SpaceWire Decoder, Documentation for Release 7.4

Lahniss Sarl, June 2014

Table of Contents

List of Figures

Introduction

The SpaceWire Decoder developed by Lahniss for Teledyne LeCroy oscilloscopes is a tool aimed at decoding SpaceWire streams emitted by various devices. This decoder is based on the ECSS-E-ST-50-12C specification.

Introductory comments to the manual:

- This document relies on the assumption that the reader is familiar with Teledyne LeCroy oscilloscopes in general.
- This document also assumes some familiarity with the above mentioned standard, ECSS-E-ST-50-12C, published by the European Space Agency.
- Dialogs are shown with black backgrounds. Some Teledyne LeCroy instruments have violet backgrounds. There are no functional differences between oscilloscopes showing tabs in violet or black, colors and shades merely reflect shifts in industrial fashion and design over the years.
- The use of the word "Sync" could lead to believe there is a SpaceWire Trigger, which there is not. "Sync", in the context of SpaceWire Decoder merely refers to the Synchronization mechanism necessary to align the bits with the N/L-Chars boundaries.

Before we get started, we need to **emphasize the methodology** underlying the software described here. The fundamental model is a 3 steps model. The signal needs to be decoded as:

- The stream is decomposed into **Bits,** using either the Strobe line or the Bitrate.
- The bits are grouped into **N-chars and L-chars,** based on the Sync Rule chosen by the operator, and provided the Rule can be applied over the span of the acquisition window.
- The **N-chars and L-chars** are grouped into **Packets** based on the rule that a Packet starts when exiting NULL stream and ends on the next EOP symbol.

The **N-chars and L-chars** conversion will not be functional until the stream is correctly separated into **Bits**, and the **Packets** extraction will not function correctly until the **N-chars and L-chars** are properly decoded.

The tuning of the Decoder must therefore be conducted in this order:

Bits → N-chars and L-chars → Packets.

The following material will guide the operator through each step of the process, beginning at the **Bit** level and ending at the **Packet** level.

Getting started

In order to get started with the SpaceWire Decoder, it is advisable to adjust the scope controls to acquire a "sufficient "sequence of relevant Data, and then **stop the acquisition**. The very notion of "sufficient "will become clearer in the following pages.

Once the SpaceWire stream is acquired, we will interpret it using the SpaceWire Decoder. This will be done gradually; starting with the extraction of the bits, then extract the N/L-chars and finally the Packets, by grouping N/L-chars into Words.

Later we will address the decoding of many Packets into the same record, therefore allowing the observation of the encoded data values over a period of time. The **decoder settings determined on a few packets** will be reused when handling many packets.

In order to start the decoding, we need to be familiar with the User Interface of the Decoder.

Firstly, in the Serial Decode dialog, you need to select the Data source (here C1) and the Strobe source (here C2), and the Protocol, "SpaceWire" in this case. Note that under certain conditions the Strobe input is not required; this will be discussed later in this document.

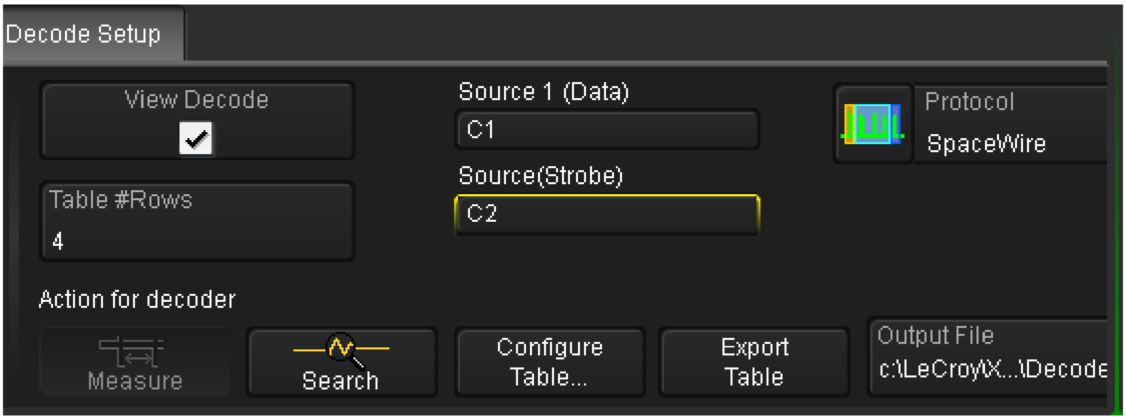


Figure 1: The selection of SpaceWire in the Decode Setup

Once the "SpaceWire" Protocol has been selected 3 tabs will appear in the Right Hand Side Dialog. We will be setting various values in these 3 tabs: **Basic, Sync, and Levels.**

To get started the Data and Strobe lines should be connected. Note that the amplitude of either signal does not matter, as the decoder solely relies on state transition. Henceforth, the SpaceWire stream can be decoded regardless of the signal amplitude.

Figure 2: The protocol selection governs the appearance of the Right Hand Side tabs.

All of the controls available in the Right Hand Side tabs are explained below.

Initially the dialog will appear in the following manner:

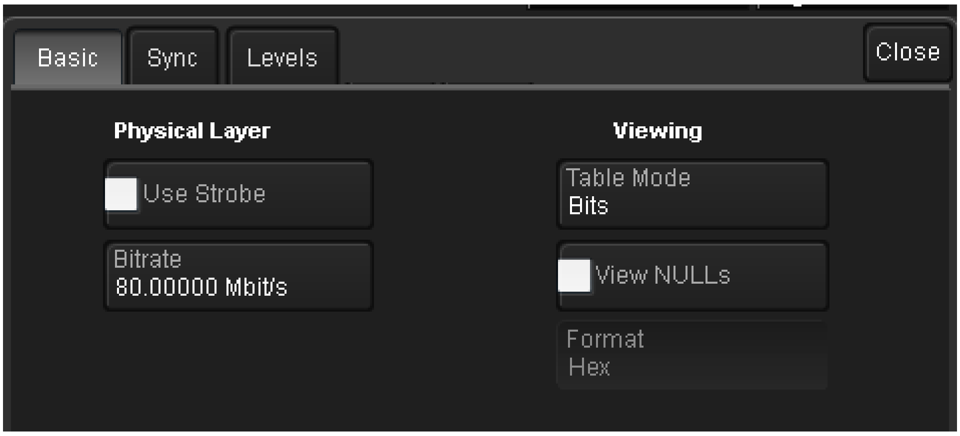


Figure 3: Initial settings of the Basic Dialog

Once the Data and Strobe lines have been connected to the oscilloscope, the first operation required is the setting of the Bitrate. On a SpaceWire stream the Bitrate can vary over time, until the final operation speed has been reached, it might therefore seem strange to set the Bitrate! Here is the explanation. Using the Strobe Signal requires an input channel on the DSO. This can be a luxury when the observation of other signals is needed and somewhat unnecessary when the SpaceWire stream is running at constant Bitrate. At constant Bitrate the signal can be assimilated to a normal, continuous NRZ signal and decoded as such, by interpolating the non-transitional bits using the declared Bitrate. This computational method is stable provided the stream is reasonably stable and saves the Strobe channel. It is therefore entirely left to the user to choose either with or without Strobe decoding.

The next operation required is the selection of the Table Mode. The SpaceWire decoder offers 3 choices.

Figure 4: The 3 possible choices of Table Mode

When working on a given signal, Physical Layer values in the tabs will not change anymore because they are strongly linked to the signal (i.e. BitRate, Absolute levels). On the other hand, the Viewing parameters can be changed at any point in time obtain optimal results, or understand the reasons for misbehaviors.

Decoding the SpaceWire stream as Bits

In order to start adjusting the decoder, it is best to select the Decode Type “Bits” In this mode every the SpaceWire Stream will appear as a continuous sequence of bits.

Rapidly verifying the correct settings of the fundamental parameters

In Bit Mode with the correct Bitrate, the expectation is that the annotated bits match the underlying signal perfectly. Using a large scale zoom it is easy to visually inspect the decoder’s output. The following figure shows an example of correct decoding. Each signal plateau, high or low is divided up into an integer number of bits. The bit transitions are aligned with the signal transitions. There are no gaps in the annotation, no overlaid annotations.

The only tolerated exceptions could be on the first and last bits of the trace, close to the edge of the grid.

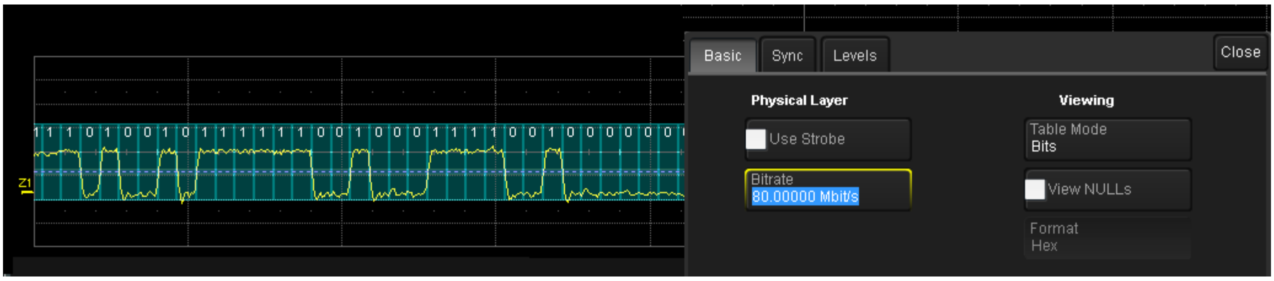


Figure 5: Example of correct Bit level decoding with zoomed trace and Basic Tab

The first rapid validation can be visual, by randomly looking at different portions of the decoded trace. The trace can be assessed, so that it is firmly established that the decoder setting are correct for the signal at hand. In principle, as long as the signal source remains identical, (measuring on same sensor, device or unit) the procedure does not need to be repeated. When measuring on different streams of the same type, and parameterized in the same way, the same settings can usually be re-used.

Using the Strobe or using the Bitrate

As explained above, it is possible to decode with or without the strobe. It is however only possible to decode without the strobe during steady-state activity of the bus, which means that the bitrate is constant over the acquisition window. We will now show that the logic equivalence of both methods using a steady state caption of a 200 Mb/s stream

Figure 6: Logical equivalence of decoding with or without Strobe

The annotated image shows, in green, when Strobe is used (right) and when it is not (left)

The bit level measurement and association with the Bitrate are shown in purple.

The resulting decoding differences are shown in pink, both on the annotated trace and the table results. On the annotation one can see that the bit contents is identical **"01000111"** in both cases. However when closely watching the bit width, it is noticeable that the bits do not have exactly the same width in both cases. **When using the strobe**, the bit boundaries (in the transition free segments) are aligned with the strobe transitions. **When not using the strobe** the bit boundaries are equidistant and solely driven by the bitrate. When the strobe is not used a slight gap can occur at the end of a sequence of identical bits when "catching up" with the next real transition.

Another difference is visible in the table below the grids. Both tables contain, vertically, the same bit sequence of**"01000111"** as in the annotation. However, the bit start times are different for interpolated bits. Start times are identical on bits starting on a real signal transition.

To conclude, we can say that the user has to decide to use or not use the strobe whenever possible, based on the factors explained above.

Detailed verification Of Bit Level Decode using the Zoom

The zoom and search functions allows a more systematic verification of the decoding, explained here. Once the decoding is in engaged, the Table appears below the grid. When starting on a new signal, it is often useful to invest a bit of time to make sure the fundamental parameters are selected correctly. By clicking into the first columns (Line Index), a zoom of the trace corresponding to the selected line will appear, for example the next Figure shows the zoom of line 5734 (of the table) in Z1. The zoom is a precious tool when studying a decoded trace because every Bit, Char or Packet can be rapidly analyzed. When clicking on the Zoom descriptor, the Zoom and Search controls appear as below.

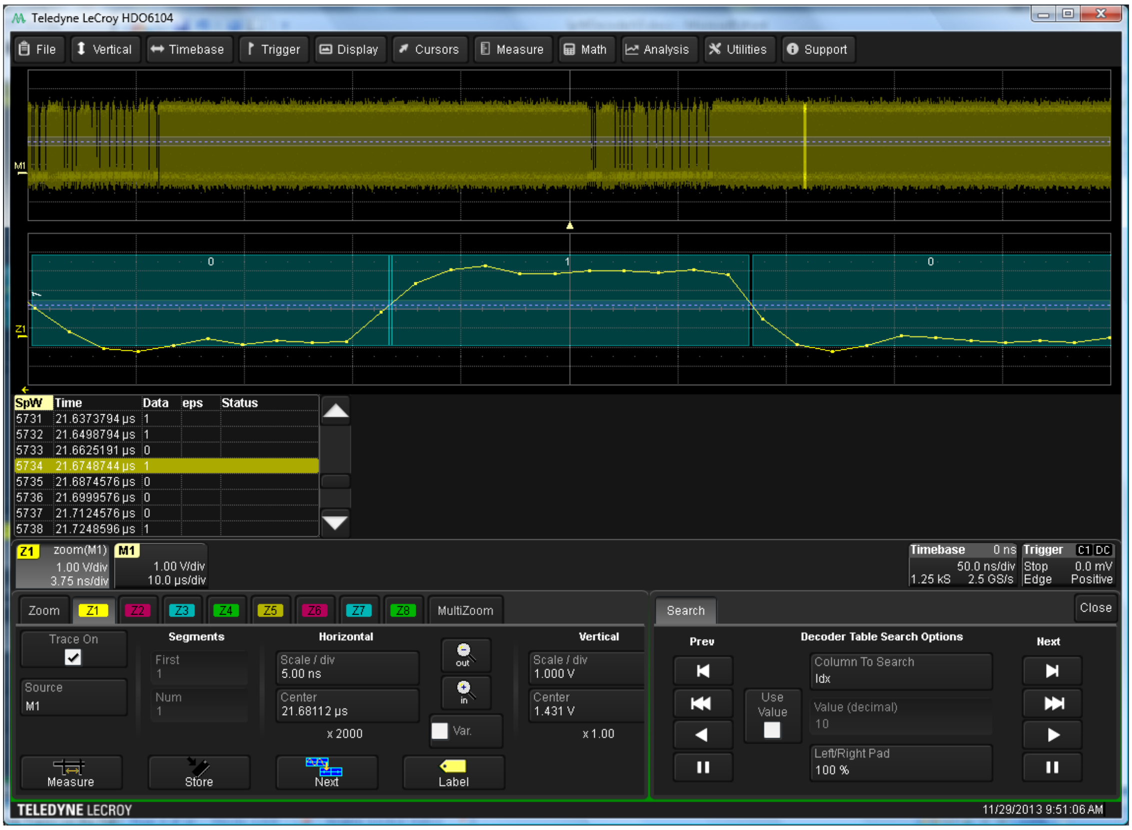


Figure 7: How to "Play" through every decoded Bit of the record

The image above highlights the procedure. First select the Framing of the zoomed packets by adjusting the Left/Right padding. A 100% padding means that a full message length will be added right and left of the message zoomed at when clicking on any line of the table.

Then jump to the very first decoded packet in the record by pushing 1. Then, push 2 to "Play" through the entire record, jumping from one packet to the next, at a rate of approximately 1 image/second. Watch the packets while the play continues and make sure that the decoding is consistent with your expectations. Note that at this stage the stream would not need to of SpaceWire nature, but could be any other continuous NRZ stream, i.e. GPON, USB or else.

More information on the Search, used by many other Lahniss decoders, can be found on:

http://www.lahniss.com/_uFAQ/FAQ.html Item 8, Decode and Search Users Manual

Controls of the Basic Tab

As intuitively shown above, we will now describe every control of the Basic Tab. These controls govern all of the basic decoding, and let the user adjust the decoding to the fundamental protocol parameters

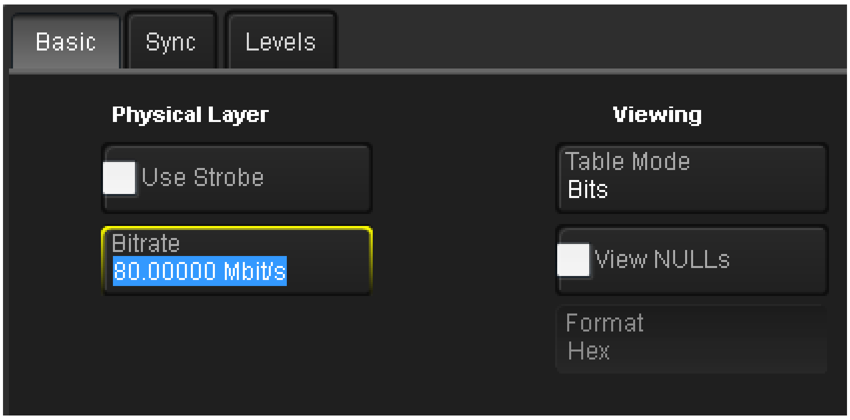


Figure 8: The Basic Tab and its Controls

UI control	Function	Range	Default
	Physical Layer		
Use Strobe	Selects bit level decoding method, using either Strobe or Bitrate.	true/false	false
Bitrate	When not using the Strobe, selects the Bitrate of the SpaceWire signal. When the Strobe is used the Bitrate is not used and disregarded by the decoding algorithm.	1-500 Mbits/s	10 Mbits/s
	Viewing		
Table Mode	Governs the level of decoding. It is usually advisable to start decoding in Bit Mode, and then switch to other modes.	Bits, N/L-Chars or Packets	Bits
View NULLs	Allows to display the NULL character separating Packets. By default the NULLs are not shown, except for 2 NULLs preceding and following Packets. Depending on the topology of the stream, the NULLs can clutter the view, or conversely can help understand a sequence of events.	true/false	false
Format	Depending on the nature of the stream observed, one or the other viewing mode is more appropriate.	Hexadecimal, ASCII or Decimal	Hexadecimal

Table 1: List of Controls in the Basic Tab

Controls of the Level Tab

The last tab of the decoder controls the levels used for determining the edge crossings of the SpaceWire signal. The default settings of Percent level = 50% and Hysteresis = 15% are usually appropriate for most signals. However certain signals can require other settings.

A known case is signals with a varying DC component, either because the probing is incorrect or because the signal is really floating. In this case the level Type Absolute allows a fixing of the threshold level, so that messages can be decoded without having the dynamic change due to the floating behavior.

Another case is very noisy signals, where a combination of level and hysteresis can be used to overcome the noise impact. Note that in this case some upstream filtering in the channel menu can also help for a better definition of the input signal.

The following image shows the interactions between the level settings in the Level tab, the Source selection assigning M1 to be Data and M2 to be Strobe and the corresponding blue dotted level line on each trace. For the sake of demonstration we chose a Level Type in Percent on the Data while the level Type on the Strobe is percent.

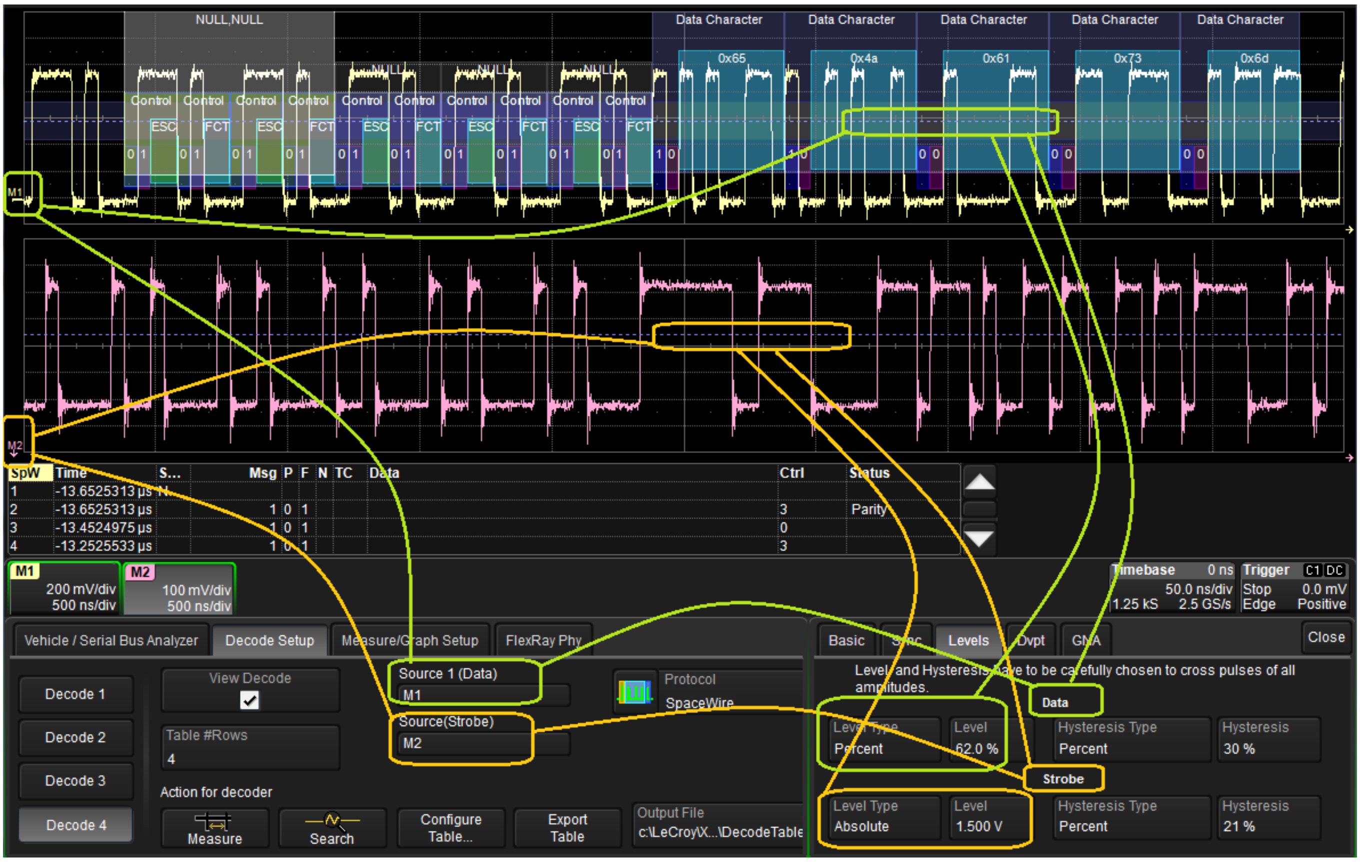


Figure 9: The Level & Hysteresis controls and there visualization on trace

The image above shows an example of healthy signals decoding. When dealing with pathological signals, please refer to the corresponding Appendix below. “Using level and Hysteresis for difficult signals”

Decoding the SpaceWire stream as N/L-Chars

Switching to N/L-Char Decode Mode

Once the decoding as Bits yields satisfactory result, the decoding as N/L-Chars can be activated by setting the Table Mode to "N/L-Chars". The SpaceWire Bits will be further analyzed to interpret the Normal and Link Characters in the context of the SpaceWire protocol structure.

For the sake of clarity in this first example, we will switch on the N/L Char Mode and show the complete screen, with the mother trace (memory trace M1 here), daughter trace (zoomed trace), table and dialog. Subsequent examples will only show trace details.

By clicking into the first columns (Line Index), a zoom of the trace corresponding to the selected line will appear. In this case the zoomed trace is centered on line 786 of the decode Table, containing the complete decoded N char 0xf5 with its infrastructure.

Interpreting the Decoded N/L Chars

The 10 bit Data Character (Light Blue grey) spans:

- char 0xf5 (0xf5 Mountain Lake Blue). This is data segment itself, containing 8 data bits and not subdivided into bits for performance reasons. The individual data bits could be seen by reversing to Bit Mode if desired.
- The preceding Parity bit (P=1 Royal blue)
- The preceding Control Flag Bit (C=0, Bordeaux).

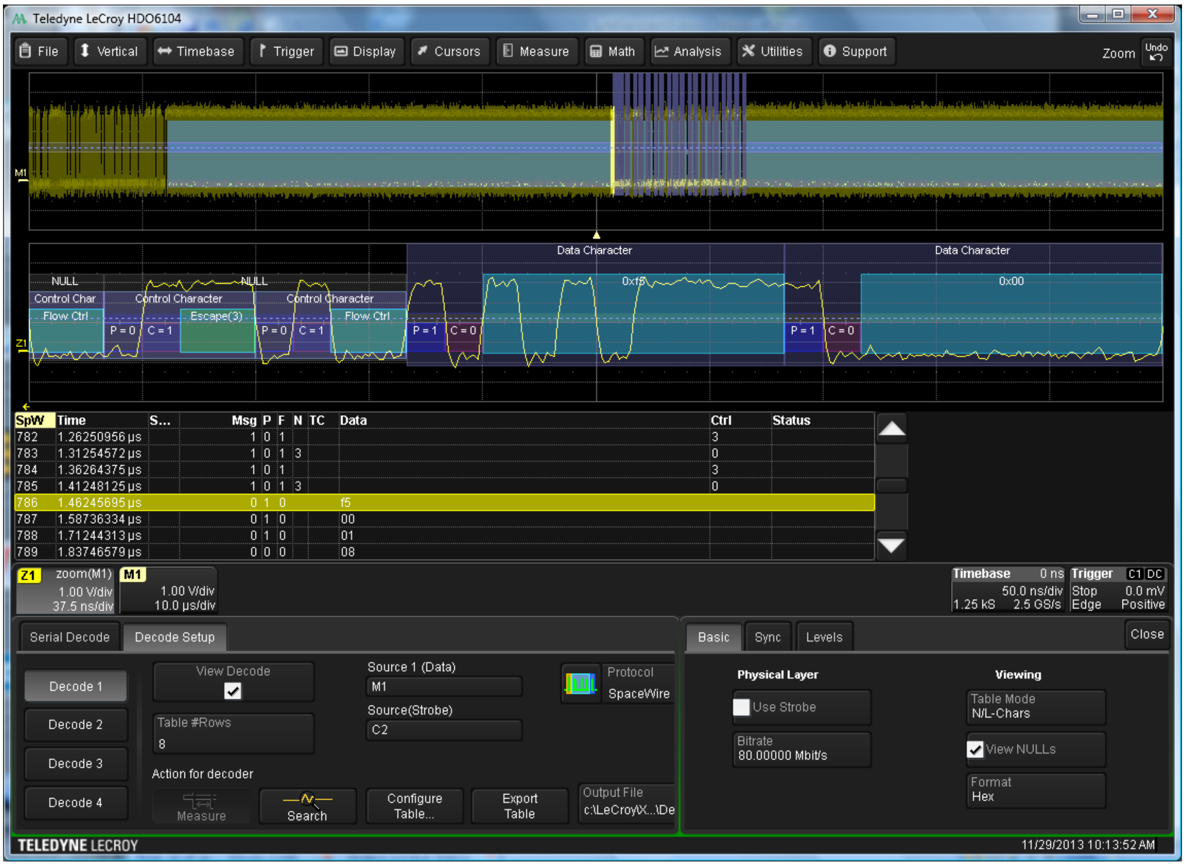


Figure 10: SpaceWire N/L-Char decode

Setting the Sync Parameters (governing the N/L-Char decoding)

The Sync parameters govern the grouping of Bits into N/L characters. This mechanism is very important and needs to be explained in details. Unlike a hardware device, connected to a SpaceWire port since the startup of the link, the decoder is used to monitor the stream at any point in time and without knowledge of the startup sequence. It is therefore presented, at its input, with a continuous stream of bits without any clue as to when a Normal or Link character starts and ends. Therefore a software mechanism is needed to Synchronize, (could also be called Aligned or Lock) the Bit stream onto the Char stream. Given the fact that several cases can occur, the SpW decoder has been equipped with several Sync mechanisms. We will now look at all of them, beginning with the default setup, which is currently thought of being the simplest and most straightforward method for most users. Time will tell is this assumption holds true.

Also note that the Sync mechanism is not a hardware trigger.

Synchronization using NULLs (default)

In this mode the decoding algorithm scans the record from left to right, looking for a bit pattern corresponding to a user select number of consecutive NULLs. The default value is 2, based on the traces used during the development phase. The following image explains the behavior with the number of NULLs selected as being 2. The Decoder has been equipped with a dedicated “Sync” column containing the elements needed to annotate the “Rule” used as Sync. In this way it is easy to verify that the Sync has locked onto the correct pattern. Furthermore the “Sync Rule” can be changed in real time and the result observed on screen

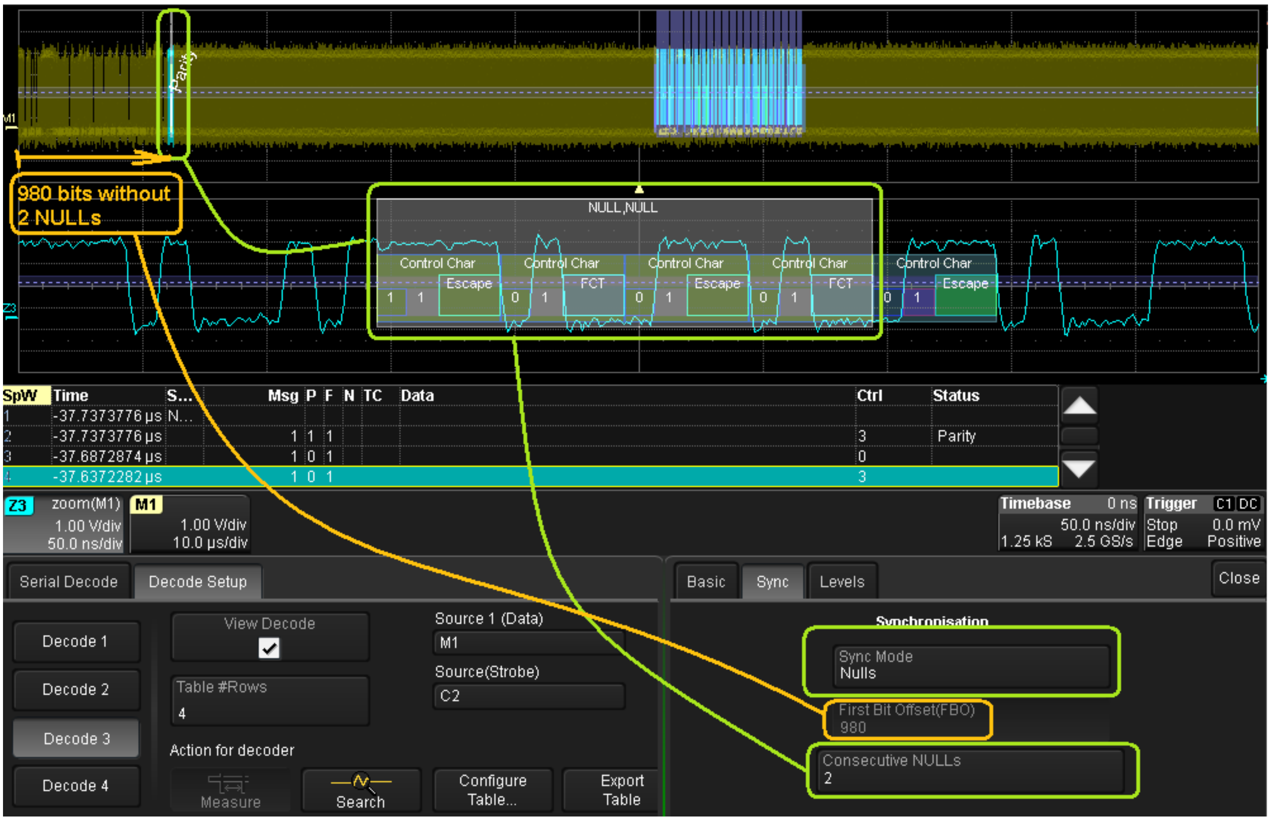


Figure 11: Behavior of the Synchronization on NULLs

It is worth noticing that the First Bit Offset (FBO) widget is grayed out, meaning that the user has no access to it. In fact the FBO is computed by the Sync Scanner and emitted to the control so that the user can see it and possibly modify it when reversing to the Manual Mode. The following images shows variations of the Sync when using a number of NULLs other then 2.

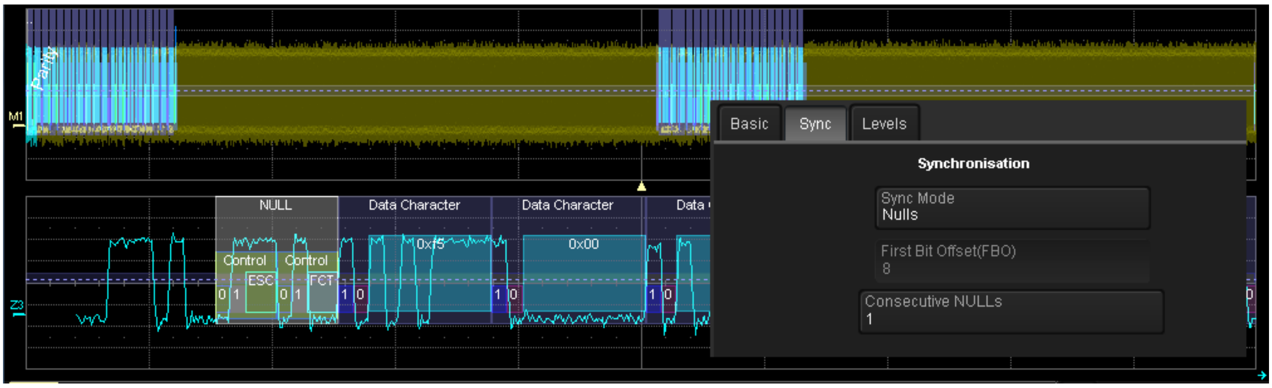


Figure 12: Synchronizing on 1 NULL only

When using only 1 NULL to Sync, the patter is found much earlier in the record. In this example the decoding proceeds correctly because there is sufficient data redundancy, however using only 2 NULLs is deemed unsafe, based on the examples used for developments of the Decoder

Conversely, here is an example using 20 NULLs. This case would be very safe since the statistical probability of finding Data that would look like 20 NULLs is very low, if not zero. Note that the larger Sync patter is now visible both on the mother trace and the daughter trace.

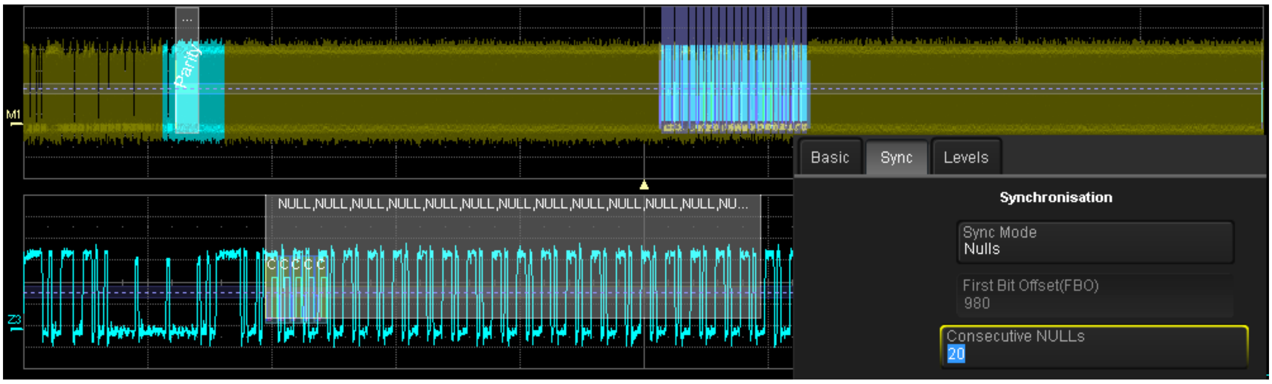


Figure 13: Synchronizing on 20 NULLs

Synchronization using a Pattern

In this mode the user can enter a symbolic pattern composed of NULLs, Data Chars, ESC, FCT and Time Codes. In the same manner as the record is searched for NULLs in NULL mode , here the record will be searched for the pattern until it is located.

As a first example we switch to Pattern mode with an empty pattern. An empty pattern provides no material for locking onto the bit stream, and therefore the computed FBO ends up being 1. The resulting decode is erroneous and riddled with Parity errors, showing up in red in the annotation and with error messages in the Decode Table.

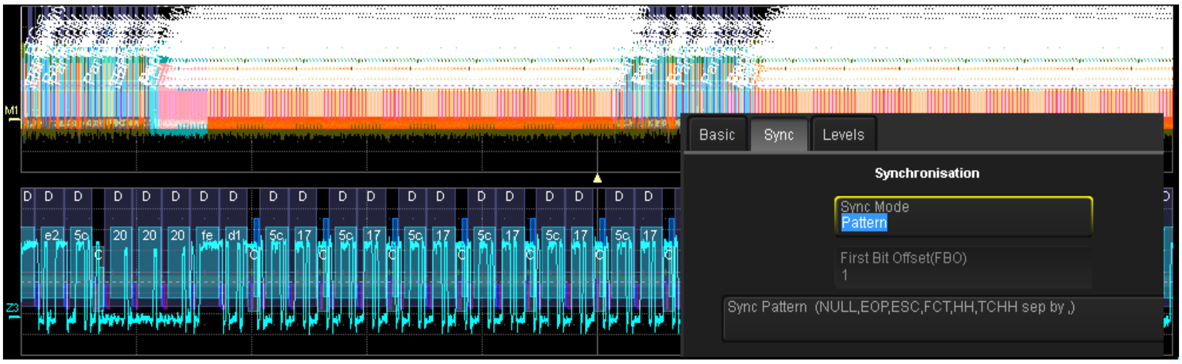


Figure 14: Synchronizing on a pattern of NULL,NULL

The next example shows various possible locking patterns.

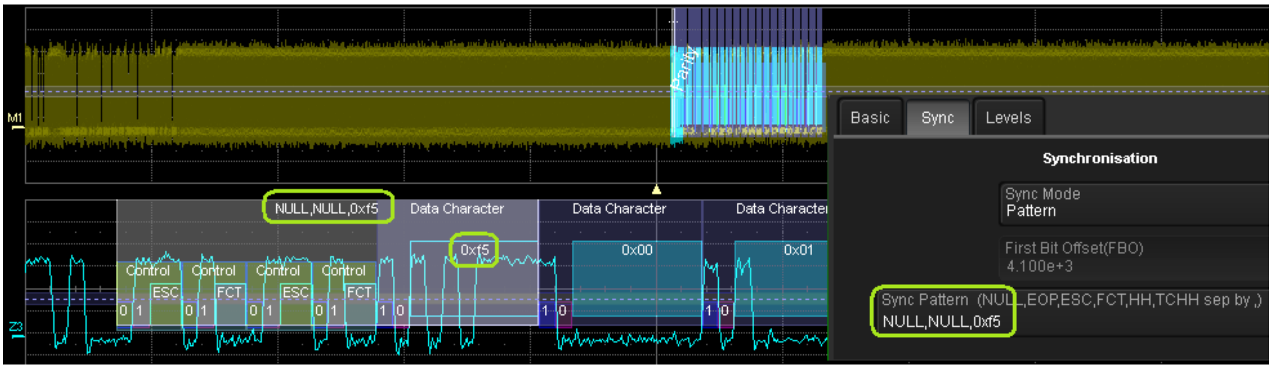


Figure 15: Synchronizing on the first character of a Packet using a Pattern

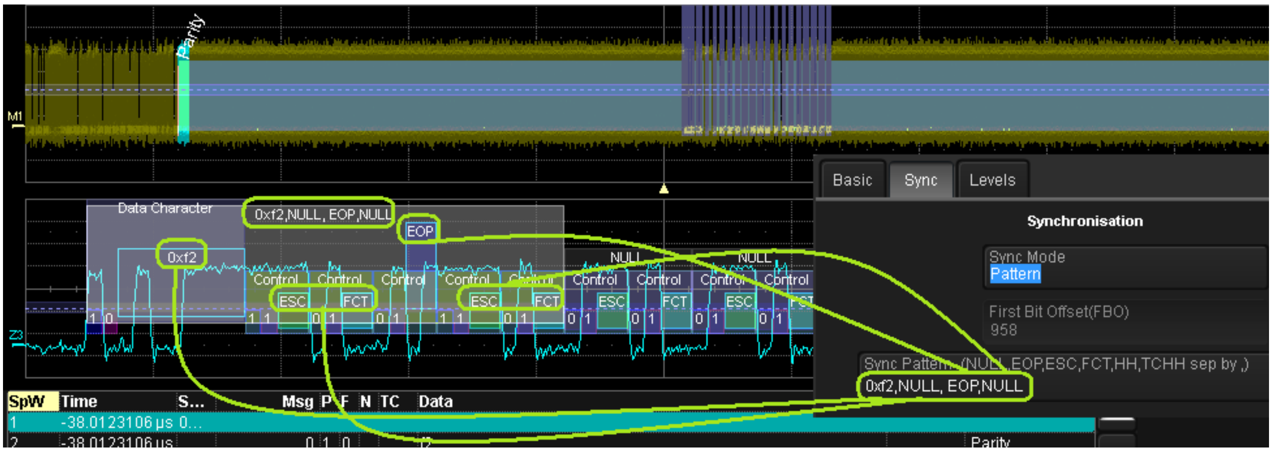


Figure 16: Synchronizing on a mix of L char and N-chars.

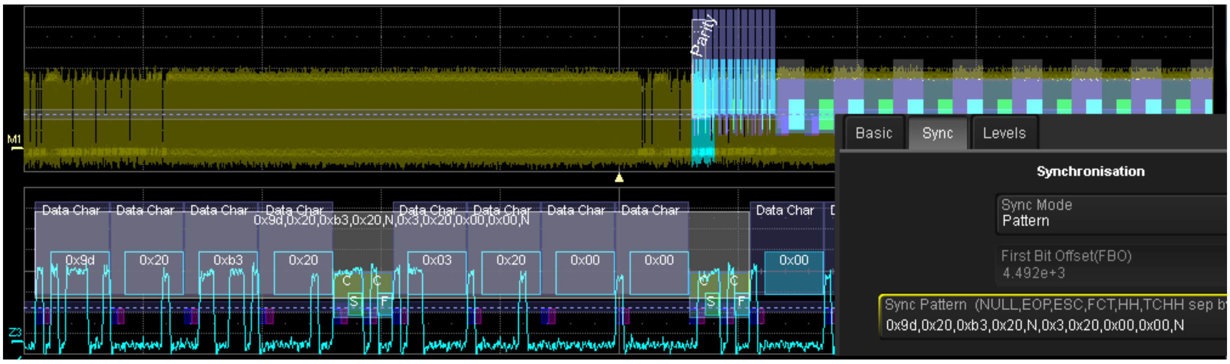


Figure 17: Synchronizing on a complex sequence

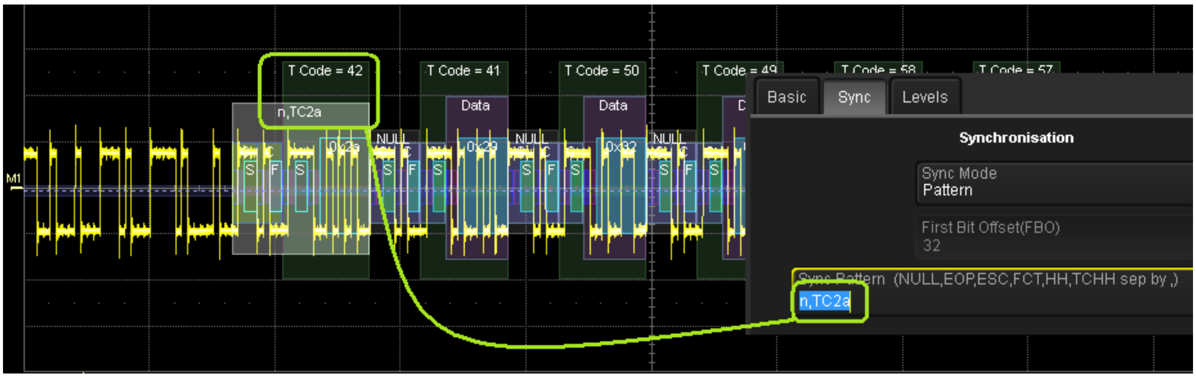


Figure 18: Synchronizing on a Time Code

The valid symbols for Synchronizing are shown in the table below

Symbol	Syntax details	Comments
NULL character	NULL or null or N or n	
ESC code	ESC or esc	
FCT code	FCT or fct	
EOP code	EOP	
Data value	0xHH	2 hex digits preceded by 0x
Time Code	TCHH	2 hex digits, no 0x

Figure 19: Table of legal Synchronization Symbols

Synchronization using the First Bit Offset

In this mode, also called "Brute Force" the user enters the index of the bits onto which to start synchronizations. This mode is used for debug purposes or when the bit stream has low level errors and all higher modes fail. When switching to Manual Mode from the example above and modifying the FBO from 32 to 24 (arbitrary value) we observe a Table populated with Parity errors.

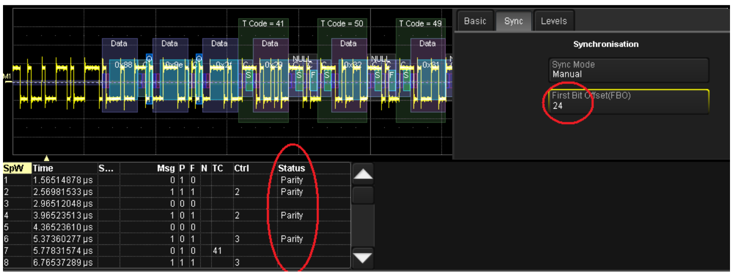


Figure 20: Manual Sync Mode, with Errors

Zoom and Search in N/L Char Mode

When a zoom is open and connected to a trace (mother trace), it also allows searching the mother trace for the occurrence of certain values. The following document explains the search concepts common to all protocols: http://www.lahniss.com/_ufaq/searchusersmanualv5.pdf

In the particular case of SpaceWire, in N/L char decode mode the following columns are available for searching:

Figure 21: Available Search Columns in N/L Char Mode

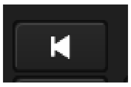
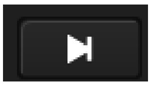
One of the most common column to be searched for is the Data column. The image below shows the details of a search for Data char 0x26, in a record of 1 millisecond stored in M1. The Value to be entered is in hexadecimal. When the search buttons in left  or right direction  are pushed, the zoom jumps to the next occurrence of char 0x26 in the record. Simultaneously, the highlighted row in the table moves up or down and the Zoom trace slides right or left. Note that it is also possible to jump to the first or the last occurrence of 0x26 in the record, or automatically play from one to the next.

Figure 22: Example of N/L char Search for 0x26

Decoding the Space Wire Stream as Packets

Usage context of Packet Mode

The Packet Mode provides a more synthetic view of the stream. SpaceWire applications using packet size anywhere between a few characters and a few hundred characters can be rapidly overviewed. Ab contrario, applications without Packets or with very large packets will not benefit from this mode. At the time of this writing, the distribution of applications is not really known, and the Packet mode is provided as a tool that can help part of the community.

Switching to Packet Mode

Packet mode is accessed by switching the Table Mode to Packets while leaving all other parameters unchanged. The Table and the annotations will react. An example follows

Interpreting the Packets

In Packet mode each line of the table contains a full packet. It is therefore easier to compare Packets, search or otherwise get an overview of the Packets compositions. The Packets are completely decoded, but the Table Line showing the characters on one line will be clipped on the right since in most case a Packet will contain too many data bytes to fit within the window. There might be exceptions when the data contains short commands, or short data from sensors emitting i.e. positional data or pitch, roll and yaw data.

The following large image shows a complete screen, with mother trace, zoom, table and dialog. The stream is constructed of roughly 20 Packets, each of which contains about 80 N-chars separated by NULLs at every 4 N-Chars.

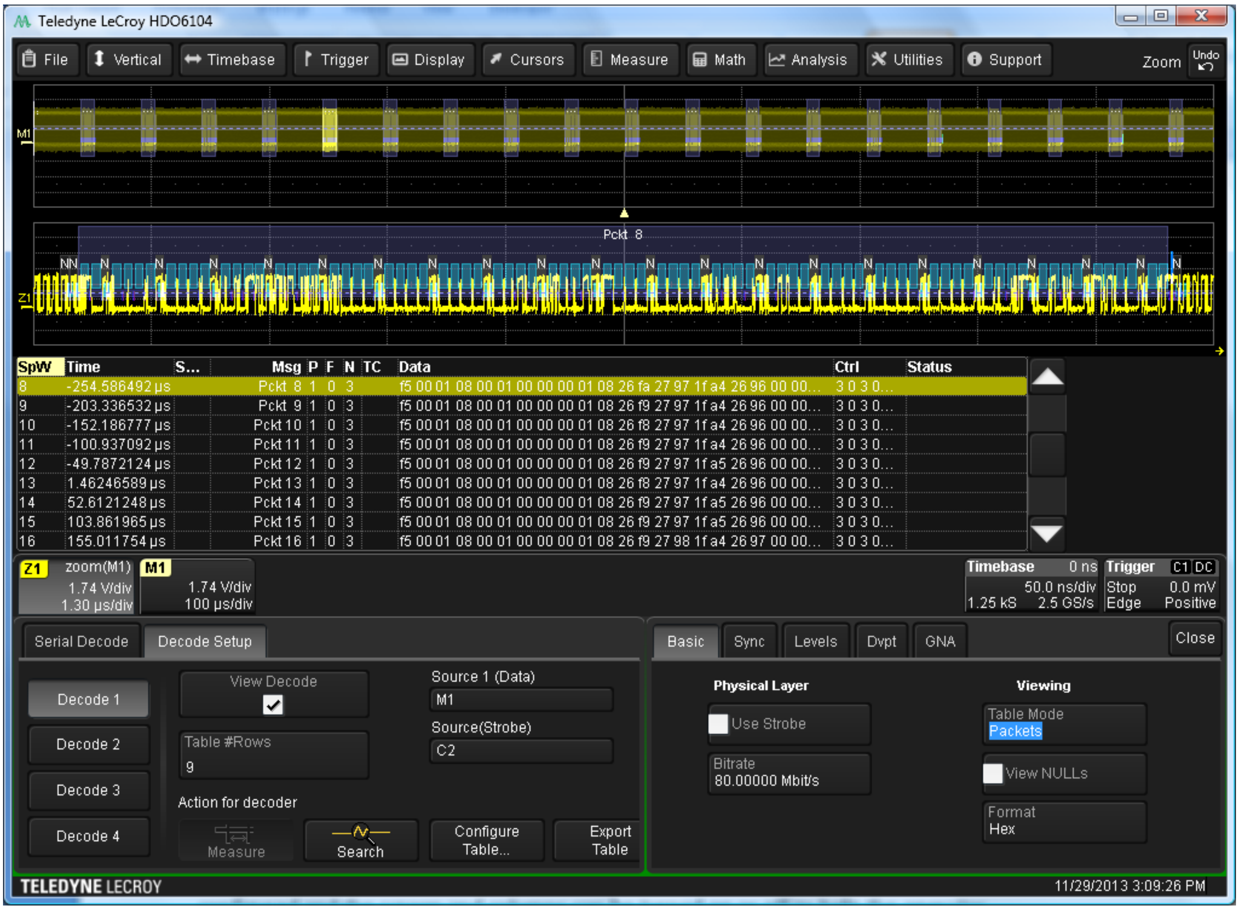


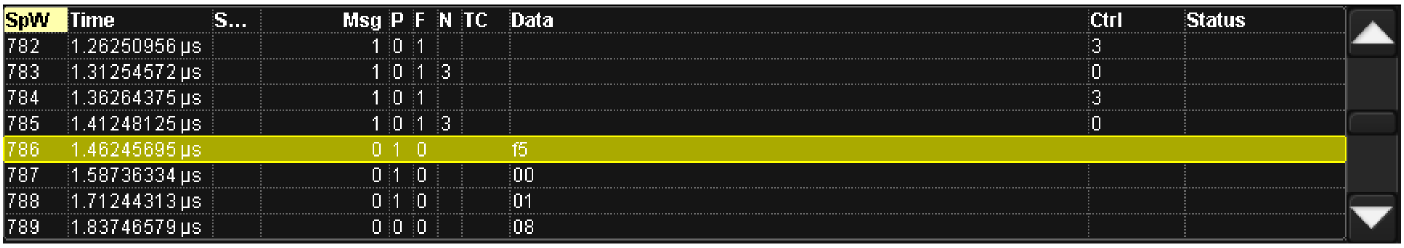
Figure 23: Complete image of decoding in Packet Mode.

Zoom and Search in Packet Mode

The Zoom and Search in Packet Mode is very similar to the N/L-Char mode and will not be explained in detail here. Please refer to the above.

Columns Contents of the Decode Table

The table below explains every column of the table, and its meaning. Note that the table can be configured and the screen and columns can be turned on or off to help the operator.



SpW	Time	S...	Msg	P	F	N	TC	Data	Ctrl	Status
782	1.26250956 µs			1	0	1			3	
783	1.31254572 µs			1	0	1	3		0	
784	1.36264375 µs			1	0	1			3	
785	1.41248125 µs			1	0	1	3		0	
786	1.46245695 µs			0	1	0		f5		
787	1.58736334 µs			0	1	0		00		
788	1.71244313 µs			0	1	0		01		
789	1.83746579 µs			0	0	0		08		

Figure 24: Columns of the SpaceWire decoder

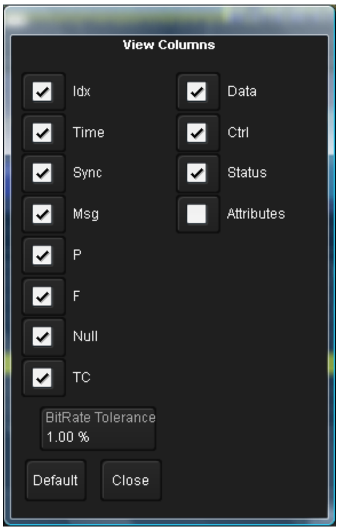


Figure 25: SpaceWire Table Column Picker

As all the other settings in the scope the table can be configured remotely for Automated Test Equipment (ATE) on large test setups. The screen visibility of the column also drives the Export of the Table to a file. The table is always exported WYSIWYG (What you see is what you get) to the CSV file. Also refer to the General Serial Decode Manual more details on the export.

Columns	Meaning of the columns contents
Idx	Index of the line in the table, and also number of the message in the annotation. Depending on the Table Mode used, a line represents either a SpaceWire Bits, a N/L-Chars or a Packet
Time	Time of the beginning of the SpaceWire Bits, N/L-Chars or Packets, with respect to the trigger point of the record.
Sync	Materializes the synchronization point resulting from the users settings in the Sync tab. The synchronizing condition appears as a grey rectangle around the user selected synchronizing pattern. See also the section dedicated to the Sync sequence.
Msg	Only appears in N/L-Chars mode and Packet modes and shows messages pertaining to the decode mode at hand.
P(arity)	Only appears in N/L-Chars mode and Packet modes and shows the parity of the characters. Note that in Packet Mode, all of the Parities of the Chars in the Packet are stacked in the same cell and therefore invisible to the user, but necessary to the algorithms.
Flag	Only appears in N/L-Chars mode and Packet modes and shows the type of Char, Control or Data
Null	Auxiliary columns for NULL handling
TC	Time Codes appear in this column whenever present in the record.
Data	The raw value of the Data characters (0 to 255)
Ctrl	The raw value of the Control characters (0 to 3)
Status	Error and Warning reporting column.
Attributes	Utility column, invisible to the user.

Table2: List of Columns in the table

Note: The BitRate Tolerance shown in the Picker is not used by the SpaceWire decoder.

APPENDIX A: Using Level and Hysteresis for difficult signals

There are cases where a poor signal still needs to be decoded. This occurs when everything else fails, normal equipment's no longer communicate, and protocol analyzers have bailed out with unexplainable errors.

A common case is the identification of a bad node on a multi node bus, such as CAN or MIL-1553. In this situation the faulty node appears as generating poor packets on the bus, which is visible even without any decoding tool. The next question is: "What is the ID or address of the offending node or slave?" Because of the signal corruption normal tools will refuse to decode the poor packet.

However, in this situation, the oscilloscope, with its combination of low and high level tools can be of great help. As the mental model for these operations is identical from protocol to protocol, a detailed explanation is provided on the ARINC 429 Lahniss Web Page

http://www.lahniss.com/_pARINC429/ARINC429.html

Items 8, 9, 10 and 11 of the second table explain how to use the Level and Hysteresis controls to overcome widely out of specification signals. This page also points at other tools, (item 11) such as filters pre and post processing filters that are part of any normal oscilloscope.

The normal work sequence for identifying a faulty bus participant usually starts with a manual acquisition of bus traffic where the fault appears in the oscilloscope capture window. The rest of the work occurs with the acquisition stopped, so that the trace contents remains unchanged. The trace to be studied can also be stored in Memories or TRC files to be recalled later or distributed among participants.

Once the trace is captured and backed-up, the identification work can start using the tools described above.

APPENDIX B: Using the NRZ pattern trigger for SpaceWire

Some LeCroy high End machines offer, as an option, a High Speed NRZ pattern trigger. In some cases this trigger can be put to use to compensate the lack of a dedicated SpW trigger. It is sometimes useful to trigger on the Packet ID at the beginning of a Packet, before the Cargo. The pattern needed to achieve a stable trigger is best composed of a NULL followed by the Address.

There is however a significant point to be explained. The hardware pattern trigger operates in a range from 150 Mb/s to 3 Gb/s. Therefore when SpaceWire runs at speeds below 150 Mb/s, the trigger is too fast. However this limitation can be circumnavigated by using the following procedure. Find the lowest possible integer multiple of the bitrate that is achievable by the trigger. Replicate each bit in the desired trigger pattern by the same integer multiple. As an example if the stream is at 40 Mb/s, the lowest possible multiple would be 160 Mb/s, therefore a multiple of 4. The 80 bit pattern is now reduced to 20 bits at 40 Mb/s. So if the desired pattern to trigger is NULL, 0x02 the signal would look like this image

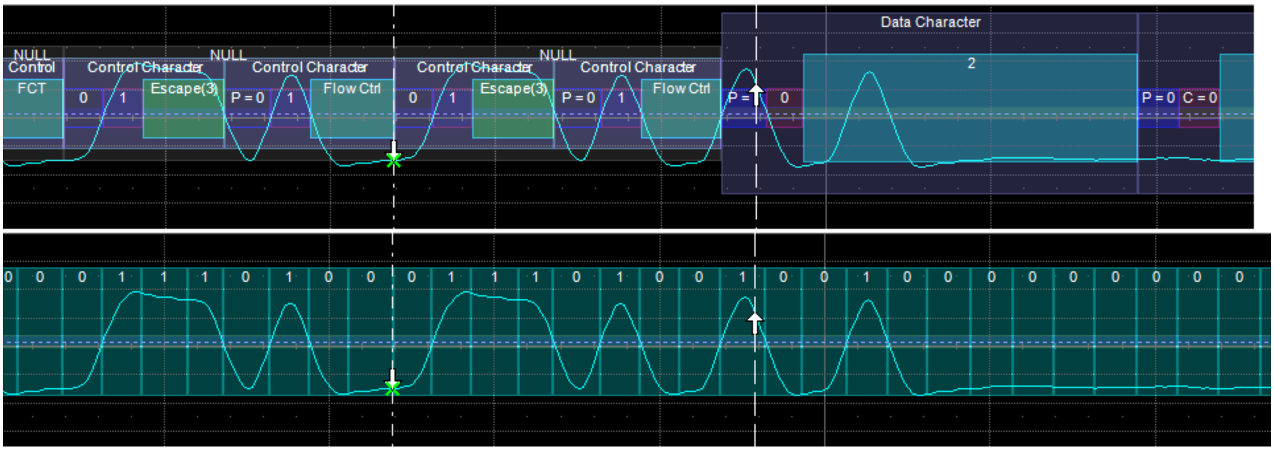


Figure 26: NRZ trigger example

The necessary trigger pattern before expansion would be:

01110100 1001000000   (NULL, 0x02)

And after expansion by 4 to reach 40 Mb/s

0000111111111111000011110000000 11110000000011110000000000000000000000000

The above patter is 72 bits long, therefor well supported by the 80 bit pattern length.

The trigger specifications are the following

High-speed Serial Protocol Triggering
Data Rates 150 Mb/s–3 Gb/s
Pattern Length 80-bits, NRZ or 8b/10b
Clock and Data Outputs Not available
Clock Recovery Jitter 1 psrms + 0.3% Unit Interval RMS for PRBS data patterns with 50% transition density
Hardware Clock Recovery Loop BW PLL Loop BW = Fbaud/5500, 100 Mb/s to 2.488 Gb/s (typical)