# AUDITORY RECOGNITION TRAINING SYSTEM (ARTSy) MANUAL

Table of Contents

## HARDWARE

The hardware components are based on the minimum requirements necessary to set up and control four training booths running the go–nogo behavioural paradigm. Additional hardware components for the other behavioural paradigms in the system are also recommended. The system requires three major hardware units: (1) computer (Table S1); (2) control interface (Table S2); (3) booth hardware (Table S4).

### *Computer*

**Table S1**

Components required for setting up and controlling four training booths

| Item | Vendor | Specifics | Quantity |
|---|---|---|---|
| Computer*† | | At least 3 PCI/PCI-E slots | 1 |
| Digital I/O card (PCI 6503) | National Instruments | 24 input/outputs, No. 777690-01 | 1 |
| Sound card | Newegg.com | M-Audio, 4-channel audio with break-out box, No. N82E16829121009 | 1 |
| Capture card | Contact GeoVision for local vendors | GeoVision GV-800 video capture card, 4 video/audio channels | 1 |
| Amplifier | BHphotovideo.com | Behringer miniamp compact 4-channel headphone, No. AMP800, | 2 (1/channel) |

Install hardware following product instructions.
* Minimum system requirements: Intel Core 2 Quad, 2.40 GHz processor; 4 GB RAM; 160 GB HDD; NVIDIA GeForce 8800 PCI-E VGA.
† OS: 32-bit Windows XP/Windows Vista/Windows 7.

*Control Interface*

**Table S2**
Control interface for integrating the hardware with the software

| Item | Vendor | Specifics | Quantity |
|---|---|---|---|
| Connector block | National Instruments | Screw terminal, CB50LP, No. 777101-01 | 1 |
| Unshielded NB1 ribbon cable | National Instruments | 1 meter, No. 180524-10; available in different sizes | 1 |
| Relay board | Simplecircuitboards.com | 10 A 4-relay TTL-driven relay board | 2 |
| Voltage Converter | Simplecircuitboards.com | 1-input to 1-output DC-to-DC converter (12 V to 5 V) | 4 |

*Assembly*

(1) Screw the connector block and relays to a firm board (our panel is made from 0.64 cm thick acrylic and measures 30.48 x 30.48 cm). Leave enough space between the connector block and relays so that wires can easily be manipulated.

(2) The National Instruments connector block screw terminal (Fig. S1) has two rows of screw pins numbered 1 through 50. The top row has even numbers and the bottom row has odd. Even numbered pins are used for grounding. Ground wires in Fig. S1 are green. Odd pins 1 to 31 are used for inputs, and odd pins 33 to 47 are used for outputs. Input/output wires in Fig. S1 are red. The connector block has a plug (Fig. S1, A) for the ribbon cable. Table S3 describes how inputs and outputs are assigned in the system.

(3) Lines from the output pins connect to relay boards to control the feeders and lights, which require an external power source (power specifications are supplied with the product information sheets of the hardware components in Tables S2, S4 and S5). When relays are activated by a 5 V TTL signal from the computer, they allow the effector devices to operate with external power sources. Odd pins 33, 35, 37 and 39 correspond to feeders 1, 2, 3 and 4, respectively. Even pins 34, 36, 38 and 40 are ground. First screw the ground wires into pins 34, 36, 38 and 40. Next, screw the output wires into pins 33, 35, 37 and 39. Now connect the wires to relay board 1 as shown in Fig. S1, B. Screw output wires from pins 33, 35, 37 and 39 into screws 1, 2, 3 and 4, respectively, on relay board 1. Twist the free ends of ground wires from pins 34, 36, 38 and 40 together and insert into screw G on relay board 1. Now connect the lights to the second relay board. Odd pins 41, 43, 45 and 47 correspond to lights 1, 2, 3 and 4, respectively. Even pins 42, 44, 46 and 48 are ground. Screw the output wires into pins 41, 43, 45 and 47 and the ground wires into pins 42, 44, 46 and 48. Connect the output and ground wires to relay board 2 following the same procedure used to connect the wires to relay board 1.

(4) The relay boards require an external power source. The power source connects to the relay board by screws 6 and 7 (Fig. S1, C). Screw 6 is positive and screw 7 is negative.

(5) Each feeder and light has two lines (one power and one to its booth) connected through the relays (Fig. S1, D, E). For instance, to set up feeder 1, first connect the power line to its power source and determine the polarity of each wire. Insert the positive wire into screw 8 (normally open contact). Twist the negative wire together with a booth line wire and insert into screw 10 (normally closed contact). Insert the other booth line wire into screw 9 (common contact). Repeat this procedure for the following three feeders and for the lights on the second relay board.

(6) Each input pin is connected to one sensor in each booth. Figure S1 (F) shows how to connect an input line to the connector block. The ground (green) wire is screwed into pin 2 and the input wire is screwed into pin 1. Refer to Table S3 for input line and booth assignment. Note: each sensor provides a 12 V signal when the beam is unbroken, and a 0 V signal when the beam is broken. The NI card is rated for signals less than 5 V. We therefore connect a voltage regulator between the sensor and the NI breakout box to drop the sensor signal within the allowable range.

(7) Connect the ribbon cable to the connector block plug and the other end to the digital I/O card.

**Table S3**

Input lines and booth assignments

| NI pin | Line | Source | Booth |
|---|---|---|---|
| 1 | Input | Sensor | 1 |
| 3 | Input | Sensor | 2 |
| 5 | Input | Sensor | 3 |
| 7 | Input | Sensor | 4 |
| 9 | Input | Sensor | 1 |
| 11 | Input | Sensor | 2 |
| 13 | Input | Sensor | 3 |
| 15 | Input | Sensor | 4 |
| 17 | Input | Sensor | 1 |
| 19 | Input | Sensor | 2 |
| 21 | Input | Sensor | 3 |
| 23 | Input | Sensor | 4 |
| 25 | Input | Sensor | 1 |
| 27 | Input | Sensor | 2 |
| 29 | Input | Sensor | 3 |
| 31 | Input | Sensor | 4 |
| 33 | Output | Feeder | 1 |
| 35 | Output | Feeder | 2 |
| 37 | Output | Feeder | 3 |
| 39 | Output | Feeder | 4 |
| 41 | Output | Lights | 1 |
| 43 | Output | Lights | 2 |
| 45 | Output | Lights | 3 |
| 47 | Output | Lights | 4 |

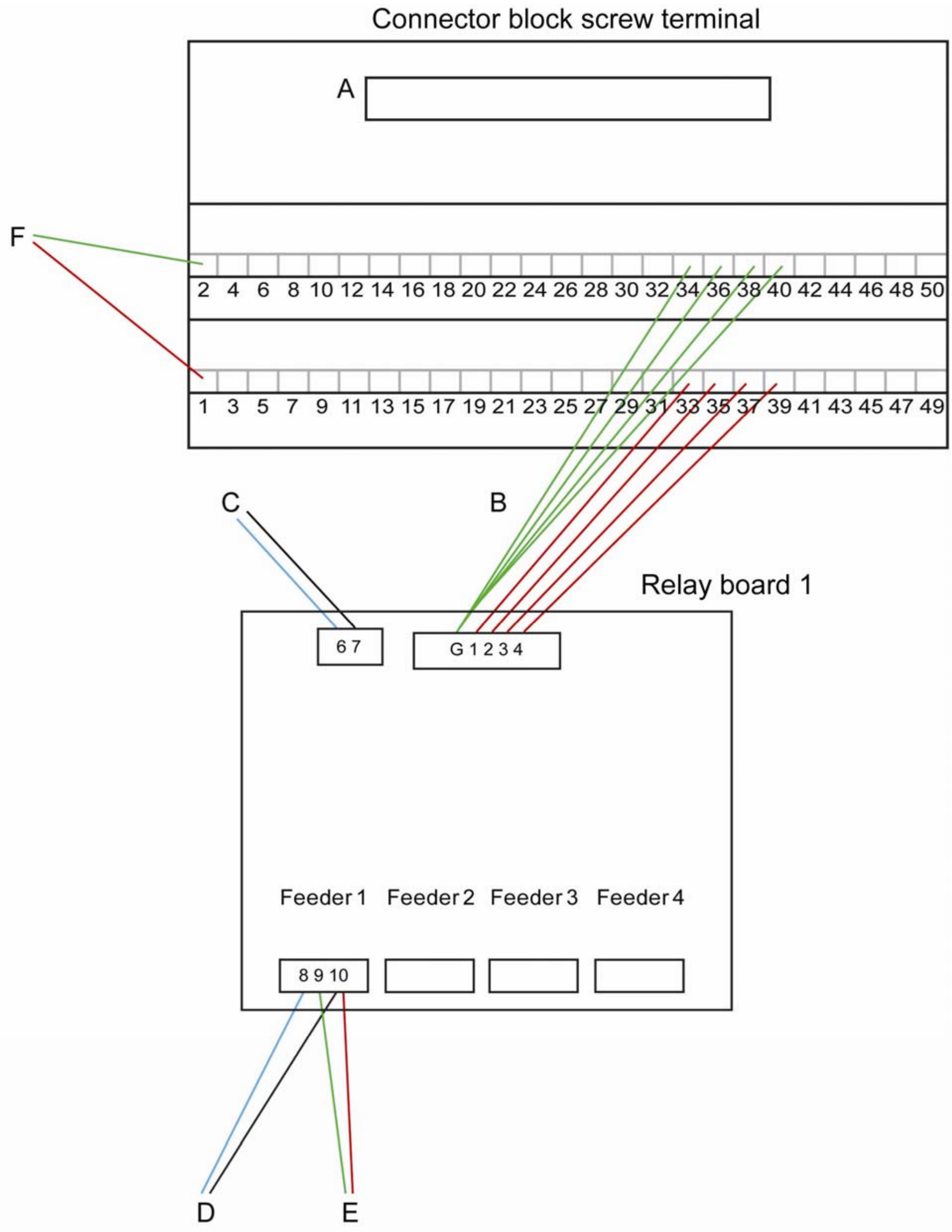


**Figure S1.** Schematic of control interface connections: (A) ribbon cable plug; (B) wires joining the connector block with relay board 1; (C) wires to relay board 1 power source; (D) wires to feeder 1 power source; (E) wires to feeder 1 in booth 1; (F) wires to sensor 1 in booth 1.

*Booths*

**Table S4**
Booth components

| Item | Vendor | Specifics | Quantity |
|---|---|---|---|
| Sound isolation cubicle | Coulbourn Instruments | No. H10-24A | 4 |
| Slot sensor | Banner Photoelectric Sensors | No. SLO30VB6Y | 4* |
| Emitter and detector† | Banner Photoelectric Sensors | Q08 series | 4 pairs |
| Light | JCWhitney.com | No. 13707G | 4 |
| Speaker | Kenwood | No. KFC-1377 | 4 |
| Camera with microphone | BHphotovideo.com | Bolide, No. KPC600 | 4 |
| Response panel | Custom | Acrylic, 30.48 x 30.48 cm | 4 |
| Wire cage | Custom | 2.54 x 1.27 cm mesh wire; cage dimensions: 43.18 x 30.48 x 30.48 cm | 4 |
| Feeder | Custom | See Table S5 | 4 |

* For go–nogo. Purchase eight to set up two alternative forced choice or operant preference tests.
† Required for two-perch preference tests.

*Assembly (final set-up shown in Fig. S2)*

(1) Cut five pieces of mesh wire to make the training cage with the following dimensions: one 35.56 x 35.56 cm piece (back), two 35.56 x 43.18 cm pieces (sides) and two 30.48 x 43.18 cm pieces (top and bottom). We find that large cages keep subjects hearty, making space for the separation of training and living activities.

(2) In the centre of the 30.48 cm side of the floor panel that will be under the response panel, cut a 2.54 x 5.08 cm space for the feeder.

(3) Cut a window, in the centre of the side piece that will be accessible when the cage is in the booth, to make a door, and a piece of mesh wire with the dimensions of the window.

(4) Sand all sharp edges on the mesh wire.

(5) Attach mesh wire pieces with wire or nylon wire ties. The floor piece should be attached to the side pieces 5.08 cm from the bottom.

(6) Cut a 30.48 x 30.48 cm piece of 0.64 cm thick acrylic to make the response panel. Drill three holes (two to mount and one to pull the wire through) for the slot sensor (for TAFC, mount sensors side by side; for preference tests, mount sensors on opposite sides of the response panel; you may want raise the sensors so that they are accessible from perches); the

mounted sensor should be 2.54 cm from the wire mesh floor. We mount our sensor on the left side of the response panel. Drill a hole in each corner of the response panel to mount it to the cage. Attach the response panel to the cage with twist ties so that it can easily be removed for maintenance.

(7) The feeder should be placed behind the response panel (fix to booth floor with tape or sticky tac) so that the lip is invisible when it is inactivated and fills the hole in the mesh floor when it is deployed.

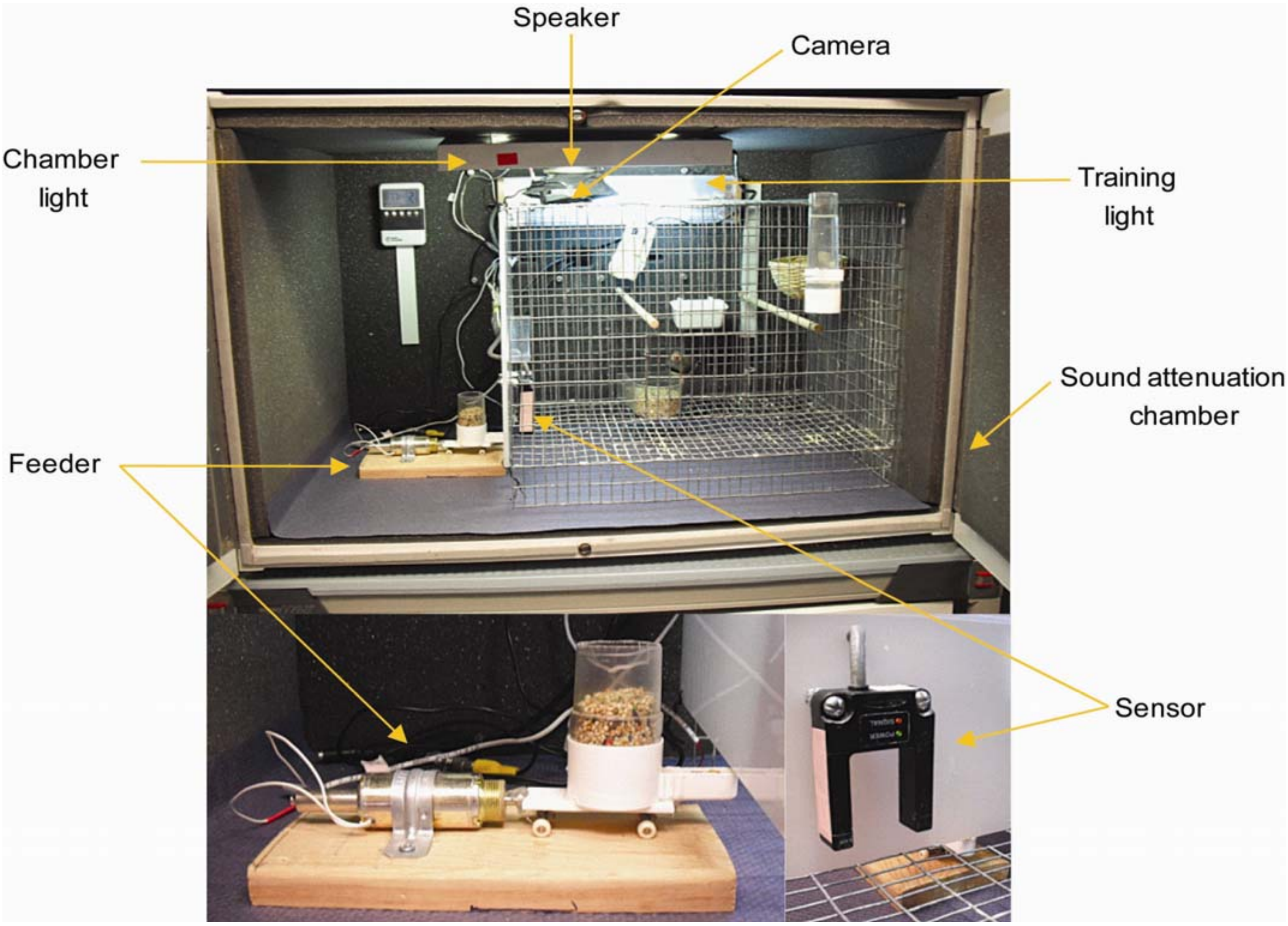


**Figure S2.** Set-up for booth assembly.

**Table S5**

Components for the feeder mechanism

| Item | Vendor | Specifics | Quantity |
|---|---|---|---|
| Solenoid | Allied Electronics | Guardian, push-type No. TP8X16-C-12D | 4 |
| Pipe clip | | For 1.91 cm diameter pipe | 4 |
| Water tube | ABBA Products Corp. | No. C005 | 4 |
| Cardboard | | | |
| Wheels | Techdeck.com | Small toy skateboards with detachable axel | 4 |
| Wood | | 17.78 x 4.45 x 10.16 cm* | 4 blocks |
| Compression Spring | McMaster Carr | 5/16” OD, 0.023” wire diameter, No. 1986K8 | 4 springs |
| Acrylic board | | 6.35 x 2.54 cm | 4 boards |

*Assembly*

(1) Make the base by cutting a 17.78 x 4.45 x 10.16 cm wood block. Mount solenoid on top with pipe clip 6.35 cm from the end (Fig. S3b).

(2) Cut the spring to a length of approximately 13 mm. Place the spring on the solenoid plunger (Fig. S3a).

(3) To assemble the seed holder, first cut a 6.35 x 2.54 cm acrylic board. Unscrew the two axels from the Techdeck. On one side of the board, attach each axel equidistantly at opposite ends with acrylic glue (Fig. S3c). The Techdeck can be used in its entirety if the curving ends are removed. Separate the water tube top from the bottom. Mount the water tube bottom to the acrylic base with acrylic glue, so that the lip protrudes over the edge of the board at the end (Fig. S3d, f). Next, saw 0.83 cm off the top of the clear water tube. This makes it easy to refill the feeder with seed. Next, cut a circular piece of cardboard that fits inside the base of the water tube. Place it on a slight angle in the base (this, in combination with the force of the solenoid, ensures that seed falls into the lip, replenishing eaten seeds during training), and then insert the top (Fig. S3f). As an alternative to mounting the feeder on the acrylic board or Techdeck, the feeder can be mounted on a mechanical slide. It consists of 2 pieces: a guide block (McMaster Carr; part no. 9829K2) and associated rail (McMaster Carr; part no. 9829K122; 120 mm length).

(4) Attach the plain end of the board to the inserted solenoid plunger (Fig. S3e). This can be achieved by several methods, but it is important that the attachment between the plunger and feeder remain plastic so the spring can be replaced. Typically, we either drill a hole in the plunger and the board and attach them with a piece of wire, or attach the two pieces with hot glue (it is stable while the feeder is in use and is also easy to peel off).

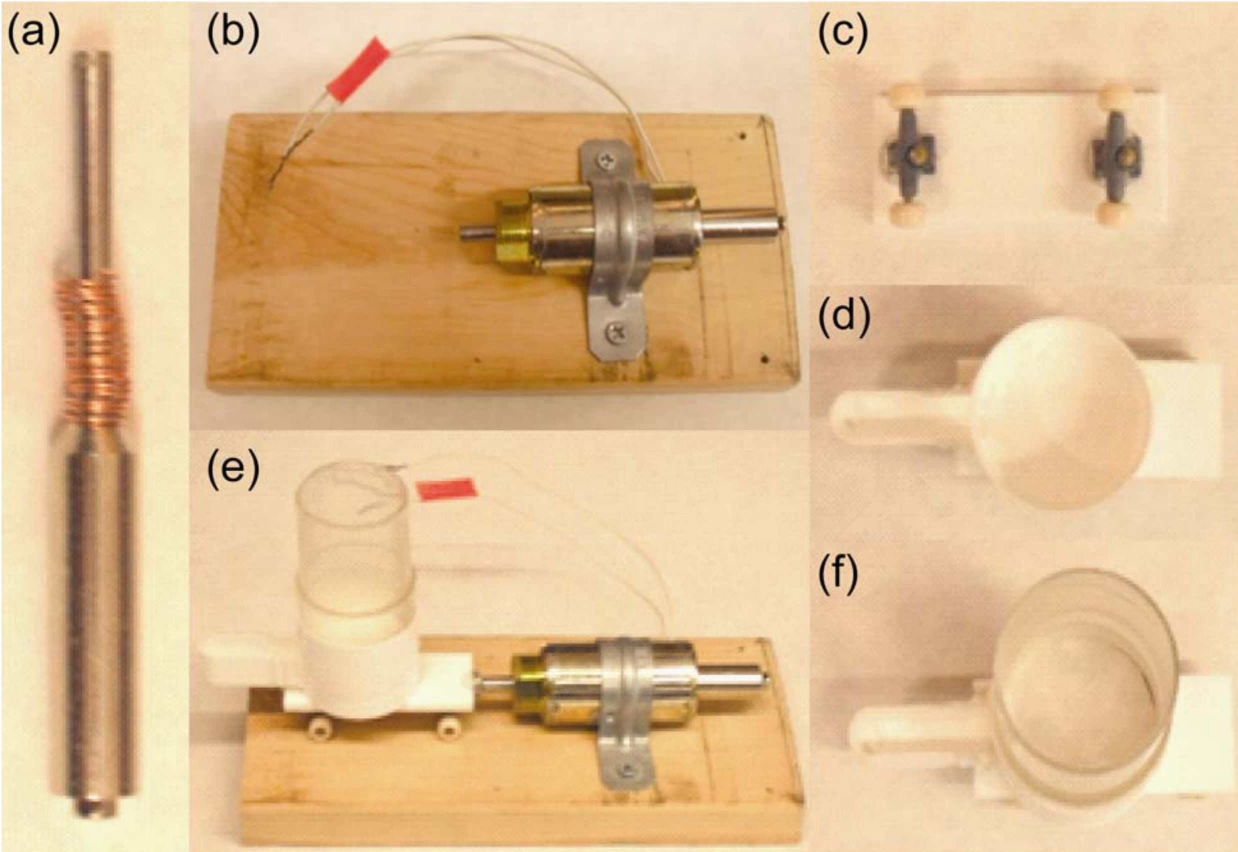


**Figure S3.** Feeder assembly: (a) spring on plunger; (b) solenoid mounted on wood base; (c) axels mounted to acrylic base; (d) bottom of water tube mounted to acrylic base; (e) final feeder apparatus; (f) water tube top.

*Modification Options*

All the hardware components (e.g. relays) we use can be purchased with more units to incorporate more channels/booths. National Instruments makes a variety of Digital I/O cards with more input/output pins. The program code needs to be modified to incorporate a larger NI card. Additionally, input and output channels can be connected to any detector (e.g. microswitch keys) and effector in the same manner as the sensors, feeders and lights. (Note: not all hardware will require additional power sources; e.g. microswitches can be powered by the computer, 5 V TTL.)

*Cost Estimates*

Estimated cost of all hardware (including the computer) for four fully functional booths will be around $11,500 (U.S.). The sound isolation cubicles and computer contribute most of the cost. Using less expensive cubicles and computer will reduce costs to around $5,000 (U.S.).

## SOFTWARE

The software bundle, ARTSy, can be downloaded as a single package that runs in the Matlab environment (Table S6). The software package monitors the infrared sensors in up to four booths simultaneously, and controls feeder, light and sound presentation. The software also provides user control of timing, rewards, punishments and acoustic stimuli.

Matlab is a software platform that is distributed by The Mathworks, Inc. Purchase Matlab through your institution or through the Mathworks website, and install it on any computer that will be used to control behavioural training. The Matlab code is all open source and can be modified for user-specific applications.

**Table S6**
Software

| Item | Vendor | Specifics | Quantity |
|---|---|---|---|
| ARTSy | Available at www.commneuro.psych.columbia.edu | | 1 |
| Matlab | Mathworks.com | V7.6 or higher | 1 |

*Download ARTSy*

ARTSy can be downloaded as a single package from www.commneuro.psych.columbia.edu. Visit the website to ask for permission and a password.

*Installation*

Once the software is downloaded, unzip the file and save the components in a directory where they will not be inadvertently modified or deleted. For example, you may want to save the

programs in a Matlab directory such as:
C:\Programs\Matlab\Work\ARTSy

Open Matlab by double-clicking the Matlab icon (Matlab must be 'run as administrator' in Windows 7). Once Matlab has started, select 'File->Set Path'. Click the 'Add with Subfolders…' button and select the ARTSy directory from the location under which it was saved. Click 'Open' to select the folder. Once back in the 'Set Path' graphical user interface (GUI), click 'Save' to save the changes and then click 'Close' to exit. Once folders have been included in Matlab's path, files within these folders are always accessible.

*Customizing the Matlab Code*

Although the Matlab code is compatible with nearly any PC (see specifications), the code must be modified to suit every set-up. There are two major changes that need to be made to the code: (1) changes to the directories in which Matlab expects to find files such as wave files and saved data; and (2) changes to the Matlab pointer that chooses the audio device through which to play sounds.

*Changing Directories*

Open Matlab and open the file titled 'InitializeProtocol.m' by typing 'edit InitializeProtocol' in the command window. This will open up the source code used to initialize all of the behavioural protocols. At the top of the file is a list of directories that must be adjusted for every computer.

*Directories*

*handles.waveDir*. Defines the directory that contains the .wav files that can be accessed during behavioural training and testing.

*handles.boothStatusDir*. Defines the directory that contains the files booth1Status.mat, input1Status.mat, booth2Status.mat, etc. The boothStatus files each contain the status of the light and the feeder in a particular booth, and the inputStatus files contain the sensor status in a particular booth. These files are all located within the unzipped 'MonitorNICard' folder, and handles.boothStatusDir must point to this directory.

*handles.saveDir*. The directory to which saved data will be stored.

*handles.settingsDir*. The directory to which saved settings will be stored.

Save ‘InitializeProtocol.m’ and close the file. Next, open ‘MonitorNICard.m’. This program has a separate entry for handles.boothStatusDir, on line 32 of the code. Change this to point to the same location as in ‘InitializeProtocol.m’. Save ‘MonitorNICard.m’ and close.

*Setting the Audio Devices*

Along with the sound card that is installed with this system (see Table S1), most computers also have an internal soundcard. Each sound card typically has more than one audio channel, such that there are many potential audio channels through which Matlab can send sounds. Users must modify the Matlab code such that the sound files are played through the appropriate audio channels.

To determine which Device Numbers to use, open Matlab and type ‘pawavplayx’ in the command window. The command window will show a list of audio devices. Find the Device Numbers that correspond with Delta 44 sound card. See below for an example in which the appropriate Device Numbers are 2 and 3.

Device 2
Device name: Line 1/2 (M-Audio Delta 44)
Max Input channels: 0 Max Output channels: 2
Native Sample Format: int32

Device 3
Device name: Line 3/4 (M-Audio Delta 44)
Max Input channels: 0 Max Output channels: 2
Native Sample Format: int32

Note: the Delta 44 M-Audio card will typically show up as four devices. Two of the devices will have 0 Max Input Channels and 2 Max Output Channels, and the other two devices will have 2 Max Input Channels and 0 Max Output Channels. Choose the Devices that have 2 output channels.

In the Matlab command window, type ‘edit InitializeBooth’. This will open the Matlab source code that initializes all of the booth-specific information. At the top of the code is a section for editing the ‘audio output device and stereo channel’. For each booth, you will need to modify handles.speaker.

*handles.speaker.* Defines the audio output device. For Booth 1 and Booth 2, enter the Device Number associated with Line 1/2 of the Delta 44 (Device 2 for the example above). For Booth 3 and Booth 4, enter the Device Number associated with Line 3/4 of the Delta 44 (Device 3 for the example above.

Although two booths are associated with each audio device, sounds will only be played from one speaker at a time. This is controlled by handles.channelGain. Booths 1 and 3 will play through the right channel (i.e. handles.channelGain = [1 0]), and booths 2 and 4 will play through the left channel (i.e. handles.channelGain = [0 1]). You do not normally need to modify this parameter.

*Creating Matlab Shortcuts*

Once the Matlab software has been edited, we find it helpful to create Matlab shortcuts for starting each of the primary programs. Matlab shortcuts are located above the command window in the Matlab screen.

To add a shortcut, right click in the open area to the right of the word ‘Shortcut’, and select ‘New Shortcut’. First, add a shortcut that will start the Matlab program for monitoring the NI card. In both the label field and the callback field, type ‘MonitorNICard’. Next, make a second shortcut with the label and callback fields entered as ‘Conditioning’.

Matlab comes loaded with some predefined shortcuts. If you would like to remove these shortcuts, right click on the desired shortcut and press delete.

## BEHAVIOURAL PROCEDURES

All behavioural procedures involve variations on the basic training stages that we describe in detail for go-nogo (GNG) and are built into the program. The program includes Acclimation, Shaping, GNG, Classical, Two Alternative Forced Choice and Preference behavioural paradigms.

*Go-Nogo (GNG)*

The following is a protocol for the behavioural procedures and program parameters used to train our subjects on the GNG task.

*Phase 1: acclimation*

At least 1 week prior to training, house subjects individually in training booths. This familiarizes subjects with the apparatus. Subjects should remain in isolation booths for the duration of the experiment(s).

*Phase 2: shaping*

There are three stages to shaping. Each stage is repeated over consecutive days until the subject masters the focal task. Fast subjects at least 3 h before starting each stage. Our birds perform optimally when the food is removed just prior to the dark cycle (2100 hours) and shaping begins immediately at the start of the light cycle (0700 hours).

*Stage 1: introducing the stationary feeder*. Fill the feeder with seed and position it so that the lip protrudes into the cage and the bird can eat from it. The bird should begin to eat regularly from the feeder after 3 to 5 h. Make sure to tap the feeder periodically so food falls into the lip (the feeder will not refill automatically when it is motionless). After the bird appears comfortable eating out of the feeder, the session is over. Give the bird its regular food dish at the end of the session. The bird should master this stage in one session. Hint: it helps to keep the regular food cup on the floor in front of the feeder; this prompts subjects to look for food in that location when the food dish is removed. Remember, zebra finches and other birds are fairly neophobic; our zebra finches rarely explore their training cages unless they are motivated.

*Stage 2: acclimating subjects to the moving feeder.* This stage has 11 steps as follows.

(1) Open Matlab (remember to ‘run as administrator’ in Windows 7). Select the Monitor NI Card tab (Fig. S4, A.1) in the upper left corner (underlined terms are defined in the Glossary). A graphical user interface (GUI) will appear (Fig. S4, A.2). Press Start/Stop. Text will appear above the Start/Stop button that reads, ‘Actively monitoring NI card’.

(2) Open a new instance of Matlab. Select the Conditioning tab (Fig. S4, B.1) in the upper left corner. A GUI will appear with a list of several training paradigms to choose from (Fig. S4, B.2). Select Acclimate. A GUI will appear (Fig. S5).

(3) Check the appropriate booth number (Fig. S5, A).

(4) Click Initialize Booth. Repeat steps 2–4 for every booth that you want to run; i.e. if you are running four booths, there should be five instances of Matlab (including Monitor NI Card instance) open.

(5) Set the Timing (sec) parameters (Fig. S5, B) as follows: Start Delay = 0; Response Duration =10; ITI minimum = 30, maximum = 60. This sets Acclimate to deliver food for 10 s, with a range from every 30 to 60 s.

(6) In Settings, click Save to Load these Timing parameters in future sessions (Fig. S5, C).

(7) Name the file (Fig. S5, D). Optional: check Verbose (Fig. S5, E).

(8) Press Start/Stop (Fig. S5, F) to begin the program. A text message will appear next to the Start/Stop button that reads ‘Waiting...’ verifying that the program is running. The bird should begin to eat regularly from the feeder in 3 to 5 h; however, some birds may require a few sessions to eat from the feeder.

(9) Press Start/Stop to stop the program at any time. A text message will appear next to the Start/Stop button that reads ‘Press start/stop to begin task’ verifying that the program is stopped.

(10) Click Close Booth (Fig. S5, A) to turn off the booth at the end of a session.

(11) Click Start/Stop on the Monitor NI Card GUI to discontinue monitoring the NI card.

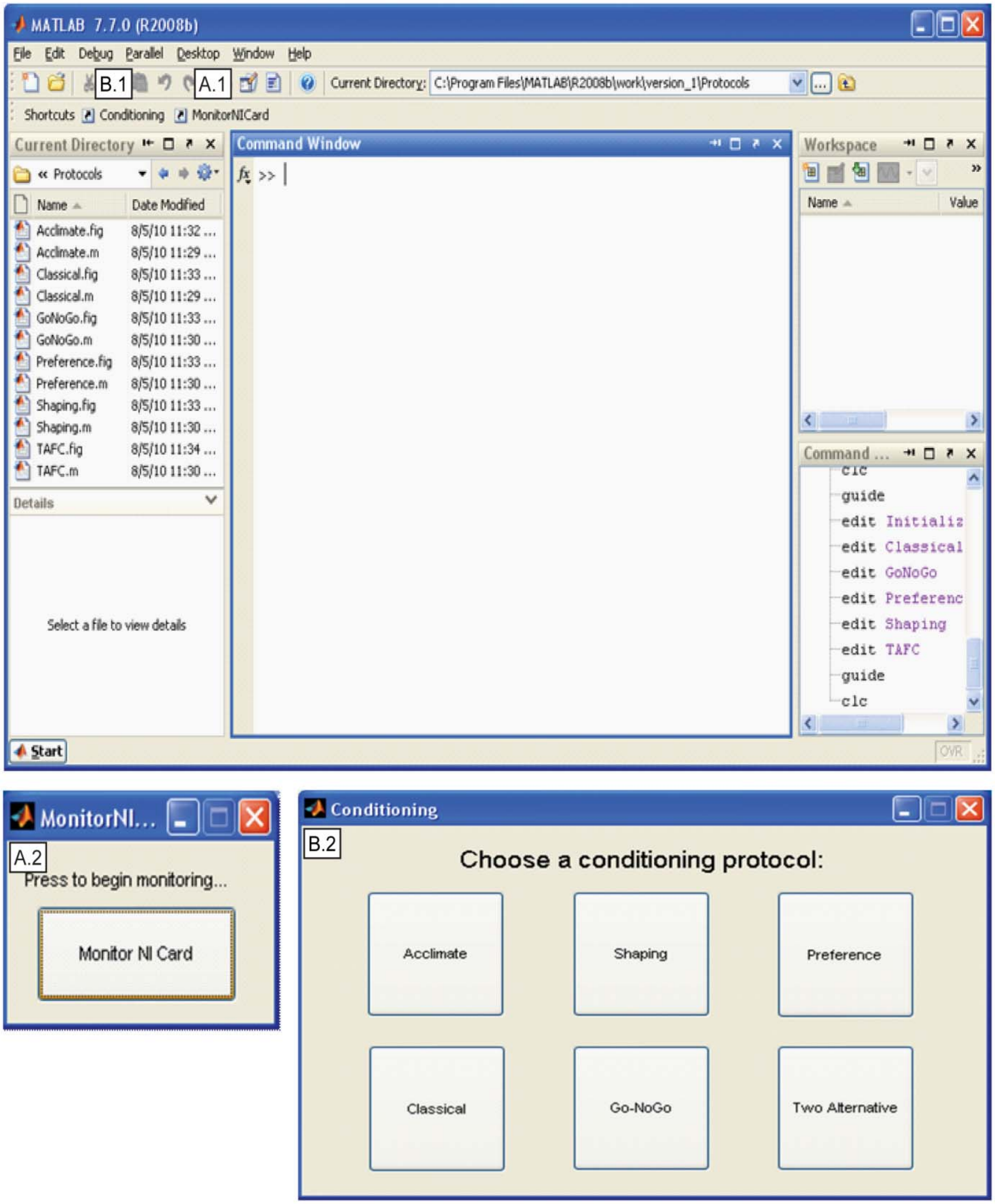


**Figure S4.** Graphical user interfaces used to test and train subjects.

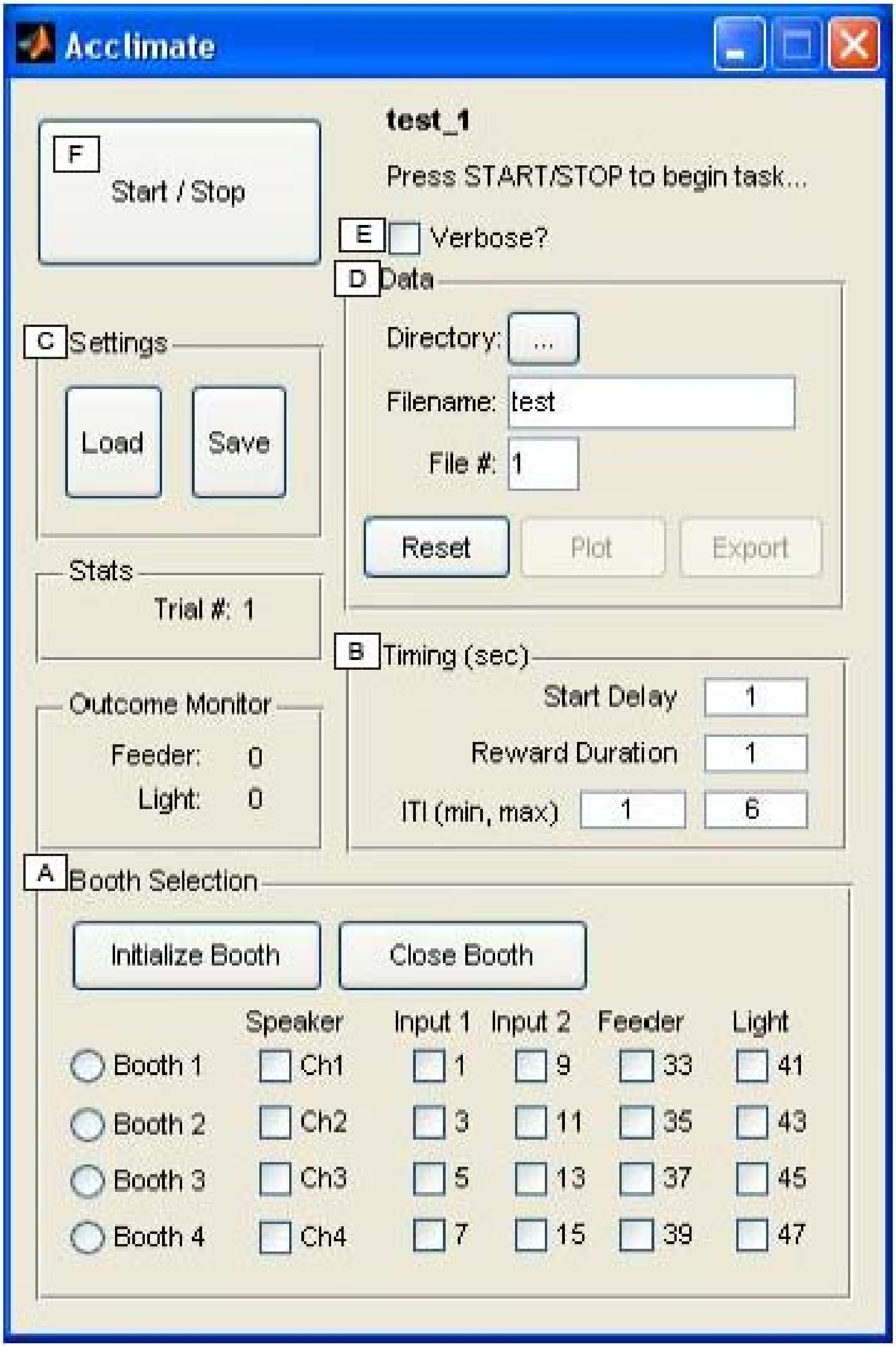


**Figure S5.** Graphical user interface used in acclimating subjects to the moving feeder.

*Stage 3: training the bird to peck the sensor for food*

(1) See Step 1 of Stage 2.

(2) Open a new instance of Matlab; select Conditioning, followed by Shaping. The GUI shown in Fig. S6 will appear.

(3) See Steps 3 and 4 of Stage 2.

(4) Set the Timing (sec) (Fig. S6, B) parameters as follows: Start Delay = 0; Reward Duration =10; ITI minimum = 0; maximum = 0. This sets Shaping to deliver food for 10 s when the bird trips the sensor.

(5) See Steps 6 and 7 of Stage 2.

(6) Roll up a piece of laboratory labelling tape, sticky-side out. Dip one side in seeds. Stick the tape with the seeded-side out, to the response panel inside the sensor just above the infrared beam (delineated by the glass windows).

(7) See Step 8 of Stage 2. After 20 min to 1 h, the bird should begin to eat the seeds off the tape, ‘accidentally’ activating the feeder by breaking an infrared beam in the sensor. The bird begins to learn the association between the feeder and the sensor after repeated, ‘unintentional’ feeder activations while eating seeds off the tape. Learning this association will not be instantaneous. The bird does not ‘grasp’ the association until it eats directly from the feeder immediately after using the sensor to deploy it. The bird may move away from the response panel after eating all the seeds on the tape, and/or not eat from the feeder despite triggering it; therefore, it is necessary to monitor the bird during this stage and to replenish seed on the tape as needed. When you observe the bird consistently eating from the feeder, leave the tape, but without seed. Let the bird train for at least one full session with tape only. The bird should master Stage 3 after about one to three sessions (about 6 h each); however, some birds may need more sessions.

(8) See Steps 9–11 of Stage 2.

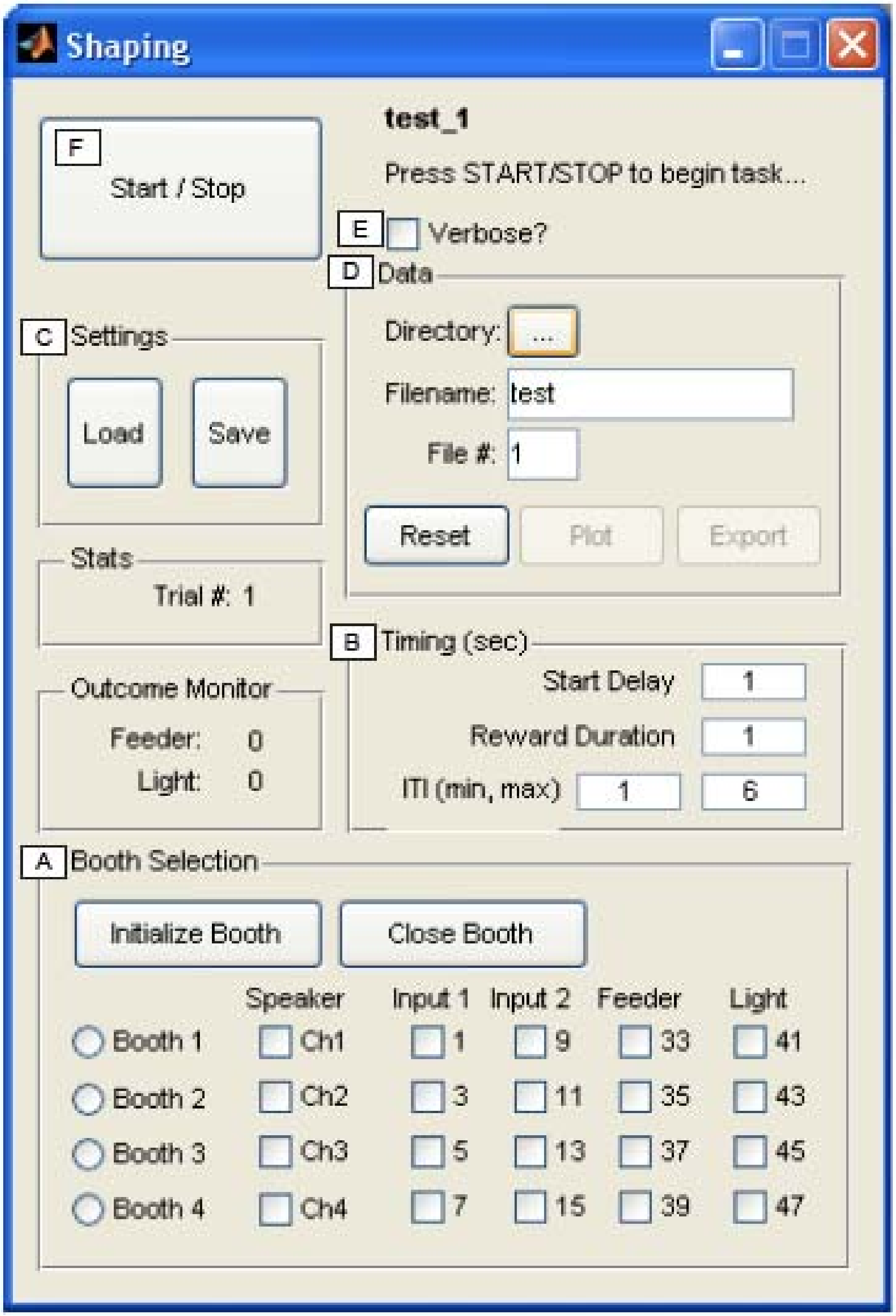


**Figure S6.** Graphical user interface used to train subjects to peck the sensor for food.

*Phase 3: GNG training*

(1) See Step 1 of Stage 2.

(2) Open a new instance of Matlab; select Conditioning, and then Go/No Go. The GUI shown in Fig. S7 will appear.

(3) See Steps 3 and 4 of Stage 2.

(4) Set the Timing parameters (Fig. S7, B) as follows: Start Delay = 0; Pre-Response Duration = 0 (mainly used with sounds of short duration; if the sounds are very short (e.g. zebra finch long calls), then the birds cannot move out of the sensor fast enough, which results in a reward or punishment regardless of their 'intended' response); Response Duration = 2; Reward Duration = 6 (when the birds are competent at the task, a 6 s reward duration strikes an optimal balance for a 10 h training session between the amount of food necessary to keep them robust and motivated.); Punishment Duration =16; Null Response Duration = 6; ITI minimum = 0; maximum = 0. This sets GNG so that the bird has 2 s to respond after initiating a trial (i.e. after stimulus playback). Following a response, there is a 6 s food reward, a 16 s lights-out punishment, or a 6 s null period (depending on the playback type; i.e. go or nogo), which is the Stimulus Outcome Delay.

(5) See Steps 6 and 7 of Stage 2.

(6) Select Go and No Go sounds (Fig. S7, G). In each box, select the desired sound file(s). To select a single file, click on the entry. To select multiple files, first click on a single entry and hold down the control (or shift) key while clicking on additional entries in a single list.

(7) Check Verbose (Fig. S7, E). Optional: check Repeat Errors (Fig. S7, H). We do not recommend using the optional setting 'check Repeat Errors' except briefly during baseline training after the bird reaches the point where it initiates at least 200 trials per session. Typically, we deselect this option when birds cease to repeat false alarms. Repeat errors are used to shift subjects' focus from only obtaining food to listening to the playbacks to obtain food.

(8) See Steps 9–11 of Stage 2.

(9) Files can be exported to Excel by clicking Export (Fig. S7, D).

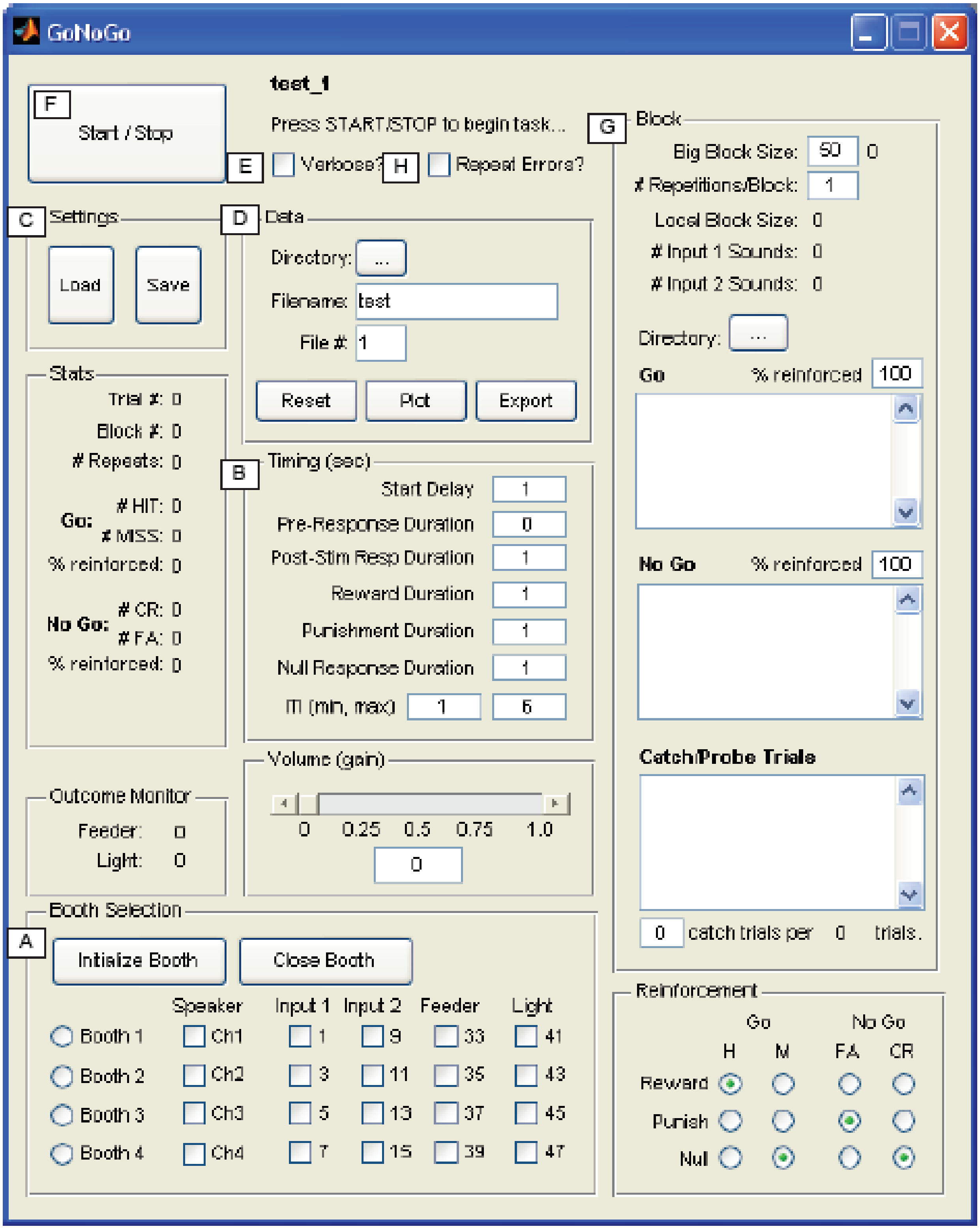


**Figure S7.** Graphical user interface used to train subjects on go-nogo tasks (Phase 3, GNG training).

*Classical Conditioning*

The following is a description of the program parameters and behavioural procedures for classical conditioning.

*Phase 1: acclimation*

See Phase 1 of GNG.

*Phase 2: shaping*

There are two stages to shaping. See Stage 1 and 2 of GNG.

*Phase 3: classical conditioning*

(1) See Step 1 in Shaping Stage 2 of GNG

(2) Open a new instance of Matlab; select Conditioning, then Classical. The GUI shown in Fig. S8 will appear.

(3) See Steps 3 and 2 in Shaping Stage 2 of GNG.

(4) Set the Timing parameters (Fig. S8, B; see Glossary).

(5) See Steps 6 and 7 in Shaping Stage 2 of GNG.

(6) Select POS, NEG and NEUTRAL sounds (Fig. S8, G) (see Step 6 in Phase 3 of GNG for sound selection).

(7) See Steps 9–11 in Shaping Stage 2 of GNG.

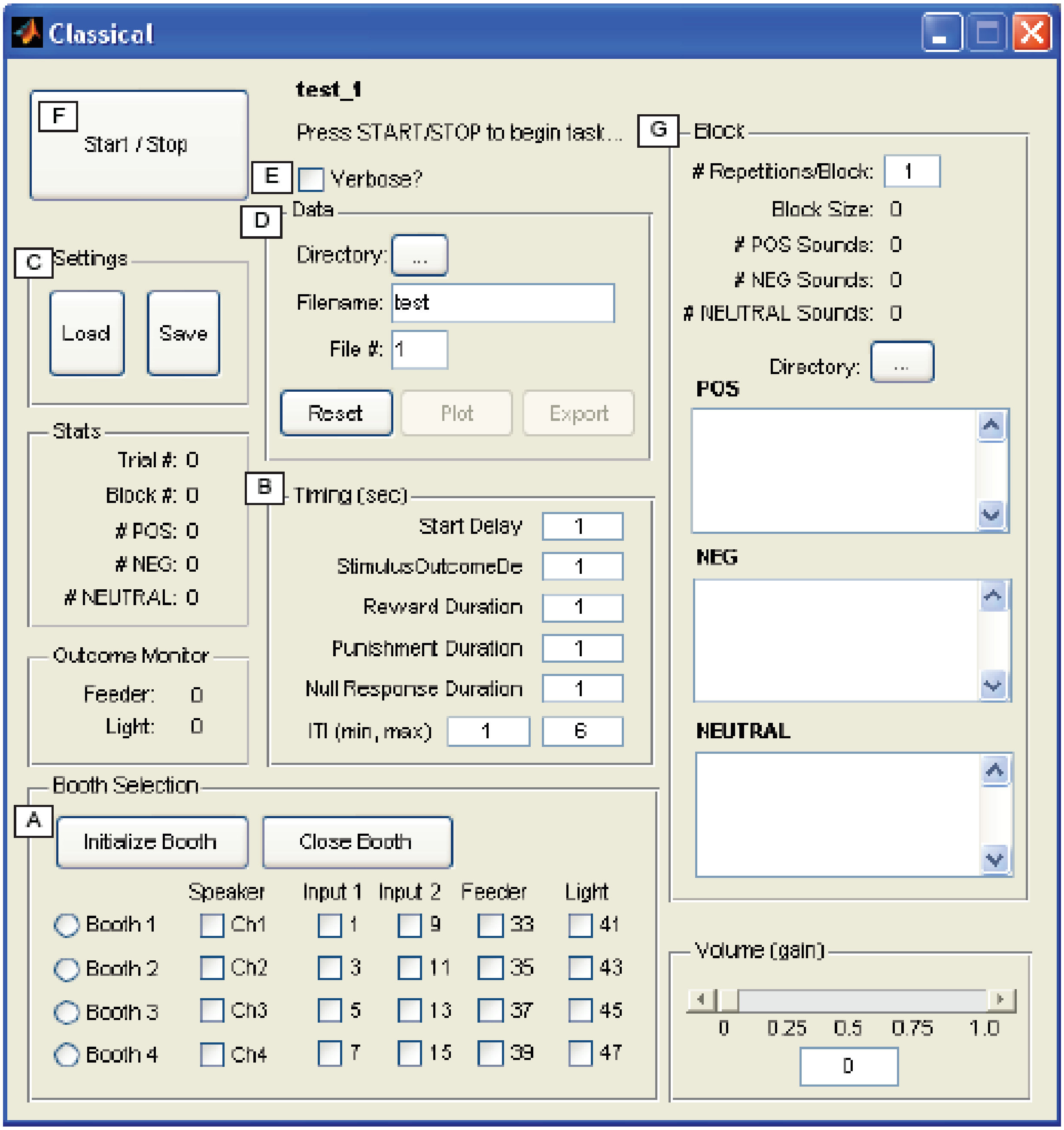


**Figure S8.** Graphical user interface for classical conditioning (Phase 3).

*Two Alternative Forced Choice (TAFC)*

The following is a description of the program parameters and suggested behavioural procedures for the two alternative forced choice task (Klump 1995; Dent et al. 2008).

*Phase 1: acclimation*

See Phase 1 of GNG.

*Phase 2: shaping*

See Shaping Stages 1 and 2 of GNG.

*Stage 3: training the bird to peck two sensors for food*. In TAFC, the subject needs to interact with two inputs. Follow the procedures in Shaping Stage 3 of GNG. When training, select the sensor that you want to use for trial initiation (i.e. left or right; Dent et al. 2008). The TAFC program does not make a distinction between the two inputs for initiation. Alternatively, train on one sensor in a single training session, alternating sensors in consecutive sessions. Hint: it may be helpful to make the tape used in each sensor a different colour.

*Phase 3: TAFC training*

(1) See Step 1 in Shaping Stage 2 of GNG.

(2) Open a new instance of Matlab; select Conditioning, then TAFC. The GUI shown in Fig. S9 will appear.

(3) See Steps 3 and 2 in Shaping Stage 2 of GNG.

(4) Set the Timing (Fig. S9, B) parameters.

(5) See Steps 6 and 7 in Shaping Stage 2 of GNG.

(6) Select Input 1, Input 2 (Fig. S9, G) (see Step 6 in Phase 3 of GNG for selecting sounds).

(7) See Steps 9–11 in Shaping Stage 2.

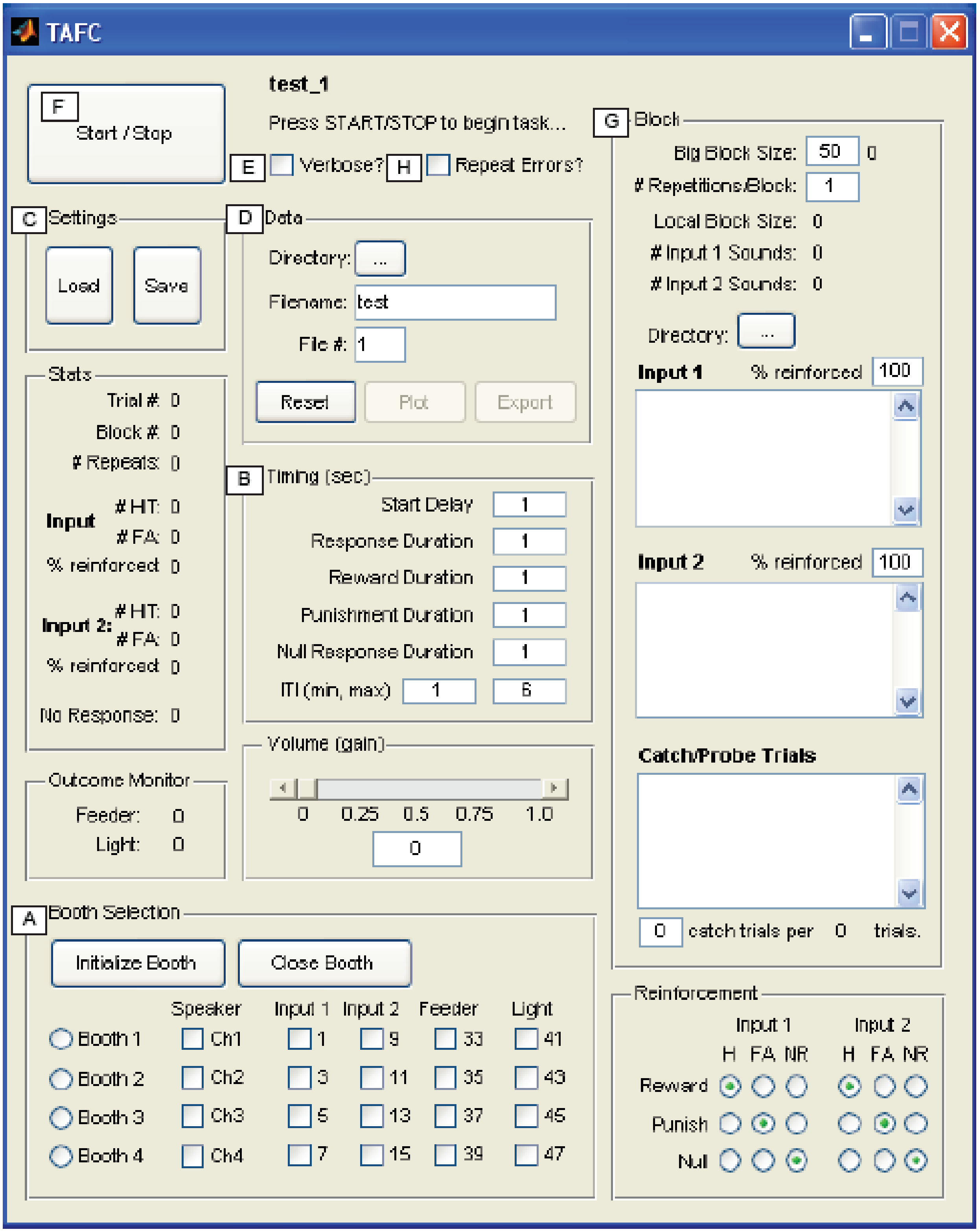


**Figure S9.** Graphical user interface used to train subjects on two alternative forced choice (TAFC) tasks (Phase 3, TAFC training).

*Preference*

The following is a description of the program parameters and overview of recommended behavioural procedures for two-input preference tests.

*Phase 1: acclimation*

Move birds to training booths 1 day prior to perch preference tests. Pecking preference tests do not require an acclimation phase.

*Phase 2: shaping (for pecking preference tests)*

Move birds to training booths. Birds need to be (auto)shaped to peck in sensors (or at keys). We recommend installing red LEDs on the response panel behind the sensors (placing food in the sensor, as previously described for our other operant paradigms, is not appropriate for a preference test). Refer to Houx & ten Cate (1999) and Holveck & Riebel (2007) for a comprehensive outline of shaping procedures using keys and LEDs.

*Phase 3: preference testing*

Preference tests may need to occur in two stages if using perches because birds may show a side bias (i.e. if the birds spend most of their time on one side of the training cage). In these cases, measure the bird's side bias for 2 h before the start of every preference test. During the test, pair the control stimulus with the perch on the preferred side.

(1) See Step 1 in Shaping Stage 2 of GNG.

(2) Open a new instance of Matlab; select Conditioning, then Preference. The GUI shown in Fig. S10 will appear.

(3) See Steps 3 and 2 in Shaping Stage 2 of GNG.

(4) Set the Timing (Fig. S10, B) parameters as follows: Start Delay = 0; if using pecking hardware, Perch Duration = 0; if using perches, Perch Duration = 0.2 to 0.7 (see Glossary); ITI minimum = 0 and maximum = 0. This sets Preference in one of two ways depending on the input hardware: if using pecking hardware, birds will initiate playbacks immediately after pecking in a sensor; if using perches, then birds must remain on a perch for 0.2 to 0.7 s to initiate a playback.

(5) See Steps 6 and 7 in Shaping Stage 2 of GNG.

(6) Select Input 1 and Input 2 sounds (Fig. S10, G) (see Step 6 in Phase 3 of GNG for selecting sounds).

(7) See Steps 9–11 in Shaping Stage 2 of GNG.

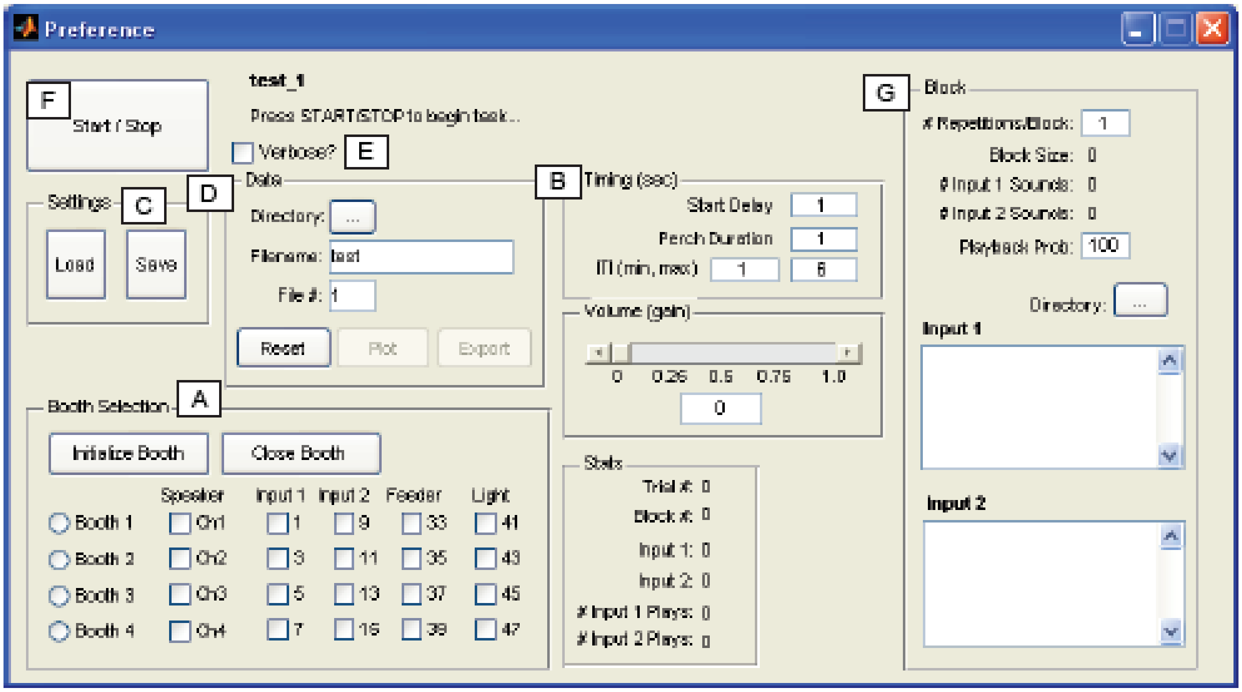


**Figure S10.** Graphical user interface used for preference tests (Phase 3, Preference testing).

## References

**Dent, M. L.,Welch, T. E., Mcclaine, E. M., Shinn-cunningham, B. G.** 2008. Species differences in the identification of acoustic stimuli by birds. *Behavioural Processes*, **77**, 184-190.

**Holveck, M. & Riebel, K.** 2007. Preferred songs predict preferred males: consistency and repeatability of zebra finch females across three test contexts. *Animal Behaviour*, **74**, 297-309.

**Houx, A. & ten Cate, C.** 1999. Song learning from playback in zebra finches: is there an effect of operant contingency? *Animal Behaviour*, **57**, 837-845.

**Klump, G. M. (Ed.)** 1995. *Methods in Comparative Psychoacoustics*. Basel: Birkhäuser Verlag.

## GLOSSARY

*(Organized by Parameter Modules in Training Graphical User Interfaces, GUIs)*

**Block**

*Big Block Size.* In GNG and TAFC: the big block is used to determine the rate at which outcomes are reinforced when using a partial reinforcement paradigm. As an example, if the local block size (see below) is 6 (e.g. 3 go songs and 3 nogo songs) and the big block size is set to 100, the actual big block size will be 102 (6 x 17), which is the nearest integer multiple of the local block size. In this case, the reinforcement rate can be controlled with approximately 1% precision, since the reinforcement rate will be determined over 102 trials. Alternatively, if the big block size is set too low (e.g. 10), the actual big block size will be 12 trials long, and the actual rate of reinforcement will only change in increments of 8.33%, even if the desired resolution is finer. Therefore, the big block size should be large enough to affect the resolution with which partial reinforcement is desired.

*Catch/Probe Trials*. Subjects are not reinforced if they respond after playback of songs in this list. This feature was designed to allow the end-user to include sounds that differ from those in the reinforced lists in order to test aspects of the subject's perception. The default is 0. These trials are inserted randomly into a local block by replacing one of the reinforced sounds.

*Go***.** Subjects are reinforced with food if they respond to sounds in this list after playback.

*Input 1*. In Preference: sounds that playback after the subject hops on perch 1. In TAFC: reinforced sounds that playback after the subject responds by activating sensor 1.

*Input 2*. In Preference: sounds that playback after the subject hops on perch 2. In TAFC: reinforced sounds that playback after the subject responds by activating sensor 2.

*Local Block Size*. In GNG and TAFC: reinforced sounds are presented pseudorandomly based on the total number of sounds in the reinforced sound lists, in random order (see #Repetitions/Block). In Preference: local blocks are generated separately for Input 1 and Input 2 (see #Repetitions/Block).

*No Go*. Subjects are reinforced with light-out/darkness if they respond to sounds in this list after playback.

*# Catch Trials per x trials*. Number of catch trials inserted into a local block.

*# Repetitions/Block*. Rate of playback for each stimulus in a local block.

*# Input 1 Sounds*. Number of selected sounds in Input 1 list.

*# Input 2 Sounds*. Number of selected sounds in Input 2 list.

*# NEG Sounds*. Number of selected sounds in NEG sounds list.

*# Neutral Sounds*. Number of selected sounds in Neutral sounds list.

*# POS Sounds*. Number of selected sounds in POS sounds list.

*NEG (negative).* Playback of these sounds is followed by punishment.

*NEUTRAL*. Playback of these sounds is not followed by a contingency.

*Playback Probability*. Likelihood of sound playback on a given perch. In our experience, performance is optimal when this parameter ranges between 0.2 and 0.33.

*POS (positive)*. Playback of these sounds is followed by reward.

*% reinforced.* Percentage of reinforced trials.

**Booth Selection**

*Booth 1*. Select to initialize booth 1.

*Booth 2.* Select to initialize booth 2.

*Booth 3*. Select to initialize booth 3.

*Booth 4*. Select to initialize booth 4.

*Close Booth*. Inactivates the booth’s detector and effector components.

*Feeder.* Activated/inactivated for selected booth when initialized or closed.

*Initialize Booth.* Activates the booth’s detector and effector components.

*Input 1*. Activated/inactivated for selected booth when initialized or closed.

*Input 2*. Activated/inactivated for selected booth when initialized or closed.

*Light*. Activated/inactivated for selected booth when initialized or closed.

*Speaker*. Activated/inactivated for selected booth when initialized or closed.

**Conditioning**

*Acclimate*. Runs classical conditioning paradigm. Designed to introduce moving feeder to subjects. Activates feeder at intervals set by the end-user.

*Classical*. Runs classical conditioning paradigm. Designed to train subjects to associate positive and negative contingencies with sound stimuli. Uses feeders, lights and speakers.

*Go-No Go.* Runs operant conditioning paradigm. Designed to train subjects to discriminate between sound stimuli through reinforcement. Uses sensors, feeders, lights and speakers.

*Preference*. Runs operant conditioning paradigm. Designed to measure subjects’ preferences for sound stimuli through reinforcement. Uses sensors and speakers.

*TAFC*. Runs operant conditioning paradigm. Designed to train subjects to categorize sound stimuli through reinforcement. Uses sensors, feeders, lights and speakers.

*Shaping*. Runs operant conditioning paradigm. Designed to train subjects to activate sensor for food. Uses sensors and feeders.

**Data**

*Directory*. Data are saved to this location.
*Filename*. Name of data file.
*File #*. Data file number. Inserts ‘ _’# at the end of the filename in the directory. Numbers are added sequentially to a given filename when the data are reset.
*Reset*. Creates a new data file with a new file #. Clears Stats.
*Plot*. Generates two plots: (1) mean percentage correct per 10 trials; (2) mean percentage correct per 10 trials by stimulus type (e.g. Go and Nogo sounds). Stop the program to access this feature; then click the Plot button.
*Export*. Generates Excel spreadsheets for GNG with five variables for each trial: (1) Response type, 1=Response, 0=No response; (2) Playback sound number, which is determined alphabetically in the wave directory and by list; (3) Time the playback started; (4) Time the playback stopped; (5) Time the subject responded.

**Monitor NI Card**

The Monitor NI Card Matlab function actively monitors sensor activity in all four booths and also monitors the feeder and light status of each booth. Monitor NI Card is the crucial interface between the Matlab behavioural protocols and the NI card, which communicates with the training booths.

**Outcome Monitor**

*Feeder*. Provides the current status of the feeder (1 = feeding; 0 = not feeding).
*Light*. Provides the current status of the light (1 = on; 0 = off).

**Reinforcement**

*CR (correct rejection).* In GNG: the subject responds correctly by withholding response to a nogo stimulus. These responses are unreinforced.
*FA (false alarm).* In GNG: the subject responds incorrectly to a nogo stimulus by pecking the sensor. These responses are reinforced with lights-out. In TAFC: the subject responds incorrectly by pecking the incorrect input/sensor. These responses are reinforced with light-out/darkness.

*H (hit).* In GNG: the subject responds correctly to a go stimulus by pecking the sensor. These responses are reinforced with food. In TAFC: the subject responds correctly by pecking the correct input/sensor in response to a stimulus. These responses are reinforced with food.
*M (miss).* In GNG: the subject responds incorrectly by withholding its response to a go stimulus. These responses are unreinforced.
*NR.* In TAFC: the subject does not respond to a stimulus by pecking either input/sensor. These responses are unreinforced.
*Null*. In GNG: period following misses and correct rejections. The subject cannot initiate a new trial during this period. In TAFC: period following no response.
*Punish*. Lights-out/darkness.
*Reward*. Feeder.

**Repeat Errors**

Repeats the previous trial; the same stimulus will play again if the bird responded incorrectly.

**Settings**

*Load*. Loads saved timing parameters saved in a directory.
*Save*. Saves timing parameters so they can be loaded in future sessions.

**Start/Stop**

Starts or stops the program. Text prompts appear next to the button verifying that the program is running or stopped.

**Stats**

*Block #*. Number of local blocks cycled through during a training session (here, a training session refers to the filename and file number).
*Input 1*. Total number of activations of sensor 1.
*Input 2*. Total number of activations of sensor 2.
*# CR*. In GNG: number of correct rejections during a training session. Shown as a function of nogo trials.
*# FA*. In GNG: number of false alarms during a training session. Shown as a function of nogo trials. In TAFC: shown for both inputs.
*# H*. In GNG: number of hits during a training session. Shown as a function of go trials. In TAFC:

shown for both inputs.

*# Input 1*. In Preference: number of perches on perch 1 that activated playback(s) from Input 1 list. This may be less than the value displayed in Input 1, since the Playback Probability may be less than 100%.

*# Input 2*. In Preference: number of perches on perch 2 that activated playback(s) from Input 2 list. This may be less than the value displayed in Input 2, since the Playback Probability may be less than 100%.

*# M*. In GNG: number of misses during a training session. Shown as a function of go trials.

*# repeats*. Number of repeated trials during a training session if the repeat errors feature is selected.

*No response*. Number of null responses in a training session.

*% reinforced*. Number of trials reinforced during a training session.

*Trial #*. Number of trials completed during a training session.

**Timing (Sec)**

*Intertrial interval (ITI)*. Minimum (left) and maximum (right) duration between the end of one trial and the beginning of the subsequent trial. On each trial, the ITI is chosen from a flat distribution. The ITI is randomized to prevent temporal pattern learning. Note: the ITI should be at least 30 s in Classical to create distinct stimulus contingencies.

*Null Response Duration*. Defines a period of no outcome after a stimulus; new trials cannot be initiated during this period.

*Perch Duration*. Depending on the activity level of your subjects, this time needs to be adjusted but should be in the range of 0.2–0.7 s.

*Pre-Response Duration.* Defines the time before a subject can respond after sound playback.

*Punishment Duration*. Defines how long the lights will be turned off.

*Response Duration*. Time a subject has to respond to a stimulus after playback.

*Reward Duration*. Defines how long the feeder will be available to subjects.

*Start Delay*. Time between trial initiation and stimulus playback.

*Stimulus Outcome Delay*. Duration between the end of a playback and the delivery of a reward, punishment or null period. It is important that the contingency occurs right after the sound (e.g. if the reward is delivered too soon, or too late, the bird will not associate a stimulus with an outcome).

**Verbose**

This feature displays detailed, real-time trial events in the command window of the selected booth.

**Volume (gain)**

Adjust the volume of the sounds presented to a booth. The volume control adjusts the gain applied to the wave files and can range from 100% (gain of 1) to 0% (gain of 0). In combination with the scale of the wave files and the volume setting on the amplifier, the Volume control sets the volume at which sounds will be presented. Hint: keep the hardware amplifier volume at a constant setting and adjust the volume with this GUI control. Use a sound level meter to convert the volume settings to dB SPL.