EPSYCH SOFTWARE MANUAL: SANES LAB

This document contains information for setting up the rig, designing an experiment using the EPSYCH toolbox, running a behavioral session, and obtaining the saved data. It assumes basic familiarity with RPVdsEx, OpenEx, MATLAB, and Git.

**SanesLab github email/password:** saneslab\_github@cns.nyu.edu/awakebehav1ng

**SanesLab SmartGit master password:** bridgeripeleepour

CONTENTS

1. SETUP
2. SPEAKER CALIBRATION
3. DESIGNING EXPERIMENTS
4. RUNNING EXPERIMENTS
5. Extracting AND ANALYZING data
6. ADDITIONAL RESOURCES

---------------------------------------------------------------------------------------------------------------

1. SETUP
   1. **Hardware**

On the rack:

Tucker Davis Technologies RZ6 Auditory Processor

Tucker Davis Technologies RZ5

Tucker Davis Technologies PP24 Patch Panel

Crown D-75A Amplifier

Custom control board from Peter Lau; [more info needed here]

New Era NE-300 Syringe Pump

PC

In the booth:

Custom made IR nose poke

Custom made IR spout sensor

Spout

Speaker

[Insert diagram of hardware set up]

\*Note: Make sure that only Ch 1 of the Crown D75 Amplifier is connected to the speaker and RZ6. Ch 2 should not be connected to anything. Also, couple out A and out B of the RZ6 to the same input of the Crown D75 Amplifier

* 1. **Software**

Windows 7 (not tested with other versions)

MATLAB 2012b (not tested with later versions)

EPSYCH toolbox (see below)

* 1. **To install EPSYCH toolbox on a new SanesLab computer:**

1. Log onto <https://github.com> *Username:* SanesLab; *Password:* awakebehav1ng
2. Clone the SanesLab/epsych repository. <https://github.com/SanesLab/epsych>. You now have an up-to-date copy of all of the code associated with the EPSYCH toolbox.
3. Open MATLAB.
4. The correct paths must be set before epsych can run experiments. Either:
   1. Enter the following into the command window:

addpath(‘C:\gits\espych\’);

epsych\_startup;

\*Note: The repository may have been cloned to a different directory. Replace the path with the appropriate location.

* 1. Or, write these lines in a script, saved to the default startup Matlab folder as “startup.m”.

1. The commands for the 2 main launch menus are Appetitive\_Training\_Menu (for early stages of appetitive training) and ep\_LaunchPad (for creating and running trial-based behavior protocols, ephys, and speaker calibration). You may wish to use the shortcuts tab to create a shortcut button for each.
2. Download the SmartGit desktop client: <http://www.syntevo.com/smartgit/>
3. Use SmartGit to keep your code updated and in sync with the SanesLab fork.

Document master SmartGit password if one is created.

*983 master SmartGit password:* bridgeripeleepour

*1012 master password:* awakebehav1ng

* 1. **Pull** from the SanesLab remote repository regularly, to ensure that all rigs are synchronized and using up-to-date programs.
  2. When a change warranting documentation is made, **Commit** and **Push** to the remote repository, the SanesLab fork URL.
  3. After significant commits, log onto github, and create a **New pull request** to merge changes back into the master dstolz/epsych repository.

---------------------------------------------------------------------------------------------------------------

1. SPEAKER CALIBRATION

**A. Launch EPSYCH toolbox:**

1. Open MATLAB
2. Type ep\_Launchpad in the command window. (Or alternatively, create and use a shortcut on the MATLAB menu bar).

**B. Calibrate speaker using the EPSYCH toolbox:**

1. Make sure that the electrostatic switch on the front of the RZ6 is set to OFF.
2. Set the attenuation of Ch1 on the Crown D75A to 30 dB (exactly 15 clicks of the knob from the far right).
3. Set up microphone in the center of the booth (but without the cage; it creates large distortions). Plug the microphone output into IN-A on RZ6.
4. Click Calibrate Utility on the EPSYCH launchpad menu.
5. Set the stimulus and acquisition modules to RZ6
6. Set the sampling rate to 100 kHz
7. Enter the microphone sensitivity (mV/dB SPL). \*Note that you may need to amplify the recorded signal so that it falls within a reasonable voltage range of the RZ6, which is +/- 10V. To amplify the recorded signal, set the gain switch on the front of the RZ6 to AMP and set the knob to the desired amplification. If amplifying signal, make sure to adjust microphone sensitivity in the GUI accordingly. (20 dB of amplification = 10X sensitivity). If not amplifying, set the switch to OFF. Ideally, your amplified signal should be above 1 V and below 9V (to avoid clipping). RECOMMENDED: 40 dB GAIN (100x)
8. Set the Signal Amplitude in the GUI (found under the Settings dropdown menu) to 1 V.
9. Select Tone or Noise from GUI dropdown menu.
10. Press RUN. \*Note: For an unknown reason (Active X bug, perhaps?) the program may not calibrate the first time Run is clicked. If this occurs, just click Run again.
11. Save file
12. Check calibration with handheld sound level meter.
13. Check calibration (how flat is the corrected output?) using PostCalibrationUtil program. \*Note: Any program using Dan’s calibration macro MUST set the normalization parameter correctly.

---------------------------------------------------------------------------------------------------------------

1. DESIGNING EXPERIMENTS

[Need documentation here about the Bitmask Generator. Also, where is bitmask file \*.epdp linked to the protocol? Used to be in “Display Prefs” but no longer an option? How is the bitmask table linked to the state machine in the RPVds circuit? Or is it hard coded in the circuit itself?]

**A. Design a behavior only experiment from scratch**

1. Click on Experiment Design on the EPSYCH launchpad menu.
2. Click on the + button on the left to add a TDT module.
3. Select “No” when asked if the experiment will use OpenEx.
4. Select RZ6 and a single module.
5. Rename device Behavior.
6. Select desired RPVds circuit file.
7. A list of editable parameters will appear.
8. Remove any parameters that you don’t want to rove systematically during the experiment by highlighting them and pressing the Remove Parameter button. Note: Remove any parameter that ends with “\_TTL” or is named “buffer,” “bufferSize,” or “BuffTrig”. These are parameter tags used for displaying data and for capturing sound from the microphone only, and should not be given values.

REQUIRED PARAMETERS (keep, no matter what you wish to rove):

Reminder, RespLatency, TrialType

1. Input values for the remaining parameters. Explanations and valid values follow. Note that you may not see some of the parameters below, depending on the RPVds file that you’ve associated with the protocol.

TrialType: 0 (GO); 1(NOGO) \*Note: REQUIRED PARAMETER

Reminder: 0 (not a reminder trial); 1 (reminder trial) \* Note: REQUIRED PARAMETER

Expected: 0 (unexpected sound onset time); 1 (expected sound onset time)

Freq: Sound frequency in Hz

FM\_Rate: Rate of frequency modulation in Hz

FM\_Depth: Depth of frequency modulation (units? Proportion? Percent?)

dBSPL: Calibrated sound level in dB SPL \*Note: If you want to rove dB SPL, you must associate a calibration file with the sound parameter (e.g. Freq, Noise, etc.). The parameter must have the same name as the Calibration Utility macro in the RPVds circuit. Currently (Nov 2015) calibrated dB SPL roving has not been tested with the Sanes Lab Appetitive\_detection\_GUI.

Stim\_Duration: Length of sound stimulus (msec)

MinPokeDur: Minimum poke duration needed to initiate a trial (msec)

Silent\_delay: Length of time between trial onset and sound onset (msec)

RespWinDelay: Length of time between trial onset and the opening of the

response window.

RespWinDur: Length of time the response window is open (msec). When open,

animal is allowed to make a behavioral response.

ITI\_dur: Length of inter-trial interval, during which time the animal cannot initiate

a new trial. The ITI begins at the end of the response window.

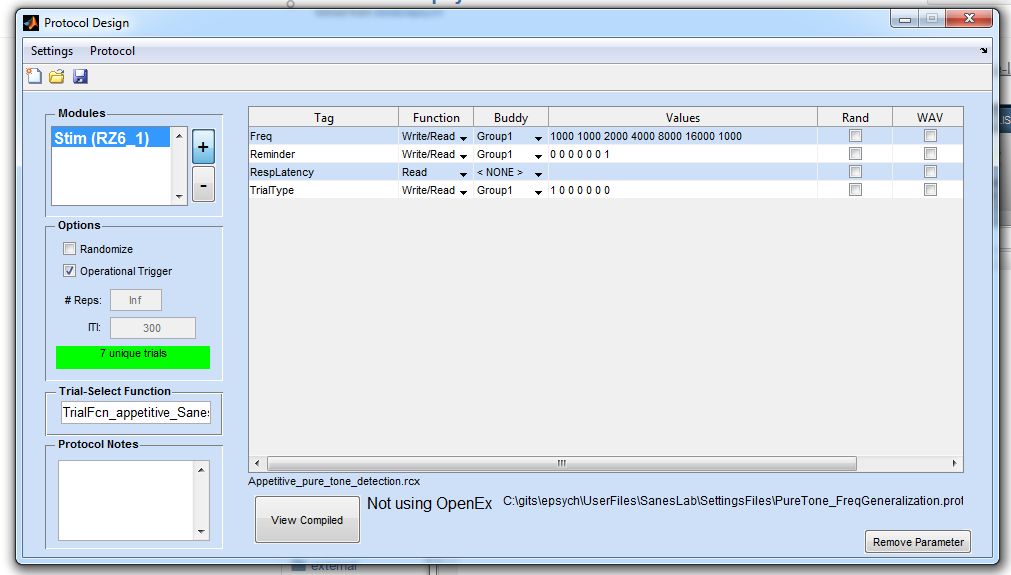
to\_duration: Length of time the animal must wait before initiating a new trial after false alarming (msec). Lights may or may not turn off during this period, depending on the circuit running.

reward\_dur: length of time (msec) pump will run for each reward.

Optostim: Optogenetic trigger initiated at trial onset. 0 (light off); 1(light on)

resp\_latency: Time from trial onset to animal’s behavioral response (msec). Set this parameter to Read. When an animal fails to make a behavioral choice before the response window has elapsed, this value is set to 0. Note: REQUIRED PARAMETER

1. Enter values for each parameter that you want to rove throughout the experiment. The values column will accept any MATLAB code that evaluates to a numeric scalar or vector. For example, 1000:2000:5000 or 1000 3000 5000 are both valid entries.
2. If you want the value of particular parameter to vary randomly from trial to trial, you can specify two values that form the boundaries of a uniform distrbution, and check the box in the Rand column. During the experiment, a random value will be selected from the uniform distribution using the rand function. \*Note: Use of the Rand option is currently untested with the SanesLab Appetitive\_detection\_GUI.m.
3. Alternatively, if you want to use pre-generated WAV files as your stimuli, you can check the box in the WAV column. A new window will open, allowing you to select WAV files and add them to a list. The order of the files in the table is the order for the specified parameter tag. \*Note: To avoid aliasing and stimulus artifacts, the sampling rate of each WAV file must be the same as the sampling rate of the TDT module on which it will run. Also, currently, WAV file stimuli have not been tested with the SanesLab Appetitive\_detection\_GUI.m.
4. If you want certain variables to co-vary, you can link them together using the Buddy column. For example, if you want to create a simple GO/NOGO paradigm, with five GO frequencies (1000-16000 Hz), and a 1000 Hz reminder trial, you would enter data as shown below. \*Note: “Buddied” parameters must have the same number of values.



1. Uncheck randomize on the left side of the screen so that we can use a custom trial selection function.
2. Check Operational Trigger on the left side of the screen to indicate that animals will be initiating their own trials by nose poking. \*Note: uncheck this if running any paradigm that does not require the animal to initiate its own trials, such as an aversive paradigm, shuttlebox paradigm, startle response paradigm, etc. Currently (Nov 2015), the SanesLab GUI does not support such paradigms.
3. Enter TrialFcn\_appetitive\_SanesLab in the Trial-Select Function box on the left side of the screen. \*Note: do not include the .m extension at the end of the file name.
4. Check the trial structure by clicking on View Complied at the bottom of the screen. If everything looks as it should, save your protocol.

**B. Design a combined behavioral/ephys experiment from scratch:**

1. Click on Experiment Design on the EPSYCH launchpad menu.
2. Click on the + button on the left to add a TDT module.
3. Select “Yes” when asked if the experiment will use OpenEx.
4. Rename device Behavior \*Note: Renaming is very important here
5. Select “Yes” when asked if you want to read parameter tags from existing RPvds file.
6. Select desired RPVds circuit file.
7. A list of editable parameters will appear.
8. Follow steps 7-14 in **Section III.A.**

**C. Design an ephys only experiment from scratch:**

1. Follow steps 1-7 in **Section III.B.**.
2. Enter values for each parameter that you want to rove throughout the experiment. When entering multiple values for a single parameter, separate them with spaces (commas might work too), or use MATLAB’s colon notation (i.e. start:step:end).
3. If you want calibrated sound stimuli, associate a calibration file with an appropriate parameter (i.e. Freq) by selecting the appropriate calibration file from the dropdown menu under the Calib. column on the far right. \*Note: The parameter must have the same name as the Calibration Utility macro in the RPVds circuit.
4. If you want to randomize the order of stimulus presentation, check randomize on the left side of the screen. \*Note: If you want a truly random order of stimuli, such that each time you run the protocol, a new order is generated, select Settings in the top menu bar, and select Compile at RunTime. If you’d prefer a pseudorandom order, such that each time the protocol is run, the same randomized order of stimuli is presented, do not select Compile at RunTime.
5. Uncheck Operational Trigger on the left side of the screen to indicate that animals will not be initiating their own trials.
6. On the left side of the window, enter the desired number of repetitions for each trial.
7. On the left side of the window, enter the desired inter-trial interval (ITI) in msec. If you wish to randomize the ITI, you can specify two values that form the boundaries of a uniform distribution. During the experiment, a random value will be drawn from this distribution for each ITI.
8. Check the trial structure by clicking the View Compiled button on the bottom of the window. \*Note: The total number of trials displayed in the table will be truncated if over (1000? 2000?) trials are generated.
9. If everything looks good, save your protocol.

---------------------------------------------------------------------------------------------------------------

1. RUNNING EXPERIMENTS
2. **Running a behavior only experiment**
3. Click on Run Behavior Experiment on the EPSYCH launchpad menu.
4. Click on Function Definitions on the top menu bar and select Timer Functions
5. In the Start Timer Function box, enter ep\_TimerFcn\_Start\_SanesLab
6. In the RunTime Timer Function box, enter ep\_TimerFcn\_RunTime\_SanesLab
7. In the Stop Timer Function box, enter ep\_TimerFcn\_Stop\_SanesLab
8. In the Error Timer Function box, enter ep\_TimerFcn\_Error. Click OK.
9. Click on Function Definitions on the top menu bar and select Saving Function
10. Enter ep\_SaveDataFcn\_SanesLab. Click OK.
11. Click on Function Definitions on the top menu bar and select Box Figure
12. Enter Appetitive\_detection\_GUI. Click OK.
13. Click on Function Definitions on the top menu bar and select Add Subject
14. Enter ep\_AddSubject\_SanesLab. Click OK. \*Note: Function definitions generally only need to be set the first time a behavior or behavior/ephys experiment is launched. Epsych should automatically remember function definitions for future launches.
15. Click + Subject.
16. Enter NYU ID of animal. \*Note: Data collected during the experiment will be saved to a file automatically named with the ID and date. It is therefore recommended that the ID name begins with a character, rather than a number (eg. ID\_123456), such that MATLAB has no difficulty opening the file for analysis later.
17. Select the sex, condition, and age of animal from the dropdown menu.
18. Add any notes about the experiment as desired. Click Done.
19. Select your desired protocol. (e.g. filename.prot) and click Open.
20. If desired, double check the trial structure by clicking View Trials.
21. If desired, save a configuration file (e.g. ID\_123456.config) by clicking the save button at the top of the window. The configuration file will save all the current settings (ID name, protocol, etc), so you can quickly launch the same configuration day after day. \*Note: To launch a previously created configuration, instead of clicking + Subject in step 13, click the open folder icon at the top of the window, and select the configuration file you want.
22. Click Run to launch the GUI
23. Select the appropriate calibration file if prompted (e.g. ceiling\_speaker.cal).
24. Run experiment and adjust parameters as needed using GUI controls. Items that are roved and items that are not present in the RPVds circuit are automatically disabled.
25. To stop the experiment, press the red Stop button in the Run Experiment Dialogue Box.
26. Save the file when prompted.

\*Note: During runtime, data is automatically backed up to C:\Users\sanesadmin\Documents\MATLAB\DATA. See line 18 of ep\_TimerFcn\_Start\_SanesLab.m

1. Close the GUI window.
2. **Running a combined behavior-ephys experiment**
3. Launch OpenProject by double clicking on the desktop shortcut.
4. Click OpenProject in the top menu bar, and select New Project.
5. Click and drag the RPVdsEx Icon from the Applications window on the left, to the Application column of the Launch Details table, on the right.
6. In the same manner, click and drag the Controller and the Scope icons to the table.
7. Name and save your project (default directory is C:\TDT\OpenEx\Myprojects). Click OK.

\*Note: All RPVds files that you associate with this project will be copied into the directory you select here. If you need to edit the RPVds file for an existing behavior/ephys experiment, be sure to edit the copy of the file in this directory. If you want to save your edited changes for others to use (i.e. to post to github) copy the changes into C:/gits/epsych/UserFiles/SanesLab/RPVdsCircuits.

1. Click Setup in the top menu bar and select Sweep Loop. Uncheck the Enabled box and click OK.
2. Repeat step 6 for Condition Loop
3. Click Setup in the top menu bar and select Stimulation Timing. Uncheck the Stimulation box and click OK.
4. Click Setup in the top menu bar and select Acquisition Timing. Uncheck the Acquisition box and click OK.
5. Set up the RZ5 to acquire ephys data. \*Note: Instructions below assume that SpikePac is used and incorporated into the RZ5 RPVds circuit. See TDT documentation for more detail.
   1. Click on the grey RZ5\_1 box on the left (should now be outlined in magenta).
   2. Click on the […] button to the right of the File Name box.
   3. Select an appropriate RPVds file to associate with the RZ5.
   4. **Name the device Phys**. Click OK. \*Note: Renaming is very important
   5. A table is now populated with various store types.
      1. Wave consists of the raw, unfiltered waveform captured at a high sampling rate. Set the mode for this tag to Store.
      2. eNeu consists of thresholded snippets captured at a high sampling rate. Set the mode for this tag to Store.
      3. pNeu consists of decimated waveforms captured at a low sampling rate. As it will be used for real-time plotting only,we do not need to store the data. Thus, set the mode for this tag to Fetch.
6. Set up the RZ6 to generate sound stimuli, and acquire behavioral data
   1. Click on the grey RZ6\_1 box on the left (should now be outlined in magenta).
   2. Click on the […] button to the right of the File Name box
   3. Select an appropriate RPVds file to associate with the RZ6.
   4. **Name the device Behavior**. Click OK. \*Note: Renaming is very important
   5. Verify that all of the RZ6 tags are set to Store.
7. Set up OpenController to allow for realtime spike sorting and viewing.
   1. Right click on the Controller\_1 icon on the left side of the screen.
   2. Select Import
   3. Select premade controller display file to use during the experiment and select “Yes” when asked if you want to discard the current setup: C:/gits/epsych/ UserFiles/SanesLab/SettingsFiles/Ephys\_Controller.xpc \*Note: if you wish to customize or change the Open Controller display, or create it from scratch, refer to the OpenController Manual from TDT.
8. Set up OpenScope to allow for semi-real time raster and PSTH generation.
   1. [Need more info here]
9. Click OpenProject on the top left menu bar and click Save Project. \*Note: Once a project is created and saved, it can be launched easily by clicking Open Project on the top left menu bar, and selecting Load Project.
10. With OpenProject still open, launch MATLAB and click on Run Behavior Experiment on the EPSYCH launchpad menu.
11. Follow the instructions in **Section IV. A** through step 20. \*Note: When selecting a protocol for a combined behavior/ephys experiment, be sure that the protocol was designed for use with OpenEx (see **Section III.B.**). If you’re unsure whether the protocol is OpenEx compatible, highlight the protocol after you’ve loaded it into the Run Experiment window, and click on the Edit Protocol on the right. The Protocol Design window will launch. On the bottom of the window, you will see text that reads either Using OpenEx or Not Using OpenEx.

\* what are the rules about matching the experiment config file to the circuit that was loaded and associated with the devices in OpenProject/WorkBench?

1. When prompted, select a TDT Tank to store the electrophysiology data by double clicking on an existing tank name on the left side of the window. Click Select to close dialogue.
2. Continue with steps 20-25 of **Section IV.A.**
3. **Running an ephys only experiment**
4. Follow steps 1-17 in **Section IV.B.**
5. With OpenProject still open, launch MATLAB click on Run Electrophys Experiment on the EPSYCH launchpad menu.
6. Click on the Locate Protocol Dir…button in the Protocol Setup section of the window: C:\gits\epsych\UserFiles\SanesLab\SettingsFiles\MLC\BasicCharacterization
7. Click Select Folder
8. A list of Electrophysiological protocols will be displayed. Re-order protocols as desired by using the up and down arrows on the right. \*must desired be on top?
9. Clicking Select Tank.
10. Select a TDT tank to store the data in by double clicking on an existing tank name on the left side of the window. Click Select to close dialogue.
11. Click Record \*Note: If you do not want to save the data collected, click Preview instead. Preview generates a temporary block that will be deleted when you click Stop.
12. Progress for the current experiment is displayed on the right. Program will automatically halt once protocol is complete. If you need to terminate the experiment early, click Halt.

---------------------------------------------------------------------------------------------------------------

1. Extracting AND ANALYZING data
2. **Behavior data**
3. [Need info here]
4. **Ephys data (TDT TANKS)**
5. [Need info here]
6. **SQL Databases**
7. [Need info here]
8. ADDITIONAL RESOURCES
   1. **GitHub basics**

Tutorial: <https://guides.github.com/activities/hello-world/>

Flow Guide: <https://guides.github.com/introduction/flow/index.html>

Command line cheat sheet: <https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>

**NOTES:**

TrialFcn\_appetitive\_SanesLab.m

* If an animal is forced to repeat a NOGO trial after a FA, and gets the trial correct, the next trial will always be a GO trial
* The count of consecutive NOGO trials includes repeat NOGOs after FAs
* Sounds occurring at unexpected latencies will never be presented back to back
* During runtime, data is continuously backed up to: C:\Users\sanesadmin\Documents\MATLAB\DATA