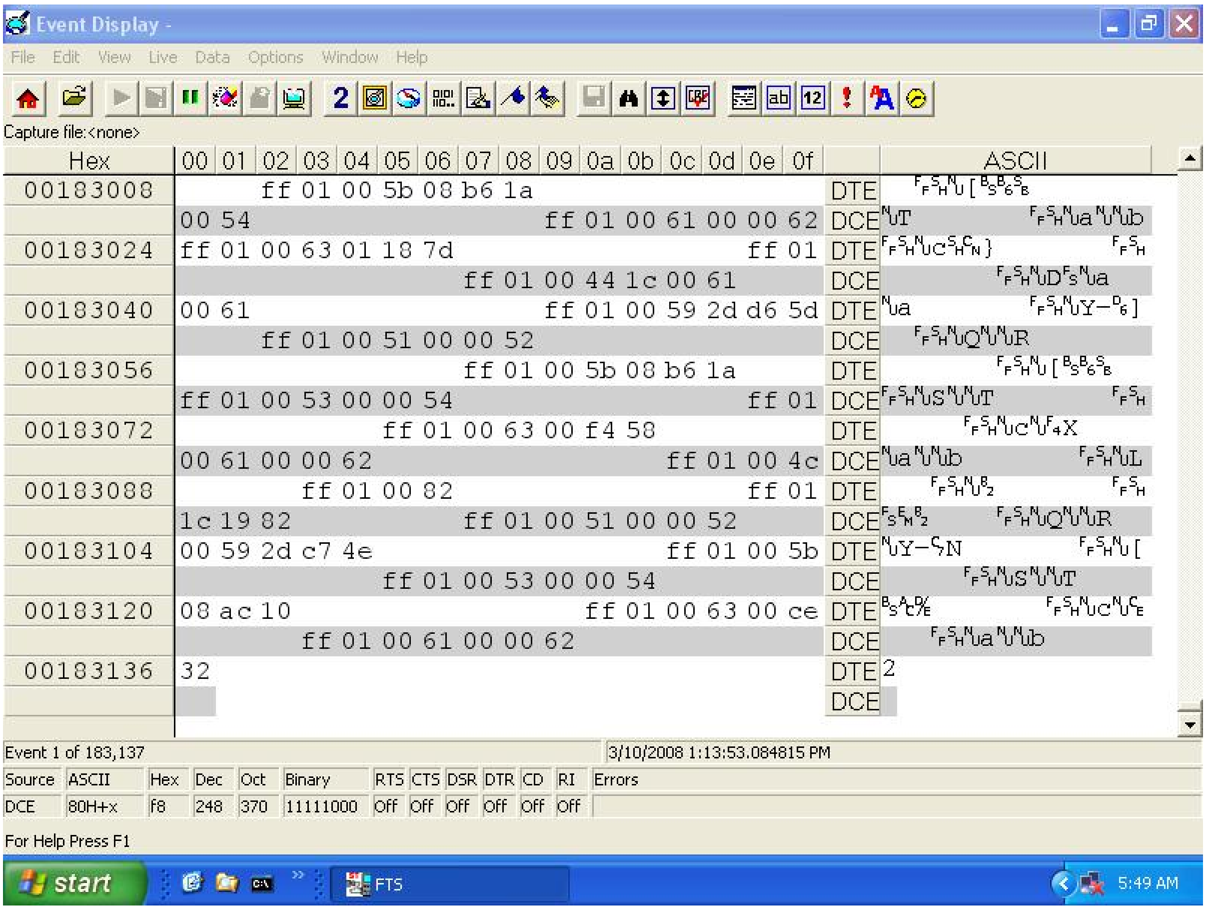


Figure 22: FTS Event screen

5.1.2 Common Data Collection Problems

When using the FTS software to collect serial data there are a couple of common problems that are encountered. These are:

1. Starting to collect data with the head end baud rate incorrect and not matching the Spectra's baud rate. The system has independant send and receive rates. Usually they are set to the same value. When both values of the FTS software are incorrect the display shown in Figure 23, page 34 will result. Note that there is data indicated as going in only one direction and that none of it "makes any sense". (Data shown in red is invalid in format.)

   Possiable data format errors include:

   1.1 Incorrect byte length. FTS software supports: 5, 6, 7 and 8 bit long bytes.

   1.2 Incorrect parity. FTS software supports: None, Odd, Even, Mark, Space and Ignore parity types.

   1.3 Incorrect baud rate. FTS software supports: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 57600 and 115200 as baud rates.

   1.4 Incorrect number of stop bits. FTS software supports: 1 or 2 stop bits. However it is very unlikley that this will cause and error.

   1.5 Incorrect order of transmitted bits. FTS software supports: LSB first (default) and MSB first.

   1.6 Physical problems with the data include:

   1.6.1. Framing errors, Usually caused by an incorrect baud rate.

   1.6.2. Parity errors, Usually caused by having an incorrect parity type selected or using an incorrect baud rate.

2. Starting to collect data when one of the baud rates is incorrect. This is shown in Figure 24, page 35. In this example the DTE baud rate is correct and the DCE baud rate is incorrect (note the red characters) in this data capture of D Protocol command/replies.

3. In Figure 25, page 36 the screen is showing normal data from a P Protocol data capture. Most of the Pelco protocol data captures will be in D Protocol, note the differences in the data format. The most common place where unexpected data formatted occurs is when working with foriegn protocols.

[28] `$Header: d:/Binder4/sda1/RCS/Problems.inc,v 1.1 2008-03-18 07:59:47-08 Hamilton Exp Hamilton $`

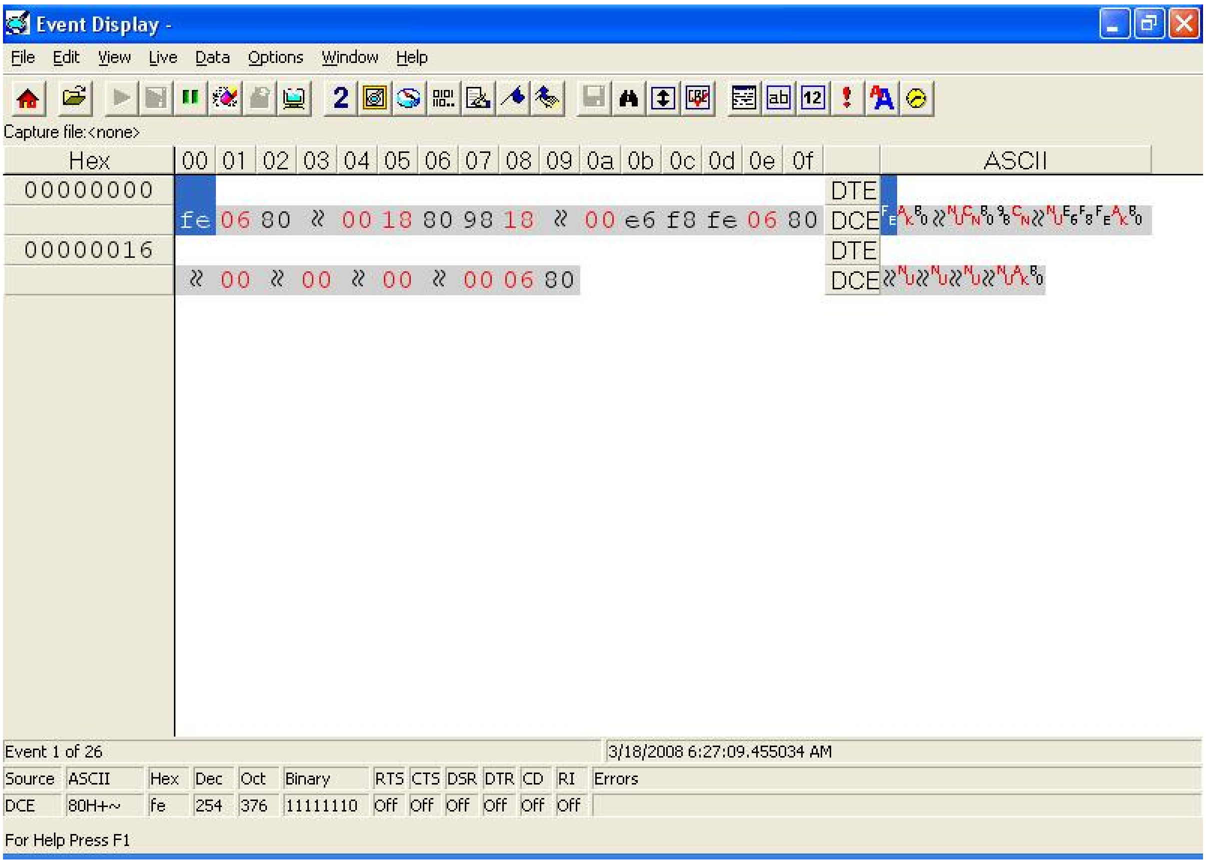


Figure 23: FTS Event screen with both baud rates incorrect

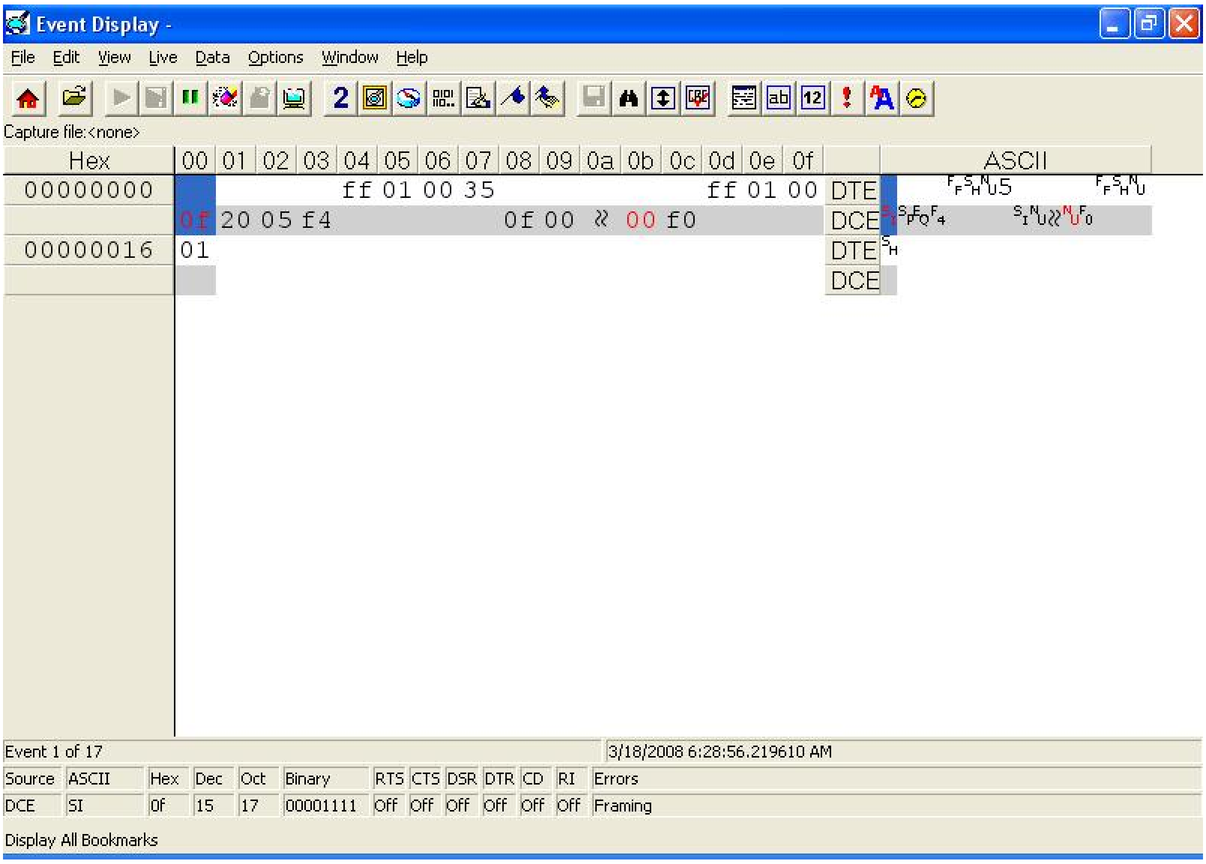


Figure 24: FTS Event screen with one baud rate incorrect

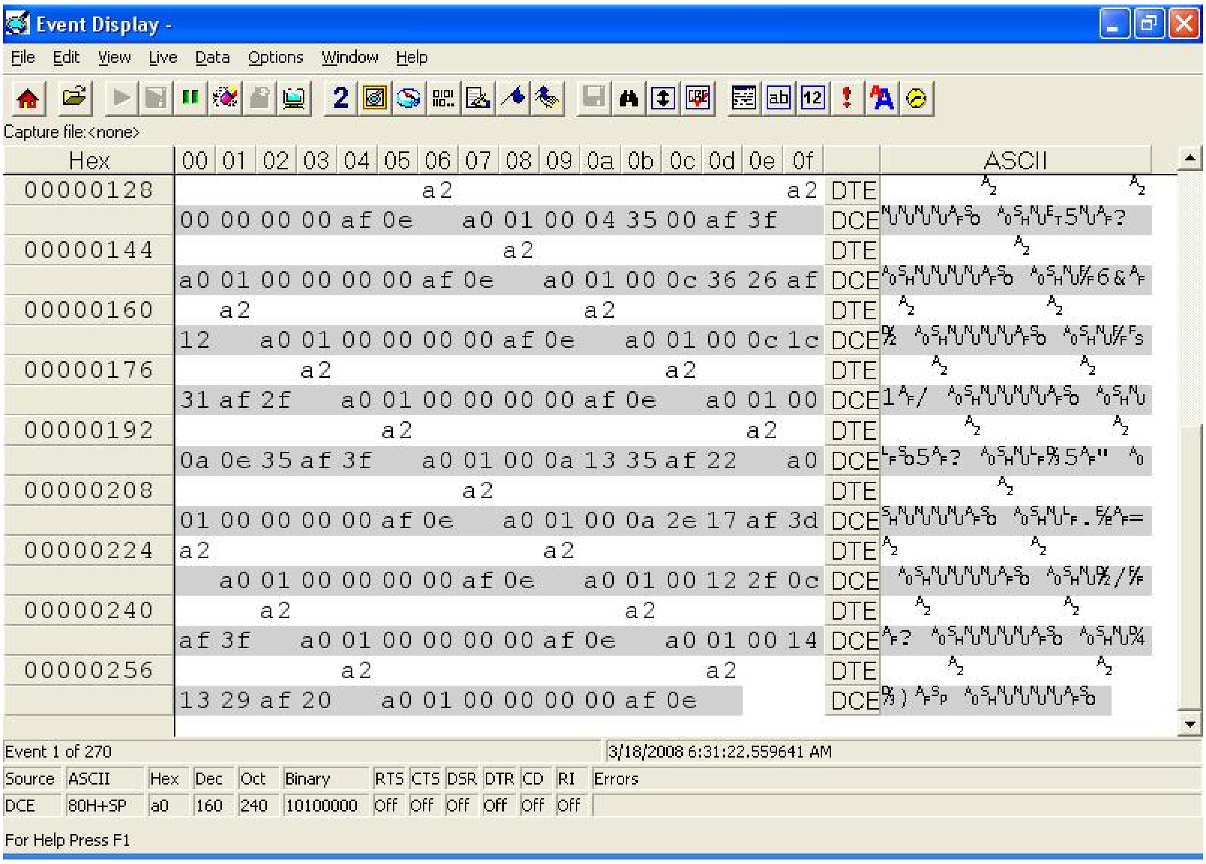


Figure 25: FTS Event screen with P Protocol move commands

5.2 Saving Oscilloscope Pictures

With a Tektronix TDS220 (and other similar Tektronix Oscilloscopes) configure the printer port as follows:

1. Hard Copy Setup:
   1.1 Layout: Portrait
   1.2 Format: EPSIMAGE
   1.3 Port: RS232
2. RS-232 setup:
   2.1 Baud: 19,200
   2.2 Flow control: None
   2.3 EOL string: CR/LF
   2.4 Parity: None
3. Computer setup.
   3.1 Select “`HyperTerminal`”
   3.2 When first started it will bring up a “`Connection Description`” display and ask for the name of an icon for the connection. Enter anyting in the window, I usually us `A` and click “`OK`.
   3.3 “`HyperTerminal`” will now bring up another window asking “`Connect To`. Down toward the bottom of the window there is an option for: “`Connect using`”. Select “`COM1''` and “`OK`”.
   3.4 “`HyperTerminal`” now brings up a helpful “`COM1 Properties`” window, where you should select:
      3.4.1. “`Bits per second`”: 19200
      3.4.2. “`Data bits`”: 8
      3.4.3. “`Parity`”: `None`
      3.4.4. “`Stop bits`”: 1
      3.4.5. “`Flow Control`”: `Hardware` or `None`
      3.4.6. Click on `OK`
   3.5 The click on the “`Transfer`” pull down menu item and select “`Capture Text...`”.
   3.6 In the “`Capture Text...`” window, select: “`C:\Captures\EPS`” and give it some logical name to save the data in. You not need to provide a file extension. It ALWAYS adds an extension of `.TXT` even if you specified one.

      This step will require some fooling around wiht the “`Browse...`” system to actually find where to put the data.

      When everything is done, click on `OK` which will bring you up to the main “`Capture Text`” window. At the “`Capture Text`” window, select “`Start`”.

      Any data that now comes in over the serial output of the oscilliscope, will now end up you your choice of file as a text file.
   3.7 Use a “null modem” or “Laplink” type cable to connect the two together.

[29] `$Header: d:/Binder4/sda1/RCS/ScopeSav.inc,v 1.3 2008-02-26 13:38:51-08 Hamilton Exp Hamilton $`

3.8 Using this configuration, it will take 1 to 2 minuets to transfer the data from the oscilloscope to the computer.

While data is being transferred to the computer it will be displayed on the computer screen.

3.9 **When all done transferring data to the computer: reselect the "`Transfer`" pulldown menu item and select the "`Stop`" item. If this is not done all the captured data will be lost.**

3.10 When you select "`Stop`" "`HyperTerminal`" will tell you that: "`You are currently connected. Are you sure you want to disconnect now?`". Select the "`Yes`" to exit the program. (And terminate data saving/collecting.)

3.11 As a last step "`HyperTerminal`" will ask if you want to: "`Do you want to sav the connection named 'x'?`". I have never been able to find where it saves the configuration data and so I answer any way that I feel will make it close its little window so that I can get out of here and exit.

4. The file saved in the above sequence, may be accessed by typeing in `X Y` at the DOS command prompt. This will take you to "`c:\captures\EPS`". At this point you may copy the set of oscilliscope pictures to your thumb drive by typing in `Y` which will copy the oscilliscope pictures to the thumb drive and remove them to `original`.

Index