



Reference Handbook Transient Recorder

LTT Labortechnik Tasler GmbH
Friedrich-Bergius-Ring 15
97076 Würzburg
Germany
www.tasler.de

Tel. +49-931-359 61-0
Fax +49-931-359 61-50

This handbook has been created to the best of our knowledge. The information contained herein is given without guarantee and may be changed without notice. The software described is covered by the included software licence.

No part of this work may be copied and distributed, independent of the method of replication, electronic or mechanical without expressed written consent of LTT GmbH.

© Copyright 2007 by LTT Labortechnik Tasler GmbH
Friedrich-Bergius-Ring 15
97076 Würzburg
Germany
E-Mail: info@tasler.de
Internet: www.tasler.de

Contents

Introduction.....	5
Chapter 1: LTT Hardware.....	6
1. Overview	6
2. LTT-184 and LTT-186	7
2.1 Front View.....	7
2.2. View from back	8
3. 19" – Rack system	10
4. Technical Specifications.....	12
Chapter 2: Cascading LTT-Systems.....	13
Chapter 3: LTTview (V. 4.2)	15
1. Installation	15
2. The Graphical User Interface	16
3. Working Online and Offline	16
4. Tab Pages Configuration	17
4.1 ADC	17
4.2 Channels.....	19
4.3 Display	27
4.4 Trigger	30
4.5 Cursor	32
4.6 Files	34
5. Graphical Display	35
5.1 Display functions.....	35
5.2 The Task Bar	37
a) General Functions.....	37
b) Graphical Display Functions	38
5.2 Additional Information	39
Chapter 4: Description of the DLL LTT2API V1.3.0	40
Fundamental concepts.....	41
Source Code samples.....	42
LTTInit.....	45
LTTDelnit	46

LTTTrigger	47
LTTTrigger_SetPreTrigger	49
LTTAnalogChannel	49
LTTDigitalChannel	52
LTT RPMChannel	53
LTTSampleTime	54
LTTGetRAMBufferSettings	55
LTT RAMBufferSettings	56
LTTFileBufferSettings	57
LTTSTREAMSettings.....	58
LTTBufferReset.....	59
LTTUploadParameters.....	60
LTTResetParameter	61
LTTStart	62
LTTStop	63
LTTCheckTrigger	64
LTTTransfer 2Buffers	65
LTTDataTransferStatus.....	66
LTTDataTransferBreak	67
LTTData_GetCounter	68
LTTData_GetBlock	69
LTTData_BlockDone.....	70
LTTGetCalibTemperature	71
LTTGetChannelIDSequence.....	72
LTTGetChannelSequenceLength	73
LTTGetResolution	74
LTTGetChannelRange	75
LTTGetTemperature	76
LTTGetDLLInfo	77
LTTGetDeviceInfo	78
LTTGetDevices	79
LTTSetInternReference	80
LTTGetChannelConverter.....	81
Index.....	82

Introduction

Thank you for deciding on a measurement system from Labortechnik Tasler. From all the currently available systems, you have chosen one of the fastest and most flexible.

Maybe you just started using LTT products and want to learn more about their features or maybe you wish to know the LTT system in and out and are using this handbook as a reference.

In any case, you'll probably be amazed by the possibilities available hidden in these compact black boxes. To enable you to take the most out of using LTT systems, we are constantly keep this hand book up to date in pace with the latest soft- and hardware features.

This handbook covers measurement hardware and software developed and manufactured by LTT GmbH. The various products might be bundled together in multi-device systems or independently.

In **Chapter 1** you'll find a detailed description and diagrams of the LTT hardware solutions pertaining to how it was developed and manufactured.

The LTT-184 and LTT-186 systems are cascadable. You're given the possibility to have hundreds of synchronised measurement channels for storage or further manipulation. **Chapter 2** of the handbook covers the installation of a cascade system.

Chapter 3 presents a complete explanation of the software LTTview. LTTview is not necessarily needed for operation of the hardware, but is specially developed for speed and ease of managing the hardware. Alternatives to LTTview include using in-house software coupled with the DLL LTT2API or even third-party developments as long as they conform to the hardware interface.

Chapter 4 is about the DLL LTT2API, which was designed for ease of integration into customer specific software. All commands in the DLL are described for merging LTT systems into third-party software. Enabling support of widely used analysis tools is very important to the LTT software development.

Obviously this handbook cannot replace actual work with new hardware and software. It is meant to relieve substantial confusion or complications. So if you have questions about our products please give us a call. Perhaps another customer will have similar questions or problems and with your feedback you'll be helping us and them to improve this handbook.

We wish you much success in using LTT products.

LTT GmbH

Würzburg, Germany, May 07, 2007

Chapter 1:

LTT Hardware

1. Overview

LTT currently produces 3 different measurement devices, which are all based on the same technology: LTT-184, LTT-186 and LTT-19X.

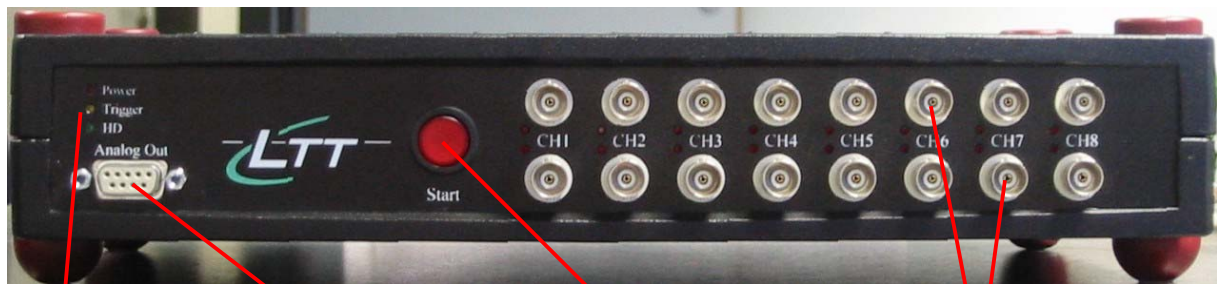
The **LTT-184** is a portable data acquisition system, which is optimized for fast long and short duration measurements. The system provides 8 or 16 individually configurable differential analog inputs and sampling rates from 1 kHz to 20 MHz. An external clock input enables the measurement of an external RPM or frequency of a pulse signal and also enables the cascading of the devices all of which are synchronised with another.

The **LTT-186** is very similar to the **LTT-184**, but contains an additional internal hard disk. Either in Stand-Alone operation or with a notebook it proves itself as a world class multifunctional measurement system. Equipped with either a 9 or 18 GB hard disk the system can even be used in difficult and rough mobile conditions.

A newly developed LTT series is the 19" rack system: **LTT-19x**. Two LTT-184 devices are configured, linked, and built into a standard 19" rack system. All the advantages of the LTT products are bundled with the advantages of a standard form factor. With a total of 32 differential analog inputs this system is predestined for rack mounted test and control systems.

2. LTT-184 and LTT-186

2.1 Front View



2. LEDs "Power",
"Trigger" and "HD"

4. Analog Out
(not used)

1. Start Button

3. BNC Inputs

1. Start Button

The LTT-186 system is capable of taking measurements without a PC. An internal hard disk allows for saving the measurement data. Press the start button to begin the measurement. After taking data, the system is hooked up to the PC and the data can then be viewed, analysed or transferred to the PC.

2. Light Emitting Diodes

Power: LED glows if power is available.

Trigger: LED glows if a trigger was defined and the system is waiting for the trigger signal. As soon as the trigger signal occurs the LED turns off.

HD: LED blinks, when data is being transferred from the LTT-device to the PC or to the internal hard drive (LTT-186 only)

3. 16 or 32 BNC-inputs

An 8 channel system has 16 BNC analog inputs, the 16 channel system 32 analog inputs. The channels are numbered sequentially, each having 2 inputs, BNC+ and BNC-, and the DLL orders them similarly.

Between the two rows of BNC inputs are LEDs which show the state of the channel.

Lower diode: A

Upper diode: B

States:

Single Ended + (SE+): LED A glows, B dark

Single Ended - (SE-): LED A dark, B glows

Differential Ended (DE): both glow

Off (GND): Both LEDs dark

8 channel devices have their analog inputs on the front side of the housing. 16 channel devices have 8 analog inputs on front side of the housing and an additional 8 channels on the back side of the housing. Every BNC shielding is tied to ground, as defined at the back of the housing. The inner conductor is the signal.

4. Analog Out

This not used.

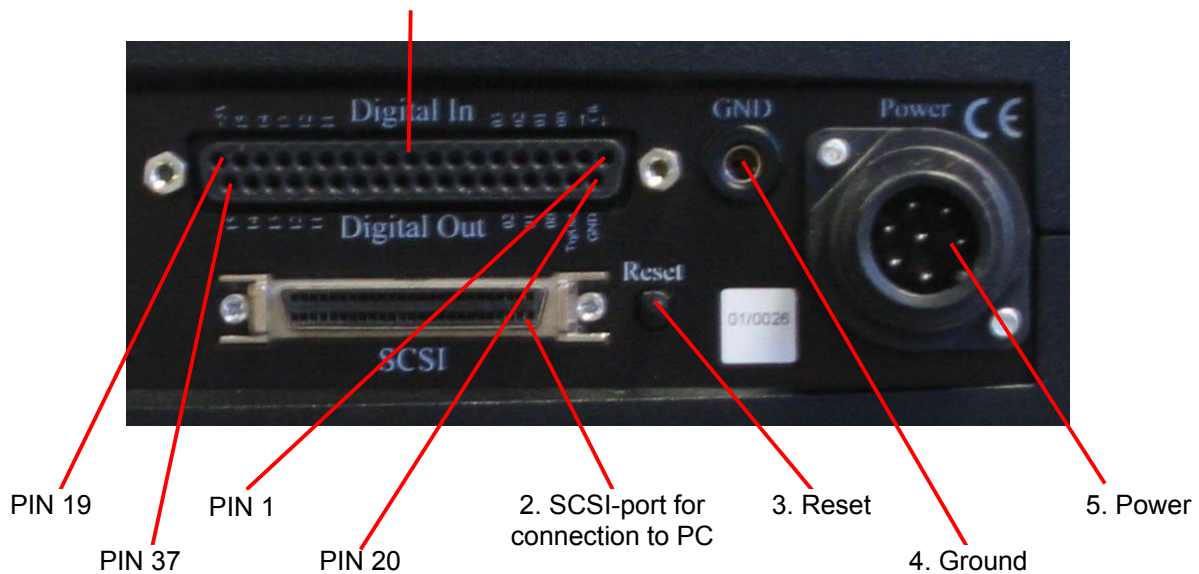
2.2 View from back

Seen is a 16 Channel Unit:



Detail view:

1. Digital In and Output



1. SUB-37 Digital Input:

Digital input can be connected using 37 sub-D socket. Digital trigger is available from Bits 9 to 15. There is a delay between the analog and digital signals which depends on the sampling rate and the number of active channels. Devices having the feature bit “Digital Delay” set do not have this delay. The delay is automatically corrected using LTT2api using version greater than 1.3.0.3 RC11

Pin layout:

Important: All pins are TTL compatible and are not protected. Therefore all inputs and outputs must be driven with 5 V or 0 V. All outputs can source a maximum of 10 mA.

PIN	Meaning	
1	Clock In	(TTL, not protected)
2	Clock Out	(TTL, not protected)
3	Digital In Bit 00	(TTL, not protected)
4	Digital In Bit 01	(TTL, not protected)
5	Digital In Bit 02	(TTL, not protected)

6	Digital In Bit 03	(TTL, not protected)
7	Digital In Bit 04	(TTL, not protected)
8	Digital In Bit 05	(TTL, not protected)
9	Digital In Bit 06	(TTL, not protected)
10	Digital In Bit 07	(TTL, not protected)
11	Digital In Bit 08	(TTL, not protected)
12	Digital In Bit 09	(TTL, not protected)
13	Digital In Bit 10	(TTL, not protected)
14	Digital In Bit 11	(TTL, not protected)
15	Digital In Bit 12	(TTL, not protected)
16	Digital In Bit 13	(TTL, not protected)
17	Digital In Bit 14	(TTL, not protected)
18	Digital In Bit 15	(TTL, not protected)
19	+5V (max. 40 mA)	(TTL, not protected)
20	Ground (max. 40 mA)	(TTL, not protected)
21	Trigger Out	(TTL, not protected)
22	Digital Out Bit 00	(TTL, not protected)
23	Digital Out Bit 01	(TTL, not protected)
24	Digital Out Bit 02	(TTL, not protected)
25	Digital Out Bit 03	(TTL, not protected)
26	Digital Out Bit 04	(TTL, not protected)
27	Digital Out Bit 05	(TTL, not protected)
28	Digital Out Bit 06	(TTL, not protected)
29	Digital Out Bit 07	(TTL, not protected)
30	Digital Out Bit 08	(TTL, not protected)
31	Digital Out Bit 09	(TTL, not protected)
32	Digital Out Bit 10	(TTL, not protected)
33	Digital Out Bit 11	(TTL, not protected)
34	Digital Out Bit 12	(TTL, not protected)
35	Digital Out Bit 13	(TTL, not protected)
36	Digital Out Bit 14	(TTL, not protected)
37	Digital Out Bit 15	(TTL, not protected)

2. SCSI-Connection for PC (50 pole)

For connecting the system to PC's and Notebooks. Also possible is the use of FireWire-SCSI and USB2-SCSI converters, which are simply connected to the SCSI port. Configuration and control of the measurement system is accomplished with this interface.

3. Reset

Button which initializes the system. Please always turn on the system first then boot the PC.

4. Ground (GND)

Banana plug for ground connection. **Important:** The connection to ground is identical to the PC ground. The measurement result is only valid when the ground is defined, i.e. a good connection to ground exists.

5. Power connection for the external power supply.

The screw down connector gives an exceptionally safe and stable connection of the measurement system to the current supply.

3. 19" – Rack system

In the following, the sockets and their designation for the 19" rack system is described. All of them are found at the back of the system (see Fig. 1).

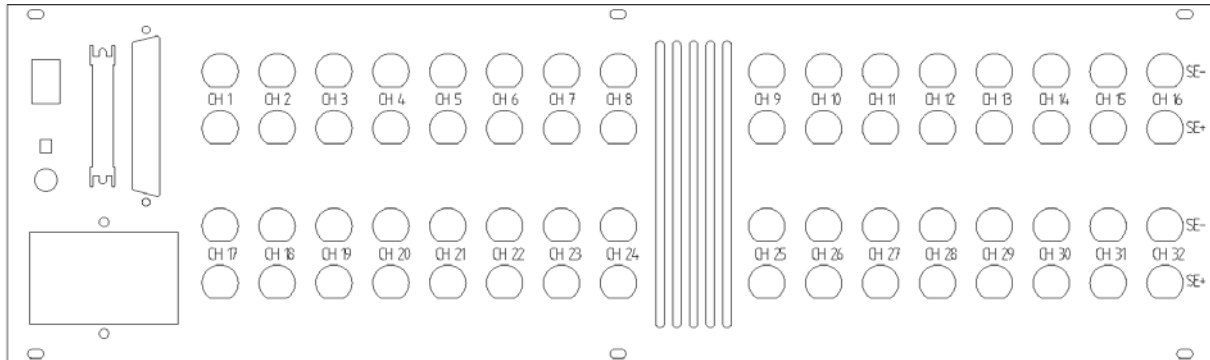


Figure 1: Rear view of the 19" rack system

The following connections are available:

1. Power (230V):

The main power switch which must be turned on before the power switch on the front can work. The 230V fuse is found here and is accessible externally.

2. GND:

Banana plug to define the LTT system ground.

3. IEEE1394 ; pass through for FireWire cable:

Not currently used.

4. SCSI-connection:

The SCSI port for communication and data transfer with the 2 LTT devices.

5. SUB-37 Digital-Input:

Unlike the LTT-18x systems the 37 pole female port of the LTT-19x system is for digital input only. 15 digital inputs are available because Bit 0 is for internal synchronisation. Triggering is available from Bits 9 to 15.

Pin layout:

PIN	Meaning
1	not used
2	not used
3	not used
4	Digital Input, Bit 1
5	Digital Input, Bit 2
6	Digital Input, Bit 3
7	Digital Input, Bit 4
8	Digital Input, Bit 5
9	Digital Input, Bit 6
10	Digital Input, Bit 7
11	Digital Input, Bit 8
12	Digital Input, Bit 9

13	Digital Input, Bit 10
14	Digital Input, Bit 11
15	Digital Input, Bit 12
16	Digital Input, Bit 13
17	Digital Input, Bit 14
18	Digital Input, Bit 15
19	+5V
20	not used
21	not used
22	not used
23	not used
24	not used
25	not used
26	not used
27	not used
28	not used
29	not used
30	not used
31	not used
32	not used
33	not used
34	not used
35	not used
36	not used
37	not used

6. 64 BNC-Connectors:

Every channel has 2 inputs BNC+ and BNC-. The channels are labelled monotonically and the DLL addresses them identically. LTTview separates the channels by device e.g. the DLL's 17 channel is device 2 channel 1. Status LED's do not exist.

Every BNC inputs exterior contact is connected to ground and the inner contact is the signal.

4. Technical Specifications

Model	LTT-184-8ch	LTT-184-16ch	LTT-186-8ch	LTT-186-16ch	LTT-19x
Number of Channels:	8 differential	16 differential	8 differential	16 differential	32 differential
AD-Converter:	8 separate simultaneous	16 separate simult..	8 separate simult.	16 separate simult.	32 separate simult.
Stand-Alone:	No	No	40 GB HD	40 GB HD	No
RAM:	128 MB	128 MB	128 MB	128 MB	2 x 128 MB
External Frequency / RPM:	Yes / No	Yes / Yes	Yes / No	Yes / Yes	No / No

Preamp				
Input coupling	Option A: differential or referenced to ground, individually configurable		Option B: differential or referenced to ground, individually configurable	
	AC	DC	DC (1:1)	DC (1:10)
Input impedance	10 nF in 0,1 MΩ, 150 pF	0,1 MΩ, 150 pF	0,1 MΩ, 150 pF	1 MΩ, 15 pF
Protection	± 200 V differential, ± 100 V for each channel			
Measurement range	± [50V,20V,10V,5V,2V,1V] per channel configurable Option A.			
Floating voltage	≤ (input range x 7.7) ≤ 100V			
Drift	≤ 25 ppm/°C			
Offset drift	≤ 0.2 LSB/°C			
Digitaler Input	16 Bit, TTL			
Signal bandwidth	6.5 MHz (-3dB)			
Crosstalk	< measurement resolution			
Sampling rate	1 kHz to 2.5 MHz (16 Bit); up to 20 MHz (12 Bit)			
Clock	internal, external			

Trigger	
Trigger	Pre and Post trigger on multiple channels
Source	Analog and digital Inputs

Overall precision		
Time axis deviation	≤ 20 ppm using internal clock	
Internal noise at 20 MHz sampling rate	≤ 5 LSB	
Internal noise at 2.5MHz sampling rate	≤ 4 LSB (0.006 %)	
Measurement Range	Zero point error	Measurement Error
± 1 V, ± 2 V	≤ 5 mV	≤ 0.08% Input Range
± 5 V, ± 10 V	≤ 10 mV	≤ 0.05% Input Range
± 20 V	≤ 20 mV	≤ 0.1% Input Range
± 50V	≤ 40 mV	≤ 0.2% Input Range
AC coupling		≤ 3% Input Range

Transfer Speeds	
Internal RAM	200 MB/s (100 MS/s)
Internal hard disk (LTT-186)	max. 19.6 MB/s (9.8 MS/s)
PC (SCSI)	typ. 17 MB/s (8.5 MS/s)
PC (IEEE1394)	typ. 9 MB/s (4.5 MS/s)
PC hard disk	Transfer speed to PC hard disk depends on PC type. Typ. (April 2003):10 MB/s (5 MS/s)

RAM	
Size of RAM	128 MB (64MS)
Access time	4 ns
Internal HD (LTT-186)	40GB data buffer not formatted (also available: 1 GB Flash Disk)

Operating Conditions	
Power	9-36VDC / 100-240VAC, 80 to 120 W
Temperature	+10 to +40 °C
PC connection	SCSI-II-Interface, 8 Bit, 20 MHz ULTRA, IEEE1394 400 MBit/s, USB 2.0
OS	Win2000, WinXP SP1, WinNT4 (Desktop only)

Dimensions	LTT-18x	9-36VDC Power Supply	LTT-19x (19" - Rack)
Height	63 mm	44 mm	3 units
Width	325 mm	75 mm	437 mm
Depth	280 mm	166 mm	565 mm
Weight	ca. 4.5 kg	ca. 660 g	ca. 20 kg

Last update: May 21, 2003

Chapter 2:

Cascading LTT-Systems

The measurement systems LTT-184 can be easily cascaded together giving a fully synchronised system with practically limitless number of channels. All the data is gathered by the PC to a single file.

Construction of such a cascaded systems is described in the following:

In general when connecting SCSI-II devices to the SCSI bus the **last** device must be **terminated** and the other devices are not terminated. LTT preconfigures the devices such that the device with the lowest serial number is the master (terminated) and all other devices are not terminated. This means that only the master can be run alone while the slave devices must be modified. Note: cascaded devices using FireWire or USB 2.0 are all terminated.

Example: 2 devices

1. Place the master on top of the slave and connect the master device to the end of the SCSI-II flatband extension cable. Master device has the **lowest** serial number.
2. Connect the slave device to the middle of the SCSI-II flatband extension cable.
3. Attach the included synchronisation cable to both LTT systems. Turn off PC if necessary and connect the SCSI cable to the flatband extension cable then to the PC.
4. Start both systems and then the PC.
5. Start LTTview (see Chapter 3 for description of LTTview). Both devices will be automatically recognised. From the tab sheet ADC one can select which LTT system to configure.

Example : 3 or more devices

Cascading 3 or more devices is analogous to cascading 2 devices. Care must be taken to order the devices with the lowest serial number on top and the highest serial number on bottom. Attach the delivered SCSI-II flatband extension cable and synchronisation cable as mentioned above, start from the top and working down. If any of the mentioned cables are not included please contact LTT. +49-931-35961-0 or service@tasler.de.

LTTview sorts the devices according to their serial number where device #1 (master) is the device with lowest serial number, device #2 the next lowest number, etc.

Important: Devices which were bought separately need to be modified:

- The devices SCSI ID must be unique. All individually delivered LTT devices have the SCSI ID "2". Call LTT for instructions on how to change the SCSI-ID.
- There is one master and it is terminated. Slave devices must be unterminated if using SCSI.

The above pertains to cascading using the SCSI bus. Cascading using FireWire instead requires all systems to be terminated.

Chapter 3:

LTTview (V. 3.0)

LTTview is the management software and the graphical user interface to work with LTT transient recorders. Using this software you can configure sophisticated measurements and see the results in real time, or offline, or export the data to your favorite analysis software.

1. Installation

The following assumes that SCSI or FireWire Adapters are already installed to your PC or Notebook:

1. Make sure both systems are off before connecting any cables.
2. Attach the LTT equipment to your PC using the SCSI interfaces and plug in the power supply.
3. Start the LTT system and afterwards your PC. Important note for everyone: Please *always* start the LTT device first! Note for users of Win2000, WinNT or WinXP: When installing the LTT device you need to have administrator's rights.

The LTTview software is in the LTT device and will now be automatically installed on your PC. By the way: Your PC or Notebook will not need any extra drivers. After the auto-installation LTTview will automatically start to add some settings and to install the Adaptec ASPI Layer.

4. In case LTTview does not start automatically please use your Windows Explorer to start LTTview from the disk drive "LTT18x".
5. When you're asked to reboot your PC, please do so.



When starting your PC again, the symbol for LTTview should be on your desktop. By double clicking it the software can be started. Immediately, the LTT equipment is ready for use.

From now on when starting the system and the PC, simply click on the LTTview icon to start the software.

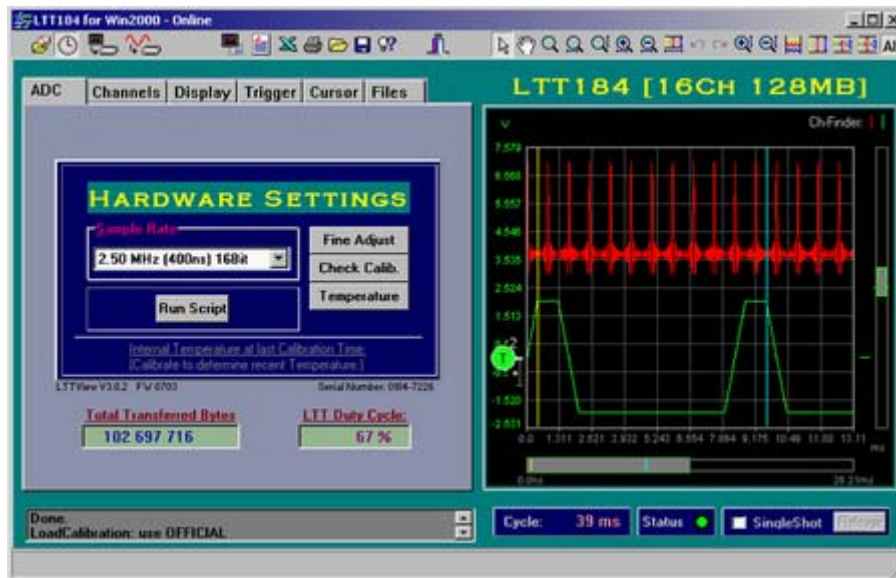
LTT-18x systems are recognized by Microsoft Windows as regular PC-hard disk. You can actually access the data using software like Windows Explorer. LTT-186 systems have an additional hard disk which is called LTTDATA.

Note: Since LTT-18x is being recognized as a hard disk, your virus protection software will possibly give false warning of a "New Boot Virus".

2. The Graphical User Interface

The graphical user interface includes on the left hand side a wide range of possibilities for your specific configuration of the system. With a simple click on the different tab sheets you can do all the configurations and adaptations necessary for your needs.

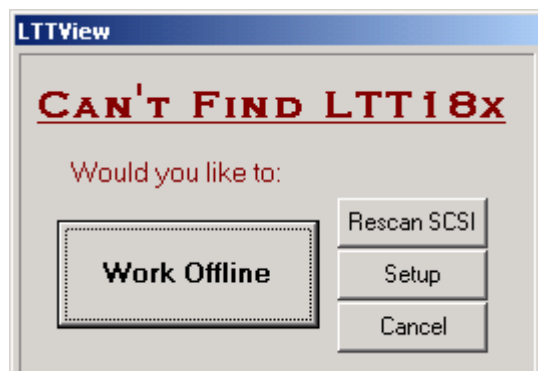
On the right hand side you see an oscilloscope-like screen. With the help of the symbols above the screen window you can access various functions like zooming in along the x- or y-axis, focus on certain signals, resolution of the displayed data and many more (see chapter 5. Graphical Display for more details).



3. Working Online and Offline

Online is working with a device and Offline is without. When started, LTTview first seeks for the compatible hardware connected to the PC. If found LTTview registers the device and shows its serial number on the tab page **ADC**, further information can be found in the grey scroll box. LTTview is online using default configuration found in the windows or Winnt directory. This default configuration can be changed using the controls on the tab page **Files**.

It is also possible to use LTTview without the hardware connected. If LTTview does not find an appropriate device a window will appear with the option **Work Offline**. After which you may load data from the hard disk by pressing the **Load Data** icon or from the tab page **Files** the button **Load Data**.



Please note, that in the offline mode the data may not be changed and neither the configuration of a measurement. The configuration remains the same at the time data was taken.

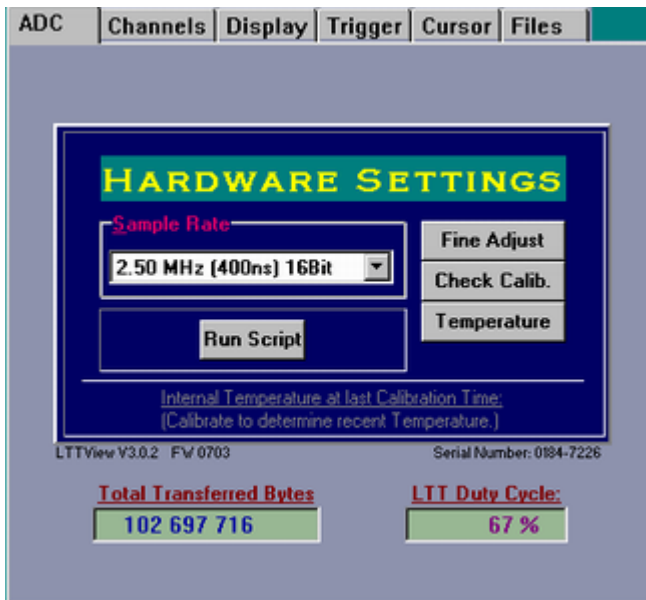
If a device is connected and can be seen from your Microsoft Windows Explorer (see above Installation) a **Rescan SCSI** might work else try **Setup**.

4. Tab Pages Configuration

LTTview has many different measurement possibilities available and the following section covers what the tab pages have to offer.

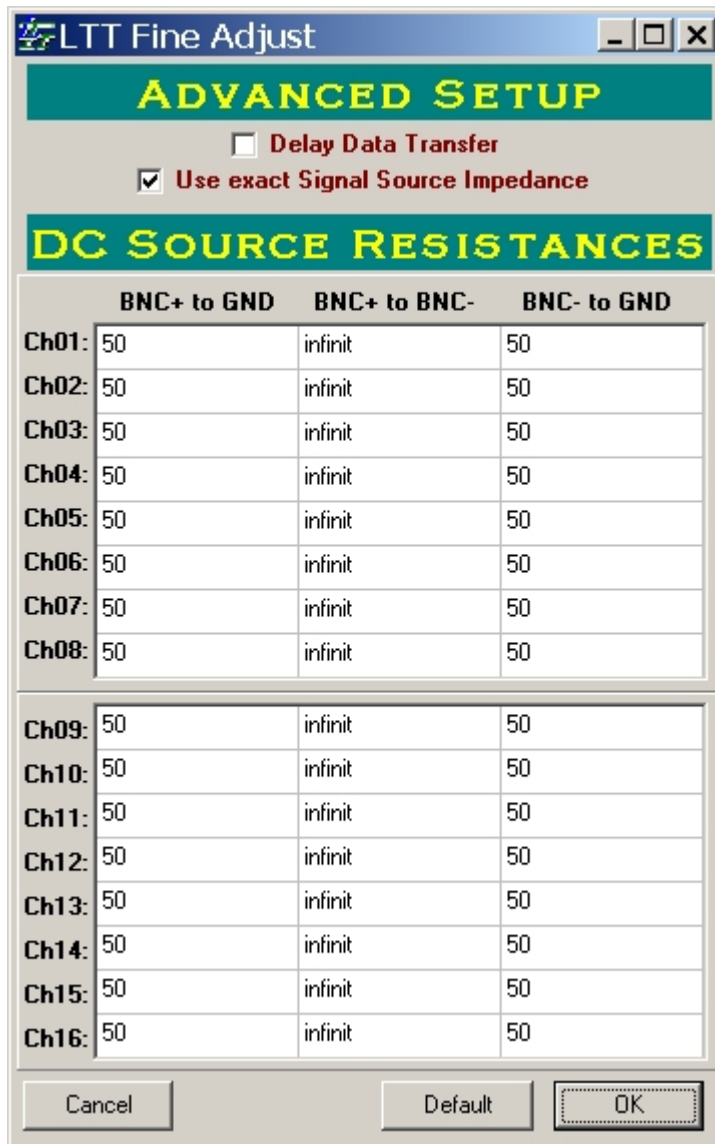
4.1 ADC

ADC (Analog to Digital Converter).



Sample Rate: Selects the sampling rate for all active channels. Total sampling rate is the number of active channels multiplied by the sampling rate. Also displayed is time interval between samples for a given channel and the resolution. The sampling rate spans from 751 Hz (16 bit) to 20 MHz (12 bit). The resolution for the external frequency can also be selected. The maximum total sampling rate is 100 MHz.

Fine Adjust: Selecting **Fine Adjust** will bring up the **LTT Fine Adjust** window which allows for advanced settings regarding the output impedance of the signal source:



Delay Data transfer: Reduces noise caused from the SCSI transfer. During the measuring process no data is being transferred to the PC. In this case the data depth is limited to the size of the internal buffer of 128 MB. Please note: This mode is not supported by all PC's.

Use exact Signal Source Impedance: This is the default. If you use something other than 50 Ω cable it is recommended to measure its resistance and enter it into the appropriate boxes.

Default: Sets all values in the table back to default values (50 Ω).

Check Calib.: Performs a quick calibration.

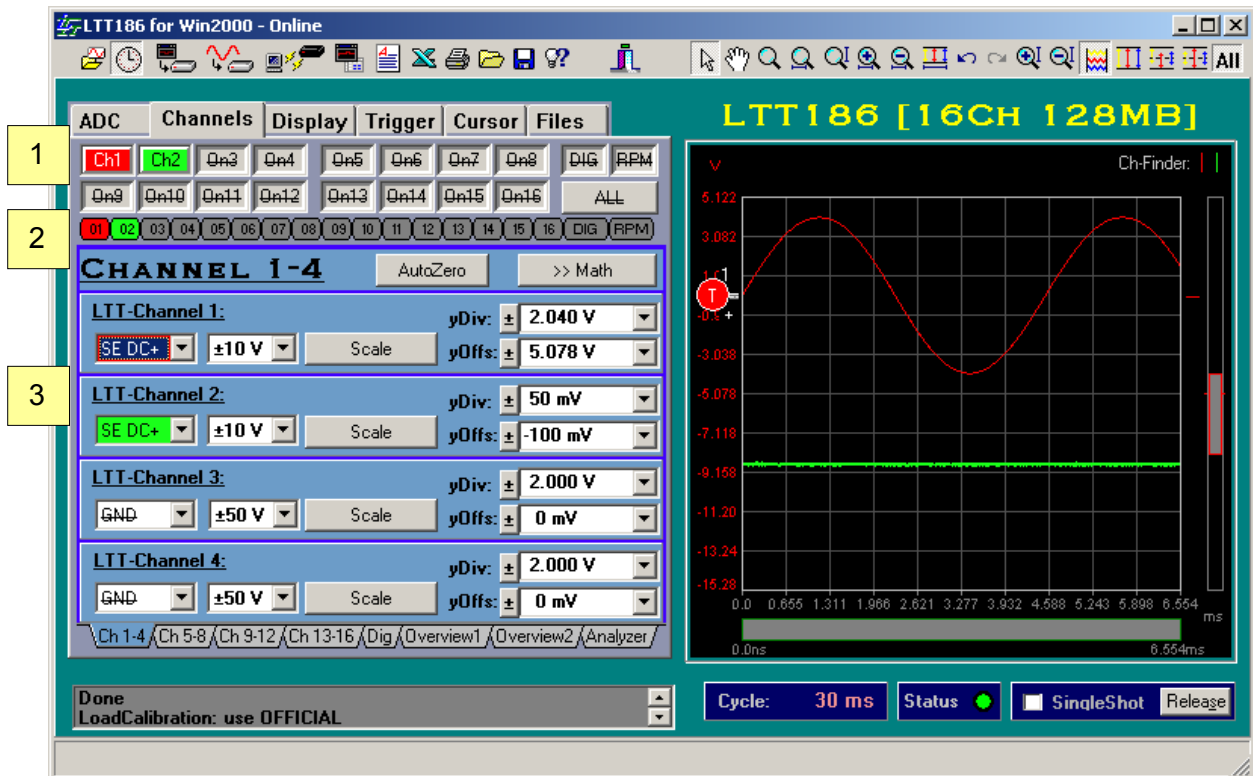
Temperature: LTT transient recorders have a built in temperature sensor. Clicking **Temperature** measures the approximate temperature within the device.

Total Transferred Bytes: Shows the total amount of data that has been transferred on the SCSI bus and includes the SCSI overhead.

LTT Duty Cycle: Shows the work load of the LTT system

Below the tab pages is a small grey status window which will show such information as the type of device found, whether a measurement has been completed, etc. Double-clicking the window opens an editor window showing all logged status information.

4.2 Channels



1

Ch1 – Ch16: Activates or deactivates the corresponding analog channel with the left mouse button. Hint: If the mouse pointer is placed over an active channel then that channel will be in the display's foreground with its corresponding Y-axis scaling. The right mouse button changes the channels color.

DIG: Main switch for the digital input which is found on the back side of the device.

RPM (Rotation per Minute): The RPM channel enables you to measure an external clock. This feature is very useful for example when measuring rotating movements, for example to measure torsional vibrations.

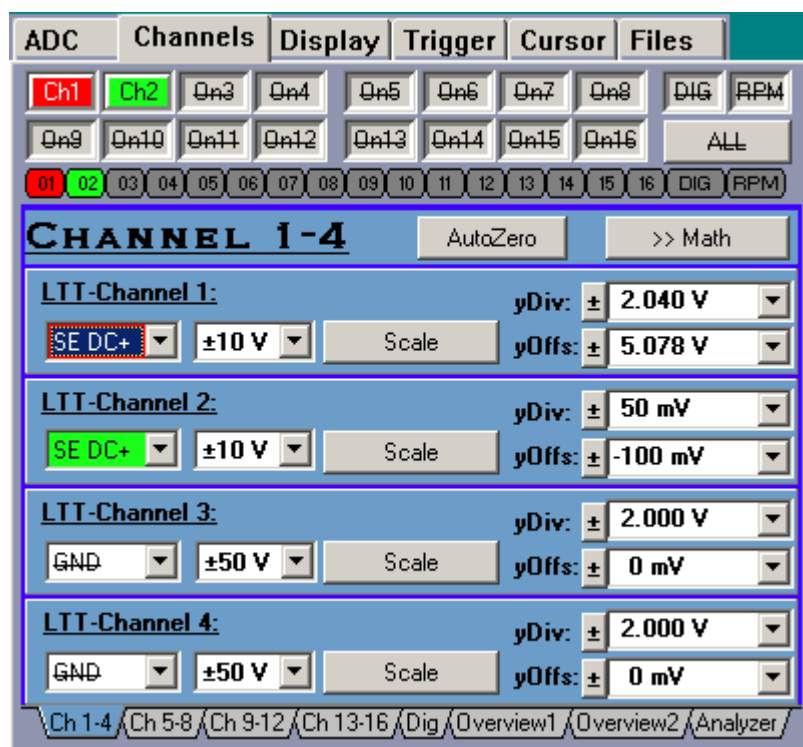
RPM allows for measurements synchronously to an external pulse signal. In addition to measuring the external pulse signal RPM delivers the time reference between the pulse signals.

2

01-16, DIG, RPM: The single row of button turns off the display of the corresponding signal. The channel is still active.

3

Next is a blue window showing various features for channels 1 through 4 such as: input coupling, range, etc.



The left combobox has the following choices for the input coupling for the individual channel:

Channel with AC/DC option	
GND	Ground
SE DC+	Single Ended DC Plus (0.1 MΩ)
SE DC-	Single Ended DC Minus (0.1 MΩ)
DE DC	Differential Ended DC (0.1 MΩ)
SE AC+	Single Ended AC Plus
SE AC-	Single Ended AC Minus
DE AC	Differential Ended AC

Channel with 1 MΩ option	
GND	Ground
SE DC+	Single Ended DC Plus (0.1 MΩ)
SE DC-	Single Ended DC Minus (0.1 MΩ)
DE DC	Differential Ended DC (0.1 MΩ)
SE 1M+	Single Ended DC 1:10 Plus (1 MΩ)
SE 1M-	Single Ended DC 1:10 Minus (1 MΩ)
DE 1M	Differential Ended DC 1:10 (1 MΩ)

Choosing “Single Ended” means that only one of the two differential analog inputs is activated. “Plus” is the lower row of inputs, “Minus” is the upper row of inputs. Choosing “Differential Ended” allows to measure differential signals. The LED’s next to the inputs show which state is activated.

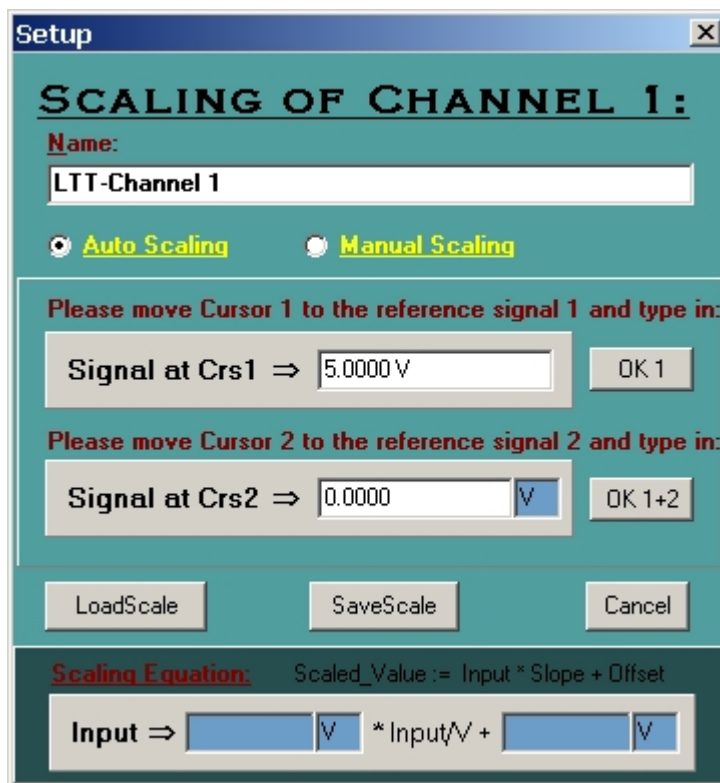
The second combobox will show the input range for the respective channel. This depends on which input impedance is chosen.

0,1 MΩ- Input Impedance
± 50 V
± 20 V
± 10 V
± 5 V
± 2 V
± 1 V
± 25V
± 2,5V

1 MΩ- Input Impedance
± 500 V
± 200 V
± 100 V
± 50 V
± 20 V
± 10 V
± 250 V
± 25 V

Scale: This option allows the user some flexibility in the type of units used giving meaningful representation to the data, i.e. pressure, counts, etc instead of volts. The scaling can be defined automatically (using the cursor in the display) or manually (by typing in the preferred scale).

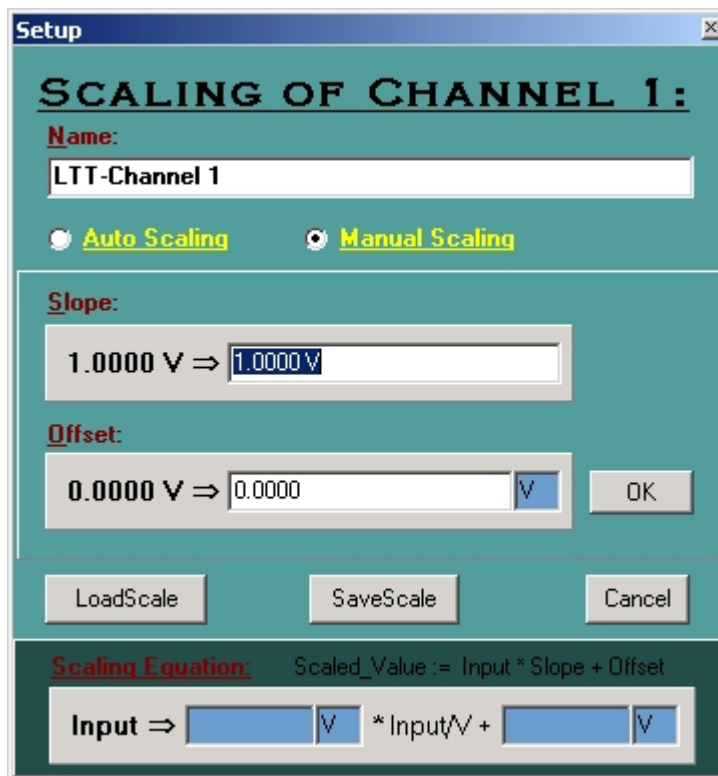
- **Auto Scaling:**



With the help of the display the scaling may be defined automatically. In order to do so please set the first cursor on the preferred position of the signal curve and type in the value (**Signal at Crs1**) which the signal should have at that position. Click **OK 1** to enter the value. (For information on how to use cursors please see Chapter 4.5).

To proceed place the second cursor at the preferred second position of the signal curve and type in the respective desired value. Accept the value clicking **OK 1+2**. The software will now automatically calculate the necessary scaling.

- **Manual Scaling:**



The scaling is:

$$\text{Val}_{\text{User}} = \frac{\text{Val}_{\text{Volt}}}{\text{Volt}} * \text{Factor}_{\text{User}} + \text{Offset}_{\text{User}}$$

The unit may be adjusted manually as well.

Example: Measurement of currents. With the output characteristics of the sensors (e.g. $2\text{V} \hat{=} 1\text{A}$) gives:

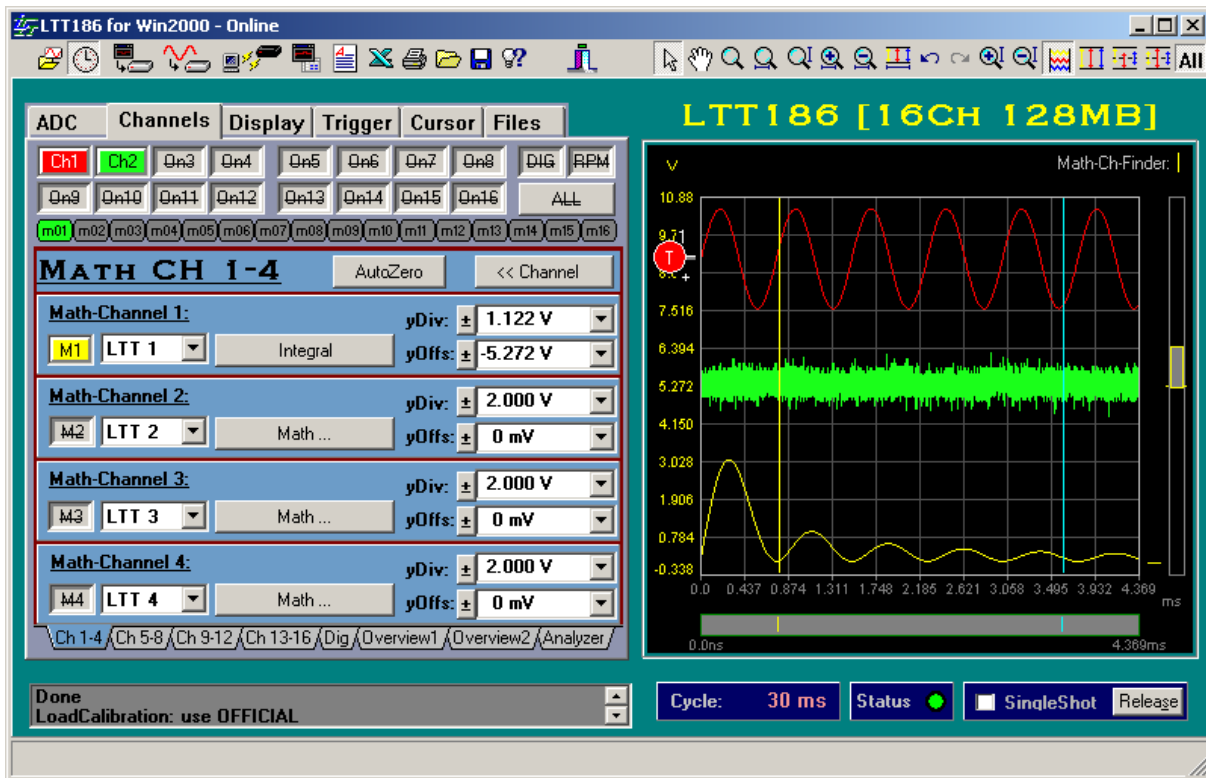
$$\text{Val} [\text{A}] = 0.5 \text{ A} * \frac{\text{Val} [\text{V}]}{\text{Volt}} + 0 \text{ A}$$

After closing the dialog window the scaling of the y-axis of the respective channel will be changed.

yDiv: Similar to oscilloscopes the zoom size based on graph divisions is shown (y-axis; vertical offset). Different to oscilloscopes this only changes the displayed zoom size of the signal without changing the actual measurement range of the transient recorder. The default configuration for the graph is 10 divisions and with this edit box the user can select or type in the size of the divisions.

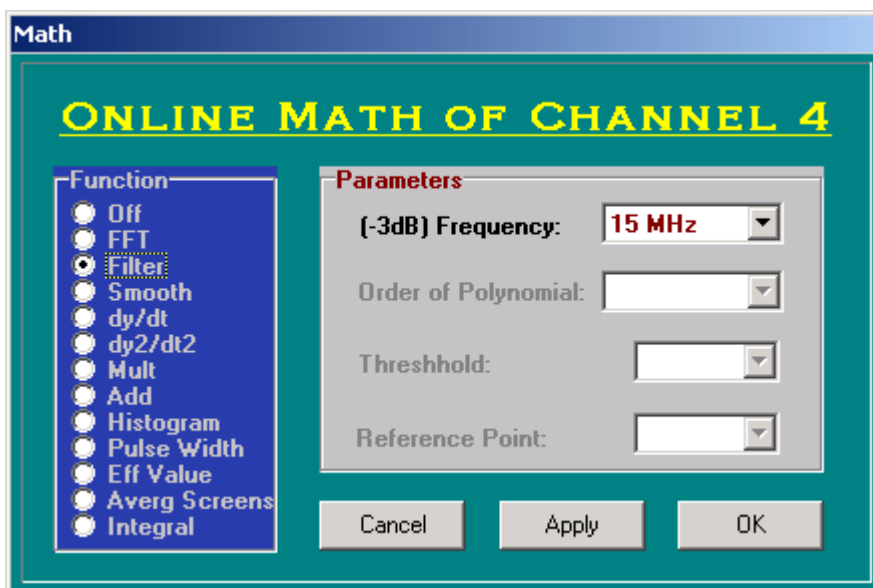
yOffs: Defines the y-offset of the screen. Enables moving the vertical position of the signal range.

Math: Toggles between the configuration windows for the analog channels and the math channels.



In math mode, the buttons found far to the left (M1, M2, ...) activate or deactivate math channels. The combobox between the buttons sets the source of the math channel which can also be another math channel.

Math...: Use this to choose the type of mathematical operation to be performed online on the source signal. After selecting a mathematical function the labelling of the buttons change (e.g. "Filter (-3dB:26kHz)").





The following options are available:

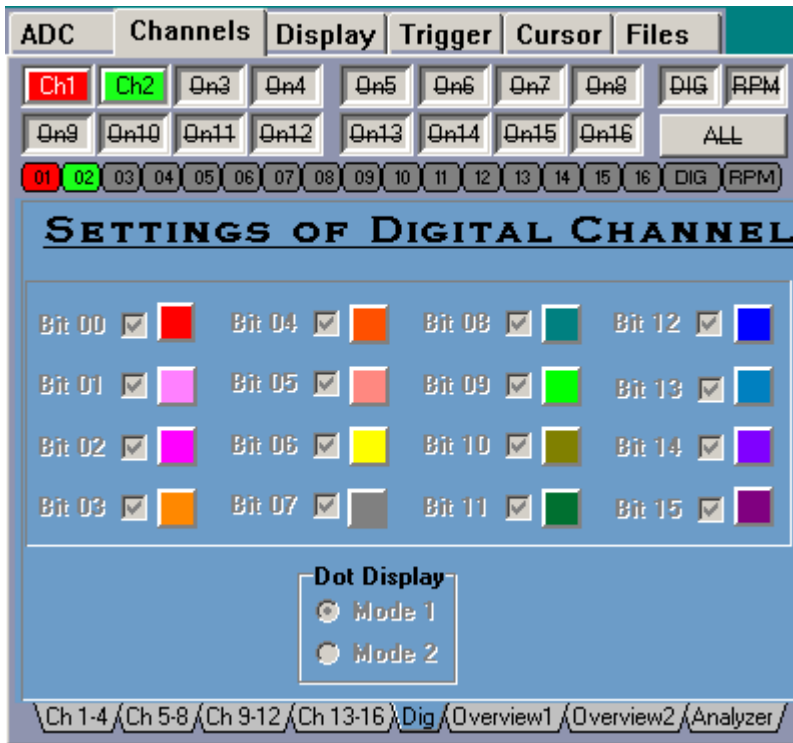
Off:	deactivates the channel and sets to default value.
FFT:	selects Fast Fourier Transform with a the resolution dependent on sampling rate and memory depth.
Filter:	low pass filter whose -3dB point depends on sampling rate.
Smooth:	Sovitsky-Golay smoothing filter, which calculates a local, unweighted average value.
dy/dt:	1 st derivative
dy2/dt2:	2 nd derivative
Mult:	multiplication of two channels
Add:	addition of two channels
Histogram:	plots the counts of a signal.
Pulse Width:	analysis of the pulse width and the frequency of the pulse
Eff Value:	effective value of a sequence of pulses.
Average Screens:	running average is computed
Integral:	integrates the source signal

The changes can be applied using **Apply** or **OK**.

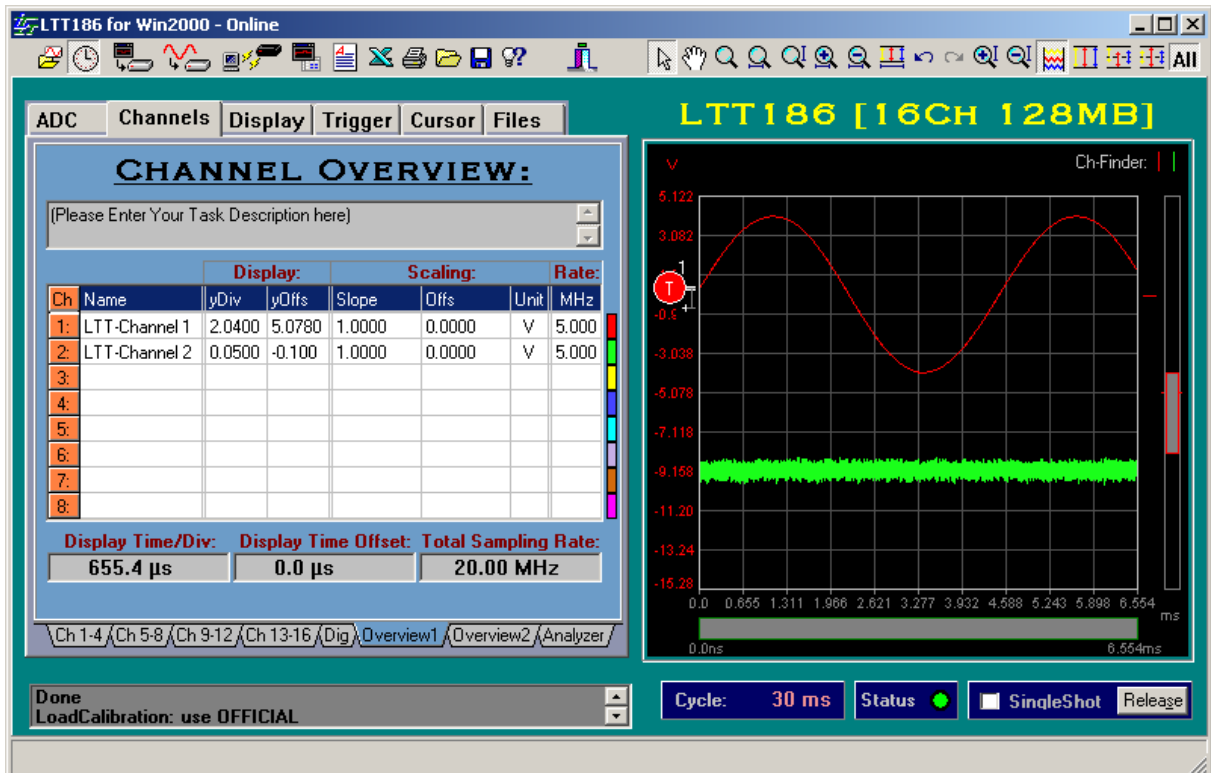
Like the analog channels, the math channels also have the display variable *yDiv* and *yOffs* to adjust the displayed curves.

yDiv:	Sets the size per division in the display in the vertical direction
yOffs:	Sets the offset of the display for the corresponding channel, moving along the y-axis.

Digital Input: For activating and changing the display of the bits from the digital input. Mode 1 separates the bits into curves  and Mode 2 is a color stream display where high is colored and low is white .

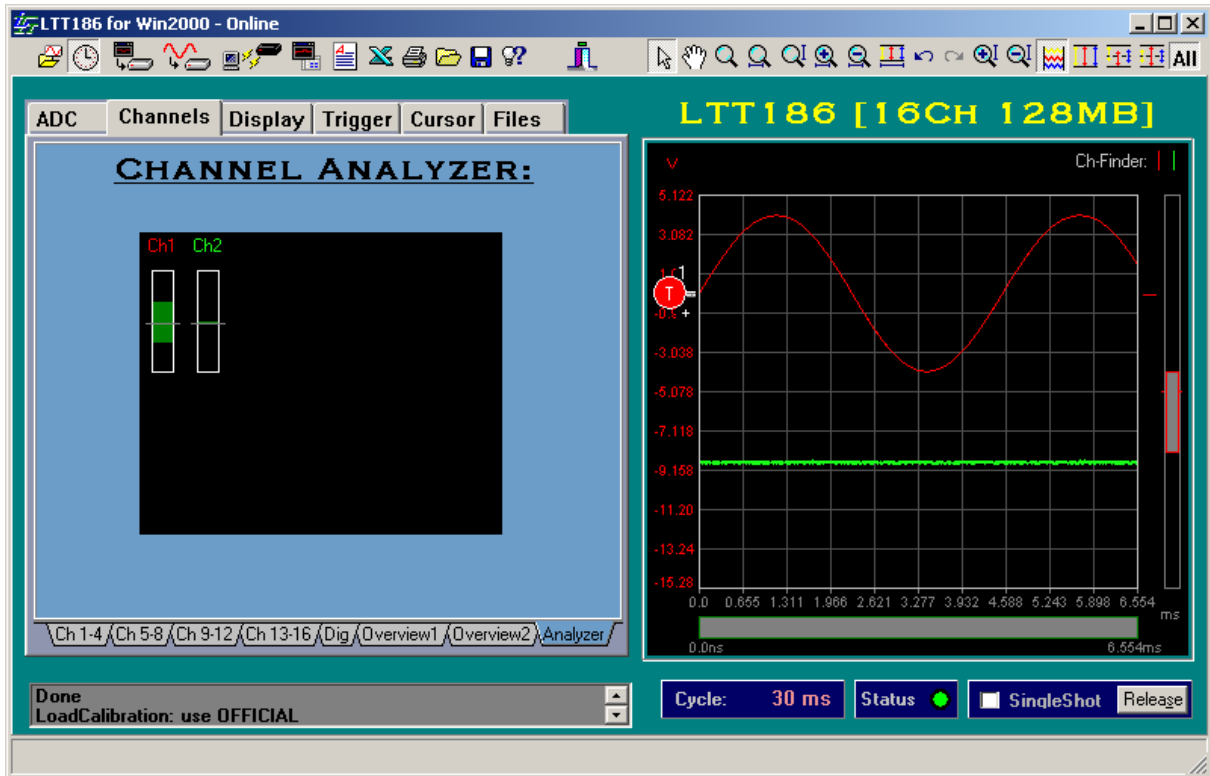


Overview1/Overview2: The overview contains a gray box for the user task information and an overview of the 1-8 or 9-16 channels.

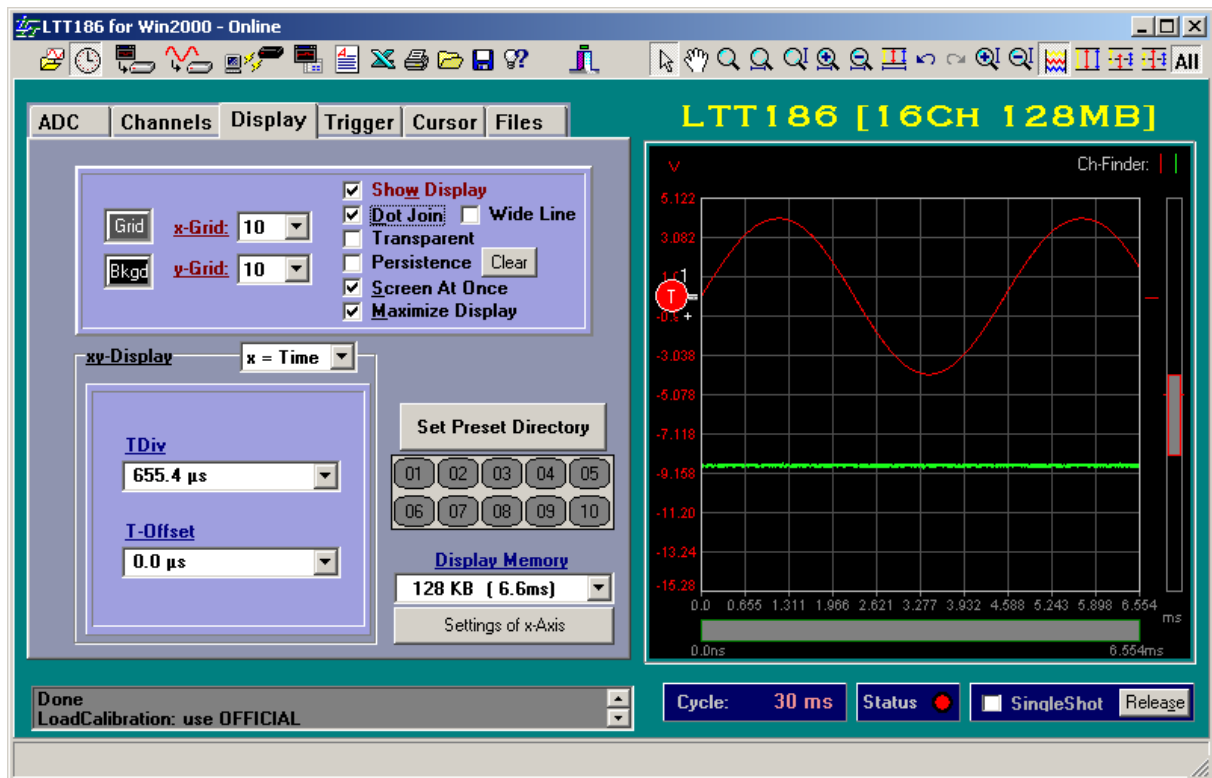


The text of the “Task Description” will be printed on the print out formula together with measurement details and the signal curves.

Analyzer: The Channel Analyzer is a convenient tool to quickly check if all channels are within $\pm 90\%$ of their input ranges. The vertical bar is green if the signal's extrema are within its input range and red if beyond the input range.



4.3 Display



Grid: Turn on or off the grid and with the left mouse button sets the color.

x-Grid: Sets the number of grid boxes in the display along the X-axis

y-Grid: Sets the number of grid boxes in the display along the Y-axis

Bkgd: Sets the color of the background. By clicking the right mouse button the standard color options appear of which you can choose.

Show Display: Turns off/on the display.

Dot Join: Plots the data using lines when checked else as points.

Wide Line: Lines will appear fatter.

Transparent: The lines plot below can be “seen” through the lines on top.

Persistence: Like in a histogram the occurrence of a signal will be counted, but color will be used (variable saturation levels). The more often the signal occurs the brighter it will be plotted. Persistence is a fantastic tool whenever recurring signals are to be controlled for deviations. Persistence can be used in online as well as offline mode. Offline it allows to scroll through a set of recorded files with focus for example on a specific waveform. That way you can easily detect abnormal behavior of the wave.

Screen at Once: Instead of plotting the data as it comes LTTview waits until the display memory is full. Default is off and data is plotted as it comes.

Maximize Display: Maximize the display window

xy-Display: Configuration of the xy display by choosing the type of x-axis.

Example:

x=Time: X-axis has units of time and the two parameters below are available.

TDiv: Size of the time divisions.

T-Offset: Start of the x-axis.

Set Preset Directory: Selects the directory where the user presets are to be stored. Directly below is a field with 10 presets. The presets are for the display only. Intent was to give the user the ability to view and compare many channels easily. Switch off the display of channels you don't wish to compare, channels are still active just not seen, and set the cursors and zoom. Press and hold a preset button till it blinks. The preset is now saved to the prescribed directory.

Display Memory: Sets the memory depth of the display (amount of the momentarily displayed data). Maximum is 16 Mbyte (8 MS).

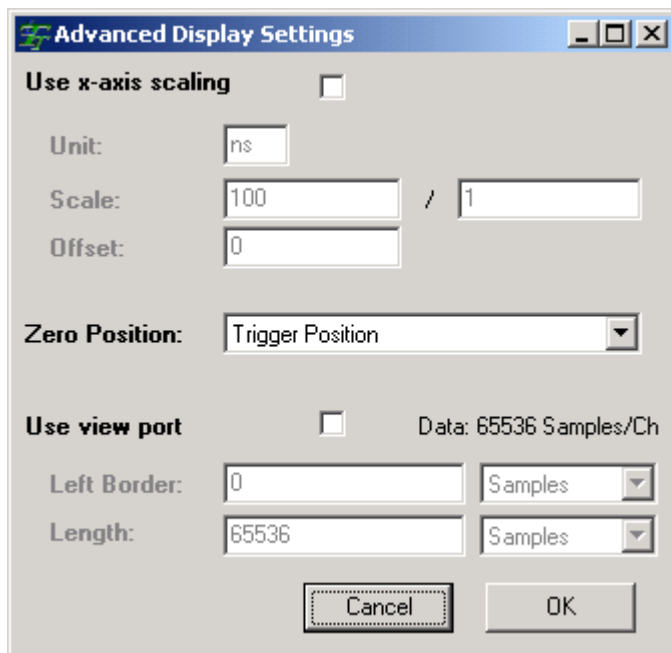
Settings of x-Axis: Opens the window **Advanced Display Settings** for advanced settings of the labelling of the x-axis and for the definition of the **View Port**. The view port is the selection of the total measured data which is to be managed in the display of LTTview. When the view port is not activated, the total measured data can be displayed by LTTview.

Example of an application:

Attached to a drive shaft is a variably rotating disk with 3600 slits which define the synchronisation frequency. In the LTTview display exactly one rotation (360°) is to be shown.

LTTview only allows powers of 2 values for the memory depth of the display (e.g. 4096 Samples = 8 kB). With the help of the view port the selection of 3600 Samples may be defined as a selection of the total measured data.

For the labelling of the x-axis the unit 1° per 10 is appropriate.



Use x-axis scaling: Adjustment of the labelling of the x-axis

- Unit:** preferred unit of the x-axis (5 characters allowed, e.g. mm/s²)
- Scale:** calculation factor for the unit per sample
- Offset:** offset of the x-scaling (samples per channel)

Zero Position: Reference point of the x-axis labelling. You can choose between Trigger Position and First Sample:

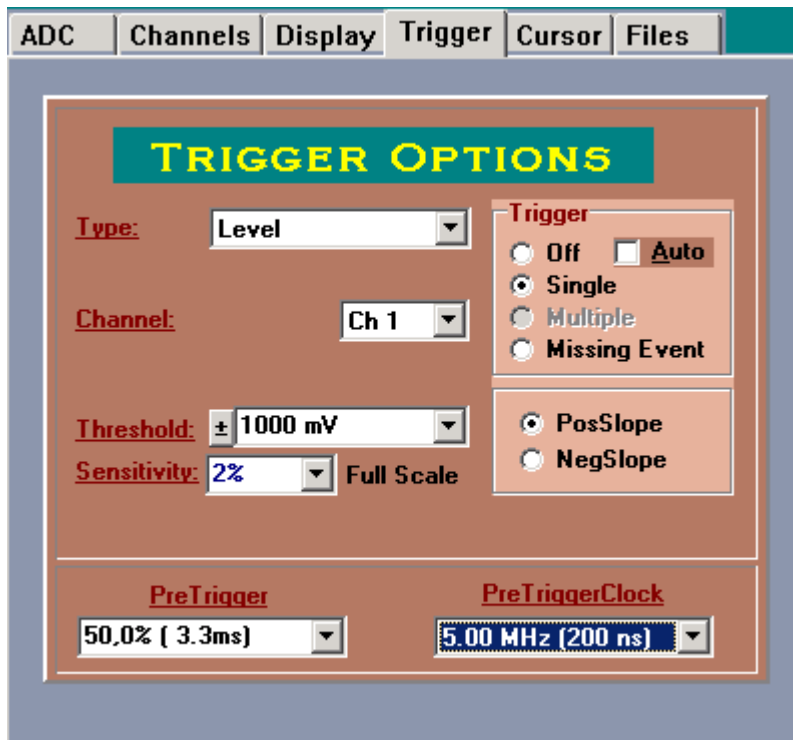
- Trigger Position:** The zero point of the x-axis is set to the trigger point. If a pre trigger is defined, this point is not equal to the beginning of the data.
- First Sample:** The zero point of the x-axis is the first data point.

Use view port: Selection of the data that is to be managed using LTTview.

- Left Border:** Number of samples per channel, that is to be ignored at the beginning of the measurement data
- Length:** Number of samples per channel beginning at “Left Border” to be handed over to LTTview

Note: If as reference point of the x-axis the trigger position is selected and the view port is defined such that the trigger moment is outside the view port, the zero position of the x-axis will be outside as well.

4.4 Trigger



Off: Turn trigger off

Single: When selecting this trigger option a single channel will be the trigger channel.

Type: Type of trigger

- **Level:** Threshold which should be surpassed or underpassed to start the measurement. The Trigger depends on the threshold, sensitivity (hysteresis) and slope. At a positive slope the sensitivity defines a lower threshold and at a negative slope an upper threshold which is to be underpassed / surpassed before a trigger event is being produced. This is helpful in the case of very noisy signals. The sensitivity is defined by percent of the full scale range. Therefore, at a range of 50 V a sensitivity of 2 % will be 1V.
- **Comparison:** Recording of the measurement will start if measured value is either greater or smaller than a prescribed threshold value. The comparison trigger therefore is a "level trigger" without defining a sensitivity or slope. The history of the signal curve is not important.
- **Region:** The recording of the signal will start when data has either entered or exited the region. The region is defined by the upper and lower limits and the enter or exit options.
- **Extrema in Region:** Using this option will trigger on extrema within a defined region of values. A delay time (noise suppressor) is specified to enable triggering on extrema smaller or greater than regular signals.

A window of values can be defined in which the maximum / minimum of an extrema must lie to trigger the measurement. To suppress noise a time frame (noise suppressor) may be specified that indicates, what time width a signal extrema (peak or dip) must at least have in order to start the recording.

- **Level+Delta:** This trigger equals a level trigger with an additional condition of a certain slope. Data will be taken if the subsequent data point after reaching the threshold level is greater than the absolute value of delta. This delta is defined in units of the channel.

Channel: Choose which channel or channels to be the source of the trigger. If the digital channel is being selected, the bit (7 to 15) of the external digital inputs may be defined. It is also possible to define that pressing the start button on the LTT system will start the recording.

The **Scan** option enables the selection of several trigger channels which may start the recording using an “Or” combination.

Threshold: Can be any value in the valid measurement range.

Sensitivity: Sometimes called hysteresis and may have values between 0,02% and 50% of the full scale range.

PosSlope / NegSlope: Sets the slope for the trigger event.

The **Multiple** trigger option can be checked to restart after a defined time using the option **Reset Trigger after**. The reset time between two triggers is few μ s.

Missing Event: The measurement will start, if after a successful trigger event no second trigger event is registered within the time set by **Delay Time**.

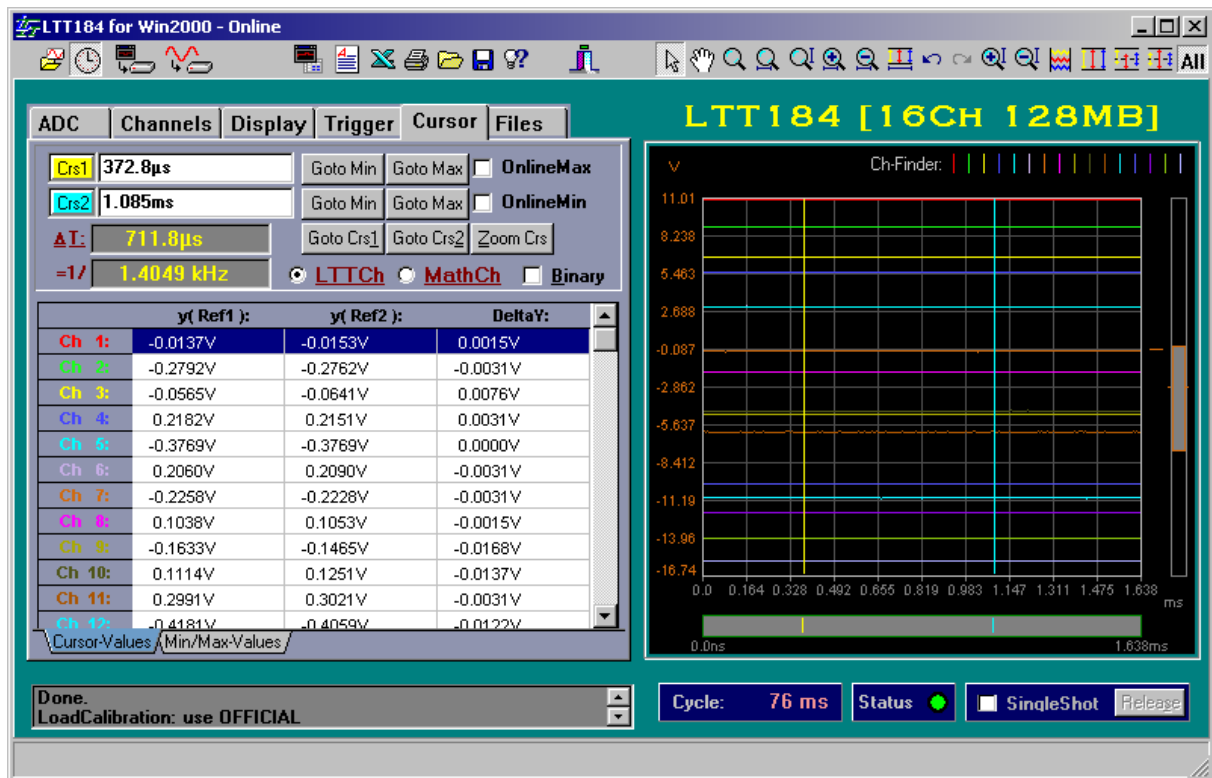
Important note: The Trigger will start only after the first trigger event has appeared. If a second trigger event is registered within the defined delay time, the timer starts over again. That way it is possible to trigger on a missing event.

The Missing Event Trigger may be used with the PreTrigger option.

PreTrigger: Data will be shown that occurred before the triggering event. The pretrigger depth is 120MB. The trigger event is scaled as a percentage of the display memory. Therefore it is possible to have a pretrigger greater than 100%.

PreTriggerClock: A-B Mode. Using a PreTrigger it is possible to have a different sample rate before the trigger event than the sample rate set on the ADC tab page (PostTrigger clock). As long as the maximum total sampling rate (# active channels times sample rate) is less than 100MS, a different PreTrigger clock may be selected.

4.5 Cursor



Crs1: Turns on/off the 1st cursor and sets the color with the right mouse button

GotoMin / GotoMax: Sets cursor 1 to the minimum or maximum of the currently selected channel on the screen. This selected channel is called the display channel, that is the channel, whose scaling is shown at the y-axis of the display. The display channel can be switched by various ways (please also see Chapter 5, Graphical Display.

1. Ch-Finder: In the right upper corner of the display there is a colored line for every channel. By right-clicking it with the mouse the display channel changes.
2. Y-Scrollbar: On the right of the display by right clicking the mouse a window will open, in which the display channel can be selected.
3. Tab sheet "Channels": By moving the mouse pointer over the Turn on/off buttons of the channels the display channel changes.
4. Tab sheet "Cursor": By moving the mouse pointer over the channel numbers on the "cursor value list" the display channel changes.

OnlineMax: Dynamically sets cursor 1 to the maximum of the display channel

Crs2: Turns on/off the 2nd cursor and sets the color with the right mouse button

GotoMin / GotoMax: Sets cursor 2 to the minimum or maximum of the display channel

OnlineMin: Dynamically sets cursor 2 to the minimum of the display channel

Goto Crs1: Adjusts the graph so that the 1st cursor is shown.

Goto Crs2: Adjusts the graph so that the 2nd cursor is shown.

Zoom Crs: Adjusts the graph so that the both cursors are shown.

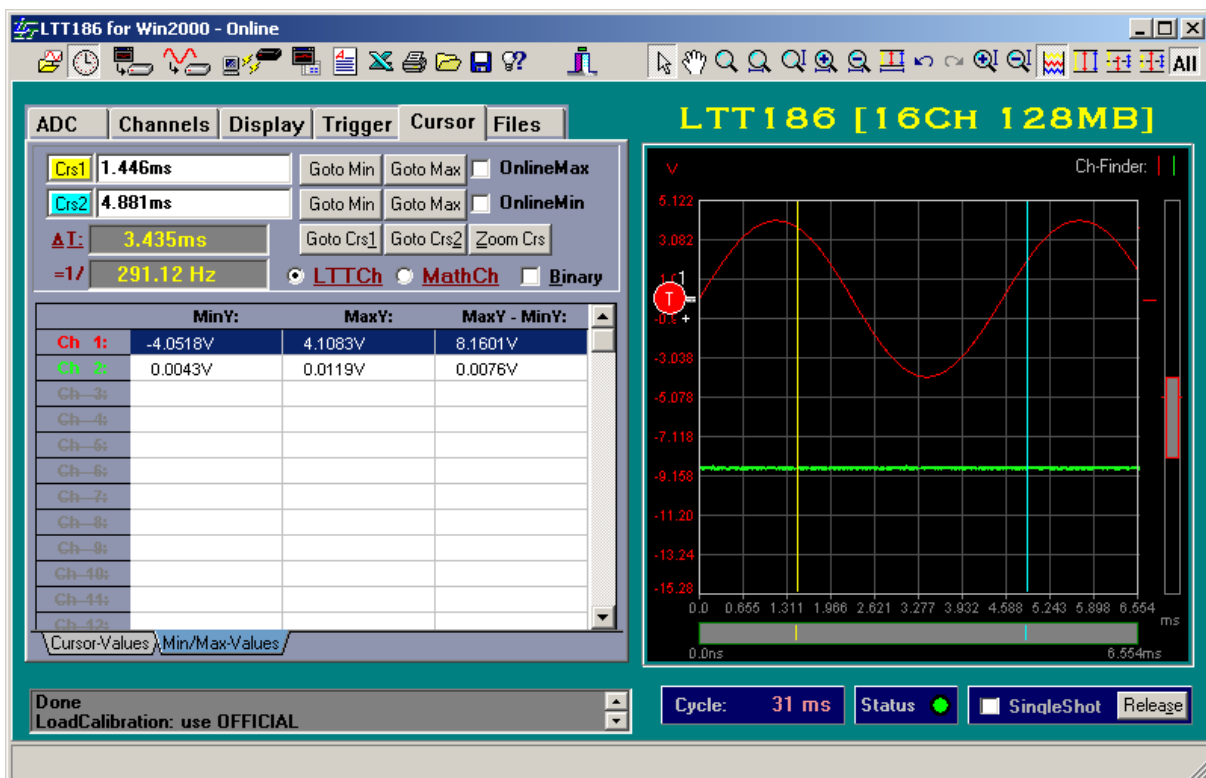
LTTCh / MathCh: Toggles the display in the table shown below between the analog and the math channels

Binary: Presents the data in 16 bit binary format.

Cursor-Values: The table Cursor-Values shows the y values for all channels at the positions of the two cursors.

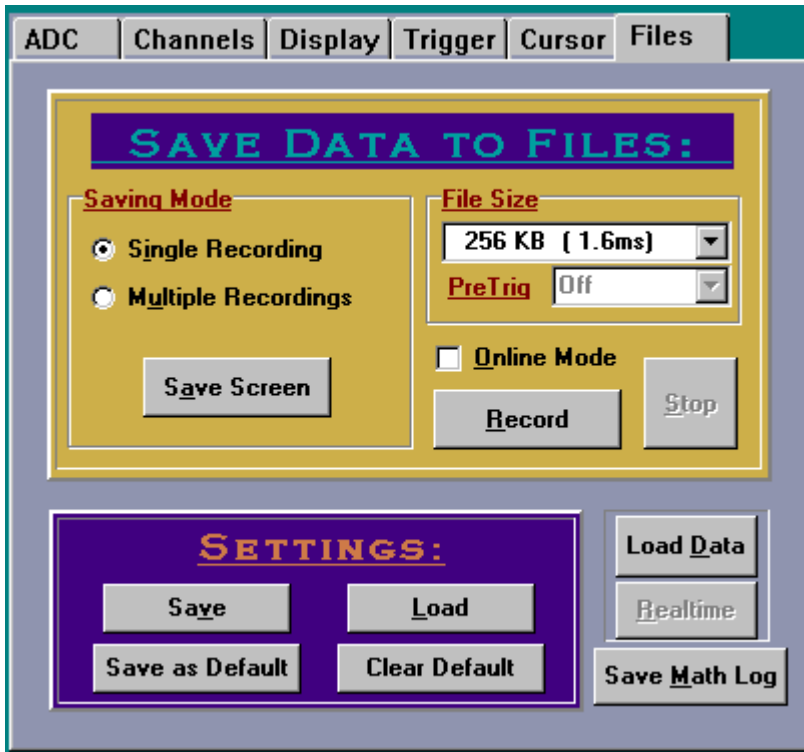
Note: The channels can be turned on and off by clicking on the 1st column in the grid with the left mouse button. Using the right mouse button the color can be changed.

Min/Max-Values: The table **Min/Max-Values** shows the minimum and maximum values for all active channels and their difference. This references the region as defined by the view port (see chapter 4.3 Display).



4.6 Files

Saving the data, configuring and starting measurements of different types and taking large amounts of data can be accomplished with this tab page.



Saving Mode: Two different types of measurements may be chosen:

Single Recording: The recording will be performed once.

Multiple Recordings: Multiple *triggered* recording will be carried out creating a file for each measurement. A trigger must be activated. *Note:* In this case the automatic reactivation of the trigger will not be performed by the LTT device (see tab sheet **Trigger, Multiple Trigger**), but by the software LTTview. LTTview stops and starts the recording on its own. Therefore the reset time will be significantly longer (~1 ms).

Save Screen: Saves the currently visible selection of the data on the screen to a file.

File Size: Definition of the file to be generated and saved:

File Size: Sets the maximum file size for the measurement. For information it is also shown how long the measurement will take to record. *Note:* If the data stream is too fast for the SCSI/FireWire/USB2 bus the measurement will be aborted. No data will be lost but the set file size will not be reached.

PreTrig: This sets the pretrigger for the file to be saved and may be different from the pretrigger specified in the tab sheet **Trigger**. Details there refer to the display memory. The maximum is 120 MB. The pretrigger time is adjusted automatically for the current sampling rate.

Online Mode: Allows the user to see the data as it comes and per default is off for optimum throughput.

Fast Save: Per default, for large file sizes a compressed file is simultaneously generated. Checking this option suppresses generation of this file allowing for higher throughput.

Record: Starts the measurement

In Offline mode is: **Export Screen:** Exports the data visible in the graph to a user defined format

Stop: Stops the measurement

Settings: All the changes that are made to LTTview can here be saved and loaded:

Save: Saves the settings to a specified directory.

Load: Loads the settings.

Save as Default: Saves the settings so that when LTTview is being restarted it will use the current settings.

Clear Default: Resets the default settings.

Load Data: Loads previously saved data.

Realtime: Switches from Offline to Online mode.

Save Math Log: Saves the log file for the applied mathematical functions in a chosen directory.

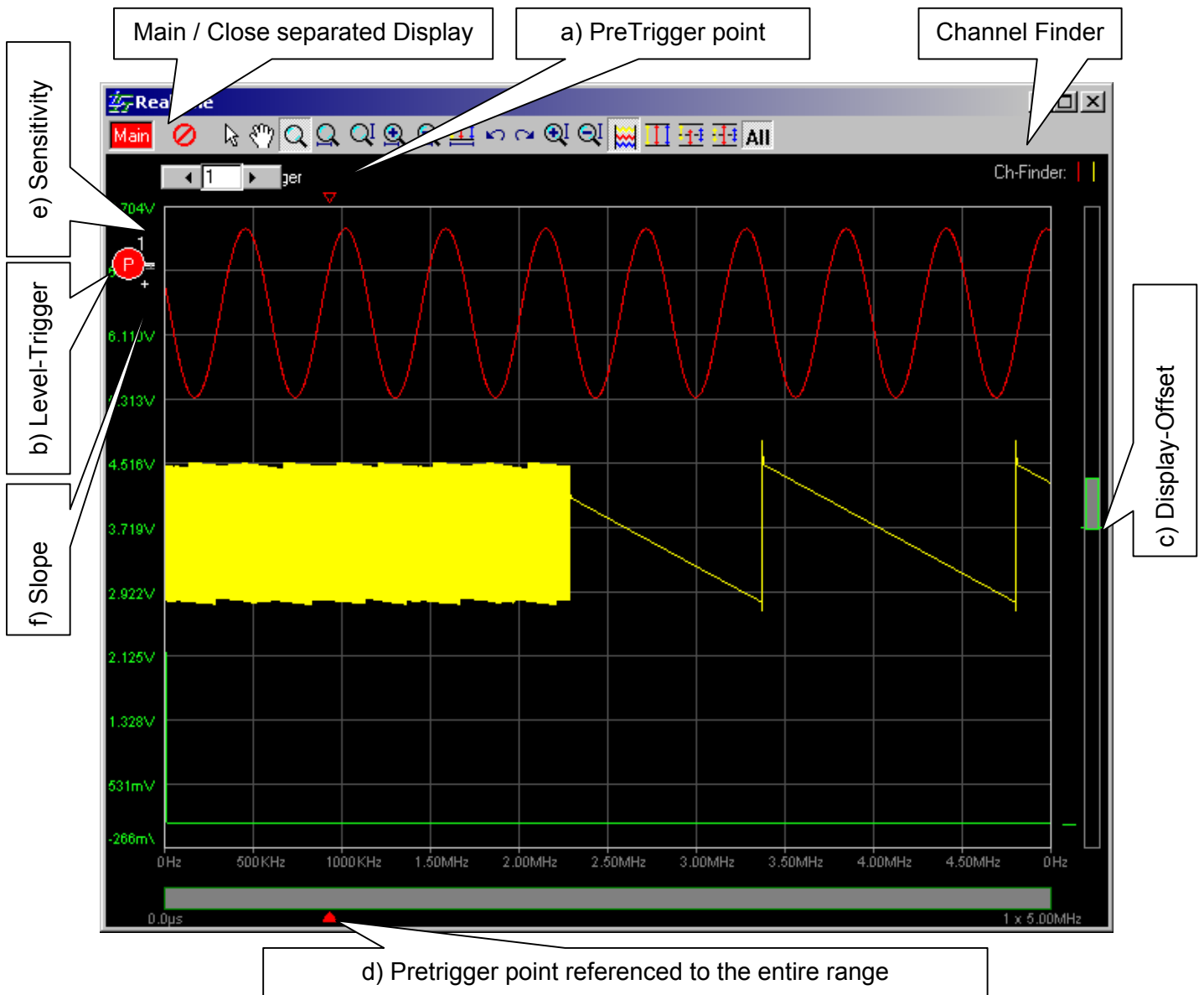
5. Graphical Display

5.1 Display functions

The graphical user interface of LTTview resembles the display of an oscilloscope. If you take a closer look at the display you will notice that the display hides various functions.

Shown below is the graphical display when separated from the tab page controls. Presented is the results of a Multiple Measurement using a 6.5 ms pretrigger and a FFT math channel in green. Not normally seen are the first two buttons in the Task Bar: **Main** and **Close Separated Display**.

Main will bring the tab page controls to the front and **Close Separated Display** will revert to the default operation with the Graphical Display docked to the tab page controls. The other icons in the Task Bar will be describe below.



Ch-Finder: The channel finder is a useful tool to display the selected signal curve on the display. Clicking one of the vertical lines found in the channel finder will put the corresponding channel on top, thereby becoming the *Display Channel (DC)*.

If a math channel is used, there will be two channel finders: One for the analog inputs and one for the math channels. Clicking the label "Channel Finder" will toggle between them.

The top, left, right and bottom regions around the graph as well as the graph itself can perform certain functions:

- Clicking along the top of the graph will set a pretrigger to the DC if possible.
- Clicking along the left y-axis will set the trigger threshold for the DC. A circled in "T" or "P" will appear (if also a PreTrigger was defined).
- Clicking along the right y-axis will set a "display offset" level referenced to the full scale. The gray slider is outlined with the same color as the DC and scaled to full scale as represented by the empty gray box.
- The grey bar at the bottom represents the whole recording length. The small arrow shows the PreTrigger point in relation to the whole recording. Clicking and/or dragging along the

bottom x-axis will move the slider or change the size of the slider and thereby the displayed data.














- e) Right of the trigger symbol you find the symbol representing the sensitivity. The symbol may be extended or shortened using the mouse which will change the sensitivity.
- f) A “+” or “-“ shows the positive or negative slope

Clicking with the right mouse button will bring a pop-up window to select various trigger options.

5.2 The Task Bar






Various functions may also be selected using the task bar. It is divided into basically 2 parts. The left side has general controls and the right side has icons pertaining to the graphical display as seen above in the separated display.

a) General Functions












	Open File: Loads saved data.
	Real Time: switches to the online operation
	Save to disk.
	Measure: start measurement to disk
	Configure Stand-Alone Operation (LTT-186 only)
	Separate the Graphical Display from the Tab Page controls
	Clipboard: Copies the Graphical Display to the Clipboard
	Exports to Excel. Note: Excel can handle only up to 30000 points
	Print: the Graphical Display with an information table.
	Load settings
	Save current settings
	Toggle the Help Mode. In Help Mode clicking the cursor will start explorer and bring up the appropriate information.
	Exit LTTview

b) Graphical Display Functions

Mouse assisted functions:

	Set mouse to cursor mode
	Hand Mode: The mouse can be used to optimize the xy-offsets
	Zoom in a selected xy region
	Zoom in the x region
	Zoom in the y region

Direct function:

	Every click will decrease tDiv by a factor of ½.
	Increases the tDiv by a factor of 2.
	Show x full scale. Maximum zoom out.
	Step back 1 in history.
	Step forward 1 in history.
	y-axis: decreases yDiv by a factor of ½ (zoom out)
	y-axis: increases yDiv by a factor of 2 (zoom out)
	Spreads the “active” & “show enabled” channels so that all channels can be seen.
	y-axis: “Zoom all of all Channels”: Optimizes the scaling such that the signals use the entire range.
	y-axis: “Zoom Range of all Channels”: Sets the range to full scale.
	y-axis: “Center all channels”: Centers the signals in the display.
All	If “All” is activated then the y-axis functions will operate on all <i>shown</i> channels.

5.2 Additional Information

By default the following information is found under the Graphical Display:

Cycle: Duration of a completed measurement including graphing the data.

Status: Green – data has been received
Red – waiting for a trigger event
Black – LTTview is hung

Single Shot: Holds the real time operation. Pushing **Release** will advance by one measurement.

Chapter 4:

Description of the DLL LTT2API

To integrate LTT measurement systems with third party applications the LTT2API-DLL was developed. Using the DLL opens a wide range of possibilities for efficient use of the LTT-Hardware. This chapter describes the programming interface **LTT2API Version 1.3.5**.

Every function is described using the following format:

<Function Name>

[Device]

Declaration:

<Declaration of the function>

Description:

<Description of the function and explanation of the use of the function>

Input:

<specifies and describes the function parameters>

Return:

<specifies the return value>

Note: The note [Device] in the 2nd line after the <Function Name> indicates a direct communication with the LTT-System.

Fundamental concepts

The API is designed to manage one or a group of LTT devices in an efficient manner. The most important design goals are stability and performance. The calls are described in STDCALL calling convention. It's also possible to access the API-calls with CDECL calling convention. You must add only CDECL to every function call:

```
__stdcall LTTInit() == > __cdecl LTTInitCDECL()  
...
```

The use of the API is divided in several steps:

Initialization:

The call **LTTInit** must be the first call to the API. With this call the internal resources are initialised and the system is scanned for connected LTT devices. Note: This function should be used **only once** at the beginning.

Setup:

The devices can now be configured with a certain setup, as turn on several channels, select its input type and range or select a trigger. Also the amount of data has to be specified. These calls don't communicate directly with the devices. Instead, all setup calls build up a certain configuration which is passed to the devices with **LTTUploadParameters**. This call does an integrity check to see if the configuration is a valid one.

Transfer:

The devices will be started with **LTTStart**, which basically means that the trigger will be active or in the case of no trigger, the measurement process of the devices begins.

Now the transfer of the data from the devices to the RAM-buffer/File can be done in two ways. In both cases the transfer is invoked with **LTTTransfer2Buffers**.

BLOCKED_MODE:

The call **LTTTransfer2Buffers** returns when the transfer of the data to the RAM-buffer/File is completed. This is the easiest way, and the most used one.

NONE_BLOCKED_MODE:

The call **LTTTransfer2Buffers** returns **immediately** after invoking the transfer. Now the transfer to the RAM-buffer/File needs the cooperation of the application. The transfer must be monitored with **LTTTransferStatus** to catch when the transfer is completed or interrupted.

In STREAM-mode the internally used buffer is mostly smaller than the requested transfer amount of data, so this buffer is used like a ring-buffer. To avoid the overwriting of already in the buffer transferred data, the application must read out blocks of already transferred data. If the call **LTTTransferStatus** signals that there is an amount of data ready for read-out, then the start index and the length of the block inside the buffer must be obtained with **LTTData_GetBlock**. When the processing of the block is finished, **LTTData_BlockDone** must be called. Now go back to monitoring the transfer with **LTTTransferStatus**.

Deinitialization:

To close all internal resources, call **LTTDeInit**.

Other calls:

The API provides a number of different informative calls. They are not necessary for the data transfer, but quite a lot of information about the status and the devices can be obtained.

Source Code samples

In the following a source code sample for each of the transfer modes is provided. Both sample are written in plain C:

BLOCKED_MODE:

```
unsigned short usMaxAn, usMaxDig, usMaxRPM, usMaxMath;
short sBuffer[128*1024];
long lAmount;

...
LTTInit( &usMaxAn, &usMaxDig, &usMaxRPM, &usMaxMath );

...
// channel 0 on, +/-10V range, single ended DC+
LTTAnalogChannel( 0, 1, 10000, 1 );
// channel 3 on, +/-2V range, diff. ended DC
LTTAnalogChannel( 3, 1, 2000, 3 );

// trigger at ch 0, LEVEL, 2V-threshold,
// pos. slope, 200mV sensitivity
LTTTrigger( 1, 0, 0, 0, 0, 2000, 200 );

// sample rate: 1,04 Mhz <==> 960 ns
LTTSampleTime( 960, 0 );

// RAMBuffer: 32kS per channel ==> 2 * 32 * 1024 * 2 = 128 kB
LTTRAMBufferSetting( &sBuffer, 32 * 1024, 2 );

// submit to device(s)
if( LTTUploadParameters() == 0 ) {
    // start LTT device(s)
    LTTStart();

    // invoke transfer
    lAmount = LTTTransfer2Buffers( 0, 0 );
}

...
// clean up
LTTDeInit();
```

NON_BLOCKED_MODE:

```
unsigned short usMaxAn, usMaxDig, usMaxRPM, usMaxMath;
unsigned long ulStartIdx, ulLength;
short sBuffer[128*1024];
long lStatus;

...
LTTInit( &usMaxAn, &usMaxDig, &usMaxRPM, &usMaxMath );

...
// channel 5 on, +/-100V range, 1MOhm, SE DC+
LTTAnalogChannel( 5, 1, 100000, 7 );
// channel 10 on, +/-5V range, SE DC-
LTTAnalogChannel( 10, 1, 5000, 1 );
// digital channel
LTTDigitalChannel( 0, 1 );

// trigger at ch 5, REGION, 70V upper threshold
// -40V lower threshold, exit region
LTTTrigger( 1, 1, 5, 0, 2, 70000, -40000 );

// sample rate: 1,04 Mhz <==> 960 ns
LTTSampleTime( 960, 0 );

// STREAMuffer: 1MS per channel, buffer 64kS
LTTSTREAMSettings( &sBuffer, 64 * 1024, 1024 * 1024, 3 );

// submit to device(s)
if( LTTUploadParameters() == 0 ) {
    // start LTT device(s)
    LTTStart();

    // invoke transfer
    LTTTransfer2Buffers( 0, 0 );

    // monitor transfer
    lStatus = 0;
    while( lStatus <= 0 ) {
        lStatus = LTTDataTransferStatus();

        if( lStatus == -255 )
            // error occurred
            break;

        else if( lStatus == -1 )
            // to something
            DoSomething();

        else if( lStatus == -2 || lStatus > 0 ) {
            LTTData_GetBlock( &ulStartIdx, &ulLength );

            // do something with data or
            // simply call done!
        }
    }
}
```

```
        LTTData_BlockDone();
    }
}

...
// clean up
LTTDeInit();
```

LTTInit

[Device]

Declaration:

```
long __stdcall LTTInit(  
    unsigned short *pNumCh,  
    unsigned short *pNumDigCh,  
    unsigned short *pNumRPM,  
    unsigned short *pNumMathCh  
);
```

Description:

Initializes internal resources and searches for all connected LTT devices.

Input:

pNumCh
number of available analog channels

pNumDigCh
number of available digital channels

pNumRPM
number of available RPM channels

pNumMathCh
number of available math channels

Return:

Number of LTT devices found.

LTTDeInit

□

Declaration:

```
void __stdcall LTTDeInit( void );
```

Description:

Shuts down internal systems and frees resources.

Input:

Return:

LTTResetParameter

[Device]

Declaration:

```
long __stdcall LTTResetParameter( long device_reset );
```

Description:

Resets API-Parameter internally to default setup.

Input:

device_reset
option to reset the devices
0 – not passed; only API is reset
>0 – additionally resets devices;

Return:

LTTTrigger



Declaration:

```
long __stdcall LTT Trigger(  
    unsigned short Trigger,  
    unsigned short Slope,  
    unsigned short Channel,  
    unsigned short ChType,  
    unsigned short TrigType,  
    long x1,  
    long x2  
);
```

Description:

Enables or disables the trigger channel and sets trigger parameters.

CAUTION:

The channels which will be used as trigger channels must be turned on!

Input:

Trig

- 0 - trigger OFF
- 1 - trigger ON

Slope

Level, Comparison + Delta:

- 0 - positive slope
- 1 - negative slope

Comparison:

- 0 - bigger then threshold
- 1 - smaller then threshold

Region:

- 0 - enter region
- 1 - exit region

Channel

set trigger channel (index begins with 0)

ChType

- 0 - analog channel
- 1 - digital channel
 - firmware 6xxx
 - x1 - digital trigger bit [8..15]
 - firmware 7xxx
 - x1 - digital trigger bit [10..15]

CAUTION:

only the specified bits are allowed !!!

TrigType

- | | |
|-----------|---------------------|
| analog | digital |
| 0 – Level | 0 - single bit mode |

1 – Comparison
2 - Region
4 - Level + Delta

1 - multi bit mode

x1

Level, Comparison, Comparison + Delta:
Threshold [mV]
Region:
Lower Threshold [mV]

x2

Level:
Sensitivity [mV]
Comparison:
not used
Region:
Upper Threshold [mV]
Comparison + Delta:
minimal difference to previous sample [mV]

Return:

0 if successful.
-1 if channel doesn't exist.
-2 level trigger (TrigType=0) can't have negative hysteresis.
-4 level+delta trigger (TrigType=4) invalid range for x2 ($0 < x2$).
-5 invalid digital trigger channel (ChType=1) .

LTTTrigger_SetPreTrigger



Declaration:

```
long __stdcall LTTTrigger_SetPreTrigger (
    unsigned long dwpretrigger
    unsigned long dwPreSampleTimeSlow
);
```

Description:

Sets the pretrigger length per channel. Units are in kilo samples.

For example, desired is a 500ms pretrigger using a 400ns sampling rate. The pretrigger length is calculated.

$$500\ 000\ 000\ \text{ns} \ (\ 1\ \text{Sample} / 400\ \text{ns}) = 125\ 000\ \text{samples}$$

$$125\ 000\ \text{samples} / (1024\ \text{samples}) = 122.070$$

Must be integer number so either 122 (499.7 ms) or 123 (503.8ms) are valid

pretrigger is only allowed under certain conditions. Per devices only one or a pair number of active channels are allowed.

The pretrigger length is limited to 60 Megasamples (120*1024*1024 byte).

So with 6 active channels the maximum pretrigger per channel is 10 Megasamples.

Input:

dwpretrigger
length of pretrigger as multiples of kilo-sample blocks

dwPreSampleTimeSlow
sample time in [ns] for pretrigger phase
can be zero (pretrigger clock equals posttrigger clock)

Return:

0 OK
-1 No trigger active
-2 Wrong pretrigger length

LTTTrigger_SetReactivationTime



Declaration:

```
long __stdcall LTTTrigger_SetReactivationTime(  
    unsigned long dwTime  
);
```

Description:

Sets reactivation time for Trigger. Typical dead-time for reactivation <5 μ s. Units are in [ns]. The timer resolution is in steps of 100ns. The valid range depends on the used sample time:

0	- Reactivation timer OFF
4 * Sampletime...1s	- Valid reactivation time

CAUTION:

This option can't be activated together with LTTTrigger_Setpretrigger!! Both are mutually exclusive.

Input:

dwTime
 reactivation time in [ns]

Return:

0	OK
-1	No trigger active
-2	Reactivation time too big

LTTTrigger_SetABAMode

□

Declaration:

```
long __stdcall LTTTrigger_SetABAMode(  
    unsigned long dwTime  
    unsigned long dwABASampleTimeSlow  
);
```

Description:

Sets trigger in ABA.Mode

CAUTION:

This function can't be used with LTTTrigger_Setpretrigger!!
They are mutually exclusive.

Input:

dwTime

Time after which the trigger
changes back to pretrigger speed
allowed values:

0 - No ABA mode
100ns..1s - allowed range

dwABASampleTimeSlow

Sample time in [ns] before
trigger has happened and to
which is switched back after
timer has expired.

CAUTION:

Not all times are allowed.

Return:

0 OK
-1 No trigger active
-2 Reactivation time too big

LTTAnalogChannel



Declaration:

```
long __stdcall LTTAnalogChannel(  
    unsigned short Channel,  
    unsigned short OnOff,  
    unsigned long chRange,  
    unsigned short AmpType  
);
```

Description:

Turns on/off analogs channels and apply the channel settings.

CAUTION:

The 16 Channel devices have some hardware constraints regarding the possible channel settings. Due to the internal design it's not possible to turn on a backside channel without at least one active front channel. The RPM channel counts as back channel and the digital channel as front channel:

0, 9, D1	valid
12, D1	valid
7, RPM	valid

but

14, 15	not valid
RPM	not valid

Input:

Channel
channel ID

OnOff
0 - channel OFF
1 - channel ON

chRange
channel range in [mV]
Allowed settings:
[1000, 2000, 2500, 5000,
10000, 20000, 25000, 50000,
100000, 200000, 250000, 500000]

AmpType

- 0 - GND
- 1 - DC, Single ended +
- 2 - DC, Single ended -
- 3 - DC, Differential ended

- 4 - AC, Single ended +
- 5 - AC, Single ended -
- 6 - AC, Differential ended

- 7 - 1MOhm, DC, Single ended +
- 8 - 1MOhm, DC, Single ended -
- 9 - 1MOhm, DC, Differential ended

Return:

- 0 if successful
- 1 if channel doesn't exist
- 2 invalid AmpType
- 3 AmpType doesn't match LTT-Hardware
- 4 Channel range doesn't match AmpType settings

LTTDigitalChannel

□

Declaration:

```
long __stdcall LTTDigitalChannel(  
    unsigned short Channel,  
    unsigned short OnOff  
);
```

Description:

Turns on/off digital channels.

CAUTION:

See `LTTAnalogChannel` for details about valis channel settings!

Input:

Channel
channel ID, which to be modified.

OnOff
0 - channel OFF
1 - channel ON

Return:

0 if successful
-1 if channel doesn't exist.

LTTTRPMChannel

□

Declaration:

```
long __stdcall LTTTRPMChannel(  
    unsigned short Channel,  
    unsigned short OnOff  
);
```

Description:

Turns on/off RPM channels.

CAUTION:

Only the RPM-channel at the master device per device stack is supported.
See `LTTAnalogChannel` for details about valid channel settings!

Input:

Channel

channel ID, which to be modified.
Must be 0 !!

OnOff

0 - channel OFF
1 - channel ON

Return:

0 if successful
-1 if channel doesn't exist.

LTTAnalogAutoZero

[Device]

Declaration:

```
long __stdcall LTTAnalogAutoZero(  
    unsigned long chnl_cnt,  
    unsigned long *p_chnl_list  
);
```

Description:

Automatic ReZero of GND-Level for selected active channels in current range.

Input:

chnl_cnt
length of channel-list

p_chnl_list
pointer to channel-list for rezero'ing

Return:

- 0 - successful
- 1 - no LTT devices present
- 2 - invalid channel-list pointer
- 3 - invalid channel in channel-list
- 4 - trying to rezero non-active channel
- 5 - upload-error for rezero; current setup is not valid
- 6 - error in data transfer

LTTAnalogAutoZeroClear

[Device]

Declaration:

```
long __stdcall LTTAnalogAutoZeroClear(  
    unsigned long chnl_cnt,  
    unsigned long *p_chnl_list  
);
```

Description:

Clear's AutoZero offsets for selected channels in current range.

Input:

chnl_cnt
length of channel-list

p_chnl_list
pointer to channel-list for rezero'ing

Return:

0 - successful
-1 - no LTT devices present
-2 - invalid channel-list pointer
-3 - invalid channel in channel-list

LTTOutputImpedance

□

Declaration:

```
long __stdcall LTTOutputImpedance(  
    unsigned short wAnalogChannel,  
    unsigned short wConnectionType  
    unsigned long dwImpedance  
);
```

Description:

Sets output impedance of signal source for specified input coupling.

Input:

wAnalogChannel
Analog channel ID

wConnectionType
type of input coupling
0 – SE+ coupling
1 – SE- coupling
2 - DE coupling

dwImpedance
impedance value

Return:

0 if successful
-1 if channel doesn't exist.
-2 invalid connection type.

LTTSampleTime

□

Declaration:

```
long __stdcall LTTSampleTime(  
    unsigned long SampTime,  
    short TimeFlag  
);
```

Description:

Sets the sampling time.

Input:

SampTime

Sample time in nano-seconds.

CAUTION: not all times are allowed!

TimeFlag

If sample time is between two allowed sample times, this parameter indicates, which of both is chosen for legal sample time. Chooses the sampling time below (1) or above (0)

Return:

The actual sampling time used.

LTTGetRAMBufferSettings

□

Declaration:

```
long __stdcall LTTGetRAMBufferSettings(  
    unsigned long dwSamplesPerChannel ,  
    unsigned long dwChannels  
);
```

Description:

Gives back needed buffer size in bytes.

This is a helper function to calculate the required buffer size for a certain channel settings and data amount.

Input:

dwSamplesPerChannel
 Number samples per channels

dwChannels
 Number of channels

Return:

The size of the buffer in bytes.

LTTRAMBufferSettings

□

Declaration:

```
long __stdcall LTTRAMBufferSettings(  
    short *pwBuffer,  
    unsigned long dwSamplesPerChannel,  
    unsigned long dwChannels  
);
```

Description:

Setting up the RAM buffer system for data transfer.

The buffer must be allocated by the application. The size must match the channel settings and data amount.

Input:

pwBuffer
 Pointer to RAM buffer.

dwSamplesPerChannel
 Number of samples per channel.

dwChannels
 Number of channels.

Return:

0 if successful
-1 STREAM already set up !!

 RAM and STREAM are exclusive buffer settings.
-2 length per channel is invalid

LTTFileBufferSettings



Declaration:

```
long __stdcall LTTFileBufferSettings(  
    char *pcFileName,  
    unsigned long dwMode,  
    unsigned long dwMaxCount,  
    unsigned long dwSamplesPerChannel,  
    unsigned long dwChannels  
);
```

Description:

Sets the parameters to measure to a file.

Input:

pcFileName
chooses which channel to be modified

dwMode
Reserved! Always 0.

dwMaxCount
Reserved! Always 0.

dwSamplesPerChannel
Number of samples per channels

dwChannels
Number of channels

Return:

0	success
-1	file extension not specified
-2	wrong mode
-3	file size too large
-4	unknown file extension

LTTSTREAMSettings



Declaration:

```
long __stdcall LTTSTREAMSettings(  
    short *pwBuffer,  
    unsigned long dwBufferSize,  
    unsigned long dwSamplesPerChannel,  
    unsigned long dwChannels  
);
```

Description:

Setting up STREAM buffer system.

In contrary to RAM buffer system, it's possible to transfer data much bigger than the specified buffer size. It gives more control on the transfer to the application, but also is not so easy to work with. It requires the cooperation of the application during the transfer.

Input:

pwBuffer
Pointer to RAM-Buffer.

dwBufferSize
Buffer size in samples

dwSamplesPerChannel
Number of samples per channel

dwChannels
Number of channels

Return:

0 everthing's OK
-1 ERROR; sample per channel too small
-2 ERROR; buffer size too small
-3 ERROR; buffer size not multiple of channel sequence size
-4 ERROR; RAM buffer already set

LTTBufferReset

□

Declaration:

```
void __stdcall LTTBufferReset( void );
```

Description:

Resets the transfer buffers systems.

Input:

Return:

LTTUploadParameters

[Device]

Declaration:

```
long __stdcall LTTUploadParameters( void );
```

Description:

Checks and sends the parameters to the LTT devices.

The parameters have been gathered using LTTAnalogChannel, LTTSampleTime, LTTRAMBufferSetting, etc. but the device has not yet been configured.

LTTUploadParameters makes a final check to see if the combination of all the parameters don't make an error and then if successful configures the devices. The error codes are additive.

Input:

Return:

The return value is a sum of the following error codes.

- 0 Everything OK
- 1 LTT device not found or ASPI dll not found
- 2 Wrong sample rate by current channel setting
- 4 No channels active
- 10 Wrong channel combination!! When backside channels used, at least one front channel must be active also!
- 20 Trigger channel not active
- 40 pretrigger settings wrong - no channel set
- 100 pretrigger settings wrong - invalid pretrigger length
- 200 No Buffer settings specified!!
- 400 amount of active channels does not fit channel parameter in buffer settings.
- 1000 Math error
- 2000 No active channel for master device.
- 4000 - trigger timer and PRE-trigger are mutually exclusive settings.
- 10000 - trigger timer too small;
condition: timer > 4 * sampletime

LTTResetParameter



Declaration:

```
void __stdcall LTTResetParameter( long device_reset );
```

Description:

Resets all parameter settings in LTT2API to default. When device_reset is 0 no upload to the device(s) occurs and only the dll parameters are affected. If, however, device_reset equals 1 the channels are set to internal ground, that is OFF.

Input:

device_reset : 1 or 0.

Return:

LTTStart

[Device]

Declaration:

```
long __stdcall LTTStart( void );
```

Description:

Starts the LTT devices and activates the trigger if on.

Input:

Return:

0 Always

LTTStop

□

Declaration:

```
void __stdcall LTTStop( void );
```

Description:

Stops the measurement at LTT devices.

Input:

Return:

LTTCheckTrigger

□

Declaration:

```
long __stdcall LTTCheckTrigger(  
    unsigned long MilliSecond  
);
```

Description:

Checks if the trigger has been fulfilled.

In the case the trigger is still active, the call waits the specified amount of time before the call returns.

Input:

MilliSecond
time in milliseconds to wait after a negative trigger result.

Return:

0	trigger not fulfilled
1	trigger successful

LTTTransfer2Buffers

[Device]

Declaration:

```
long __stdcall LTTTransfer2Buffers(  
    unsigned short BufferType,  
    unsigned short BlockMode  
);
```

Description:

Transfers the data from the recorder to the assigned buffers. This call initiates the real transfer. The transfer can work in BLOCKED or NONE_BLOCKED mode.

BLOCKED:

The call returns, when the transfer is done.
This is not possible with the STREAM buffer system.

NONE_BLOCKED:

The transfer is initiated and the call returns immediately.
The Transfer must be monitored with LTTDataTransferStatus.

Input:

BufferType
Reserved; Always 0.

BlockMode
0 - (Blocked Mode)
1 - (NON-Blocked Mode)

Return:

-1 No LTT devices.
-2 No buffer type set.
-3 Transfer is active, so can't invoke another transfer !!
-4 File transfer already done. Reconfigure file buffer settings!

BLOCKED mode:

>=0 Number of transferred samples

CAUTION:

The amount of transferred samples must be checked against the requested data length. If the amount of transferred samples is less than the requested amount, an irrecoverable error has occurred during the transfer, which led to a break of the transfer. In the data buffer the correctly transferred data are stored up to the point where the error has occurred!!

NONE_BLOCKED mode:

⓪transfer

invoked

LTTDataTransferStatus



Declaration:

```
long __stdcall LTTDataTransferStatus( void );
```

Description:

Monitors the status of the transfer.

This function is necessary for the **NONE_BLOCKED** Mode of transfer.

Input:

Return:

- 0 nothing TODO
- 1 waiting for trigger
- 2 new block available
- 255 error happened

- >0 transfer complete, number of transmitted samples

CAUTION:

The amount of transferred samples must be checked against the requested data length. If the amount of transferred samples is less than the requested amount, an irrecoverable error has occurred during the transfer, which led to a break of the transfer. In the data buffer the correctly transferred data are stored up to the point where the error has occurred!!

LTTDataTransferBreak

[Device]

Declaration:

```
long __stdcall LTTDataTransferBreak( void );
```

Description:

Breaks a running transfer.

Input:

Return:

0	OK
-1	error occurred

LTTData_GetCounter



Declaration:

```
long __stdcall LTTData_GetCounter( void );
```

Description:

Gives back the amount of transferred bytes.

This call can be used in **NONE_BLOCKED** Mode of transfer to monitor the progress of the transfer.

Input:

Return:

amount of up to now transferred samples

LTTData_GetBlock

□

Declaration:

```
long __stdcall LTTData_GetBlock(  
    unsigned long *dwStartIndex,  
    unsigned long *dwAmount  
);
```

Description:

Gives back a data block, which can safely be read by the application.

CAUTION:

Only used in **NONE_BLOCKED** mode of transfer. Transfer must be monitored with LTTTransferStatus, which indicates if a data block is ready for read-out.

Input:

dwStartIndex

Gives back the start index of the block of data.

dwAmount

Gives back the total number of sample available in the block.

Return:

0	if no LTT devices found
1	if successful

LTTData_BlockDone



Declaration:

```
long __stdcall LTTData_BlockDone( void );
```

Description:

Tells LTTAPI that the block obtained with the call LTTData-GetBlock is not anymore used by the application.

Input:

Return:

ⓈAlways

LTTGetCalibTemperature

□

Declaration:

```
long __stdcall LTTGetCalibTemperature(  
    unsigned long dwDevice  
);
```

Description:

Gives back internal temperature of specified device at last calibration.

Input:

dwDevice

Device ID for which the calibration temperature is requested

Return:

Internal devices temperature in degrees C

LTTGetChannelIDSequence

□

Declaration:

```
void __stdcall LTTGetChannelIDSequence(  
    unsigned short *pSequenceList  
);
```

Description:

Gives back channel ID in channel sequence.

Input:

pSequenceList
pointer to list. Application must provide sufficient memory !!

Return:

LTTGetChannelSequenceLength

□

Declaration:

```
long __stdcall LTTGetChannelSequenceLength( void );
```

Description:

Gives back length of channel sequence.

Input:

Return:

Length of channel sequence.

LTTGetResolution

□

Declaration:

```
long __stdcall LTTGetResolution( void );
```

Description:

Gets the current bit resolution for the sampling time.

Input:

Return:

- 12 12Bit resolution;
- 14 14Bit resolution;
- Ⓣ 15Bit resolution;
- Ⓣ. 16Bit resolution;

LTTGetChannelRange

□

Declaration:

```
long __stdcall LTTGetChannelRange(  
    unsigned long dwAnalogChannel  
);
```

Description:

gives back range for the requested channel.

Input:

dwAnalogChannel
Analog channel index

Return:

maximum range; +/- value in [mV]

LTTGetTemperature

[Device]

Declaration:

```
long __stdcall LTTGetTemperature(  
    unsigned short wDevice  
);
```

Description:

Measure temperature in specified device.

Input:

wDevice

Device ID for which the calibration temperature is requested

Return:

current internal devices temperature

LTTGetDLLInfo

□

Declaration:

```
long __stdcall LTTGetDLLInfo(  
    char *pcBuffer  
);
```

Description:

Obtain identification string of the DLL.

Input:

pcBuffer

pointer to buffer where the null terminated string will be put.
Must be at least 64 bytes.

Return:

Ⓢ Always

LTTGetDLLVersion

□

Declaration:

```
long __stdcall LTTGetDLLVersion( void );
```

Description:

Obtain version number of the DLL.

The version number is build in the following manner:

```
( LTT2API_MAJOR << 16 ) | ( LTT2API_MINOR << 8 ) | LTT2API_BUG
```

Input:

Return:

Version number

LTTGetDeviceInfo



Declaration:

```
long __stdcall LTTGetDeviceInfo(  
    unsigned long dwDevice,  
    char *pcBuffer  
);
```

Description:

Obtain identification string of the LTT device.

Input:

dwDevice

Selects the LTT device.

pcBuffer

pointer to buffer where the null terminated string will be put.
Must be at least 64 bytes.

Example:

```
LTT-184/16 SN:3214 FW:6.0.607 ch:( 1 - 16)
```

Return:

0 if successfull
-1 if LTT device not found.

LTTGetDevices

□

Declaration:

```
long __stdcall LTTGetDevices( void );
```

Description:

Returns the number of devices found.

Input:

Return:

number of devices found.

LTTSetInternReference

□

Declaration:

```
long __stdcall LTTSetInternReference(  
    unsigned long dwDevice,  
    unsigned long dwRefIdx  
);
```

Description:

Sets all the channels for a given device to the internal reference.

Input:

dwDevice

Selects the LTT device.

dwRefIdx

index of reference value table

0 <=> GND

1 <=> 857mV

2 <=> 3.00V

3 <=> V_{temp}

4 <=> $V_{temp} - 3.00V$

5 <=> $3.00V - 3.00V$

6 <=> 0 - 857mV

Return:

0 OK

-1 invalid device index

-2 invalid reference index

LTTGetChannelConverter

□

Declaration:

```
double __stdcall LTTGetChannelConverter(  
    unsigned long dwAnalogChannel  
);
```

Description:

Gives back conversion factor for requested analog channel.
Applying this factor to a 16-Bit data sample value will convert the value in units of [mV]:

$$[\text{mV}] = [16\text{Bit}] * \text{LTTGetChannelConverter}([\text{channel}])$$

Input:

dwAnalogChannel
Analog channel index

Return:

conversion factor

Index

19" – Rack system	10	Missing Event	31
ADC	17	New Boot Virus	15
Analog Out	7	Online and Offline	16
Analog to Digital Converter	17	Online Mode	35
Analyzer	26	Overview	25
Auto Scaling	21	Persistence	27
Average Screens	24	Power connection	9
Cascading LTT-Systems	13	PreTrig	34
Channels	19	PreTrigger	31, 36
Check Calib	18	PreTriggerClock	31
Ch-Finder	36	Print out	26
Close Separated Display	35	Pulse Width	24
Comparison Trigger	30	Range	22
Cursor	32	Realtime	35
Cursor-Values	33	Record	34, 35
Cycle	39	Region Trigger	30
Delay Data transfer	18	Reset	9
Differential Ended	20	Rotation	28
Digital Input	8, 19, 25	RPM (Rotation per Minute)	19
Display	27, 35	Sample Rate	17
Display Channel (DC)	36	Save Math Log	35
Drivers	15	Saving Mode	34
Eff Value	24	Scale	21, 29
External frequency	17, 19	SCSI-Connection	9
Extrema in Region	30	Sensitivity	31
FFT	24	Set Preset Directory	28
Files	34	Settings	35
Filter	24	Signal Source Impedance	18
Fine Adjust	17	Single Ended	20
FireWire	15	Single Shot	39
Frequency	17	Slave	13
Graphical User Interface	16	Slope	31
Grid	27	Start Button	7
Ground	9	Status	39
Histogram	24	Synchronisation	13
Impedance	20	Task Description	26
Input coupling	20	Technical Specifications	12
Inputs	7	Temperature	18
Installation of LTTview	15	Terminated	14
Integral	24	Threshold	31
LED	7	Total sampling rate	17
Level Trigger	30	Total Transferred Bytes	18
Load	35	Trigger	30
LTT Duty Cycle	19	Unit	22
LTTDATA	15	Vertical offset	22
LTTview	15	View Port	28
Main	35	Work Offline	16
Manual Scaling	22	x-Axis	28
Master	13	yDiv	22
Math	23	yOffs	22
Math channels	23	Zero Position	29

