multichannel systems

nsMCDLibrary
Neuroshare implementation for MC_Rack data

Multi Channel Systems MCS GmbH
Aspenhaustrasse 21
72770 Reutlingen
Germany

Fon +49-71 21-90 92 5 – 0
Fax +49-71 21-90 92 5 – 11

info@multichannelsystems.com

www.multichannelsystems.com

nsMCDLibrary Version: 3.4
Stand: 2012-01-30

Autor: Jens Pätzold

General

To access MC_Rack data files from the Neuroshare API, the data are mapped to the structure given by Neuroshare API. This API is implemented in a Windows DLL or Linux or Mac (Intel) shared library by a set of predefined functions.

MC_Rack files recorded with MC_Rack Version 4.3.2 or older could be read with this version of the library.

Up to now, not all possible data streams within a MC_Rack file could be read by the Neuroshare API.

The nsMCDLibrary.dll is implemented according to the Neuroshare API Version 1.3.

This paper only handles what is specific to data from MC_Rack. For the documentation of the Neuroshare API see there: http://neuroshare.sourceforge.net/

Installation

The Neuroshare library can be downloaded from our webpage at http://www.multichannelsystems.com/downloads.

All packages contain this documentation and the library for the respective system, example source files to access the Neuroshare library directly from your own program code, the sources for the „mexprog“ library that is needed for the interface to Matlab as well as the Matlab functions itself.

Operating System	Package Name	Neuroshare Library Name
Windows 32bit	nsMCDLibrary32_3.3.zip	nsMCDLibrary.dll
Windows 64bit	nsMCDLibrary64_3.3.zip	nsMCDLibrary64.dll
Linux 32bit (build with Ubuntu 10.04 LTS)	sMCDLibrary_Linux32_3.3.tar.gz or	nsMCDLibrary.so
Linux 64bit (build with Ubuntu 10.04 LTS)	sMCDLibrary_Linux64_3.3.tar.gz or	nsMCDLibrary.so
Mac OSX (Intel) 32bit and 64bit 'fat' binary	nsMCDLibrary_MacOSXI_3.3.tar.gz	nsMCDLibrary.dylib

We recommend one of the toolboxes FIND (http://find.bccn.uni-freiburg.de/) or sigTOOL (http://sigtool.sourceforge.net/). They contain already the necessary Matlab interface to the Neuroshare Dll.

Just unzip the file anywhere on your computer. If your FIND or sigTOOL installations have only an older version of the nsMCDLibrary.dll, just replace the nsMCDLibrary.dll with the newer one.

„mexprog“ could be compiled from within Matlab for you specific OS.

The plain Matlab interface files could be direcly used from within our package.

Implementationdetails

MC_Rack has two different recording types. Both continuous and triggered data can be handled with the neuroshare library. The mapping is a little bit different for each type, so the mapping is shown for each type separately.

Continuous data mapping

Stream	**Short name**	**Entity**
Electrode Raw Data	elec	Analog Entity
Analog Raw Data	anlg	Analog Entity
Filtered Data	filt	Analog Entity
Channel Tool Data	chtl	Analog Entity
Digital Data	digi	Analog Entity, each binary digit as Event Entity
Spikes	spks	Segment Entity
Trigger	trig	Event Entity

Other stream are not mapped up to now.

Triggered data mapping

Stream	**Short name**	**Entity**
Electrode Raw Data	elec	Analog Entity and Segment Entity
Analog Raw Data	anlg	Analog Entity and Segment Entity
Filtered Data	filt	Analog Entity and Segment Entity
Channel Tool Data	chtl	Analog Entity and Segment Entity
Digital Data	digi	Analog Entity and Segment Entity, each binary digit as Event Entity
Spikes	spks	Segment Entity
Trigger	trig	Event Entity

Please be aware that some streams are mapped to two entities. The contents is essentially the same. Other stream are not mapped up to now.

Missing streams

All other possible streams as average and different parameters are up to now not implemented in the nsMCSLibrary.dll.

Entities

Event entities

Each of the 16 bits of the digital data channel is represented as one event entity. Each change on the individual digital line results on a new event in the entity with the next index. The event information is in one 8bit (byte) value. Its value is 1 for a change from 0 to 1 on the digital line and −1 (255) for a change from 1 to 0 on the digital line.

Triggers are mapped to one event entity for each trigger stream. The event is represented in two 16bit (word) values. The first one gives the slope of the Trigger, the second the Trigger value. For manual triggers the second value gives the trigger number. If trial synchronisation is used the trigger event contains another two 16bit values. The 3rd one is the trial number and the 4th one is the stimulus number.

Analog entities

Raw electrode data, analog data and filtered data are mapped to analog entities. Each channel gives one analog entity in Neuroshare. The different gain is considered. The data are given in Volt of the original input.

The 16 bit of the digital data channel are represented in one analog entity as raw data with a range from 0 to 65535, with minimum step size of one. Each sample value is included in the entity.

For continuous data the first index is starting at time 0. Then all data are following with the next index for the next sample point. The time difference of each index is the reciprocal value of the sample rate.

For triggered data the indices are not always continuous in time. If you are reading many indices at one time, you get, according to the Neuroshare specifications, in `pdwContCount` the number of samples that are continous:

„Although the samples in an analog entity are indexed, they are not guaranteed to be continuous in time and may contain gaps between some of the indexes. When the requested data is returned, `pdwContCount` contains the number of Analog items, starting from `dwStartIndex`, which do not contain a time gap“. [NeuroshareAPI-1-3.pdf]

For analog entities there is no timestamp directly included in the API for reading the data. You get the timestamp with the function `ns_GetTimeByIndex`.

Segment entities

Spike streams are represented in segment entities. Spikes (or wave forms) are sampled independently for each channel. This means there is always only one source per segment entity. The number of sources is always 1. Each original channel from MC_Rack data forms one entity in Neuroshare. The different gain is considered. The data are given in Volt of the original input.

For triggered data all analog entities are also represented as segment entities. One segment contains the data of one sweep for one channel.

Neural event entities

There are no streams in MC_Rack data that are mapped to neural events.

Naming convention for entities

The entities are named as follow:

- 8 letter: stream name with:
    - 4 letters name
    - 4 digits stream number.
- Space
- 4 letters: HWID
- Space
- 4 digits: index of channel
- Space
- 8 letters: name of channel (right justified)

From here on, this applies only to event entities of digital data:

- Space
- 2 digits sub channel for events of digital data

for example:

```
aaaa0000 0000 0000 xxxxxxxx 00
elec0001 0001 0000       21
digi0001 0063 0001       D1
digi0001 0063 0001       D1 02
```

Configure Behaviour

The behaviour of the Dll can be configured by a file in the ini-file format. All parameters are in the section **[Settings]**.

BaseIsPressedStart configures how times are interpreted.

0: times start at 0 for each file (default and old behaviour)
1: times start at the time when the start button is pressed.

Example for the configuration file:

```
[Settings]
BaseIsPressedStart = 1
```

Location of the file

Windows: The configuration file is searched in the same directory and with the same name as the Dll has, but with the extension „.ini“ (nsMCDLibrary.ini)

Linux / Apple: not implemented yet

Missing

The function `ns_GetLastErrorMsg` is not implemented yet.