SIGMA2

LOGIC ANALYZER

Reference Manual

ASIX s.r.o.
Staropramenna 4
150 00 Prague
Czech Republic

www.asix.net

support@asix.net

sales@asix.net

ASIX s.r.o. reserves the right to make changes to this document, the latest version of which can be found on the Internet.

ASIX s.r.o. renounces responsibility for any damage caused by the use of ASIX s.r.o. products.

Table of Contents

1 General Information 5
1.1 Product Overview 5
1.2 Package Contents 5
1.3 Panel Overview 6
1.4 Product Version 6
2 Getting started 7
2.1 Installation on Windows 7
2.2 Installation on Linux 7
2.3 Target connection 7
2.4 Acquire the data 8
3 Controls 9
3.1 Indicators and button 9
3.2 Target Connection 9
4 Using SIGMA & OMEGA Logic Analyzers Software 11
4.1 Clock Source 11
4.1.1 Modes of operation 11
4.1.2 Synchronous Clock Timing 12
4.1.3 Asynchronous Clock Timing 13
4.2 Input pins 13
4.3 Traces 13
4.4 Trigger Settings 14
4.4.1 Basic Trigger Settings 15
4.4.2 Advanced Trigger Settings 15
4.4.3 Trigger Position within Acquisition 16
4.4.4 External Triggering 17
4.4.5 Other Trigger Settings 17
4.5 Working with the acquired data 18
4.5.1 Navigation and analysis 18
4.5.2 UART Protocol Decoder 19
4.5.3 SPI Bus Decoder 20
4.5.4 I2C Bus decoder 20
4.5.5 USB 1.1 Analyzer Plugin 21
Installation 21
What to measure 22
Measure tool attachment 22
Measuring 22
Processing 22
Viewing 23
Searching 24
Linking the events window with the analyzer window 25
Gathering of related communication into trees 25
4.6 Auxiliary functions 25
4.6.1 Insider 25
4.7 Frequency Measuring 28
4.8 Pin View 28
4.9 Availability of Auxiliary Functions 29
4.10 Using the Application for Automated Acquisition 29
4.11 Plugins 29
5 Using the Logic Analyzer 31
5.1 Sampling Frequency 31
5.2 Using the Application as a Portable Application 31

6 Using SIGMA2 Logic Analyzer under Linux 33

7 OMEGA and SIGMA2 Comparison 35

8 Specifications 36

9 Document history 38

1 General Information

1.1 Product Overview

SIGMA2 is a logic analyzer - development tool designated for tracing and debugging of TTL (and compatible) digital signals.

The SIGMA2 Logic Analyzer is equipped with 256 Mb of memory and provides with up to 16 digital inputs with sampling rate 50 Msps. Built-in data compression allows for tracing of long running signals without exhausting logic analyzer memory. When using all 16 inputs, guaranteed minimum capacity is 14 million samples. SIGMA2 uses USB (Full-Speed, 12 Mbps) which ensures both data transfer and power delivery with a single cable. There is no additional power supply needed.

Features:

- Up to 16 TTL compatible inputs
- Up to 200 Msps sampling rate (limited number of inputs)
- External clock up to 49.9 MHz
- 256 Mb of internal memory
- RLE hardware compression
- Flexible trigger options
- USB interface for data and power

Note: The SIGMA Logic Analyzer is an original version of the SIGMA2 Logic Analyzer, it was being delivered until November 2011. SIGMA2 is a new version which replaces original SIGMA. SIGMA2 is virtually the same logic analyzer as SIGMA, it differs from its predecessor by new enclosure, different LED indication and new button inspired by PRESTO and FORTE programmers. SIGMA and SIGMA2 Logic Analyzers are binary compatible, therefore any utility which was originally intended for SIGMA will work with SIGMA2.

1.2 Package Contents

Please inspect the logic analyzer mechanically and electrically upon receiving it. Unpack all items from the shipping box and check for any obvious signs of physical damage that may have occurred during transportation. Report any damage to the shipping agent immediately. We recommend to save the original packing carton for possible future reshipment. Every logic analyzer is shipped with the following contents:

- SIGMA2 Logic Analyzer
- Target cables:
  - 20 individual pins (SIGMACAB)
  - one-to-one 20 pins (SIGCAB20)
  - one-to-one 10 pins (SIGCAB10)
- USB cable (A-B)
- CD-ROM (software, drivers)
- Optional accessory (must be ordered separately):
  - Set of 10 variously colored hooks (PicoHook10)

Verify that all ordered items are included in the shipping container. If anything is missing, please contact your local distributor.

1.3 Panel Overview

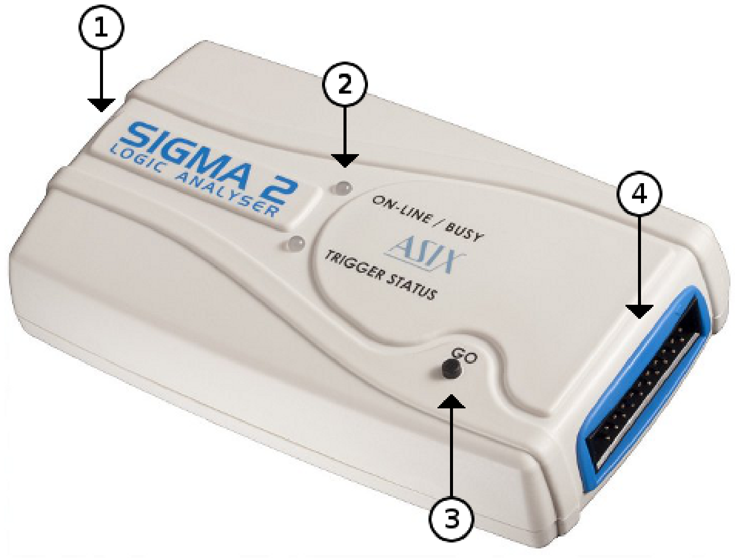


Fig.2: SIGMA2 Panel Overview

Panel Overview

1	USB port
2	Indication LEDs
3	Multifunction Start/Stop/Trigger button
4	Target interface

1.4 Product Version

The logic analyzer may come in different hardware or software version. This manual reflects features and options of logic analyzer software equipment **Logic Analyzer version 3.00**. Updates to the latest version of the software are always available free of charge on the internet at www.asix.net. The main software comes with many supporting utilities, which may be different version.

Trade Name	Serial Number	Availability	
SIGMA	Since A6010001	Since 2007	No longer available
SIGMA 2	Since A6020001	Since 2011	Low cost
OMEGA	Since A6030001	Since 2012	Flagship

Table 1: Logic Analyzer Versions

The logic analyzer software supports basic functionality of all logic analyzer hardware versions in the table, but advanced features availability may differ. This manual describes the hardware and software features available only to the SIGMA2 Logic Analyzer.

Detailed comparison of the logic analyzers is in the chapter OMEGA and SIGMA2 Comparison.

2

Getting started

Before connecting and powering up the logic analyzer, please review and go through all the instructions in this chapter. You will learn all basic functions required to debug your first application with the logic analyzer.

2.1 Installation on Windows

Install the ASIX SIGMA & OMEGA Application Package available on CD or at www.asix.net to your computer. Check the web page periodically for software updates. Software updates are free and may address a discovered problems and add new features. The SIGMA2 Logic Analyzer is a USB device, therefore it requires USB drivers. The drivers are installed automatically during software package installation. Connect SIGMA2 to a USB port or a USB hub using supplied cable. In a while green ON-LINE LED should turn on and the SIGMA2 Logic Analyzer should appear in Device manager as correctly installed.

2.2 Installation on Linux

The software for the SIGMA2 Logic Analyzer is designated for Windows, but it runs on most distributions of Linux using Wine.

For installation instructions of ASIX SIGMA & OMEGA Application Package on Linux see the chapter Using SIGMA2 Logic Analyzer under Linux.

2.3 Target connection

The SIGMA2 Logic Analyzer is equipped with 16 high impedance inputs with logic levels compatible to TTL and auxiliary *Trigger In* and *Trigger Out* pins.

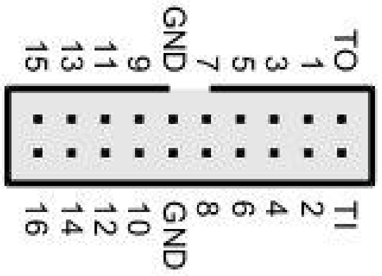


Fig.3: Target connector

Always connect the ground between the application and the analyzer and then connect desired input pins. The SIGMA2 Logic Analyzer do not isolate ground between the PC and the application.

If you want to use *Trigger In* and *Trigger Out* pins, use ***Settings → Trigger Options...***, tab ***Other Settings***.

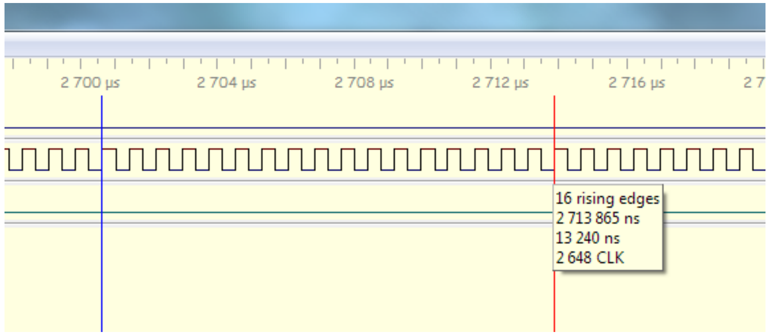
Warning: *Trigger In* and *Trigger Out* pins are not 5V tolerant!

Note: Capacitance and length of the probe cables should be taken into consideration when connecting to a debugged application, otherwise a cross-talks of fast signals may occur. The leads of the supplied cable with individual pins may be split to reduce capacitance between adjacent wires for mid-range signal speeds. For high-speed signals, using of any cable is not recommended, it is recommended to connect directly the logic analyzer to the application.

2.4 Acquire the data

Start the application ASIX SIGMA & OMEGA Logic Analyzers from start menu and launch the data acquisition by pressing **Enter**. The SIGMA2 memory will last most probably for minutes, so you can stop data acquisition any time by pressing button *Stop Acquisition Now*.

- To zoom, select a range with mouse or use key **+** or ***** on keyboard.
- To take back last zoom operation, use key **-**, **Backspace** or **/**.
- To move over the measured data, use arrow keys **→**, **←**, **Page Down** and **Page Up**, mouse wheel or hold **Ctrl** key while moving mouse pointer.
- Using **Alt**+**←**/**Alt**+**→** you can jump to next change on a line where the mouse pointer is.
- Jump to another line is possible with arrows **↑** and **↓**.
- To measure time or frequency or count number of edges, use keys **Spacebar**, **F** and **Q**.
- To select a trigger, press **T** key and to use different clocking options, press **C** key.



16 rising edges
2 713 865 ns
13 240 ns
2 648 CLK

Fig.4: Counting edges in an acquisition

To start using protocol analyzers, double click any trace label on left margin of the window to add a new line. Every new protocol analyzer is on a separate line, called a *trace*.

3 Controls

3.1 Indicators and button

Main panel contains two bi-color LED indicators providing an operator with quick status information.

🟢🟡**ONLINE / BUSY (green/yellow LED)**

⚪**off:** SIGMA2 is in low-power (Sleep) mode or no USB driver has been installed (Windows only) or no synchronization signal is being received during Daisy-Chain operation

🟢**green:** SIGMA2 is ready to operate

🟡**yellow:** SIGMA2 is acquiring data

🔴🟡**TRIGGER STATUS (red/yellow LED)**

⚪**off:** Trigger logic is inactive - no trigger condition has been detected

🔴**red:** SIGMA2 is waiting for trigger condition

🟡**yellow:** flashes when trigger condition or trigger pattern has been matched

The Go button helps to control the analyzer comfortably - it cyclically switches among essential operation states. When it is pressed in idle state the data acquisition is launched. When it is pressed in running state the software trigger is initiated. When it is pressed in triggered state the acquisition is stopped, idle mode is launched and the data transfer from logic analyzer memory to PC begins.

3.2 Target Connection

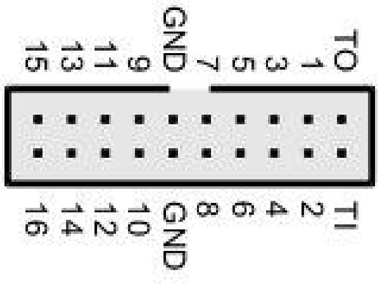


Fig.5: Target connector

The SIGMA2 Logic Analyzer is equipped with sixteen high impedance inputs with TTL input logic levels and 1 MΩ pull-down resistors to park the pins when they are unused. The functions *Trigger In* and *Trigger Out* are available. The function *Power Output*[1] is available on *Trigger In*.

Warning: *Trigger In* and *Trigger Out* pins are not 5V tolerant!

If you want to use *Trigger In* and *Trigger Out* pins, use ***Settings → Trigger Options...***, tab ***Other Settings***.

The digital inputs are organized as two 8-pin ports (inputs 1 to 8 are merged into port 1, inputs 9 to 16 are merged into port 2). Pin-to-pin skew between inputs on a single port is rather low while it may be considerably higher between the ports.

Always connect the ground between the application and the analyzer and then connect desired input pins. SIGMA2 Logic Analyzer do not isolate ground between the PC and the application [2].

	min.	typ.	max.	
V_{IL} input low voltage			0.8	V
V_{IH} input high voltage	2.0			V
V_{IN} absolute rating, inputs 1..16	-0.3		5.5	V
V_{IN} absolute rating, trigger I/O	-0.3		3.6	V
t_{sksp} pin-to-pin skew within single port		1		ns
t_{skbp} pin-to-pin skew between ports		4.8		ns

Table 2: Inputs Electrical Specifications

Note: Capacitance and length of the probe cables should be taken into consideration when connecting to a debugged application, otherwise a cross-talks of fast signals may occur. The leads of the supplied cable with individual pins may be split to reduce capacitance between adjacent wires for mid-range signal speeds. For high-speed signals, using of any cable is not recommended, it is recommended to connect directly the logic analyzer to the application.

[1] Usage of any SIGMA2-powered logic level translator is possible thanks to possibility to power it from the logic analyzer using *Power Output* feature on *Trigger In* pin.

[2] When using a USB optoisolator designed for USB Full-Speed (12 Mbps) you will benefit from unique SIGMA2 feature: The logic analyzer will download the data you are currently looking at in preference. Therefore, the data you are looking at will be available virtually in the same time you will focus on them.

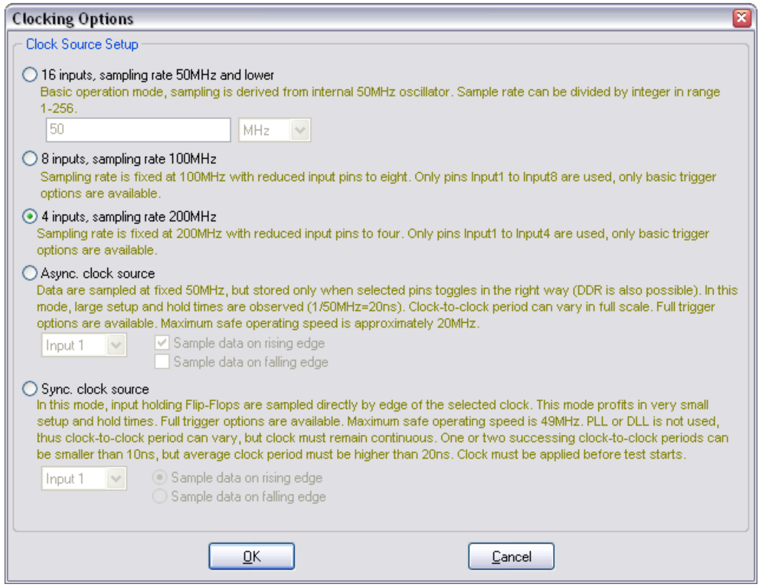
4

Using SIGMA & OMEGA Logic Analyzers Software

4.1 Clock Source

4.1.1 Modes of operation

SIGMA2 can operate in one of several modes adapted to actual user needs and particular debugged application (number of inputs, sampling period etc...). The mode of operation can be selected in ***Settings → Clock source***.



Clocking Options

Clock Source Setup

○ 16 inputs, sampling rate 50MHz and lower
Basic operation mode, sampling is derived from internal 50MHz oscillator. Sample rate can be divided by integer in range 1-256.
50 MHz

○ 8 inputs, sampling rate 100MHz
Sampling rate is fixed at 100MHz with reduced input pins to eight. Only pins Input1 to Input8 are used, only basic trigger options are available.

● 4 inputs, sampling rate 200MHz
Sampling rate is fixed at 200MHz with reduced input pins to four. Only pins Input1 to Input4 are used, only basic trigger options are available.

○ Async. clock source
Data are sampled at fixed 50MHz, but stored only when selected pins toggles in the right way (DDR is also possible). In this mode, large setup and hold times are observed (1/50MHz=20ns). Clock-to-clock period can vary in full scale. Full trigger options are available. Maximum safe operating speed is approximately 20MHz.
Input 1
☑ Sample data on rising edge
☐ Sample data on falling edge

○ Sync. clock source
In this mode, input holding Flip-Flops are sampled directly by edge of the selected clock. This mode profits in very small setup and hold times. Full trigger options are available. Maximum safe operating speed is 49MHz. PLL or DLL is not used, thus clock-to-clock period can vary, but clock must remain continuous. One or two successing clock-to-clock periods can be smaller than 10ns, but average clock period must be higher than 20ns. Clock must be applied before test starts.
Input 1
● Sample data on rising edge
○ Sample data on falling edge

Fig.6: Modes of operation

Available modes of operation:

Basic Mode

16 inputs, sampling rate 50 Msps.
Basic operation mode, sampling is derived from internal oscillator. Divisor of 2 to 256 can be used to achieve longer measurements.[1]

Higher Sampling Rate Mode 100 Msps

8 inputs, sampling rate 100 Msps.
Number of input pins is reduced to eight with benefit of twice sampling rate. Inputs are limited to first port only (Inputs 1-8).

Higher Sampling Rate Mode 200 Msps

4 inputs, sampling rate 200 Msps.
Number of input pins is reduced to four with benefit of four times sampling rate. Inputs are limited to Input 1-4.

Asynchronous Clock Source

16 inputs, stored only when particular input is updated.

The setup is same as in basic mode (16 inputs, 50 Msps), but they are updated and stored only when selected pin has changed (asynchronous clock). The trigger can be rising edge, falling edge or both edges, maximum clock speed is ~20 MHz with 1:1 duty cycle. The asynchronous mode can achieve much lower clock rate than synchronous mode, but can store clock timing. Because of that, the asynchronous mode consumes larger amount of the analyzer memory, because the fact that the sample capture time must be saved as well.

Synchronous Clock Source

15 inputs, external clock.

Input1 or Input9 is utilized as external clock input. Rising or falling clock edge can be chosen, but not both. Clock speed should be within the range of 1 MHz to 49.9 MHz. The data are buffered in FIFO buffer of depth 16 clocks, therefore the data may be transferred in bursts of up to 16 clocks with period as low as 10 ns, but the overall transfer rate must be lower than 16 clocks per 320.64 ns. Due to internal pipeline circuits, the clock signal must be present before start of the data acquisition and some time after the end of the acquisition otherwise several first and last samples will not be contained in the captured data. Using this mode for measuring synchronous bus (e.g. processor bus) can be advantageous.

Note: Data compression (RLE and Huffman coding) is used in every case, disregarding selected mode, giving possibility to capture long time running signals with precise timing. The actual compression ratio depends on characteristics of the particular signal.

4.1.2 Synchronous Clock Timing

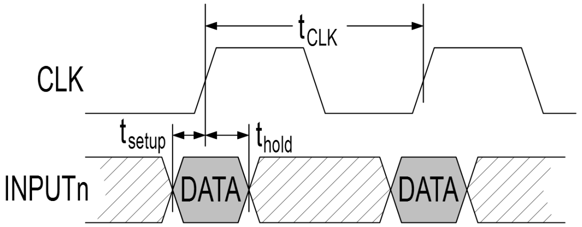


Fig.7: Synchronous clock timing

The synchronous clock timings are measured at the input connector. At the maximum clock rates, use of custom cables and amplifiers may be necessary.

	Typ.	Max.	
t_{setup} Data setup before clock	3.55	8.30	ns
t_{hold} Data hold after clock	-0.55	3.75	ns

Table 3: Synchronous Clock Timing

If more tight t_{setup} and t_{hold} timings are required, use OMEGA Logic Analyzer.

4.1.3 Asynchronous Clock Timing

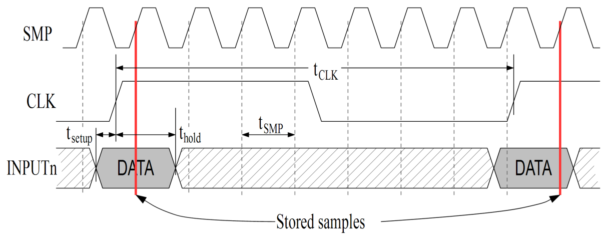


Fig.8: Sampling with asynchronous clock

The asynchronous mode clock timings are defined for the input connector.

	Min.	Typ.	Max.	
t_{SMP} Sampling period		20		ns
t_{CLK} Input clock period	50			ns
t_{setup} Data setup before clock		2.5	7.3	ns
t_{hold} Data hold after clock		22.7	27.5	ns

Table 4: Asynchronous Clock Timing

If more tight t_{setup} and t_{hold} timings are required, use the OMEGA Logic Analyzer in *Synchronous clock Mode With Asynchronous Time Scale* mode.

4.2 Input pins

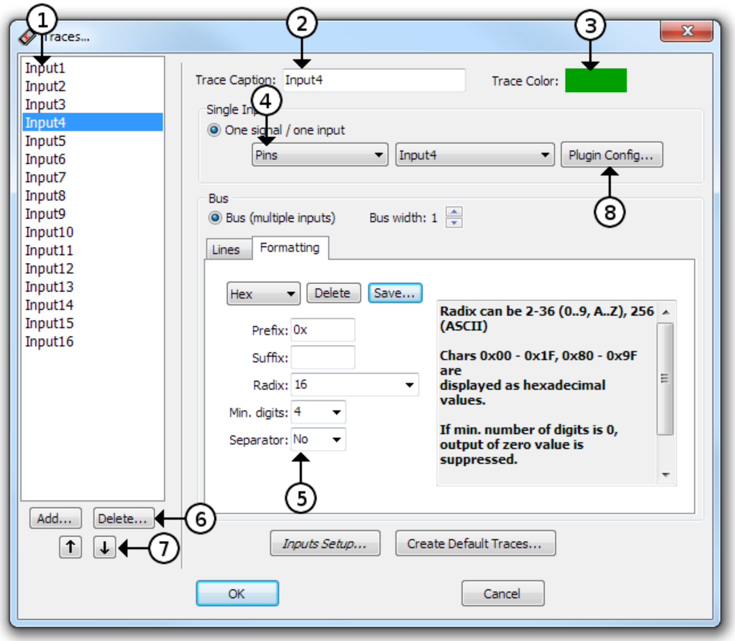
The term **input pin** refers to physical input of SIGMA2 Logic Analyzer. The logic analyzer use fixed input threshold levels compatible with TTL or 5V / 3.3V CMOS. If a particular input is not required by user, but its value differs from logic 0, SIGMA2 can disable some of the unused inputs to save amount of required memory for data acquisition. If the unused pins are in logic 0 (weak internal pull-down will guarantee this), the amount of saved memory is negligible.

The number and placement of used input pins can be selected in *Inputs Dialog*. The dialog can be opened using ***Settings → Inputs*** menu or using *I* hotkey.

4.3 Traces

The term **trace** refers to visualization of acquired data. A trace can be composed of several inputs as well, otherwise a single input may be used in multiple traces, e.g. it is possible to visualize several inputs as a bus while still having the possibility to display individual signals.

Traces are defined in *Traces dialog*. The dialog can be opened by double-clicking a name of each trace in the main viewer window, using ***Settings → Traces*** menu or using **Ctrl+T** hotkey.



Trace Caption: Input4

Trace Color: 🟩

● One signal / one input

Pins ▾ Input4 ▾

● Bus (multiple inputs) Bus width: 1

Prefix: 0x

Suffix:

Radix: 16

Min. digits: 4

Separator: No

Radix can be 2-36 (0..9, A..Z), 256 (ASCII)

Chars 0x00 - 0x1F, 0x80 - 0x9F are displayed as hexadecimal values.

If min. number of digits is 0, output of zero value is suppressed.

Fig.9: Traces Dialog

Traces Dialog

1 **List of traces**
The selected trace is being edited.

2 **Trace caption**
Trace caption can have arbitrary name. Common negation characters in the expression are treaded as a negation.

3 **Trace color**
The color is mixed with color for logic 0 and logic 1.

4 **Input selector**
When the selector selects a decoder, it can be directly configured via *Plugin Config...* button (**8**).

5 **Bus number setup**
Any textual *prefix* and *suffix* can be selected, *radix* can be in range of 2 to 36.

6 **Add and Delete Trace buttons**
Click the button to add a new trace or delete one.

7 **Trace move buttons**
Click the button to move the trace up and down. The shortcut to use the buttons is **Shift+↑** and **Shift+↓**.

8 **Plugin Config Dialog**
When a decoder is selected as source of the data, the configuration of the decoder can be invoked by pressing the *Plugin Config Dialog* button.

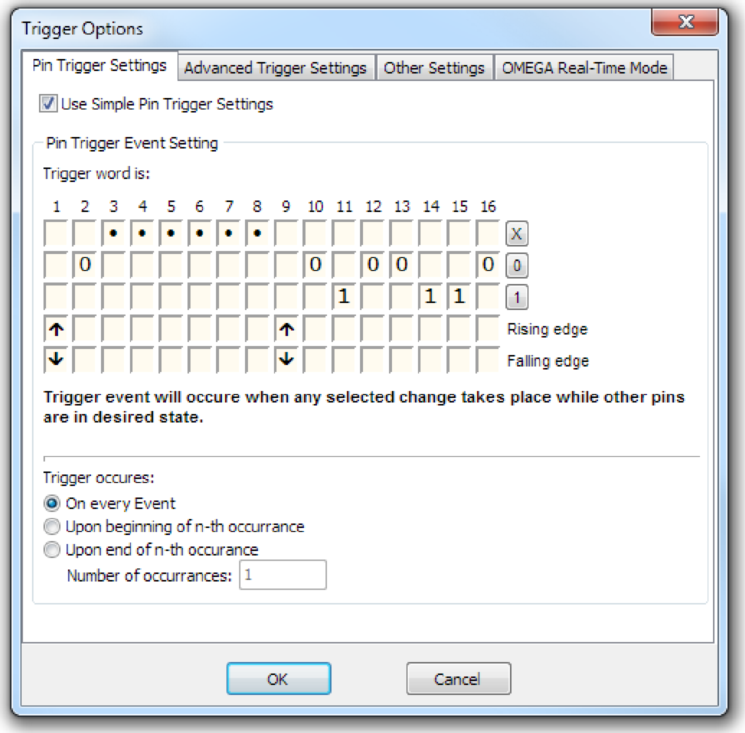
If a trace is defined as a bus, the value on the bus will be displayed according to configurable formatting. Radix from 2 to 36 can be used to format the value as a number using alphabetical characters A-Z for digits 10-35. There is also special formatting option for displaying data as ASCII characters; values which do not represent a printable character in selected set are shown as hexadecimal numbers. The output may be prefixed by a text, suffixed by a text or padded with zeros from the left to given particular width and likewise digit grouping can be used.

4.4 Trigger Settings

Trigger options are defined in the *Trigger settings dialog*. The dialog can be invoked from menu by ***Settings → Trigger Setup*** or using a **T** hotkey.

Availability of certain trigger settings depends on clock settings. For higher sample rates, only basic trigger on an edge of a selected input signal is available. In the other modes either a pin trigger or an advanced trigger can be used. The advanced trigger allows user to set up precise specification of a trigger condition.

4.4.1 Basic Trigger Settings



Trigger Options

Pin Trigger Settings | Advanced Trigger Settings | Other Settings | OMEGA Real-Time Mode

☑ Use Simple Pin Trigger Settings

Pin Trigger Event Setting

Trigger word is:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
		•	•	•	•	•	•									X
	0								0		0	0			0	0
										1			1	1		1
↑								↑								Rising edge
↓								↓								Falling edge

Trigger event will occure when any selected change takes place while other pins are in desired state.

Trigger occures:

● On every Event
○ Upon beginning of n-th occurrance
○ Upon end of n-th occurance
Number of occurrances: 1

OK Cancel

Fig.10: Basic Trigger Settings

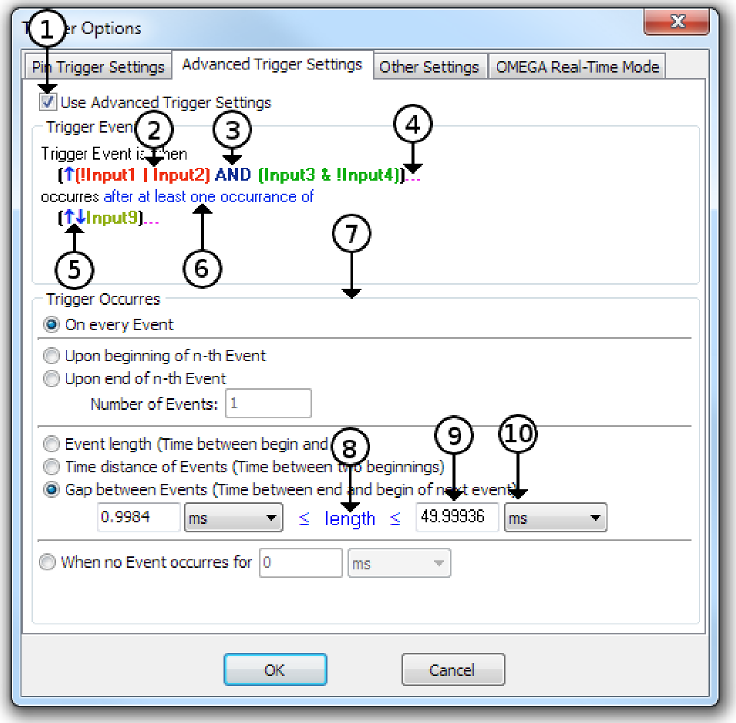
Basic trigger settings define a trigger event as a combination of desired levels and edges on input pins.

Note: Although setting the edge detector on more than one input pin lacks a little sense, it is possible to set the edge detector on up to two input pins. The acquisition is then triggered only when an edge is detected within one clock period of the detector logic, which is 20 ns. This can be somehow useful for race condition hunting, but the probability of the detection is disputable.

The trigger can occur immediately (occurs as soon as the defined combination turns up) or delayed by a counter.

4.4.2 Advanced Trigger Settings

Advanced trigger settings define the trigger event by a set of boolean expressions in combination with an advanced event and delay counter.



Trigger Options

Pin Trigger Settings | Advanced Trigger Settings | Other Settings | OMEGA Real-Time Mode

☑ Use Advanced Trigger Settings

Trigger Event

Trigger Event is then
(↑(!Input1 | Input2) AND (Input3 & !Input4))...
occurres after at least one occurrance of
(↑↓Input9)...

Trigger Occurres

● On every Event
○ Upon beginning of n-th Event
○ Upon end of n-th Event
Number of Events: 1

○ Event length (Time between begin and
○ Time distance of Events (Time between two beginnings)
● Gap between Events (Time between end and begin of next event)
0.9984 ms ≤ length ≤ 49.99936 ms

○ When no Event occurres for 0 ms

OK Cancel

Fig.11: Advanced Trigger Settings

Advanced Trigger Dialog Overview

1	**Advanced Trigger Selector**
2	**Trigger** ***Mask***

The mask made of any number of inputs can be selected. The inputs in the mask can be either *ANDed*, *ORed*, *NANDed* or *NORed*, but not their combination.

3 **Boolean** *Function* **between** *Masks*
There can be any boolean function of list *AND*, *NAND*, *OR*, *NOR*, *XOR*, *XNOR*.

4 **Adder of new** *Functions* **and** *Masks*
A new *function* and a *mask* can be added by clicking the *adder* mark.

5 **Inversion or Edge** *Detector*
Nothing, inversion or any edge detectors can be set up here. The detector include *rising edge*, *falling edge* and *any edge*.

6 *Precondition* **toggle button**
A precondition can be enabled and disabled by clicking on the *precondition* button. The trigger is then detected only when a boolean condition is matched after at least one occurrence of the precondition.

7 **Advanced event and delay** *Counter*
The event and delay counter enables a time and count related condition.

8 **Lower than / Higher than / Constraints toggle button**
By clicking the *length* button the condition will toggle between a *Lower than* / *Higher Than* / *Constraints* condition.

9 **Value with prescaler**
The value of the counter. Due to a prescaler, the value may be rounded to nearest value achievable with the prescaler.

10 **Unit selector**

Note: There are three *masks* available for the *condition* and *precondition* in total. The *condition* must use at least one, therefore at most two *masks* are available for the *precondition*.

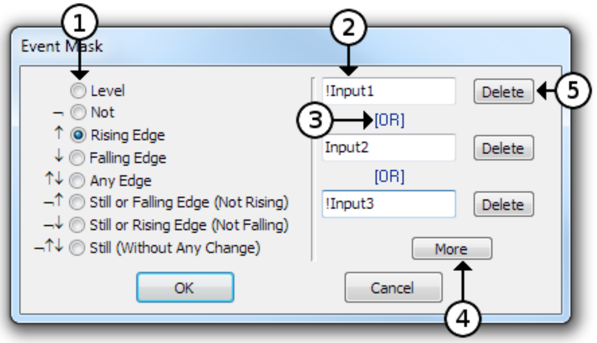


Fig.12: Advanced Trigger Masks

Mask selection

1 **Edge and inversion selector**

2 *Input* **or** *trace Term*
The *term* can also have negation and in case of busses constant comparison[2].

3 **Mask function selector**
The *selector* can be either *AND* or *OR*.

4 **Add new input** *term*

5 **Delete one** *term*

Although this approach makes the description of very complex situations possible, accordingly it allows to define the moment to be captured precisely, there are certain limitations determined by capabilities of the hardware. If the expression is too complex to be implemented in the SIGMA2 Logic Analyzer hardware, an exclamation icon appears to indicate this fact.

4.4.3 Trigger Position within Acquisition

The *Post-Trigger Time* can be selected on the ***Other Trigger Settings*** tab of the ***Trigger Settings*** dialog. This *Post-Trigger Time* selects the amount of logic analyzer **memory** which can be used after the acquisition was triggered. The rest of the memory is used as *Pre-*

Trigger Time. If the trigger was detected before the whole *Pre-Trigger Time* memory was wasted, the remaining memory will not be used, even for the *Post-Trigger* acquisition. If the amount of the memory used is greater than the *Pre-Trigger* memory amount, the very beginning of the acquisition is dropped.

The *Post-Trigger Time* setting can be set from range of 1-99% with resolution of 1%. The accuracy of the setting is within ±1%.

4.4.4 External Triggering

The SIGMA2 Logic Analyzer comes with *Trigger In* and *Trigger Out* pins (on the SIGCAB20 cable described as ***TI*** and ***TO***). The *Trigger Out* can be configured as 3.3V CMOS output with impedance 1kΩ with negative or positive polarity or as an open collector output. The *Trigger Out* pulse length can be set to either 1 µs or 1 ms.

The *Trigger In* input can be configured as positive or negative polarity. The function of *Power Out* can be configured on *Trigger In*.

The source of the activation of the *Trigger Out* pin can be selected from variety of sources:

- By external *Trigger In*.
- By trigger by trigger condition.
- By trigger by Go button.
- By trigger in PC software.
- During acquisition.
- During acquisition after it has been triggered.

	Min.	Typ.	Max.	
V_{IL} input low voltage			0.8	V
V_{IH} input high voltage	2.0			V
V_{IN} absolute rating, trigger I/O	-0.3		3.6	V
V_{PO} power output on Trigger In		3.3		V
I_{PO} power output on Trigger In			100	mA

Table 5: Electrical Specification on the Trigger In/Out Pin

Warning: The absolute maximum voltage on the *Trigger In* and *Trigger Out* pins is 3.6V.

4.4.5 Other Trigger Settings

During normal acquisition, the acquisition is triggered by first occurrence of the trigger from beginning of the acquisition. If any successive triggers are detected, they are ignored by the *Post-Trigger* acquisition termination logic, but the *Trigger LED* can be configured to either blink on every detected trigger or only on the first trigger launching the acquisition.

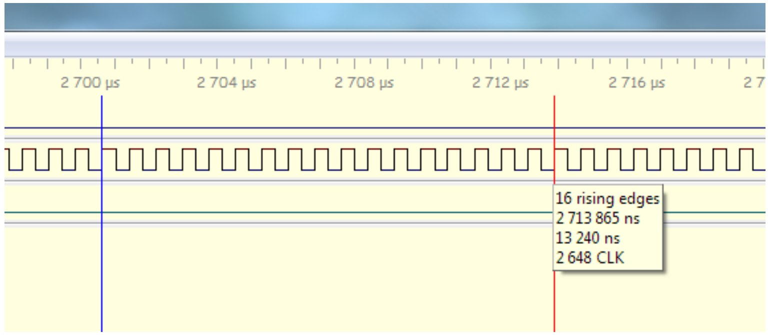
4.5 Working with the acquired data

4.5.1 Navigation and analysis

Navigation in the main viewer can be controlled by keyboard, mouse or by a combination of both.

Action	Key or mouse action
Viewer window sliding along the time axis	← or →
	mouse wheel
	Ctrl and mouse move
Zoom	**+** or **-**
	Ctrl and mouse wheel
	Select by mouse drag
Undo last zoom / move	**Backspace**
Zoom 50× in	*****
Zoom whole acquisition[3]	**/**
Jump to acquisition end[3]	**End**
Jump to trigger[3]	**Home**
Move mouse to another trace	↑ or ↓
Jump to next edge on selected trace	**Alt**+→ or **Alt**+←
Place bookmark	**Ctrl**+**Shift**+**0** to **9**
Jump to bookmark	**Ctrl**+**0** to **9**
Place marker	**Space**
Count number of edges	**Q**
Toggle between period and frequency	**F**
Display counting options	**QQ**

Table 6: Mouse and Keyboard action for navigation and analysis



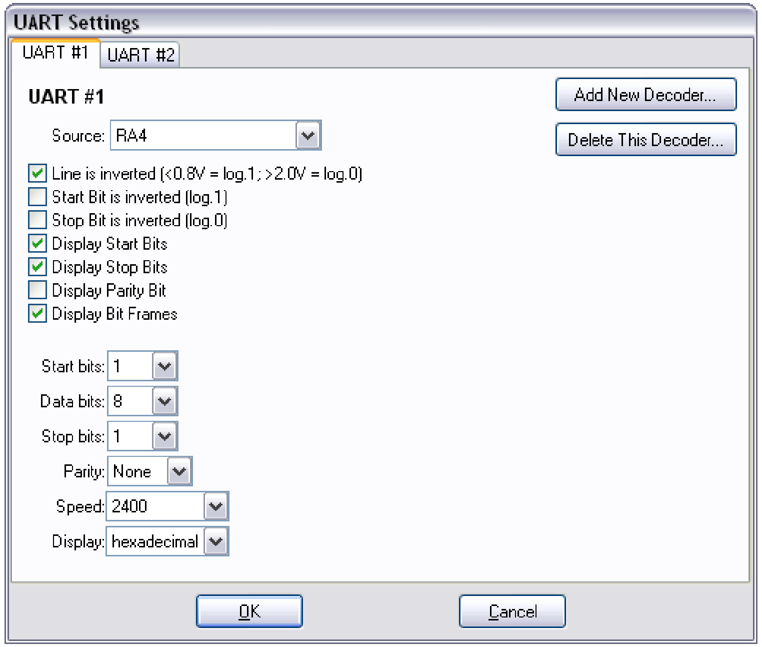
16 rising edges
2 713 865 ns
13 240 ns
2 648 CLK

Fig.13: Counting edges in an acquisition

Note: Several functionality of the software is coded in plugins. All described functionality is within plugins distributed together with the software. By disabling or replacing the plugins, the functionality will differ.

4.5.2 UART Protocol Decoder

This decoder decodes captured UART signal and displays ASCII characters, decimal or hexadecimal values.



UART #1

Source: RA4

☑ Line is inverted (<0.8V = log.1; >2.0V = log.0)
☐ Start Bit is inverted (log.1)
☐ Stop Bit is inverted (log.0)
☑ Display Start Bits
☑ Display Stop Bits
☐ Display Parity Bit
☑ Display Bit Frames

Start bits: 1
Data bits: 8
Stop bits: 1
Parity: None
Speed: 2400
Display: hexadecimal

Fig.14: UART Decoder

Several options can be set up:

Input

The input pin.

Line Polarity

The polarity of the line. This is useful when using direct connection of voltage limited RS-232 (be aware of maximum ratings on SIGMA2 pins).

Start Bit Inversion

The polarity of the start bit and idle bus logic level.

Stop Bit Inversion

The polarity of the stop bit.

Bit Frames Visibility

This option enables display of bit frames.

Data Bits

The number of start bits can be configured from **1** to **16**.

Start Bits

The number of start bits can be configured to **1** or **2**.

Stop Bits

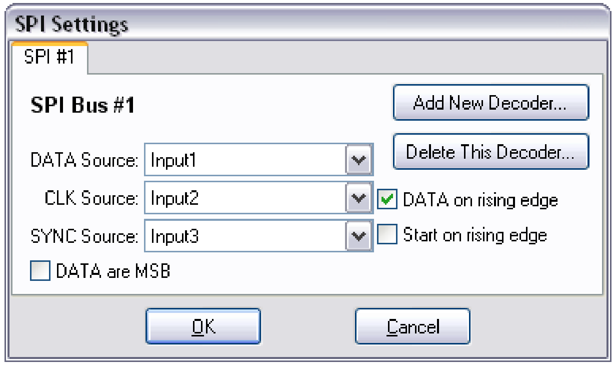
The number of stop bits can be configured to **0**, **1** or **2**.

Parity

The decoder can check parity bit. The parity can be set to **None**, **Even**, **Odd**, **Mark**, **Space**.

4.5.3 SPI Bus Decoder

This decoder decodes captured SPI signal and displays it as hexadecimal values. For correct function it is necessary to set up an appropriate data input, a clock input and an input that trigger counting of bits in a byte.



SPI Settings

SPI #1

SPI Bus #1

DATA Source: Input1

CLK Source: Input2

SYNC Source: Input3

☑ DATA on rising edge

□ Start on rising edge

□ DATA are MSB

Fig.15: SPI Plugin settings

Several options can be set up:

Input pins

Three inputs pins for *Data*, *Clock* and *Synchronization* (i.e. -CS).

Clock polarity

Data on *rising* or *falling* edge of Clock.

Data Sequence

The data sequence *MSB first* or *LSB first*.

Sync polarity

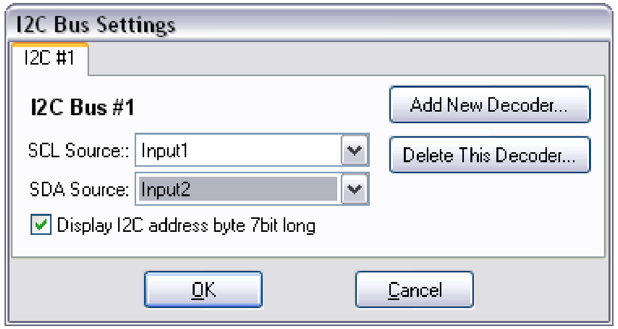
The edge on which is started decoding can be set to **rising** or **falling** edge.

Field bit length

Allows user to define length of each field in order how they come after start of the frame (SYNC). One or more last fields can be closed in brackets indicating repetition (Ex.: 12,(8,16)).

4.5.4 I2C Bus decoder

This decoder decodes captured I2C signal and displays start bits, stop bits, addresses, acknowledge bits and data in hexadecimal values.



I2C Bus #1

SCL Source:: Input1

SDA Source: Input2

☑ Display I2C address byte 7bit long

Fig.16: I2C Plugin settings

Several options can be set up:

Input pins
Two input pins for **SDA** and **SCL** signal.

Display address 7bit long
There are two possibilities how to display I2C address: with or without the LSB displayed (e.g. A0/A1 device address is displayed as A0W/A1R in first case or 50W/50R in the latter).

4.5.5 USB 1.1 Analyzer Plugin

The USB 1.1 Analyzer decodes captured signals as USB 1.1 signals. Reading of the USB specification is highly recommended before using the USB decoder.

First of all a new decoder must be added in the ***Settings* → *Plugin Settings* → *USB Plugin Configurations*** menu using the *Add New Decoder* button and the captured signals to be decoded must be chosen. After the OK button is pressed, a decoded data window is opened. The data decoding is triggered by menu ***Other* → *Decode Now!*** or by the **F9** key press. The communication are decoded automatically after the data has been completely downloaded from the analyzer if the field "*Decode protocol automatically upon data download*" is checked in the settings.

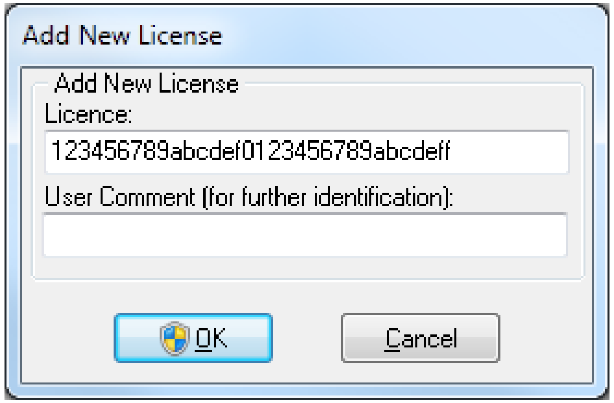
After decoding, the communication is displayed in a tree structure where all the packets are listed. The decoded packets can be itemized to the bits level. After clicking the decoded packet or some of its part, the appropriate part of the captured traces are highlighted. After right mouse button clicking a ***Zoom*** function from the local menu can be used. The ***Search* → *Find...*** function of the main menu provides various possibilities how to search in the decoded data.

Note: It is required that a license is bought for the USB plugin functionality. The license is assigned to the logic analyzer hardware.

Note: A dedicated hardware probe for easy connection of the USB signals to the logic analyzer can be purchased optionally. It is equipped with two USB A connectors (plug and receptacle) and pins for the logic analyzer connection. The logic analyzer can be connected either directly to the USB signals or to the buffered USB signals. On the USB cable there must be found a suitable position for the probe, where the captured signals are the best quality.

Installation

The USB decoder has been drawn up as one of the plugins and it is a part of the installation package and so there is no need to install it separately. For setting up the license hit ***License* → *Install New License...*** in ASIX SIGMA & OMEGA Logic Analyzers application main menu.


Add New License

Add New License

Licence:

123456789abcdef0123456789abcdeff

User Comment (for further identification):

Fig.17: Adding a new license

What to measure

With USB decoder USB 1.5 Mbps (Low-Speed) and 12 Mbps (Full-Speed) data communication captured by the SIGMA2 Logic Analyzer can be analyzed. The logic analyzer is not capable of capturing higher data rates (High-Speed, Super-Speed).

Measure tool attachment

Although the USB data communication is partly differential, GND and both USB data signal (DATA +,DATA-) must be connected to the SIGMA2 Logic Analyzer. SIGMA2 samples that signal with enough accuracy as a common TTL signal. Due to NRZI coding, which is used by USB, it is not needed to distinguish between DATA+ and DATA-, so they are interchangeable. Unfortunately there are some states on the USB using single-ended signaling (*Bus Reset* and *End-Of-Packet*). This is the reason why the connecting of both signals DATA+ and DATA- is required. Connecting of single signal DATA line is not enough. The data rate is chosen by swapping the data lines. Behind the USB hub on the lines to a Low-Speed (1.5 Mbps) device there is only Low-Speed communication, whereas on the lines to Full-Speed (12 Mpbs) device there are both Low-Speed and Full-Speed communication observable. The 480 Mbps communication is called High-Speed and SIGMA2 is not capable to measure that.

The *USBprobe* includes two 74AHCT125 TTL gates and two USB connectors of type A and B wired so that it acts as a USB extension. The SIGMA2 Logic Analyzer can be connected before or beyond the buffer, it depends on an application – better to try. Generally, best results are reached with USBprobe connected directly to an USB hub and the shorter USB cable, the better. Likewise it is important to shorten the way between USBprobe and SIGMA2.

On *USBprobe* there are 5V directly accessible from PC (through a 800mA irreversible fuse) – avoid a shortage! *It is highly inadvisable to connect USBprobe directly to PC ports*. We suggest using a USB hub with an external power source and connecting *USBprobe* directly into the hub.

Measuring

Measuring is possible only with a purchased license.

The USB data signals (DATA+, DATA-) can be connected to any two inputs and the other inputs can by utilized for measuring of any other signals (e.g. for measuring of another USB communication). It is possible to analyze more than one USB communication.

Processing

When signals have been captured (= the acquisition has been done) it is necessary to decode it. It can take tens of seconds depending on amount of the captured data. Decoding is proceed automatically, promptly when the data have been captured, if it is enabled in the USB analyzer settings dialog or it can be launch in the menu by choosing ***Decode → Decode Now!*** or hitting **F9**.

When decoding have finished there is a list of measured

USB events in the events window (which is simultaneously the main window).

Viewing

After the decoding have finished the list of measured USB events is in the events window.

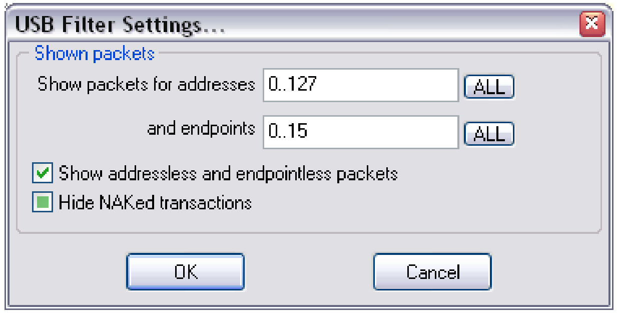
To decrease the number of events in the event window a configurable filter can be applied so that only requested events are displayed. For that option open the menu ***Settings → Filter Settings***. One or more USB addresses can be marked. The formatting can contain single addresses or address ranges (e.g. *`0,5..7`*) within range of 0 to 127 (USB uses 7bit wide address range). The same principle applies to endpoints within range of 0 to 15 (the endpoint direction – bit 7 - does not matter).

The *address zero* is dedicated for the device without any address yet. The *zero endpoint* is special *control endpoint*, which each device must implement. It is the only endpoint with bi-directional transactions.

Due to the fact that the USB specification does not allow devices to send data by their own, the major part of the traffic is occupied by a master device (PC) asking slave devices whether they have some data to be transferred.

Therefore it can be useful to mask out the frames without any useful data (terminated by a *NAK* token). For doing that, as well as masking frames which are not designated for any particular device (e.g. a *Start-Of-Frame* token, *Bus Reset*), use the menu ***Settings → Filter Settings***.

On the highest level of the decoded data tree there is a possibility to display or hide transactions ended by a *NAK* token or a *ACK* token, for that option use the right mouse button and control the filter from the local menu.



Show packets for addresses 0..127

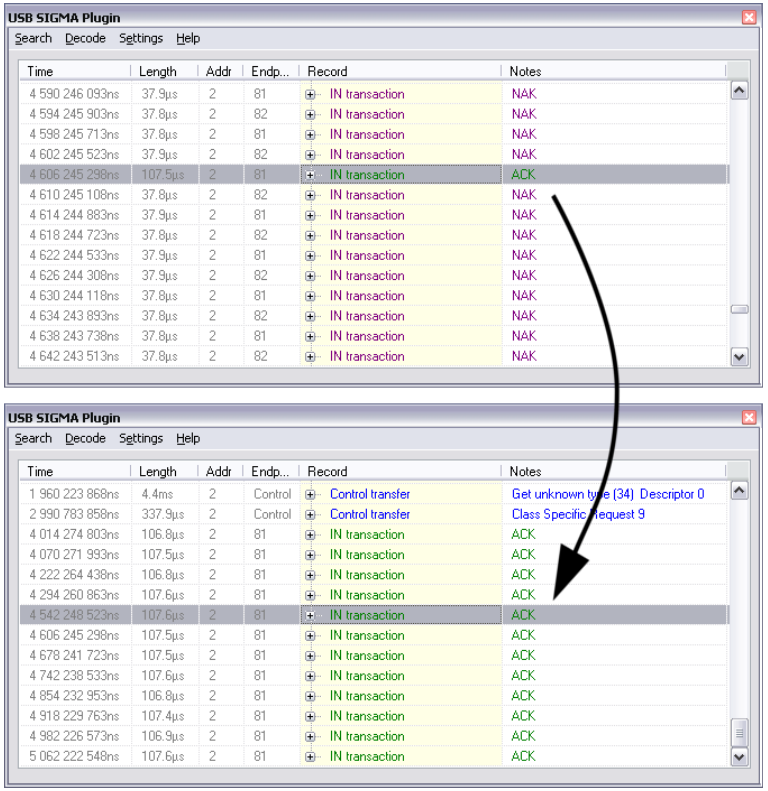
and endpoints 0..15

☑ Show addressless and endpointless packets

□ Hide NAKed transactions

Fig.18: USB Filter settings

Another way how to pop-up the dialog *Settings → Filter Settings* menu is to click on the title of *Addr* or *Endpoint* column.



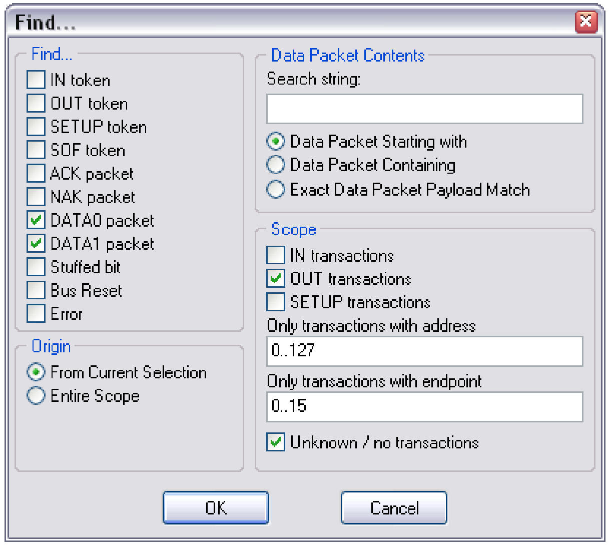
Time	Length	Addr	Endp...	Record	Notes
4 590 246 093ns	37.9µs	2	81	IN transaction	NAK
4 594 245 903ns	37.8µs	2	82	IN transaction	NAK
4 598 245 713ns	37.8µs	2	81	IN transaction	NAK
4 602 245 523ns	37.9µs	2	82	IN transaction	NAK
4 606 245 298ns	107.5µs	2	81	IN transaction	ACK
4 610 245 108ns	37.8µs	2	82	IN transaction	NAK
4 614 244 883ns	37.9µs	2	81	IN transaction	NAK
4 618 244 723ns	37.8µs	2	82	IN transaction	NAK
4 622 244 533ns	37.9µs	2	81	IN transaction	NAK
4 626 244 308ns	37.9µs	2	82	IN transaction	NAK
4 630 244 118ns	37.8µs	2	81	IN transaction	NAK
4 634 243 893ns	37.8µs	2	82	IN transaction	NAK
4 638 243 738ns	37.8µs	2	81	IN transaction	NAK
4 642 243 513ns	37.8µs	2	82	IN transaction	NAK

Time	Length	Addr	Endp...	Record	Notes
1 960 223 868ns	4.4ms	2	Control	Control transfer	Get unknown type (34) Descriptor 0
2 990 783 858ns	337.9µs	2	Control	Control transfer	Class Specific Request 9
4 014 274 803ns	106.8µs	2	81	IN transaction	ACK
4 070 271 993ns	107.5µs	2	81	IN transaction	ACK
4 222 264 438ns	106.8µs	2	81	IN transaction	ACK
4 294 260 863ns	107.6µs	2	81	IN transaction	ACK
4 542 248 523ns	107.6µs	2	81	IN transaction	ACK
4 606 245 298ns	107.5µs	2	81	IN transaction	ACK
4 678 241 723ns	107.5µs	2	81	IN transaction	ACK
4 742 238 533ns	107.6µs	2	81	IN transaction	ACK
4 854 232 953ns	106.8µs	2	81	IN transaction	ACK
4 918 229 763ns	107.4µs	2	81	IN transaction	ACK
4 982 226 573ns	106.9µs	2	81	IN transaction	ACK
5 062 222 548ns	107.6µs	2	81	IN transaction	ACK

Fig.19: Window with hidden transactions which are ended with NAK

Searching

For searching for a specific type of packet or event (*Bus Reset*, any error, *Stuffed Bit*) open the ***Search → Find...*** menu or hit **Ctrl+F** and then for another occurrence hit the **F3** key.



Find...
- □ IN token
- □ OUT token
- □ SETUP token
- □ SOF token
- □ ACK packet
- □ NAK packet
- ☑ DATA0 packet
- ☑ DATA1 packet
- □ Stuffed bit
- □ Bus Reset
- □ Error

Origin
- ● From Current Selection
- ○ Entire Scope

Data Packet Contents

Search string:

- ● Data Packet Starting with
- ○ Data Packet Containing
- ○ Exact Data Packet Payload Match

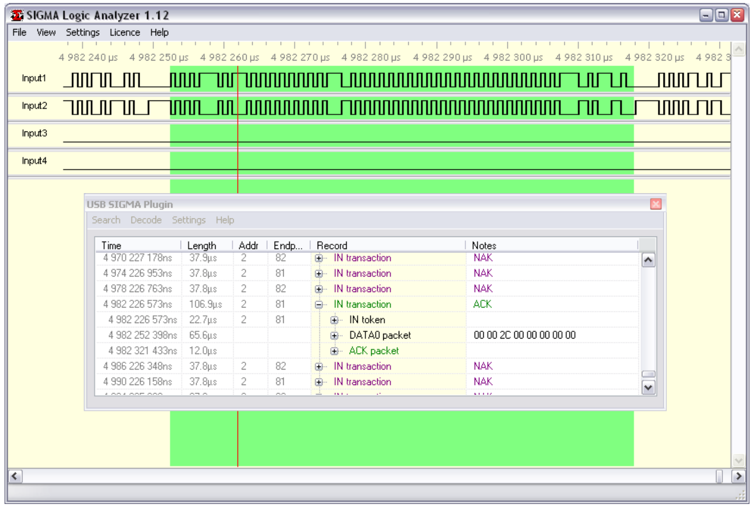
Scope
- □ IN transactions
- ☑ OUT transactions
- □ SETUP transactions

Only transactions with address: 0..127

Only transactions with endpoint: 0..15

- ☑ Unknown / no transactions

Fig.20: Searching Window



Time	Length	Addr	Endp...	Record	Notes
4 970 227 178ns	37.9µs	2	82	IN transaction	NAK
4 974 226 953ns	37.8µs	2	81	IN transaction	NAK
4 978 226 763ns	37.8µs	2	82	IN transaction	NAK
4 982 226 573ns	106.9µs	2	81	IN transaction	ACK
4 982 226 573ns	22.7µs	2	81	IN token	
4 982 252 398ns	65.6µs			DATA0 packet	00 00 2C 00 00 00 00 00
4 982 321 433ns	12.0µs			ACK packet	
4 986 226 348ns	37.8µs	2	82	IN transaction	NAK
4 990 226 158ns	37.8µs	2	81	IN transaction	NAK

Fig.21: DATA0 Packet Highlighted

If data packets (DATA0, DATA1) are being searched, searching can be limited to particular endpoint, device address or by a hexadecimal string.

Linking the events window with the analyzer window

When a particular USB event has been chosen the real place of the occurrence is highlighted in the analyzer window. The place can be also zoomed in by hitting the right mouse button and then choosing ***Zoom***. Hitting the right mouse button in the analyzer window and then choosing ***Lookup*** in USB Communication highlights position in the events window.

Gathering of related communication into trees

In the basic USB decoder settings, related consecutive events are gathered into a tree (e.g. whole *Control Transfer*). This behavioral leeds to potential ambiguity in the order of USB events. In this case ***Flat Decoding*** can be chosen in the ***Settings → Settings...*** menu. It disables gathering of USB events so they are sorted top-down strictly by the time.

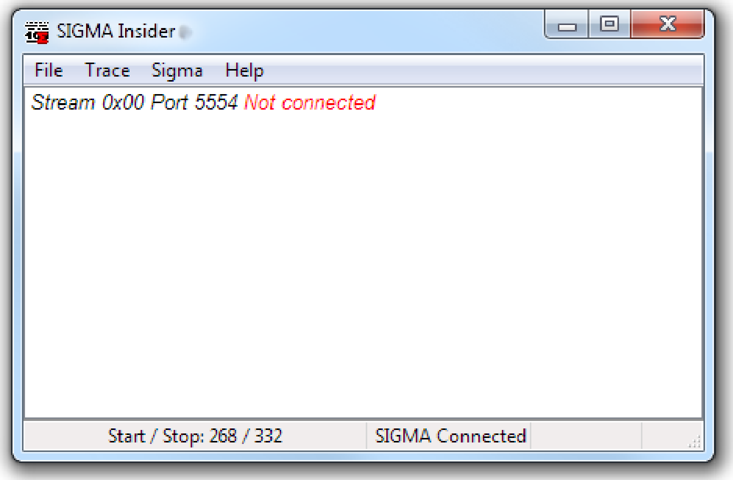
4.6 Auxiliary functions

The SIGMA2 Logic Analyzer have some useful auxiliary functions. This functions are standalone utilities accessible from ***Start menu***.

4.6.1 Insider

SIGMA2 Insider is a tool for real-time sniffing of most common busses and streaming them to a TCP/IP port. The output stream can be displayed by a common terminal program, such as PuTTY. Home page of PuTTY is http://www.chiark.greenend.org.uk/~sgtatham/putty/. SIGMA2 Insider supports additionally displaying contents of up to 256 shift registers as debug outputs.

SIGMA2 Insider is controlled using a standalone application. The application contains window for displaying contents of shift registers called *traces*. Some traces can be set up as streams, redirected to a TCP/IP and can be used as debug outputs. The number of streams is limited, because they require buffering.



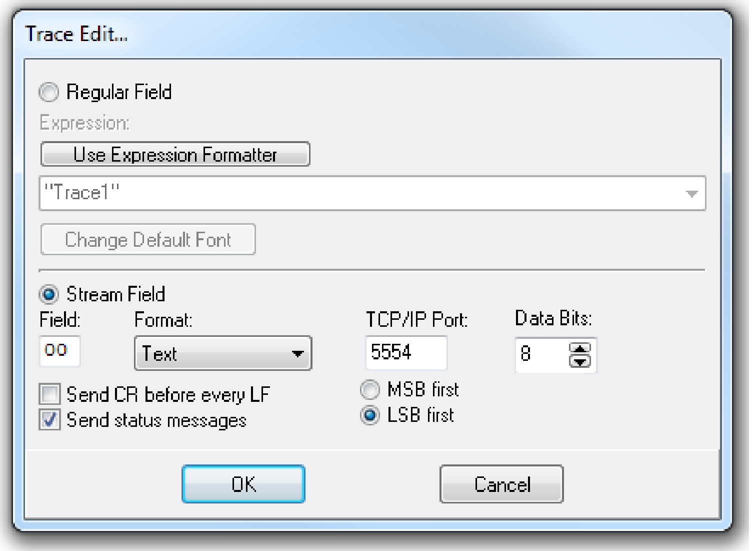
Stream 0x00 Port 5554 Not connected

Start / Stop: 268 / 332

SIGMA Connected

Fig.22: SIGMA Insider main window

Traces are controlled using menu ***Trace*** to add, modify or delete them.



Trace Edit...

○ Regular Field

Expression:

Use Expression Formatter

"Trace1"

Change Default Font

● Stream Field

Field: 00

Format: Text

TCP/IP Port: 5554

Data Bits: 8

□ Send CR before every LF

☑ Send status messages

○ MSB first

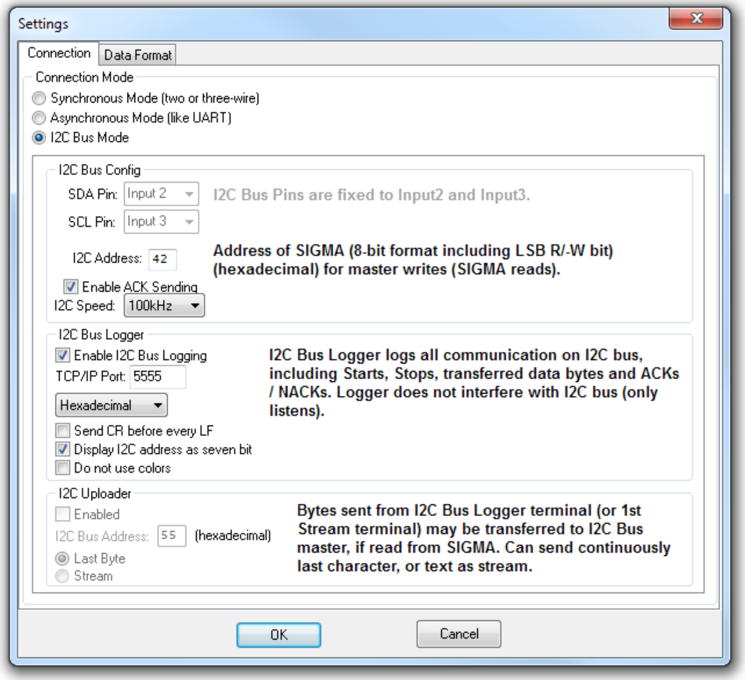
● LSB first

OK Cancel

Fig.23: SIGMA Insider trace window

To use the trace as a debug output, use the option *Stream Field*, if the trace should be displayed in the window, use option *Regular Field*. If the *field numbering* is not used (default), always enter value 0. The *field numbering* can be enabled in menu ***Sigma → Insider Setup...***, tab ***Data Format***. The field number can be then selected to be consisted of arbitrary number of first bits in the message (on byte-oriented busses it should be set to 8 bits).

SIGMA2 Insider can work also as a I^{2}C Bus Logger.



Settings

Connection | Data Format

Connection Mode

○ Synchronous Mode (two or three-wire)

○ Asynchronous Mode (like UART)

● I2C Bus Mode

I2C Bus Config

SDA Pin: Input 2

SCL Pin: Input 3

I2C Bus Pins are fixed to Input2 and Input3.

I2C Address: 42

Address of SIGMA (8-bit format including LSB R/-W bit) (hexadecimal) for master writes (SIGMA reads).

☑ Enable ACK Sending

I2C Speed: 100kHz

I2C Bus Logger

☑ Enable I2C Bus Logging

TCP/IP Port: 5555

Hexadecimal

□ Send CR before every LF

☑ Display I2C address as seven bit

□ Do not use colors

I2C Bus Logger logs all communication on I2C bus, including Starts, Stops, transferred data bytes and ACKs / NACKs. Logger does not interfere with I2C bus (only listens).

I2C Uploader

□ Enabled

I2C Bus Address: 55 (hexadecimal)

● Last Byte

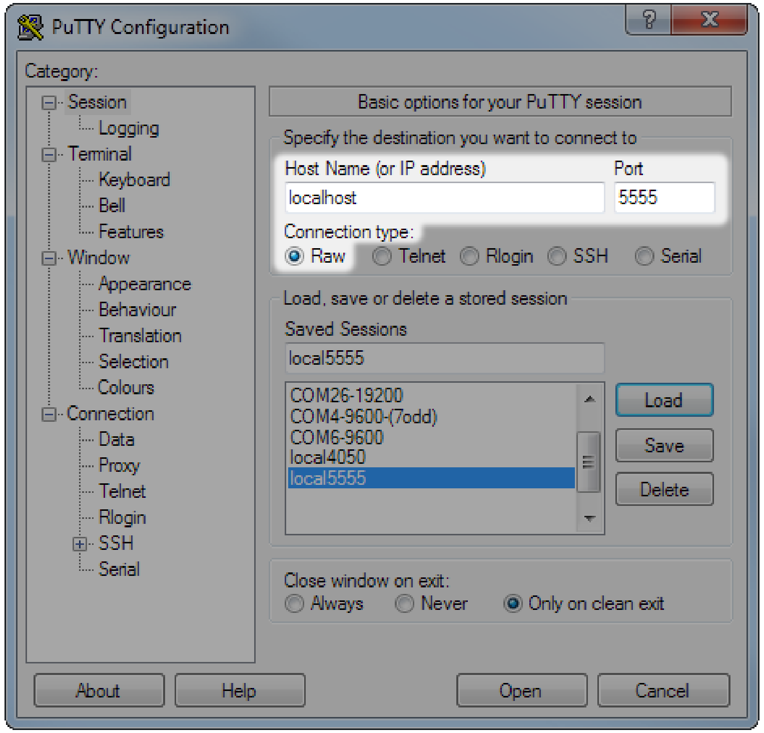
○ Stream

Bytes sent from I2C Bus Logger terminal (or 1st Stream terminal) may be transferred to I2C Bus master, if read from SIGMA. Can send continuously last character, or text as stream.

OK Cancel

Fig.24: SIGMA Insider setup

The debugging or logging texts are available on a selected TCP/IP port. Connect to the port with your favorite terminal application.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): localhost

Port: 5555

Connection type: ● Raw ○ Telnet ○ Rlogin ○ SSH ○ Serial

Load, save or delete a stored session

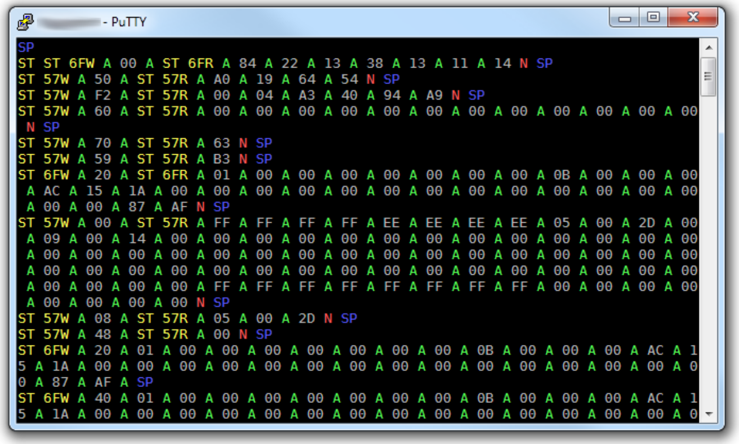
Saved Sessions: local5555

COM26-19200
COM4-9600-(7odd)
COM6-9600
local4050
local5555

Close window on exit: ○ Always ○ Never ● Only on clean exit

Fig.25: Opening Insider Connection in PuTTY

The picture above shows the basic options required to open the connection in the PuTTY terminal application. For connecting within one PC, as host use "*localhost*". The host can be, if required, filled with any other host reachable on the network.



```
SP
ST ST 6FW A 00 A ST 6FR A 84 A 22 A 13 A 38 A 13 A 11 A 14 N SP
ST 57W A 50 A ST 57R A A0 A 19 A 64 A 54 N SP
ST 57W A F2 A ST 57R A 00 A 04 A A3 A 40 A 94 A A9 N SP
ST 57W A 60 A ST 57R A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00
 N SP
ST 57W A 70 A ST 57R A 63 N SP
ST 57W A 59 A ST 57R A B3 N SP
ST 6FW A 20 A ST 6FR A 01 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 0B A 00 A 00 A 00
 A AC A 15 A 1A A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00
 A 00 A 00 A 87 A AF N SP
ST 57W A 00 A ST 57R A FF A FF A FF A FF A EE A EE A EE A EE A 05 A 00 A 2D A 00
 A 09 A 00 A 14 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00
 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00
 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00
 A 00 A 00 A 00 A 00 A FF A FF A FF A FF A FF A FF A FF A FF A 00 A 00 A 00 A 00
 A 00 A 00 A 00 A 00 N SP
ST 57W A 08 A ST 57R A 05 A 00 A 2D N SP
ST 57W A 48 A ST 57R A 00 N SP
ST 6FW A 20 A 01 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 0B A 00 A 00 A 00 A AC A 1
5 A 1A A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 0
0 A 87 A AF A SP
ST 6FW A 40 A 01 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 0B A 00 A 00 A 00 A AC A 1
5 A 1A A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 00 A 0
```

Fig.26: Example I2C Log in PuTTY window

In the picture there is a PuTTY output of an example logging on an I^{2}C bus with multiple slave devices.

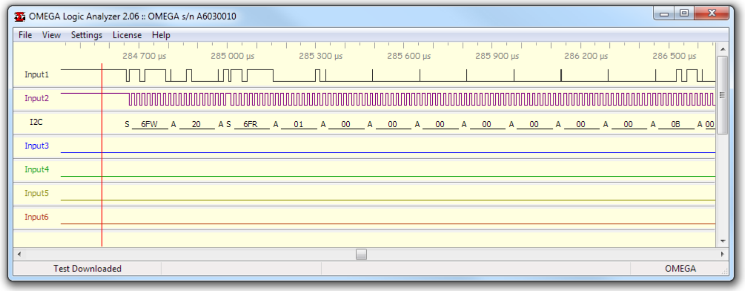
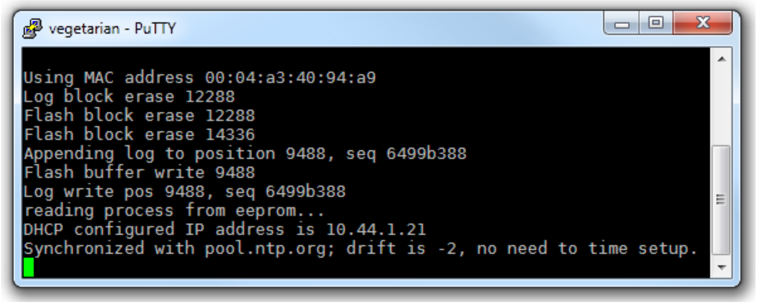


Fig.27: Example I2C Bus activity

The same I^{2}C Bus activity as it was measured with the logic analyzer. The activity is on the line 9 of the PuTTY window.

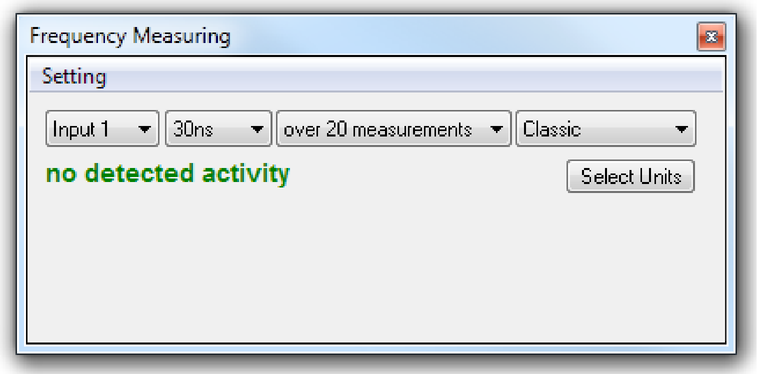


```
Using MAC address 00:04:a3:40:94:a9
Log block erase 12288
Flash block erase 12288
Flash block erase 14336
Appending log to position 9488, seq 6499b388
Flash buffer write 9488
Log write pos 9488, seq 6499b388
reading process from eeprom...
DHCP configured IP address is 10.44.1.21
Synchronized with pool.ntp.org; drift is -2, no need to time setup.
```

Fig.28: Example of I2C debug in PuTTY window

4.7 Frequency Measuring

SIGMA2 Frequency Measuring is a tool for measuring a frequency on an input pin. SIGMA2 Frequency Measuring supports measuring on up to four pins simultaneously. SIGMA2 Frequency Measuring is a standalone application.



Input 1 | 30ns | over 20 measurements | Classic

no detected activity

Fig.29: SIGMA2 Frequency Measuring

4.8 Pin View

SIGMA2 Pin View is a logic probe tool for inspecting of the correct connection of the SIGMA2 Logic Analyzer. It can be opened using menu ***View → Pin View...***. The logic probe shows not only the digital logic value, but also shows simplified waveforms if there is some activity.

Note: The operation may not be available during acquisition or some advanced modes of operation.

4.9 Availability of Auxiliary Functions

Logic Analyzer	Mode of operation	Acquisition	Pin View	Insider	Function Generator	Frequency Measuring
SIGMA	50 Msps	16 Inputs	Yes, only used inputs	No	No	No
	100 Msps	8 Inputs, scope Input 1 to 8	Yes, only used inputs, scope Input 1 to 8	No	No	No
	200 Msps	4 Inputs, scope Input 1 to 4	Yes, only used inputs, scope Input 1 to 4	No	No	No
	Synchronous Inputs	15 Inputs + Input 1 or 9 as Clock	Yes, only used inputs except Clock	No	No	No
	Asynchronous Inputs	15 Inputs + any Input as Clock	Yes, only used inputs	No	No	No
	Frequency Measuring	No	No	No	No	Yes, up to four simultaneous measurements
	Insider	No	No	Yes	No	No
OMEGA	200 Msps	16 Inputs	Yes, for all inputs	Yes	Yes, either Input 1 to 8 or Input 9 to 16	Yes, single measurement
	400 Msps	8 Inputs, either Input 1 to 8 or Input 9 to 16	Yes, for all inputs	Yes	Yes, either Input 1 to 8 or Input 9 to 16	Yes, single measurement
	Daisy Chain Master	16 Inputs	Yes, for all inputs	Yes	Yes, either Input 1 to 8 or Input 9 to 16	Yes, single measurement
	Daisy Chain Slave	16 Inputs	Yes, for all inputs	No	No	No
	Synchronous Inputs	15 Inputs + Input 1 as Clock	Yes, for all inputs + frequency measuring for Clock	Yes	Yes, either Input 1 to 8 or Input 9 to 16	Yes, single measurement
	Real-Time mode	16 Inputs	TBD	TBD	TBD	TBD

Table 7: Detailed Availability of Auxiliary Functions

4.10 Using the Application for Automated Acquisition

The logic analyzer application sigmalogan.exe recognize a parameter `-export` . Invoking the application with this parameter will cause the application to start an acquisition immediately using stored parameters (see chapter Using the Application as a Portable Application), wait for acquisition end, download and export the acquisition into selected file using function ***File → Export Current View...***. The the file name given in by parameter `-export` has extension *.stf, the saved file is in STF format.

4.11 Plugins

The ASIX SIGMA & OMEGA Logic Analyzer software features modular design to add functionality according to particular user's needs. This modularity is achieved by using **plugins**.

The plugins are dynamically loadable libraries (*DLL* files) located in the plugins subdirectory of the main program directory. Individual plugins can be enabled or disabled in ***Settings → Plugins*** dialog and configured in ***Settings → Plugin Settings*** (if applicable). Several plugins are part of the ASIX SIGMA&OMEGA APPLICATION PACKAGE by default.

The **Plugin API** is described in a separate document. SIGMAP02 Plugin Developer's Manual

Some data decoded by some plugins (UART, SPI, I^2C) can be inserted among captured signals as a virtual track. For that feature go to the ***Settings → Traces*** Setup menu.

Additional plugins can be provided in the future.

Source codes of some plugins are released under GPL, thus users are free to modify or create new plugins.

[1] Maximum acquisition duration at 50 Msps is 45 minutes. Due to RLE properties, lower sampling rate has benefit **only** when longer duration is demanded.

[2] Examples of the syntax are ***!Input1***, ***Input1=0***, ***BUS=A6***, ***BUS=h'A6'***, ***BUS=b'10100110'***, ***BUS=d'166'***.

[3] Live in Real-Time Mode

5

Using the Logic Analyzer

5.1 Sampling Frequency

The SIGMA2 Logic Analyzer samples at a sampling rate, for example 50 Msps. It means the inputs are sampled every 20 ns.

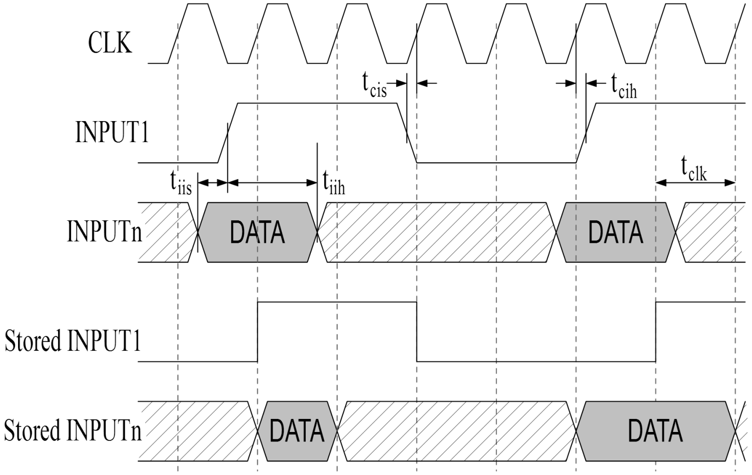


Fig.30: Sampling of SIGMA2 Inputs

	Min.	Typ.	Max.	
t_{clk} Sampling clock period[1]		20		ns
t_{cis}+t_{cih} Data valid window[2]	2.6			ns
t_{iis} Input (data) setup time before input (clock) within one port	3.6			ns
t_{iih} Input (data) hold time after input (clock) within one port	3.6			ns
t_{iispp} Input (data) setup time before input (clock) between ports	7.4			ns
t_{iihpp} Input (data) hold time after input (clock) between ports	7.4			ns

Table 8: Recommended sampling timing (not applicable in synchronous mode)

The minimum input low time (t_{il}), high time (t_{ih}), period (t_{ip}) must be selected according to required data integrity. If input-to-input setup and hold times (t_{iis}, t_{iih}) are not met, the data are not valid on the same sample as the clock signal changes.

5.2 Using the Application as a Portable Application

The logic analyzer application can be used as a portable application. The files required by the application are the original executable file (*sigmalogan.exe*), the FTChipID library (*ftchipid.dll*) and the plugins located in the plugins subdirectory. These files can be copied to any directory and the executable file may be renamed to any new name.

The application stores the setting into registry. By creating a blank ini file with the same base file name as the executable file, the application will start using this ini file. For example if executable file is logan.exe, create blank ini file called logan.ini. Application search for the file in the current directory and in the directory of the executable.

The contents of ini file are also checked against special code. To use the code, create a blank ini file or delete its contents and let the ini file contain only a single line with the code. The file should end with a new line.

Ini file special codes	
:NULL	Do not use, load or store any settings
:REG	Use registry (default setting)
:REG_HKCU	
:REG_HKCU/*path*	
:REG_HKLM	Use HKEY_LOCAL_MACHINE registry. May require grant access rights to the registry.
:FILE=*path*	Use specific *file*.

Table 9: Ini file special codes

[1] SIGMA2 Logic Analyzer default sampling period.

[2] If the input changes during this window, data are indeterminate.

6

Using SIGMA2 Logic Analyzer under Linux

The software for the SIGMA2 Logic Analyzer is capable of working under Wine. For USB access it uses libftd2xx.

Step 1: Install libftd2xx and libftchipid

Install 32-bit versions of libftd2xx and libftchipid by FTDI, even if you use 64-bit kernel. The application is a 32-bit binary and requires 32-bit libraries.

- Extract `libftd2xx.so.1.1.0` (in case of newer version replace 1.1.0 with the latest version) and `libftchipid` and copy the files `libftd2xx.1.1.0.so` and `libftchipid0.1.0` into the directory for 32-bit libraries (typically `/usr/lib/i386-linux-gnu/`).
- `ln -s libftd2xx.so.1.1.0 /usr/lib/i386-linux-gnu/libftd2xx.so.1` (it is typically sufficient to run ldconfig to achieve this)
- `ln -s libftd2xx.so.1.1.0 /usr/lib/i386-linux-gnu/libftd2xx.so.0` (must be made manually)
- `ln -s libftchipid.so.0.1.0 /usr/lib/i386-linux-gnu/libftchipid.so.0` (it is typically sufficient to run ldconfig to achieve this)
- The library searches for device files in `/dev/bus/usb`. Please ensure that `/dev/bus/usb` directory contains special files to access USB devices.
- Check that your SIGMA2 Logic Analyzer is recognized by the system (use command `lsusb`).
- Check your access rights to the corresponding files in `/dev/bus/usb` (command `ls -la /dev/bus/usb/`). Probably you will not have as a user the r+w access rights for these files.
- If you have not access rights and you are using udev:

Create a user group (or use in any of the suitable existing groups) which will have access to ASIX USB devices.
Add a new file with udev rules to the directory `/etc/udev/rules.d` or `/lib/udev/rules.d` (Depending on your distribution. Suitable name for this new file is 51-asix_tools.rules. Insert the following lines to this file (where mygroup is a name of the group you have chosen, suitable group is e.g. plugdev):

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="f1a0", MODE="0664", GROUP="mygroup" # PRESTO
SUBSYSTEM=="usb", ATTRS{idVendor}=="a600", ATTRS{idProduct}=="a000", MODE="0664", GROUP="mygroup" # SIGMA/SIGMA2
SUBSYSTEM=="usb", ATTRS{idVendor}=="a600", ATTRS{idProduct}=="a003", MODE="0664", GROUP="mygroup" # FORTE
SUBSYSTEM=="usb", ATTRS{idVendor}=="a600", ATTRS{idProduct}=="a004", MODE="0664", GROUP="mygroup" # OMEGA
```

VID and PID values can be determined using the listing of connected devices by the lsusb command.
The simplest option is to give the access rights to devices to all programs and all users (highly not recommended!). In this case, use `MODE="0666"` and do not use the GROUP parameter.

Step 2: Install wine

It is necessary to install 32-bit version of wine (for example `wine-1.4:i386`).

Step 3: Install lin_ftd2xx

Check environment variable *WINEDLLPATH*. It should point to directory where are 32-bit wine DLLs, typically `/`

usr/lib/i386-linux-gnu/wine. Install lin_ftd2xx by ASIX into this directory.

Installation of the Microsoft™ TrueType core fonts are recommended. These fonts may be obtained by installing msttcorefonts package from Ubuntu package repository.

Note:

Library libftd2xx requires also access rights during opening of the programmer or logic analyzer to all FTDI serial devices to check that this is not the device it wants to open.

7 OMEGA and SIGMA2 Comparison

Parameter	SIGMA2	OMEGA
On the market since	Since 2007[1]	Since 2012
PC interface	USB 2.0 Full Speed (12 Mbps) powered from USB, no external supply required	USB 2.0 High Speed (480 Mbps) powered from USB, no external supply required
Basic mode (with advanced trigger logic)	16 inputs / 50 Msps	16 inputs / 200 Msps
Accelerated mode (with simple triggering)	8 inputs / 100 Msps, 4 inputs / 200 Msps	8 inputs / 400 Msps
Synchronous clock mode	15 inputs / 49.975 MHz	15 inputs / 99.95 MHz
Real-Time operation mode	N/A	Available up to 2^{31} B-tree nodes
Daisy chain operation	N/A	2 analyzers (up to 32 inputs): ±5 ns 3 analyzers (up to 48 inputs): ±10 ns more: possible but without timing specification
Memory	SDRAM, 256 Mbit, 16-bit bus, ~66 MHz	SDRAM, 512 Mbit, 32-bit bus, ~133 MHz
Compression method	RLE	RLE + Huffman coding
Max. RLE run count	2^{16}	2^{15}
Sample memory size[2]	14.7×10^{6}	29.7×10^{6}
Typical number of samples[3]	2×10^{6} input signal changes	approximately 20-30×10^{6} input signal changes
Max. test length/ max. acquisition time[4]	128×10^{9} / 45 min.	862×10^{9} / 77 min.[5]
Worst conditions acquisition length	0.29 s	0.15 s
Worst case compressed data flow	915 Mbit/s	3.6 Gbit/s
External Trigger-In	LVTTL (max. 3.3 V)	
External Trigger-Out	LVCMOS (3.3 V) with 1 kOhm serial resistor or open collector with pull-up	LVCMOS (3.3 V)
Auxiliary Power Output	Trigger-In pin 3.3 V, max. 100 mA	Trigger-In pin 2.4 - 3.0 V, max. 100 mA

Table 10: OMEGA and SIGMA2 comparison

[1] Prior to 2011 as SIGMA.

[2] Maximal test length - each successive sample differs (data cannot be compressed).

[3] Tested with I2C, SPI or UART serial protocols.

[4] Test length / time with no signal changes on the inputs.

[5] In real-time mode the maximum test length is 65×10^{12} samples = 90 hours.

8 Specifications

Input Voltage Range				
	Min.	Typ.	Max.	
V_{IL} input low voltage			0.8	V
V_{IH} input high voltage	2.0			V
V_{IN} absolute rating, inputs 1..16	-0.3		5.5	V
V_{IN} absolute rating, trigger I/O	-0.3		3.6	V
Power Output				
V_{PO} power output on Trigger In		3.3		V
I_{PO} power output on Trigger In			100	mA
Input Pins Skew				
t_{sksp} pin-to-pin skew within single port		1		ns
t_{skbp} pin-to-pin skew between ports		4.8		ns
Recommended Sampling Timing[1]				
t_{cis}+t_{cih} Data valid window	2.6			ns
t_{iis} Input (data) setup time before input (clock) within one port	3.6			ns
t_{iih} Input (data) hold time after input (clock) within one post	3.6			ns
t_{iispp} Input (data) setup time before input (clock) between ports	7.4			ns
t_{iihpp} Input (data) hold time after input (clock) between ports	7.4			ns
Synchronous Clock Timing				
t_{setup} Data setup before clock		3.55	8.30	ns
t_{hold} Data hold after clock		-0.55	3.75	ns
Asynchronous Clock Timing				
t_{SMP} Sampling period		20		ns
t_{CLK} Input clock period	50			ns
t_{setup} Data setup before clock		2.5	7.3	ns
t_{hold} Data hold after clock		22.7	27.5	ns
$\Delta f/f_{typ}$ internal clock precision		±50		ppm
T_A ambient temperature[2]	0		50	°C

Table 11: Electrical specification

[1] Not applicable in synchronous timing
[2] Indoor use only

9

Document history

Document revision	Modifications made
2014-12-18	Initial release of a new version of manual.
2015-04-15	Updated info on parameters of -out and -export of sigmalogan.exe and omegacli.exe.
2016-12-24	Added info about utilites omegacli, omegartmcli, stf2bin, bin2stf, binconvert.
2017-02-10	Updated Linux info