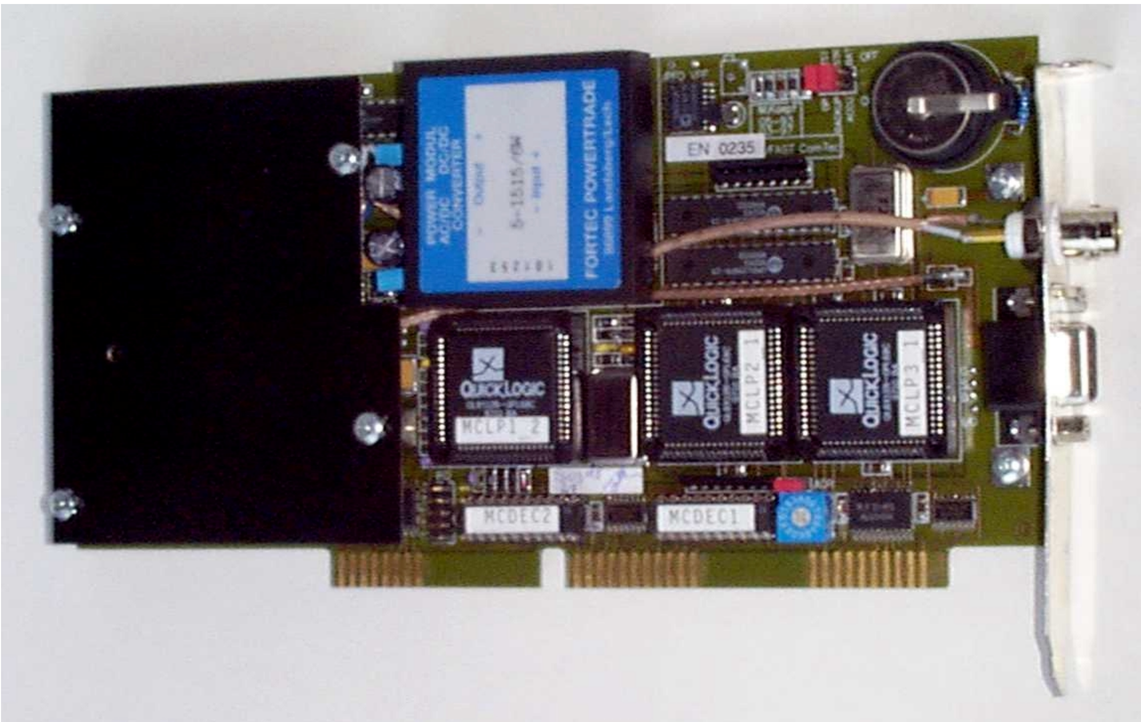
MCDLAP

ADC & Multichannel Data Processor

User Manual

Grünwalder Weg 28a, D-82041 Oberhaching
Tel ++49 (0)89 66518050; FAX ++49 (0)89 66518040
Germany



Version 2.85, October 7, 2004

Software Warranty

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. This license allows you to use this software on only one compatible computer at a time. 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.

Table of Contents

Table of Figures

1. Introduction

The MCDLAP is a complete highly sophisticated data acquisition system for IBM or compatible personal computers. It is based on a 2/3 length low profile PC-AT-card and therefore also ideal for most laptop computers. The usage of state-of-the-art integrated circuits and the sophisticated design lead to the excellent performance and stability previously reserved for standalone systems.

The MCDLAP card not only provides the user with a high resolution Pulse Height Analyzing (PHA) ADC but also with a complete Multichannel Data Processor (MCD). The ADC is a 16kchannel / 14bit Wilkinson type with a 100MHz clock rate. It includes Lower and Upper Level Discriminators (LLD, ULD) providing Single Channel Analyzer (SCA) capabilities for data reduction etc. The MCD section has a 16k x 32bit large ultra fast histogramming memory and hardware data processing providing high count rates and low data transfer time.

The MCDLAP card is particularly designed for use in nuclear- and x-ray spectroscopy but also ideal for other applications. It exhibits excellent integral- and differential linearity. The high conversion speed and the good channel profile make it an ideal choice for high resolution Germanium detectors.

The battery backup of the MCDLAP memory provides for no data loss on power failures. This is especially important in long-term experiments or when the experimentation time is limited. With the MCDLAP the user can continue his measurement after a power failure without losing any data.

The onboard Digital to Analog Converter (DAC) enables software control of external devices (e.g. High Voltage supplies) without added hardware.

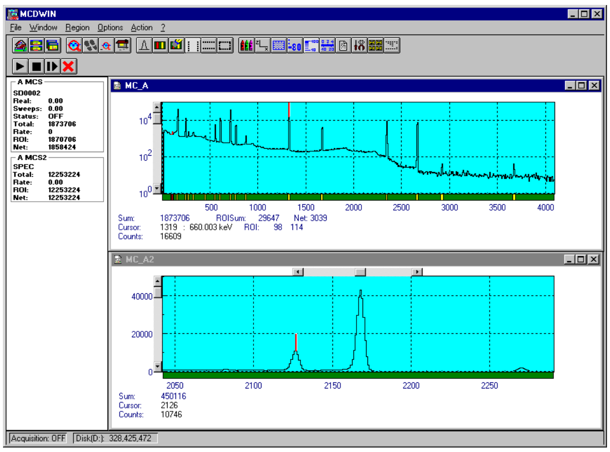
The onboard Digital I/O port can be used to remotely control data acquisition or to switch external devices on and off.

A GATE input allows coincidence / anticoincidence measurements.

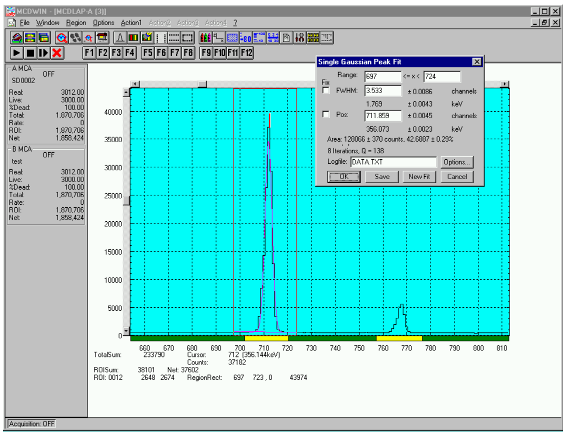
In addition the highly sophisticated operating software allows the user to easily control all the features of the MCDLAP data acquisition system. Since all the functions are fully under software control no jumpers or switches have to be set. This means once the MCDLAP card is installed in your computer no further hardware changes are necessary.

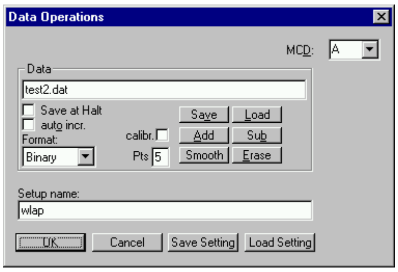
Once started the card operates fully independently. Thus the operating program can be exited and the computer resources are free for other tasks while data acquisition is in progress. When reentering the MCDLAP operating software you can view the measurement results.

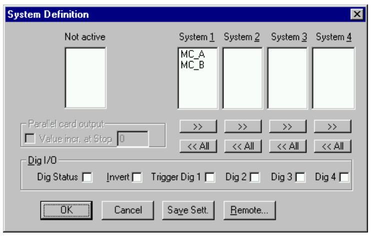
The sophisticated WINDOWS based control and analysis software MCDWIN ensures quick learning and easy usage.

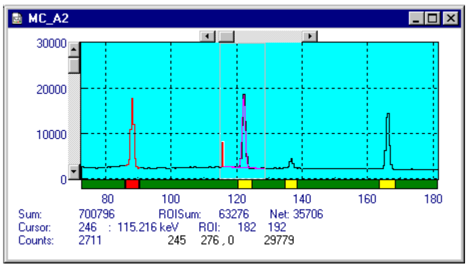


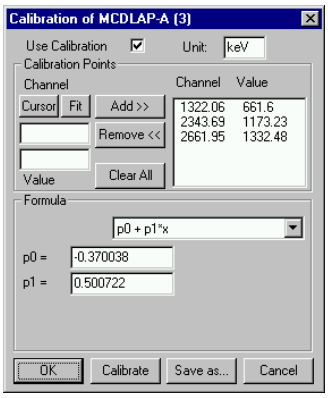
Some of MCDWIN's features are high resolution graphics displays with zoom, linear and logarithmic (auto)scaling, grids, ROIs, Gaussian fit, calibration using diverse formulas. and FWHM calculations. Macro generation using the powerfull command language allows task oriented batch processing and selfrunning experiments. An IAEA compatible software interface allows to directly use such analysis packages as GANAAS, QXAS, POSFIT or others.




Data Operations
MCD:
A
Data
test2.dat
Save at Halt
auto incr.
Format:
Binary
calibr.
Pts 5
Save
Load
Add
Sub
Smooth
Erase
Setup name:
wlap
OK
Cancel
Save Setting
Load Setting


System Definition
Not active
System 1
System 2
System 3
System 4
MC_A
MC_B
>>
<< All
Dig I/O
Dig Status
Invert
Trigger Dig 1
Dig 2
Dig 3
Dig 4
OK
Cancel
Save Sett.
Remote...


MC_A2
30000
20000
10000
0
80
100
120
140
160
180
Sum: 700796
ROISum: 63276
Net: 35706
Cursor: 246 : 115.216 keV
ROI: 182 192
Counts: 2711
245 276 , 0
29779


Calibration of MCDLAP-A (3)
Use Calibration
Unit: keV
Calibration Points
Channel
Channel Value
Cursor
Fit
Add >>
Remove <<
Clear All
Value
1322.06 661.6
2343.69 1173.23
2661.95 1332.48
Formula
p0 + p1*x
p0 = -0.370038
p1 = 0.500722
OK
Calibrate
Save as...
Cancel

2. Installation Procedure

2.1. Hard- and Software Requirements

The MCDLAP requires an IBM AT or compatible computer with a 386, 486, Pentium or higher processor and an available 16 bit slot. Several MCDLAP cards might be installed in your computer if you have enough available slots.

A PC with Microsoft Windows 3.1 / 95 / 98 / NT / 2000 installed is required for use of the supplied control and analysis software MCDWIN.

2.2. Hardware Installation

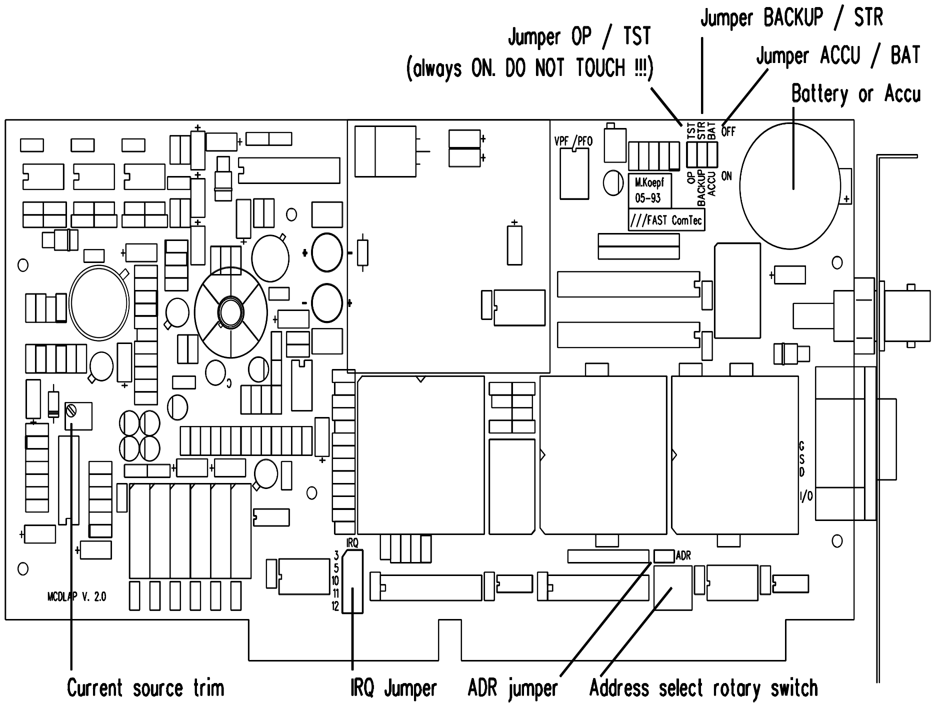


Figure 2.1: MCDLAP PC card

First you should locate an unused address in the I/O address space of your computer. The MCDLAP has a small rotary switch and a jumper labeled ADR (ref. Figure 2.1) that determines the base I/O address of the card. The MCDLAP occupies 8 I/O addresses starting at this base address. The supported base addresses and corresponding switch settings are:

Switch	Base Address [hex]	Switch	Base Address [hex]	Switch	Base Address [hex]	Switch	Base Address [hex]
0	200	4	240	8	300	C	340
1	210	5	250	9	310	D	350
2	220	6	260	A	320	E	360
3	230	7	270	B	330	F	370

Figure 2.2: Table of the base I/O addresses

The factory setting is 320_{hex} - an address commonly not used by other devices. With the ADR jumper OFF a value of 1000hex is added (i.e. with the switch set to A and the jumper removed the base address is 1320hex).

2.3. Software Installation

To install the MCDLAP software on your hard disk insert the MCDLAP disk into drive A. Log to drive A: by clicking from the explorer and start the installation program for Windows 9x/NT/2000/XP by double clicking the SETUP.EXE. For Windows 3.x start the INSTALL.BAT in the win3x directory from a DOS box.

A directory called C:\LAP (or C:\WLAP for Windows 3.x) is created on the hard disk and all MCDLAP and MCDWIN files are transferred to this directory. Drive C: is taken as default drive and the LAP working directory as default directory. It is not mandatory that the MCDLAP operating software is located in this directory, you may specify another directory during the installation.

For Windows NT/2000/XP it is necessary that you install the device driver FASTMCD.SYS from the WINNT\DRIVER directory. Run INSTALL.BAT if not already done from the setup program and restart the system.

Under Windows 9x/NT/2000/XP the Setup has installed an icon on the desktop. To start the software from the WINDOWS 3.x program manager install an icon for WLAP.EXE (server to control the hardware) in the WINDOWS program manager using the File - New... menu item. WLAP.EXE is the MCDLAP Hardware Server program. This program will automatically call the MCDWIN.EXE program when it is executed. The MCDLAP Server program controls the MCDLAP board but provides no graphics display capability by itself. By using the MCDWIN program, the user has complete control of the MCDLAP along with the MCDWIN display capabilities.

3. Hardware Description

3.1. Overview

The MCDLAP is a complete data acquisition system comprising a peak sensing analog-to-digital converter and a fast multichannel data processor. The inclusion of lower and upper level discriminators provide higher count rates by the discrimination of uninteresting events. A GATE input enables for coincidence measurements. The conversion gain is programmable: 256, 512, 1k, 2k, 4k, 8k and 16k channels resolution.

Due to the 100% background operation capability even a hardware reset of the computer does not affect data acquisition.

The high sophisticated design with state-of-the-art integrated circuits lead to the excellent performance, reliability and stability of the MCDLAP so far reserved to standalone devices. The usage of an onboard DC/DC converter (not the PC power supply for the analog section), a multiple DAC, 3 field programmable gate arrays, high speed cache RAMs etc. stand for the considerable amount of technology applied

3.2. ADC Section

The ADC is a Wilkinson type with a clock frequency of 100MHz. The input is connected to a fast peak detector and a lower threshold comparator. Every event that passes the lower threshold triggers the control logic. An ultra fast OP-amp charges a hold capacitor and when the maximum amplitude level is reached the storage capacitor holds this peak value. The control logic now checks if the peak value is inside the lower and upper level window (LLD and ULD setting). If the peak is outside the window region the event will be denied and the capacitor is discharged rapidly and the peak detector rearmed. If a valid - inside the discriminator window - event occured a conversion cycle is started and the input stage is disabled to prevent disturbance by further input signals. A 500ns pulse is output at SCA OUT and a constant current source now discharges the hold capacitor while a fast counter measures the required time to reach ZERO level. To move the zero point of a spectrum this ZERO level can be changed by about ±200mV but it is recommended to leave this to the calibration routine of the operating software. After zero crossing has occured the counter value is latched in the ADC output register and the converter is rearmed. The required conversion time is

conversion time = 1µs + (0.01µs * channel address)

and thus depends on the event amplitude level and the selected conversion gain.

The dead time of the converter is

dead time = t_{rise} + 1µs + MAX (0.01µs * channel address ; t_{decay})

where t_{rise} is the time the input signal needs to reach the maximum after threshold crossing and t_{decay} is the time the input signal needs to decay. The dead time is choosen this way since not only the conversion time affects the real dead time of the ADC but also the slopes of the input signal. This is true since further events must not trigger the converter before the old pulse has ended

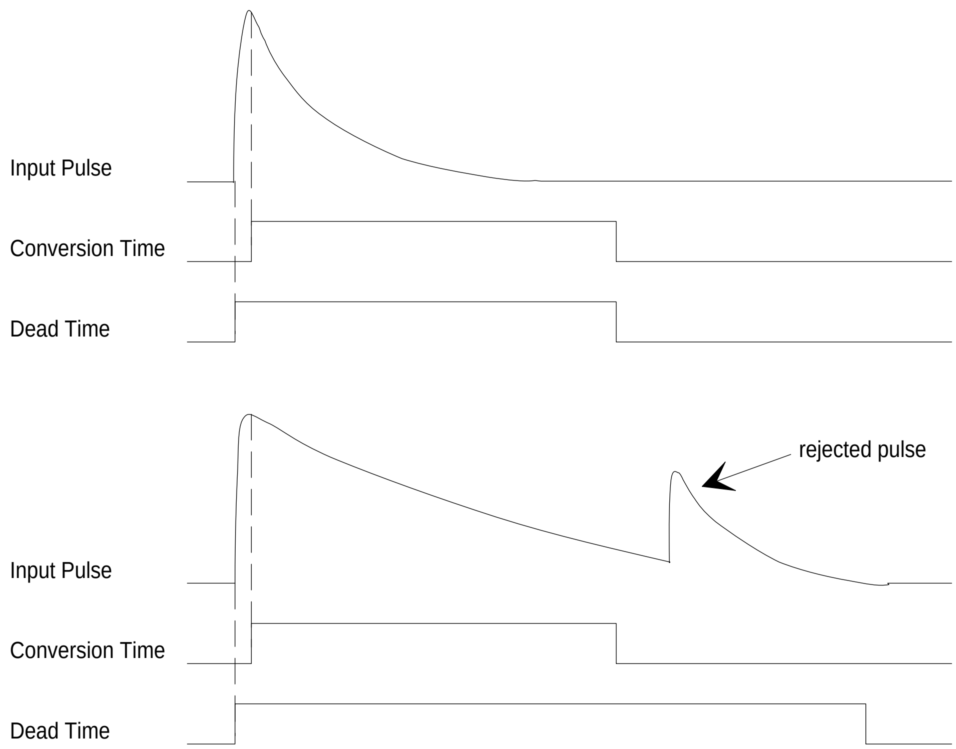


Figure 3.1: ADC dead time and conversion time in relation to pulse shapes (fast (upper) and slow (lower curve) decay)

A signal showing the dead time of the converter is provided at DEAD TIME OUT of the 9 pin SUB-D connector. Since there exists an output register for the converted value and the ultra high speed of the multichannel analyzer (MCA) no further dead time is added by the data storage and processing sequences in the MCA. This means that after completion of a conversion the ADC is ready for a new conversion cycle immediately.

To allow several spectra to be stored in memory in parallel a digital offset can be added to the conversion result. The offset can be added in multiples of 1024 channels thus providing writing to different memory locations. When adding an offset to the ADC result care must be taken that the conversion gain is not higher than the used offset stepwidth. This is particularly important for not to overwrite other spectra sections.

3.3. MCA Section

The MCA section of the MCDLAP card is high sophisticated data processing unit providing ultra fast data storage cycles. The average storage and processing time for one event is only 430ns much higher than the maximum data rate of the ADC. This ensures that no dead time is added by data storage or through data read operations while collection of a spectrum is in progress.

When valid ADC data are available the conversion result is used as address to the MCA memory and the corresponding memory contents is incremented. The 32kB x 16bit MCA memory is organized in 16k channels each 32bit wide thus providing 4G counts per channel before an overflow will occur. The meory is fully controllable by the software. Thus spectrum data can be written back to the memory and interrupted experiments continued.

The channels 0 and 1 are used to store real and live time information with a resolution of 1ms each. When the timer preset mode is enabled these two channels are observed an acquisition is stopped when the preset values are reached.

The battery backup of the MCDLAP ensures that no data losses occur even when a power failure happens. A watchdog continuously observes the power supply and when it would fade the MCDLAP will finish his progressing cycle and goto a reset state. This ensures that even a power loss would not confuse MCDLAP operations and the battery preserves the memory contents. Depending on the charge condition of the battery the data retain time can be several days.

3.4. Signal Connectors

All external signal connectors are placed on the mounting bracket of the card. There is an BNC plug for the analog input signal and a 9 pin SUB-D connector for all other signals. The pin assignment of the SUB-D connector is shown here.

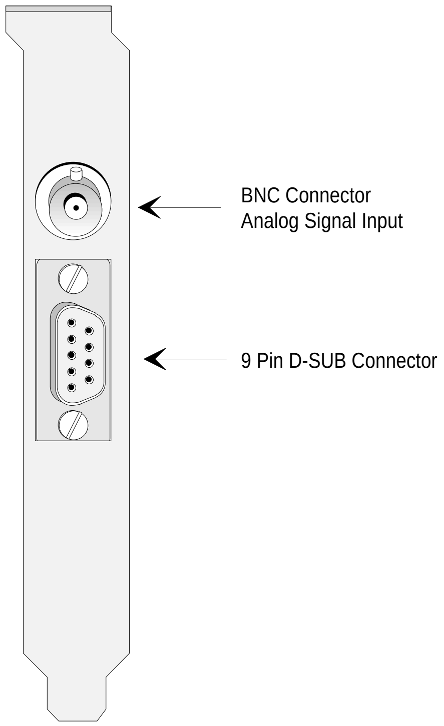


Figure 3.2: MCDLAP connectors at the mounting bracket

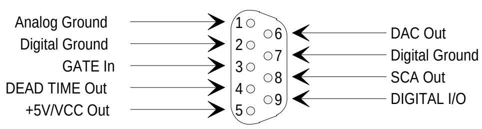


Figure 3.3: 9 Pin D-SUB connector

ANALOG SIGNAL INPUT: The analog signal input accepts positive or positive slope leading bipolar pulses. The input is DC coupled with an input impedance of 1kΩ. The nominal input range is 25µV to 8V. Risetime 100ns to 100µs, flat top, width ≥0.5µs, fall time 200ns to 100µs.

GATE INPUT: The GATE input accepts pos. TTL level signals ≥100ns width. The input impedance is approximately 4.7kΩ (internal pullup).

SCA OUTPUT: Positive TTL output pulse - width ≅500ns - for every valid event. A valid event is a pulse that corresponds to the lower and upper level discriminators.

DEAD TIME OUTPUT: Positive TTL output signal when the ADC is *dead*.

DIGITAL I/O: The DIGITAL I/O port is an open drain bidirectional port. To use it as an output a pullup resistor (≈1 to 10kΩ) to +5V is reqired.

DAC OUTPUT: The output of an 8bit digital to analog converter. It provides an output voltage of 0 to 9.961V in 255 steps (39mV per step).

ANALOG & DIGITAL GROUND: Internally connected but seperately output to minimize digital noise on the analog output line.

3.5. Battery Backup

An onboard battery allows for data retention even on power failures. There are two options available:

a rechargeable accu soldered on the board or an exchangeable battery in a socket. With an accu the jumper ACCU / BAT must be set to allow recharge.

With a battery the jumper ACCU / BAT must be removed to prevent damage!

If the MCDLAP is supplied with an accu and you store the card for longer periodes without usage the jumper BACKUP / STR can be removed to prevent low discharge. A battery shpould be removed in this case. For normal operation this jumper must be set.

3.6. Controls

All operation control is done by software. Thus, once installed there is no need to open your computer again.

3.6.1. Current Source Trim

To calibrate the full range input amplitude level adjust the current source with the 10 turn potentiometer.

It is recommended to leave this to authorized personel since wrong adjustment can inhibit proper operation!

3.6.2. Lower Level Discriminator

The lower discriminator level is controlled by an 8bit digital to analog converter providing levels from 0 to 10V in 255 steps.

3.6.3. Upper Level Discriminator

The upper discriminator level is controlled by an 8bit digital to analog converter providing levels from 0 to 10V in 255 steps. Care should be taken that the upper level is always higher than the lower level or the MCDLAP will be disabled.

3.6.4. Zero

The zero level is controlled by an 8bit digital to analog converter providing levels of approximately ±200mV in 255 steps.

3.6.5. Coincidence / Anticoincidence

Toggle mode of operation. For normal operation no GATE signal is needed and all valid (LLD ↔ ULD) pulses are accumulated. For coincidence mode a positive TTL signal must be present at the GATE input during the linear gate time. In anticoincidence mode a TTL *low* (zero) must be applied at the GATE input.

3.6.6. Real / Life Timer

In the Real Time register the real runtime since start is stored while the Life Time register contains the dead time corrected time.

A preset of the real or life timer can be used to interrupt the measurement when the preset time is reached.

3.7. Interrupt

The MCDLAP provides an interrupt capability. This allows to interrupt the CPU on an event that falls inside the discriminator levels. The interupt lines IRQ 3, 5, 10, 11 and 12 can be selected by setting a jumper on the corresponding field of the IRQ pins near the PC busconnector. Normally no jumper will be set and no interrupt is used.

4. Windows Server Program

The window of the MCDLAP server program WLAP.EXE is shown here. It enables the full control of the MCDLAP to perform measurements and save data. This program has no own graphic capabilities, 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.

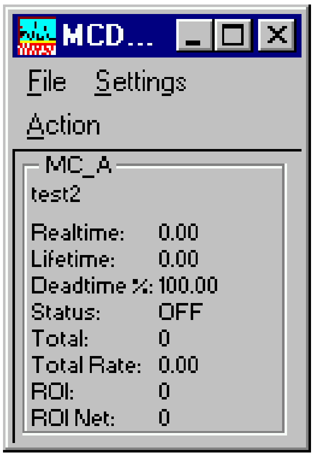


Figure 4.1: MCDLAP Server program with Status display

At program start the configuration files WLAP.INI (contains - for example - the I/O port base address in a format base=320; see Figure 4.2) and WLAPA.INF are loaded. Instead of this WLAPA.INF file any other setup file can be used if its name - excluding the appendix 'A.INF' - is used as command line parameter (e.g. WLAP TEST to load TESTA.INF) . The server program is normally shown as an icon. After a double click it is opened to show a status window.

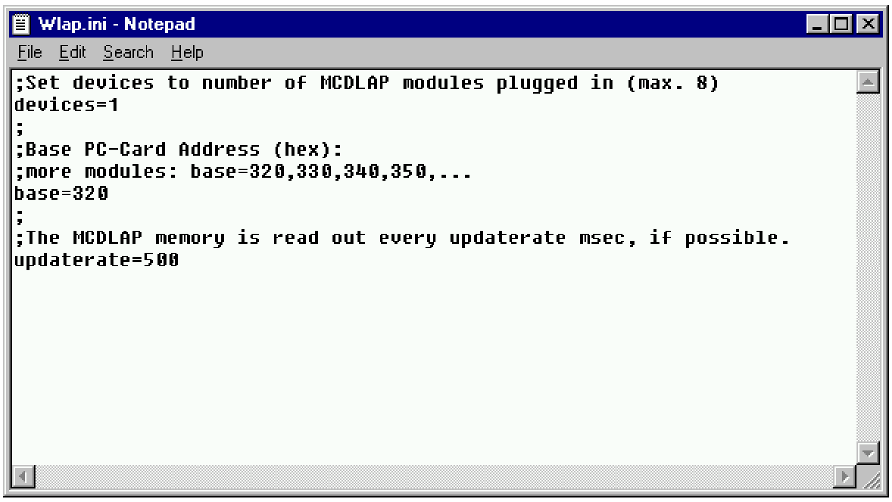


Figure 4.2: Sample WLAP.INI file

In the following the several dialogs are described in detail:

Clicking in the File menu on the Data... item opens the Data Operations dialog box.

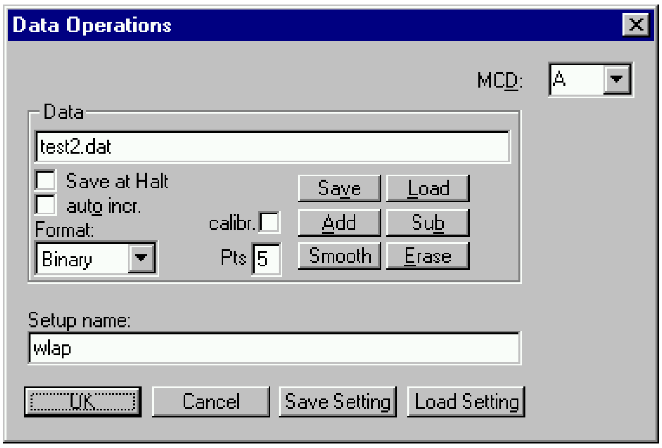


Figure 4.3: Data Operations dialog box

This dialog allows to edit the data settings. Mark the checkbox „**Save at Halt**" to store a spectrum- and a configuration file at the end of a measurement. The filename can be entered. If the checkbox **auto incr.** is checked, a 3-digit number is appended to the filename that is automatically incremented with each saving. The format of the data file can be ASCII (extension .ASC), binary (.DAT), or GANAAS (.SPE). The buttons **Save**, **Load**, and **Erase** perform the respective operation. With **Add** and **Sub** a spectrum can be added or subtracted from the present data. The Checkbox **calibr.** is checked if a calibration is used and the data is then adjusted according to the calibration. The **Smooth** button performs an n-point smoothing of the data. The number of points to average can be set with the **Pts** edit field between 3 and 21.

Clicking in the Settings menu on MCD... item opens the MCLAP Settings dialog box. Here parameters like presets, range parameters, etc. for the MCDLAP card can be set.

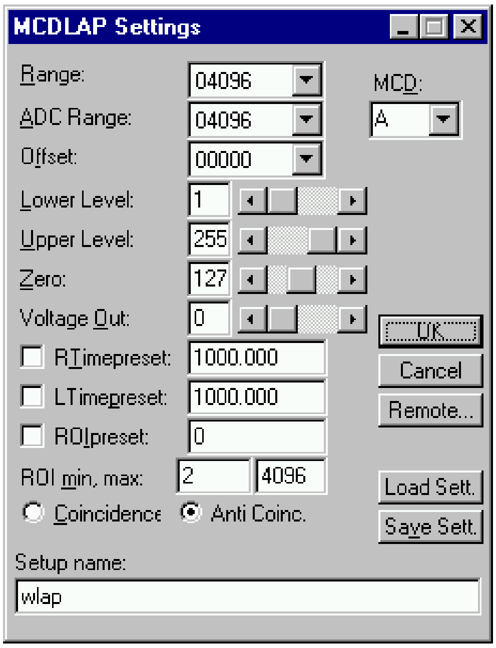


Figure 4.4: MCDLAP Settings dialog box

With the combo box „**Range**" the length of the spectrum can be chosen. Similar the **ADC Range** and **Offset** can be selected. 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 in the „**ROI**" limits, i.e. >= the lower limit and < the upper limit. Another possibility is to acquire data for a given realtime via the „**RTimepreset**" or livetime via the „**Ltimepreset**". The DAC values **Lower level**, **Upper level** and **Zero**, which are important for the ADC can be entered either via a scrollbar or in the edit field (the values must be in a range 0 ...

255). Also the DAC output value can be entered this way. The voltage can be defined in 256 steps by setting the value in the edit field to a number between 0 and 255 or the corresponding scrollbar can be used. The voltage is between 0 and 10 V in steps of 39 mV. The **Coincidence** or **Anticoincidence** mode of the coincidence input can be chosen via the radio buttons.

The „System...“ item in the settings menu opens the System Definition dialog box.

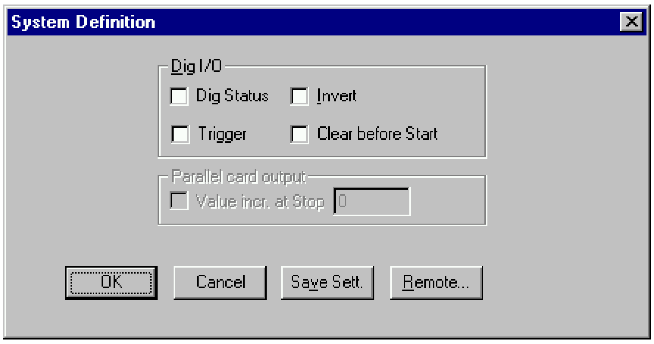


Figure 4.5: System Definition dialog box for a single MCDLAP card

The digital input / output pin 9 of the D-SUB connector can be used either as an output to show the status of the MCA if the checkbox „**Dig Status**“ is marked. The level is high for OFF and low for ON or vice versa if „**Invert**“ is checked. Alternatively, it is possible to use it as an external trigger input for starting and stopping the system („DESY control line“). If the checkbox **Trigger** is marked, a start command will not immediately start the system. After the start command, the digital input will be permanently checked for its logical level. If the level is low, the data is cleared if the corresponding checkbox **Clear before Start** is marked and the MCA will immediately be started. It will stop if the level returns to high (or vice versa if „**Invert** “ is marked). A stop command for the system will finish the digital input checking.

Via an optional available parallel port card it is possible to control for example a sample changer by checking **"Value incr. At Stop"**. The 8 bit value entered in the edit field (a number between 0 and 255) is output at the digital I/O port. The value will always be incremented by one if MC_A is stopped.

If more than one MCDLAP card is used, the system definition dialog box comes up as shown in Figure 4.6. Here the several units can be combined to form up to 4 separate systems that can be started, stopped and erased by one command.

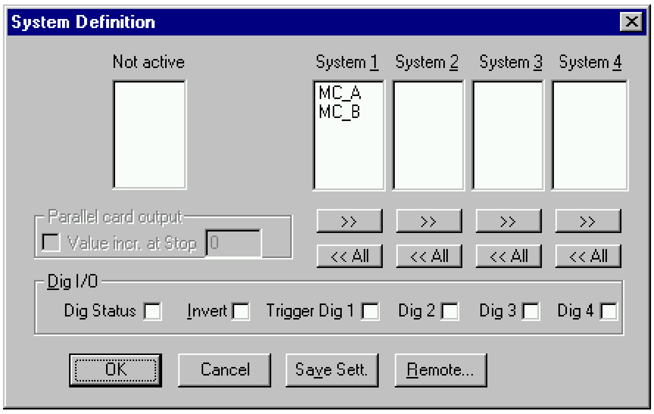


Figure 4.6: System Definition dialog box, two MCDLAP cards

In the shown setting a single system is formed. The two MCDLAPs A and B are combined, both operate in one system. 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.

Click **OK to** accept all settings. **Cancel** cancels all changes. Clicking „**Save Settings**“ stores all settings in the file **WLAPA.INF** in a form:

wndwidth=128
wndheight=186
range=4096
adcrange=4096
offset=0
lowlevel=1
uplevel=255
zerolevel=127
outlevel=0
rtpreset=1000
rtprena=0
ltpreset=1000
ltprena=0
roipreset=0
roiprena=0
anti=1
smpluse=0
smplval=0
roimin=2
roimax=4096
autoinc=0
datname=test2.dat
savedata=0
fmt=dat

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

The „**Remote mode...**“ item in the settings menu or the „**Remote“** button in the System Definition dialog box opens the Remote Control dialog box. Here all settings can be made for the control of the MCDLAP server program via a serial port.

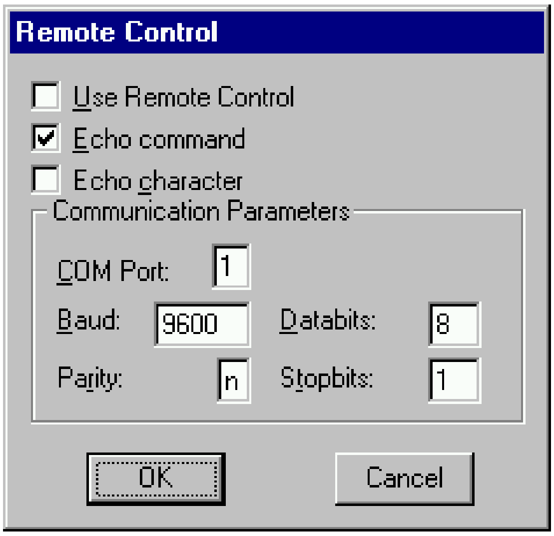


Figure 4.7: Remote Control dialog box

If the Checkbox „Use Remote Control" is marked and the COMCTL.DLL is available, the specified COM port will be used for accepting commands. 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. The possible commands and their syntax are listed in the following section.

4.1. Control Language

A sequence of commands that are stored in a file with extension .CTL can be executed by the MCDLAP server program or MCDWIN with the „Load " command. Also the configuration files WLAPx.INF or the header files with extension .MLP contain such commands to set the parameters. Each command starts at the beginning of a new line with a typical keyword, the case is ignored. Any other 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 section 4.2. The application name for opening the DDE connection with the standard MCDLAP server program WLAP.EXE is WLAP, the topic is MCDLAP. 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 a TCP/IP Winsock installed like the Trumpet winsockets 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 WLAPA.INF contains a complete list of commands for setting parameters; an example is:

```
wndwidth=128        ; Sets width of server window
wndheight=186       ; Sets height of server window
range=4096          ; Memory size for spectra
```

```
adcrange=4096        ; ADC resolution
offset=0             ; start point of spectra in memory (in steps of 1024)
lowlevel=1           ; Lower discriminator threshold value
uplevel=255          ; Upper discriminator value
zerolevel=127        ; Zero level value
outlevel=0           ; Level at DAC output
rtpreset=1000        ; Realtimepreset value (seconds)
rtprena=0            ; Realtime preset enable (1=enabled)
ltpreset=1000        ; Lifetime preset value (seconds)
ltprena=0            ; Lifetime preset enable
roipreset=0          ; ROI preset value
roiprena=0           ; ROI preset enable
anti=1               ; Coincidence ANTI
smpluse=0            ; Use of optional DIG I/O output
smplval=0            ; Value for optional DIG I/O output
roimin=2             ; ROI lower limit (inclusive)
roimax=4096          ; ROI upper limit (exclusive)
autoinc=0            ; Enable Auto increment of filename
datname=test2.dat    ; Filename
savedata=0           ; Save at Halt
fmt=dat              ; Format (ASCII: asc, Binary: dat, GANAAS: spe, Dual Binary: da2)
smoothpts=5          ; Number of points to average for a smooth operation
```

A data header file with extension .MLP contains a subset of above parameters and some additional information typical for the special measurement. An example is the file Sd0002.MLP:

```
REPORT-FILE from 11/22/94 13:52:56 written 01/30/96 9:57:55
                          ; the first time is when the measurement was started,
                          ; the 2nd when the data file was written
REALTIME:
3012.000                  ; real time in seconds
LIFETIME:
3000.000                  ; life time in seconds
TOTALSUM:
1870706                   ; total sum of counts
ROISUM:
1870706                   ; sum of counts in ROI
NETTOSUM:
1858424                   ; sum in ROI with background subtracted
cmline0= 11/22/94 13:52:56    ; comment lines: the first line always contains the start time
cmline1=2
cmline2=Calibration source
cmline3=Oberhaching
cmline4=1/1/93 12:00:00
cmline5=10
cmline6=mg
cmline7=3
cmline8=Ge
cmline9=test
range=4096                ; subset of parameters as in a WLAPx.INF file...
adcrange=4096
offset=0
lowlevel=15
uplevel=240
zerolevel=115
outlevel=0
rtpreset=1000
rtprena=0
ltpreset=1000
ltprena=0
roipreset=0
roiprena=0
anti=1
roimin=2
roimax=4096
autoinc=0
```

```
datname=C:\WLAP\DATA\SD0002.mlp
savedata=1
fmt=dat
caluse=1                ; Use Energy calibration
calch00=2344.00         ; Calibration points..
calvl00=1173.264000
calch01=2662.00
calvl01=1332.500000
caloff=-0.475572        ; Calibration formula polynomial coefficients..
calfact=0.500742
calfact2=0
calfact3=0
calunit=keV             ; Calibration unit
roi=702 721             ; ROI limits..
roi=597 614
roi=544 561
roi=757 777
roi=863 882
roi=151 172
roi=236 252
roi=267 281
roi=315 328
roi=1311 1336
roi=1657 1682
roi=2332 2360
roi=2648 2675
roi=2891 2947
roi=3652 3694
```

The following commands perform actions and therefore usually are not included in a MCDx.INF 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 suspended (see start).
start3      ; Clears and starts system 3. Further execution suspended (see start).
start4      ; Clears and starts system 4. Further execution suspended (see start).
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 cont).
cont3	; Continues acquisition of system 3 (see cont).
cont4	; Continues acquisition of system 4 (see cont).
savecnf	; Writes the settings into WLAPA.INF (, WLAPB.INF,...)
MC_A	; Sets actual multichannel analyzer to MC_A for the rest of the ; control file.
MC_B	; Sets actual multichannel analyzer to MC_B for the rest of the ; control file.
MC_C	; Sets actual multichannel analyzer to MC_C for the rest of the ; control file.
MC_D	; Sets actual multichannel analyzer to MC_D for the rest of the ; control file.
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 to 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 WLAP.exe (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 pin 9 of D Sub connector.
waitpin 4000	; Waits 4000 ms for going the level low at pin 12 of optional parallel card. ; After a timeout a Message box warns and waits for pressing OK. ; Can be used for connecting a sample changer.

run controlfile	; Runs a sequence of commands stored in control file. This command ; cannot be nested, i.e. it is not possible to execute a run command ; from the control file 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.
exec program	; Executes a Windows program or .PIF file. ; Example: exec notepad test.ctl ; opens the notepad editor and loads test.ctl.
fitrois	; Makes a single peak Gaussian fit for all ROIs 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 fitrois 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 and uses the result 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 TCP/IP control:

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

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

4.2. Controlling the MCDLAP Windows Server via DDE

The MCDLAP program can be a server for a 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 server program are described together with a demo VI („Virtual Instrument“) for LabVIEW. It is not recommended to use the DDE protocol for LabVIEW, as a DLL interface is (optionally) available which works much faster. The following should be seen as a general description of the DDE conversation capabilities of the server program.

4.2.1. Open Conversation

application: WLAP
topic: MCDLAP

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

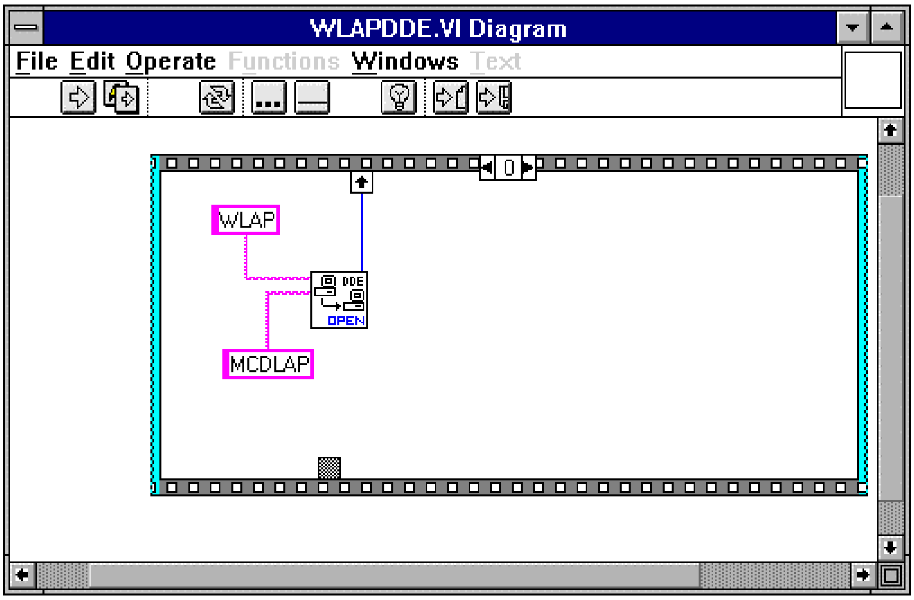


Figure 4.8: Opening the DDE conversation with the MCDLAP server in LabVIEW

4.2.2. DDE Execute

The DDE Execute command can be used to perform any action of the WLAP program. Any of the Control command lines described in chapter 4.1 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 WLAP program then executes the command and, after finishing, it sends an Acknowledge message to the DDE client. This can be used for synchronizing the actions in both applications.

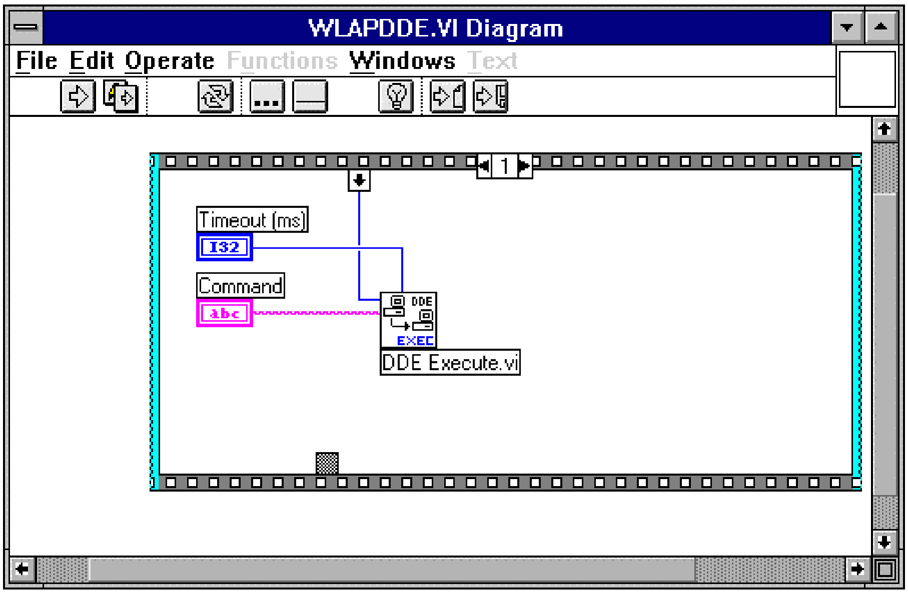


Figure 4.9: Executing a control command from a LabVIEW application

4.2.3. 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 MCDLAP 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 in the actual multichannel analyzer. The desired multichannel analyzer can be selected before by a command MC_A, ..., MC_D.

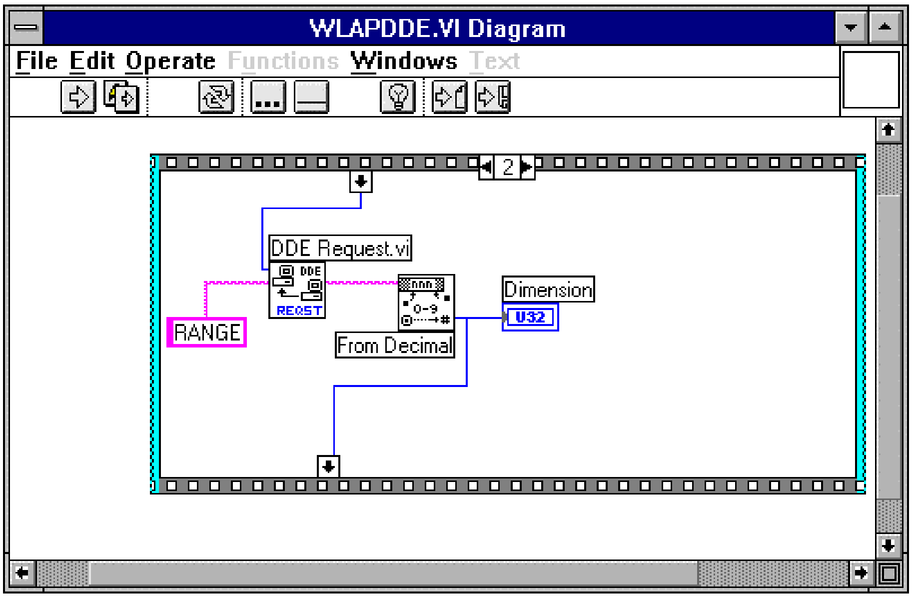


Figure 4.10: Getting the total number of data with LabVIEW

DATA

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

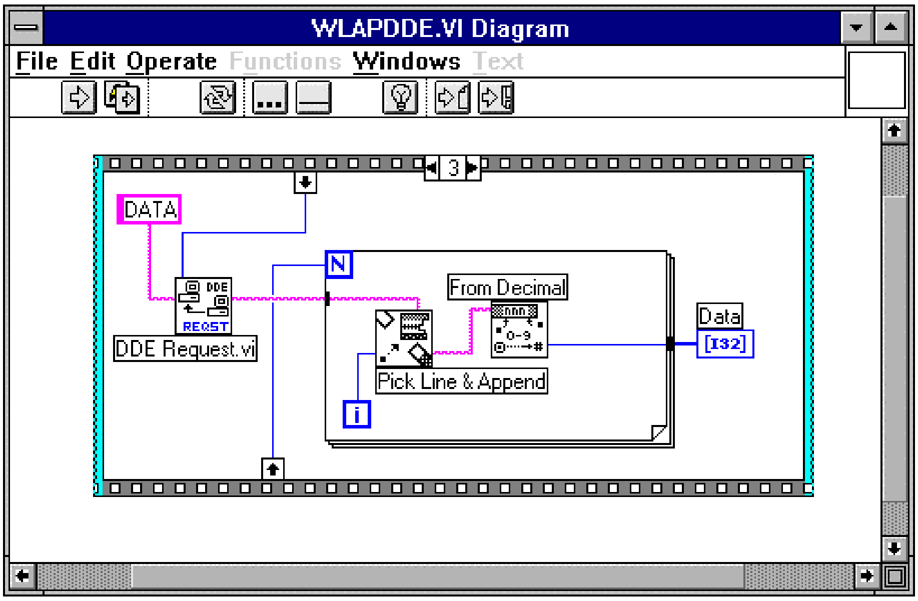


Figure 4.11: Getting the data with LabVIEW

4.2.4. Close Conversation

After finishing the DDE communication with the 7882 program, it must be closed.

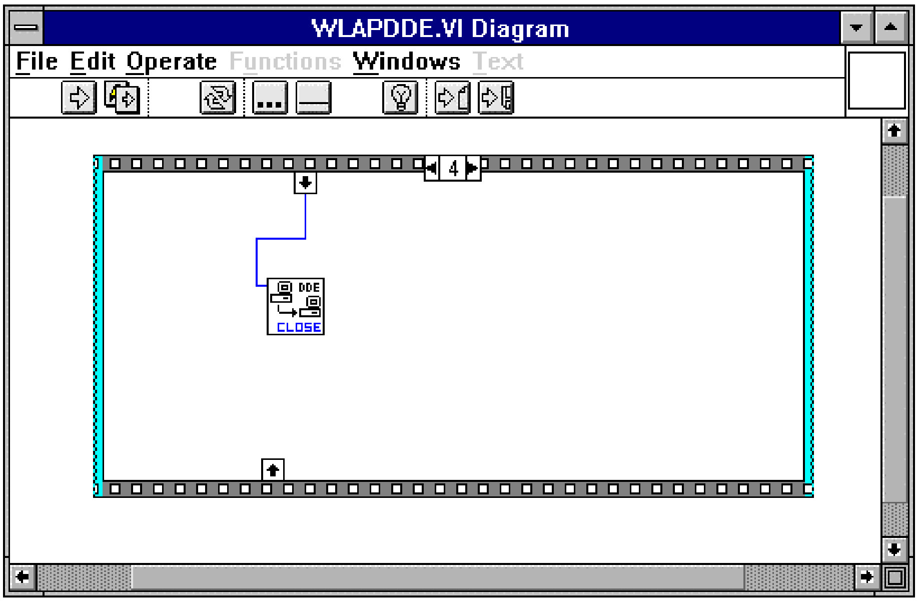


Figure 4.12: Closing the DDE communication in LabVIEW

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

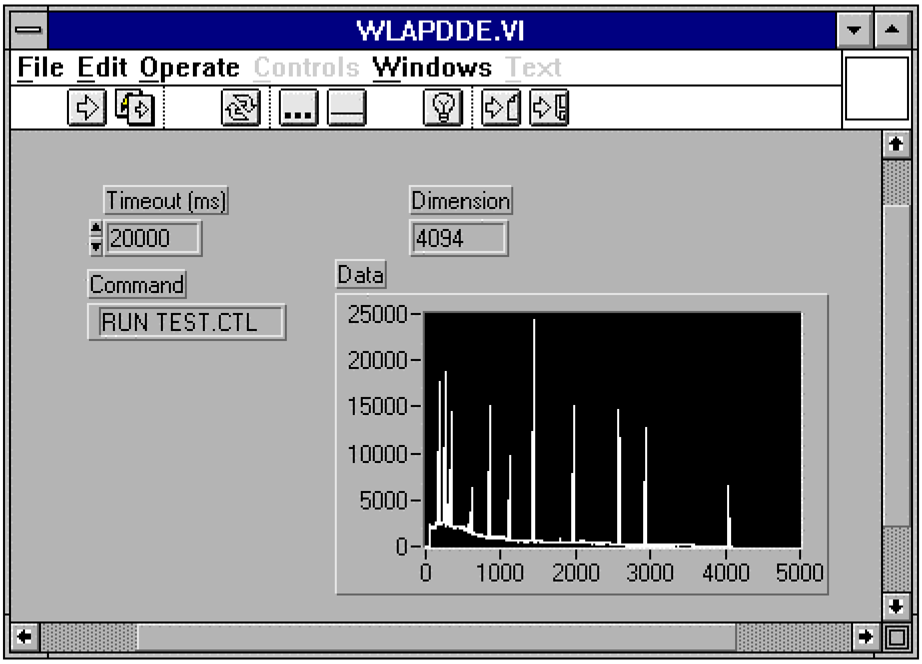


Figure 4.13: Control Panel of the demo VI for LabVIEW

4.3. Controlling the MCDLAP Windows Server via DLL

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

In the following some parts of the header and definition files of the DLAP.DLL are listed, that may help an experienced programmer to use the DLL for own written applications. Please note that the complete documented source code of the DLL including fundamental VI's and an example VI for LabVIEW and example program in Visual Basic is available as an option.

```
typedef struct{
  int started;          // acquisition status: 1 if running, 0 else
  double realtime;      // real time in seconds
  double totalsum;      // total events
  double roisum;        // events within ROI
  double totalrate;     // acquired events per second
  double nettosum;      // ROI sum with background subtracted
  double livetime;      // Lifetime in seconds
  double deadtime;      // Dead time in percent
  unsigned long maxval; // Maximum value in spectrum
} ACQSTATUS;

typedef struct{
  long range;                 // spectrum length
  int rtprena;                // 1 if realtime preset enabled, 0 else
  int roiprena;               // 1 if ROI event preset enabled, 0 else
  long roimin;                // lower ROI limit
  long roimax;                // upper limit: roimin <= channel < roimax
  double roipreset;           // ROI preset value
  double rtpreset;            // time preset value
  int savedata;               // 1 if auto save after stop
  int fmt;                    // format type: 0 == ASCII, 1 == binary
  int autoinc;                // 1 if auto increment filename
  int anti;                   // 1 if anticoincidence, 0 == coinc.
  int ltprena;                // 1 if livetime preset enabled, 0 else
  int outlevel;               // DAC output level, 0...255
  int adcrange;               // ADC range
  int offset;                 // ADC offset
  int lowlevel;               // ADC lower level limit
  int uplevel;                // ADC upper level limit
  int zerolevel;              // ADC zero level
  double ltpreset;            // live time preset
  int nregions;               // number of regions
  int caluse;                 // use calibration
  double scalpreset;          // dummy
  int active;                 // always 1
  int calpoints;              // number of calibration points
} ACQSETTING;



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
} ACQDATA;

typedef struct {
  int nDevices;           // Number of devices: always 4
  int nDisplays;          // Number of displays (active MCA's): 0...4
  int nSystems;           // Number of systems
  int bRemote;            // 1 if server controled by MCDWIN
  int sys;                // System definition word
} ACQDEF;

/*** FUNCTION PROTOTYPES (do not change) ***/
int  FAR PASCAL LibMain(HANDLE, WORD, WORD, LPSTR);
VOID FAR PASCAL StoreSettingData(ACQSETTING FAR *Setting, int nDisplay);
```

```
                                    // Stores Settings into the DLL
int FAR PASCAL GetSettingData(ACQSETTING FAR *Setting, int nDisplay);
                                    // Get Settings stored in the DLL
VOID FAR PASCAL StoreStatusData(ACQSTATUS FAR *Status, int nDisplay);
                                    // Store the Status into the DLL
int FAR PASCAL GetStatusData(ACQSTATUS FAR *Status, int nDisplay);
                                    // Get the Status
VOID FAR PASCAL Start(int nSystem);             // Start
VOID FAR PASCAL Halt(int nSystem);              // Halt
VOID FAR PASCAL Continue(int nSystem);          // Continue
VOID FAR PASCAL NewSetting(int nDevice);
                                    // Indicate new Settings to Server
UINT FAR PASCAL ServExec(HWND ClientWnd);// Execute the Server WLAP.EXE
VOID FAR PASCAL StoreData(ACQDATA FAR *Data, int nDisplay);
                                    // Stores Data pointers into the DLL
int FAR PASCAL GetData(ACQDATA FAR *Data, int nDisplay);
                                                // Get Data pointers
long FAR PASCAL GetSpec(long i, int nDisplay);
                                                // Get a spectrum value
VOID FAR PASCAL SaveSetting(void);              // Save Settings
int FAR PASCAL GetStatus(int nDevice);
                                    // Request actual Status from Server
VOID FAR PASCAL Erase(int nSystem);             // Erase spectrum
VOID FAR PASCAL SaveData(int nDevice);          // Saves data
VOID FAR PASCAL GetBlock(long FAR *hist, int start, int end, int step,
  int nDisplay);                    // Get a block of spectrum data
VOID FAR PASCAL StoreDefData(ACQDEF FAR *Def);
                                    // Store System Definition into DLL
int FAR PASCAL GetDefData(ACQDEF FAR *Def);     // Get System Definition
VOID FAR PASCAL LoadData(int nDevice);          // Loads data
VOID FAR PASCAL AddData(int nDevice);           // Adds data
VOID FAR PASCAL SubData(int nDevice);           // Subtracts data
VOID FAR PASCAL Smooth(int nDevice);            // Smooth data
VOID FAR PASCAL NewData(void);      // Indicate new ROI or string Data
VOID FAR PASCAL HardwareDlg(int item);
                                    // Calls the Settings dialog box
VOID FAR PASCAL UnregisterClient(void);
                                    // Clears remote mode from MCDWIN
VOID FAR PASCAL DestroyClient(void);            // Close MCDWIN
UINT FAR PASCAL ClientExec(HWND ServerWnd);
                                    // Execute the Client MCDWIN.EXE
int FAR PASCAL LVGetDat(unsigned long huge *datp);
                                    // Copies the spectrum to an array
VOID FAR PASCAL RunCmd(int nDevice, LPSTR Cmd);
                                                // Executes command
int FAR PASCAL LVGetRoi(unsigned long far *roip, int nDevice);
                              // Copies the ROI boundaries to an array
int FAR PASCAL LVGetCnt(double far *cntp, int nDevice);
                              // Copies Cnt numbers to an array
int FAR PASCAL LVGetStr(char far *strp, int nDevice);
                              // Copies strings to an array
EXPORTS
        WEP       @1 RESIDENTNAME
        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
```

5. MCDWIN Program

The window of the MCDWIN program is shown here. It enables the full control of the MCDLAP card via the server program to perform measurements and save data, and shows the data on-line in several windows.

The server program WLAP.EXE automatically starts MCDWIN. If you try to start MCDWIN before the server is started, a message box warns that you should start the server first.

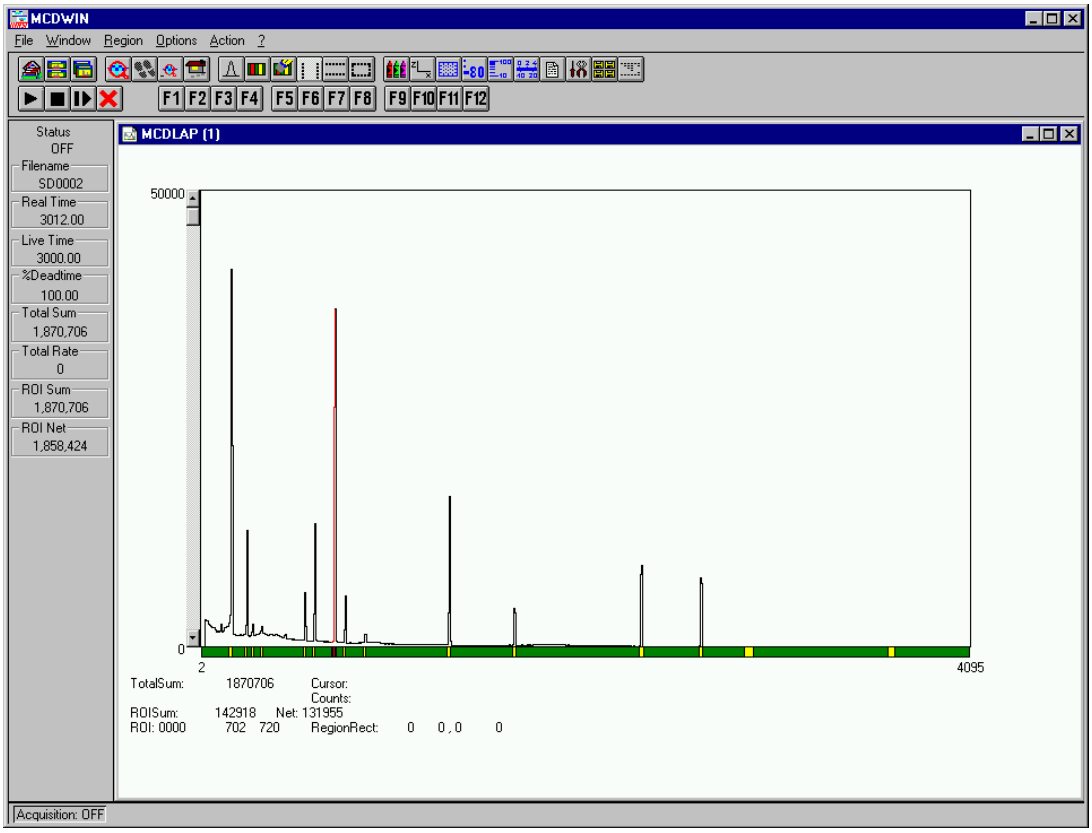


Figure 5.1: MCDWIN main window

A status window at the left side gives all information about the status of the MCDLAP. A toolbar provides fast access to many used functions in the menu. A status bar at the bottom indicates the meaning of the toolbar icons. A cursor appears when clicking the left mouse button inside the graphic area. To clear the cursor, move the cursor outside the spectrum display and double click with the right mouse button. To define a region, press the right mouse button, and while keeping the button pressed, drag a rectangle. In the 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. For the MCDLAP this may be a usefull feature if for example several 1k spectra are acquired into the RAM using different offsets and you want display the several spectra in an isometric picture

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.

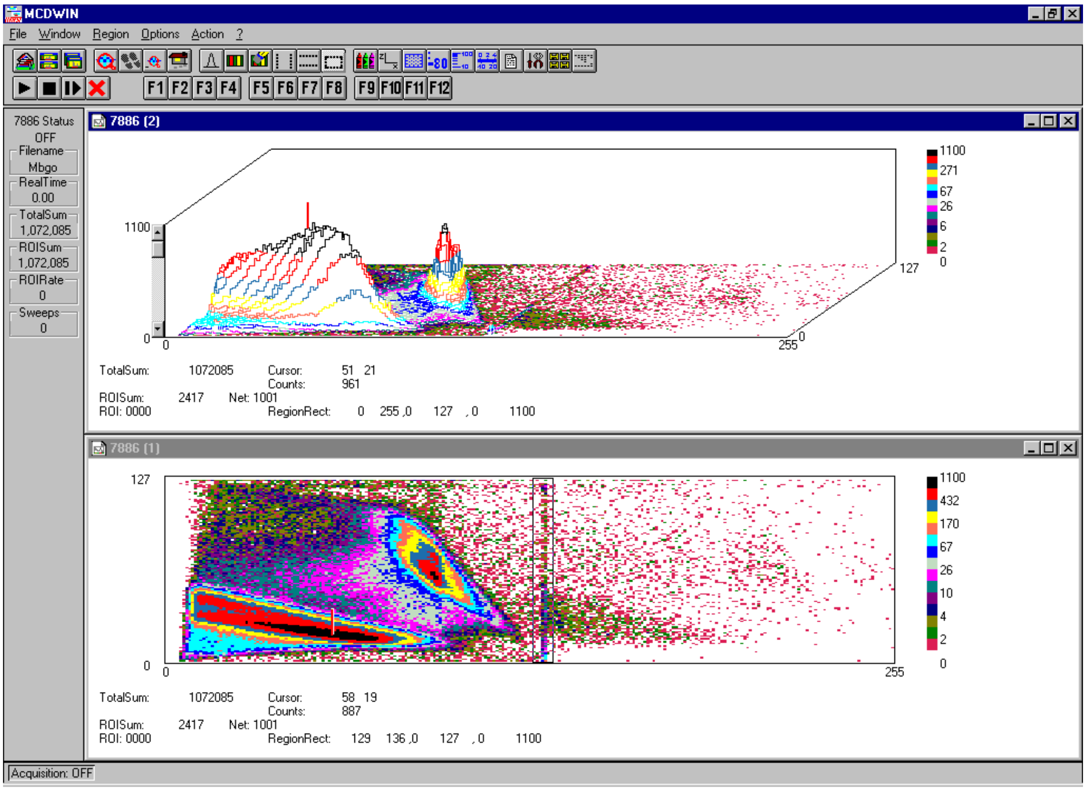


Figure 5.2: MCDWIN Map and Isometric display

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 into the MCA selected by the active window. When saving data, you have the choice between binary (.DAT), ASCII (.ASC), and GANAAS (.SPE) format. When you load data, select a header file (extension .MLP). This file contains the information about the size and format of the data file, which is then automatically read. With „**Add**“ the data is added to the present data. The data read from a file is shifted according to the calibration, if it is available.

New Display...

With the Open New menu item or the corresponding icon a new Display window can be created and shown as the active window. If more than one MCDLAPs are installed (by writing a line devices=2 into the WLAP.INI file), in the „**Open New Display**“ dialog box the MCA for the new display can be selected.

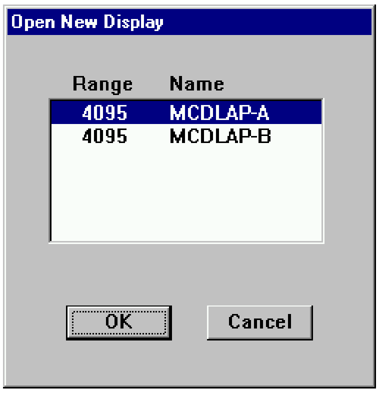


Figure 5.3: File New Display dialog box

Open All

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

Close All

By selecting the Close All menu item, all available Displays are closed.

Compare...

The Compare... menu item allows to compare single spectra.

To use this feature it is necessary to have at least two MCDLAP devices defined in the WLAP.INI file also if only one MCDLAP is physically available.

A Display window containing a single spectrum must be active; clicking this menu item opens then a dialog box to select one or more of available other single spectra Displays. The selected spectra are then plotted into the active Display window in a different color.

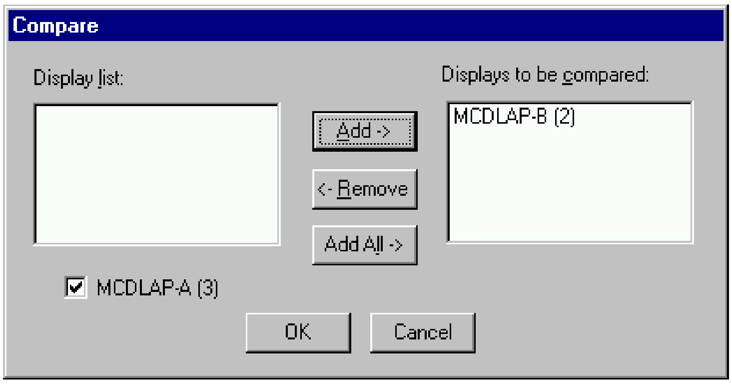


Figure 5.4: Compare dialog box

Print...

The Print menu item prints a Display to the printer. Only the visible part of the spectra will be printed. The size and position of the graphic on paper can be adjusted in a dialog box.

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.

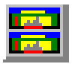
Exit

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 allocate the same size for each window.

Cascade

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

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.

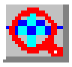
Window list

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 window and is displayed in front of all the others.

5.3. Region Menu

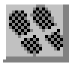
The Region menu contains commands for Regions and ROIs (Regions of Interest). A Region can be defined by marking it in a display, with the mouse using the right mouse button and dragging a rectangle over the area one is interested in. A ROI, i.e. an already defined region in a single spectrum can be 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



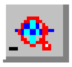
The Zoom item or respective icon enlarges a Region to the maximum Spectrum Display size.

Back



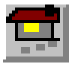
The Back menu item or clicking the corresponding icon restores the last zoom view. Each time a Back command is clicked the view is stepped back one step.

Zoom Out



The Zoom Out menu item or clicking the corresponding icon enlarges the actual zoom view by a factor 2, if possible.

Home

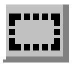


Clicking the Home menu item or the corresponding icon restores a Display to the basic configuration.

Shape

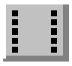
Selecting the Shape menu item opens a submenu with the items Rectangle, X-Slice, Y-Slice, and Polygon to choose the ROI shape.

Rectangle



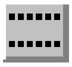
Sets the region shape to a rectangle with arbitrary dimensions. To enter the rectangular region, press the right mouse button, drag a rectangle, and release the button to define the region.

X-Slice



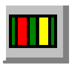
Sets the Region shape to the rectangle with maximal height.

Y-Slice



Sets the Region shape to the rectangle with maximal 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 or delete, change and activate an existing ROI. Also the peak values 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 it can be written into a file with extension .CTL, which can be directly loaded into the server to restore the ROI list.

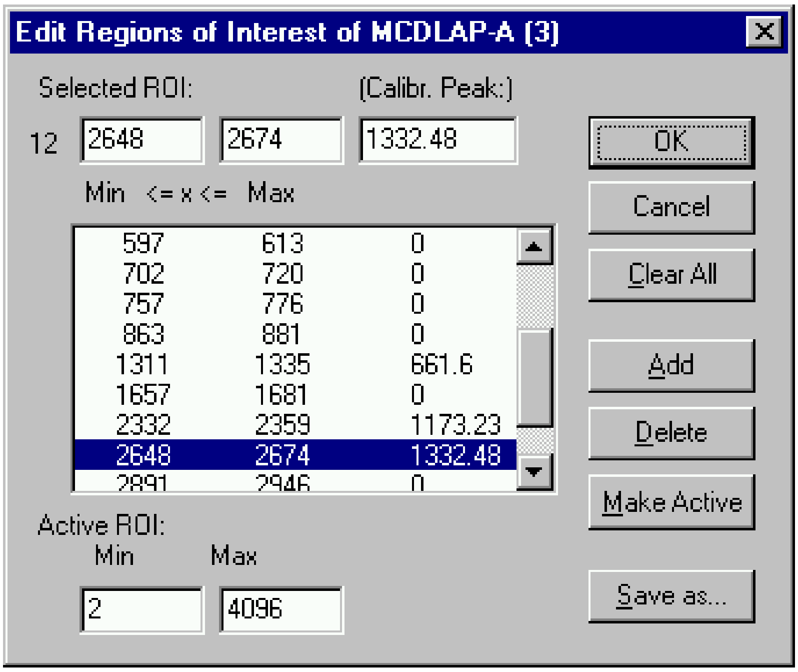
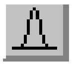


Figure 5.5: ROI Editing dialog box

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:

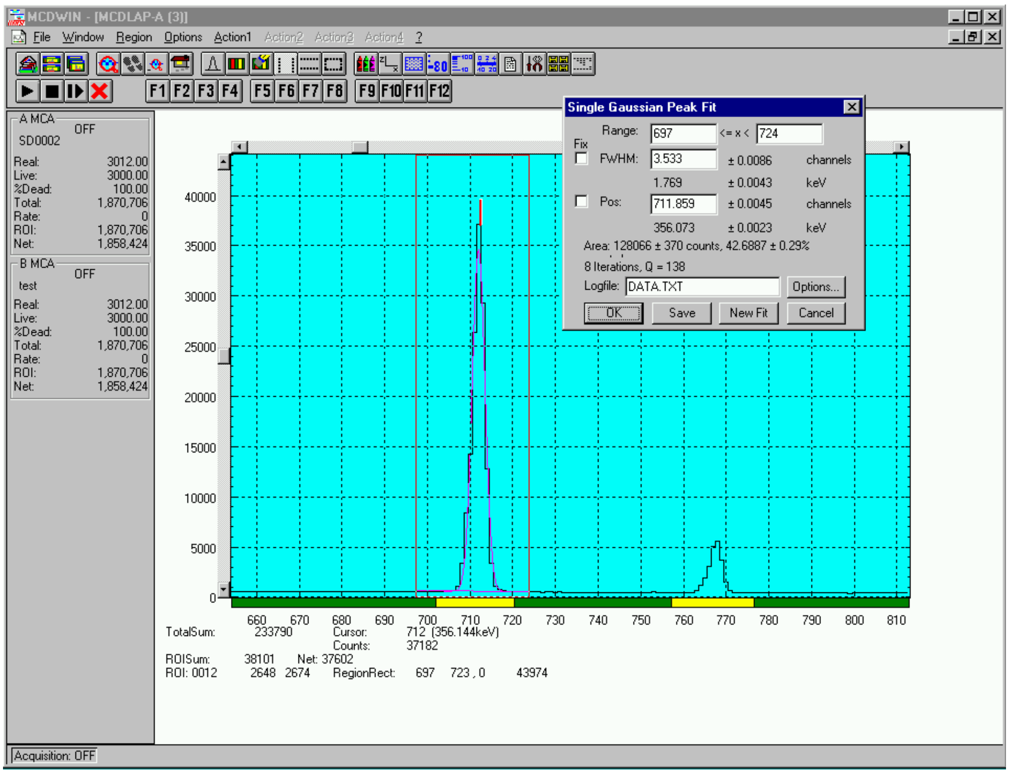


Figure 5.6: Single Gaussian Peak Fit

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 line shape, you may multiply the errors with the square root of Q. Click on Save to append a line containing the results to a Logfile with the specified name. OK closes the dialog and lets the fitted function in the display also if it 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:

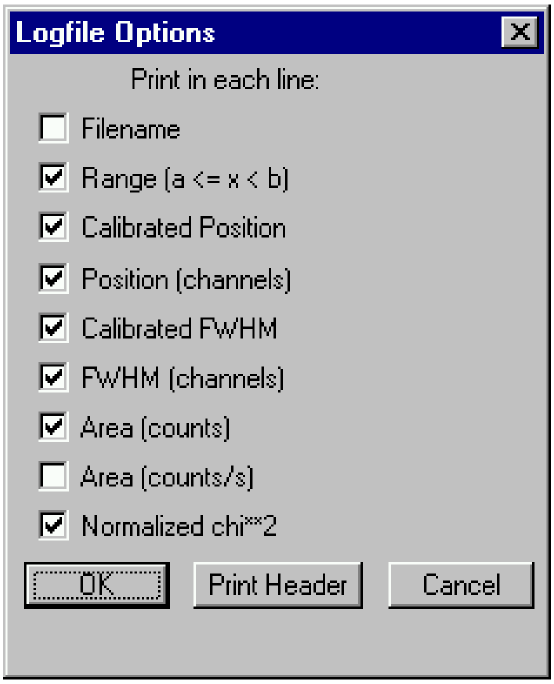


Figure 5.7: Log file Options for the Single Gaussian Peak Fit

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

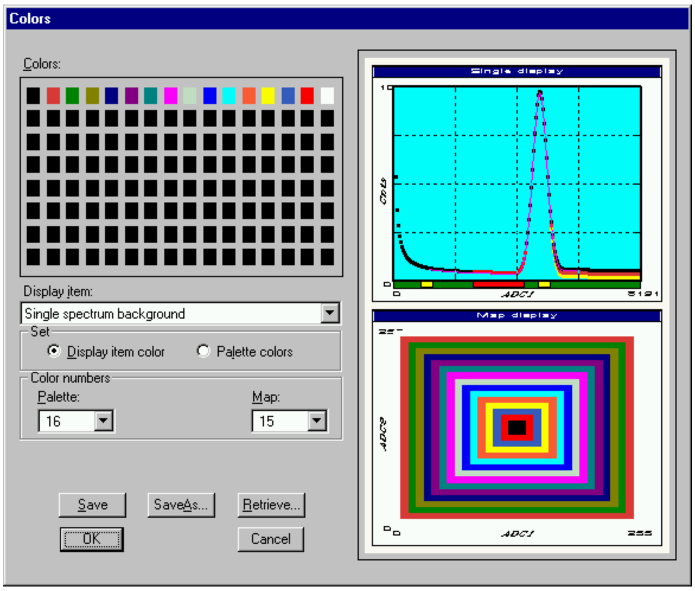


Figure 5.8: Colors dialog box

Display...

The Display menu item or the corresponding icon opens the Display Options 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**.

'Dot' means that each spectra point is shown as a small rectangle, the size of this rectangle 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 that are below this value are then not taken into account for the mean value calculation.

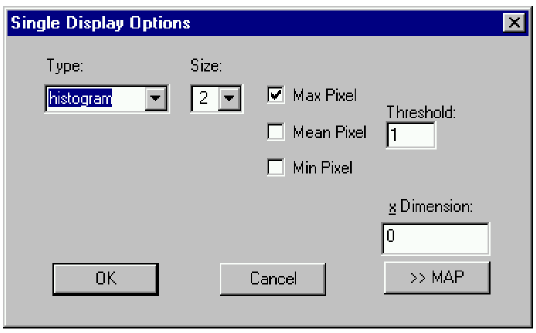
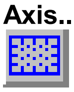


Figure 5.9: Display Options dialog box

Axis...



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

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.

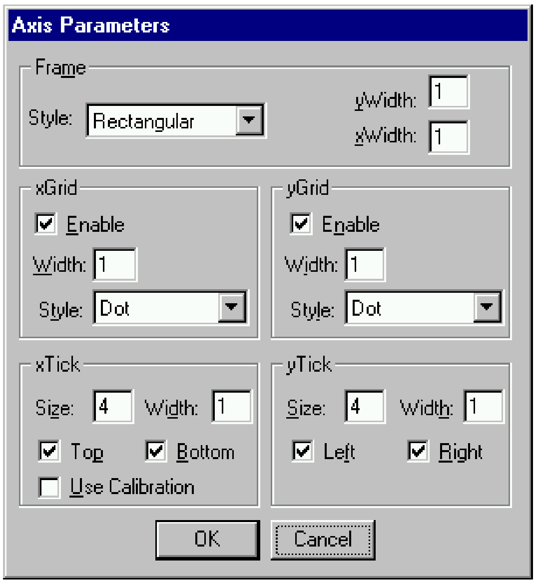


Figure 5.10: Axis Parameter dialog box

Scaling...

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

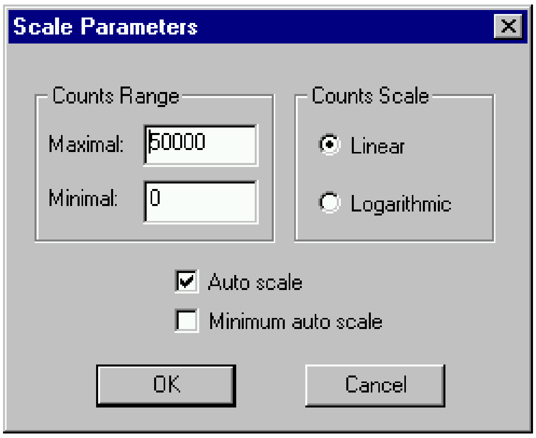


Figure 5.11: :Scale Parameters dialog box

It allows to change the ranges and attributes of a Spectrum axis. By setting the Auto scaling mode, the MCDWIN will automatically recalculate the maximum y axes of the visible Spectrum region only. To keep the same height of the visible region for a longer time, set the Auto scaling mode off. 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

For a Lin scale all data intervals have the same size. With Log scale the intervals will be small for small y values and large for large y values. All options have effect only on the active Display.

Calibration...

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

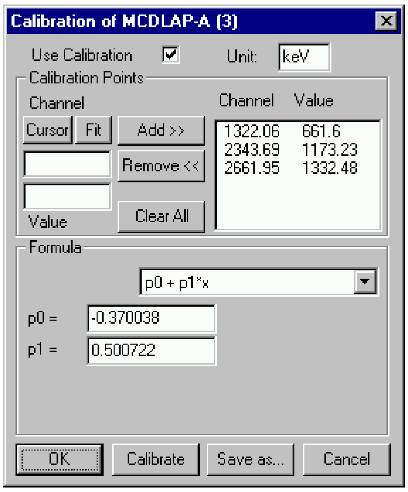
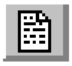


Figure 5.12: Calibration dialog box

Make a choice of several calibration formulas. Enter some cursor positions and the corresponding values, click on Add, 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 on, the calibrated values are displayed together with the channel position of the cursor.

Comments...



Up to eleven 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 contains automatically the time and date when a measurement was started. The titles of each line can be changed by editing the file COMMENT.TXT.

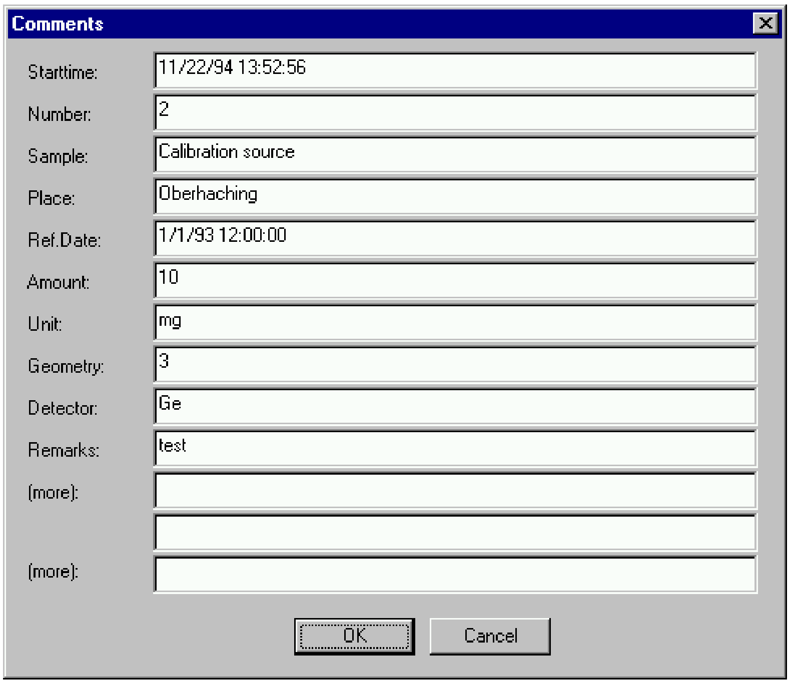


Figure 5.13: Comments dialog box

Range, Preset...

The Range, Preset dialog box allows to make all the respective MCDLAP settings (See MCDLAP Server documentation).

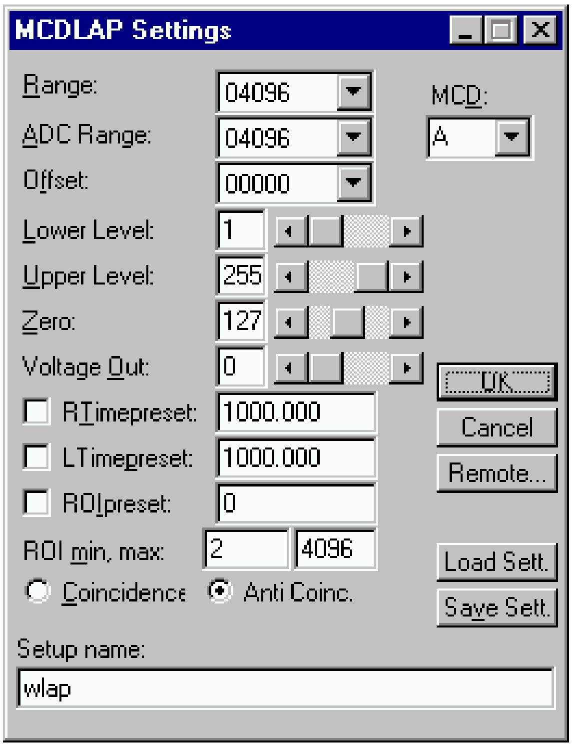
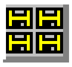


Figure 5.14: Settings dialog box

Data...



The Data dialog box allows to make all the respective MCDLAP settings (See MCDLAP Server documentation).

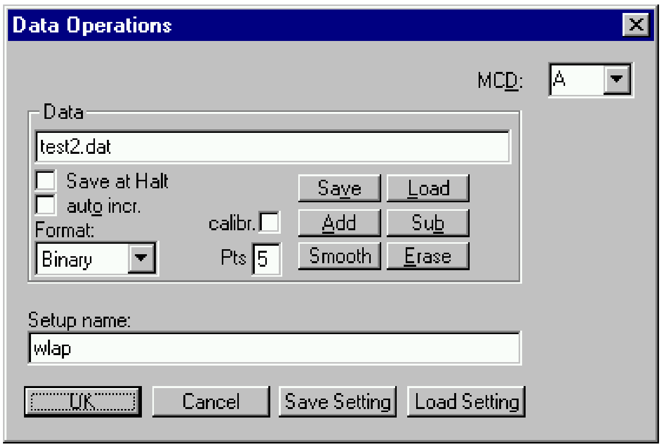
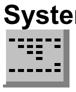


Figure 5.15: Data Operations dialog box

System...



The System Definition dialog box allows to make all the respective MCDLAP settings (See MCDLAP Server documentation).

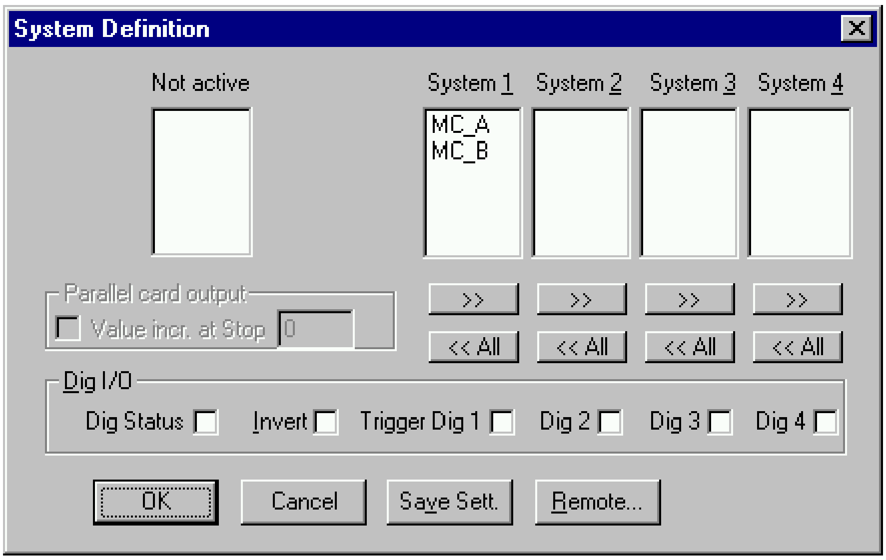
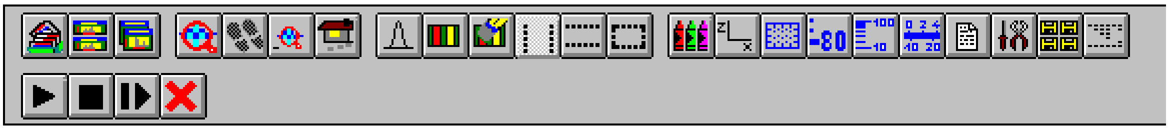


Figure 5.16: System Definition dialog box

Tool Bar...



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

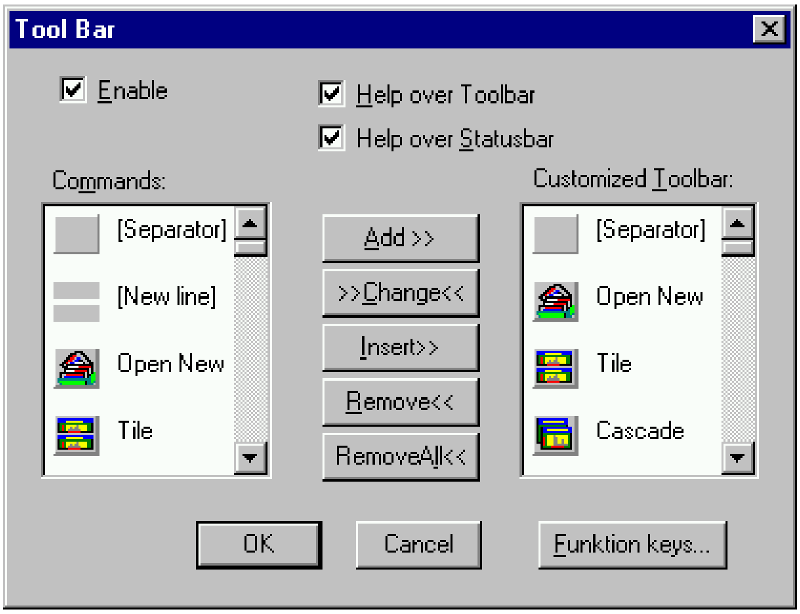


Figure 5.17: Tool Bar dialog box

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.

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 check mark shows if it is active or not. The Status bar usually shows if an acquisition is active. When the left mouse button is pressed while the mouse cursor is within a toolbar icon, it displays a short help message what the meaning of the toolbar icon is.

Status window

The same way it is possible to hide or show the status window at the left side of the MCDWIN main window. The fonts can be chosen between a larger and smaller set if again selecting this item.

Save

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

Save As...

Saves all MCDWIN parameters defined in the Options menu to a user defined config file. The default settings in MCDWIN.CNF are loaded when starting MCDWIN.

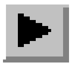
Retrieve...

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 systems are formed, also more actions menus are available, otherwise they are grayed.

Start



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

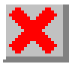
Halt

The Halt toolbar button stops a measurement.

Continue

The Continue toolbar button continues a measurement.

Erase



The Erase toolbar button erases the data.

6. MCDLAP Programming

6.1. Register Specification

The MCD/LAP card is controlled via input and output to some I/O port registers. The base address is defined by the rotary switch setting as described in chapter 2.

Address BASE +	**Width [bits]**	**Write Operation**	**Read Operation**
0	16	CONTROL	STATUS
1	8	DAC 0	*Reserved*
2	16	RAM ADDRESS	*Reserved*
3	8	DAC 1	*Reserved*
4	16	RAM WRITE	RAM READ LOW
5	8	DAC 2	*Reserved*
6	16	ADC RANGE	RAM READ HIGH
7	8	DAC 3	*Reserved*

Figure 6.1: Register Overview

Control register
Base + 0

The control register is accessed by a 16 bit output to the base address. The bits in the control register are defined and used as follows:

Bit	**Name**	**Meaning**
0	PRE REAL	Controls a Realtime preset. A value of one enables an automatic stop of the measurement, if the real time counter (channel 0), which must be set to a value of 2^{32} - preset, after overflow, reaches the value zero. Note that the counter is incremented every millisecond.
1	PRE LIFE	Controls a Lifetime preset. A value of one enables an automatic stop of the measurement, if the life time counter (channel 1), which must be set to a value of 2^{32} - preset, after overflow, reaches the value zero.
2	CLEAR INT	A value of one clears the interrupt. It is reset to zero automatically.
3	---	Not used.
4	ENC	Enable Conversion. A value of one starts acquiring data, a value of zero stops the measurement.
5	ANTI	Controls the coincidence mode. A value of one means anticoincidence, i.e. a TTL low (zero) must be applied at the GATE input. A zero means coincidence mode, i.e. a TTL high signal must be at the GATE input.
6	RESET	A value of one resets the card.
7	RDDIR	A value of one enables writing data to the card RAM, a zero enables reading data.
8	RN0	Bit 0 of RN (conversion range).
9	RN1	Bit 1 of RN.
10	RN2	Bit 2 of RN. The number RN, which can have a value between 0 and 6, defines the memory range used for data aquisition, according to the following table:

RN	memory range
0	256
1	512

		2 1k
		3 2k
		4 4k
		5 8k
		6 16k
11	OFF0	Bit 0 of OFF (offset).
12	OFF1	Bit 1 of OFF
13	OFF2	Bit 2 of OFF
14	OFF3	Bit 3 of OFF. The number OFF, which can have a value between 0 and 15, defines a memory offset in steps of 1k (=1024), where the data are to be stored. This allows to hold several spectra in the card RAM.
15	DIGIO	A value of one sets the DIGITAL I/O pin to low (open drain output). Allows to control external devices.

The default values after a reset are all zero.

Status register
Base + 0

The status register can be read via 16 bit input from the base address. The most bits, but not all, have the same meaning as the corresponding bits in the control register and allow to read the status of the card.

Bit	Name	Meaning
0	PRE REAL	Realtime preset. A value of one means that automatic stop of the measurement is enabled, if the real time counter (channel 0), which must be set to a value of 2^{32} - preset, after overflow, reaches the value zero. Note that the counter is incremented each millisecond.
1	PRE LIFE	Lifetime preset. A value of one means that automatic stop of the measurement is enabled, if the life time counter (channel 1), which must be set to a value of 2^{32} - preset, after overflow, reaches the value zero.
2	RDFLAG	Read flag. This flag is used for the handshake with the card when reading data from the RAM. If it has a value of zero, data can be read, otherwise one has to wait.
3	WMOD	Write mode. This flag must have the same value as RDDIR, before data can be read or written to the RAM.
4	ENC	Enable Conversion. A value of one means that the card is acquiring data, a value of zero means that the measurement is stopped.
5	ANTI	Coincidence mode. A value of one means anticoincidence, i.e. a TTL low (zero) must be applied at the GATE input. A zero means coincidence mode, i.e. a TTL high signal must be at the GATE input.
6	RESET	A value oof one means card reset is on
7	RDDIR	A value of one means writing data to the card RAM is enabled, a zero means reading data is enabled.
8	RN0	Bit 0 of RN (range).
9	RN1	Bit 1 of RN.
10	RN2	Bit 2 of RN. The number RN, which can have a value between 0 and 6, defines the the memory range used for data aquisition, according to the following table:

RN	memory range
0	256
1	512
2	1k
3	2k
4	4k
5	8k
6	16k

11	OFF0	Bit 0 of OFF (offset).
12	OFF1	Bit 1 of OFF
13	OFF2	Bit 2 of OFF
14	OFF3	Bit 3 of OFF. The number OFF, which can have a value between 0 and 15, defines a memory offset in steps of 1k (=1024), where the data are to be stored. This allows to hold several spectra in the card RAM.

15 DIGIO Status of the DIGITAL I/O pin.

DAC 0 register (DAC Output)
Base + 1

The DAC 0 register can be used to control the DAC output voltage at pin 6 of the 9 Pin D-Sub connector, by an 8 bit output to address Base + 1.

DAC 1 register (ZERO)
Base + 3

The DAC1 register is used to control the ZERO level of the ADC, by an 8 bit output to address Base + 3.

DAC 2 register (Lower Level)
Base + 5

The DAC2 register is used to control the Lower Level of the ADC, by an 8 bit output to address Base + 5.

DAC 3 register (Upper Level)
Base + 7

The DAC3 register is used to control the Upper Level of the ADC, by an 8 bit output to address Base + 7.

RAM address register
Base + 2

A 16 bit word output to Base + 2 specifies the RAM address, where data are to be read or written. As double words (32 bit) are stored, the RAM address is the channel number times two. A read or write cycle is always done in blocks of 256 byte, the channel number therefore must be a multiple of 64, the RAM address a multiple of 128 (the last 7 bits are ignored).

RAM write data register
Base + 4

A 16 bit word output to BASE + 4 writes data to the RAM. This must be done in a cycle of writing 128 words, always first the lower word and then the upper word of a channel content.

RAM read data register (lower word)
Base + 4

A 16 bit word input from BASE + 4 reads data from the RAM. This must be done in a cycle of reading 128 words, always first lower word from BASE + 4 and then the upper word from BASE + 6.

RAM read data register (upper word)
Base + 6

A 16 bit word input from BASE + 6 reads data from the RAM. This must be done in a cycle of reading 128 words, always first the lower word from BASE + 4 and then the upper word from BASE + 6.

ADC range register
Base + 6

An 8 bit output to BASE + 6 defines the ADC range (conversion gain) by programming the current source. Depending on the ADC range, the following value rn0 must be sent:

ADC range	rn0
256	0
512	1
1k	2
2k	3
4k	4

8k	5
16	6

6.2. The Subroutines for controlling the MCDLAP

In the following the low level subroutines used by the MCDVGA DOS software for controling the MCDLAP are listed and commented. The program language is C. Variables and constants, if not clear from the context, are discussed if they appear first time.

(Microsoft C; for use with Turbo-C replace:

```
outp    with    outportb     (writes a byte to an I/O port)
outpw   with    outport      (writes a 16 bit word to an I/O port)
inp     with    inportb      (reads a byte from an I/O port)
inpw    with    inport).     (reads a 16 bit word from an I/O port)
```

The following functions are listed to illustrate the hardware programming, but do not form a complete executable program. Note that not all variables and functions are declared, if they are not necessary to understand the hardware programming. Standard Microsoft C functions like fopen, strcpy and so on are not documented.

The complete source code of the DLL DLAP.DLL that controls the hardware via the server program WLAP.exe with example programs for Visual Basic and LabVIEW is available as an option.

MCA_Init

The MCA_Init Routine selects a special MCDLAP and initializes the communication to it.

```
void MCA_Init()
{
  int i,k,rn0;

  i = mcdAR
  i >>= 9;
  k = 0;
  while(i) {i >>= 1; k++;}
  rn=k;
  base = MCA_base[MCD_NUM];
  base1 = base + 1;
  base2 = base + 2;
  base3 = base + 3;
  base4 = base + 4;
  base5 = base + 5;
  base6 = base + 6;
  base7 = base + 7;
  creg = adcoffs<<1;
  if (anti) creg = (creg & 0xffdf) | 0x0020;
  else creg = (creg & 0xffdf);
  creg = (creg & 0xf8ff) | (rn << 8);
  if (demomod == ON)
    return;
  rn0 = rn;
  sreg = inpw(base);
  if (sreg & 0x10) {  /* ENC == 1 , ON */
    Sstate = ON;
    creg = sreg;
    rn = (creg & 0x0700) >> 8;
    mcdAR = 1 << (rn+8);
    adcoffs = (creg & 0x7800) >> 1;
  }
  outpw(base, creg);
  outp(base6, rn0);
  outp(base3,(unsigned char)dac1);
  outp(base5,(unsigned char)dac2);
  outp(base7,(unsigned char)dac3);
}
```

Variable	**Meaning**
MCD_NUM	an integer variable which specifies the selected MCDLAP, 1 ≤ MCD_NUM ≤ 8.
MCA_base	an unsigned integer array containing the base addresses of the MCD/LAPs.
mcdAR	the ADC range (between 256 and 16384).
adcoffs	the offset, where the data are stored in the RAM.
creg	the control register

sreg	the status register
rn	defines the memory range (equivalent to RN described in the register specification)
rn0	defines the adc range, i.e. the current source.

writemem

The writemem Routine writes a block of 256 byte to the MCD/LAP RAM.

```
void writemem(unsigned int adr)
{
  int wmod,rdflg;
  int i,c;
  unsigned *pt;
  int errcnt=1000;
  unsigned a;

  creg = inpw(base);
  creg = (creg & 0xff7f) | 0x80; /* RDDIR = 1 */
  outpw(base, creg);
  sreg = inpw(base);
  wmod = sreg & 0x08;
  while (!wmod) {
    sreg = inpw(base);
    wmod = sreg & 0x08;
    errcnt--; if (errcnt <= 0) goto userbreak;
  }
  a = adr<<1;
  outpw(base2,a);
  pt = (unsigned *)(PDAT32(adr));

  for (i=0; i<64; i++) {
    outpw(base4, *pt++);
    outpw(base4, *pt++);
  }
  return;
userbreak:
  ;
}
```

Note the way it is done: RDDIR is set to one, WMOD is read and checked if it is one, then the address is output to Base+2 and the data are dumped to Base+4.

PDAT32(adr) is a pointer to the data in the computer memory.

readmem

The readmem Routine reads a block of 256 byte from the MCDLAP RAM.

```
int readmem(unsigned int adr)
{
  int wmod,rdflg;
  int i,c,cnt;
  unsigned *pt;
  unsigned a;

  creg = inpw(base);
  creg = (creg & 0xff7f); /* RDDIR = 0 */
  outpw(base, creg);
  sreg = inpw(base);
  wmod = sreg & 0x08;
  cnt=1000;
  while (wmod) {
    sreg = inpw(base);
    wmod = sreg & 0x08;
    if (--cnt <= 0) goto userbreak;
  }
  adr &= 0xFFC0;
  a = adr<<1;
  outpw(base2,a);
  pt = (unsigned *)(PDAT32(adr));
  for (i=0; i<64; i++) {
    sreg = inpw(base);
    rdflg = sreg & 4;
    cnt=1000;
```

```
    while (rdflg) {
      sreg = inpw(base);
      rdflg = sreg & 4;
      if (--cnt <= 0) goto userbreak;
    }
    *pt++ = inpw(base4);
    *pt++ = inpw(base6);
  }
  pt = (unsigned *)PDAT32(adr);
  *pt++ = inpw(base4);
  *pt = inpw(base6);
  return TRUE;
userbreak:
  return FALSE;
}
```

Note the way it is done: RDDIR is set to zero, WMOD is read and checked if it is zero, then the address is output to Base+2. In a loop always the rdflag is checked and the data are read from Base + 4 (lower word) and Base + 6 (upper word).

Note that at the end the contents of the first channel is read again since the RAM output register may contain wrong data when the loop is started.

MCA_Newtime

MCA_Newtime routine sets the real- and lifetime counters.

```
void MCA_Newtime(unsigned long rtim, unsigned long ltim)
{
  if (demomod == ON)
    return;

  readmem(0);
  DAT32(0) = rtim; /* Realtime */
  DAT32(1) = ltim; /* Livetime */
  writemem(0);
}
```

The time counters are in channel zero and one; therefore the first block of MCD/LAP RAM is read, modified, and rewritten.

DAT32(adr) specifies the data in the computer memory.

MCA_Start

The MCA_Start routine sets all parameters, clears the time counters and starts a measurement.

```
void MCA_Start()
{
  if (demomod == ON)
    return;
  MCA_Init();
  if(ltbpreset==2) {
    creg = ((creg & 0xfff6) | 0x0001);
    outpw(base, creg);
    MCA_Newtime((long)(-rtpreset*1000L), 0L);
  }
  else if(ltbpreset==1) {
    creg = ((creg & 0xfff6) | 0x0002);
    outpw(base, creg);
    MCA_Newtime(0L, (long)(-ltapreset*1000L));
  }
  else {
    creg = (creg & 0xfff6);
    outpw(base, creg);
    MCA_Newtime(0L,0L);
  }

  creg = (creg & 0xffef) | 0x10; /* ENC = 1 */
  outpw(base, creg);
  Sstate=ON;
}
```

ltbpreset defines the preset type:

value	meaning
0	no preset
1	life time preset
2	real time preset.

rtpreset is the realtime preset in seconds,
ltapreset is the lifetime preset.

MCA_Stop

The MCA_Stop routine stops the measurement.

```
void MCA_Stop()
{
  if (demomod == ON)
    return;
  creg = (creg & 0xffef); /* ENC = 0 */
  outpw(base, creg);
  readmem(0);
  if(ltbpreset==2) DAT32(0) += rtpreset*1000;
  if(ltbpreset==1) DAT32(1) += ltapreset*1000;
  writemem(0);

  Sstate=HALT;
}
```

Note that in case of a preset the time counters are corrected to show the time of the measurement.

MCA_Continue

MCA_Continue routine continues a measurement, leaving all parameters and the time counters at their present values.

```
void MCA_Continue()
{
  if (demomod == ON)
    return;
  if(ltbpreset==2) {
    readmem(0);
    if(DAT32(0) < rtpreset*1000)
      MCA_Newtime((long)DAT32(0) - (long)(rtpreset*1000L), 0L);
    else {
      warnpiep("Preset reached");
      return;
    }
    creg = ((creg & 0xfff6) | 0x0001);
    outpw(base, creg);
  }
  else if(ltbpreset==1) {
    readmem(0);
    if (DAT32(1) < ltapreset*1000)
      MCA_Newtime(0L, (long)DAT32(1) - (long)(ltapreset*1000L));
    else {
      warnpiep("Preset reached");
      return;
    }
    creg = ((creg & 0xfff6) | 0x0002);
    outpw(base, creg);
  }
  else {
    creg = (creg & 0xfff6);
    outpw(base, creg);
  }

  creg = (creg & 0xffef) | 0x10; /* ENC = 1 */
  outpw(base, creg);
  Sstate=ON;
}
```

Of course the timers have to be corrected for the preset values to ensure the stop if a preset is reached.

MCA_find

The MCA_find routine detects the installed MCDLAP's.

```
void MCA_find()
{
  base = MCA_base[MCD_NUM];
  base1 = base+1;
  base2 = base+2;
  base3 = base+3;
  base4 = base+4;
  base5 = base+5;
  base6 = base+6;
  base7 = base+7;
  if (demomod == ON)
    return;
  if (readmem(0)) ADCinstalled[MCD_NUM]=1;
}
```

adcinst

The adcinst routine detects the connected MCDLAP's and initialises it.

```
void adcinst()
{
  int i;
  int flag = 1;
  int help = 0;

  for (i=1; i<MAX_MCDPC; i++) {
    MCD_NUM = i;
    takecnf();
    MCA_find();
    if (ADCinstalled[i]) {
      printf(
      "\nMCD/LAP Nr. %d with Base Adress %x    found", i,
      MCA_base[i]);
      if (flag) {
        flag = 0;
        help = MCD_NUM;
                  /* note first found MCDLAP */
      }
    }
  }
  if (flag) {    /* no MCDLAP found */
    demomod = ON;
    printf("\nNo hardware found ---> Demomode\n");
  }
  MCD_NUM = help;
  takecnf();
  MCA_Init();
}
```

readportadress

The readportadress routine reads the base addresses from a file PORT.MCD.

```
int readportadress()
{
  int i;
  FILE *stream;
  char buff[80];

  if (stream = fopen("PORT.MCD", "r+t")) {
    for (i=1; i<MAX_MCDPC; i++) {
      if(!freadstr(stream,buff)) {
        sscanf(buff, "%x",&MCA_base[i]);
      }
      else break;
    }
    fclose(stream);
    return TRUE;
  }
  else
```

```
  return FALSE;
}
```

7. Technical Data

ADC:	Type: .... 100MHz Wilkinson
	Conversion Gain: 256, 512, 1k, 2k, 4k, 8k, 16k
	Conversion Time: .... 1µs + (0.01µs * channel address)
	.... independent of digital offset
	Dead Time: .... t_{rise} + 1µs + MAX (0.01µs * channel address ; t_{decay})
	.... independent of digital offset
	Integral Nonlinearity: .... ±0.05 % of full scale over top 99 % of range
	Differential Nonlinearity: .... ±1 % of full scale over top 99 % of range
	Stability: .... gain: 50ppm/°C
	.... zero: 50µV/°C
	Count Rate Shift: .... ±0.5 channels at rates up to 50k events / s
MCD:	Memory: .... 32k x 16bit
	Organization: .... 16k channels x 32bit
	Average Storage Time: 430ns
Power Requirements:	Computer Power Supply: .... +5V, 1400mA, 7W typ.
	.... 1600mA, 8W max.
Physical:	Size: .... 2/3 length low profile PC-card, ≈22 x 11cm
	Weight: 0.3kg