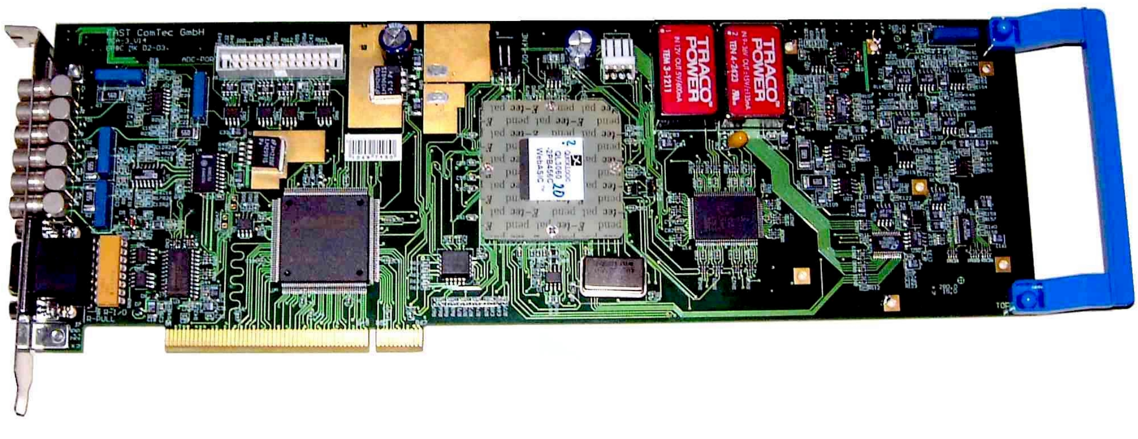
MCA-3 Series / P7882

Multichannel Analyzer / Dual Input Multiscaler

User Manual

Grünwalder Weg 28a, D-82041 Oberhaching
Germany



Version 3.5, März 7, 2016

Warranty Information

FAST ComTec warrants proper operation of this software only when used with software and hardware supplied by FAST ComTec. FAST ComTec assumes no responsibility for modifications made to this software by third parties, or for the use or reliability of this software if used with hardware or software not supplied by FAST ComTec. FAST ComTec makes no other warranty, expressed or implied, as to the merchantability or fitness for an intended purpose of this software.

Software License

You have purchased the license to use this software, not the software itself. Since title to this software remains with FAST ComTec, you may not sell or transfer this software. You must get FAST ComTec's written permission for any exception to this license.

Backup Copy

This software is protected by German Copyright Law and by International Copyright Treaties. You have FAST ComTec's express permission to make one archival copy of this software for backup protection. You may not otherwise copy this software or any part of it for any other purpose.

The information in this manual describes the hardware and the software as accurately as possible, but is subject to change without notice.

Important Information on Hardware Compatibility

The MCA-3 Series / P7882 Multichannel Analyzers / Multiscalers are PCI Local Bus compliant devices. As such the board contains the configuration space register organization as defined by the PCI Local Bus Specification. Among the functions of the configuration registers is the storage of unique identification values for our devices as well as storage of base address size requirements for correct operation specific to each of our products.

The host computer that our products are installed in is responsible for reading and writing to/from the PCI configuration registers to enable proper operation. This functionality is referred to as 'Plug and Play' (PnP). As such, the host computer PnP BIOS must be capable of automatically identifying a PCI compliant device, determining the system resources required by the device, and assigning the necessary resources to the device. Failure of the host computer to execute any of these operations will prohibit the use of the MCA-3 Series Multichannel Analyzers / P7882 Multiscalers in such a host computer system.

It has been determined that systems that implement PnP BIOS, and contain only fully compliant PnP boards and drivers, operate properly. However, systems that do not have a PnP BIOS installed, or contain hardware or software drivers, which are not PnP compatible, may not successfully execute PnP initialization. This can render the MCA-3 / P7882 Series inoperable. It is beyond the ability of FAST ComTec's hardware or software to force a non-PnP system to operate MCA-3 Series Multichannel Analyzers / P7882 Multiscalers.

IMPORTANT NOTEs

This manual is written for all of the several options of MCA-3 / P7882 boards available. Due to this fact most of the time the manual will talk of "MCA-3". In case you purchased a P7882 please replace "MCA-3" by "P7882" in your mind.

The MCA-3 Series Multichannel Analyzers / P7882 Multiscalers are available in a number of optional versions. Thus, depending on the version purchased some features described in this manual may or may not be implemented / functioning on your particular board.

Some models in the MCA-3 series (i.e. P7882, MCA-3, MCA-3FADC) offer the Multichannel Scaling (MCS) mode and some (i.e. MCA-3, MCA-3FADC, MCA-3A) the pulse height analysis (PHA) mode. The MCS mode is used for recording, into appropriate time bins, the time of arrival of pulses at the STOP inputs relative to the START input, over a specified time sweep length. The PHA mode is used for analyzing and sorting analog input pulses by amplitude, either via an external Analog-to-Digital Converter (ADC) or via a fast internal ADC (only present on MCA-3A and MCA-3FADC).

WARNINGS

Damage to the MCA-3 / P7882 board, the computer or injury to yourself may result if power is applied during installation.

Static electricity discharges can severely damage the MCA-3 / P7882. Use strict antistatic procedures during the installation of the board.

Take care to provide ample airflow around the MCA-3 / P7882 board.

Take care not to exceed the maximum input values as described in the technical data.

Particularly the COUNT inputs are very high speed, high sensitivity inputs and thus, susceptible to oscillation. Take care to apply low impedance (≤ 50 Ω) source signals and well shielded, 50 Ω cables.

Table of Contents

Table of Figures

1. Introduction

The MCA-3 Series / P7882[1] PCI board is a versatile family of advanced multichannel analyzers / dual input multiscalers. The design incorporates more features and functions than any other MCA in its price range. The backbone is a fast MCA[2] with 512 k onboard memory channels and 32 bit capacity each which allows to accumulate 100% in the background. The memory can also be configured as a large FIFO buffer for listmode data acquisition.

A standard nuclear ADC port allows to connect a large number of external devices like ADCs, TDCs etc. The data width is 16 bit corresponding to 64 k channels. The design is capable to collect external ADC data at rates of up to 5,000,000 events/s.

Features available include a built-in pulse height analyzing ADC with 500 ns conversion time and 8 k conversion range. The throughput capacity is in excess of 1,000,000 events/s depending on the shape and duration of the input pulses.

Multiscaling mode is far advanced and offers two count inputs with 100 ns bin width (200 ns in dual input mode) and counting rates far in excess of 350 MHz. The two counting inputs allow to acquire - for example - two Moessbauer spectra at a time from a single Moessbauer drive. This is a very economical solution requiring just one Moessbauer system. The very short dwell times starting at 100 ns make the MCA-3 ideal for many types of spectrometers and LIDAR applications where high count rates are required as well as high sweep repetition rates.

A versatile 8 bit digital I/O[3] port may further satisfy your experimental needs. Usage as TAG inputs allows for a wide range of spectra routing, multi detector experiments, sequential acquisition etc.

Two software configurable SYNC outputs provide synchronization and triggering of external devices or experiment monitoring.

A presettable 32 bit sweep counter in MCS[4], deadtime and livetime counters in PHA[5] mode enable for exact normalization calculations.

An open-drain 'GO'-line (compatible to other products of FAST ComTec) allows for overall experiment synchronization.

The high-performance hardware is matched by sophisticated software delivered with each MCA-3. MCDWIN - the MS-WINDOWS™ based operating software - provides a powerful graphical user interface for setup, data transfer and spectral data display.

Some of MCDWIN´s features are high-resolution graphics displays with zoom, linear and logarithmic (auto)scaling, grids, ROIs[6], Gaussian fit, calibration using diverse formulas and FWHM[7] calculations. Macro generation using the powerful command language allows task oriented batch processing and self-running experiments.

"C"-Libraries are available for controlling functions from the user's application program. A DLL (Dynamic Link Library) is available for operation in a Laboratory Automation environment.

---

[1] From here on, the manual will refer to MCA-3 / P7882 most of the time as "MCA-3" only. If you purchased a P7882 please replace "MCA-3" by "P7882" in your mind.

[2] Mulichannel Analyzer

[3] I/O: Input / Output

[4] Multiscaling

[5] Pulse Height Analyzer

[6] ROI: Region Of Interest

[7] FWHM: Full Width at Half Maximum

2. Installation Procedure

2.1. Hard- and Software Requirements

The MCA-3 requires a personal computer (with INTEL™ compatible processor) with an available PCI slot.

A Pentium II or higher processor and at least 64 MB of memory are recommended.

A Microsoft WINDOWS NT 4.0, XP, Vista, 7 or higher operating system must be installed.

2.2. Hardware Installation

Turn off the power to your computer system and remove the line cord. Discharge your body from any static electricity by touching a grounded surface – e.g. the metal surface of the power supply – before performing any further hardware procedure.

FAST ComTec assumes no liability for any damage, caused directly or indirectly, by improper installation of any components by unqualified service personnel. If you do not feel comfortable performing the installation, consult a qualified technician.

WARNING

Damage to the MCA-3 board, the computer or injury to yourself may result if power is applied during installation.

Static electricity discharges can severely damage the MCA-3. Use strict antistatic procedures during the installation of the board.

Open the cover of the computer case and insert the MCA-3 PCI board in an unused PCI slot. You might first have to remove the cover from the rear of the PCI expansion slot you selected. After the board is carefully seated in the PCI slot, make sure you fasten the board with a screw to the mounting bracket.

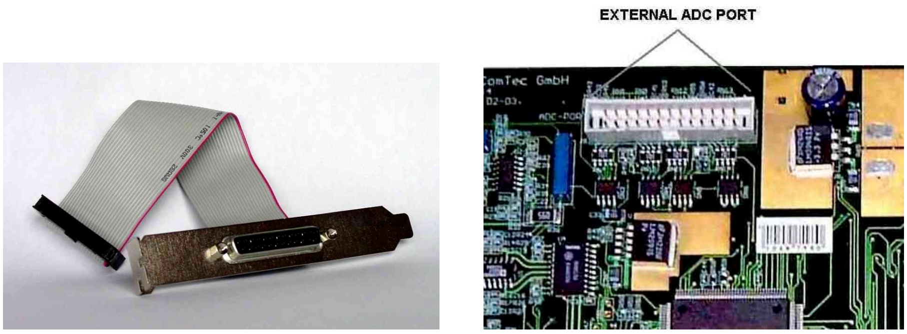


Figure 2.1: External ADC port connector

If purchased, install the external ADC port connector now. In some computer cases special fittings are available for some types of D-SUB connectors. These are particularly useful to save slots for additional plug in boards. Otherwise mount the housing bracket with the 25-pin D-SUB connector in another available slot of your computer. Plug in the 26-pin socket connector (at the end of the ribbon cable) into the 26 pin four-walled header named ADC PORT on the MCA-3 PCI board.

2.3. Driver Installation

To install the driver, please start the device manager. You can do it by right-clicking the computer shortcut on the desktop and selecting “Manage” from the drop-down-menu, then “Device Manager”. Another way is to select in the Start Menu Settings..Control Panel, then System..Device Manager.

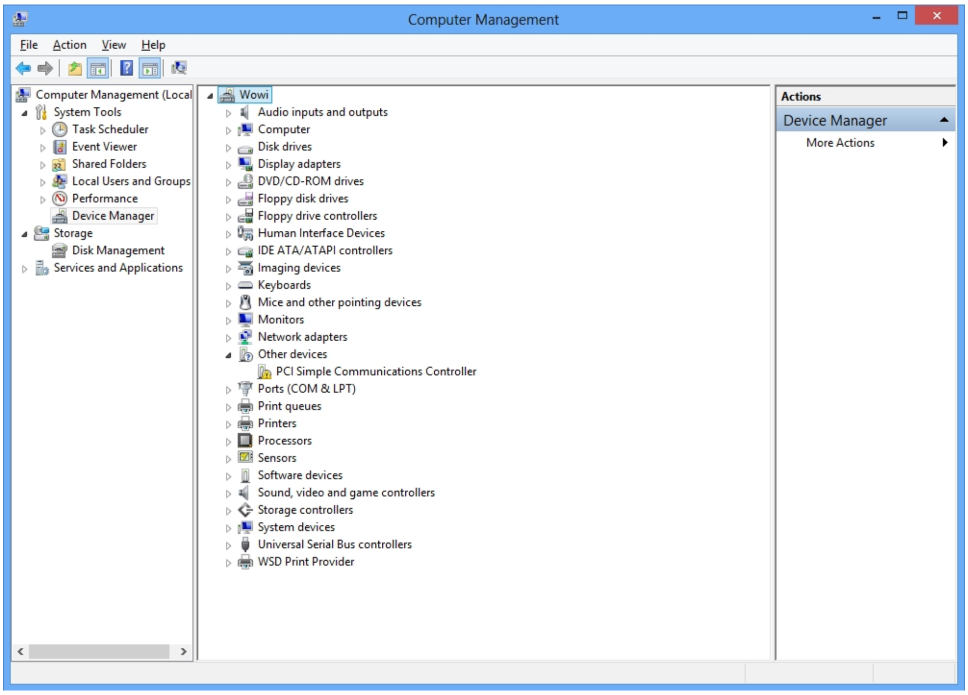


Fig. 2.2: Device manager

The new device is displayed as “PCI Simple Communication Controller” in a class “Other devices”. Please right-click it and select “Update driver” from the drop-down menu.

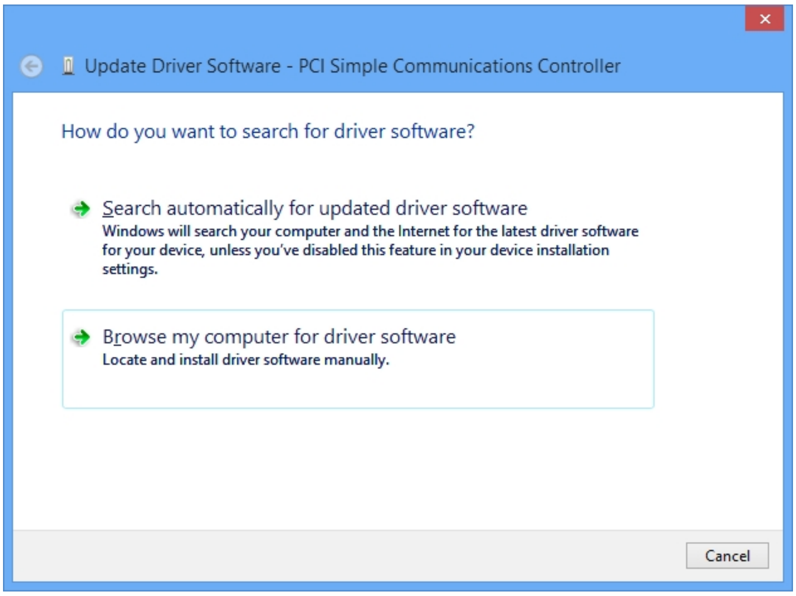


Fig. 2.3: Driver installation

Windows offers to search automatically for updated driver software or browse the computer for the driver software. Don’t try the automatic installation, it will probably not work. Select to browse manually for the driver and select for example for Windows 7 or 8 the wdmdriv\win7 folder on the CD or Software stick.

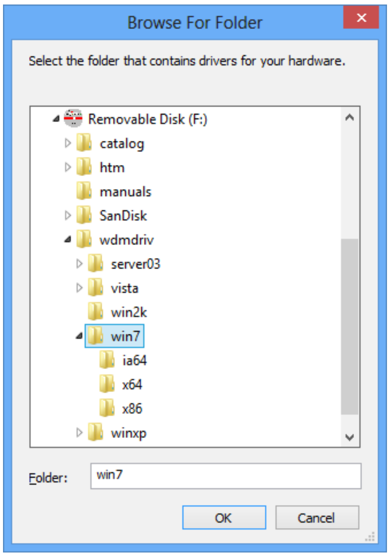


Fig. 2.4: Choosing the right folder

Please don’t go deeper into the folder structure, the processor type (x86 , x64, etc.) is found automatically. After pressing “OK” the driver installation will proceed. The device manager will then show a “FAST ComTec MPA-3, MCA3 and P7882..P7888 Driver” in a class “Multifunction adapters” as shown in Fig. 2.5

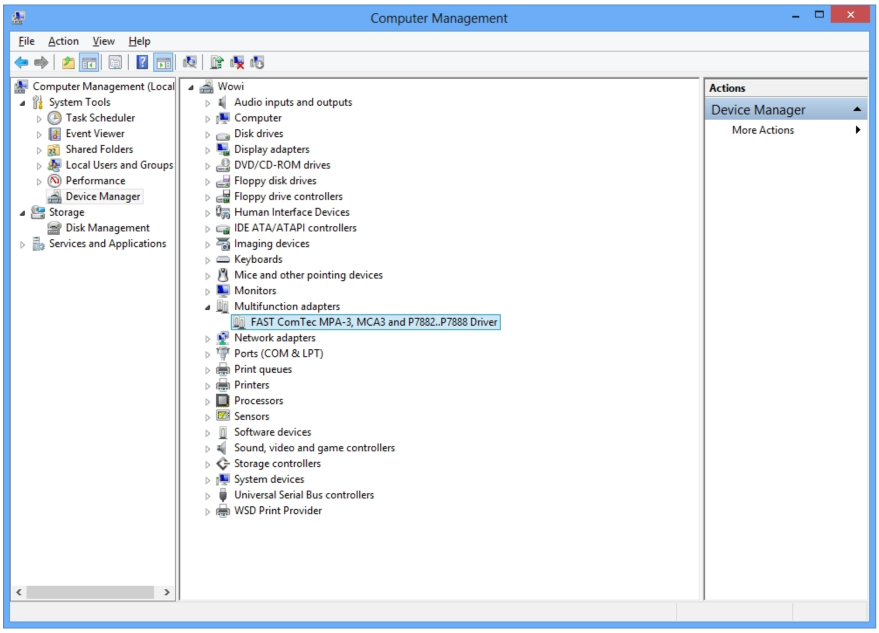


Fig. 2.5: Device Manager after successfull driver installation

2.4. Software Installation

To install the MCA-3 software on your hard disk insert the installation medium (CD or USB-Stick) and start the installation program **setup.exe** for the 32-bit software or **setupx64.exe** for the native 64-bit software.

A directory called C:\MCA-3 for the 32 bit software or C:\MCA-3(x64) for the 64 bit software is created on the hard disk and all MCA3 and MCDWIN files are transferred to this directory. Drive C: is taken as default drive and \MCA-3 or \MCA-3(x64) as default directory. It is not mandatory that the P7888 operating software is located in this directory. You may specify another directory during the installation or may copy the files later to any other directory.

The Setup program has installed one shortcut on the desktop that starts the MCA-3 server program. The server program will automatically call the MCDWIN.EXE program when it is executed. The MCA-3 Server program controls the MCA-3 board but provides no graphics display capability by itself. By using the MCDWIN program, the user has complete control of the MCA-3 along with the MCDWIN display capabilities.

If you have more than one MCA-3 modules installed, edit the line devices=1 in the file MCA3.INI and enter the number of modules.

To run the MCA-3 software, simply double click on the “MCA-3“ icon. To close it, close the MCA-3 server in the Taskbar.

2.5. Getting Started with some basic Measurement

To ease getting familiar with the use of the MCA-3 we will now setup some basic measurements in MCS mode. We use a simple TTL signal generator to supply START and STOP signals.

First we will measure the arrival time of multiple STOP events in a time window of 400 µs after a START (Trigger) pulse. The time resolution (bin width) shall be 100 ns. The measurement should run for exactly 100,000 sweeps (scans, shots) until it ends. With an input signal frequency of 100 kHz the resulting spectrum is suggested to look like a garden fence with peaks every 10 µs or 100 time bins.

First let's setup up the wire connections to the board and then start the software to run the measurement.

2.5.1. Connecting the test signals for a MCS measurement

The generator should be able to drive two 50 Ω inputs to some hundred millivolts and should not exceed ± 5 V as not to exceed the absolute maximum ratings of the inputs. For this, a 50 Ω power splitter divides the 100 kHz TTL signal into two branches. The two output signals of the power splitter are connected to the ± 5 V discriminator TRG and COUNT 1 inputs on the PCI bracket (ref. Figure 2.7: MCA-3 bracket connectors).

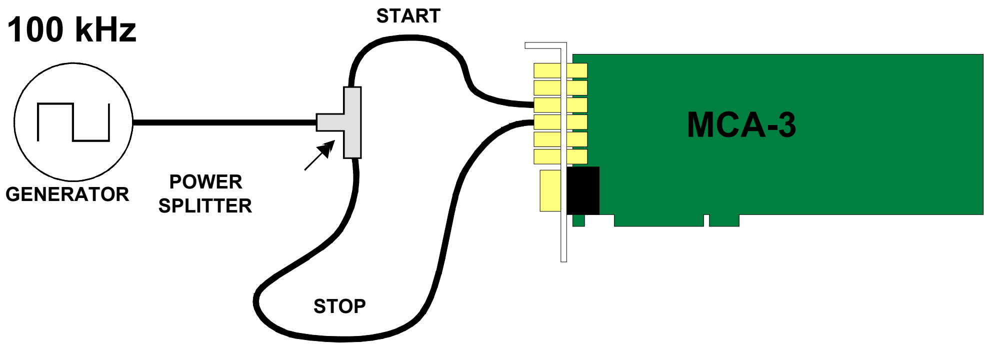


Figure 2.6: Basic measurement setup

2.5.2. Starting MCDWIN and setup for the measurement

Next step is to start the MCA-3 software by double clicking the corresponding icon. This will automatically start the MCDWIN program. On startup a picture appears that shows the bracket connectors (ref. Figure 2.7) and their meaning. You can review this picture any time by clicking on File - About… in the MCA-3 server program.

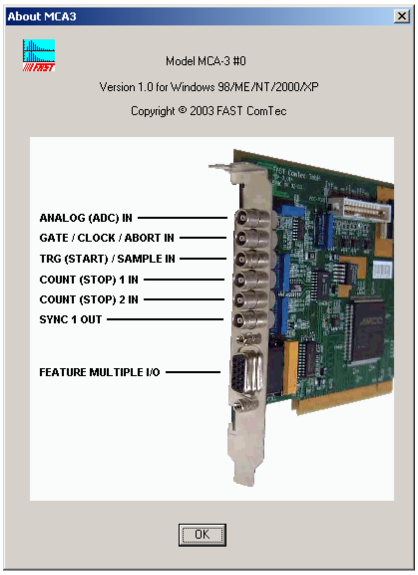


Figure 2.7: MCA-3 bracket connectors

The MCA-3 Server is iconized then and one does not have to worry about it since all hardware settings are also accessible from the MCDWIN program which actually is the graphical user interface and which will appear now on your screen (ref. Figure 2.8).

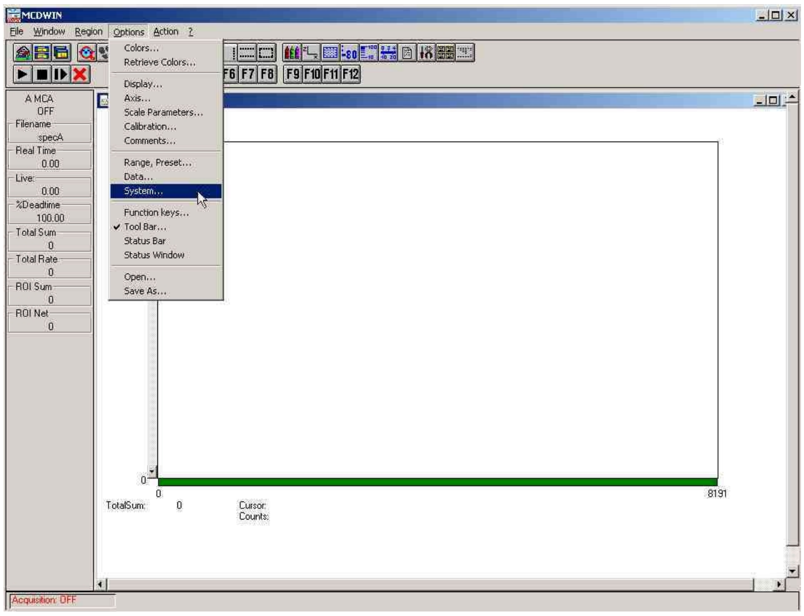


Figure 2.8: MCA-3 / MCDWIN startup window

Now we first have to setup the MCA-3. First we need to set it to MCS mode. Click on Options – System … to get the System Definition window pop up. Set Mode to MCS (ref.: Figure 2.9) and click OK.

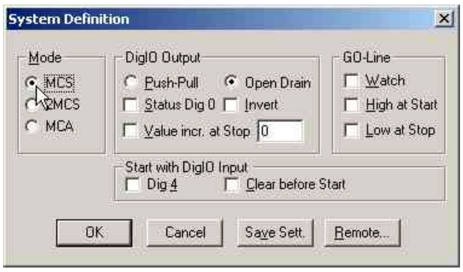


Figure 2.9: System Definition window

Then click on Options – Range, Preset … to find the MCA-3 Settings window pop up. Set the Range to 4000 time bins and Dwelltime to 100 ns which corresponds to the desired 400 µs time range.

Enable the sweep preset and type in the number of sweeps as 100,000 (ref. Figure 2.10). Take care to have "Use SCA" disabled.

Then click on Inputs to select the desired input threshold levels.

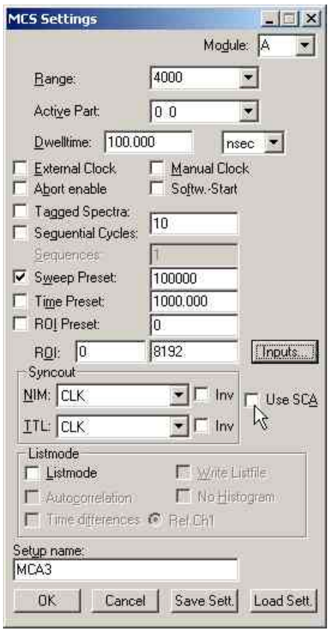


Figure 2.10: MCS Settings window

Select the Start and Stop inputs and set them to 'Customized' and a voltage level corresponding to your signal amplitude (e.g. +0.800 V, ref. Figure 2.11 and Figure 2.12). For best results use a 50 Ω Impedance setting. You also may select either edge settings. Now click OK to get back to the MCS Settings window. Again click OK.

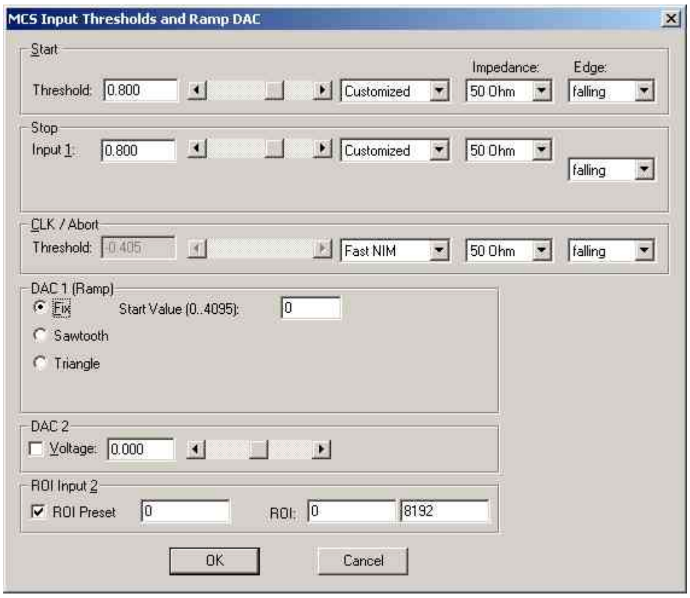


Figure 2.11: MCS Input Threshold window

To verify the quality of the discriminated signals you might select SAMPLE/TRG resp. COUNT 1 on the Fast NIM[1] SYNC output (ref. Figure 2.10) and connect the SYNC_1 output to an oscilloscope. Take care to terminate the cable with 50 Ω. Now you can online watch the effect of changing the input thresholds.

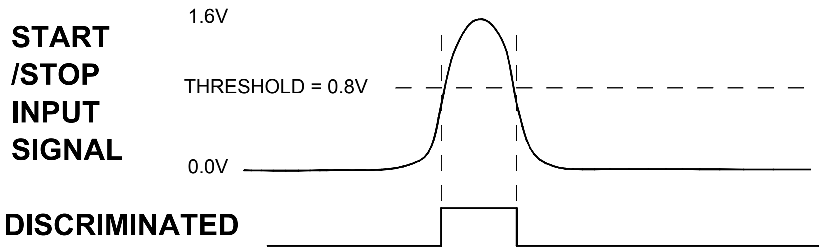


Figure 2.12: START/STOP discriminator levels

Let's change the display to have a grid and the axis numbered. Click on Options – Axis…. Enable the grid and the axis ticks (ref. Figure 2.13). Also enable 'Use Calibration' to see the x-axis in time units rather than channels. Then click OK.

[1] NIM: Nuclear Instrument Modules. This is a standard for mechanical and electronic properties of such modules.

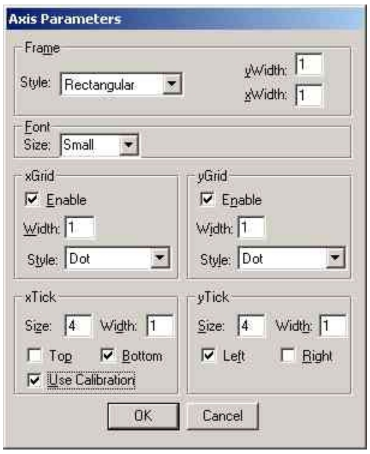


Figure 2.13: Axis Parameters window

Now lets setup the scale calibration feature to see the actual time data in the spectrum. Click on Options – Calibration…. and make sure 'Use Calibration' is enabled (ref. Figure 2.14), 'Unit:' is 'usec' and the calibration formula is set to 'p0 + p1*x' with 'p0 = 0' (offset) and 'p1 = 0.1' (time bin width).

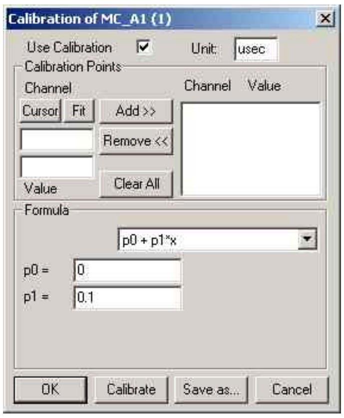


Figure 2.14: Calibration of MCA-3

The hardware is initialized properly now and also the display should appear as in Figure 2.15. To start the measurement now click on the Start button.

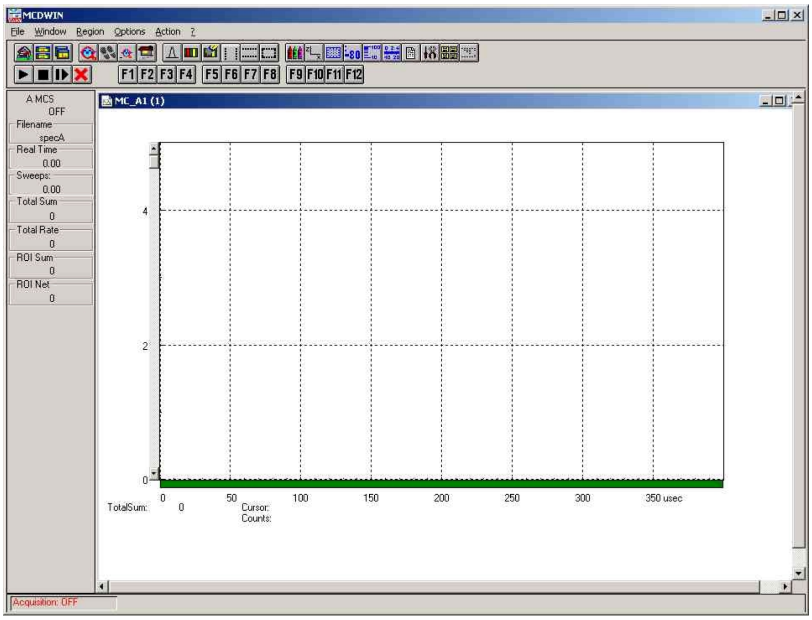


Figure 2.15: MCDWIN properly setup

The measurement will begin to run and ends when 100,000 sweeps are done. The resulting spectrum should look as in Figure 2.16. The peaks are separated by 100 channels or 10 µs. The sweep counter shows that exactly 100,000 sweeps have been acquired. Each peak should contain 100,000 events (one event per sweep).

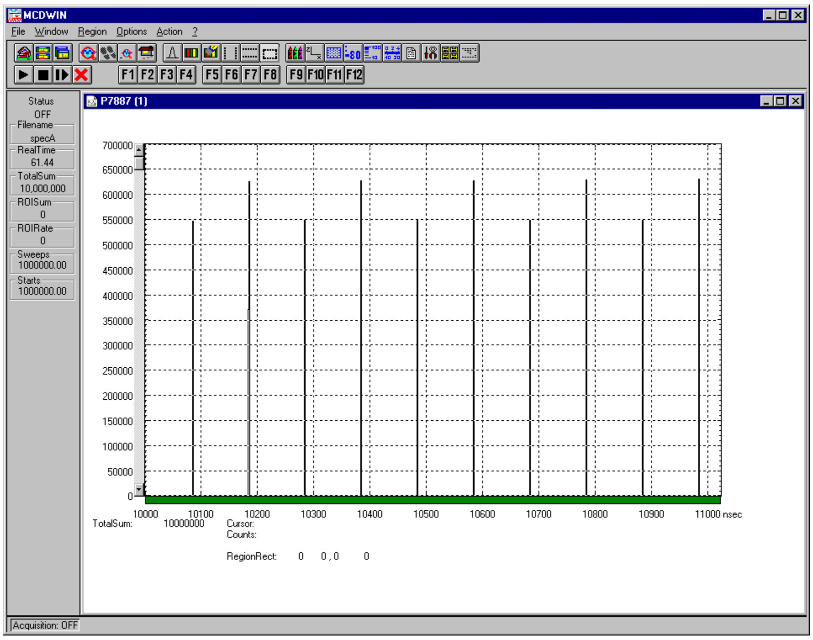


Figure 2.16: Resulting spectrum of the basic measurement

2.5.3. Signal Setup of the analog input

The analog input can be used for pulse height analysis using the internal ADC or for MCS mode using the SCA (Single-Channel-Analyzer) as count source.

The important point is to properly setup the THRESHOLD, LOWER and UPPER LEVEL DISCRIMONATORS (ref. Figure 2.17 and Figure 3.13). This is quite similar to any nuclear ADC.

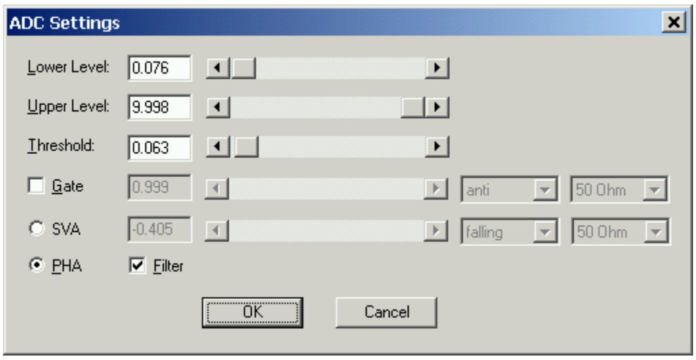


Figure 2.17: Analog input Settings

The threshold level is defined as that voltage at which the input signal is regarded to be above noise. The lower and upper level provide a voltage window that defines "valid" (interesting) pulses. Pulses that have peak values falling outside this (e.g. energy) window are discarded. Thus, you only convert/count pulses inside the LLD/ULD window.

3. Functional Hardware Description

3.1. Overview

The MCA-3 is a family of full size PCI PC board based multichannel analyzers. All settings are software selectable. No jumper, switch, etc. configurations are necessary. The large 512k x 32 bit data memory can be segmented to accumulate successive measurements. Online reading of the stored data is possible without additional deadtime affecting the actual measurement.

A switchable input lowpass filter at the analog ADC input allows to improve the signal quality by reducing high frequency noise on the signal.

Additional features are four onboard discriminators. These enable the inputs to be adjusted for a large range of input signals. Two of them provide extremely high input bandwidth allowing for count rates in MCS mode of typically well over 200 MHz.

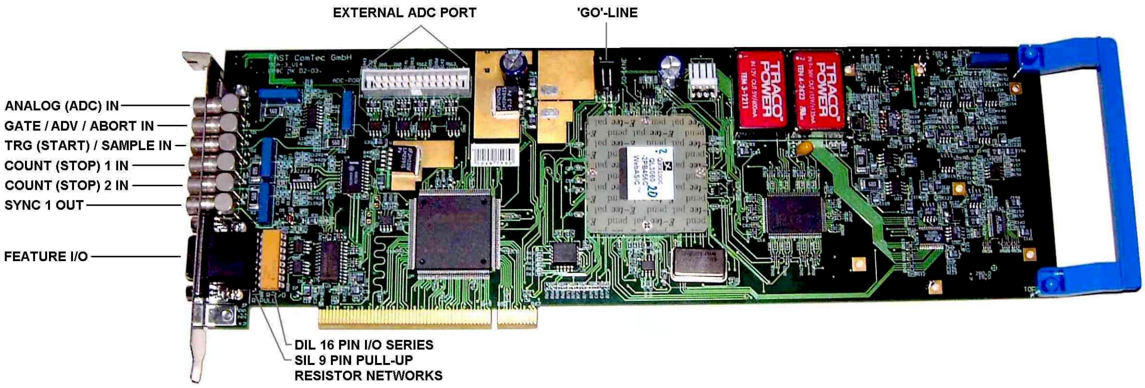


Figure 3.1: MCA-3 PCI board

Besides, two SYNC outputs with a large variety of output signal options (all software selectable) and the 'GO'-line (compatible to other FAST products) allow for easy synchronization or triggering of other measurement equipment.

Furthermore a versatile, user configurable 8 bit digital I/O port allows for a whole bunch of experimental control, monitor or whatsoever other tasks. Moreover, the 8 bit digital I/O port can be used for spectra tagging to allow for multi-detector configurations, sequential data acquisition etc.

Additionally two 12 bit analog voltage outputs are available. One of it can be configured to generate a sawtooth or triangle x-ramp voltage synchronous to the MCS time sweep.

3.2. Connectors

3.2.1. External ADC Port

The external ADC input port is a standard nuclear ADC interface that supports up to 16 bit (64k) ADCs, TOFs etc. The used Data Ready (DRDY) / Data Accepted (DACC) handshake is common to a large number of devices available. The principle timing is scetched in the picture below.

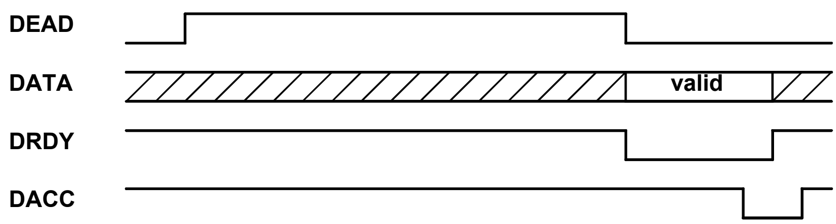


Figure 3.2: Principle DRDY / DACC Handshake

Since all control signal polarities are software selectable any known ADC, TOF etc. might be connected.

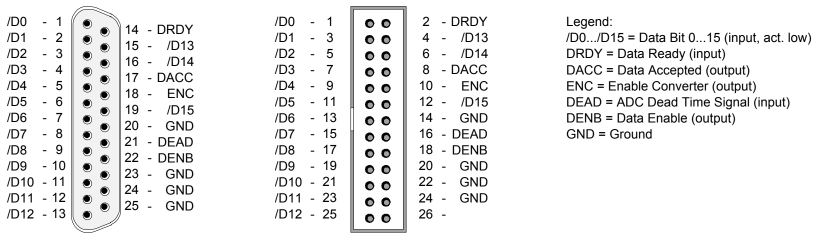


Figure 3.3: External ADC Port

3.2.2. Analog (ADC) Input

The analog signal input is a LEMO type connector on the mounting bracket. It accepts positive voltages in the range of 0 … +10 V. The input impedance is 1 kΩ, dc coupled. An optional 1.8 MHz lowpass filter can be enabled to allow for reduction of high frequency noise.

In pulse height analysis (PHA) mode it is optimized for Gaussian shaped pulses with shaping times in the range of 250 ns … 25 µs.

In MCS mode the associated single channel analyzer can also be used for counting (ref. chapter 3.4.4).

WARNING

Take care not to exceed the maximum input values as described in the technical data (ref. chapter 7.2.1).

3.2.3. GATE / ADV / ABORT Input

This input is used for either

- GATE the internal ADC
- external channel ADVance
- MCS sweep ABORT

The GATE / ADV /ABORT input is a LEMO type connector on the mounting bracket. It accepts signals in the range of ±5 V and provides a bandwidth of 100 MHz. The input impedance is a selectable 1 kΩ or 50 Ω. The threshold voltage is adjustable in the range of ±2.5 V. For clean switching and oscillation free operation the slew-rate should be higher than 0.5 V/µs. The input polarity is also software selectable. Refer Figure 3.4 for a simplified circuit schematic. Diodes are provided for overvoltage protection.

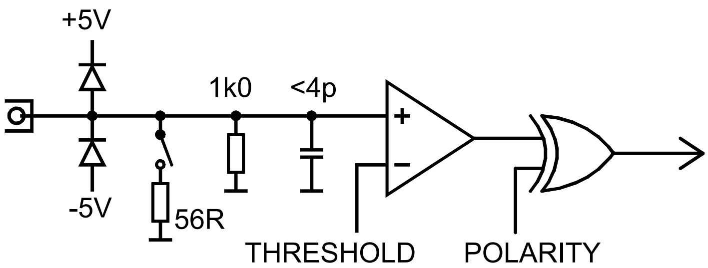


Figure 3.4: Discriminator Inputs simplified Schematic

WARNING

Take care not to exceed the maximum input values as described in the technical data (ref. chapter 7.2.1).

The discriminator signals, as detected by the input circuitry, may be monitored online with an oscilloscope on the SYNC outputs. Thus, optimization of the threshold voltages was never as easy. It is recommended to use the Fast NIM SYNC output for this purpose due to it's higher bandwidth.

3.2.4. TRIGGER (START) / SAMPLE Input

This input is used as either

- TRIGGER (START) of a MCS sweep
- SAMPLE command for the internal ADC

The TRIGGER (START) / SAMPLE input is a LEMO type connector on the mounting bracket. It accepts signals in the range of ±5 V and provides a bandwidth of 100 MHz. The input impedance is a selectable 1 kΩ or 50 Ω. The threshold voltage is adjustable in the range of ±2.5 V. For clean switching and oscillation free operation the slew-rate should be higher than 0.5 V/µs. The input polarity is also software selectable. Refer Figure 3.4 for a simplified circuit schematic. Diodes are provided for overvoltage protection.

WARNING

Take care not to exceed the maximum input values as described in the technical data (ref. chapter 7.2.1).

The discriminator signals, as detected by the input circuitry, may be monitored online with an oscilloscope on the SYNC outputs. Thus, optimization of the threshold voltages was never as easy. It is recommended to use the Fast NIM SYNC output for this purpose due to it's higher bandwidth.

3.2.5. COUNT (STOP) 1 & 2 Inputs

These inputs are for MCS counting.

The COUNT (STOP) 1 & 2 inputs are LEMO type connectors on the mounting bracket. They accept signals in the range of ±5 V and provide a bandwidth of typically well over 200 MHz. The input impedance is individually selectable between 1 kΩ and 50 Ω. The threshold voltage is adjustable in the range of ±2.5 V. For clean switching and oscillation free operation the slew-rate should be higher than 0.5 V/µs. The input polarity is also software selectable. Refer Figure 3.5 for a simplified circuit schematic. Diodes are provided for overvoltage protection.

> **WARNING**
>
> Take care not to exceed the maximum input values as described in the technical data (ref. chapter 7.2.1).

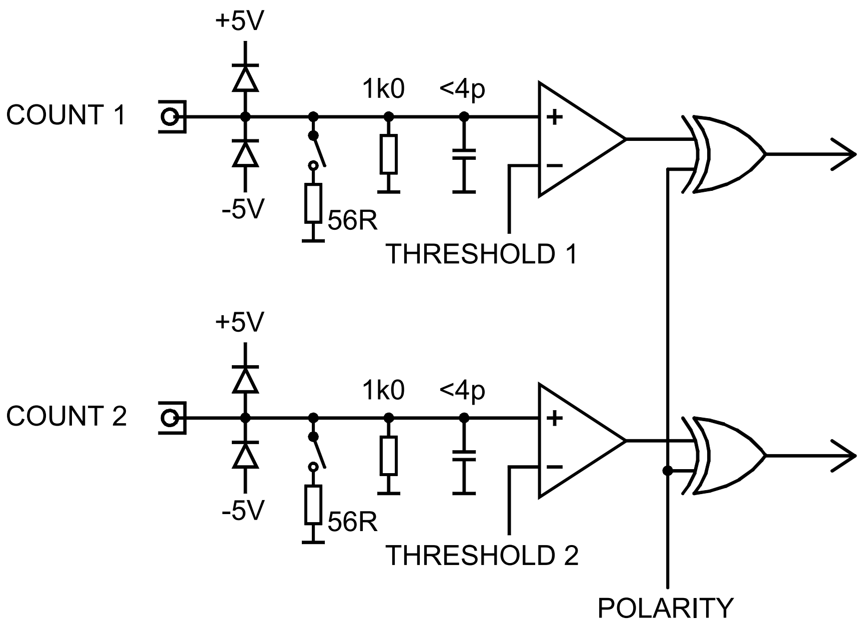


Figure 3.5: Count 1 & 2 Inputs simplified Schematic

> **WARNING**
>
> The COUNT inputs are ultra high speed, high sensitivity inputs and thus, susceptible to oscillation. Take care to apply low impedance (≤ 50 Ω) source signals and well shielded, 50 Ω cables.

The discriminator signals, as detected by the input circuitry, may be monitored online with an oscilloscope on the SYNC outputs. Thus, optimization of the threshold voltages was never as easy. It is recommended to use the Fast NIM SYNC output for this purpose due to it's higher bandwidth.

3.2.6. SYNC / Monitor Outputs

The SYNC outputs provide a large variety of output signals for a lot of synchronizing, triggering, monitoring or whatsoever application. The selectable output signals are:

- CLK: Internal 40 MHz clock
- GATE/ADV/ABORT: Discriminated input signal
- SAMPLE/TRG: Discriminated input signal
- COUNT (STOP) 1: Discriminated input signal
- COUNT (STOP) 2: Discriminated input signal
- LIVE_INT: Internal ADC livetime signal
- LIVE_EXT: External ADC port livetime signal
- 1 kHz: Internal 1 kHz signal
- SCA: Single Channel Analyzer output
- ON: indicates a running sweep in MCS mode when logic '1'
- EoBin: End-of-Time bin signal
- BIN_DIV: Divided End-of-Time bin signal used to increment the x-ramp DAC
- STEP_CNT: Signals when number of set steps of the x-ramp DAC is reached
- GO: 'GO'-line
- RAMP_DOWN: signals falling portion of a triangle x-ramp
- '1': logic '1'

All these signals can be inverted and they all may be output on the Fast-NIM SYNC 1 output on the mounting bracket and on the TTL SYNC 2 output on the FEATURE connector (ref. Figure 3.9) as well.

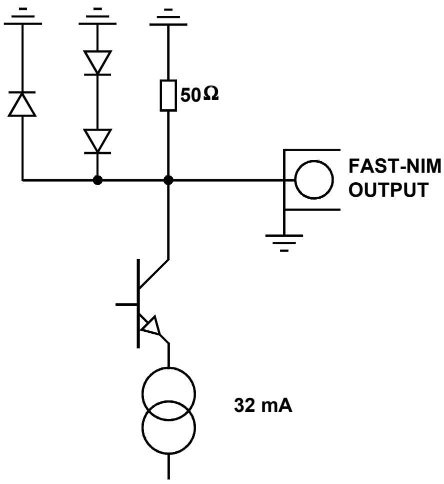


Figure 3.6: Fast-NIM SYNC 1 output schematic

The Fast-NIM SYNC output supplies standard Fast-NIM (0…-0.8 V / 16 mA) signals into a 50 Ω load. The output impedance also is 50 Ω. For Fast-NIM signals a logical 'TRUE' corresponds to a low voltage (-0.8 V), e.g. while a sweep is running 'ON' will result in –0.8 V (= 'TRUE') output.

3.2.7. 'GO'-Line

The system-wide open-drain 'GO' line enables any connected device to start and stop all participating measurement equipment simultaneously. This allows for easy synchronization of electronic devices previously often not possible.

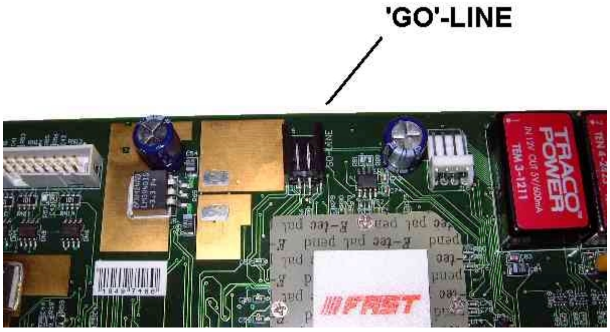


Figure 3.7: 'GO'-line connector

The 'GO' line is a system-wide open-drain wired-AND signal that can start and stop a measurement. This line is also available on the Multi I/O port connector (ref Figure 3.9). The 'GO'-line may be enabled, disabled, set and reset by the software.

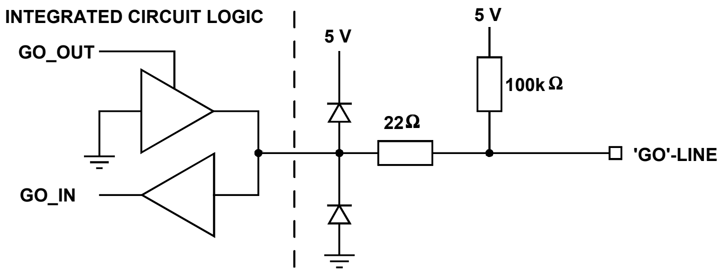


Figure 3.8: 'GO'-line logic circuit schematic

When watching of the 'GO'-line is enabled a low voltage will halt the measurement. When output to the 'GO'-line is enabled starting a measurement will release (high impedance output) the 'GO'-line whereas a halt of the measurement will pull down the 'GO'-line to a low state. Since it is an open drain output wired AND connection with other devices is possible.

3.2.8. FEATURE (Multi) I/O Connector

A very versatile 8 bit digital I/O port is implemented on the female 15 pin high density D-SUB connector fixed on the mounting bracket. Since the corresponding resistors are socket mounted (ref. Figure 3.10) they can be easily user configured in a most flexible way.

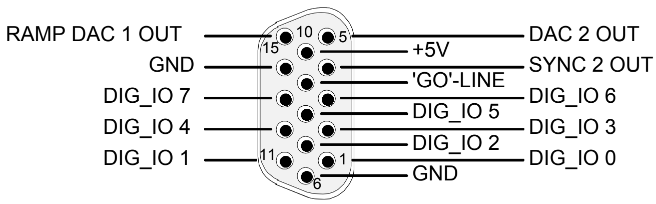


Figure 3.9: FEATURE (multi) I/O connector pinning

This I/O port is fully software controllable and each single (1-bit) port is individually configurable. It might be used for external alert signals, sample changer control, status inputs / outputs etc.

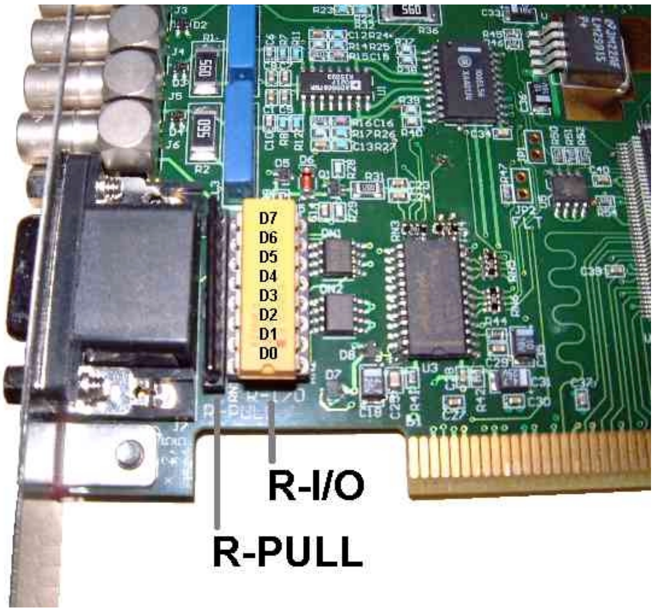


Figure 3.10: Digital I/O port configuration resistors

As can be seen from Figure 3.11 each individual bit of the digital I/O port might be configured as input only (tri-stated output), pull-up (open drain) or driver output (small R-I/O) with readback capability. Wired-OR / AND connections are also feasible.

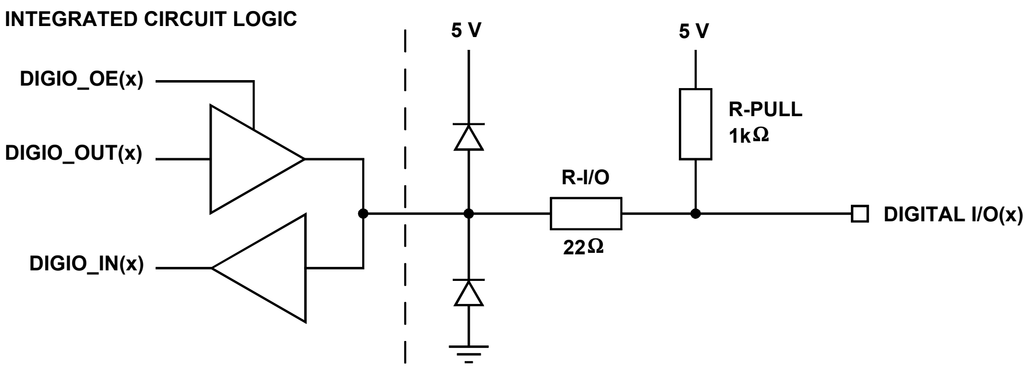


Figure 3.11: FEATURE (multi) I/O port schematic

3.2.9. Analog DAC Outputs

Two 12 bit DACs provide analog output voltages.

DAC 1 has an output voltage range of 0 … +10 V with a resolution of 2.44 mV. On power-up or reset the output is reset to 0 V. Beside a fixed output voltage it can also be configured in MCS mode as x-ramp output with a triangle or sawtooth waveform.

The x-ramp can be defined by a start value, the stepping time in multiples of the dwell time, the step size and the number of steps until restart (sawtooth) or change of slope (triangle).

The time for each slope is defined as:

STEPS x STEPPING TIME = "Step Bin Width" x "Number of Steps" x "Dwell Time"

The height of each slope beginning at the Start Value is:

STEPS x STEP SIZE = ("Step Height" x "Number of Steps" - "Start Value") x 2.44mV

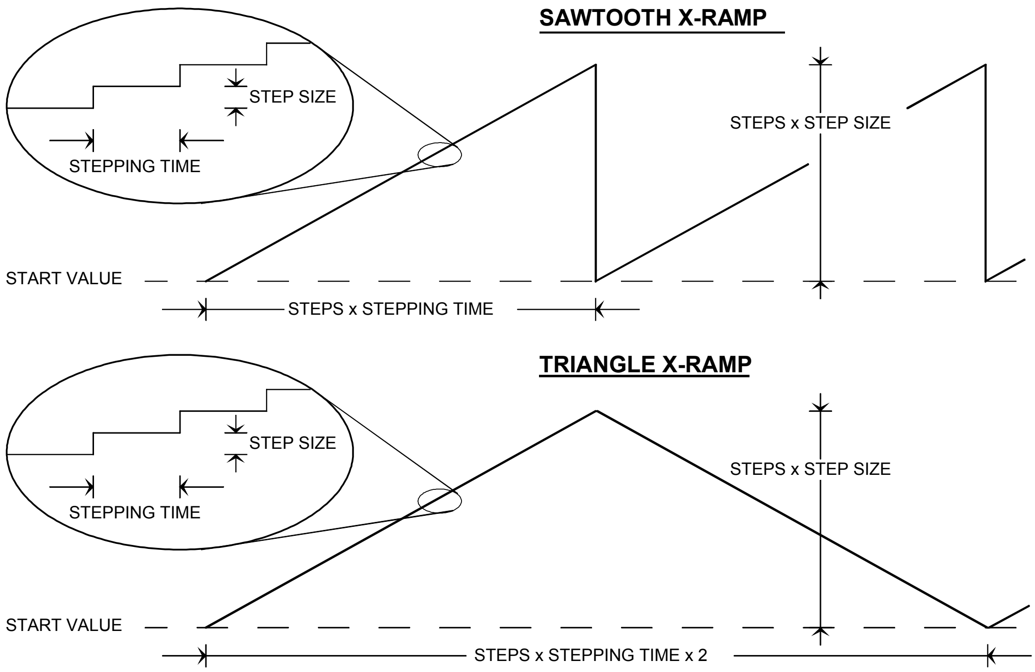


Figure 3.12: X-Ramp Details

A good idea is to stick to the following rules for not more than one ramp cycle per sweep:

For a sawtooth x-ramp:

- START VALUE [LSBs] + STEPS x STEP SIZE [LSBs] ≤ 4095
- STEPPING TIME [Time Bins] x STEPS ≤ Sweep Range [Channels]

For a triangle x-ramp:

- START VALUE [LSBs] + STEPS x STEP SIZE [LSBs] ≤ 4095
- STEPPING TIME [Time Bins] x STEPS ≤ Sweep Range [Channels]

Where the LSB is 2.44 mV, STEPPING TIME is in multiples of Time Bin which is the actual dwell time and Channels is the number of channels in the spectrum . Thus, you will get just a single ramp per sweep.

DAC 2 has an output voltage range of ±10V with a resolution of 4.88 mV. On power-up or reset the output is reset to 0 V.

3.3. Multichannel Analyzer Section

The MCA supports either the external ADC port with upto 16 bit or the internal ADC.

3.3.1. Live / Real Timer

Two 32 bit timers are implemented featuring live and real time measurement with a resolution of 1 ms.

Each of the timers is presettable in the range of 2 ms to more than 1193 hours.

3.3.2. Listmode

The onboard memory can also be configured for listmode data acquisition thus working like a large FIFO memory.

3.3.3. Spectra offset

A digital offset might be applied to the spectra shifting its actual location in the memory. Thus, multiple spectra might be sequentially accumulated in the onboard memory. The offset applied by software selection can be in steps of 64.

3.3.4. Event Tagging / Spectrum Multiscaling

The 8 digital I/O ports can be used for spectrum tagging as well. Each bit is individually selectable for tagging. The enabled TAG bits are used as the most significant memory address bits thus, shifting the acquired spectrum corresponding to the actual TAG input.

Digital I/O bit 7 is used as address bit 18 (MSB) while Digital I/O bit 0 is used as address bit 11.

When all 8 TAG bits are used the remaining memory range is 512k / 256 = 2048, i.e. 2k.

In MCA mode the digital I/O port (TAG input in this case) is sampled when the DEADTIME signal becomes active (edge sensitive).

3.4. Internal ADC / Single Channel Analyzer

3.4.1. General

The internal ADC is an ultra fast pulse height analyzing analog-to-digital converter (PHA ADC) with a typical conversion time of 500 ns. The peak detection circuitry is optimized for Gaussian shaped input signals. It is also possible to use the ADC in a sampling mode. This ADC is ideally suited for high counting rate applications.

Two discriminator levels provide for Single Channel Analyzer (SCA) functionality. The built-in SCA can be output on any SYNC output. This can be used in such applications like Moessbauer Spectroscopy. The GATE input can be used to gate out specific events.

An additional analog RC low-pass filter (f_g=1.8MHz) can be enabled to reduce high frequency noise superimposed on the input.

3.4.2. Pulse Height Analysis

In PHA mode each input pulse with its peak value falling into the window between the lower and upper level discriminators is converted. When enabled the GATE input must be valid (polarity is software selectable) at the peak (ref. chapter 7.1.3 for setup and hold requirements).

The threshold voltage (THD) defines the upper limit of the noise floor to prevent conversion triggering by noise.

The simplified operational flow is to detect a signal is coming by an upward crossing of the threshold level (i.e. input is above noise). This arms the peak detector. Then, at the signal's peak you check if the pulse falls between LLD and ULD. If yes, the peak voltage level is converted. This way you can discard pulses that fall outside an interesting e.g. energy window.

Pile up rejection is provided by allowing a new conversion only after the signal has fallen back below the threshold level.

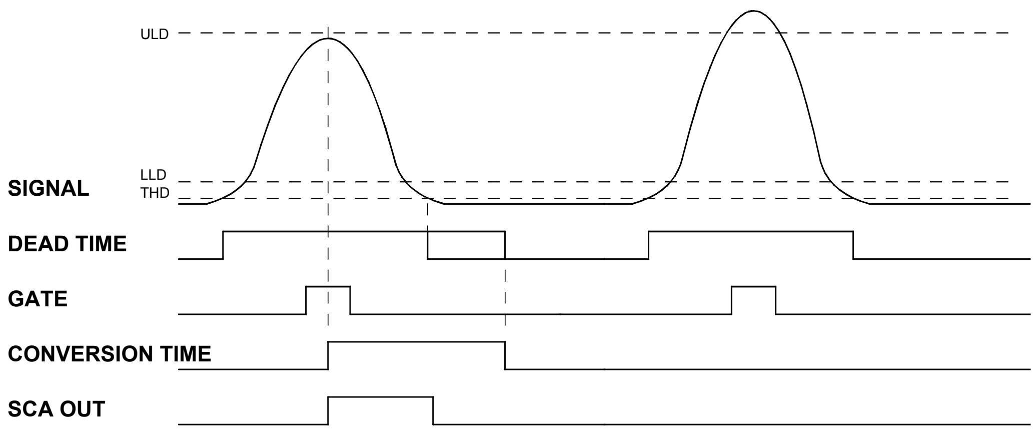


Figure 3.13: Internal ADC PHA Principle

3.4.3. Sampling Voltage Analysis

In SVA mode the input voltage is converted on the selected edge of the SAMPLE input signal. Different from PHA mode the LLD / ULD levels are ignored and the SCA does not work in SVA mode. An enabled GATE input must be valid at the sampling signal edge.

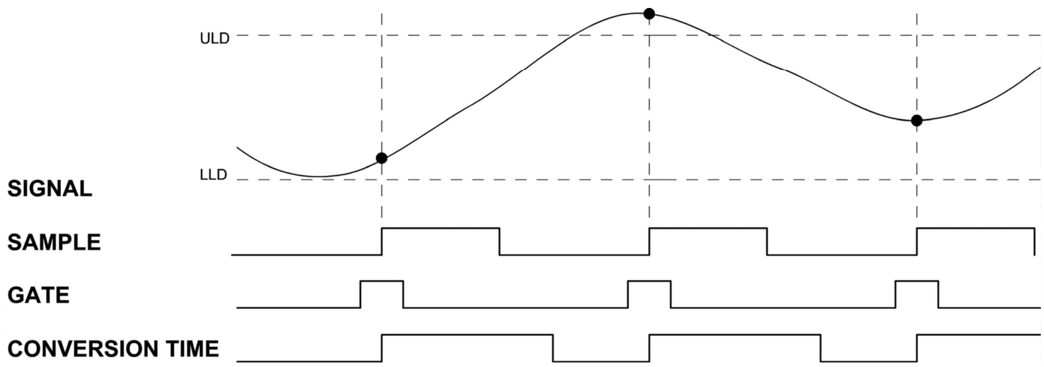


Figure 3.14: Internal ADC SVA Principle

3.4.4. Single Channel Analyzer

The Single Channel Analyzer function filters a signal input by amplitude in the MCS mode of operation using the SCA Threshold, Lower Level (LLD) and Upper Level (ULD) Discriminator settings. An event count is then accepted only if it is above the Threshold and LLD and below the ULD. This function enables the selection of only certain amplitude signals by the SCA for further processing by the MCS function.

To activate the internal SCA (Single Channel Analyzer) function, go to Options...Range/Presets. Check the “Use SCA” checkbox. See Figure 4.4 “MCS Settings” on page 4-4 .

Note: When the SCA function is used, the input signal should be connected to the Analog (ADC) Input normally used with the MCA (Pulse Height Analysis) mode. The filtered output of the SCA function is then internally connected to the COUNT 1 input of the MCS function. This feature is available only at the model MCA-3FADC, as it is the only model in the MCA-3 series having both an analog input to the internal ADC and MCS functionality.

If the GATE function checkbox is enabled, then a suitable GATE function signal must be connected via the GATE/ADV/ABORT input connector.

3.5. Multiscaler Section

3.5.1. Introduction

In MCS mode the MCA-3 counts STOP events relative to the TRIGGER (START) signal. The time resolution is equal to the selected dwell time. The total measurement time is the spectrum length in time bins multiplied by the dwell time. The maximum count rate is typically well over 200 MHz.

The number of time bins (= spectrum length) may be selected in the range of 64 to 512k (256k in dual input mode) in steps of 1. This fine granularity allows to optimize the sweep length for a specific experiment.

When a sweep (scan) is finished – i.e. the last time bin has elapsed – the MCA-3 is instantly ready for the next TRIGGER. Thus, a maximum sweep repetition rate with no end-of-sweep deadtime is guaranteed.

3.5.2. Dwell Timer

The internal dwell timer provides automatic time bin advance in the range of min. 100 ns (200 ns in dual input mode) upto over 107 s in steps of 25 ns.

External channel advance is supported via the corresponding ±5 V discriminator input. Please note that the external signal is internally synchronized to the 40 MHz clock. An integrated circuitry asures that the minimum cycle time is 150 ns guaranteeing no triggering on fast ringing or successive spikes. Please make sure not to under-run the minimum bin width requirement of 200 ns in dual input mode.

For test and experiment setup purposes a manual or software channel advance is provided.

3.5.3. Sweep Counter

In MCS mode one of the internal 32 bit counters acts as sweep counter. It is incremented on each valid TRIGGER event.

Sweep presets may be selected in the range of 1 to 2^{32}.

3.5.4. Abort a Sweep

In some cases it might be desirable to abort a running sweep. The ABORT input signal aborts the actual sweep at the end of the actual time bin. Thus, the minimum ABORT pulse width must be equal or greater than the dwell time.

3.5.5. Spectra offset

As in MCA mode a digital offset might be applied to the spectra shifting its actual location in the memory. Thus, multiple spectra might be sequentially accumulated in the onboard memory. The offset applied by software selection can be in steps of 64.

3.5.6. Spectrum Tagging

As in PHA mode the 8 digital I/O ports can be used for spectrum tagging. Each bit is individually selectable for tagging. The enabled TAG bits are used as the most significant memory address bits thus, shifting the acquired spectrum corresponding to the actual TAG input.

Tag bits can be used to record a 1-8 bit external digital input and “append” or “tag” it to the time information recorded by the multiscaler, providing the tag bits are present at the tag inputs at the appropriate time. A typical application, for example is to identify a specific detector from several (using a digital code fed to the tag input) that generated the time count. Normally there are a total of 19 bits of digital data available (corresponding to the 512 K time bin memory on the board), which enables a maximum of 524,288 time channels (time bins) to be used in 1 input mode, and 262,144 in each input in 2-input mode.

Digital I/O bit 7 is used as address bit 18 (MSB) while Digital I/O bit 0 is used as address bit 11. In 1 input mode, a maximum of 8 tag bits are available, and when using all 8 tag bits 2048 time bins are then available (corresponding to 19-8 = 11 bits). Conversely, for example, for a sweep length of 4096 channels (4K, 12 bits), a maximum of 7 tag bits can be used.

In 2-input mode, a maximum of 7 tag bits are available (the 8th bit is used to identify input 1 or input 2). When 7 bits are used in 2 input mode, the maximum time sweep length is limited to 2048 channels (corresponding to 19-1-7 = 11 bits) for each input.

In MCS mode the digital I/O port (TAG input in this case) is sampled when the TRIGGER signal becomes active. Thus, whole sweeps are tagged.

3.5.7. MCS Listmode

Using the listmode data acquisition in MCS mode allows for sheer endless sweeps, i.e. to extend the total sweep length beyond the onboard memory limit of 512k time bins.

The minimum dwell time applicable depends on the maximum data transfer rate into the PC and thus, largely depends on the computer used.

4. Windows Server Program

The window of the MCA-3 server program is shown here. It enables the full control of the MCA-3 card to perform measurements and save data. This program has no own spectra display, but it provides - via a DLL („dynamic link library“) - access to all functions, parameters and data. The server can be completely controlled from the MCDWIN software that provides all necessary graphic displays.

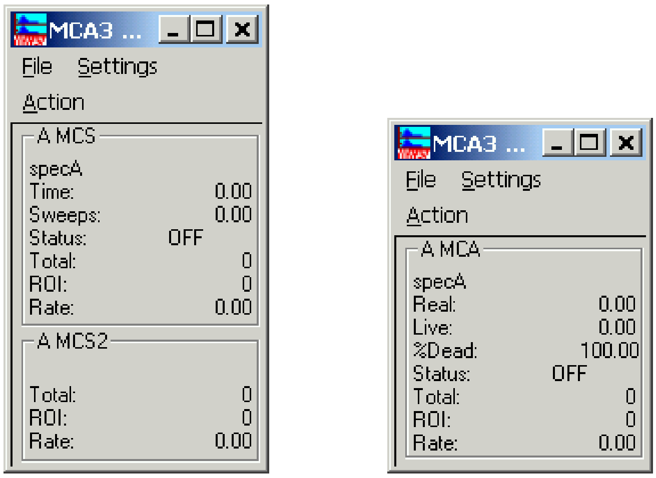


Figure 4.1: MCA-3 Server Window in dual MCS mode (left) or MCA mode (right)

4.1. Server functions

To start the software, just double click a shortcut icon linking to the server program. The server program performs some tests and then starts MCDWIN and gets iconized. Usually you will control everything from MCDWIN, but it is possible to work with the server alone and independently from MCDWIN.

4.1.1. Initialisation files

At program start the configuration files MCA3.INI and MCA3A.CFG are loaded. Up to 4 MCA-3 boards can be used. Specify the number of boards in the MCA3.INI file with a line devices=n. You can also specify more than one if you have only one physical module. The software runs then for the not physical boards in demo mode and it is then possible to load spectra for comparing them in MCDWIN.

```
;Set devices to number of MCA-3 modules plugged in (max. 4)
devices=1
;
; Board type: 22 MCA-3,
;             24 MCA-3FADC,
;             26 P7882,
;             33 MCA-3B,
;             34 MCA-3A,
; (Bit0=/SCA, Bit1=/int.ADC, Bit2=/ext.ADC, Bit3=/listmode, Bit4=/MCSmode)
type=20,20,20,20
; Slider range for internal ADC
adcslider=1024,1024,1024,1024
; External ADC Port Control Signal Polarities (hex):
; 1) pol0:  [DENB;ENC;DACC;DEAD;DRDY]
;    DRDY    :   0 = active high, 1 = active low
;    DEAD    :   0 = active high, 1 = active low
;    DACC    :   0 = active high, 1 = active low
;    ENC     :   0 = active high, 1 = active low
;    DENB    :   0 = active low, 1 = active high
; The first parameter belongs to the first devcice,
; for more devices add parameters seperated with comma
; Silena ADC's:
;pol0=13,13,13,13
; FAST ADC's:
pol0=5,5,5,5
;
;The MCA-3 memory is read out every updaterate msec, if possible.
updaterate=500
```

Figure 4.2: MCA3.INI File

In the MCA3.INI file the hardware configuration and some settings are defined, that can be changed only by editing this file using a text editor like Notepad. The board type must be specified. If it is different from the type found, the software shows the type found, terminates and opens the notepad editor with the MCA3.INI loaded. Change then the line type= , enter the correct type number(s) and save the file. Very important if using the external ADC port is the correct setting of the **ADC handshake polarities**. If you have for example Silena or Laben ADC's, remove the semicolon at the first position of the line after "; Silena ADC's:" and insert a semicolon in the line after "; FAST ADC's:"

The file MCA3A.CFG (MCA3B.CFG... for more modules) contains the default settings. It is not necessary to edit this file, it is saved automatically. Instead of this .CFG file any other setup file can be used if its name without the appendix ‘A.CFG’ is used as command line parameter (e.g. MCA3 *TEST* to load TESTA.CFG).

4.1.2. Action menu

The server program normally is shown as an icon in the taskbar. After clicking the icon it is opened to show the status window. Using the „**Start**“ menu item from the action menu a measurement can be started. In the status window every second the acquired events, the counting rate and the time are shown. Clicking the „**Halt**“ menu item the measurement is stopped and via „**Continue**“ proceeded.

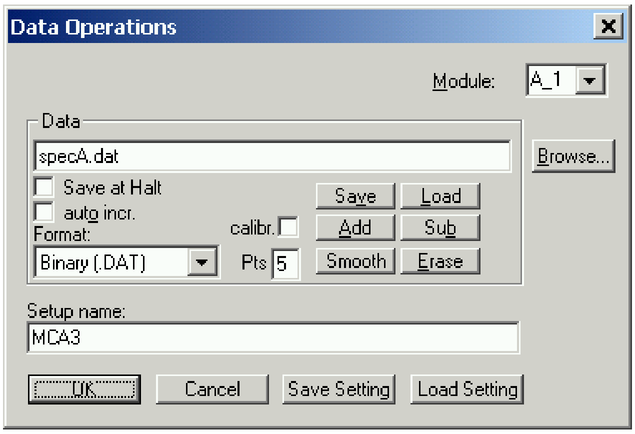


Figure 4.3: Data Operations dialog

4.1.3. File menu

The Data... item in the File menu opens the Data Operations dialog box. Mark the checkbox „**Save at Halt**" to write a spectrum- and a configuration file at the stop of a measurement. The filename can be entered. If the checkbox „**auto incr.**" is marked, a 3-digit number is appended to the filename that is automatically incremented when starting a new acquisition. The format of the data file can be ASCII or binary (extension .ASC or .DAT). Click on „**Save**" to write a data- and configuration file of the actual data with the specified name. By pressing „**Load**" previously stored data can be loaded or a **control file** (extension .CTL) executed. With „**Add**" or „**Sub**" a spectrum saved on disk can be added to or subtracted from the present data. Eventually resulting negative counts will be truncated to zero. Check the checkbox „**calib.**" to enforce using a calibration and shift the data to be added according to the calibration. The „**Smooth**" button performs a n-point smoothing of the spectrum data. The number of points to average can be set with the „**Pts**" edit field between 2 and 21. „**Erase**" clears the spectrum.

The menu item File – Replay... opens the Replay dialog.

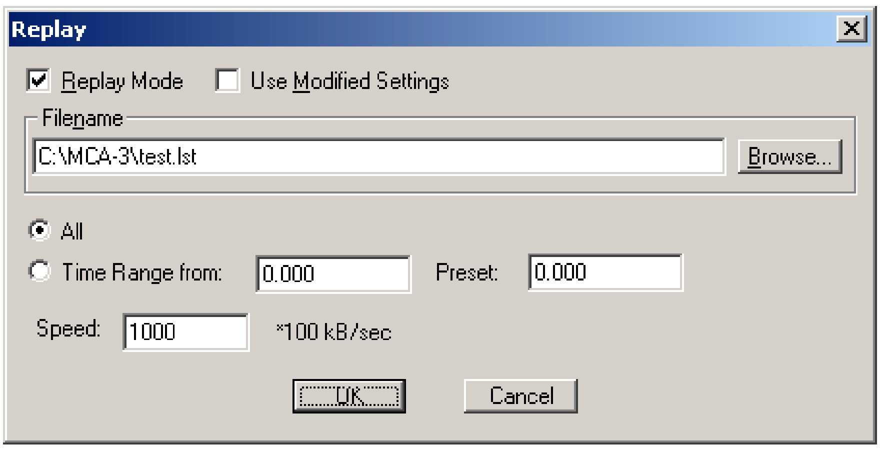


Figure 4.4: Replay Settings dialog

Enable **Replay Mode** using the checkbox and specify a **Filename** of a list file (extension .LST) or search one by pressing **Browse...** In PHA mode it is possible either to choose the complete listfile by selecting **ALL** or a selected **Time Range**, if the listfile contains time stamps. The Replay **Speed** can be specified in units of 100 kB per sec. To **Use Modified Settings** enable the corresponding checkbox; otherwise the original settings are used. To start Replay press then Start in the Action menu or the corresponding MCDWIN toolbar icon.

The MCDWIN menu item in the file menu starts the MCDWIN program if it is not running.

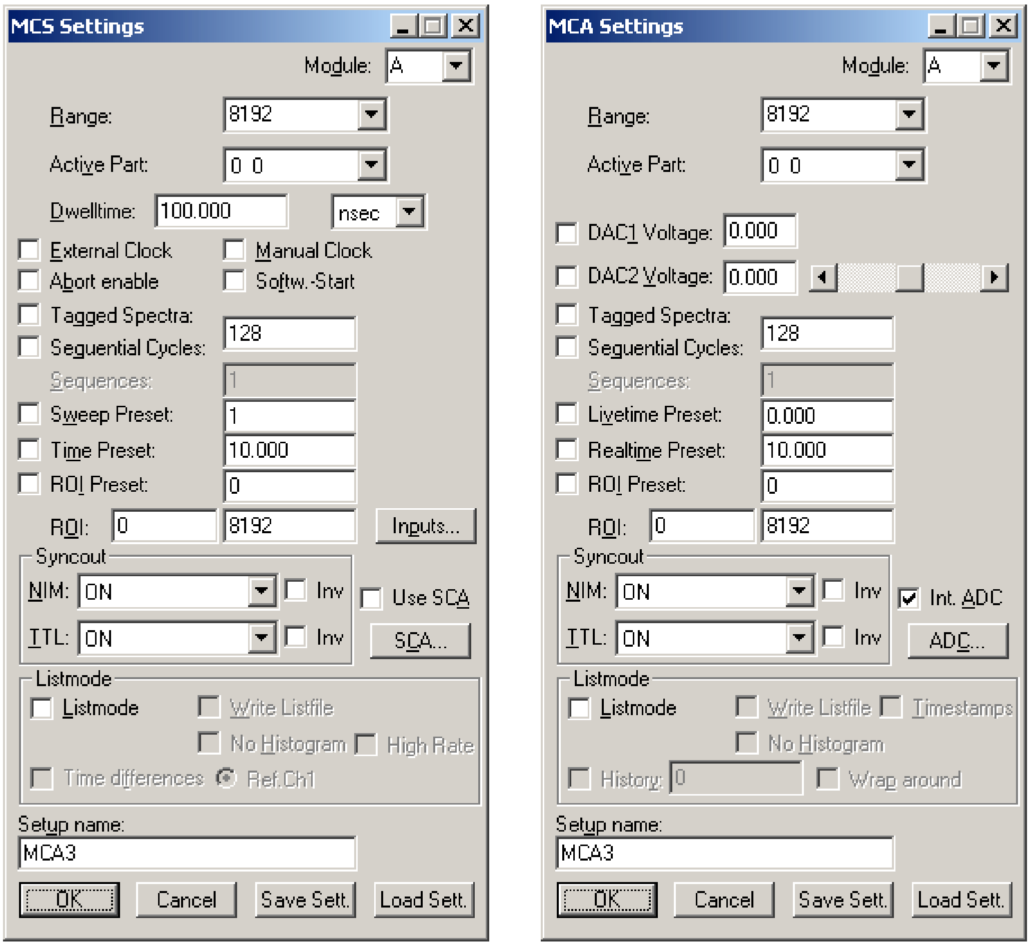


Figure 4.5: MCS and MCA Settings dialog

4.1.4. Settings dialog

The Hardware... item in the Settings menu opens the MCA-3 Settings dialog box. Here depending on the mode of operation (MCS or MCA/PHA) parameters like **Presets**, **Range** parameters, **Dwell time**, etc can be set. In the edit field **Range** the length of the spectrum can be chosen between 64 and 524288 (in 2MCS mode up to 262144). If the checkbox **ROIpreset** is marked, the measurement will be stopped after acquiring more events than specified in the corresponding edit field. The events are counted only if they are within the **ROI** limits, i.e. >= the lower limit and < the upper limit. Another possibility is to acquire data for a given real time via the **Realtime Preset** or a given live time via the **Livetime Preset**. A measurement will be stopped if the corresponding checkbox is marked. Note that the start of the acquired spectrum in the MCA-3 memory can be defined in the **Active Part** drop down listbox. Also the ERASE, LOAD and SAVEDAT commands make an effect only on this part of the MCA-3 memory. The range parameter defines the length of the spectrum starting from the active part. The actual displayed memorysize in MCDWIN may be greater if a nonzero Active Part is specified. **Dwelltime** is the width of a timebin in MCS mode, it can be entered in units of 25 nsec. The minimum dwelltime is 100 ns in MCS mode or 200 ns in 2MCS mode. The option **External clock** respective **Manual clock** is only in MCS mode available. If **Softw. Start** is marked, no start signal is necessary. The time-counter is masked corresponding to the chosen range and the higher bits are not evaluated. The signal for the synchronisation of the experiment can be obtained from one of the two Sync Out outputs.

Via the **Sync out** - combo boxes the Synchronisation / Monitor signals specified in chapter 3.2.6 can be selected: **Gate/Abort/Adv. TRG/SAMPLE**, **COUNT1**, **COUNT2**, **LIVE_INT**, **LIVE_EXT**, **1kHz**, **SCA**, **ON**, **EoBin**, **BinDiv**, **STEP_CNT**, **GO**, **RAMP_DOWN**, **1**, and **40 MHz** as a synchronized time signal in MCS mode.

(Dec-23-2004): In **MCA listmode** it is now possible to record and display a **History** of the ADC data, like a transient recorder (see Fig. 4.5). The **Length** of the history can be defined arbitrarily.

A **Wrap around mode** allows to get a repeated view by overwriting old data. In MCA listmode now optional timestamps can be written into the listfile. (Mar-21-2005): a new checkbox "**Timestamps**" in the settings dialog in the listmode box allows to dump the realtime and livetime counters into the listfile every time when ADC event data are transfered, see chapter 4.1.6.

In **MCS List mode** an acquisition mode "**Time differences**" is implemented for analyzing pulse trails. In this mode the first stop event is used as a reference point and for following stop events the time difference to the reference is calculated. The displayed spectra is then a relative time distribution of stop events related to the reference point. The first stop event that falls out of the chosen time range after a reference event is taken as a new reference point. Note that the MCS List mode is a wrap-around mode intended for a very long single sweep. The sweep can be much longer than the card RAM size. In this mode the maximum number of counts per time bin is usually 255. This limit can be overcome if the "**High Rate**" box is checked, but in this case the minimum dwelltime is about 2 mikrosec for proper operation, a warning message will be shown when trying wrong settings (Aug-25-2009).

The measurement can be stopped automatically after a specified number of sweeps by checking **Sweep preset**. A List file can be written by checking the corresponding checkbox **Listmode** and **Write List file**. If **No Histogram** is checked, no histogramming is made. A series of measurements can be acquired into separate memory parts by checking **Sequential cycles** and specifying the number of cycles. Each single measurement should be terminated by any of the preset conditions, the complete run stops after performing the specified number of cycles or is repeated accordingly if the specified number of **Sequences** is greater than 1. Check **Tagged spectra** if you want to acquire up to 256 seperated spectra marked by tag bits as mentioned in chapter 3.3.4. (MCDWIN will show the spectra in a 2 dimensional view).

If the checkbox **ROI Preset** is marked, the measurement will be stopped after acquiring more events than specified in the corresponding edit field. The events are counted only if they are within the **ROI** limits, i.e. >= the lower limit and < the upper limit. It is not necessary that this ROI is within the spectra range. Another possibility is to acquire data for a given time via the **Time Preset**.

The **Inputs...** button opens the Input Thresholds and DAC's dialog box. Here you can specify the threshold level at the falling edge of the input signal. The combo box provides a choice between standard Fast NIM (-0.4 V) and customized, i.e. Voltage level set by hand between -2.5 .. +2.5 V (scroll bar or edit field). Also the voltages for the free usable DAC outputs DAC1 and DAC2 can be set in this dialog.

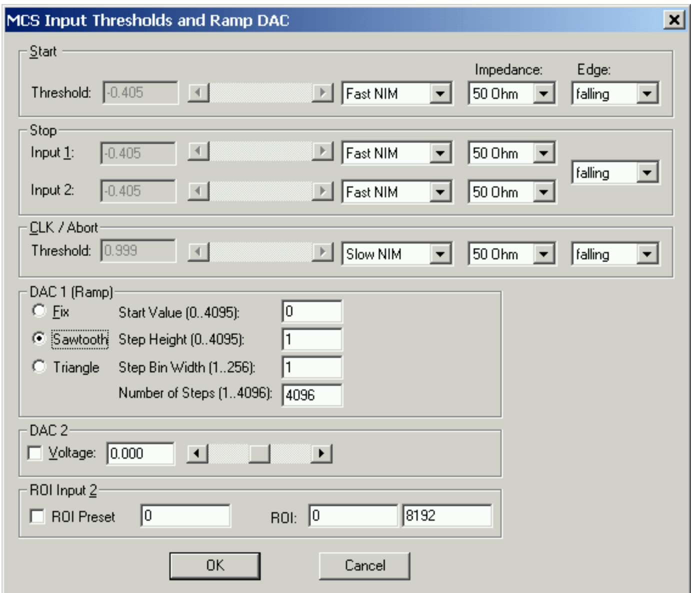


Figure 4.6: Input Thresholds and DAC's dialog

All parameters of the Ramp DAC as described in chapter 3.2.9 and the second MCS input ROI can be set here too.

The **ADC...** button is available in MCA mode if the internal ADC is available and opens the ADC Settings dialog box. Here you can specify the Lower Level and the Upper Level of the SCA, i.e.

the Voltage level between 0 .. +10 V (scroll bar or edit field) and the Threshold between 0 and 2.5 V. Also the Gate and SVA (Sampled Voltage Analysis) can be enabled and the threshold voltages for the Gate and SVA inputs can be set in this dialog, see chapter 3.4.

Checking the Filter checkbox activates a 1.8 MHz cutoff frequency analog input filter on the Analog Input connector. This is useful in some cases to filter high-frequency noise that may be present on the input signal.

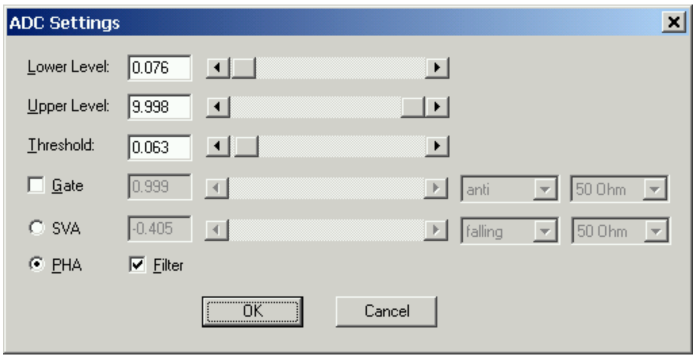


Figure 4.7: ADC Settings dialog

In MCS mode the **SCA...** button opens the SCA Settings dialog that allows to specify the same SCA settings when using the analog input together with the SCA for multiscaling.

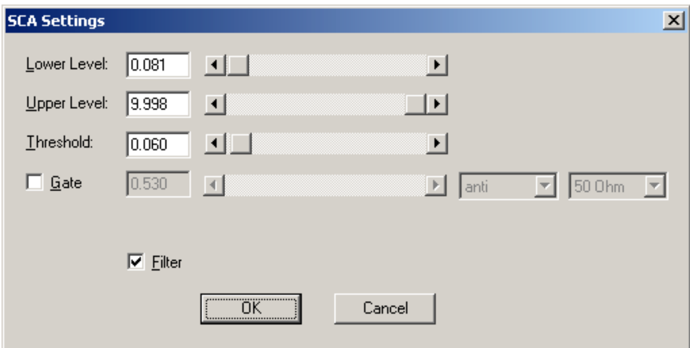


Figure 4.8: SCA Settings dialog

4.1.5. System definition dialog

The „System..." item in the settings menu opens the System Definition dialog box. If more than one MCA-3 modules are used, several cards can be combined to form one or up to 4 seperate systems that can be started, stopped and erased by one command. In addition the use of the Digital Input / Output and the GO-Line can defined: It can be used either to show the status of the MCA if the checkbox **Status Dig 0** (0..3 for more modules) is marked. At the respective pins +5 Volt are output if an acquisition is running and 0 V if not. The polarity can be inverted by checking **Invert**. Alternatively, it can be used for example with a sample changer by checking **"Value inc. at Stop"**. Here, the 8 bit value entered in the edit field (a number between 0 and 255) is output at the Dig I/O port. This value will always be incremented by 1 if the P7887 is stopped. The **Invert** checkbox allows to invert the logical level. See also the commands **pulse** and **waitpin** how to handshake a sample changer. The Radio buttons **Push-Pull** and **Open Drain** describe the output mode of the Dig I/O ports.

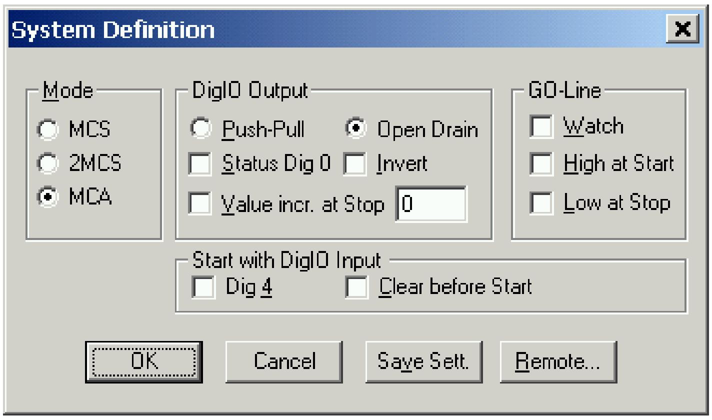


Figure 4.9: System Definition dialog box for a single MCA-3 card

It is also possible to use the digital input 4 as an external trigger for starting the system (more modules: Dig inputs 4..7 start systems 1..4) (**DESY control line**). If the corresponding checkbox is marked, a start command for the respective system will not immediately start the system. After the start command, the digital input will be permanently checked for its logical level. If the level changes from high to low, the data for the system is cleared and it will then be started. It will stop if the level returns to high (or vice versa if **Invert** is marked) and can again be restarted with the next level change. A stop command for the system will finish the digital input checking. By checking **Clear before Start** the spectra is cleared before the start. A stop command for the system will finish the digital input checking.

The Use of the GO-Line is controled via the 3 checkboxes **Watch**, **Release at Start**, and **Low at Sweep Preset reached**. The GO line gates directly the hardware. "Low at Sweep preset reached" means that the GO line is immediately pulled down when a sweep preset is reached.

If more than one MCA-3 card is used, the system definition dialog box comes up as shown in Figure 4.10. Here the several units can be combined to form up to 4 separate systems that can be started, stopped and erased by one command.

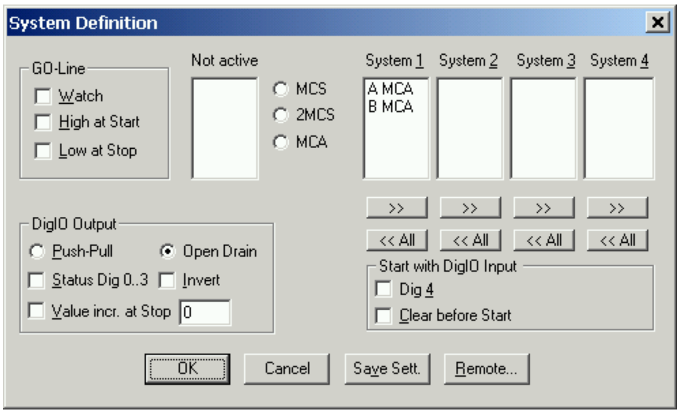


Figure 4.10: System Definition dialog box, two MCA-3 cards

In the shown setting a single system is formed. The two modules MC_A and MC_B are combined. System 1 can be started, stopped, erased, and continued with the respective commands in the Action 1 menu. It is also possible for example to form two independent systems 1 and 2: Click on the button labeled **<<All** below the list box „System1" to remove all units from system 1. They are then shown in the „Not active" list box. Then select unit A and click on the button labeled **>>** below the „System 1" list box to include it into system 1 and perform the respective action for unit B and System 2.

OK accepts all settings. **Cancel** rejects all changes. Pressing „**Save Settings**" stores all settings in the file **MCA3A.CFG** using the control language (see the following section)

This file is loaded at program start automatically and the parameters set. Together with each data file a header file with extension .mcd is saved. This header also contains all settings and in addition some information like the date and time of the measurement and comments entered in the MCDWIN program.

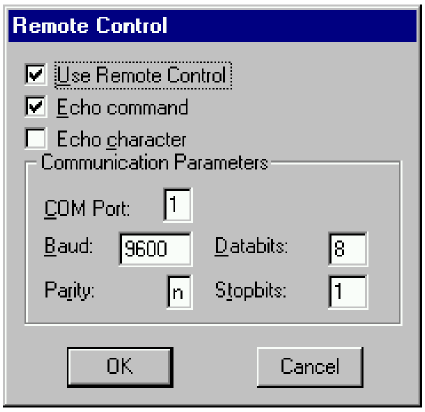


Figure 4.11: Remote control dialog

The **Remote...** button opens the Remote control dialog box. Here all settings can be made for the control of the MCA-3 server program via a serial port. If the Checkbox **Use Remote Control** is marked and the COMCTL.DLL is available (i.e. you have the **optional MCDLAN software**), the specified COM port will be used for accepting commands (see Control language). If **Echo command** is marked, the input line will be echoed after the newline character was sent. **Echo character**, on the other hand, immediately echoes each character.

4.1.6. File formats

Spectra data is written into two seperate files, one with extension .mcd containing configuration data and one containing pure spectra data with an extension indicating the chosen format. The **.mcd** file contains the settings in ASCII format using the control language described in section 4.2.

Spectra data files with extension **.asc** contain in each line one decimal number in ASCII containing the corresponding count value in the histogram.

Binary data files with extension **.dat** are written with 4 bytes per data value, as usual in the Intel world in reverse order i.e. the least significant byte comes first.

GANAAS files can be read with the popular gamma spectra evaluation software by the IAEA and have the extension **.spe**. It is an ASCII format with a small header containing after a tag $SPEC_ID: a title, after a tag $MEAS.TIM: a line containing two numbers for the livetime and realtime, then a tag $DATE_MEA: and the date and time of the measurement start, then a tag $MCA_CAL: and two lines containing the order of the polynomial function for the energy calibration and the polynome coefficients. After a tag $DATA: follow the spectra data in ASCII with 10 values per line.

Another ASCII file format is the x y format with extension **.txt**. It can be read for example with Excel and contains the channel number and content as two decimal numbers in ASCII per line seperated by a TAB character.

A special ASCII format for 2D files with extension **.csv** can be read with the older MPAWIN software for the FAST ComTec MPA/PC multiparameter system (if the extension is changed into .asc). It has got a small header starting with a line [DISPLAY] and ending with a line [DATA] and then only for each non zero data point a line containing 3 values seperated by TAB characters, the x and y channel numbers and the channel content.

A variant of the standard binary format is the dual binary format with extension **.da2**. It is especially suited for multiscaling data recorded using both inputs and combines both time spectra into a single file, first from Count 1 Input and then from Count 2 Input. For data not recorded in MCS2 mode it is the same as the usual binary **.dat** format.

List files have the extension **.lst** and start with a header containing the usual report and configuration data in ASCII as in the .MCD files. The header ends with a line containing [DATA]. Then follows the data, depending on the format chosen for the data file either in ASCII one number per line, or in binary 4 bytes per number, as usual in the Intel world in the reverse order, i.e. the least significant byte comes first. For **MCA mode** the number must be masked with 0xffff to get the channel to be incremented in the spectra, the higher bits may contain tag bits if used, see section 3.5.6. (Mar-21-2005): If "**Timestamps**" are enabled in the settings dialog in the listmode box, the realtime and livetime counters are dumped into the listfile every time when event data are transfered. The time belongs to the event data immediately preceding the time stamp. The timestamp consists of three 32 bit words: a -1 as a characteristic tag, then the realtime and the livetime counters. Note that this counter is less than -1 if a corresponding preset is set, the preset is reached at a value of -1. In ASCII format the 3 numbers are in a single line and seperated by tabs.

List files in **MCS mode** are similar, but for each dwelltime two numbers are recorded, the memory address and the number of counts. In ASCII format these numbers are in the same line, seperated by a tab. At the beginning the memory address is the channel number, but it wraps around when the end of the board RAM is reached.

In **MCS2 mode** for each dwell time 3 numbers are recorded, the memory address and the numbers counted from both inputs. In ASCII format these numbers are in the same line, seperated by tabs.

4.2. Control Language

A sequence of commands that is stored in a file with extension .CTL can be executed by the MCA-3 server program with the „**Load**“ command. A lot of these commands are used in the configuration file MCA3A.CFG, also the header files with extension .mcd contain such commands to set the parameters. Each command starts at the beginning of a new line with a typical keyword. Any further characters in a line may contain a value or a comment. Following methods are available to execute commands:

- Load the command file using the Load command in the file menu.
- Enable remote mode in the server and send commands via the serial connection. The COMCTL.DLL is necessary which is part of the optional available MCDLAN software.
- Open a DDE connection and send the commands via DDE as described in chapter 4.3. The application name for opening the DDE connection with the standard MCA-3 server program MCA3.EXE is MCA3, the topic is MCA3-. Implemented are the DDE Execute to perform any command, and the DDE Request with items RANGE and DATA.
- Send the commands over a TCP/IP net using a remote shell and the optional available MCDLAN software. It is necessary to have TCP/IP networking installed and that the remote shell daemon program MCWNET is running. See the readme file on the installation disk.
- Send the commands via the DLL interface from LabVIEW, a Visual Basic program or any other application (software including the complete source code of the DLL and examples optional available).
- From your own Windows application, register a Windows message and then send the command as can be seen in the DLL source code.

The file MCA3A.CFG contains a complete list of commands for setting parameters. An example is:

lld=33	; Lower Level Discriminator
uld=4095	; Upper Level Discriminator
thd=98	; ADC Threshold
gate=0	; Gate enable
coin=0	; Gate mode: 0 means Anticoincidence, 1 Coincidence

sva=0	; 1 means Sampling mode for internal ADC
adcfilter=1	; Use Filter for ADC
range=8192	; Sets histogram length
prena=0	; Preset enable bits (hex): ; bit 0: Realtime preset enable ; bit 1: Lifetime preset enable ; bit 2: Sweep preset enable ; bit 3: ROI preset enable ; bit 5: ROI 2 preset enable in 2nd MCS spectrum
mcsmode=6	; (Operation mode bits (hex) ; bit 0: single MCS, ; bit1: dual MCS, bit0 and bit1=0: MCA mode ; bit 2: MCA Internal ADC ; bit 3: sequential mode ; bit 4: tagged ; bit 5: listmode ; bit 6: Software start (MCS) ; bit 7: time differences mode (MCS) ; bit 8: CH2 ref (diff. mode) (MCS) ; bit 10: enable sweep abort (MCS) ; bit 12: Use SCA input for Cnt1 (MCS)
roimin=0	; ROI lower limit (inclusive)
roimax=8192	; ROI upper limit (exclusive)
rtpreset=1000	; Realtimepreset value (seconds)
ltpreset=1000	; Livetimepreset value (seconds) (MCA mode only)
evpreset=10000	; ROIpreset value
swpreset=100	; Sweep preset value
ev2preset=10000	; ROI preset value for 2nd MCS spectrum
roi2min=1	; ROI lower limit (inclusive) in 2nd MCS spectrum
roi2max=8192	; ROI upper limit (exclusive) in 2nd MCS spectrum
rampmode=0;	; Usage of Ramp DAC: 0=fix, 1=triangle, 2=sawtooth
rampstart=0	; Ramp start or fix value (0..4095)
rampstep=1	; Ramp step height
rampbin=1	; Ramp step bin width
ramplen=4096	; Ramp no. of steps
qdacuse=0	; enable dac2 output
qdac=2048	; dac2 value (0..4095)
dac1use=0	; use ramp dac fixed in MCA mode
dac1val=0	; fix ramp dac value in MCA mode (0..4095)
impsel=f	; impedance + edge select (hex): ; bit 0..3 50 Ohm Clk, Start, In1, In2 ; bit 4..6 rising edge Clk, Start, In1=In2
thrstart=1716	; Threshold value for the trigger start (0..4095)
thr1=1716	; Threshold value for first MCS input
thr2=1716	; Threshold value for second MCS input
thrclk=1716	; Threshold value for Gate / Clock / Abort input

diguse=0	; Usage of DIG I/O port (hex): ; bit 0: pins 0-3 output status of MCD's ; bit 1: output digval and increment digval after stop ; bit 2: Invert polarity ; bit 3: Push-Pull output ; bit 4..7: Input pins 4..7 Trigger System 1..4 ; bit 8: GOWATCH ; bit 9: GO High at Start ; bit 10: GO Low at Stop
digval=0	; DIG I/O output value
cycles=10	; number of measurements for sequential mode
sequences=1	; number of sequential cycles
mempart=0	; memory offset in channels
dwelltime=7	; defines the dwelltime = (N+1) x 25 ns
dwellunit=0	; (hex) defines the dwelltime unit for the dialog ; bits 0..3: choice of units 0=ns, 1=usec, 2=msec, 3=sec ; bit 4: external clock (or manual channel advance)
syncout=f0f	; sync out (hex); bit 0..3 SyncOut1, 8..11 SyncOut2 ; bit 7 Inv SyncOut1, bit 15 Inv SyncOut2 ; 0=CLK, 1=GATE/ABORT/ADV, 2=TRG/SAMPLE, 3=COUNT1, ; 4=COUNT2, 5=LIVE_INT, 6=LIVE_EXT, 7=1kHz 8=SCA 9=ON, ; 10=EoBin, 11=BIN_DIV, 12=STEP_CNT, 13=GO, ; 14=RAMP_DOWN, 15="1"
autoinc=0	; 1=Enable Auto increment of filename
datname=SPECA.dat	; Filename
datname2=SPECB.dat	; Filename for 2nd MCS channel
savedata=0	; bit 0: save at Halt ; bit 1: write listfile (listmode only) ; bit 2 no Histogram (listmode only)
fmt=dat	; Format (ASCII: asc, Binary: dat, GANAAS: spe)
smoothpts=5	; Number of points to average for a smooth operation
caluse=0	; bit 0=1: Use calibration, higher bits: calibration formula
calch0=0.00	; First calibration point channel
calvl0=0.000000	; First calibration point value
calch1=100.00	; Second calibration point channel
calvl1=100.000000	; Second calibration point value
caloff=0.000000	; Calibration parameter: Offset
calfact=1.000000	; Calibration parameter: Factor
calunit=keV	; Calibration unit

The following commands perform actions and therefore usually are not included in a MCDx.CFG file:

start	; Clears the data and starts a new acquisition for system 1. Further ; execution of the .CTL file is suspended until any acquisition stops due ; to a preset.
start2	; Clears and starts system 2. Further execution is NOT suspended.
start3	; Clears and starts system 3. Further execution is NOT suspended.
start4	; Clears and starts system 4. Further execution is NOT suspended.
halt	; Stops acquisition of system 1 if one is running.

halt2	; Stops acquisition of system 2 if one is running.
halt3	; Stops acquisition of system 3 if one is running.
halt4	; Stops acquisition of system 4 if one is running.
cont	; Continues acquisition of system 1. If a time preset is already reached, ; the time preset is prolongated by the value which was valid when the ; „start„ command was executed. Further execution of the .CTL file is ; suspended (see start).
cont2	; Continues acquisition of system 2 (see start2).
cont3	; Continues acquisition of system 3 (see start3).
cont4	; Continues acquisition of system 4 (see start4).
savecnf	; Writes the settings into MCDA.INF (, MCDB.INF...)
MC_A	; Sets actual multichannel analyzer to MC_A for the rest of ; the controlfile.
MC_B ... **MC_D**	; Sets actual multichannel analyzer to MC_B ... MC_D for the ; rest of the controlfile.
savedat	; Saves data of actual multichannel analyzer. An existing file ; is overwritten.
pushname	; pushes the actual filename on an internal stack that can hold 4 ; names.
popname	; pops the last filename from the internal stack.
load	; Loads data of actual multichannel analyzer; the filename ; must be specified before with a command datname=...
add	; Adds data into actual multichannel analyzer; the filename ; must be specified before with a command datname=...
sub	; Subtracts data from actual multichannel analyzer; the filename ; must be specified before with a command datname=...
smooth	; Smoothes the data in actual multichannel analyzer
eras	; Clears the data of system 1.
eras2	; Clears the data of system 2.
eras3	; Clears the data of system 3.
eras4	; Clears the data of system 4.
exit	; Exits the server (and MCDWIN) programs
alert Message	; Displays a Messagebox containing Message and an OK ; button that must be pressed before execution can continue.
waitinfo 5000 Message	; Displays a Messagebox containing Message, an OK ; and an END button. After the specified time (5000 msec) ; the Messagebox vanishes and execution continues. OK ; continues immediately, END escapes execution.
beep *	; Makes a beep. The character '*' may be replaced with '?', '!' or ; left empty. The corresponding sound is defined in the WIN.INI ; file in the [sounds] section.
delay 4000	; Waits specified time (4000 msec = 4 sec).
pulse 100	; Outputs a pulse of 100 ms duration at dig 3 (pin 2).
waitpin 4000	; Waits 4000 ms for going the level at dig 7 (pin 13) going low. ; After a timeout a Messagebox warns and waits for pressing OK ; Can be used for connecting a sample changer.
sweep	; Starts a Sweep (Software-Trigger).

advance	; Advances the timebin channel in manual mode.
run controlfile	; Runs a sequence of commands stored in controlfile. This ; command cannot be nested, i.e. it is not possible to execute ; a run command from the controlfile called.
onstart command	; The command is executed always after a start action when the ; acquisition is already running. The command can be any valid ; command, also 'run controlfile' is possible.
onstart off	; Switches off the 'onstart' feature. Also a manual Stop command ; switches it off.
onstop command	; The command is executed always after a stop caused by a ; preset reached or trigger. This can be used to program measure ; cycles. For example the command 'onstop start' makes a ; loop of this kind.
onstop off	; Switches off the 'onstop' feature. Also a manual Stop command ; switches it off.
lastrun=5	; Defines the file count for the last run in a measure cycle. After a ; file with this count or greater was saved with autoinc on, instead ; of the 'onstop command' the 'onlast command' is executed.
numruns=5	; Defines the file count for the last run in a measure cycle. The ; last count is the present one plus the numruns number.After a ; file with this count was saved with autoinc on, instead of the ; 'onstop command' the 'onlast command' is executed.
onlast command	; The command is executed after a stop caused by a preset ; reached or trigger instead of the 'onstop command', when the ; last file count is reached with autoinc on. This can be used to ; finish programmed measure cycles.
onlast off	; Switches off the 'onlast' feature. Also a manual Stop command ; switches it off.
abortloop	; Aborts smoothly an acquisition loop programmed with "onstop" ; and "onlast". Terminates immediately the running acquisition ; and executes the "onlast" command.
endloop	; Ends smoothly an acquisition loop programmed with "onstop" ; and "onlast". The presently running acquisition is continued until ; a preset is reached, then the "onlast" command is executed.
onstart2 command	; The command is executed always after a start2 action.
onstart3 command	; The command is executed always after a start3 action.
onstart4 command	; The command is executed always after a start4 action.
onstart2 off	; Switches off the 'onstart2' feature.
onstart3 off	; Switches off the 'onstart3' feature.
onstart4 off	; Switches off the 'onstart4' feature.
onstop2 command	; The command is executed always after a stop of system2 ; caused by a preset reached. This can be used to ; program measure cycles. For example the command ; 'onstop2 start2' makes a loop of this kind.
onstop3 command	; The command is executed always after a stop of system3 ; caused by a preset reached.
onstop4 command	; The command is executed always after a stop of system4 ; caused by a preset reached.
onstop2 off	; Switches off the 'onstop2' feature. Also a manual Stop2 ; command switches it off.

onstop3 off	; Switches off the 'onstop3' feature. Also a manual Stop3 ; command switches it off.
onstop4 off	; Switches off the 'onstop4' feature. Also a manual Stop4 ; command switches it off.
numruns2=5	; Defines the number of runs in a measure cycle with "onstop2".
numruns3=5	; Defines the number of runs in a measure cycle with "onstop3".
numruns4=5	; Defines the number of runs in a measure cycle with "onstop4".
onlast2 command	; The command is executed after a stop caused by a preset ; reached or trigger instead of the 'onstop2' command, when the ; last file count is reached with autoinc on. This can be used to ; finish programmed measure cycles.
onlast3 command	; See onlast2, for loops with 'onstop3'.
onlast4 command	; See onlast2, for loops with 'onstop4'.
onlast2 off	; Switches off the 'onlast2' feature. Also a manual Stop2 command ; switches it off.
onlast3 off	; Switches off the 'onlast3' feature. Also a manual Stop3 ommand ; switches it off.
onlast4 off	; Switches off the 'onlast4' feature. Also a manual Stop4 ommand ; switches it off.
abortloop2	; Terminates immediately the running acquisition and executes ; the "onlast2" command.
abortloop3	; Terminates immediately the running acquisition and executes ; the "onlast3" command.
abortloop4	; Terminates immediately the running acquisition and executes ; the "onlast4" command.
endloop2	; The presently running acquisition is continued until a preset is ; reached, then the "onlast2" command is executed.
endloop3	; The presently running acquisition is continued until a preset is ; reached, then the "onlast3" command is executed.
endloop4	; The presently running acquisition is continued until a preset is ; reached, then the "onlast4" command is executed.
exec program	; Executes a Windows program or .PIF file. ; Example: exec notepad test.ctl ; opens the notepad editor and loads test.ctl.
deleteallrois	; Deletes all ROIs in the active Display of MCDWIN or the active ; multichannel analyzer if MCDWIN is not running.
deleteallrois MC_A	; Similar to the deleteallrois command, but using the argument ; allows to specify which spectrum should be treated ; independently of which child window is activated in MCDWIN
fitrois	; Makes a single peak Gaussian fit for all ROIs in the active ; Display of MCDWIN and dumps the result into a logfile. This is ; performed by the MCDWIN program and therefore can be ; made only if this application is running.
fitrois MC_A	; Similar to the fitroi command, but using the argument allows to ; specify which spectrum should be evaluated independently of ; which child window is activated in MCDWIN
autocal	; Makes a single peak Gaussian fit for all ROIs in the active ; Display of MCDWIN for which a peak value was entered in the ; MCDWIN Region Edit dialog and uses the results for a

; calibration. This is performed by the MCDWIN program and
; therefore can be made only if this application is running.

autocal MC_A ; Similar to the autocal command, but using the argument allows
; to specify which spectrum should be evaluated independently of
; which child window is activated in MCDWIN

The following commands make sense only when using the serial line or LAN control:

MC_A? ; Sends the status of MC_A via the serial port and make
; MC_A actual.

MC_B? ... MC_D? ; Sends the status of MC_B ... MC_D via the serial port and
; make MC_B ... MC_D actual.

? ; Send the status of the actual multichannel analyzer

RROI(0,1) ; Sends the sum, mean value and max positive and negative
; deviation from mean of rectangular ROI #1 in spectra #0

sendfile filename ; Sends the ASCII file with name ‘filename’ via the serial line connection.

The execution of a control file can be finished from the Server or MCDWIN with any Halt command.

4.3. Controlling the MCA-3 Windows Server via DDE

The MCA3 program can be a server for DDE (Dynamic Data Exchange). Many Windows software packages can use the DDE standard protocols to communicate with other Windows programs, for example GRAMS, FAMOS or LabVIEW. In the following the DDE capabilities of the MCA3 program are described together with a demo VI („Virtual Instrument") for LabVIEW. It is not recommended to use the DDE protocol for LabVIEW, as also a DLL interface is available that is much faster. The following should be seen as a general description of the DDE conversation capabilities of the MCA3 program.

4.3.1. Open Conversation

application: MCA3
topic: MCA3-

Any application that wants to be a client of a DDE server, must open the conversation first by specifying an application and a topic name. The application name is MCA3 and the topic is MCA3-.

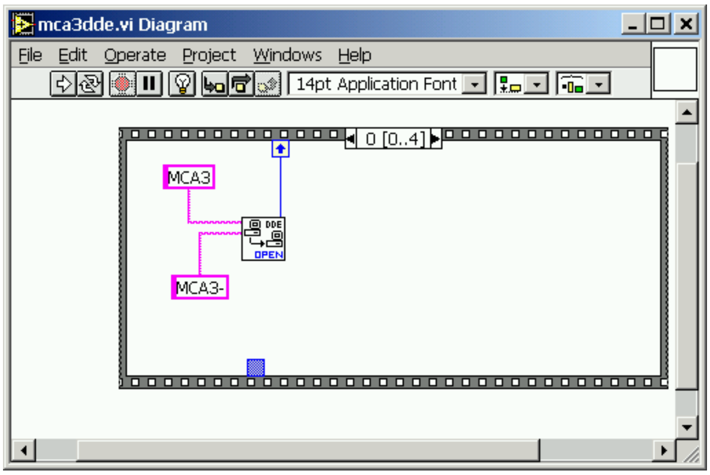


Figure 4.12: Opening the DDE conversation with the MCA-3 in LabVIEW

4.3.2. DDE Execute

The DDE Execute command can be used to perform any action of the MCA3 program. Any of the Control command lines described in section 4.2 can be used. For example a sequence of control commands saved in a file TEST.CTL can be executed by specifying the command

RUN TEST.CTL

The MCA3 program then executes the command and, after finishing, it sends an Acknowledge message to the DDE client. This can be used to synchronize the actions in both applications.

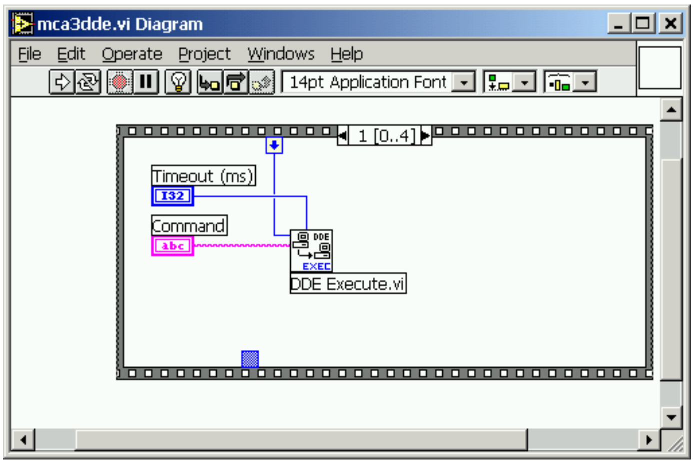


Figure 4.13: Executing a MCA-3 command from a LabVIEW application

4.3.3. DDE Request

The DDE Request is a message exchange to obtain the value of a specified item. Only two items are defined for DDE request up to now: RANGE and DATA. The value is obtained as an ASCII string, i.e. it must be converted by the client to get the numbers. All other parameters concerning the setup can be obtained by the client application by reading and evaluating the configuration file.

RANGE

The RANGE item can be used to obtain the total number of data.

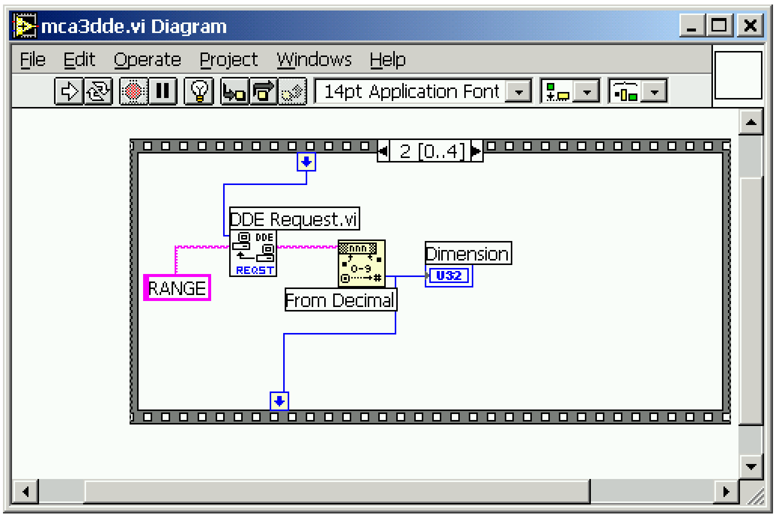


Figure 4.14: Getting the total number of data with LabVIEW

DATA

With the DATA item the data is obtained. The value of this item is a multiline string that contains in each line a decimal number as an ASCII string.

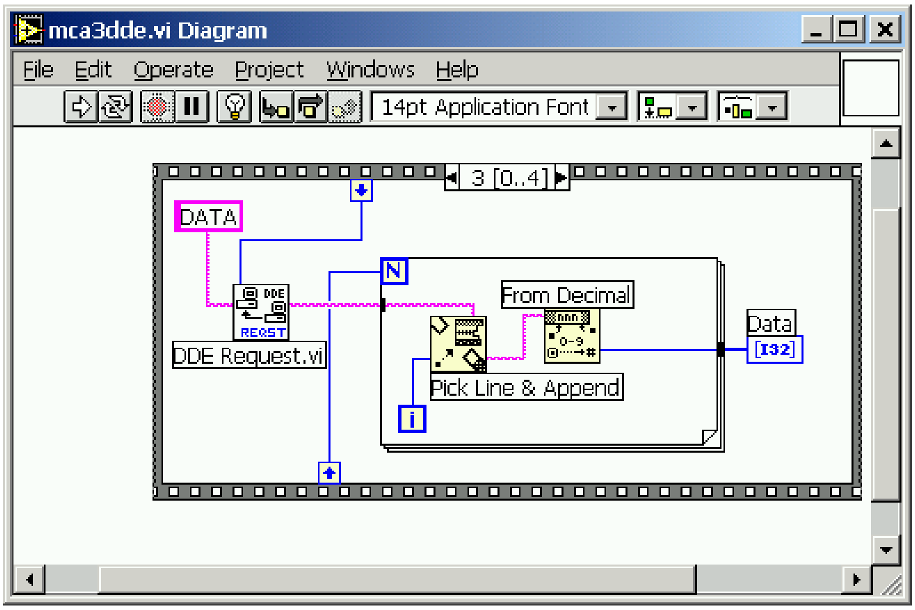


Figure 4.15: Getting the data with LabVIEW

4.3.4. Close Conversation

After finishing the DDE communication with the server program, it must be closed.

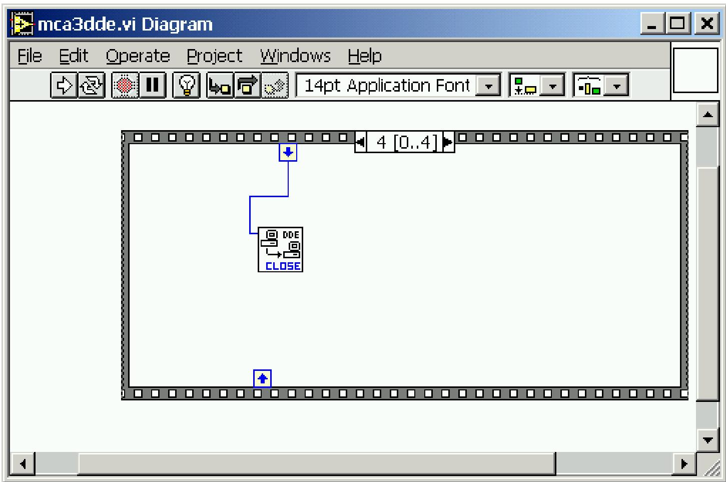


Figure 4.16: Closing the DDE communication in LabVIEW

The following figure shows the „Panel“ of the described VI for LabVIEW.

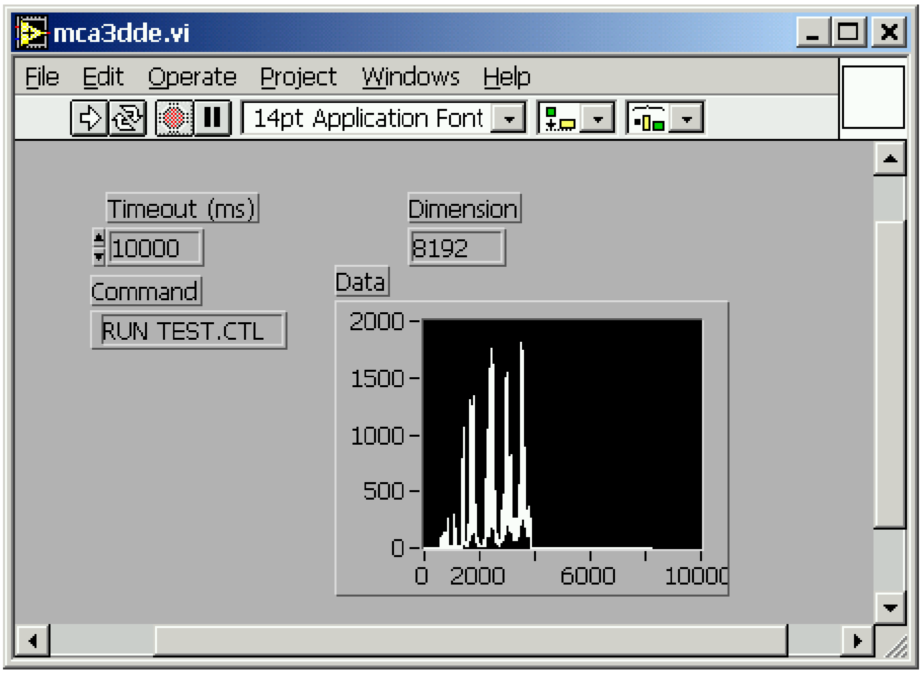


Figure 4.17: Control Panel of the demo VI for LabVIEW

4.3.5. DDE Conversation with GRAMS/386

The following file GRAMSMCA3.CIF can be used to get the MCA3 data into GRAMS/386 via DDE using the „Collect" menu:

```
MCA3 DDE Test
Query
MCA3
MCA3-
DATA


save

end
spc
1 second
```

4.4. Controlling the MCA3 Windows Server via DLL

The MCA-3 server program provides access to all functions, parameters and data via a DLL („dynamic link library"). So the server can be completely controlled by the MCDWIN software that provides all necessary graphic displays.

In the following some parts of the header and definition files of the DMCA3.DLL are listed, that may help an experienced programmer to use the DLL for own applications. The arguments „item, nDevice, nDisplay, nSystem" are only for compatibility with other devices controlled by the MCDWIN software and must be zero.

NOTE:

The complete documented **sourcecode of the DLL** including fundamental VI's and an example VI for **LabVIEW** and an example **Visual Basic** and **C** program is available as an **option**.

```
typedef struct{
  int started;          // aquisition status: 1 if running, 0 else
```

```
  double runtime;       // realtime in seconds
  double totalsum;      // total events
  double roisum;        // events within ROI
  double totalrate;     // acquired total events per second

  double nettosum;      // ROI sum with background subtracted
  double sweeps;        // Livetime in seconds or sweeps for MCS mode
  double deadtime;      // deadtime in percent
  unsigned long maxval; // Maximum value in spectrum
} ACQSTATUS;

typedef struct{
  unsigned long range;  // spectrum length, ADC range
  long prena;           // bit 0: realtime preset enabled
                                 // bit 1: livetime preset enabled
                        // bit 2: sweep preset enabled
                        // bit 3: ROI preset enabled
                        // bit 5: ROI2 preset enabled
  long mcsmode;         //
                        // bit 0: MCS 1 channel
                        // bit 1: MCS 2 channel
                        // mcsmode & 0x3==0: MCA mode
                        // bit 2: MCA int. ADC        (MCA)
                        // bit 3: sequential mode (MCA+MCS)
                        // bit 4: tagged          (MCA+MCS)
                        // bit 5: listmode        (MCA+MCS)
                        // bit 6: Software Start      (MCS)
                        // bit 7: differential mode   (MCS)
                        // bit 8: Ch2 ref (diff.mode) (MCS)
                        // bit 9: Autocorrelation     (MCS)
                        // bit 10: enable sweep abort (MCS)
                        // (bit 11: synchron tag) always on)   (MCA+MCS)
                        // bit 12: use SCA Input  (MCS)
  unsigned long roimin; // lower ROI limit
  unsigned long roimax; // upper limit: roimin <= channel < roimax
  double eventpreset;   // ROI preset value
  double rtpreset;      // time preset value
  long savedata;        // bit 0: 1 if auto save after stop
                        // bit 1: write listfile
                        // bit 2: listfile only, no histogram
  long fmt;             // format type: 0 == ASCII, 1 == binary
  long autoinc;         // 1 if auto increment filename
  long diguse;          // Use of Dig I/O, GO Line:
                        // bit 0: status dig 0..3
                        // bit 1: Output digval and
                        // increment digval after stop
                        // bit 2: Invert polarity
                        // bit 3: Push-Pull output
                       // bit 4..7:  Input pins 4..7 Trigger System 1..4
                        // bit 8: GOWATCH
                        // bit 9: GO High at Start
                        // bit 10: GO Low at Stop
  long digval;          // digval=0..255 value for samplechanger
  long cycles;          // for sequential mode
  long sequences;       // for repeated sequential mode
  long mempart;         // number of active part
  unsigned long dwelltime;  // 32 bit value for FPGA
  long dwellunit;       // choice of units in dialog
                        // 0=ns, 1=usec, 2=msec, 3=sec
                        // bit 4: external clock (or manual channel
advance)
  long syncout;         // sync out; bit 0..3 SyncOut1, 8..11 SyncOut2
                        // bit 7 Inv SyncOut1, bit 15 Inv SyncOut2
                        // 0=CLK, 1=GATE/ABORT/ADV,
                        // 2=TRG/SAMPLE, 3=COUNT1, 4=COUNT2, 5=LIVE_INT,
                        // 6=LIVE_EXT, 7=1kHz 8=SCA 9=ON, 10=EoBin,
                        // 11=BIN_DIV, 12=STEP_CNT, 13=GO,
                        // 14=RAMP_DOWN, 15="1"
  double ltpreset;      // Livetime preset value (MCA)
  long nregions;        // number of regions
  long caluse;          // bit 0 == 1 if calibration used,
```

```
                        // higher bits: formula
  double swpreset;      // sweep preset value
  long active;          // 1 for module enabled in system 1
  long calpoints;       // number of calibration points
} ACQSETTING;

typedef struct{
  long pol0;            // Handshake polarities
  long timerticks;      // realtime, livetime clocks per sec
  long timebase;        // elementary dwelltime unit nsec
  long dwellmin;        // minimum dwelltime
  long sernum;          //
  long version;         //
  long type;            // Bit0=0: SCA present,
                        // Bit1=0: Internal ADC,
                        // Bit2=0: extern ADC,
                        // Bit3=0: Listmode
                        // Bit4=0: MCS mode
                        // Bit5: reserved, presently=1
} BOARDPROPERTY;

typedef struct{
  long thrstart;          // DAC 6
  long thr1;              // DAC 7
  long thr2;              // DAC 8
  long thrclk;            // DAC 5
  long qdacuse;           // enable dac2 output
  long qdac;              // DAC 1 (=dac2)
  long dac1use;           // use ramp dac fixed in MCA mode
  long dac1val;           // fix ramp dac value in MCA mode
  long rampmode;          // 0=fix, 1=triangle, 2=sawtooth
  long rampstart;             // ramp start value,
  long rampstep;          // step height
  long rampbin;           // step bin width
  long ramplen;           // no. of steps
  long impsel;            // impedance + edge select:
                          // bit 0..3 50 Ohm Clk, Start, In1, In2
                          // bit 4..6 rising Clk, Start, In1=In2
  unsigned long roi2min;
  unsigned long roi2max;
  double roi2preset;
} DACSETTING;

typedef struct{
  long uld;                   // upper level discriminator = DAC 2
  long lld;                   // lower level discriminator = DAC 3
  long thd;                 // threshold = DAC 4
  long gate;                  // gate enable
  long coin;                  // 1=coin or rising, 0=anti or falling
  long filter;                // 1=filter on
  long sva;                   // external sampled conversion
  long slider;                // 0=OFF, 256, 1024, 8192 sliderrange
  // auxiliary for dialog box:
  long thrstart;              // DAC 6 threshold for sampling
  long thrclk;                // DAC 5 threshold for gate
  long impsel;                // impedance + edge select:
                              // bit 0..1 50 Ohm Gate, Start
                              // bit    5 rising  Start
} ADCSETTING;

typedef struct{
  unsigned long HUGE *s0;          // pointer to spectrum
  unsigned long far *region;       // pointer to regions
  unsigned char far *comment0;     // pointer to strings
  double far *cnt;                 // pointer to counters
  HANDLE hs0;
  HANDLE hrg;
  HANDLE hcm;
  HANDLE hct;
} ACQDATA;
```

```
typedef struct {
  int nDevices;              // Number of spectra = number of modules
  int nDisplays;             // Number of active displays 0...nDevices
  int nSystems;              // Number of systems 0...4
  int bRemote;               // 1 if server controlled by MCDWIN
  unsigned int sys;          // System definition word:
                           // bit0=0, bit1=0: MCD#0 in system 1
                           // bit0=1, bit1=0: MCD#0 in system 2
                           // bit0=0, bit1=1: MCD#0 in system 3
                           // bit0=1, bit1=1: MCD#0 in system 4
                           // bit2=0, bit3=0: MCD#1 in system 1 ...
                           // bit6=1, bit7=1: MCD#3 in system 4
} ACQDEF;


/*** FUNCTION PROTOTYPES (do not change) ***/
BOOL APIENTRY DllMain(HANDLE hInst, DWORD ul_reason_being_called, LPVOID
lpReserved);
VOID APIENTRY StoreSettingData(ACQSETTING FAR *Setting, int nDisplay);
                                   // Stores Settings into the DLL
int APIENTRY GetSettingData(ACQSETTING FAR *Setting, int nDisplay);
                                   // Get Settings stored in the DLL
VOID APIENTRY StoreExtSettingData(EXTACQSETTING FAR *Setting, int
nDisplay);
                                   // Stores extended Settings into the
DLL
int APIENTRY GetExtSettingData(EXTACQSETTING FAR *Setting, int
nDisplay);
                            // Get extended Settings stored in the DLL
VOID APIENTRY StoreStatusData(ACQSTATUS FAR *Status, int nDisplay);
                                       // Store the Status into the DLL
int APIENTRY GetStatusData(ACQSTATUS FAR *Status, int nDisplay);
                                      // Get the Status
VOID APIENTRY Start(int nSystem);     // Start
VOID APIENTRY Halt(int nSystem);      // Halt
VOID APIENTRY Continue(int nSystem);  // Continue
VOID APIENTRY NewSetting(int nDevice);// Indicate new Settings to Server
UINT APIENTRY ServExec(HWND ClientWnd);  // Execute the Server WLAP.EXE
VOID APIENTRY StoreData(ACQDATA FAR *Data, int nDisplay);
                                   // Stores Data pointers into the DLL
int APIENTRY GetData(ACQDATA FAR *Data, int nDisplay);
                                   // Get Data pointers
long APIENTRY GetSpec(long i, int nDisplay);
                                         // Get a spectrum value
VOID APIENTRY SaveSetting(void);         // Save Settings
int APIENTRY GetStatus(int nDevice);// Request actual Status from Server
VOID APIENTRY Erase(int nSystem);        // Erase spectrum
VOID APIENTRY SaveData(int nDevice);     // Saves data
VOID APIENTRY GetBlock(long FAR *hist, int start, int end, int step,
  int nDisplay);                        // Get a block of spectrum data
VOID APIENTRY StoreDefData(ACQDEF FAR *Def);
                                     // Store System Definition into DLL
int APIENTRY GetDefData(ACQDEF FAR *Def);
                                          // Get System Definition
VOID APIENTRY LoadData(int nDisplay);     // Loads data
VOID APIENTRY AddData(int nDisplay);      // Adds data
VOID APIENTRY SubData(int nDisplay);      // Subtracts data
VOID APIENTRY Smooth(int nDisplay);       // Smooth data
VOID APIENTRY NewData(void);          // Indicate new ROI or string Data
VOID APIENTRY HardwareDlg(int item);  // Calls the Settings dialog box
VOID APIENTRY UnregisterClient(void); // Clears remote mode from MCDWIN
VOID APIENTRY DestroyClient(void);        // Close MCDWIN
UINT APIENTRY ClientExec(HWND ServerWnd);
                                      // Execute the Client MCDWIN.EXE
int APIENTRY LVGetDat(unsigned long HUGE *datp, int nDisplay);
                                      // Copies the spectrum to an array
VOID APIENTRY RunCmd(int nDisplay, LPSTR Cmd);
                                      // Executes command
int APIENTRY LVGetRoi(unsigned long FAR *roip, int nDisplay);
                               // Copies the ROI boundaries to an array
```

```
int APIENTRY LVGetCnt(double far *cntp, int nDisplay);
                               // Copies Cnt numbers to an array
int APIENTRY LVGetStr(char far *strp, int nDisplay);
                               // Copies strings to an array

EXPORTS
;       Functions in dmca3.c
        StoreSettingData            @2
        GetSettingData              @3
        StoreStatusData             @4
        GetStatusData               @5
        Start                       @6
        Halt                        @7
        Continue                    @8
        NewSetting                  @9
        ServExec                    @10
        StoreData                   @11
        GetData                     @12
        GetSpec                     @13
        SaveSetting                 @14
        GetStatus                   @15
        Erase                       @16
        SaveData                    @17
        GetBlock                    @18
        StoreDefData                @19
        GetDefData                  @20
        LoadData                    @21
        NewData                     @22
        HardwareDlg                 @23
        UnregisterClient            @24
        DestroyClient               @25
        ClientExec                  @26
        LVGetDat                    @27
        RunCmd                      @28
        AddData                     @29
        LVGetRoi                    @30
        LVGetCnt                    @31
        LVGetStr                    @32
        SubData                     @33
        Smooth                      @34
        StoreExtSettingData         @35
        GetExtSettingData           @36
```

5. MCDWIN Software

The window of the MCDWIN program is shown here. It enables the full control of the MCA-3 card via the server program to perform measurements, save data and to show the data online in several windows.

The server program MCA3.EXE automatically starts MCDWIN. If you try to start MCDWIN in advance to the server, a message box warns that you should start the server first.

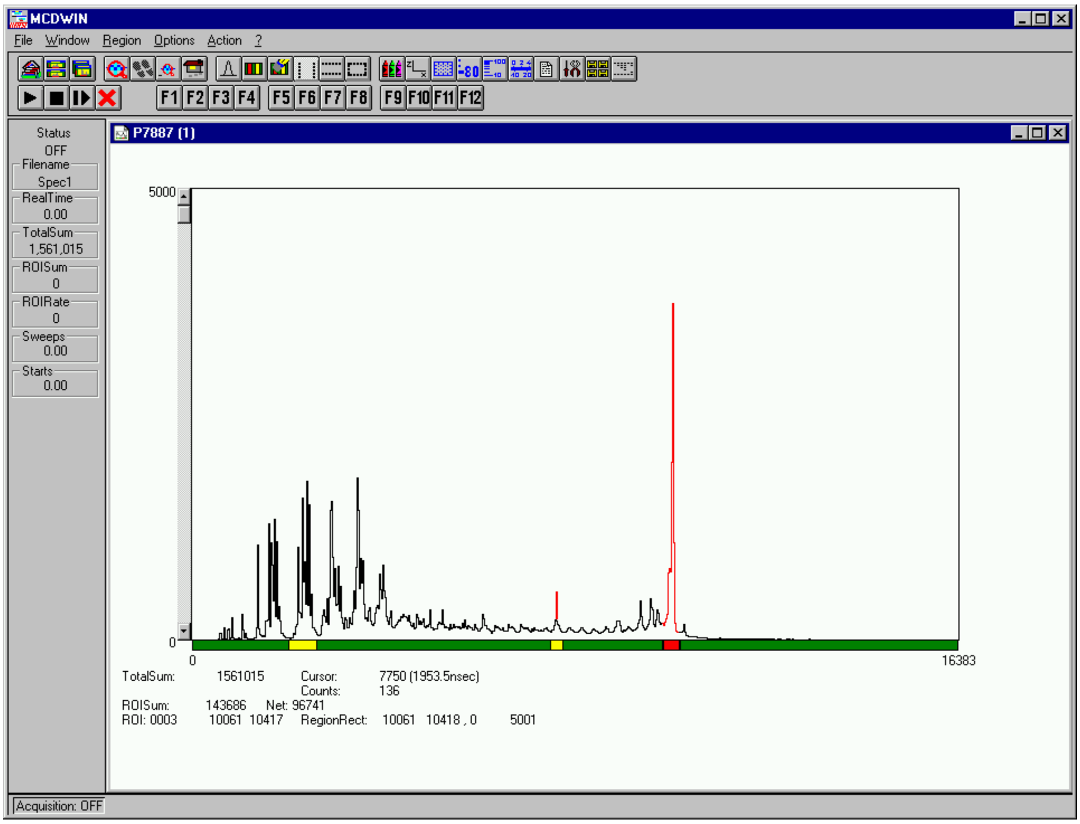


Figure 5.1: MCDWIN main window

A status window at the left side gives all information about the status of the MCA-3. A toolbar provides fast access to many used functions in the menu. A status bar at the bottom gives help about the meaning of the toolbar icons. A cursor appears when clicking the left mouse button inside the graphics area. To get rid of the cursor, make a double click with the right mouse button outside the graphics area. To define a region, press the right mouse button, and while keeping the button pressed, drag a rectangle. In zoomed state a scrollbar appears that allows to scroll through the spectrum.

MCDWIN has also viewing capabilities for two dimensional spectra. A single spectrum can be converted into a two dimensional one by specifying the x dimension in the display option dialog. It is possible to drag a rectangle and zoom into this rectangle. Rectangular ROIs can be set and the ROISum and Net ROISum is displayed. The Net Sum is calculated the same way like in the single view, by subtracting a linear interpolated background from the both outmost channels in x-direction. This Net sums are then summed up in y-direction. The ROI editing dialog is changed into a Rectangular Editing dialog for MAP and ISO displays. The Cursor can be moved in x and y direction using the mouse and the arrow keys, in ISO display only using the arrow keys.

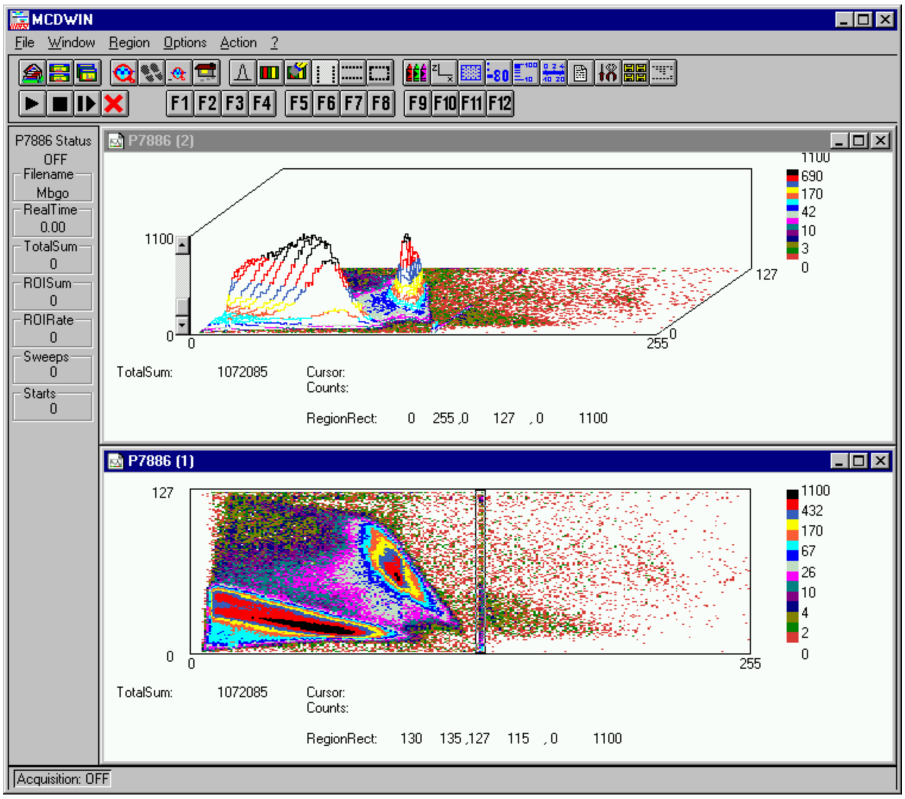


Figure 5.2: MCDWIN Map and Isometric display

A status window at the left side gives all information about the status of the MCA-3. A toolbar provides fast access to many used functions in the menu. A status bar at the bottom gives help about the meaning of the toolbar icons. A cursor appears when clicking the left mouse button inside the graphics area. The cursor can be moved using the arrow keys. To get rid of the cursor, make a double click with the right mouse button outside the graphics area. To define a region, press the right mouse button, and while keeping the button pressed, drag a rectangle. In zoomed state a scrollbar appears that allows to scroll through the spectrum.

In the following the several menu functions are described together with the corresponding toolbar icons.

5.1. File Menu

Load..., Add..., Save, Save As...

These menu items provide the usual functions for loading and saving data common to most Windows programs. When saving data, you have the choice between binary (.DAT) and ASCII (.ASC) format. When you load data, select a header file (extension .mcd). This file contains the information about the length and format of the data file, which is then automatically read.

It is also possible to load a file with extension .CTL containing commands which are then executed. With „**Add**“ the data is added to the present data. The data loaded from a file is corrected according to the calibration, if available.

Open New...

With the Open New menu item or the corresponding icon a new Display window will be created and shown as the active window.

Open All

By selecting the Open All menu item, all available Displays are shown. The window of the last opened Display becomes active.

Print...

The Print menu item opens the print dialog. It allows to arrange several pictures on a page into zones. The number of zones in vertical and horizontal direction can be specified. The Color can be black/white, RGB (colored) or Gray scale. RGB is recommended also for black laser printers. Some info lines containing date, filename and title can be added. For each page a temporary file PRINT1.WMF, PRINT2.WMF... will be created. This file is in Windows Metafile format and can be exported into some other Windows applications.

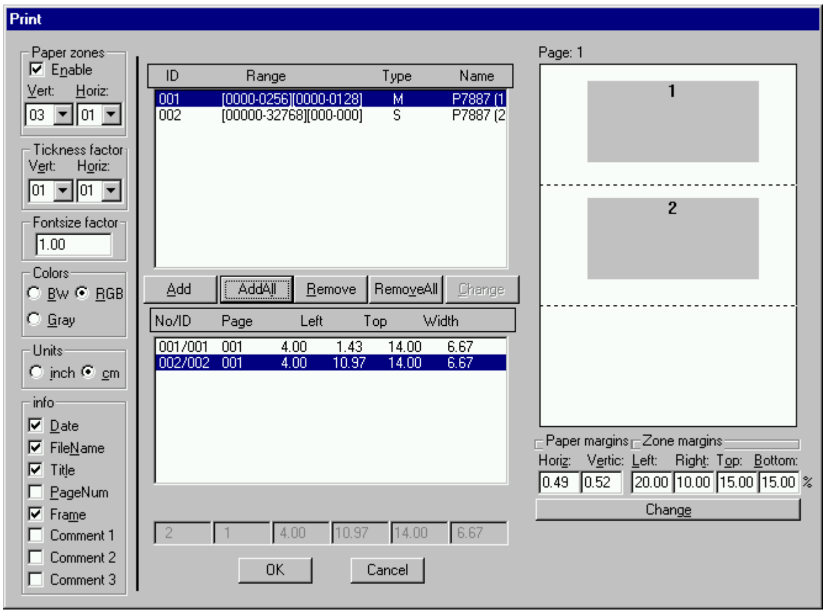


Figure 5.3: Print dialog box

> **NOTE:**
>
> If printing takes a long time and disk activity is high, please note the following: The picture for the printing is first built in the memory, but it may need quite a lot of memory if the printer resolution is high and therefore Windows makes intense virtual memory swapping to disk if for example only 8 MB RAM are available. Therefore it is recommended: never use a 600 dpi printer driver for the printout of spectra. For example for an HP Laser 4, install the PCL driver and use 300 dpi. The PCL driver is also much more effective than a Postscript driver, printing is much faster. With 600 dpi, the maximum figure size is indeed limited to about 12 cm x 7 cm (Windows 9x cannot handle on an easy way bitmaps larger than 16 MB).

Setup Printer...

The Setup Printer menu item allows to configure the printer.

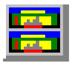
Exit

The Exit menu item exits the MCDWIN.

5.2. Window Menu

The Window menu allows to arrange the Display windows.

Tile



With the Tile menu item or clicking the corresponding icon, all opened and displayed MCDWIN Display windows are arranged over the full MCDWIN client area trying to make the same size for all windows.

Cascade

The Cascade menu item or respective icon arranges all windows in a cascade.

Arrange Icons

By the Arrange Icons menu item, the minimized MCDWIN Display windows are arranged in a series at the bottom of the MCDWIN client area.

Close All

By selecting the Close All menu item, all Display windows are closed.

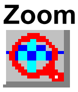
Window list

At the end of the Window menu, all created Display windows are listed with their names, the current active window is checked. By selecting any of the names, this window becomes the active and is displayed in front of all others.

5.3. Region Menu

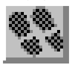
The Region menu contains commands for Regions and ROIs (Regions of Interest). A Region can be marked in a display with the mouse using the right mouse button by dragging a rectangle over the area one is interested in. A ROI, i.e. an already defined region in a single spectra can be shown zoomed by double-clicking with the left mouse button on the corresponding colored area in the bar at the bottom of the spectra display. A single mouse click with the left button on the corresponding colored area makes this to the active ROI and lets the counts contained in this ROI be displayed in the information lines of the respective window.

Zoom



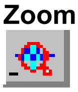
The Zoom item or respective icon enlarges a Region to the maximum Display Spectrum size.

Back



The Back menu item or clicking the corresponding icon restores the last zoom view. A successive Back command returns to the previous view.

Zoom Out



The Zoom Out menu item or clicking the corresponding icon reduces the actual zoom factor by 2, if applicable.

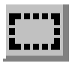
Home

Clicking the Home menu item or the corresponding icon restores a Display to the basic configuration.

Shape

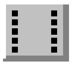
Selecting the Shape menu item opens a submenu with the items Rectangle, X-Slice, Y-Slice, and Polygon to choose the ROI shape.

Rectangle



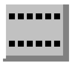
Sets the region shape to a rectangle with arbitrary dimensions. To enter the rectangular region, press the right mouse button, drag a rectangle, and release the button to define the region.

X-Slice



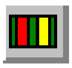
Sets the Region shape to the rectangle with maximum height.

Y-Slice



Sets the Region shape to the rectangle with maximum width.

Create



The Create menu item creates a new ROI from the current marked Region.

Delete

By selecting the Delete menu item or the respective icon, the current active ROI is deleted and the previously defined ROI is activated.

Edit...

With the Edit item, a dialog box is opened which allows to edit the ROI list, i.e. create a new one, delete, change and activate an existing ROI. Also the peak values (e.g. energy, mass etc.) for an automatic calibration can be entered here. A ROI can be edited and added to the list. It can also be made to the „Active ROI“, that is the special ROI that is used by the server program to calculate the events within this ROI and look for an event preset. The ROI list can be cleared and can be written to a file with extension .CTL, which can be directly loaded into the server to restore the ROI list.

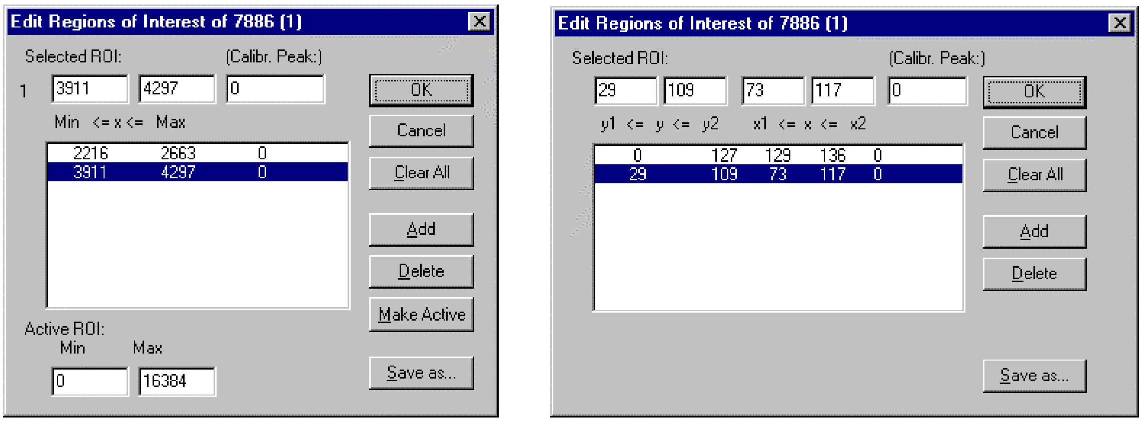
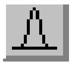


Figure 5.4: ROI Editing dialog box, left: Single spectra, right: 2D spectra

The selected ROI can be changed by clicking on it in the ROI list. In the MCDWIN spectrum display the total and net sum of the selected ROI is displayed.

Fit...



By selecting the Fit... menu item or the respective icon, A single Gaussian peak fit with linear background is performed for the currently marked region. The fitted curve is displayed and a dialog box shows the results:

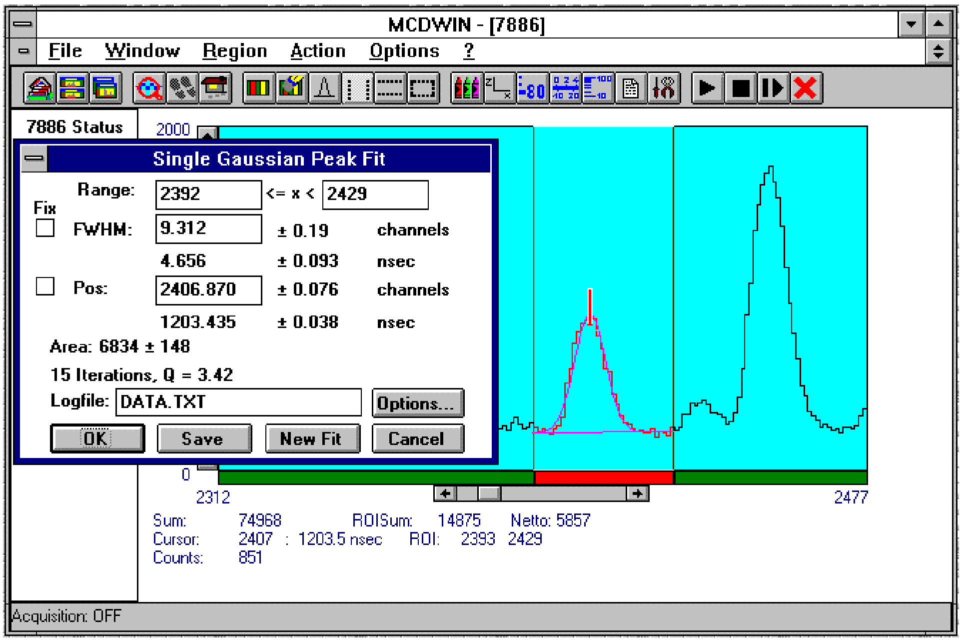


Figure 5.5: Single Gaussian Peak Fit

The full width at half maximum FWHM and Position of the Gaussian can be changed and a **New Fit** can be performed, they even can be fixed to the entered value by marking the respective checkbox. The Position and FWHM are displayed in channels and also in calibrated units, if a calibration is available. The area of the Gaussian is also shown. For all values also the standard deviations are given. The value of Q is the normalized chi**2. To take into account the systematic error of the lineshape, you may multiply the errors with the squareroot of Q. Click on **Save** to append a line containing the results to a **Logfile** with the specified name. **OK** closes the dialog and the fitted function remains in the display - also if the display is refreshed -, whereas after **Cancel** the curve no longer will be shown in a refreshed display. **Options...** opens a new dialog box to define the information in the logfile:

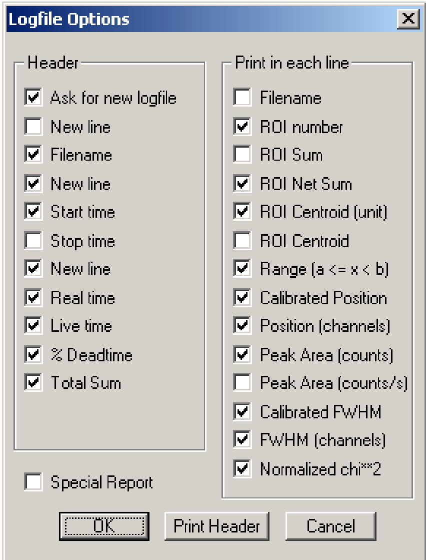


Figure 5.6: Log file Options for the Single Gaussian Peak Fit

The several quantities are written in standard text format with Tabs as separators and a Newline character at the end of each line, so the file can be read with standard calculation programs like EXCEL. Click on **Print Header** to write a header line.

Fit ROIs

With the Fit ROIs item, for all ROIs a Single Gaussian Peak Fit is performed and the results are dumped into the logfile.

Auto Calib

Makes a Gauss fit for all ROIs in the active Display for which a peak value was entered, and performs a calibration using the fit results.

5.4. Options Menu

The Options Menu contains commands for changing display properties like scale, colors etc., hardware settings, calibration and comments.

Colors...

The Colors menu item or respective icon opens the Colors dialog box.

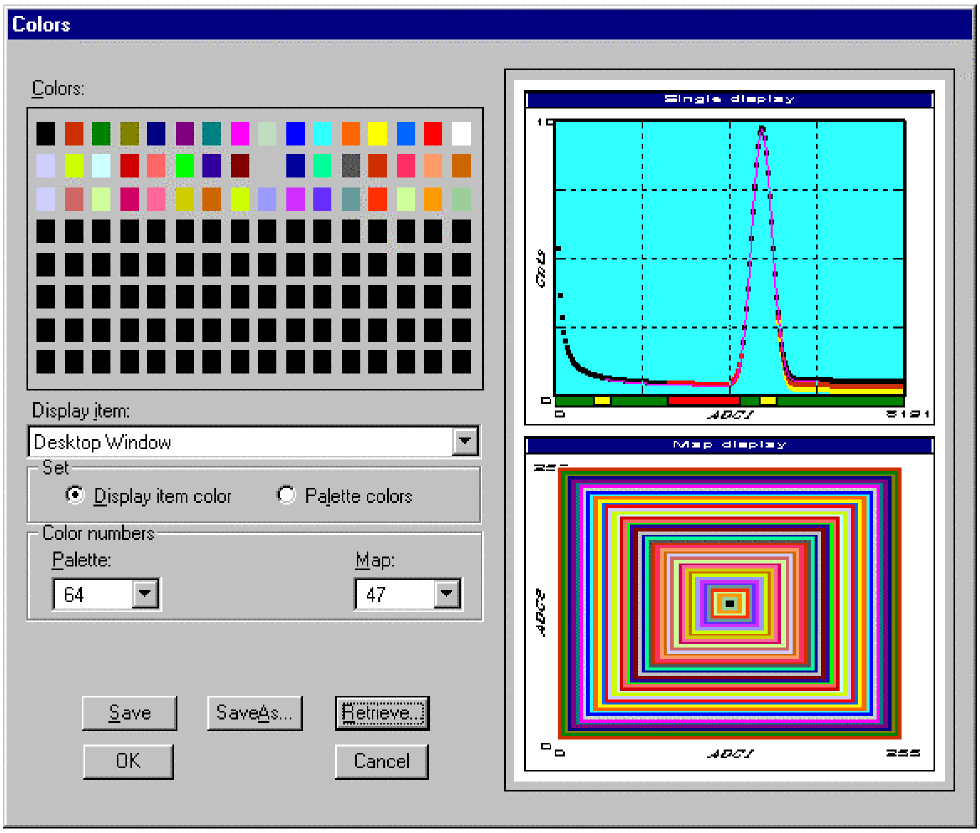


Figure 5.7: Colors dialog box

It changes the palette or Display element color depending on which mode is chosen. The current color and palette setup may be saved or a new one can be loaded.

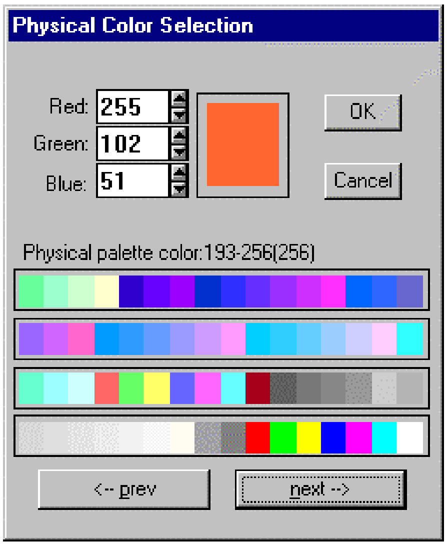
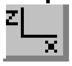


Figure 5.8: Color Palette dialog box

To change on of the colors, select "Palette colors" and click on one of the colors. In the Color Palette dialog box the RGB values can be edited or for a 256 color video driver one of the Physical palette colors can be chosen.

Display...



The Display menu item or the corresponding icon opens for single spectra the Single view dialog box.

Here the graphic display mode of single spectra can be chosen. The '**Type**' combo box gives a choice between dot, histogram, spline **I** and line. The '**Symbol**' combo box gives a choice between None, Circle, Triangle down, Triangle up, Cross, Snow-flake and Diamond. The symbols can be filled by checking **Fill**, error bars can be displayed by checking **Error Bar**.

'Dot' means that each spectra point is shown as a small rectangle or the specified symbol, the size can be adjusted with the **size** combo box. 'Histogram' is the usual display with horizontal and vertical lines, 'spline I' means linear interpolation between the points, and 'line' means vertical lines from the ground to each spectra point.

If the displayed spectra range contains more channels as pixel columns are available in the video graphic display, usually only the maximum value of the channels falling into that pixel columns is displayed. But it can also explicitly specified by marking the checkboxes „**Max Pixel**“, „**Mean Pixel**“ or „**Min Pixel**“ which value will be displayed. It is also possible to display all three possible values in different colors that can be chosen in the colors dialog. For the „Mean Pixel“ a Threshold value can be entered; channel contents below this value then aren't taken into account for the mean value calculation.

It is possible to change to a two dimensional view of the spectrum by specifying the x Dimension and clicking the button ">> MAP".

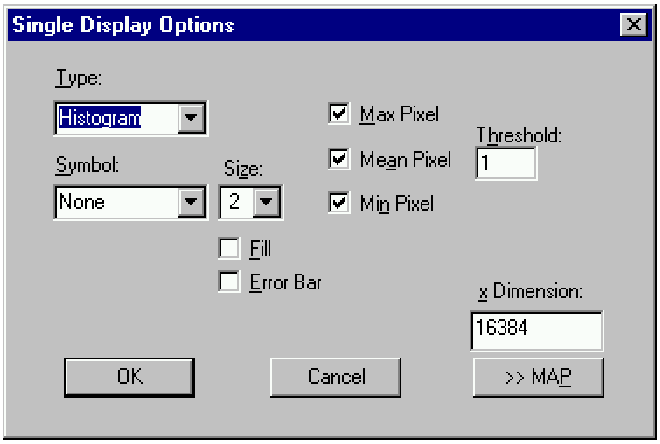


Figure 5.9: Single View dialog box

For MAP displays the Display Options dialog is changed and allows a choice between four Graphic types: **bitmap dot**, **vector dot**, **bitmap contour** and **vector contour**. Bitmap Dot is recommended as a standard, because it makes a good and fast display. Vector Contour paints colored contour lines. To calculate the lines takes a lot of time and causes the mouse pointer changing to an hourglass. But it gives very impressive colored pictures suited especially for presentation and when looking carefully at spectra details.

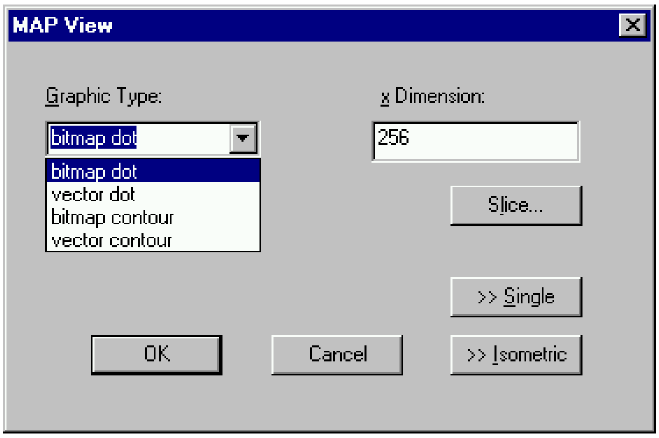


Figure 5.10: MAP View dialog box

Clicking the Slice button allows to create new single spectra displays, showing a slice in a 2D spectra. The Slice dialog box is displayed.

Select “x=const” or “y=const” for the slice direction, and the coordinate. Clicking the "create" button creates the new display window. In the title bar of the new window the name of the 2D spectra and the slice coordinate is shown.

The slice position can be changed using the scroll bar in the Slice dialog, or by entering the value in the edit fileld and pressing the button which is labeled “Set” after creation of the slice view.

The Slice dialog can be closed by clicking its close field. Created slice spectra displays remain visible and their coordinates can be changed later using the Slice dialog again. The position of the Slice dialog with respect to the MCDWIN main window can be saved in the MCDWIN.CNF file. Rectangular ROIs are visible in the slice spectra display and can be created here.

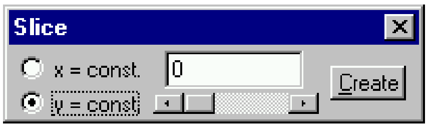


Figure 5.11: Slice dialog box

From the MAP View dialog it is possible to change to Single view by clicking ">> Single" or change to Isometric View by clicking ">> Isometric".

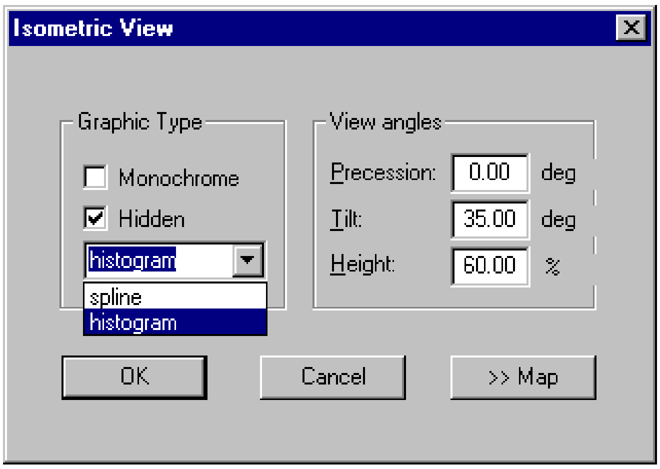
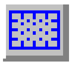


Figure 5.12: Isometric View dialog box

In isometric mode several single spectra are drawn behind each other. The **Precession** angle around the vertical axis can be chosen in multiples of 90 degrees. The **Tilt** angle is between the x and y axis and can be chosen between 15 and 89 degrees. The **Height** specifies the percentage of the z-axis length respective to the whole drawing, it can be entered between 0 and 99. With **hidden** it can be specified whether the hidden parts are not drawn. If "Monochrome" is checked, the spectra are painted monochrome, otherwise in color.

Axis...



By the Axis... menu item or the respective icon, the Axis Parameters dialog box is opened.

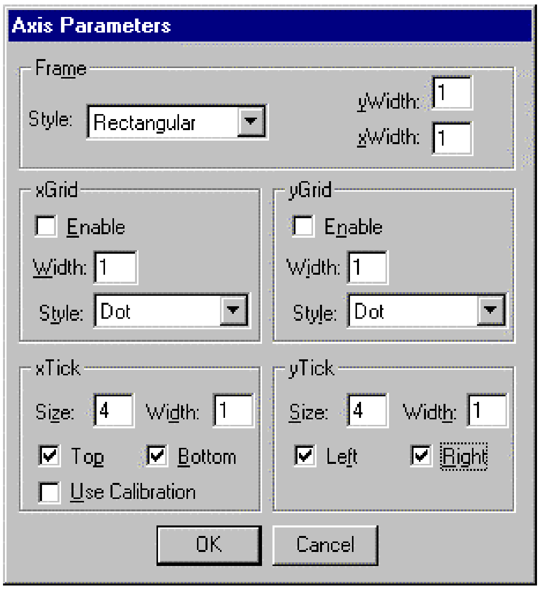


Figure 5.13: Axis Parameter dialog box

It provides many choices for the axis of a display. The frame can be rectangular or L-shape, the frame thickness can be adjusted (xWidth, yWidth). A grid for x and y can be enabled, the style can be chosen between Solid, Dash, DashDot and DashDotDot. Ticks on each of the four frame borders can be enabled, the tick length and thickness can be chosen. The style of the axis labeling depends on enabled ticks at the bottom respective left side: If no ticks are enabled there, only the lowest and highest values are displayed at the axis, otherwise the ticks are labeled.

Scaling...

The Scaling menu item or the corresponding icon opens the Scale Parameters dialog box.

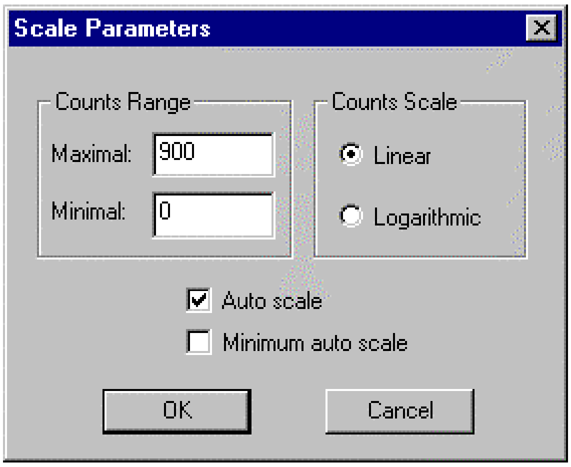
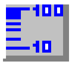


Figure 5.14: Scale Parameters dialog box

It allows to change the ranges and attributes of a Spectrum axis. By setting the Auto scaling mode, the MCDWIN will automatically recalculate the y-axe's maximum value for the visible Spectrum region only. To keep the same height of the visible region for a longer time, deselect the Auto scaling mode. Then with the scroll bar thumb one can quickly change the visible region scale, otherwise the scale will be changed automatically. The Minimum auto scale mode helps to display weak structures on a large background.

Lin / Log scale



Chose between Linear or Logarithmic scaling. All options have effect only on the active Display.

Calibration...

Using the Calibration menu item or the corresponding icon opens the Calibration dialog box.

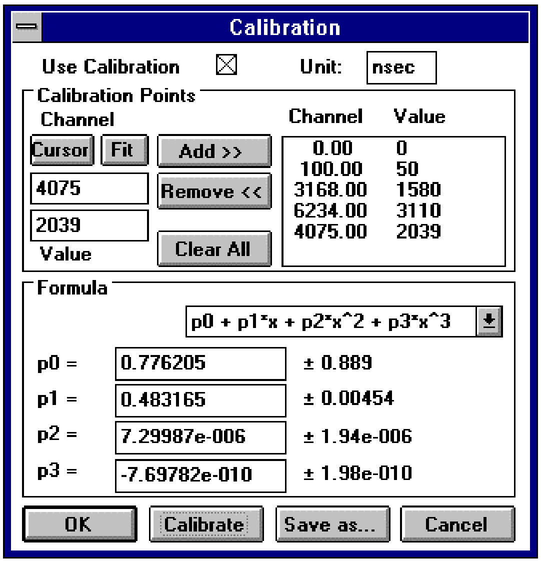
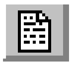


Figure 5.15: Calibration dialog box

Make a choice of several calibration formulas. Enter some cursor positions and the corresponding values (e.g. energy, mass etc.), click on Add and then on Calibrate. The obtained coefficients can be inspected together with the statistical error, or they can be changed and entered by hand. If 'use calibration' is enabled, the calibrated values are displayed together with the channel position of the cursor.

Comments...



Up to 13 comment lines with each 60 characters can be entered using the Comments dialog box. The content of these lines is saved in the data header file. The first line automatically contains the time and date when a measurement was started. The titles of each line can be changed by editing the file COMMENT.TXT.

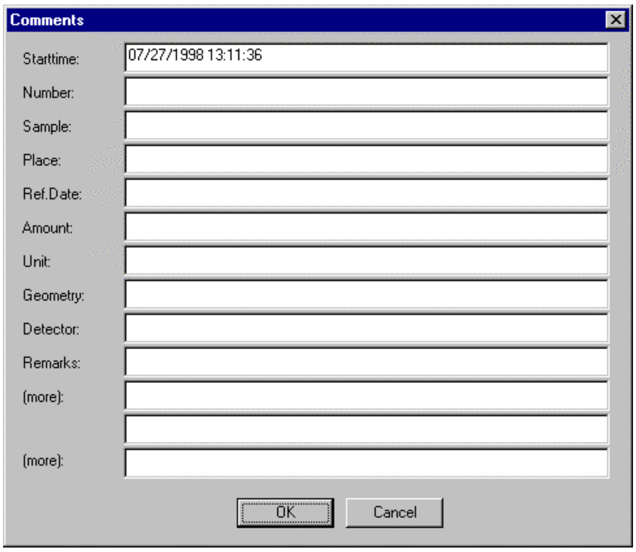


Figure 5.16: Comments dialog box

Range, Preset...

This dialog box allows to make all MCA-3 settings (ref. chapter 4.1.4).

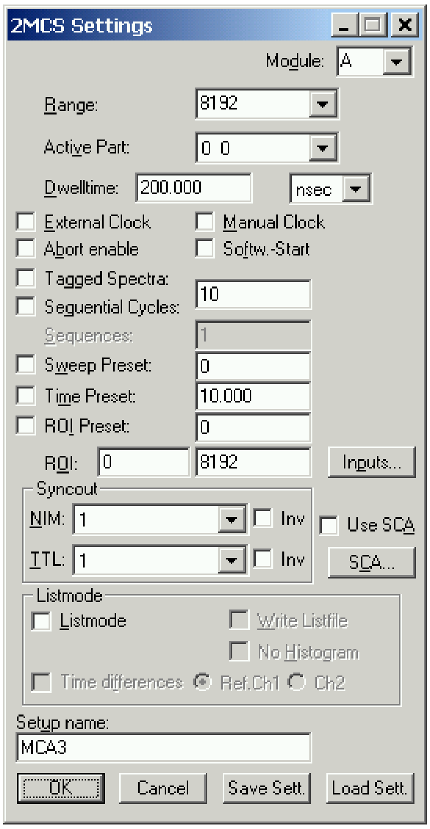
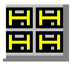


Figure 5.17: MCA-3 Settings dialog box

Data...



The Data dialog box allows to perform all the MCA-3 data operations (ref. chapter 4.1.3).

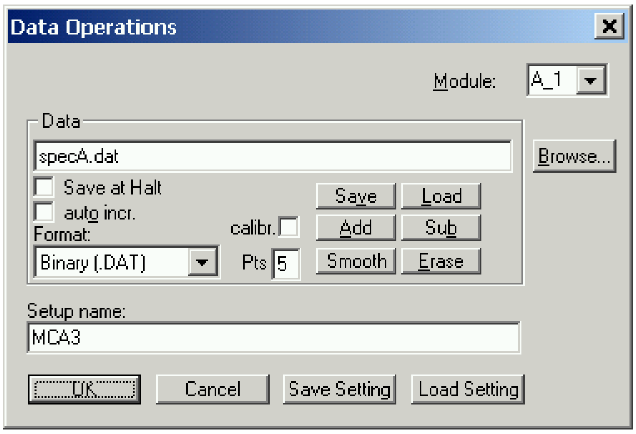
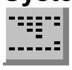


Figure 5.18: Data Operations dialog box

System...



The System Definition dialog box allows to make all the respective MCA-3 settings (See chapter 4.1.5).

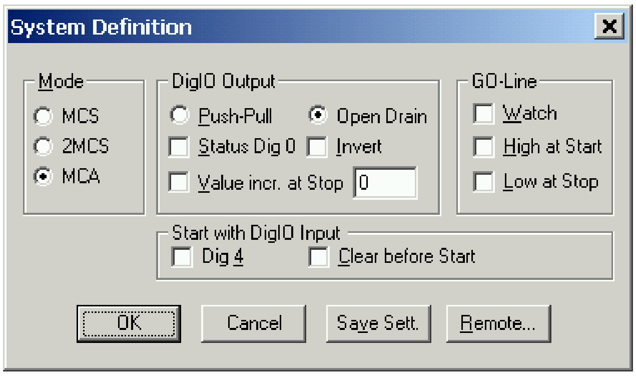
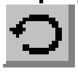


Figure 5.19: System Definition dialog box

Replay...



The menu item Options – Replay... opens the Replay settings dialog (ref. chapter 4.1.3).

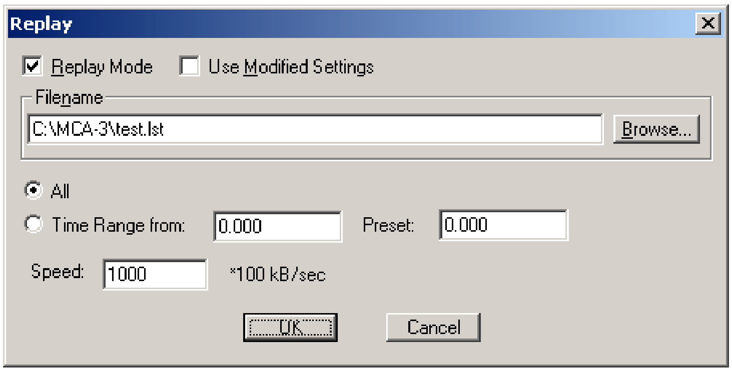
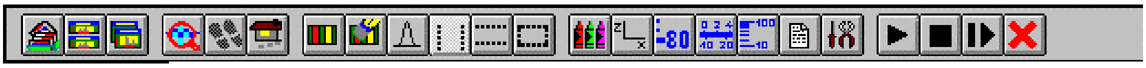


Figure 5.20: Replay dialog box

Tool Bar...



Selecting the Tool Bar Menu item opens the Tool Bar Dialog Box. It allows to arrange the icons in the Tool Bar.

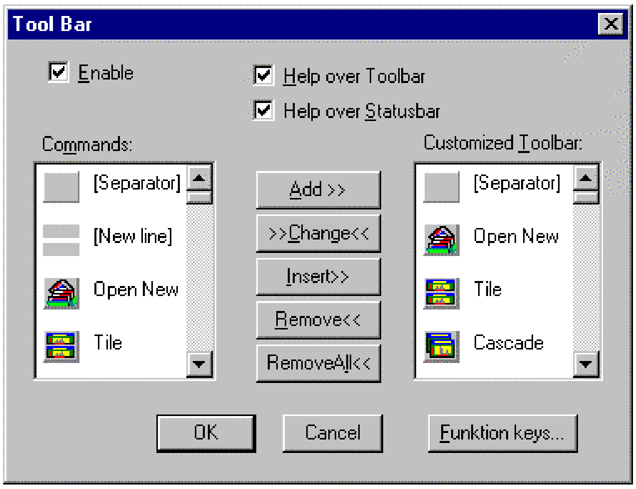


Figure 5.21: Tool Bar dialog box

If it is enabled, an array of icons in the MCDWIN Menu is shown. Clicking the left mouse button with the cursor positioned on an icon, the user can perform a corresponding MCDWIN Menu command very quick.

It is also possible to include icons for free programmable function keys F1...F12 into the Toolbar. The function keys can be programmed in the Function keys dialog. It can be accessed either by clicking the "Function keys..." button or directly from the options menu.

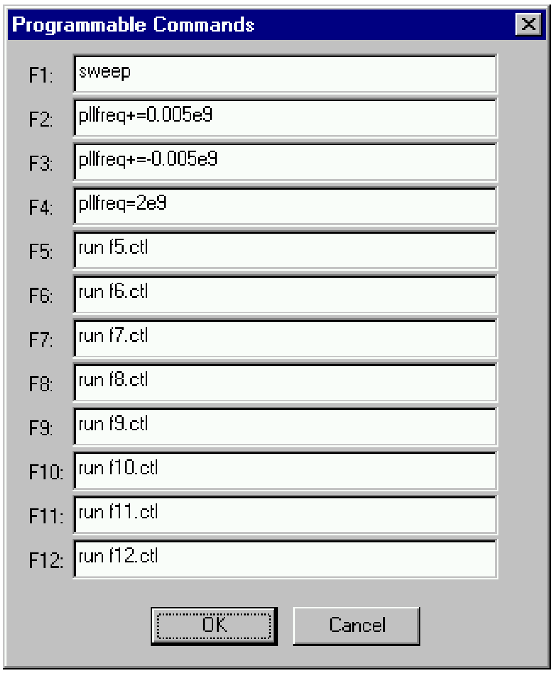


Figure 5.22: Function keys dialog box

The functions can be executed by clicking the corresponding icon in the toolbar or by the corresponding function key on the keyboard simultaneously with the CTRL key. The MCDWIN window must be the active on the desktop and have the focus.

Status bar

With this menu item the Status bar at the bottom of the MCDWIN main window can be switched on or off. A corresponding checkmark shows if it is active or not. The Status bar usually shows if an acquisition is running. When the left mouse button is held down while the mouse cursor is on a toolbar icon, it displays a short help message what the toolbar icon does.

Status window

The same way it is possible to hide or show the status window at the left side of the MCDWIN main window.

Save

Stores all parameters defined in the Options menu to the MCDWIN.CNF config file.

Save As...

Stores all parameters defined in the Options menu to a user defined config file.

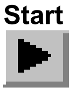
Retrieve...

Loads a new configuration.

5.5. Action Menu

The Action Menu or corresponding toolbar icons contain the commands to start, stop, continue and erase a measurement. If more than one system is formed, also more actions menus are available, otherwise they are grayed or disabled.

Start



The Start toolbar button erases the histogram data and starts a new measurement.

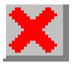
Halt

The Halt toolbar button stops a measurement.

Continue

The Continue toolbar button continues a measurement.

Erase



The Erase toolbar button clears the histogram data.

6. Programming and Software Options

The MCA-3 can be controlled by user-written programs using the DLL software interface with example programs for Visual Basic, LabVIEW and C that is available as an option. Furthermore, LINUX software is available as an option containing a driver, library and console test program. A Windows software similar to the LINUX package that runs without the server using a stand-alone DLL is also available on demand for customers who own one of the two available library packages.

Auto-Correlation: an optional available expansion of the Server program allows to acquire data into a two dimensional array M(i,j). The channel (i,j) is incremented when in a single sweep the channel i and i+j has an event. The two dimensional MAP can be viewed in MCDWIN even during the acquisition. Use the display options and switch to MAP and later to ISOMETRIC.

7. Appendix

7.1. Performance Characteristics

7.1.1. General

Memory:	....512k x 32 bit
Basic operating modes:	.... Multiscaling ....Pulse Height / Multichannel Analysis
Memory modes:	(histogramming) .... Cardmode (FIFO) ....Listmode
Memory segments:	programmable in binary steps....64, 128, 256, … , 512k
Memory offset:	programmable in steps of 64.... ....64, 128, 192, … , (512k – 64) = 524,224
Memory tags:	most significant memory address bits .... DIG I/O 0…7
Timer / counter:	2x presettable: .... 32 bit

7.1.2. External ADC Port

Input data rate:	(depends on source device) ....max. 5 x 10^6 events/s
Live / Real time resolution:	.... 1 ms
Timer preset range:	....2 ms … > 1193 h

7.1.3. Internal ADC

Conversion time:	....370 … 700 ns ....typ. 500 ns
Deadtime:	.... = Time-to-peak + MAX(Conversion time; Signal fall time)
Data throughput:	.... Deadtime^{-1}
Live / Real time resolution:	.... 1 ms
Timer preset range:	....2 ms … > 1193 h
SCA pulsewidth:	....typ. 300 ns
Peak to sampling delay:	....TBD
Resolution:	8k range, 500 ns flat top pulses ....typ. 2.1 chs
Integral non-linearity:	.... typ. ±0.25%
Differential linearity:	including effects from integral non-linearity.... typ. ±1.1% for 99% of usable channels.... typ. ±0.8%
Peak shift:	upto 700 kcps, SVA mode....TBD upto 1000 kcps, SVA mode....TBD
Unused channels:	....TBD

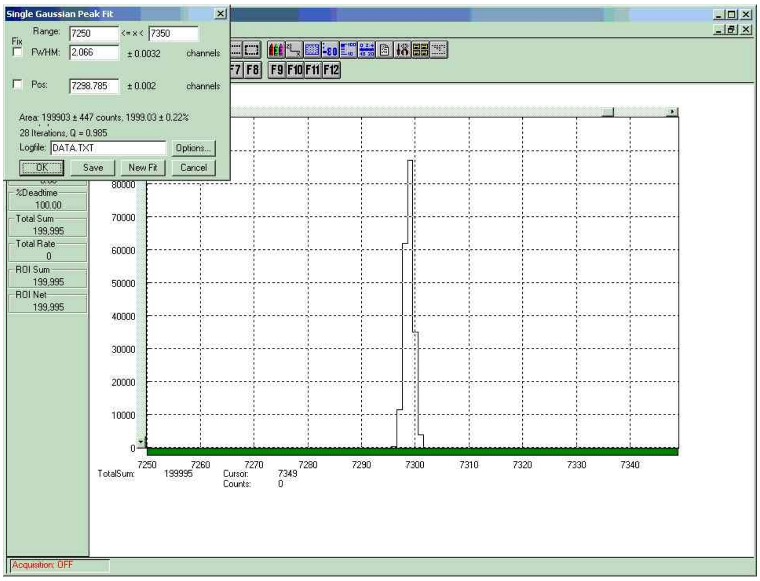


Figure 7.1: Internal ADC Peak Resolution

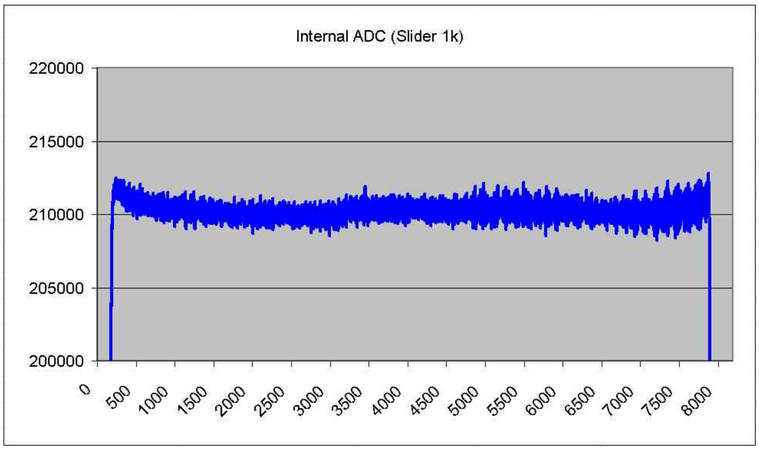


Figure 7.2: Internal ADC Linearity Measurement

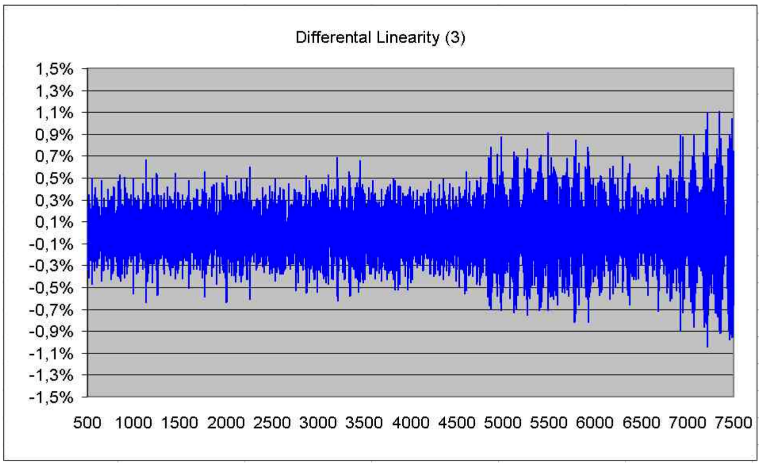


Figure 7.3: Internal ADC Differential Linearity Plot

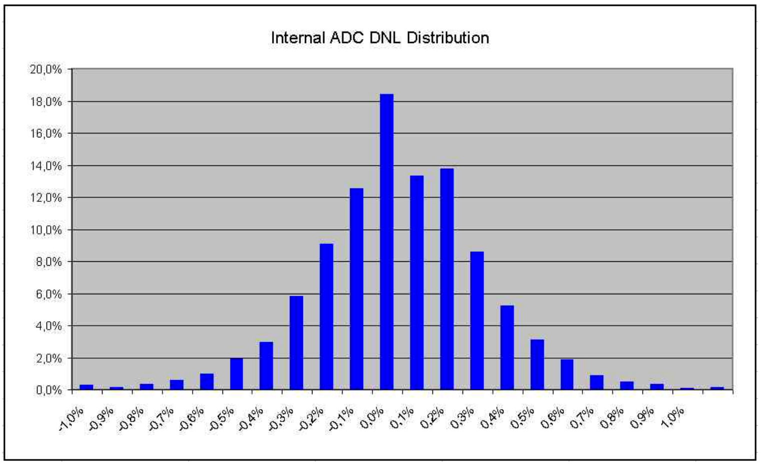


Figure 7.4: Internal ADC Differential Linearity Distribution

GATE to peak:	t_{setup} ....(5 + min. delay) TBD t_{hold} ....(5 + max. delay) TBD
SAMPLE delay:	SVA mode: ....TBD
TAG :	t_{setup} to begin-of-pulse (deadtime rising) .... ≥ -10 ns t_{hold} after begin-of-pulse (deadtime rising) .... ≥ 60 ns

7.1.4. Multiscaler

Dwell time modes:	.................................internal (auto), external, manual (software)
Internal dwell time:	1 MCS input channel ....min. 100 ns 2 MCS input channel ....min. 200 ns .... max. > 107 s in steps of: .... 25 ns

External channel advance:	edge: ....rising / falling selectable period (internally asured): ....min. 150 ns uncertainty: .... ± 12.5 ns
Start / trigger delay:	.... 25 ns ± 12.5 ns
Deadtime:	End of sweep: .... none between time bins: .... max. ± 0.15 ns .... typ. ± 0.07 ns
Sweep repetition time:	.... ≥ (Range + 25 ns)
ABORT reaction time:	.... at end of time bin
ABORT pulse width:	.... > dwell time + 10 ns
Differential linearity:	.... typ. ±0.07%
Sweep counter:	.... 32 bit
Sweep counter preset range:	.... 1 … 2^{32}
TAG to START:	t_{setup} (DIG I/O_i).................... 65 ns t_{hold} (DIG I/O_i).................... 65 ns

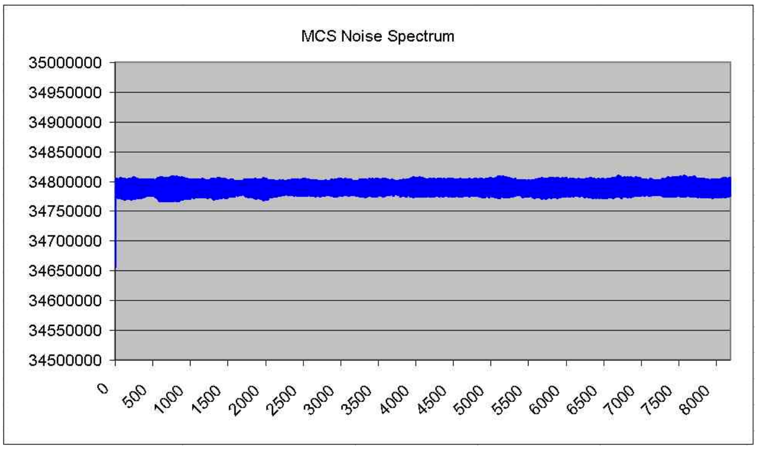


Figure 7.5: MCS Linearity Measurement

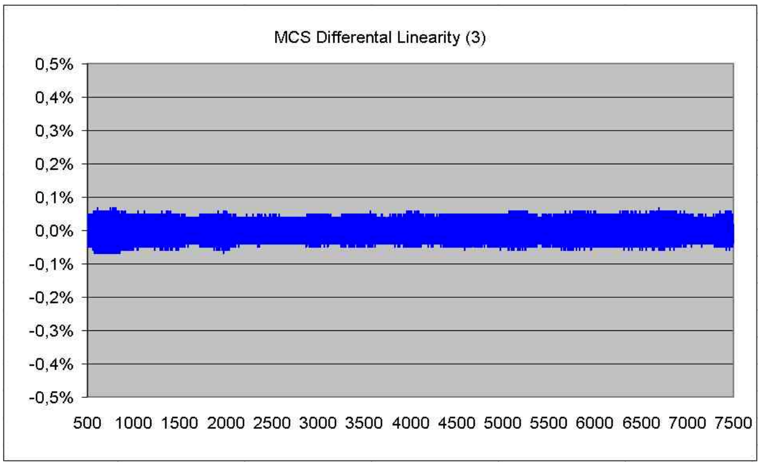


Figure 7.6: MCS Differential Linearity Plot

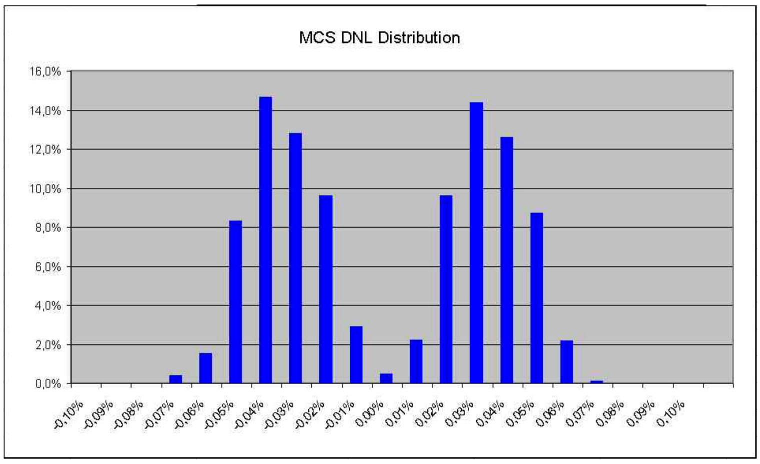


Figure 7.7: MCS Differential Linearity Distribution

7.1.5. Digital-to-Analog Converters

DAC 1 (RAMP):

Operating modes:		fixed voltage
	in MCS modes only:	x-ramp
X-Ramp modes:		triangle / sawtooth
Ramp start value:	in steps of 2.44 mV	0 … +10V
Ramp step size:	in steps of 2.44 mV	0 … +10V
Ramp stepping time:	in steps of bin width	(1 … 255) x bin width
		min. 200 ns
Ramp steps:	until change of slope (triangle)	1 … 4096
	until restart at start value (sawtooth)	1 … 4095
Relative accuracy:	INL	± 1 LSB
Differential nonlinearity:	DNL	± 1 LSB
Offset error:		± 5 LSB
Full scale voltage error:		± 60 mV
Full scale voltage stability:		± 30 mV
Full scale temp coefficient:		± 0.3 mV/°C
Output slew rate:		≥ 2 V/µs
Settling time:	to ± ½ LSB for full scale change	5 µs
Digital glitch impulse:		≤ 30 nV s

DAC 2:

Relative accuracy:	INL	± 4 LSB, typ. ± 1 LSB
Differential nonlinearity:	DNL	± 1 LSB

Offset error:	....± 2.7 %/FS[1], typ. ± 0.5 %/FS	
Offset temp coefficient:	....± 300 μV/°C	
Full scale voltage error:	.... ± 3%/FS, typ. ± 1%/FS	
Full scale temp coefficient:	....± 110 ppm/°C, typ. ± 20 ppm/°C	
Output slew rate:	....≥ 1 V/μs	
Settling time:	for half scale change .... 5 μs	

7.1.6. Timebase

Reference oscillator: Nominal frequency: ....40.000 MHz
Initial accuracy (25°C): ....≤ ± 50 ppm
Frequency stability:
in operating temperature range: ....≤ ± 100 ppm

7.1.7. Data Throughput

Listmode data rate to the PC[2]: .... TBD sec^{-1}

---

1 Full Scale

2 depends largely on the computer used

7.2. Specification

7.2.1. Absolute Maximum Ratings

Input voltage:

analog signal input:	(PC power ON)	-10 to +20 V
	(PC power OFF)	-10 to +12 V
any discriminator input:		± (VCC[1] + 1.25 V)
any digital I/O port:		-0.5 to VCC + 0.5 V

DC current:

analog signal input:	(PC power ON)	-17 to +25 mA
	(PC power OFF)	± 17 mA
any discriminator input:		± 200 mA
any digital I/O port:		± 20 mA
any DAC output:		± 25 mA

7.2.2. Recommended Operating Conditions

Supply voltage:	(from PC power supply)	+5 V, +12 V
Temperature range:		0 to 50°C
GO Line load:		min. 1 kΩ to VCC
	or	min. 2 kΩ to GND

7.2.3. Power Requirements

Supply voltage:	VCC:	+5 V ± 0.25 V
		+12 V ± 0.6 V
Supply current:	+5 V:	typ. 0.55 A, max. 0.70 A
	+12 V:	typ. 0.56 A, max. 0.75 A

7.2.4. Connectors

Analog (ADC) Input

Location:		mounting bracket
Connector:		LEMOSA series 00 NIM-CAMAC
Input impedance:		1 kΩ
Input filter:	(software selectable on/off)	lowpass fg = 1800 kHz
Amplitude:		+0.025 … +10 V

[1] VCC nominally is +5V when power is ON

Pulse form:		Gaussian shaped
Signal shaping time:		250 ns to 25 µs
Threshold voltage:	(power-up default = 21 mV)	0 … +2.5 V
		in steps of 0.61 mV
Lower Level Discriminator:	(power-up default = 61 mV)	0 … +10 V
		in steps of 2.44 mV
Upper Level Discriminator:	(power-up default = 10.0 V)	0 … +10 V
		in steps of 2.44 mV

±5 V Discriminator Inputs

Location:		mounting bracket
Connector:		LEMOSA series 00 NIM-CAMAC
Impedance:	(software selectable)	1 kΩ / 50 Ω
Input voltage range:		± 5.0 V
Threshold voltage:	(power-up default = 0.0 V)	±2.5 V
		in steps of 1.22 mV
Sensitivity:		typ. < 10 mVpp
Bandwidth:	(GATE / ADV / ABORT)	$\leq$ 100 MHz
	(TRG / SAMPLE)	$\leq$ 100 MHz
	(COUNT 1)	typ. $\geq$ 200 MHz
	(COUNT 2)	typ. $\geq$ 200 MHz
Pulse width high/low:	(GATE / ADV / ABORT)	typ. $\geq$ 5 ns
	(TRG / SAMPLE)	typ. $\geq$ 5 ns
	(COUNT 1)	typ. $\geq$ 2 ns
	(COUNT 2)	typ. $\geq$ 2 ns
Slew rate requirement:		> 0.5 V/µs
Logical polarity / slope:		software selectable

Fast-NIM SYNC_1 Output

Location:		mounting bracket
Connector:		LEMOSA series 00 NIM-CAMAC
Impedance:		50 Ω
Output HIGH voltage:	(50 Ω load)	typ. –0.0 V
Output LOW voltage:	(50 Ω load)	typ. –0.8 V
Output short circuit current:	V_{OUT} = GND	typ. 32 mA
Logical polarity:		software selectable

SYNC_2 Output

Location:		ref. FEATURE (multi) I/O port connector
Output HIGH voltage:	(at $POUT_i$) $I_{OutHIGH}$ = -12mA	min. 2.4 V
Output LOW voltage:	(at $POUT_i$) I_{OutLOW} = 16mA	max. 0.45 V
Output short circuit current:	V_{OUT} = GND ($R_{PULL} = \infty$, $R_{I/O} = 0\Omega$)	-15 to –180 mA
	V_{OUT} = 3.3 V ($R_{PULL} = \infty$, $R_{I/O} = 0\Omega$)	40 to 210 mA
Logical polarity:		software selectable

DAC 1 Output

Location:	.... ref. FEATURE (multi) I/O port connector
Output voltage range:	.... 0 … +10V
Resolution:	.... 12 bit @ 2.44 mV

DAC 2 Output

Location:	.... ref. FEATURE (multi) I/O port connector
Output voltage range:	....$\pm$10 V
Resolution:	.... 12 bit @ 4.88 mV

Digital I/O 0…7

Location:	.... ref. FEATURE (multi) I/O port connector
R PULL :	(default) ....1.0 kΩ
R I/O:	(default) .... 22 Ω
Input HIGH voltage:	(at PIN_i, ref Figure 3.11.)[1] .... min. 2.0 V
Input LOW voltage:	(at PIN_i) .... max. 1.2 V
Output HIGH voltage:	(at $POUT_i$) $I_{OutHIGH}$ = -16mA .... min. 2.4 V
Output LOW voltage:	(at $POUT_i$) I_{OutLOW} = 15mA .... max. 0.45 V
Output short circuit current[2]:	V_{OUT} = GND ($R_{PULL} = \infty$, $R_{I/O} = 0\Omega$) .... -15 to –180 mA V_{OUT} = 3.3 V ($R_{PULL} = \infty$, $R_{I/O} = 0\Omega$) ....40 to 210 mA

GO-Line

Location:	....PCI board .... ref. multi I/O port connector
Connector:	2 pin header on PCI board: ....LUMBERG 2,5 MSFW 2(MBX) suitable socket connector: .... LUMBERG 2,5 MBX 2
Line Type :	....open drain / wired-AND
Pull-Resistors:	....100kΩ to VCC
Input HIGH voltage:	.... min. 2.0 V
Input LOW voltage:	.... max. 1.2 V
Output LOW voltage:	I_{OutLOW} = 15mA .... max. 0.8 V

7.2.5. Physical

PCI long board (ISA assembly, 5V)

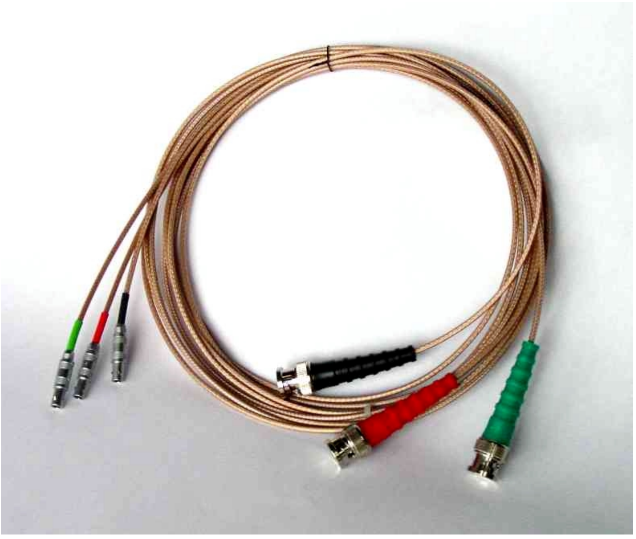
Size:	(incl. retainer) .... 341 x 107 mm
Weight:	(board alone)..≈ TBD g

[1] Note: input and output voltages are measured at the internal logic pads not at the external connectors. Thus, the corresponding pull and series resistors must be considered to get the external voltages

[2] Only one output at a time. Duration should not exceed 30 seconds.

7.3. Accessories

LEMO – BNC adapter cables



ADC port connector cable

7.4. Trouble Shooting

- System hangs on power-up: Take care that the board is well seated in the PCI connector. Push it towards the bracket to ensure proper connections.
- PCI device is not properly detected: Push the board in the PCI slot towards the bracket to ensure proper connections.
- Error message **"MCA-3 not found or FASTMPA device driver not installed!"** at the first start of the software:
  Maybe you did not install the device drivers. If Windows 9x/2000/XP is installed with the MCA-3 board plugged in, a wrong device driver for a "general PCI communication device" may be installed. Check it using the Device manager, remove the wrong driver and install the correct driver from the WDMDRIV directory on the diskette. On Windows NT change to the subdirectory \ntdriver on your MCA-3 installation disk and run install.bat to install the driver for the MCA-3 board.
- Error message **" Communication problem with board! Resource conflict?"** at the first start of the software: Try another PCI slot and remove all other PCI cards that are not urgently used for the moment. Some computers (for example most DELL computers) have a problem to recognize a second I/O port range reserved by a PCI card like the MCA-3 using the AMCC S5933/S5935 PCI interface chip. In this case please use another computer.

7.5. Frequently Asked Questions

7.5.1. Tag bits

Q: I have a question regarding the P7882. I want to use the D I/O bits to tag three different spectra, but I can't find in the manual how to assign the DIO bits to be used as TAG bits. I want to integrate in a cyclical manner, these 3 sources. I understand they will be shifted in memory and so forth.

I have two TTL lines that indicates which input is 'active', i.e. TTL 1 & 2 low = source 1. TTL 1 high and TTL 2 low = source 2. TTL 1 low and TTL 2 high = source 3. So, again, how do I programatically set which D I/O are to be used as TAG bits? and how do I achieve the necessary shift in terms of bins, if say, I want each source to have 8k bins?

A: Selecting [Options > Range, Presets ...] will open the MCS Settings dialog window. Check the box to the left of "Tagged Spectra"; in the box to the right of that enter "3" (the default is 128). Set the "Range:" to 8192.

Using 13-bits (8k) of time resolution would leave the remaining most significant 6 bits available for tagging out of the total of 19 bits in 1-input MCS mode (5 bits would be available in 2-input MCS mode as the MSB is used to identify the input). According to sections 3.3.4 and 3.5.6 in the manual, Digital I/O bit 0 is used to address bit 11; since bit 11 and bit 12 will be use in an 8k sweep length (13-bits), the next available bits for tagging would correspond to Digital I/O bit 2 and Digital I/O bit 3 (bit 13 and bit 14). Your TTL signal should be connected to these D I/O's (your LSB to D I/O bit 2).

Q: Thanks for your fast reply, That cleared things up. This leads me to think that in order to create the possibility of using the most bins I should set my two TAG bits on bit 16 and 17 (DIO 5 and 6). Is it correct that I could still use less bins in the three spectra tag mode, and I could swap between 1 or 2 input channels, without having to change my DIO connector cabling? or is it something I'm not getting here?

What I'm after is a configuration that will allow the most flexible setup, without rewiring the two TTL DIO cables.

A: You could use DIO 5 and 6 only and not rewire the DIO cabling but it will require adjustment of the "Tagged Spectra:" setting from "3" if you are to change the "Range:" setting to something other than 65k. The reason for this is the way the tag bits are allocated relative to the MSB of the sweep range; a corresponding number of bits (more significant) to MSB of the sweep range are used as the tag bits. So for two tag bit usage and an 8k sweep range (bits 0-12 used), the next two bits (bits 13 and 14, corresponding to DIO 2 and 3) get used as the tag bits; for a sweep range of 32k (bits 0-14 used) bits 15 and 16 (DIO 4 and 5) get used; and for a the sweep range of 65k (bits 0-15 used), bits 16 and 17 (DIO 5 and 6) get used.

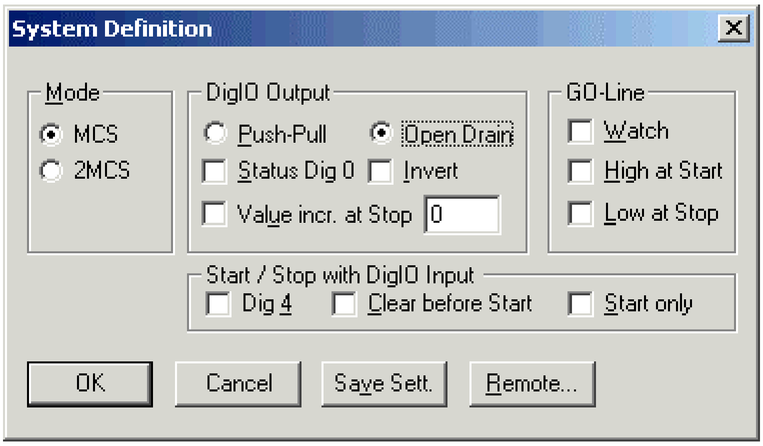
If you were to use bit 16 and 17 (DIO 5 and 6) and the 65k range, the tag bit patterns of 00, 01 and 10 (DIO6, DIO5) will be produced corresponding to the Y-axis levels of 0, 1 and 2 (your source 1, 2 and 3, respectively). If you were to only change the sweep range to 32k ("Tagged Spectra:" remains at "3"), bits 15 and 16 (DIO 4 and 5) get used as the tag bits and you will only be toggling bit 16 (bit 15 is pulled-up if no connection is made to it). The resulting tag bit patterns of 01, 11, and 01 (DIO5, DIO4) is produced corresponding to 1, 3, and 1 on the Y-axis, which of course is not what you want (source 1 and 3 get combined and source 2 not displayed). Likewise if you change the sweep range to 8k, the resulting tag bit pattern of 11, 11, 11 (DIO 2, DIO3 - bits 13 and 14 pulled-up with no connections) is produced with all sources combined in one spectra (Y-axis level of 3 and also not displayed).

To overcome this error when changing sweep length ranges and using only bit 16 and 17 (DIO 5 and 6), the "Tagged Spectra:" setting needs to correspondingly change to accommodate the differing number bits used while including bits 16 and 17 in the make up of the tag bits. For the sweep range of 32k, bits 15, 16 and 17 (DIO 4, 5 and 6) will need to be used as the tag bits. With toggling of bits 16 and 17 and bit 15 held high, the resulting tag bit patterns produced would be 001, 011, and 101 (DIO 6, DIO5, DIO4) corresponding to Y-axis levels of 1, 3 and 5.; the minimum "Tagged Spectra:" setting needed would be 5. For the sweep range of 8k, bits 13 through 17 (DIO2 through DIO6) will need to be used as the tag bits. The resulting tag bit patterns produced would be 00111, 01111, and 10111 (DIO 6, DIO5, DIO4, DIO3, DIO2) corresponding to Y-axis levels of 7, 15 and 23.; the minimum "Tagged Spectra:" setting would be

23. The results of these settings is a 2D display with with 3 spectra corresponding to the 3 sources and a number of empty spectra (Y-axis levels).

The shortest sweep range setting using the tag bit feature is 2048 channels (2k), 11-bits, leaving a maximum of 8-bits for tags (256 total) in 1-input MCS mode and 7-bits for tags (128 total) in 2-input MCS mode. Similarly, you would need to set the "Tagged Spectra:" setting appropriately (7 tag bits used) to accommodate using the 2k range while using only DIO 5 and 6.

With this technique you won't need to rewire your cabling and only need to change software settings. Also I should mention that the "DigIO Output" setting should be changed to "Open Drain" in the System Definintion dialog window [Options > System...] to get the DIO's to work as stated as inputs.



7.6. Personal Notes