AWAKE-BEHAVING MANUAL: SANES LAB

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

**SanesLab github email:** [saneslab\_github@cns.nyu.edu](mailto:saneslab_github@cns.nyu.edu)

**TABLE OF CONTENTS**

1. SETUP
2. SPEAKER CALIBRATION
3. DESIGNING EXPERIMENTS
4. RUNNING EXPERIMENTS
5. Extracting AND ANALYZING data
6. ADDITIONAL RESOURCES
7. SETUP
   1. **Tucker Davis Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| Tucker-Davis Technologies | RZ6 | **Auditory processor** |
| Tucker-Davis Technologies | RZ5D *or*  RZ2-2 with QZDSP and optional DSP-S (for use with the RS4 data streamer) | **Bioamp Base Processor** for 16 ch (RZ5) or up to 128 ch (RZ2) |
| Tucker-Davis Technologies | PP24 | **BNC Patch Panel** |
| Tucker-Davis Technologies | RS4 (requires RZ2 DSP-S) | **Data Streamer** (only necessary for 64+ channel recording) |
| Tucker-Davis Technologies | PO5e | **PC Interface Card** |
| Tucker-Davis Technologies | TB32 *or*  PZ5 with custom DB37 to DB26 cables | **32 Channel digitizer** (TB32)  **64 Channel digitizer** (PZ5)  (Interfaces with TBSI wireless receiver) |
| Cables on Demand | CS-DSSMDB37MM-022.5 | **37 Pin Connector** for TB32 and TBSI wireless receiver: DB 37 Standard shielded D-Sub Cable Male/Male, 2.5 ft  (Only necessary if using TB32) |

* 1. **Audio hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| JBL Commercial | 280Z | **Auditory Preamplifier** (replaces Crown D75, which is now discontinued) |
| Vifa | DX25TG05-04 | **Speaker** (4 Ohm Tweeter) |
| In house machine shop | N/A | **Magnetic speaker holders** |

* 1. **Triangle Biosystems Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| Triangle Biosystems | 911-1063-00 | **64ch Wireless Analog Receiver** |
| Triangle Biosystems | B10-3163-GK  X50-4113-GP | **64ch wireless recording headstage**  Optional optogenetic add-on |

* 1. **Water Delivery Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| New-Era | NE-1000 | **Single channel programmable syringe pump** |
| Celestron NexStar | 93920 | **Serial to RS-232 cable:** Connects pump to computer |
| Soldered in house | N/A | **DB9 to BNC connector:** Connects PP24 to pump for TTL triggering. TTL signal should be routed to Pin 2. BNC shield (ground) should be routed to pin 9. |
| VWR | 427450 | **Tubing:** PE240, 0.066” ID, 0.095” OD, 10’ length |
| Nordson | 7018068 | **Luer-Lock connector:** 0.5” length, 0.053” ID, 0.065” OD, 15 guage |
| Shapeways | SKP file available upon request | **Plastic Spout and spout holder** |
| In house machine shop | N/A | **Metal spout** |
| In house machine shop | N/A | **Metal Floorplate** |
| Precision Brand | 33040 | **Metal hose clamp:** Precision brand M6S Micro Seal, Miniature All Stainless Worm Gear Hose Clamp, 5/16”. Used for attaching shocker cables. |

* 1. **Food Delivery Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| Med Associates | ENV-203-201R | **Pellet Dispenser** |
| Med Associates | SG-230RC | **Powered Adapter:** TTL to 28V DC with Coax Connector |
| Med Associates | SG-501 | **Power Supply:** 28V, 1A DC |
| Med Associates | SG-210CP-25 | **Power Cable** (25’) |
| Med Associates | SG-216A-20 | **Mini-molex Extension**  (3 pin) |
| Shapeways | Food Hopper V2 | **Food Hopper Tray** |
| BioServ | F0163 | **Food pellets:** Dustless precision grain-based (20 mg) |

* 1. **IR Sensor/Emitter Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| Digi-Key (Lite-On Inc) | 160-1063-ND (LTE-302) | **IR Emitter**: 940 nm, 1.2V, 50 mA 0.0662 mW/sr @ 20 mA 40 Radial, Side View |
| Digi-Key (TT Electronics/Optek Technology) | 365-1085-ND (OP950) | **IR Sensor:** Photodiode 935 nm 5 nS |
| Digi-Key (Pomona Electronics) | 501-1030-ND | **BNC Breakouts:** Breakout BNC Female- 20AWG Leads |
| Grainger | 2JA46 | **Socket Head Cap Screw:** Nylon, #4 Thread Diameter, 1 ½” Length under head |
| Grainger | 2JE25 | **Socket Head Cap Screw:** Nylon ¼” Thread Dia., 1” Length Under Head |
| Grainger | 1TRP7 | **Spiral Point Tap:** Right Hand, Plug, ¼”, 20 Pitch |
| Weller Precision Prototype | Refer to order #NYU00101 | **IR Sensor/Emitter Housing for Spout** |
| In house 3D printed or can be ordered commercially with CAD | N/A. CAD designed by Brad Buran available upon request. | **Nose Poke** |

* 1. **Aversive hardware**

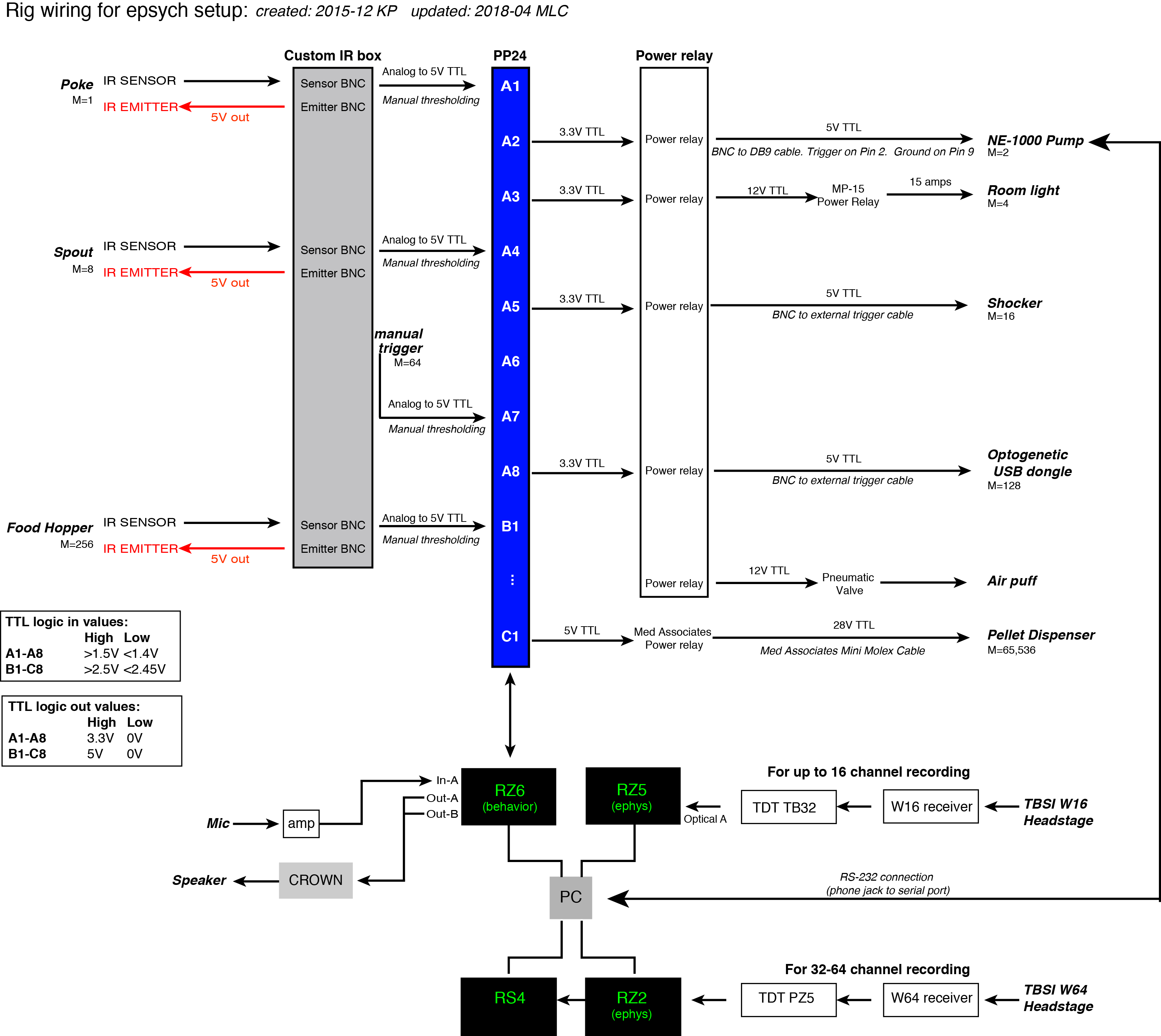
|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| Coulbourn | H13-15 | **Precision** **Animal Shocker** |
| Coulbourn | H93-19-25 | **Shocker cables:** 25 ft shock cable, 2 pole/ Universal Clips |
| Coulbourn | 802-249-FF-072 | **External trigger cable:** Plastics One 249 Plug, 5-30 V DC, 72”. Wire BNC shielding to rounded (-) side of 249 plug. Wire BNC center pin to flat (+) side of 249 plug. |
| Digi-Key (Pomona Electronics) | 501-1137-ND (3789) | **BNC Breakouts:** BNC Male-Minigrabber Test Clip |
| Ingersoll-Rand | P251SS-012-D | **Pneumatic Valve:** 3-way valve; must insert stop-valve in one of the ports to turn it into an on/off valve. Ports are 1/8”. Delivers air puff manually (via button) or automatically (via 12 V DC trigger). |
| Furman | MP-15 | **Power relay:** 120V AC. Used to control room light during time outs. |

* 1. **Other and Custom Hardware**

|  |  |  |
| --- | --- | --- |
| **Vendor** | **Product Number** | **Description** |
| N/A | N/A | **Power relay and analog to TTL converter:** (Peter Lau). Combination of a power relay (converts 3V DC input to 5 or 12 V DC output) for 8 channels, and an analog to TTL sensor (converts an analog input to a 5 V DC TTL output using a manually configurable threshold) for 8 channels. |
| Macrofab | Design and gerber files upon request (caras@nyu.edu) | **Analog to TTL converter PCBs:** Converts an analog input to a 5V DC TTL output using a manually configurable threshold. Designed by Nima in Antje’s lab. Brad designed a version that is not configurable for manual thresholding of the analog signal (must be done digitally). Each board supports one IR sensor/emitter pair. Replaces the Peter Lau box. |
| Protocase | Component list upon request (caras@nyu.edu) | **Rack mountable chassis for PCBs:** Fits 19” rack, 2U height. Each chassis houses 5 PCBs and associated hardware components. |
| McMaster Carr | 9344T31 | **Metal chicken wire for test cage** |
| Industrial Netting | XN1673-44 | **Plastic flooring for test cage** |
| CCMI Plastics | Refer to order for  NJ-CAGE | **Test Cage:** Sketchup design by Brad Buran. Ordered by Antje. |
| CyberPower | CPS-1220RMS | **Surge protector:** Rack mountable |
| B&H Video (Canare) | CA5B5B10 | **BNC-to-BNC:** Color coded cables for running from rack to booth. 5 BNCs. 10 ft length. |
| Best Buy | 9928354 | **Webcam:** Logitech C270 |

* 1. **Software**

|  |
| --- |
| Windows 7 or later (tested with Windows 7 and Windows 10) |
| MATLAB 2012b (probably good through MATLAB 2014a, but may run into active X problems with 2014b or later according to TDT). |
| Opto StimWare (TBSI) |
| EPSYCH toolbox (see below) |
| SmartGit (see below) |
| TDT v88 drivers and software (if using Synapse or only doing behavior, or using OpenEx without Spike Pac).  Use v80 drivers if using OpenEx with Spike Pac macros.  -RPVdsEx/TDT Drivers  -Active X controls  -Open Developer (only if doing ephys)  -Synapse Suite or OpenEx (optional, only if doing ephys)  -TtankMin |



\*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. **To install EPSYCH toolbox on a new computer:**

1. Log onto <https://github.com>
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. \*Note: MATLAB must be run in administrator mode. Either ensure that you’re running MATLAB while logged into an administrator account on the computer, or create a shortcut to MATLAB on the computer desktop, and configure the shortcut to always run in administrator mode by editing the shortcut properties (accessed by right clicking the shortcut and navigating to the shortcut tab).
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.

* 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). Also, for conversion purposes, note that 1 Pa = 94 dB SPL.
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.
14. DESIGNING EXPERIMENTS

**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 if desired. Any name will do.
6. Select desired RPVds circuit file.
7. A list of editable parameters will appear. These parameters correspond to parameter tags in the RPVds circuit.

Remove any parameters that you don’t want to rove systematically during the experiment by highlighting them and pressing the Remove Parameter button. Notes about parameter tags and designing RPVds circuits: If you do not wish a particular parameter tag to display in the list of editable tags, and do not wish the value to be saved each trial, preface the tag in the RPVds circuit with a tilde (~). **Avoid using parameter tag names that begin with the letter “z.” Many proprietary TDT and OpenEx tags being with “z,” and the epsych code ignores them.** Also, note that parameter tags are **case insensitive.**

You can also choose to have a parameter tag show up in the list, but not be updated during runtime by adding an asterisk (\*) prefix. Additionally, you can add a dollar sign ($) prefix to the parameter tag to prompt for values when beginning an experiment. These features are part of the native EPSYCH toolbox, but have not been tested with the SanesLab GUIs. Use with caution.

Known supported “hidden” parameter tags include:

~TrialDelivery (allows user to start or pause trial delivery dynamically

~InTrial\_TTL (allows MATLAB to display when we’re in a trial during runtime)

~Spout\_TTL (allows MATLAB to display when the spout TTL is high during runtime)

~RespWin\_TTL (allows MATLAB to display when the Response Window is open during runtime)

~Poke\_TTL (allows MATLAB to display when the nose poke TTL is high during runtime)

~Water\_TTL (allows MATLAB to display when water is being delivered during runtime)

~Light\_TTL (allows MATLAB to display when house light is off during runtime)

~buffer (microphone buffer)

~BuffTrig (trigger for microphone buffer)

~bufferSize (size of microphone buffer)

All of the above parameter tags except ~TrialDelivery are also functional without the tilde, as found in older circuits. In those cases, remove these parameter tags from the list of editable parameters. These tags are used for displaying data and for capturing sound from the microphone only, and should not be given values.

Finally, there are two REQUIRED PARAMETERS. Keep these, no matter what you wish to rove:

Reminder, 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.

AMdepth: Depth of amplitude modulation (proportion)

AMrate: Rate of amplitude modulation in Hz

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.

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

FMdepth: Depth of frequency modulation (proportion)

FMrate: Rate of frequency modulation in Hz

Freq: Sound frequency in Hz

Highpass: Highpass filter cutoff for noise stimulus in Hz

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. (msec)

Lowpass: Lowpass filter cutoff for noise stimulus in Hz

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

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

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

RespLatency: Time from trial onset to animal’s behavioral response (msec). Set this parameter to Read or remove it from the list. When an animal fails to make a behavioral choice before the response window has elapsed, this value is set to 0.

RespWinDelay: Length of time between trial onset and the opening of the response window (msec).

RespWinDur: Length of time the response window is open (msec). When open, animal is allowed to make a behavioral response.

reward\_dur: length of time (msec) pump will run for each reward. This value will be automatically calculated by the MATLAB GUI based on the desired pump rate and reward volume, so it can be removed unless you specifically want to rove.

ShockFlag: Determines if shock delivery is allowed (1) or not allowed (0)

ShockDur: Length of time (msec) for shock delivery

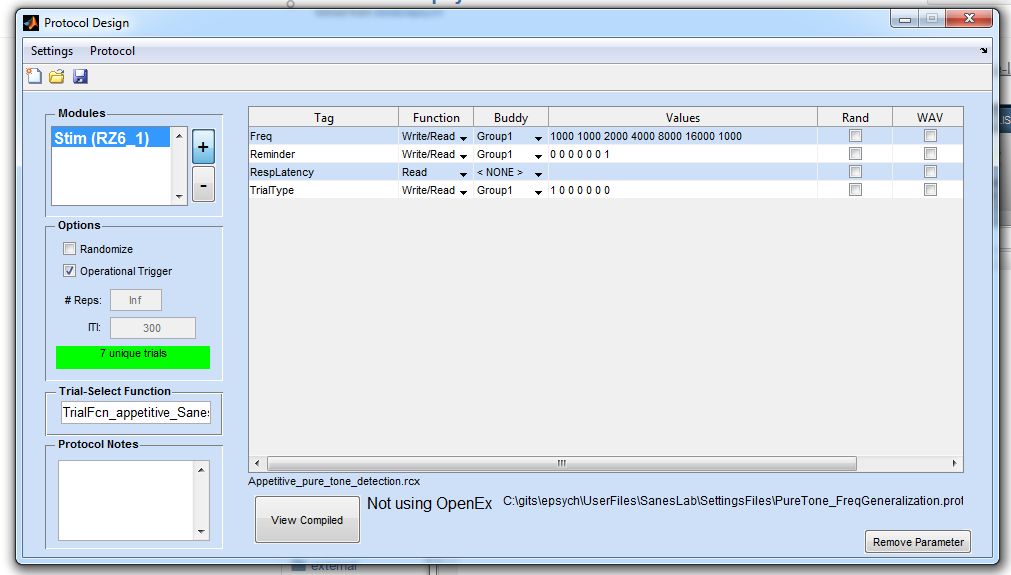
Silent\_delay: Length of time between trial onset and sound onset (msec)

Stim\_Duration: Length of sound stimulus (msec)

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.

TrialType: 0 (GO); 1(NOGO) \*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 paradigms.
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.
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. \*Note: uncheck this if running any paradigm that does not require the animal to initiate its own trials, such as shuttlebox paradigm, startle response paradigm, etc.
3. Set the number of reps to Inf
4. Enter TrialFcn\_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.
5. 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 if you wish. Any name will do.
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 7-17 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. If the protocol requires manual control over sound level or frequency (such as during basic characterization, for instance), you’ll want to load in one or more sound calibration file, so you can access voltage correction values. You can include relevant code for this, and other tasks, in a custom trial function. If you’ve already written this function, enter the name of the function 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.
9. 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 2000 trials are generated.
10. If everything looks good, save your protocol.
11. RUNNING EXPERIMENTS
12. **Running a behavior only experiment**
13. Click on Run Behavior Experiment on the EPSYCH launchpad menu.
14. Click on Function Definitions on the top menu bar and select Timer Functions
15. In the Start Timer Function box, enter ep\_TimerFcn\_Start\_SanesLab\_v2
16. In the RunTime Timer Function box, enter ep\_TimerFcn\_RunTime\_SanesLab\_v2
17. In the Stop Timer Function box, enter ep\_TimerFcn\_Stop
18. In the Error Timer Function box, enter ep\_TimerFcn\_Error. Click OK.
19. Click on Function Definitions on the top menu bar and select Saving Function
20. Enter ep\_SaveDataFcn\_SanesLab. Click OK.
21. Click on Function Definitions on the top menu bar and select Box Figure
22. Enter Appetitive\_detection\_GUI\_v2 or Aversive\_detection\_GUI Click OK. The specific GUI you choose must be compatible with the RPVds circuit and protocol.
23. Click on Function Definitions on the top menu bar and select Add Subject
24. Enter ep\_AddSubject\_SanesLab. Click OK. \*Note: Function definitions will stay set across consecutive runs. A quick way to load in the correct GUI and timer functions is to save a configuration file (see below).
25. Click + Subject.
26. 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.
27. Select the sex, condition, and age of animal from the dropdown menu.
28. Add any notes about the experiment as desired. Click Done.
29. Select your desired protocol. (e.g. filename.prot) and click Open.
30. If desired, double check the trial structure by clicking View Trials.
31. 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, box figure, timer functions, 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.
32. Click Run to launch the GUI
33. Select the appropriate calibration file if prompted (e.g. ceiling\_speaker.cal).
34. 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.
35. To stop the experiment, press the red Stop button in the Run Experiment Dialogue Box.
36. Save the file when prompted.

\*Note: During runtime, data is automatically backed up to C:\Users\sanesadmin\Documents\MATLAB\DATA. See line 63-82 of ep\_TimerFcn\_Start\_SanesLab\_v2.m

1. Close the GUI window.
2. **Running a combined behavior-ephys experiment using OpenEx**
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. Also note that SpikePac is not compatible with later versions of the TDT drivers (v86+ are known to be incompatible. v80 is definitely compatible).
   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. – **“Physiology.rcx”** \*Note: Make sure that you are acquiring the “RealSignal” and not the “TestSignal” in the circuit.
   4. Name the device. Any name will do. Click OK.
   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. \*Note: This circuit MUST be compatible with the protocol file you wish to run.
   4. Name the device. Any name will do. Click OK.
   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/SanesLabMembers/MLC/Ephys/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.
   4. Select “Yes” when asked if you want to discard the current setup.
8. If desired, 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.**), and verify that it controls the same circuit you loaded in for the RZ6 in step 11c, above. 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.
12. 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.
13. Continue with steps 20-25 of **Section IV.A.**
14. **Running an ephys only experiment using OpenEx**
15. Follow steps 1-17 in **Section IV.B.**
16. With OpenProject still open, launch MATLAB click on Run Electrophys Experiment on the EPSYCH launchpad menu.
17. Click on the Locate Protocol Dir…button in the Protocol Setup section of the window: C:\gits\epsych\UserFiles\SanesLab\SettingsFiles\General\Protocols\BasicCharacterization
18. Click Select Folder
19. A list of Electrophysiological protocols will be displayed. Re-order protocols as desired by using the up and down arrows on the right. The protocol at the top of the list is the first one to be presented.
20. Click Select Tank. 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. Server and Tank details should appear in the box to the right.
21. 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.
22. 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.
23. **Running a combined behavior-ephys experiment using Synapse**
24. Launch Synapse by clicking on desktop shortcut
25. If Synapse has already been configured on your machine, choose the appropriate User, Experiment and Subject, and then skip to step 11 below. Otherwise, if this is the first time launching Synapse, configure the rig by clicking on Menu at the top left of the screen, then selecting Edit Rig. Click Detect to prompt Synapse to automatically detect all TDT hardware. Ensure that all hardware was properly identified and selected (indicated by a checkbox next to the module name).
26. Configure each module as needed. While the specifics will differ according to the hardware set-up for each rig, the following is what the SanesLab currently uses. After configuring the hardware, click OK.

PZ5: 64 Analog Channnels, 0 Digital Boards

RS4: Direct communication. Set IP address to address found on the status tab of the touchsceen interface on the RS4. Note that the RS4 should have already been configured prior to this step.

RZ6: Right click and select Legacy Mode. Legacy mode allows a complex RPVds circuit to be run on a module without first being converted into a Synapse “Gizmo”.

1. The identified modules should now be listed in the processing tree. Additional configuration parameters (sampling rates, number of channels, etc.) should be set now by clicking on each module and adjusting the settings in the Options section on the right hand side of the window. Press Commit after you’ve made all of your desired changes for each device. Again, while the specific will depend on your hardware setup, here’s what the SanesLab uses. Items not specifically mentioned here were left in their default state. The following settings assume that the user is planning to acquire single or multi-unit physiology using single ended (non-differential) recording.

RZ2: Under the “Main” tab, set the Master Device Rate by unchecking the Auto box, and selecting 25K from the drop down menu. Enable Automatic Load Optimization by checking the Auto box on the second row. Disable the Tick Store.

RS4: Uncheck the Max box and set the sampling rate to 24414 Hz using the slider. Set the Scaling parameter to unity. Check the box next to Send SEV Rename Packets.This last step ensures that the data streamed to the RS4 will have the same name as the data in the tanks.

PZ5: Uncheck the Use Sub Amps box and check the Enable box. Set the Type to Analog. Set the Input Channels to 64. Under the Options section at the bottom of the screen, set the Sampling Rate to System Rate. Uncheck the DC Coupled box. Set the Reference Mode to None for single ended recording. Set the Set to base type option to Single Unit.

RZ6: Select the RPVds circuit to run on the device in legacy mode by clicking the component icon next to the File Name box on the right. \*Note: This circuit must be identical to the circuit identified in the MATLAB protocol you wish to run.

1. Once all of your changes to the hardware have been committed, click on Menu at the top left of the screen, then select Preferences. Under the General tab, set User Login to Required. Then check the Experiments and Subject boxes to make them Private per user. This step allows users to have privately owned experiments and subjects. If you wish for even greater security, you can set the User Login to Required with password.
2. Still under the General tab of the Preferences Menu, check the boxes to enable Standby Mode and Synapse Server. Standby Mode must be enabled for epsych to work properly. The Synapse Server must be enabled to access Synapse API controls via MATLAB.
3. Under the Data Saving Tab, set the desired path for data tank storage, and set up your data saving heirarchy as you wish.
4. Now that the rig is configured, either create a new user by checking the user icon at the top left of the screen and selecting New or select an existing user.
5. Create an experiment by highlighting a desired module, and double clicking or dragging a chosen Gizmo. You will only be able to add Gizmos to modules that are not running in Legacy Mode. For a better understanding of Gizmos and Synapse see the TDT Synapse Manual (referenced at the end of this document). Once you are satisfied with your Experimental Design, save it by clicking on the Experiment icon on the top left of the screen, and clicking Save As. *\*Note: You can access saved experiments in the future by simply clicking on the Experiment icon and selecting the desired one.*
6. Choose an experimental subject by checking the Subject icon on the left of the screen and either selecting New to create a new subject, or simply select an existing subject.
7. Once Synapse is configured, launch MATLAB and click on Run Behavior Experiment on the EPSYCH launchpad menu.
8. Follow all of the instructions in **Section IV. A**. \*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.**), and verify that it controls the same circuit you loaded in for the RZ6 in step 4, above. **All combined ephys/behavior experiments must use protocols designed for use with OpenEx, even if you’re using Synapse instead of OpenEx.**  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.
9. **Running an ephys-only experiment using Synapse**
10. Follow steps 1-10 in **Section IV.D.**
11. Once Synapse is configured, launch MATLAB click on Run Electrophys Experiment on the EPSYCH launchpad menu.
12. Click on the Locate Protocol Dir…button in the Protocol Setup section of the window: C:\gits\epsych\UserFiles\SanesLab\SettingsFiles\General\Protocols\BasicCharacterization
13. Click Select Folder
14. A list of Electrophysiological protocols will be displayed. Re-order protocols as desired by using the up and down arrows on the right. The protocol at the top of the list is the first one to be presented.
15. 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.
16. 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.
17. Extracting AND ANALYZING data
18. **Behavior data**
19. [Need info here- KP? JY?]
20. **Ephys data (TDT TANKS)**
21. [Need info here- KP? JY?]
22. 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>

* 1. **Logitech HD Webcam C270**

Specs and software download: [http://support.logitech.com/en\_gb/product/hd-webcam-c270#download](http://support.logitech.com/en_gb/product/hd-webcam-c270%23download)

* 1. **TDT software**

You can download drivers here: <http://www.tdt.com/downloads.html>

You may also need to update the microcode on your hardware for the latest versions of the programs. Instructions may be found here. <http://www.tdt.com/files/fastfacts/Microcode.pdf>

OpenEx Documentation: <http://www.tdt.com/user-manuals.html>

Synapse Documentation: <http://www.tdt.com/files/manuals/SynapseManual.pdf>

Active X Documentation: <http://www.tdt.com/files/manuals/ActiveX_User_Reference.pdf>

Open Developer Documentation: <http://www.tdt.com/files/manuals/OpenDeveloper_Manual.pdf>

Synapse API Documentation: <http://www.tdt.com/files/manuals/SynapseAPIManual.pdf>

* 1. **Matlab setup notes**

1. Startup file

%% Load paths for EPsych

addpath('C:\gits\epsych');

epsych\_startup('C:\gits\epsych');

Save as “startup.m” in main \Documents\Matlab\ folder

1. Shortcuts

**e**: *ep\_LaunchPad*

**A**: *Appetitive\_Training\_Menu* (for early spout training)

**NOTES:**

In the appetitive GO/NOGO paradigm:

* 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

In the aversive GO/NOGO paradigm:

* The paradigm will always start with a NOGO trial
* The shock will only be delivered if the animal missed the GO. (The animal will never receive a shock for getting back on the spout too early after a hit).