RSM Setup Guide

Jan 12, 2015

Bill Vinje

This guide is divided into three sections: Part I, initial setup; Part II, running a session; Part III, technical appendices including gamma calibration.

**Part I: Initial Setup.**

These are the steps that must be undertaken to situation RSM successfully on a new machine.

**Prerequisites for setup:**

> Intel-Based Mac.

> Internet access for downloads.

> Mac OS X 10.8 or 10.9.

> Matlab (at least R2012b).

> NI-6501 National Instruments USB digital-IO module.

**Step 1: Set up of hardware environment.**

> RSM assumes a 2nd display is in use.

If necessary, shut down computer and reconfigure monitors.

Note native frame rate and resolutions of 2nd monitor.

> Connect USB cable of NI-6501 cable to computer.

See NI-6501 wiring appendix for connection to rig.

**Step 2: Set up of software environment.**

> Check on Matlab java, should be: Java 1.6 or Java 1.7.

This can be accomplished from the Matlab command line by >> *version –java.*

> Upgrade X-11 environment: Xquartz 2.7.5

Download page: <http://xquartz.macosforge.org/trac/wiki/X112.7.5>

Install.

Reboot.

> Download NI-DAQmx Base 3.6 for Mac OS X

User account: [william.vinje@gmail.com](mailto:william.vinje@gmail.com)

Password: ejlab123

Download page: <http://www.ni.com/download/ni-daqmx-base-3.6/3431/en/>

Run installer with NI 6501 attached.

Reboot.

> Check that NI 6501 firmware is up to date. (Update if needed).

Checking and updating is done by running FWUpdate which is a Mac application that lives in Applications/NI- DAQmx Base/bin

> Download Xcode (if not already XCode 5 or later).

<https://developer.apple.com/xcode/downloads/>

> Download Xcode command line tools (note these are only used for compiling digital IO utilities).

Starting from the above website: Click on View Downloads in the Additional Tools section. When it asks you to sign in as an Apple Developer you can try your App store account info. If that fails you may have to create an Apple Developer account (which is free). When you are let into the <https://developer.apple.com/downloads/index.action> site then select “Command Line Tools (OS X Mountain Lion) for Xcode”

Download and mount the .dmg and run the installer .mpkg file.

**Step 3: RSM configuration:**

> Get hold of current RSM version from Dropbox.

Place the RSM directory wherever desired on stimulus machine. However, for file permissions reasons this should live in the home directory of whatever user runs the stimulus program.

> Bring necessary movies into appropriate locations:

A pre-run initialization movie is needed in RSM\_Utils/RAWMOV\_UTILS. It needs to be named: prerun\_dummy.rawMovie. For the RSM\_Batch.m demonstration a further movie is needed in RSM\_Movie\_Vault. The demonstration is designed for catcam\_forest.rawMovie

> Recompile RSM’s mgl routines if necessary.

The RSM bundle assumes a 64bit mac. If you are not using such a platform you will certainly have to recompile. Likewise, you may need to recompile if you experience difficulties. If necessary, re-compile mex files. (See re-compiling appendix).

> Change necessary RSM paths in Setup\_Rig.m file.

Go to RSM\_Utils/Setup\_Rig.m and change the path to the RSM home directory name on the new machine.

> If necessary, construct or update monitor object.

The monitor object describes all relevant monitor properties.

A default monitor object is provided with the bundle for setup purposes.

These are stored in {RSM}/RSM\_Monitors.

An appendix discussing the Gamma calibration process is included below.

**Step 4: Matlab configuration issues.**

> Set your Matlab path to include (with subdirectories) the RSM directory.

**Part II: Running an RSM session.**

**Step 1: Shut down search-light indexing utility in Mac OS X.**

From a terminal window run: "sudo mdutil -i off".

Note, this step only needs to occur once per reboot of computer. However, repetition does no harm.

**Step 2: Setup RSM Session on Matlab.**

> Shut down all non-Matlab applications. To obtain proper performance of RSM cannot tolerate any competition beyond the operating system.

> Start Matlab.

> Set Matlab working directory to RSM directory.

Things are now ready to run *RSM\_Batch* from the Matlab command line to see the stimulus demos that have been packaged for study.

Alternatively you can manually start a session with the following commands:

**>>** *Start\_RSM;*

**>>** *mglSetGammaTable( RSM\_GLOBAL.monitor.red\_table, RSM\_GLOBAL.monitor.green\_table, RSM\_GLOBAL.monitor.blue\_table );*

Note the second command. This is to ensure that the gamma table is set to our specifications. In current testing work there have been problems putting this command into the *Start\_RSM* script. Future testing work may determine whether this is explicit second step can be avoided in some way.

**Part III: Technical Appendices.**

**NI-6501 attachment wiring appendix:**

The wiring breakout consists of two lines. Digital-out and digital-in. We follow the same convention used by Justin Gardner in mgl. Digital-output is accomplished by the ability to set Port 1 of the NI-6501. Digital-input is accomplished by reading Port 2 of the NI-6501 (see http://gru.brain.riken.jp/doku.php/mgl/functionReferenceDigIO).

This translates into the following pinouts:

Digital-out, high (red): connection pin 27 on NI-6501.

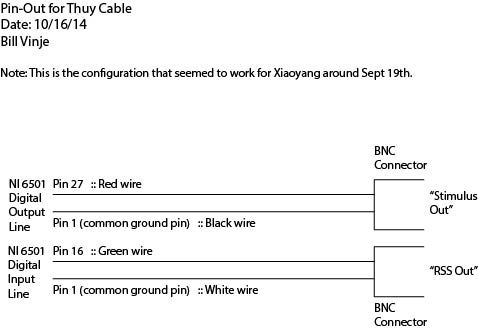
Digital-out, low (black): connection pin 1 on NI-6501.

The digital output connects with the “Stimulus out” line of the rig.

Digital-in, high (white): connection pin 16 on NI-6501.

Digital-in, low (green): connection pin 1 on NI-6501 (same as digital out).

The digital input connects with the “RSS Out” line of the rig.



**Machine / Software Environment Testing appendix:**

So far three machines have been by BV for RSM.

Development machine:

OS X 10.8.2.

Matlab version 8.0.0.783 (R2012b).

Java: 1.6

Extensive testing occurred on this machine. This includes tests of timing performance conducted with (i) photometer and oscilloscope and (ii) tests conducted in with physiology data collected using RSM as the stimulus computer in an active recording rig.

First successful install on 2nd machine:

OS X 10.8.5.

Matlab 2013B.

Java: 1.7

Currently, only minimal checks have been conducted. The *RSM\_Batch.m* file can successfully display all stimulus classes.

It is worth note that to accomplish the second install required updating the version of the mgl code that deals with Core GL. As of 04/25/14 the code used has features that Mac has depreciated as of OS X 10.8. This means that future OS updates might very well break something. If problems with graphics performance occur after OS updating (especially a black screen on the 2nd monitor) then investigating the relevant mgl code would be a good starting point.

Successful install on 3rd machine:

OS X 10.9.3.

Matlab 2013B

Java: 1.7

This is the installation on my laptop. It has not been tested with a second monitor yet.

So far RSM functions seem to operate properly.

**Recompiling Appendix:**

Currently, Justin’s recompile instructions prove sufficient.

To recompile the basic mgl executables (everything except digio): Navigate the matlab working directory to the RSM\_MGL/mgllib directory. Run: “ mglMake(1); ”.

To recompile the digio executables: Navigate to RSM\_MGL/utils/ReadDigPort. Run: “ mglMake(‘digio’); ”

Note: you must be careful that you do not accidentally use the unmodified version of Justin’s mexopt scripts (i.e mexopt script from an unmodified download from Justin). The mexopt scripts provided with the RSM bundle have been modified to not call on eyelink libraries that we don’t have (and don’t want to care about).

**Monitor Setup and Gamma Calibration Appendix**:

(With documentation help from Xiaoyang)

The goal is to generate a suitable “Monitor\_Obj” which is a matlab data structure of the object type. RSM uses this structure to control things like the gamma tables.

Here is an example of the properties embodied in a monitor object:

Monitor\_Obj with properties:

monitor\_name: 'RigA\_CRT1'

num\_lut\_bits: 8

gamma\_model: @BrainardModel

optometer\_raw\_data: [153x4 double]

red\_table: [1x256 double]

green\_table: [1x256 double]

blue\_table: [1x256 double]

red\_params: [2.5200 32 3.2181]

green\_params: [2.5200 32 3.1320]

blue\_params: [2.5200 32 3.1645]

gamma\_test\_date: '09-Jan-2014'

width: 640

physical\_width: 28.5000

height: 480

physical\_height: 21.5000

screen\_refresh\_freq: 119.5175

cen\_width: 320

cen\_height: 240

default\_mondesc\_path: './RSM\_Monitors/'

mon\_num: 2

backgrndcolor: [0.5000 0.5000 0.5000]

Step 1: Set up basic information

> Set your matlab path to include the RSM home directory with subdirectories.

> Navigate your matlab working directory to the RSM\_Monitors directory.

> At the matlab command prompt: "new\_display = Configure\_Display". You should see the following menu:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* RSM Monitor Configuration Utility. \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Options:

[ 1] Create new monitor object.

[ 2] Load existing monitor object.

[ 3] Edit existing monitor object.

[ 4] Collect data for new gamma profile.

[ 5] Visualize gamma profile. (doesn’t yet work)

[ 6] Remake gamma tables using best-fit gamma parameters.

[ 7] Remake gamma tables using values interpolated from data.

[ 8] Test existing gamma profile linearity.

[ 9] Determine frame refresh frequency.

[10] Save new monitor object.

[11] Exit.

\* Choose 1 to initialize. You’ll be asked to fill in some basic information about the display including:

\* Human name of monitor (remember to use single quotation mark)

\* Width of stimulus display screen [pixels]

\* Width of stimulus display screen [pixels]

\* Physical width of stimulus display screen [cm]

\* Height of stimulus display screen [pixels]

\* Physical height of stimulus display screen [cm]

\* Vertical refresh frequency of stimulus display screen [Hz] (round numbers are fine)

\* Number of bits in LUT [bits]:

\* If you want to change any of these information, choose 3 in the main menu and choose whatever you want to change in the submenu. Also you can add the following information:

\* Default monitor description path. (should be answered with './RSM\_Monitors/‘)

\* Monitor location number.

\* Monitor default background color.

\* Save the object by choosing 10 in the main menu.

\* Choose 9 to empirically adjust the frame rate to high precision. Here the refresh frequency will be calculated in real-time. You will be asked to input the data collection time.

\* Save again.

Step 2: Gamma calibration

\* Choose 4 to collect data for gamma profile

\* You will be prompted with questions specific for a “Graseby 350 Linear/Log Optometer”. If this is the optometer model you are using, please check the setting as those questions instruct. If not, just choose 1 for all these questions.

\* Enter the power readings of optometer as data collection starts. At the end of each round, you can choose to collect another repetition or stop.

\* Once you have collected data the program will generate default gamma tables via matlab’s interpolation functions. The interpolated gamma tables can be changed to values based on a best fit of a classic gamma function model (Brainards). This is done by using option “6” in main menu (and can be set back to the interpolated value by option “7”).

\* Save again.

\* Choose 8 to test linearity of gamma calibrations. This part is mostly same with 4.

\* Save again.

\* Choose 11 to exit.