Bastian Löher

Photomultiplier Control Software for the Low-Energy Photon Tagger NEPTUN User's Manual

Mini Research Project

Nuclear Physics Institute
Technical University Darmstadt

September 2007

Abstract

This manual contains practical information on how to use the control software for the photomultipliers at the low-energy photon tagger NEPTUN. The software controls the photomultipliers' voltage settings and provides diagnostic tools for easy supervision during beam time. In parts the software is based on the work of Michael Elvers.

It is split into two major parts to provide convenient remote access to the photomultipliers. The server part takes care of communication with the VME crate via a TinyCAN module, while the client part offers either a console based or graphical user interface. Client and Server communication is done via UDP protocol. The graphical client is able to save the current voltage settings to disk and reload them on demand.

Contents

1 Software

1.1 Introduction

The software is split into four smaller applications. **PmServer** takes part of crate-client communication. The **PmClient** module provides a console interface or works in batch mode. **PmLogger** is a simple logging client and **PmControl** is the graphical user interface written in Qt version 4.[1]

1.2 PmServer

1.2.1 Introduction

This is the main server application for controlling the NEPTUN Tagger photomultipliers. It receives commands via UDP from its clients and dispatches them through the TinyCAN module to the crate. It also recieves messages from the TinyCAN module and broadcasts them to the clients. The server must be running if clients want to connect to it.

The reason why we have a server in the middle and do not send any commands directly to the module is the fact that the module cannot handle large amounts of commands in quick succession. The server's main purpose is to moderate the TinyCAN input and output and also provide some environmental amenities for the client user.

1.2.2 Syntax

Use this command to directly start the server:

```
server [ -p port ]
```

or call the script this way:

```
./pmserver
```

1.2.3 Quick Start

1. Make sure that the file **remoteport** exists in the same directory as the server executable. If this is not the case, create a simple text file and name it remoteport. In the first line enter the UDP port the server should use. The default port number is 50000.

[1] The Qt framework provides easy to use and platform independent classes for programmers.

2. Type the command `./pmserver` to call the starting script. The server will now try to start. If the server starts successfully then proceed to the client Quick Start in section 1.3 on page 5

3. If the server does not start read the error message carefully and check the section Errors (1.6.1).

1.2.4 Options

The command `server` starts the server. It has only one option:

-p Specifies the UDP port number that the server will listen to. The port should be a free system port in the range from 1024 to 65535. The clients must then connect to that port for the handshake process to take place. The server's response will come through the next higher port. The client should listen there. During handshake the client will receive its own port number out of the server port range (Check server output) for further communication. Example: Default port number is 50000, so in this case the response comes through 50001. The first connecting client will then be assigned the first free port number in the port range, which is 50005.

1.2.5 Remarks

It is useful to also start the logger client after starting the server. This will log all output to a file. Check the PmLogger section (1.4, page 8) for details.

The server can only be started once. If one instance of the server is already running, any attempt to start another one will result in an error message.

Quitting the server can in some cases leave the voltage turned on. Please make sure to always use the `pmks` command to quit the server. You can verify a proper server shutdown by checking that the LEDs 5V (green) and 24V (yellow) are switched off. These LEDs can be found on the photomultiplier crate front panel.

1.3 PmClient

1.3.1 Introduction

This is the main client application for controlling the NEPTUN Tagger photomultipliers. When run, it starts the client that connects via UDP protocol to the server. The server should be running and its IP address and listen port should be known. Upon start the client will read the server configuration and provide the user with a console based interface. The user may then enter commands to control the server or the photomultipliers. If the 'b' option was set, then only one batch command will be executed and the client will close automatically afterwards.

1.3.2 Syntax

Use this command to directly start the client:

`client [-a address] [-p port] [-b command] [values]`

or call the script this way:

`./pmclient`

1.3.3 Quick Start

1. Make sure that the files **remoteport** and **remoteip** exist in the same directory as the client executable. If this is not the case, create simple text files and name them. In the first line of the respective files enter the UDP port the server uses or the IP number of the machine the server is running on. The default port number is 50000 and the default IP is 127.0.0.1, which is the local loopback address.

2. Make sure the server is running. To do this, either you know that the server is running or `ssh` to the server machine and `ps -e | grep server` to see if a server process is active.

3. Type the command `./pmclient` to call the starting script. The client will now try to start and connect to the server. If the client starts successfully then you should be presented with a command line. Here you can enter any of the commands described below (1.3.5)

4. If the client does not start, read the error message carefully and check the section Errors (1.6.2).

1.3.4 Options

The command `client` starts the client. It has several options:

-a Specify a custom IP address that the client will use for connecting to the server. Default is 127.0.0.1, the local loopback address.

-b Specify the command that the client should execute directly on startup. In this mode the client will terminate after receiving a response. The response will go to standard output. Check the list below (1.3.5) to learn the various commands.

-p Specifies the UDP port number that the client will connect to. The port should be a free system port in the range from 1024 to 65535. The server's response

will come through the next higher port. The default port number is 50000, so in this case the response comes through 50001.

values Values are only necessary if using batch mode. These values will be used in conjunction with the supplied batch command. Check the list below for the correct syntax.

1.3.5 Commands

Client Control

rm <portNumber>
Unregisters a dead client with the portNumber given.

quit (q)
Unregisters the current client and quits the console.

Server Control:

config
Reads the photomultiplier configuration from the server and prints it out. The server checks if the "pmtables.dat" file has changed and updates its data beforehand.

killserver (ks)
Remotely quit the server. The connection to the TinyCAN module will be closed properly and the lockfile will be deleted. All clients must then be terminated manually using the `quit` command.

who
Prints all the connected clients as stored in the server memory.

Photomultiplier Control:

gethv
Reads the photomultiplier settings from the TinyCAN Module. Returns voltage, currents, and threshold values.

send <command>
Sends the command 'as is' directly to the CAN Module.

sethv <voltage> [<pmnr>]
Sets the voltage of photomultiplier pmnr to given voltage. If no photomultiplier is specified, the voltage will be applied to all photomultipliers.

switch [<on|off>]
Switches the photomultipliers on or off. If no argument is given switch defaults to off.

1.3.6 Batch Scripts

In the same directory as the client executable exist several batch files named `pm*`. These scripts can be run directly from the command line and execute a server command according to their name.

For example `pmgethv` will connect to the server and get the current photomultiplier settings. This is equal to running `pmclient`, typing the command `gethv` on the command line and quitting the client.

1.3.7 Remarks

The maximum number of clients that can connect to the same server is limited to 20. This should be enough for most applications.

All clients receive all output that originates from the TinyCAN module. This way all clients will also receive error messages.[2]

1.4 PmLogger

1.4.1 Introduction

This is the logger client application logging any activity at the NEPTUN Tagger photomultipliers. When run, it starts the logging process that connects via UDP protocol to the server. The server should be running and its IP address and listen port should be known. Upon start the logger will read the server configuration and output all messages the server sends to its logger port. The logger port is defined as the Maximum port in the server port range. Check server output for details.

1.4.2 Syntax

Use this command to directly start the logger:

```
logger [-a address] [-p port]
```

there is no starting script

[2] The asynchronous nature of the communication requires this measure.

1.4.3 Quick Start

1. The default port number is 50000 and the default IP is 127.0.0.1, which is the local loopback address. Make sure you know your server IP and port. Usually the logger is running on the same machine as the server, but this is not necessary.

2. Make sure the server is running. To do this, either you know that the server is running or `ssh` to the server machine and `ps -e | grep server` to see if a server process is active.

3. Type the command `./logger` to start logging. Use the `-p` option to specify a port number. For example type: `./logger -p 3666`

4. If the logger does not start, read the error message carefully and check the section Errors (1.6.2).

1.4.4 Options

The command `logger` starts the program. It has two options:

-a Specify a custom IP address that the logger will use for connecting to the server. Default is 127.0.0.1, the local loopback address.

-p Specifies the UDP port number that the logger will connect to. The port should be a free system port in the range from 1024 to 65535. The server's response will come through the next higher port. The default port number is 50000, so in this case the response comes through 50001.

1.4.5 Remarks

You can start the logger and write the logging output directly to a logfile using this command:

```
./logger >> logfile 2>> %1.
```

This way the logger output is appended to the file.[Dit03]

1.5 PmControl

1.5.1 Introduction

PmControl is the graphical user interface for controlling the photomultipliers. Note that this program must be started from the command line. The graphical user interface provides almost all functionality that the command line tools

'client' and 'pm*' have. The Photomultipliers' properties like Voltage, Currents and Threshold value can be checked. The photomultipliers can be switched on and off and their voltage can be changed either one by one or simultaneously. The interface can be locked to avoid accidental user input. All settings can be saved to settings (*.psf) files and can also be restored from those files. Use the Quick Start info below to get started with the program, or use the really small built-in help (not recommended).

1.5.2 Syntax

On the command line type:

```
pmcontrol
```

1.5.3 Quick Start

1. Open up the program window (`./pmcontrol`).
2. Go to the 'options' tab and check that the settings are correct. If not then change the settings according to your server settings.
3. Check if the server is running.
4. Hit the connect button. The connection should be established and the values should get refreshed automatically.
5. Set the voltages one by one or use the Main Voltage Box below the tabs to set all Voltages to the same value.
6. Hit the '»' button. The settings will be sent. Wait until automatic refresh is done.
7. Now use either the 'All On' button to switch all Photomultipliers on, or flick the red switches individually to switch only single photomultipliers.
8. Use the Lock Menu to Lock the interface or press Ctrl-L.
9. Save your settings via the File Menu.
10. Use the overview tab to check on all photomultipliers at once.

1.5.4 Menus

File > Open
Open a settings (*.psf) file from disk (File format in 3 on page 15).

File > Save
Save a settings (*.psf) file to disk.

File > Quit
Quit the application.

Lock > Lock
Lock the user interface. No Input can be made in this mode.

Lock > Unlock
Unlock the interface again.

Help > About
Display the about box.

1.5.5 Controls

There are several buttons at the interface bottom:

Connect/Disconnect
This is the most important button. It opens and closes the connection to the server.

All on/All off
These buttons switch all photomultipliers on or off.

«/»
The « button inserts all current values into the spinbox voltage edit boxes. When the values have been set, the » button sends the new values to the server.

Refresh
Refreshes all values. The on/off state is updated as well.

Set
Press this button to assign the value in the Main Voltage Box to all photomultipliers. This is useful for setting a common level before finetuning the individual voltages.

Each photomultiplier has individual controls:

On/Off
Switches this photomultiplier on and off (instantaneously).

«/»
« Grabs the current voltage and fills the edit box, while » sends the value in the edit box to the server.

Progress bars
The progress bars display (from left to right) the currently set voltage (in V), the photomultiplier threshold and the positive and negative currents.

1.6 Error Messages and Troubleshooting

This section covers most of the error messages the applications can produce and shows how to handle them.

1.6.1 Server Messages

Unable to open 'pmtable.dat'
The file 'pmtable.dat' could not be found. Either the file does not exist or the server application has no permission to read it. Refer to 3 for details on the file structure. This error may also occur during runtime. The best solution in this case is to restart the server and all clients. This error has not yet been resolved.

Warning no other cratenumber than 0 supported, omitting line
Only one crate is supported so far.

Warning pmnr xx is invalid, omitting line
The photomultiplier number in this line is not valid. The valid range is 0 to 127. Chose a valid number to avoid this warning.

Warning pmnr xx already defined, omitting line
The number is already defined in the file. To avoid ambiguities, this line will be omitted. Delete the line to avoid the warning.

Warning position xx is invalid, omitting line
The position is invalid. Only positions in the range 0 to 7 are valid. Change the setting to avoid this message.

Unable to initialize can (errorcode xx)

The can module could not be initialized. Unplug the USB cable, and reconnect it. This should solve the problem.

Unable to open can devicenumber xx (errorcode xx)

The can device can not be opened. Try reconnecting it. If this does not help, restart the computer. This message can also occur when the server is already started.

Server already running!

This message occurs when the server is already running or has not been quit properly. Use the `./pmks` command or the `Ctrl+C` key sequence to quit the server properly (Using `Ctrl+C` can be unsafe, so be careful). If the message still occurs, delete the lock file located in `/tmp/pmserver_lock`. This should solve the problem.

1.6.2 Client and logger Messages

Bind failed!

This message occurs when either the server is having troubles or when the client tries to use the same port as another one. Quit your current client and try again. If this fails, restart the server and try again.

2 Future Development

Some ideas for future development of the software.

- Enable the console client to understand and save *.psf files.
- Enable the PmControl to shutdown the server.
- New tab in the PmControl for UDP message output.
- Fix 'Segmentation fault' bug when closing PmControl.
- Fix crash in PmControl when clicking Watch tab.
- Notify PmControl of server shutdown.
- Automated voltage setting based on energy histogram.

References

[Elv07] M. Elvers, Aufbau und Entwicklung eines Datenaufnahmesystems für den Niederenergie-Photonentagger NEPTUN, Diploma Thesis, Institute of Nuclear Physics, Technische Universität Darmstadt, unpublished (2007).

[Dit03] P. Ditchen, Shell-Skript Programmierung, mitp (2003).

[Ker85] B. Kernighan, D. Richie, Programmieren in C, Carl Hanser (1985).

3 Appendix A: Formats

TinyCAN command format

The format for the TinyCAN module follows this general structure:

Command	Length	Data
4 Byte	1 Byte	Specified by length

The command is constructed from the position of the module it controls and the type of command. Brakedown of the four bytes:

1	2	3	4	
Crate	High Module Byte	Low Module Byte	Command byte	Data
starts at 06	starts at 04	starts at 04	00 -	—
			⋮	⋮
			0B - Read switch status	—
			0C - Switch on/off	00 / FF

Settings file format (*.psf)

The Qt client saves the photomultiplier configuration in a settings file. Usually the settings are saved with the current date as filename.

# PmNr	Id	OnOff	Voltage
0	0x06044000	0	999.98
1	0x06044000	1	999.98
2	0x06044000	0	999.98
⋮			

pmtable.dat file format

This file stores information about the connected photomultipliers.

#nr	crate	platine	position
0	0	0	0
1	0	0	1
2	0	0	2
⋮			

More information in [Elv07] (section 2.8.3)

4 Appendix B: Client-Server communication

Structure

The communication between client and server follows this simple structure:

Port Number	Number of Commands	Command(s)
2 Bytes	1 Byte	Specified by length times command length

Commands

The command can be any of these listed below:

Command	Parameters	Description
0	L	Client connects to server use the L option to connect as logger
1	—	Client disconnects from server
2	[1|2|3] [Command]	Sends command 1 - sethv Command 2 - switch Command 3 - gethv Command The send command uses 1 as well[3] TinyCAN command as described in 3.
3	—	Shutdown the Server properly
4	—	Get all clients Similar to the Unix `who` command
5	L	Tell Pm Configuration and assign a free port. use L option to be assigned the logger port
9	—	Unknown command[4]

Please note that numbers are transferred as ASCII numbers, meaning that for example the number '4' would be sent as ASCII code 0x34.

Example

This example command requests that the photomultiplier number 5 should be set to a voltage of 500V.

Console command

```
pmsethv 500 5
```

Server command

```
0xC3550132310604404103066666
```

Details

0x	This is a HEX number
C355	Port number 50005
01	One command only
32	Send a command
31	Sethv
0644041	ID for the first board (0x44) and sethv command
03	Length of command Data
06	Channel of Photomultiplier Channel 06 means position 5
66	High voltage byte
66	Low voltage byte

Voltage value calculation

The server can set the voltage value in a range from 0 to 1250V. The TinyCAN module only understands values in the range from 0 to 65535. Thus the input value (500V in the above example) must be mapped to a different output value via this simple formula:

$$U_o = \frac{65536 \cdot U_i}{1250} \tag{1}$$

[3] Correct implementation would require another number for the send command, but it is not necessary for the server to function correctly. It just can not distinguish between send commands and sethv commands.

[4] This command is useless at first glance, but it can be used if unknown commands should be logged, or exploited to test server connection without actually doing anything.

Acknowledgements

I want to thank Michael Elvers for his exceptionally helpful ideas and remarks, for his engagement to help me with this project, and of course for giving me this challange. I have learned alot during those few weeks.

I also thank Angelo Calci for initially mentioning my name in the Zilges group which got it all started.

Kay Lindenberg has always been there to give helpful answers to all questions I might have had regarding Linux, the Network or anything else. Thank you.

Also thanks to Linda Schnorrenberger for asking questions from somewhere on the left.

More thanks go out to the rest of the Zilges group which have been very nice and kind to me, providing a very comfortable working environment. Thanks!