SENT Decoder, Documentation for Release 7.8

Lahniss Sarl, April 2017

Table of Contents

List of Figures

Introductions

Introduction (English)

- The SENT Decoder for LeCroy oscilloscopes supports SENT stream emitted by various sensors, both in the 2008 and the 2010 SENT format, as well as the upcoming 2015 version.
- This document relies on the assumption that the reader is familiar with Teledyne LeCroy oscilloscopes in general, and also assumes some familiarity with the SENT standard published by SAE.
- Experience shows that readers familiar with NRZ or Manchester coding need to adjust to the Pulse width coding technique used by SENT to transmit the Fast Channel Information. The Slow channels require one more abstraction level: the distribution of bits over several SENT messages.
- The use of the word “Channels” is confusing in the combined context of the oscilloscope and the SENT decoder. On oscilloscopes, **channels normally refer to the signal input channels**, often called C1 through C4. The SENT protocol on the other hand uses **Fast and Slow channels** to designate 2 different streams of information carried by the SENT messages. We hope that the document will provide sufficient contextual clues to always allow the distinction.
- The following material will guide the operator through every step of the process.
- Some recent features in the decoder reflect concepts under discussion in various SENT communities and are not yet documented in the SAE specifications.
- The SENT SPC format, generated outside the SAE SENT committee is also supported.

Introduction (Français)

- Le décodeur SENT permet l’interprétation complète de l’ensemble des informations transportées par le flux continus de messages émanant de nombreux capteurs, autant dans sa version 2008 que 2010 et de la future version 2015 en préparation.
- La forme et le fonds de ce document partent du principe que le lecteur est raisonnablement familier avec les oscilloscopes de Teledyne LeCroy ainsi que la spécification du protocole publiée par la SAE.
- L’expérience montre que les utilisateurs familiers des encodages Manchester et NRZ doivent ajuster leur modèle mental à la technique en modulation de longueur d’impulsion utilisée par SENT. Par ailleurs le mode de distribution des informations des canaux lents requière une abstraction supplémentaire, due au fait que l’information des canaux lents est distribuée sur 16 ou 18 messages rapides.
- L’utilisation du terme « Canaux » est déroutante dans le contexte de l’oscilloscope. En effet, sur les oscilloscopes « Canal » désigne une entrée analogique de l’appareil, souvent nommées C1 à C4, à laquelle est connectée un signal électrique. En revanche SENT utilise « Canal »pour le concept des flux d’information (rapides et lents) transportés par les messages. Nous espérons que la rédaction du texte contient suffisamment d’information contextuelle pour permettre la distinction.
- Ce manuel guide l’utilisateur pas à pas à travers chaque étape du processus, pour l’ensemble des fonctionnalités des canaux rapides, des canaux lents et des calculs en aval utilisant MessageToValue et ColumnsToValue.
- Certaines fonctionnalités du décodeur sont en avance sur la spécification officielle, mais répondent à des besoins réels de certains constructeurs.
- Le format SENT SPC, élaboré en dehors du comité de la SAE est aussi supporté.

Einführung (Deutsch)

- Der SENT- Decoder erlaubt eine vollständige Interpretation des Datenstroms eines SENT- Sensors. Die Versionen 2008, 2010 und 2016 der Spezifikation werden unterstützt.
- Diese Anleitung beruht auf der Annahme, dass der Leser Teledyne LeCroy Oszilloskopen bereits kennt, und dass er die SAE- Spezifikation des SENT- Protokolls grundsätzlich beherrscht.
- Die Erfahrung zeigt, dass Ingenieure, die mit NRZ- und/ oder Manchester- Kodierung vertraut sind, etwas umdenken müssen um mit der Puls- Breiten- Codierung von SENT zurecht zu kommen. Eine zusätzliche Abstraktionsstufe bieten die Slow- Channels an, die auf der Basis von 16 oder 18 bit serielle Daten verteilen. Die graphische Darstellung (der einzelnen Bits bzw. der gesamten Information) durch den Dekoder dürfte aber das Verständnis dieser Methode erleichtern.
- Dieses Dokument ist eine stufenweise Einführung zum SENT- Dekoder und eine umfassende Beschreibung seiner Funktionalität.
- Erfahrene Benutzer können natürlich auch direkt zu den gewünschten Erläuterungen gehen.
- Die Einstellung des Decoders erfolgt allmählich, beginnend mit der richtigen Paketierung der Daten in SENT- Botschaften, gefolgt von der korrekten Identifikation der Nibbles und danach der Datenwörter. Nur die korrekte Einstellung der Nibble- Dekodierung ermöglicht die richtige Wörter- Berechnung.
- Das Wort “Channel” ist verwirrend im Zusammenhang der Beschreibung des Oszilloskopes und des SENT- Decoders. Das Wort “Channel” (Kanal) auf dem Oszilloskopen bezieht sich auf die Eingangskanäle des Gerätes, oft auch mit C1 bis C4 beschrieben, die mit einem elektrischen Signal verknüpft sind. Das SENT- Protokoll verwendet das Wort „Channel“ für zwei verschiedene Datenpakete (die Fast- und Slow- Channels), die in den SENT- Botschaften enthalten sind. Der Verfasser hofft, dass dies damit ausreichend erklärt ist (siehe auch detaillierte Beschreibung).
- Dieses Dokument führt den Anwender Schritt für Schritt von der basierenden Nibble SENT- Dekodierung zur graphischen Aufzeichnung der Wörter mit Hilfe der Parameter MessageToValue und ColumnsToValue.
- Das SENT- SPC- Format arbeitet außerhalb des SENT- Komitees, wird jedoch unterstützt. SENT-SPC beruht auf einer Abfrage des Sensors beim Controller, während SENT auf einem kontinuierlichen Sensor- Signal beruht, mit der Annahme, dass der Controller schnell genug ist, um den Informationsstrom zu bearbeiten.
- Anhang A und B enthalten Dekodierung und Einstellungsbeispiele für häufige Fälle.
- Anhang C bietet Erklärungen zum Pegel und zur Hysteresen Einstellung.
- Anhang D erläutert die Darstellung der Sensor- Werte im Zeitverlauf mit Hilfe von zusätzlichen Funktionen des Oszilloskopes.
- Anhang E schlägt einige Verfahren zur Beschleunigung der Berechnungen vor.

Getting started

As we start, we need to **emphasize the methodology** underlying the software described here. The fundamental model is a 2 step model. The signal needs to be decoded as:

- Each Burst needs to be sliced into **Nibbles**
- Then the nibbles can be converted to **Word(s), therefore constituting the fast channels**
- Then the Slow channels can be decoded, and the SCDF file constituted.

The **Words** extraction will not function correctly until the **Nibbles** are properly decoded. The tuning of the Decoder must therefore be conducted in this order:

Nibbles → Words.

In order to get started with the SENT Decoder, it is advisable to adjust the scope controls to acquire 5 to 8 Burst or Frame of relevant Data, and then **stop the acquisition**.

Once the SENT messages are acquired, we will proceed to its interpretation using the SENT Decoder. We will proceed gradually; starting with the identification of the Burst controls, then extract the Nibbles and finally the Words, by grouping nibbles 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 familiarized with the User Interface of the Decoder.

Firstly, in the Serial Decode dialog, you need to select the signal source (here Memory 1), and the Protocol, “SENT” in this case.

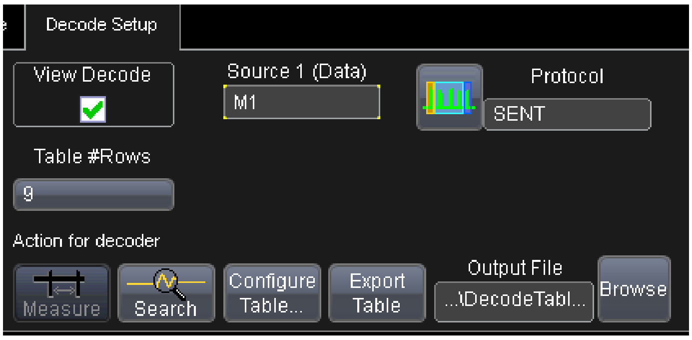


Figure 1: The selection of SENT in the Decode Setup

Once the “SENT” Protocol has been selected 2 to 4 tabs will appear in the Right Hand Side Dialog. We will be setting various values in these 3 tabs: **Basic, Fast Ch, Slow Ch and Levels**

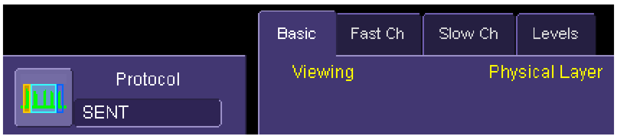


Figure 2: The protocol selection governs the appearance of the Right Hand Side tabs.

The Right Hand Side tabs composition is governed by selections in the Basic tab, as explained below.

When decoding by Nibbles, the tabs only have one possible incarnation.

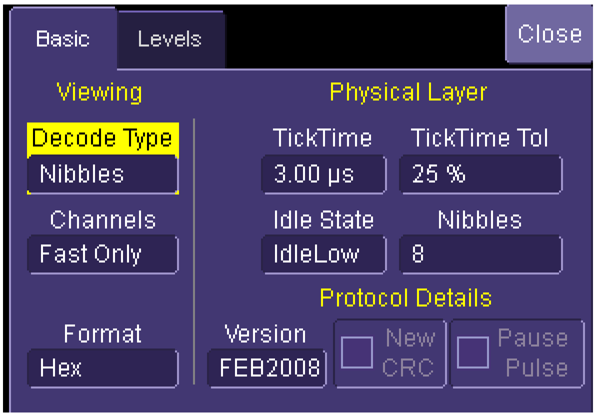


Figure 3 When Decode Type is Nibble, only the Basic and Levels tabs are required

When decoding as Words, the selection of tables depends on the Channels selection.

Figure 4 When either Fast Only or Slow Only is selected, the corresponding tab will appear

When decoding as Words, Channels=Both, all of the 4 tabs will be presented.

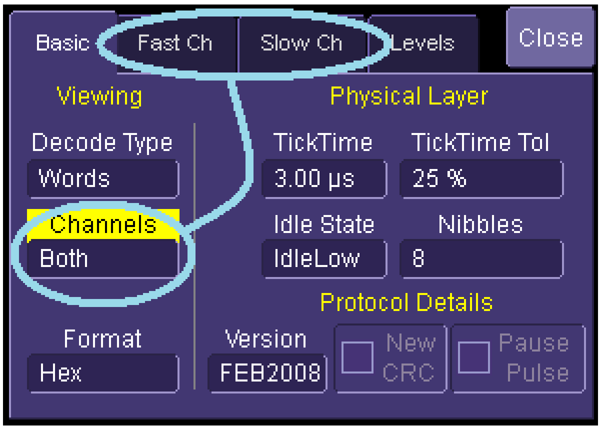


Figure 5 When Channels = Both are selected, Fast and Slow tabs will be shown

When working on a given signal, some of the values in the tabs will not change anymore because they are strongly linked to the signal (i.e. Tick Time or Idle State). Other values can or will have to be tuned to obtain optimal results, or understand the reasons for misbehaviors.

Decoding Messages as raw Nibbles (Physical Layer)

As its name indicates, SENT is based on Nibbles and Nibble length. Therefore the first level of decoding is the nibbles.

In order to start adjusting the decoder, it is best to select the Decode Type "Nibbles" In this mode every SENT Packet will appear, with:

- its SYNC (also named CAL) pulse at the beginning.
- its 5 to 8 constituting nibbles, labeled with values ranging from 0 through 15, depending on the pulse length, as per SENT specifications.
- the Inter Frame Gap (IFG) between the packets .

Verifying the correct settings of the fundamental parameters

Every complete packet in the trace should be decoded, the Tck values in the table should be coherent with the TickTime set in the Basic Tab, and the IFG times should be within specifications. In this mode, it is easy to visually verify that every packet is correctly identified and outlined. There might be exception on the first and last packet of the trace, close to the edge of the grid. The following figure shows an example of correct decoding

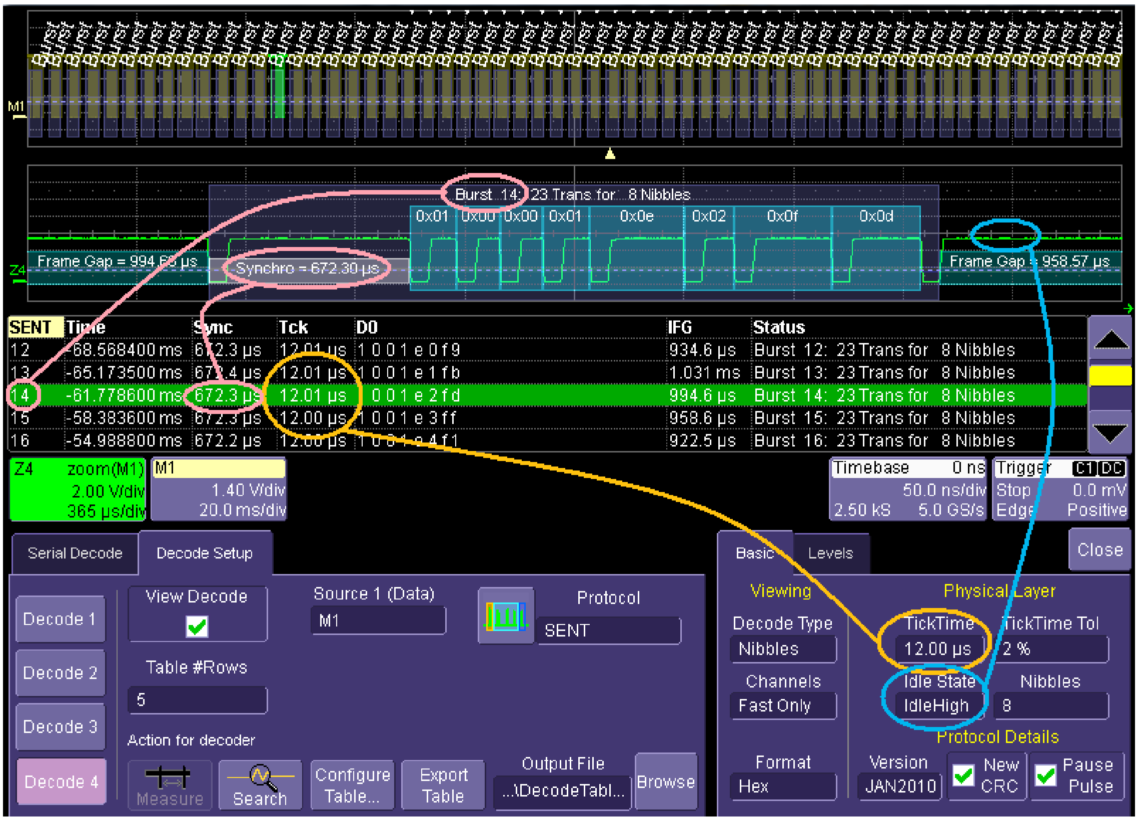


Figure 6 Example of correct Nibbles decoding with Mother Trace, Zoom and Basic Tab

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), this procedure does not need to be repeated. When measuring on different sensors of the same type, and parameterized in the same way, the same settings can usually be re-used. However, the SENT specification allows large variations of the TickTime. So it is possible that similar sensors built over the years from successive fabrication batches and wafers differ from one another.

Using the zoom to rapidly verify the initial decoding

The zoom 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 make sure the fundamental parameters (TickTime, Polarity and Levels), 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 Fig shows the zoom of line 21 (of the table) in Z4. The zoom is a precious tool when studying a decoded trace because every packet can be rapidly analyzed. When clicking on the Zoom descriptor, the Zoom and Search controls appear as below.

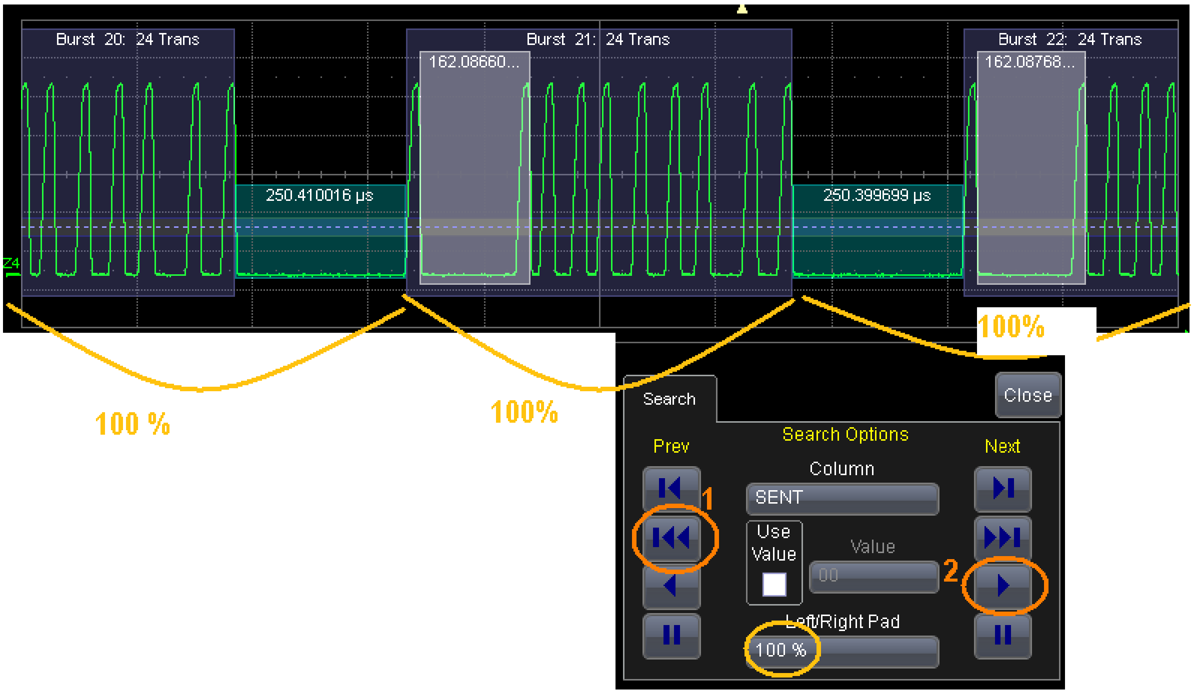


Figure 7 How to "Play" through every decoded packet of the record

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 records 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. For the time being we are not looking at the nibbles values at all.

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 and version.

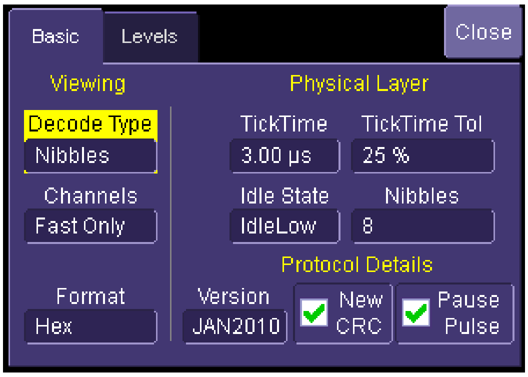


Figure 8: The Basic Tab and its Controls

UI control	Function	Range	Default
	Viewing		
Decode Type	Changes the way the signal is decoded, in terms of details and depth of decoding. It is recommended to start with Nibbles, then move on to Words when needed.	Nibbles, and Words	Nibbles
Channels	Selection of the SENT channels to be processed.	Fast Only, Slow Only or both	Fast Only
Format	Viewing format of the Nibble values and D0 through D3 data values.	Hexadecimal or Decimal	Hexadecimal
	Physical Layer		
Tick Time	Controls the SENT Ticktime, which is the Time increment between a nibble of value N and a nibble of value N+1	400 ns to 3 ms	3 us
Tick Time Tolerance	Defines how the CAL pulse is filtered with respect to the Ticktime indicated by the user. i.e. if a tick Time of 3us is set, with a tolerance of 10% the CAL pulse is expected to be 56 * 3 us +- 10%, therefore 168 us +- 10%	1 % to 30%	10%
Idle State	Defines where the idle state lies, therefore opposite of pulse direction.	High or Low	High
Nibbles	The number of nibbles contained in a single SENT message, including Status, Data and CRC nibbles.	5 or 8	8
	Protocol Details		
SENT Version	Selection between the 2008 and 2010 Version. In the 2008 Version CRC and Pause Pulse are turned off and grayed out since they are not supported. In the	2008 or 2010	2010

	2010 Version both controls are available, and turned on by default.		
New CRC	When checked, the CRC computation will be performed as per 2010 recommended implementation under 5.4.2.2. Otherwise it will follow the 2008 guidelines, under 5.4.2.1 (Legacy). Note that the CRC computation is only active in Word mode.	On/off	On (recommended 2010 implementation)
Pause Pulse	When checked, algorithm expects a Pause pulse as per 2010 definition under 5.2.6. The Pause Pulse follows the CRC of message N and precedes the CAL pulse of message N+1	On/Off	On (with Pause Pulse)

Table 1: List of Controls in the Basic Tab

Setting the correct transition level in the Level Tab

The last tab of the decoder controls the levels used for determining the edge crossings of the SENT signal. The default settings of Percent level = 25% 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.

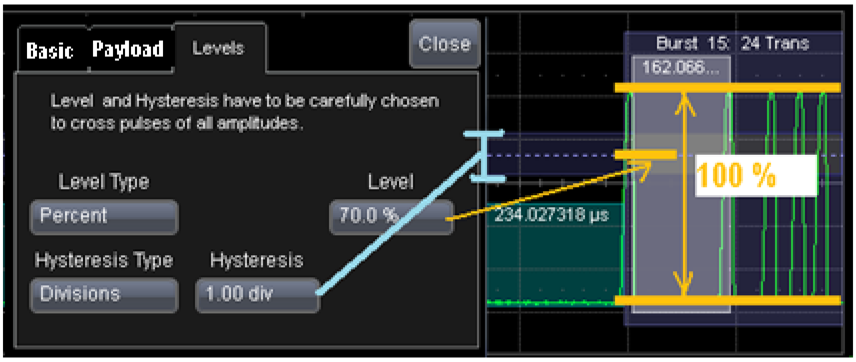


Figure 9: The Level controls and there visualization on trace

Deeper level of understanding

This nibble decoding allows another set of validations. The D0 column contains the nibble values. When wrong nibble values appear (less than 0 or > 15) just click on the line to examine the faulty packet. When faulty data appears, the reason could be wrong decoder settings, but it could also reveal really pathological signals. This is usually when the oscilloscope is the most useful tool, since the signal shape can be examined in great details, using the normal toolbox provided by the instrument (zooms, cursors, parameters, functions, etc.)

Nibbles
088a0270
4889037e
88890470
08890572
08890674
08880772
08880871
08880973
08880a75
08870b71
08870c7f

Figure 10: The “Nibble” column shows the raw nibble values, here in Decimal, for 8 nibbles.

The image above shows an example of healthy decoding. Note that users of oscilloscopes with an orientable screen can rotate the screen and expand the table to the maximum number of lines to have a better overview of the complete decode.

Decoding Fast Channels as Words (Data Link Layer)

Once the decoding as nibbles yields satisfying result, the decoding as Words can be switched on by simply switching the Decode Type to “Words”. The SENT packets will be further analyzed to interpret the nibbles in the context of the SENT protocol structure.

For the sake of clarity we will switch on the Word mode and show a SENT message with only its infrastructure nibbles, SYNC and Pause Pulse.

The first nibble represents the Status (red) and the last nibble the CRC (blue) whilst the remaining nibbles in the middle will carry the payload of the SENT message. For the sake of explanation a message is shown here, only with its infrastructure nibbles.

Figure 11: SENT message #20 decoded with its Status and CRC nibbles only, no Fast Channel used

The same message, with the payload nibbles interpreted as words, appears in the following commented image, also showing the setting of the Fast Channel. This dialog drives the interpretation of all the nibbles, except the Status and CRC nibbles.

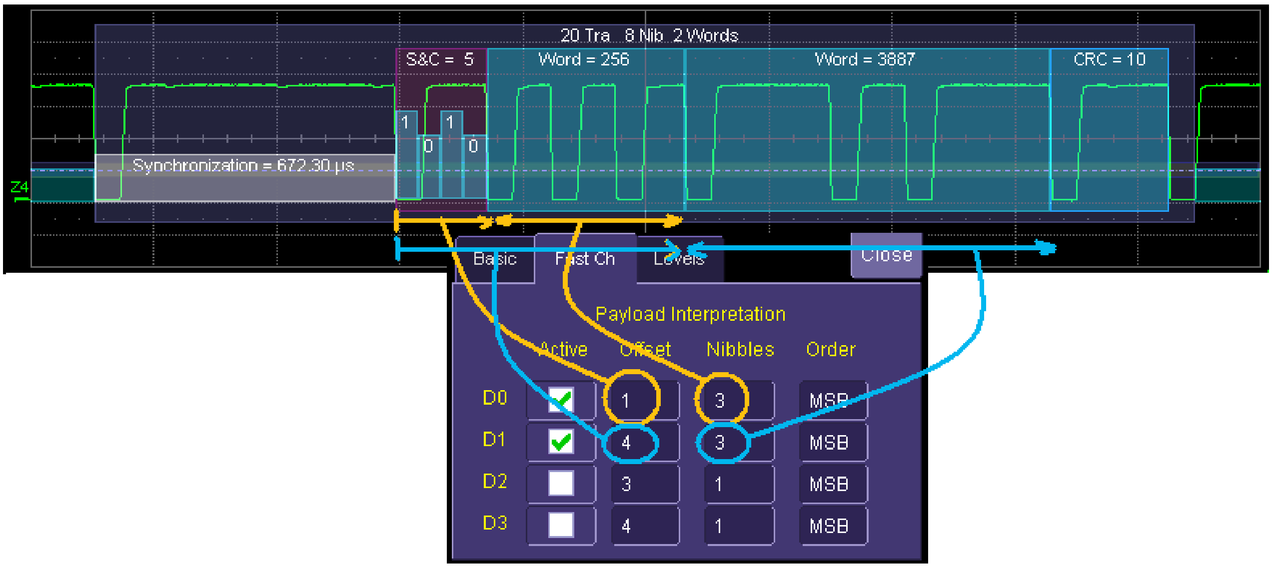


Figure 12: Same SENT message #20 with Payload Words interpreted, as 2x3 nibbles

The Fast Channel dialog allows the user to select the grouping of the nibbles (4 bits) into 4, 8, 12,16,20,24 bit words. This information transportation method helps carrying many bits with a minimum number of transitions. As shown in the dialog, the user is free to select any offset and any number of nibbles to form the words. Some sensors broadcast information as 2 x 12 bits for Pressure and temperature. Other sensors emit Pressure with 12 bits, a running counter with 8 bits and the invert of the most significant Nibble with 4 bits. Other sensors might emit only 16 or 24 bit value. Any of these combinations can be accommodated by the algorithm.

Furthermore, the CRC nibble validates the encoding/decoding of the Frame. Should a Frame have a bad CRC, its line in the table would present a warning message “CRC error”. A bad CRC can have many reasons originating in the signal transmitter hardware or software, the transmission lines or the receiver.

Controls of the Fast Channel Tab (governing the Word decoding)

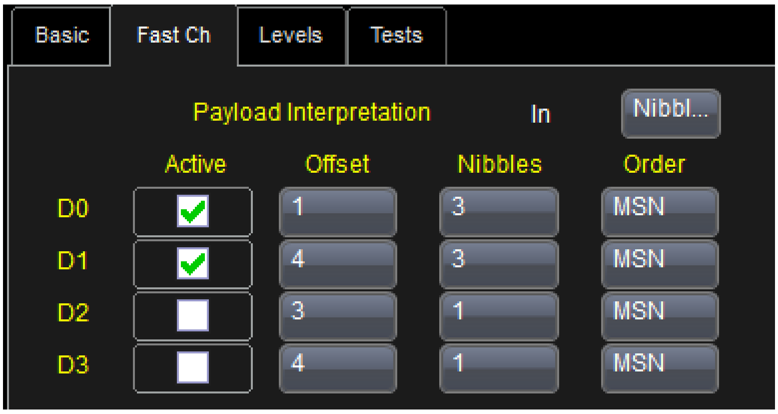


Figure 13: The Fast Channel Tab

UI control	Function	Range	Default
In	Defines the unit used for setting Offset and Span (currently still called nibbles). The units in bit are only used when there is a need to save bits so that more information can be compacted into the relatively short message	Nibbles, Bits	Nibbles
Active	Up to 4 Fast Channels can be defined, independently of one another. Note that the algorithm does not preclude from superposing active channels. This is very useful for secure sensors when the MSN of a fast channel is also transmitted as its opposite in another channel.	On/off	off
Offset	Defines the position of the fast channel in the SENT Payload. An Offset of zero will point at the very first nibble, containing the Status and Communication data.	0 to Nibble Number in Nibble mode 0 to 32 in Bit mode	1,2,3,4
Nibbles/Bits	Defines the number of Nibbles used from the offset to form the fast channel value. Also drives column width in the table, see details below	0 to 6	1
Order	Defines how to combine the several nibbles of a fats channel.	MSN or LSN	MSN

The number of nibbles defined in the dialog drives the width of the DX column, as well as the number of nibbles represented in those columns. The following image shows examples of various nibble width and their impact on the renderings.

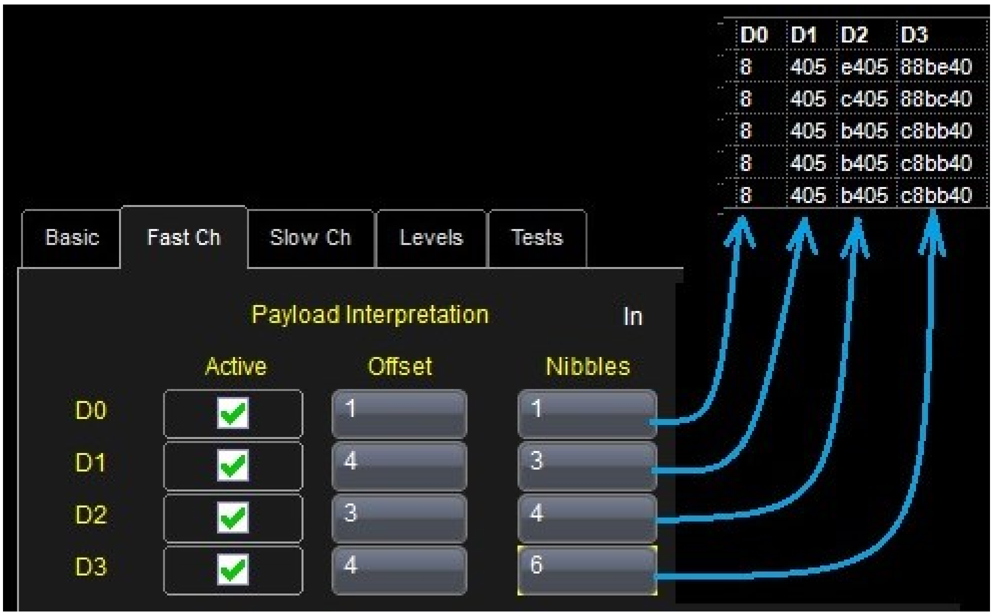


Figure 14: Number of Nibbles action on Table view

Using the Fast Channel tab to verify Secure Sensors decoding

In secure counters, some nibbles can be assigned to a rolling counter. The counter ensures that the receiving ECU never misses a messages, or if a message is missing can be detected. A rolling counter of 2 nibbles, hence 8 bits, would take values of 0 through 255 and revolve back to 0. This ramps lives as long as the SENT sensor is powered. It is similar to a heartbeat in other devices. Any deviation from that behavior is considered as pathological and will raise severe warnings, especially in mission critical sensors such as steering wheel angles, brake pedal angles and altitude sensors.

The verification of this counter occurs via the test tab.

The visualization of the counter can be conducted using a ColumnToValue parameter on the associated Dx columns, followed by a Track of the value.

Numerous examples can be seen under:

http://www.lahniss.com/_p/_psent/sent.shtml

As well as in the SENT Signal Source LAH10x's manuals:

http://www.lahniss.com/_u/_ulah10x/lah10xmanual_v6.pdf

Decoding the Slow Channel (Application Layer)

SENT provides a creative mechanism to broadcast information that was usually to be found in data sheets, specifications and other auxiliary literature, such as conversion coefficients from raw data to physical values, manufacturer's name, type of sensor, calibration data, wafer production references, etc.

This mechanism relies on 2 bits in the Status & Communication Nibble of the SENT message. These bits are fetched from 16 or 18 adjacent fast messages and grouped to form a Message ID, a message Value and a CRC. The grouping of 16 fast messages to form a single slow message is usually referred to as "Short Serial Messages" whereas the grouping over 18 fast messages is referred to as "Enhanced Serial Message Format".

Switching the decoder to view the Slow Channel

The following image shows what happens when switching the Channels to "Slow Only" mode. The annotation is modified and groups messages into chunks of 16 or 18 messages, depending on which version of SENT is used.

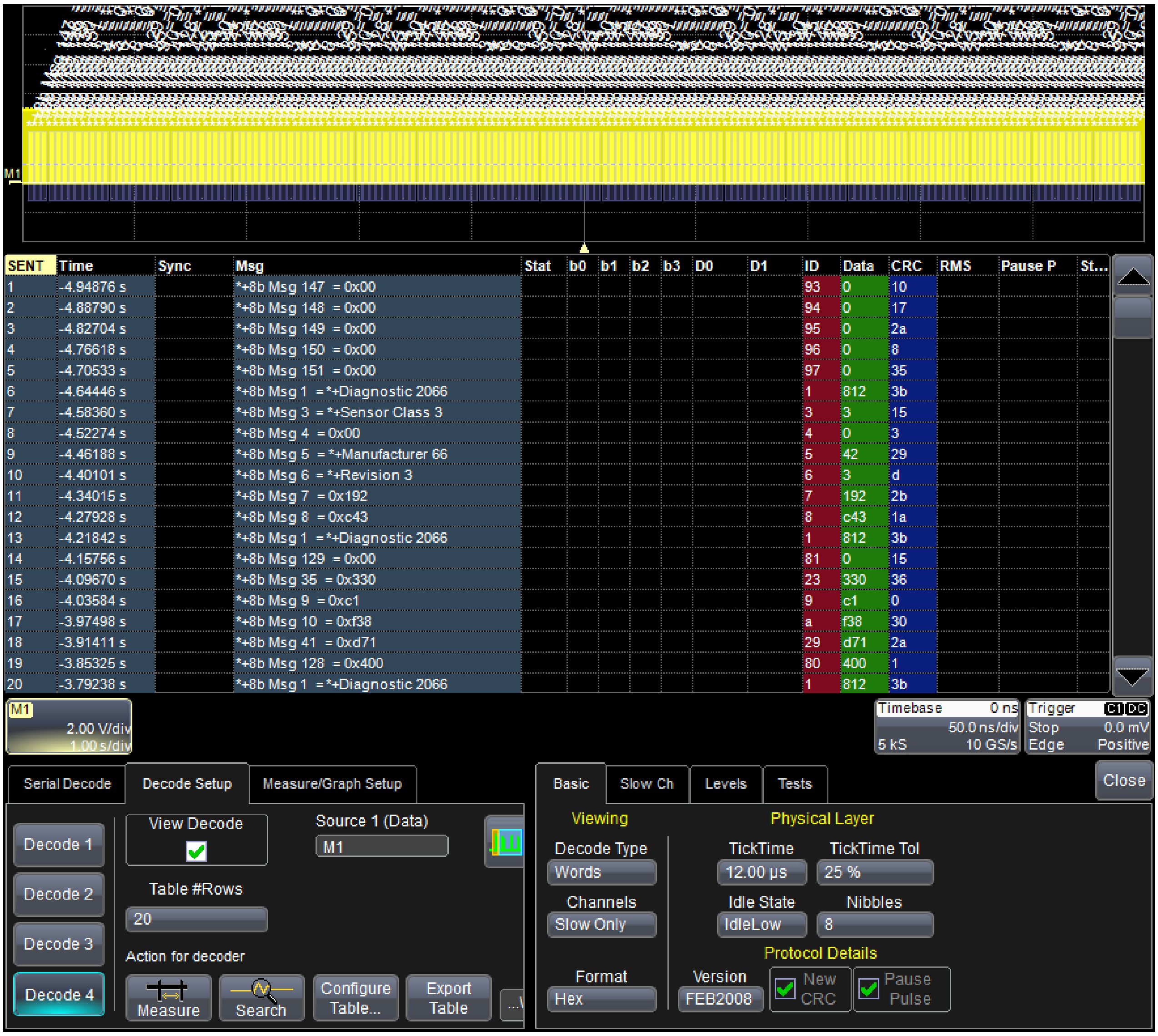


Figure 15 Slow Channel Only Decode, with slow messages IDs and Data columns, no SCDF file in use.

The decode Table dynamically changes its contents to reflect the new situation. The time columns shows the beginning of each slow message. The “Msg” columns show the decoded Name and value of the slow message. Dedicated columns **ID** and **Data** show the contents of the slow message. Other columns are not used in this mode, and can be turned off using the standard column picker found in every decoder.

As visible on the image, names and values of the messages currently set to default texts, and in some cases default value.

We will see in the next section how the default values can be changed and loaded into the decoder by means of a user constructed file.

Before we move on to this file, also note the various colored background of the slow messages (grey, red, green and blue). This will become significant when the Channel Mode “Both” is used, so that we can distinguish the Fast and Slow Messages, and further interpret them.

Creating and Editing the Slow Channel Definition File (SCDF)

We will now explain how to enhance the decoding of the slow messages using the Slow Channel Definition File (SCDF) mechanism. Before we go into details, let’s look at the resulting evolution of the Slow Only decode mode with the appropriate file.

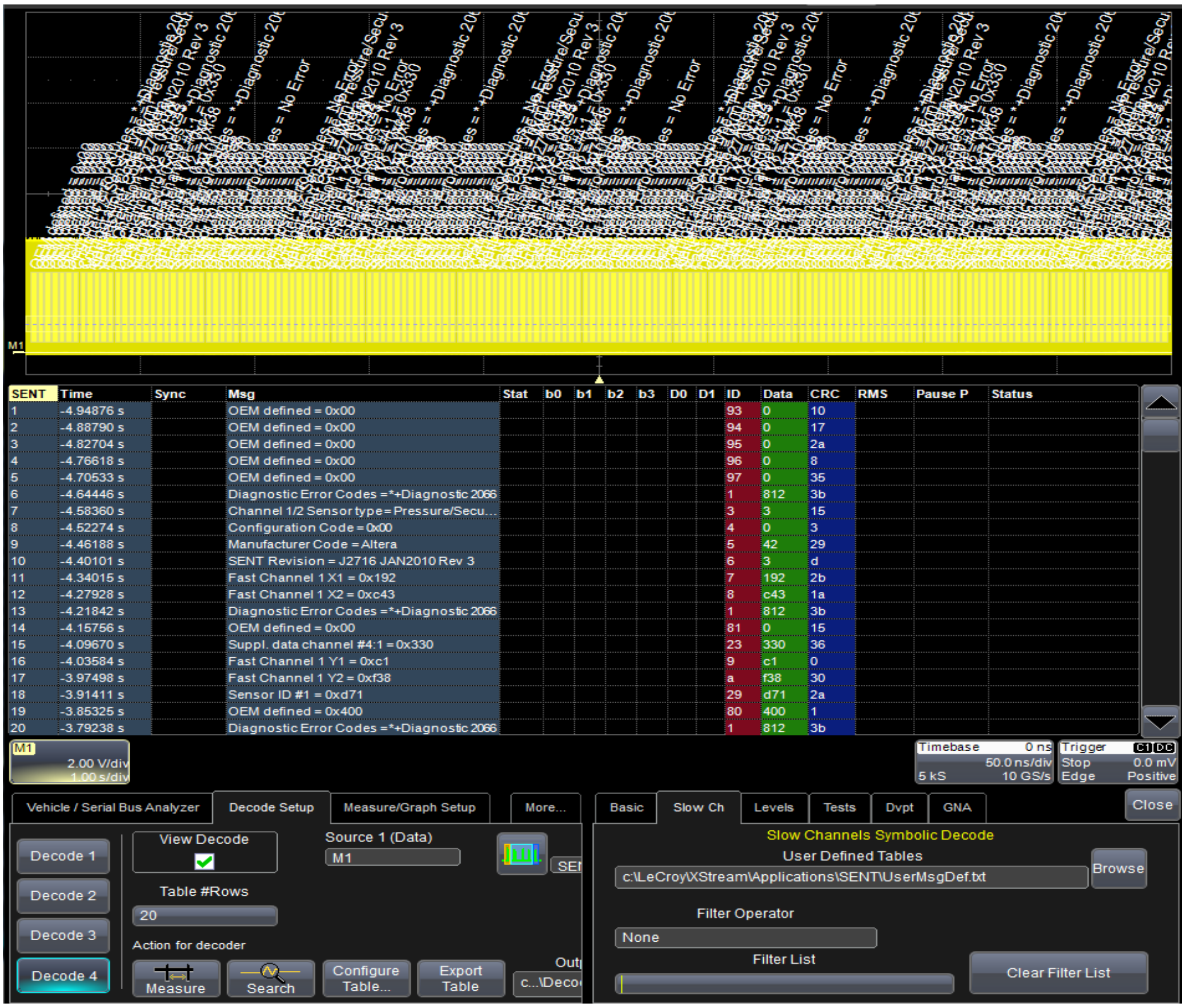


Figure 16 Interpretation of the Slow Messages using the SCDF file, same example as previous Figure

The image shows that the default Message IDs are transformed into real text and some of the values into text (Like "Manufacturer's code = Altera") or values (such as Fast Channel 1 X1 = 0x192).

The image also shows the "Slow Ch" dialog containing the name and path of the SCDF used to interpret the Message Labels, Diagnostic Labels, as well as some other ancillary values such as Manufacturer's codes and SENT revision number. The syntax of the SCDF file is explained in the next section, the slow channel filter in another section.

Structure of the SCDF

The mechanism used to translate IDs and values into text is a simple TXT file containing table definitions. The beginning of the SCDF default file installed on every instrument is shown here in a simple text editor, which is enough to edit the file.

SlowChanDefFile.txt - Notepad

File Edit Format View Help

```
//
//                   Syntax description at the end of the SCDF file
//
// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for 8 bit message ID, max is 254
// ---------------------------------------------------------------------------
Table,SlowChannel8BitMessageID
0,Undefined
1,Diagnostic Code         // interpreted as per user defined Table DiagnosticMessages
2,Undefined
3,Sensor Class            // interpreted as per user defined Table SENTSensorClasses
4,Undefined
5,Manufacturer Code       // interpreted as per user defined Table ManufacturerCodes
6,SENT Rev                // interpreted as per user defined Table SENTRevisionCodes
7-254,OEM defined

// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for 4 bit message ID, max is 15
// ---------------------------------------------------------------------------
Table,SlowChannel4BitMessageID
0-15,User Defined

// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for SENT rev, max is 9
// ---------------------------------------------------------------------------
Table,SENTRevisionCodes
0,Undefined Rev
1,Rev 1
2,Rev 2
3,Rev 3
4-9,Future Revs
```

Figure 17 Default SCDF file provided on every instrument.

The syntax is documented in the file itself, towards the end, as comments, as well as the syntax errors detected during the parsing. Syntax and errors are shown in the images below.

The SCDF syntax defines several reserved names to identify the auxiliary tables used for

- **Table, SlowChannel8BitMessageID**, used to interpret the 8 bit message ID
- **Table, SlowChannel4BitMessageID**, used to interpret the 8 bit message ID
- **Table, SENTRevisionCodes**, used to interpret the value conveyed by message ID 6
- **Table, SENTSensorClasses**, used to interpret the value conveyed by message ID 3
- **Table, DiagnosticMessages**, used to interpret the value conveyed by message ID 1
- **Table, ManufacturerCodes**, used to interpret the value conveyed by message ID 5

The parsing errors are always emitted at the bottom of the oscilloscope screen, with the last error overwriting the previous ones. It is therefore advisable to fix the errors as they occur, beginning with the last emitted error, then reload the file and see if any errors are left.

SlowChanDefFile.txt - Notepad

File Edit Format View Help

```
// --------------------------- SYNTAX EXPLANATION ---------------------------
//
// This file drives the constitution of several tables, whose names are defined in
// the statements of the form:
//
//                     <Table,TableName>
//
// Each statement of this type sets the "Current Table"
//
// Each table line definition consists of a statement of the form:
//
//                     <LineX, Text>
//                     <LineStart-LineEnd, Text>
//
// The first version will fill Line X with "Text" of the "Current Table"
// The second version will fill a Table region from LineStart to LineEnd
// with "Text" of the "Current Table"
//
//
// The "LineX", "LineStart", LineEnd""  keywords must be Decimal or Hexadecimal.
// A Decimal Line value must contain only values: 0123456789
// An Hexadecimal Line value must begin with 0x followed by Hex digits
// Hex digits are: 0123456789abcdefABCDEF
//
// The <LineStart-LineEnd, Text>  syntax allows filling of the table with very few Line
// statements. This feature is used i.e in the default SCDF, or for test purposes.
//
// At boot time all of the tables are filled with default texts beginning with "*+"
//
```

And the parsing errors are listed at the very end of the file. Note that the parsing errors are very useful when beginning to use the SCDF, in order to located syntax errors. Also note that the syntax is very strict, in order to keep the parser as simple as possible.

SlowChanDefFile.txt - Notepad

File Edit Format View Help

```
// At boot time all of the tables are filled with default texts beginning with "*+"
//
// --------------------------- SYNTAX ERRORS EXPLANATION ---------------------
//
// 1-Line Value and Range Syntax Errors
//
//  "Line XX, index YYY out of range (0 to Max Line for Current Table)       i.e.  1234,Some 4 bit ID message
//  "Line XX has more then 1 '-' separator in Line Range: "                  i.e.  1-3-8,User Text
//  "Line XX, Line Range has Start:SS > End:EE"                              i.e.  25-4,User Text
//  "Line XX, line contains non digits on position %d: "                     i.e.  2$3, User Text
//
// 2-General Line Syntax Errors
//
//  "Line XX has CC Commas Instead Of 1"                                     i.e.  21,User Text,,,,
//  "Line XX has No Commas"                                                  i.e.  A Line without commas
//  "Table Line Definition XX Has UU Tokens Instead Of 2"                    i.e.  34,User Text1,User Text2
//  "Line XX, Line Definition exceeds 10 characters"                         i.e.  12345-13345,User Text
//  "Line XX, Message Length YY exceeds 60 characters"                       i.e.  3,Some Very Long User Text
//  "Line XX, Warning,empty Message?"                                        i.e.  3,
//
// 3-Table Selection Syntax Errors
//
//  "Line XX, Warning: Table at YY replaced by User Entry "                  Line is redefined
//  "Line XX defines a table line, but no table is selected"                 Line Definition starts before Table Definition
//  "Line XX, unknown error, rejects all tokens"                             Catch all when Line Parser is overwhelmed
//  "Table Selector Definition %d Has %d Tokens Instead Of 2"                i.e. Table,ManufacturerCodes,Some More Useless Text
//  "Line XX specifies unknown Table"                                        i.e. Table,UnknownTableName
```

Figure 18 Syntax error emitted by the SCDF Parser

Controls of the Slow Channel Tab

The Slow Channel Tab contains one filename picker control, allowing to select the name and path of the SCDF, as well as the Slow Channel Filter Controls

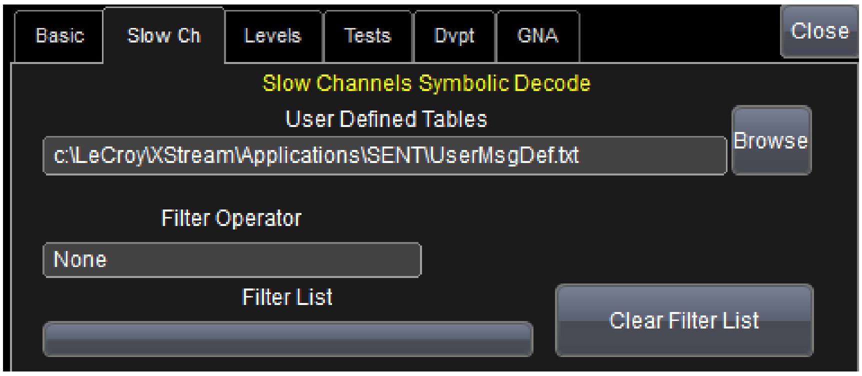


Figure 19 Slow Channel Tab controls

It is expected that different sensors will require different SCDF files, each one based on the manufacturer's definition of the slow channel contents. A small common subset of the message space might end up being standardized messages and can be reused between sensors and applications. The next revision of the SENT protocol might clarify this point.

The Slow Channel Filter Control is explained in the following section.

Slow Channel Filter

Slow Channels of the SENT protocol convey a wealth of information's that do not need and cannot be transmitted over the fast channel(s). As an example, some SENT sensors emit their manufacturing date and serial number over the slow channels. It stands to reason that neither manufacturing date nor serial number will vary over time. However, the slow channels can also convey slowly varying information, such as the core temperature of the sensor, or IC parameters evolving over time. Some advanced sensors embed complete micro-controllers capable of emitting dynamic diagnostic messages. Therefore the slow channels carry a mix of static and dynamic information. Often SENT engineers are interested in monitoring dynamic values, the most concrete example being the core temperature of the sensor. Since the temperature message might only be one of many slow messages, it is useful to filter it out (InRange) of the other messages. Conversely, some frequent messages might be filtered away, to avoid clobbering the users view. We use the example of Figure 15 above (No Filter) to show how the filter works.

In this example, we apply a filter that will only show messages "InRange" of the "Filter List"

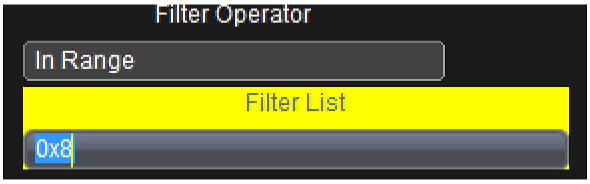


Figure 20: InRange filter example

The resulting table and annotation will only display slow message 0x8, as per image below

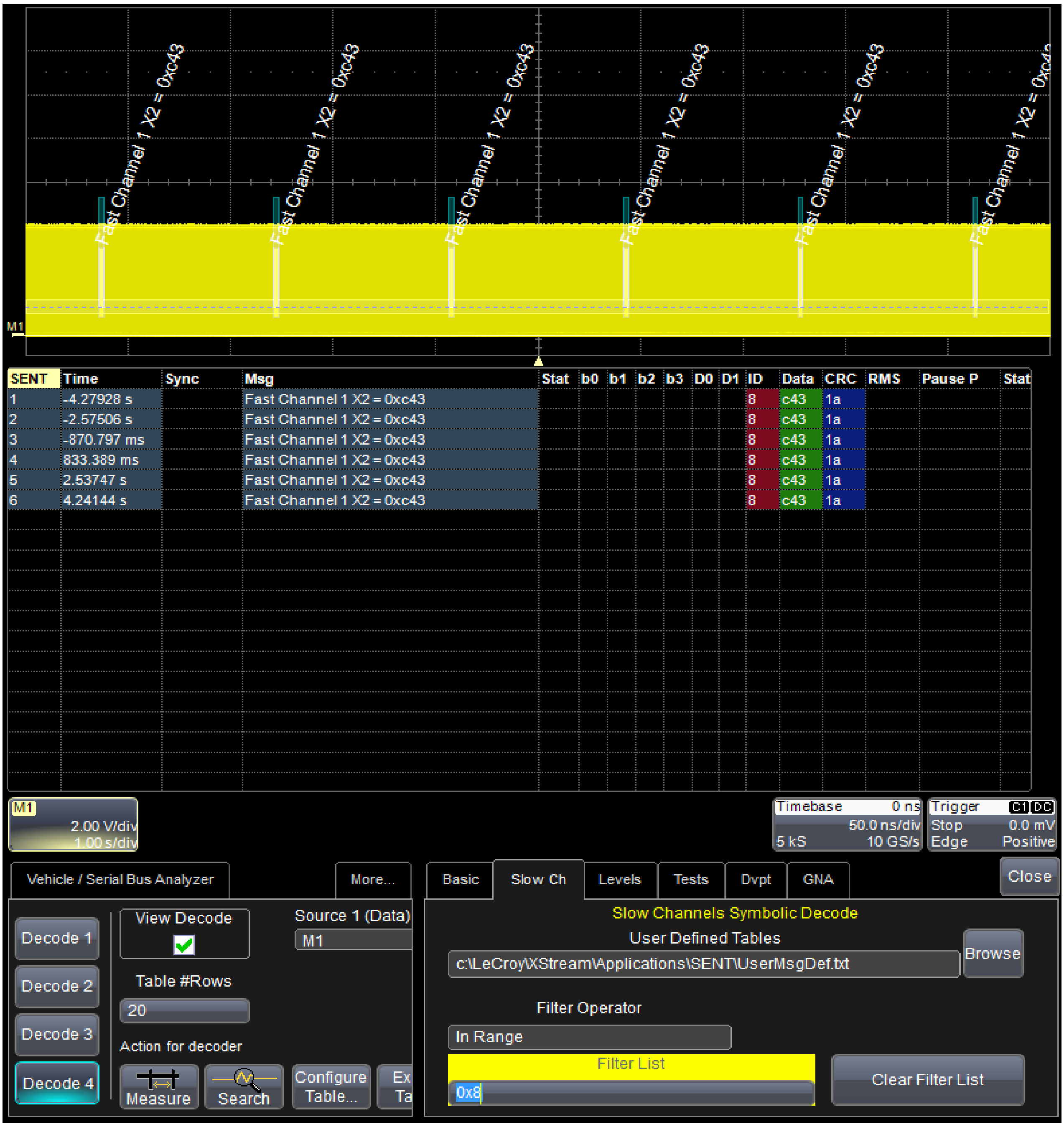


Figure 21: Example of InRange filter on Slow Message 0x8

The definition of the “Filter List” can span several messages

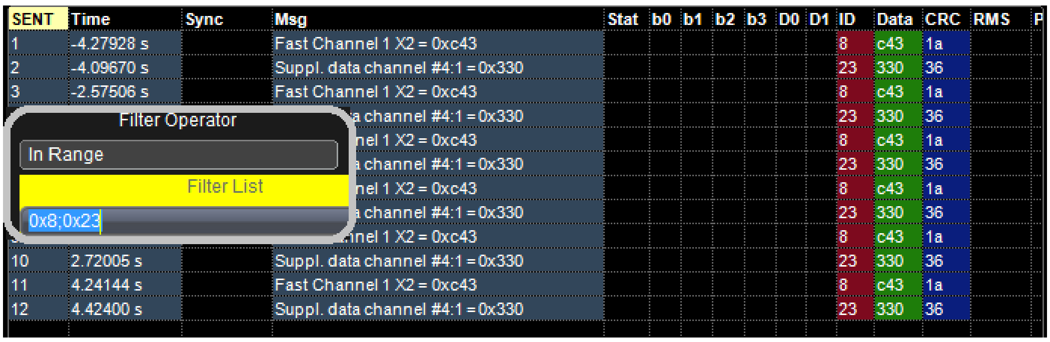


Figure 22: Extracting several messages using an InRange List

In a symmetrical way, the “Out Of Range” operator allows the viewing of all the messages that are not listed in the “Filter List”. The following example shows all of the interesting messages documented in the SCDF file by excluding those that do not have a documented meaning.

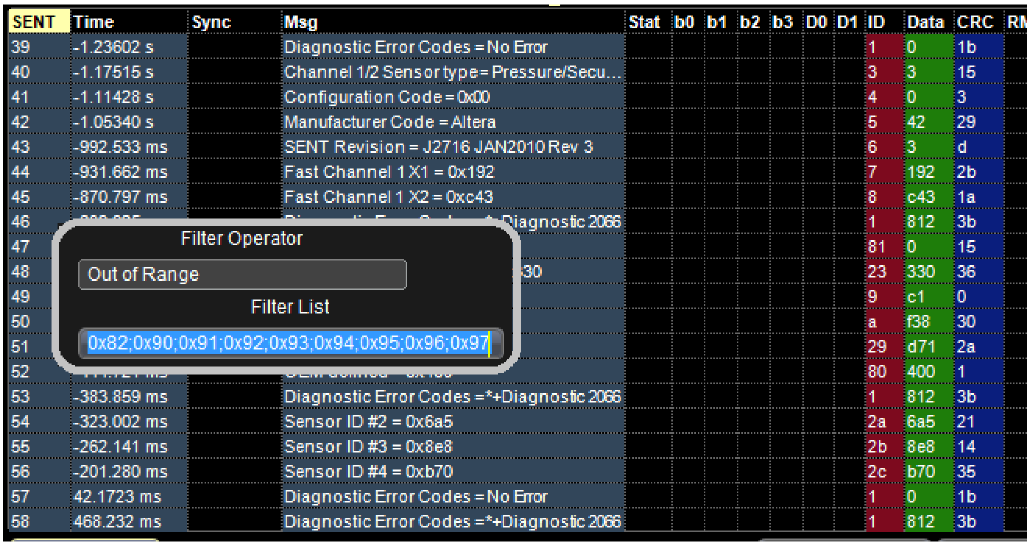


Figure 23: Out Of Range Slow Channel filter

Also note that the “Filter List” can use either hexadecimal values as well as decimal values. The filter definitions below will yield exactly the same results.

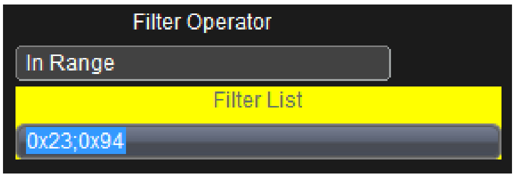


Figure 24: Filter definition in hexadecimal

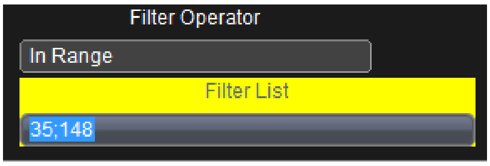


Figure 25: Filter definition in decimal

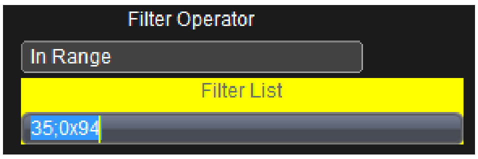
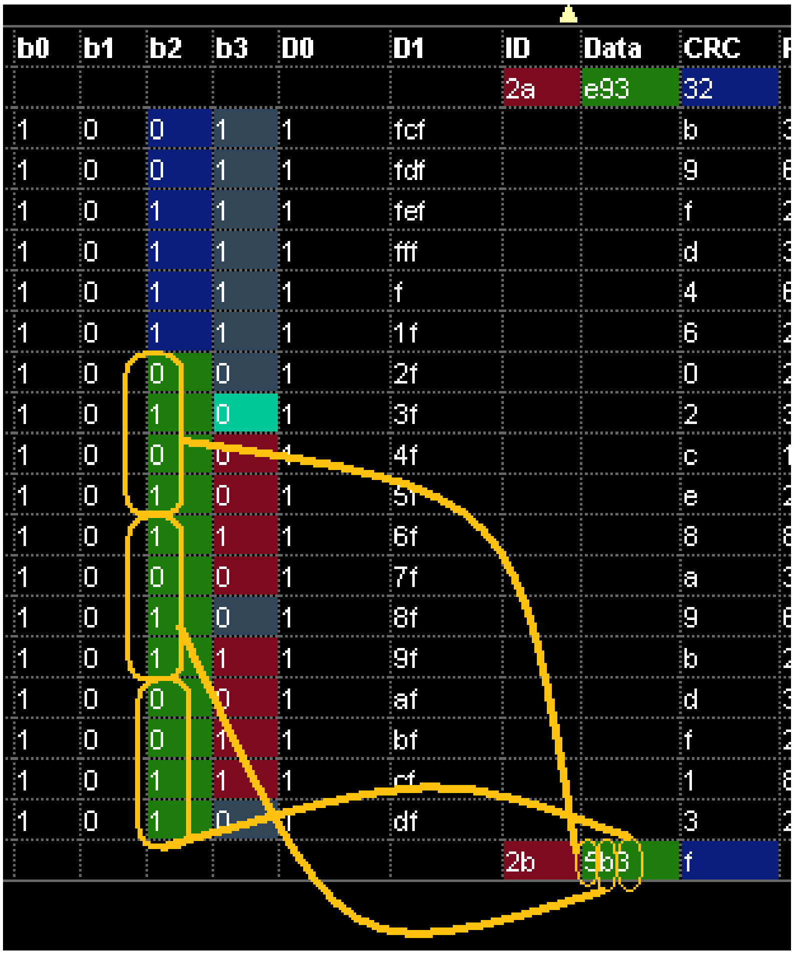


Figure 26: Filter definition in Hexadecimal and Decimal, mixed

Furthermore, it can be observed that the filter effect acts upon the table as well as the annotation on the trace and all of the downstream processing functions such as the ProtoBusMag parameters, the Export and the Search.

Detailed explanation of the Slow Channel decoding

The slow channel encoding scheme is not really complex, but requires a little more detailed explanation. The chromacoding of Status and Communication bits provides a didactic path to full comprehension. This feature helps understanding how the SC values are distributed on many Fast Messages (16 or 18). The example below shows an Enhanced Slow message, where Sync, ID, Data and CRC are distributed on 18 Fast Messages. The grey color shows the Slow Channel Sync bits, red is the 8 bit ID, green is the 12 bit Data and blue the CRC. The following image shows the exact decoding of the 12 bit value. The 8 bit ID(red) and the 6 bit CRC (blue) are decode using the same method.



b0	b1	b2	b3	D0	D1	ID	Data	CRC
						2a	e93	32
1	0	0	1	1	fcf			b
1	0	0	1	1	fdf			9
1	0	1	1	1	fef			f
1	0	1	1	1	fff			d
1	0	1	1	1	f			4
1	0	1	1	1	1f			6
1	0	0	0	1	2f			0
1	0	1	0	1	3f			2
1	0	0	0	1	4f			c
1	0	1	0	1	5f			e
1	0	1	1	1	6f			8
1	0	0	0	1	7f			a
1	0	1	0	1	8f			9
1	0	1	1	1	9f			b
1	0	0	0	1	af			d
1	0	0	1	1	bf			f
1	0	1	1	1	cf			1
1	0	1	0	1	df			3
						2b	5b3	f

Figure 27: Explanation of Slow Channel Bit interpretation

The detailed mechanism is not useful in everyday work, and it is expected that either the word mode or the “Slow Only” mode will be primarily used. However when doubts arise about the slow channels contents, the mixed display mode could be turned on.

EMC susceptibility testing Using the Test Features

Activating the Test System

The Test procedure relies on the Test tab containing the Test criterions enunciated so far by SAE. Each criterion is independent of the others and can be activated/deactivated at will. Furthermore thresholds specified by SAE using a constant (i.e. 2) are implemented with a variable.   This flexibility lets users tighten or relax the condition in the course of their work. The default value for each criterion is the SAE specified value.

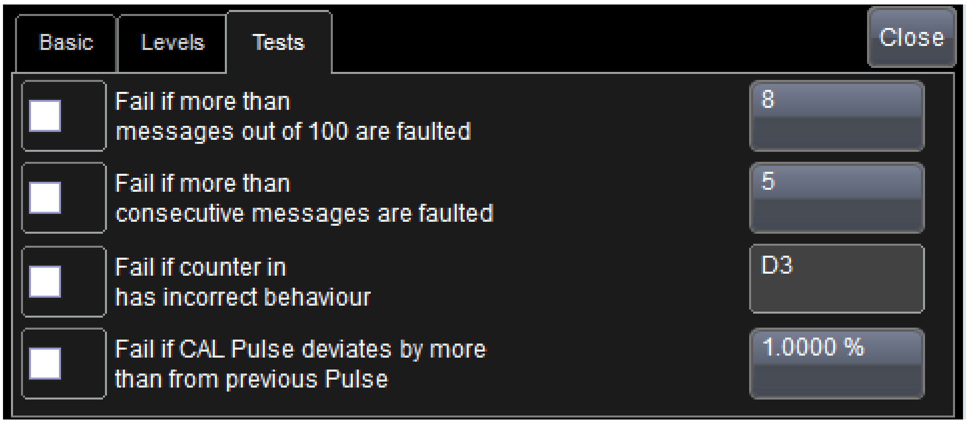


Figure 28: The test tab and its controls

The image above shows the currently know set of test conditions, with the adjustable theresholds.

Several **intra-message tests** are implemented and listed in a following section, such as CRC errors and nibbles outside the 0-15 range. The tests are performed on the fly during the decoding of a message and can be emitted with the message onto the corresponding table line.

The tests above are different in the sense that they span several messages (**inter-message tests**) and rely on the results of the intra-message tests. Therefore they need to be generated through a rescan of the decoded table after the decoding is finished.

The tests discussed in this section emit their results to the Status column as well as to the S column. The Status column provides a textual view of the error whereas the S column is a numerical indicator suitable for further processing using the Pass-Fail system, as explained in the next section.

Using the Test System as part of a Pass Fail Procedure

This is an example of the complete setup, with both a trace containing SENT errors and one without errors

Computational flow:

- Decoder Fills the table
- Decoder detects all specified error conditions (intra-message and inter-message)
- P1 is MessageToValue or ColumnToValue of S columns
- Q1 is condition on P1
- F1 is Track of P1 (comfort feature only)

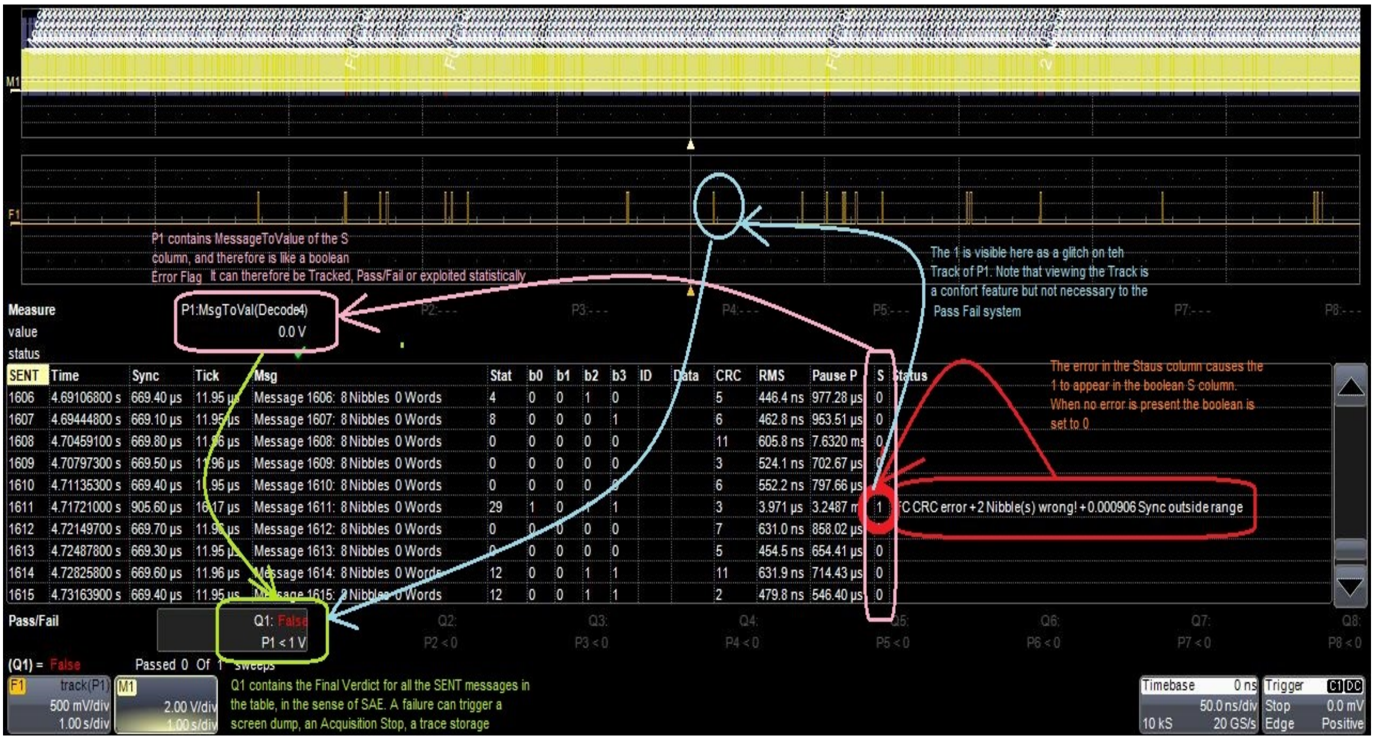


Figure 29: Complete Test setup for EMC susceptibility, with errors

The next image shows the same setup, but with an error-free trace

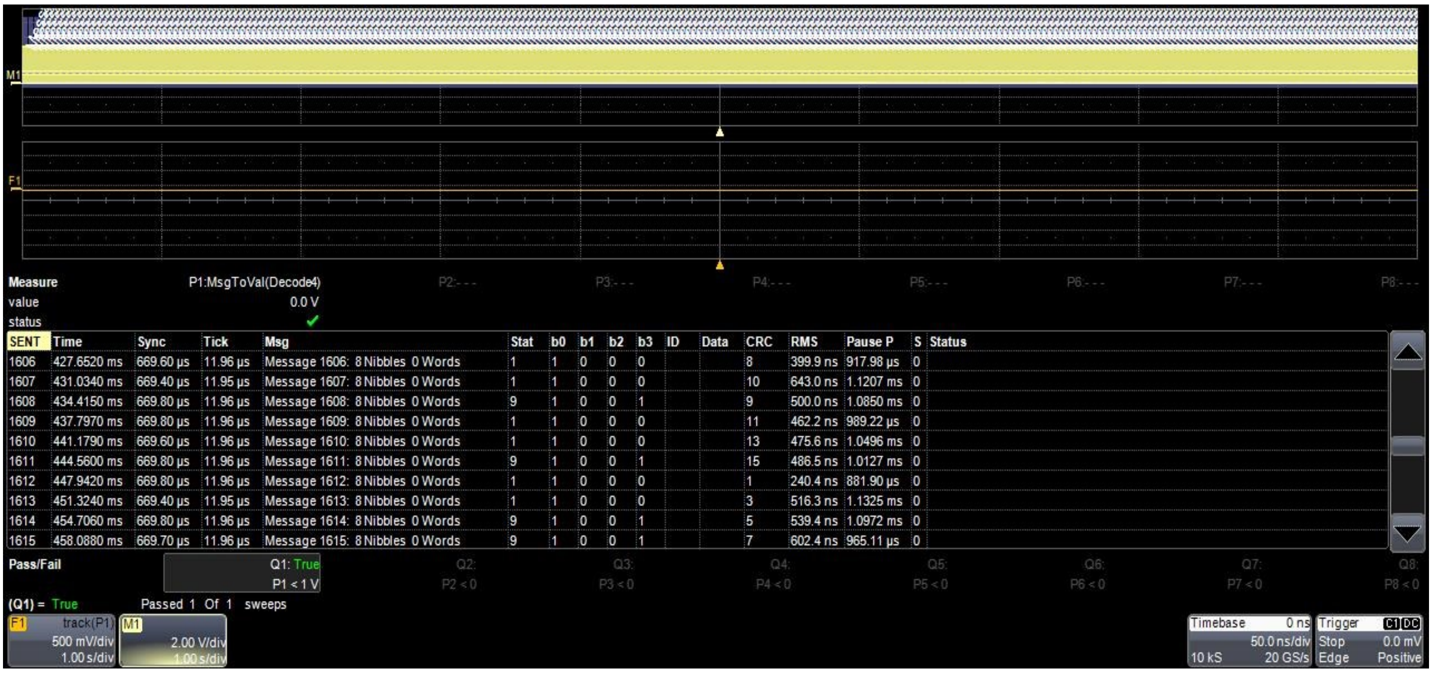


Figure 30: Complete Test setup for EMC susceptibility, without errors

Note that in this case P1 only contains 0 (zeroes) as displayed by the Track. The Pass Fail condition being set as “Pass if All values < 1”is met, and therefore the green Success Flag is set.

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 on the screen and columns can be turned on or off to help the operator.

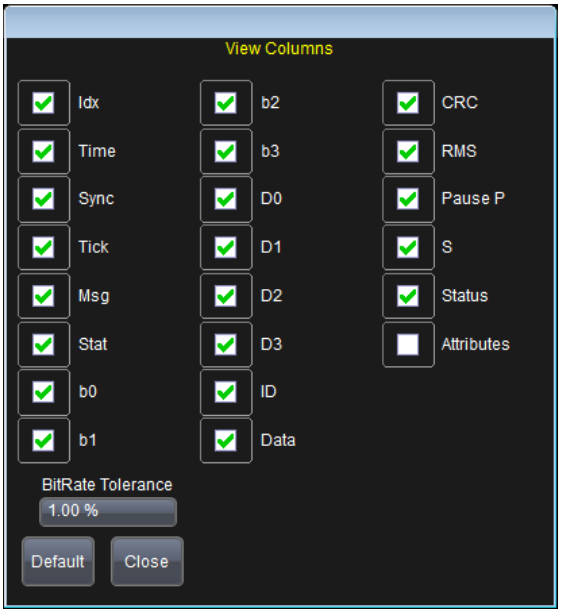


Figure 31: The columns of the SENT decoder

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.
Time	Time of the beginning of the SENT message, with respect to the trigger point of the record.
Sync	Real Measured Length of the Sync Pulse for the message on this line of the table. The pulse width is measured between the 2 falling edges of the Sync pulse, at the intersection of the signal and the Level selected in the “Level Tab” Note that large hysteresis also impact this value read out.
Tick	This is the Real **TickTime** value computed as the Sync pulse divided by 56. This value should be close to the selected TickTime in the Basic Dialog
Msg	This is the message summary, with the number of transitions, nibbles and words.
Stat	The value of the Status and Communication Nibble. This value is also split into its bit components in the next 4 columns to help interpreting the contents (Status and slow channels).
b0	Reserved for specific applications.
b1	Reserved for specific applications.
b2	Serial Data Message Bits (slow channels).
b3	Message Start.
D0-D3	The payload nibbles interpreted according to the settings of D0 in tab “Fast Ch”. This column only appears when Dx is active. These columns are also used for tracking values when using ProtoBusMAG. The D0 column is used when in Nibble mode

ID	
Data	
CRC	Value of the CRC nibble that gets compared to the computed value based on the other nibbles of the message. When these 2 values do not match, a message is emitted in the status column. It is normal that the first and last message of a record, when truncated, generate such a CRC error.
RMS	Room Mean Square value of the falling edge crossings, usually in nanoseconds. rms = $\sqrt{\sum_{j=0}^{j=n}(crossing\ j-(n*realTickLength))^2} / n$ n is so far always 8, for the 8 falling edges following the Sync pulse. So far values observed in the lab range from 10 to 50 ns, reflecting the local stability of the edge generations of the sensor.
Pause P	Pause Pulse, measured from the end of a message to the beginning of the next message. Note that this column is empty when the stream does not use a Pause Pulse to warrant message equidistance.
S	Boolean Status column. This column is only useful for tests, and reflects the contents of the "Status" column. The value is 0 when the "Status" column is empty and non-zero when the "Status" column contains text.
Status	Error and Warning reporting column.
Attributes	Utility column, invisible to the user.

Table2: List of Columns in the table

Note: BitRate Tolerance is not used by the SENT decoder

Error Messages emitted to the Status column

The Status column contains all of the error messages emitted by the decoder.

Channel	Error description	Error message	Status	S
Fast	Nibble Value outside Range 0-15	" %d Nibble(s) wrong! "	OK	1
Fast	Sync Outside Range +-25% of Sync computed as 56 * user TickTime	" %f Sync outside range "	OK	1
Fast	Fast Channel CRC Error	" FC CRC error "	OK	1
Slow	In enhanced Slow Msg, if Bits 7, or 13 or 18 not 0	" B 7|13|18 != 0 ! "	Tested under debugger, but not with real signals as	1
Slow	Enhanced Slow Msg CRC	" SC(18) CRC Error "	OK	1
Slow	Legacy Slow Msg CRC	" SC(16) CRC Error "	OK	1
Compound fast	More than N in 100	" %d/100 messages faulted"	OK	2
Compound fast	More than N consec.	" %d consecutive messages faulted"	OK	3

Compound fast	Counter error	" Counter D%x faulted"	OK	4
Compound fast	CAL > % from prev	" Sync faulted"	OK	5

Figure 32: List of Error Messages

APPENDIX A 3 SENT Examples

The images in the Appendix document various topologies of SENT signals, with a variety of Tick Times, Polarity and Pause Pulse. Refer to the main document for the detailed explanations of the controls.

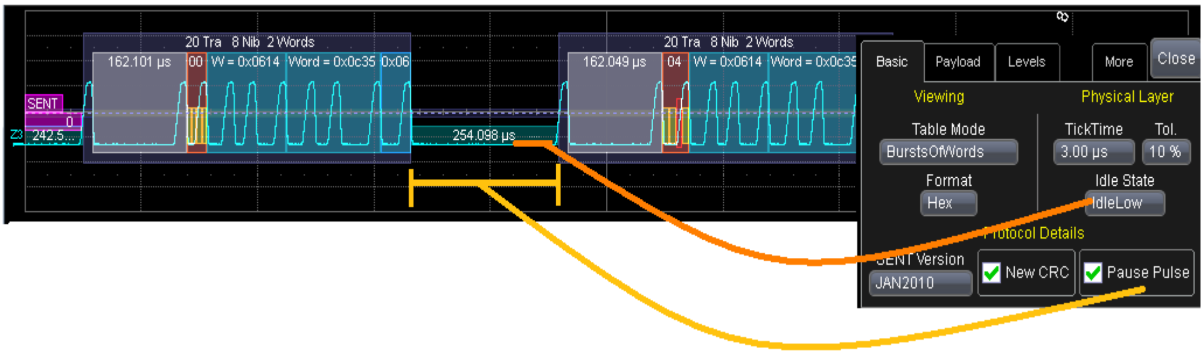


Figure 33: Example of SENT signal, 3us TickTime, Idle Low with Pause Pulse

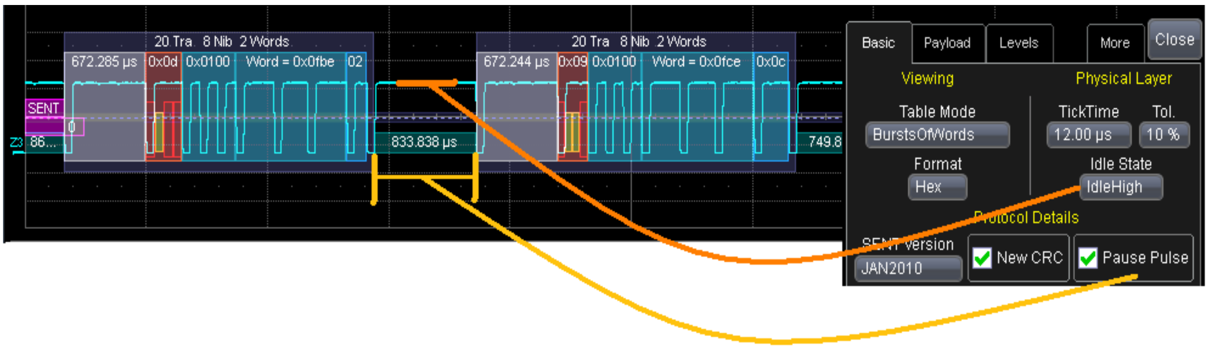


Figure 34: Example of SENT signal, 12 us TickTime, Idle high with Pause Pulse

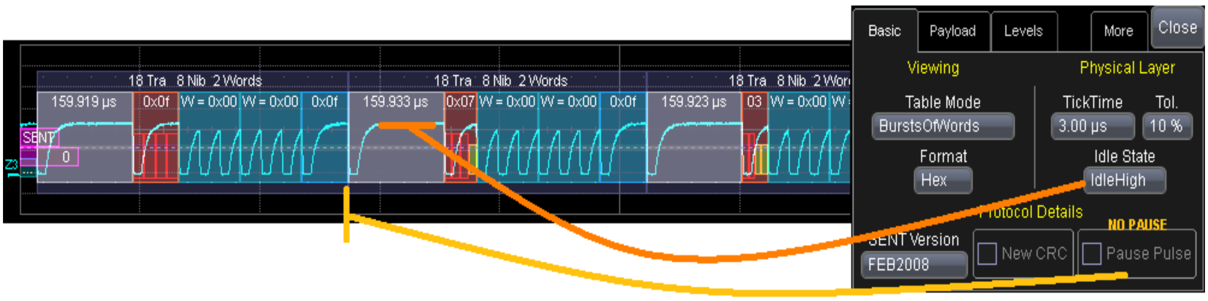


Figure 35: Example of SENT signal, 3us Tick Time, Idle High, without Pause Pulse

APPENDIX B 2 SENT SPC Examples

SENT SPC is a half-duplex variation of SENT variation. The micro-controller initiates the transfer of the data from the sensor to himself by emitting a Master trigger Pulse (MTP). The decoding is selected by choosing “SENT SPC” in the Protocol Drop down Box. The MTP length is then entered in the Basic Tab of the Decoder controls.

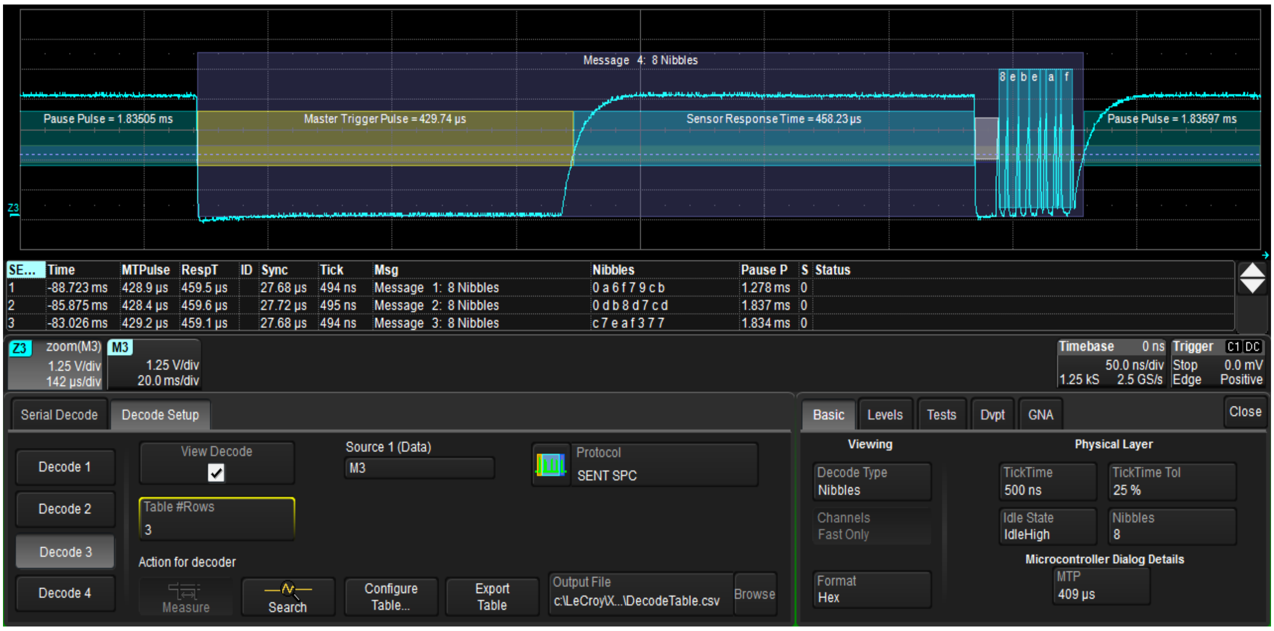


Figure 36: Example of SENT SPC signal, 500 ns Tick Time, 409 us Master Pulse, 8 Nibbles

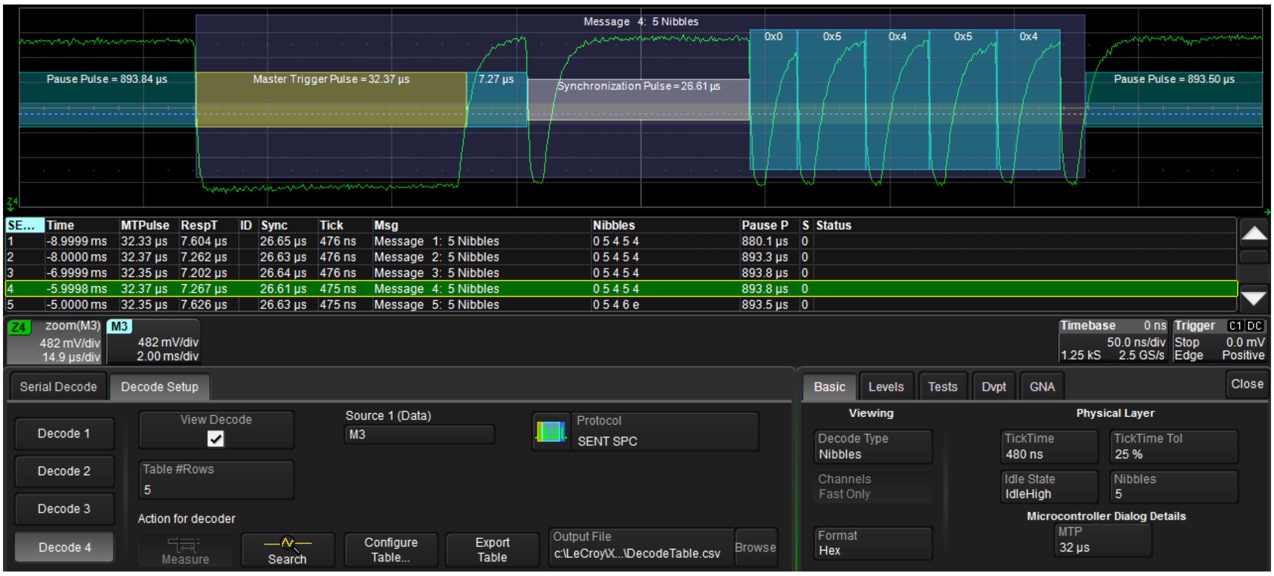


Figure 37: Example of SENT SPC signal, 480 ns Tick Time, 32 us Master Pulse, 5 Nibbles

APPENDIX C Using Level and Hysteresis for difficult signals

Please refer to the ARINC 429 web page for an example of this functionality:

http://www.lahniss.com/_p/_parinc429/arimg/decodewithtunedlevels.png

The method described uses 2 levels instead of one level with hysteresis. The goal is to set levels and hysteresis in such a way that the noise is ignored. The principles explained on this avionics protocol are applicable to any other protocol regardless of its physical layer definition.

APPENDIX D: Using ProtoBusMAG in connection with SENT

Please also refer to examples shown on pages:

http://www.lahniss.com/_p/_psent/sent.shtml Measurements on Melexis, Hella and Micronas sensors are shown.

http://www.lahniss.com/_u/_ulah10x/lah10x.shtml Measurements on LAH 10x signal source containing different types of P/T sensors and Hall Effect sensors with various output configurations.

Example of MessageToValue

A fairly typical output of an angle sensor, manipulated manually, would look like:

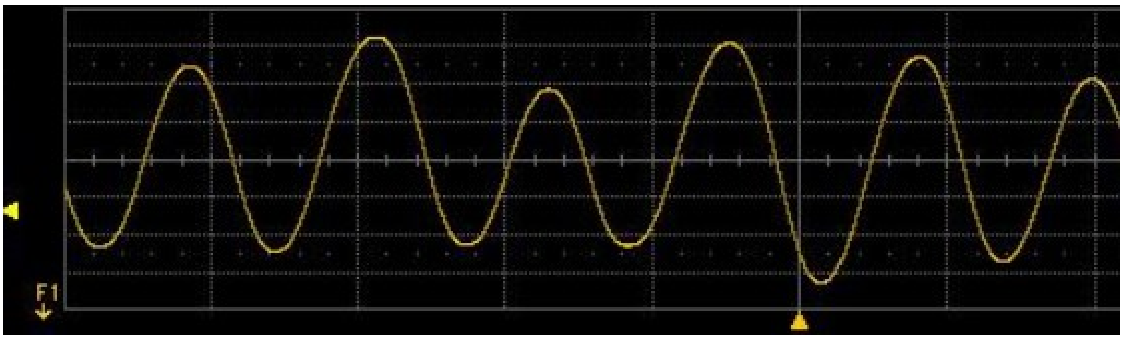


Figure 38: Typical Trend of a 12 bit Angle sensor when rotated

The monitoring of a Secure Sensor with counter and Inverse of MSN can be conducted as follows:

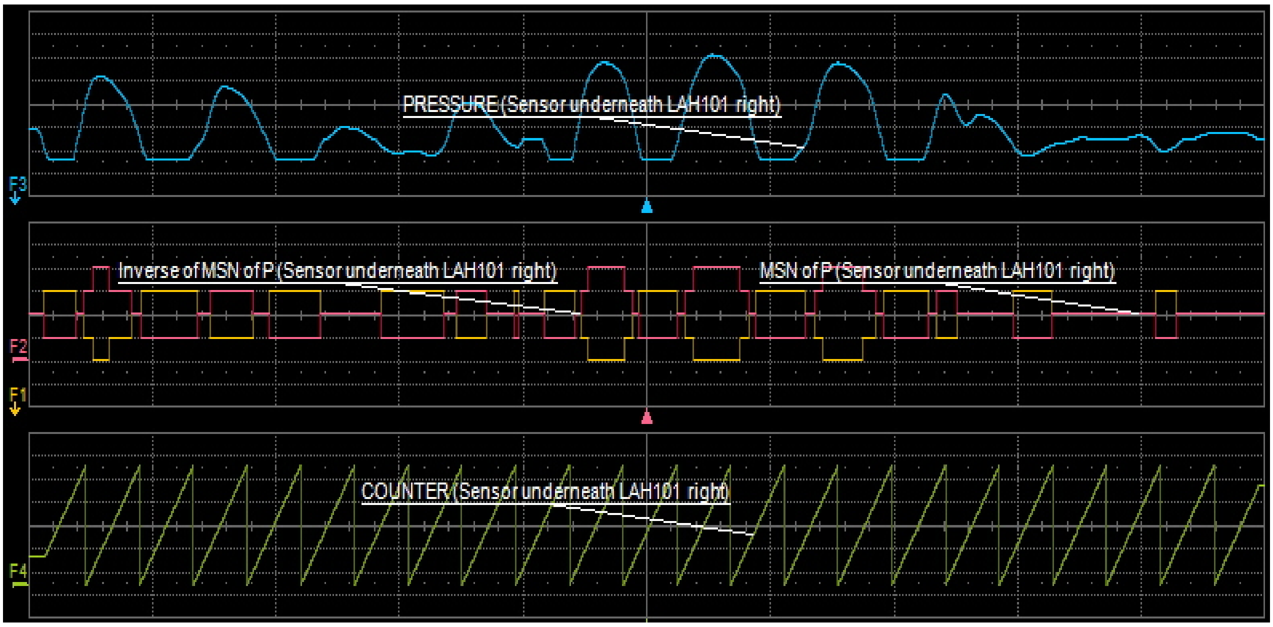


Figure 39: Monitoring all the values emitted by a Secure Sensor

Example of ColumnToValue

Another useful parameter is ColumnToValue which can sometimes replace MessageToValue when neither filtering nor rescaling is needed. **ColumnToValue combined with its Track** works as the graph of a column on a spreadsheet. The image below shows a triple example on a SENT SPC decode.

Figure 40: Graphing the MTP, the Response Time and the Pause Pulse in an SENT SPC communication

Generally speaking SENT SPC is easier to setup then its big brother MessageToValue because it requires less settings and immediately works on the column chosen. When used on non-addressed buses like SENT it can be used for many of the decoded columns. It can also be mixed with MessageToValue on the Slow Channels when filtering by ID is required. Both measurements can coexist.

APPENDIX E: Exploiting the memory Depth and optimizing for speed

LeCroy oscilloscopes have large or very large memories. This memory depth allows capture of **long time spans of signals**, especially when signals are relatively slow, as it is the case for SENT. The long time spans can allow the observation of the message payloads over seconds, sometimes minutes. When used on sensors this can help in monitoring the sensors behavior. The very useful option called ProtoBusMAG (Protocol Bus Measure Analyze and Graph) yields excellent results on long traces.

Automation access to relevant parameters

In many cases the result need to be exploited via a host computer and commands in this style:

app.PassFail.LastPass.Result.Value

Comments on Acquisition window and statistics

An oscilloscope repeatedly captures **windows in time**. Typically for SENT the window is 20 to 70% of the time. The remaining time the instrument is blind due to the processing of the previous acquisition.

The capture however can then be **repeated over a long time**, with the result that the percentage of the coverage will extend to the entire measurement session. For example if the oscilloscope is set to 100ms per division, (therefore 1 second per acquisition window) there will be about 300ms of processing time. If the test is repeated over an hour the same proportion will apply. This is largely sufficient to meet the test requirements in most cases and has the great advantage over any other system that if and when errors occur they can be analyzed immediately and easily using the large tool-box on-board the oscilloscope.

Depending on the requested processing load, the time coverage might vary between 10% and 70 %.

Parallelizing tests using all of the oscilloscope channels

The tests could be parallelized using all 4 channels of the oscilloscope. In that case All 4 channels would be fed into each of the available decoder, and the processing chain above would be cloned 4 times. This mode is statistically interesting because multi-channel acquisitions occur in parallel. The processing is serialized, but i.e in 80/20 % mode the monitoring of one more sensor only requires 20% additional time. This property could be used for production tests to test more sensors in less time.

It is also possible to use digital channels as input.

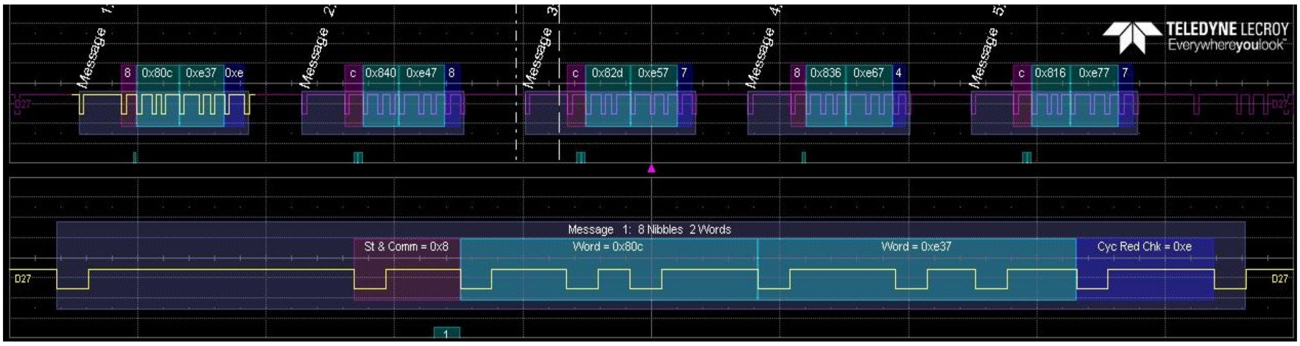


Figure 41: Decoding a SENT digital trace

Hints to optimize for speed

Depending on the regime of operations several tricks can be used to speed up the acquisition/processing/display loop. These tricks are independent of one another and might be combined.

Avoid oversampling

It is not necessary to oversample the SENT signal to decode it. 20 to 50 samples per pulse are sufficient. Having too many samples slow down the processing chain.

Use triggered waveforms

Having a solid trigger speeds up acquisition. A convenient trigger for SENT is the Interval Trigger on falling edges set to match the CAL pulse of the SENT signal. The picture shows the trigger setup, with a TT of 3us. Using the AUTO trigger normally slows down the acquisition loop.

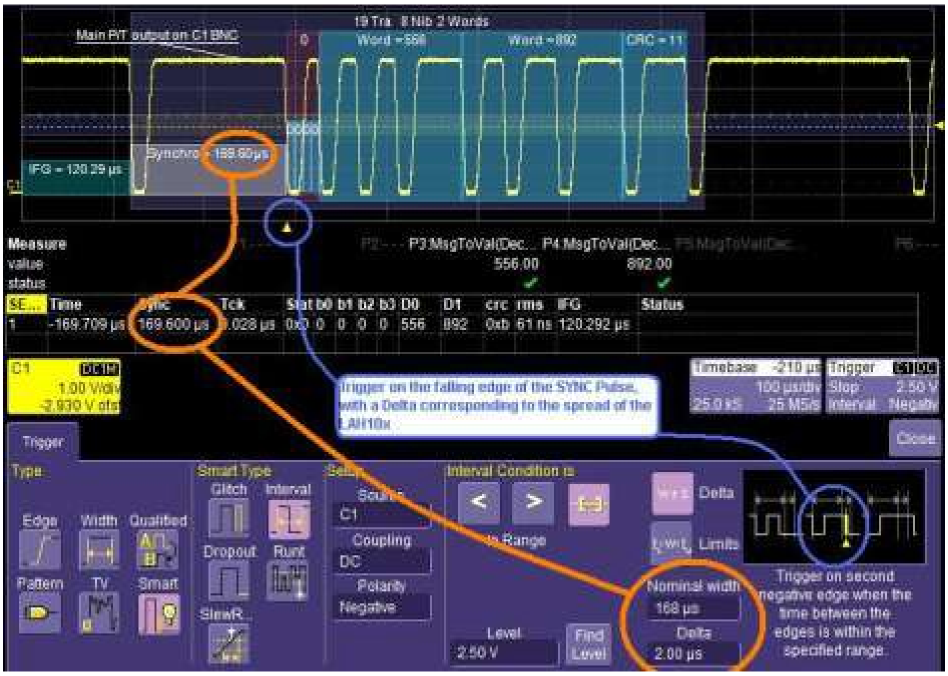


Figure 42: Setting up the Interval trigger for SENT

Optimize for Analysis and not for display

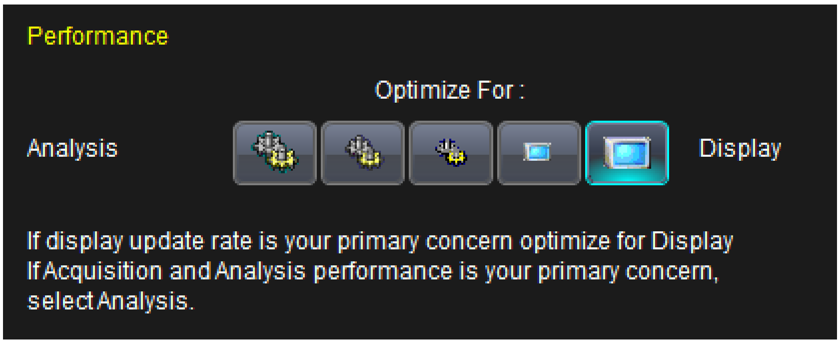


Figure 43: Performance selection control, Analysis vs. Display

This feature allows the user to have a certain control over the CPU time allocated to Display versus Analysis. It can help in certain cases.

Turn off all traces and annotators

As strange as it seems, the decoder can work **without showing its results**. The mere fact that a Parameter (MessageToValue in our case) is connected downstream from the decoder will for the decoder to remain instantiated. However, it will save CPU time, and therefore accelerate the processing loop, to not display the decoded result, and only use them for the Pass/Fail processor.

There are 2 items possible in this category: Turn off the decoded trace and reduce the decode table to one single line.

It is possible via automation to turn the entire table off also but still keep the computation active (app.SerialDecode.Decode1.View = 'false')

Decrease number of columns in Export of Tables

If the Decode Table needs to be exported, it is best to decrease its number of columns to the minimum necessary. The export time to the file is proportional to the amount of data exported. Fewer columns consequently translate into a faster export. Generally speaking, anything that can be computed on-board the oscilloscope accelerates the whole test.

Turn off automatic calibrations

Digital oscilloscopes usually have an automatic recalibration system. This system kicks in when certain temperature or humidity thresholds are crossed. **In automated tests** it would be best to programmatically turn off these CALs during a batch of measurement and **programmatically re-enable** them after.

Resources for optimizing performance

http://cdn.teledynelecroy.com/files/appnotes/an_019_techbrief_optimzg_perf.pdf