

RICOH MEG Reader Toolbox

Revision 1.0

Specifications

4 June 2018

Notice

This software is written for MATLAB® version 9.3 (R2017b) onwards.
MATLAB is a registered trademark of The MathWorks, Inc.

The contents of this software are subject to change without prior notice as a result of continuing improvements to the software's performance and functions.

This software consists of following functions:

Category	Function name	Purpose
Read MEG data	getRData	Get measurement data.
Read MEG header	getRHdrSystem	Get information about system.
	getRHdrChannel	Get information about channel.
	getRHdrAcqCond	Get information about data acquisition condition.
	getRHdrEvent	Get information about trigger event.
	getRHdrCoregist	Get information about coregistration.
	getRHdrDigitize	Get information about digitization.
	getRHdrSubject	Get information about subject.
	getRHdrAnnotation	Get information about annotation
	getRHdrSource	Get information about analyzed sources.
Read MRI	getRMriHdr	Get information about header of MRI file (*.mri).
Others	getRVersion	Get information about version of this toolbox.

getRData

This function retrieves the measurement data of whole channels by the specified file path and sample range.

```
data = getRData(
    filepath,
    start_sample,
    sample_length
);
```

Arguments:

<i>filepath</i>	string	[in] File path
<i>start_sample</i>	double	[in] Start sample or trial(frame) number for retrieving data. The start number corresponding to each acquisition type is as follows : - Continuous Raw : Start sample number for retrieving data. (0 origin)
<i>sample_length</i>	double	When both <i>start_sample</i> and <i>sample_length</i> are omitted, you can get data of whole samples. [in] Sample length for retrieving data. The number of samples or trials(frames) corresponding to each acquisition type is as follows : - Continuous Raw : Number of samples for retrieving data. When this parameter is omitted or is specified as 'Inf', you can get data from <i>start_sample</i> to the end of sample(frame).

Return values:

<i>data</i>	matrix(double)	[out] double matrix of measurement data. Row : number of channels(whole channel), Column : number of samples Unit of the each channel depends on channel type as follows: Magnetometer [Tesla] AxialGradioMeter [Tesla] PlanarGradioMeter [Tesla] ReferenceMagnetometer [Tesla] ReferenceAxialGradioMeter [Tesla] ReferencePlanarGradioMeter [Tesla] TriggerChannel [Volt] EegChannel [Volt] *This has already been reflected EEG gain EcgChannel [Volt] *This has already been reflected ECG gain EtcChannel [Volt] NullChannel [Volt]
-------------	----------------	--

getRHdrSystem

This function retrieves information of the system.

```
system_info = getRHdrSystem(
    filepath
);
```

Arguments:

<i>filepath</i>	string	[in] File path
-----------------	--------	----------------

Return values:

<i>system_info</i>	structure	[out] The structure of system information.
<i>.version</i>	double	Data version
<i>.revision</i>	double	Data revision
<i>.system_id</i>	double	System ID
<i>.system_name</i>	string	System name
<i>.model_name</i>	string	Model name

getRHdrChannel

This function retrieves information about channel.

```
channel_info = getRHdrChannel(filepath);
```

Arguments:

filepath string [in] File path

Return values:

channel_info structure [out] The structure of channel information.
.channel_count double The number of whole channels.
.channel structure array The detail information of channels. ('index 1' corresponds to 'channel 0')
.type double Channel type as follow table:

NullChannel	= 0;
MagnetoMeter	= 1;
AxialGradioMeter	= 2;
PlanarGradioMeter	= 3;
ReferenceMagnetoMeter	= 257;
ReferenceAxialGradioMeter	= 258;
ReferencePlanarGradioMeter	= 259;
TriggerChannel	= -1;
EegChannel	= -2;
EcgChannel	= -3;
EtcChannel	= -4;

.data structure The geometrical information of a channel.
 These fields is based on MEG device coordinate system.
 These fields of each channel type are as follows:
 See Figure.1 and Figure.2.

If channel type is AxialGradioMeter or ReferenceAxialGradioMeter (see Figure.3),

.x	double	x coordinate of inner coil position [meter]
.y	double	y coordinate of inner coil position [meter]
.z	double	z coordinate of inner coil position [meter]
.zdir	double	Sensor orientation from z-axis [degree]
.xdir	double	Sensor orientation from x-axis [degree]
.baseline	double	Baseline length [meter]
.size	double	Inner coil size [meter]
.name	string	Abbreviation name

If channel type is PlanarGradioMeter or ReferencePlanarGradioMeter (see Figure 4),

.x	double	x coordinate of inner coil position [meter]
.y	double	y coordinate of inner coil position [meter]
.z	double	z coordinate of inner coil position [meter]
.zdir1	double	Sensor orientation from z-axis [degree]
.xdir1	double	Sensor orientation from x-axis [degree]
.zdir2	double	Baseline orientation from z-axis [degree]
.xdir2	double	Baseline orientation from x-axis [degree]
.baseline	double	Baseline length [meter]
.size	double	Inner coil size [meter]

If channel type is MagnetoMeter or ReferenceMagnetoMeter,

.x	double	x coordinate of coil position [meter]
.y	double	y coordinate of coil position [meter]
.z	double	z coordinate of coil position [meter]
.zdir	double	Sensor orientation from z-axis [degree]
.xdir	double	Sensor orientation from x-axis [degree]
.size	double	Inner coil size [meter]
.name	string	Abbreviation name

If channel type is EegChannel or EcgChannel,

.type	double	Type (0: Analog input, 1: Nihon-Kohden EEG)
.id	double	ID
.name	string	Abbreviation name
.gain	double	Gain (This field exists if type=0)

If channel type is TriggerChannel or EtcChannel,

.type	double	Type
.id	double	ID
.name	string	Abbreviation name

If channel type is NullChannel, there is no field.

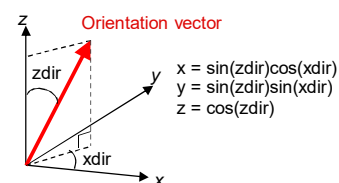


Figure.1 Orientation vector

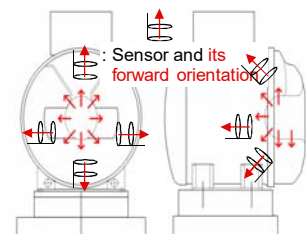


Figure.2 Sensor orientation in the dewar

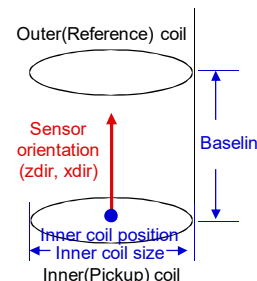


Figure.3 AxialGradioMeter parameter

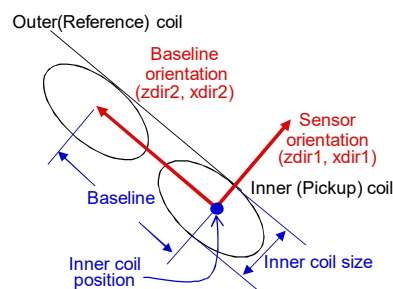


Figure.4 PlanarGradioMeter parameter

getRHdrAcqCond

This function retrieves information about data acquisition condition.

```
acq_cond = getRHdrAcqCond(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>acq_cond</i>	structure	[out] The structure of information about data acquisition condition.
<i>.acq_type</i>	double	Acquisition type
		AcqTypeContinuousRaw = 1;
		AcqTypeEvokedAve = 2;
		AcqTypeEvokedRaw = 3;
If acquisition type is AcqTypeContinuousRaw,		
<i>.sample_rate</i>	double	Sampling rate [Hz]
<i>.sample_count</i>	double	The number of samples which were actually acquired [sample]
<i>.specified_sample_count</i>	double	The number of samples which were specified before starting acquisition [sample]
If acquisition type is AcqTypeEvokedAve or AcqTypeEvokedRaw,		
<i>.sample_rate</i>	double	Sampling rate [Hz]
<i>.frame_length</i>	double	Frame length (The number of samples per one trial) [sample]
<i>.pretrigger_length</i>	double	Pretrigger length (The number of samples before trigger per one trial) [sample]
<i>.average_count</i>	double	The number of trials(frames) which were actually acquired [trial]
<i>.specified_average_count</i>	double	The number of trials(frames) which were specified before starting acquisition [trials]
<i>.multi_trigger</i>	structure	The structure of multi trigger information.
<i>.enable</i>	boolean	Is multi trigger mode ? (true : multi trigger mode)
<i>.count</i>	double	Number of multi triggers
<i>.list</i>	structure array	List of multi triggers (If not multi trigger mode, this structure array is set to empty.)
<i>.enable</i>	boolean	Is current multi trigger set to enable ? (true : enable)
<i>.code</i>	double	Event code (1 origin)
<i>.name</i>	string	Event name
<i>.average_count</i>	double	The number of trials(frames) which were actually acquired [trial]
<i>.specified_average_count</i>	double	The number of trials(frames) which were specified before starting acquisition [trials]

getRHdrEvent

This function retrieves information about trigger event.

```
event = getRHdrEvent(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>event</i>	structure array	[out] The structure array of trigger event corresponding to each trial.
<i>.sample_no</i>	double	Sample number of current event (0 origin)
<i>.code</i>	double	Event code (1 origin)
<i>.name</i>	string	Event name

getRHdrCoregist

This function retrieves information about coregistration.

```
coregist = getRHdrCoregist(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>coregist</i>	structure	[out] The structure of information about coregistration.
<i>.done</i>	boolean	Is coregistration done? (true : done)
<i>.mri_type</i>	double	MRI type NoMriFile = 0; NormalMriFile = 1; VirtualMriFile = 2;
<i>.mri_file</i>	string	File path of MRI file (*.mri)
<i>.hpi_file</i>	string	File path of HPI(Head Position Indicator) file (*.mrk)
<i>.meg2mri</i>	matrix(double)	4 x 4 matrix which transforms MEG device coordinate to MRI coordinate [meter] usage: $[xmri, ymri, zmri, 1]' = coregist.meg2mri * [xmeg, ymeg, zmeg, 1]'$
<i>.mri2meg</i>	matrix(double)	4 x 4 matrix which transforms MRI coordinate to MEG device coordinate [meter] usage: $[xmeg, ymeg, zmeg, 1]' = coregist.meg2mri * [xmri, ymri, zmri, 1]'$
<i>.hpi</i>	structure array	The structure array of HPI(Head Position Indicator)
<i>.meg_pos</i>	matrix(double)	HPI position $[x, y, z]$ on MEG device coordinate [meter]
<i>.mri_pos</i>	matrix(double)	HPI position $[x, y, z]$ on MRI coordinate [meter] (Before coregistration, this field is set to $[0,0,0]$)
<i>.label</i>	string	HPI label as follows: 'LPA' : Left PreAuricular 'RPA' : Right PreAuricular 'CPF' : Center PreFrontal 'LPF' : Left PreFrontal 'RPF' : Right PreFrontal
<i>.model</i>	structure	The structure of conductor model.
<i>.type</i>	double	Conductor model type UNKNOWN_MODEL = -1; NO_MODEL = 0; SPHERICAL_MODEL = 1; LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

<i>.cx</i>	double	x coordinate of spherical center position on MRI coordinate [meter]
<i>.cy</i>	double	y coordinate of spherical center position on MRI coordinate [meter]
<i>.cz</i>	double	z coordinate of spherical center position on MRI coordinate [meter]
<i>.radius</i>	double	radius of spherical conductor on MRI coordinate [meter]

If Conductor model type is LAYERED_MODEL,

<i>.ax</i>	double	Coefficient 'ax' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.ay</i>	double	Coefficient 'ay' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.az</i>	double	Coefficient 'az' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.c</i>	double	Coefficient 'c' of planar equation $'ax * x + ay * y + az * z = c'$

getRHdrDigitize

This function retrieves information of the digitization.

```
digitize = getRHdrDigitize(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>digitize</i>	structure	[out] The structure of information and points about digitization.
<i>.info</i>	structure	The structure of information about digitization.
<i>.digitizer_file</i>	string	File path of digitizer file
<i>.done</i>	boolean	Is matching done? (true : done)
<i>.meg2digitizer</i>	matrix(double)	4 x 4 matrix which transforms MEG coordinate to Digitizer coordinate.
<i>.digitizer2meg</i>	matrix(double)	4 x 4 matrix which transforms Digitizer coordinate to MEG coordinate.
<i>.point</i>	structure array	The structure of point data about digitization.
<i>.name</i>	string	Point name
<i>.x</i>	double	x-coordinate on digitizer coordinate [meter]
<i>.y</i>	double	y-coordinate on digitizer coordinate [meter]
<i>.z</i>	double	z-coordinate on digitizer coordinate [meter]

getRHdrSubject

This function retrieves information of the subject.

```
subject = getRHdrSubject(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>subject</i>		[out] The structure of subject information.
<i>.id</i>	string	ID
<i>.name</i>	string	Name
<i>.birthday</i>	string	Birthday
<i>.sex</i>	string	Sex
<i>.handed</i>	string	Handed

getRHdrAnnotation

This function retrieves information about annotation.

```
annotation = getRHdrAnnotation(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>annotation</i>	structure array	[out] The structure array of annotation information.
<i>.sample_no</i>	double	Sample number of annotation
<i>.label</i>	double	Label of annotation
<i>.comment</i>	string	Comment of annotation
<i>.annotationCategory</i>	uchar	Category of annotation

getRHdrSource

This function retrieves information of the sources.

```
source = getRHdrSource(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>source</i>	structure array	[out] The structure array of analyzed source information. Note : Sources are arranged in order of estimated time.
<i>.type</i>	double	Type of source DipoleModel = 1; DistributedSourceModel = 2;
<i>.time</i>	double	Analyzed Time [second] from 1970.1.1
<i>.sample_no</i>	double	Time sample index of source
<i>.channel_list</i>	row vector(double)	Channel number (0 origin) list which used to estimate
<i>.model</i>	structure	The structure of conductor model.
<i>.type</i>	double	Conductor model type UNKNOWN_MODEL = -1; NO_MODEL = 0; SPHERICAL_MODEL = 1; LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

<i>.cx</i>	double	x coordinate of spherical center position on MEG coordinate [meter]
<i>.cy</i>	double	y coordinate of spherical center position on MEG coordinate [meter]
<i>.cz</i>	double	z coordinate of spherical center position on MEG coordinate [meter]
<i>.radius</i>	double	radius of spherical conductor on MEG coordinate [meter]

If Conductor model type is LAYERED_MODEL,

<i>.ax</i>	double	Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
<i>.ay</i>	double	Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
<i>.az</i>	double	Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
<i>.c</i>	double	Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

.algorithm	structure	The structure of conductor algorithm.
.magnetic_field_calc	double	Algorithm of magnetic field calculation BiotSavartLaw = 1; SarvasLaw = 2; MagneticDipoleLaw = 3;
.variable_restraint	double	Algorithm of variable restraint NoRestraint = 0; PositionRestraint = 1; DirectionRestraint = 2; IntensityRestraint = 3;
.optimization	double	Algorithm of optimization GradientAlgorithm = 1; LeadFieldReconstructionAlgorithm = 2; ManualSetAlgorithm = 3; UserAlgorithm = 4;
.filter	structure	The structure of spectral filter setting.
.hpf, .lpf	structure	The structure of high-pass / low-pass filter setting.
.enable	boolean	Does this filter enable?
.cutoff_frequency	double	Cutoff frequency [Hz]
.window_type	double	Window type NoWindow = 0; HanningWindow = 1; HammingWindow = 2;
.width	double	Filter width
.bpf, .bef	structure	The structure of band-pass / band-eliminate filter setting.
.enable	boolean	Does this filter enable?
.low_frequency	double	Low frequency [Hz]
.high_frequency	double	High frequency [Hz]
.window_type	double	Window type
.width	double	Filter width
.moveave	structure	The structure of moving average setting.
.enable	boolean	Does this filter enable?
.width	double	Filter width
.baseadj	structure	The structure of baseline adjustment setting.
.enable	boolean	Does this filter enable?
.type	double	Type of baseline adjustment PretriggerBaselineAdjust = 0; PosttriggerBaselineAdjust = 1; AllRangeBaselineAdjust = 2; ExplicitBaselineAdjust = 3;
.start_time	double	Start time [millisecond]
.end_time	double	End time [millisecond]
.gof	double	Goodness-of-fit (GOF)
.correlation	double	Correlation Coefficiency
.label	double	Label
.comment	string	Comment
.total_intensity	double	Total intensity of sources
.dipole_count	double	Number of dipole sources
.dipole_list	structure array	The structure array of dipole sources
.x	double	x coordinate of dipole position on MEG coordinate [meter]
.y	double	y coordinate of dipole position on MEG coordinate [meter]
.z	double	z coordinate of dipole position on MEG coordinate [meter]
.zdir	double	Dipole orientation from z-axis [degree]
.xdir	double	Dipole orientation from z-axis [degree]
.intensity	double	Dipole intensity (moment) [Ampere Meter]

getRMriHdr

This function retrieves header information of specified mri file (*.mri).

```
mri_header = getRMriHdr(
    filepath
);
```

Arguments:

filepath string [in] File path

Return values:

mri_header structure [out] The structure of mri header information.
.data_style double Data style (0 : DICOM, others : Polhemus)
.model structure The structure of conductor model.
.done boolean Is conductor model defined ? (true : defined)
.type double Conductor model type
 UNKNOWN_MODEL = -1;
 NO_MODEL = 0;
 SPHERICAL_MODEL = 1;
 LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

.cx double x coordinate of spherical center position on MRI coordinate [meter]
.cy double y coordinate of spherical center position on MRI coordinate [meter]
.cz double z coordinate of spherical center position on MRI coordinate [meter]
.radius double radius of spherical conductor on MRI coordinate [meter]

If Conductor model type is LAYERED_MODEL,

.ax double Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
.ay double Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
.az double Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
.c double Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

.hpi structure array The structure of point data about picked HPI.
.done boolean Is pick-up of a HPI point done ? (true : done)
.mri_pos double HPI position [x, y, z] on MRI coordinate [meter]
.label string HPI label as follows:
 'LPA' : Left PreAuricular
 'RPA' : Right PreAuricular
 'CPF' : Center PreFrontal
 'LPF' : Left PreFrontal
 'RPF' : Right PreFrontal

.image_parameter structure The structure of image parameters.
.intensity vector(double) 1 x 2 row vector, minimum and maximum of image values
.initial_color vector(double) 1 x 2 row vector, minimum and maximum of initial brightness
.color vector(double) 1 x 2 row vector, minimum and maximum of current brightness

.normalize structure The structure of normalized HEAD coordinate system (LPA(x-), RPA(x+), nasion(y+)). See Figure.5.
.done boolean Is HEAD coordinate system defined ? (true : defined)
.mri2normalize matrix(double) 4 x 4 matrix which transforms MRI coordinate to HEAD coordinate [meter]
 usage: [xhead, yhead, zhead, 1]' = *mri_header.normalize.mri2normalize* * [xmri, ymri, zmr., 1]'

.point structure array The structure of point data about HEAD fiducial points.
.done boolean Is pick-up of a HEAD fiducial point done ? (true : done)
.name string Name of HEAD fiducial points.
.x double x coordinate of a HEAD fiducial point on MRI coordinate [meter]
.y double y coordinate of a HEAD fiducial point on MRI coordinate [meter]
.z double z coordinate of a HEAD fiducial point on MRI coordinate [meter]

.besa_fiducial structure The structure of BESA fiducial information.
.point structure array The structure of point data about BESA fiducial points.
.done boolean Is pick-up of a BESA fiducial point done ? (true : done)
.x double x coordinate of a BESA fiducial point on MRI coordinate [meter]
.y double y coordinate of a BESA fiducial point on MRI coordinate [meter]
.z double z coordinate of a BESA fiducial point on MRI coordinate [meter]

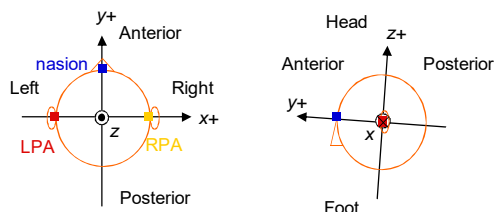


Figure.5 Normalized HEAD coordinate system

getRVersion

This function retrieves version of this toolbox.

version = getRVersion;

Arguments:

none

Return values:

<i>version</i>		[out] structure of toolbox version
<i>.version</i>	string	toolbox version : major.minor
<i>.major</i>	double	toolbox major version
<i>.minor</i>	double	toolbox minor version
<i>.revision</i>	double	toolbox revision version
<i>.build</i>	double	toolbox build version
<i>.date</i>	string	release date yyyy.mm.dd