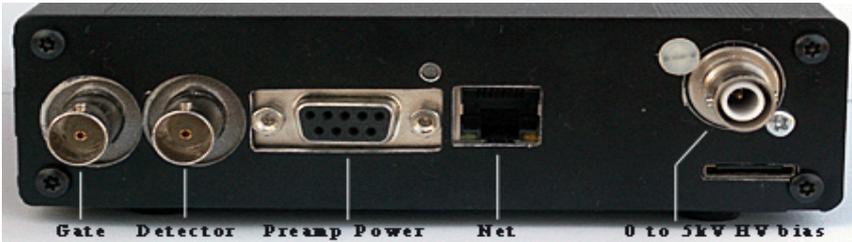
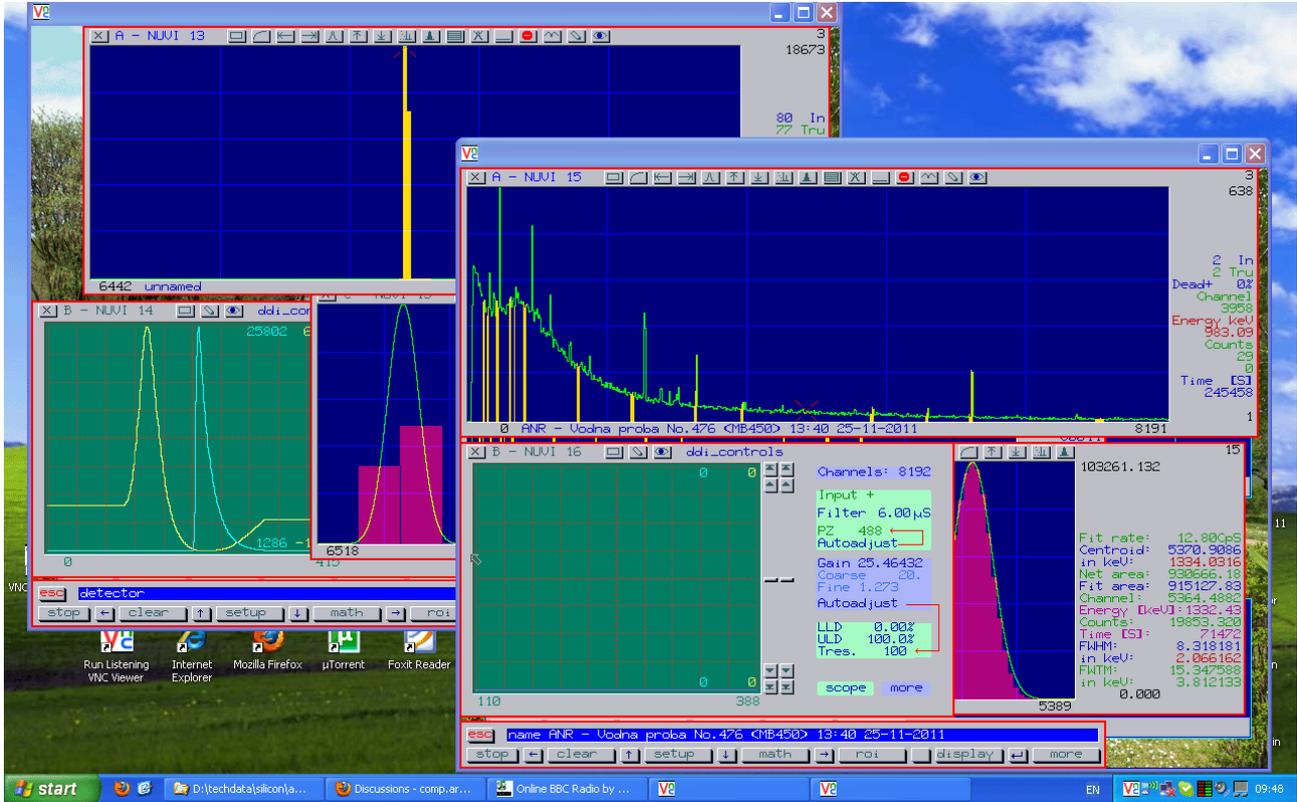


netMCA Series Reference Manual



```

NMSETUP 11 Serial no.: 1 Local:13:18:59 Sat 30 Jan 2010 Timezone:+0200
MAC address: 00C6 66666667 GMT: 11:18:59 Sat 30 Jan 2010
Fallback IP: 192.168.100.103 Home domain: tgi-sci.com Active IP: 85.130.49.8
Subnet mask: 0.0.0.0 User name: netmca2service Subnet: 255.255.192.0
Router IP: 192.168.100.1 Password: ***** Router: 85.130.0.1
Attempt DHCP: Y Report IP home: N DNS 1: 208.67.222.222 DNS 1: 217.9.224.2
DNS 2: 208.67.220.220 Override DHCP DNS server DNS 2: 217.9.224.3
DNS works over: udp
    
```

```

MCASET 14 Apply
Port A signal polarities
Data ready: low Inhibit: low
Accepted: low Dead time: low
Enable: high Output en: low
AND to: 00003FFF EOR to: 00000000
Port B signal polarities
Data ready: low Inhibit: low
Accepted: high Dead time: low
Enable: low Output en: low
AND to: 00003FFF EOR to: 0000FFFF
    
```

Table of Contents

1. Overview.....	4
2. Internet connection.....	4
3. Online support.....	5
4. Online updates.....	6
5. Backing up of data.....	7
6. Some DPS system commands.....	8
6.1. Command line input.....	8
6.1.1. Typical menu usage in DPS.....	9
6.2. Shell command line syntax.....	9
6.2.1. General.....	9
6.2.2. Command line special characters & strings.....	10
6.2.2.1 Standard output (stdo).....	10
6.2.2.2. Standard error (stderr).....	10
6.2.2.3. Standard input (stdi).....	11
6.2.2.4. Changing priority.....	11
6.2.2.5. Starting an application in background.....	11
6.3. Script commands summary.....	12
6.4. Some DPS commands:.....	13
6.5. Window Manipulation.....	15
7. Using the nuvi MCA software.....	16
7.1. Overview.....	16
7.2. Commands.....	16
7.2.1. Command entry overview.....	16
7.2.2. Description of some commands.....	17
7.2.2.1. Type independent commands.....	17
7.2.2.1. PHA spectrum type commands.....	18
7.2.2.2. Mouse functions within a PHA spectrum window.....	20
8. Spectrum evaluation.....	20
8.1. Evaluation command sequence.....	21
8.2. Efficiency calibration.....	21
9. Using the DDI (Digital Detector Interface).....	22
9.1. General.....	22
9.2. The DDI control menu.....	22
9.2.1. Main "MCA" page.....	22
9.2.1.1. Channels.....	22
9.2.1.2. Signal polarity.....	23
9.2.1.3. Filter.....	23
9.2.1.4. PZ.....	23
9.2.1.5. LLD, ULD.....	23

9.2.1.6. Threshold.....	23
9.2.2. Filtering MCA control page.....	24
9.2.2.1. Filter equivalent time.....	24
9.2.2.2. Differentiator.....	24
9.2.2.3. PZ.....	24
9.2.2.4. "BLR" scaling mode.....	24
9.2.2.5. Filter shape.....	25
9.2.3. Timing MCA control page.....	25
9.2.3.1. Tot.....	26
9.2.3.2. Pret.....	26
9.2.3.3. Prem.....	26
9.2.3.4. Pud.....	26
9.2.3.5. Skip.....	26
9.2.3.6. Mtot.....	26
9.2.3.7. Fmax.....	26
9.2.3.8. Fmod.....	26
9.2.3.9. Mmx.....	27
9.2.3.10. Pepe.....	27
9.2.4. (Oscillo)Scope page.....	27
9.2.5. Debug page.....	27
9.3. Detector power block control & monitoring menu.....	28
9.3.1. HV output.....	28
9.3.2. Target.....	28
9.3.3. Set to.....	28
9.3.4. Sensed.....	28
9.3.5. Current.....	28
9.3.6. Shutdown.....	29
9.3.7. Input.....	29
9.3.8. Upper level.....	29
9.3.9. Lower level.....	29
9.3.10. Pullup.....	29
9.3.11. Preamp voltages.....	29
9.3.12. Overcurrent.....	29
9.3.13. Current.....	29
9.3.14. Voltage.....	29
9.3.15. Rate ^.....	29
9.3.16. Rate v.....	30
Appendix A – Detailed update and reinstall procedures.....	31
A.1. Automatic update and/or reinstall of the system HDD.....	31
A.2. Manual (re)install of the system HDD partition 0.....	32
A.3. Recovering a non-accessible HDD partition 0.....	33
A.4. Recovering HDD partitions other than 0.....	34
A.5. Updating the CPLD.....	34

1. Overview

The netMCA series of multichannel analyzers includes a variety of models with different capabilities. It was introduced with its first member, the netMCA-2.

The netMCA devices are accessed through their 10/100 Ethernet connection. They act as internet hosts, providing access to the complete functionality via RFB (VNC) and to a subset of their functionality via http. Since various free VNC clients are available for both Windows and Linux, e.g. realVNC, as are various internet browsers, there is no need for any spectrometry software to run on the PC from which the netMCA is accessed. The devices are shipped with all the necessary MCA software for acquisition and spectrum evaluation included. The devices come with a hard disk drive inside; a diskless option is also offered.

2. Internet connection

When the device is powered, it will boot DPS off its hard drive or off its internal flash; it will then proceed to connect to the internet. This is indicated by the LED close to the RJ-45 connector; it will glow red when initial boot is done and DPS begins to run its turnon.bat script, and will turn green after the device has determined its IP address.

How the IP address is determined is configurable - it can either be acquired via DHCP or it can be set to a static value.

The netMCA is shipped with the following configuration:

- will request an IP over DHCP;
- will request DNS server addresses over DHCP;
- will fall back to IP 192.168.100.xxx, router (gateway) 192.168.100.1, and subnet mask of 0.0.0.0 if DHCP fails;
- will attempt to "call home" so the user can see which IP address has been assigned to the netMCA using a browser.

If DHCP has succeeded and the device has acquired an Internet connection, it will post its IP address to its home domain (which is set to tgi-sci.com upon shipping) and the user can then see it with an internet browser.

The complete address is <http://tgi-sci.com/nmchome/nmcX.txt> , where X is serial number of the netMCA.

Knowing the IP address the user can then use an RFB client (we recommend realVNC, www.realvnc.com) to connect to the netMCA [note: realVNC has to be set to use "all available colours" which is not its default setting].

The IP address will be reported as it is "inside" the netMCA, real or not; if not real (e.g. behind a NAT router) the device will be only accessible from within the local subnet (behind the router) by RFB clients.

The device can also be set to initiate RFB sessions to an Internet host; this will work also if the IP address is not real. It can be set to repeatedly connect to a remote host. The latter will have to run the VNC client in listening mode at port 5900; the netMCA can persistently retry the connection if it gets lost or fails to be established each 21 seconds if this is done using the "call" script (just type in "call <host><enter>" without the quotation marks to launch it).

If DHCP has failed, the device will use its static fallback values. These can be changed by the user and saved to the sset.xobj file and/or to the EEPROM.

If a static IP address needs to be set, the following procedure is recommended:

- Restart the device,
- run the system setup (the setup button on the shell window bottom).
- Click and edit the Fallback IP, Subnet mask, Router IP address and the two DNS servers to the desired fixed values you want to set.
- Set the "Attempt DHCP" to No.
- Save the setup to a file named sset.xobj in the root directory. Either click save -> file -> type in the name in the text line of the file browser box or type in the menu text line "save file sset.xobj" <enter> .
- Restart the netmca - you can do so by typing in reset.xxx in a shell window or by repowering it.

[The new settings will not have taken effect yet. The netmca will read its settings from its EEPROM and if it finds DHCP to be on there (Attempt DHCP: Y has been last saved to the eeprom) it will use these settings as found there.]

- run the system setup again. Change only the "Attempt DHCP" to N; then save the settings to the EEPROM (takes a few seconds, as it ends the buttons at the menu bottom will return to their "root" mode (Exit Save Load More)).
- Restart the netMCA. Now as it finds DHCP being off in the eeprom settings it will load the settings from sset.xobj in the root directory of the boot drive (the file the settings were saved to during the first part of the sequence).

It is possible that the new settings will now not work for whatever reason and the netMCA will not be accessible at all.

Should this happen, you can still boot off the ROM drive by plugging in the D-9 cap into the D-9 connector next to the RJ45. During this boot neither the eeprom settings are used nor is the hdd accessed at all; the boot drive will now be 2: (full name is #scsi\2:). You can then go to drive 0 - the boot partition of the hdd by typing

cd 0: , start the setup menu and try again, restore the previous configuration etc.

3. Online support.

TGI provides online support for netMCA users.

If you have an issue - either technical or training or both - with your netMCA you can set it to connect to the TGI support terminal where our personell will access your device and do their best to solve the issue. You can stay VNC connected while this lasts and watch what they are doing. Notice that for this to be viable you will need to have something like 500 kbps outgoing internet speed at least.

To connect to the TGI support VNC terminal you can type in a shell window

```
vnscrv -g a support.tgi-sci.com & <enter> .
```

This may or may not work depending on whether there is a listening terminal slot at the moment at TGI. You can also activate the support seek loop by running the "support" script: type in

```
support <enter>
```

in a shell window and follow the instructions. After you activate the support seek loop you will have to restart the netMCA for it to start to run; it will then attempt to connect to support.tgi-sci.com every 21 seconds until success and will persistently retry it if it loses the connection. You can monitor the progress of this connection in a small shell window which appears above the initial boot shell (also small) window.

Once the issue is resolved TGI will switch your support seek loop off. You can do this yourself by running the same "support" script and following the instructions.

4. Online updates

While updates are available online checking for some is left to the user (i.e. it is not automated) since it is believed that most MCA users will prefer a system they are happy with unchanged.

Users can check for updates available online by typing
update <enter>

in a shell window and follow the instructions, if any (these will depend on the particular update).

This will transfer the minimum needed for the update; it may include also other operations.

To completely reinstall the latest version onto your drive 0, run the "reinst" script; it will download a complete disk image and write it to drive 0. Make sure to backup any user data you may

have on this drive prior to that as it will be lost after reinstalling. The rest of the partitions (most of the HDD as it is shipped) will be preserved. Reinstall will always be followed by a reset.

If updating a really outdated version (so there are numerous updates between it and the latest one) it may be more practical to simply reinstall the latest.

*** It is important to run the update or reinst scripts as the first and only command the netMCA will receive after boot. ***

5. Backing up of data

Since most netMCA devices come without their internal backup device (like say a DVD writer), they rely on ftp-ing to backup data.

A typical way of backing up data is to create a "disk" image file and use the "backup" script to copy to it a whole directory tree which is to be copied; then this image file can be ftp-d and written to a CD, DVD etc. from where it can be subsequently restored to the netMCA disk again via ftp.

This is how a disk image is to be created:

```
pfid <template> <destination><<CR>
```

Various source type images are located in \syslib\devices\diskimg\:

hdd06p.disk - this is the image used for the boot partition of the netMCA coming with a 120G disk; it is an 815M disk image with a partitioning table which splits the drive in the following partitions:
0: - 815M, B: - 8G, C: - 20G, D: - 20G, E: - 20G, F: - 40G.

hdd8g.disk - 8G template image for 512 byte sector drives,
hdd20g.disk - 20G template image for 512 byte sector drives,
hdd40g.disk - 40G template image for 512 byte sector drives,
vnudvd.disk - 2G template image for 2048 byte sector drives,
dvd4g.disk - 4G template image for 2048 byte sector drives.

The template images only contain the disk system info, root directory, CAT etc.; the pfid command uses them to create images of the respective full size.

If some other image template is needed, the "newdisk" script can be run; type in "newdisk" and follow the instructions.

Once the new disk image has been created using the pfid command it must be set as a disk-drive like accessible device.

For this purpose the fdk device driver must have been installed (if it is not you can do so by typing in a shell window `install \syslib\devices\fdk<CR>`).

To "mount" a disk image file as one of the possible 16 LUNs (0 to f) of the fdk device type
`setfdk <N> [disk image file path]<disk image file name>`
where N is the LUN (Logical Unit Number), 0 to 9 or a to f .

Once this is done the new image is ready for access as `#fdk\N:` etc.

For example, let's say we want to backup the entire drive 0 and we want to do that as an 815M image (just as the physical drive 0 is). We want to back it up say in the an `f:\backups\` directory which we have previously created, the new image will be named `sysbkp.disk`.

Here is how to do this:

```
cd 0:
install \syslib\devices\fdk <--- install fdk device driver if needed
pfd \syslib\devices\diskimg\hdd06p.disk f:\backups\sysbkp.disk
setfdk 0 f:\backups\sysbkp.disk
backup \ =fdk\0: noecho%% <--- recursively copies all directories
setfdk 0 <--- disconnects image file, truncates it to its used size
```

Once the `sysbkp.disk` file is thus ready, it can be transferred using ftp and saved elsewhere.

To retrieve data from a previously backed file, make sure to download it via ftp on some of the local drive partitions; then set it as an fdk LUN and access the files inside it. Say you download a previous backup image named `databkp.disk` and put that in `f:\backups\` using ftp; then

```
install 0:\syslib\devices\fdk (if not already installed)
setfdk 0 f:\backups\databkp.disk
```

You can then access the newly mounted image as `=fdk\0:`, say, `dir =fdk\0:` etc.

6. Some DPS system commands

6.1. Command line input

DPS command line input is done via two similar functions depending on the interface and the program which inputs the command line.

Shell (the general user command interface) and `nuvi` input commands through the more sophisticated "menu" line input. Simpler programs - such as say copy asking whether to overwrite etc. - use a simpler "keyin" call.

In either of these, the following keys are treated more

or less the same:

ctrl-x - cancels all text input so far and wait for input;

esc - cancels all input text and exits input with "aborted" status;

del (rubout, $\$7f$; sometimes marked as "backspace" on the keyboard) deletes the character at the left of the cursor.

The "menu" style input - used by shell, nuvi and some others - also has the following line editing functions:

cursor left, cursor right move the cursor within the line;

shift cursor right - move to next word;

shift cursor left - move to previous word;

alt cursor left- delete previous word;

alt cursor right - delete until end of the word;

ctrl cursor right - delete until end of line;

INS key - toggle insert/replace mode (default being replace);

del - delete character under cursor;

alt-I - paste top line of "clipboard" data at cursor position.

Also, the cursor can be positioned by a mouse click within the edited line. A double mouse click will have the effect of alt-I after positioning the cursor.

6.1.1. Typical menu usage in DPS.

Whenever a menu appears (e.g. the netmca system setup menu) it usually has a row of buttons at the bottom and a text bar above them. One can either click the commands and follow prompts which may be displayed in the text bar or choose from subsequent button pages which are offered, or type in commands in the text bar. As soon as one has typed in an unambiguous beginning of an acceptable command, it is displayed in a shaded colour to the right of the text cursor; press the space bar to have the command complete as text. If text typed, some commands may execute immediately after space is typed while others may wait for more input, this is context sensitive (decided at programming time).

Some parameters can be clicked on and edited to a new value inside their boxes just like one would edit a text line. Clicking outside of the box or typing a CR will end the editing setting the new value, ESC will abort and restore the old value. Also, one can click edit numbers; the first time you click on a number in a parameter the text cursor will move over it, the second time you left click it the character under the cursor will be incremented by 1 - and decremented by 1 by right clicking it.

6.2. Shell command line syntax

6.2.1. General

Shell is the application which is given control to

at boot time. The first time it is given control to it starts the \turnon.bat script and after it stops execution it gives control to the user. Commands can be entered from a command line, some of them being readily available via the buttons at the shell window bottom.

Shell begins to process a command line after CR (enter) has been pressed.

If a command line has an @ as the first non-space character, it is assumed to be a command issued to the script processor and is passed to it (described in dpsyntax.txt).

If the first non-space character of the command line has not been a @, a file with the name of the first word (default suffix is .com) is searched for in the current directory and in all directories defined via the @path command; if found, it is run as a program. If a .com is not found, a .bat file with the same name is searched; if found, it is started as a script.

In both cases (line starting with an @ or not) before the shell begins to process the line, it is passed to the script processor for filtering to replace variable(s) and/or expression(s) and then is filtered by the shell itself for certain key characters.

6.2.2. Command line special characters & strings

Commands are spawned with their standard input (stdi), standard output (stdo) and standard error (stderr) being the same as those of the shell unless other(s) are explicitly specified. In the normal case after boot the shell will have its stdi, stdo and stderr from/to its window, thus reading its commands from the keyboard and writing its output to the window.

Here are the shell's key characters and how they can be used:

6.2.2.1 Standard output (stdo)

The > character followed by a file name defines the stdo. For example,

```
dir *.com >test
```

will spawn the dir.com which will list all files matching *.com to the file test.txt (.txt is the default suffix). If the file test.com exists, the "Duplicate file name" error will occur. To overwrite an existing file (or create a new if not existing, the > must be followed by a ! character, thus the example above will look like

```
dir *.com >!test
```

To append the output to an existing file, >> must be specified. In this case the example will become

```
dir *.com >>test
```

6.2.2.2. Standard error (stderr)

The >< combination defines the stderr of the spawned task (error messages are output to the stderr).

For example,

```
dir *.com ><test
```

will cause dir to output the list of files matching the *.com wildcard to the default output (e.g. the window), but will output any error messages to the file test.txt. To overwrite or to append, the same syntax as with stdo is valid:

```
dir *.com ><!test      will overwrite test.txt
dir *.com ><>test      will append to test.txt.
```

6.2.2.3. Standard input (stdi)

The < character specifies the stdi of the spawned task. For an example, the "ekey.com" is chosen because it uses some stdi. What it does is to output to its stdo the hex equivalent of every character it reads from its stdi (it exists as it input a CR (enter), \$0d, or an eot (\$04). Thus

```
ekey <test
```

will echo as hex to its stdo every character it reads from test.txt until it encounters the first \$4 or \$od.

6.2.2.4. Changing priority

When shell spawns a new task, it assigns it the same priority as its own. To change this, priority can be specified either in units relative to the default (shells) priority or in absolute units. The ^ (\$5E, ascii circumflex) character designates priority.

For example,

```
dir *.com ^+5
```

will start dir with priority which is equal to that of the shell plus 5.

```
dir *.com ^-5
```

will start dir with priority equal to the current (that of the shell) minus 5.

```
dir *.com ^5789
```

will start dir with a priority equal to 5789. The valid priority range is 1 to 65535.

To change the shell's own priority is a different thing since the shell we are talking to is already started. It can be done by issuing the "priority" command to the script processor, like that:

```
@priority +8
@priority -9
@priority 5599
```

The first line will increment the shell's priority by 8, the second line will decrement it by 9, and the third line will set it to 5599.

6.2.2.5. Starting an application in background

If the & (\$26, ascii ampersand) is encountered in the command line, the task is spawned in background. This means that shell will not wait it to exit but will come back immediately. For example

```
dir *.com >#uart\1: &
```

will start dir with its output redirected to uart port 1 and come back immediately. Now, the uart may be either slow or held for handshake etc. so the dir will execute for a long time; nevertheless, shell will be ready to accept new commands immediately.

6.3. Script commands summary

Here is a summary of the commands which can be issued to the script processor (see dpsyntax.txt for details).

In order a line to be passed to the script processor by the shell, it must start with a @ (\$40, ascii commercial at).

```
@setl <variable name> [string] - set a local variable to a string,
@setg <variable name> [string] - set a global variable to a string,
@if<condition> <string for evaluation> - conditional operators, the conditions
available being:
@ifc ,string1,string2, - if strings compare (case sensitive),
@ifnc ,string1,string2, - if strings do not compare (case sensitive),
@ifr ,string1,string2, - if strings compare (read same, case insensitive),
@ifnr ,string1,string2, - if strings do not compare (case insensitive),
@ifeq <expression> - if the expression equals to 0,
@ifne <expression> - if the expression is not equal to 0,
@ifgt <expression> - if the expression is greater than 0,
@iflt <expression> - if the expression is lower than 0,
@ifge <expression> - if the expression is greater than 0,
@ifle <expression> - if the expression is lower than or equal to 0,
@ifdefc <variable name> - if such a command line variable is defined,
@ifudfc <variable name> - if such a command line variable is not defined,
@ifdefg <variable name> - if such a global variable is defined,
@ifudfg <variable name> - if such a global variable is not defined,
@ifdefl <variable name> - if such a local variable is defined,
@ifudfl <variable name> - if such a local variable is not defined,
@endc - end of latest effective conditional operator effect,
@write <variable name> <file to write to> [options] - write var. to file,
@lbl <name> define a location in a script where the "goto" can go,
@goto <label name> - change script flow - resume execution at <label>.
@abort - abort a script with "aborted" error status,
@priority [+N,-N,N] - display/change shell's priority,
@end - end a script with no error status,
@wait <time in seconds> - wait the time to pass before next statement,
@retry <script intermediate file name> - retry line which aborted the script,
@resume <script intermediate file name> - continue after aborting line,
@r - retry last script,
@g - resume last script,
@io - return the names and current file positions of stdi, stdo and stderr,
@input - input a line from stdi,
@inline <input name> [options] - input a line,
@timeout <value in seconds> - define max. time to wait for input,
@eoff - set shell's stdo to #null (display no echo),
@eon - set shell's stdo to original stdo,
```

@path [path name, -] - add/delete/delete all/show all path(s) to search,
 @mask <value> - set the error mask value,
 @level <value> set the error level value,
 @rem [any text] - remark to be displayed during execution,
 @make <object name> [string to pass to "create" action] - make an object,
 @do <action> <object identifier> [text to pass to action code] - the obvious,
 @date - returns the current data as text in i@0 - i@5 local variables (in i@1
 as a whole),
 @xdate - returns date in unix-like format, the last word being seconds since
 JAN 1 1970,

Those commands which return some text do so by setting the following
 local variables:

i@1 - contains the entire returned line,
 i@ - number of returned words,
 i@0 - left most word in of the returned text,
 i@1 - next right word and so on up,
 i@r - error index returned by the command (0=no error),
 i@t - terminating character if the comand was "inline" or "input".

Notice that existing i@N (N being word nubur) variables are only overwritten
 if a word with that nubur is returned; they stay unchanged otherwise, check
 the i@ for validity of each i@N var.

6.4. Some DPS commands:

cmp <file 1> <file 2> [-o] - compare files.
 The o option causes display of different bytes (hex).

copy source [destination] [-options] - copy file(s).
 options:
 w - overwrite existing destination without asking,
 c <N> - clip destination file size to N,
 s N - force source file type to N,
 d N - force destination file type to N.

del <file name> [-options] - delete file(s).
 options:
 y - delete matching files without asking,
 a - delete any type of file (including dir and cat).

dir [path] [file name] [-options]
 options:
 w - wide display,
 s - show system (hidden files),
 a - display file allocation.

free [device name][lun] - display device size and free space there.

kill <task id> - kill a task. Id can be seen via the "t" command.

md <directory name> [-options] - make directory.
 options:
 s <N> - set initial size to N entries (default 203),
 d - set directory type to time updated (its own directory entry
 in the parent directory will be updated every time the directory
 is written to).

rd - <directory name> - remove directory.

t [-options] - show running tasks.

options:

w - wide display,

f - show task "flag" values.

te [file name] - text editor.

shift cursor up - 4 lines up,

shift cursor down - 4 lines down,

shift cursor right - move to next word,

shift cursor left - move to previous word,

alt cursor left- delete previous word,

alt cursor right - delete until end of the word,

ctrl cursor right - delete until end of line,

alt m - set one corner of marked field,

alt l - set one end of marked field marking the whole line,

alt k - delete (kill) marked area,

alt c - insert marked area (as new lines),

alt z - insert marked area as new lines at cursor x position,

alt ctrl v - remove marked area (not the lines it occupies),

alt g - delete all lines of marked area,

alt f - fine next (case insensitive),

alt s - search next (case sensitive),

ctrl u - map to upper case to end of word,

alt d - map to lower case to end of word,

alt ctrl u - map to upper case the whole line,

alt ctrl d - map to lower case the whole line,

ctrl a - map to upper case character under cursor,

ctrl f - map to lower case character under cursor,

ctrl b - switch to bulgarian,

ctrl e - switch to english,

ctrl-pgdn or F9 - move to EOF,

ctrl-pgup or F10 - move to BOF,

esc - enter command mode (many take 2-3 escapes at times).

vpa, a32, a54, l32 - programming tools.

ntp <host> [-options]:

s - set system time to the time received,

q - quiet mode, do not output date info,

t - get time via rfc868 format,

o - show system clock offset (+ if ahead),

d - debug mode, list entire packet as hex.

timeserv - time server, serves both NTP and rfc868 requests.

Uses system time.

ftp - ftp client.

Has the following commands:

name - define user name to log in with,

pass - define password to log in with,

cd <directory specification> - change remote directory,

lcd <directory specification> - change local directory,

ls [name specification] - list remote directory,

dir - same as ls (see above),

pwd - list current remote directory,

lpwd - list current local directory specification,

get <remote name> [local name] - get a file,

put <local name> [remote name] - put a file,

putl - same as put, but remote name converted to lower case,
putu - same as put, but remote name converted to upper case,
close or cl - disconnect from remote host,
open or op <host name> - connect to remote host,
quit - diconnect and exit.

Some useful scripts:

backup <source> <destination> [noecho%%] - copy all directory trees as they are from source to destination. To back your disk drive 0 to drive 1, for example, you may specify "backup 0: 1:". If the "noecho%%" is present in the commmands line, echo will be off during backup allowing it to run in background (while measuring, for example).

rm <path name> <-R> - remove all files and directories below this path (only the specified top directory remains).

xm - start nuvi with the original (factory) settings,

fixdisk [drive] [c%%] - run a disk repair for that disk and store the recovered CAT in recover.cat in the current directory; then compare recover.cat to the drive's unitcat.syst and if different, ask if to overwrite. The "c" option is for listing the differences found, if any.

date - display current date and time,

ntime - gets time from time-a.nist.gov and displays it along with the current local clock error,

stime - sets the local clock to the time taken from time-a.nist.gov.

lookup <host> <type> - do a DNS lookup of <host>, type being one of:

A - IP adress record,

NS - nameserver ... etc., please consult rfc1034 & rfc1035

MD

MF

CNAME

SOA

MB

MG

MR

NULL

WKS

PTR

HINFO

MINFO

MX

TXT .

6.5. Window Manipulation.

DPS allows window size & screen position manipulation with the mouse like most other operating systems, however catching them to move around is a bit different - one has to right-click the window corner (or border if to move it just over one axis).

Resizing windows is as usual, left click the window corner or

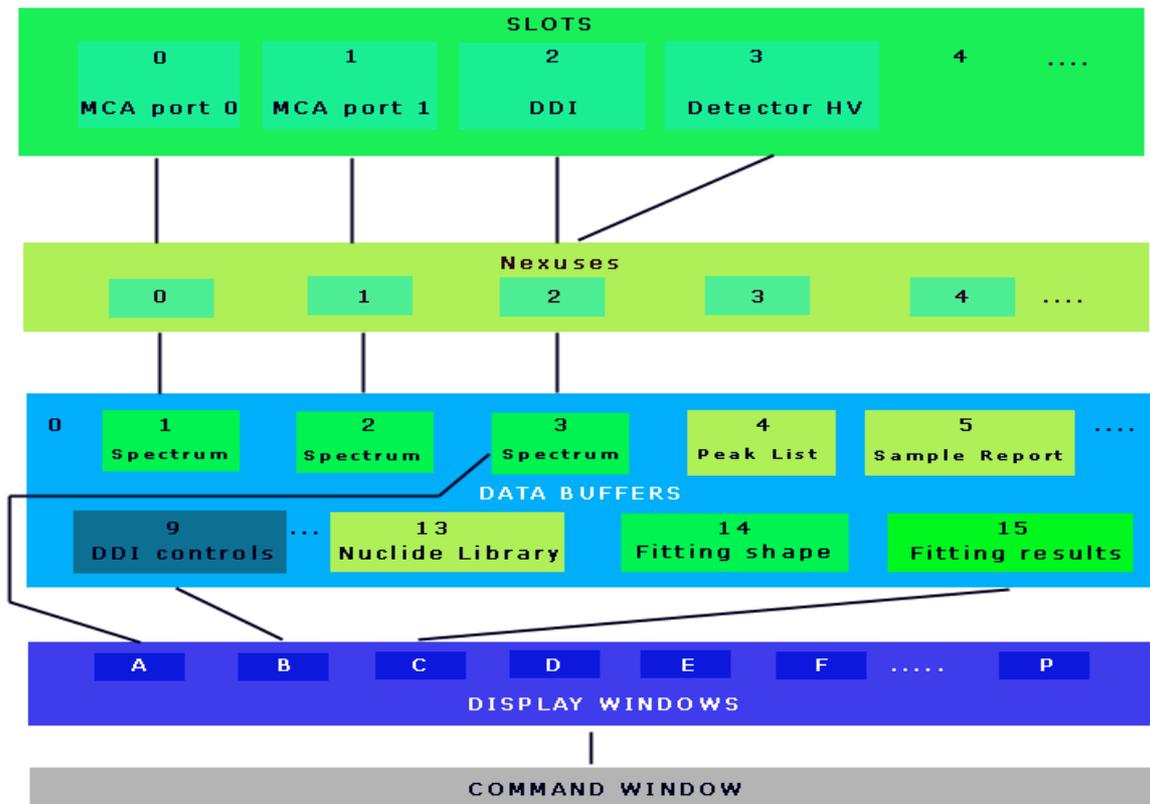
border and drag.

The current window is the window which owns the keyboard. To change it, either click anywhere inside the window to become current, or select it with the alt-tab key combination (press alt and then hit tab as many times as it takes until you are offered the task you need, then release both alt and tab).

7. Using the nuvi MCA software

7.1. Overview

The nuvi software is organized in a way to allow maximum flexibility and expandability. It is organized as a pool of data buffers, acquisition nexuses and hardware slots. The user can connect a number of slots to an acquisition nexus and an acquisition nexus to an object residing in a data buffer (e.g. a spectrum).



The objects in the data buffers can be visualized through the "display" windows, and commands related to the whole chain display-buffer-nexus-slot can be issued via the user interface and/or a command script. There are 16 data buffers, 16 nexuses and 16 slots in nuvi. Data buffers 13, 14 and 15 are by default occupied by the nuclide library, the gaussian curve (which is fitted to peaks) and the peaks resulting from the last fit operation respectively.

Our systems are delivered with ready to use connection definitions, e.g. in the netMCA-2 slot 1 (which is MCA port A) is connected to nexus 1 which is connected to data buffer1 where the default1.spc spectrum is loaded upon start and, similarly, slot 2 (MCA port B) is connected to nexus2 which is connected to buffer 2 where default2.spc is loaded. Obviously default1 and default2 spectra can be modified by the user to be of the desired size, to carry some known energy calibration etc.

Nuvi will store its connection configurations upon exit and will restore them same upon next start.

Nuvi uses its own spectrum format, which is a DPS object type. Saving spectra in this format to a file allows complete data retrieval afterwards, with the ROI map, energy calibration, name etc. Nuvi can also export and import spectra in 3 other formats - plain ACII, competitor ASCII (also referred to in many of our documents as IAEA-like, perhaps wrongly) and Nucleus-2. Spectra can be exported and saved to a file or can be accessed with a browser over http if the http server has been started by nuvi (netMCA-2 is shipped with this on).

7.2. Commands

7.2.1. Command entry overview

Issuing commands to nuvi by the user is done via the command window or by using the buttons on the display window.

The buttons on the display window are just clickable, one time action buttons with some hieroglyphic marking, like clear, load, save etc. What buttons will appear on top of a display window depends on the object which is shown in that window.

The command window has a set of buttons and accepts typed in commands as well. The buttons are marked with text and are object specific; the text input will offer a command for finishing with the space bar as soon as enough is typed in to make this choice unambiguous.

The command window is connected to one of the display windows and thus to the object which is contained in the data buffer which is connected to this window. This connection (command window <-> display window) is indicated by illuminating the display identifying text in the upper left part of the display window, e.g. A - NUVI 5).

Managing the connections and active display can be done using the "connect" and "display" commands; they give access to the complete functionality in that respect.

There is a quick way to connect the command window to a display window - by simply typing in its associated character (a to p) followed by a CR (enter on the keyboard).

The same applies to connecting one of the first 10 buffers (0 to 9) to the current display (the display window currently connected to the command window): simply type in the buffer number followed by a CR.

Commands of any text can be entered by typing them into the command window. If a command is not recognized as a known nuvi command - this search being object specific, i.e. will depend on what object is shown through the current "display" window - a file with that name and a .wxc suffix will be searched in the nuvi directory and, if found, will be run as a nuvi executable. If a .wxc file is not found, a file with that name and a .bat suffix will be searched in the nuvi directory and, if found, it will be run as a nuvi inherent DPS command script (so it may contain all command sequences the user could type in manually plus script internal commands with variable processing, see DPS command scripts). If nothing is found, an error message will be shown.

7.2.2. Description of some commands

There is a great and constantly growing variety of commands. Here are described the most basic and frequently used ones. In this description, "current window" or "current display" refers to the display window currently connected to the command window, and "current buffer" refers to the buffer currently connected to this window.

7.2.2.1. Type independent commands.

These are commands which nuvi will accept regardless of the object type in the current buffer, including the case of an empty buffer (obviously save does not apply to an empty buffer).

load [file name] - load an object from a file into the current buffer. If no name is specified the command will open a filebrowser window. If a name is specified without a suffix a .spc (pha spectrum type) will be assumed. If the command is started from the command window without a file name specified, a "timeout error" may be shown in the command window. It is safe to ignore this error in this case.

save [file name] - save the current object to a file. If no name is specified a filebrowser window will be opened. If write is attempted to an existing file, an overwrite confirmation will be requested.

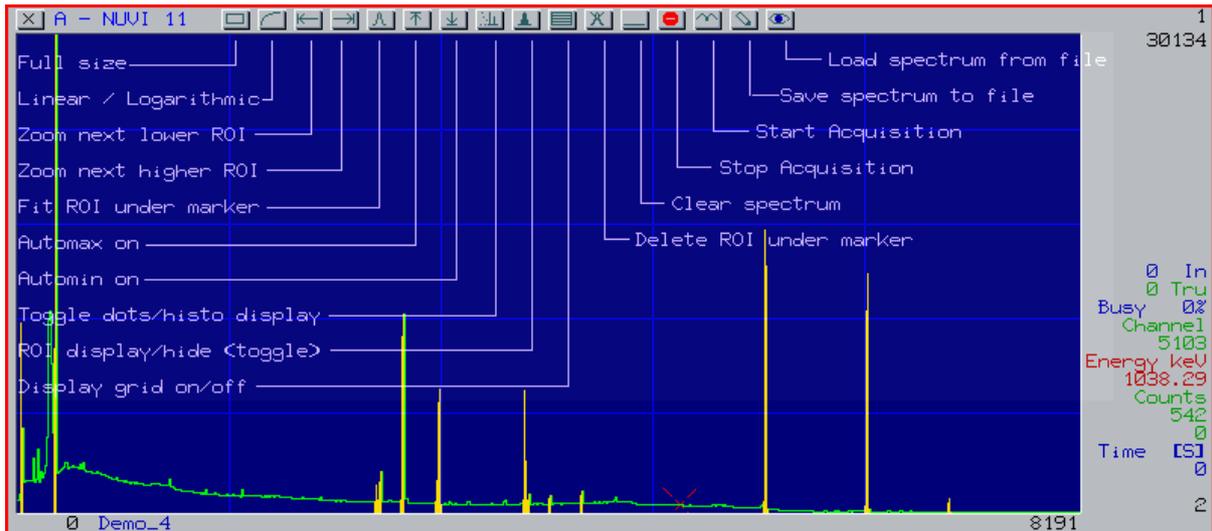
savew [file name] - same as save, but will not ask for overwrite confirmation (useful in command scripts).

connect - manage connections. Options will be offered in the command display buttons; used to connect a buffer to a display, a nexus to a buffer, a slot to a nexus etc.

display - control display window behaviour. Object specific options will be offered in the button area. The following object independent options are available:

- rate - sets minimum interval between display refreshes (in mS),
- connect - offers to connect one of the displays to the command window,
- size - to change the display window size, expects all 4 corners of the window to be specified as % full screen (left limit, upper limit, right limit, bottom limit; top being 0),
- remove - removes that display window, to have it again simply reconnect it.

7.2.2.1. PHA spectrum type commands.



clear - set all channel data and the acquisition time to 0,

start - start acquisition. The time begins (or resumes) or to count, event data are processed into the spectrum. Applies only if a startable nexus is connected to the spectrum, if not it won't appear on the command list.

stop - stop acquisition. Appears only while acquisition is in progress.

setup - offers a choice of objects to be setup, system dependent. On the netMCA-2, the mca port can be setup.

math - a variety of math functions are offered:

add <N> - add spectrum N to current spectrum (channel to channel),

subtract <N> - subtract spectrum N from current spectrum,

scale <A> - scale current spectrum by X axis so that channel B gets down into channel A (A and B need not be integers), for example scale 2 1 will result in an 8192 channel spectrum to be scaled to 4096 channels.

multiply <A> - multiply each channel contents by A. A needs not be an integer.

fit - fit the curve in buffer 14 to the peak under the marker and put the results in buffer 15. Calculates centroid, FWHM, FWTM, net area, fit area etc. The fitting results are also graphically visualised and are viewable through a nuvi display window. Normally the curve in buffer 14 is gaussian (gausscur.spc is loaded during start).

roi - manage the Regions Of Interest (ROI) in the spectrum. Offers

mark_beg - set the beginning of a new ROI at present marker,

set - sets a new ROI between the marked beginning and the marker,

remove - remove the ROI under the marker,

next - zoom next higher (by X) ROI,

previous - zoom next lower ROI,

whole_data - unzooms all, i.e. shows the entire spectrum in the display,

del_all - delete all ROIs.

left, right, top, bottom <N> - set the respective display limit,

define - define a ROI (usually from within a script) or a buffer address and size,

define ROI <A> define a ROI between and including channels A and B,

define buffer N address <address> <size> . This is useful after having

created an object and knowing its address and size in memory (usually returned by the object create action). It takes an advanced understanding of the nuvi software.

import - import a non-nuvi spectrum. Offers ascii, nucleus and iaea formats.

export - export spectrum in non-nuvi format. Same formats as import available.

new - create a new entity. The following choices are offered:

calibration - deletes all calibration data in the current spectrum,

spectrum - create a new spectrum. The following is prompted:

size [-r <max. ROIs>][n <name>]:

size must be specified in number of channels,

maximum number of ROIs can be specified (default is 128),

name can specify a spectrum name (default "unnamed"),

c <max. number of calibration points> - default is 128.

peak_list - creates a peak list (first step of spectrum evaluation),

library <file name> - create a nuclide library (usually the default nuclide library has been loaded in buffer 13 by begin.bat)

calibrate <channel> <energy> - add a point to the calibration polynomial.

The order of the polynomial is incremented by 1 and the point is stored.

While the number of allowed calibration points is very high (default 128),

it rarely would make sense to use more than 4 as beyond that the polynomial

will likely "oscillate". For HPGe detectors with competitor conversion

hardware - like NIM main amps and ADCs - usually a 2 point calibration is

quite adequate. With our DSP based conversion devices, e.g. the hi-spec

nukebus module or the netMCA-3, a single point calibration is sufficient

(these devices have a 0 offset zero guaranteed by design).

Hint: you don't need to type the channel centroid position. Just click

the centroid value in the fit results window, this will place this

text on the clipboard; then press ALT-I to paste it within the command

line.

set - define acquisition stop condition.

set realtime <N> - stop after acquiring for N seconds,

set livetime <N> - stop after acquiring for N seconds of live time,

set gross <M> - acquire untill the gross integral in the specified

region reaches <M> counts,

set grossll <A> - set lowest channel number of gross calc. region,

set grossul - set upper limit (highest channel number) of gross

calculate region.

Notice that the grossll and grossul subcommands are denoted as

"left" and "right" respectively on the menu buttons.

There are also some useful scripts included within the nuvi installation.

They can be viewed and edited by using the text editor; they are all

located in the \nuvi\ directory.

Here are some of them:

sm <N> - smooth the spectrum in buffer N. The script chooses different filtering coefficients depending on spectrum length.

am <N> - automark the spectrum in buffer N. It sets search parameters depending on spectrum length. Old ROIs will not be deleted; normally one would first delete all ROIs in this spectrum.

cn <A> - connect buffer A to display B,

shot - creates a screenshot in .gif format, puts it in \wwdir\shot.gif.

7.2.2.2. Mouse functions within a PHA spectrum window

Clicking any of the numeric dials will copy their contents to the clipboard (pastable in the command line by ALT-I).

Left click within the spectrum places the marker on the clicked position.

Left click and hold within the spectrum initiates zoom drag; moving left and right pans the spectrum, moving up and down zooms/unzooms it by its X axis.

Right click and hold modifies the top or bottom display limit; this switches the respective autoranging off. Click the top or bottom limit dial to set which limit is to be modified by right click & drag.

Click and hold both keys creates a new ROI. Drag left or right to the desired end of the ROI, release both keys to finish.

If at the beginning the left key is pressed first, all ROIs overlapped by the new ROI while dragging will be deleted. If the right key has been clicked first, the overlapped ROIs will be left as they were.

8. Spectrum evaluation

This description is based on the following assumptions:

- a nuclide library exists in buffer 13
- a gaussian curve is loaded in buffer 14.

This is the case practically in all shipped configurations, the begin.bat script loads nuclides.nlb in buffer 13 and gausscur.spc in buffer 14.

8.1. Evaluation command sequence

For the sake of description simplicity, let us evaluate the spectrum in buffer 1 using display window A to view our proceedings.

- ensure access to the spectrum to evaluate
- a <cr> - select current display window
- 1 <cr> - make sure to connect buffer 1 (where the spectrum is) to window A

- mark regions of interest to be used for evaluation (these may have been loaded with the default spectrum prior to starting acquisition for routine measurements, done manually etc.).

If the regions of interest have been previously prepared, skip this step.

To do it automatically, issue the following commands:

- roi del_all <cr> - delete all ROIs in the spectrum
- am 1 <cr> - automatically mark the peaks in the spectrum. Some manual interventions may be needed afterwards, depending on statistics

or other factors.

- if the spectrum has not been energy calibrated, do it now.

- create a peak list

new peak_list 3 <cr> - create a new peak list, place it in buffer 3

3 <cr> - connect buffer 3 to current window to view the peak list

identify <cr> - identify nuclides, assign each recognised peak a nuclide

- select efficiency calibration to be used

@setg effname <name> <cr> - name is the filename under which the efficiency table has been saved in \nuvi\data\ .

If this step is omitted (and has never been done since boottime) the default.eff table will be used.

- create a sample report

new report 4 - create a sample report, place it in buffer 4.

To see the activities in the report referenced to a particular date, click and edit either the global report date on top of the report (this will refer all nuclide activities to this moment of time) or click and edit each nuclide individually to see its activity at a particular moment of time. All dates are initially set to the moment the spectrum acquisition was started. Correction for activity drop during acquisition (especially needed for short lived nuclides) is always done.

8.2. Efficiency calibration

To create a new efficiency table, create a sample report with the supplied default.eff table (it gives a 100% efficiency at any energy). Click and edit the activities of all nuclides in the report to set them to the actual (usually taken from the source certificate) values. Save the efficiency table:

write efficiency <filename> <cr>

9. Using the DDI (Digital Detector Interface)

9.1. General

The DDI comes with an analog input for the detector signal and a gate input allowing gating the input on a per event basis.

On the netMCA-3 the DDI is accompanied by a detector power block, which consists of a power output featuring +/-12V and +/-24V for the detector preamp and a HV bias output, plus a HV shutdown sense input and RxD and TxD of a serial interface meant for communication with detectors which are capable of it.

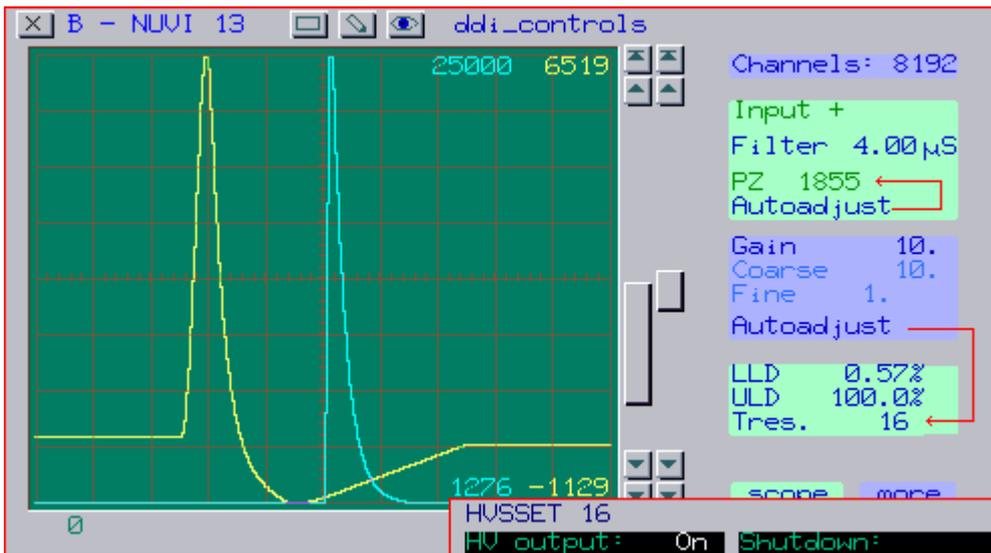
9.2. The DDI control menu

The DDI control menu is invoked from nuvi using the "setup" command, which is available while a spectrum or peak object is connected to the current nuvi display. Press setup -> ddi_2; this will create a ddi_setup in buffer 9 and connect buffer 9 to display B. Both the display and the buffer number can be changed by the user by editing the \nuvi\ddi_2.bat file.

The DDI will save its state upon exiting nuvi and will restore it upon its start; the user can also save and restore the state by using the load/save buttons on the ddi_setup menu.

The DDI control menu consists of several pages; main, detailed filtering and timing which control the MCA related parameters, scope which controls the oscilloscope part of the menu and debug which is only for TGI debugging/maintenance purposes.

9.2.1. Main "MCA" page.



9.2.1.1. Channels

Sets the conversion range, allows up to 16384 channels. Setting the conversion range to a larger value than the connected spectrum length has will result in events falling above the spectrum upper limit being discarded.

9.2.1.2. Signal polarity

Choose the correct one for the detector.

9.2.1.3. Filter

Specified in microseconds. Sets the signal filtering the closest to what an analog shaping amplifier from the old days would do at this shaping time. There are a lot more details about the filtering available on the subsequent MCA related pages - but often using the factory settings with this parameter only will produce excellent results.

9.2.1.4. PZ

Pole zero. One can set the PZ compensation value manually (and watch the result in the oscilloscope trace at the left) or have the PZ being adjusted automatically by clicking "autoadjust". For the autoadjust to work properly, make sure to have the rest of the settings in a reasonable range; gain should be set such that at least a few events/second which would convert in the second half of a spectrum would come in, and of course input polarity has to be set correctly. The input count rate should not be much higher than 1000 - 2000 cps.

It is convenient to use the Y autoranging feature for the input scope trace to verify there are pulses high enough (> apr. 16000) and that there are not too many pulses clipped on the top (i.e. gain is not too high).

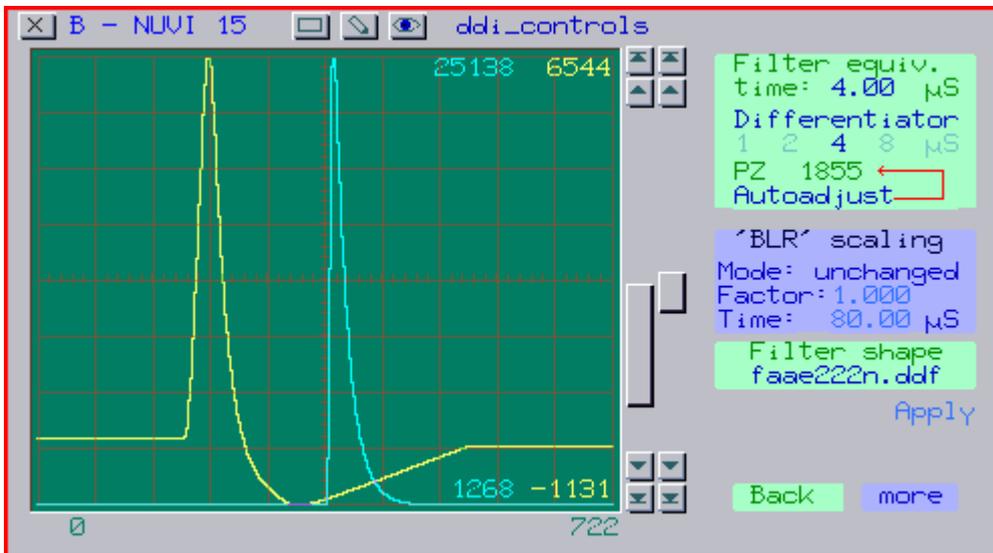
9.2.1.5. LLD, ULD

These are the post-conversion lower- and upper level discriminator values. Set them in % of the fullscale conversion range; setting them to 0 and 100 respectively is equivalent to switching them off.

9.2.1.6. Threshold

this is the fast channel event recognition threshold. It is in internal conversion units and is autoadjusted every time the gain is changed; it can also be manually made to autoadjust by clicking the button marked so. This can be practical if the input signal characteristics have changed significantly, say, changing the detector. It can also be manually set - which may be useful during experimentation with some new setup for experienced users, but generally it should be left to autoadjust for normal usage.

9.2.2. Filtering MCA control page.



It allows more subtle changes to the used filtering. While it may be necessary to use it to extract the last bits of performance from your detector, it can rarely be useful before things have been set to work in some reasonable fashion from the main MCA page alone (provided the filtering and timing pages are with their factory settings).

9.2.2.1. Filter equivalent time

This has a similar effect like the "filter" setting on the main MCA page, but it does not change the differentiator setting. The filter is rescaled and all relevant parameters (except those explicitly locked in the timing page) are updated.

9.2.2.2. Differentiator

Sets the (hardware) input differentiator. The standard values the DDI comes with on the netMCA are 1, 2, 4 and 8

microseconds. Just click the desired value and it will be switched to. Notice that modifying the "filter" parameter on the MCA main page will also set the closest differentiator value (and will thus modify this setting).

9.2.2.3. PZ

same as on the main MCA page, provided under the differentiator for convenience as changing its value will also usually if not always require PZ being adjusted.

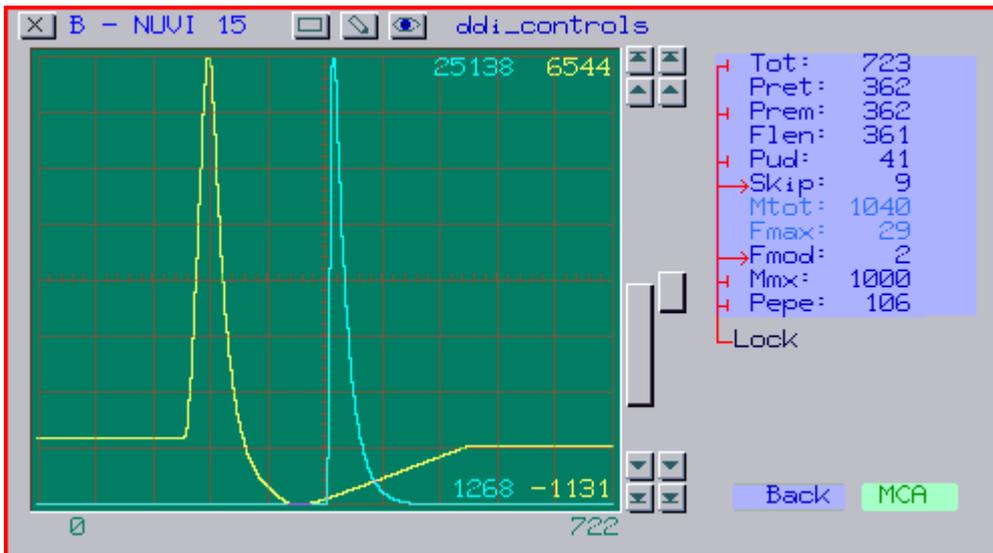
9.2.2.4. "BLR" scaling mode

Influences how the filter will be scaled down for the chosen equivalent time. "Unchanged" means the entire filter will be scaled the same, "by factor" means the "BLR part" of the filter will be scaled such that it will be times "factor" the length of what it would have been if "unchanged"; "fixed time" will set the "BLR part" of the filter being fixed as time (specified in uS). Notice that "BLR" is highly conventional in this context, there is nothing like the old days' BLR hardware on the DDI, all a BLR used to do is done by software.

9.2.2.5. Filter shape

The filter shape can be chosen from here. Various filter shapes come readily available with the netMCA as .ddf files, and newer (which were non-existent etc. upon purchase) are available in netmca updates or manually from tgi-sci.com. For detectors which used to be happy with 2 to 6 uS shaping time the faae222n filter will likely be the most suitable; it may not work so well at 1 uS, the 1us.ddf is a better choice then. The 6us.ddf has been tested to work very well with high efficiency detectors, 70+ %. For high-resolution Peltier cooled Si detectors the 1us.ddf filter has also demonstrated excellent resolution and throughput. Filters can be edited using the filter editor object; while the editing is graphics and on-screen, mark-drag etc., it takes some experience and expertise to produce useful results with it.

9.2.3. Timing MCA control page.



This is the third of the MCA control pages. It takes a high level of expertise to be able to utilize it; most users will practically never have to use it, except perhaps for the "prem" parameter.

Whenever the filtering has been modified - either by the "filter" parameter on the first MCA page or by (re)applying the parameters at the second MCA page, these parameters will be recalculated and updated. Some of them allow being locked; if locked, a parameter will not be changed after the above mentioned recalculation is done. Normally, all the parameters will be unlocked except for "skip" and "fmod", which are normally locked and set to 9 and 2 respectively.

9.2.3.1. Tot

total event length, number of samples. This is the total length which will be seen in the oscilloscope window.

9.2.3.2. Pret

Pretrigger length, number of samples. In the scope window this will be the number of samples preceding the event edge which caused triggering to occur.

9.2.3.3. Prem

minimum pretrigger length, number of samples.

Must be between 1 and the value of "pret"; it determines the number of edge-free samples prior to the event edge. If there is an edge between the trigger edge and the beginning of the "prem" interval, the event will not be converted (will be considered "piled up").

This is the only parameter on this page which may be worth tinkering with in real life; setting it to the highest possible value (that of "pret") will result in more rejected events but perhaps somewhat better resolution. A good rule of thumb is to set it initially highest and then gradually reduce it by watching the throughput and resolution. For very low count rates (below 1000 cps) there is not much if anything to be gained by setting it below the pret value.

9.2.3.4. Pud

pile up detect interval, number of samples. Sets the number of samples past the trigger edge within which an edge detection should cause the trigger event to be rejected.

9.2.3.5. Skip

internal parameter, must be set to 9. 8 and 10 can perhaps be useful values to try out at high count rates in relation with pileup pulse pair resolution, but this should be done last, after the rest of the settings have been optimized.

9.2.3.6. Mtot

highest value "Tot" may take, non-editable.

9.2.3.7. Fmax

internal parameter of no value to the user.

9.2.3.8. Fmod

filtering mode. Default value is 2; 1 and 0 are also possible and would utilize somewhat less computing power than 2, 0 being the least. It is recommended to keep this at 2, the power benefit is well below 30% for 0 vs. 2.

9.2.3.9. Mmx

Internal operation setting, must be within 500 to 32767. Keeping it to the factory set value should work OK in all practical cases.

9.2.3.10. Pepe

Pulse edge to pulse edge distance. An event coming sooner than this value after the trigger edge will be rejected.

9.2.4. (Oscillo)Scope page.

This is the oscilloscope trace control page. Triggering can be set to either "norm" or "single" shot, both equivalent to what these terms mean in a conventional scope. Notice that the scope sweep will not start after starting the menu, "norm" must be pressed once to do that even though the last setting was "norm".

The green scope trace shows the raw sampled signal with a range between 0 and 32677. The sampling is actually 14 bits but bit 0 of the shown sampled data (the least significant bit, that is) represents the state of the gate input, being 0 if the gate is open (open circuit on the gate input) and 1 if the gate is closed (gate input shorted).

The yellow scope trace shows the result of the signal filtering. It is time shifted so it can be shown in the same window, i.e. there is no valid timing relationship between the two traces to be seen.

Setting the scope limits is possible from the command window using the `1_top`, `1_bottom` (green trace), `2_top`, `2_bottom` (yellow trace) commands, as well as `left_limit` and `right_limit`.

The X range can be dragged by left clicking within the scope window and moving the mouse up/down to change the x scale and left/right to pan through the trace.

Y autoranging is separately controllable for the top and bottom limit of each trace by the tabbed arrow buttons. Moving a slider or clicking a scroll button will switch both the top and bottom autoranging off for that trace.

Right clicking and dragging within the scope area will modify the Y scale for the trace last addressed by the sliders/buttons or clicked onto.

9.2.5. Debug page

while this page is left accessible it is only meant for TGI online assistance/support purposes. Changing any DAC value will disrupt normal operation; changing the HV driving DAC CAN RESULT IN DAMAGE to your detector.

9.3. Detector power block control & monitoring menu.

```
HVSSET 16
HV output: On Shutdown: On Preamp: Overcurrent: On
Target: -3500 V Input: 4.863V +24: 24.14V Current: 10µA
Set to: -3501 V Upper level: 6V +12: 11.84V Voltage: Off
Sensed: -3510 V Lower level: 4V -12: -12.15V Rate ↑: 100V/S
Current: 0µA Pullup: On -24: -24.00V Rate ↓: 0V/S
```

This menu is used to control the HV output and its automatic shutdown conditions. It also allows monitoring of the HV output current and voltage, the voltage at the shutdown input and the voltages at 3 of the preamp power outputs (-24V is not monitored).

The sensed HV value is of worse precision than the regulated one (8 bits vs. 12 bits), it is meant to give a general feeling only.

The output current is measured with an approx. 200nA resolution, which is way above what a HPGe detector would draw.

It can be useful detecting some disastrous failure in the detector or cabling.

The HV is set to 0 every time the netMCA is reset, including via command line reset.

The HV settings can be saved to a file and restored subsequently; also, the settings upon exiting nuvi are saved and will be restored upon the next nuvi start. The output voltage and state will not be affected by starting/quitting nuvi, the HV state is controlled and maintained by a task started the first time nuvi is started after boot and left alive until reset or power down.

9.3.1. HV output

click this to turn the HV on and off.

9.3.2. Target

target voltage for the HV output. The "set to" value will be driven towards it at the allowed speed (set via rate ^ and rate v parameters) until reaching the value.

9.3.3. Set to

value currently set to the HV source, automatically driven to follow the "target" value.

9.3.4. Sensed

sensed value. Notice that the sensing precision is much worse than the actual set precision, this is just for monitoring purposes.

9.3.5. Current

the current drawn from the HV output, in uA.

9.3.6. Shutdown

click this to enable/disable the shutdown input and to recover from an activated shutdown condition.

9.3.7 Input

Voltage measured at the shutdown input.

9.3.8. Upper level

if the shutdown input exceeds the voltage set here, the HV will be shut down, this being indicated in its on/off parameter colour.

9.3.9. Lower level

if the shutdown input falls below the voltage set here, the HV will be shut down, this being indicated in its on/off parameter colour.

9.3.10. Pullup

switches on and off a pullup of 3k to the shutdown input. The pullup puls towards 5V.

9.3.11. Preamp voltages

shows the sensed values of the +24, +12 and -12V preamp power outputs. The -24V is not sensed (thus shown gray on the menu).

9.3.12. Overcurrent

toggles the HV output overcurrent protection on/off. While the typical detector bias consumption will be much below the detectable level of the current sense circuit, it is wise to keep this on with some sane value so the HV will be shut down in the even of some failure, short etc.

9.3.13. Current

the current above which the overcurrent protection will be triggered. With the standard HPGe HV output type it is factory set to 10 uA; it can be set to somewhat above 100 uA (considering 100 uA fullscale; this may differ vastly on other HV output options).

9.3.14. Voltage

voltage deviation of the sensed vs. the set value at which the HV will be shut down.

9.3.15. Rate ^

highest rate at which the set value will be increasing while trying to reach the target value, in V/S. If 0, no speed

limit will apply (i.e. the set value will instantly become equal to the target value). While the set voltage is $< 100V$, the rate up may be additionally limited to allow for initial transients in the control loop.

9.3.16. Rate v

highest rate at which the set value will be decreasing while trying to reach the target value, in V/S. If 0, no speed limit will apply (i.e. the set value will instantly become equal to the target value).

Appendix A – Detailed update and reinstall procedures

The netMCA update and reinstall disk images are http downloadable here along with the respective usage procedure descriptions.

[Automatic reinstall and/or update](#)

[Manual \(re\)install of the system HDD partition 0](#)

[Recovering a non-accessible HDD partition 0](#)

[Recovering HDD partitions other than 0](#)

[Updating the CPLD](#)

A.1. Automatic update and/or reinstall of the system HDD

If the netMCA is connected to the Internet it can update itself by issuing a single command.

It is also possible to completely reinstall the system HDD partition by a single command.

The commands are "update" and "reinst" respectively and must be typed in the shell window as the first thing after reset.

On the very first shipped devices (version/revision prior to 1_3) these two commands may not be found;

in this case, the [manual reinstall procedure](#) must be followed once.

[hdd06sys.disk](#) - latest complete disk image of #scsi\0:, apr. 10M

The easy way to install this is by using the reinst script on the netMCA.

reinst --will download the file via ftp and install it; will ask whether to preserve the active sset.xobj.

The script can also be redirected:

reinst domain%host% user%login name% pass%password%

Where host is host name or IP address, login name is the login name to the ftp account where the file - hdd06sys.disk - will be downloaded from, and pass is the password to that account.

Defaults are:

host: tgi-sci.com

user: netmca

pass: update

Specifying user and pass only makes sense if host has been specified to something other than tgi-sci.com,

e.g. the IP address of a local machine where a dedicated ftp account has been setup.

A.2. Manual (re)install of the system HDD partition 0

Reinstallation can also be done manually - if you don't have the "reinst" script or for whatever reason.

To do this, download this image using ftp to any other drive than #scsi\0: ; restart the netMCA to make sure there are no tasks running which will be accessing drive 0:, then copy the image to #scsi\0:.

Here is an example:

restart the netMCA

md b:newimage --prepare some destination directory (or use some existing, just not on 0:)

md b:newimage

cd b:newimage --set the destination directory to be the current one

ftp

name netmca

pass update

op tgi-sci.com

get hdd06sys.disk

quit

copy hdd06sys.disk #scsi\0: --install the image - old data on 0: is lost.

2:syslib\commands\reset.xxx --reset the netMCA - must be done, 0: has been overhauled "under" the filesystem

Notice that this will result in overwriting also 0:sset.xobj - where the boot settings are.

If you want to preserve these settings - say, you have set DHCP to N in the EEPROM settings and you have set a static IP in 0:sset.xobj - you can do the following before issuing the copy command in the above sequence:

install 0:syslib\devices\fdk --install the file disk image driver

setfdk 0 hdd06sys.disk --mount the newly loaded image to =fdk\0:

copy 0:sset.xobj =fdk\ -w --copy the active settings to the new image

setfdk 0 -l --flush =fdk\0: cache, unmount disk image

Now do the actual installation,

copy hdd06sys.disk #scsi\0: --install the image - old data on 0: is lost.

2:syslib\commands\reset.xxx

A.3. Recovering a non-accessible HDD partition 0

[hdd06p.disk](#) - blank recovery image of #scsi\0:, apr. 50k

This image is needed to recover drive 0 if it has become non-accessible.

If it has become non-bootable (LED won't turn red >20 seconds after poweron) the device must be booted off the internal ROM disk, drive 2.

If the HDD partition 0 is accessible - e.g. **dir 0:** works - the [manual reinstall procedure](#) will probably be sufficient.

If the HDD partition 0 is not accessible - returns "Invalid data format" to **dir 0:**, here is the last-resort recovery procedure to be followed:

Boot off drive 2 - the ROM disk, on the internal flash, download this image - [hdd06p.disk](#) - and copy it to #scsi\0: .

This will recover drive 0 to a completely blank state (non-bootable) and will recover access to all other partitions.

Here is the detailed sequence:

1. Unpower the netMCA
2. Insert the D9 cap into the D9 connector
3. Power up the netMCA, connect to it via RFB/VNC as usual; it will have booted off 2:
4. Download the blank disk image:

cd =ramd\ -- will use RAM disk to download the disk image

ftp

name netmca

pass update

op tgi-sci.com

get hdd06p.disk

quit

copy hdd06p.disk #scsi\0:

This will make 0: on the HDD accessible and will restore physical access to all other partitions on it.

Notice that the HDD will not yet be bootable.

5. Reset the netMCA - either repower it or type "reset.xxx"< enter>

6. Make sure some HDD partition other than 0 is accessible and has enough space on it for hdd06sys.disk, e.g. b: .

7. Install hdd06sys.disk on #scsi\0: using that partition:

md b:newimage --prepare some destination directory (or use some existing, just not on 0:)

cd b:newimage --set the destination directory to be the current one

ftp

name netmca

pass update

op tgi-sci.com

get hdd06sys.disk

quit

copy hdd06sys.disk #scsi\0: --install the image - HDD is now bootable

8. Unpower the netMCA

9. Unplug the D9 cap from the D9 connector

10. Power up the netMCA - it must now boot off drive 0 as usual.

A.4. Recovering HDD partitions other than 0

If a partition other than 0 has become unaccessible, it can be recovered to a blank state by using some of the images found in 0:syslib\devices\diskimg\ . The netMCA-2 comes with the following partitions (defined in block 0 of hdd06p.disk and/or in hdd06sys.disk): 0: - 815M, B: - 8G, C: - 20G, D: - 20G, E: - 20G, F: - 40G. 0: recovery is explained extensively in the rest of this document.

To recover B:, issue:

```
copy 0:syslib\devices\diskimg\hdd8g.disk #scsi\b:
```

To recover C: issue:

```
copy 0:syslib\devices\diskimg\hdd20g.disk #scsi\c:
```

To recover D: issue:

```
copy 0:syslib\devices\diskimg\hdd20g.disk #scsi\d:
```

To recover E: issue:

```
copy 0:syslib\devices\diskimg\hdd20g.disk #scsi\e:
```

To recover F: issue:

```
copy 0:syslib\devices\diskimg\hdd40g.disk #scsi\f:
```

Beware - use only some of the above copy commands exactly as specified.

Any modification may result in data loss.

A.5. Updating the CPLD

The ADC interface CPLD is not updated automatically during the system updates. This is done so because it is rarely if ever changed and has only a limited number of allowed programming cycles.

To update the CPLD, enter:

```
updcpld
```

This will update the CPLD and its version information.

Again, the CPLD has only a limited number of programming cycles.

Normally it will rarely if ever be necessary to reprogram it.