CMOS HD-MEA based file format description

This document describes the meta file format for high density multielectrode data that uses the HDF5 container as persistence layer.

# Introduction into the organization of the HDF5 file

### Experiment

Every stored data file will represent one experiment. An experiment is defined as a (not necessarily continuous) recording with a single chip. Its length is determined by the user. The experiment defines the meta information of the data file like the chip-id. An experiment contains one or more sessions.

* Experiments start with the first setting on an configuration
* Experiments end with the ending of the last session

### Session

Every experiment consists of one or more recording sessions. A recording session is defined by a common configuration of the HDMEA chip. In case the configuration (e.g. choice of electrodes that are being recorded) is changed, a new session is started.

* Sessions contain the meta information of the recording like the configuration details
* Sessions belong to exactly one experiment
* The data that is recorded during one session is stored in chunks
* Sessions contain zero or more chunks (thus they can contain no data at all)
* Sessions start with the setting of a new configuration
* Sessions end with the last sample of the last chunk they contain
* The length of a session is the time between setting of the configuration and its end
* Sessions that contain no data (and thus no chunks) have a length of 0
* Sessions do not need to cover a whole experiment, there can be gaps between sessions

### Chunk

A chunk is a continuous recording of extracellular data with a constant configuration. If data is not recorded for a period of time a new chunk is started.

* Chunks contain the actual recorded data
* Chunks belong to exactly one session
* Chunks start with the recording of data after a period of not recording data
* Chunks end with the last sample of recorded data they contain
* The ending of the last chunk of a session defines the end of the session
* Chunks do not need to cover the whole length of a session

### Configuration

A configuration is the collection of meta data that is related to the HDMEA chip. Every session has exactly one configuration.

### Examples

* 1. The probably most common way to use the setup is the following. The chip is configured with a certain setup. Data is recorded. The experiment is stopped.   
     This will lead to a data file (experiment) containing exactly one recording session with exactly one chunk. The chunk will be of the same length as the session.
  2. Another use case is the not-continuous recording of data. An experiment is started by setting a configuration for the chip. Whenever a certain event (spike, burst, trigger) occurs, data is recorded.   
     This will lead to a data file (experiment) with one session that contains many chunks of shorter length.
  3. In an initialization phase the chip is scanned with different configurations to find active electrodes. Then a certain configuration (e.g. that uses only active/desired electrodes) is chosen and kept constant to record continuous data.  
     This will lead to a data file that has many shorter sessions in the beginning. These sessions have only one chunk which reflects the recording during the initialization phase. The last session is the longest and contains also only one chunk representing the actual recording with the final configuration.

### Time Frames

Due to the experiment – session – chunk arrangement there exist different time frames inside a data file.

* The “global” time is the time related to the beginning of the experiment.
* The session time relates to the start of the session
* The within chunk time is the time inside a chunk and provides the index to define a hyperslab inside the HDF5 file.

### Matlab access to the data file

The matlab access to the data file is provided by the XXXX class. An example usage



**Fig.1:** Sketch of the data organization inside the HDF5 file

# Detailed implementation of the file format

### The root node “/”

The root node “/” represents the file. Any file wide meta information can be added to that node. Since every file contains only one experiment, this node could also contain the experiment specific information. To keep things separated, file information is stored directly in the root node while experiment information is stored in the “/experiment” node.

The file format specifies the following necessary information:

1. “/\_\_version” Version of the file format
2. “/\_\_date\_created” Date when the file was created
3. “/\_\_date\_last\_modified” Date when the file was last modified

The file format specifies the following optional information:

1. “/User/ “ contains variables that can be freely set by any user and is not used by the standard data interfaces.

The root node contains the sessions in the node “/sessions”.

### The experiment node “/experiment”

The experiment node contains the file wide information concerned with the experiment.

1. “/experiment/chip\_id” The ID of the HD chip used

### The session node “/sessions”

### The single session node “/sessions/session<session#>”

### The session configuration node “/sessions/session<session#>/config”

### The chunk node “/session/session<session#>/chunks”

### The single chunk node “/session/session<session#>/chunks/chunk<chunk#>”