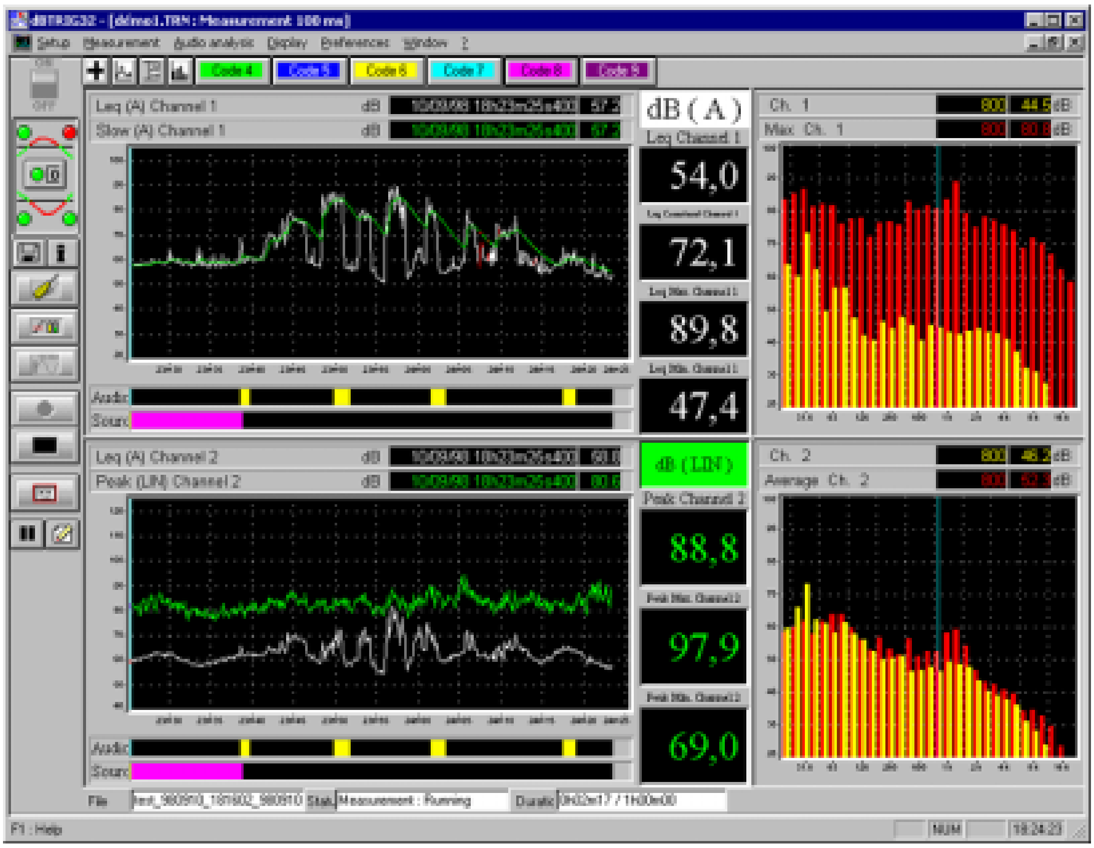
dBTRIG32 SOFTWARE

Acquisition part of dBENV32



01dB-Stell
MVI technologies group

USER MANUAL

dBTRIG32

ACQUISITION PART OF dBENV32

dBTRIG32 acquisition software
User manual
Standard and Expert modes

💻 **infogb@01dB-stell.com**
🌍 **www.01dB-stell.com**

01dB-Stell Head Office
565, rue de Sans-Souci
F-69760 Limonest
FRANCE
☎ (33) 4 72 20 91 00
🖹 (33) 4 72 20 91 01

01dB-Stell Italia s.r.l.
Via Berchet, 13
35131 Padova
ITALY
☎ (39) 49 92 00 966
🖹 (39) 49 92 01 239

01dB-Stell Inc.
1583 East Genesse Street
P.O. Box 796
Skaneateles NY 13152
USA
☎ (1) 315 685 3141
🖹 (1) 315 685 3194

ise data using
T32 - which
mits detailed
nalysis of the
ise
mate in both time
⚖ frequency

The specifications are subject to change without notice (E&OE)

P83MAN20-01 NOT1103

gb_dBTRIG32_4.0_NE_manual.doc – Updated on: December 2001

TABLE OF CONTENTS [1]

[1] **Important Notice**: Because this software package is modular in structure, some of the functions described in this manual may not be available in your copy of the software. To upgrade your version with optional modules, contact your 01dB agent.

1. DESCRIPTION OF A 01DB PC-BASED MEASUREMENT CHAIN

You have purchased a 01dB PC-based system to perform sound and vibration analysis. The 01dB concept is to combine sound and vibration instrumentation with the computational and management facilities of a PC.

For first time users of a PC based measurement chain, the following chapter describes the concept with respect to environmental noise measurements.

For more general information on the concept of a PC – based measurement system, see the "getting started user manual" delivered with your system.

1.1. Introduction

In recent years, there has been much written and said about the use of personal computers in acoustics and vibrations, both for measurement and results processing. The advances in PCs have been nothing short of phenomenal, driven by the fast moving requirements of information technology in the office environment, and these benefits are starting to filter into our world of acoustics, in the form of more flexible measurement tools.

The pure technologists amongst us would doubtless plunge headlong into building an instrument on a PC, which in principle is not a difficult job, but the resulting system must offer clear benefits to the user before it can be accepted as a true alternative to more dedicated instrumentation.

This is where one of the traditional divides has emerged between instruments and PCs. The dedicated instrument has always been used to provide the measurements, and the computer has been used simply as a storage device for archiving and displaying results, with simple post-processing functions. The interface between the two has either been in the form of a simple RS-232 serial communication, or via manual entry of results from a paper printout. This latter in particular has been responsible for many errors, with the tedium of copying numbers into a spreadsheet, for example, from a long roll of silver paper!

It is only relatively recently that computers have moved into the measurement arena, supported by the massive increase in available processing power. But to build a sensible instrument using a PC, we need to consider which processes are better handled by dedicated hardware, and what can be transferred to the PC environment.

The use of personal computers for acoustical measurement and data processing has been a topic of much discussion, even controversy over recent years. The phenomenal growth in PC technology now offers the benefits of computational speed and flexibility that are being employed in many areas of research and engineering applications.

Traditionally the function of acoustical measurement has belonged to the dedicated instrument while the computer has carried out the tasks of data storage, display and basic processing. The interface between the two has been either an RS-232 serial communication or manual entry via a keyboard. Human error, excessive time and long rolls of silver paper are just some of the disadvantages of this system.

An attractive solution for the pure technologist is to build a PC based instrument, though the benefits over the dedicated instrument should be apparent. Alternatively, the design of an optimum PC based measurement system will depend on the management of data handling between the dedicated hardware and the PC environment. The possibility of flexible and creative solutions that offer a very high degree of accuracy over a wide range of applications have been sought and developed over the last ten years by the team at 01dB.

1.2. General description

01dB have developed a modular PC - based measurement chain, similar to that used by traditional stand-alone measurement systems. This generic approach also applies to sound level meters, tape recorders or multi-channel analysers.

First, the **transducer** transforms a physical phenomena into an electrical input signal. Secondly **the signal conditioning** block transforms and/or amplifies the input signal for treatment in an acquisition unit. Conditioners require power supply.

The conditioned signal is then fed to a **digital signal acquisition unit**, with some dedicated DSP function performing the actual measurements.

Control of the instrument and output of the results is required. Control is via key commands on a control panel. Results such as a spectrum or time history graphs may also be stored, for example as raw audio data, similar to a DAT recording or as a spectrum memory or set of Ln measurements. Results are then available at the convenience of the user for post processing operations, such as building acoustics calculations.

Finally, the instrument may have an interface to a computer for further analysis and reporting.

The **personal computer** may be used as a host to a dedicated instrument. Duplication of function, however, quickly becomes apparent. A PC may handle many internal functions of the dedicated instrument more efficiently. Benefits of the PC include access to greater storage capacity, input / output devices, higher resolution with a Windows™ style graphical interface and general integration with the computer based design environment. The computer may also take responsibility for system control. Note that some existing analysers actually feature an internal x86 PC processor as the system controller!

As graphics, user interface, storage and post-processing are all handled by the PC platform, and the type of measurement performed is defined simply by the application software running on the computer, which sends calls to the hardware resources as necessary.

Application software packages run under Microsoft Windows. These define the nature of the instrument such as environmental noise analyser, a building acoustics analyser, a signal/frequency analyser, etc. To change the instrument, the user simply calls up different software 'modules'.

The **Level of Accuracy** achieved by a Sound Level Meter is determined by its grade, and is regulated by the legal requirements of the Standards IEC651, IEC804, IEC1043 and their national equivalents. Minimum requirements are given in respect of dynamic range, linearity, frequency response and indications.

What is a PC based measuring system? It is a system comprising hardware resources, software modules and a host computer, which must meet the current Standard specifications for its class. The requirements currently reflect the use of dedicated instruments, but until they are revised, the new instruments must still comply.

01dB is dedicated to the design and development of portable PC based measurement systems that offer a very high degree of accuracy and are intuitive to use in all aspects. Our systems are type 1 approved in several countries. This accuracy rating applies to the use of generic computing hardware, that in practice enables any brand of computer to be used that meets current minimum standards.

1.3. Environmental noise applications

To perform acoustical in the environment with a 01dB PC - based measurement chain, this is the list of the hardware elements required. This list is non-exhaustive and may vary from application to application.

Transducer unit (dual channel measurements in option)

- Type I or Type 2 condenser microphone (pre-polarised, externally polarised)
- Associated preamplifier. It should supply the polarisation voltage for the condenser microphone if required.
- Outdoor microphone unit, containing both a microphone and a preamplifier, can be used for long term noise monitoring applications.

Accessories

- Windshield to protect the microphone (or an all weather windshield or an outdoors microphone unit).
- Extension cable for connection to the acquisition unit.
- Tripod for the microphone unit.
- Measurement case for outdoors measurements.
- Type 1 or Type 2 acoustical calibrator to perform calibrated measurements.

Measuring instrument

- Notebook, industrial or desktop computer, that meets the minimum requirements specified by 01dB, with a Windows operating system.
- Acquisition unit connected to the Notebook (e.g. SYMPHONIE box).
- dBTRIG32 application software for measurement, dBTRAIT32 application software for processing and various optional modules.

The photograph illustrates a field environmental noise measurement with a 01dB measurement chain.

All these elements can be purchased from 01dB. Contact your sales' representative for more information.

A complete description of how these different elements interconnected is given in the getting started manual delivered with your system

2. MEASURING WITH DBTRIG32

2.1. Principle of measurement

There is a sequence of steps that the user should follow. This will lead not only to successful measurement but also to rapid familiarity with the software. Each stage is described in the manual.
Before proceeding with the measurement procedure shown below, it is necessary to set up the measurement chain.

- **Stage n° 1: Set up of the measurement chain**

Connection of the hardware elements, definitions of their characteristics in the databases' utility dBCONFIG32, set up of the signal conditioning options of both the transducer and the hardware peripheral. Refer to the getting started user manual delivered with your system for more details.

- **Stage n° 2: Selection of acquisition hardware, transducer and calibrator**

Choose **Setup** / **Hardware configuration.** Select the correct transducer, calibrator, remote control if required and hardware platform for each measurement channel.

📖 *This command is not available if the main measurement window is open.*

- **Stage n° 3: Opening a new Measurement file**

Select **Setup/New** or **Open** an existing file. Measurement files contain all the measurement parameters. At this stage the measurement window appears. Once the ON/OFF switch is set to ON, the screen presents real time visualisation of the time histories of the recorded quantities. The recording is in *Play* mode and data will not be saved to a datafile until the measurement start is activated according to the programmed conditions

- **Stage n° 4: Calibrating of the system**

Select **Setup/Calibration** and perform the measurement chain calibration.

- **Stage n° 5: Setting the system parameters**

Choose **Setup/Parameters**. These include acquisition, storage, audio recording, period definition, threshold and gain settings, automatic calibration, coding and system alarms.

- **Stage n° 6: Setting the dynamic range for the measurement**

Choose **Setup/ Dynamic range** and select the appropriate dynamic range so that no overloads or underloads occur. Automatic adjustment of the dynamic range by the software is also possible by using the command **Setup / Parameters / Advanced parameters** and **Automatic gain shift** tab.

- **Stage n° 7: Setting the visualisation parameters**.

Via the **Display** menu or the icons on the toolbar. Options include Layout, Time history, Digital Indicators, Spectrum. Customise the visual interface of the instrument.

- **Stage n° 8: Starting the measurement...**

Easy to use icons for start, stop and pause are found on the vertical command bar of the measurement window. Alternatively, pre-programme the timing with periods and time slots.

- **Stage n° 9: Coding noise events on-line.**

dBTRIG32 offers direct coding of time events during acquisition either from the tool bar or from the pull down menu.

Each stage is described in the manual. Data processing is performed in the software module dBTRAIT32.

2.2. New functions and optional modules

Further to the 32-bit environment of 01dB application software, **dBTRIG32** features many new functions that did not exist in previous versions of the software:

- **New data file format (CMG) common to all 01dB application software packages**

All the noise data acquired by **dBTRIG32** are now saved to the computer hard disk into a single datafile called **measurement session (*.CMG)**, which stand for measurement campaign in French. It replaces the LEQ datafiles (for noise levels and spectra) and the WAV datafiles (for audio records) of the 16-bit versions of 01dB software.

- **Measurement in physical units**

It is now possible to acquire and display the measured quantities in physical units rather than decibels (for example, sound pressure can be expressed in Pascal directly).

- **Edition of the reference values for calculation of levels in decibels (dB)**

The user may now define the reference values of any physical unit for calculation of its level in decibels. This function may be useful for specific industrial applications.

- **Real time measurement of the spectrum in octaves and third octaves of time weighted levels (Lin, Fast, Slow or user-defined time constant)**

The real-time acquisition of an octave or third octave spectrum is now possible for Leq, Fast, Slow or user-defined time constant quantities. Furthermore, the time weighted quantities are no longer based on short Leqs calculations but also real Slow and Fast sound pressure level calculations, as defined in standard IEC651.

- **Triggering capabilities for various events**

The acquisition of auxiliary events in dBTRIG32 (audio records, spectra, noise source codes and alarms) according to a user-defined threshold has been improved : management of various triggers (in a database).

- **Dual channel acquisition module (option)**

It is now possible to perform noise measurements on two channels simultaneously with this optional module. With a special cable, the user may obtain a single measurement of 115 dB dynamic (from 20dB to 135dB), using both acquisition channels.

- **Online analysis of audio records during acquisition (option)**

The simultaneous analysis of audio records during acquisition is no longer performed by an external application software module (dBAUDIO) but by a calculation server. In addition to ‘classical’ octave and third octave analysis, dBTRIG32 computes in real-time spectrum and multispectrum in 1/6th, 1/8th, 1/12th, 1/24th and 1/48th octave bands.

- **Vibration module (option)**

With this optional module, computation of overall levels according to ISO2631 standard has been implemented. The third octave frequency range is extended down to 1Hz and the sampling frequency can be set under 40 Hz, depending on hardware, allowing extended analysis of long vibration signals.

- **Expert module (option)**

The expert module allows the user to define any combination of trigger conditions for recording audio or spectrum events and generate alarm signals.
Thresholds can be either relative or absolute providing very flexible event detection and data storage.

- **Psychoacoustic module (option)**

Overall levels used in the field of airport noise assessment (PNL, PNLT) are calculated by the software and can be displayed and stored with other global values.
Similarly, the Loudness level according to ISO532B (Zwicker) is also computed in real-time.

- **Use with a digital tape recorder DAT (system JAZZ)**

Jazz features real-time analysis of DAT recordings by using AES/EBU digital interface. When using a sound level meter connected to a DAT recorder (environmental monitoring for example), **Jazz** will read the tape in digital format, without loss of quality. The signal does not have to be recorded A weighted on the tape, therefore simultaneous global readings and 1/3 octave analysis are possible.
dBTRIG32, used with the **Jazz** acquisition card, allows the operator to manage band indexes of digital audio tapes (DAT).

- **Use of a sound level meter as an acquisition front-end (ACL mode)**

The ACL mode allows using a 01dB sound level meter as acquisition platform in dBTRIG32. The sound level meter sends calibrated data to dBTRIG32 via serial interface.
The user can carry out long-term measurements even on sound level meters without storage function. A continuous surveillance can be included without any interruption of the measurement.

📖 *Optional modules are described in **chapter 13.***

2.3. User level of the software

In the **Preferences** menu of dBTRIG32, the user may choose in between three different levels : Light, Standard and Expert modes. Each level gives access to different functions.

The table below shows all the functions per software module:

	Light version	Standard version	Expert version
Acquisition	- Quantities : Leq (A or Lin frequency weightings), Peak (C or Lin frequency weightings), time weightings - Single channel - Noise measurements only - Fixed Leq and audio pass band fixed to 20kHz	- Quantities : Leq (A, B, C, G or Lin frequency weightings), Peak (C or Lin frequency weightings), time weightings, 1 statistical indice Ln, 1 user-defined time constant - Dual channel (option) - maximum dynamic (option) - All type of transducers - Spectra (Leq, Fast, Slow) in real-time in octaves and third octaves **Psychoacoustics module**: - Loudness, PNL, and PNLT. **Vibration module:** - Audio pass band upper limit ranging from 40 Hz to 20 kHz - ISO2631 and 8041 vibrational weightings - Spectra : octaves 2Hz - 16kHz 1/3 octaves 1Hz - 20 kHz **Use with a DAT recorder (system Jazz only)**	Identical to standard version
Storage	- Manual - 1 or more quantities - Audio recording manually or according to a simple absolute threshold	- Manual, clock or user-defined periods - All acquired quantities + spectrum - Audio recording manually or according to a trigger	- Identical to standard version + spectrum or multispectrum event, acquired according to a trigger
Coding	- Manual (6 codes from F4 to F9) - Delayed coding possible	- Manual (6 codes from F4 to F9) - Delayed coding possible - Definition of independent codes per channel	- Identical to standard version - Simultaneous coding - Coding according to a trigger
Display	- Only one quantity time history of fixed duration (1000pts) + associated digital indicators	- Up to 6 acquired quantities - Averaged Leq display - Digital indicators associated to one displayed quantity - Display on the same plot the time histories of active channels, only if the acquisition parameters of both channels are identical	- Identical to standard version

📖 *Consult the dBTRIG32 light version manual for more details on this mode.*

📖 *When the user selects a new user level, the application software has to be restarted*

2.4. Modes of operations

dBTRIG32 has three modes of operation:

2.4.1. Inactive mode

The inactive mode closes the interaction between the acquisition platform and computer interface. It is accessible either from the ON/OFF icon found at the top right corner of the measurement screen or from the Menu **Setup / Hardware ON/OFF**. The inactive mode reduces power consumption. Active mode will be resumed automatically if a pre-programmed acquisition begins.

2.4.2. Play mode

While configuration parameters are set, dBTRIG32 displays the signal reading without recording (saving to hard disc). The real time visualisation of the results enables configuration parameters to be set before the measurement is recorded.

- The configuration parameters may be saved to a file for use with subsequent measurements under the same conditions.
- Record mode may be initiated manually by the user or automatically at a pre-programmed time.

2.4.3. Record mode

Data acquisition is recorded and saved to hard disk. Note that once recording, the acquisition parameters are unavailable for modification.

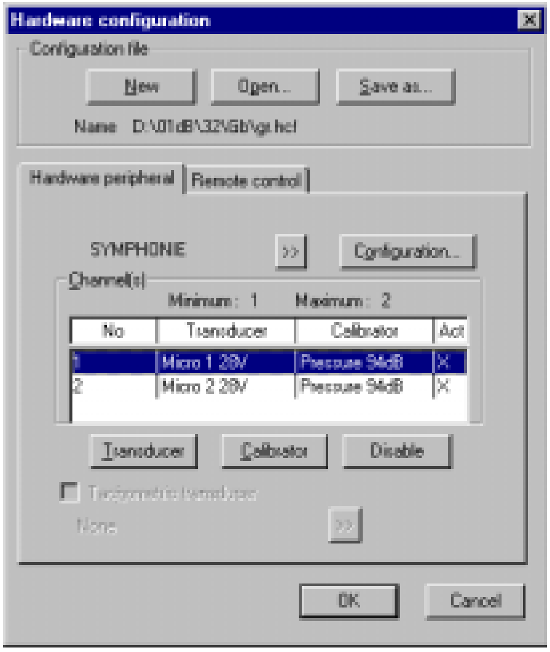
- During recording, the acoustical quantities identified for measurement are saved to hard disk. If audio recording is active, dBTRIG32 records the whole of the signal to disk according to a trigger
- During the measurement, the operator can activate the pause button as a simple method of editing unwanted signal input. Under record mode, the user has six coding options to identify noises.

📖 *See the following chapters for further details on the command available for each of the functions of dBTRIG32.*

3. MEASUREMENT HARDWARE CONFIGURATION

Hardware specification and settings are required before any measurement. The **Hardware configuration** option is found under the **Setup** menu of **dBTRIG32** main window. This dialog box (see below) features various tabs : **Hardware Peripheral** is used to define which hardware elements are used to perform an acquisition (hardware peripheral, transducers, calibrators, active channels), **Remote control** is used to define and configure a remote control object (this tab is displayed only if the file DBCD32.INI is present in the 01dB program files directory).

If you are using a 01dB Sound Level Meter as an acquisition front-end, see chapter 13.6.



From the **hardware peripheral** tab, define:

- The type of hardware platform
- The active measurement channels
- For each channel, a couple transducer / calibrator of same type
- The signal conditioning options of the selected hardware peripheral (**Configuration** key)

The hardware configuration defined here will be recalled automatically next time the program is used.

The acquisition platforms, transducers and calibrators are selected from hardware elements' databases defined under the hardware configuration programme dBCONFIG32.

Access to tachometric transducers is only available in dBFA32.

- **Hardware**

The hardware board configuration sets up the computer so that it will be able to record data generated by the specified board. It sets the number of possible active channels. The Configuration key gives access to signal conditioning options, built-in the hardware unit.

- **Active channels**

Among the possible channel(s) available on the acquisition unit, define which channels will be active for both acquisition and calibration.

- **Transducer**

The transducer configuration loads the transfer function of the selected transducer and allows the conversion of measured data into an input voltage and the reverse process after data analysis to display the results. A transducer must be prescribed to each active channel.

- **Calibrator**

The calibrator configuration allows the user to perform the calibration routine, which adjusts the transfer function of the transducer in order to perform calibrated and accurate measurements. A calibrator must be declared for each active channel.

To enable direct power supply of a transducer from a SYMPHONIE unit or a JAZZ acquisition card, define the same option(s) for the transducer(s) and for the hardware platform (Configuration command).

Selection, use and configuration of remote controls are dealt with in ***chapter 12****.*

For more information concerning hardware configuration, refer to the getting started manual delivered with your measurement system.

4. MEASUREMENT CONFIGURATION FILES

The measurement configuration files of **dBTRIG32** determine the parameters under which measurements are triggered, recorded and saved. (Commands **Setup / New, Setup / Save and Setup / Open**).

This facility for pre-programming a measurement configuration enables non-experienced operators to carry out measurements. The configuration files contains the following elements:

Configuration parameters that are saved

- Acquisition parameters
- Storage parameters
- Audio recording parameters
- Recording time settings
- Threshold/Gains settings
- Automatic calibration, where applicable
- Coding parameters
- Alarm management, where applicable
- Source names

Screen visualisation parameters, defined from the Display menu

- General layout parameters
- Time history selection
- Digital Indicator selection
- Spectrum display where applicable.
- Fine tuning of axis and cursor settings

The colours
The user may change element colours, to suit personal preferences.

📖 *Note that the current dynamic range of the measurement is also saved.*

When an existing measurement configuration file is loaded, the acquisition software set-up the acquisition, storage and display parameters such as defined in the configuration file.

The main acquisition window of dBTRIG32 is displayed once a new or existing measurement configuration is loaded.

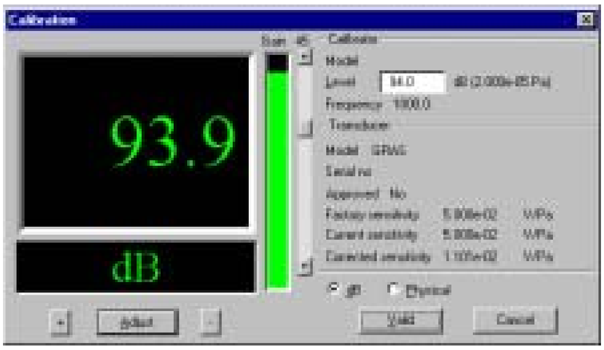
📖 *In standard mode, the measurement configuration files have the extension **TRN**, while in the expert mode, the measurement configuration files have the extension **TRE**.*

5. MEASUREMENT CHAIN CALIBRATION

Calibration is recommended before every measurement. Calibration guarantees the reliability of the results.

Calibration affects the sensitivity of the selected transducer by adjusting it as a function of measured and expected values (defined by the frequency and level characteristics of the calibrator). The calibrators and transducers are defined by using dBCONFIG32 and they are selected using the **Hardware configuration** command in the **Setup** menu.

The current calibration is done using Leq over a 125-millisecond period. It measures the Leq value of the input signal and converts it into the unit set in the transducer's characteristics. By adjusting the level to the expected level, it changes the sensitivity of the transducer. By validating it, the adjusted value will now become the default value for the next time the program is used.



Access **Calibration** via the **Setup** menu.

The input gain and transducer sensitivity may be calibrated from the control panel Values may be modified using the '+', '-' and 'Adjust' buttons.
Calibration levels can be expressed either in **dB** or in **physical** units.

On validation, the system is ready to carry out calibrated measurements.

Caution! Before calibration:

- Verify that the calibration signal remains constant for a sufficiently long period.
- Verify that the gain view meter is correctly positioned (neither too weak, nor overloading).
- It is preferable to place the calibrator on foam to reduce the effect of vibrations.

Caution! After calibration:

- If, for the same transducer / calibrator pair, the sensitivity after calibration differs greatly from the original sensitivity, damage to the microphone may have occurred.

- If the measured values are not correct but the calibration value is OK, it could mean that the sensitivity of the microphone is correct only at 1 000 Hz. Check the microphone membrane.

A microphone is very fragile equipment. A fall of 10-cm may damage the microphone membrane. As general rule, if the measured value in dB varies by +/- 1.5 dB from the value that would be measured with the microphone according to the original sensitivity (see calibration data sheet), consider your microphone as faulty.

***Example*: For a microphone that as a factory sensitivity of 50 mV / Pa and a calibrator that delivers 94 dB at 1000 Hz.**

The microphone is able to perform correct measurements if:

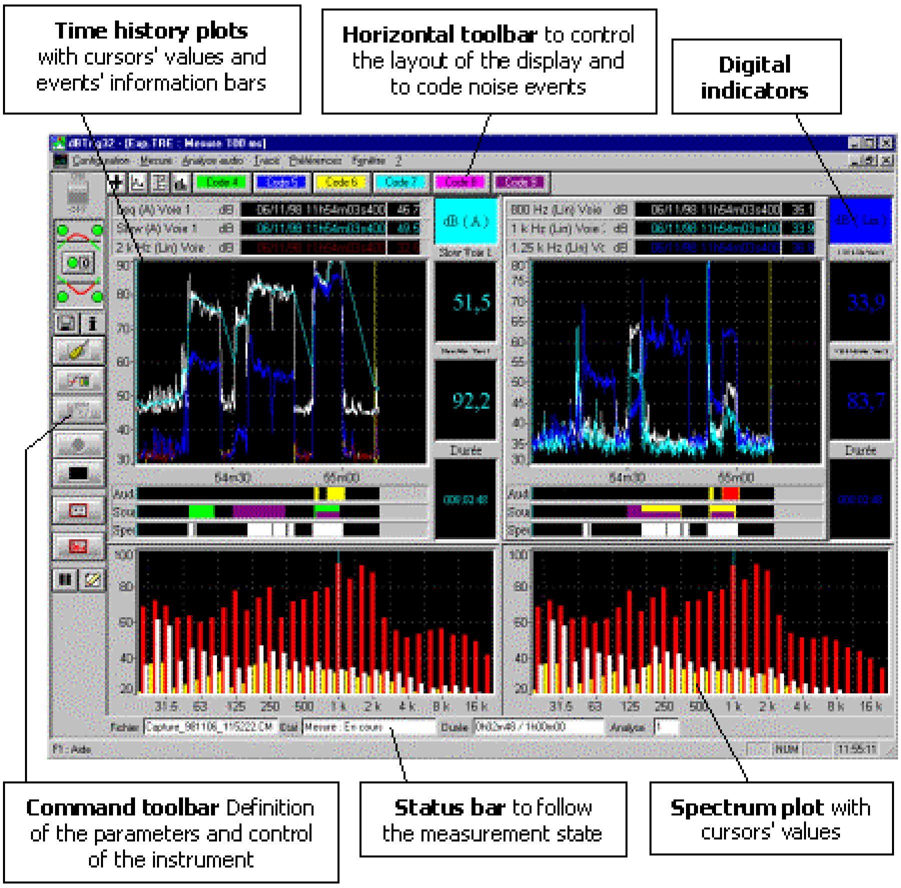
- **The measured calibration level lies between 92.5 dB and 95.5 dB.**
- **The current microphone 'sensitivity lies between (around) 40 mV/Pa and 60 mV/Pa (multiply or divide the original value by a factor of 1.1885)**

For greater or lower microphone 'sensitivities, consider the microphone as faulty. Return it to your 01dB agent.

6. DBTRIG32 MEASUREMENT WINDOW

Complete control of the measuring instrument is available from the **dBTRIG32** measurement window. It is displayed when a measurement configuration file is opened. The main windows features the following elements:

- Command toolbar
- Horizontal toolbar
- Time history plot
- Spectrum plot
- Information bars
- Status bar



The measurement window is slightly different in standard and expert modes.

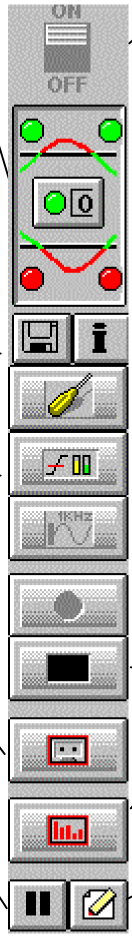
In expert mode, it is possible to record spectrum as events. An event information for the spectrum and a key in the command bar are therefore added to access this functionality.

📖 *The measurement window parameters can be saved in a configuration file by using the commands **Setup / Save** or **Save as**.*

📖 *It is possible to display full screen the time history plot or the spectrum plot by double clicking on it. Double click again on the plot to go back to a full display.*

6.1. Measurement window command bar

This toolbar is used to configure and run the measurement.



Show **overloads and underloads** and reset digital indicators (click on the centre button). The top section presents the overload indicators (on the left, an instantaneous indication, on the right, hold if an overload occurred. The bottom section presents underload indications

Start / Stop dialog between the application software and the acquisition peripheral

Display an **information window** about the current measurement parameters.

Generic **filename** (measurement session *.CMG)

Measurement parameters (acquisition, storage, coding, automatic calibration and gain adjustment)

Selection of a fixed **dynamic range** for the measurement.
It is however possible to automatically adjust the dynamic range while storing data by using the command Configuration / Parameters - Advanced - option Automatic gain.

Calibration (recommended before any measurement).

Storage of the noise data on the hard disk.

Stop storage of the noise data on the hard disk.

Start / Stop an **audio** recording manually, whatever the trigger conditions defined to acquire audio events.

Start / Stop recording a **spectrum** event manually

Pause : noise data is still stored on the hard disk but they are automatically coded, so that they are no taken into account for later processing operations.

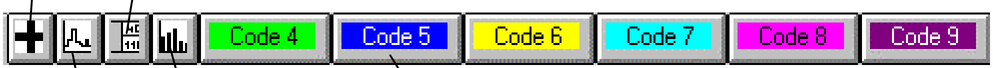
Insert **comments** during measurement

6.2. Measurement window horizontal toolbar

The toolbar defines the content of the visualisation, and enables dynamic coding of noise events:

Set-up of the **contents** of the measurement window (curves' display mode, cursors' values, display information bars, legend, type of scale used, etc.)

Set-up the **digital indicators** in the measurement window. Shown quantity (Leq, Peak, Slow, ...), displayed values (Instantaneous, cumulated, Maximum, Minimum, cumulated duration), space share between the indicators and the time history plot.



Start **source coding** by clicking on the adequate coding button. The noise data will be associated to this code number until another code is applied or if the user stops the current coding operation.

Set-up of the **spectrum** plot : display dynamic, show max / min and averaged spectra.

Set-up of the **time history** plot : select the quantities to display, duration of the display, display dynamic range.

6.3. Measurement window status bar

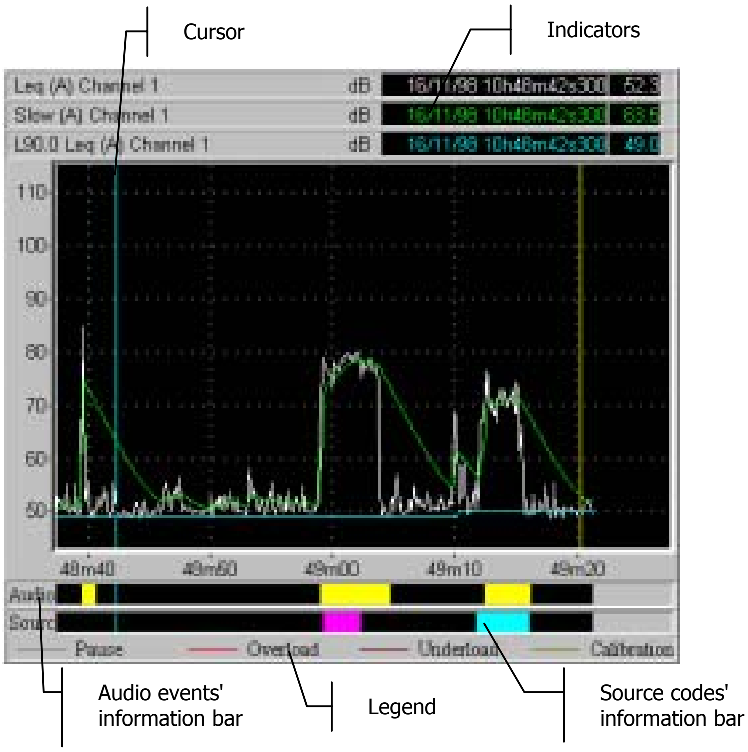
This status bar presents from left to right : the name of the measurement session where data is logged, the status of the measurement or the type of trigger used to start data logging, the duration and remaining duration of the measurement and the number of audio records that have not yet been analysed by the calculation server.

To change the datafile name, use the command **Measurement / New filename**.

To modify the measurement duration, use the command **Setup / Parameters / Storage.**

6.4. Time history plot

The time history plot window displays the time history of the any acquired quantities (see paragraph 8.2).



The indicators at the top of the display window indicate the instantaneous values of the displayed noise quantities at the cursor location (at the contrary of the digital indicators of the measurement window that present overall noise levels).

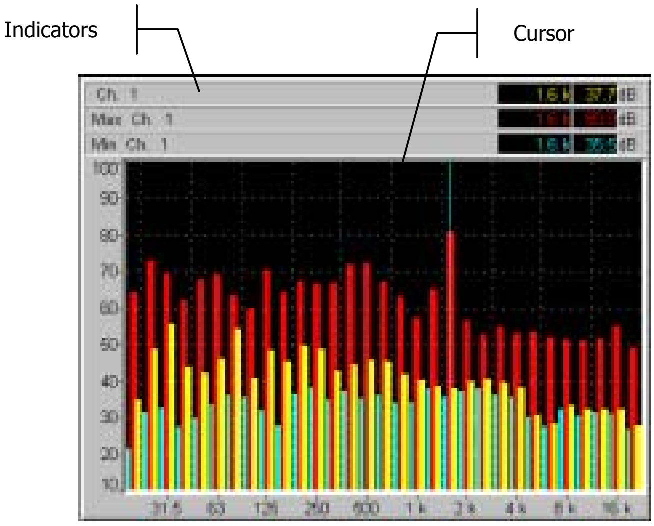
Double click on these indicators to select which quantities will be displayed.

The audio information bar indicates when audio records are acquired. It gives the time limits of any single audio event. When audio analysis is activated, the colour of the audio event reference changes when this particular event is being processed.

The code information bar indicates the time limits of the active code. Six different sources may be defined by the user.

In expert mode only, an additional information bar is shown for spectrum events.

6.5. Spectrum plot



The real-time octave or third octave spectrum acquired (see paragraph 7.1.2) is shown in this display window. The plot is refreshed for each integration time.

The minimum, maximum and averaged spectra can also be shown along with the instantaneous spectrum. When the reset indicator command is used, the display is cancelled as for the digital indicators (see paragraph 8.4).

The indicators at the top of the display window indicate the instantaneous values of the displayed Spectra at the cursor location.

Double click on these indicators to select which quantities will be displayed.

6.6. Definition of auxiliary event (expert mode)

dBTRIG32 in expert mode allows the user to record different type of events simultaneously to the main data flow (overall noise quantities, spectra). The following auxiliary data can be acquired:

- Audio records
- Averaged spectra over the event duration
- Mutlispectra
- Codes
- Alarms

Audio events and spectrum events may be triggered either **manually** from the measurement window command bar or by using a **trigger**, that consists of threshold and time conditions.

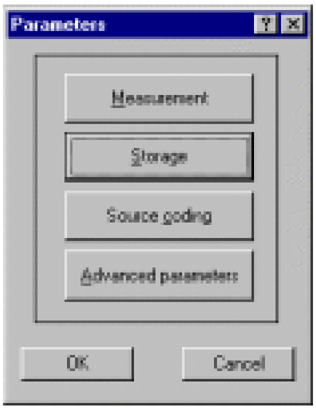
Coding of a single or several noise sources may be triggered either **manually** from the measurement window command bar or by using a **trigger** that consists of various threshold conditions

Signal alarms may be activated only for a trigger made of threshold conditions. To trigger an alarm, a remote control object has to be defined at the hardware configuration stage.

For each type of event (audio, spectra, codes, alarms), the user may define overall parameters (maximum event duration, event stretching before and after the acquisition, etc.) specific to each event as well as various triggers.

Triggers defined in **dBTRIG32** may be used for all types of auxiliary events. The way triggers are defined and used is dealt with in the following chapter.

7. MEASUREMENT PARAMETERS (SET-UP)



The command  **Setup / Parameters** regroups most of the parameters that have to be defined before starting a measurement. This command allows the user to define:

- Acquisition parameters
- Storage parameters
- Source coding parameters
- Advanced parameters (automatic gain shift, automatic calibration, alarm event in expert mode)

By clicking on **OK**, all the parameters that have been defined are applied to the measurement system.

Some parameters are different in standard and expert modes. Both settings are described and clearly identified in this manual.

7.1. Acquisition parameters

Use this command (**Acquisition** key) to define the measurement acquisition parameters. The dialog shown below features three different tabs in order to define :

- Overall acquisition parameters (**Overall** tab), such as the measurement passband.
- Acquired quantities per measurement channel (**Channel** tab), such as a third-octave spectrum.
- Any additional quantities to measure (**Advanced** tab), such as a Ln statistical indice.

7.1.1. Overall tab

- **Maximum frequency** : Fixed to 20 kHz.
It corresponds to the maximum analysis frequency for acquisition of time quantities. It defines the frequency pass band taken into account for the calculation of Leq, spectrum and other indicators.

- **Maximum audio frequency**
Some hardware peripherals authorise a different frequency pass band for audio recording than for time level quantities. Any subsequent frequency analysis of an audio record will be performed up to this frequency only.

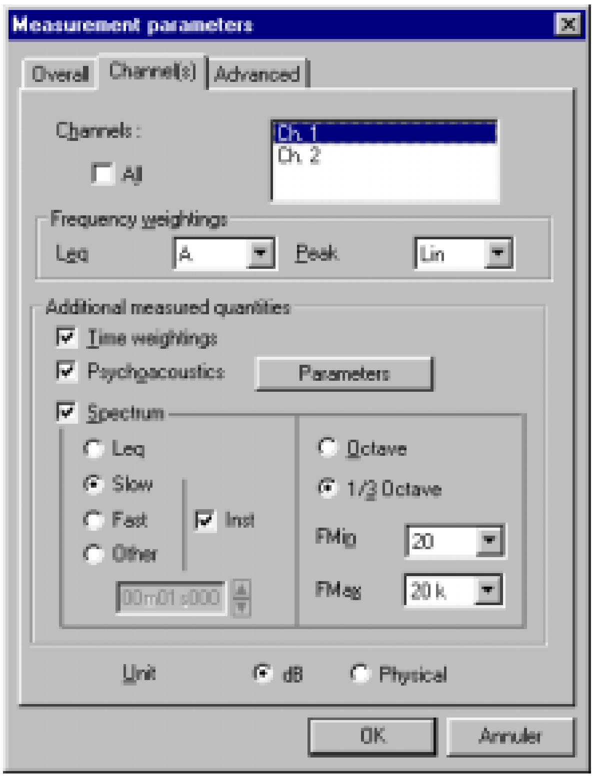
- **Time base**
It corresponds to the acquisition rate. This period also corresponds to the logging and display rate. One value per time base will be calculated. The values of this parameter depend upon the selected hardware platform.

- **Input –Output Loop.** By activating this option, the user can listen to the input signal (from the microphone) with earphones plugged to the output channel.

- **Management of indexes on DAT bands.** By activating this option, the user can manage DAT tape indexes with a JAZZ system. *Refer to the **JAZZ** information manual for more details.*

- **Maximum dynamic.** With a special cable, the user can link two measurement channels in order to make a single one of greatest dynamic (typically 115dB with SYMPHONIE). A single measurement range from 20dB to 135dB is thus available. *Refer to **paragraphs 11.4** and **13.2** for more details.*

7.1.2. Channel(s) tab



This tab is used to define the following parameters, either for all active measurement channels or for each channel individually.

Weightings

- Frequency weightings to apply to all acquired quantities except the Peak level that has an independent network. A, B, C, G and Lin frequency weightings are available.
- For peak values, C and Lin weightings are available.

📖 *If the **Vibration module** is available, the ISO2631 vibrational weightings can be selected in this field. See paragraph 13.3.*

Additional measured quantities

- Calculation of sound pressure levels according to the following **time weightings** : Fast, Slow and Impulse.

If this option is selected, the following quantities will be calculated : **Slow Inst, Slow, Slow Min, Slow Max, Fast Inst, Fast, Fast Min, Fast Max, Impulse, Impulse Max.**

📖 *For more details on the calculation of time weighted sound pressure levels, refer to **paragraph 16.1***

- **Psychoacoustics** calculations **(option)** : **PNL, PNLT, and Loudness.** Click on the **configuration** key to define additional calculation parameters for Loudness and PNLT calculations.

📖 *For more details on the calculation of these psychoacoustics criteria, refer to **paragraph 13.4**.*

- **Spectrum :** this option is used to calculate (or not) the real-time spectrum in order to obtain the time history of sound pressure levels per octave or third octave frequency bands (digital filtering technique).

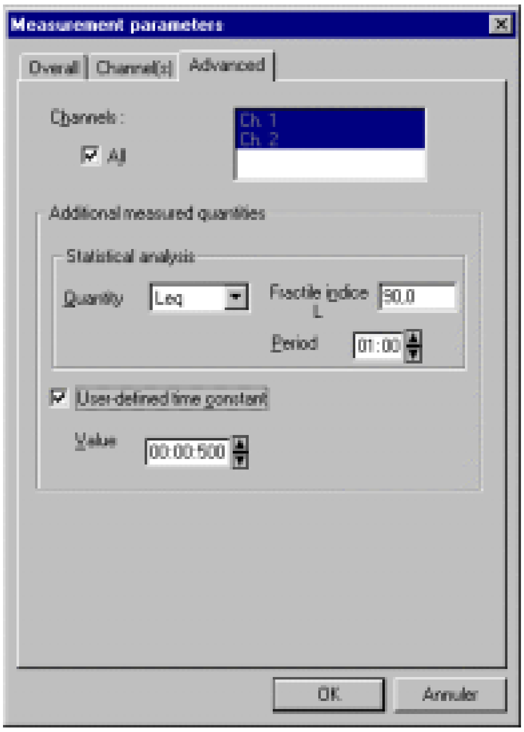
Select to the left the type of quantity to measure per frequency band (**Leq, Slow, Fast, Other**) and to the right the analysis resolution (**octave or third octave**) and the frequency limits (**from 31.5Hz to 20kHz in octave bands and from 20Hz to 20kHz in third octave bands**).
Tick the "**Inst**" box to calculate instantaneous time weighted sound pressure level.

💣 **The spectral values are only guaranteed if they are greater or equal to the lower limit of the active dynamic range - 5dB**

📖 *For more details on the calculation of time weighted sound pressure levels, refer to **paragraph 16.1***

Choose as well if the data will be logged in **physical units** or in **decibels**.

7.1.3. Advanced tab



This tab is used to define the following parameters, either for all active measurement channels or for each channel individually.

- **Statistical indice Ln**

A statistical indice Ln can be measured in real-time. Define which acquired **quantity** will be used for the calculation of the **fractile indice Ln** to calculate and the floating **period** of the calculation.

The indice L corresponds to the percentage of time during which the selected quantity is exceeded.

The software calculates a floating indice : it will be calculated for each time base taking into account the period of the calculation.

In the example shown aside, the statistical indice that will be calculated corresponds to the Leq level that is exceeded during 90% of the time over the last minute of measurement.

📖 *In practice, use L90,0 or L95,0 to approximate the background noise level of a noise climate and L10,0 to approximate the loudest noise source over the measurement duration.*

- **User-defined time constant**

Select if you wish (or not) to calculate time weighting to apply to sound pressure level calculations. Enter its value. This quantity is identified as a RC in the software.

Once the acquisition parameters are defined, define the storage parameters. In other words, define the quantities that will be logged on the computer hard disk.

7.2. Storage parameters (standard mode)

Use this command (**Storage** key) in order to define the data logging parameters. This dialog bow (shown below) features three different tabs to define :

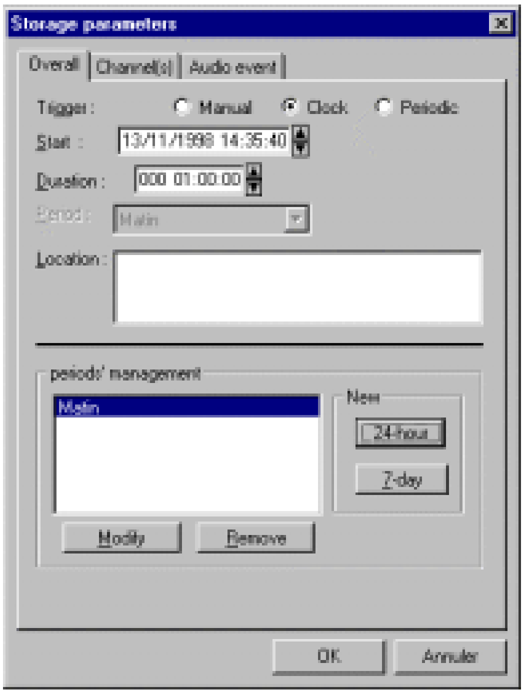
- The way the measurement will be started and its duration (**Overall** tab)
- The acquired quantities to store per measurement channel (**Channels** tab)
- The way audio events will be triggered (**Audio event** tab)

WARNING

The data logging parameters in standard mode and in expert mode are not exactly the same. Each mode is described in this manual.

Refer to paragraph 7.3 for the definition of storage parameters in expert mode.

7.2.1. Overall tab



This tab is used to define the way the measurement will be started. A measurement in **dBTRIG32** may be triggered in three different ways:

- **Manual**

Manually start the acquisition for a given **duration** (format day / hour / minutes / seconds).

- **Clock**

The time and date is programmed to **start** measuring

- **Periodic**

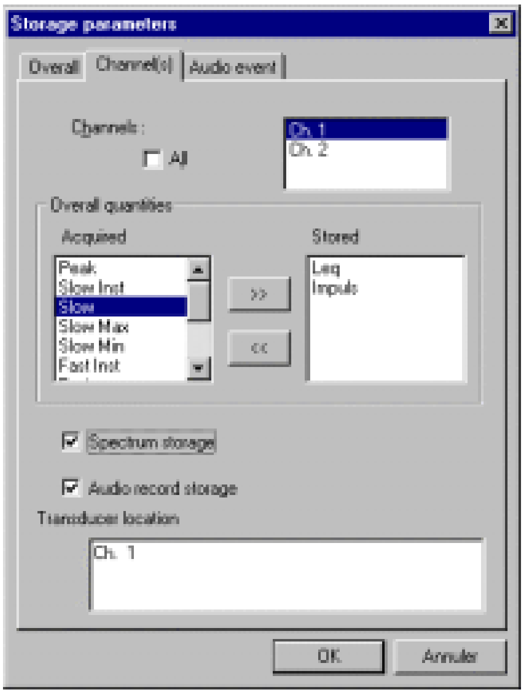
Choose from one of the existing **periods**. Recording is activated during time slots specified by the user, each time the period repeats itself. (for example, daily from 8 a.m. to 8 p.m.)

Define the measurement **location** as well to later identify this measurement session.

The data logging periods allows the user to define periods without overlapping, over a day cycle or a week cycle.
The choice **24 hours** means that it will be repeated identically whatever the day of the week. (For example, no difference will be made between results in the time slice 8h-20h of a week day and a bank holiday day).
The choice **7-day**, means that the software can differentiate between weekdays' periods (for example 8h – 20h) and weekend periods (for example from Friday 18h to Monday 6h).

The periods' definition for data logging is also done in this dialog box. Refer to ***paragraph 7.7*** *for more information.*

7.2.2. Channels tab



This tab is used to define the following parameters, either for all active measurement channels or for each channel individually.

Data logging of overall quantities
dBTRIG32 allows the user to store the time history of all the acquired quantities. The following overall quantities can be stored in a measurement session file:

- **LEQ, Peak**
- **Time weightings** (Fast, Fast Min, Fast Max, Fast Inst, Slow, Slow Min, Slow Max, Slow Inst, Impulse, Impulse Max)
- **User defined time constant** (RC, RC Min, RC Max, RC Inst)
- **Psychoacoustic criteria** (PNL, PNLT, Loudness)
- **User-defined statistical indice**

Select overall quantities in the **Acquired quantities** list and click on  to add them to the **Stored quantities** list.

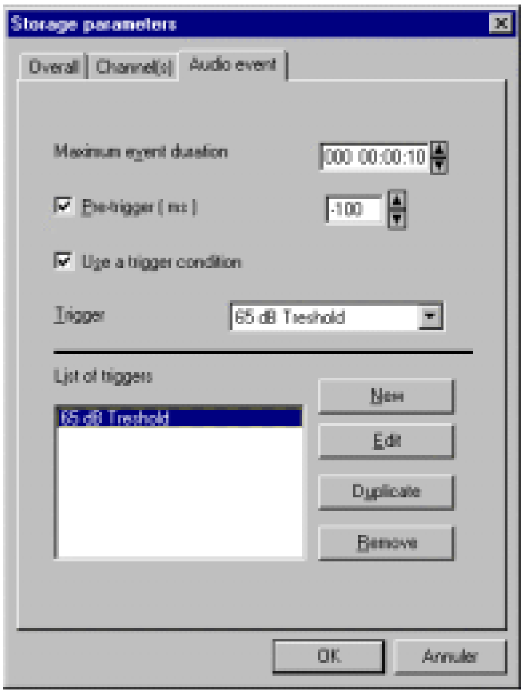
At the contrary, select overall quantities in the **Stored quantities** list and click on  to remove them from the list. There are passed into the **Acquired quantities** list.

One cannot log quantities that are not defined in the acquisition parameters (see section 7.1). If, for example, the Spectrum function is not activated in the Channels tab of the acquisition parameters, noise levels per frequency band cannot be acquired and saved in a measurement session file.

Tick respectively the box **Spectrum storage** and **Audio record storage** to save the spectrum and audio events into a measurement session file.

The field **Transducer location** can be useful for dual-channel measurements with different data logging parameters for each channel (box **All** not activated).

7.2.3. Audio event tab



Simultaneously to the acquisition of overall quantities, **dBTRIG32** may acquire and save audio signals, according to several parameters.

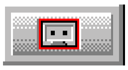
In this tab, define first general parameters for the acquisition of audio events.

Define the **maximum event duration** at the format days / hours / minutes / seconds.

Define a **pre-trigger** option allowing audios recording to start X milliseconds before or after the trigger condition is fulfilled.

Then select the **trigger** that will be used to trigger the event in the **list of triggers**.

If the box **use a trigger condition** is not ticked, and if at least one trigger has been defined, the events can only be recorded manually.

📖 *Click on the icon*  *in the measurement window command bar to manually record an audio event.*

If the box **use a trigger condition** is ticked, and if at least one trigger has been defined, an event will be recorded when the trigger conditions are fulfilled.

For audio events, a trigger can be composed of two types of conditions :

- **Periodic** : Audio records are made periodically according to a period defined by the user.

- **Threshold** : Audio records are made when a user-defined threshold condition (absolute, relative to an acquired quantity or relative between measurement channels) is fulfilled.

As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress.

📖 *The definition and configuration of event triggers are dealt with in* ***paragraph 7.8*** *for dBTRIG32 standard mode.*

7.3. *Storage parameters (expert mode)*

Use this command (**Storage** key) in order to define the data logging parameters. This dialog bow (shown below) features three different tabs to define :

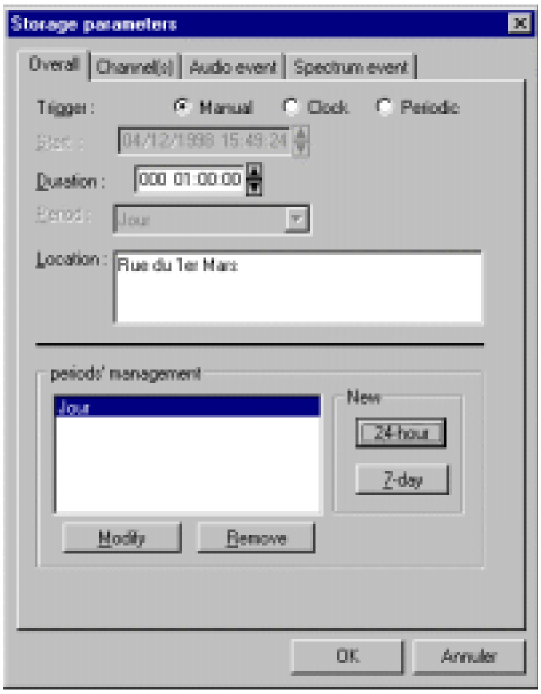
- The way the measurement will be started and its duration (**Overall** tab)
- The acquired quantities to store per measurement channel (**Channels** tab)
- The way audio events will be triggered (**Audio event** tab)
- The way spectrum events will be triggered (**Spectrum event** tab)

WARNING

The data logging parameters in standard mode and in expert mode are not exactly the same. Each mode is described in this manual.

Refer to paragraph 7.2 for the definition of storage parameters in standard mode.

7.3.1. *Overall tab*



This tab is used to define the way the measurement will be started. A measurement in **dBTRIG32** may be triggered in three different ways:

- **Manual**

Manually start the acquisition for a given **duration** (format day / hour / minutes / seconds).

- **Clock**

The time and date is programmed to **start** measuring

- **Periodic**

Choose from one of the existing **periods**. Recording is activated during time slots specified by the user, each time the period repeats itself. (for example, daily from 8 a.m. to 8 p.m.)

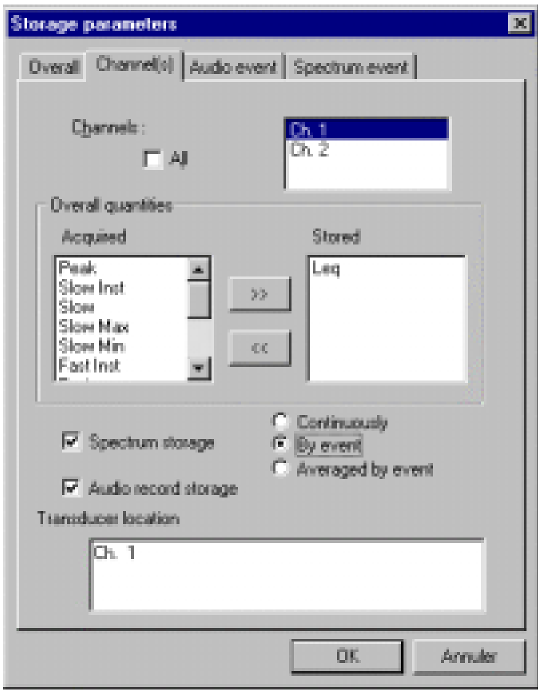
Define the measurement **location** as well to later identify this measurement session.

The data logging periods allows the user to define periods without overlapping, over a day cycle or a week cycle.
The choice **24 hours** means that it will be repeated identically whatever the day of the week. (For example, no difference will be made between results in the time slice 8h-20h of a week day and a bank holiday day).

The choice **7-day**, means that the software can differentiate between weekdays' periods (for example 8h – 20h) and weekend periods (for example from Friday 18h to Monday 6h).

The periods' definition for data logging is also done in this dialog box. Refer to ***paragraph 7.7*** *for more information.*

7.3.2. Channels tab



This tab is used to define the following parameters, either for all active measurement channels or for each channel individually.

Data logging of overall quantities
dBTRIG32 allows the user to store the time history of all the acquired quantities. The following overall quantities can be stored in a measurement session file:

- **LEQ, Peak**
- **Time weightings** (Fast, Fast Min, Fast Max, Fast Inst, Slow, Slow Min, Slow Max, Slow Inst, Impulse, Impulse Max)
- **User defined time constant** (RC, RC Min, RC Max, RC Inst)
- **Psychoacoustic criteria** (PNL, PNLT, Loudness)
- **User-defined statistical indice**

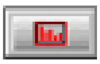
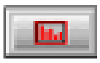
Select overall quantities in the **Acquired quantities** list and click on  to add them to the **Stored quantities** list.

At the contrary, select overall quantities in the **Stored quantities** list and click on  to remove them from the list. There are passed into the **Acquired quantities** list.

One cannot log quantities that are not defined in the acquisition parameters (see section 7.1). If, for example, the Spectrum function is not activated in the Channels tab of the acquisition parameters, noise levels per frequency band cannot be acquired and saved in a measurement session file.

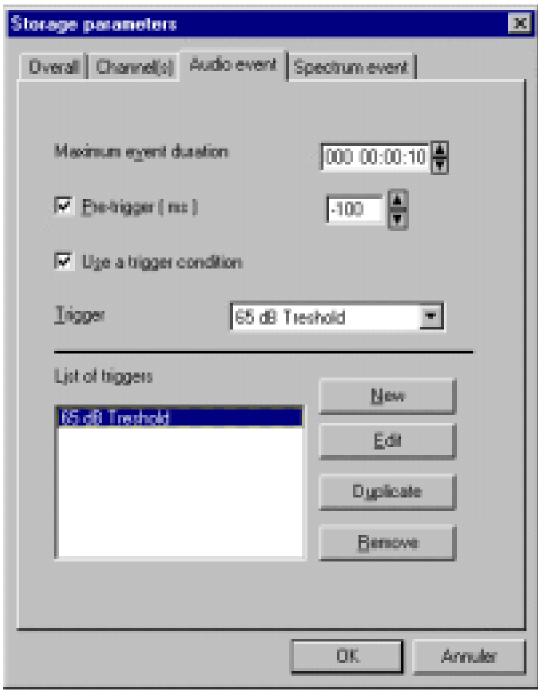
Tick the box **Audio record storage** to save into a measurement session file the audio events.

Tick the box **Spectrum storage** to save the real-time spectrum, as defined in the acquisition parameters. In the expert mode, the spectrum may be stored by three different means:

- **Continuously** : a spectrum is stored for each time base. We obtain the spectral time history of a noise level per frequency band over the complete measurement duration.
- **By event** : a spectrum is stored manually by the key  of the measurement window command bar or when the state of a defined trigger is "true". We obtain the spectral time history of a noise level per frequency band over the event duration.
- **Averaged by event** : a spectrum is stored manually by the key  of the measurement window command bar or when the state of a defined trigger is "true". We obtain a spectrum averaged over the event duration.

The field **Transducer location** can be useful for dual-channel measurements with different data logging parameters for each channel (box **All** not activated).

7.3.3. *Audio event tab*



Simultaneously to the acquisition of overall quantities, **dBTRIG32** may acquire and save audio signals, according to several parameters.

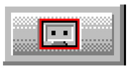
In this tab, define first general parameters for the acquisition of audio events.

Define the **maximum event duration** at the format days / hours / minutes / seconds.

Define a **pre-trigger** option allowing audios recording to start X milliseconds before or after the trigger condition is fulfilled.

Then select the **trigger** that will be used to trigger the event in the **list of triggers**.

If the box **use a trigger condition** is not ticked, and if at least one trigger has been defined, the events can only be recorded manually.

📖 *Click on the icon*  *in the measurement window command bar to manually record an audio event.*

If the box **use a trigger condition** is ticked, and if at least one trigger has been defined, an event will be recorded when the trigger conditions are fulfilled.

For audio events, a trigger can be composed of two types of conditions :

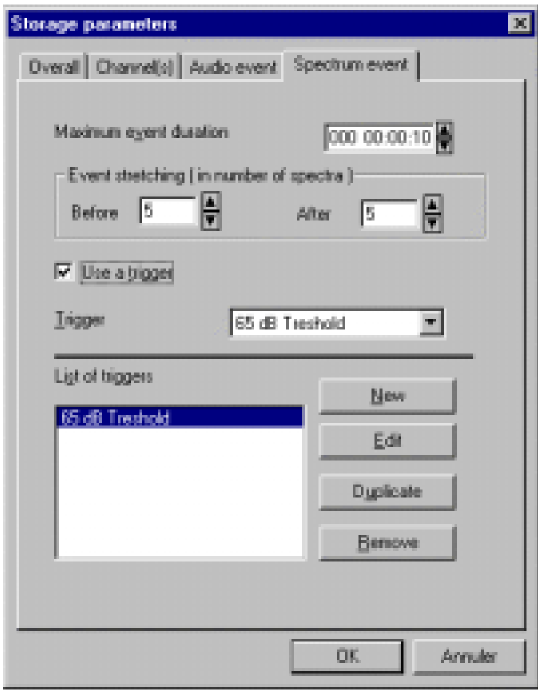
- **Periodic** : Audio records are made periodically according to a period defined by the user.

- **Threshold** : Audio records are made when a user-defined threshold condition (absolute, relative to an acquired quantity or relative between measurement channels) is fulfilled.

As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress.

📖 *The definition and configuration of event triggers are dealt with in **paragraph 7.9** for dBTRIG32 expert mode.*

7.3.4. Spectrum event tab



Simultaneously to the acquisition of overall quantities, **dBTRIG32** may acquire and save spectrum events, according to several parameters (see paragraph 7.3.2).

In this tab, define first general parameters for the acquisition of spectrum events.

Define the **maximum event duration** at the format days / hours / minutes / seconds.

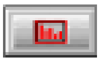
Define an **event stretching** in number of spectra before and after the true event, knowing that **dBTRIG32** acquire a spectrum for each time base (see the acquisition parameters, overall tab, to change this time base).

We therefore can calculate the stretching duration D corresponding to the number of spectra selected N for a time base t by a simple multiplication : D = N * t

In the example shown aside, and for a time base of 200ms, a spectrum event will start 1 second before (5 * 200ms) and stop 1 second after (5 * 200ms) the duration for which the trigger state is true.

Then select the **trigger** that will be used to trigger the event in the **list of triggers**.

If the box **use a trigger condition** is not ticked, and if at least one trigger has been defined, the events can only be recorded manually.

Click on the icon  *in the measurement window command bar to manually record a spectrum event.*

If the box **use a trigger condition** is ticked, and if at least one trigger has been defined, an event will be recorded when the trigger conditions are fulfilled (trigger state = TRUE).

For spectrum events, a trigger can be composed of two types of conditions :

- **Periodic** : Spectrum records are made periodically according to a period defined by the user.

- **Threshold** : Spectrum records are made when a user-defined threshold condition (absolute, relative to an acquired quantity or relative between measurement channels) is fulfilled.

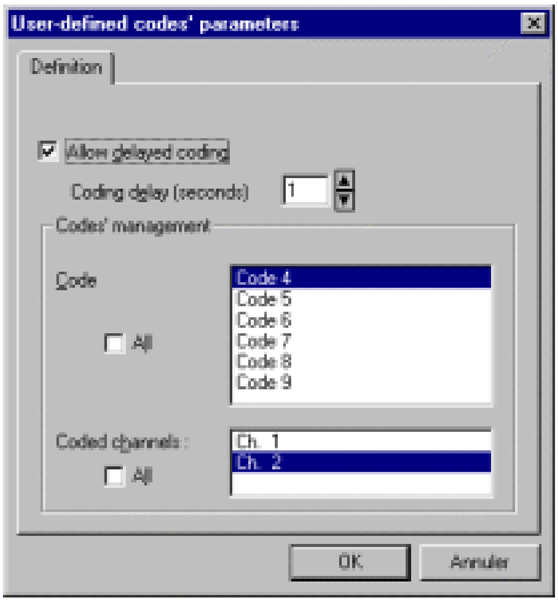
As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress

*The definition and configuration of event triggers are dealt with in **paragraph 7.9** for dBTRIG32 expert mode.*

7.4. Source coding parameters (standard mode)

Use this command (Source coding key) to define noise source coding parameters.. **dBTRIG32** allows the user to perform dynamic coding of noise events during the course of a measurement. The coding option during parameter definition allows direct or delayed coding to be specified.

7.4.1. Definition tab



Coding noise events can be performed in two ways:

- **Direct** (**Allow delayed coding box** not activated): coding of a noise source can be activated from the horizontal toolbar of the measurement window by using the coding buttons.
- **Delayed** (**Allow delayed coding box** activated) : coding of the noise event begins **X** seconds (where X is the **coding delay** parameter). This delay can be visualised on the time history with a cursor.

When performing dual-channel measurements, it is possible to affect code numbers to a given measurement channel only.

In the group **codes' management,** select in the **Code** list a given code number and then affect it to a given measurement channel in the **Coded channels** list. Repeat this operation for all the code numbers of interest.

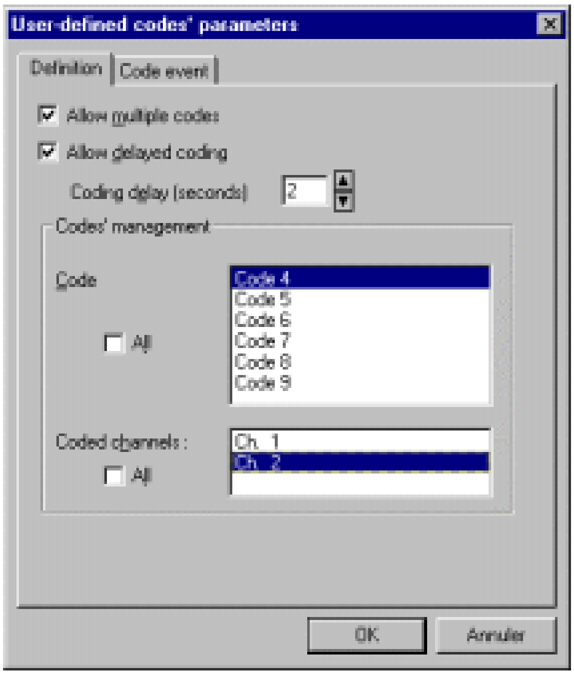
In the above example, the source 'Code 4' is affected to Channel 2 only. When using this code number, only noise data from the second measurement channel will be coded.

If the user wishes to affect all codes numbers to all the active measurement channels, tick the box **All** by the code list and the box **All** by the coded channel list.

7.5. Source coding parameters (expert mode)

Use this command (Source coding key) to define noise source coding parameters. **dBTRIG32** allows the user to perform dynamic coding of noise events during the course of a measurement. Coding can be performed either manually (using the coding keys of the measurement window) or when a user-defined threshold is exceeded The coding option during parameter definition allows direct or delayed coding to be specified. A special feature allows coding noise data with several codes simultaneously.

7.5.1. Definition tab



Activate the option **Allow multiple codes** for simultaneously code noise data with several noise source codes.

If this option is not activated, the current coding operation will be stopped when another code is selected by the user. In other words, only one coding operation at a time is allowed.

Coding noise events manually can be performed in two ways:

- **Direct** (**Allow delayed coding box** not activated): coding of a noise source can be activated from the horizontal toolbar of the measurement window by using the coding buttons.

- **Delayed** (**Allow delayed coding box** activated) : coding of the noise event begins **X** seconds (where X is the **coding delay** parameter). This delay can be visualised on the time history with a cursor.

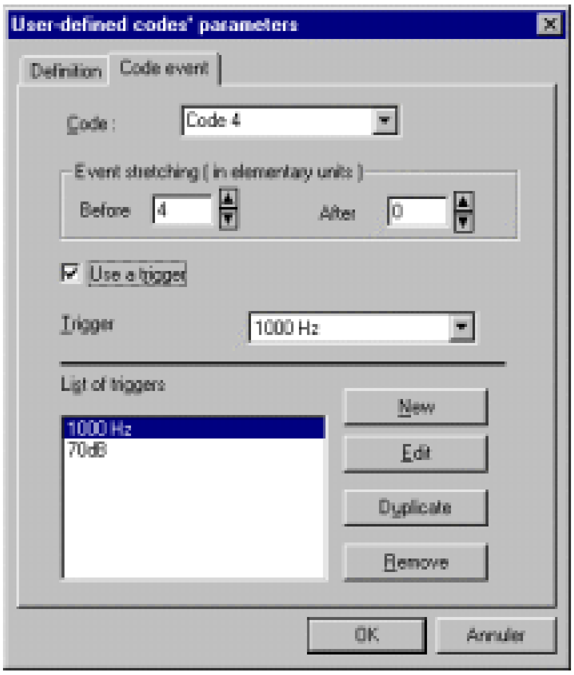
When performing dual-channel measurements, it is possible to affect code numbers to a given measurement channel only.

In the group **codes' management,** select in the **Code** list a given code number and then affect it to a given measurement channel in the **Coded channels** list. Repeat this operation for all the code numbers of interest.

In the above example, the source 'Code 4' is affected to Channel 2 only. When using this code number, only noise data from the second measurement channel will be coded.

If the user wishes to affect all codes numbers to all the active measurement channels, tick the box **All** by the code list and the box **All** by the coded channel list.

7.5.2. Code event tab



Its is possible to trigger noise event coding when a user-defined threshold is exceeded. For each available code number (up to 6), define a threshold trigger.

For a given code (Code 4 in the example), the user may **use a trigger** (threshold) selected in the **list of triggers**. Coding noise data with this code number will be performed automatically when the trigger state is "TRUE".

Define an **event stretching** in number of elementary units before and after the true event, knowing that an elementary unit corresponds to the time base of acquisition in **dBTRIG32** (see the acquisition parameters, overall tab, to change this time base).

We therefore can calculate the stretching duration D corresponding to the number of elementary units selected N for a time base t by a simple multiplication : D = N * t.

In the above example, and for a time base of 200ms, a code event (code 4) will start 0.8 second before (4 * 200ms) and stop 0 second after (0 * 200ms) the duration for which the trigger state is true

For each available code, the user may define different parameters (event stretching, trigger used).

Beware that when using a threshold condition to trigger a noise event, delayed coding should not be activated (in the definition tab) otherwise, the software will start coding at the delayed coding cursor position. The coded noise data will not correspond to the event because of that delay.

If the box **use a trigger** is ticked, and if at least one trigger has been defined, an event will be recorded when the trigger conditions are fulfilled (trigger state = TRUE).

As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress

*The definition and configuration of event triggers are dealt with in **paragraph 7.9** for dBTRIG32 expert mode.*

7.6. Advanced parameters

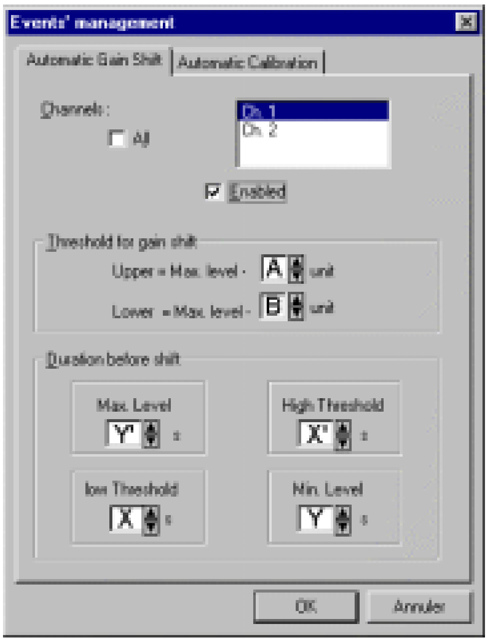
Use this command (**Advanced parameters** key) to define the following parameters:

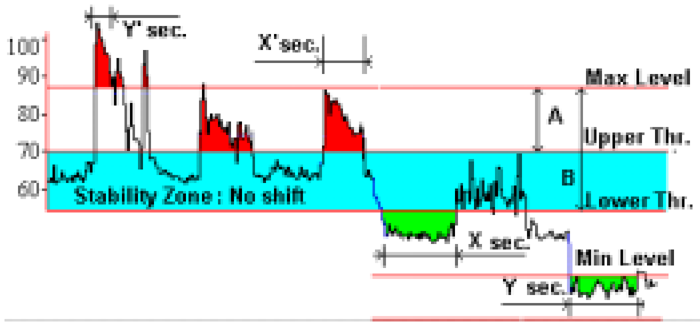
- Activate an automatic gain shift for unattended measurements (**Automatic gain shift** tab)
- Activate automatic calibration and / or calibration check for adapted microphone (**Automatic calibration** tab).
- Define alarm events in **expert mode only** (**Alarm event** tab)

7.6.1. Automatic gain shift tab

dBTRIG32 offers an automatic control option for the dynamic range during the course of measurement. Two control settings are available, automatic or fixed by activating / deactivating the **Enabled** option in the dialog bow shown below.

For automatic control, an underload or overload detection algorithm adjusts the dynamic range automatically.
For a fixed dynamic range (enabled box not activated), the operator chooses a dynamic scale according to the noise level of the measurement. See **paragraph 7.10.**





Shift up if level above :

- **Upper threshold during X' seconds.**
- **Maximum level during Y' seconds.**

Shift down if level below :

- **lower threshold during X seconds**
- **minimum level during Y seconds.**

The principle of automatic control, as a function of the defined parameters is illustrated above. Automatic control enables modification to the dynamic range corresponding to the measured noise levels. Overload criteria are defined, based on minimum and maximum gain levels as well as relative threshold levels with respect to the overload levels.

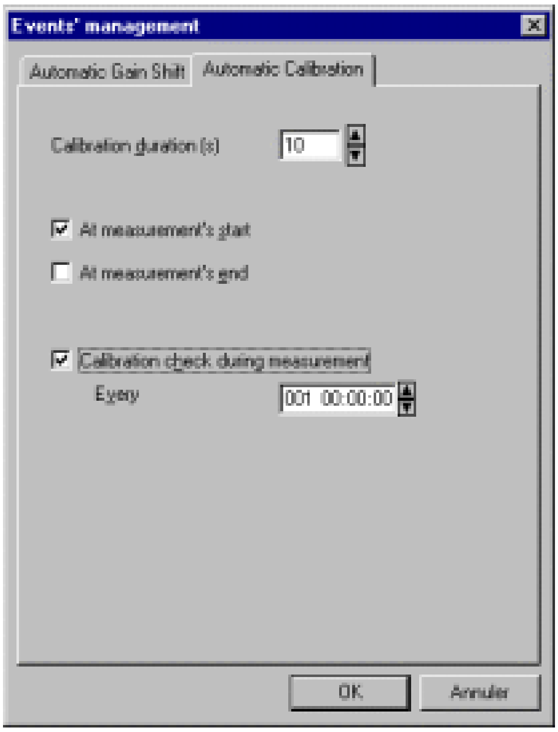
Each time an overload or underload occurs, the data will not be recorded for a short duration corresponding to the time necessary to initialise the amplifiers of the acquisition unit. This initialisation duration is equal to 500 ms when A weighting is selected and to 4 seconds when no weighting (Lin) is selected. The "measured" values during the initialisation period are not taken into account in the logged data and in the digital indicators of the measurement window.

Furthermore, the digital indicators, the overload and underload indicators are reset for each dynamic range shift.

Automatic gain shift allows the user to measure over a wide range of sound levels (typically 20 -140dB. It is strongly recommended to select careful the parameters, as poor settings may result in continuous adjustments and hence important data loss.

*Refer to **chapter 11** for more information on unattended measurements and automatic functions of dBTRIG32.*

7.6.2. Automatic calibration tab



dBTRIG32 manages automatic calibration and calibration check during measurements when an adequate microphone unit is connected to the measurement system.

The dialog box shown aside allows the user to define if automatic calibration is active at the start and/or at the end of the measurement and sets the time period for which it is active (in seconds).

Select as well if a calibration check during measurement should be performed and its periodicity (every day, every hour, etc.)

Access automatic calibration functions in **dBTRIG32**, when the appropriate remote control (to activate an electrostatic actuator) is defined at the hardware configuration stage.

*Refer to **chapter 11** for more information on unattended measurements and automatic functions of dBTRIG32.*

7.6.3. Alarm event tab (expert mode)

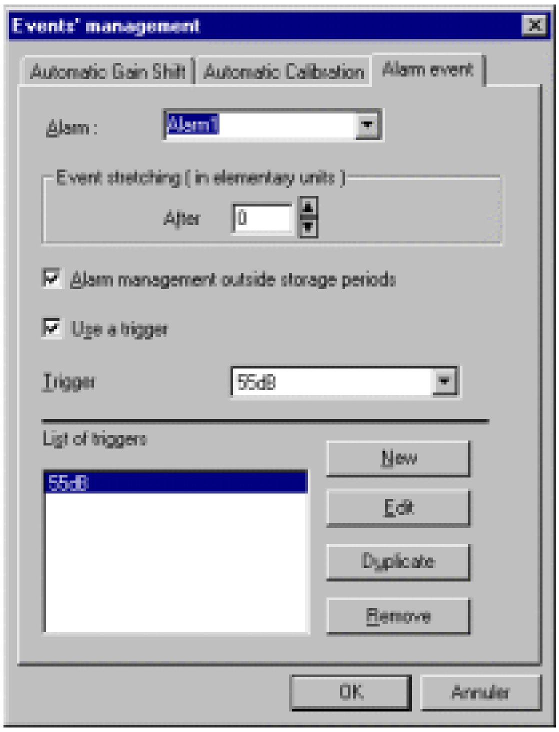
It is possible to define two alarm event that allows the user to generate a light signal when a user-defined threshold is exceeded (e.g.: flashing light system connected to the serial port of the PC enabled when a noise level is exceeded).

Define an **event stretching** in number of elementary units after the true event, knowing that an elementary unit corresponds to the time base of acquisition in **dBTRIG32** (see the acquisition parameters, overall tab, to change this time base).

We therefore can calculate the stretching duration D corresponding to the number of elementary units selected N for a time base t by a simple multiplication : D = N * t.

In the above example, and for a time base of 200ms, an alarm event (for the alarm n°1) will stop 0 second after (0 * 200ms) the duration for which the trigger state is true.

For a given alarm (Alarm 1 in this example) the user may **use a trigger** (threshold) selected in the **list of triggers**. Enabling an alarm signal will be performed automatically when the trigger state is "TRUE".

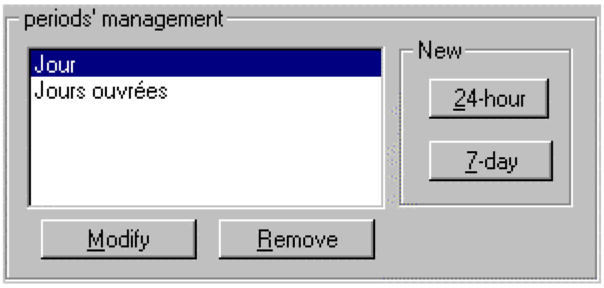


*The definition and configuration of event triggers are dealt with in **paragraph 7.9** for dBTRIG32 expert mode*

*The use of a remote control object to generate an alarm event is dealt with in **chapter 12**.*

7.7. Definition of storage periods

When selecting the start mode of data logging in dBTRIG32 (Overall tab of the storage parameters' dialog box), it is possible to define storage periods without overlapping, over a day cycle or a week cycle.



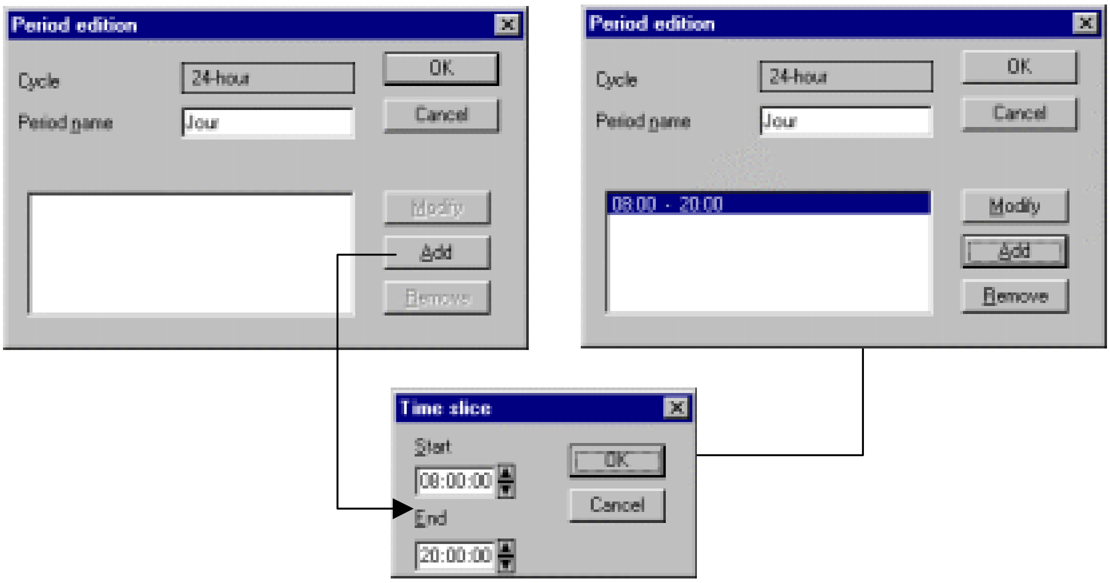
The choice **24 hours** means that it will be repeated identically whatever the day of the week. (For example, no difference will be made between results in the time slice 8h-20h of a week day and a bank holiday day).

The choice **7-day**, means that the software can differentiate between weekdays' periods (for example 8h – 20h) and weekend periods (for example from Friday 18h to Monday 6h).

Several slots may be specified during a period but only one period may be active during measurements.

7.7.1. 24 hour cycle

This section describes the procedure to program a 24-hour cycle.

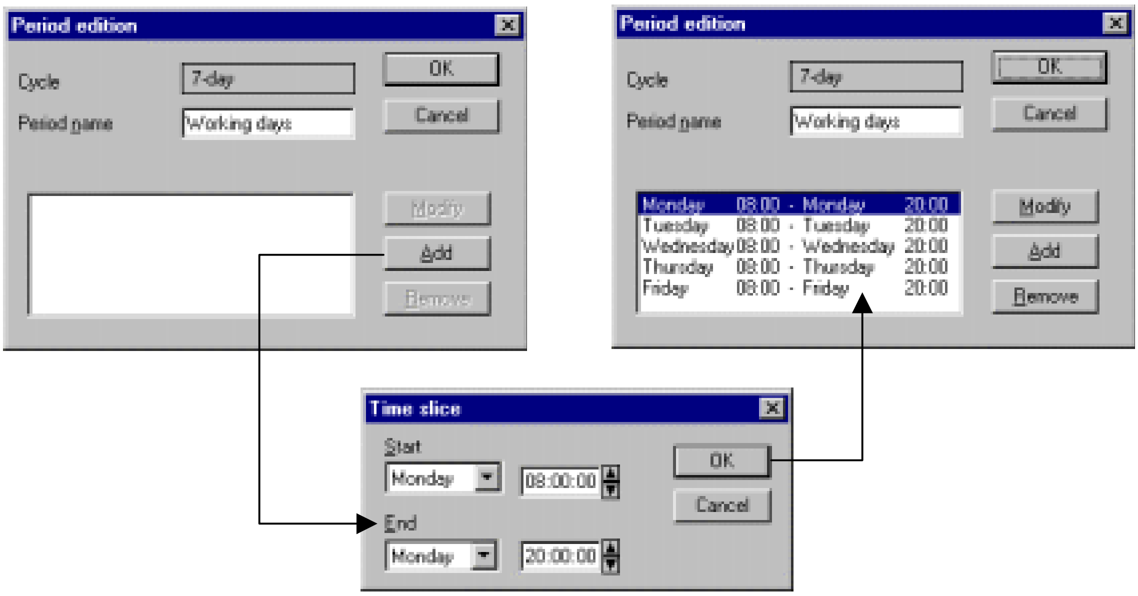


Give a name to the period (for example, day). To accept the time slot, click on **Add**. Use the up/down arrows to specify the exact start and end time. Validate with "OK". The preceding window is re-displayed with the current settings.

Other time slots may be added, while existing entries may be modified or deleted, using respectively the buttons **Modify** and **Remove**.

7.7.2. 7 day cycle

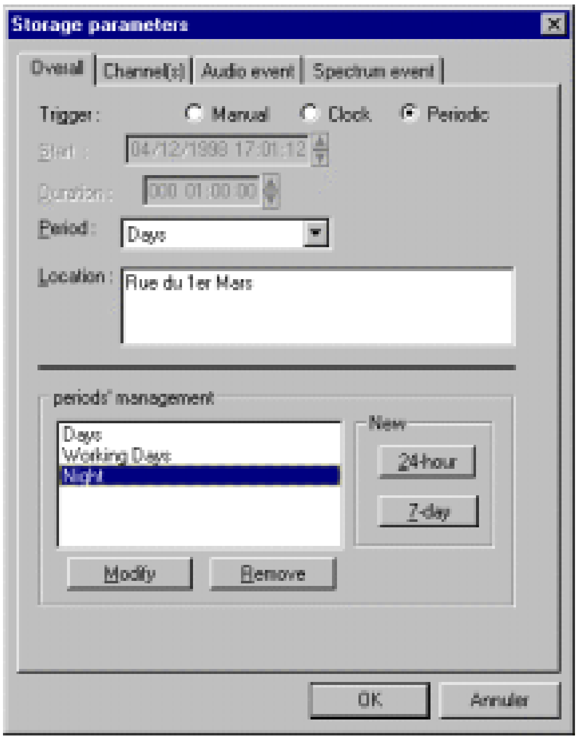
In order to define a 7-day cycle, proceed as follows:



For example, a period named “working days” has a 7day cycle, with time slots set between 8 a.m. and 8 p.m.; from Monday to Friday inclusive.

Add the first time slot: Monday, between 8 hrs and 20 hrs. Access the list of days and increment the time using the arrows adjacent to the boxes.

Validate then repeat this operation for the other days until Friday. The Period Edition window contains 5-hour slots corresponding to the 5 days of the week.



The list of periods is also adjusted The periods are saved by default and are resumed at the next use of the program.

Choose the storage **period** in the list (Overall tab, storage parameters dialog box). Data logging in dBTRIG32 will start and end according to the defined period.

In our example, data logging will occur every day between 08:00 and 20:00 hours.

Note that an overlap of two time slots is not possible; and that from a calculation point of view, two time slots that are concurrent will be considered as one slot.

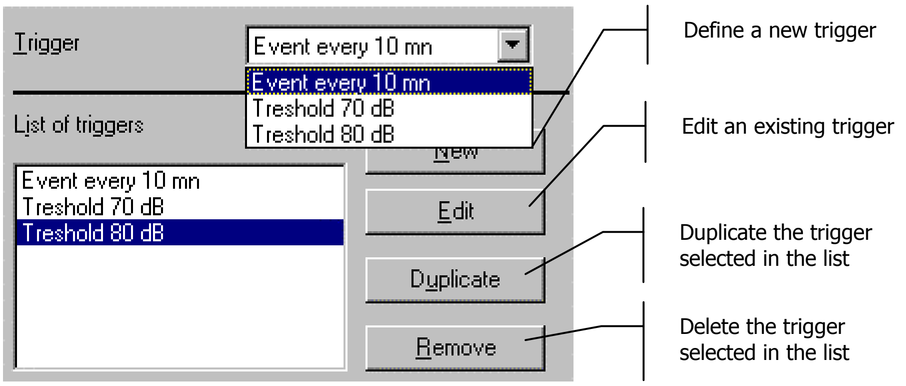
7.8. Event triggers (standard mode)

In **dBTRIG32**, it is now possible to define several trigger conditions (threshold or clock) to automatically acquire an auxiliary event (audio record for example).
Several periods, for which the event acquisition according to a user-defined trigger is activated, can be managed by **dBTRIG32**. For example, define different threshold for day and night time to trigger an audio record.

As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress.

By convention, we call a **trigger** the set of conditions defined by the operator. Several triggers may be defined and used during a measurement session.

In the example below, three different triggers have been defined : a clock trigger (acquisition of an event every 10 minutes), and two threshold triggers (acquisition of an event when the measured Leq level exceeds 80dB and when the statistical indice L10 exceeds 70dB).

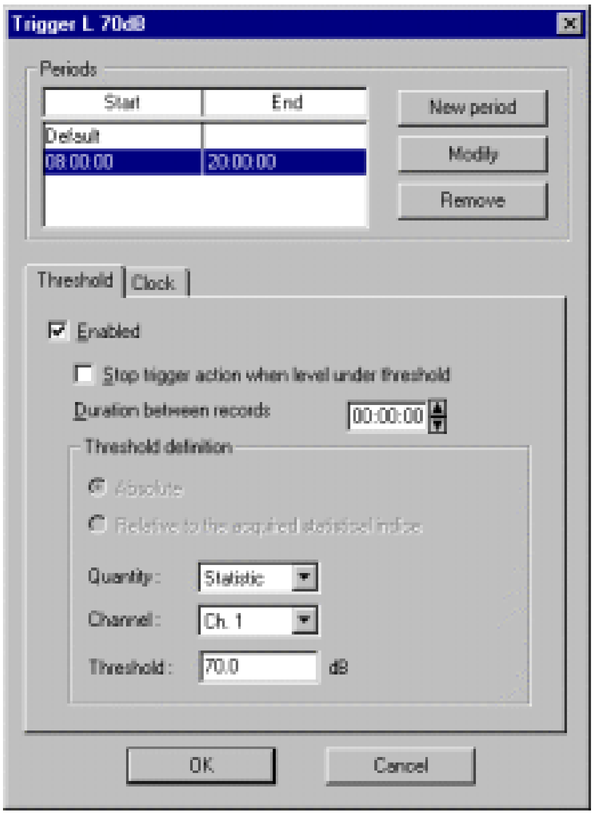


Access this dialog box by the tab Audio record of the storage parameters dialog box (see **paragraph 7.2.3**).

Select now the trigger that will be used to acquire automatically audio events. First, tick the box **Use a trigger** then select the appropriate trigger in the list.

To define a trigger, proceed as follow:

- Access the **Audio event** tab in the **storage parameters** dialog box
- Click on the **New** key to define a new trigger
- Give a name to this trigger
- Define the time periods for which the trigger is active
- Define the trigger conditions (clock and/or threshold) for each period of activity
- Validate the configuration dialog box of the trigger
- Repeat the above operations to define as many triggers as necessary



Here is an example of definition of a trigger. This dialog box is displayed when defining a new trigger or edit an existing trigger (with the **Edit** key).

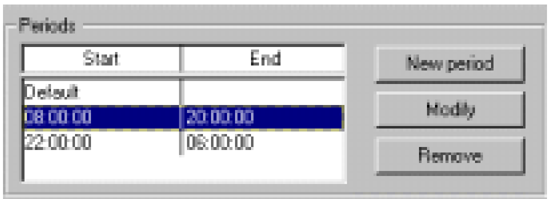
The upper part of the dialog shows the **periods** of activities of the trigger conditions.

The **threshold** tab is used to define threshold trigger conditions.

The **clock** tab is used to define clock trigger conditions.

7.8.1. Periods of activity of a trigger

In the trigger configuration dialog box, edit periods (over a 24-hour cycle) for which the trigger conditions are activated. Alternatively, the user may define different trigger conditions for different times of a day.



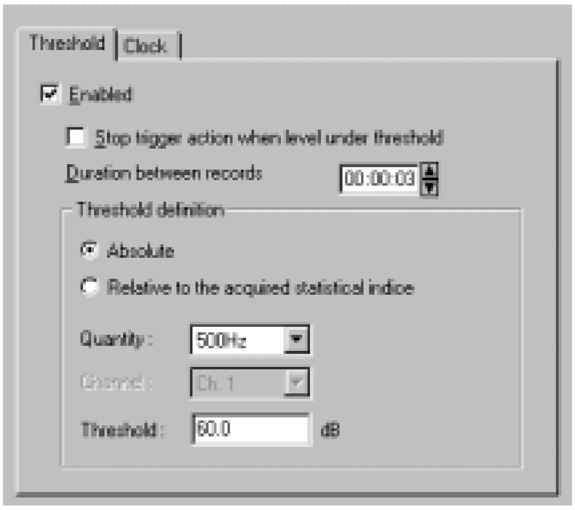
Click on **new period** to define the start and end times (over a 24-hour cycle) of a period of activity.

The trigger conditions defined for a **Default** period are activated when no other periods are defined (in this case, the trigger conditions are valid all the time) or for the time slots not covered by other periods of activity.

📖 *In order to define trigger conditions for each activity period, first select with the mouse a period in the list. The active period is shown in inverse video.*

7.8.2. Threshold trigger conditions

Once the periods of activity of the trigger have been defined, the user may define threshold and/or clock trigger conditions.



In the configuration dialog box, select the **threshold** tab to define a threshold trigger condition.

Tick first the box **Enabled** to activate a threshold condition.

Tick the box **Stop trigger action when level under threshold** to stop the event acquisition when the threshold condition is not fulfilled anymore. For example, audio recording will be stop when the measured level passes below a defined threshold level)

Define as well a minimum **duration between two successive records**.

To **define a threshold** itself, select the following parameters:

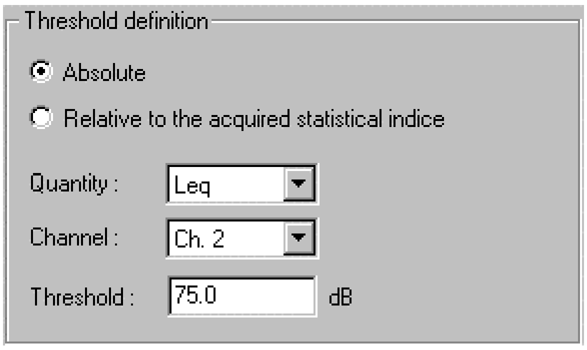
- The type of threshold (**absolute** or **relative to the acquired statistical indice**)
- The measured **quantity** considered defining a threshold level (Leq, Fast, Slow, frequency band, etc.).
- For dual channel measurements, the measurement **channel** considered for calculating the threshold level
- The level in dB of the **threshold** trigger for an absolute threshold, or the difference in dB between the level of the selected measured quantity and the level of the acquired statistical indice for a relative threshold

🕮 *Refer to **paragraph 7.1** for more information on the acquired quantities and the acquired statistical indice*

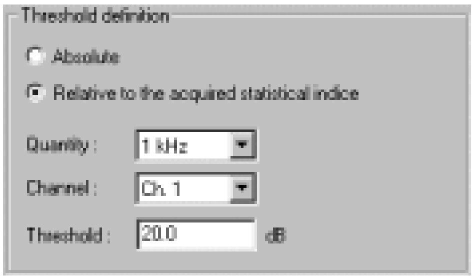
🕮 *It is possible to combine threshold and clock trigger conditions.*

7.8.2.1. Example of absolute threshold

In the example shown below, an event will be acquired when the Leq level measured on channel 2 is greater or equal to 75dB.



7.8.2.2. Example of relative threshold



In the example shown aside, an event will be triggered when the noise level measured in the third octave band centred at 1000 Hz on channel 1 is 20dB greater than the level of the acquired statistical indice.

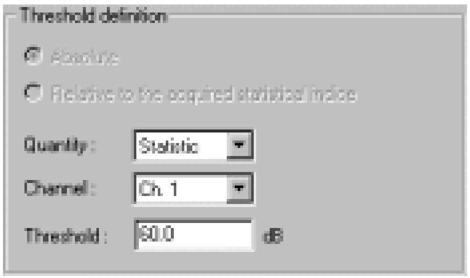
If, for example, the indice L95 is acquired (it is the noise level exceeded during 95% of the time corresponding to the background noise), the threshold trigger will vary according to the background noise level.

For example, for a L95 level of 55dB, an event will be triggered each time the noise level in the 1000 Hz frequency band exceed 75dB, that is 20dB above the background noise.
This function can be useful for environmental noise measurements over day and night periods, when the background noise varies greatly.

7.8.2.3. Example of absolute threshold : particular case

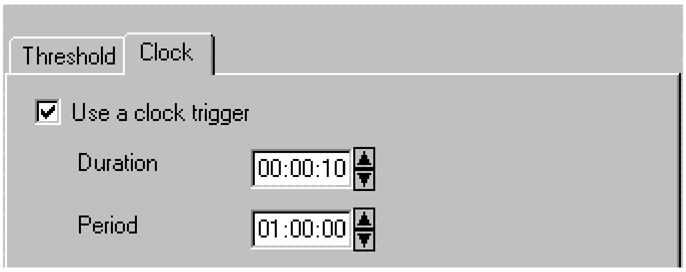
If the user selects the statistical quantity in the list, the fields absolute and relative are greyed.

If for example, the L10 statistical indice is acquired (it is the noise level exceeded during 10% of the time), an event will be triggered when the L10 level is greater or equal to 60dB.



7.8.3. Clock trigger condition

Once the periods of activity of the trigger have been defined, the user may define threshold and/or clock trigger conditions.
In the configuration dialog box, select the **clock** tab to define a clock trigger condition.



To activate clock triggering of events, tick the box **Use a clock trigger** then define the duration of the events and the **periodicity** of their acquisition (format: hour / minutes / seconds).

In the above example, 10 second long events will be acquired every hour since measurement start.

📖 *It is possible to combine threshold and clock trigger conditions.*

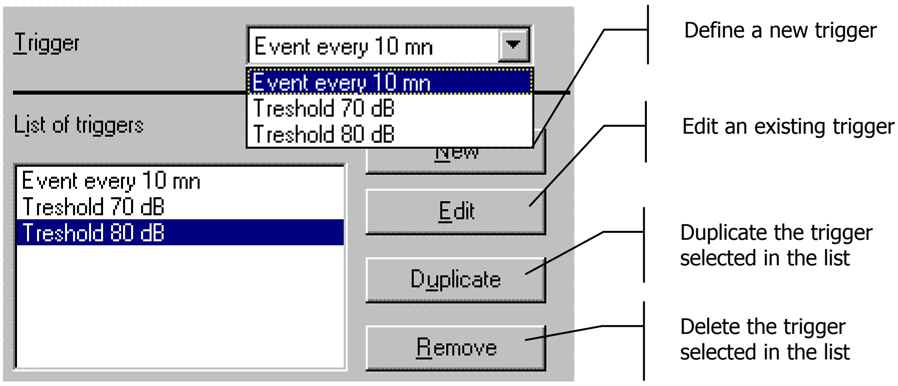
7.9. Event triggers (expert mode)

In **dBTRIG32**, it is now possible to define several triggers conditions (threshold or clock) to automatically acquiring an auxiliary event (audio record, spectrum event, and code event or alarm event).
Several periods, for which the event acquisition according to a user-defined trigger is activated, can be managed by **dBTRIG32**. For example, define different threshold for day and night time to trigger an audio record

As the acquisition of auxiliary events is independent of the measurement of noise quantities, the trigger conditions may be defined at any time, even during a measurement session in progress.

By convention, we call a **trigger** the set of conditions defined by the operator. Several triggers may be defined and used during a measurement session.

In the example below, three different triggers have been defined : a clock trigger (acquisition of an event every 10 minutes), and two threshold triggers (acquisition of an event when the measured Leq level exceeds 80dB and when the statistical indice L10 exceeds 70dB).



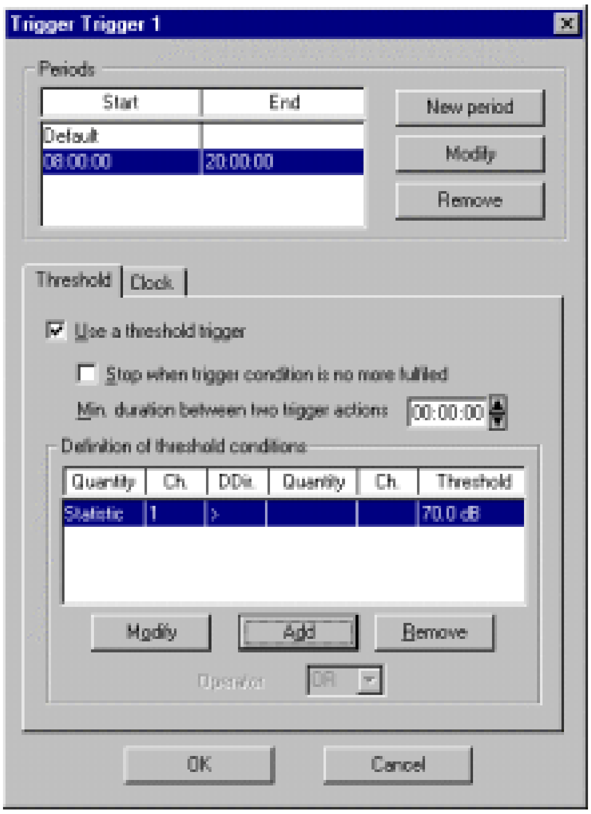
Access this dialog box by:

- The tab **Audio event** (paragraph 7.3.3) and the **tab Spectrum** event (paragraph 7.3.4) of the **storage parameters** dialog box
- The tab **Code event** (paragraph 7.5.2) of the **source coding parameters** dialog box
- The tab **Alarm event** (paragraph 7.6.3) of the **advanced parameters** dialog box

Select now the trigger that will be used to acquire automatically audio events. First, tick the box **Use a trigger** then select the appropriate trigger in the list.

Proceed as follow to define a trigger:

- Access the **event (audio, spectrum, code, alarm)** tab from a dialog box of the measurement parameters
- Click on the **New** key to define a new trigger
- Give a name to this trigger
- Define the time periods for which the trigger is active
- Define the trigger conditions (clock and/or threshold) for each period of activity
- Validate the configuration dialog box of the trigger
- Repeat the above operations to define as many triggers as necessary



Here is an example of definition of a trigger. This dialog box is displayed when defining a new trigger or edit an existing trigger (with the **Edit** key).

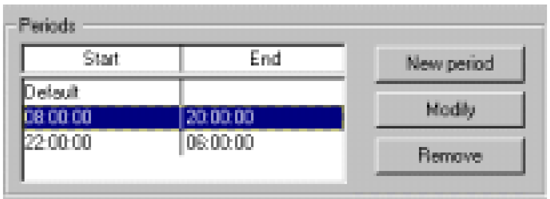
The upper part of the dialog shows the **periods** of activities of the trigger conditions.

The **threshold** tab is used to define threshold trigger conditions.

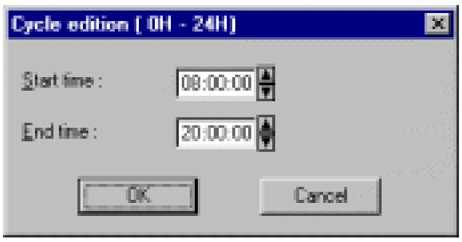
The **clock** tab is used to define clock trigger conditions.

7.9.1. Periods of activity of a trigger

In the trigger configuration dialog box, edit periods (over a 24-hour cycle) for which the trigger conditions are activated. Alternatively, the user may define different trigger conditions for different times of a day.



Click on **new period** to define the start and end times (over a 24-hour cycle) of a period of activity.

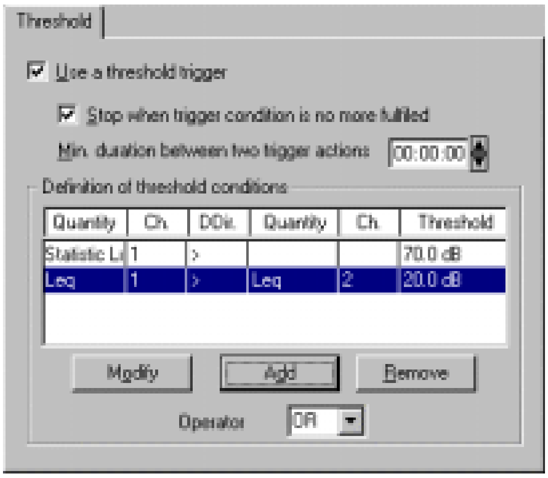


The trigger conditions defined for a **Default** period are activated when no other periods are defined (in this case, the trigger conditions are valid all the time) or for the time slots not covered by other periods of activity.

In order to define trigger conditions for each activity period, first select with the mouse a period in the list. The active period is shown in inverse video.

7.9.2. Threshold trigger conditions

Once the periods of activity of the trigger have been defined, the user may define threshold and/or clock trigger conditions.



In the configuration dialog box, select the **threshold** tab to define a threshold trigger condition.

Tick first the box **Enabled** to activate one (or more) threshold conditions.

Tick the box **Stop trigger action when level under threshold** to stop the event acquisition when the threshold condition is not fulfilled anymore. For example, audio recording will be stop when the measured level passes below a defined threshold level).

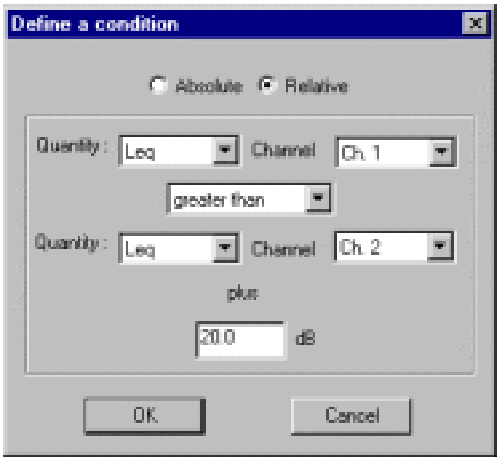
Define as well a minimum **duration between two successive records.**

To define **threshold conditions**, first click on the **add** button to define one condition, then repeat this operation for as many threshold conditions as required.
The keys Modify and remove allow the user to respectively edit or delete the threshold condition selected in the list (appears in inverse video).

When several conditions are defined, a boolean operator (**AND/OR**) has to be defined. A threshold trigger will be activated when all the threshold conditions are fulfilled (**AND operator**) or when at least one condition is fulfilled (**OR operator**).

The condition definition dialog box appears on screen. Define the following parameters:

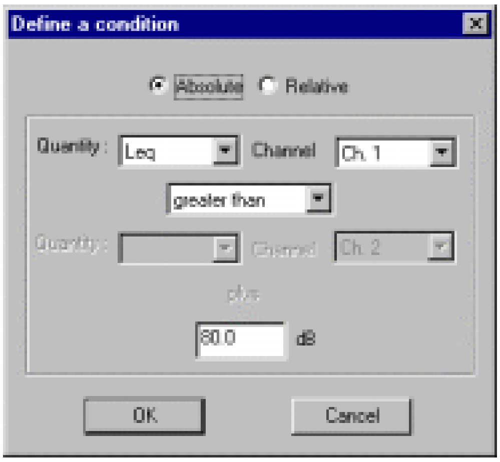
- The type of threshold (**absolute** or **relative**)
- The measured **quantity** considered defining a threshold level (Leq, Fast, Slow, frequency band, etc.).
- The measurement **channel** considered for calculating the threshold level
- The way of the threshold (quantity **greater than** or **less than** the threshold value)
- For relative threshold, the second **quantity** (and associated measurement **channel**) to compare to the first quantity.
- The **threshold** level in dB for an absolute threshold, and the difference in dB between the levels of the first and the second quantities selected.



📖 *Refer to **paragraph 7.1** for more information on the acquired quantities and the acquired statistical indice*

📖 *It is possible to combine threshold and clock trigger conditions.*

7.9.2.1. Example of absolute threshold condition



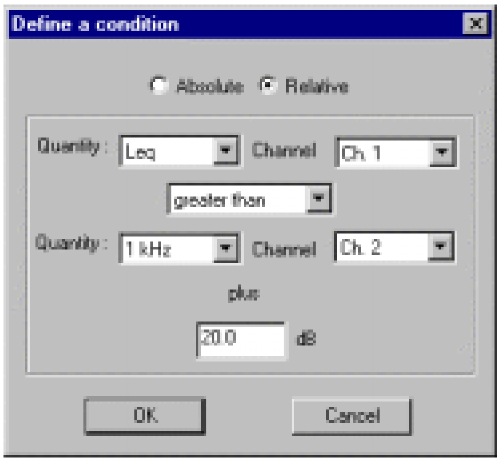
In the example shown aside, the threshold, condition will be fulfilled when the Leq level measured on channel 1 is greater than or equal to 80dB.

If only this condition as been defined, or if the OR operator is selected, an event will be triggered when this threshold condition is fulfilled.

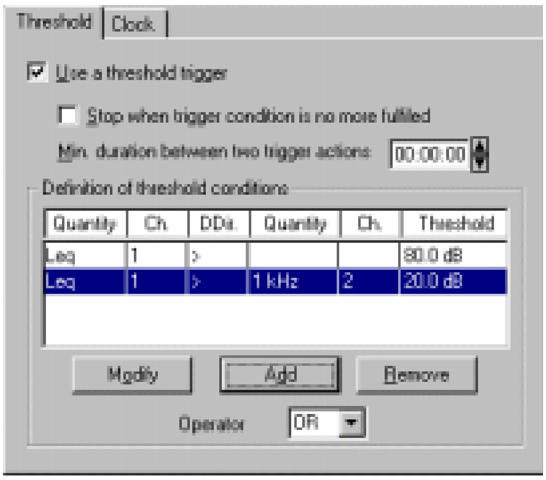
7.9.2.2. Example of relative threshold condition

In the example shown aside, the threshold condition will be fulfilled when the Leq level measured on channel 1 is 20dB greater than the noise level measured in the 1000Hz frequency band on channel 2.

If only this condition as been defined, or if the OR operator is selected, an event will be triggered when this threshold condition is fulfilled.



7.9.2.3. Example of trigger with two threshold conditions

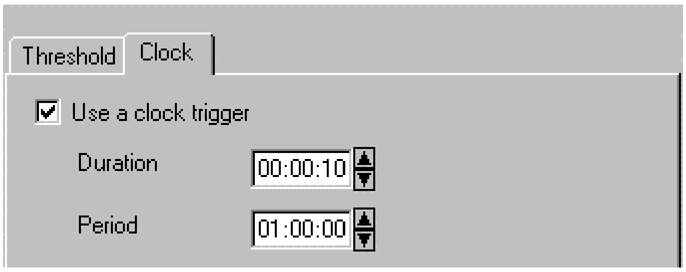


If for a trigger, we define the two previous threshold conditions and if we select the boolean operator OR, an event will be triggered when the first condition is fulfilled (*Leq channel 1 > 80dB*) or when the second condition is fulfilled (*Leq channel 1 > level 1kHz + 20dB*)

*To trigger events according to a **spectral pattern**, the user has to define threshold conditions for each frequency band of interest then to select the boolean operator AND. An event will be acquired when all the threshold conditions are fulfilled.*

7.9.3. Clock trigger condition

Once the periods of activity of the trigger have been defined, the user may define threshold and/or clock trigger conditions.
In the configuration dialog box, select the **clock** tab to define a clock trigger condition.

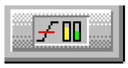


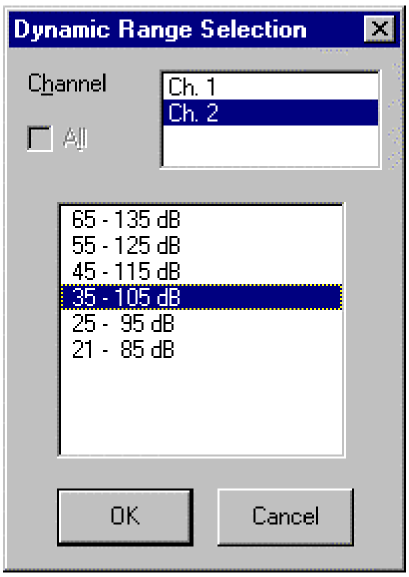
To activate clock triggering of events, tick the box **Use a clock trigger** then define the duration of the events and the **periodicity** of their acquisition (format: hour / minutes / seconds).

In the above example, 10 second long events will be acquired every hour since measurement start.

It is possible to combine threshold and clock trigger conditions.

7.10. Dynamic range parameters

dBTRIG32 allows manual selection of the dynamic range using the icon  (or **Setup** menu **/ Dynamic range**). This command allows the user to select the dynamic range of the measurement as follow.



The number of dynamic ranges available depends upon the hardware platform used. It usually varies between 5 or 6 ranges. It is very important to carefully select the dynamic range as any level measured outside this range will coded as an overload (code 2) or an underload (code 3) and the overload indicators in the command bar of the measurement window will be lit in red.

dBTRIG32 features dynamic ranges of 65dB maximum. It is the greatest value for any one range of 16-bit acquisition hardware that comply with Type 1 specifications of the IEC804.
The suggested dynamic ranges are absolute electrical ranges for a given type of transducer. This means that they are electrically identical to the transducer being used (i.e. the highest range will always cover the transducer being used).

However, given that values are expressed in "acoustical" dB, the final values in the panel next to this will be different depending upon the type and its sensitivity. In effect, the same electrical level expressed in "acoustical" dB for two transducers with different sensitivities will give different levels (in dB).

This initialisation duration when a dynamic range is selected is equal to 500 ms when A weighting is selected and to 4 seconds when no weighting (Lin) is selected. The "measured" values during the initialisation period are not taken into account in the logged data and in the digital indicators of the measurement window.

Furthermore, the digital indicators, the overload and underload indicators are reset for each dynamic range shift.

7.11. Selective configuration tasks

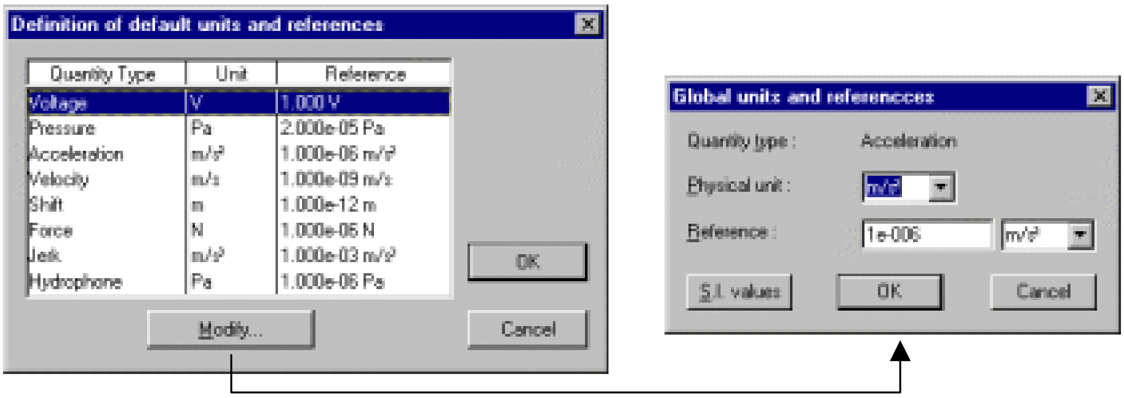
Before setting up a measurement session, the user may perform the following operations in **dBTRIG32**:

- Edition of the physical units and references for calculation of logarithmic levels in dB
- Display line spectra rather than bar spectra
- Define names for the noise source codes

7.11.1. Conversion units and references management (from a physical unit to a dB level)

Use the command **Units' management** under the **Preferences** menu. The following dialog box appears on-screen:

dBTRIG32 allows the user to edit the default references and the units used for each type of transducer.



Click on the **modify** key to edit the physical unit and/or the reference value of the selected quantity in the list (appears in inverse video). Click on **SI values** key to select the values of the International system as standard.

💣 **This function cannot be accessed when a measurement is running. Close the measurement window to edit the references values and the units.**

Let us consider an example of conversion.

For pressure type transducers, the reference value of the SI system is $2*10^{e-5}$ Pa. The sound pressure level is given by the formula :

Lp = 10 * log (p²/p0²) where p0 is the reference value

Therefore, for an acoustic pressure level of $2*10^{e-2}$ Pa, **dBTRIG32** will use a reference of $p0=2*10^{e-5}$ Pa, to calculate a sound pressure level of 60dB (that is $10*\log[2(10^{e-2})^2/2(10^{e-5})^2]$).

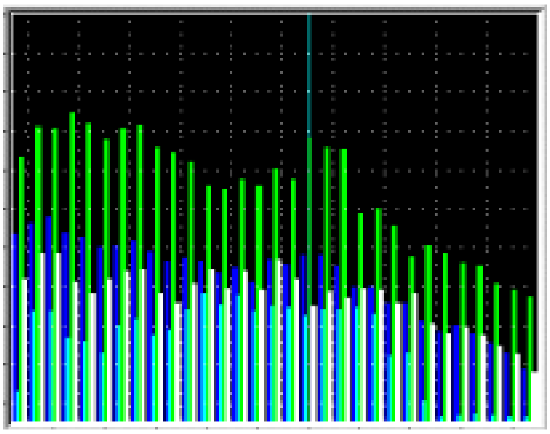
If one define a reference level of $p0=2*10^{e-6}$ Pa, the sound pressure level displayed in **dBTRIG32** will be equal to 80 dB.

Likewise, it will be possible to edit these parameters for each type of transducer, depending on the application.

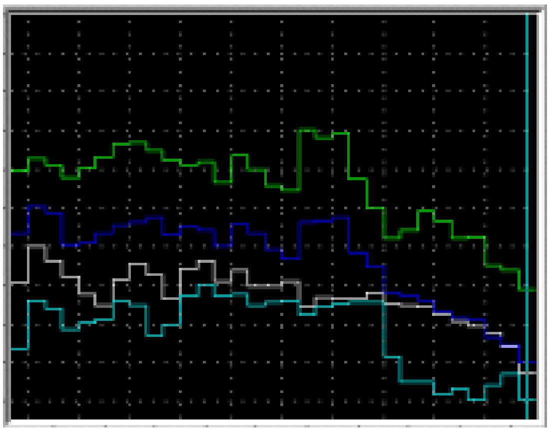
7.11.2. Display line spectra

Use the command **Line spectra display** under the **Preferences** menu.
When this function is activated, the bar spectrum display is changed to a line spectrum display.

This function cannot be accessed when a measurement is running. Close the measurement window to change the spectrum display



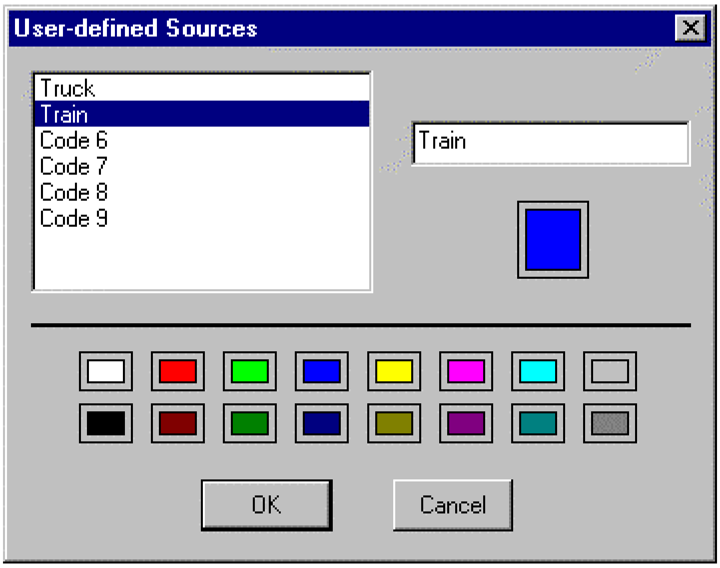
Bar sepctrum display



Line spectrum display

7.11.3. Definition of sources' names

Use the command **Setup / Sources' names** to modify the names and the colours associated with a code number.



dBTRIG32 allows the user to dynamically code noise data. This command is used to change the default names of each code. The name of a source is automatically updated in the horizontal toolbar of the measurement window.

1. Select the source to edit in the list (appears in inverse video) and give a name to it in the right hand corner of the dialog box.

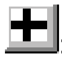
2. The colour associated is displayed in the right hand corner of the dialog box.

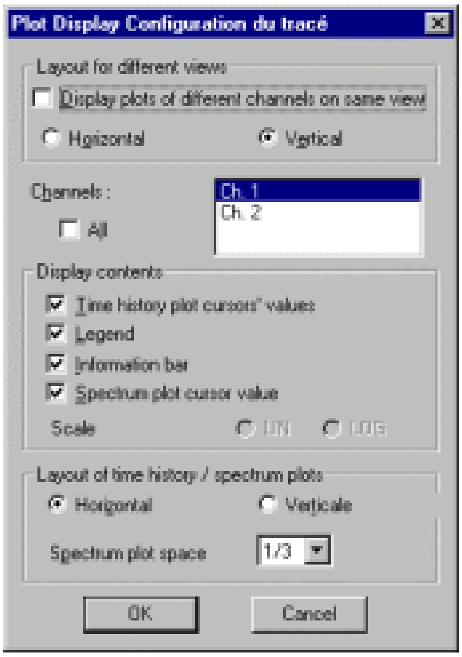
3. Select a different colour in the lower part of the dialog box to change the colour associated with this noise source.

8. MEASUREMENT WINDOW DISPLAY PARAMETERS

The content and general format of the measurement window may be defined by the **Display** menu commands or from the horizontal toolbar. Options include display of cursor values and information bars, time history and spectra readings in landscape or portrait formats, selection of quantities to read, definition of the time period of the displayed time history (e.g. from 30s of data to over an hour), choice of digital indicators, etc.

8.1. Command Display / Layout

The following options are found under the **Display / Layout** menu, or the associated icon :



- **Layout for different views**

Tick the box **Display plots of different channels on same view** if you wish to show measured data from both channels on the same plot. The measurement configuration must identical on both channels (same type of transducers, same frequency weightings, etc.)

The displayed dynamic range will fit the minimum and maximum levels of the two ranges previously used.
As the spectrum plots will be displayed on the same graph, it may be useful to display line spectra instead of bar spectra (See paragraph 7.11.2).

Select the layout for the different views of two measurement channels : **vertical** or **horizontal**.

- **Display contents**

Tick the box **Time history plot cursors' values** to display measured levels at the cursor location on the time history plots.

When several overall quantities are displayed on a time history plot, a cursor value bar is shown for each one of them. Double click with the mouse on this field to select which indicators have to be shown.

Tick the box **Legend** to display noise source names managed by **dBTRIG32** as a legend on the time history plot.

Tick the box Information bar to display the information bars relative to the acquisition of auxiliary events (audio, source and spectrum event bars).

Tick the box **Spectrum plot cursor value** to display measured levels at the cursor location on the spectrum plot. . When several spectra are displayed on a spectrum plot, a cursor value bar is shown for each one of them. Double click with the mouse on this field to select which indicators have to be shown.

If you chose to display and acquire data in physical units (see paragraph 7.1.1), you may select the type of scale (**linear** or **logarithmic**) to use for both time history and spectrum plots.

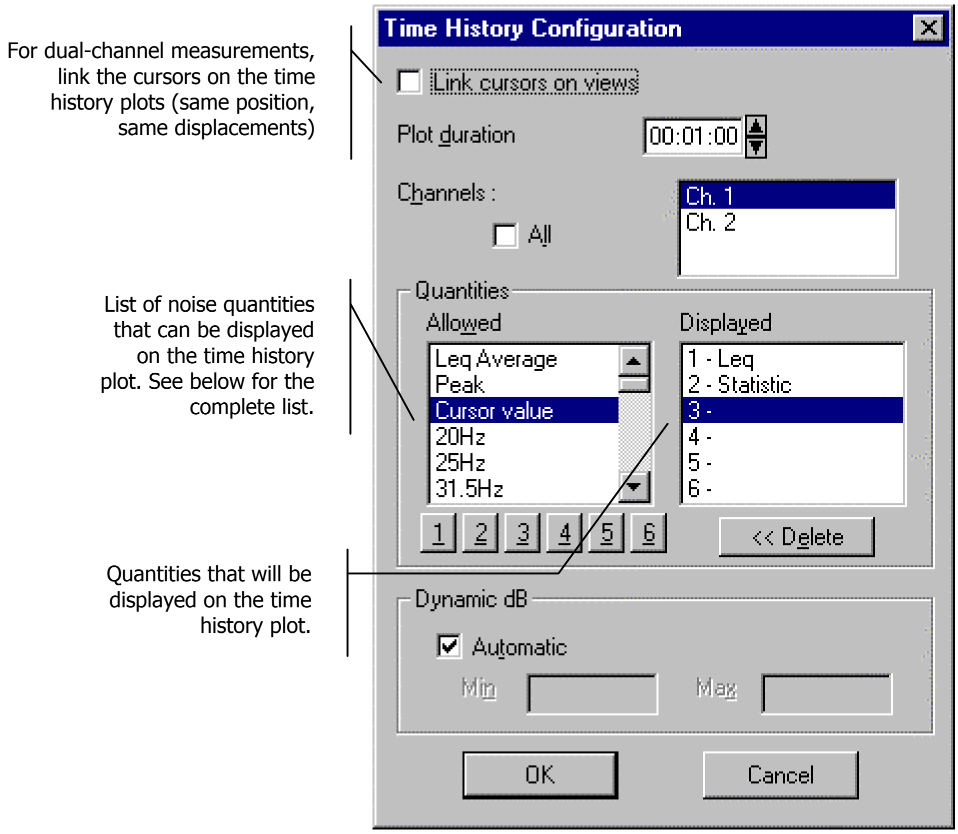
If the measurement window is too small, most of these display indications are not shown, in order to clearly read the graphics.

- **Layout of time history / Spectrum plots**

Define if the plots, for a given measurement channel, have to be shown side by side (**horizontal**) or on top of one another (**vertical**). The **spectrum plot space** option is used to define how much space in the display is reserved to the spectrum plot.

8.2. Command Display / Time history

Use this option to define the time history display characteristics:



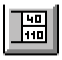
Define the **plot duration** shown on the time history plot (format : hour / minutes / seconds).

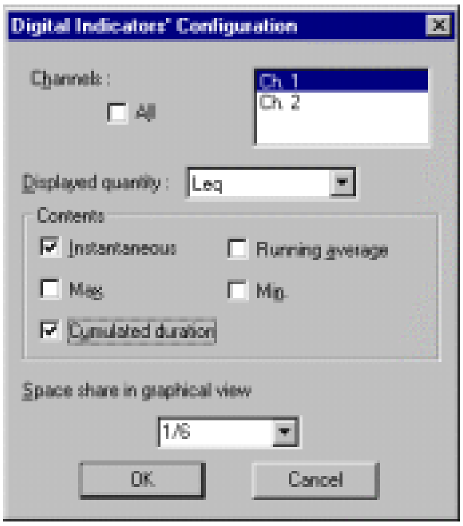
Furthermore, for each active measurement channel (or all of them), define:

- The time history plots **displayed** to choose from a list of **allowed** quantities. Use the keys **1, 2, 3, 4, 5 and 6** to choose, in a given order, the plots to display. The **remove** key is used to delete a quantity from the displayed plots list.
- The display dynamic range: **automatic** (the display dynamic range on the time history plot is adjusted to the values of the measurement dynamic range), or manual (define the **minimum** and **maximum** values to display on the plot)

The list of allowed quantities that can be displayed correspond to the list of acquired quantities (see **paragraph 7.1.**)

8.3. *Command Display / Digital indicators*

Use this command  to define the numerical indicator characteristics, for one or all the active measurement channels:



- **Displayed quantity**

The quantity will be shown in the digital indicators. The list of quantities to choose from correspond to all the quantities displayed on the time history plot (see paragraph 8.2).

- **Contents**

The type of indicators is: **instantaneous, running average, maximum, minimum** and the **cumulated duration**.

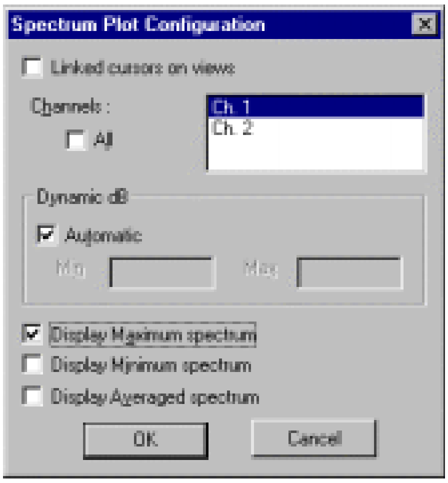
These indicators are reset when the reset indicators command is used.

- **Space share in graphical view**

This option is used to define how much space in the display is reserved to the indicators. In the above example, the digital indicators will take a 1/6th of the space reserved to the time history plot, and the graphical view will take 5/6th of that space.

8.4. *Command Display / Spectrum*

Use this  command to define the spectrum display characteristics, for one or all the active measurement channels:



Tick the box **Linked cursors on views** to link the cursors on the spectrum plots of each active measurement channels.

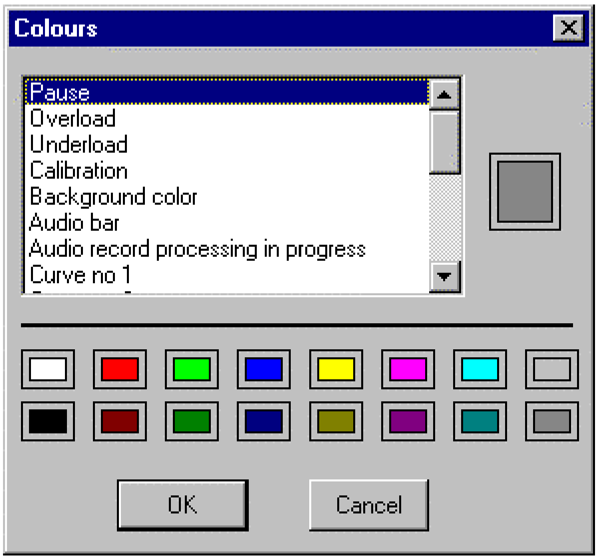
The display dynamic range can be configured in two ways: **automatic** (the display dynamic range on the spectrum plot is adjusted to the values of the measurement dynamic range), or manual (define the **minimum** and **maximum** values to display on the plot)

3 other spectra may be displayed: **maximum / minimum / averaged spectra.**

These spectra are reset when the reset indicators command is used

8.5. Command Preferences / Colours

Use this command to set-up the colours of the different elements displayed in the measurement window.

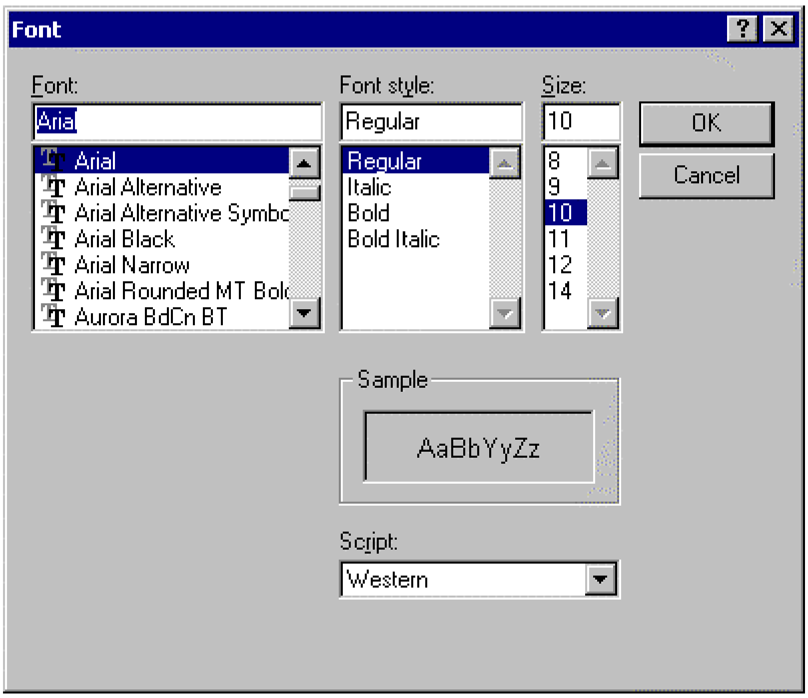


The colour of the element selected in the list (it appears in inverse video) is displayed in the right part of the dialog box. Select another colour (pick from the colours available in the lower part of the dialog box) and valid by OK.

This colour scheme is global for the display and will be applied to all the plots. On validation of this dialog box, the new colour scheme to all the existing plots.

8.6. Command Preferences / Font

Use this command to select a different font for data display in the software.



9. RECORDING A MEASUREMENT

After measurement set-up and display configuration, data logging of the acquired quantities may begin. The data will be saved to the hard disk on the microcomputer in a measurement session file.

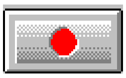
9.1. Command Measurement / New filename

This command is used to define a generic filename for the measurement session file. **dBTRIG32** allows the user to log data, using storage periods, by keeping the same file root name.

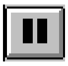
A dialog box is displayed for the user to define the file name of the measurement session (*.CMG). Any binary file (.BID) attached to the measurement session file is stored in the same directory on the hard disk. If no file name has been defined, the software will automatically prompt the user for this definition.

9.2. Measurement : Start / Pause / Stop

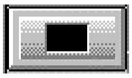
The start, stop and pause instructions can be activated either from the command bar to the right of the measurement window or from the **Measurement** pull down menu. The icons are shown below:



Start data logging (CTRL + B)



Switch to Pause (free run) mode and inversely



Stop data logging (CTRL + E)

See paragraph 2.4 for more information on Pause and storage modes in dBTRIG32

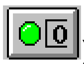
9.3. Actions during a measurement session

9.3.1. Comments

The comment window ( or command **Measurement / Comment**) may be displayed at any time during measurement. Select the **Start** and **End** keys to associate comments and the time they were made to a particular section of the measurement.

The comments are directly accessible from an information bar during the analysis stage of **dBTRAIT32**.

9.3.2. Reset the indicators

Use this command (**Measurement / Reset indicators** or ) to reset the digital indicators, the overload and underloads flags and the elapsed time. The reset command can be applied at any time: it does not affect data logging.

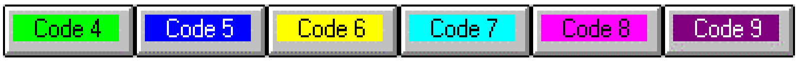
The quantities calculated in the digital indicators display is independent from data logging. Calculations restart each time a rest command has been used.

9.3.3. Dynamic noise source coding while measuring

The use of dynamic coding to identify a noise source is a very powerful tool. The user is able to compile several noise source identifications during the acquisition phase or later during analysis of the results.

During measurement, the coding algorithm determines the number of appearances for each identified noise source. This enables the contributions of each source with respect to its overall input to be appreciated.

dBTRIG32 offers time event coding during the measurement. **dBTRIG32** automatically identifies and codes values that cause over or under-loading. In addition to this, the program allows 6 sources to be coded by the user. Dynamic coding may be operated from the horizontal toolbar of the measurement window or by using the F4 to F9 keys. To end stop dynamic coding, press the key or icon corresponding to the active source.



The icons for 6 time event codes are illustrated above. Their colour and names may be changed to the user's preference **Preferences / Colours** and **Set-up / Source names.**

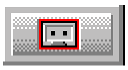
The first stage in coding is to define which noise sources to consider. A source is defined by an identifier (a series of characters) and an internal identifier (numerical code). While a noise source is present, a section of the measurement is cut in order to act as an identification template.

When the noise source is identified, the following results are calculated:**: Particular Leq, Partial Leq partial, SEL and LMAX** and, by way of graphical display: **Time history, Histogram**

*Consult the **dBTRAIT32** manual or the on line help file for more details on how to use the coding facility*

9.3.4. Manual trigger of audio recording during measurement

An audio recording, independent of current threshold or clock trigger settings, may be initialised by this command (**Measurement / Audio record**). As soon as data logging starts the following icon is available for use:

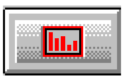


Start / Stop audio recording manually (CTRL+A)

*If the button is greyed, it may mean that audio storage has not been activated. See **paragraphs 7.2.2 and 7.3.2.***

9.3.5. Manual trigger of spectrum event during measurement (expert mode)

A spectrum event, independent of current threshold or clock trigger settings, may be initialised by this command (**Measurement / Spectrum record**). As soon as data logging starts the following icon is available for use:

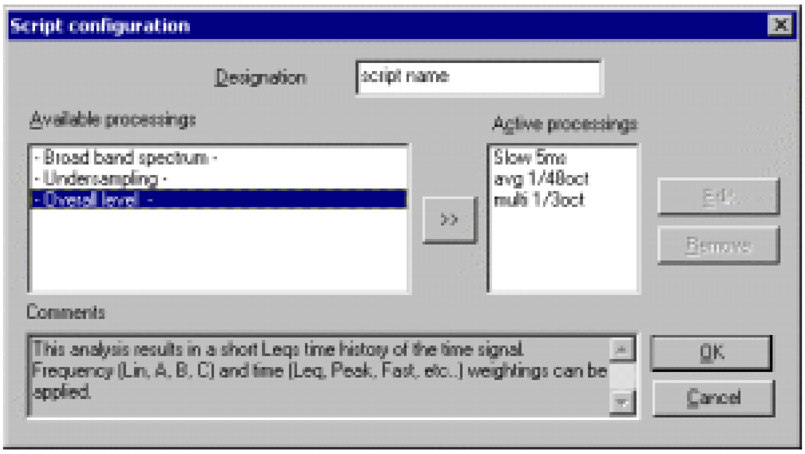


Start / Stop spectrum event recording manually

*If the button is greyed, it may mean that audio storage has not been activated. See **paragraph 7.3.2***

10. ONLINE ANALYSIS OF AUDIO RECORDS (OPTION)

It is possible to user to analyse, in both frequency and time domains, audio records made with the software **dBTRIG32** either automatically at the time of acquisition in **dBTRIG32** or at the processing stage in **dBTRAIT32**, using the multitasking capabilities of Windows:



The analysis types are:

- Frequency analysis (averaged spectrum over the audio event duration) in octaves, 1/3rd octaves
- Frequency analysis in 1/6th, 1/12th, 1/24th and 1/48th octaves **(option)**
- Decimation or deletion of audio records to optimise audio storage capacity on the PC hard disk.
- Multispectrum analysis of audio records (spectrum time history and A, Lin overall levels) **(option)**
- Detailed time history (A, C and Lin in parallel, several time constants - Fast, Slow, Impulse, etc. – using very small time step,) **(option)**

You can then evaluate tagged noise data using **dBTRAIT32** - which permits detailed analysis of the noise climate in both time and frequency domains.

All these analyses are available through an **operation (script) server**, common to all 32-bit 01dB application software packages. The definition and configuration of such analyses can be accessed through the command **Aux / Frequency analysis & advanced processing / Configure** in **dBTRAIT32** and **Audio Analysis / Configure analysis**, key **scripts' edition** in **dBTRIG32**. The dialog box shown above is displayed on-screen.
The user then defines an analysis script that consists of a list of processings that will be applied to each audio event. For each type of processing (defined also as an operator), the user sets various parameters.

This utility saves time as you will not have to process the audio recordings manually any more. Furthermore, the multispectrum analysis is ideal for tonal components, which vary in frequency and level.

📖 *If the option cannot be accessed, it means that the CMG datafile does not contain any audio recordings.*

Once the analysis script is configured, it will execute **automatically** in the background each time an audio event has been recorded.

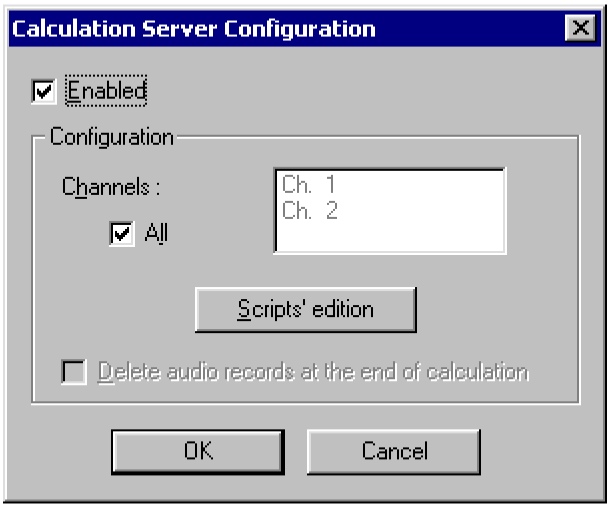
In the audio information bar of the measurement window, the colour of an audio event changes when being processed (by default, an audio being analysed changes from a yellow colour to a red one).

It is possible to stop manually the analysis process by using the command **Interrupt analysis** in the menu **Audio analysis.**

The following paragraphs deal with the configuration of each type of analysis, the activation of the script server and the principle of analysis script.

10.1. Activation of the analysis script server

Activate the calculation server by using the command **Audio analysis / Configure analysis**. The following dialog box is displayed on-screen:



This dialog box allows the user to activate the calculation server an edit an analysis script for audio events' analyses.

The following parameters have to be defined :

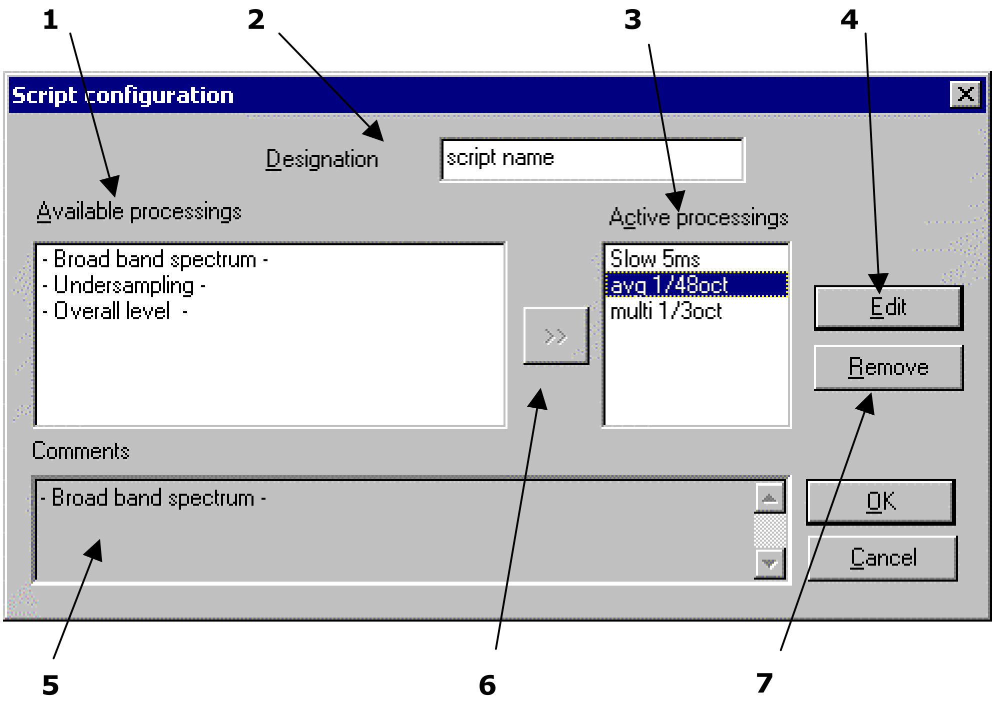
- **Enabled** : tick this box to activate the calculation server.
- **Channels** : select the measurement channels on which recorded audio events will be analysed. If the box **All** is ticked, it means that identical audio event analyses will be performed on all the active measurement channels. If this box is not ticked, the user may define an analysis script for each active measurement channels.
- **Scripts' edition** : Click on this key to define an analysis script (select the type of analyses and define the calculation parameters).
- **Delete audio records at the end of calculation** : tick this box if you do not want to keep audio events after they have been analysis (No replay)

Once a script is defined (and configured), click on **OK**.

When **dBTRIG32** records an audio event, the calculation server will automatically launch an analysis script, and process audio records as soon as they have been acquired. By default, the last script used is recalled when another measurement session is started.

10.2. How to define and configure a script

In order to define the list of processings to apply to audio records, access this dialog box (command **Aux / Frequency analysis & advanced processing / Configure** in **dBTRAIT32** and **Audio Analysis / Configure analysis**, key **scripts' edition** in **dBTRIG32**):



1. List of **available processings**: it shows all the types of processings compatible with the software packages **dBTRIG32** and **dBTRAIT32**. The same operator can be used several times with different parameters to process the same audio recording.
2. **Designation**: give a name to the script.
3. Set of **active processings** currently part of the script. These operators will be activated to analyse independently a set of audio events.
4. **Edit** the parameters of the selected operator (in the list of available processings). Each operator shows its own configuration dialog box.
5. Comments describing the operator selected in the available processings' list. If an operator from the active processings' list is selected, the comment field shows the type of the operator.
6. **Key >>** that allows adding an operator into the active processings' list. Different operators of the same type may be added (the user may for example define a multispectrum analysis in third octave band with different time steps). A generic name for each operator is given by default.
7. The key **remove** allows the user to remove an operator from the list of the active processings.

Once the list of active processings has been defined, edit each one of them to define the calculation parameters.

10.3. Configuration of each active processing.

Once the list of active processings has been defined in the script configuration dialog box, define the calculation parameters of each operator.

To do so, select an operator in the list and click on the **Edit** key.

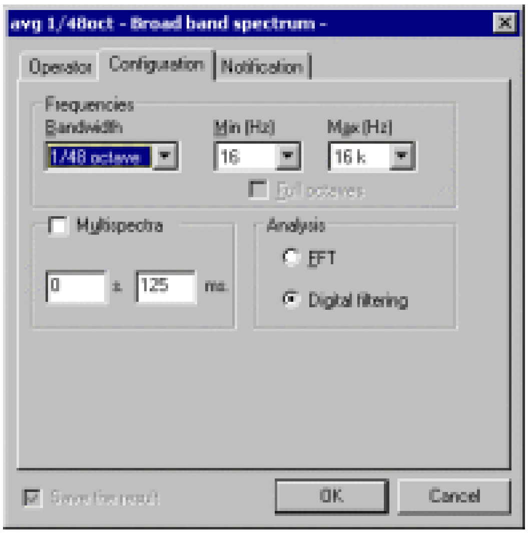
An operator applies a given processing on a quantity or a set of quantities (audio data in the present application software). Each type of operator has its own name and a set of analysis parameters. Furthermore, The progress of each calculation is shown to the user at user-defined intervals.

In **dBTRIG32** and **dBTRAIT32**, the following operators may be selected:

- Broad band spectra (averaged spectra et multispectra)
- Audio record undersampling
- Detailed time history of an overall level

10.3.1. Broad brand analysis (averaged spectra and multispectra

In the script configuration dialog box, select an operator of type - **octave and third octave spectra -** or, if you have the 1/N octaves option, **- Broad band spectrum -** in the list of available processings and place it in the list of active processings. On operator of name FC is then created.
Select this operator and click on the **Edit** key. The dialog box shown below is displayed on screen. The tab **Operator** gives a detailed description of the operator and allows the user to define a more specific name for it.



In the tab **Configuration**, the following analysis parameters may be defined:

- **Bandwidth** to define the spectrum bandwidth for the analysis: Octaves and 1/3 octaves by default, 1/1, 1/3, 1/6, 1/12, 1/24 et 1/48th octaves in option.
- **Min** : Minimum centre frequency of analysis
- **Max** : Maximum centre frequency of analysis
- **Full octaves:** If this option is activated, the software will set the minimum and maximum frequency limits so that only full octaves are calculated.
- **Analysis**: Select the method used to perform frequency analysis: FFT or digital filtering.

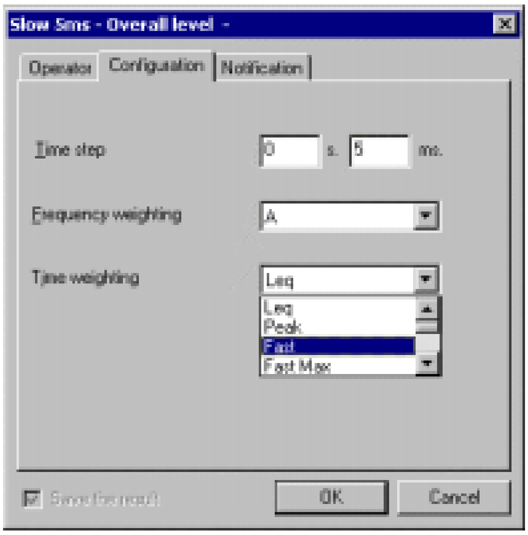
- **Multispectra**: If this option is activated, the user will obtain the time history over the event duration of the calculated spectrum. The time step may vary from 1ms to 1 s.

Furthermore, The choice of maximum frequency for the analysis will depend upon the sampling frequency (approximately twice the measurement pass band) at which the measurements have been taken from **dBTRIG32**: if a pass band of 10 kHz was defined for audio records, frequency bands greater than this cannot be obtained.

10.3.2. Overall level analysis (detailed time history)

A detailed time history analysis offers a zoom facility for the time history data over the duration of the event.

In the script configuration dialog box, select an operator of type **- Overall level -** in the list of available processings and place it in the list of active processings. On operator of name LEQ is then created. Select this operator and click on the **Edit** key. The dialog box shown below is displayed on screen. The tab **Operator** gives a detailed description of the operator and allows the user to define a more specific name for it.



In the tab **Configuration**, the following analysis parameters may be defined:

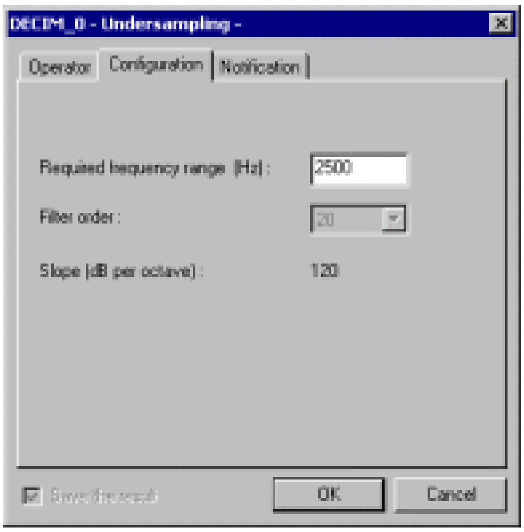
- **Time step**: it is the integration time for the calculation of an overall Leq level. It may vary between 1 ms and 1s.
- **Frequency weighting** : select a frequency weighting to calculate the overall level (A, B, C, Lin)
- **Time weighting** : select the overall noise quantity to calculate (Leq, Peak, Slow, Fast, Impulse)

If, for example, a time step of 1 s is chosen for a 2 s long audio recording, only two values will be calculated (one value per second). The result will therefore be useless. It is recommended to specify as small a step as possible without increasing calculation time too much.

Excessive calculation time may cause problems if the computer has insufficient power. This should also be remembered with respect to the number of quantities and weightings involved in the calculation.

10.3.3. Undersampling

In the script configuration dialog box, select an operator of type **- Undersampling -** in the list of available processings and place it in the list of active processings. On operator of name DECIM is then created. Select this operator and click on the **Edit** key. The dialog box shown below is displayed on screen. The tab **Operator** gives a detailed description of the operator and allows the user to define a more specific name for it.



In the tab **Configuration**, the following analysis parameters may be defined:

- **Required frequency range**: define the maximum frequency of the audio event after undersampling.

It is recommended not to undersample audio files below 2000 Hz unless source recognition, by playing back the file, is unimportant.

The size of audio recordings, in term of memory space, depends on the pass band declared in the configuration box. A table shows the size in kilobytes of audio records as a function of the pass band.

Even if audio files have been analysed at the time of acquisition, you may process them again, with different parameters, in **dBTRAIT32.** It is therefore recommended not to undersample the audio files in **dBTRIG32** since the sampling rate becomes the upper band limit for a further analysis.

The audio file decimation factor is equal to the ratio of measurement frequency and under-sampling rate. Be aware that only whole integers multiples of 2 or 5 are taken into account. The following results are valid for audio records of 10 s long with a passband of 20 kHz. The equivalent sound pressure level measurements are integrated over a period of 500 ms.

Frequency pass band (Hz)	decimation factor	decimated file sizes (kilobytes)
20000 ⇒ 10001	0	1001*
10000 ⇒ 5001	2	501
5000 ⇒ 4001	4	251
4000 ⇒ 2001	5	201
2001 ⇒ 1001	10	101
1000 ⇒ -	-	63 **

* original size of the file
** minimum size

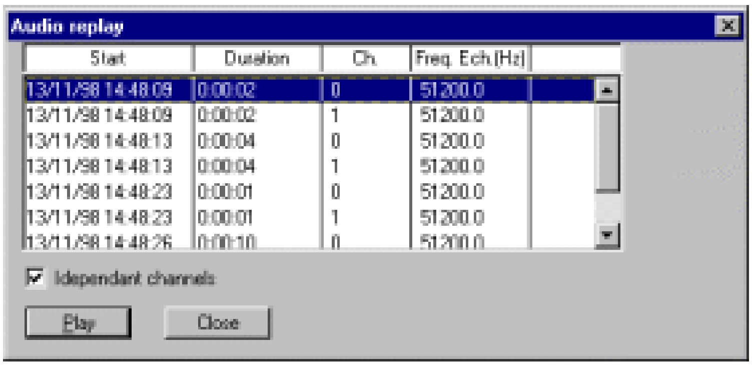
To playback decimated audios file, unless the same hardware is used for recording and playback of the audio signal, some problems may be encountered.

This is due to the sampling rate of standard multimedia cards (44,1 kHz standard) which is inferior to **SYMPHONIE** system- when a 20 kHz passband is chosen (f_e = 51.2 kHz). A file decimation factor of at least 2 will be used to play them back on a standard multimedia card.

The minimal undersampling passband for 01dB hardware is 1414.21 Hz. Any signal decimated below this frequency cannot be played back. Typically, a 2 kHz cut-off frequency is also the minimum required to accurately identify events in environmental noise.

10.4. Replay audio records

It is possible to replay directly audio records in **dBTRIG32** on the multimedia sound system of the computer (**SYMPHONIE, JAZZ** outputs or built-in sound card, connected to a loudspeaker) by using the command **Replay audio** in the audio analysis menu. The following dialog box is displayed on-screen.



Select in the list the audio records to playback then click on the **Play** key.

The option Independent channels (for dual channel measurements) allows the user to play back separately the audio events recorded on channel 1 and channel 2. If this option is not activated, both audio events will be played back of the output channels (on each loudspeaker connected to the dual channel output).

Play back of audio records is not possible when an acquisition (with data logging) is in progress)

11. UNATTENTED MEASUREMENTS : AUTOMATIC FUNCTIONS OF DBTRIG32

dBTRIG32 features many functions useful to perform unattended measurements over long periods of time from a few days to several years of continuous acquisition).

The following operations (described in this chapter) can be performed :

- **Automatic calibration** and / or **calibration check** for adequate microphone units.
- **Auto reboot facility** of the complete measurement system in case of power shortage.
- **Automatic adjustment of the dynamic range** as a function of the measured sound levels.
- Use of a **single extended dynamic range of 115dB**

11.1. Automatic calibration - Calibration check

This paragraph presents the principle, the configuration and the use of the automatic calibration and calibration facilities.

11.1.1. Principle and procedure

dBTRIG32 offers **automatic calibration** and **calibration check** via a remote control object and a suitable acquisition head (microphone unit with a built-in electrostatic actuator for automatic calibration and / or microphone preamplifier allowing a calibration check by voltage insertion or Phantom reference).

The following actions are performed when these functions are activated :

For a transducer that supports actuator calibration

- Activation of a calibration signal implanted in the acquisition head by a remote control object and or a user-defined period.
- The program switches to calibration mode for a defined calibration duration and re-computation of the measured values if there is a difference less than 3dB with respect to the last calibration results. Adjust transducer sensitivity within that range.
- If the transducer sensitivity drift is too far, the user must manually calibrate the system and check hardware elements. See **chapter 5**.

For a transducer that supports calibration check by voltage insertion (Phantom reference)

- Activation of a reference voltage (corresponding for example to 90 dB at 1 000 Hz) on a given pin of the preamplifier connector for a user-defined period. This voltage (or Phantom reference) is usually generated by the acquisition hardware (e.g. SYMPHONIE)
- The program switches to calibration mode for a defined duration. This operation is repeated periodically at user-defined intervals.
- If the transducer sensitivity drift is too far, the user must manually calibrate the system and check hardware elements. See **chapter 5**.

During a calibration check, **dBTRIG32** switches to Pause mode. If a drift occurs, the data values may be adjusted in **dBTRAIT32** by applying a correction factor.

The following procedure describe how to activate automatic calibration and calibration check in **dBTRIG32**:

Measurement chain hardware configuration

1. Define adequate transducer and calibrator in the utility dBCONFIG32
2. Select an appropriate transducer / calibrator couple, and activate the Phantom reference signal conditioning option of the hardware platform in the hardware configuration of dBTRIG32.
3. Select and configure an appropriate remote control object at the hardware configuration stage in dBTRIG32 (for automatic calibration only)

Software configuration

4. Use the command **Setup / Parameters / Advanced parameters / Automatic calibration** tab to configure the calibration duration, the periodicity of calibration checks, etc.

Automatic calibration is then performed automatically at measurement 'start and/or measurement's end. Calibration check is then performed automatically at user-defined intervals or manually by using the command **Measurement / Calibration check** during a measurement session.

These functions cannot be used with all types of microphone units on the market. Please, refer to the list of transducers that can be used for automatic calibration and calibration check before attempting to configure these functions.

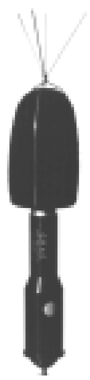
11.1.2. List of transducers that can be used

The table shown below gives the list of transducers that can be used for automatic calibration at measurement 'start and end and for calibration check at user-defined intervals.

Transducer	Automatic calibration	Calibration check
41AM and **41CM** with power supply and calibration interface box	Yes	Yes
41AL with Phantom reference	No	Yes
0V or 200V polarised microphone associated with **PRE12H** preamplifier	No	Yes

For the outdoor microphone units of type 41AM/CM, a remote control cable (between the acquisition unit and the interface box) is required to activate the electrostatic actuator from the application software **dBTRIG32**.


41AM / CM


41AL

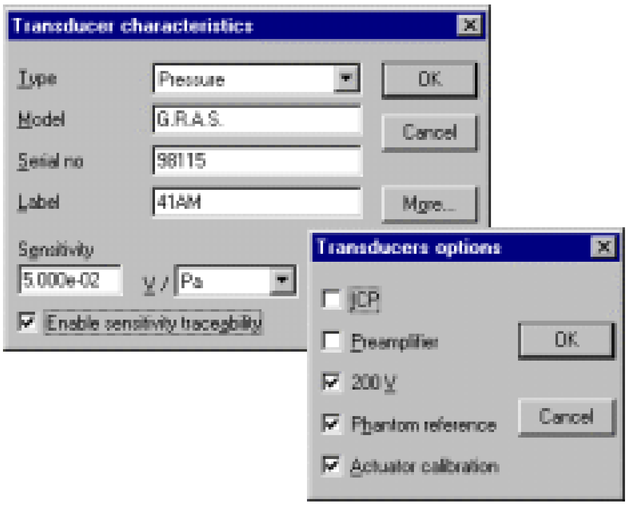

Microphone with PRE12H

Contact your 01dB representative for more information on these products

11.1.3. Hardware configuration of the measurement chain

Proceed as follow to configure the measurement chain for automatic calibration and calibration check.

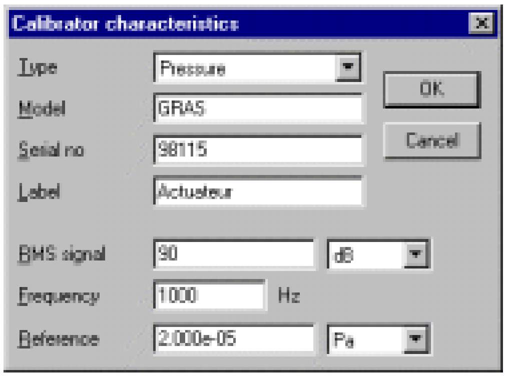
11.1.3.1.Definition of a transducer and a calibrator in dBCONFIG32



In the utility dBCONFIG32, define a transducer with the options **Actuator calibration** for automatic calibration and/or **Phantom reference** for calibration check.

📖 *Activate the option **200V** if the microphone requires a polarisation voltage. For electret microphones activate the option **Preamplifier.***

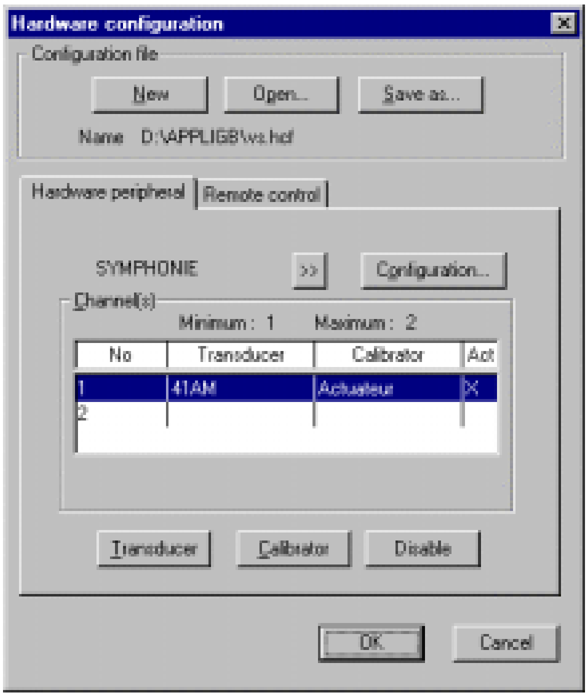
Validate all these dialog box.



For automatic calibration, define a "dummy" calibrator, that will correspond to the electrostatic actuator. For calibration check only, no calibrator needs to be defined.

📖 *For the 41AM/CM unit, define a calibration level of 90dB at 1000Hz.*

11.1.3.2.Hardware configuration in dBTRIG32



Once adequate transducers and calibrators have been defined in dBCONFIG32, use the command **Setup / Hardware configuration**. The dialog box shown aside appears on-screen.

In the **hardware peripheral** tab, select an acquisition platform (SYMPHONIE for example), a transducer and a calibrator that supports automatic calibration and / or calibration check. **Enable** the measurement channel.

📖 *Refer to **chapter 3** relative to hardware configuration of the measurement chain for more information.*

Once the hardware elements of the measurement chain have been selected, click on the **Configuration** key to define the signal conditioning options of the hardware platform. For SYMPHONIE, the dialog box shown aside appears on screen.

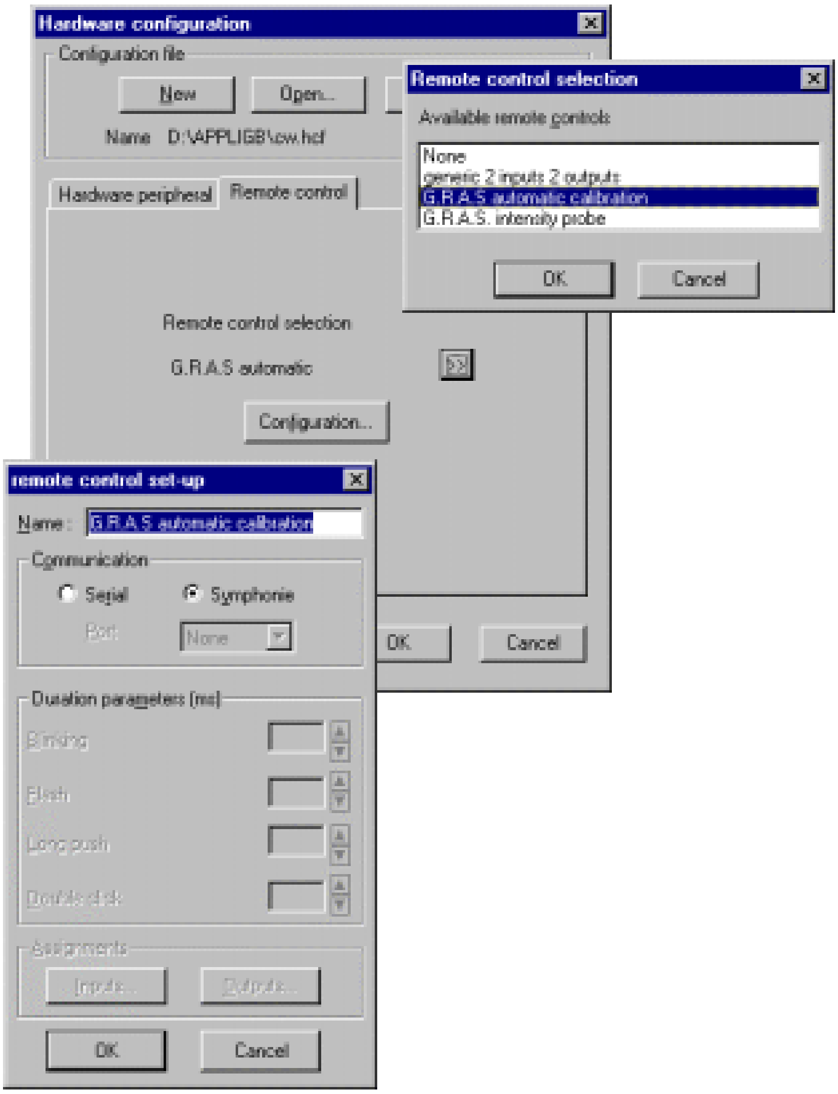
Activate the **Phantom** option for all the active measurement channels. A reference voltage for the calibration check will be generated by the acquisition platform. It corresponds to a level of 90 dB at 1000 Hz.

*Use the option **200V** if the microphone requires an external polarisation voltage.*

For microphones (Pressure type transducers), the input filters are set to 10Hz, whatever the choice made by the user.

11.1.3.3.Selection and configuration of the remote control for automatic calibration (41AM/CM unit)

In order to activate the electrostatic actuator of the GRAS outdoor microphone units (41AM and 41CM) automatically from the application software **dBTRIG32**, define and configure a remote control object in the **Remote control** tab (see below).



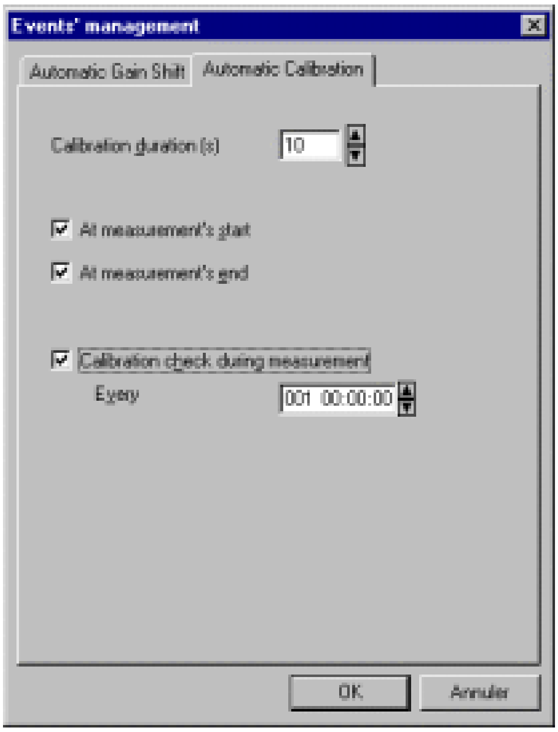
Click on the **>>** key and select the remote control type **GRAS automatic calibration** (defined as Type = 7 in the file DBCD32INI).

After the selection of the remote control, click on the **Configuration** key and select the communication port : either **Serial** (COM1, COM2, etc.) RS232 interface, or **Symphonie** (a special remote control cable that connects between the digital inputs / outputs of the acquisition unit and the interface box of the 41AM/CM unit.

Hardware configuration of the measurement chain is now completed. Define now the automatic calibration and calibration check parameters in the measurement parameters of **dBTRIG32**.

Refer to chapter 3 for hardware configuration and to chapter 12 for remote controls.

11.1.3.4.Measurement parameters in dBTRIG32



Once hardware configuration is completed, the user has to define in the acquisition parameters of **dBTRIG32** the automatic calibration and / or calibration check parameters.

To do so, use the command **Setup / Parameters / Advanced parameters / Automatic calibration** tab.

Configure the **calibration duration** for automatic calibration and / or calibration check in seconds. Define as well when automatic calibration will be performed (at **measurement 'start and / or end**) and the **periodicity** of the calibration check during a measurement.

11.1.4. Automatic and manual use of these functions

Once the measurement chain has been completely configured (for both hardware and software elements), **automatic calibration** is performed as follow:

1. If calibration at measurement' start has been selected (command **Setup / Parameters / Advanced parameters / Automatic calibration** tab), the application software will activate the electrostatic actuator for the defined calibration duration when a measurement session is started (command **Measurement / Start**).
2. The software adjusts the transducer sensitivity after calibration.
3. Data logging of noise data is the n carried out.
4. If calibration at measurement's end has been selected (command **Setup / Parameters / Advanced parameters / Automatic calibration** tab), the application software will activate the electrostatic actuator for the defined calibration duration when a measurement session is ended (command **Measurement / End**).
5. The software adjusts the transducer sensitivity after calibration.

Calibration check can be performed automatically according to a user-defined periodicity (command **Setup / Parameters / Advanced parameters / Automatic calibration** tab) or manually by using the command **Measurement / Calibration check**.

In the latter case, **dBTRIG32** switches to Pause mode (no data logging) and to calibration mode (The Phantom reference is activated on a pin of the preamplifier connector) until the operator use the command **Measurement / Calibration check** again. Data logging is then carried out normally.

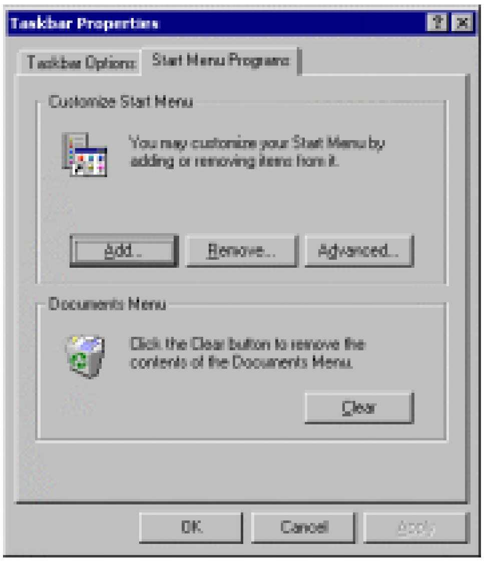
11.2. Auto reboot facility of the measurement system

dBTRIG32 features an auto reboot facility in case power shortage occurs during unattended measurement sessions (mains cut off or battery pack of the notebook completely discharged). Data logging will be started automatically as soon as the mains power supply is switched back on. Configure this facility in 3 steps : in dBTRIG32, by creating a shortcut in the start up group of Windows 95 and by modifying system files MSDOS.SYS and SCANDISK.INI.

11.2.1.1.In dBTRIG32

Go to the **Setup** menu of dBTRIG32 and activate the command **Automatic restart**.

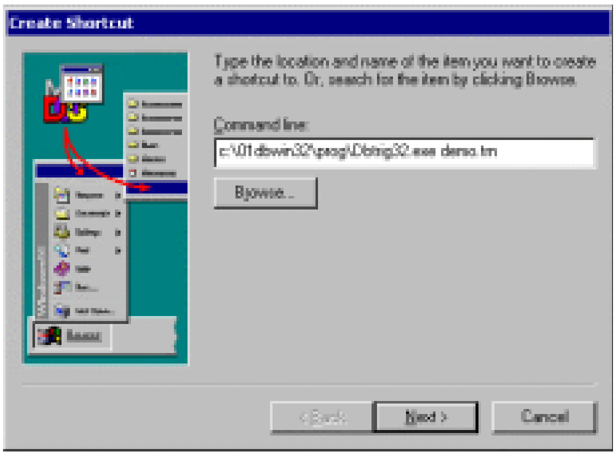
11.2.1.2.Shortcut for dBTRIG32 in the Start menu of Windows 95



Create a shortcut for dBTRIG32 in the start-up group of Windows 95. To do so, proceed as follow:

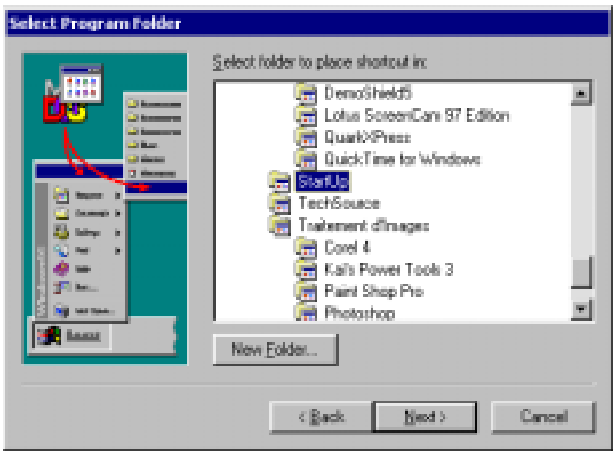
Click on the right button of the mouse while on the Start bar of Windows 95 and select **Properties** in the contextual menu. The following dialog box is displayed on screen.

Select the **Start Menu Options** folder and press the **Add** key.



In the **Create Shortcut** dialogue box, browse to find the path of the dBTRIG32 program (by default C:\ PROGRAM FILES\ 01dB PROGRAMS\ DBTRIG32.EXE).

Add to this line the name of the configuration file for the active measurement session (example : DEMO.TRN) then click on **Next**.



The configuration file DEMO.TRN must be located in the same directory than the program dBTRIG32.EXE in order to use the above syntax. The full path for the configuration file must be specified if not located in the same directory.

In the next dialog, place the shortcut in the **Start-up** group then click on the **Next** key.

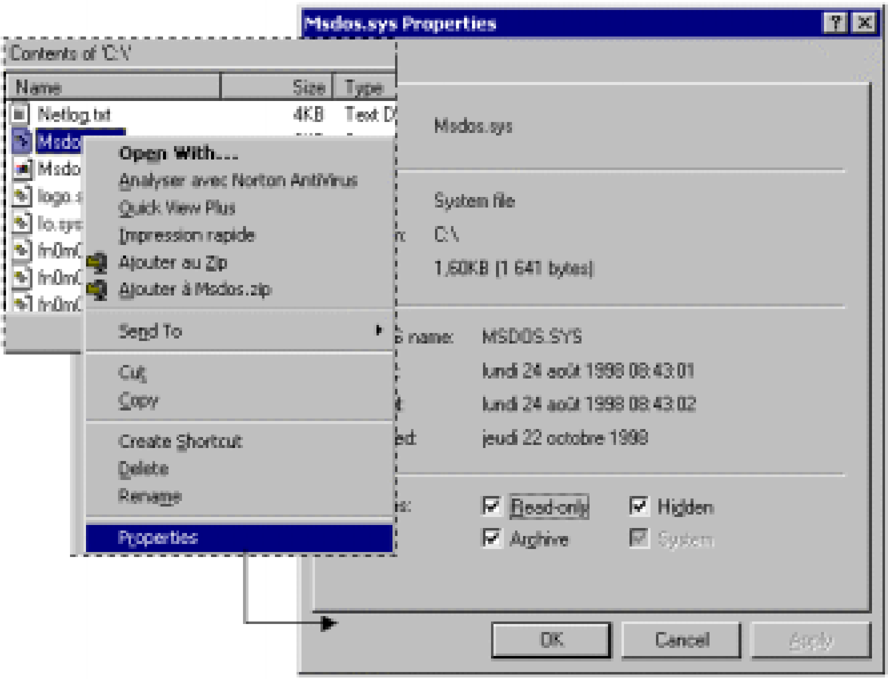
Give a name to this shortcut and validate all the dialog boxes.

Open the start menu and check that the shortcut for **dBTRIG32** is located in the start-up group.

11.2.1.3.Edition of Windows 95 system files

In case of power shortage of the computer, Windows 95 may not be properly shut down. When powering up the computer, the exploitation system may ask the user to execute Scandisk before loading Windows 95.
In order to automatically reboot the system at start-up, the following parameters need to be adjusted:

- For the file C:\WINDOWS\MSDOS.SYS or C:\MSDOS.SYS, modify the file attributes to be able to edit the file (select the file, click on the right mouse button and select **properties** in the contextual menu). Remove the attributes **Read-only** and **Hidden**.



📖 *To display system files in Windows explorer, select the command **View / Options** then the tab **View** and tick the option **Show all files**.*

📖 *Once the modifications to the file have been carried out, select again the attributes **Read-only** and **Hidden***

- In the file MSDOS.SYS, add the following to automatically execute ScanDisk when rebooting the computer :

[Options]
AutoScan=2

- In the file C:\WINDOWS\COMMAND\SCANDISK.INI, modify all the PROMPT fields to ALWAYS, FIX or NEVER (see other choices) in order to automate all the operations of Scandisk.

11.3. Automatic adjustment of the dynamic range

dBTRIG32 offers a control option for the dynamic range during the course of measurement, according to user-defined criteria.

Automatic control enables modification to the dynamic range corresponding to the measured noise levels. Overload criteria are defined, based on minimum and maximum gain levels as well as relative threshold levels with respect to the overload levels.

Each time a surcharge or underload occurs, the data will not be recorded for a short duration corresponding to the time necessary to initialise the amplifiers.

Each time an overload or underload occurs, the data will not be recorded for a short duration corresponding to the time necessary to initialise the amplifiers of the acquisition unit. This initialisation duration is equal to 500 ms when A weighting is selected and to 4 seconds when no weighting (Lin) is selected. The "measured" values during the initialisation period are not taken into account in the logged data and in the digital indicators of the measurement window.

Furthermore, the digital indicators, the overload and underload indicators are reset for each dynamic range shift.

Automatic gain shift allow the user to measure over a wide range of sound levels (typically 20 -140dB. It is strongly recommended to select careful the parameters, as poor settings may result in continuous adjustments and hence important data loss.

Refer to paragraph 7.6.1 relative to automatic gain shift for more information.

11.4. Using the maximum dynamic option (2 channels -> 1 channel)

11.4.1. Principle and procedure

It is possible, with a specific cable, to link two input channels of the acquisition unit in order to form a single measurement channel of greater dynamic (115dB for the **Symphonie** unit, from 20dB to 135dB). The user can therefore perform single channel measurements with an extended dynamic range, in order to avoid automatic or manual gain shift during acquisition.

This function finds its applications for long term noise monitoring at a single location. It can be coupled easily with automatic calibration (with an appropriate front end).

Proceed as follow to configure and use the maximum dynamic option in **dBTRIG32**:

Measurement chain hardware configuration

1. Define adequate transducer and calibrator in the utility dBCONFIG32
2. Define another transducer with exactly the same characteristics as the first one.
3. Select two transducer / calibrator couples in the hardware configuration of dBTRIG32

Software configuration

4. Use the command **Setup / Parameters / Measurement / Overall** tab to setup the maximum dynamic option.

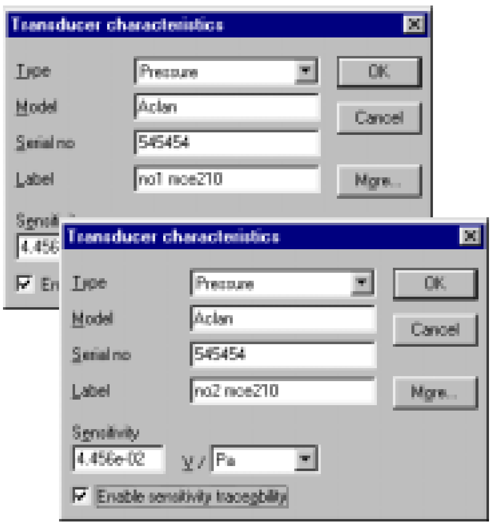
Then independently calibrate each measurement channel before starting any acquisition.

In order to use the maximum dynamic option, the user must have a specific cable allowing to connect the microphone to both input channels of the acquisition unit.

11.4.2. Hardware configuration of the measurement chain

Proceed as follow to configure the measurement chain.

11.4.2.1.Definition of two identical transducers and a calibrator in dBCONFIG32



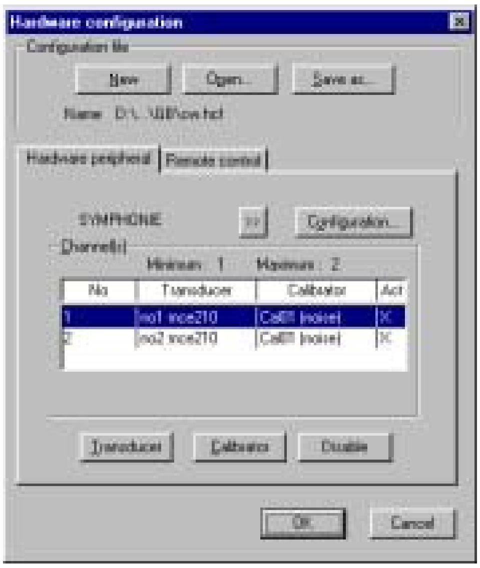
In the utility dBCONFIG32, define two transducers with **identical characteristics**, including signal conditioning options.

*Activate the option **200V** if the microphone requires a polarisation voltage. For electret microphones activate the option **Preamplifier.***

Define as well an adequate calibrator.

If, for example, the user wishes to get the maximum dynamic with automatic calibration (see **paragraph 11.1**), just define an identical transducer as shown above to access the option.

11.4.2.2.Hardware configuration in dBTRIG32



Once adequate transducers and calibrators have been defined in dBCONFIG32, use the command **Setup / Hardware configuration**. The dialog box shown aside appears on-screen.

In the **hardware peripheral** tab, select an acquisition platform (SYMPHONIE for example), a transducer and a calibrator for each channel and **enable** these channels. Select the transducers previously defined.

Once the hardware elements of the measurement chain have been selected, click on the **Configuration** key to define the signal conditioning options of the hardware platform.

Activate all options required for the transducer in use.

📖 *Refer to **chapter 3** relative to hardware configuration of the measurement chain for more information*

11.4.2.3.Measurement parameters and calibration in dBTRIG32

Once hardware configuration is completed, the user has to define in the acquisition parameters of **dBTRIG32** for the use of the maximum dynamic.

To do so, use the command **Setup / Parameters / Measurement / Overall** tab**.**

Tick the box **Maximum dynamic (2 channels -> 1 channel)** then validates.

Then, for the first use, perform a manual calibration of each measurement channel independently by using the command **Setup / Calibration**. Refer to **chapter 5** for more details.

11.4.3. Use of this function

Once the measurement chain has been completely configured (for both hardware and software elements), the **maximum dynamic** option is activated. The user will then not be able to access automatic or manual gain settings.

This function may be activated simultaneously to automatic calibration.

When recordings audio signals, the dynamic range of acquisition is equal to the dynamic range of the input channels that is less amplified (that is the highest range, typically 65 - 135dB for Symphonie)

12. REMOTE CONTROLS AND DBTRIG32

It is possible to use remote control objects with 01dB PC based measurement systems to control the measurement process (start data logging, record an audio signal, etc.) or to enable a specific event when a user-defined condition is fulfilled (alarm triggering when a user-define threshold is exceeded, for example).

The interface between a physical remote control and the measurement system is performed either by the RS232 serial interface of the computer or the digital inputs / outputs of the acquisition unit. In the measurement software module, at the hardware configuration stage, a remote control object can be defined.

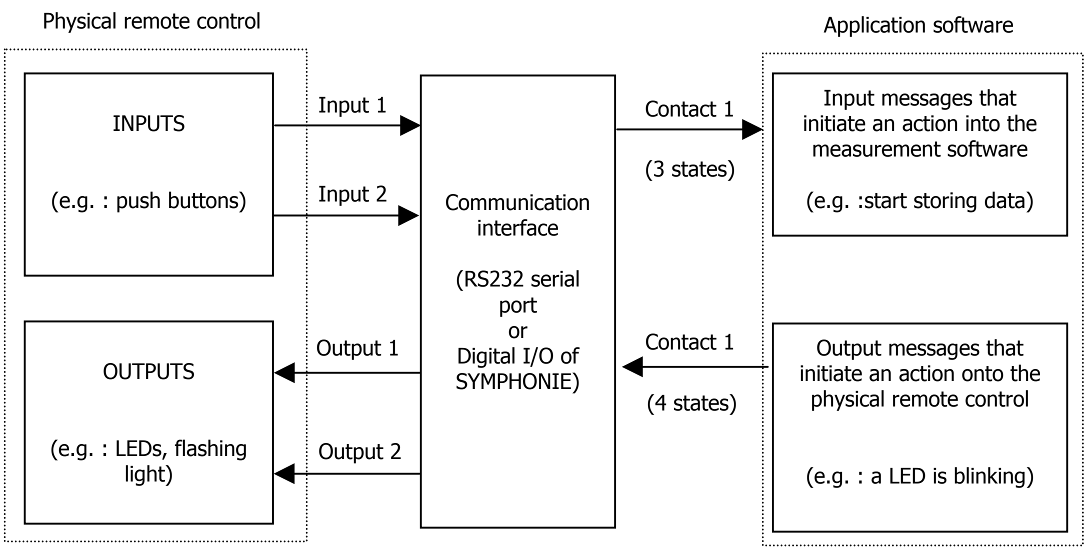
We describe in this chapter the generic remote controls of **dBTRIG32** (definition, configuration and mode of operation).

Then, we present various applications for **dBTRIG32** remote controls:

- Activation of an electrostatic actuator for automatic calibration at measurements' start or end (specific remote control) (see **paragraph 11.1**)
- Triggering audio recordings from a simple one button physical remote control
- Dynamic coding of noise event with a two button physical remote control
- Triggering of a flashing light alarm when a user defined threshold is exceeded (**expert mode only**)

The electrical drawings of the physical remote controls and the communication interfaces (RS232 port, SYMPHONIE digital inputs / outputs) are also given.

12.1. Definition of a generic remote control object



A generic remote control object for 01dB programs is made of 2 physical inputs and 2 physical outputs. It features **16 input functions** and **16 output functions**, each one of them having a START and a STOP state.

It means that **32 input messages** can be received by an application software by mean of the remote control object (to remote control the measurement for example) and **32 output messages** can be sent by the application software to a physical remote control (to trigger a light alarm for example).

Messages are assigned to each **contact** (corresponding to an input or an output of a physical remote control) in order to perform a specific operation.

The input contacts (generally connected to push buttons of a physical remote control) can present 3 different states : click, double click and long push of a button. An input message is assigned to each state.
The output contacts (generally connected to flashing lights or LEDs) can present 4 different states : active, inactive, blinking and flash. A contact state is assigned to an output message.

In fact, each contact corresponds to a specific pin of the communication interface (RS232 connector of a microcomputer, MiniDyn connector of the Symphonie acquisition unit).

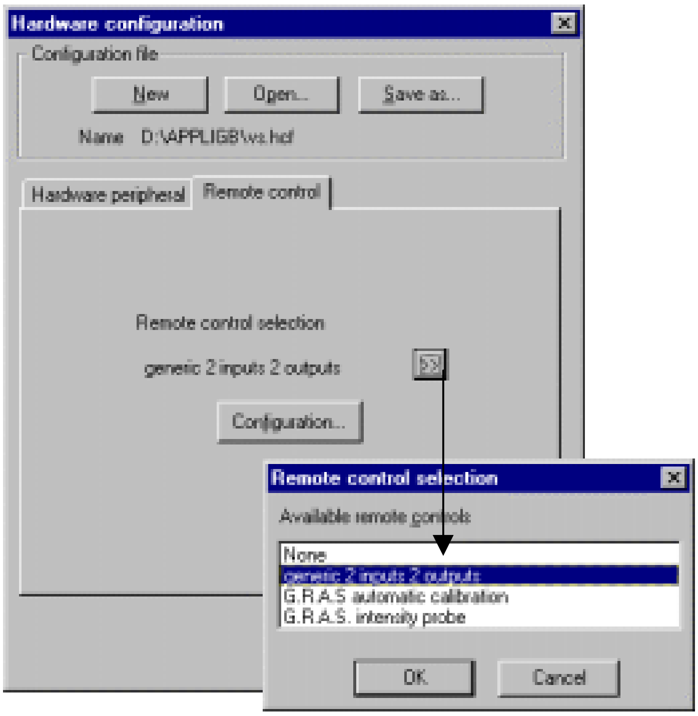
When an input contact is activated, a voltage is generated on this pin. This voltage is then interpreted by the remote control object and the application software execute the appropriate input message.
When an output message is sent by the application software to the remote control object, the output contact is activated and, depending on its state, a voltage is generated on the pin of the communication interface. This voltage signal is then interpreted by the physical remote control.

Remote control objects are defined in the file DBCD32.INI. If you do not have this file in the 01dB program folder, contact 01dB technical support to obtain a copy of it. A complete description of this file is given in this manual.

Follow the procedure below to define a generic remote control object :

1. Selection of the remote control
2. Configuration of the remote control
3. Assignment of the input functions
4. Assignment of the output functions
5. Configuration of event triggering parameters in the application software (if required)
6. Use of the remote control

12.1.1. Selection of a generic remote control



At the hardware configuration stage, select the **remote control** tab (shown only if the DBCD32.INI file is present in the 01dB programs directory). The dialog box shown aside is displayed on screen.

Click on the **>>** key, and select the remote control type **generic 2 inputs 2 outputs** and click on OK.

The specific remote control **G.R.A.S automatic calibration** and **G.R.A.S intensity probes** are predefined object to respectively activate the electrostatic actuator of the GRAS outdoor microphone units (41AM and 41CM) automatically from the application software **dBTRIG32** and to drive sound intensity and sound power measurements from the 50AI intensity probe handle in **dBFA32**.

After the selection of the remote control, configure its parameters.

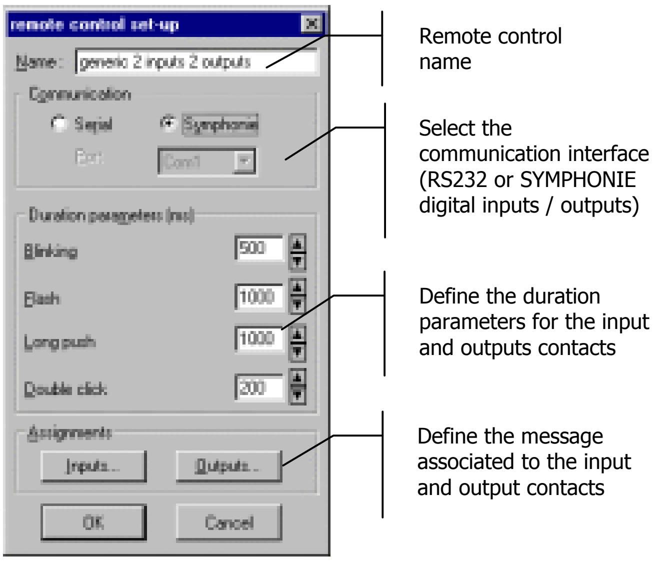
12.1.2. Configuration of a generic remote control

After the selection of the remote control, click on the **Configuration** key to configure the remote control object parameters. The following dialog box appears on-screen:

Define in this dialog box the **name** of the remote control and the **communication** interface to use (either a **Serial** port or the **Symphonie** digital inputs / outputs).

The **duration parameters** refer to the states of the input and output contacts of the remote control.

In the example shown aside, and for input contacts, the state **Long push** corresponds to a continuous essential push of 1000 ms minimum; the state Double **click** corresponds to two successive key push of time interval less than 200 ms.



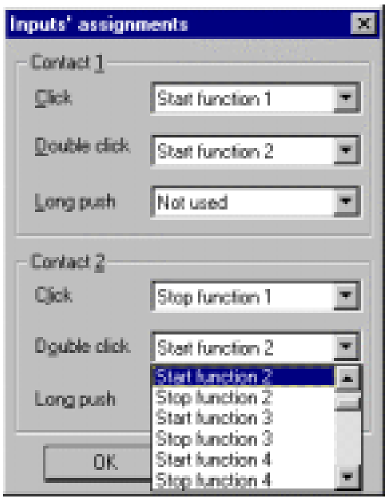
For output contacts, the state **Blinking** generates a voltage on a pin of the connector of the communication interface every 500 ms and the state **Flash** generates a voltage on this pin for 1000 ms only one time.

The configuration of the input and output functions can be accessed by the keys **Inputs** and **Outputs**.

12.1.3. Assignment of input functions

After configuration of the general parameters of the remote control, define which input function will be assigned to each input contact. Access the dialog box shown below by the **Inputs** key.

Select for each state of an input contact a function in the list of input functions. The input contacts present three different states:



- **Click : Simple push of a key**. The voltage on the appropriate pin of the connector of the communication interface passes from 0V to 9V (RS232) or 5V (SYMPHONIE) while the key is pushed.
- **Double click: Two successive pushes of a key**. The voltage on the appropriate pin of the connector of the communication interface passes from 0V to 9V (RS232) or 5V (SYMPHONIE) two times over a user-defined time interval.
- **Long push : Continuous push of a key.** The voltage on the appropriate pin of the connector of the communication interface passes from 0V to 9V (RS232) or 5V (SYMPHONIE) two times over a user-defined period.

When one of these states occur, an input message will be interpreted by the application software. The table below shows a list of the actions that will interpreted by **dBTRIG32** for each input function:

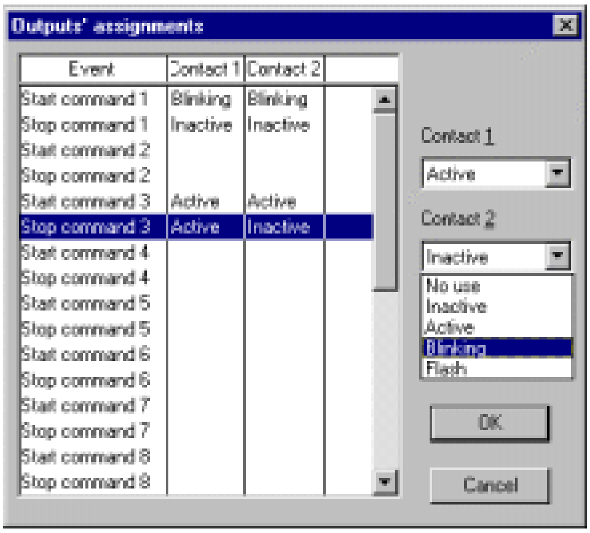
FUNCTION , STATE	ACTION
MEASUREMENT - STORAGE	
Start function 1	Start measurement (data logging)
Stop function 1	Stop measurement (data logging)
AUDIO RECORD	
Start function 10	Start an audio record
Stop function 10	Stop an audio record
CODAGE DES SOURCES	
Start function 11	Start coding (code 4) and end current coding operation
Stop function 11	End coding (code 4)
Start function 12	Start coding (code 5) and end current coding operation
Stop function 12	End coding (code 5)
Start function 13	Start coding (code 6) and end current coding operation
Stop function 13	End coding (code 6)
Start function 14	Start coding (code 7) and end current coding operation
Stop function 14	End coding (code 7)
Start function 15	Start coding (code 8) and end current coding operation
Stop function 15	End coding (code 8)
Start function 16	Start coding (code 9) and end current coding operation
Stop function 16	End coding (code 9)

📖 *In expert mode, and if simultaneous coding has been activated, the end of the current coding operation will not happen.*

12.1.4. Assignment of output functions

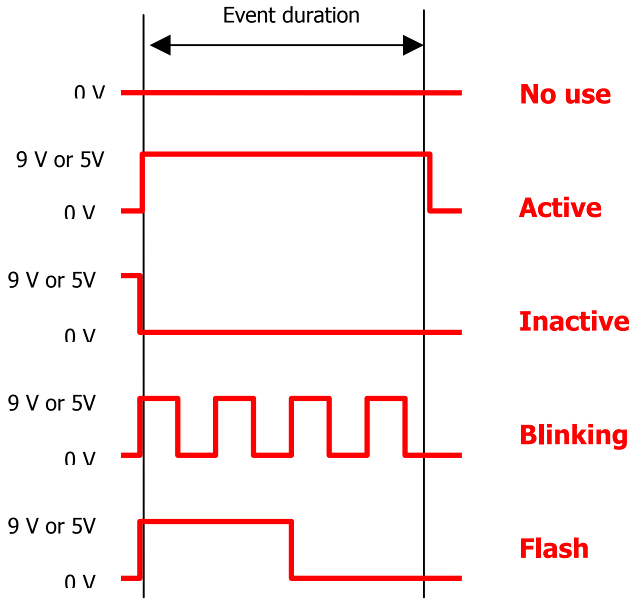
After configuration of the general parameters of the remote control and the input contact assignments, define the assignment of the output contacts that the application software will send to the physical remote control when a user-defined condition has been fulfilled (for example, activation of an electrostatic actuator for automatic calibration or activation of a flashing light a user-defined threshold has been exceeded). Access this dialog box by the **Outputs** key.

For adequate output functions (or events), define the state of each output contact. Output contacts present four different states:



- **No use.**
- **Active.** While an event is activated, the voltage on the appropriate pin of the connector of the communication interface passes from the current state to 9V (RS232) or 5V (SYMPHONIE).
- **Inactive**. While an event is activated, the voltage on the appropriate pin of the connector of the communication interface passes from the current state to 0V.
- **Blinking**. While an event is activated, the voltage on the appropriate pin of the connector of the communication interface passes from 0V to 9V (RS232) or 5V (SYMPHONIE) at a user-defined rate.
- **Flash**. While an event is activated, the voltage on the appropriate pin of the connector of the communication interface passes from 0V to 9V (RS232) or 5V (SYMPHONIE) for a user-defined period.

The drawing below illustrates the different states of the output contacts :



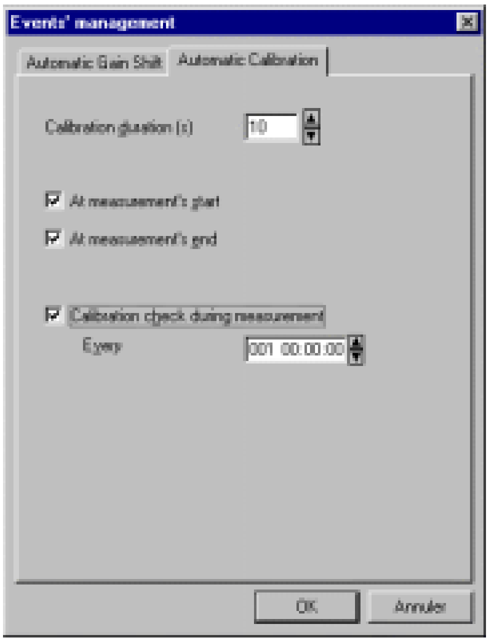
For **dBTRIG32**, the table below gives the list of actions for each output contact:

FUNCTION , STATE	ACTION
MEASUREMENT - STORAGE	
Start command 1	Notify the remote control that the application software waits to start an acquisition
Stop command 1	Notify the remote control that an acquisition is in progress
AUTOMATIC CALIBRATION	
Start command 5	Activate the electrostatic actuator
Stop command 5	Stop the electrostatic actuator
ALARM EVENT (THRESHOLD TRIGGER)	
Start command 8	Activate the alarm n°1 event when a user-defined threshold condition is fulfilled
Stop command 8	Stop the alarm event n°1 when the threshold condition is not valid
Start command 9	Activate the alarm n°2 event when a user-defined threshold condition is fulfilled
Stop command 9	Stop the alarm event n°2 when the threshold condition is not valid
AUDIO RECORD	
Start command 10	Notify the remote control that the application software starts an audio record
Stop command 10	Notify the remote control that the audio record has stopped
CODING	
Start command 11	Notify the remote control that coding starts (code 4)
Stop command 11	Notify the remote control that coding stops (code 4)
Start command 12	Notify the remote control that coding starts (code 5)
Stop command 12	Notify the remote control that coding stops (code 5)
Start command 13	Notify the remote control that coding starts (code 6)
Stop command 13	Notify the remote control that coding stops (code 6)
Start command 14	Notify the remote control that coding starts (code 7)
Stop command 14	Notify the remote control that coding stops (code 7)
Start command 15	Notify the remote control that coding starts (code 8)
Stop command 15	Notify the remote control that coding stops (code 8)
Start command 16	Notify the remote control that coding starts (code 9)
Stop command 16	Notify the remote control that coding stops (code 9)

12.1.5. Configuration of dBTRIG32

Depending on the application software, it may be possible to define triggering parameters. For dBTRIG32, define the following parameters for the application :

- **Automatic calibration**



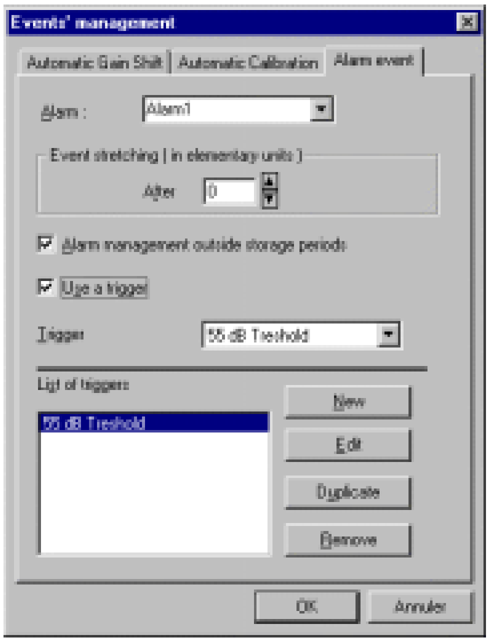
Use the command **Setup / Parameters / Advanced parameters / Automatic calibration** tab

Configure the **calibration duration** for automatic calibration and / or calibration check in seconds. Define as well when automatic calibration will be performed (at **measurement 'start and / or end**) and the **periodicity** of the calibration check during a measurement

- **Alarms (mode expert only)**

Use the command **Setup / Parameters / Advanced parameters / Alarm event** tab

Define for each alarm event (n°1 and n°2), the threshold trigger conditions.



📖 *Consult paragraph 7.6.3 for more information.*

12.1.6. Operating process

Once a remote control object has been completely configured and (if required) once triggering parameters have been defined in the application software, the operator may use the remote control. The operating process is very simple.

Simply connect the physical remote control to the communication interface that was selected. When the triggering conditions are fulfilled, or when the user clicks on a button, the physical remote control or the software executes the actions defined.

📖 *Consult the examples for more information on the use of the remote controls.*

If you wish to develop a physical remote control for specific needs, do not hesitate to contact your 01dB representative to discuss your application.

📖 *Consult the connection diagrams of the communication interface as well.*

12.2. Description of dBCD32.INI

The dBCD32.INI file contains information relative to the definition of remote control objects used in 01dB measurement software programs. The syntax in the file is as follow:

[Com Device]
Count=4

It is the **number of remote control objects** defined in the file. They are numbered in this example from 0 to 3 - therefore a total of 4 objects. This number is incremented each type a new remote control object is defined.

[CDEV 0]
Type=1

This is the **type of remote control**. Type = 1 define a generic 2 inputs 2 outputs remote control.

Name=generic 2 inputs 2 outputs
Comm=-1
Clign=500
Flash=1000
Appui=1000
DblClk=200

These parameters are defined in the **configuration of the remote control** (name, communication port, and duration parameters). See paragraph 12.1.2

Fct Clk0=0
Fct DblClk0=0
Fct Appui0=0
Fct Clk1=0
Fct DblClk1=0
Fct Appui1=0

These parameters are defined in the **configuration of the input contacts, associated to an input message** (See paragraph 12.1.3). The number corresponds to one of the 32 input messages. It varies between 0 (not used) and 32 (stop function 16).

Out1_9=2
Out2_9=2
Out1_10=1
Out2_10=1
Out1_15=2
Out2_15=1
Out1_16=1
Out2_16=1
Out1_17=2
Out2_17=1
Out1_18=1
Out2_18=1

These parameters are defined in the **configuration of the output contacts** (See paragraph 12.1.4). A state of an output contact is associated with this output messages. The number corresponds to one of the 32 output messages. It varies between 0 (not used) and 32 (stop command 16).

[CDEV 1]
Type=7
Name=G.R.A.S automatic calibration
Comm=-1

This specific remote control is used to activate an electrostatic actuator of GRAS outdoors permanent microphone units 41AM/CM.

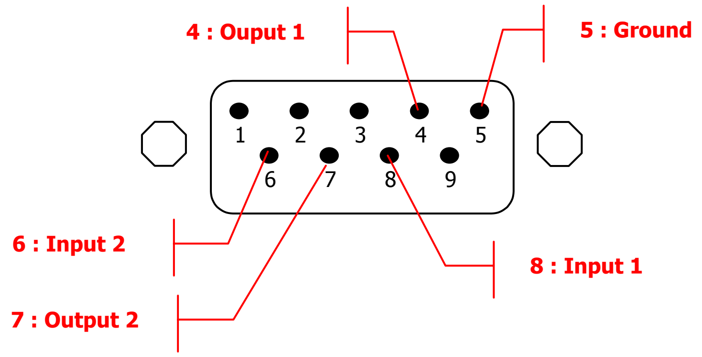
[CDEV 2]
Type=2
Name=G.R.A.S. sound intensity probe
Comm=-1
Clign=500
Flash=200
Appui=1000

This specific remote control is used to control measurements of sound intensity and sound power in **dBFA32** from the handle of a GRAS sound intensity probe 50AI.

12.3. Communication interface

Find in this paragraph the electrical drawings that indicates which pins correspond to input and output contacts of a generic remote control.

12.3.1. For a RS232 9-pin serial port (male connector on the PC)

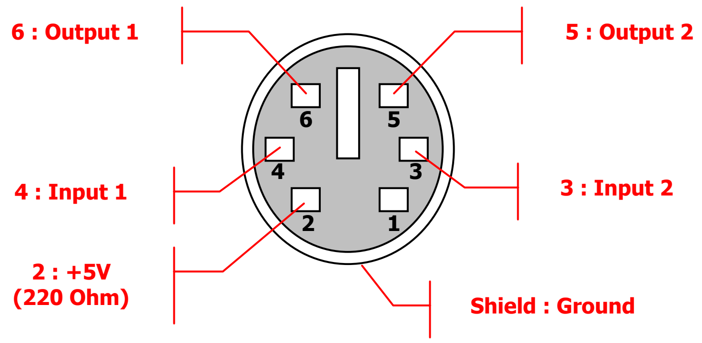


Input contact n°1	Pin 8
Input contact n° 2	Pin 6
Output contact n°1	Pin 4
Output contact n°2	Pin 7
Ground	Pin 5

The generated output signal from a RS232 serial port varies from -9V or -12V (lower state) to +9V or +12V (higher state) for a current of 5mA.

📖 *In most cases, an input remote control requires a 9V-power supply for the push key. This voltage is not supplied by the RS232 interface. A solution is to generate an output signal on the RS232 interface (output contact always activated) and to power supply the push key with it, or to use a capacitance network or a battery.*

12.3.2. For SYMPHONIE digital inputs / outputs (female MiniDyn connector)



Input contact n°1	Pin 4
Input contact n° 2	Pin 3
Output contact n°1	Pin 6
Output contact n°2	Pin 5
Ground e	Body
Signal (+5V / 20mA)	Pin 2

The generated output signal from the MiniDyn connector vary from 0V (lower state) to +5V (higher state) for a current 20 mA.

Pin 2 of the MiniDyn connector always supply +5V / 20 mA in order to power supply various elements of the physical remote controls connected to it.

12.4. Remote control examples

In this paragraph, we give several examples to define, configure and use a remote control object in the application software **dBTRIG32**. All the electrical drawings of the physical remote controls evoked are also given.

Simple examples

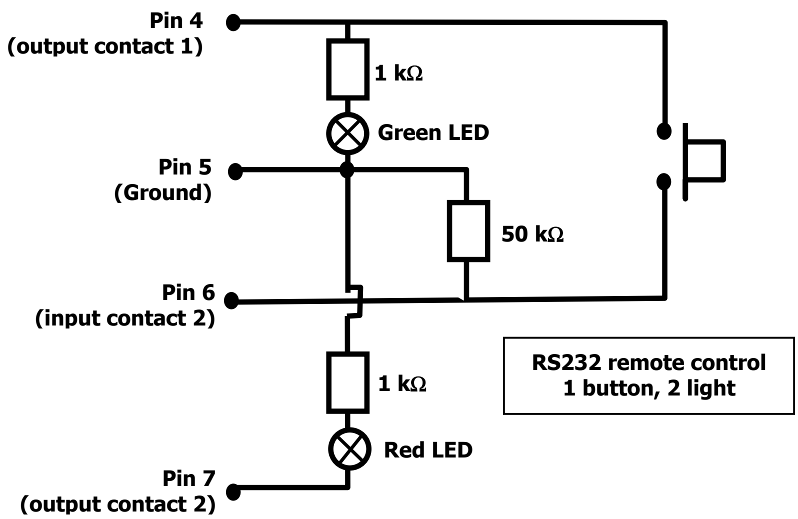
- 1 button, 2 LEDs remote control to manually start audio records during a measurement session
- 2 buttons, 2 LEDs remote control to code noise sources during a measurement session

Complex example (expert mode only)

- Activate a light alarm when a user-defined condition has been fulfilled and activate automatic calibration at measurement 'start and end.

12.4.1. 1 button, 2 LEDs remote control to manually start audio records during a measurement session

The aim of this application is to start an audio record during a measurement session by a simple key push of a physical remote control connected to a serial port of the computer.



The electrical drawing of this remote control (one button, two lights) use a RS232 interface.

When the user clicks once on the push key, an audio record is started. When he (or she) double clicks on the push keys, the audio record is stopped.

The green LED will be lit when a measurement (data logging) is in progress. It will blink when the measurement session is ended.

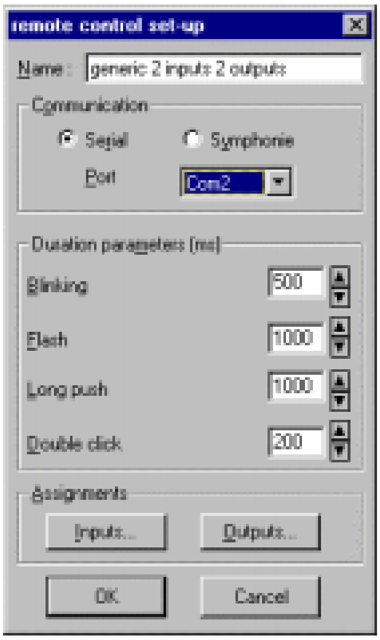
The red LED will be lit when an audio record is being acquired. It will flash once when audio recording has stopped.

🕮 *This remote control requires a 9V-power supply for the push key. This voltage is not supplied by the RS232 interface. A solution is to generate an output signal on the RS232 interface (output contact always activated. It is why the green LED is always lit during a measurement session.*

To configure this remote control, perform the following operations:

- Choice and configuration of the remote control object
- Assignments of input and output contacts
- Connection and operating process

12.4.1.1.Choice and configuration of the remote control object



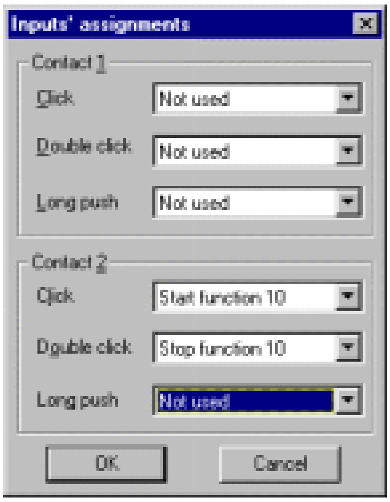
At the hardware configuration stage, select the **remote control** tab (shown only if the DBCD32.INI file is present in the 01dB programs directory).

Click on the **>>** key, and select the remote control type **generic 2 inputs 2 outputs** and click on **OK.** Click on the **Configuration** key to configure the remote control object parameters. The dialog box shown aside appears on-screen.

Select the communication interface to use (a **serial** one) and the communication **port** . In this example, the physical remote control will be connected on the COM2 serial port of the computer.

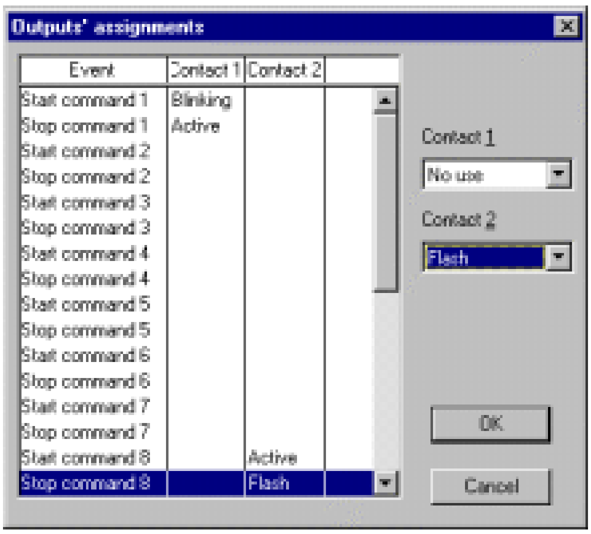
Define the duration parameters **Blinking** and **Flash**, corresponding to the two output states, and **Long push** and **double click**, corresponding to two input contacts

12.4.1.2.Assignments of input and output contacts



Click now on the **Inputs** key of the remote control set-up dialog box.

In our example, we have to start an audio record (**Start function 10**) when clicking on the push key of the remote control and to stop audio recording (**Stop function 10**) when double clicking on the push key.



Click now on the **Outputs** key of the remote control set-up dialog box.

The green LED (on contact 1) will blink (**blinking**) while data logging has not started (**Start command 1**). It will stay lit all the time (**active**) during the measurement session (**stop command 1**).

The red LED (on contact 2) will be lit all the time (**active**) when an audio record is acquired (**Start command 10**) and it will flash for 1 second at the end of the record (**Stop command 10** and **Flash**).

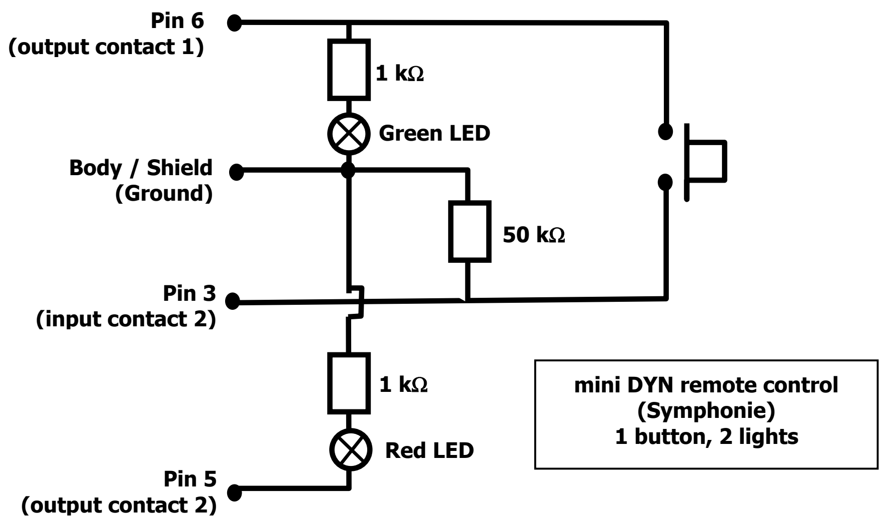
12.4.1.3.Connection and operating process

Once the remote control is configured, switch on the measurement system (connect the microphone, set-up the measurement, etc.) and connect the remote control to the COM2 RS232 interface.

The green LED will blink. Start the measurement session (Measurement / Start). The green LED will continuously switched on.

Click once on the push key to start an audio record. The red LED will be continuously switched on. Double click on the push key to stop audio recording. The red LED will be switched off after 1 second.

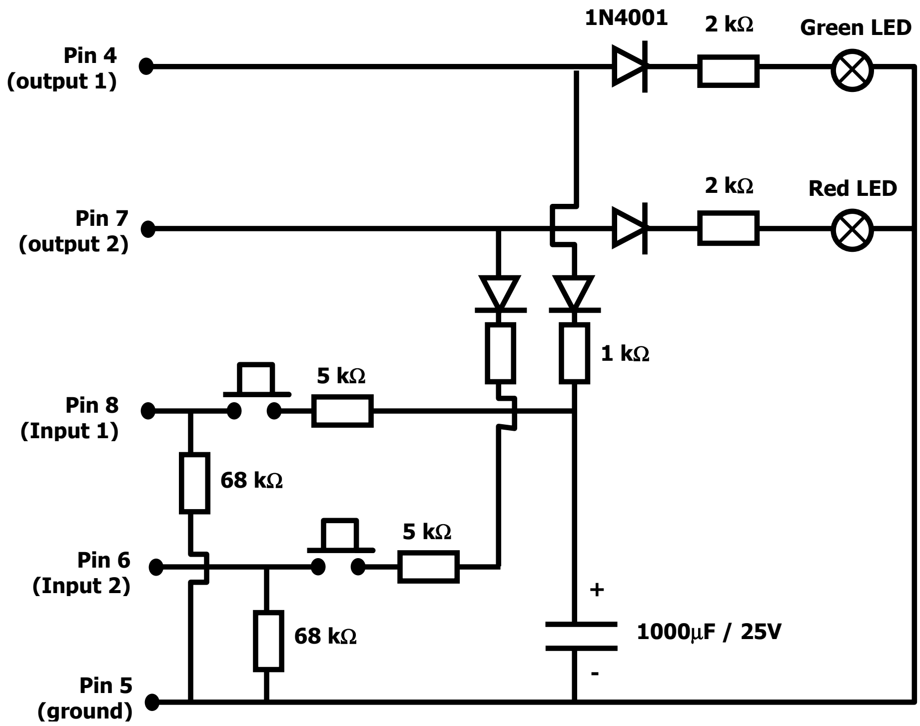
🕮 *Note that if we used a SYMPHONIE remote control instead (the push key being power supplied by the 5V signal output of the MiniDyn connector), no keyboard action would have been necessary before using the remote control and it will also be possible to start data logging using the long push state of the input contact (**Start function 1** for **long push**). The electrical drawing of a Symphonie remote control is given below:*



12.4.2. 2 buttons, 2 LEDs remote control to code noise sources during a measurement session

Let us consider a measurement of road traffic noise. The user wishes to code during a measurement all the plane passing by (code 4) and all the trains as well (code 5) in order to latter eliminate them from overall calculations in **dBTRAIT32** and therefore estimate only the impact of road traffic noise.

To address this application, we can use a 2 buttons, 2 LEDs remote control connected to the RSS232 serial port of the computer. The electrical diagram of such a remote control is as follows:



When the user clicks once of the first push key (input 1) to code plane noise events (code 4), the green LED blinks. To stop coding the plane event, the user double clicks on the first push key. The green LED is then switched on continuously.

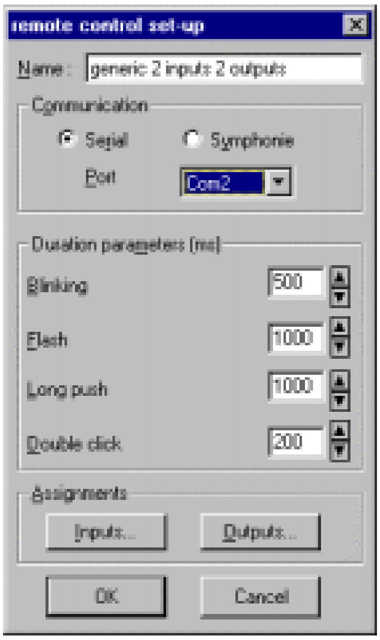
When the user clicks once of the first push key (input 2) to code train noise events (code 5), the red LED blinks. To stop coding the train event, the user double clicks on the first push key. The red LED is then switched on continuously.

A long push on the first key or the second key allows the user to restart data logging.

To configure this remote control, perform the following operations:

- Choice and configuration of the remote control object
- Assignments of input and output contacts
- Connection and operating process

12.4.2.1.Choice and configuration of the remote control object



At the hardware configuration stage, select the **remote control** tab (shown only if the DBCD32.INI file is present in the 01dB programs directory).

Click on the **>>** key, and select the remote control type **generic 2 inputs 2 outputs** and click on **OK.** Click on the **Configuration** key to configure the remote control object parameters. The dialog box shown aside appears on-screen.

Select the communication interface to use (a **serial** one) and the communication **port** . In this example, the physical remote control will be connected on the COM2 serial port of the computer.

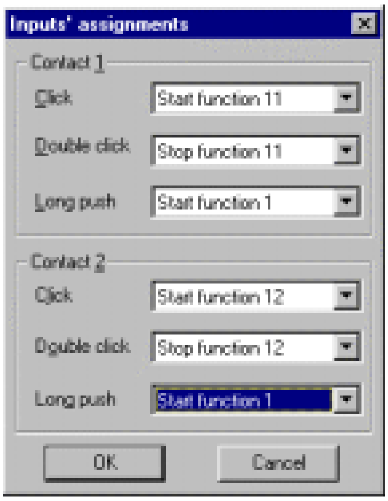
Define the duration parameters **Blinking** and **Flash**, corresponding to the two output states, and **Long push** and **double click**, corresponding to two input contacts.

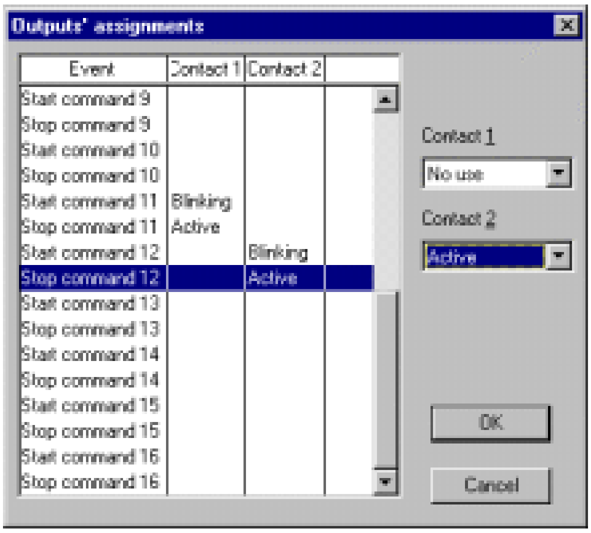
12.4.2.2.Assignments of input and output contacts

Click now on the **Inputs** key of the remote control set-up dialog box.

In our example, we start coding noise data with the code 4 (**Start function 11**) for a simple click of the first push key of the remote control (contact 1) and we stop coding noise data (**Stop function 11**) for a double click.
Similarly, we start coding noise data with the code 5 (**Start function 12**) for a simple click of the second push key of the remote control (contact 2) and we stop coding noise data (**Stop function 12**) for a double click.

A long push of the first or second key will start data storage (**Start function 1**)





Click now on the **Outputs** key of the remote control set-up dialog box.

The green LED (on contact 1) will blink (**blinking**) when coding plane events (code 4). It will be switched on continuously (**active**) at the end of the coding operation.

The red LED (on contact 2) will blink (**blinking**) when coding train events (code 5). It will be switched on continuously (**active**) at the end of the coding operation.

12.4.2.3.Connection and operating process

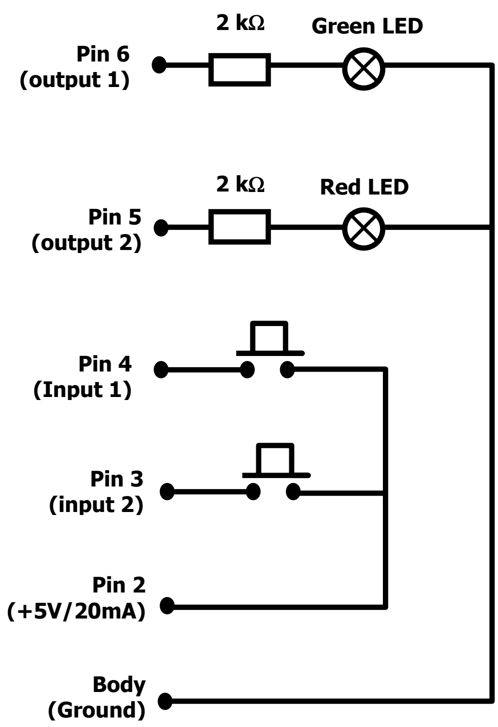
Once the remote control is configured, switch on the measurement system (connect the microphone, set-up the measurement, etc.) and connect the remote control to the COM2 RS232 interface.

Start the measurement session (Measurement / Start) and test the codes 4 and 5 (software interface). This will continuously switch on the two LEDs, in order to charge up the capacitance network of the remote control, in order to power supply the push keys.

Click once on the first key to start coding a plane event (code 4). The green LED blinks. Double click on the same button to stop coding. The green LED is switched on continuously.

Click once on the second key to start coding a train event (code 5). The red LED blinks. Double click on the same button to stop coding. The red LED is switched on continuously.

📖 *Note that if we used a SYMPHONIE remote control instead (the push key being power supplied by the 5V-signal output of the MiniDyn connector), no keyboard action would have been necessary before using the remote control. The electrical drawing of a Symphonie remote control is given below:*



mini DYN remote control
(Symphonie)
2 buttons, 2 lights

12.4.3. Alarm triggering and automatic calibration (expert mode)

Let us now consider a more complex example: the user wishes to activate automatic calibration at measurement 'start and end on one measurement channel of Symphonie (acquisition head with a built-in electrostatic actuator type 41AM/CM only) and to trigger a light alarm event when a Leq level of 65dB has been exceeded.

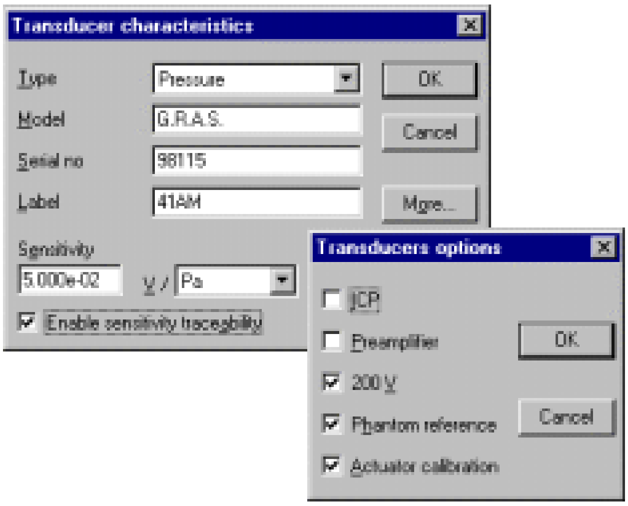
Although automatic calibration and alarm settings are independent from one another, we will consider them simultaneously.

📖 *The electrical drawings for this application are not provided. Contact your 01dB representative for more details.*

To configure this remote control, perform the following operations:

- Hardware configuration of the measurement chain
- Choice and configuration of the remote control object for SYMPHONIE
- Assignments of input and output contacts
- Definition of automatic calibration and alarm events in dBTRIG32
- Connection and operating process

*12.4.3.1.*Hardware configuration of the measurement chain (automatic calibration)



In the utility dBCONFIG32, define a transducer with the options **Actuator calibration** for automatic calibration and/or **Phantom reference** for calibration check.

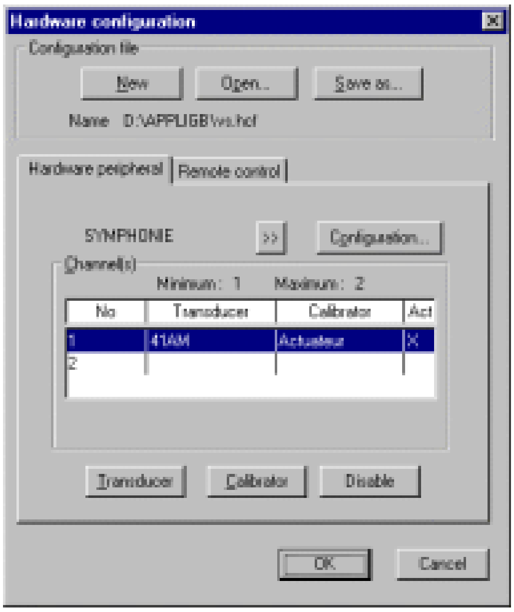
📖 *Activate the option* ***200V*** *if the microphone requires a polarisation voltage. For electret microphones activate the option* ***Preamplifier.***

Validate all these dialog box.

For automatic calibration, define a "dummy" calibrator that will correspond to the electrostatic actuator. For calibration check only, no calibrator needs to be defined.

📖 *For the 41AM/CM unit, define a calibration level of 90dB at 1000Hz.*

Once adequate transducers and calibrators have been defined in dBCONFIG32, use the command **Setup / Hardware configuration**. The dialog box shown aside appears on-screen.



In the hardware peripheral tab, select an acquisition platform (SYMPHONIE for example), a transducer and a calibrator that supports automatic calibration and / or calibration check. **Enable** the measurement channel.

📖 *Refer to chapter 3 relative to hardware configuration of the measurement chain for more information.*

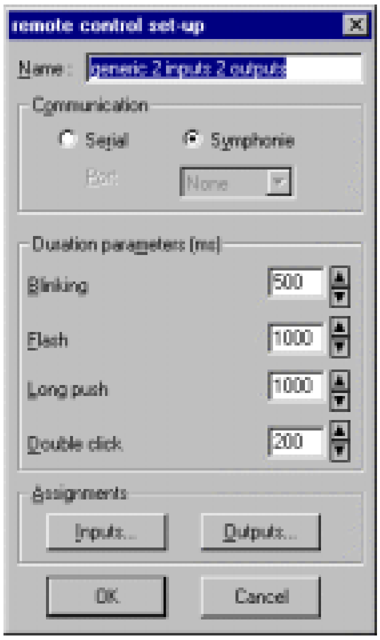
Once the hardware elements of the measurement chain have been selected, click on the **Configuration** key to define the signal conditioning options of the hardware platform. For SYMPHONIE, the dialog box shown aside appears on screen.

Activate the **Phantom** option for all the active measurement channels. A reference voltage for the calibration check will be generated by the acquisition platform. It corresponds to a level of 90 dB at 1000 Hz.

📖 *Use the option **200V** if the microphone requires an external polarisation voltage.*

📖 *For microphones (Pressure type transducers), the input filters are set to 10Hz, whatever the choice made by the user.*

12.4.3.2.Choice and configuration of the remote control object for SYMPHONIE



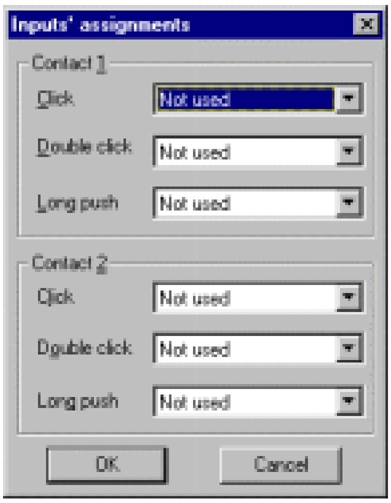
At the hardware configuration stage, select the **remote control** tab (shown only if the DBCD32.INI file is present in the 01dB programs directory). Click on the **>>** key, and select the remote control type **generic 2 inputs 2 outputs** and click on **OK.**

Click on the **Configuration** key to configure the remote control object parameters. The dialog box shown aside appears on-screen.

Select the communication interface type **SYMPHONIE**. The alarm system will have to be connected on the acquisition unit digital inputs / outputs.

Define the duration parameters **Blinking** and **Flash**, corresponding to the two output states, and **Long push** and **double click**, corresponding to two input contacts.

12.4.3.3.Assignments of input and output contacts

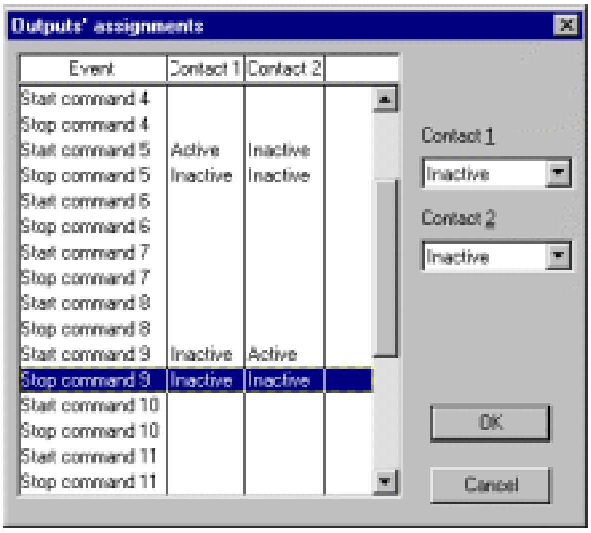


Click now on the **Inputs** key of the remote control set-up dialog box.

In our example, no input function needs to be defined. Indeed, it is only the software that initiates an action (automatic calibration, alarm event) and not the operator.

Each state of each contact have therefore to be set to **Not used**. Validate this dialog box by OK.

Click now on the **Outputs** key of the remote control set-up dialog box.



For automatic calibration, start the command 5 (start calibration) on contact 1 (active) and stop command 5 (stop calibration) on contact 1 (inactive).

Physically, a voltage of 5V will be generated on Pin 6 (LINE1) of the acquisition unit digital I/O when command 5 is active. The voltage will be equal to 0V when command 5 is inactive, at the end of the calibration.

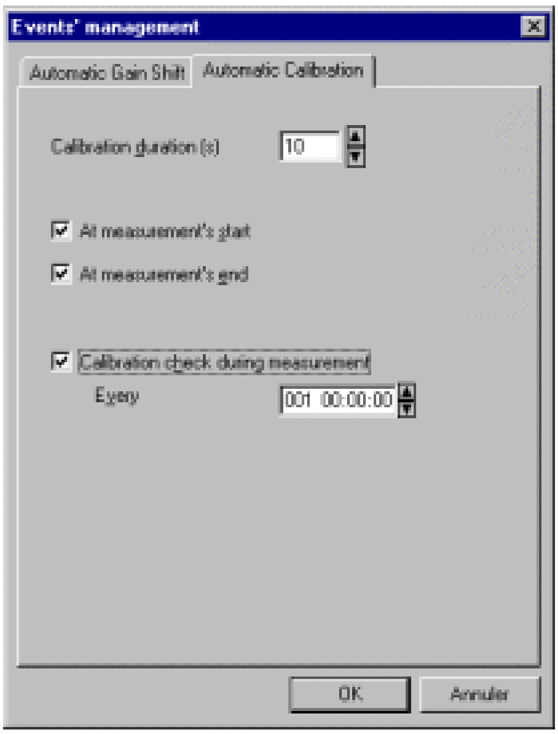
For the **alarm n°2**, start command 9 (trigger alarm when a threshold is exceeded) on contact 2 (active) and stop command 9 (stop triggering alarm event when the measured level goes below the threshold) on contact 2 (inactive).

Physically, a voltage of 5V will be generated on Pin 5 (LINE2) of the acquisition unit digital I/O when command 9 is active. The voltage will be equal to -5V when command 9 is inactive.

Furthermore, the states of contact 2 for the command 5 are set to **inactive** so that no alarm event is triggered during a calibration. Similarly, the states of contact 1 for the command 9 are set to **inactive** so that no calibration occurs when an alarm event is triggered.

📖 *The command 8, corresponding to alarm 1 is not used in this example.*

12.4.3.4.Definition of automatic calibration parameters



Once hardware configuration is completed, the user has to define in the acquisition parameters of **dBTRIG32** the automatic calibration and / or calibration check parameters.

To do so, use the command **Setup / Parameters / Advanced parameters / Automatic calibration** tab

Tick the boxes **Calibration at measurements' start** and **at measurement's end** then define the **calibration duration** (in seconds).

If you wish, define a periodicity to perform a calibration check during a measurement session then validate by **OK**.

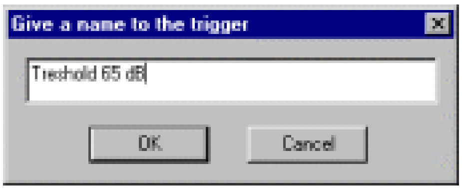
12.4.3.5.Definition of alarm event parameters

Use the command **Setup / Parameters / Advanced parameters / Alarm event** tab to access this dialog box.

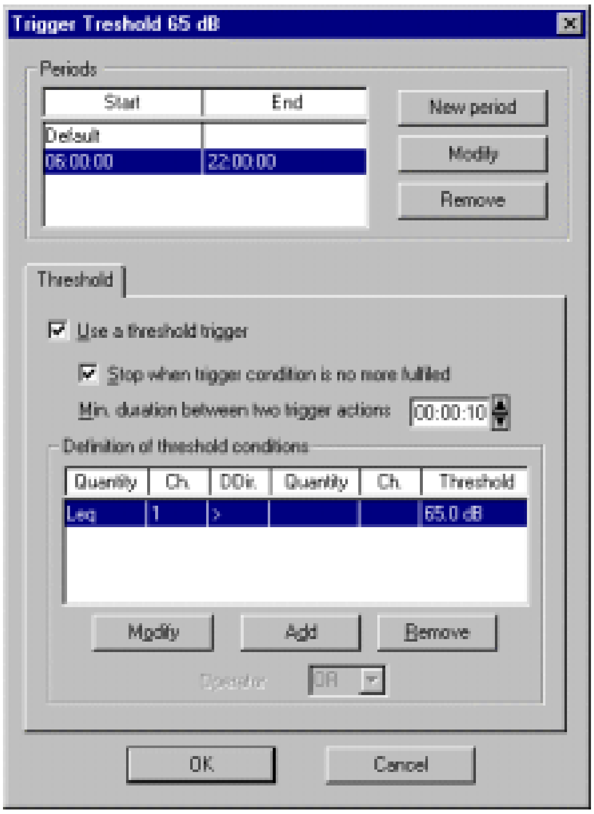
Select the active **alarm** (**Alarm2** in this example, corresponding to the command n°9). Define an **event stretching** in number of elementary units after the true event, knowing that an elementary unit corresponds to the time base of acquisition in **dBTRIG32** (see the acquisition parameters, overall tab, to change this time base).

In this example, and for an elementary unit of 1 second, the alarm event will still be active 2 seconds after the threshold condition is not fulfilled.

Define as well if the alarm event will be triggered when no data logging is in progress.



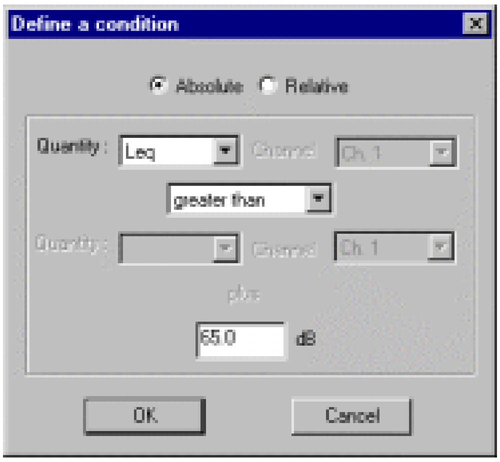
Define now the trigger condition itself. To do so, click on the **New** key and give a name to that trigger.



In the trigger configuration dialog box, first define the periods of a day for which alarm triggering is active (from 06:00 to 22:00 in our example).

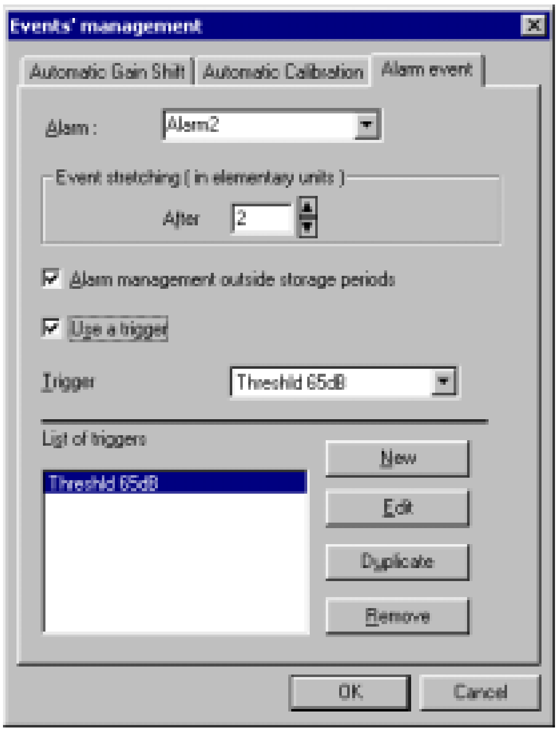
Then, tick the boxes **Use a threshold trigger**, **Stop when trigger condition is no more fulfilled** and define the **minimum duration between two trigger actions** (10 seconds in our example).

To define the threshold condition itself, click on the Add key.



In our example, an alarm event will be triggered when the Leq level is greater than 65dB. Validate this dialog box.

📖 *Several threshold conditions could be defined to trigger the alarm.*



Once the threshold trigger has been defined, tick the box **use a trigger** and select the trigger **Threshold 65dB** in the list. Validate this dialog box.

We have define an Alarm2 event, active from 6am to 10pm, that will be triggered when the Leq level of channel 1 exceed 65 dB.

The event will still be active 2 seconds after the measured level goes below the 65dB threshold.

12.4.3.6.Connection and operating process

From the above configuration, the light alarm system (e.g. flashing light) will be switched on when a voltage is generated on the output contact n°2 (Pin 5 of the MiniDyn connector of the Symphonie unit).

The electrostatic actuator of the microphone unit will be activated before and after any noise data is stored in a measurement session file when a voltage is generated on the output contact n°1 (Pin 6 of the MiniDyn connector of the Symphonie unit).

After setting up the measurement system, simply start an acquisition in **dBTRIG32.** All these operations will be performed automatically by the system.

13. OPTIONAL MODULES OF DBTRIG32

The **dBTRIG32** application software is available in two modes:

- A **light version**, that works like a single channel integrating data logging sound level meter with a simple and easy-to-use graphical interface. Audio recording is also possible.
- A **standard version**, described in this manual.

📖 *See paragraph 2.3 for more detailed information on the functions of each version, and how to switch from one mode to another.*

The following optional modules can be added to dBTRIG32 standard version:

- **Dual channel acquisition module (option)**

It is now possible to perform noise measurements on two channels simultaneously with this optional module. With a special cable, the user may obtain a single measurement of 115 dB dynamic (from 20dB to 135dB), using both acquisition channels.

- **Online analysis of audio records during acquisition**

In addition to ‘classical’ octave and third octave analysis, dBTRIG32 computes in real-time spectrum and multispectrum in 1/6th, 1/8th, 1/12th, 1/24th and 1/48th octave bands. See **chapter 10.**

- **Vibration module**

With this optional module, computation of overall levels according to ISO2631 standard has been implemented. The third octave frequency range is extended down to 1Hz and the sampling frequency can be set under 40 Hz, depending on hardware, allowing extended analysis of long vibration signals.

- **Expert module**

The expert module allows the user to define any combination of trigger conditions for recording audio or spectrum events and generate alarm signals.
Thresholds can be either relative or absolute providing very flexible event detection and data storage.

- **Psychoacoustic module**

The PNL and PNLT criteria are used by the civil aviation in order to evaluate the subjective response of human beings affected by the noise of jet-powered aircraft and helicopters (in flight) in neighbouring communities around airports. They are also used to calculate the EPNL level for certification of aircraft.
Similarly, the Loudness level according to ISO532B (Zwicker) is also computed in real-time.

- **Use with a digital tape recorder DAT (system JAZZ)**

Jazz features real-time analysis of DAT recordings by using AES/EBU digital interface. When using a sound level meter connected to a DAT recorder (environmental monitoring for example), **Jazz** will read the tape in digital format, without loss of quality. The signal does not have to be recorded A weighted on the tape, therefore simultaneous global readings and 1/3 octave analysis are possible. **dBTRIG32,** used with the **Jazz** acquisition card, allows the operator to manage band indexes of digital audio tapes (DAT).

📖 *Refer to **Jazz** getting started user manual or **dBTRIG32** on-line help for more details on the use of the software with a digital tape recorder.*

- **Use of a sound level meter as an acquisition front-end (ACL mode)**

The ACL mode allows using a 01dB sound level meter as acquisition platform in dBTRIG32. The sound level meter sends calibrated data to dBTRIG32 via serial interface.
The user can carry out long-term measurements even on sound level meters without storage function. A continuous surveillance can be included without any interruption of the measurement.

📖 *See chapter 13.6 for more informations about ACL mode.*

📖 *The expert module excepted, corresponding in fact to a version of the software (see **paragraph 2.3**), the other modules do not affect the software user interface.*

In this chapter, we describe the main differences in the use of dBTRIG32 when the optional modules are installed.

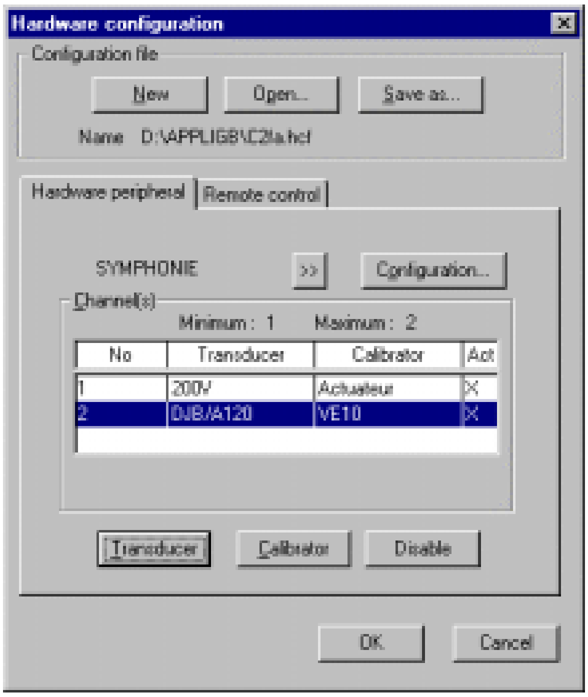
13.1. Dual-channel measurements

It is now possible to perform noise (and vibration, with the corresponding optional module) measurements on two channels simultaneously with this optional module.

All the data logging, source coding and display parameters and most of the acquisition parameters can be defined independently for each measurement channel. **The sampling frequency, the audio maximum frequency and the time base of acquisition are the only parameters common to both channels. See paragraph 7.1.1.**

The following paragraphs describe how to use and configure the software for dual-channel measurements:

13.1.1. Hardware configuration



At the hardware configuration stage, it will be now possible to select a **maximum** of two transducers.

Define a couple transducer / calibrator for each measurement channel and activate each channel for acquisition.

Do not forget to activate the adequate signal conditioning options of the acquisition platform (Configuration key) and to define the corresponding options in the transducers' characteristics in dBCONFIG32.

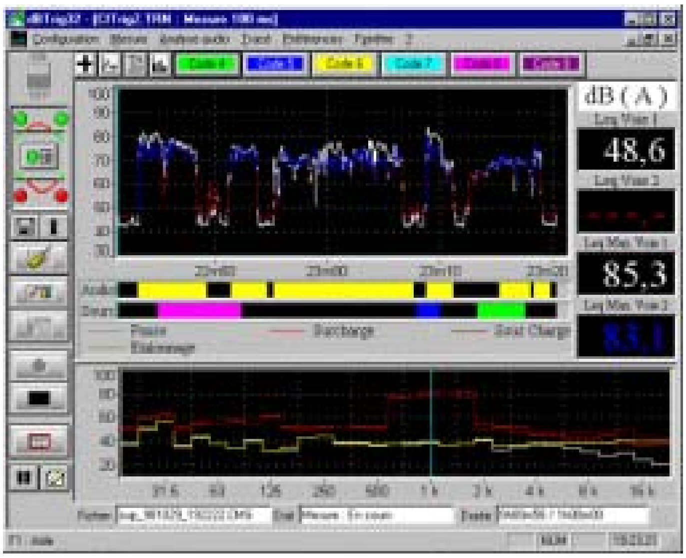
13.1.2. Measurement window and display

When two active channels have been activated during hardware configuration, any measurement window (new or existing configuration files) will feature a time history plot, digital indicators and a spectrum plot for each active measurement channel.

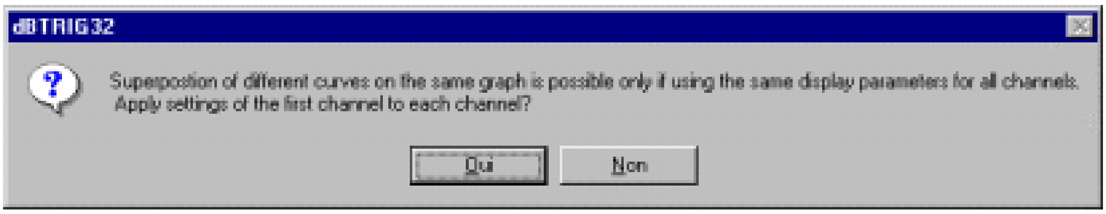
On the display, choose to **link the cursors** on the time histories and /or the spectrum plots (command **Display / Time history** and **Display / Spectra**).

If the measurement channels have identical settings (same transducers, same acquisition parameters on both channels), it will be possible to display data from channel 1 and channel 2 on the same graph by using the command Display / Layout. Tick the box **Display plots of different channels on same view** in this dialog box. This option will not be accessible when the measurement channels do not have the same settings.

The measurement window will look like this:



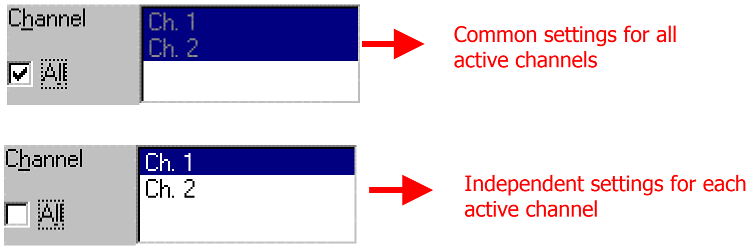
If different display settings have been defined for each channel, a message is displayed on screen. Choose Yes to automatically apply the settings of the first channel to the common view.



dBTRIG32 may adjust the measurement set-up file if, for example, this configuration file has been saved for a single channel configuration. A dialog box is displayed on screen to warn the user that the measurement configuration will be adapted for dual channel measurements.

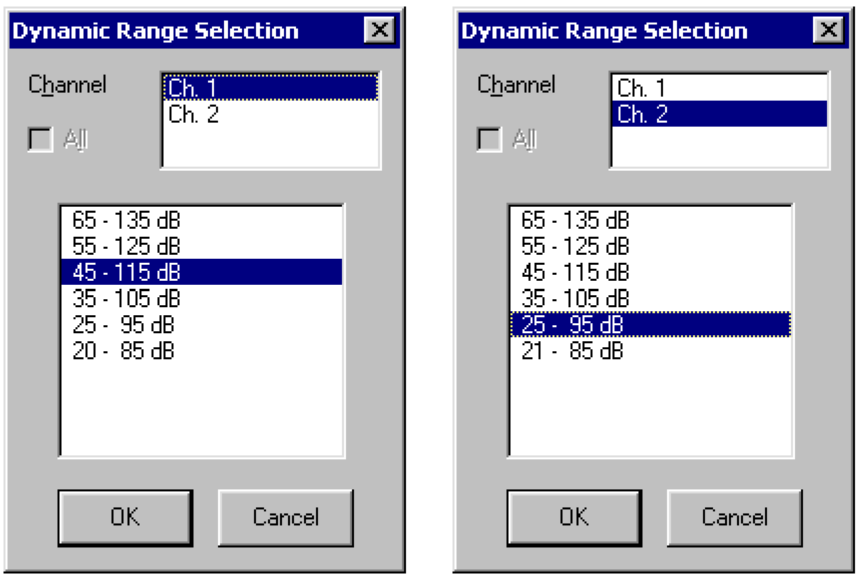
13.1.3. Independent settings for each measurement channel

For dual channel measurements in **dBTRIG32**, it is possible to define independent parameters per measurement channel (acquisition, storage and various display parameters).



As a general rule of thumb, if the case All is ticked, the parameters defined by the user will be applied to all the active measurement channels.
If the case All is not ticked, the parameters defined by the user will be applied to the selected measurement channel in the list (it appears in inverse video).

Let us consider the example of dynamic range manual selection. The user wishes to select two different dynamic ranges for channel 1 and channel 2:



Uncheck the **All** case. Select a measurement channel in the list with the mouse, to choose the appropriate dynamic range. Proceed similarly for all active measurement channels.

In our example, the dynamic range on channel 1 is 45-115dB and the dynamic range of channel 2 is 25-95 dB.

If, the case All is ticked after the user defined independent parameters on each active measurement channel, check the parameters again before performing an acquisition.

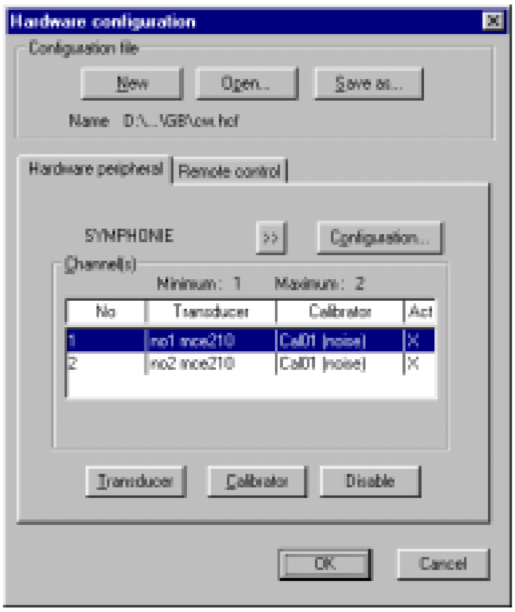
13.2. Maximum dynamic option (2 channels -> 1 channel)

With the dual channel acquisition module and a specific cable, it is possible to link two input channels to form a single channel of greater dynamic (115dB for the **Symphonie** acquisition unit, ranging from 20dB to 135dB).

The user can then perform single channel measurements with an extended dynamic range, without manual or automatic gain shift.

The following paragraphs describe how to use and configure the software with the maximum dynamic option:

13.2.1. Hardware configuration



In the utility dBCONFIG32, define two transducers with **identical characteristics**, including signal conditioning options.

Define as well an adequate calibrator.

During hardware configuration, select an acquisition platform (SYMPHONIE for example) in the **hardware peripheral** tab, a transducer and a calibrator for each channel and **enable** these channels. Select the transducers previously defined.

Do not forget to activate the adequate signal conditioning options of the acquisition platform (Configuration key) and to define the corresponding options in the transducers' characteristics in dBCONFIG32.

13.2.2. dBTRIG32 software configuration

Define in the acquisition parameters of **dBTRIG32** for the use of the maximum dynamic

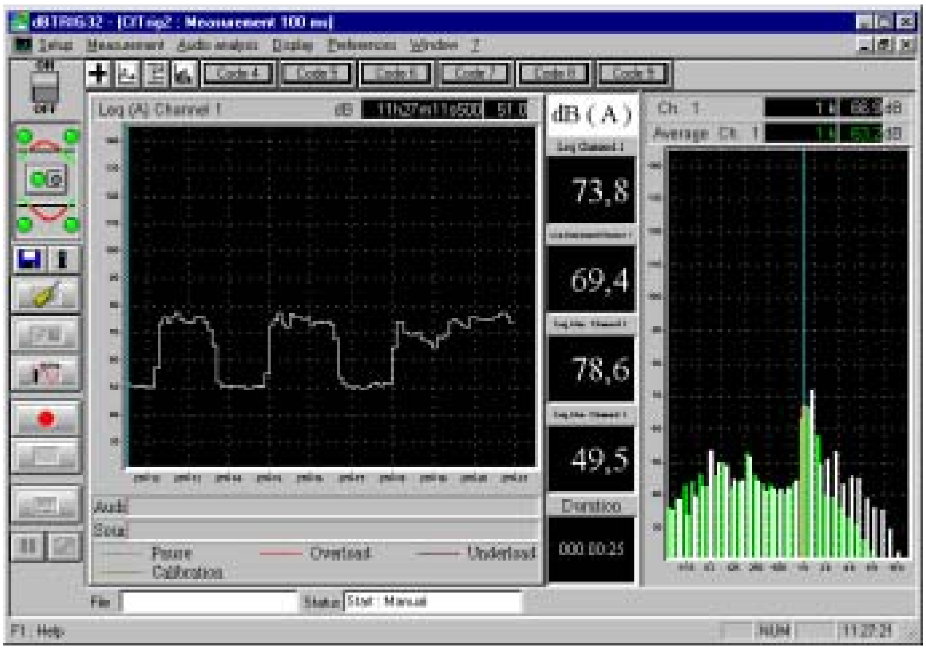
To do so, use the command **Setup / Parameters / Measurement / Overall** tab**.**

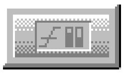
Tick the box **Maximum dynamic (2 channels -> 1 channel)** then validates.

Then, for the first use, perform a manual calibration of each measurement channel independently by using the command **Setup / Calibration**. Refer to **chapter 5** for more details.

13.2.3. General use

Once the measurement chain has been completely configured (for both hardware and software elements), the **maximum dynamic** option is activated. Everything happens as for single channel measurements although no gain settings can be performed. An example of measurement window is given below:



The use cannot access manual gain settings (command  Setup / **Dynamic range** greyed) or automatic gain shift (command **Setup / Parameters / Advanced parameters / Automatic gain shift** tab greyed).

Furthermore, when recordings audio signals, the dynamic range of acquisition is equal to the dynamic range of the input channels that is less amplified (that is the highest range, typically 65 - 135dB for Symphonie).

13.3. Vibration monitoring

The vibration module allows the user to connect an accelerometer on one (or two) channel of the acquisition unit, and to measure vibration levels as a function of time. Overall levels can be calculated according to the standard **ISO2631 part 1** dealing with the evaluation of human exposure to whole-body vibration (general requirement) and **ISO8041** dealing with human response to vibration (measuring instrumentation).

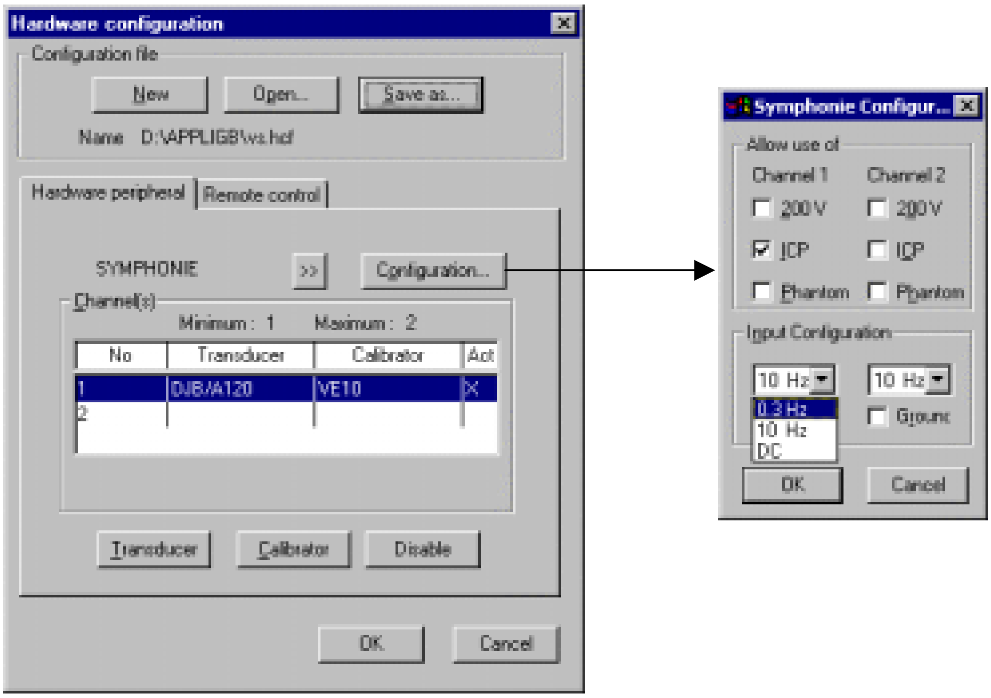
The acquisition parameters are modified with this module. The user may select:

- A **maximum audio frequency** that vary from 20 kHz down to 40Hz (instead off 2.5kHz to 20kHz)
- The following **frequency weightings** C, Lin, Wd, Wk, WBc, Wf, Wh, Wc, We, Wj and Wb according to the standard ISO8041 and ISO2631. (instead of A, B, C, G, Lin)
- The acquisition of octave spectra from 2Hz to 16 kHz and third-octave spectra from 1Hz to 20 kHz (instead of 31.5Hz to 16kHz in octaves and 20Hz to 20kHz in third-octaves)

The following paragraphs describe how to use and configure the software for vibration measurements:

13.3.1. Hardware configuration

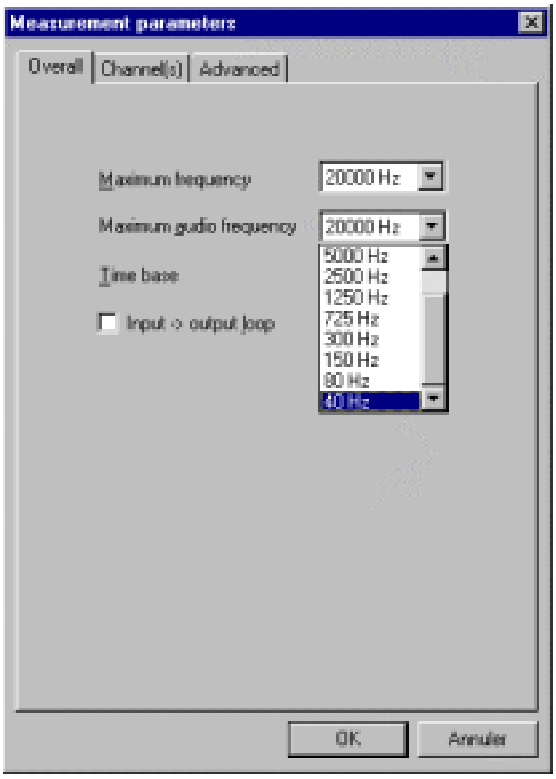
At the hardware configuration stage, select an acceleration type transducer (e.g. DJB A120 ICP® accelerometer) and acceleration type calibrator (e.g. VE-10 vibration calibrator from RION).



Define a couple transducer / calibrator for each measurement channel and activate each channel for acquisition.

Do not forget to activate the adequate signal conditioning options and to select appropriate input filters of the acquisition platform (Configuration key) and to define the corresponding options in the transducers' characteristics in dBCONFIG32.

13.3.2. Maximum audio frequency



When an accelerometer has been selected at the hardware configuration stage, it becomes possible to select a maximum audio frequency down to 40Hz in order to record vibration signals.

Use the command **Setup / Parameters / Acquisition** and select the **Overall** tab. This dialog box appears on-screen.

Select in the list the maximum frequency for recording vibration signals.

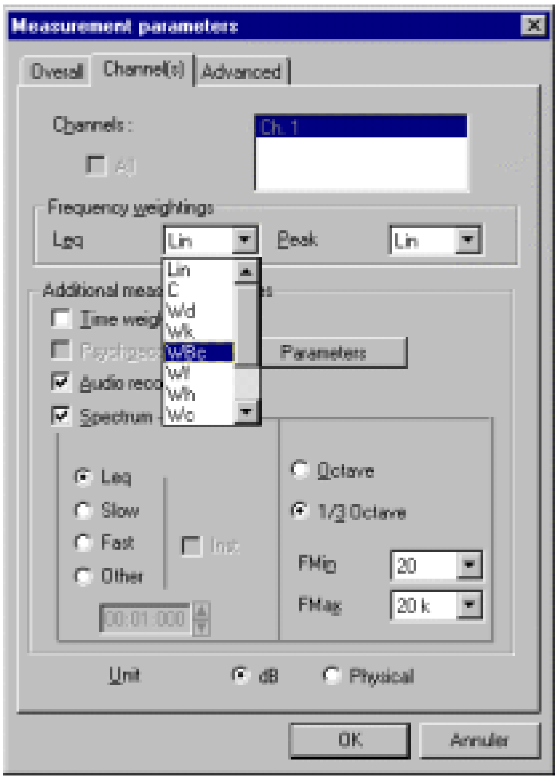
With the vibration module, the following pass bands for signal recording can be selected: **0-2500Hz, 0-1250Hz, 0-725Hz, 0-300Hz, 0-150Hz, 0-80Hz and 0-40Hz.**

13.3.3. Vibration frequency weightings

When an accelerometer has been selected at the hardware configuration stage, it becomes possible to select vibration frequency weightings for calculation of overall time-varying levels (Leq, Fast, Slow, etc.)

Use the command **Setup / Parameters / Acquisition** and select the **Channel(s)** tab. This dialog box appears on-screen.

Select in the list of **Leq frequency weightings** the desired vibration weighting.



With the vibration module, the following frequency weightings can be selected

- **Lin** (no weighting) and **C weighting networks**

ISO2631 part 1

- **Principal weightings Wd** (horizontal vibrations) **and Wk** (vertical vibrations) to evaluate the adverse effects of vibrations on human beings relative to health, comfort and perception issues.
- **Additional weighting Wc** (seat back vibration measurement), **We** (rotation vibration measurement) and **Wj** (vibration measurement under head of recumbent person) related to comfort issues, in specific cases.
- **Principal weighting Wf** to evaluate the adverse effects of vibrations on human beings relative to motion sickness
- **Principal weighting Wb** to evaluate the adverse effects of vibrations on human beings relative to comfort issues in given environments (vehicles on rails, for example)

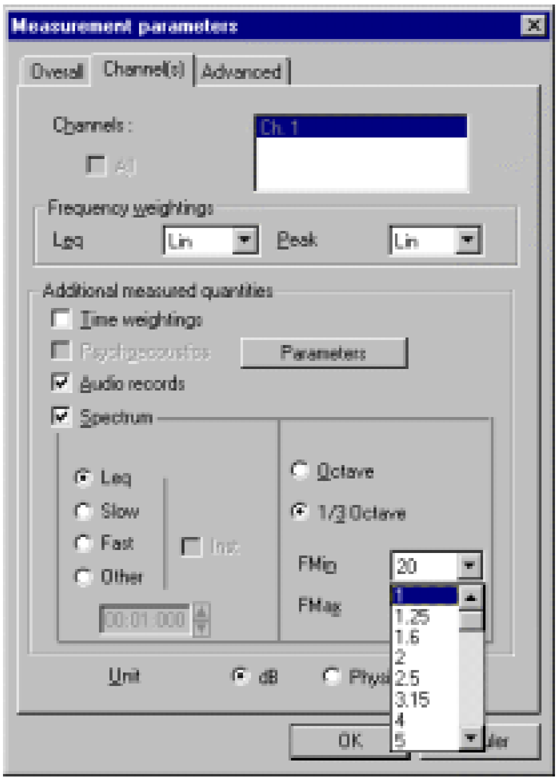
ISO8041 (and ISO2631 part 2)

- **Principal weighting WBc** (all directions combined) to evaluate the adverse effects of continuous and shock-induced (1Hz à 80Hz) vibrations in buildings.

ISO8041 (and ISO5349)

- **Hand-arm weighting Wh** (all directions) to evaluate hand-transmitted vibrations.

13.3.4. Octave and third-octave real-time spectrum acquisition



When an accelerometer has been selected at the hardware configuration stage, it becomes possible to select third octave frequency limits from 1Hz to 20kHz and octave frequency limits from 2Hz to 16kHz for real-time acquisition of a spectrum, required for the calculations of vibration levels according to the standard ISO2631 part 1 and ISO8041.

Use the command **Setup / Parameters / Acquisition** and select the **Channel(s)** tab. This dialog box appears on-screen.

Select the spectrum resolution (**octave** or **1/3 octave**) and the minimum (**Fmin**) and maximum (**Fmax**) centre frequencies in the scroll down lists.

13.4. Psychoacoustics criteria

The psychoacoustics module allows the user to calculate in real-time overall levels used in the field of airport noise assessment (PNL, PNLT) to evaluate the subjective response of human beings affected by the noise of jet-powered aircraft and helicopters (in flight) in neighbouring communities around airports. They are also used to calculate the EPNL level for certification of aircraft.. These levels can be displayed and stored with other global values.

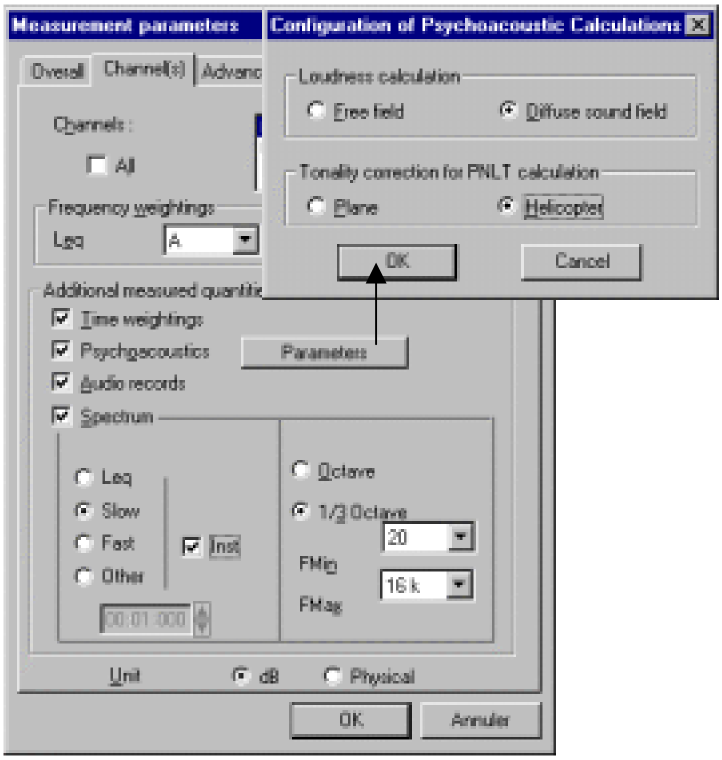
Similarly, the Loudness level according to ISO532B (Zwicker) is also computed in real-time.

The following paragraphs describe how to use and configure the software for psychoacoustic measurements:

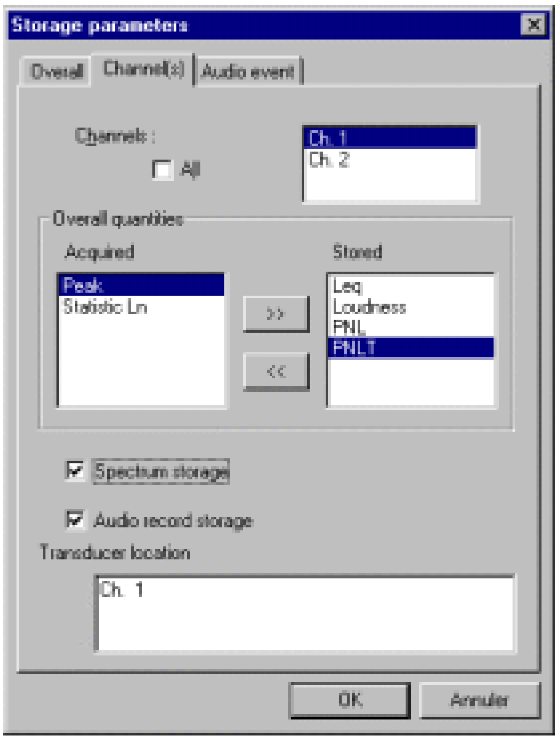
13.4.1. Acquisition and storage parameters

When a microphone has been selected at the hardware configuration stage, it becomes possible to select the acquisition of psychoacoustic criteria.

Use the command **Setup / Parameters / Acquisition** and select the **Channel(s)** tab. This dialog box appears on-screen.



Tick the box **Psychoacoustics** and click on the Parameters key to define additional calculation parameters for the Loudness (**free field** or **diffuse sound field**) and PNLT levels (tonality correction for a **plane** or a **helicopter**).



In order to store psychoacoustic criteria in a measurement file, use the command **Setup / Parameters / Storage** and select the **Channel(s)** tab. This dialog box appears on-screen.

In the list of **acquired** overall quantities, select as appropriate the **Loudness, PNL, PLNT** quantities and use the key **>>** to pass them into the stored quantities list.

In the measurement window, display psychoacoustic criteria time histories by using the **command Display / Time history**.

13.4.2. PNL, PNLT criteria and acoustical assessment of aircraft

The PNL and PNLT criteria are used by the civil aviation in order to evaluate the subjective response of human beings affected by the noise of jet-powered aircraft and helicopters (in flight) in neighbouring communities around airports. They are also used to calculate the EPNL level for certification of aircraft.

The main criterion for certification of aircraft is the Effective Perceived Noise Level **EPNL**, expressed in **EPNdB** units, used to evaluate the subjective response of human beings affected by aircraft noise. The computation of this criterion require a measurement of the Perceived Noise Level **PNL** weighted by a pure-tone correction factor (it gives a Perceived Noise Level Tone corrected **PNLT**).

The procedures to measure and compute these criteria are available from the International Civil Aviation Organisation ICAO. There are described in the *Environmental Technical Manual on the use of Procedures in the Noise Certification of Aircraft* (Doc 9501).

These criteria apply to stationary (or permanent) noises.

Here is an overview of the definition and the measurement procedure of these criteria computed in **dBTRIG32** according to the ICAO document.

13.4.2.1.PNL : Perceived Noise Level

The initial aim of the Perceived Noise Level criterion was to express in comparable units the adverse effects on human beings of the noise from jet-powered engines and helicopters, which spectra are different. It only deals with the subjective effect that disappears after exposition.

The **PNL** is expressed in **PNdB** (perceived noise decibel) over a logarithmic scale. For a linear scale, this quantity is denominated by the letter **N** and is expressed in **Noys**.

PNL computation is based on a real-time measurement of a third-octave spectrum. The measurement parameters are fixed by the ICAO technical manual.

Measure a real-time third-octave spectrum from 50Hz to 10 kHz, weighted with the time constant Slow Instantaneous, and for an acquisition time base of 500ms.

For each time base and for each third-octave frequency band, the Slow Instantaneous time weighted sound pressure level **SPL** is converted to a perceived noise **n(i)** by means of a conversion table.

The overall perceived noise N is given by the formula :

$$N(k) = 0.85n(k) + 0.15\left[\sum_{i=1}^{24} n(i,k)\right]$$

Where:

k represent a time base (an instant)
i represent a given third octave frequency band (24 bands from 50Hz to 10kHz)
n(k) represent the greatest value of n(i,k) over the whole spectrum
N(k) represent the overall perceived noise

The perceived noise N(k) is then converted into a perceived noise level PNL(k) for each time base by using the formula:

$$PNL(k) = 40 + \frac{10}{\log 2}\log N(k)$$

13.4.2.2.PNLT : Perceived Noise Level Tone Corrected

If the measured SPL level spectrum features uneven spectral components (for example a masking effect: perception of a sound is completely masked by a louder noise, of different frequency), experience proved that the noise is more annoying that the equivalent noise for a continuous spectrum and for the same perceived noise level value.
This correction is a function of the greatest level in a given third octave frequency band, with respect to the levels in the adjacent frequency bands. If the level difference is greater than 5dB for neighbouring bands, a correction of 4 to 5 PNdB could be applied to the PNL calculation.

Spectral irregularities (for example, the maximum value for discrete frequency bands or pure tone) are characterised by a correction factor **C**. The computation of this correction factor is standardised by the International Civil Aviation Organisation.

This correction factor is calculated from the sound pressure level in the third-octave frequency band centred on **80 Hz for jet-powered aircraft and on 50Hz for helicopters**. We then calculate the sound pressure fluctuations of the other third-octave frequency bands.
By an iterative method, as a function of the sound level differences between frequency bands, a corrected sound pressure level is calculated for each band. We then consider the difference between the sound pressure levels before and after applying the correction, taking into account non-negligible values only. For each non-negligible third-octave frequency band, a pure-tone correction factor is calculated.

The greatest of these values is the pure-tone correction factor C, that should be added to the calculated perceived noise level PNL.

The perceived noise level tone corrected PNLT is then given by the formula:

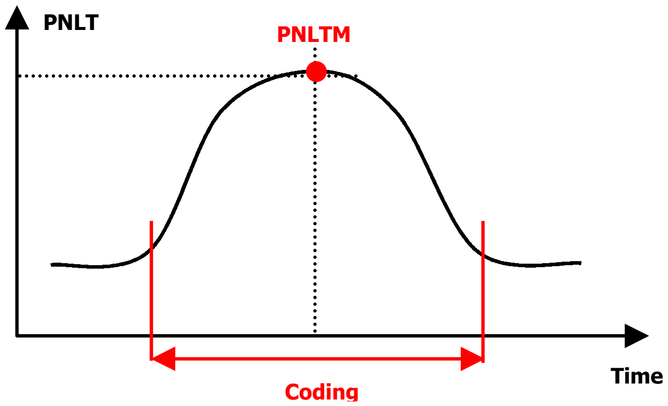
$$PNLT(k) = PNL(k) + C(k)$$

13.4.2.3.EPNL : Effective Perceived Noise Level

The effective perceived noise level EPNL is equal to the instantaneous perceived noise level, PNL, corrected for spectral variations (Perceived Noise Level Tone corrected PNLT) and duration correction factor.

The EPNL calculation method, based on PNL and PNLT measurements, is described below:

1. For each PNLT value of the duration of a plane event, calculate the maximum value PNLTM



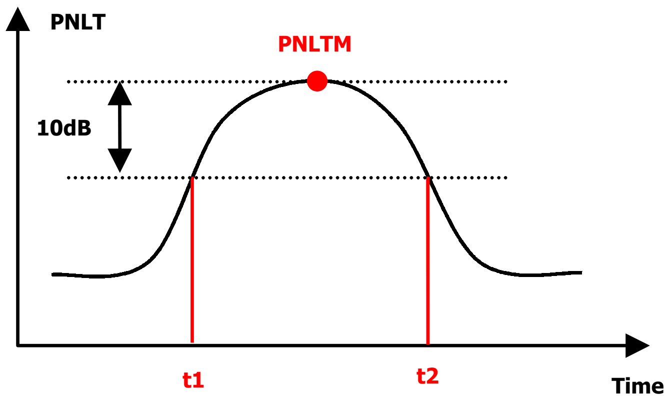
2. Calculate a duration correction factor D from the PNLT values.

The duration correction factor D is given by the formula:

$$D = 10\log\left[\frac{1}{T}\int_{t1}^{t2} anti\log\frac{PNLT}{10}\,dt\right] - PNLTM$$

Where:

T is a standardised time constant
PNLTM is the maximum value of PNLT over the event
t1 is the first time when PNLT is greater than PNLTM - 10
t2 is the time when PNLT becomes less than PNLTM - 10



3. Calculate the effective perceived noise level EPNL by the formula:

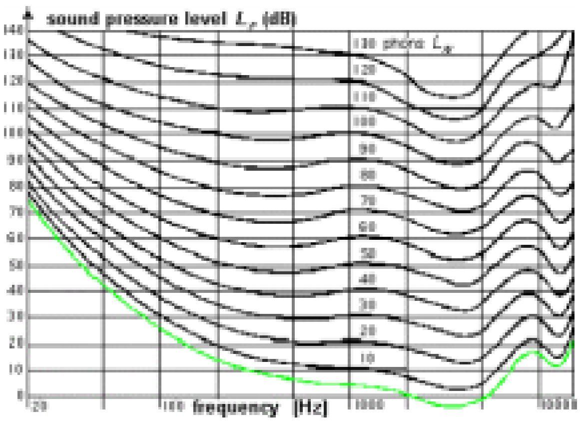
$$EPNL = PNLTM + D$$

This criterion is not calculated in dBTRIG32, because it cannot be calculated in real-time.

13.4.3. Loudness criteria

Loudness is a measure of sound energy and is measured in *Sones*. It represents the intensity of the sound as perceived by the ear. As a reference, 1 sone is the level given by a pure sound of 1 kHz at 40 dB. This level increases by powers of two.

In this way, a relationship between the dB scale and sones may be established for 1kHz. For the other frequencies, the sound level and frequency are altered to give the same impression of Loudness. This method may appear user dependent, however studies have shown considerable independence of this factor from the results



The resulting Isosone graphs are composed of points corresponding to frequency levels that have equal Loudness obtained from pure sounds.
To compute in real-time the Loudness level of complex sounds, **dBTRIG32** use the standardised method described in the standard ISO R532B. Refer to this standard for more information.

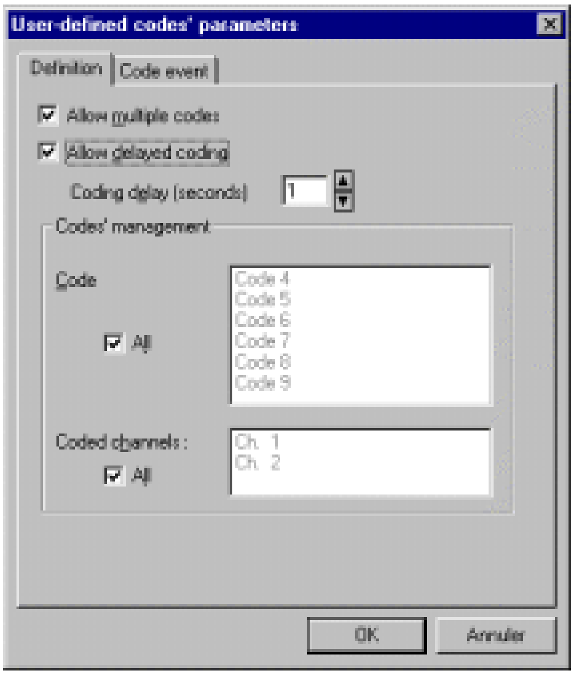
13.5. Expert mode

The expert mode of **dBTRIG32** allows the user to define any combination of trigger conditions for recording audio or spectrum events and generate alarm signals. Thresholds can be either relative or absolute providing very flexible event detection and data storage. Noise data can also be coded when a user-defined threshold is exceeded and according to several codes simultaneously.

At the contrary of the other optional modules, the expert mode correspond a software user level (see **paragraph 2.3**).

The following paragraphs describe how to use and configure the software in expert mode. **Refer to the chapters of this manual relative to the expert mode for more detailed information**.

13.5.1. Coding noise data with multiple codes



In expert mode, it is possible to code noise data simultaneously according to several codes. Use the command **Setup / Parameters / Source coding** and select the **Definition** tab to define these parameters. The following dialog box is displayed on screen.

On top of the "usual" coding operations, tick the box **Allow multiple codes**.

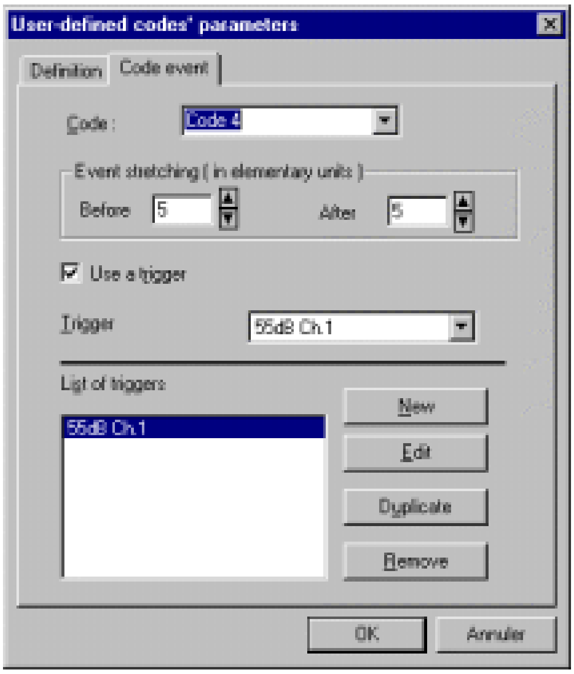
If this function is not selected, a coding operation in progress would be automatically stopped when the operator uses a different code.

Simultaneous Coding

Standard Coding

During a measurement, several coding operations may be performed simultaneously if this function is activated.

13.5.2. Coding events (threshold triggering)



In expert mode, it is possible to code noise data when a threshold condition is fulfilled. Use the command **Setup / Parameters / Source coding** and select the **Code event** tab to define these parameters. The following dialog box is displayed on screen.

For a given **code** (Code 4 in this example), the operator can **use a trigger** (threshold) in order to automatically code noise data with the code 4 during a measurement session.

An **event stretching before and after** the trigger state is TRUE can also be defined.

In this example, and for an acquisition time base of 100ms, the 500 ms of noise data preceding and succeeding the event will be coded as a Code 4 noise source when the measured Leq level on channel 1 is greater than 55dB.

For each available code, the user may define different parameters (event stretching, trigger used). See paragraph 13.5.5 for more details on the definition of a threshold trigger.

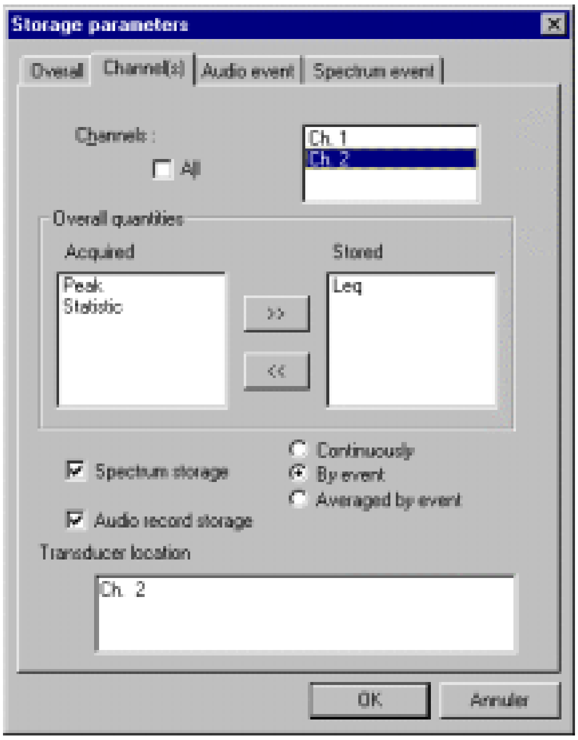
Beware that when using a threshold condition to trigger a noise event, delayed coding should not be activated (in the definition tab) otherwise, the software will start coding at the delayed coding cursor position. The coded noise data will not correspond to the event because of that delay.

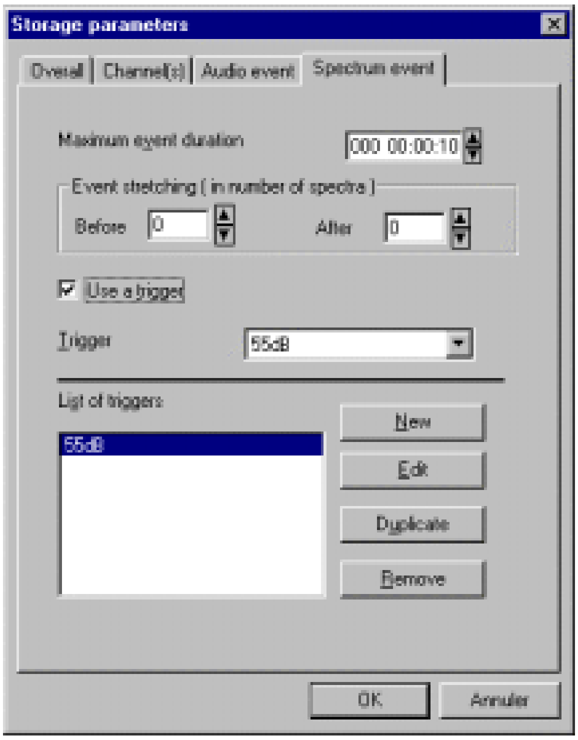
13.5.3. Spectrum events

In expert mode, it is possible to store the acquired spectrum in octaves and third octaves continuously or by event.
For spectrum event data logging, store either the spectrum averaged over the event duration or the multispectrum (one spectrum per acquisition time base).

Use the command **Setup / Parameters / Storage** and select the **Channel(s)** tab to define these parameters. The following dialog box is displayed on screen.

Tick the box **Spectrum storage** and choose the data logging mode (**continuously, by event or averaged by event**).





Then select the tab **spectrum event** to define the triggering parameters to acquire the event. A threshold or a clock trigger may be used.

Definer the maximum event duration, an event stretching before and after the trigger condition is fulfilled and the trigger to use.

📖 *See paragraph 13.5.5 for more details on the definition of a threshold and/or a clock trigger.*

13.5.4. Triggering alarm events

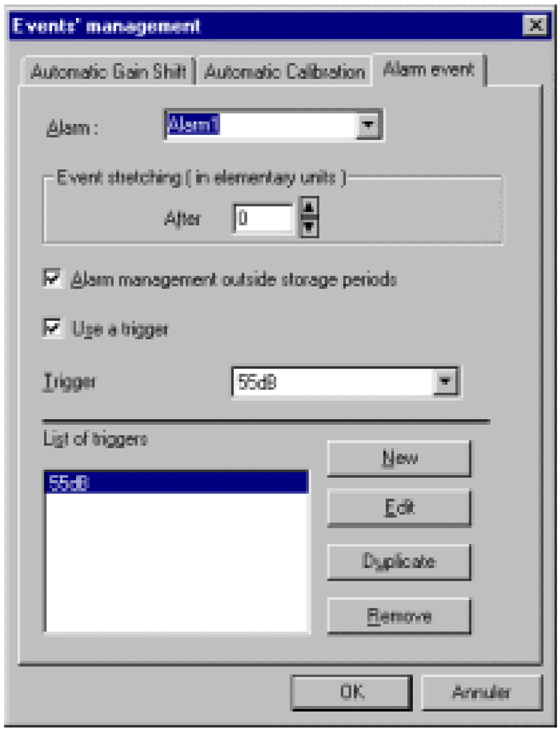
In expert mode, it is possible to trigger a light alarm when a threshold condition is fulfilled and by using a remote control object (See chapter 12). E.g.: flashing light system connected to the serial port of the PC enabled when a noise level is exceeded).

Once the measurement chain is set up (definition of the remote control to enable alarm triggering), use the command Setup / Parameters / Advanced parameters and select the alarm event tab. This dialog box appears on screen.

Select the active **alarm** (**Alarm1** in this example). Define an **event stretching** in number of elementary units after the true event, knowing that an elementary unit correspond to the time base of acquisition in **dBTRIG32**.

Define as well if the alarm event will be triggered when no data logging is in progress.

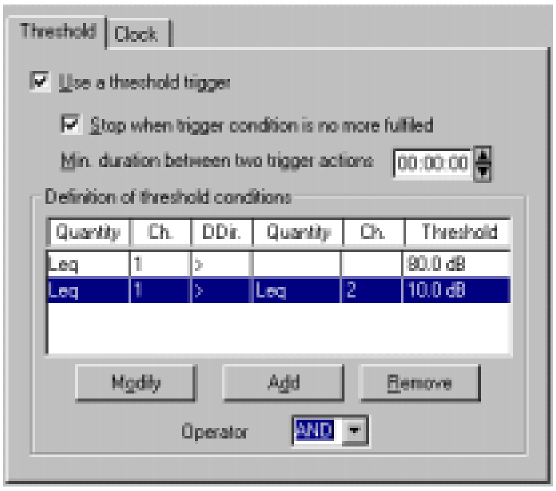
Select the trigger used (threshold conditions) to start an alarm.



📖 *See paragraph 13.5.5 for more details on the definition of a threshold trigger.*

13.5.5. Event triggers in expert mode

In expert mode, the definition of threshold trigger conditions is much more evolved than in **dBTRIG32** standard mode.



To define the threshold conditions of a trigger, use the key **New** or **Edit** to access this dialog box. Select the threshold tab.

To define **threshold conditions**, first click on the **add** button to define one condition, then repeat this operation for as many threshold conditions as required. When several conditions are defined, a boolean operator (**AND/OR**) has to be defined. A threshold trigger will be activated when all the threshold conditions are fulfilled (**AND operator**) or when at least one condition is fulfilled (**OR operator**).

In this example, the trigger will be active when the Leq measured on channel 1 is greater than 80dB AND when the Leq level on channel 1 is 10dB greater than the Leq level on channel 2.

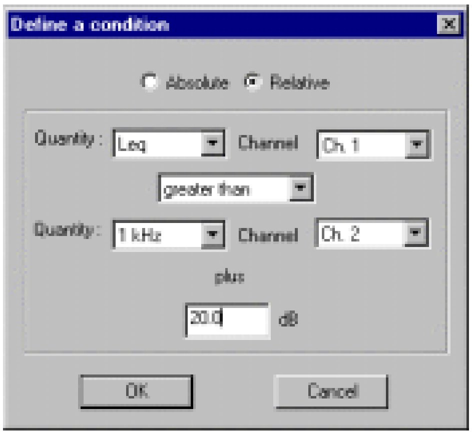
Two types of threshold may be defined : **absolute** or **relative**.

13.5.5.1.Definition of an absolute threshold

First define the **quantity** considered (choose between all the acquired quantities), the measurement **channel** considered, the way of the threshold (**greater than or less than**) and the threshold **value**.

In this example, the threshold condition will be fulfilled when the Leq level measured on channel 1 is greater than 80dB.

13.5.5.2.Definition of a relative threshold



The threshold condition will be fulfilled by comparing the two acquired quantities.

Define the quantities, the measurement channels, the way and the relative threshold value.

In this example, the threshold condition will be fulfilled when the Leq level measured on channel 1 is greater than the sound pressure level of the 1kHz third octave band measured on channel 2.

13.6. Use of a sound level meter as an acquisition front-end in dBTRIG32 (ACL mode)

The ACL mode allows using a 01dB sound level meter as acquisition platform in dBTRIG32.
The sound level meter data are sent in real time to the computer via serial interface.

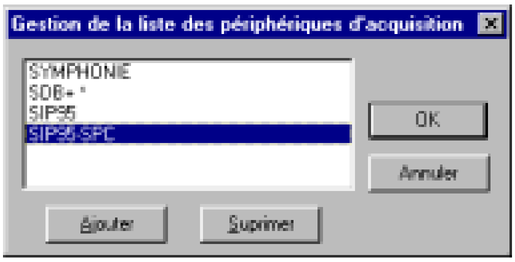
The sound level meter sends calibrated data to dBTRIG32.

With the ACL mode one use the high **storage** capacities given by the computer. The user can carry out long-term measurements even on sound level meters without storage function. A continuous surveillance can be included without any interruption of the measurement.
Now there are the coding of 6 noise sources and the event management available in **dBTRIG32**.

The **ACL mode limitations are the same as for the used sound level meter**.
Only with SLM including the real time option the user can obtain a spectrum history in dBTRIG32.
No SLM connected to dBTRIG32 allows audio signal records.

13.6.1. Definition of a SLM hardware platform

The user has to define a new platform in the 01dB hardware platform database with **dBCONFIG32**, in order to use a 01dB sound level meter as acquisition front-end with **dBTRIG32**.



Click on Hardware platform to display the Hardware platform database management box.

Then click on Add to display the hardware peripheral table.
Select a SLM to be connected as an acquisition head.

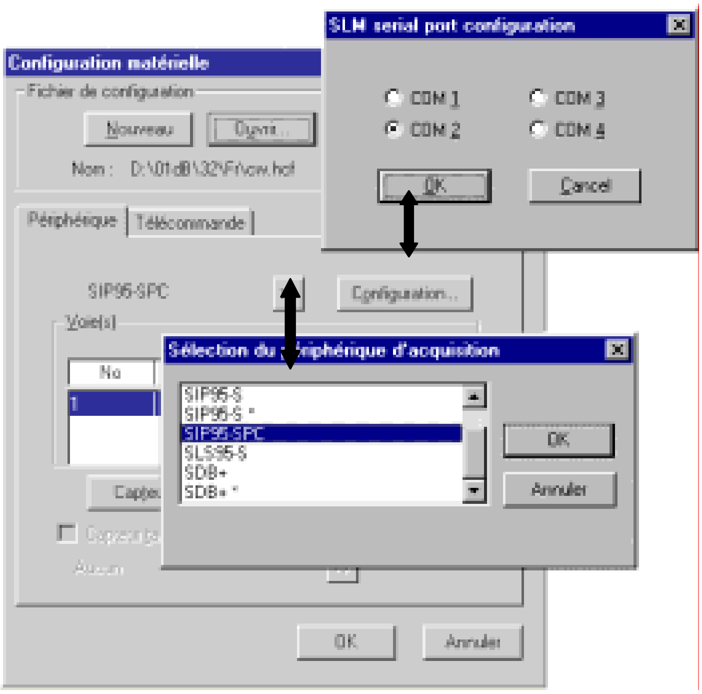
The three main ranges of 01dB sound level meters are available in ACL mode: **SIP95**, **SLS95** and **SdB**.
The sound level meters SdB without the Leq option are not available.
The available sound level meters are defined in the dBCARD32.INI file.

SPC = option analyse fréquentielle temps réel
** = option gamme de mesure 20-100 dB*

Click on OK to validate. **The chosen sound level meter is now available in dBTRIG32**.

13.6.2. Hardware configuration in dBTRIG32

Choose **Hardware configuration** in the **Setup** menu.



Choose the SLM in the list previous defined in **dBCONFIG32**.

The only setting for a hardware platform is the input of the **serial interface number**.

dBTRIG32 impose the input of a transducer (and a possible calibrator) at least on one active channel to carry out an acquisition.

So the user has to define a transducer like for a "classical" acquisition.
The selection of a transducer has no importance, because the sensitivity values are not taken into account for the measurement. In fact, **the SLM provides data already calibrated**.
Later, there is no calibration possible in **dBTRIG32**.

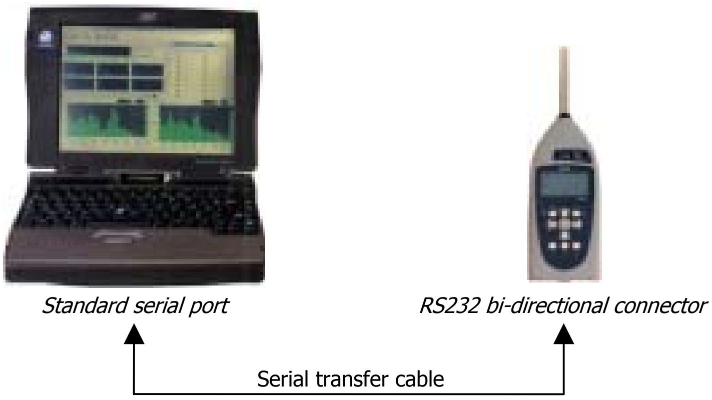
13.6.3. Sound level meter calibration

Before any measurement a calibration is needed, in order to carry out precise measurements.
The ACL mode is the only operation that needs a sound level meter manipulation during measurement.
📖 **Refer to chapter "Sound level meter calibration" of your SLM user manual to follow the procedure.**

13.6.4. Connection – Switching on

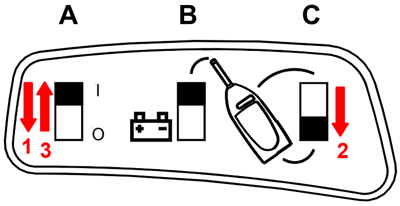
- Connect the sound level meter to the serial interface of the computer :

- **SdB** : use CTA09 cable
- **SLS / SIP** : use CTS009 cable (provided with the device)



- Switch the device on for transfer :

SLS / SIP



- **Switch off** the sound level meter: Start/Stop switch (A) down.
- Put the sound level meter in **external operating mode**: SLM switch (C) down.
- **Switch on** again the sound level meter: Start/Stop switch (A) up.

SdB

- **Switch off** the sound level meter
- **Press** the VAL key (in the middle) when switching on the instrument

The communication mode of the SLM is blocked now and all keys are desactivated. The screen appears as mentioned on the opposite.

Communication
in progress

There is no setting needed (parity, transfer rate …), because it is fixed in the instrument.

All the following operation will be carried out in dBTRIG32.

13.6.5. Acquisition in dBTRIG32

The user can now carry out an acquisition in **dBTRIG32** with an hardware platform. A lot of functions are not accessible, because they are not compatible with the SLM use: calibration (has to be carried out with the sound level meter), audio records, psycho-acoustics, vibration weighting …

With the **dBTRIG32** storage functions the user can carry out long-term measurements (monitoring…).
There are 6 source numbers for noise source coding available that can be activated with a trigger.

We use the SIP95 sound level meter with frequency analysis in ACL real time mode in the opposite example..

14. TROUBLESHOOTING

Below are a few common problems and solutions encountered with **dBTRIG32**. For other problems, please consult the troubleshooting section of the getting started manual that was delivered with your system before calling technical support.

Nothing happens when launching a measurement session in dBTRIG32.

- Check that the hardware platform has been correctly installed and configured on the computer and in the software utility **dBCONFIG32**.
- Check in the hardware configuration dialogue box that a microphone / calibrator couple has been declared and enabled.

The measured levels are identical whatever the noise level..

- Check that the transducer and extension cable are correctly connected to the acquisition platform and that the transducer is plugged to the active measurement channel.
- Check that, for SYMPHONIE and the JAZZ acquisition card, the correct signal conditioning options (Click on **configure** in the hardware configuration dialogue box) are activated if using a 200V polarised microphone.

The measured level seems incorrect

Before anything else, perform calibration as explained in chapter 5. If the problem is not solved, proceed as follow.

Check your microphone. Microphones are fragile equipment that can be damaged by a (small) fall or water. First, check its sensitivity. You may also check the appearance of its membrane. To do so, unscrew the protection grid. If the surface is uneven or scratched, the microphone is damaged.

Check the calibrator. Over the years, the signal amplitude of the calibrator may vary. It will therefore induce a systematic error in all measurements. Send the calibrator back every two years to the manufacturer to test it.

Check the extension cable. Directly plug the microphone to the acquisition unit (if possible) and compare the measured values. If a large difference is observed, the cable may present a problem. Send it back to the manufacturer.

Check the hardware signal conditioning options (SYMPHONIE and JAZZ). Please, refer to the getting started manual that was delivered with your system for details.

Check transducers' and calibrators' definitions in dBCONFIG32. A simple test is to define a new transducer and calibrator in dBCONFIG32 using the factory characteristics. Then perform a calibration of the system. Compare the sensitivities of the old and new transducers in dBCONFIG32. If they are greatly different, it may come from the definitions in dBCONFIG32. In this case, remove the old transducer and calibrators from the databases. **Do not forget that a great difference of sensitivity could also mean that the microphone is damaged**.

No values are displayed.

- Check that the colour of the curves are not the same as the colour of the background.
- Check that the analyser has been switched on.

15. MEASUREMENT DATAFILES IN DBTRIG32

15.1. Data storage type

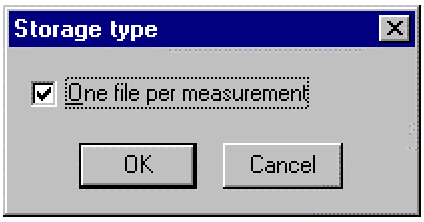
It is possible to store the measurements in two different ways:

- One data file per measurement session
- One global datafile for all the measurement session

Use the command **Preferences / Storage type** when there is no measurement window opened, to select the type of datafile naming.

The 16-bit version of 01dB software modules are using a lot of different file formats where noise and vibration data, acquired with different programs, are stored (for example, LEQ files for time varying noise quantities, WAV files for audio records, FC files for averaged octave and third octave spectra, etc.). **The new 32-bit software modules are now using only one type of data file to store any acquired quantity. They are called measurement session files (*CMG)**

15.1.1. One data file per measurement session (default)



By default, **dBTRIG32** creates a datafile per measurement.

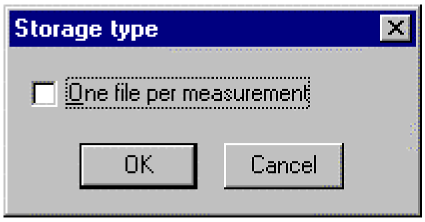
If the generic datafile name is TEST, **dBTRIG32** will create a measurement session file for each measurement. Its name contains the date and time of the measurement.

Syntax :TEST_year/month/day_hour/minute/seconds.CMG

For example, the file **TEST_981015_175444.CMG** corresponds to a measurement started 15 October 1998 at 17h 54 min 44 sec.
TEST_981015_1755 is the mesurement started one minute after.

Advantages : Analysis can be performed while mesurement continue (**monitoring**...)

15.1.2. One global datafile for all the measurement session



It is also possible to store all the measurement sessions (for a given configuration file) into the same datafile.

In this case, the file will be named after the generic file name only TEST.CMG

Advantages : One unique CMG file (copy, backup...).

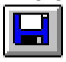
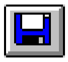
15.1.3. Binary data files

*.**BID** files are also created (Binary Item Data). It contents binary data associated to the measurement session.
The name of this files is : session name + increment number + .BID extension

So you must use the command **Delete session(s)** in your analysis software (dBTRAIT32...) to delete a session and its BID files.

It is possible to concatenate CMG and BID files in one unique CMG file using dBTRAIT32 software.

15.1.4. Overview of datafile names

One data file per measurement session	One global datafile for all the measurement session
	
Example of a measurement serie all 5 minutes from 6 o'clock p.m.  : SERIE	 : SURV
Measurement session files SERIE_981015_180000.CMG SERIE_981015_180500.CMG SERIE_981015_181000.CMG etc... (according to the number of executed records)	**Measurement session file** SURV.CMG
Binary files SERIE_981015_180000_1.BID SERIE_981015_180000_2.BID SERIE_981015_180000_3.BID etc... (according to the number of recorded items) SERIE_981015_180500_1.BID SERIE_981015_180500_2.BID SERIE_981015_180500_3.BID etc... (according to the number of recorded items) and so on for all sessions...	**Binary files** SURV_1.BID SURV_2.BID SURV_3.BID etc... (according to the number of items recorded)

15.2. Datafile management

dBTRIG32 contains a specific file management routine since the signal files not only take up a lot of space on the hard disc but the number of recordings is unknown at the start of the measurement session, (particularly if the recording is programmed for threshold trigger conditions) dBTRIG32 always gives priority to storage of time level data : **any measurement of time data will be carried out thoroughly** - It means that data storage of user-selected time quantities will always be completed.
The file management procedure is shown below:

15.2.1. First degree tests

- **Storage parameters**

In order to protect the data from any problems that develop during acquisition, the user must define the recording parameters before any data storage on the computer hard disk occurs.

dBTRIG32 checks the available memory on the hard disk, taking into account the measurement length, the acquisition data type and quantity.

- **Acquisition parameters**

An identical test is performed each time the time basis of acquisition is modified (if a measurement data file name has been defined.

Each time a test is performed, a warning message is displayed on the screen if there is insufficient memory capacity for the measurement, under current parameters. If this occurs, reduce the measurement duration, free some memory space on the hard disk or reduce the number of time quantities to store.

15.2.2. Second degree tests

At the start of audio recording and during the acquisition, dBTRIG32 carries out checks on available hard disc memory for audio recordings and for complete measurement of time related quantities.

If the memory space is insufficient and a threshold trigger was used dBTRIG32 automatically switches to manual audio trigger. If manual trigger was defined, and insufficient memory is apparent, an error message is displayed.

This enables the user to free some memory space during acquisition and re-start the trigger mode by using the audio parameters command.

Alternatively, by using threshold conditions, an audio record begins as soon as the preceding one end but it can be created later. If memory shortage occurs during the interval between two audio records, it could mean that there is not enough space to record overall noise data.

15.3. Structure of measurement session data files (*.CMG)

Because of the increasing needs for data logging of a great variety of noise and vibration quantities in environmental and industrial applications (multi-channels, long term monitoring, wide range of units, transducers, measured static and dynamic quantities, etc.), **01dB developed a new data file format well adapted to store, display and process measurement results.**

All the previous datafile formats have therefore been dropped and a unique file format that can address all our applications (and much more) has been developed. Based on a database structure, it rationalises the edition, the display, the processing and the various "Office" operations that can be performed on measurement results in 01dB PC-based measurement chains.

This chapter describe the new structure of the datafiles as well as all the processing operations that depend on it.

15.3.1. Measurement session

The first requirement in the definition of this format was to identify a structure of the "container" type, intuitive enough for being used by an non-experienced operator, and flexible enough to store all type of noise, vibration and other quantities measured by 01dB systems. We therefore decided to call this type of file "measurement session".

Any 01dB application software module will therefore store, handle and display MEASUREMENT SESSIONS: Genuine containers of a set of measurements performed for environmental, industrial and building acoustics applications.

A measurement session is therefore defined by its name (generic root name), a comment (to amply describe the context of a measurement).

But what do we found in a measurement session file and how do we handle the data?

Going down one level in the structure, we reach the heart of the data file structure, that is the elementary entity used to store a given type of data. **This generic entity is called an ITEM. Its structure is identical for all quantities stored in a measurement session**.

Physical data storage of the items is not shown to the user. Depending on the requirements of the application software used, the measurement sessions, which reference in its header all these item boxes, will contain the complete set of measured quantities in a single file or in individual files (one by item). In both cases, later computing operations (file copies, architecture on the computer hard disk, etc.) will be easier to integrate and to perform.

15.3.2. Item

Let us now describe this structure : the item is a stand-alone quantity that must be able to contain any type of data sets (description, metrological and reference information, actual measurement values) in order to address the present and future measurement applications.

A non-exhaustive list of items currently stored as items is given below.

The following quantities, measured by 01dB systems, can be stored as items of a measurement session data file:

- Audio records
- Impulse responses acquired by MLS technique
- Time histories of overall quantities: Overall levels (frequency and time weighted), civil aviation criteria (PNL,PNLT), psychoacoustic criteria (Loudness, Sharpness, etc.), spectral time history
- Autospectra and interspectra, in real or complex narrow bands, time averaged or not, computed in real time or off-line
- Broad band spectra (form octave to 1/48th octave resolution), computed in real time or off-line
- Bark band spectra (specific Loudness)
- Transfer functions (cross-spectra, coherence, etc.)
- Noise source codes
- Measurement chain information (dynamic range selection, overload, etc.)
- Comments
- Table of results (psychoacoustic)
- Histograms
- Echograms
- Room acoustic criteria
- etc.

The container therefore has a database structure well adapted to handle batch processings, for example.

15.4. Measurement session file size calculation

15.4.1. General formula

The user may calculate manually the datafile size before starting a measurement, in order to estimate the hard disk space required to store the measurement (**dBTRIG32** automatically calculates the file size required to perform a complete measurement.

The formula below can be used to calculate the file size of measurement session CMG, when concatenated (no binary files):

$$T_{CMG}(bytes) = \left(\frac{2 \times N_1 \times D}{DI}\right) + Y + N_2 \times T_{AUDIO} + N_3 \times T_{OCTAVE}$$

Where:

- **T_{CMG}:** measurement data file size (in bytes)
- **DI (in seconds):** acquisition time base selected by the user (from 10ms to 1s)
- **N_1: number of stored quantities.** Selected by the user during definition of the storage parameters.
- **D :** complete measurement duration (in seconds)
- **Y : constant**. The value of this constant can vary as a function of several parameters. This constant contains the file header with the number of active channels, the transducers' location, the audio recording references, etc. can vary from 1kB to 50 kB.
- **N_2 :** number of audio events
- **T_{AUDIO} :** size of an audio event (see below)
- **N_3 :** number of spectra events
- **T_{OCTAVE} :** size of a spectrum event (see below)

For dual-channel measurements, multiply the overall file size by 2, if identical settings on both channels are selected or calculate the data size for each channel and add them, if independent settings are selected (the constant excepted).

Example 1 : A measurement with :

- One channel
- 100ms time base
- Leq and Peak data logging
- 10 days of continuous measurements
- No events
- Constant Y ≈ 2000 bytes

T_{CMG} = ((1/0,100) x2 x2 x 10 x 24 x3600) + 2000 = 3 456 000 bytes

15.4.2. Audio event size

The audio files can fill a hard disc very quickly as they contain all the signal information required for detailed analysis. The size of these files depend not only on the length of recording but on the sampling rate.

The formula below can be used to calculate the file size of an audio record:

$$T_{AUDIO}(bytes) = Fe \times 2 \times D$$

Where:

- **T_{AUDIO}** : Size of the audio event in bytes
- **Fe** : Sampling frequency in Hertz. This frequency depends on the pass band and has a minimum value of 2,2 times the pass band. This factor varies according to the acquisition cards used. See the following list :

Hardware	Pass Band (Hertz)	Factor	Sampling Rate (Hertz)
Sonata PRO	20000	2,4	48000
M942	20000	2,56	51200
JAZZ	20000	2,4	48000
SYMPHONIE	20000	2,56	51200
Multimedia card	20000	2,205	44100

- **D :** duration of the record in seconds

Example 2 : An audio event with :

- Symphonie : factor 2.56
- Pass band : 20 kHz
- Record duration : 10 seconds

T_{AUDIO} = ((20000 x 2.56) x2 x10)+ 500 = 1 024 500 bytes
(Around 1 Mbytes every 10 seconds)

Therefore, with the preceding result, if we had 50 audio records over the 10-day measurement session, the overall file size would be about **85.8 Mbytes.**

15.4.3. Spectrum event size (expert mode)

The formula below can be used to calculate the file size of a spectrum event (not averaged):

$$T_{OCTAVE}(bytes) = \left(\frac{2 \times N \times D}{DI}\right)$$

With :

- **T_{OCTAVE}:** size of the spectrum event in bytes
- **DI (in seconds):** acquisition time base selected by the user (from 10ms to 1s)
- **N :** number of frequency bands.
- **D :** duration of the event (in seconds)

Example 3 : A spectrum event with :

- Time base : 100ms
- Stored spectrum : third octave bands from 20Hz to 20kHz (31 bands)
- Event duration : 60 seconds

T_{OCTAVE} = ((1/0,100) x2 x 31 x 60) = 37200 bytes

Therefore, with the preceding result, if we had 100 spectrum events over the 10-day measurement session, the overall file size would be about **89.5 Moctets**.

15.5. Abstract : Files used by dBENV32 (dBTRIG32 + dBTRAIT32)

01dB sotwares use or create different type of files :
- **Program files**
- **Parameter files**
- **Data files**

- **Program files**

These files are an integral part of the measurement system. They combine libraries of functions (***.DLL files)** that are common to several programmes, executable files **(*.EXE files)** for each specific software modules and driver files (***.DRV files**) specific to each acquisition platform. The drivers establish a communication protocol between the computer, the software module and the acquisition unit.
The programme files also contain the software **help files (*.CHM**)

By default, they are located in **C:\ PROGRAM FILES\01DB\PROGRAMS**. The drivers of the acquisition cards are located under **C:\WINDOWS\SYSTEM.**

- **Parameter files**

These are initialisation and configuration files containing the measurement parameters, the list of characteristics of the different measurement chain elements and the measurement chain configurations of different applications, for example. These files are also located in the same directory as the programmes but are not delivered with the installation CDROM.

When the acquisition software is first used, a file is created to save the essential parameters. If this file is destroyed, it will be necessary to configure the software and hardware again.

See the sketch below.

- **Data files**

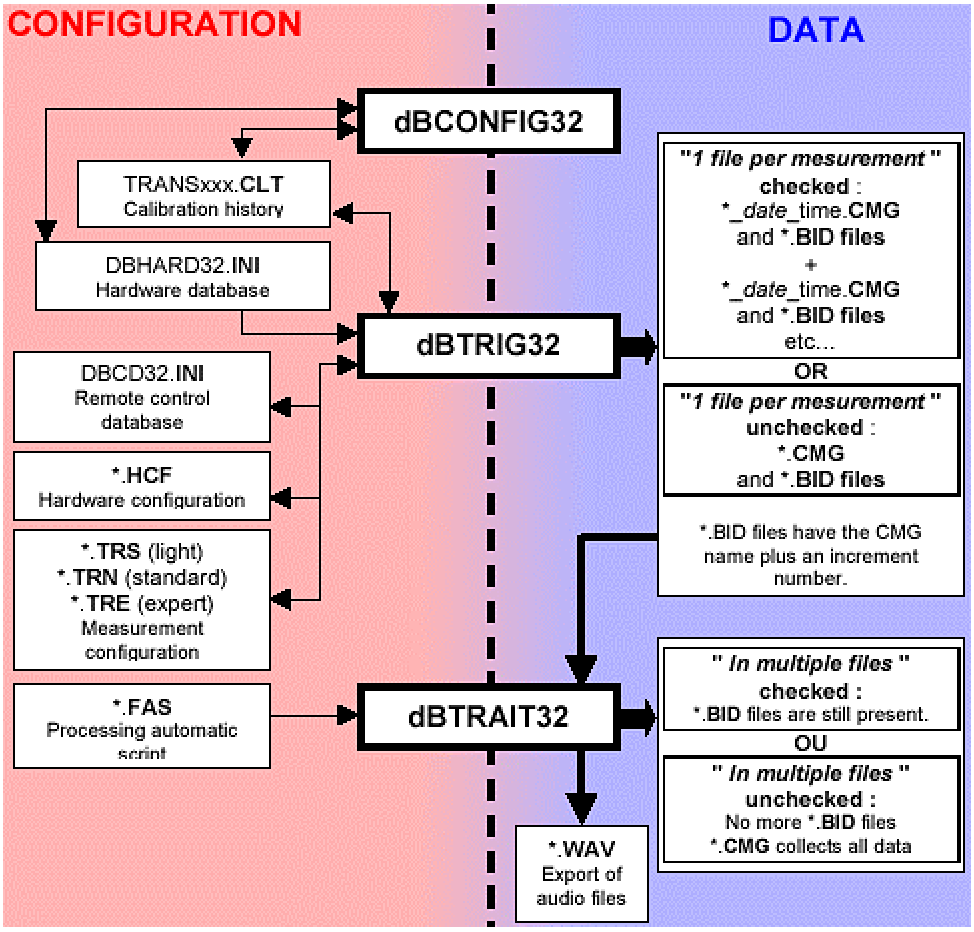
These files contain the measurement results. They can be located anywhere on the computer hard disc. A directory **C:\ MY DOCUMENTS \ 01DB MEASUREMENT SESSIONS** is created by default.

For users that have evolved from a 01dB measurement system running with 16-bit applications to a system running with 32-bit applications, the format of data files has evolved. The files can be imported with the command **File / Import** in the measurement software.

To avoid deleting everything by mistake we advise not to put the measurement files under the same path directory as the programme files.

See the sketch below.

Configuration and Data files used by dBENV32 (see below for further information).



More information are available in :
- **dBCONFIG32** on-line help
- **dBTRAIT32** on-line help or manual

In this manual, see pages :
- 12.2 - Description of dBCD32.INI (remote control database file **dBCD32**.INI)
- 3 - Measurement hardware Configuration (*.**HCF** files use)
- 4 - Measurement configuration files(*.**TRN** and *.**TRE** files use)
- 15.1 - Data storage type (**CMG** / BID files management)

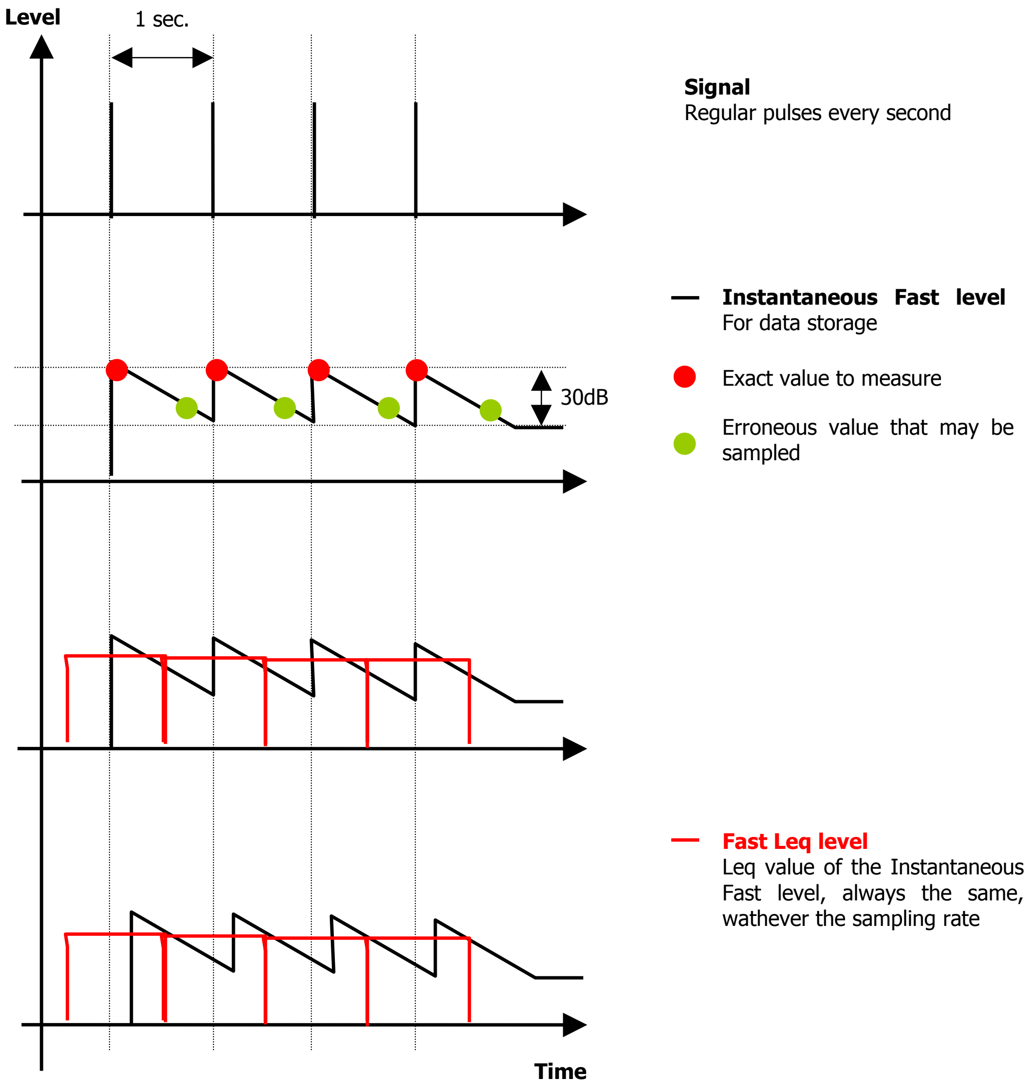
16. APPENDIX

16.1. Calculation of time-weighted sound pressure levels in dBTRIG32

In **dBTRIG32**, the calculation of time weighted sound pressure levels SPL, overall or per frequency band, can be performed in two ways :

- According to the classical method implemented a wide range of sound level meters (see glossary)
- Or according to a continuous equivalent sound pressure level Leq calculation of this quantity.

Let us consider the computation of a Fast sound pressure level. The diagram below illustrates the difference between the measured **Fast** level (corresponding to a Fast Leq level) and the **Fast Inst.** level, acquired at a rate of 1 second:



If the measured signal is made of regular impulses (one impulsion per second), the measured Fast Leq level will match closely the original signal.

The measured Instantaneous Fast level cannot be correct because, as we sample the signal according to discrete values, the measured Fast Inst. level has little chance to match the real value of the impulse.

For a Fast time weighting, the falling slope is equal to 30dB / second. Therefore, in our example, if only one value per second is sampled, the Fast Instantaneous level measured may vary over a 30dB range.

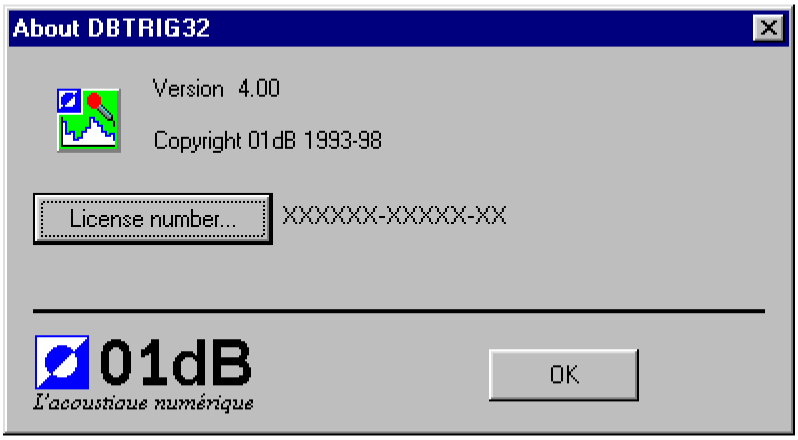
At the contrary, if we calculate the Fast Leq level, we are considering an averaged level of the instantaneous Fast level. We will therefore obtain the same value, whatever the sampling rate, because of the energy conservation principle.

Furthermore, we are then able to calculate a cumulated value of the Fast Leq level, as well as a minimum (Fast Min) and maximum (Fast Max) values.

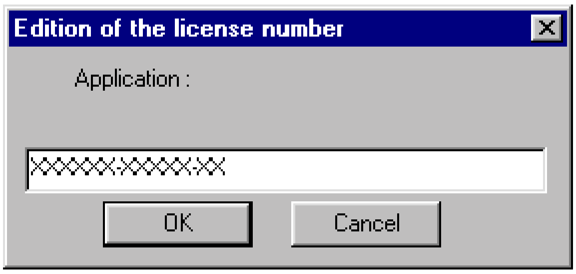
The differences between the two calculations are not as marked as for this example for real-life signals and a shorter integration time.

16.2. Edition of the software licence number

Use this command (menu **? / About dBTRIG32**) to obtain general information on the software version, copyright and licence number.



If the licence number of the software module has to be modified, click on the key **Licence number**. The following dialog box appears on screen:



Enter the new licence number, provided by 01dB technical support.

Start again the application software in order to account for this modification.