Startle Reflex Software

Developed by Daniel Stolzberg (2011)

Table of Contents

Introduction

The Startle Reflex software is specifically designed to run the startle reflex paradigm and its derivatives, such as pre-pulse inhibition (PPI) of the acoustic startle reflex. The software provides graphical user interfaces (GUI) built and run in the MathWorks MATLAB (www.mathworks.com) environment. The MATLAB environment was selected because of its powerful data handling, analysis toolboxes, and graphical presentation features. Furthermore, since MATLAB is a "interpreted language," future feature development can be seamlessly implemented. The software interfaces with Tucker-Davis Technologies (www.tdt.com) hardware using proprietary ActiveX software interface. TDT hardware handles all stimulus generation and signal acquisition. While the primary use of the Startle Reflex software is for acoustic paradigms, other startle paradigms using air-puff, lights, floor shock or some combination of stimulus modalities can be implemented using the same software (only modality specific stimulus generators may be needed).

The intent of the software is to facilitate the creation and implementation of startle reflex experiments without any software development by the end-user. Despite this simplicity directive, the software provides a powerful schema to create relatively complex, multiparametric startle reflex paradigms in a simple GUI. Furthermore, the Startle Reflex software was designed to be scalable from the same GUI; i.e. the number of subject boxes run simultaneously is limited only by the number of analog/digital channels available on the TDT hardware.

Getting the Gist of It

The Startle Reflex software setup is split into three main parts:

1) ***Experiment file***: Created using the Experiment Generator GUI (`SRExpGen`). This GUI is primarily used to define the basic parameters needed to connect to the TDT hardware, which RPvds files will be used, as well as some other information about the experiment design. An experiment file is created for different types of protocols. For example, one experiment file is created for noise-burst PPI and a different experiment file is created for gap-PPI since these require different stimulus parameters.
2) ***Schedule file***: Created using the Schedule Generator GUI (`SRSchedGen`). A single experiment can have many different definitions for its parameters. This GUI creates a 'schedule' which defines these parameters with a particular experiment in mind. Many different schedules can be created for a single experiment.
3) ***Control Panel Setup***: The control panel (`SRControlPanel`) is used to locate an experiment file and a schedule file, as well as define which startle reflex boxes are

being run and which subject is in each box. This setup can be saved and recalled for subsequent experiments to save time and reduce mistakes.

Once experiment and schedule files are generated and the control panel setup is configured, running a session is as simple as launching the control panel (`SRControlPanel`), loading a setup file and running the session.

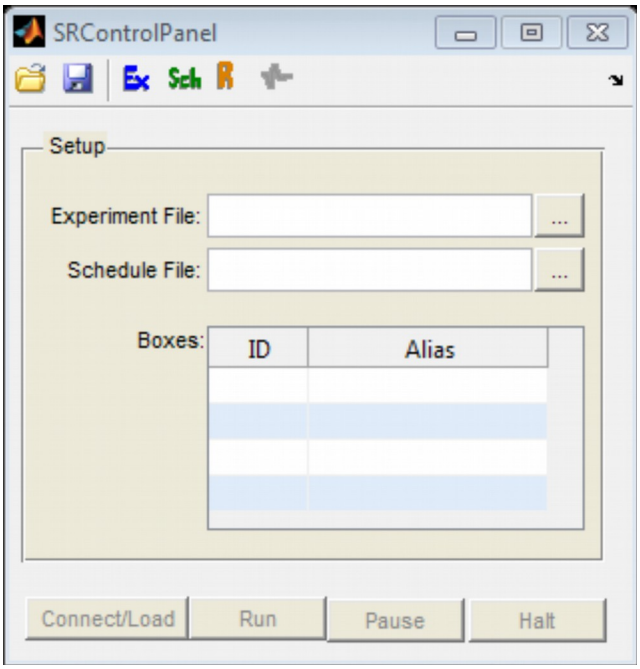


Fig 1. Startle Reflex Control Panel

Control Panel (`SRControlPanel`)

The Startle Reflex software is based around a single control panel (Fig 1). The primary purpose of this GUI is to set up and run a startle reflex session. The toolbar of the control panel provides access to all of the other GUIs used to create experiments and schedules and to access schedule results. Each of these GUIs are described in detail in later sections. Three steps are required before a session can be run.

1) The “Experiment File:” field locates a file on disk generated using the Experiment Generator GUI . These files have the extension .SRES.
2) The “Schedule File:” field locates a file on disk generated using the Schedule Generator GUI. These files have the extension .SRSF.
3) The “Boxes:” table is used to define which subject boxes are to be used. The “ID” column accepts integer values (1,2,3, etc.) which indicates which subject boxes (really which analog/digital channels) are being acquired for the current session. The “Alias” column is a name or some textual indication of which subject is located in the corresponding box ID. The Alias defaults to “Box_#” where ‘#’ is the corresponding ID.

Once these three fields are defined (Fig 2), then the “Connect/Load” button will be enabled. Clicking the “Connect/Load” button will establish a connection with the TDT hardware. If a connection is successful, then the “Run” button will be enabled. If an error occurs, make sure the hardware is turned on and it is visible in TDT’s zBUSmon program. If the hardware modules are visible in zBUSmon, then check the experiment file Experiment Generator.

Clicking the disk icon on the toolbar saves the selected experiment and schedule files as well as the box information into a single settings file. In subsequent experiments, clicking the folder icon on the toolbar will prompt to locate previously saved setup files. Note that the setup files contain file locations and not the files themselves. If an experiment or schedule file is moved from its original location then an error will be generated. To resolve this, simply locate the new locations of the files on disk and save a new setup file.

Click “Run” to start a session. A new window will be launched to plot the piezoelectric signal from each box for each trial.

Please note that clicking the “Pause” button does not immediately pause the session. A message dialog will appear asking you to wait until the current trial is triggered. Once the current trial is triggered, a “PAUSED” dialog will appear.

Clicking the “Halt” button will stop the session. A dialog will appear asking if you would like to save the session data acquired so far.

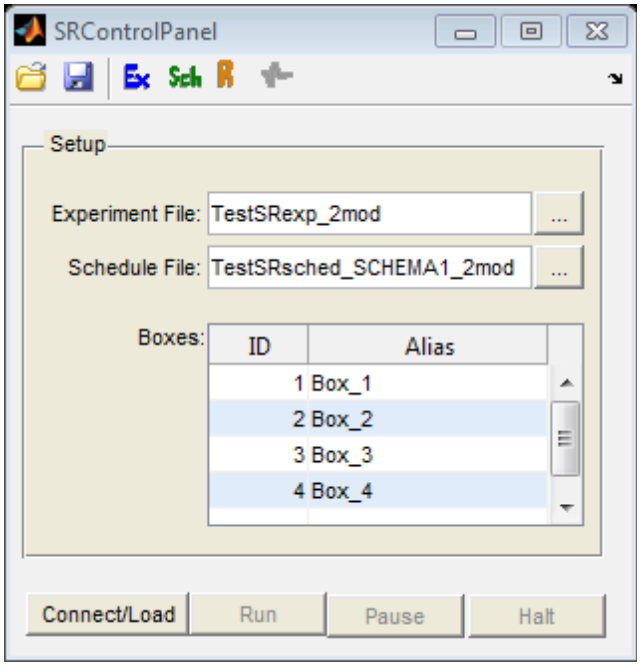


Fig 2. Startle Reflex Control Panel with fields filled out and ready to connect to hardware.

If the session runs through all trials (defined in Schedule Generator GUI), then the control panel will ‘halt’ itself and prompt the experimenter to save the completed session data.

Experiment Generator GUI (SRExpGen)

The Startle Reflex software takes advantage of Tucker-Davis Technologies' real-time processing modules and the RPvds design environment. The Exxperiment Generator GUI (Fig 3) gathers information regarding which TDT hardware modules are to be used, the function of each module, and the RPvds files associated with each module. Standard RPvds files are available for running nearly any form of startle reflex or PPI paradigm.

Hardware Setup

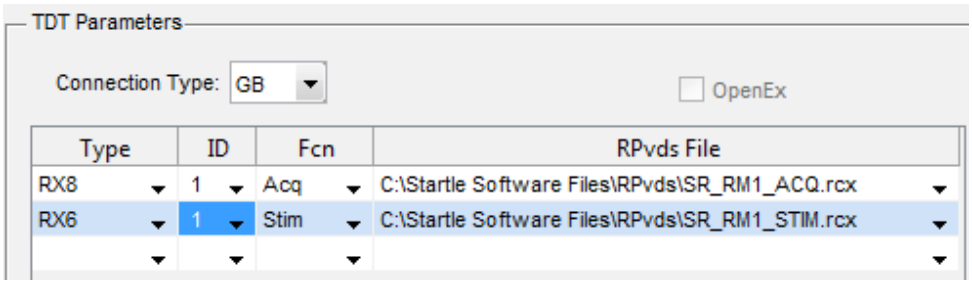


Fig 4. TDT Hardware/RPvds parameters

The TDT modules setup (Fig 4) is used to define how the TDT hardware is connected to the computer (USB or GB for Gigabit or Optibit), which hardware modules are being used for generating stimulus and which are being used to acquire the piezoelectric signal, i.e. the startle response.

The module type is specified by selecting the appropriate module in the 'Type' column. In the example in figure 4, there two modules selected, an RX8 for acquisition and RX6 for stimulus generation. Both have ID set to '1' because they are the only RX8 and RX6 modules being used with this setup. If more than one module of the same type is being used (ex: two RP2 modules), then the ID associated with each module must be specified.

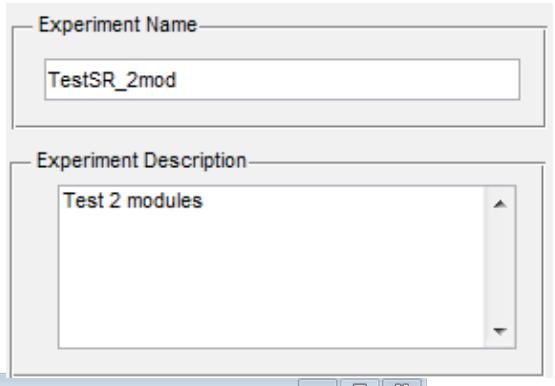


Fig 5. Experiment info

The RPvds file

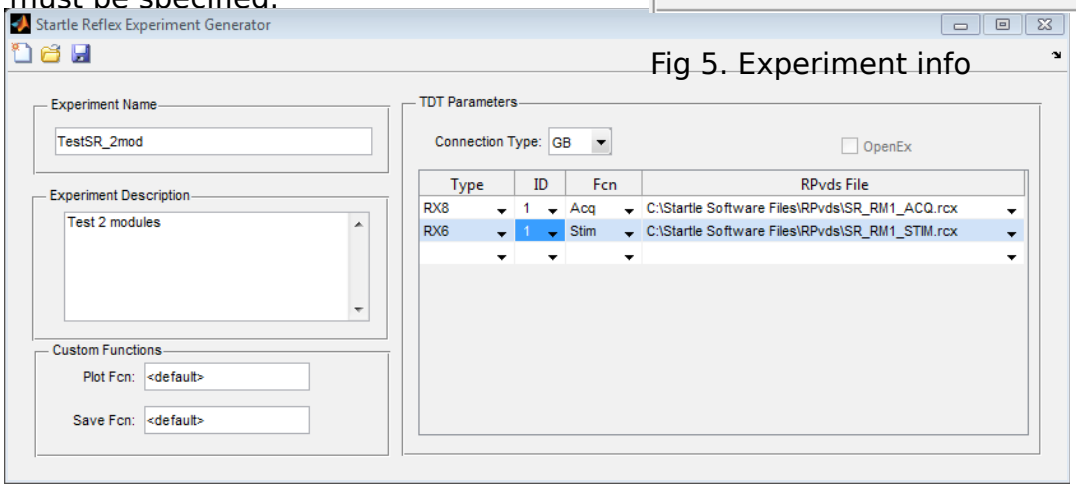


Fig 3. Experiment Generator GUI

column is used to locate and identify which pregenerated RPvds file is associated with each module. Each module must have an RPvds file associated with it. Click this field and select "< LOCATE >" to browse to the RPvds file location on disk.

Experiment Info

Other information may be entered to name (required field) and describe (optional field) the project (Fig 5).

Custom Functions

Customized functions can be specified for further control over the Startle Reflex software during run time. Although this is currently implemented, specifics are not discussed in this manual. Contact Dan for more info.

Schedule GUI (SRSchedGen)

The startle reflex paradigm has a tremendous number of parameters which can be modified to test different hypotheses regarding neurological function and dysfunction. This software package uses 'schedules' to facility the creation of nearly any startle reflex paradigm. Furthermore, the Schedule Generator GUI (Fig 6) generates a basic schematic of the protocol design.

The Schedule Generator GUI is composed of basically three parts; however, only two need attention for most protocol designs.

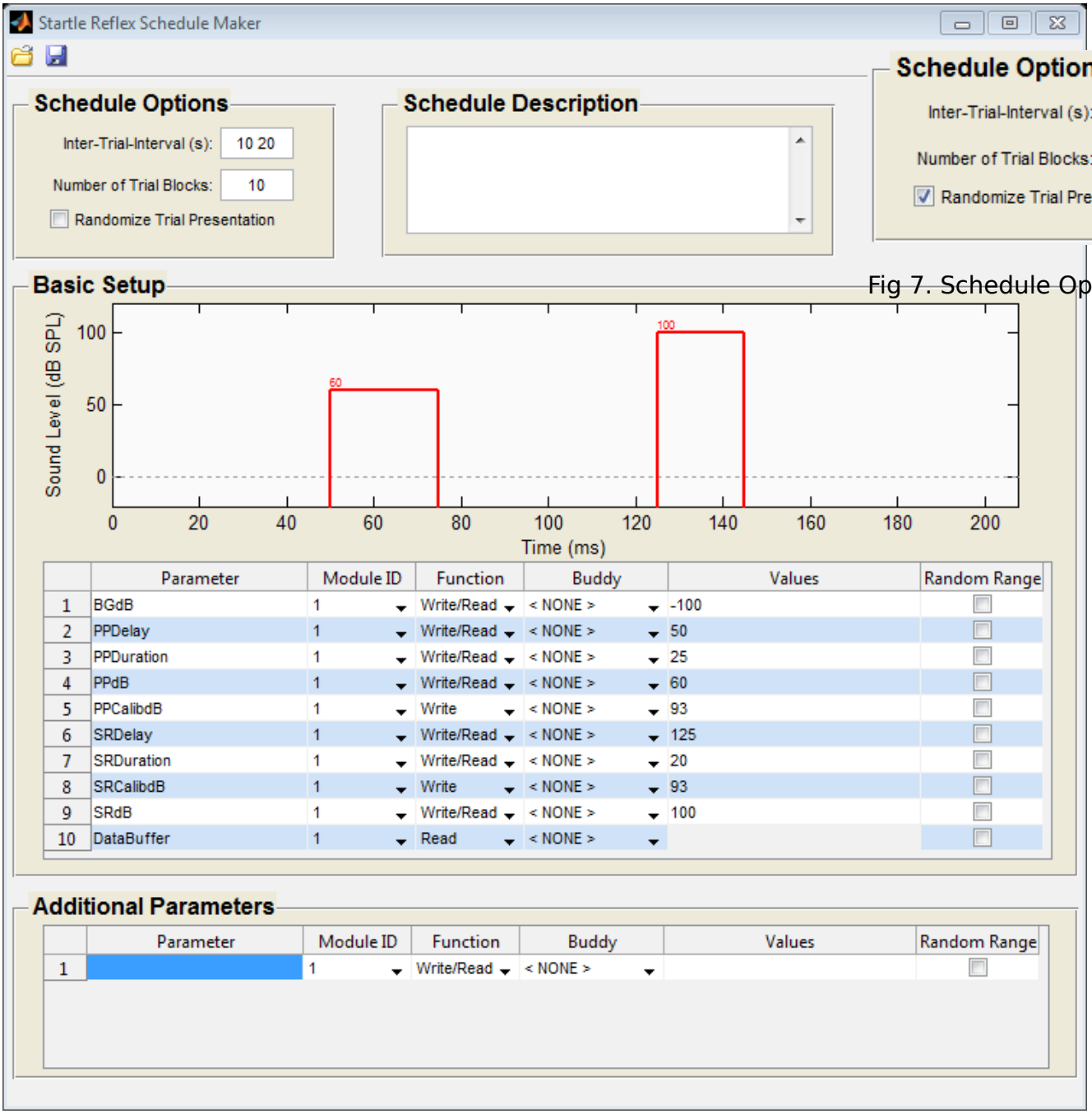


Fig 6. Schedule Generator GUI

Fig 7. Schedule Options

Schedule Options

These fields control the presentation of trials within a session. The Inter-Trial Interval specifies (in seconds) the time delay between consecutive trials. There are two

ways to use this field. If a static inter-trial interval is desired, then a single scalar value can be entered; ex: ‘5’ (without quotes) indicates 5 seconds will be inserted between trials. Alternatively, it is often preferred to randomize the inter-inter trial interval. When two values are entered into this field (with a space between numbers), it is understood as a range from which to generate a random delay before each trial is triggered; ex ’10 20’ (without quotes) indicates that a random inter-trial interval will be chosen for each trial with the shortest possible being 10 seconds and the longest possible being 20 seconds. In this case, the interval is chosen from a flat distribution within the specified range.

The “Number of Trial Blocks” field specifies the number of replicates for each parameter specified for the schedule (described in the next section). The basic purpose of this is to specify how many times each stimulus type is presented during a session. The example in figure 7 indicates 10 repetitions of each stimulus.

The “Randomize Trial Presentation” checkbox tells the software to present stimuli in a randomized fashion.

Basic Setup

The basic setup table contains parameters most often used to generate startle reflex protocols. Additional parameters can be included (described later). Modifying the values and options in this table may take some training, but with practice it is a powerful tool for rapidly generating simple and complex protocols.

The “Parameter” column cannot be edited by the user. Each parameter has a corresponding parameter tag in the RPvds file associated with the stimulus module (with the exception of DataBuffer). Here is a list of the basic function of the basic setup parameters:

- BGdB Background sound level in decibels
- PPDelay Delay in milliseconds from trial trigger to the onset of the pre-pulse stimulus
- PPDuration Duration in milliseconds of the pre-pulse stimulus
- PPdB Sound level of the pre-pulse stimulus
- PPCalibdB Calibrated sound level (ex: at 1V; discussed later) of the pre-pulse stimulus
- SRDelay Delay in milliseconds from trial trigger to the onset of the startle eliciting stimulus
- SRDuration Duration in milliseconds of the startle eliciting stimulus
- SRCalibdB Calibrated sound level (ex: at 1V; discussed later) of the startle eliciting stimulus
- SRdB Sound level of the startle eliciting stimulus
- DataBuffer Pointer to data buffer(s) on acquisition module

While these parameters are not required for the Startle Reflex software to run a session, they were selected to facilitate the creation of PPI and classic startle reflex paradigms.

Additional parameters can be manually specified in the corresponding table at the bottom of the Schedule Generator GUI.

The values for each parameter are specified in the “Values” column. This field can accept a few different types of input:

	examples
• a single numerical value	5
• an array of numerical values	1 2 3 4 5 6
• simple MATLAB expression	logspace(1,2,6)
	1000 * 2 .^ (0:0.5:4)

	Parameter	Module ID	Function	Buddy	Values	Random Range
1	BGdB	1	Write/Read	< NONE >	-100	
2	PPDelay	1	Write/Read	< NONE >	50	
3	PPDuration	1	Write/Read	< NONE >	25	
4	PPdB	1	Write/Read	< NONE >	60	
5	PPCalibdB	1	Write	< NONE >	93	
6	SRDelay	1	Write/Read	< NONE >	125	
7	SRDuration	1	Write/Read	< NONE >	20	
8	SRCalibdB	1	Write	< NONE >	93	
9	SRdB	1	Write/Read	< NONE >	100	
10	DataBuffer	1	Read	< NONE >		

Fig 8. Basic Setup

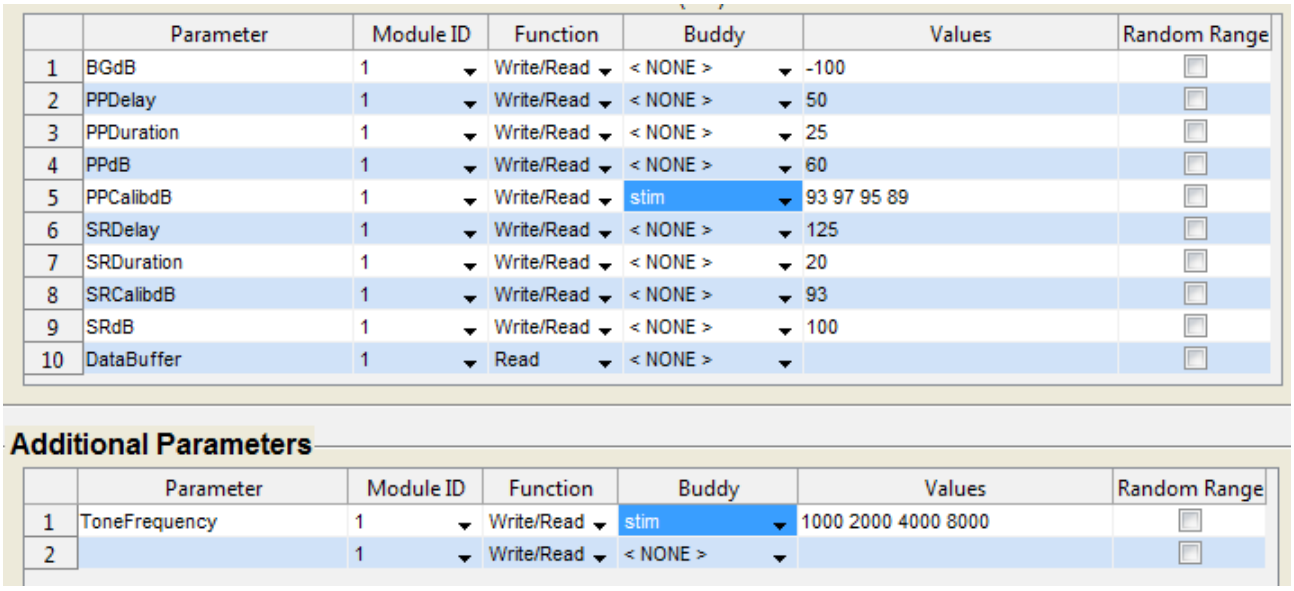
The “Module ID” column (fig 8) specifies which module (stimulus or acquisition) ID the RPvds parameters are present. All of the basic parameters will be located on the stimulus module (stimulus module 1 in Fig 8). Only the “DataBuffer” parameter is located on the acquisition module (acquisition module 1 in Fig 8). If, for example, the pre-pulse stimulus is being generated on stimulus module 1 and the startle eliciting stimulus on stimulus module 2, then the “PP” parameters would have a 1 (as in Fig 8) and the “SR” parameters would have a 2 in the module ID column.

The “Function” column is used to specify if a value is written, read, or both to a module. The “Read” option indicates that the values for these parameters will be retrieved by the startle software on each trial and paired with the startle response data (DataBuffer). The DataBuffer parameter function must be set to “Read.”

The “Buddy” column provides a method for having the values of more than one parameter co-vary in the generated schedule. To add a ‘Buddy’ select a dropdown box and select “< ADD NEW >”. A prompt will ask for a buddy name. Enter some textual value, such as “stim”. This will create a new “Buddy” which can be used by multiple parameters across both basic and frequency parameters. This feature is especially useful

when entering different calibration values for multiple frequencies or adjusting bandwidth of a narrow-band noise as frequency increases.

In the example in figure 9, the additional parameter “ToneFrequency” was given four values. The basic parameter, “PPCalibdB” was also given four values. The buddy “stim” was created and selected for both parameters. This setup will generate a schedule in which the “ToneFrequency” value of 1000 would always be paired with the “PPCalibdB” value of 93. The same would be for 2000 and 97, 4000 and 95, 8000 and 89.



	Parameter	Module ID	Function	Buddy	Values	Random Range
1	BGdB	1	Write/Read	< NONE >	-100	
2	PPDelay	1	Write/Read	< NONE >	50	
3	PPDuration	1	Write/Read	< NONE >	25	
4	PPdB	1	Write/Read	< NONE >	60	
5	PPCalibdB	1	Write/Read	stim	93 97 95 89	
6	SRDelay	1	Write/Read	< NONE >	125	
7	SRDuration	1	Write/Read	< NONE >	20	
8	SRCalibdB	1	Write/Read	< NONE >	93	
9	SRdB	1	Write/Read	< NONE >	100	
10	DataBuffer	1	Read	< NONE >		

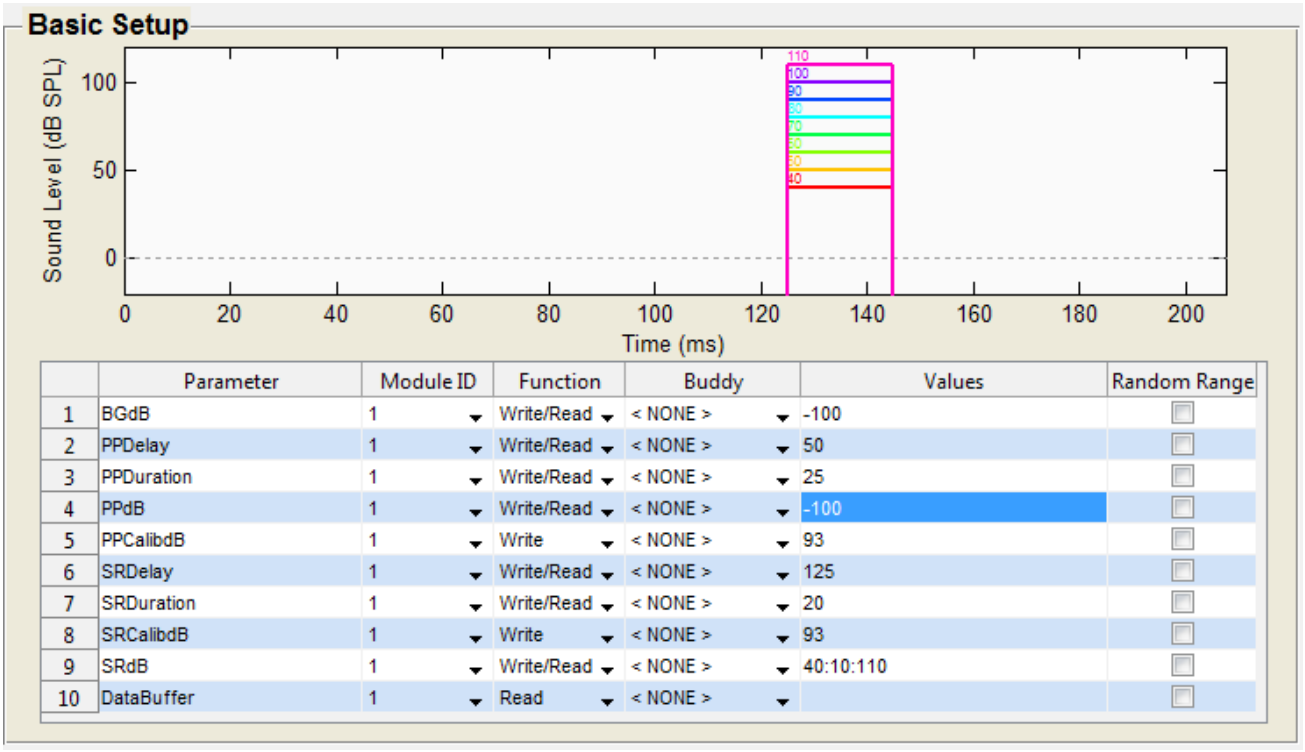
	Parameter	Module ID	Function	Buddy	Values	Random Range
1	ToneFrequency	1	Write/Read	stim	1000 2000 4000 8000	
2		1	Write/Read	< NONE >		

Fig 9. Using “Buddy” to co-vary multiple parameter values

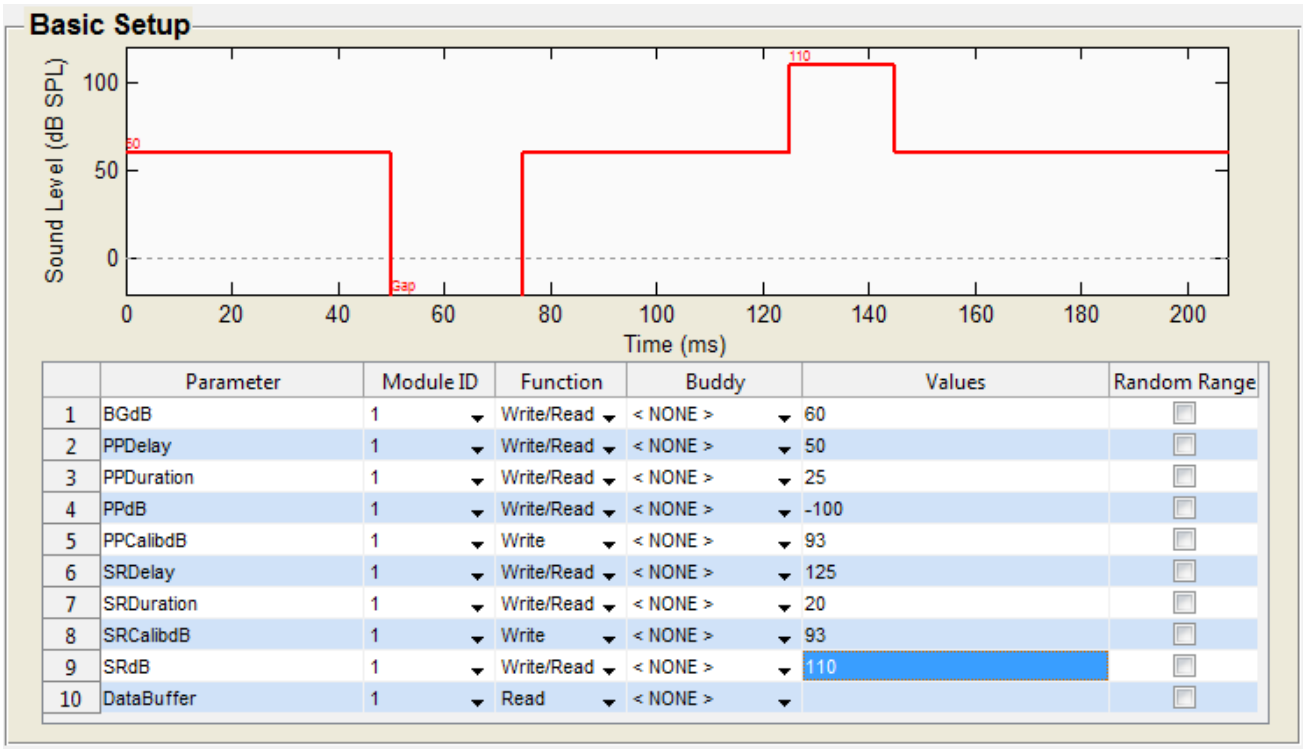
The final option available to parameters with “Write” functionality is the “Random Range” column. Checking this box will create a randomized (flat distribution) value between a range specified in corresponding “Values” field. If this field is checked, then the corresponding “Values” field must have two values, such as ‘4 10’ (without the quotes). This will generate a random value between 4 and 10.

You will notice that updating any of the basic parameters will automatically update the schematic. Adding or modifying additional parameters will not update this schematic. Figure 10 (top) is an example of a startle reflex input/output function. Figure 10 (bottom) is an example of a startle reflex gap-detection protocol.

Once parameters are specified, the trials generated for the schedule can be previewed by clicking the “View Trials” button in the top right of the screen.



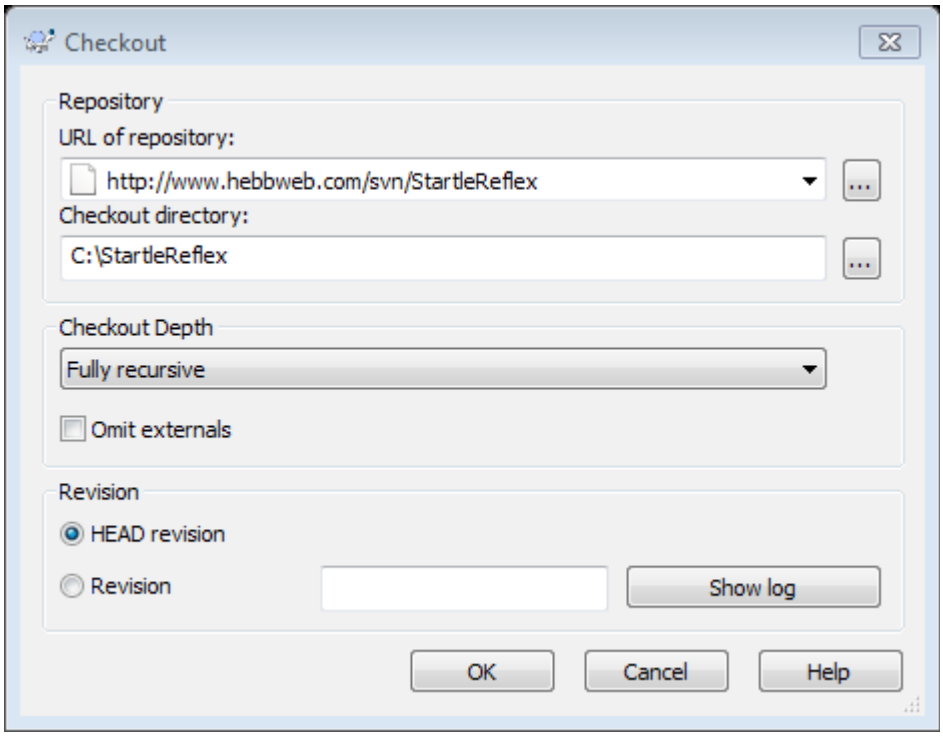
	Parameter	Module ID	Function	Buddy	Values	Random Range
1	BGdB	1	Write/Read	< NONE >	-100	
2	PPDelay	1	Write/Read	< NONE >	50	
3	PPDuration	1	Write/Read	< NONE >	25	
4	PPdB	1	Write/Read	< NONE >	-100	
5	PPCalibdB	1	Write	< NONE >	93	
6	SRDelay	1	Write/Read	< NONE >	125	
7	SRDuration	1	Write/Read	< NONE >	20	
8	SRCalibdB	1	Write	< NONE >	93	
9	SRdB	1	Write/Read	< NONE >	40:10:110	
10	DataBuffer	1	Read	< NONE >		



	Parameter	Module ID	Function	Buddy	Values	Random Range
1	BGdB	1	Write/Read	< NONE >	60	
2	PPDelay	1	Write/Read	< NONE >	50	
3	PPDuration	1	Write/Read	< NONE >	25	
4	PPdB	1	Write/Read	< NONE >	-100	
5	PPCalibdB	1	Write	< NONE >	93	
6	SRDelay	1	Write/Read	< NONE >	125	
7	SRDuration	1	Write/Read	< NONE >	20	
8	SRCalibdB	1	Write	< NONE >	93	
9	SRdB	1	Write/Read	< NONE >	110	
10	DataBuffer	1	Read	< NONE >		

Fig. 10 Schedule examples. Top: Startle reflex input/output function. Bottom: Gap-prepulse inhibition of the acoustic startle reflex.

Setting up SVN to retrieve Startle Reflex software

1. Download and install latest version of Tortoise SVN
    a. http://tortoisesvn.net/downloads
    b. Download the 32-bit or 64-bit version based on your operating system
    c. You will need to restart your system
2. SVN Checkout
    a. Navigate to Matlab directory
        i. Usually something like: C:\Program Files (x86)\MATLAB\R2007b\
    b. Add a directory called 'work' if one does not yet exist
    c. Open the work directory
    d. Right-click and select 'SVN Checkout' from the context menu
    e. Enter "http://www.hebbweb.net/svn/StartleReflex" in the URL of repository field.
    f. 

    g. Click OK and you will be prompted for a Username and Password
    h. Ask Dan for the Username and Password
    i. The program should begin downloading all of the files in the repository from the server
3. A Matlab script called "startup.m" needs to be obtained and put into the "work" directory. This file will be automatically called when Matlab launches and it will add the necessary files to the Matlab path.