|  |  |
| --- | --- |
| Title | MATLAB Structure Conventions |
| Document No. | SPE-003-001 |
| Revision | 1 |
| Approvals |  |

Table of Contents

[2 Document Scope 1](#_Toc448249985)

[3 Glossary of terms 1](#_Toc448249986)

[4 References 2](#_Toc448249987)

[5 Data access 2](#_Toc448249988)

[6 Sweep structure 2](#_Toc448249989)

[7 Process structure 5](#_Toc448249990)

[8 Other functions and examples 6](#_Toc448249991)

[9 List of tables 8](#_Toc448249992)

[10 Version History 8](#_Toc448249993)

# Document Scope

This document defines the MATLAB variables used to store BSXinsigth device [R1] data. It encompasses all versions of the product, including late prototypes, Gen1 and Gen2 devices.

# Glossary of terms

The following table lists the terms used in this document.

Table 1: Glossary of terms used in this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| IMU | Inertial measurement unit |
| LT | Lactate threshold |
| ID | Identity |
| PC | Personal computer |
| CSV | Comma separated values |
| MAC | Media access control |
| UTC | Universal time coordinated |
| LED | Light emitting diode |
| MPU | Micro processing unit |
| NaN | Not a number |

# References

The following is a list of other documents referenced within this document.

Table 2: List of other documents referenced within this document.

|  |  |  |
| --- | --- | --- |
| **Reference** | **Document No.** | **Title** |
| R1 | 101-0002 | BSX INSIGHT PRODUCT SPECIFICATION |
| R2 | SPE-001-001 | Real time muscle oxygenation and assessment register specifications |
| R3 | SPE-002-002 | Total hemoglobin parameter specifications |
| R4 |  | Optical\_Tissue\_Detection\_Definition\_02Sep2015.pdf |

# Introduction: Data access

When users perform *LT assessments* or *Daily Activities* using the BSXinsight device the *Mobile Application* automatically creates LT assessments in BSX’s cloud server and the device stores optical and IMU data on its FLASH memory. Once users dock their devices the resulting data file is automatically uploaded to the cloud server using the *PC Desktop Application*. In the cloud database files are associated either with an LT assessment or with an activity, each one with a unique ID, allowing BSX engineers to download the data files using their ID numbers as a reference. The files downloaded this way are in the “optical” or “binary” format, with an “.optical” or “.bin” extension. Note that in spite of the different extensions these files are of identical format.

Alternatively, BSX engineers can also access and download data files directly from a device using the *Science Application*, resulting in a .bin file downloaded to their desktop PCs. The Science Application also saves the same data file in the CSV format, which is less memory efficient but easier to read and parse into other applications, including MATLAB. The Science Application includes a dynamically linked library (*BSX.Insight.Lib.Algorithm.dll*) in its installation directory that is capable of converting .bin files into CSV files. Using this library though a simple executable (*BSX.Insight.DeviceRecordParserApp.exe*) MATLAB is thus capable of reading and parsing BSXinsight data files into any format (.bin, .optical or .csv) using the function *getSweep.m*, which takes the file path as its argument and returns a structure known as the *sweep structure*.

# Sweep structure

The function getSweep.m converts BSXinsight data files into a MATLAB structure with the fields listed in Table 3, which lists fields in the same order they are presented in MATLAB. Note that getSweep.m is backward compatible with earlier versions of the data file (known as flash *schema*). As such, some of the fields that are available today did not exist in previous schema (e.g., tHb, SmO2 and Alert fields did not exist in Gen1 of the BSXinsight device). Thus, it is advisable to use the MATLAB function *isfield(sweep, ‘fieldname’)* when working with legacy MATLAB files to prevent program crashes. Currently the latest schema number is 37.

Table 3: Fields included in sweep file structure.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Explanation** | **Data format** | **Default value** |
| flash\_schema | File saving scheme used in FLASH | unit16 | N/A |
| samp\_rate | Sampling rate of optical data | single | N/A |
| imu\_samp\_rate | Sampling rate of IMU data | single | N/A |
| date | UTC date of string | string ‘YYYY/MM/DD’ | ‘0001/01/01’ |
| UTCtime | Time of sweep in UTC format. | string ‘HH:MM:SS’ | ’00:00:00’ |
| device\_id | Device serial number. Usually the device MAC ID. | string (12 hex chars) | N/A |
| assessment | Assessment unique ID. | string (24 hex chars) | 'FFFFFFFFFFFFFFFFFFFFFFFF' |
| sport | Sport type of data file. Usually ‘run’, ‘bike’ or ‘calibration’ | string | N/A |
| FW\_version | Firmware version of device used to collect data | string | N/A |
| Is\_Daily\_Activity | Flag indicating whether the data file is a Daily Activity | logical | N/A |
| SportType | Device sport type. Usually ‘Run’, ‘Cycling’ or ‘MultiSport’ | string | N/A |
| ambient | Ambient light signal (counts) | single array Nx1 | N/A |
| count15 | Optical signals generated by 15mm LEDs (counts) | single array Nx4 | N/A |
| ccode15 | Current codes used to drive 15mm LEDs | single array Nx4 | N/A |
| count27 | Optical signals generated by 27mm LEDs (counts) | single array Nx4 | N/A |
| ccode27 | Current codes used to drive 27mm LEDs | single array Nx4 | N/A |
| Acc\_y | Accelerometer reading (g), y-axis | single array Mx1 | Empty array [] |
| Acc\_z | Accelerometer reading (g), z-axis | single array Mx1 | Empty array [] |
| Gyro\_x | Gyroscope reading (deg/s), x-axis | single array Mx1 | Empty array [] |
| Gyro\_y | Gyroscope reading (deg/s), y-axis | single array Mx1 | Empty array [] |
| Pace\_on\_dev | IMU pace information (m/s) | single array Mx1 | Empty array [] |
| cpuTemp | Temperature reported by MPU (Celsius) | single array Nx1 | N/A |
| battVolt | Battery voltage (V) | single array Nx1 | N/A |
| HR | Heart rate (bpm) reported by heart rate monitor attached to device via ANT+ protocol (if any) | single array Nx1 | NaN |
| PacePower | Pace (m/s, in running mode) or power (W, in cycling mode) reported by ANT+-connected device (if any) | single array Nx1 | NaN |
| cHhb\_15mm | Deoxyhemoglobin concentration (a.u.), 15mm geometry (used by LT assessment algorithm) | single array Nx1 | N/A |
| cHbO2\_15mm | Oxyhemoglobin concentration (a.u.), 15mm geometry (used by LT assessment algorithm) | single array Nx1 | N/A |
| cHhb\_27mm | Deoxyhemoglobin concentration (a.u.), 27mm geometry (used by LT assessment algorithm) | single array Nx1 | N/A |
| cHbO2\_27mm | Oxyhemoglobin concentration (a.u.), 27mm geometry (used by LT assessment algorithm) | single array Nx1 | N/A |
| SmO2 | Tissue oxygenation (%, projection method) | single array Nx1 | N/A |
| cHHb\_SmO2 | Deoxyhemoglobin concentration (a.u., projection method) | single array Nx1 | N/A |
| cHbO2\_SmO2 | Oxyhemoglobin concentration (a.u., projection method) | single array Nx1 | N/A |
| cH2O\_SmO2 | Water concentration (a.u., projection method) | single array Nx1 | N/A |
| Alert | Alert bits (see [R2]) | logical Nx1 | N/A |
| SDS\_speed | Speed, as reported by ANT+-connected device (m/s) | single array Nx1 | Empty array [] |
| SDS\_cadence | Speed, as reported by ANT+-connected device (steps/m) | single array Nx1 | Empty array [] |
| tHb | Total hemoglobin concentration (g/dL, see [R3]) | single array Nx1 | N/A |
| InstPower | Instantaneous power reported by ANT+-connected device (W) | single array Nx1 | NaN |
| BikeCadence | Bicycle cadence reported by ANT+-connected device (rpm) | single array Nx1 | NaN |

In Table 3 ‘*N’* refers to the number of samples recorded in a given optical time series, while ‘*M’* refers to the number of samples recorded in a given IMU time series. Note that, since the optical and IMU data may (and often do) have different sampling rates, *M* is usually different from *N*. Also, even in data files in which both have the same sampling rate IMU data collection does not start at the same time as the optical data and, hence, *M* and *N* are still likely to be different. Nevertheless, the time series fields *time* and *imu\_time* are synchronized, meaning that events that take place at the same time in both time series are guaranteed to have taken place within no more than a sampling period. Thus one can, for example, visualize IMU and optical events by using

plot(sweep.time, sweep.ambient)

hold on

plot(sweep.imu\_time, sweep.Acc\_y, ‘r’)

By convention *sweep* structures are saved using MATLAB’s *save* command and their file names finish with the suffix ‘*\_sweep.mat*’.

# Process structure

A *process* variable consists of a structure containing fields that are calculated (i.e., processed) using the original data collected by a device data file and stored in a *sweep* structure. Field definitions are listed in Table 4. Note that *process* fields have no default values. That is, if the *sweep* fields in which they are dependent do not exist, they are not created.

Table 4: Fields included in a process file structure.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Explanation** | **Data format** |
| OD15 | Optical density, 15mm | single Nx1 |
| OD27 | Optical density, 27mm | single Nx1 |
| Resid | Residual of fit using multivariate analysis | single Nx4 |
| mu\_eff | Effective absorption coefficient of each LED (1/mm) | single Nx4 |
| Pk | Solution of multivariate fit | single Nx3 |
| mu\_s | Reduced scattering coefficients (1/mm) | double 1x4 |
| mu\_a | Absorption coefficients for each LED (1/mm) | single Nx4 |
| pH2O | Water concentration (a.u). same as cH2O\_SmO2 in sweep structure | single Nx1 |
| HbF | Total hemoglobin coefficient (a.u.). Sum of 1st and 2nd columns of Pk. Used to calculate tHb. | single Nx1 |
| HbConc | Hb concentration (old method) | single Nx1 |
| tHb | Total hemoglobin (g/dL) | single Nx1 |
| pH2Oproj | Vector projection towards water spectrum after fit using Hb spectra (a.u.) | single Nx1 |
| pH2Ocos | Angle cosine towards water spectrum after fit using Hb spectra | single Nx1 |
| SmO2 | Tissue oxygenation (%) | single Nx1 |
| tissueTF | Boolean indicating whether tissue is detected | logical Nx1 |
| tissue | Length of tissue detection error vector (a.u.) | single Nx1 |
| HR | heart rate calculated from optical signals (bpm) | single Nx1 |
| Acc | acceleration magnitude (g) decimated to same sampling rate as optical data | single Nx1 |
| Gyro | angular velocity magnitude (deg/s) decimated to same sampling rate as optical data | single Nx1 |
| date | Date when sweep processing was executed | string |

Table 5 lists fields that are redundant and are likely to be excluded from future revisions of the *process* structure. For future compatibility use their substitutes when writing new code.

Table 5: List of fields likely to be excluded from future revisions of process structure.

|  |  |
| --- | --- |
| **Field name** | **Reason for future obsolescence** |
| pH2O | Already calculated by device as field cH2O\_SmO2 |
| SmO2 | Already calculated by device as field SmO2 |
| tHb | Already calculated by device as field tHb |
| tissueTF | Already calculated by device as bit 8 of field Alert |
| HbF | Simply given by sum of first two columns of Pk (or sum of cHHb\_SmO2 and cHbO2\_SmO2 in sweep structure). |

Note that a MATLAB *sweep* structure can be converted into a *process* structure simply using the function *sweep2process.m* with the *sweep* structure as the input argument and the *process* structure as the output. By convention *process* structures are saved using MATLAB’s *save* command and their file names finish with the suffix ‘*\_process.mat*’.

# Other functions and examples

Table 6 lists other relevant MATLAB functions and a short description of each one of them. Each function is fully documented and their documentation is accessible by using MATLAB’s *help* or *doc* commands.

Table 6: List of MATLAB functions used to access, process and display BSXinsight data files.

|  |  |  |
| --- | --- | --- |
| **Function name** | **Short description** | **Example** |
| getAssessment.m | Returns information about an assessment (or a list of assessments) from the server. | assessment = getAssessment( '568bdac6adac1820728b456d') |
| getActivity.m | Returns information about an activity (or a list of activities) from the server. | activity = getActivity( '56c7a0aeadac181a408b4568') |
| getSweep.m | Reads a BSXInsight sweep (or assessment) data file. Accepts either binary, .optical, CSV or \_sweep.mat files, or a server URL pointing at the assessment  optical data. Also accepts an assessment or activity structures, as output by a getAssessment or getActivity function calls. | sweep = getSweep(assessment) |
| sweep2process.m | Calculates process structure from a sweep file structure. Uses Default parameters. | process = sweep2process(sweep) |
| generateAssessmentPlot.m | Function used to generate plots summarizing assessment or activity results. | generateAssessmentPlot(sweep, process) |

The following is an example of how to access, process and display device data. An LT assessment was used in this example but it is also applicable to an activity simply replacing the *getAssessment* function with getActivity:

assessment = getAssessment('568bdac6adac1820728b456d') % reads assessment structure

sweep = getSweep(assessment) % reads sweep structure

sweep = getSweep(assessment) % calculates process structure

generateAssessmentPlot(sweep, process) % plots assessment data

The plot created by this example is shown in Figure 1.



Figure 1: Sample output plot generated using generateAssessmentPlot.m function. Window was resize and tHb plot was zoomed in for better visualization.

# List of tables

[Table 1: Glossary of terms used in this document. 1](#_Toc448250015)

[Table 2: List of other documents referenced within this document. 2](#_Toc448250016)

[Table 3: Fields included in sweep file structure. 3](#_Toc448250017)

[Table 4: Fields included in a process file structure. 5](#_Toc448250018)

[Table 5: List of fields likely to be excluded from future revisions of process structure. 6](#_Toc448250019)

[Table 6: List of MATLAB functions used to access, process and display BSXinsight data files. 6](#_Toc448250020)

# Version History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **ECO #** |
| First release. | 4/12/2016 |  |