**Getting started**

**System Requirements:**

* MATLAB 2019 or greater.
* MATLAB support package for Arduino hardware.
* Tested on Windows 10.
* Arduino Uno board.

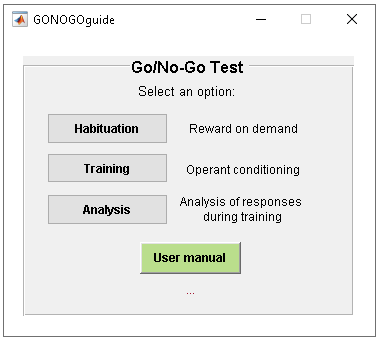
**Installation:**

* Download the source code as a zip file from: <https://github.com/NeuralCircuitsLabC01/GUIDE_GO-NOGO>.
* Run MATLAB and add the folder “GUIDE\_GO-NOGO” to the MATLAB path. See MATLAB documentation for help.

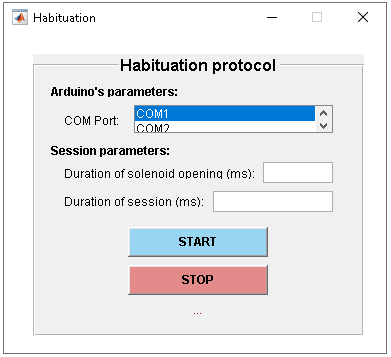
**GUIDE overview:**

* We have created separated MATLAB routines to perform the Go/No-Go test. Automated tasks include “Habituation” and “Training” (see Hernandez-Soto et al., 2021 for more information about the Go/No-Go protocol).
* In addition, a MATLAB routine has been created to analyze the results obtained from the Training routine.
* To run the MATLAB routines, the Arduino board is needed to be connected to the computer.

**Run the GUIDE:**

* In the MATLAB Command Window, type “GONOGOguide”and press “enter”.
* The following graphical user interface window will appear:
* Select a protocol: “Habituation”, “Training “or “Analysis”.
* The button “User manual” will open this document.

**Habituation protocol:**

* By pressing the “Habituation” button, the following graphical user interface will emerge:
* Habituation protocol allows to reward with water every time that animal licks the sensor (see Hernandez-Soto et al., 2021 for more information about the Go/No-Go protocol).
* Specify the COM Port in which the Arduino board is connected. Use the next septs to know the COM Port:
  + Click on the “Start Menu” (Windows Taskbar) and open up the “Control Panel”.
  + While in the “Control Panel”, navigate to “System and Security”. Next, click on “System”. Once the “System” window is up, open the “Device Manager”.
  + Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)".
  + For more details see <https://www.arduino.cc/en/Guide/ArduinoUno>.
* Type the session parameters (“Duration of solenoid opening (ms)” and “Duration of session (ms)”).
  + We recommend a 50 ms solenoid opening for a ~20µl water output.
  + A session duration of 10 min (600000 ms) is recommended to avoid animals’ satiety (see Hernandez-Soto et al., 2021 for more information).
* Press the START button to run the code.
* While MATLAB routine is running, a new window will emerge showing the animals’ performance during session.
* User may press the STOP button to abort the session.

**Training protocol:**

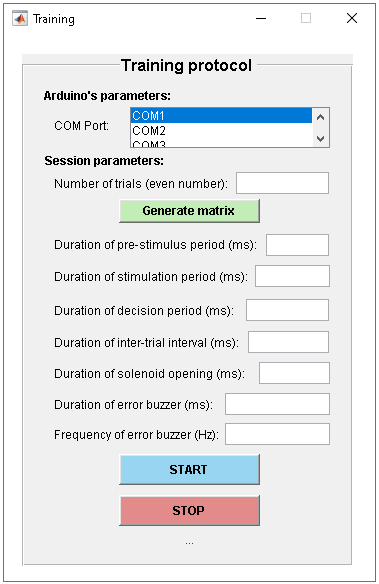
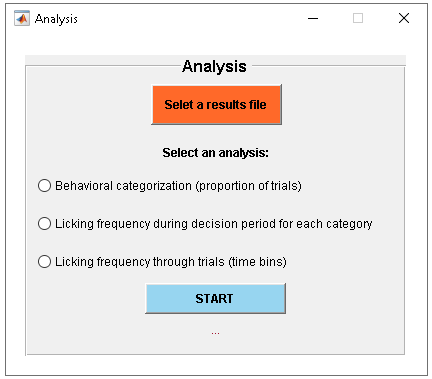
* By pressing the “Training” button, the following graphical user interface will emerge:
* Training protocol allows to perform the Go/No-GO olfactory conditioning by rewarding the animal with water every time it responds to the Go olfactory stimulus and with a buzzer tone when animal respond to the No-Go olfactory stimulus (see Hernandez-Soto et al., 2021 for more information about the Go/No-Go task).
* Specify the COM Port in which the Arduino board is connected. To know the COM Port, use the steps specified in the “Habituation” section.
* Type the number of trials during the Go/No-Go session. An even number is needed.
* Press “Generate matrix” to generate the matrix of olfactory stimuli.
  + The matrix of olfactory stimuli is a [n x 10] matrix with the numbers separated by commas. n = No.Trials.
  + The matrix will be saved in a “.dat” file.
  + To stablish the sequence of olfactory stimuli delivered by the olfactometer, it’s needed that user add the stimuli matrix to the code at the olfactometer software (for more information about the olfactometer software, see Slotnick et al., 1991 and Slotnick & Restrepo, 2005).
* Specify the duration (ms) of the session parameters according to the following scheme:

Gráfico de barras

Descripción generada automáticamente con confianza baja

* Type the duration (ms) of the solenoid opening (for water reward) and the duration and frequency of the error buzzer (for punishment).
  + We recommend a 50 ms solenoid opening for a ~20µl water output.
  + An error buzzer duration of 200-500 ms is also recommended.
  + An error buzzer frequency of 5000 Hz is recommended to differentiate it to the tone (8000 Hz) that starts each trial. MATLAB only accepts a buzzer frequency between 0 Hz and 32767 Hz.
* Press the START button to run the code.
* While MATLAB routine is running, a new window will emerge showing the animals’ performance during trials.
* User may press the STOP button to abort the session.
* When the session has finished, a results window will appear showing a raw raster plot of licks through trials and the proportion of Correct Responses and Errors in the session.
* Specify a folder and a file name to save the results of the session (data and graphics). Do not close the figures until data be saved.

**Analysis:**

* By pressing the “Analysis” button, the following graphical user interface will emerge:
* The “Analysis” routine permit to analyze the behavioral categorization of each Go/No-Go trial (Hits, Misses, Correct Rejections and False Alarms) from the raster plot of licks that results from a Training session (see Hernandez-Soto et al., 2021 for more information about the Go/No-Go task and the behavioral categorization). Also, the routine allows to analyze the licking frequency during decision period per behavioral category and both the raw and z-scored licking frequency through trials.
* Press the “Select a results file” button to select a results file from a training session.
* Select one of the following analyses:
  + “Behavioral categorization (proportion of trials)” analyzes the proportion of Hits, Misses, Correct Rejections and False Alarms trials of a Go/No-Go session. This routine also divides the raw raster plot of licks in their respective behavioral categorization.
  + “Licking frequency during decision period for each category” analyzes the mean licking frequency during the decision period for each behavioral category (Hits, Misses, Correct Rejections and False Alarms).
  + “Licking frequency through trials (time bins)” analyzes the licking frequency through trials for each behavioral category (Hits, Misses, Correct Rejections and False Alarms). To calculate the licking frequency through time, a time bin (ms) needs to be specified. Time bin must be at least 2 times higher than the Arduino Uno sampling interval (it depends on computer settings and USB connection). Both the absolute and z-scored licking frequency through time are calculated.
* User may specify a folder and a file name to save the results obtained from each analysis routine. Results are saved as a MATLAB a cell array.

General comments:

* Send questions or suggestions for improvement to:
  + Hernández-Soto R: rebecahernandezsoto@comunidad.unam.mx
  + Villasana-Salazar B: benvillasanasalazar@gmail.com
* To cite this software, please refer to the following paper:
  + Hernandez-Soto R et al. (2021). An open source and low-cost operant Go/No-go conditioning olfactory system based on Arduino microcontroller.