MuLES: Open source EEG acquisition and streaming

Documentation version 1.0

Raymundo Cassani
Hubert Banville

January 2015

Contents

1 Licence

2 Overview of features

MuLES (MuSAE Lab EEG Server) is a piece of software developed in LabVIEW, that aims at simplifying the use of common commercial electroencephalography (EEG) devices. It allows easy EEG data acquisition, recording as well as interfacing with any programming language supporting

basic network socket programming. In this way, it is not necessary for the user to delve into the available SDKs, APIs and proprietary communication protocols of the different devices. Moreover, the provided common interface allows complete interchangeability between the devices, making it easier to create applications that work with different EEG headsets. Currently supported features include:

- Acquire raw EEG data from 5 different consumer EEG devices:
  1. InteraXon Muse
  2. Emotiv Epoc
  3. Neurosky Mindset
  4. Neuroelectrics Enobio
  5. OpenBCI V3
- Recording and saving of EEG signals and triggers in CSV and/or EDF* format
- Streaming of live EEG data using a common, simple socket interface (can be used over the a single computer, a local network or the internet)
- Streaming of pre-recorded CSV or EDF* files
- Allows the use of multiple instances for multi-user acquisition, recording and streaming
- Example clients are provided in MATLAB and Python

*The feature of writing and reading EEG data in EDF ensures the compatibility of the recordings with other software (such as EEGLAB) and allows to stream data from already existent EEG databases.

3 Download and installation

MuLES is distributed as an executable for Windows (32bit and 64 bit versions) that requires LabVIEW Run-Time Engine 2013 SP1. Additionaly, LabVIEW source files are provided and can be compiled using the LabVIEW Application Builder.

MuLes compiled (for Windows) and Examples of clients can be found on `https://github.com/MuSAELab/MuLES`.

3.1 Dependencies

The LabVIEW Run-Time Engine 2013 SP1 (32-bit) (available for Windows 8 32-bit/8 64-bit/7 64-bit/7 32-bit/Vista 64-bit/Vista 32-bit/XP (SP3) 32-bit/Server 2008 R2 64-bit/Server 2003 R2 32-bit) must be installed by executing the Run-Time installer as Administrator. Note that even with a 64-bit OS the 32-bit Run-Time is required.

The respective software for each supported device must be installed prior to using MuLES. Where applicable, the device must also be paired to the computer via Bluetooth. Below is a list of resources on where to find the necessary software:

1. InteraXon Muse: The Muse SDK, including the MuseIO software, must be installed. This is available on the developper's site (`https://sites.google.com/a/interaxon.ca/muse-developer-sit home`).

2. Emotiv Epoc: The Emotiv SDK with access to raw EEG signals (a specific license must thus be bought) is needed for using the device with MuLES. The SDK can be obtained at `https://emotiv.com/store/sdk/`.

3. Neurosky Mindset: Mindset drivers must be installed. Instructions manual: `http://developer.neurosky.com/docs/doku.php?id=mindset_instruction_manual`

4. Neuroelectrics Enobio: The NIC software must be installed. This is available at `http://www.neuroelectrics.com/support/download`.

5. OpenBCI V3: Follow the tutorial "Prepare your OpenBCI hardware" at http://docs.openbci.com/.

3.2 Extracting the executable

Unzip the provided file into the desired folder. MuLES can then be started by launching the **mules.exe** file. Make sure the required dependencies are installed as described above and the paths to the respective software are correctly set in the **config.ini** file (see Section 4.6).

4 Usage

Quick access to the features of MuLES is allowed through its graphical user interface (GUI), as shown in Figure 1. The GUI consists in four different sections, described below:

(a) Input device selection

(b) Recording and streaming controls

(c) Start / Stop acquisition

(d) Current server state information

4.1 Selecting a device or pre-recorded data

MuLES currently allows the selection of one of six EEG data sources:

1. InteraXon Muse
2. Emotiv Epoc
3. Neurosky Mindset
4. Neuroelectrics Enobio
5. OpenBCI V3
6. Pre-recorded data in a CSV or EDF file

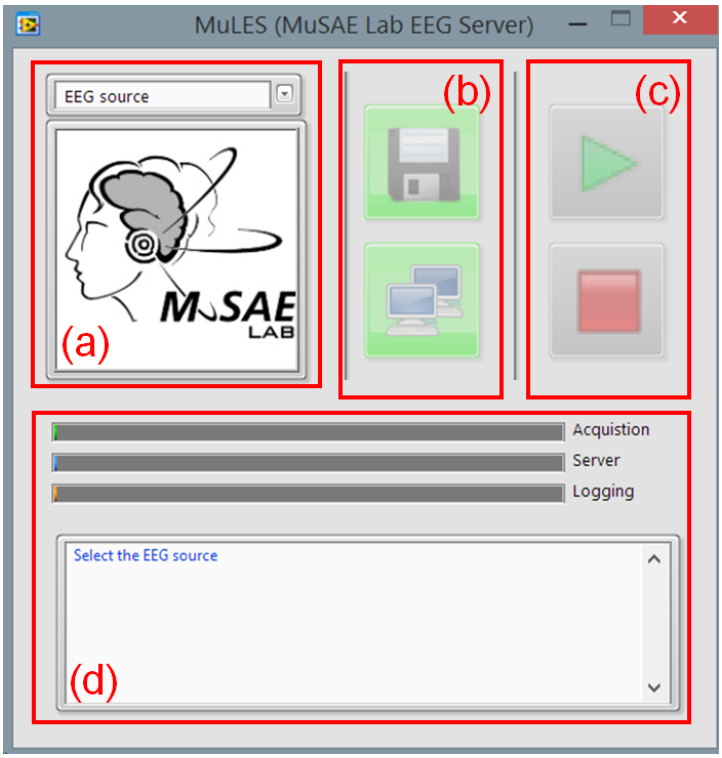

Figure 1: The graphical user interface and its sections: (a) input device selection, (b) recording and streaming controls, (c) start/stop acquisition and (d) current server state information

To stream data from a device, at least one of the supported devices (as well as the complete installation for that device, as mentioned in Section 3.1) is required. The path to the installation should be updated in the **config.ini** file (see Section 4.6).

First, power on the device and set it up following the manufacturer's instructions. Next, in the drop-down menu ((a) in Figure 1), select the desired device.

To stream from a pre-recorded data file, select *File* in the drop-down menu. In the prompt window, select the file from which you want to stream the data, and click OK.

After a successful selection (and/or connection), a message is displayed in the information text box in the GUI ((d) in Figure 1).

4.2 Recording and Streaming EEG data

4.2.1 Streaming the data through TCP/IP

To enable the live streaming of the data through the TCP/IP protocol, select the Network button in section (b) of Figure 1). Once the server is started, the data will available for the client application through the TCP/IP port specified in the **config.ini** file (default: 30000), following the format explained in Section 5.1.

4.2.2 Saving the data in a CSV or EDF format

To enable the saving of the streamed data in a CSV or EDF file, select the Save button in section (b) of Figure 1). The data is saved by default in the folder **eeg_files** of the main installation

path. To change the default folder, see instructions in Section 4.6.

Please note that at least one option (Recording or Streaming) must be chosen when EEG data comes from a device. When pre-recorded data file is source, only Recording OFF and Streaming ON is available.

4.3 Starting and stopping the acquisition

Once a device selection and an action (Recording or Streaming) have been made, the streaming and/or recording of the data can be started or stopped by clicking on the *Start* and *Stop* buttons (section (c) in Figure 1)

4.4 Current server state information

Once the streaming and/or recording has been started, information about the current state of the server is displayed in section 4 of Figure 1. The *Acquisition*, *Server* and *Logging* bars are animated whenever the connection to the device/file is ongoing, the data is streamed over TCP/IP, and the data is recorded in a file, respectively. Moreover, the information text box displays confirmation messages on the current state of the server.

4.5 Execution from the command line

MuLES can also be executed from the command line. A typical call would take the form:

```
1 server.exe --"DEVICEXX" PORT=30000 LOG=T TCP=T
```

The parameters **PORT**, **LOG** and **TCP** are optional, with default values 30000, T and T, respectively. **DEVICEXX**, on the other hand, is required, and can take the following values:

- The device label present in the **config.ini** file.
- The name of a CSV or EDF file located in the **eeg_files**/ subfolder.
- An absolute path for a CSV or EDF file.

4.6 Modifying the default parameters

The default parameters for each compatible device can be modified in the **config.ini** file, in the main folder containing the server executable. First, device-independent server parameters are found under the section starting with the **[GENERAL]** key, and are listed in Table 1. These parameters can be used to specify a different TCP/IP port for streaming the data, modify the default behaviour to enable or disable data recording or streaming, select a default saved file format, and select the interface language (English, French or Spanish).

Similarly, general device parameters are set for each device separately under the sections starting with [DEVICEXX] (Table 2). If using a device with the manufacturer's standard configuration, these parameters do not need to be modified except for **PATH**.

Finally, some additional fields are specific to the different hardware types and must be included in the EXTRA parameter listed in Table 2. These fields are shown in Table 3. For example, in the case of a Muse device, the EXTRA parameter would become:
"FS=220,#CH=4,DATA=ffffi,PRESET=14,BTNAME=Muse,OSCPORT=5000".

It is to be noted that new devices can be added as desired following the above conventions.

Parameter	Description	Value
PORT	TCP/IP server port	Default: 30000
LOG?	Enable the recording of the data	TRUE or FALSE
TCP?	Enable the streaming of the data	TRUE or FALSE
FILE-EXT	File format for saved data	"CSV" or "CSV;EDF"
LANGUAGE	Language of the interface	"eng.txt", "fra.txt" or "spa.txt"

Table 1: General server parameters found in **config.ini**.

Parameter	Description	Example
NAME	User-specified name for the device	"MUSE 1"
HARDWARE	Type of hardware of the device. Must be one of the following: "INTERAXON-MUSE", "NEUROSKY-MINDSET", "NEUROELECTRONICS-ENOBIO", "EMOTIV-EPOCH", "OPENBCI-V3" or "FILE"	"INTERAXON-MUSE"
IMAGE	Default image for the device	"1.jpg"
PATH	Path to the device drivers (for a list of necessary drivers, refer to Section 3.1)	"/C/Muse/muse-io.exe"
CHANNELS	List of channel names (as defined by the respective device drivers)	"A1,FP1,FP2,A2,STATUS"
EXTRA (for devices other than "FILE" only)	This parameter has three basic fields common to all hardware types other than "FILE": FS (sampling frequency in Hertz), $\#CH$ (number of channels provided by the device) and $DATA$ (format used to interpret the data stream generated by the server; the number of elements is always $\#CH + 1$, with f=float32 and i=int32)	"FS=220,#CH=4,DATA=ffffi"

Table 2: Device parameters found in **config.ini**.

Hardware	Field	Description	Example
INTERAXON-MUSE	PRESET	Data referencing presets as defined here.	PRESET=14
	BTNAME	Bluetooth name of the device (default: "Muse").	BTNAME=MUSEabc
	OSCPORT	OSC data is sent to localhost:5000 by default. In the case of multiple Muse devices being used at once on the same computer, you must assign a different port for each one.	OSCPORT=6000
NEUROSKY-MINDSET	COM	Port where the Mindset is connected (this information can be found in Device Manager in Windows).	COM=COM4
EMOTIV-EPOC	None		
NEUROELECTRONICS-ENOBIO	NICv	NIC software version	NICv=1.3
OPENBCI-V3	COM	Port where the OpenBCI is connected (this information can be found in Device Manager in Windows).	COM=COM14

Table 3: Device-specific fields to be included in the EXTRA parameter of each device.

Name	String	Decimal	Hexadecimal	Description
Flush	"F"	70d	46x	Flushes all accumulated data (only for devices, not for files).
Request	"R"	82d	52x	Request all samples obtained since the last Flush or Request command).
Stop	"P"	80d	50x	Stops server with code 66; equivalent to a client disconnection.
Kill	"K"	75d	4Bx	Stops server with code 6666, send exit to all loops and kills the exectuable.
Header	"H"	72d	48x	Requests header information.
Names	"N"	78d	4Ex	Request channel names.
Triggers	-	1d–64d	1x–40x	Sends one of 64 available triggers.

Table 4: TCP/IP commands that can be sent to the server.

5 Using the server from a client side

To receive the streamed data in a client application, one has to use basic socket programming. We provide example scripts for clients in MATLAB and Python, available by default in the zipped installation file under the **/code/client examples** subfolder.

5.1 Description of the communication protocol

In all streamed packages, the first four bytes correspond to an int32 that indicates the length of the following packet in bytes. The packet content is different depending on which command was sent to the server socket. The available 1-byte commands are listed in Table 4.

The following responses are expected when sending the Header, Names and Request commands:

- Header response
  - NAME
  - HARDWARE
  - FS
  - #CH
  - DATA
  - others
- Names response (channel labels are separated by commas)
  - CH1
  - CH2
  - ...
  - CHi

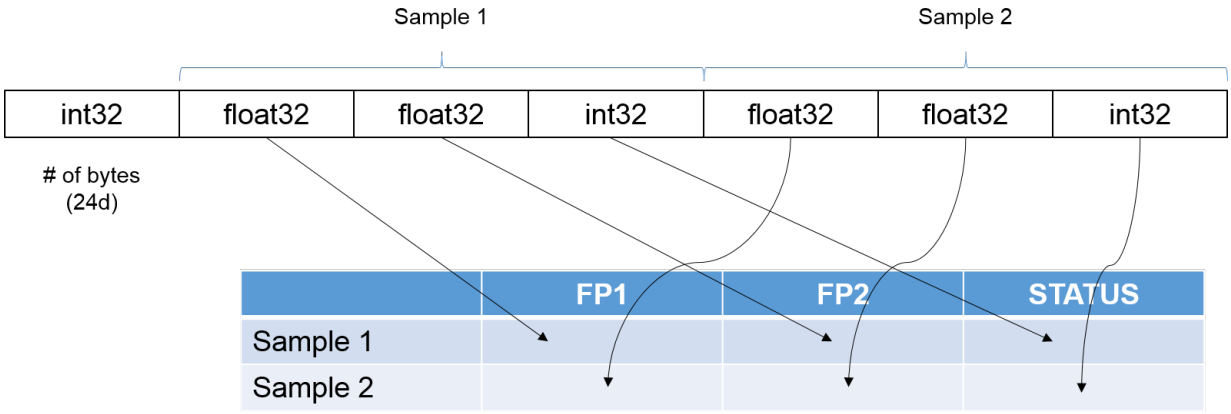

Figure 2: Example format of the Request response packet.

  - STATUS

- Request response (EEG data is provided, and need to be decoded using the parameters DATA provided by the Header command). Each sample in each channel is represented as a Single (float32). Each sample in STATUS is represented as a int32.

  - e.g.: Having a parameter **DATA=ffi** and channel names **Channels=FP1,FP2,STATUS**, a packet containing two samples is illustrated in Figure 2.