STAR-Dundee

SpaceWire Engineering Excellence

SpaceWire Link Analyser Mk2: Debugging a SpaceWire Hardware Link Fault

The SpaceWire Link Analyser Mk2 is designed specifically to support the testing and debugging of SpaceWire systems. This application note provides an example of debugging a SpaceWire hardware link fault using a Link Analyser Mk2, saving the engineer significant time and effort in the process.

Scenario

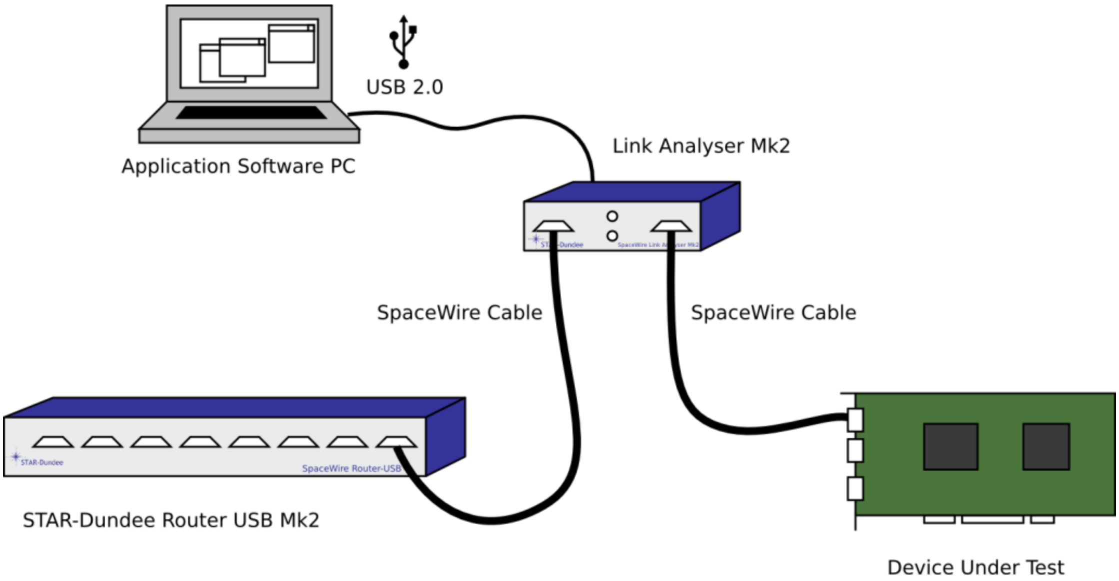
During the development of a piece of SpaceWire equipment we need to debug a hardware problem with a SpaceWire link. When the unit under test is connected to another piece of SpaceWire equipment we observe a link disconnect suggesting an error on the link. For example, the UUT may be connected to a STAR-Dundee Router-USB Mk2 in which case the disconnect is indicated by the link status LED flashing red.

Debugging with the Link Analyser Mk2

In this scenario the Link Analyser Mk2 is used to confirm that the link is operating as expected prior to the fault, identify the type of errors occurring in the link and allow the specific error to be targeted to capture the trace at the time of the fault. The trace can then be inspected for the cause of the fault. The following sections describe how this can be done.

Test Setup

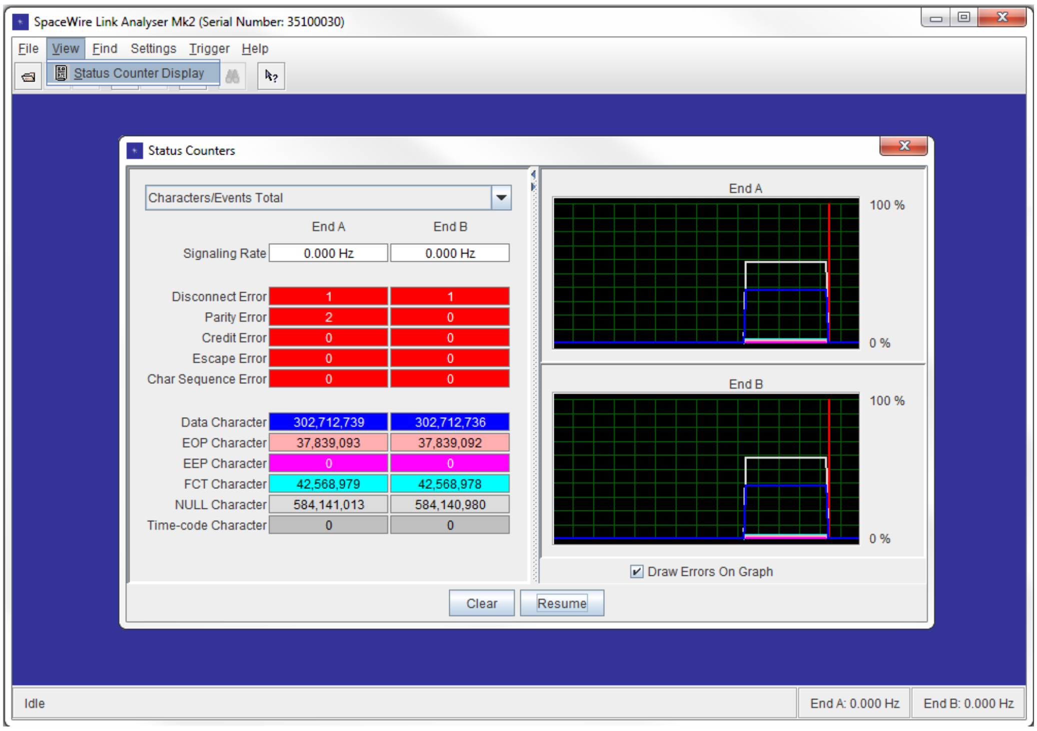
Connect the Link Analyser between the two pieces of equipment using the two SpaceWire interfaces on the front panel. Connect it to a host PC via USB. The diagram below illustrates this configuration. The Link Analyser Mk2 will allow the traffic to flow on the link between the equipment as before while allowing the traffic to be monitored unobtrusively.



Link Analyser Mk2 Connected

View Link Statistics

Once connected, launch the SpaceWire Link Analyser Mk2 software on the host PC. The Link Analyser Mk2 immediately starts to record link statistics. To view the link statistics open the Status Counters display. This shows the number of errors and characters recorded and the current signalling rate. This view can quickly show if the link is operating as expected i.e. the signalling rate is correct and the number and types of characters recorded are as expected. It is also a good immediate indicator of link faults. The screenshot below shows in this example parity errors have occurred on End A and a disconnect error on End A and End B. The signalling rate is zero because at this stage the link has been disconnected. The status counters display can help indicate errors or characters of importance that can be used to trigger the Link Analyser Mk2 and narrow the search for the fault (see below for more on triggering). Alongside the link statistics are graphs that illustrate the character and error activity over time. In this example we see data, EOP, FCT and NULL characters (represented by the blue, salmon, cyan and light grey lines) are flowing in both directions of the link up until a disconnect error occurs on End A and End B (represented by the red line).



Characters/Events Total

	End A	End B
Signaling Rate	0.000 Hz	0.000 Hz
Disconnect Error	1	1
Parity Error	2	0
Credit Error	0	0
Escape Error	0	0
Char Sequence Error	0	0
Data Character	302,712,739	302,712,736
EOP Character	37,839,093	37,839,092
EEP Character	0	0
FCT Character	42,568,979	42,568,978
NULL Character	584,141,013	584,140,980
Time-code Character	0	0

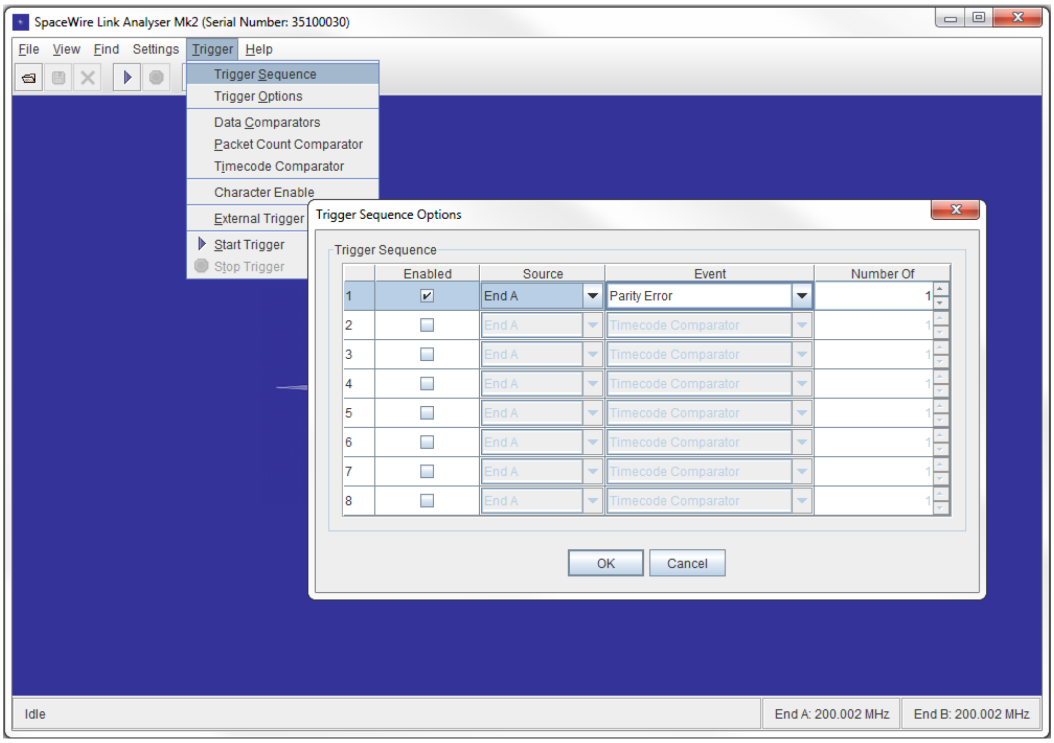
☑ Draw Errors On Graph

End A: 0.000 Hz End B: 0.000 Hz

Status Counter Display

Capture the SpaceWire Traffic

In order to view and debug the SpaceWire traffic between the two pieces of equipment we must first capture it. SpaceWire link traffic is captured when a user defined trigger sequence is detected on the link. The trigger can be configured to occur on a range of different events including time-codes, data characters, control characters and errors. In this scenario we want to identify what error(s) led to the disconnect therefore it may be useful to trigger on a character sequence error, escape error, credit error, parity error, disconnect error or all errors. The status counter display has however already indicated the cause of the disconnect as a parity error on End A. To confirm this we want to analyse the link traffic surrounding the parity errors. The trigger is set to occur when a parity error is detected on End A as shown below. In this example the trigger is set to occur when a single event is detected as there is no need to do anything else. We can however configure the trigger to occur when a complex sequence of multiple different events is detected.

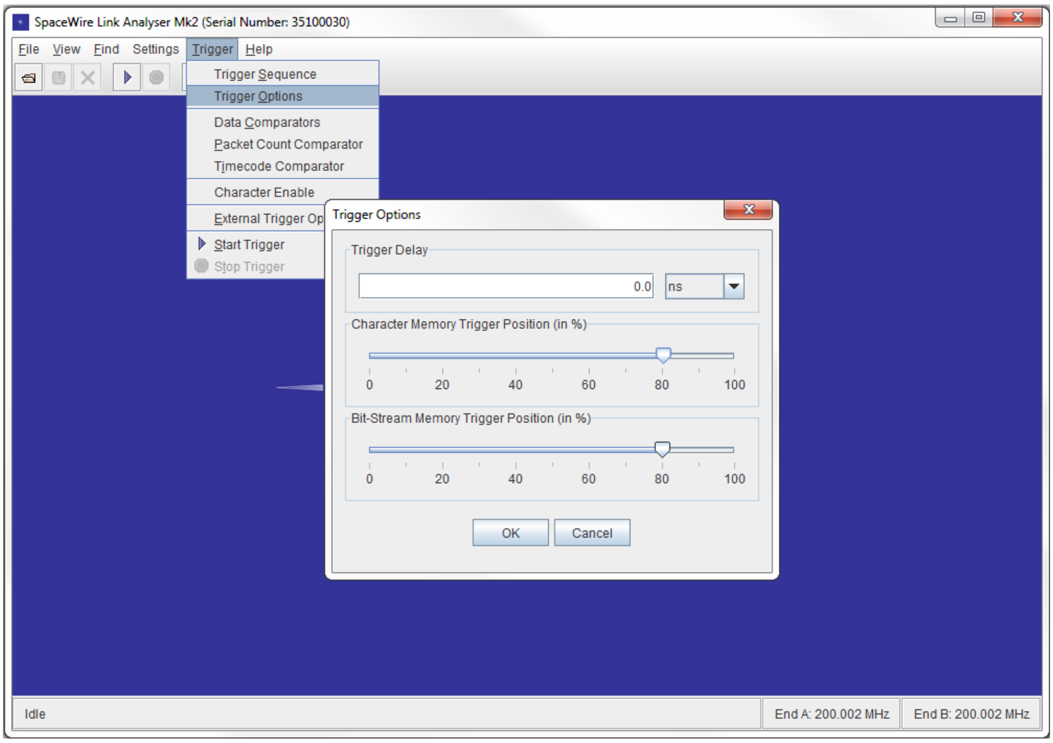


Trigger Sequence

	Enabled	Source	Event	Number Of
1	☑	End A	Parity Error	1
2	□	End A	Timecode Comparator	1
3	□	End A	Timecode Comparator	1
4	□	End A	Timecode Comparator	1
5	□	End A	Timecode Comparator	1
6	□	End A	Timecode Comparator	1
7	□	End A	Timecode Comparator	1
8	□	End A	Timecode Comparator	1

Trigger on Parity Error

The Link Analyser passively buffers the SpaceWire traffic on the link in anticipation of the trigger. When the trigger occurs, traffic before and after the trigger is captured. The memory reserved for before and after the trigger is configurable. In this example it is known that a parity error occurs before the disconnect. Therefore we need only reserve a small amount of memory to capture the trace post trigger to confirm the cause of the disconnect error is the parity error. The majority of the memory is used to store the trace before the trigger to try and capture the cause of any faults.



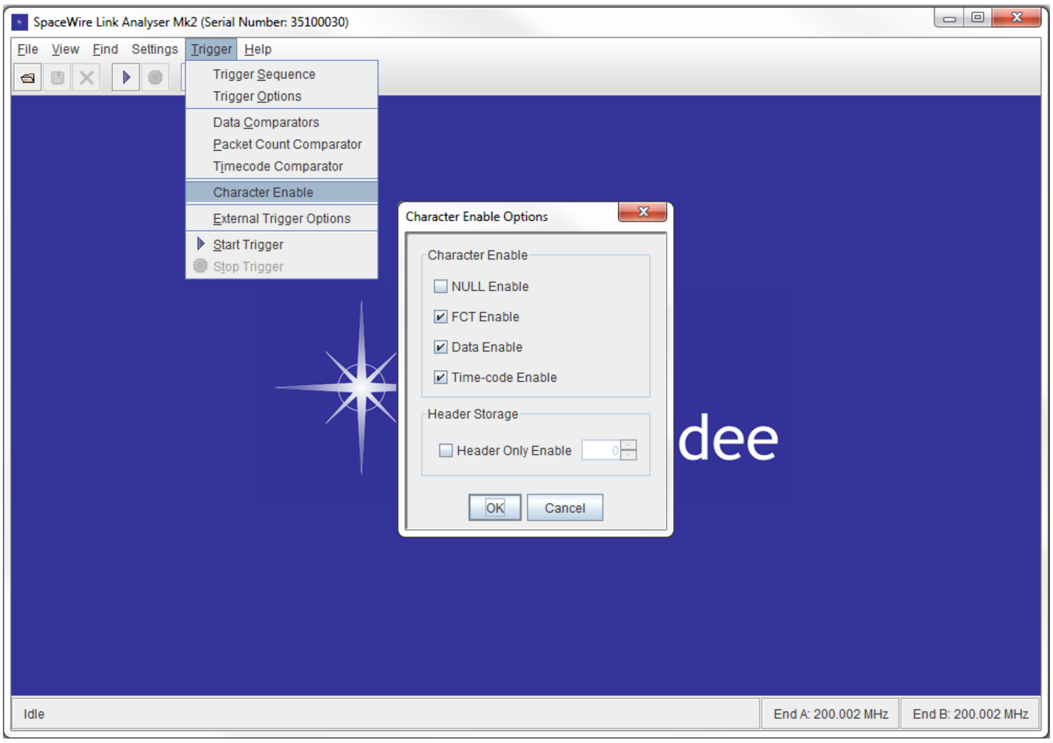
Trigger Delay: 0.0 ns

Character Memory Trigger Position (in %): 80

Bit-Stream Memory Trigger Position (in %): 80

Trigger Options

It is possible to enable and disable recording of specific character types. Disabling the recording of one character type means more memory is available to those remaining enabled character types. As disabled character types are not recorded it also means they are not present in the displays, effectively filtering the character types. For these reasons, in this example, NULL control code recording is disabled.



Character Enable Options

Character Enable

☐ NULL Enable

☑ FCT Enable

☑ Data Enable

☑ Time-code Enable

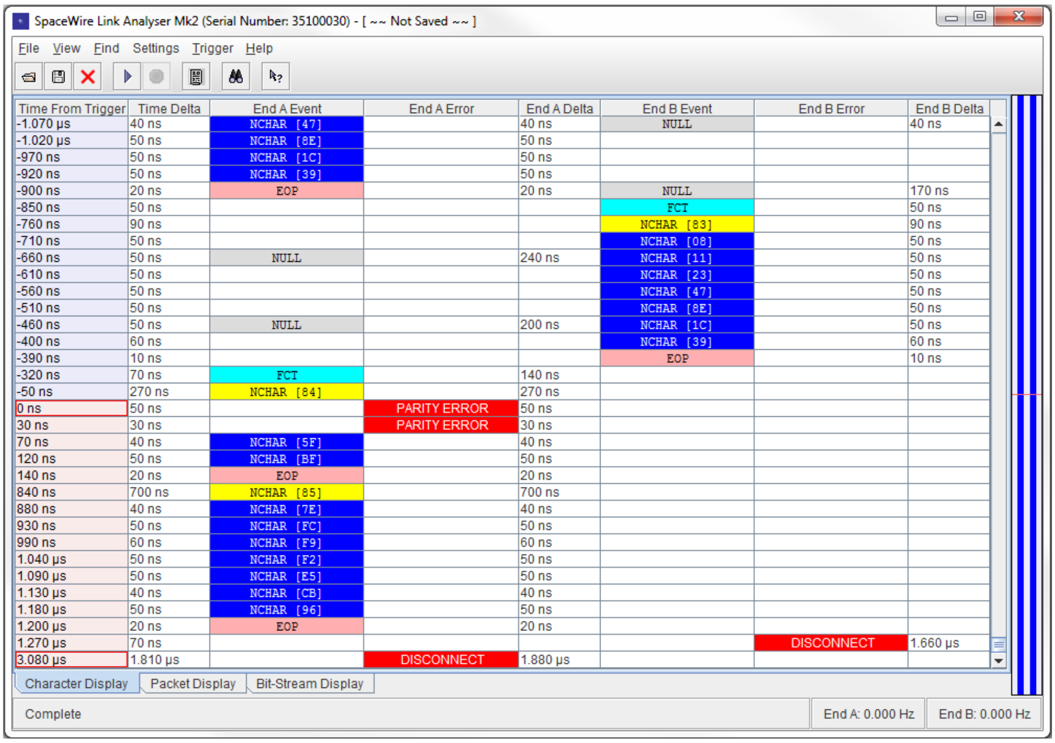
Header Storage

☐ Header Only Enable 0

Disable NULL Control Code Recording

Analyse the SpaceWire Traffic

Once captured, we can inspect the SpaceWire link traffic using the character, packet and bit-stream displays. The character display shows the characters, events and errors captured. In this trace we can see the trigger has occurred on the parity error as setup in the trigger sequence. There are a number of indicators of this in the “Time From Trigger” column: the “Time From Trigger” value equals zero, the cell is outlined in red representing a marker and the column background colour is red (blue indicates before trigger, red after). This is shortly followed by disconnect errors on both ends of the link, confirming the cause of the disconnect is a parity error introduced by the unit under test.



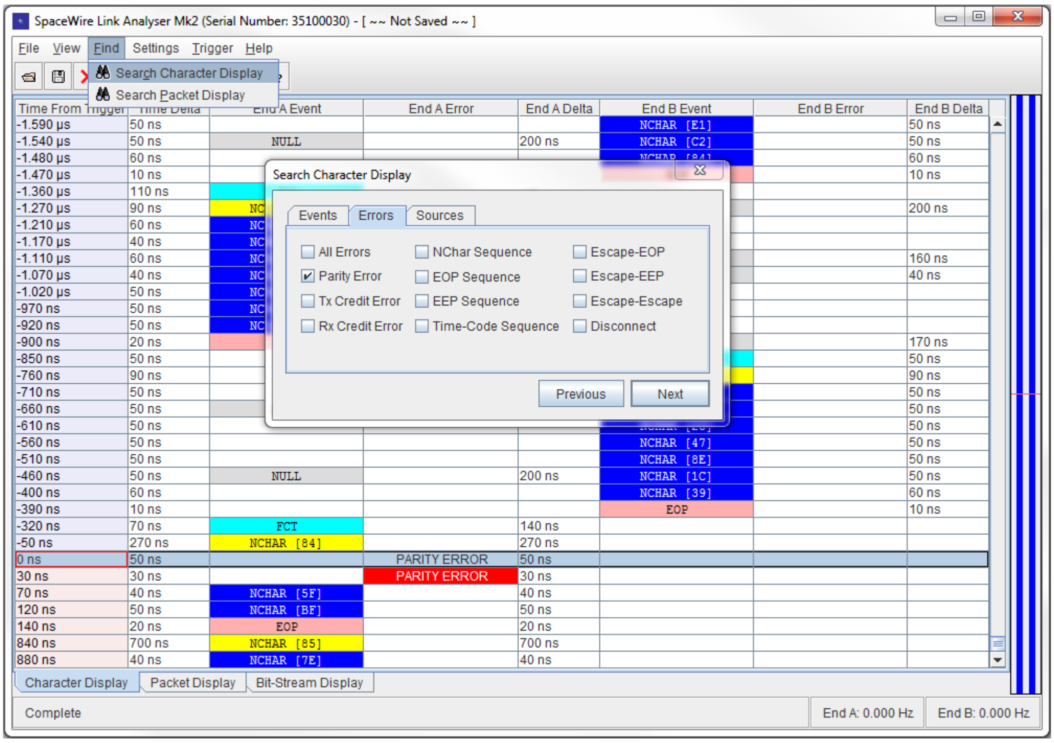
SpaceWire Link Analyser Mk2 (Serial Number: 35100030) - [~~ Not Saved ~~]

Time From Trigger	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta
-1.070 µs	40 ns	NCHAR [47]		40 ns	NULL		40 ns
-1.020 µs	50 ns	NCHAR [8E]		50 ns			
-970 ns	50 ns	NCHAR [1C]		50 ns			
-920 ns	50 ns	NCHAR [39]		50 ns			
-900 ns	20 ns	EOP		20 ns	NULL		170 ns
-850 ns	50 ns				FCT		50 ns
-760 ns	90 ns				NCHAR [83]		90 ns
-710 ns	50 ns				NCHAR [08]		50 ns
-660 ns	50 ns	NULL		240 ns	NCHAR [11]		50 ns
-610 ns	50 ns				NCHAR [23]		50 ns
-560 ns	50 ns				NCHAR [47]		50 ns
-510 ns	50 ns				NCHAR [8E]		50 ns
-460 ns	50 ns	NULL		200 ns	NCHAR [1C]		50 ns
-400 ns	60 ns				NCHAR [39]		60 ns
-390 ns	10 ns				EOP		10 ns
-320 ns	70 ns	FCT		140 ns			
-50 ns	270 ns	NCHAR [84]		270 ns			
0 ns	50 ns		PARITY ERROR	50 ns			
30 ns	30 ns		PARITY ERROR	30 ns			
70 ns	40 ns	NCHAR [5F]		40 ns			
120 ns	50 ns	NCHAR [BF]		50 ns			
140 ns	20 ns	EOP		20 ns			
840 ns	700 ns	NCHAR [85]		700 ns			
880 ns	40 ns	NCHAR [7E]		40 ns			
930 ns	50 ns	NCHAR [FC]		50 ns			
990 ns	60 ns	NCHAR [F9]		60 ns			
1.040 µs	50 ns	NCHAR [F2]		50 ns			
1.090 µs	50 ns	NCHAR [E5]		50 ns			
1.130 µs	40 ns	NCHAR [CB]		40 ns			
1.180 µs	50 ns	NCHAR [96]		50 ns			
1.200 µs	20 ns	EOP		20 ns			
1.270 µs	70 ns					DISCONNECT	1.660 µs
3.080 µs	1.810 µs		DISCONNECT	1.880 µs			

Complete | End A: 0.000 Hz | End B: 0.000 Hz

Character Display

In this scenario the parity error has already been identified as the cause of the disconnect but in a different scenario it may not be immediately obvious what the problem is. The Link Analyser software can display up to one million events and so searching for problems manually can be time consuming. To save time and effort the character display has a search feature to aid navigation and debugging that can be used to find specific events and errors.



SpaceWire Link Analyser Mk2 (Serial Number: 35100030) - [~~ Not Saved ~~]

Search Character Display

Events | Errors | Sources

□ All Errors	□ NChar Sequence	□ Escape-EOP
☑ Parity Error	□ EOP Sequence	□ Escape-EEP
□ Tx Credit Error	□ EEP Sequence	□ Escape-Escape
□ Rx Credit Error	□ Time-Code Sequence	□ Disconnect

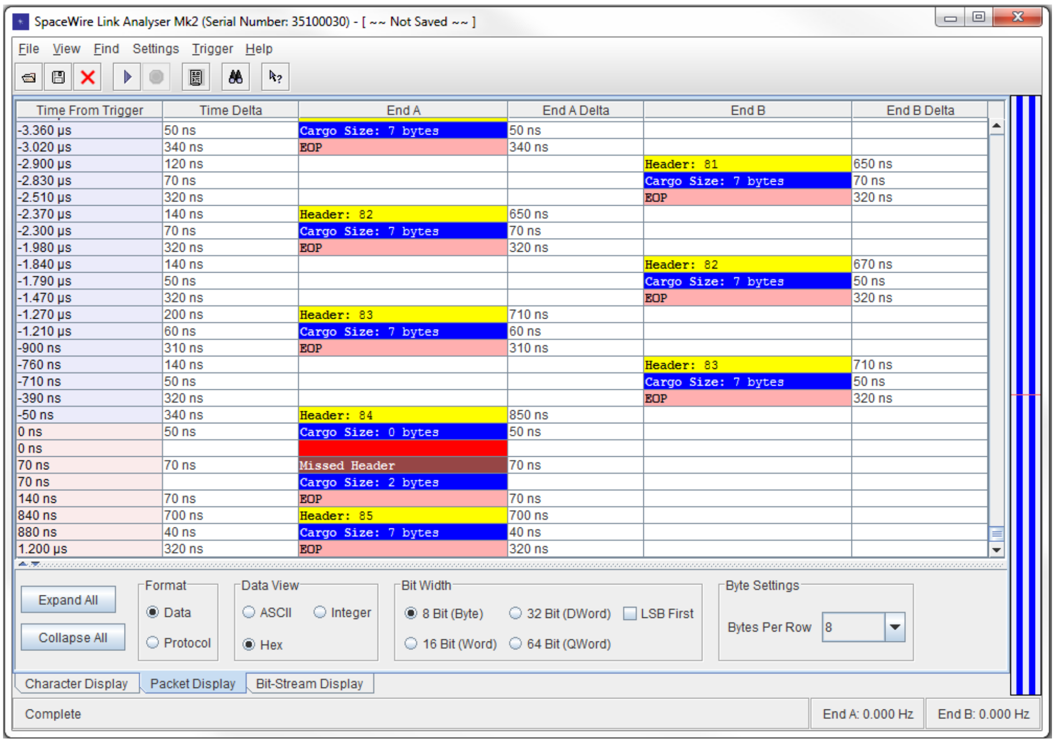
Previous | Next

Time From Trigger	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta
-1.590 µs	50 ns				NCHAR [E1]		50 ns
-1.540 µs	50 ns	NULL		200 ns	NCHAR [C2]		50 ns
-1.480 µs	60 ns				NCHAR [84]		60 ns
-1.470 µs	10 ns						10 ns
-1.360 µs	110 ns						
-1.270 µs	90 ns	NC					200 ns
-1.210 µs	60 ns	NC					
-1.170 µs	40 ns	NC					
-1.110 µs	60 ns	NC					160 ns
-1.070 µs	40 ns	NC					40 ns
-1.020 µs	50 ns	NC					
-970 ns	50 ns	NC					
-920 ns	50 ns	NC					
-900 ns	20 ns						170 ns
-850 ns	50 ns						50 ns
-760 ns	90 ns						90 ns
-710 ns	50 ns						50 ns
-660 ns	50 ns						50 ns
-610 ns	50 ns						50 ns
-560 ns	50 ns				NCHAR [47]		50 ns
-510 ns	50 ns				NCHAR [8E]		50 ns
-460 ns	50 ns	NULL		200 ns	NCHAR [1C]		50 ns
-400 ns	60 ns				NCHAR [39]		60 ns
-390 ns	10 ns				EOP		10 ns
-320 ns	70 ns	FCT		140 ns			
-50 ns	270 ns	NCHAR [84]		270 ns			
0 ns	50 ns		PARITY ERROR	50 ns			
30 ns	30 ns		PARITY ERROR	30 ns			
70 ns	40 ns	NCHAR [5F]		40 ns			
120 ns	50 ns	NCHAR [BF]		50 ns			
140 ns	20 ns	EOP		20 ns			
840 ns	700 ns	NCHAR [85]		700 ns			
880 ns	40 ns	NCHAR [7E]		40 ns			

Complete | End A: 0.000 Hz | End B: 0.000 Hz

Search Character Display for Parity Error

The packet display provides a packet level view of the captured data. In this scenario we can see the link errors (indicated by the red cell) have affected two packets on End A: one is missing an EOP marker and another is missing the packet header.


SpaceWire Link Analyser Mk2 (Serial Number: 35100030) - [~~ Not Saved ~~]
File View Find Settings Trigger Help
Expand All
Collapse All
Character Display
Packet Display
Bit-Stream Display
Complete
End A: 0.000 Hz
End B: 0.000 Hz

Time From Trigger	Time Delta	End A	End A Delta	End B	End B Delta
-3.360 µs	50 ns	Cargo Size: 7 bytes	50 ns		
-3.020 µs	340 ns	EOP	340 ns		
-2.900 µs	120 ns			Header: 81	650 ns
-2.830 µs	70 ns			Cargo Size: 7 bytes	70 ns
-2.510 µs	320 ns			EOP	320 ns
-2.370 µs	140 ns	Header: 82	650 ns		
-2.300 µs	70 ns	Cargo Size: 7 bytes	70 ns		
-1.980 µs	320 ns	EOP	320 ns		
-1.840 µs	140 ns			Header: 82	670 ns
-1.790 µs	50 ns			Cargo Size: 7 bytes	50 ns
-1.470 µs	320 ns			EOP	320 ns
-1.270 µs	200 ns	Header: 83	710 ns		
-1.210 µs	60 ns	Cargo Size: 7 bytes	60 ns		
-900 ns	310 ns	EOP	310 ns		
-760 ns	140 ns			Header: 83	710 ns
-710 ns	50 ns			Cargo Size: 7 bytes	50 ns
-390 ns	320 ns			EOP	320 ns
-50 ns	340 ns	Header: 84	850 ns		
0 ns	50 ns	Cargo Size: 0 bytes	50 ns		
0 ns					
70 ns	70 ns	Missed Header	70 ns		
70 ns		Cargo Size: 2 bytes			
140 ns	70 ns	EOP	70 ns		
840 ns	700 ns	Header: 85	700 ns		
880 ns	40 ns	Cargo Size: 7 bytes	40 ns		
1.200 µs	320 ns	EOP	320 ns		

Format: ● Data ○ Protocol

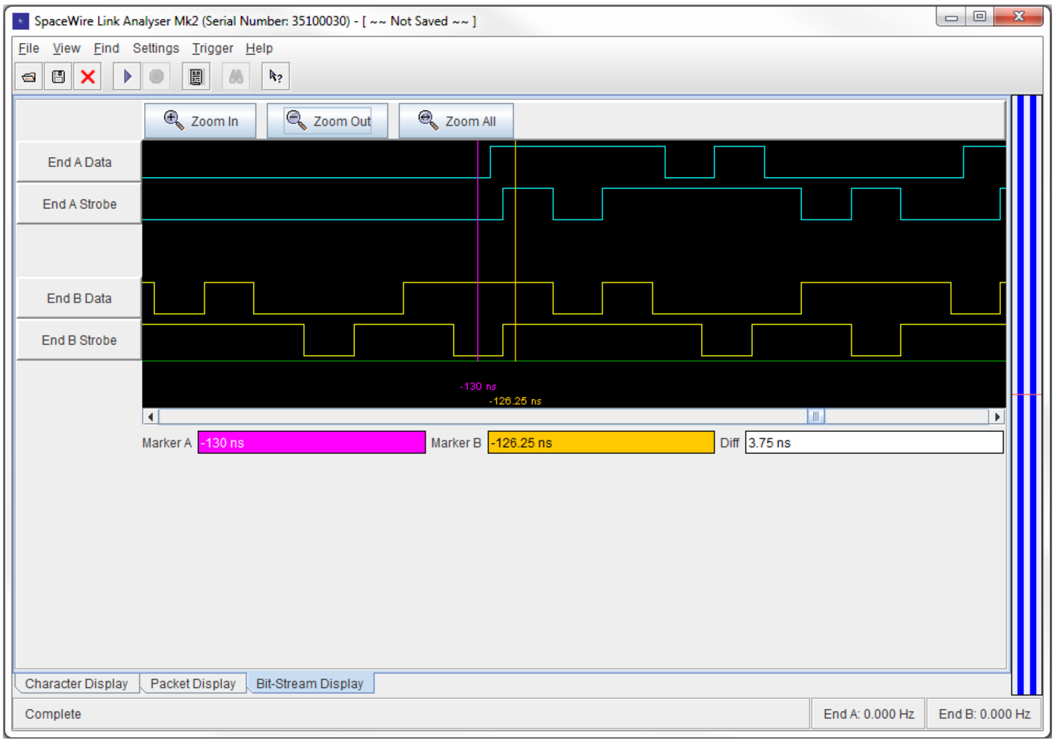
Data View: ○ ASCII ○ Integer ● Hex

Bit Width: ● 8 Bit (Byte) ○ 32 Bit (DWord) □ LSB First ○ 16 Bit (Word) ○ 64 Bit (QWord)

Byte Settings: Bytes Per Row 8

Packet Display

The bit-stream display provides a graph of the data and strobe signals captured on both SpaceWire links. The screenshot of the bit-stream display below shows the cause of the parity errors in this scenario. Between the marker A (the purple vertical line) and marker B (the orange vertical line) a simultaneous transition is highlighted on the data and strobe signals on End A.



Marker A -130 ns Marker B -126.25 ns Diff 3.75 ns

Bit-Stream Display

Inspecting these views and comparing expected SpaceWire link traffic with actual traffic helps trace the cause of SpaceWire link problems. In this example the character display has confirmed the cause of the disconnect is a parity error, the packet display has shown those packets affected by the errors and the bit-stream display has shown the parity errors are a result of a simultaneous transition.

Conclusion

When debugging a SpaceWire hardware link problem, using a Link Analyser Mk2 allows the engineer to inspect the link traffic; target specific event sequences and filter information, and therefore enhances their ability to solve the problem quickly and efficiently.

In this application note we have seen how the Link Analyser is connected and how to set a trigger to capture SpaceWire link traffic at the most appropriate time. Once captured, the Link Analyser’s displays let us inspect and debug the traffic flowing in each direction at the character, packet and bit-stream level.

This document only briefly describes some of the Link Analyser’s capabilities in relation to a specific problem. The Link Analyser Mk2 is provided with context sensitive help and a user manual that comprehensively describes the Link Analyser Mk2.

For more information please visit our website at www.star-dundee.com or contact us at enquiries@star-dundee.com.