ASIX

OMEGA

LOGIC ANALYZER

Reference Manual

ASIX s.r.o.
Na Popelce 38/17
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

1.1 Product Overview

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

OMEGA Logic Analyzer is the highest performance logic analyzer by ASIX s.r.o. from series of SIGMA/SIGMA2/ OMEGA Logic Analyzers.

The OMEGA Logic Analyzer is equipped with 512 Mb of memory (equivalent to 64 MB) and provides with up to 16 digital inputs and sampling rate of 200 Msps by a single device. More inputs and memory can be utilized using synchronization between multiple logic analyzers units. Built-in data compression allows for tracing of long running signals without exhausting logic analyzer memory. When using all 16 inputs per analyzer, guaranteed minimum capacity is over 29 million samples[1] at full speed. OMEGA uses High-Speed USB 2.0 (480 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 per unit
- Synchronization between multiple units
- Up to 400 Msps sampling rate (limited number of inputs)
- External clock up to 99.95 MHz
- 512 Mb of internal memory
- Real-Time mode utilizing the memory only as a FIFO buffer
- Advanced hardware scompression
- Flexible trigger options
- USB interface for data and power

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:

- OMEGA Logic Analyzer
- Target cables:
  - 20 individual pins (SIGMACAB)
  - one-to-one 20 pins (SIGCAB20)
  - one-to-one 10 pins (SIGCAB10)
- USB cable (A-B)
- Synchronization header
- Synchronization cable
- 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: OMEGA 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 OMEGA Logic Analyzer.

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

[1] OMEGA will use approximately 18.1 bits of memory per 16 input sample.

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

2.2 Target connection

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

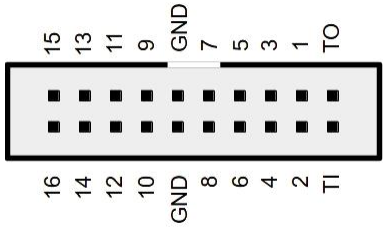

Fig. 3: Target connector

Always connect the ground between the application and the analyzer and then connect desired input pins. The OMEGA 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.

2.3 Acquire the data

Start the application ASIX SIGMA & OMEGA Logic Analyzers from start menu and launch the data acquisition by pressing **Enter**. The OMEGA 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.
- OMEGA Logic Analyzer supports many auxiliary functions. You can setup them with **U** key (U for *utilities*).

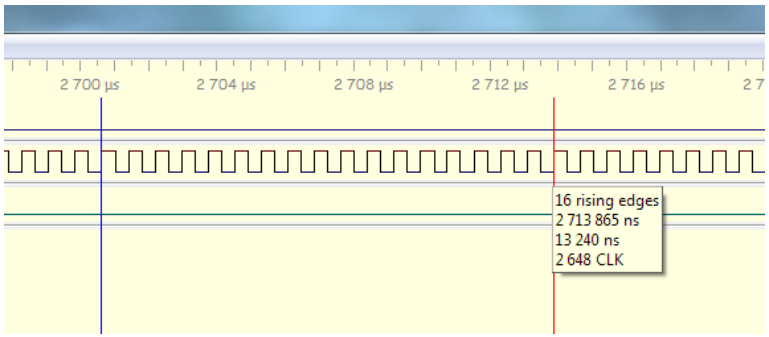

Fig. 4: Counting edges in an acquisition

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:** OMEGA 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:** OMEGA is ready to operate
- **yellow:** OMEGA is acquiring data

TRIGGER STATUS (red/yellow LED)

- **off:** Trigger logic is inactive - no trigger condition has been detected
- **red:** OMEGA 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 [1] the acquisition is stopped, idle mode is launched and the data transfer from logic analyzer memory to PC begins.

3.2 Target Connection

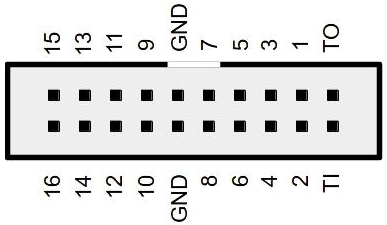

Fig. 5: Target connector

The OMEGA 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*[2] is available on *Trigger In*. The function of Synchronization (Daisy-Chain connection) is available on the *Trigger In*/*Trigger Out* pair.

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. OMEGA Logic Analyzer do not isolate ground between the PC and the application [3].

	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] When using OMEGA Real-Time mode where is unconstrained number of triggers, end of acquisition can be performed by long pressing of the GO button.

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

[3] When using a USB optoisolator designed for USB Full-Speed (12 Mbps) you will benefit from unique OMEGA 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

OMEGA 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***.

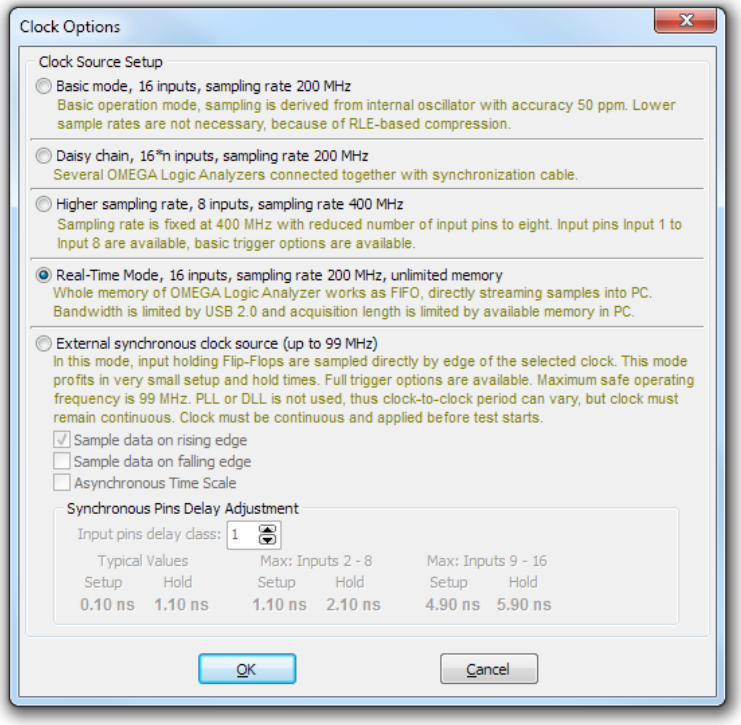

Fig. 6: Modes of operation

Available modes of operation:

Basic Mode

16 inputs, sampling rate 200 Msps.
Basic operation mode, sampling is derived from internal oscillator.

Daisy Chain Mode

16×n inputs, sampling rate 200 Msps.
Several OMEGA Logic Analyzers connected together with a synchronization cable.

Higher Sampling Rate Mode

8 inputs, sampling rate 400 Msps.
Number of input pins is reduced to eight with benefit of twice sampling rate. Inputs are limited to single port due to higher demands on the delay skew between inputs.

Real-Time Mode

16 inputs, sampling rate 200 Msps.
The whole internal memory works as FIFO buffer and data are streamed in real-time to the PC. This mode is very memory demanding, therefore lot of free RAM and free space on hard drive is required.

Synchronous Clock Mode

15 inputs, external clock.
Input1 is utilized as external clock input. Rising or falling sampling edge or both (DDR) can be chosen. Clock speed should be within the range of ~100 kHz to 99.95 MHz. 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 last samples will not be contained in the captured data. Using this mode for measuring synchronous bus (e.g. processor bus) can be advantageous. The synchronous clock can be used with or without asynchronous time information[1].

Synchronous clock Mode Without Asynchronous Time Scale

The OMEGA Logic Analyzer tries optionally to measure the clock speed with auxiliary low resolution frequency counter and uses this information for time scale. Because time information is not included in the acquisition data, the memory usage is lower compared to the mode with Asynchronous Time Scale enabled.

Synchronous clock Mode With Asynchronous Time Scale

With external clock frequencies <40 MHz, The OMEGA Logic Analyzer can together with the data also save the time with resolution 10 ns. Higher frequencies up to the maximum possible external clock frequency are allowed, but the accuracy is degraded due to higher utilization of the clock synchronization circuitry. It is not possible to use both DDR and Asynchronous Time Scale at the same time.

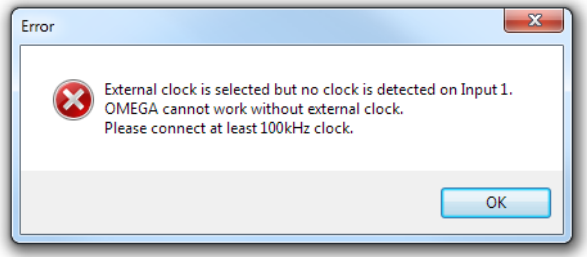

Fig. 7: Clock not present on Input 1 during OMEGA Synchronous Clock Mode

The clock on Input 1 must be continuous and present before the acquisition starts.

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

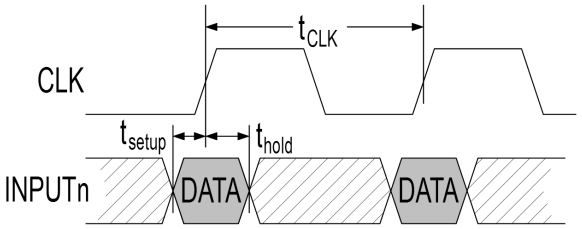

Fig. 8: Synchronous clock timing

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

Delay Class	Typical		Maximum Inputs 2-8		Maximum Inputs 9-16		
	Setup t_{setup}	Hold t_{hold}	Setup t_{setup}	Hold t_{hold}	Setup t_{setup}	Hold t_{hold}	
1	0.10	1.10	1.10	2.10	4.90	5.90	ns
2	-0.15	1.45	0.85	2.45	4.65	6.25	ns
3	-0.40	1.10	1.10	2.10	4.90	5.90	ns
4	-0.90	2.45	0.10	3.45	3.90	7.25	ns
5	-1.10	2.70	-0.10	3.70	3.70	7.50	ns
6	-1.40	3.10	-0.40	4.10	3.40	7.90	ns
7	-1.80	3.55	-0.80	4.55	3.00	8.35	ns
8	-1.95	3.75	-0.95	4.75	2.85	8.55	ns

Table 3: Synchronous Clock Timing

Delay Class can be selected in a dialog ***Settings → Clock Source***. All inputs must be set up for the same *Delay Class*.

4.2 Input pins

The term **input pin** refers to physical input of OMEGA 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, OMEGA 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.

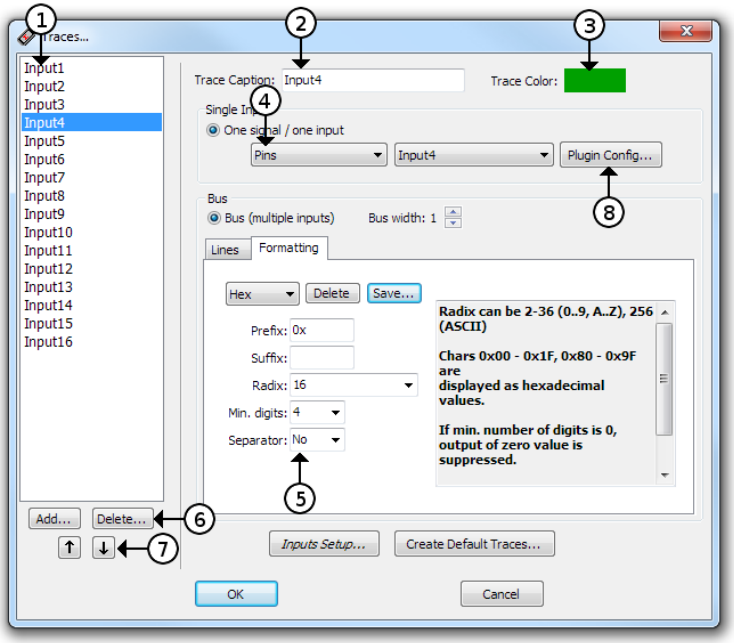

Fig. 9: Traces Dialog

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

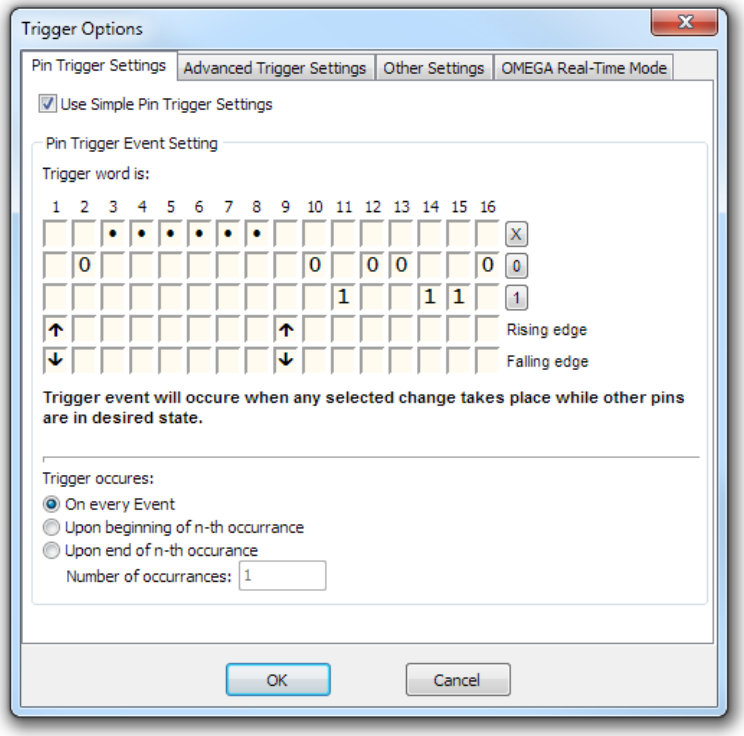

Fig. 10: Basic Trigger Settings

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 10 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.

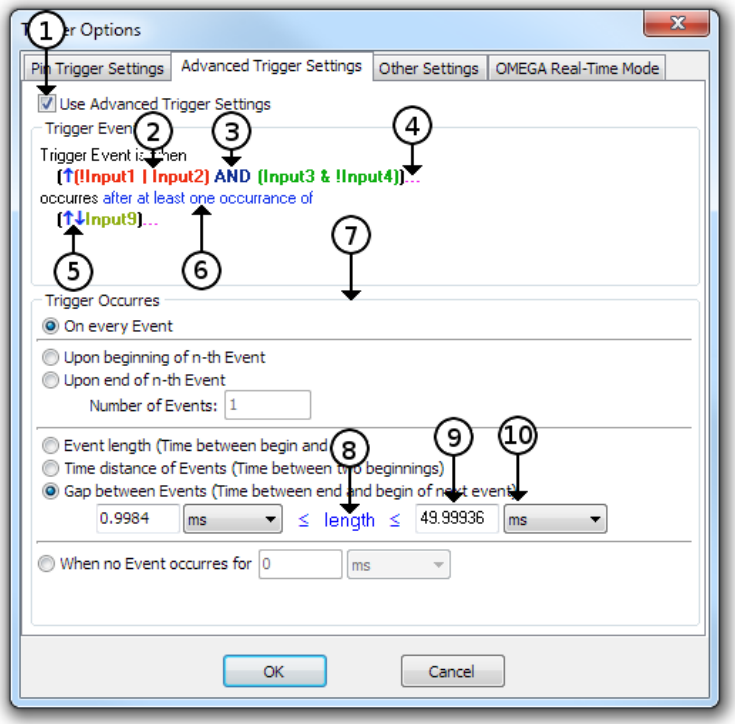

Fig. 11: Advanced Trigger Settings

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*.

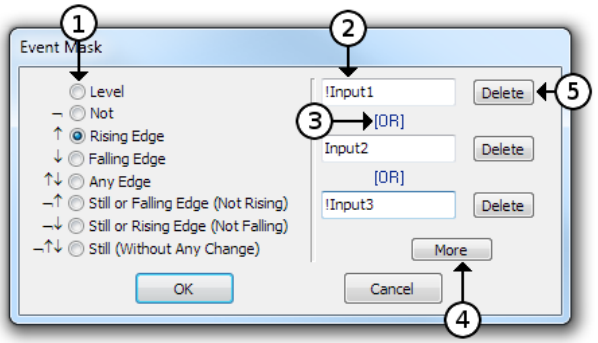

Fig. 12: Advanced Trigger Masks

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 OMEGA 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 OMEGA 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 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 *Trigger In* and *Trigger Out* pins are shared with *Daisy Chain* functionality on the OMEGA Logic Analyzer.

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	2.0	2.4	3.3	V
I_{PO} power output on Trigger In			100	mA

Table 4: 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.

4.4.6 Amount of triggers

When using *OMEGA Real-Time Mode*, the trigger is being detected and stored during whole acquisition, not only for the first time to trigger the acquisition.

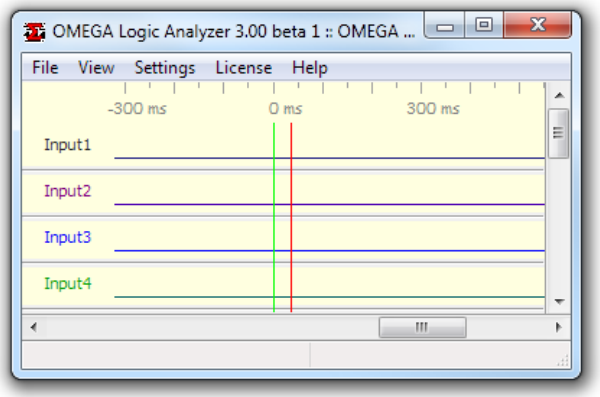

Fig. 13: Time scale around trigger

The time scale zero time can be set to:

- The beginning of the acquisition.
- The first trigger.
- The last trigger.

The viewer can be set to display:

- Only the first trigger.
- Only the last trigger.
- Every trigger.

Note:	Setting the trigger as an event which is very often (e.g. rise edge of a communication clock) is **highly discouraged** in the *real-time mode* because the trigger is stored in the FIFO and consumes significant amount of the buffer. The performance of the software is also reduced when displaying lot of triggers.

Trigger Filter

The trigger filter is a filter which reduces the number of triggers to one per millisecond, but allowing bursts of up to 256 triggers. The filter should be always turned on during normal operation.

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 5: Mouse and Keyboard action for navigation and analysis

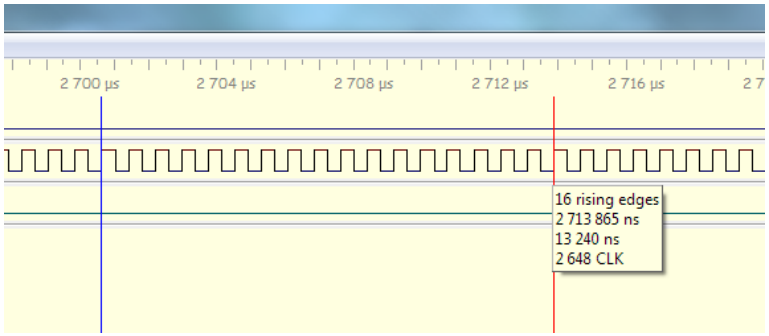

Fig. 14: Counting edges in an acquisition

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.

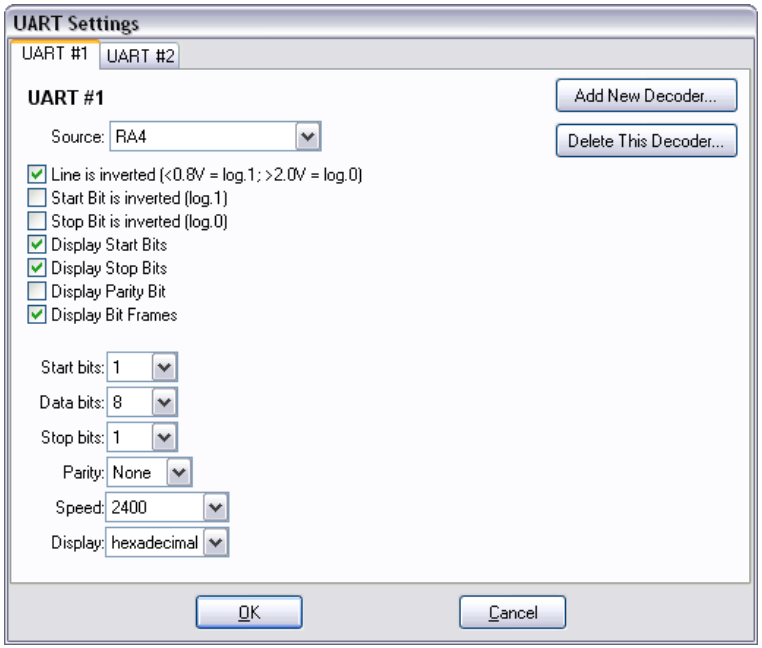

Fig. 15: UART Decoder

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 OMEGA 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.

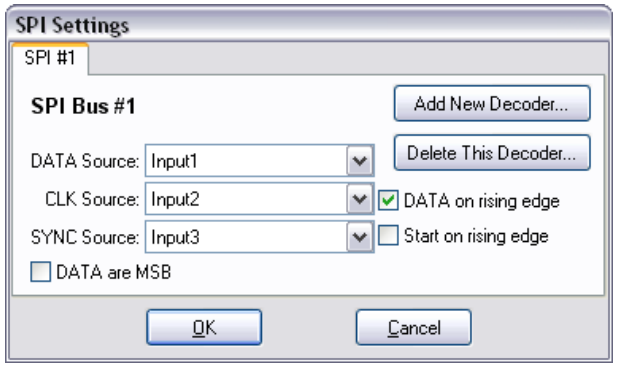

Fig. 16: SPI Plugin settings

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.

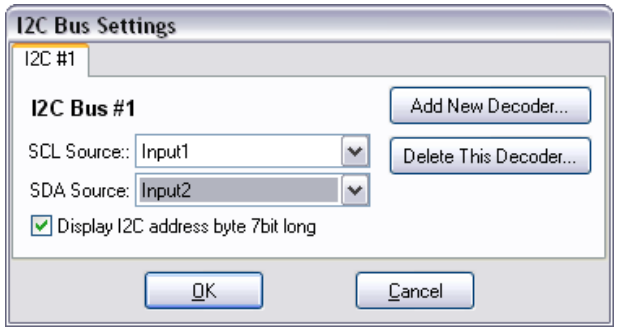

Fig. 17: I2C Plugin settings

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 1-Wire Bus decoder

This decoder decodes captured 1-Wire signal. It shows reset pulses (R), presence pulses (P) and data in hexadecimal format.

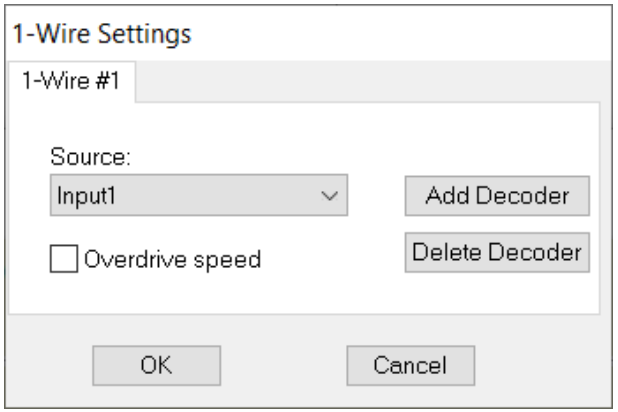

Fig. 18: 1-Wire plugin settings

Several options can be set up:

Source
: Selects an input pin containing the signal to be decoded.

Overdrive speed
: By setting this option the overdrive speed is selected, when not set it is supposed that there is normal communication speed on the 1-Wire bus.

4.5.6 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. The communication is decoded automatically after the data have 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 is 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:	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.

What to measure

With USB decoder USB 1.5 Mbps (Low-Speed) and 12 Mbps (Full-Speed) data communication captured by the OMEGA 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 signals (DATA+, DATA-) must be connected to the OMEGA Logic Analyzer. OMEGA 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-, 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 OMEGA 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 OMEGA 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 a USB hub and the shorter USB cable, the better. Likewise it is important to shorten the way between USBprobe and OMEGA.

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

The USB data signals (DATA+, DATA-) can be connected to any two inputs and the other inputs can be 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) they have to be decoded. It can take tens of seconds depending on amount of the captured data. Decoding is done automatically, immediately when the data have been captured, if it is enabled in the USB analyzer settings dialog. Or it can be started 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 a single address 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 for 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.

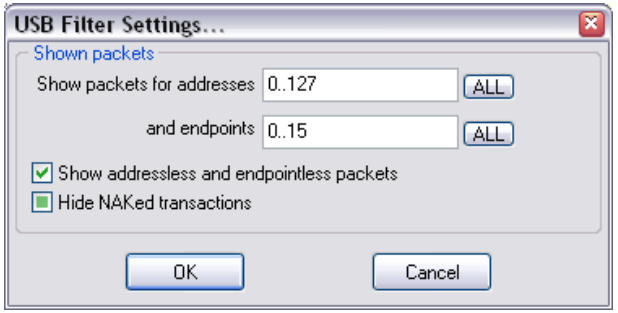

Fig. 19: USB Filter settings

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

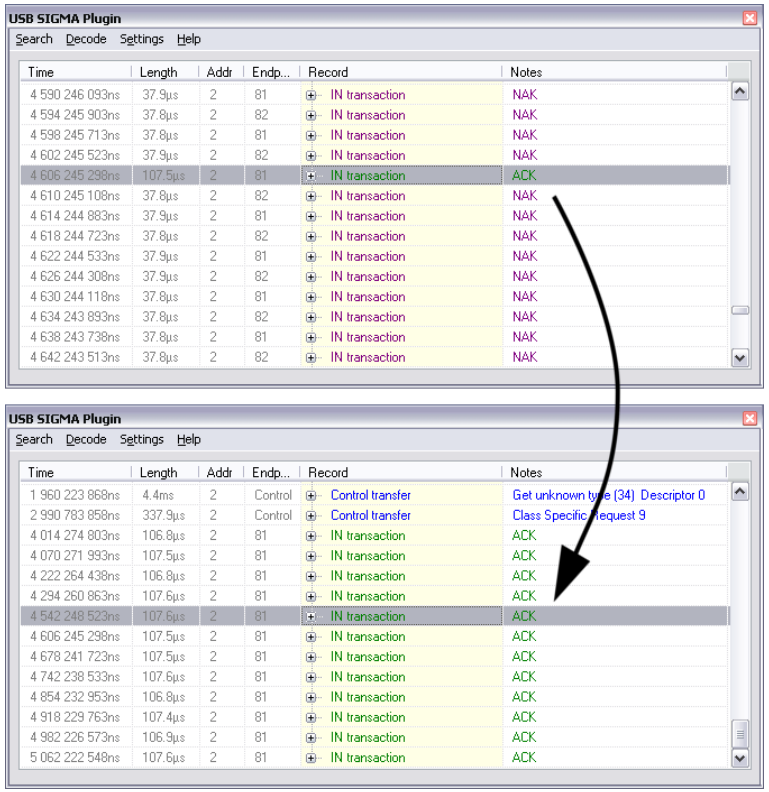

Fig. 20: Window with hidden transactions which are ended with NAK

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.

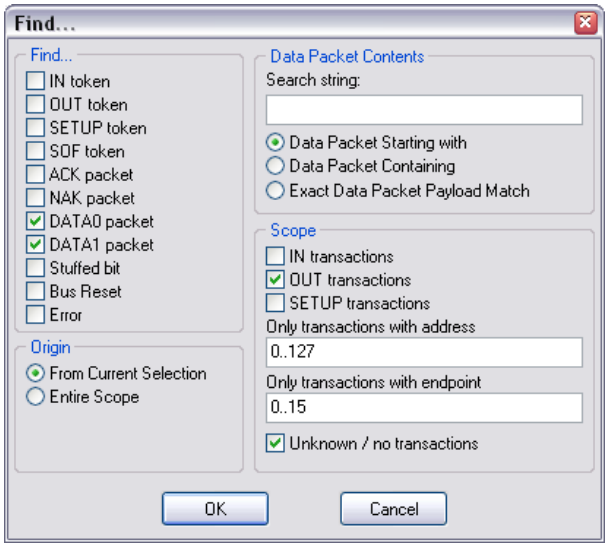

Fig. 21: Searching Window

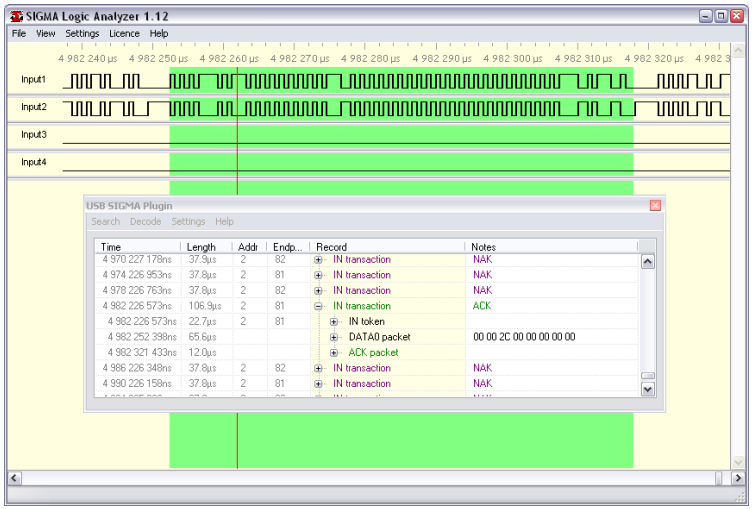

Fig. 22: DATA0 Packet Highlighted

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 behavior leads 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 OMEGA Logic Analyzer have some useful auxiliary functions. This functions are controlled all at once using a gateway dialog ***Settings → Utilities Setup...***.

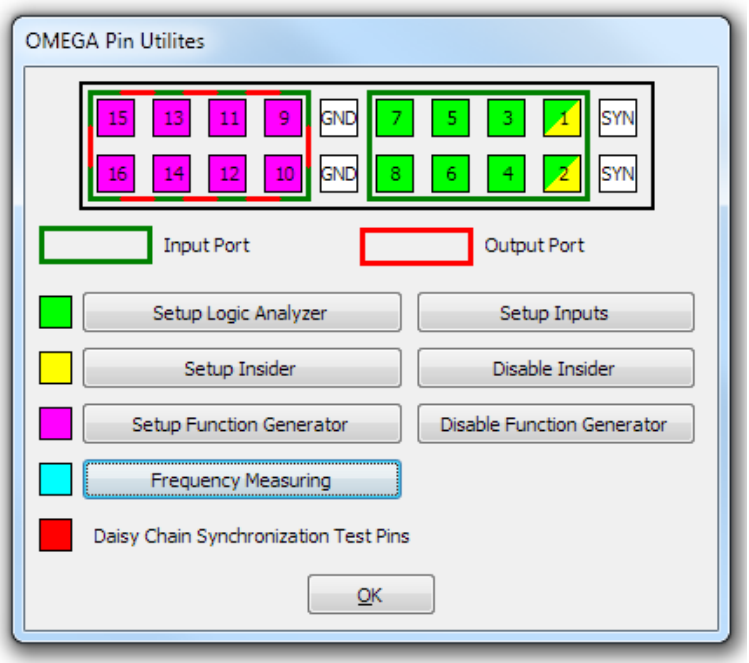

Fig. 23: Dialog Utilities Setup

Setup Logic Analyzer
This opens *Clock Source* dialog and allows setup operation mode and number of input pins.

Setup Inputs
This opens *Inputs Setup* dialog where unused inputs can be selected.

Setup Insider
This opens *Insider Setup Wizard*. See chapter Insider.

Setup Function Generator
This opens *Function Generator Setup*. See chapter Function Generator.

Frequency Measuring

This opens *Frequency Measuring* and allows measuring of duty cycle and frequency on one pin.

4.6.1 Insider

OMEGA 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 https://www.chiark.greenend.org.uk/~sgtatham/putty/.

OMEGA Insider can be set up using OMEGA Insider Wizard. The wizard can be launched from *Utilities dialog* (can be opened using ***Settings → Utilities Setup...*** or by pressing the **U** key).

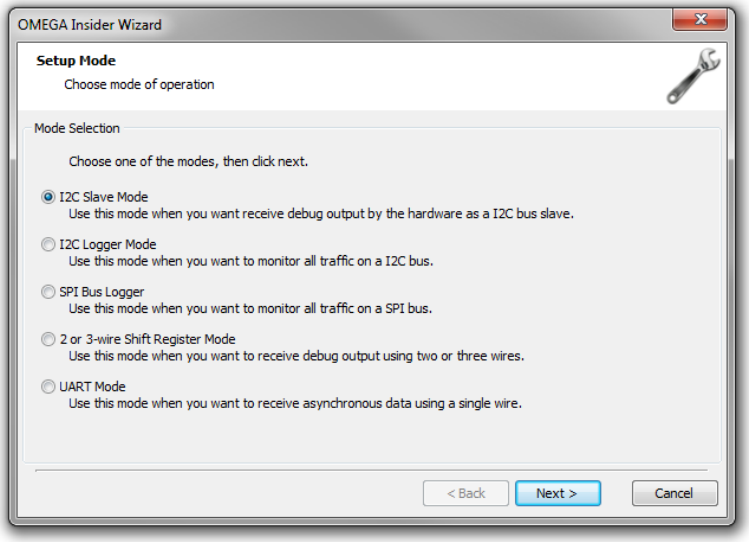

Fig. 24: OMEGA Insider Wizard

The wizard can work in one of five modes.

Mode	Bus	Type	Purpose	Example of Required Settings	Pin Requirements
I^2C Bus Slave	I^2C	Debug texts	Log texts written to OMEGA as I^2C bus slave device	OMEGA I^2C Bus Slave Address I^2C Bus Speed (100 kHz / 400 kHz)	Input 1 (SDA), Input 2 (SCL), Input 9, Inputs 10-16 unusable
I^2C Bus Logger	I^2C	Bus logger	Log all bus activity including addresses, data, ACKs and NAKs	I^2C Bus Speed (100 kHz / 400 kHz)	Any two inputs
SPI Bus Logger	SPI	Bus logger	Log all bus activity using single -CS wire.	Single (DATA) or dual data (MISO, MOSI) lines, CS polarity, SPI Bus mode (0/1/2/3)	Any three or four inputs.
Universal Synchronous Receiver	Synchronous	Debug texts	Log texts sent using DATA / CLK wires.	Optional byte synchronization using third wire or timer.	Any two or three inputs.
Universal Asynchronous Receiver	UART	Debug texts	Log texts sent using TXD wire.	Baud rate, start bit inversion.	Any single input.

Table 6: OMEGA Insider Modes

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

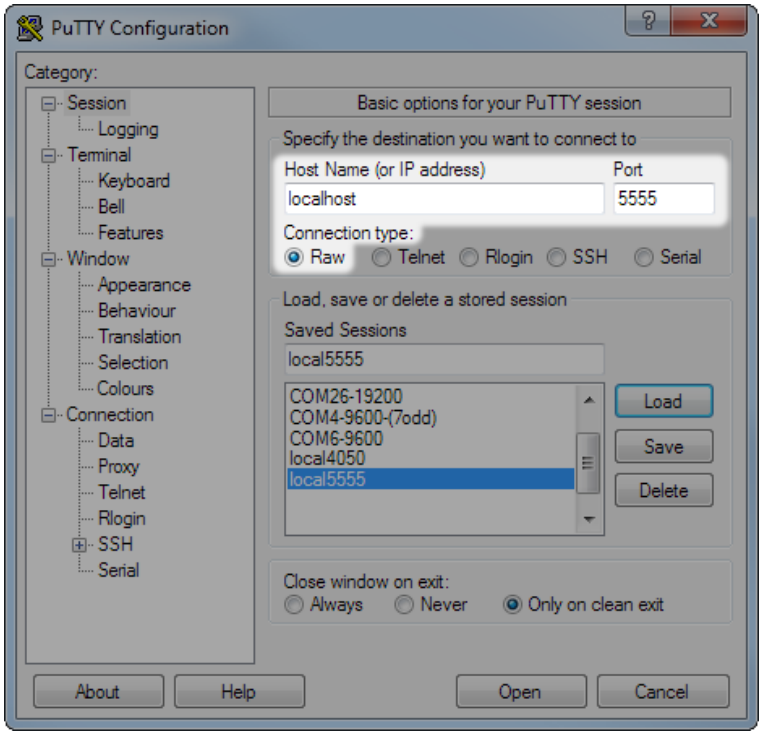

Fig. 25: Opening Insider Connection in PuTTY

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.

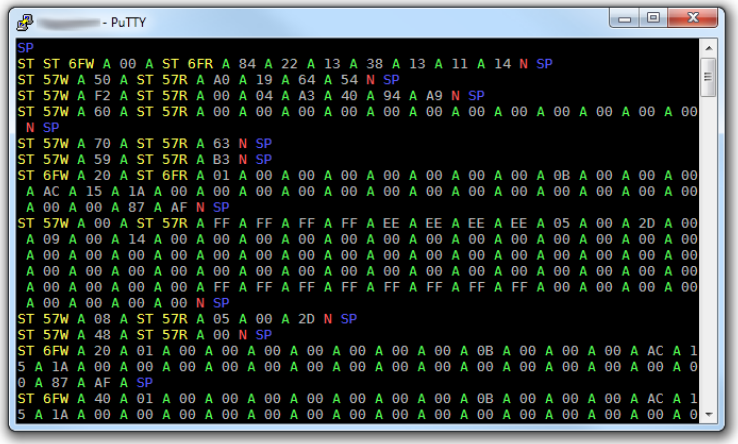

Fig. 26: Example I2C Log in PuTTY window

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

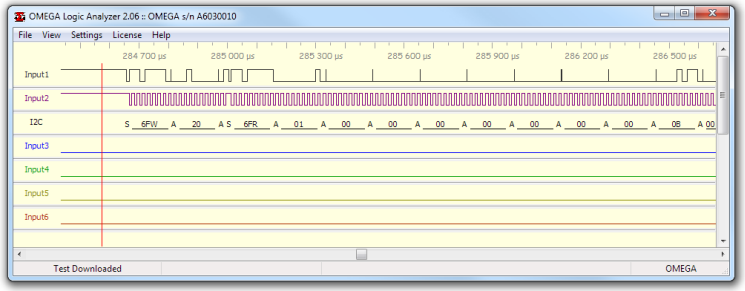

Fig. 27: Example I2C Bus activity

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.

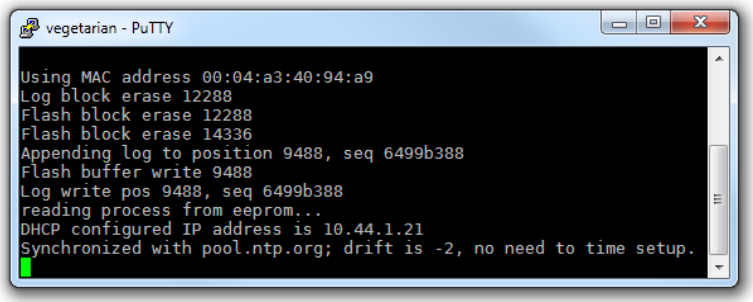

Fig. 28: Example of I2C debug in PuTTY window

Note:	The Insider functions are available only in Basic mode during idle and acquisition phase. The Insider is not available during acquisition download.

4.6.2 Function Generator

The OMEGA Logic Analyzer can change the direction of a single port (either Inputs 1-8 or Inputs 9-16) and the port become 3.3V CMOS output. The Logic Analyzer will then control that eight output lines according to instructions in a file. The instructions can be followed either once or in a loop. The function generator output is controlled by controls in acquisition progress window during acquisition. The output can by optionally postponed to a detection of the trigger used by the acquisition.

The function generator can be set up from Utilities dialog (accessible by opening ***Settings → Utilities Setup...*** or by pressing **U** key).

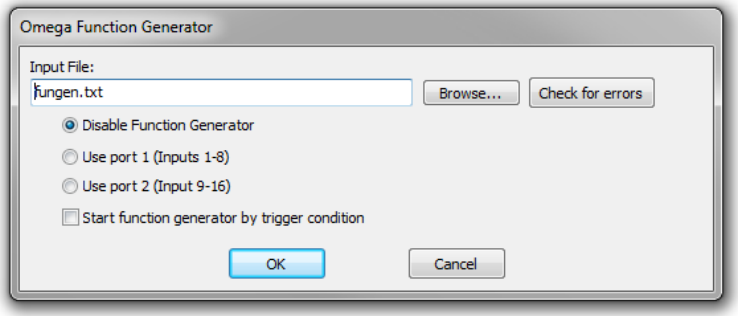

Fig. 29: OMEGA Function Generator Setup

Explaining example of a file containing function generator instructions.

```
0us     00
; this line is a comment
1us     T
2us     loop
```

The first column contains time from beginning of the file. It can be entered in nanoseconds (***ns***), microseconds (***us***) or milliseconds (***ms***). Please note that in the case of microseconds, the character **is not** the Greek letter mu **µ**, but the Latin letter **u**. The time on the first line **must be** zero. There must be no whitespace between number and the unit identifier (ns/us/ms).

The second column contains the output of a single port (eight output pins) to be used. The value is an eight bit value.

- Two characters are interpreted as a hexadecimal value (for example ***A6*** as 1001 0110).
- Three characters are interpreted as an octal value (for example ***266*** as 10 010 110).
- Eight characters are interpreted as a binary value (for example ***10100110***).
- Keyword ***T*** is interpreted as a third state (for whole port).

- Keyword ***END*** for end of file without looping.
- Keyword ***LOOP*** for end of file and continue looping.

First and second column are delimited by any number of whitespace characters.

The file is case insensitive.

4.7 Frequency Measuring

OMEGA Frequency Measuring is a tool for measuring a frequency on an input pin. The OMEGA Frequency Measuring can be enabled using menu ***View → Frequency Measuring***.

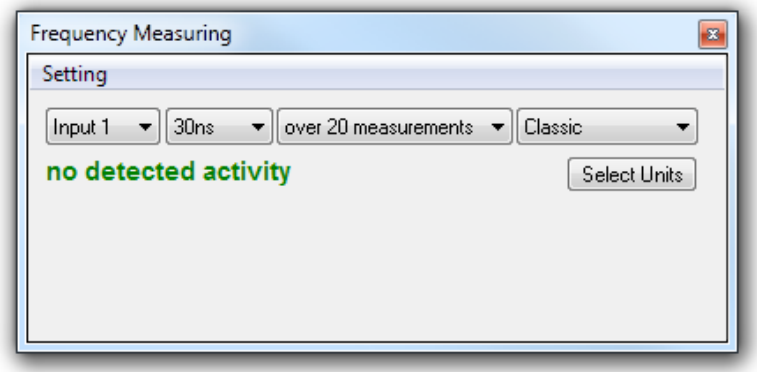

Fig. 30: OMEGA Frequency Measuring

4.8 Pin View

OMEGA Pin View is a logic probe tool for inspecting of the correct connection of the OMEGA 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 file name given in by parameter ***-export*** has extension *.stf, the saved file is in STF format.

4.11 Command Line Interface omegacli.exe

The command line utility *omegacli.exe* is a command line interface for simple interfacing of The OMEGA Logic Analyzer. It can be found in directory where SIGMA2/OMEGA software was installed. The utility does not require any other files, therefore it can be copied to any directory and treated as a portable application.

Once *omegacli.exe* is started, it performs one acquisition with basic settings and prints the results to the standard output. The standard output can be redirected in many different shells to a file using "***>***" character. The acquisition progress is being printed to standard error output. To disable standard error output in shell, use "***2>/dev/null***" redirection command.

List of parameters:

- ***-h***, ***-help*** - prints short description and exits
- ***-version*** - prints program version and exits
- ***-post xxx*** - selects post trigger time or test length (if there is no trigger)
  - must be one of these variants:
    - ***-post num*** (range of 1 to 254) - a number of 256kB blocks
    - ***-post nn%*** (range of 1% to 99%) - per cent memory amount, rounded to 256kB blocks
    - ***-post time*** (units "s" or "ms") - this method is very inaccurate, resolution is hundreds of milliseconds
  - example:
    - ***-post 1s***
    - ***-post 50%***
- {-serial sernum}{code] - use OMEGA with this serial number (must be in format A6031234 or 031234)
  - example:
    - ***-serial 031234***
- ***-trg xxx*** - trigger condition; must be entered as 16 characters, each pin can be 0,1,R,F or X (don't care), only one pin can be selected as R/F
- ***-trg none*** - do not use trigger condition, only button
- ***-trg any*** - do not use trigger at all
  - example:
    - ***-trg 000000000000001R***
- ***-bin*** - acquisition data are in binary format Omega.Data) see SIGMAP01 - Reading STF File. This format is suitable for other ulitilies, like bin2stf and binconvert.
- ***-out file.stf*** - write acquisition to a file in stf format, instead of stdout

The output format of *omegacli.exe* on the standard output:

- The first column is a timestamp (5ns units) relative to the position of the trigger.
- The second column are pin states, always represented as a 16 digit binary number.
- The columns are delimited by a tabulator character.

Both text output format and binary output format are the same for both omegacli.exe and omegartmcli.exe utilities. The binary output format us useful for other utilities, like bin2stf.exe and binconvert.exe.

Note: The *omegacli.exe* command line interface is currently a beta version.

4.12 Data streaming using Command Line Interface omegartmcli.exe

The command line utility *omegartmcli.exe* is a command line interface for simple interfacing of The OMEGA Logic Analyzer. It can be found in directory where SIGMA2/ OMEGA software was installed. The utility does not require any other files, therefore it can be copied to any directory and treated as a portable application.

The utility *omegartmcli.exe* starts sending acquired data by Logic Analyzer to the standard output. The memory in the logic analyzer is turned into a FIFO. The acquisition will last as long as data from standard output are taken. The acquisition can be terminated by terminating of the standard input of the utility or by pressing Ctrl+C on console. The standard output can be redirected in many different shells to a file using "`>`" character or into another utility using "`|`".

List of parameters:

- ***`-h`***, ***`-help`*** - prints short description and exits
- ***`-version`*** - prints program version and exits
- {-serial sernum}{code] - use OMEGA with this serial number (must be in format A6031234 or 031234)
  - example:
    - ***`-serial 031234`***
- ***`-bin`*** - acquisition data are in binary format Omega.Data) see SIGMAP01 - Reading STF File. This format is suitable for other ulitilies, like bin2stf and binconvert.

The output format of *omegartmcli.exe* on the standard output:

- The first column is a timestamp (5ns units) relative to the position of the trigger.
- The second column are pin states, always represented as a 16 digit binary number.
- The columns are delimited by a tabulator character.

Both text output format and binary output format are the same for both omegacli.exe and omegartmcli.exe utilities. The binary output format us useful for other utilities, like bin2stf.exe and binconvert.exe.

Note: The *omegacli.exe* command line interface is currently a beta version.

4.13 Conversion to and from binary data stream using bin2stf.exe a stf2bin.exe

Utility *bin2stf.exe* can save binary data stream into STF file. This file can be further opened in logic analyzer application, starting at version 3.04. A STF file saved this way does not include any information about used logic analyzer or its settings. The utility bin2stf.exe can save data only into STF file, output to standard output is not possible.

Utility *stf2bin.exe* can convert STF file into binary data

stream. Input can be STF file acquired and saved using OMEGA Logic Analyzer in Real-Time mode. Data acquired in other modes cannot be converted into data stream. Output binary stream can be written to standard output.

Poznámka: The utilities *bin2stf.exe* and *stf2bin.exe* are currently beta version.

4.14 Working with binary data stream using binconvert.exe

Utility *binconvert.exe* can take binary data stream, filter, decode and convert its data into further processable CSV format. The utility is designed to be fast and efficient. There are several routines, which can manipulate or tag the data in the stream and several other routines, which can print information from these tags. The sequence of the parameters defining these routines matter.

Full parameters description can be obtained by calling ***binconvert*** without parameters.

Example usage:

```
binconvert -qual1 10 1000 -uart 10 115200,8,None -uartline
```

The result will look like:

```
14831687 10 $GNGSA,A,...

15876496 10 $GNGSA,A,...
```

Columns are separated by tab. First column contain absolute time in 5ns units. Second column contains number of input (numbered from 1) and third column contain the received UART line.

Example usage:

```
stf2bin in.stf | binconvert -qual1 10 1000 -bin | bin2stf - out.stf
```

This will filter Input 10 from in.stf and save the result to out.stf.

Other decoders may be available. Please contact support@asix.net

4.15 Tips on working with binary data stream

Binary data stream can be large amount of data. It is useful to manipulate them using other utilities which can work on binary streams, like gzip, zcat, lzop. The utility omegartmcli.exe can generate data at such very high rates, that storing data directly to harddisk drive without compression can be limiting.

The command ***omegartmcli -bin | lzop -o data.lzo*** will compress and write to disk (file data.lzo) data compressed by LZO.

The command ***lzop -dc data.lzo | gzip > data.gz*** will re-compress the data from LZO to GZIP.

The command ***zcat data.gz | binconvert ... | gzip > output.txt.gz*** will feed bincovert utility with data directly decompressed from a file. The output from binconvert utility is also directly compressed and written to a file. The command ***zcat*** is equivalent to ***gzip -dc***.

The GZIP compression is quite slow compared to LZO, but compression rate is better. Decompression is faster than compression. Each command separated by pipe **|** occupies one core of your CPU, therefore if you have enough CPU cores, whole chain will be as fast as the slowest command in the chain.

4.16 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] *Asynchronous mode* which is available with SIGMA/ SIGMA2 analyzers has been replaced with *Synchronous mode with Asynchronous Time Scale* option. This mode benefits from better accuracy and higher achievable clock frequency than the original *Asynchronous mode*.

[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 OMEGA Logic Analyzer samples at a sampling rate, for example 200 Msps. It means the inputs are sampled every 5 ns.

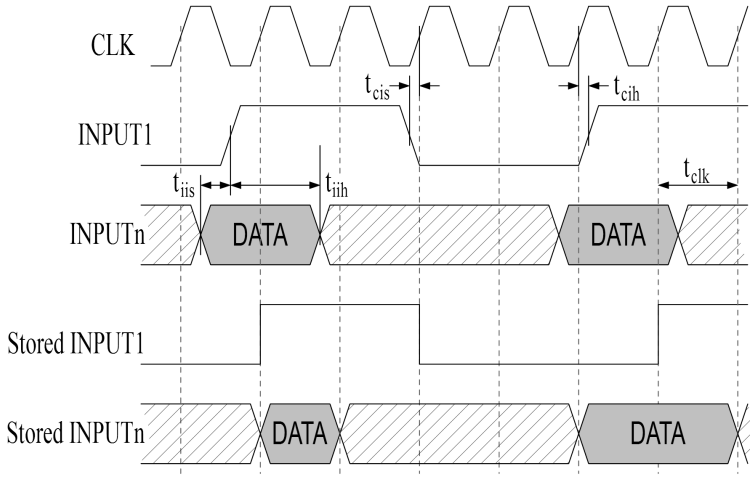

Fig. 31: Sampling of OMEGA Inputs

	Min.	Typ.	Max.	
t_{clk} Sampling clock period[1]		5		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 Powering the Analyzer

The OMEGA Logic Analyzer is powered directly from a USB bus. The USB bus should provide voltage V_{BUS} = 4.75 to 5.25 V on high-powered ports at maximum current I_{CCPRT} = 500 mA. The OMEGA Logic Analyzer meets and significantly exceeds these requirements, therefore it can be powered in bad power conditions, like an external battery power source in field. The sequence of applying of the USB power (*GND*, *VCC*) and the USB data (*GND*, *DATA +*, *DATA-*) is arbitrary, but always remember to connect *GND* first (USB connector have longer contact on *GND* pin).

Pin number	Function	Cable color	Required for
1	VCC	Red	USB Power
2	DATA-	White	USB Data
3	DATA+	Green	USB Data
4	GND	Black	USB Data + USB Power

Table 9: USB Cable Colors

	Min.	Typ.	Max.	
V_{BUS} Supply voltage at device connector	4.1	5.0	5.5	V
V_{BUSMIN} Minimum voltage for correct operation		3.9	4.1	V
I_{CCSLP} USB Data not connected or PC in sleep		300	500[3]	µA
I_{CCDIS} Device disabled in control panel		50	100	mA
I_{CCCNT} Device connected[4]		100	300	mA
I_{CCRDY} Device ready for measuring		200	300	mA
I_{CCMEA} Device measuring[5]		230	300[6]	mA

Table 10: Power Requirements

5.3 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 11: Ini file special codes

[1] OMEGA Logic Analyzer default sampling period.
[2] If the input changes during this window, data are indeterminate.
[3] For V_{BUS} lower than 5.0 V
[4] Software not yet started.
[5] Slow inputs, no outputs, no power on Trigger In
[6] Maximum current claimed to USB bus. If power on Trigger In and digital outputs are drained significantly, this power may be exceeded.

6 Synchronization

Two or more OMEGA Logic Analyzers can be connected together with included synchronization header and cable. Each OMEGA Logic Analyzer in synchronization setup can measure 16 inputs with sampling rate 200 Msps.

6.1 Logic Analyzer Connection to PC when Using Synchronization

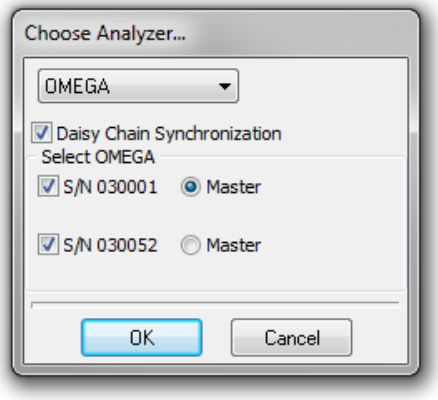

Fig. 32: Selection of Daisy Chained Analyzers

Synchronization mode among two or more OMEGA analyzers can be enabled in ***Settings → Select Analyzer*** menu by checking the option ***Daisy Chain Synchronization***. From this point it is possible to choose two or more OMEGA Logic Analyzers which will measure together. Each OMEGA Logic Analyzer must be connected to the same PC (using or not using an USB hub). Because of higher power consumption of the OMEGA Logic Analyzer, it is not recommended to power several Logic Analyzers from a bus-powered USB hub[1].

6.2 Synchronization Interconnection

6.2.1 Using Synchronization Cable

It is necessary to interconnect analyzers with a *Synchronization Cable* through the use of the *Synchronization Header* included with every OMEGA Logic Analyzer.

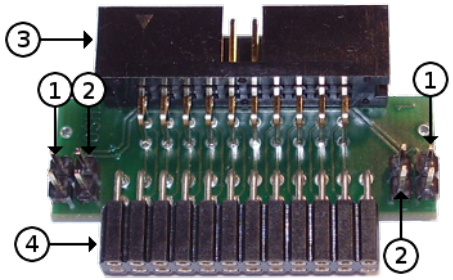

Fig. 33: Using the synchronization header

The synchronization header

1	Synchronization cable connection
2	Termination jumper
3	Target interface
4	OMEGA interface

The type of the interconnection is a bus with terminators. The synchronization bus is available on both sides of the *Synchronization header*. The bus must be terminated on both ends by means of the application of the *Terminator Jumper* (**2**). There must be either a synchronization cable or a jumper on each side of the synchronization header, but both the synchronization cable and the jumper must not be on the same side simultaneously.

6.2.2 Interconnection of two Logic Analyzers

When using two OMEGA Logic Analyzers in daisy-chain operation, number of used inputs can be up to 32 with sampling period 200 Msps. The synchronization cable can be attached in any way, the analyzers will automatically detect cable cross and correct it. One logic analyzer is declared as *Master* and the trigger condition can be made only of its inputs. The *Master* analyzer controls the *Pre-Trigger* and *Post-Trigger Times* (for more about *Post-Trigger Times* see chapter Trigger Position within Acquisition).

6.2.3 Interconnection of three Logic Analyzers

When using three OMEGA Logic Analyzers in daisy-chain operation, number of used inputs can be up to 48 with sampling period 200 Msps. The optimal timing (typical accuracy ±2 ns) is achieved when the *Master* logic analyzer is in the middle of the daisy chain and the *Slave* logic analyzers are on the both ends.

Fig. 34: Interconnection of three Logic Analyzers

Out of four (2^2) possible cable cross possibilities, only two are valid. The Hamming distance of any invalid connection to a valid one is always exactly one crossing, therefore if the synchronization cables are connected in an invalid combination, the software will call to cross one end of any cable.

Slave ← Master	Master → Slave	Validity
Straight	Straight	Valid
Straight	Crossed	Invalid
Crossed	Straight	Invalid
Crossed	Crossed	Valid

Table 12: Synchronization cable crossing

6.2.4 Interconnection of more than three Logic Analyzers

When using more than three OMEGA Logic Analyzers in daisy-chain operation, number of used inputs can be up to 16×n with sampling period 200 Msps. The optimal

timing is achieved when the *Master* logic analyzer is in the middle of the daisy chain. The further a *Slave* analyzer is from the *Master* analyzer, the greater is the sampling time misalignment. All Logic Analyzers should be connected with Synchronization cable without crossing. If any Slave Logic Analyzer receive crossed synchronization signal, the software indicate it (analyzers are identified by a serial number).

The maximum sampling misalignment is not guaranteed when using synchronization of more than three logic analyzers.

Note:	Please note that when using more than five logic analyzers in the daisy chain, it is probable the misalignment is higher than the sampling rate (5 ns).

6.3 Synchronization Accuracy

Every but the first logic analyzer tunes its oscillator frequency and phase to oscillator of the first one. The correct phase is important for accurate sampling of all logic analyzers at the same time (note 1 ns ≈ 30 cm at the speed of light). The length of the synchronization cable is tuned to compensate the delays of the internal circuits of the OMEGA Logic Analyzer. The compensation is correct for interconnection of two or three logic analyzers when the *Master* is placed in-between.

	Typ.	Max.	ns
$t_{\Delta 2}$ Sampling misalignment between 2 analyzers	±2	±5	ns
$t_{\Delta 3}$ Sampling misalignment between 3 analyzers	±2	±10	ns
$t_{\Delta >3}$ Sampling misalignment between *n* analyzers	±(*n*+1) ×2.5		ns

Table 13: OMEGA Synchronization Accuracy

The GO button and the trigger detection work only on the master logic analyzer.

Warning:	Tests measured with the OMEGA Logic Analyzer especially in case of using more than one analyzer are highly memory demanding. We recommend at least 4GB operating memory for using two or more OMEGA Logic Analyzers.

[1] Many bus-powered USB hubs claims themselves as self-powered. In case of bus-powered USB hub Windows will deny to power more than one OMEGA Logic Analyzer from a single USB port.

7

Using OMEGA Logic Analyzer under Linux

The support of ASIX products in Linux was discontinued.

Last version where we tested our products was UBUNTU 20.04.2 LTS and Wine 5.0.

Our customers informed us that our products could be successfully used up to Wine 6.0, but not in higher versions.

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

These files are needed for the installation:

- libftd2xx - Linux driver by FTDI, download here: https://ftdichip.com/drivers/d2xx-drivers/
- libftchipid - library by FTDI, download here: https://asix.tech/support_linux_en.html
- lin_ftd2xx - library od ASIX s.r.o., download here: https://asix.tech/support_linux_en.html

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.

The driver can be found on FTDI web at "Drivers/D2XX Drivers" section.

- 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 device 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:

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 :

```
SUBSYSTEMS=="usb",      ATTRS{idVendor}=="0403",
ATTRS{idProduct}=="f1a0",  MODE:="0666",  SYMLINK
+="asix_presto"
SUBSYSTEMS=="usb",      ATTRS{idVendor}=="a600",
ATTRS{idProduct}=="a000",  MODE:="0666",  SYMLINK
+="asix_sigma"
SUBSYSTEMS=="usb",      ATTRS{idVendor}=="a600",
ATTRS{idProduct}=="a003",  MODE:="0666",  SYMLINK
+="asix_forte"
SUBSYSTEMS=="usb",      ATTRS{idVendor}=="a600",
```

ATTRS{idProduct}=="a004", MODE:="0666", SYMLINK+="asix_omega"
VID and PID values can be determined using the listing of connected devices by the lsusb command.

Step 2: Install wine

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

The Wine versions over 6.0 are not supported.

Step 3: Install lin_ftd2xx

The lin_ftd2xx is available at ASIX web.

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.

8

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 14: 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.

9

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	2.0	2.4	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		-1.95 - 0.10	2.85 - 4.90[2]	ns
t_{hold} Data hold after clock		3.75 - 1.10	8.55 - 5.90[2]	ns
OMEGA Synchronization Accuracy				
$t_{\Delta 2}$ Sampling misalignment between 2 analyzers		±2	±5	ns
$t_{\Delta 3}$ Sampling misalignment between 3 analyzers		±2	±10	ns
$t_{\Delta >3}$ Sampling misalignment between *n* analyzers		±(*n*+1) ×2.5		ns
$\Delta f/f_{typ}$ internal clock precision		±50[3]		ppm
T_A ambient temperature[4]	0		50	°C

Table 15: Electrical specification

[1] Not applicable in synchronous timing
[2] Adjustable
[3] OMEGA logic analyzers prior to serial number A6030165 have degraded clock precision to ±200 ppm
[4] Indoor use only

10

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
2020-07-22	A link has been fixed.
2020-08-06	A typing error has been fixed.
2021-01-08	Installation instructions have been updated.
2021-03-10	Added links to Linux driver files.
2021-07-02	Added 1-Wire decoder describing chapter.
2023-08-02	Updated Linux info.
	Pictures 3 and 5 and a typing error have been fixed.
2024-01-10	Added a note that Linux support was discontinued.
2024-01-18	USB plugin licence information has been deleted.