WebMCA

Multichannel Data Processor

MCDWIN User Manual

Grünwalder Weg 28a, 82041 Oberhaching, Germany
Tel ++49 89 665180 50; FAX ++49 89 665180 40

Version 1.0, March 6, 2006

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 webMCA is an Ethernet based, high performance Multi Channel Analyzer (MCA), interfacing NIM ADC's to local networks as well as to the internet. Existing and new NIM based electronics are easily accessible through the Ethernet. The webMCA is a standalone MCA in cigarette box form with real time and live time measurement, various presets and multiple spectra storage. It contains a large spectrum memory, a battery powered real time clock.

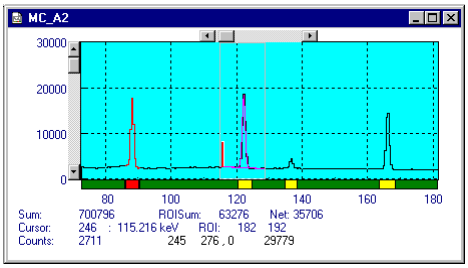
The webMCA is a compelling RISC based computer, powered by Motorola's Coldfire controller. Embedded Linux, a stable, robust operating system contributes to ist superior performance. The XML transfer protocol offers access and data transfer to and from all operating systems.

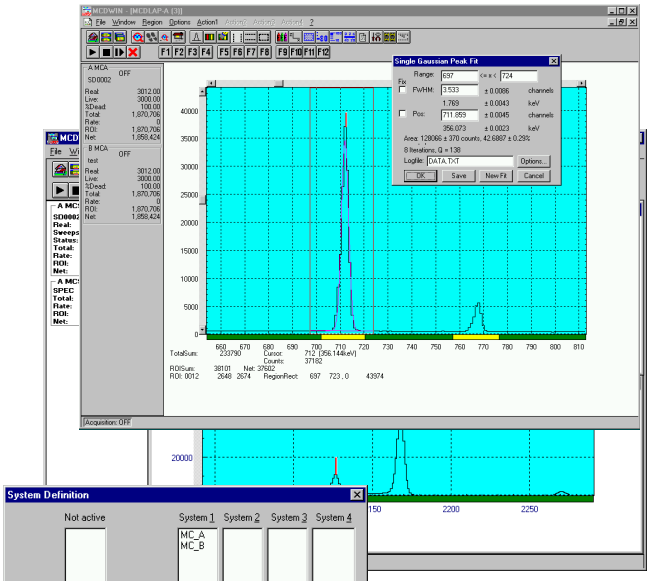
The webMCA is ideal for remote monitoring applications in power plants, laboratories, industry, universities and research centers. The novel power over Ethernet, PoEIEEE802.3af, connects and powers as easy as USB, however, providing the speed, flexibility and connectivity of the Ethernet, by far outperforming USB.

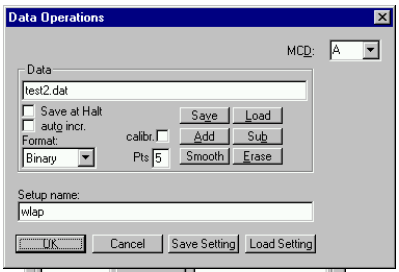
The webMCA offers Ethernet for virtually unlimited expansion: There is no restriction for the number of devices which can be addressed. There is no distance limit on remote operation.

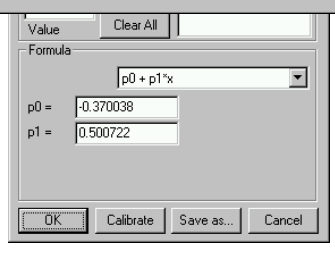
The sophisticated WINDOWS based control and analysis software MCDWIN ensures quick learning and easy usage.

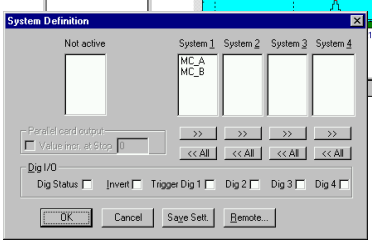
Some of MCDWIN's features are high resolution graphics displays with zoom, linear and logarithmic (auto)scaling, grids, ROIs, Gaussian fit, calibration using diverse formulas. and FWHM calculations. Macro generation using the powerfull command language allows task oriented batch processing and selfrunning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











2. Installation Procedure

2.1. Hard- and Software Requirements

The following items are required using a webMCA:

- Power over Ethernet adapter (DWL-P100 included with the webMCA)
- 10+ Mbit network connection
    - A PC with Microsoft Windows 98 / ME / 2000 / XP installed is required for use of the supplied control and analysis software MCDWIN.

2.2. Hardware Installation

The webMCA has to be connected to the local network either with a Power over Ethernet (PoE) adapter DWL-P100 or a PoE switch using a +48V power supply for the PoE device (optional).

The desired spectroscopy ADC has to be connected to the 25-pin connector of the webMCA using the provided cable.

Figure 2.1: Connecting the webMCA

After connecting to the PoE device the webMCA needs about 30 sec for initialization. The “MCA” LED indicates when the webMCA can be accessed.

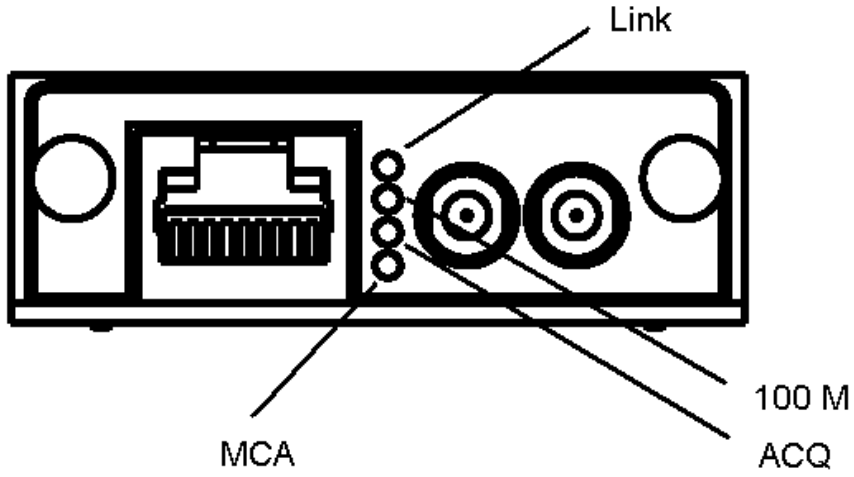


Figure 2.2: LED panel of the webMCA

The “ACQ” LED signals the acquisition status of the device. “LINK” and “100M” correspond to the network interface where “100M” indicates a 100 Mbit connection.

2.3. Software Installation

To install the MCDWIN software on your hard disk insert the MCDWIN CD and start the installation program file by double clicking

SETUP

A directory called C:\WEBMCA is created on the hard disk and all webMCA and MCDWIN files are transferred to this directory. Drive C: is taken as default drive and \WEBMCA as default directory. It is not mandatory that the webMCA operating software is located in this directory. You may specify another directory during the setup.

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

To run the WebMCA software, simply double click on the “WEBMCA“ icon. To close it, close the WebMCA server in the Taskbar.

2.4. Network Configuration

The default IP address of a webMCA is 192.168.1.2. To change this address the “WebMCA Config” utility has to be used. It can be found using the start button in the program folder "FAST ComTec WebMCA"

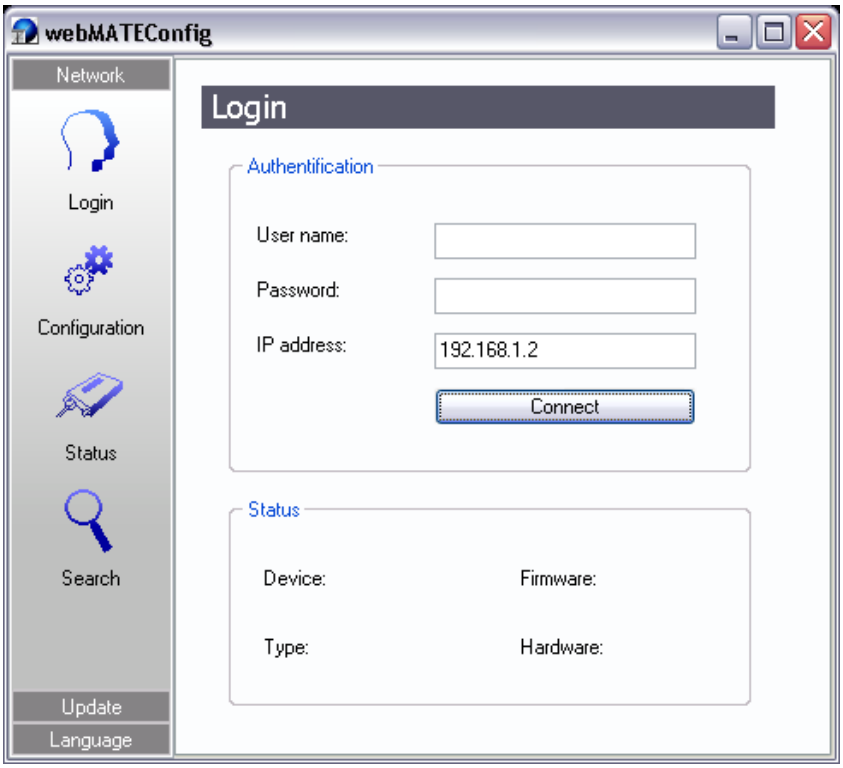


Figure 2.3: WebMCA Config utility

The entries for user name and password in the login dialog are not evaluated currently and can be left blank. After pushing the "Connect" button the firmware and hardware revision will be displayed on success.

Selecting the “Configuration” page several network related properties can be changed.

The IP address should be set to match the used intranet. To allow a time synchronization a NTP server IP address can be specified as well as a poll interval for the NTP requests.

The webMCA uses UTC time internally. When spectra are transferred to the winTMCA32 to start and stop time are adjusted to the local time of the PC with respect to the time zone and daylight saving settings.

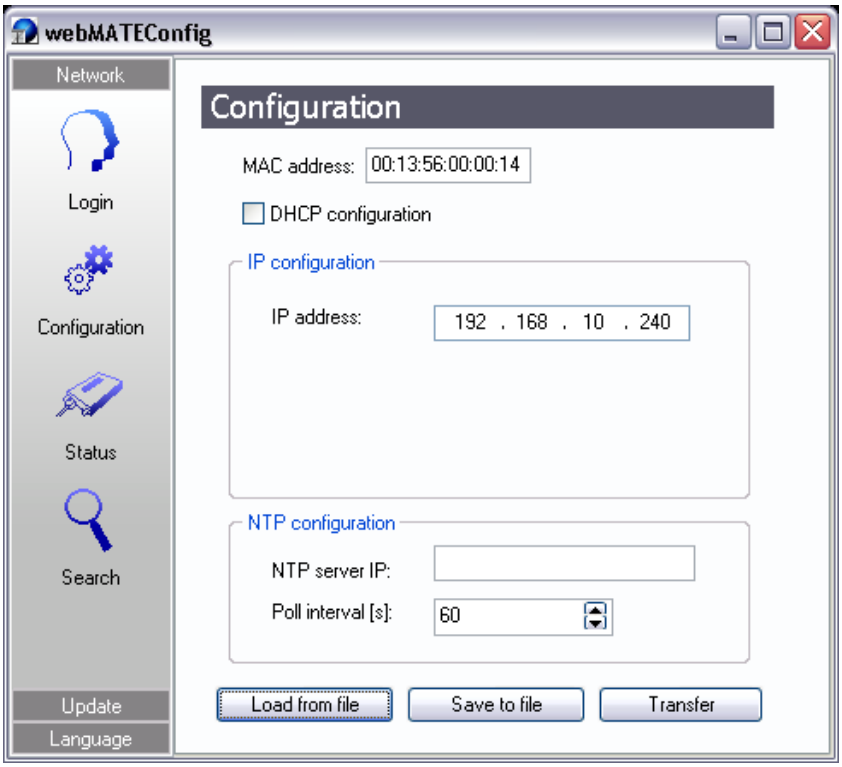


Figure 2.4: Network configuration dialog

To transmit these parameters to the webMCA the “Configure” button has to be pressed. After the transmission is done you will be asked to reboot the webMCA in order to accept these settings.

The “Status” page of the configuration tool can be used to check the device list and the network configuration of the webMCA.

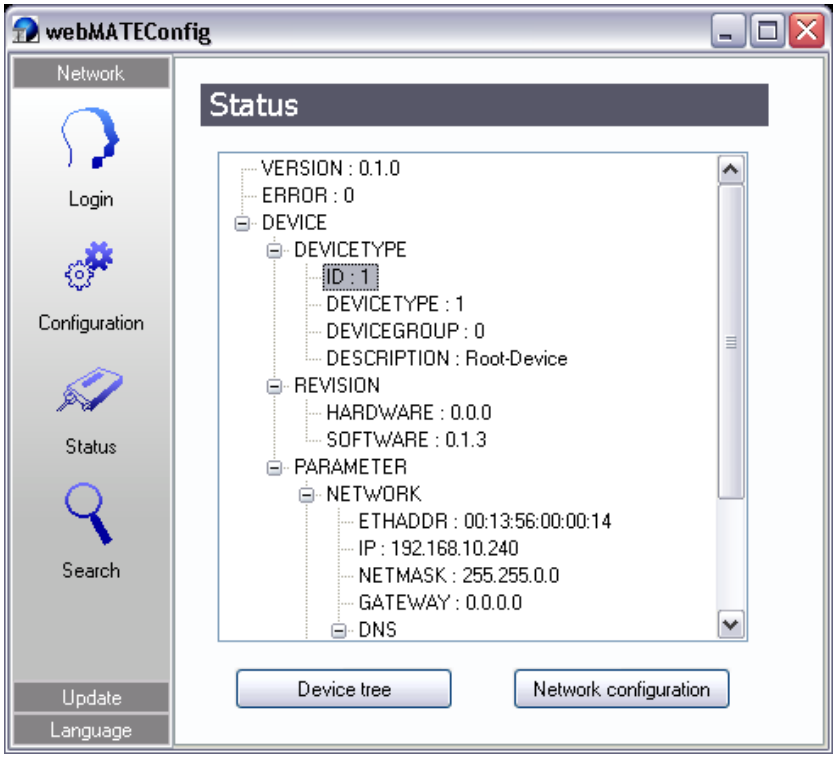


Figure 2.5: Status display

On the “Search” page it is possible to scan for all available devices in the current subnet. All found devices will be listed with their IP address and their MAC address.

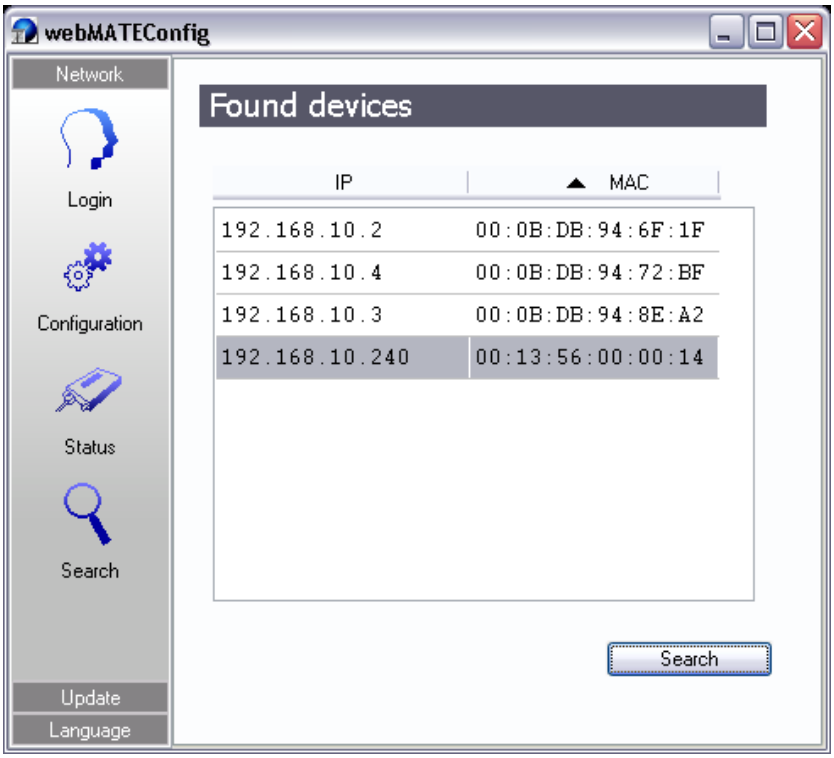


Figure 2.6: Scan result

2.5. Updating the Firmware

On the page “Update” a dialog to update the firmware of the webMCA is available.

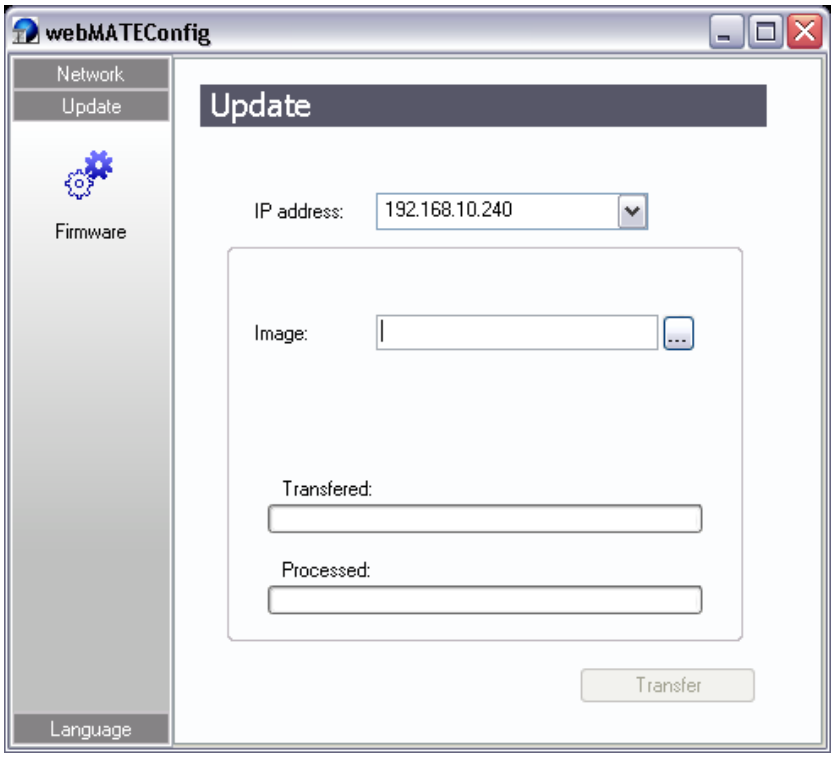


Figure 2.7: Update dialog

The IP-address has to be selected from the selection box. If the list is empty a network scan has to be done (s.a.). After selecting a firmware image file (“local*.bin”) the image will be transferred pushing the “Transfer” button. The transfer state is displayed by the progress bar.

On success the data will be stored in the flash memory of the webMCA. This will be indicated by the progress bar below “Processed:”. If the storage is done the following message box will appear:

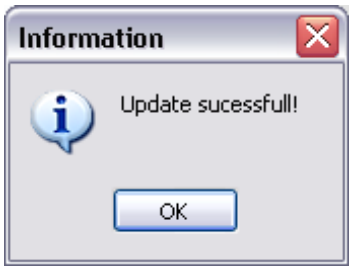


Figure 2.8: Update successfull

The webMCA will restart automatically.

3. Hardware Description

3.1. Signal description of the 25-pin ADC connector

Sub-D male pin layout

Pin	Direction	Mnemonic	Signal
1-12	In	D[0-12]	Inverted Data D0-D12
13	In	*DRDY	Inverted Data Ready
14,15	In	D13,D14	Inverted Data D13, D14
17	Out	*ACC	Data Accepted
18	Out	*STOP	Stop ADC
19	In	D15	Inverted Data D15
20	In	INH	Inhibit, Drop Data
21	In	DTIME	Dead Time
22	Out	*DEN	Inverted Data Bus Enable
23	Out	DWELL	Dwell
24	Pwr	GND	Signal Ground
25	-	E5V	Do not connect

Signal Description

*D0-*D15 16 Bit inverted ADC data input.

Datalines are terminated with 2K2 pull-up Resistor against 5 Volt internal webMCA supply. Therefore, inactive line state is approximate 5 Volt. Signal level is inverted, i.e. logic level 0xffff means ADC signal value 0.

*DRDY Data Ready handshake signal from ADC.

Signal level inverted, active low asserted. If data on D[0-15] are setup the ADC should assert this signal to denote valid data available. DRDY must be negated immediately after *ACC is asserted.

*ACC Data Accepted handshake from webMCA.

Signal level inverted, active low asserted. If the ADC asserts DRDY, after latching the data the webMCA asserts this Signal to denote data taken. The webMCA will not respond with ACC if the internal FIFO buffer signals full state. The data transfer is pending until FIFO buffer is available. ACC will immediately negate after DRDY is negated.

INH Data Inhibit from ADC. If INH is sampled asserted high the current ADC data will be ignored. The transfer cycle handshake proceeds.

*DEN Data Bus enable output. Always asserted low, will not be switched.

DWELL Dwell output to an MCS device, not used for the ADC interface. High asserted signals to switch to the next bin.

DTIME Deadtime signal from ADC. If asserted high stops counting the Livetime.

GND Signal Ground.

E5V Do not connect, provides test voltage to external test circuit.

Timing diagram

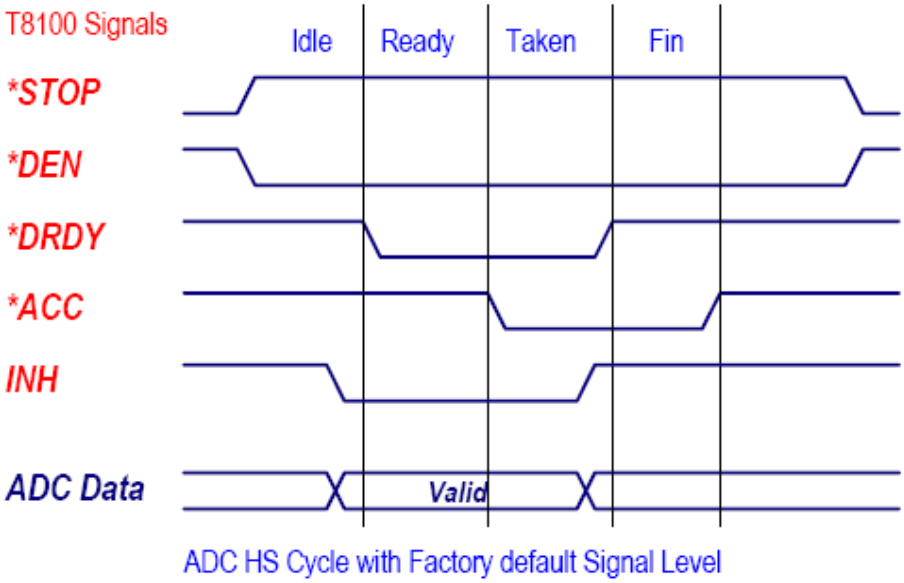


Figure 3.1: Timing of the handshake signals

3.2. Jumper settings for the FAST ComTec 7070 ADC

To use the webMCA with the 7070 ADC the jumpers of the 7070 must be set as follows:

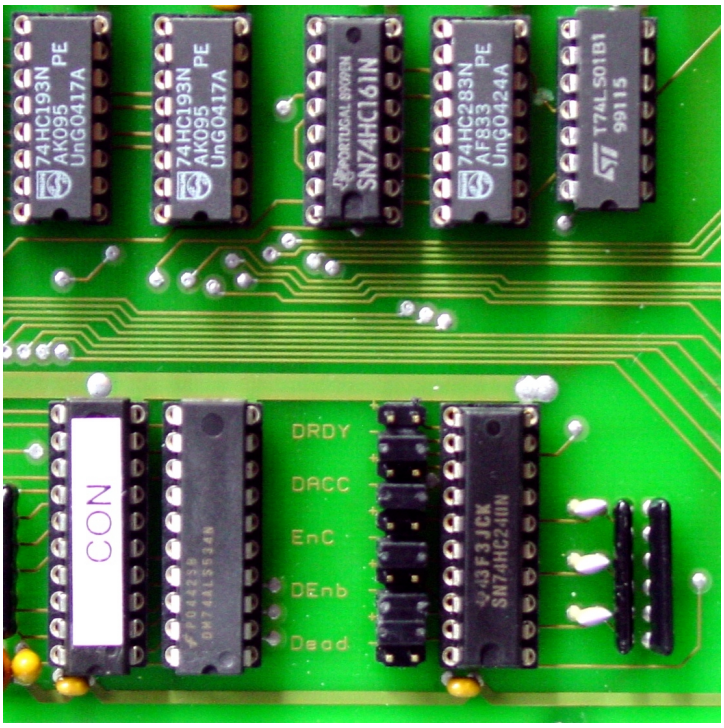


Figure 3.2: Jumper setting for FAST ComTec 7070 ADC

3.3. Network parameters

For the communication with the webMCA the following IP protocols and ports are used:

Kind of communication	**Protocol**	**Port**
Standard communication	TCP	6668
Network broadcast	UPD	9176-9178
Update	TCP	9176
NTP time synchronization	SNTP	123

4. Windows Server Program

The window of the WebMCA server program WEBMCA.EXE is shown here. It provides the full control of the webMCA 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.

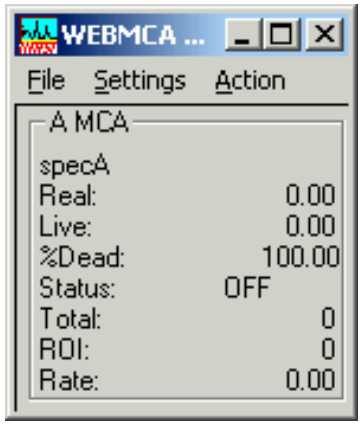


Figure 4.1: WEBMCA Server program

4.1.1. WEBMCA.INI file

At program start the configuration files WEBMCA.INI (contains - for example - the number of webMCA modules and IP address in a format ip=address; see Figure 4.2) and WEBMCAA.CFG are loaded. Instead of this WEBMCAA.CFG file any other setup file can be used if its name - excluding the appendix ‘A.CFG’ - is used as command line parameter (e.g. WEBMCA TEST to load TESTA.CFG) . The server program is normally shown inconized in the task bar. After a double click it is opened to show the status window.

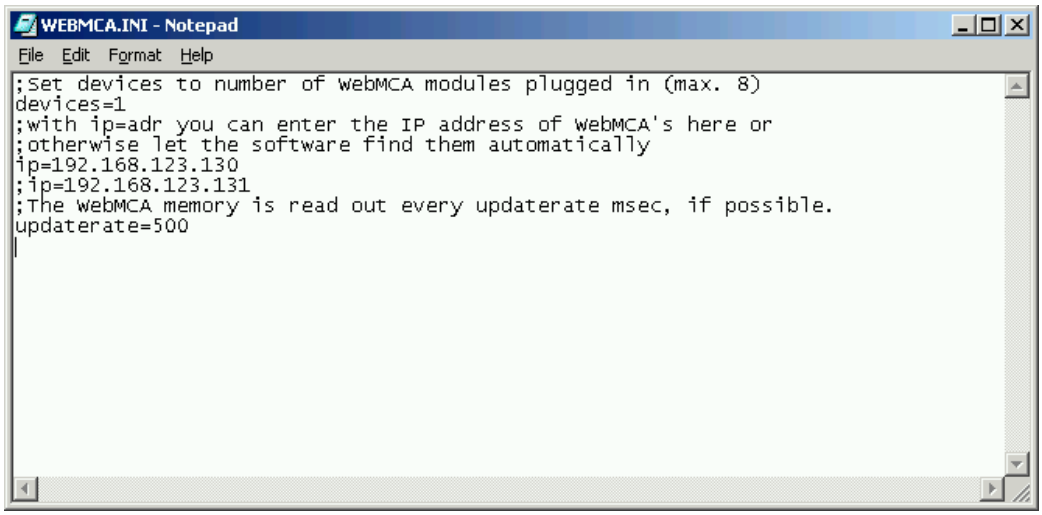


Figure 4.2: Sample WEBMCA.INI file

In the following the several dialogs are described in detail:

4.1.2. Data Operations

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

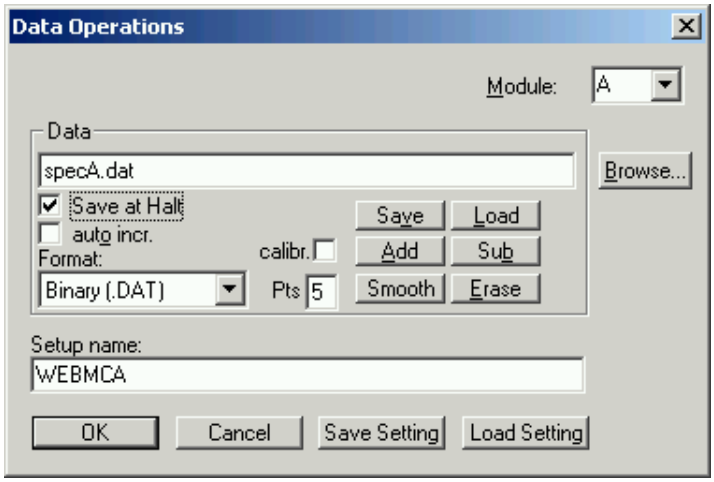


Figure 4.3: Data Operations dialog box

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.

4.1.3. MCA Settings

Clicking in the Settings menu on Hardware... item opens the MCA Settings dialog box. Here parameters like presets, range parameters, etc. for the webMCA module can be set.

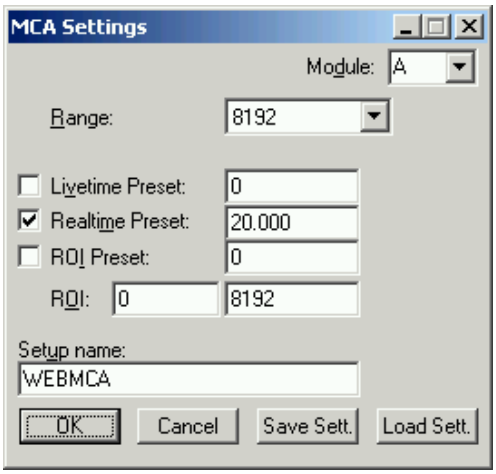


Figure 4.4: MCA Settings dialog box

With the combo box „**Range**“ the length of the spectrum can be chosen. 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 „**RealTime Preset**“ or livetime via the „**Livetime Preset**“. **OK** takes all settings. **Cancel** cancels all changes. Pressing **Save Sett.** writes all settings in the file **WEBCAA.CFG** or the name entered in the **Setup name** edit field with "A.CFG" appended.

4.1.4. System Settings

If more than one MCA is used, the system definition dialog box comes up as shown in Figure 4.5. Here the several units can be combined to form up to 4 separate systems that can be started, stopped and erased by one command.

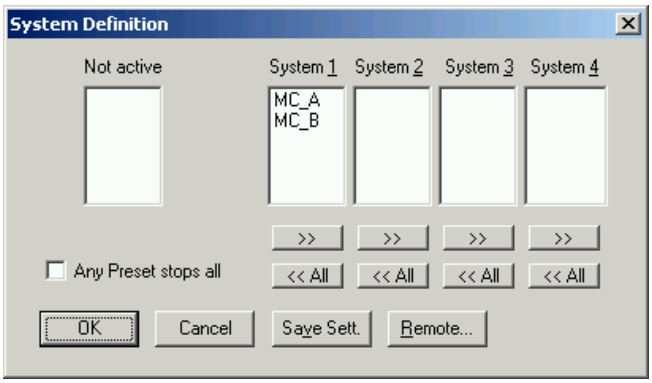


Figure 4.5: System Definition dialog box, two webMCAs

In the shown setting a single system is formed. The two webMCAs 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.

4.1.5. Config file

Clicking „**Save Settings**“ stores all settings in the file **WEBMCAA.CFG** in a form:

```
sysdef=0
range=8192
rtpreset=20.000
rtprena=1
ltpreset=0
ltprena=0
roimin=0
roimax=8192
roipreset=0
roiprena=0
autoinc=0
datname=specA.dat
savedata=0
fmt=dat
smoothpts=5
caluse=0
```

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 .MCD 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.

4.1.6. Remote mode

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 WEBMCA server program via a serial port. An optional available software expansion MCDLAN is necessary, that allows also the control over a LAN connection.

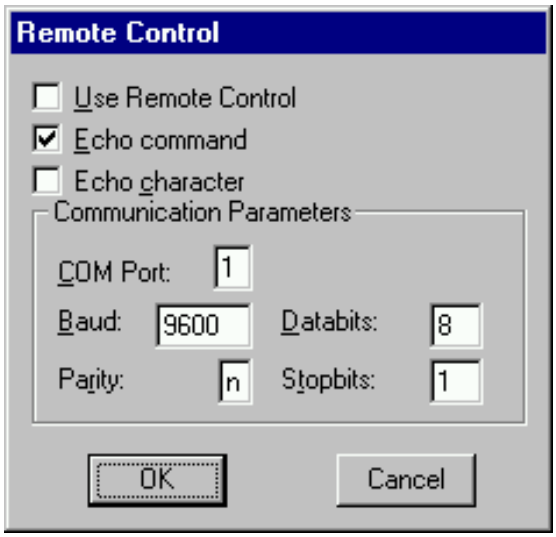


Figure 4.6: Remote Control dialog box

If the Checkbox „Use Remote Control“ is marked and the MCACOM.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.

4.2. Control Language

A sequence of commands that are stored in a file with extension .CTL can be executed by the WEBMCA server program or MCDWIN with the „Load “ command. Also the configuration files WEBMCAA.CFG or the header files with extension .MCD ontain 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 MCACOM.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 4.3. The application name for opening the DDE connection with the standard WEBMCA server program WEBMCA.EXE is WEBMCA, the topic is WEBMCA-. 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 that the remote shell daemon program MCANET 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 WEBMCAA.CFG contains a complete list of commands for setting parameters; an example is:

```
range=4096          ; Memory size for spectra
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
roimin=0            ; ROI lower limit (inclusive)
roimax=2048         ; 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)
smoothpts=5         ; Number of points to average for a smooth operation
```

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

```
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
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=2048 ; subset of parameters as in a WEBMCAA.CFG file...

rtpreset=1000

rtprena=0

ltpreset=1000

ltprena=0

roipreset=0

roiprena=0

roimin=0

roimax=2048

autoinc=0

datname=C:\MCA\DATA\SD0002.mcd

savedata=1

fmt=dat

caluse=1 ; Use Energy calibration

calch00=1172.00 ; Calibration points..

calvl00=1173.264000

calch01=1331.00

calvl01=1332.500000

caloff=-0.506315 ; Calibration formula polynomial coefficients..

calfact=1.000750

calfact2=0

calfact3=0

calunit=keV ; Calibration unit

roi=1156 1199 ; ROI limits..

peak=1173.230000 ; calibration peak in this ROI

roi=1323 1353

peak=1332.480000 :

The following commands perform actions and therefore usually are not included in a WEBMCAx.CFG or .MCD 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.

halt ; Stops acquisition of system 1 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).

savecnf ; Writes the settings into MCAA.CFG

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.

exit ; Exits the MCA.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).

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.
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 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.
?	; 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
ADC?	; Sends the AXAS Status
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.

4.3. Controlling the WEBMCA Server via DDE

The WEBMCA 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.

4.3.1. Open Conversation

application: WEBMCA
topic: WEBMCA-

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 WEBMCA and the topic is WEBMCA-.

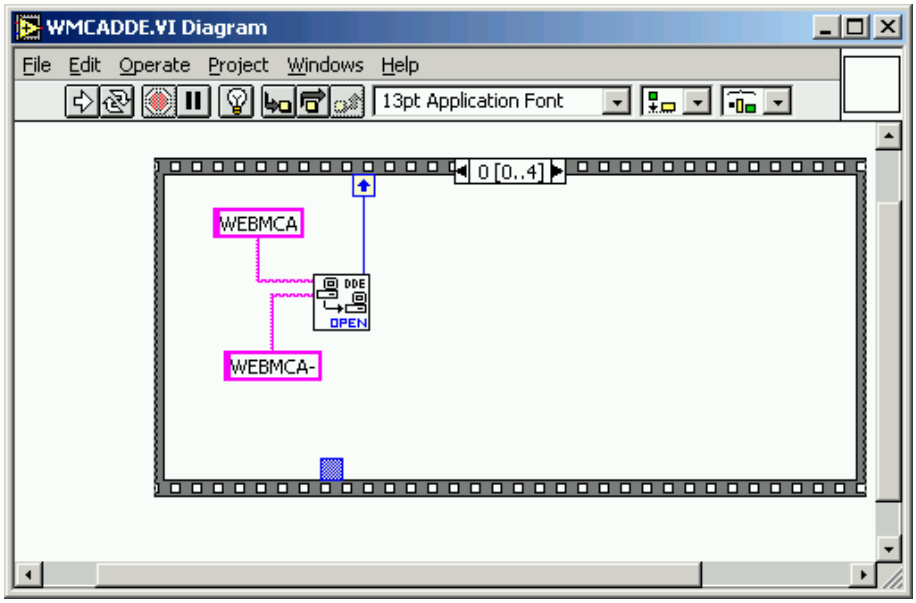


Figure 4.7: Opening the DDE conversation with the WEBMCA server in LabVIEW

4.3.2. DDE Execute

The DDE Execute command can be used to perform any action of the WEBMCA program. Any of the Control command lines described in chapter 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 WEBMCA 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.

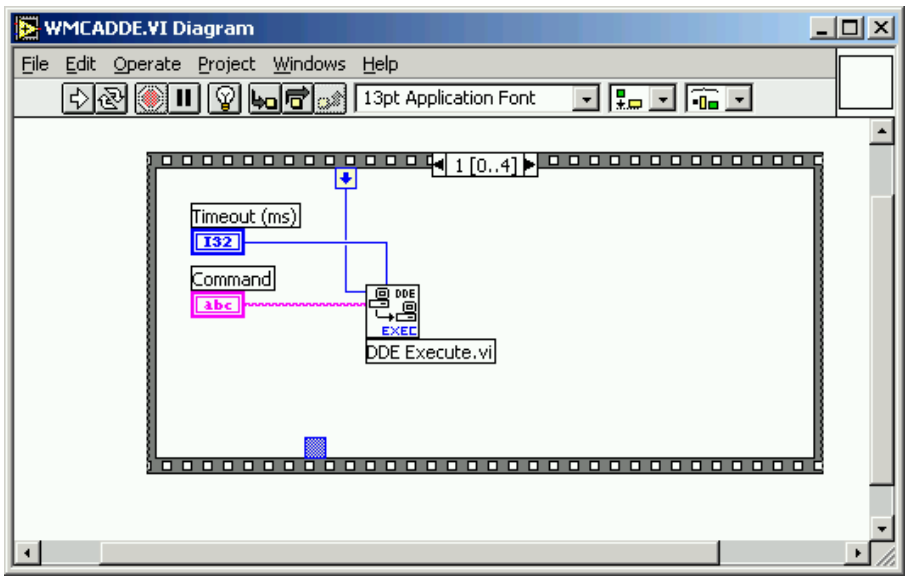


Figure 4.8: Executing a control command from a LabVIEW application

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 WEBMCA 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 (if more than 1).

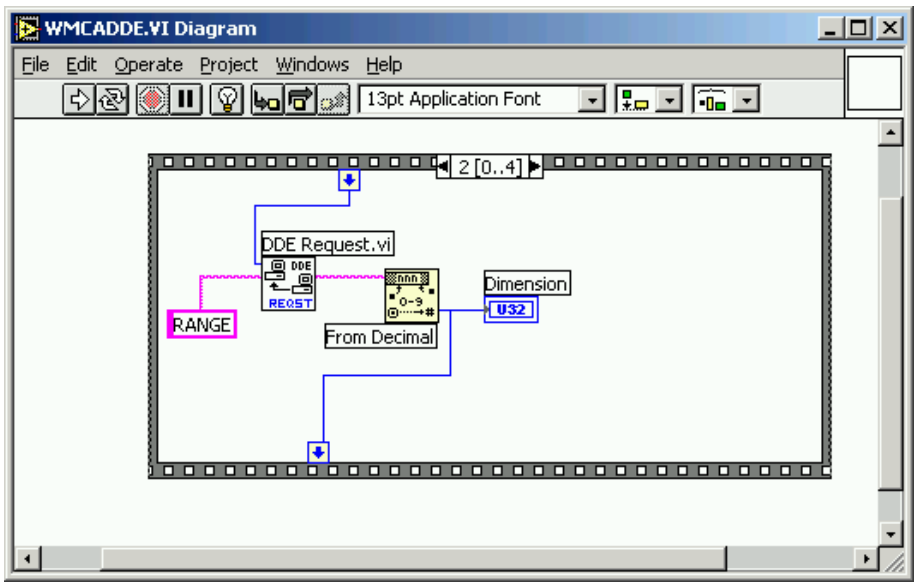


Figure 4.9: Getting the total number of data with LabVIEW

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.

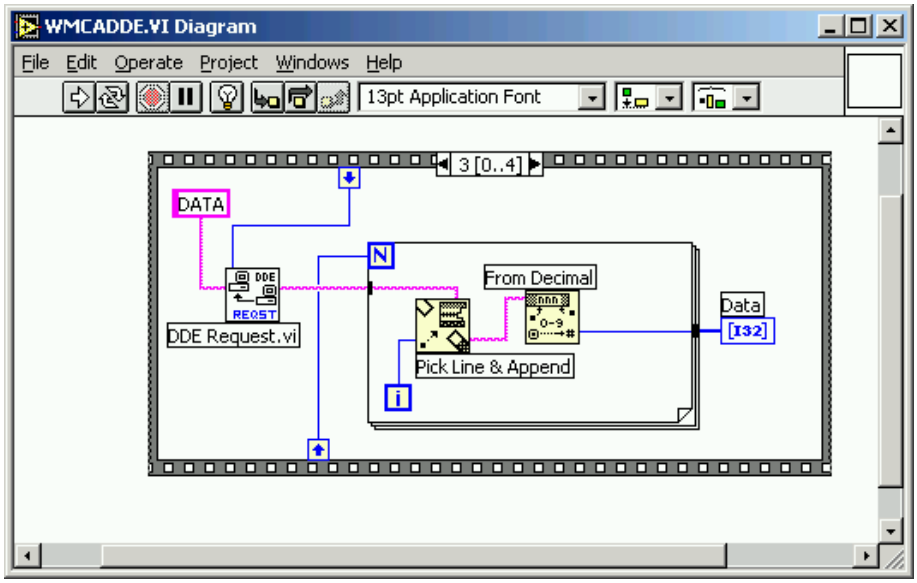


Figure 4.10: Getting the data with LabVIEW

4.3.4. Close Conversation

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

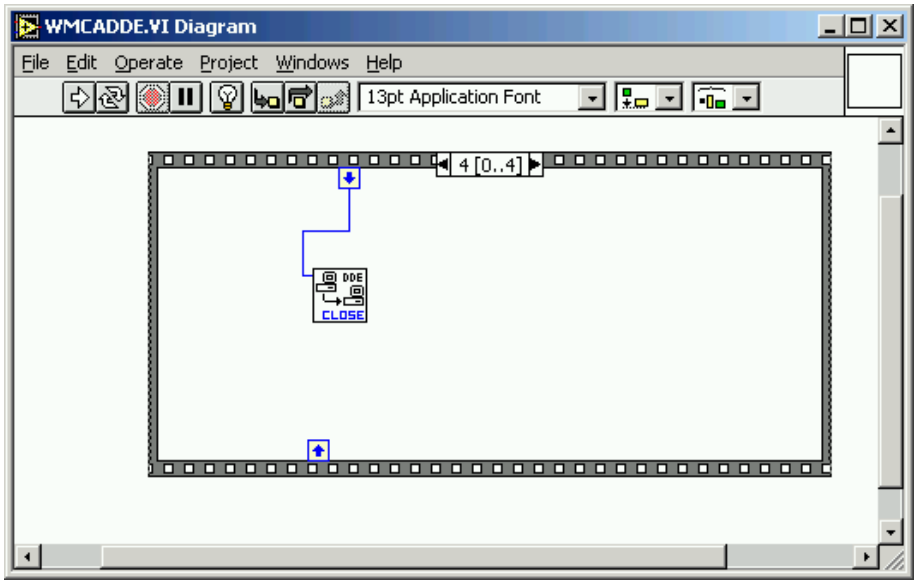


Figure 4.11: Closing the DDE communication in LabVIEW

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

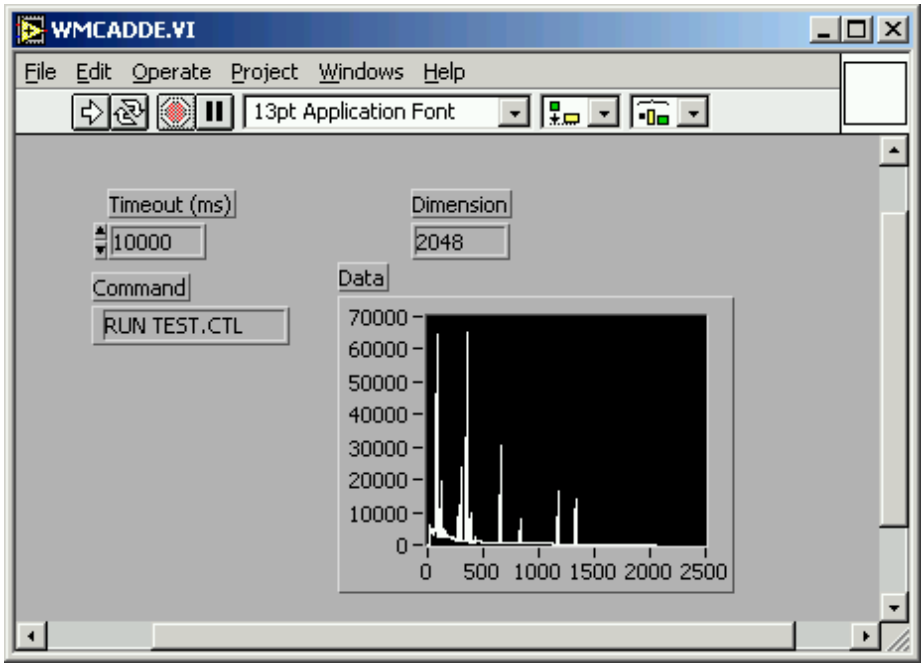


Figure 4.12: Control Panel of the demo VI for LabVIEW

4.4. Controlling the WEBMCA Server via DLL

The WEBMCA 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 DMCA.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, 2 == GANAAS
  int autoinc;           // 1 if auto increment filename
  int diguse;            // (for future use) Optional Parallel Port:
                        // bit 1: Dig I/O Status
                        // bit 2: Dig I/O Trigger
                        // bit 3: Dig I/O Invert polarity
                        // bit 4: Trigger System2
                        // bit 5: Trigger System3
                        // bit 6: Trigger System4
                        // bit 7: Clear before triggered start
  int ltprena;           // 1 if livetime preset enabled, 0 else
  int outlevel;          // (for future use) DAC output level, 0...255
  int adcrange;          // (for future use) ADC range
  int offset;            // (for future use) ADC offset
  int lowlevel;          // (for future use) ADC lower level limit
  int uplevel;           // (for future use) ADC upper level limit
  int dtlevel;           // (for future use) ADC DT level
  double ltpreset;       // live time preset
  int nregions;          // number of regions
  int caluse;            // use calibration
  double scalpreset;     // dummy
  int active;            // 0 if disabled, 1..4 if active in system 1..4
  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) ***/
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
                                          // Store System Definition
                                          // into 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 MCA.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);
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
        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 webMCA via the server program to perform measurements and save data, and shows the data on-line in several windows.

The server program WEBMCA.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.

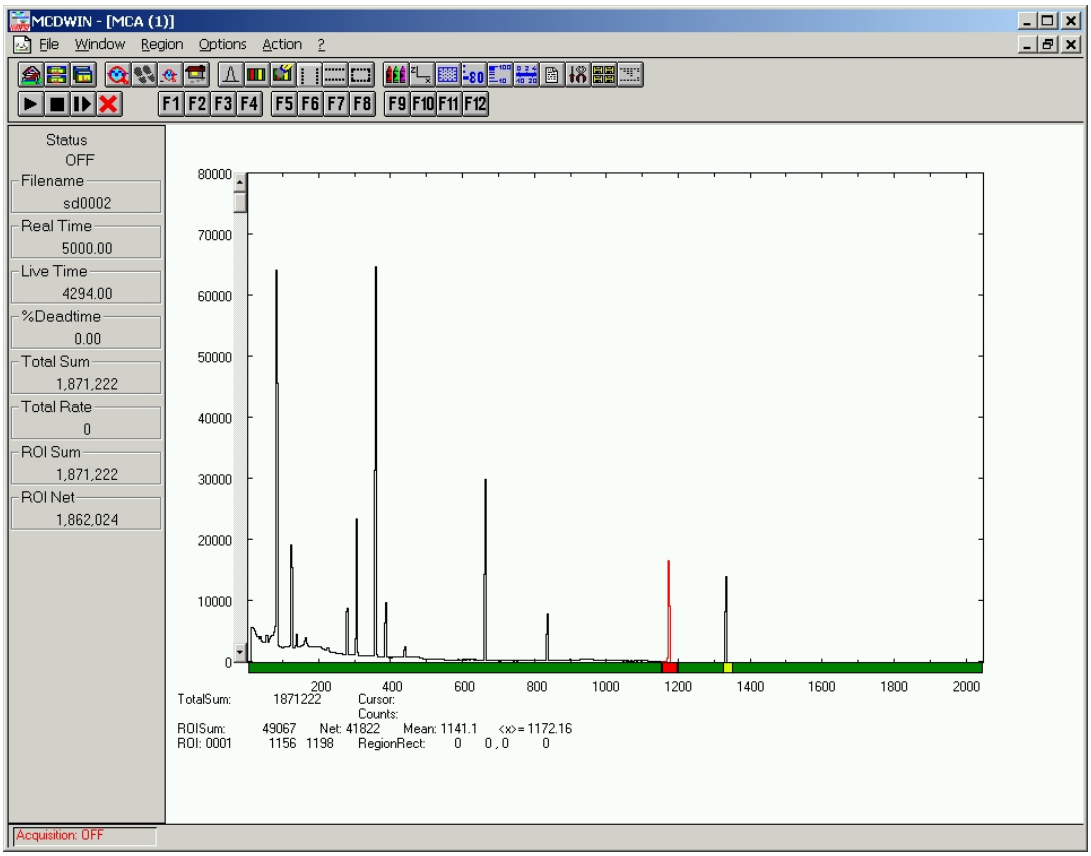


Figure 5.1: MCDWIN main window

A status window at the left side gives all information about the status of the MCA. 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.

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.

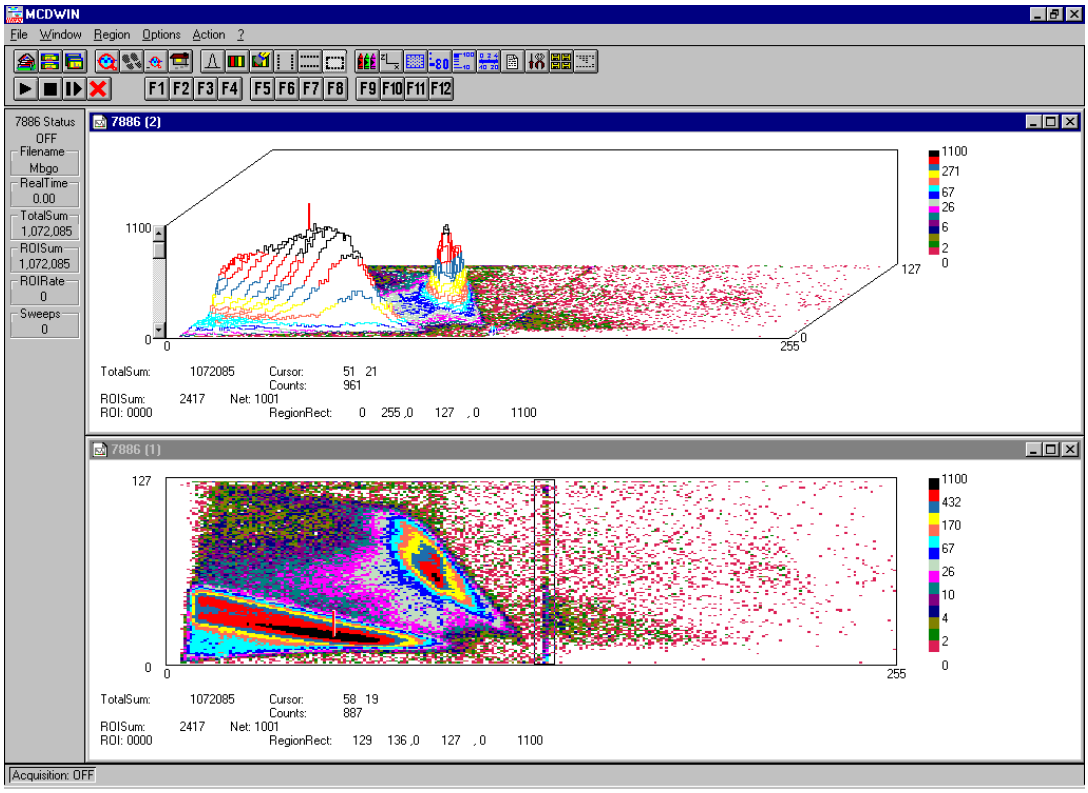


Figure 5.2: MCDWIN Map and Isometric display

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 .MCD). 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 MCAs are installed by writing a line devices=2 into the MCA.INI file, in the „**Open New Display**“ dialog box the MCA for the new display can be selected.

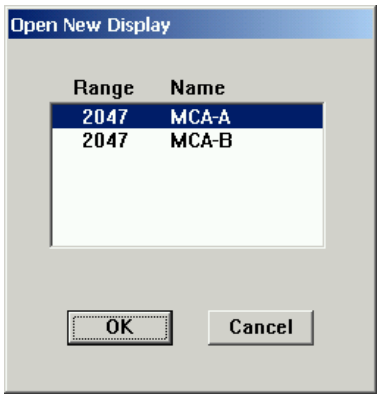


Figure 5.3: File New Display dialog box

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 webMCA devices defined in the WEBMCA.INI file also if only one webMCA is physically available. The software can run for the second webMCA in demo mode.

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.

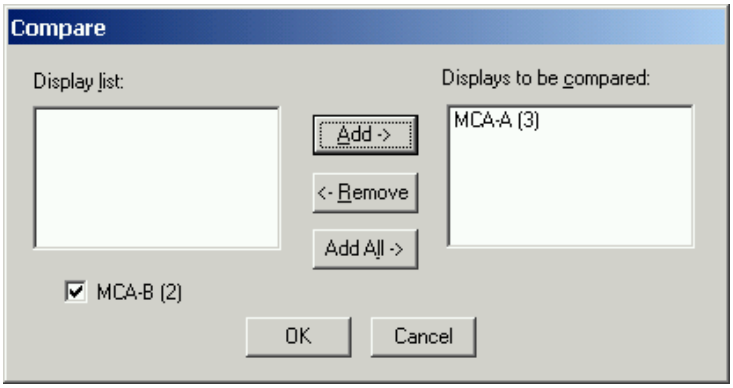


Figure 5.4: Compare dialog box

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 9x 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). For Windows 2000/XP these restrictions do not apply.

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.

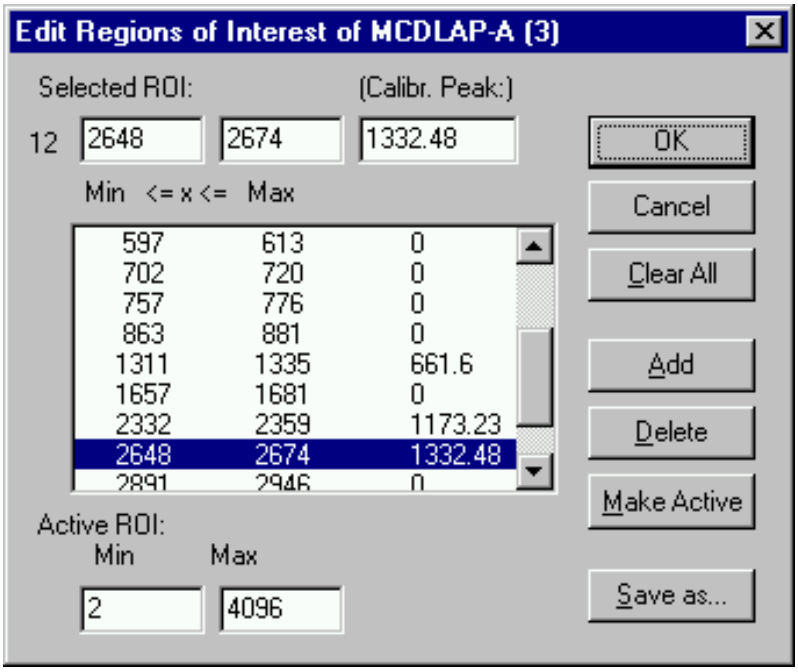


Figure 5.5: ROI Editing dialog box

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:

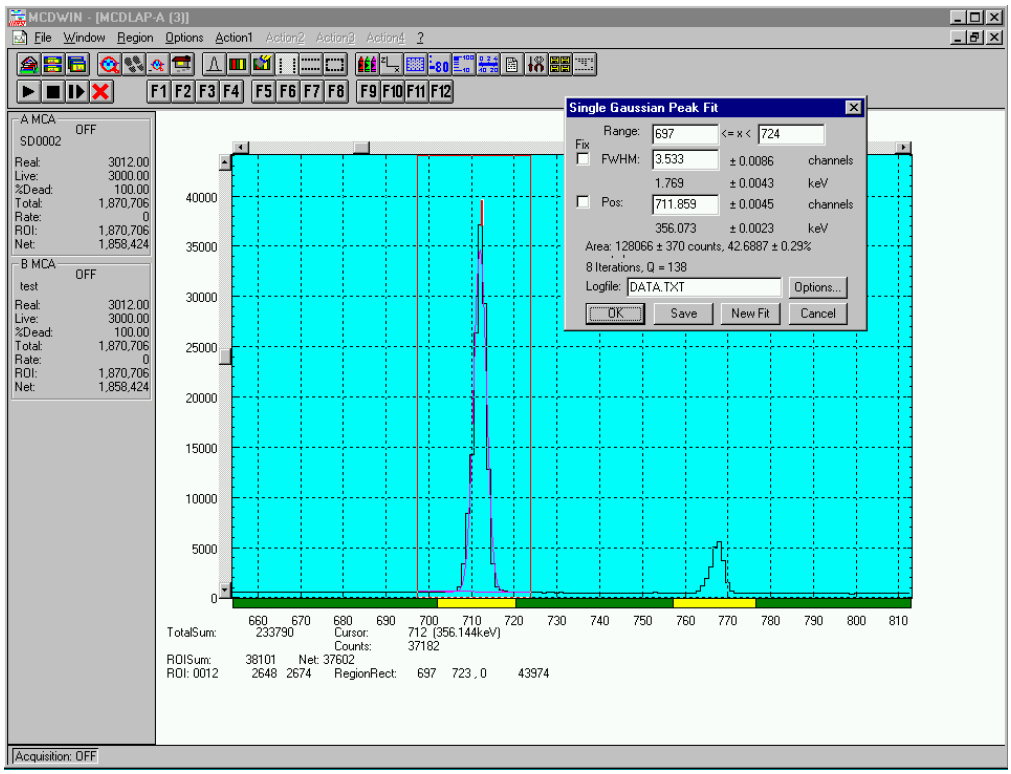


Figure 5.6: Single Gaussian Peak Fit

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:

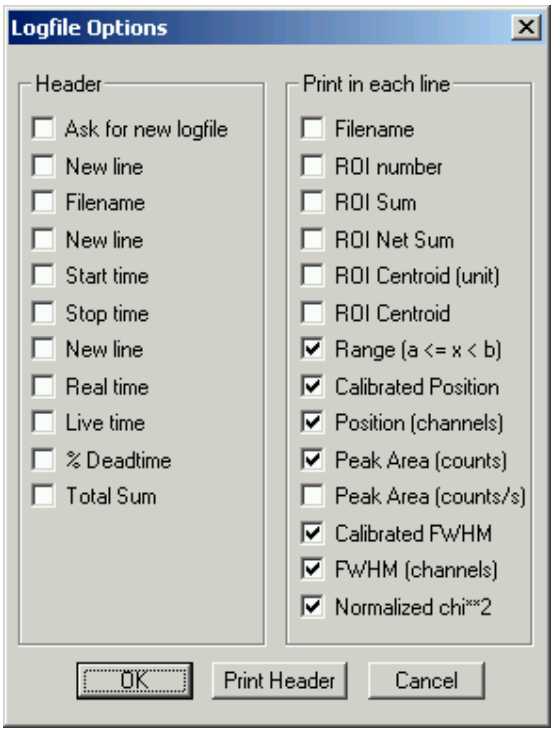


Figure 5.7: Log file Options for the Single Gaussian Peak Fit

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.

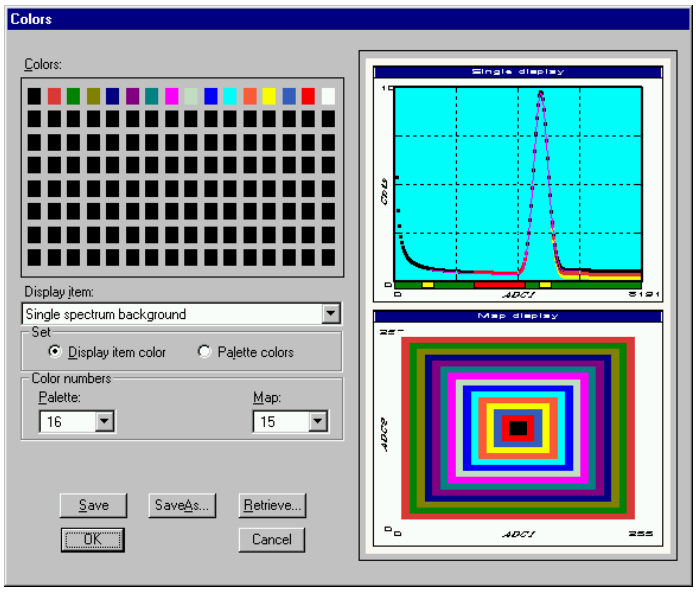


Figure 5.8: Colors dialog box

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.

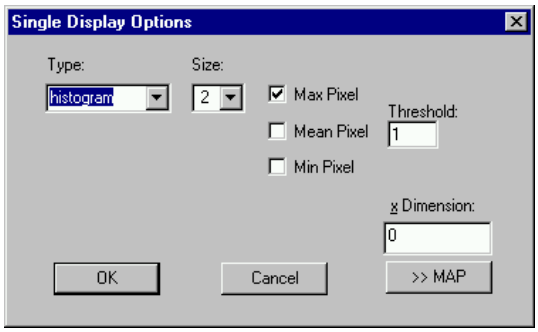


Figure 5.9: Display Options dialog box

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.

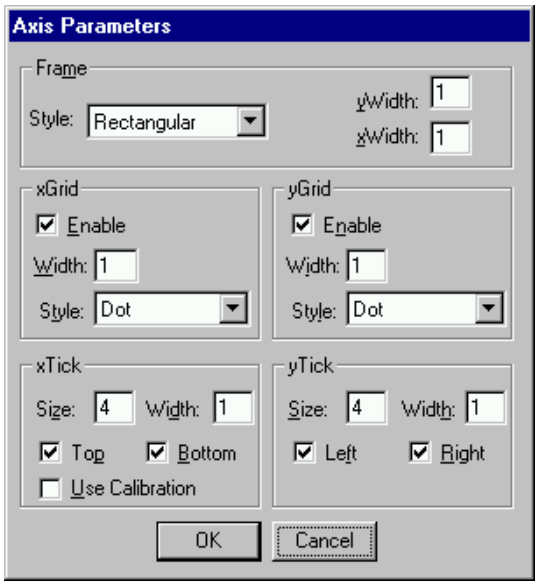


Figure 5.10: Axis Parameter dialog box

Scaling...

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

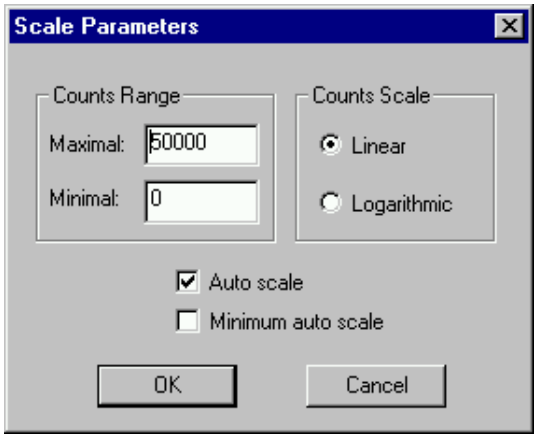


Figure 5.11: :Scale Parameters dialog box

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.

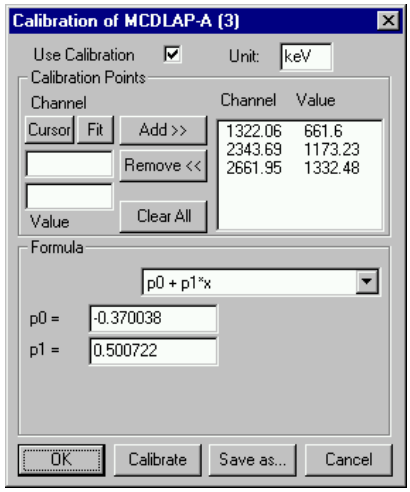


Figure 5.12: Calibration dialog box

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.

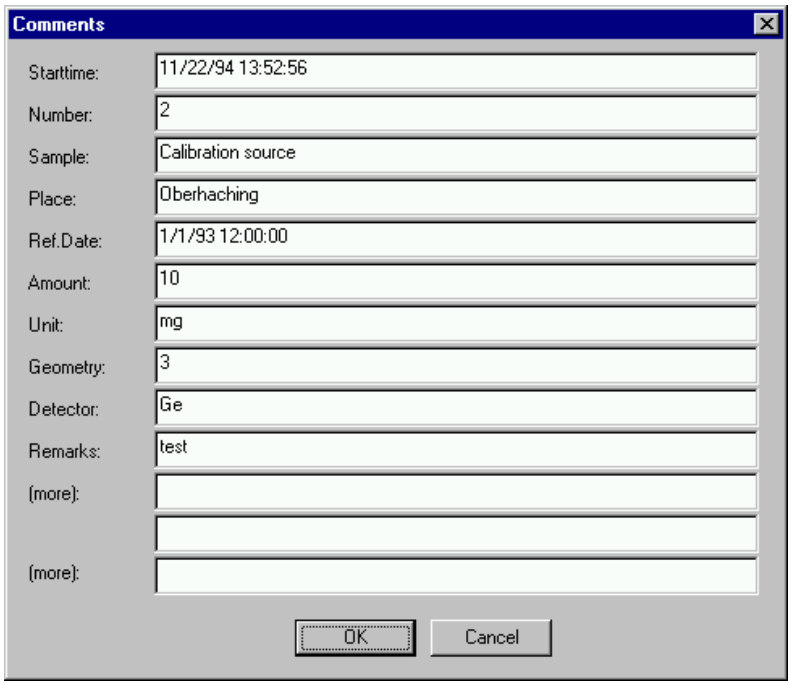


Figure 5.13: Comments dialog box

Range, Preset...

The Range, Preset dialog box allows to make all the respective MCA settings (See chapter 4.1.3).

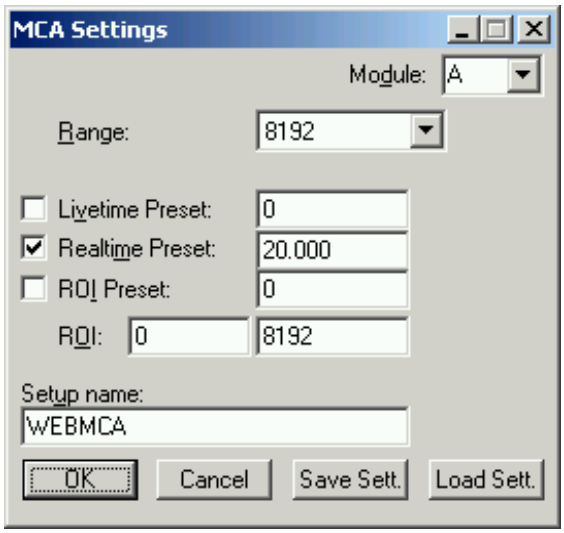


Figure 5.14: Settings dialog box

Data...

The Data dialog box allows to make all the respective Data operations and settings (See chapter 4.1.2).

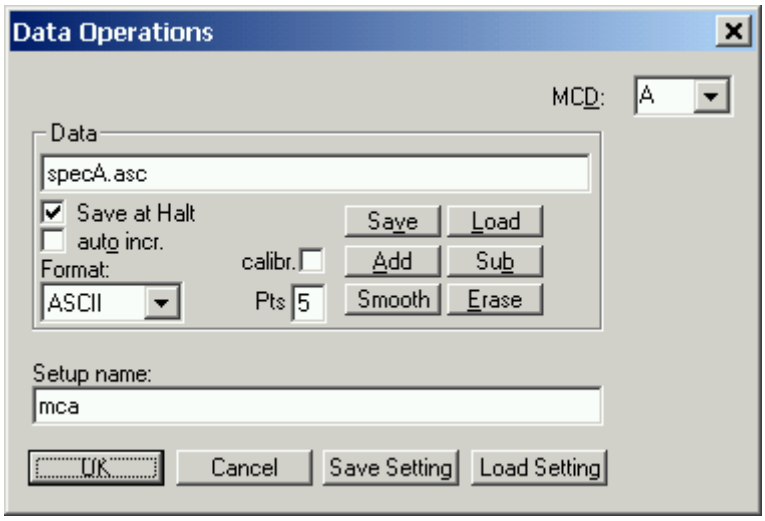


Figure 5.15: Data Operations dialog box

System...

The System Definition dialog box allows to make all the respective settings (See chapter 4.1.4).

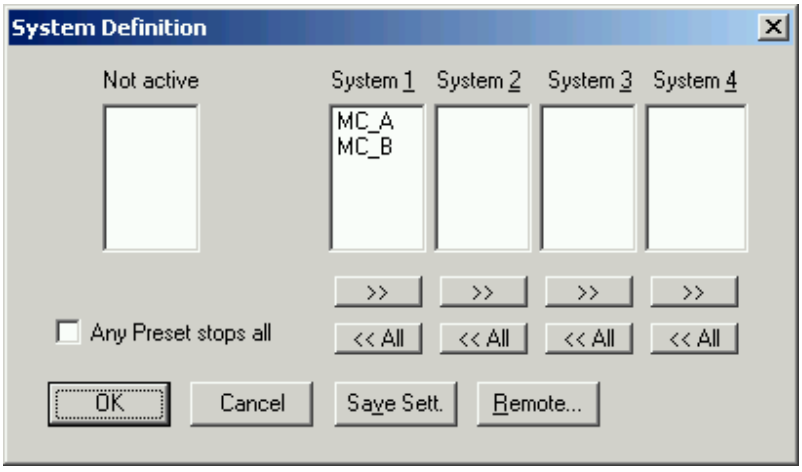
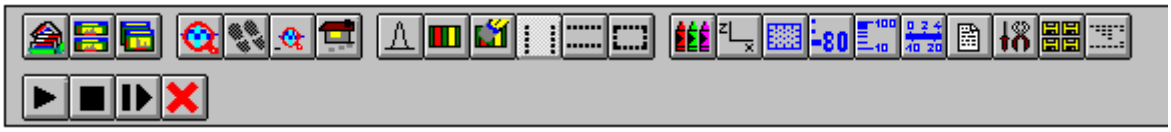


Figure 5.16: System Definition dialog box

Tool Bar...



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

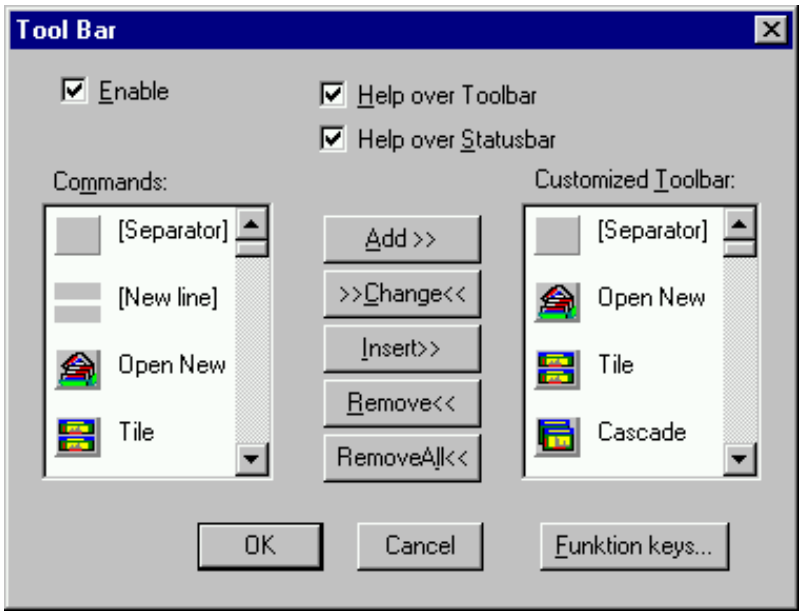


Figure 5.17: Tool Bar dialog box

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. WEBMCA Programming

6.1. Overview of the XML commands for controlling the webMCA

6.1.1. Version-Object

VERSION

Contains the actual protocol version.

Attribute(s)	req.	Content	Description
ID	X	Protocol version	Protocol version

6.1.2. Error-Object

VERSION

Contains the error number of a REQUEST

Element	req.	Content	Description
ERROR	X	Error number	Error number

0 no error
1 memory error
2 wrong version
3 DEVICE not existing
4 COMMAND not supported or wrong parameter(s)
5 access to DEVICE failed
6 OBJECT not existing
7 error in REQUEST-handling
8 wrong PARAMETER

6.1.3. PING-Object

PING

Contains a counter returned at an answer with PONG

Attribute(s)	req.	Content	Description
ID	X	32 Bit unsigned Integer	Request for PONG

6.1.4. PONG-Object

PONG

Answer to PING. Contains the ID entered with PING.

Attribute(s)	req.	Content	Description
ID	X	32 Bit unsigned Integer	Counter transfered via PING

6.1.5. Serial-Object

SERIAL

Contains the srial number

Element	req.	Content	Description
SERIAL	X	Serial (max. 25 characters)	Serial number as string

6.1.6. REVISION-Object

REVISION

Contains the hard- and software revision

Element	req.	Content	Description
SOFTWARE	X		Software version
HARDWARE	X		Hardware version

Attribute(s)	req.	Content
VERSION	X	Version
DATE		Date

6.1.7. DEVICETYPE-Object

DEVICETYPE

Contains the unique device type with an optional description.
Possible device types are:

Element	req.	Content	Description
ID	X	Device-ID (32 Bit)	unique device-ID
DEVICETYPE	X	Device type (32 Bit)	Device type
DEVICEGROUP	X	Device group (32 Bit)	Group assignment
DESCRIPTION		Description (50 characters)	Optional description

DEVICETYPE	0 - ROOT	describes the main device
	1 - MCA	describres connected MCA
	2 - Sensor	describes a sensor

6.1.8. DEVICE-Object

DEVICE

Elements	req.	Content
DEVICETYPE	X	Device type
REVISION	X	Hard- and software revision
SERIAL	X	Device serial number
PARAMETER	X	Parameter

6.1.9. PARAMETER-Object

PARAMETER

Elements	req.	Content
MEASUREMENT		Measurement parameter
NETWORK		Network parameter
AUTH		Auth.-parameter
OBJECTS		Data objects

6.1.10. MEASUREMENT-Object (ext. ADC)

MEASUREMENT

Elements	Req.	Content
ACQUISITION	X	Flag measurement active (r/w)
ACQUISITION_STOP_CONDITION	X	Condition for stop of measurement 0 - no stop 1 - manual 2 - extern 3 - via preset
ADC_TYPE	X	Type of connected ADC (r)
DWELLTIME_EXTERNAL	X	Flag for MCS advance external via connector (r/w)
COARSEGAIN		Not for external ADC
FINEGAIN		Not for external ADC
HIGHVOLTAGE		Not for external ADC
OFFSET		Not for external ADC
PRESET		List of settings for start-/stop conditions
ACQUISITION_MODE	X	Measurement mode (r/w) - PHA - PHAS (stabilized) (for future use) - MCS (for future use)

6.1.11. Preset-Object

PRESET

Start-/Stop conditions

Elements	req.	Content
START_CONDITION		Value (only for date/time)
STOP_CONDITION		Value (for date/time, realtime, livetime, counts)

6.1.12. START_CONDITION-Object

START_CONDITION

Attribute(s)	req.	Content
TYPE	X	Type of start condition 0 - manual 1 - external via gate signal 2 - via date/time specification

6.1.13. STOP_CONDITION-Object

STOP_CONDITION

Attribute(s)	req.	Content
TYPE	X	Type of stop condition 0 - manual 1 - external via gate signal 2 - via date/time specification 3 - Realtime preset 4 - Livetime preset 5 - Counts preset

6.1.14. NETWORK_CONDITION-Object

// TODO

6.1.15. AUTH-Object

// TODO

6.1.16. COMMAND-Object

COMMAND

Elements	req.	Content
NAME	X	Name of command

NAME:

GETDEVICES	results a list of connected devices.
GETDEVICE	results the parameters of a device.
CLEARSPECTRUM	cleares spectra; select spectra via parameter OBJECTID=n
ACQUISITION	starts or stops measurement; select device via DEVICETYPE=TYPE and ACQUISITIOPN=0/1

6.1.17. OBJECTS-Object

OBJECTS

Elements	req.	Content
SPECTRUM		Spectra structure

6.1.18. SPECTRUM-Object

SPECTRUM

Elements		Content	req.
SPECTRUM	ID	Number of spectra	X
	STORED	Spectra is saved	
	CHANNELS	Number of channels (r/w)	X
	BITS_PER_CHANNEL	Data depth of channels in BIT (r)	X
	ENDIAN	ENDIAN of CONTENT 0 - BIG 1 - LITTLE 2 - PDP	X
	CONTENT	Channel contents of total spectra as stream (r/w)	X
	CONTENT_COMPRESSION	Content was compressed with ZLIB	X
	REALTIME	Elapsed measuring time [s] (r/w)	X
	LIVETIME	dead time corrected measuring time[s](r/w)	X
	DEADTIME	Dead time [s] (r/w)	X
	STARTTIME	Start time of measurement hh:mm:ss:nn dd.mm.yyyy	X
	STOPTIME	Stop time of measurement hh:mm:ss:nn dd.mm.yyyy	X
	CALIBRATION_X	Polynome parameter array for X-axis (r/w)	
	CALIBRATION_Y	Polynome parameter array for Y-axis (r/w)	

	COMMENT	Comment (r/w)	
	NAME	Spectra name (r/w)	
	UNIT_X	Unit of X-axis (r/w)	
	UNIT_Y	Unit of Y-axis (r/w)	
	STABILIZATION	Parameter for software-controlled peak position stabilization im PHAS-mode (r/w) (for future use)	
	MCS	Parameter for software MCS-mode (r/w) (for future use)	
	LINEARIZATION	Structure array for linearization of spectra (each structure contains the coefficients of a 3rd degree polynome, as wellas the left and right boundary channel of the region where the polynome has to be used) (r/w) (for future use)	

6.1.19. STABILIZATION-Object

STABILIZATION

Parameter for software-peak stabilization in PHAS-mode

Elements	req.	Content
CENTROID	X	Centroid of stabilized peak
WIDTH	X	Capture range for stabilization in channels
COUNTS	X	Number of required counts in capture region for a stabilization operation

6.1.20. MCS-Object

MCS

Parameter for software MCS-mode

Elements	req.	Content
ID	X	ID of source spectra for data evaluation
ROI_L	X	Left integration boundary in spectra
ROI_R	X	Right integration boundary
DWELLTIME	X	Dwell time for channel advance

6.1.21. Calibration-Object

CALIBRATION_X/_Y

Parameter for spectra

Elements	req.	Content
DATA	X	Polynome parameter

6.1.22. DATA-Object

DATA

Polynome parameters for calibration

Attribute(s)	req.	Content
ID	X	Actual content of array

6.2. XML Examples

6.2.1. command_acquisition.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<REQUEST ID="1">
 <!--Protocol-Version-->
 <VERSION>0.1.0</VERSION>
 <!--Command-->
 <COMMAND NAME="ACQUISITION" ACQUISITION="0" DEVICEID="0" />
</REQUEST>
```

6.2.2. Preset.xml

```
<PRESET>
        // first start condition
  <START_CONDITION TYPE="n">VALUE</START_CONDITION>
        // 0 manual, 1 extern, 2 Date
        // ...
        // first stop condition

  </STOP_CONDITION TYPE="n">VALUE</STOP_CONDITION>
       // 0 none, 1 extern, 2 Date, 3 realtime, 4 livetime, 5 counts
       // ...
</PRESET>
```

6.2.3. Set_Preset.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<REQUEST ID="34"><VERSION>0.1.0</VERSION>
<DEVICE>
   <DEVICETYPE>
      <ID>0</ID>
   </DEVICETYPE>
   <PARAMETER>
      <MEASUREMENT>
         <PRESET>
            <STOP_CONDITION TYPE="3">11</STOP_CONDITION>
         </PRESET>
      </MEASUREMENT>
      <OBJECTS>
         <SPECTRUM>
            <ID>0</ID>
         </SPECTRUM>
      </OBJECTS>
   </PARAMETER>
</DEVICE>
</REQUEST>
```

6.2.4. Response_getdevice_root.xml

```
<?xml version="1.0" encoding="UTF-8"?>
 <RESPONSE ID="1">
 <VERSION>0.1.0</VERSION>
 <ERROR>0</ERROR>
 <DEVICE>

  <DEVICETYPE>
   <ID>1</ID>
```

```
  <DEVICETYPE>1</DEVICETYPE>
  <DESCRIPTION>Root-Device</DESCRIPTION>
 </DEVICETYPE>

 <REVISION>
  <HARDWARE>0.0.0</HARDWARE>
  <SOFTWARE>0.0.0</SOFTWARE>
 </REVISION>

 <PARAMETER>
  <!-- network parameters -->
  <NETWORK>
   <!-- mac -->
   <ETHADDR>00:aa:aa:aa:aa:aa</ETHADDR>
   <!-- ip -->
   <IP>192.168.1.2</IP>
   <!-- netmask, empty default netmask -->
   <NETMASK>255.255.255.0</NETMASK>
   <!-- gateway, empty default gateway -->
   <GATEWAY>192.168.1.1</GATEWAY>
   <!-- dns server list, empty no dns server -->
   <DNS>
    <IP>192.168.1.1</IP>
    <IP>192.168.1.2</IP>
    <IP>192.168.1.3</IP>
   </DNS>
   <!-- dhcp, 0 - disabled, 1 - enabled -->
   <DHCP>0</DHCP>
  </NETWORK>
  <!-- authentification parameters -->
  <AUTH>
   <PASSWORD></PASSWORD>
  </AUTH>
 </PARAMETER>
</DEVICE>
</RESPONSE>
```

6.2.5. Set_Spectrumlength.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<REQUEST ID="65">
   <VERSION>0.1.0</VERSION>
   <DEVICE>
      <DEVICETYPE>
         <ID>0</ID>
      </DEVICETYPE>
      <PARAMETER>
         <OBJECTS>
            <SPECTRUM>
               <CHANNELS>2048</CHANNELS>
               <ID>0</ID>
            </SPECTRUM>
         </OBJECTS>
      </PARAMETER>
   </DEVICE>
</REQUEST>
```

6.3. Delphi programming example

The example is based on the ICS - Internet Component Suite
http://www.overbyte.be
http://www.rtfm.be/fpiette/indexuk.htm
http://users.swing.be/francois.piette/indexuk.htm

6.3.1. TCPTest.dpr

```
program TCPTest;

uses
  Forms,
  TCPTestUnit in 'TCPTestUnit.pas' {Form1};
{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

6.3.2. TCPTestUnit.pas

```
unit TCPTestUnit;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, WSocket, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Memo1: TMemo;
    Memo2: TMemo;
    WSocket: TWSocket;
    Edit1: TEdit;
    Label1: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure WSocketDataAvailable(Sender: TObject; ErrCode: Word);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button2Click(Sender: TObject);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  WSocket.SendStr(Memo1.Text+#0);
end;

procedure TForm1.WSocketDataAvailable(Sender: TObject; ErrCode: Word);
```

```
begin
  Memo2.Lines.Add(WSocket.ReceiveStr);
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  WSocket.Close;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  WSocket.Addr:=Edit1.Text;
  WSocket.Connect;
end;
end.
```

6.3.3. TCPTestUnit.dfm

```
object Form1: TForm1
  Left = 465
  Top = 240
  Width = 449
  Height = 554
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  OnClose = FormClose
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 24
    Top = 16
    Width = 63
    Height = 13
    Caption = 'webMATE IP'
  end
  object Button1: TButton
    Left = 24
    Top = 480
    Width = 75
    Height = 25
    Caption = 'Send'
    TabOrder = 0
    OnClick = Button1Click
  end
  object Memo1: TMemo
    Left = 24
    Top = 56
    Width = 401
    Height = 137
    Lines.Strings = (
      '<?xml version="1.0" encoding="UTF-8"?>'
      '<REQUEST ID="1">'
      ' <VERSION>0.1.0</VERSION>'
      ' <COMMAND NAME="GETDEVICES" />'
      '</REQUEST>')
    TabOrder = 1
```

```
  end
  object Memo2: TMemo
    Left = 24
    Top = 216
    Width = 401
    Height = 249
    ScrollBars = ssVertical
    TabOrder = 2
  end
  object Edit1: TEdit
    Left = 104
    Top = 16
    Width = 121
    Height = 21
    TabOrder = 3
    Text = 'Edit1'
  end
  object Button2: TButton
    Left = 264
    Top = 16
    Width = 75
    Height = 25
    Caption = 'Connect'
    TabOrder = 4
    OnClick = Button2Click
  end
  object WSocket: TWSocket
    LineMode = False
    LineLimit = 65536
    LineEnd = #13#10
    LineEcho = False
    LineEdit = False
    Addr = '192.168.10.200'
    Port = '6668'
    Proto = 'tcp'
    LocalAddr = '0.0.0.0'
    LocalPort = '0'
    MultiThreaded = False
    MultiCast = False
    MultiCastIpTTL = 1
    ReuseAddr = False
    ComponentOptions = []
    ListenBacklog = 5
    ReqVerLow = 1
    ReqVerHigh = 1
    OnDataAvailable = WSocketDataAvailable
    FlushTimeout = 60
    SendFlags = wsSendNormal
    LingerOnOff = wsLingerOn
    LingerTimeout = 0
    SocksLevel = '5'
    SocksAuthentication = socksNoAuthentication
    Left = 200
    Top = 160
  end
end
```

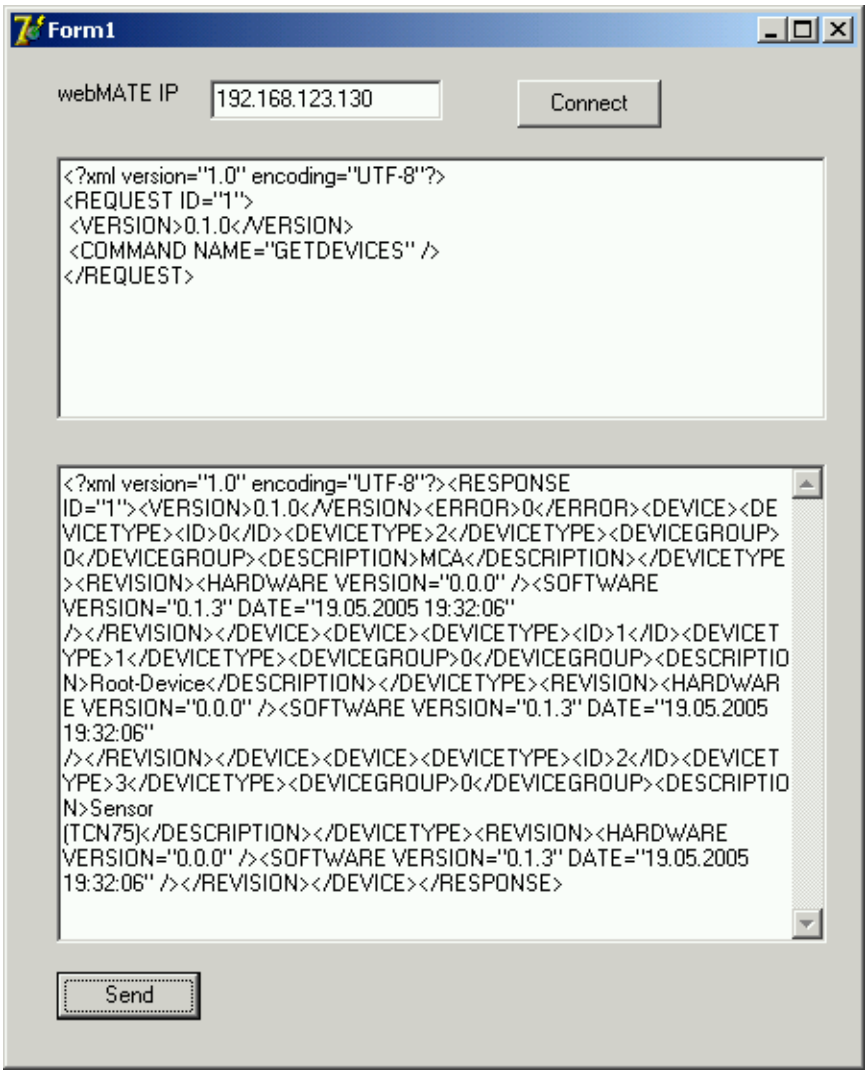


Figure 6.1: Delphi example

7. Technical Data

General:	Device Type: .....Network MCA for NIM-ADCs
	Processor: ..... 32 Bit RISC 66 MHz
	ADC-Control: ..... FPGA and adapter board
	IP protocols: ..... ICMP, NTP, TCP, UDP
	Communication: ..... XML based protocols
MCA:	Operating modes:..... PHA
	Spectrum length: ..... 256, 512, 1k, 2k, 4k, 8k channels
	Presets: ..... Real time, Live time, Counts
	Data: .....32 bit
	Presets: ..... Real time, Live time, Counts
Power Supply:	Power over Ethernet (PoE) .....48V; 55 mA (typical)
Physical:	Material: ..... Aluminum
	Size: ..... 20 x 57 x 90 mm^3
	Weight: ..... 130 g
Connectors:	Serial: .....RS232 (consol) internal
	ADC:..... 25 Pin SUB-D, see 3.1
	ADC cable: ..... ADC 40 cm connection cable included
	Network: .....RJ 45 IEEE 802.3; 10/100 MBit
	External: ..... MCX HF-Socket (input) start/stop
	Acquisition state: .....MCX HF-Socket (output)
Environmental:	Temperature: ..... -0 to +55 ^oC
	Humidity: .....up to 80 %, non condensing