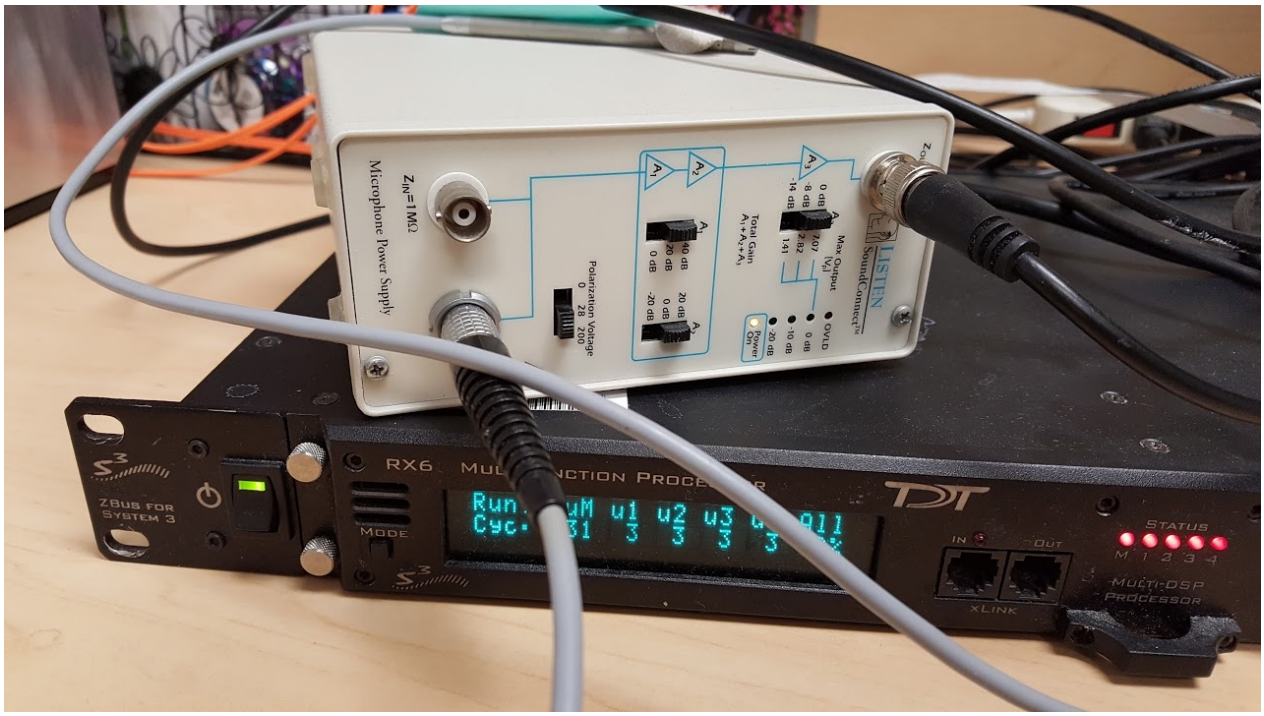
Ultrasonic Recording Manual

Table of contents

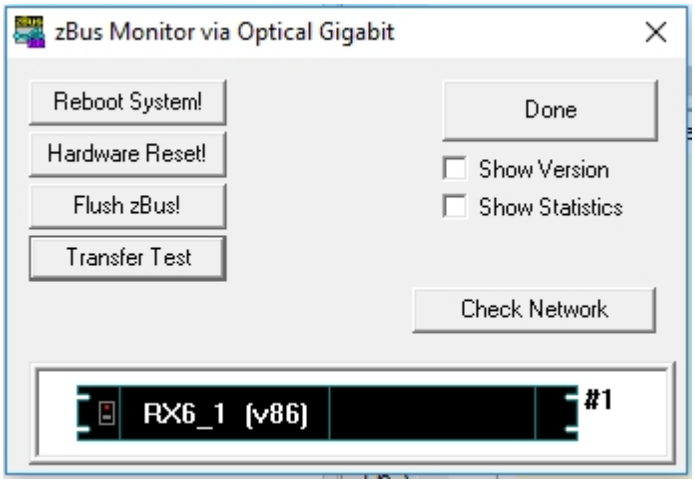
Introduction

Welcome to the Ultrasonic Recording Manual, your go-to source of information for recording ultrasonic rat vocalizations with a Tucker-David Technologies (TDT) RX6 Multifunction Processor via the Ultrasonic_Recording_Interface software.
This manual provides information about navigating the graphical user interface (GUI) and possible errors that you may encounter. Should the program become insufficient for new applications, the manual contains a section on modifying the source code. If you encounter issues that are not addressed in this manual, please contact Erica Nordin at erica.nordin@uleth.ca.

Starting the RX6 and microphone



The RX6 and microphone each have a hardware power switch which must be started independently before opening the program. If the RX6 is on, it will display information on the front display. If the microphone is on, the "Power On" button will be lit.
If the RX6 is not already on, the user must run a procedure through the zBUSmon program to prevent DC offset. A shortcut to zBUSmon is located on the desktop.



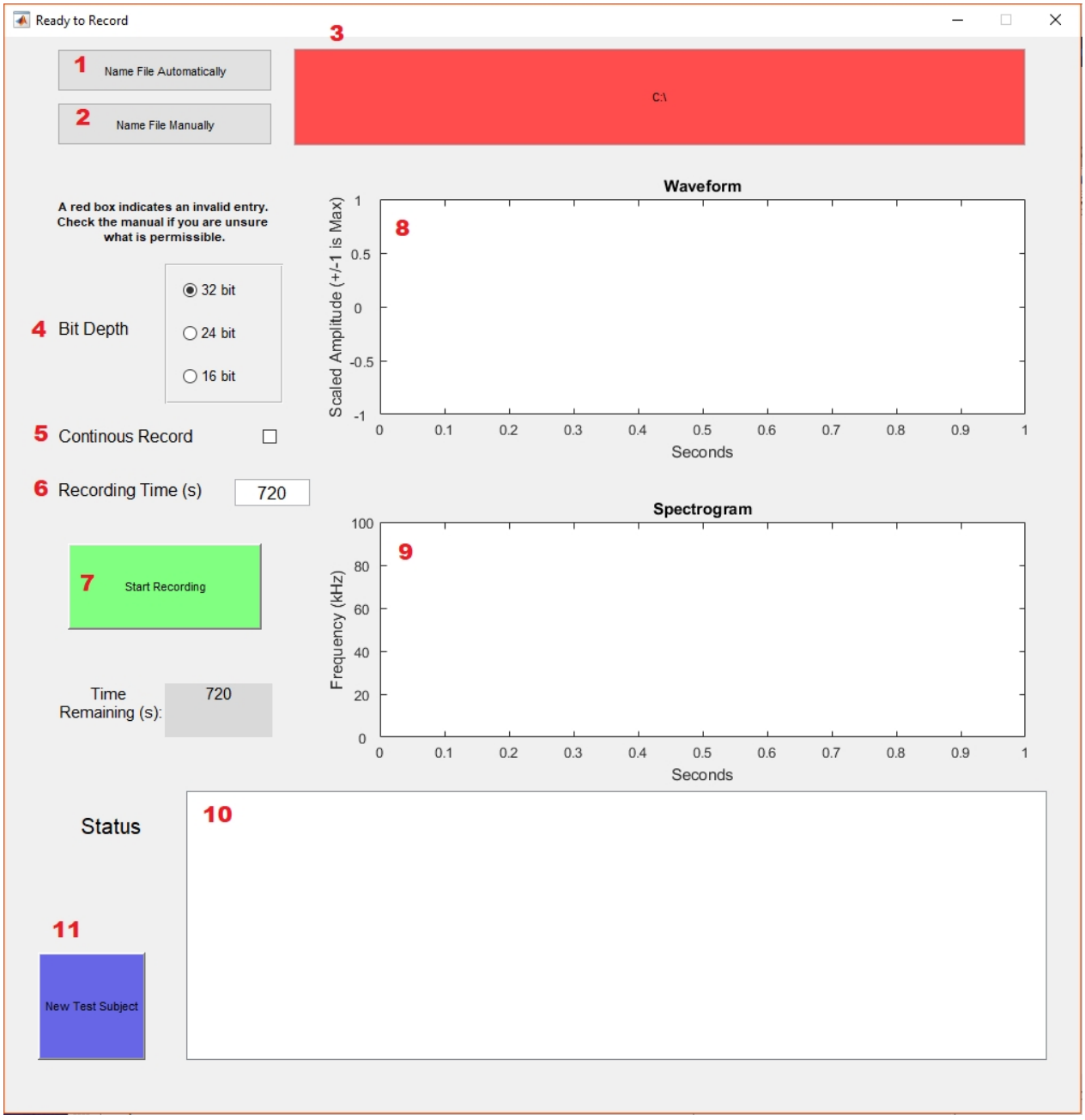
The RX6 should be visible at the bottom of the window. Press "Reboot System!", wait for the reboot, and

then press "Transfer Test". If the "Test Passed" message appears, you may press "Done" to close zBUSmon and proceed to the acquisition program.

Opening the program

A shortcut for the program is placed on the desktop. If the shortcut is absent, the program can be found at C:\Ultrasonic Recording Program\Ultrasonic_Recording_Interface. Opening the program will display the main GUI, outlined in the next section. Don't fret if nothing happens immediately; the program takes a bit of time to open.

GUI Overview



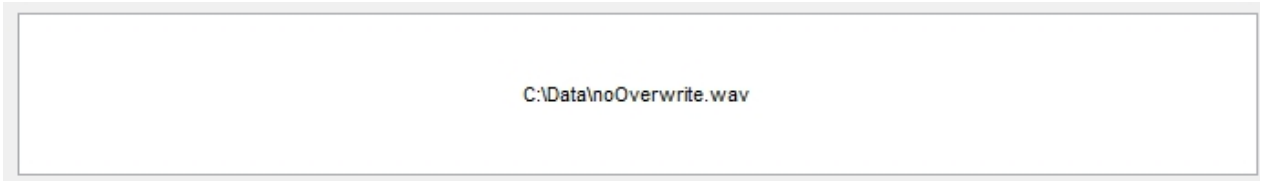
1. Automatic Naming: enables the user to enter key information and let the program fill in the blanks to name the file

2. Manual Naming: enables the user to enter the file name and pathway themselves
3. File Field: displays the file name and pathway
4. Bit Depth: allows the user to specify the desired bit depth
5. Continuous Checkbox: toggles whether or not the user wants the program to record indefinitely
6. Recording Time Field: specifies the number of seconds the program will record for
7. Start/Stop Button: begins and ends the recording
8. Waveform: graphs amplitude vs time of the live recording
9. Spectrogram: graphs frequency vs time of the live recording
10. Status Output: displays updates on the program's activity
11. New Test Subject Button: resets the GUI for a new rat

Terminating the recording session

The program can be closed by exiting the main GUI. However, closing is disabled while the program is actively recording or saving so as to prevent data loss. If you are attempting to close the program and your attempts are failing, check the Status window to ensure that the program isn't saving or recording. If it is, either stop the recording or wait for saving to complete.
The newly recorded .wav files are not backed up automatically; it is recommended that you copy your files to an external hard drive at the end of every recording session to prevent data loss. Please purge your data from the PC regularly to ensure there is always ample memory for new recordings.
The microphone should be turned off between program uses. The RX6 can be left on if no recordings are planned within the next week.

File Field

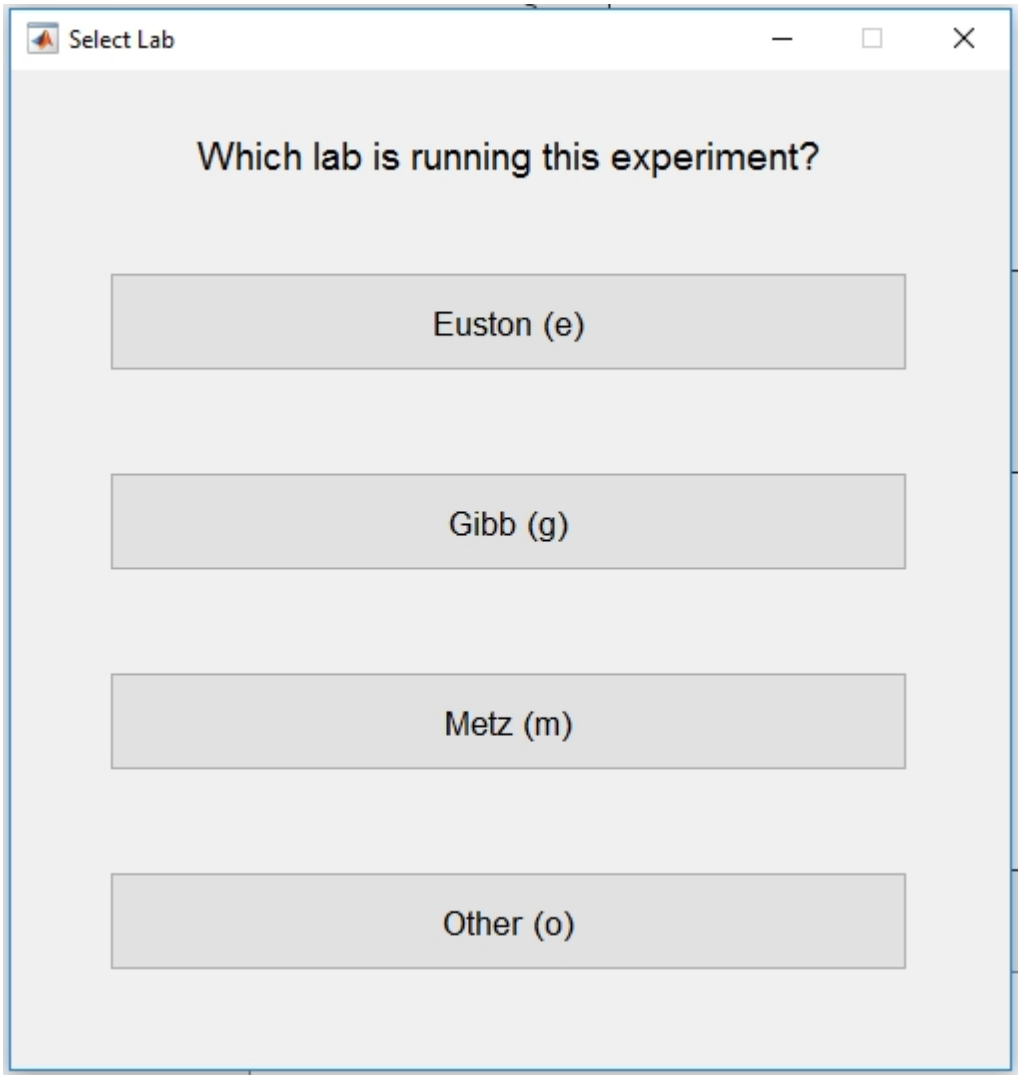


The name of the audio file is displayed at the top of the main GUI. The save name defaults to the C drive when the program is started. (The computer does not allow files to be saved directly to the C drive; a subfolder must be chosen or created.)
A white background indicates that the file name is valid; a red background indicates that the file name is invalid and must be changed before the program allows the recording to be started. A valid name fulfills these criteria:
1. The name ends in .wav
2. The root of the pathway is a drive (eg C:\)
3. The file has a local name (eg filename.wav) within its directory
4. The file is saved in a subfolder instead of the root (files cannot be saved directly to the C: drive)
When "New Recording" is pressed for a new test subject, the local file name is wiped but the current directory is maintained.

Manual naming

The file can be manually named either by pressing "Name File Manually" or clicking directly on the name. Going through Name File Manually allows the user to select the appropriate directory and then type in the file name.
Clicking directly on the name enables the user to type out the save name and location without going through a directory selection window. This method is intended for minor tweaks to the existing pathway and is not recommended as a primary naming technique.

Automatic naming

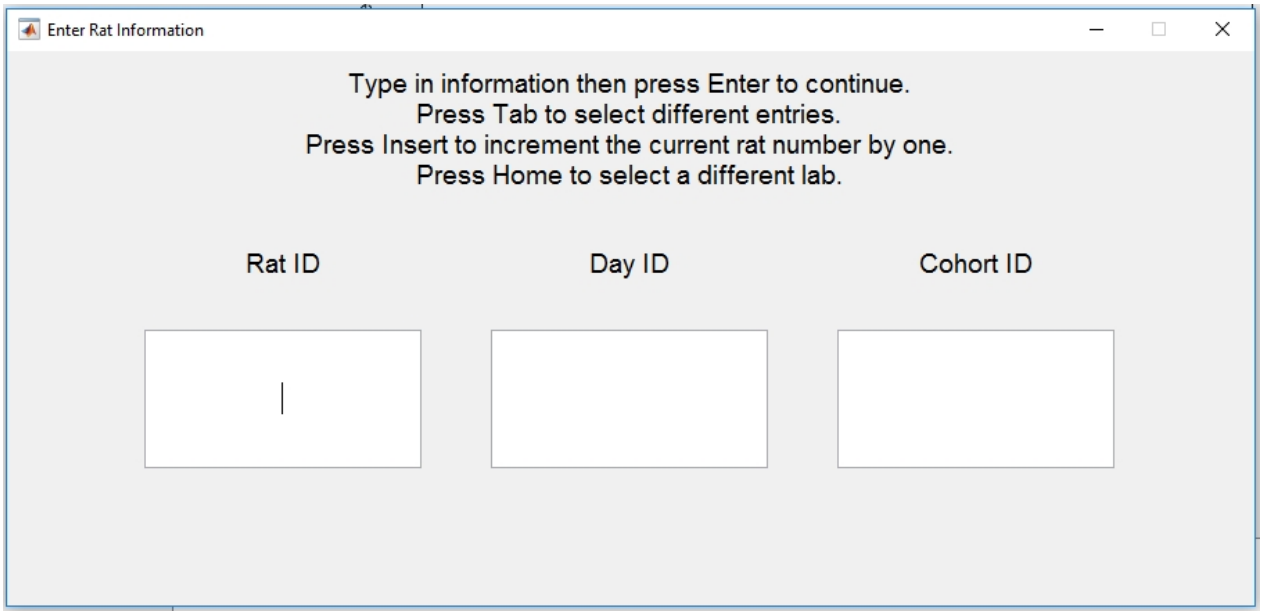


When the Automatic Naming button is pressed, a window opens which offers four options for naming based on the lab which the experiment is being run under: Euston, Gibb, Metz, or Other. The selection determines where the .wav file will be saved and what the naming format will be.

Rather than clicking the buttons, the appropriate lab can be selected by using the following keyboard shortcuts:
e = Euston
g = Gibb
m = Metz
o = Other

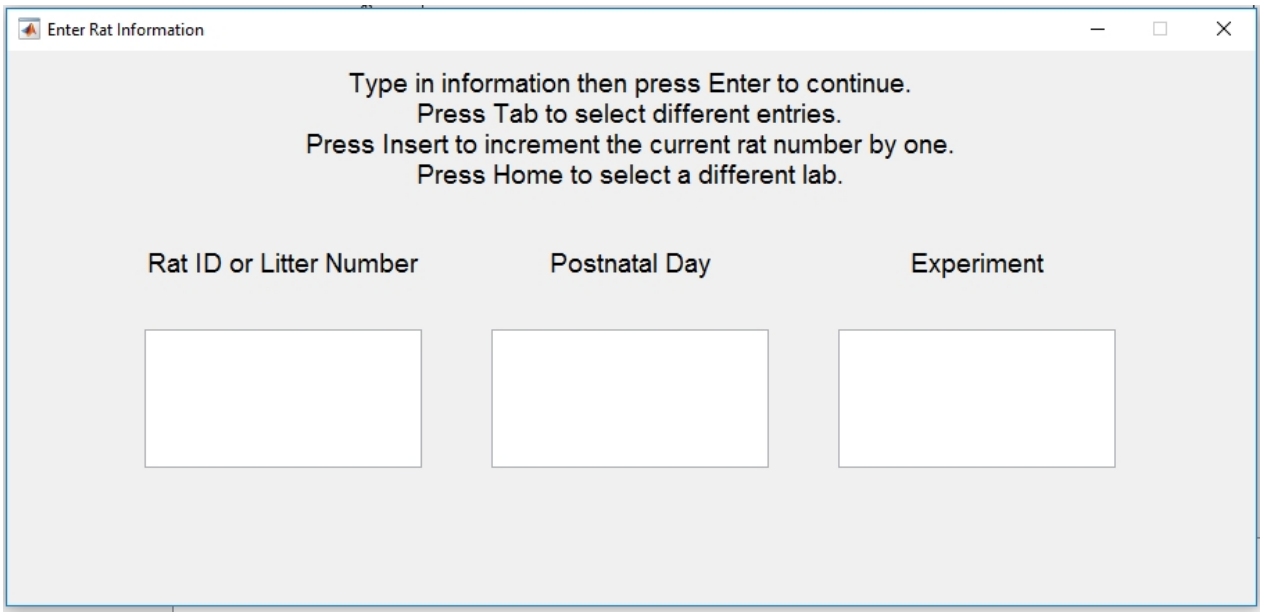
Selecting a lab will bring the user to a new window with entry fields for data relevant to the file name. The user can switch between the fields by pressing 'Tab' and finalize their data by pressing 'Enter'. The program will not allow the user to proceed if one or more fields are empty.
Once a lab has been selected, it is automatically assigned for every recording while the program is open. The lab can be changed by pressing 'Home' to bring back the lab selection screen.
If all data remains the same between back-to-back recordings except for the rat ID number, pressing 'Insert' increments the ID by 1 and keeps all other data identical.

Euston



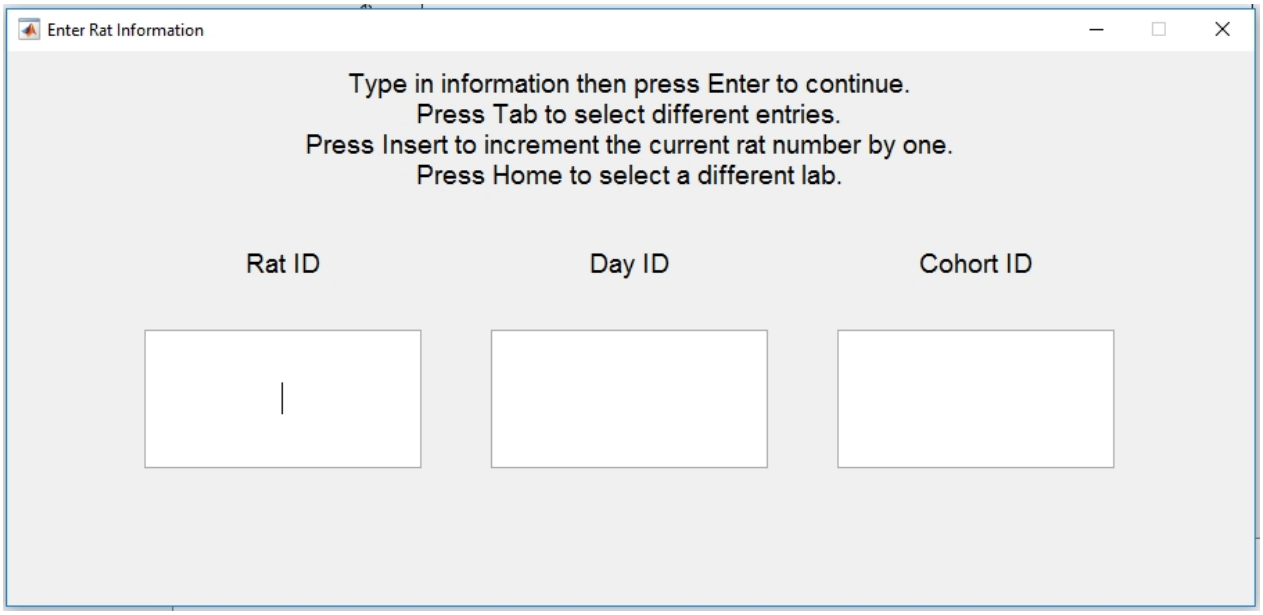
When the Euston automatic naming format is chosen, the entry fields are "Rat ID", "Day ID", and "Cohort ID". Upon entry, the file will be named based on this data and the date and time of recording. For example, if the recording is made on September 3rd, 2017 at 4:30 pm for Rat #63 on Day #8, belonging to Cohort #4, the file will be named R63D8C4_2017-09-03_T16-30.wav.
The file will be saved in C:\Euston\ and will not be segregated into a subfolder.

Gibb

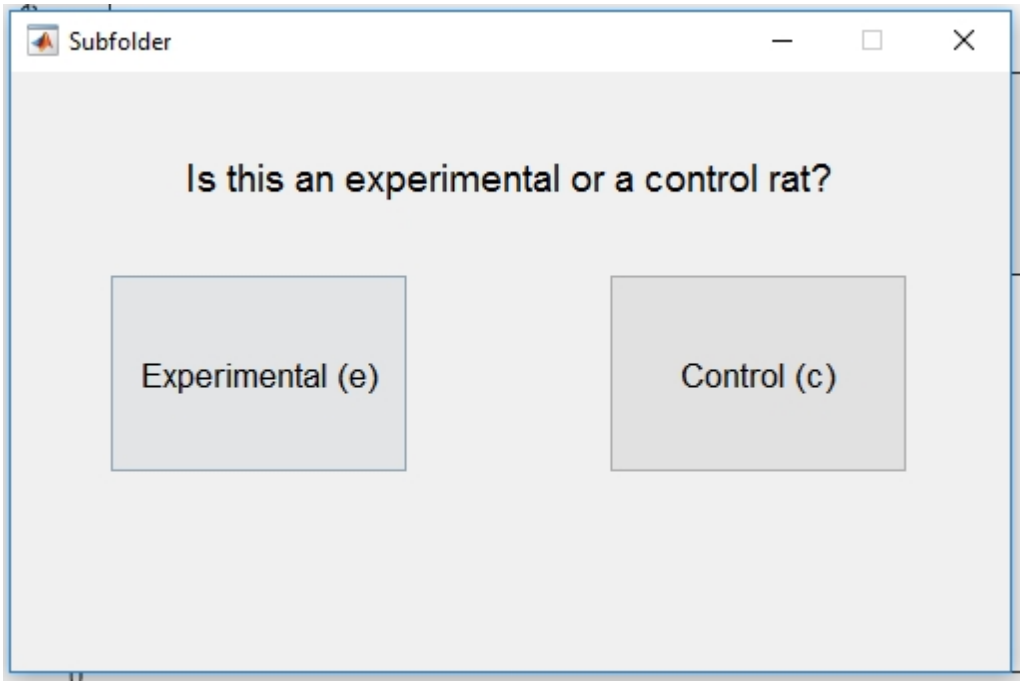


When the Gibb automatic naming format is chosen, the entry fields are "Rat ID or Litter Number", "Postnatal Day", and "Experiment". Upon entry, the file will be named based on this data and the date and time of recording. For example, if the recording is made on September 3rd, 2017 at 4:30 pm for Rat or Litter #63 on Postnatal Day #8, in Experiment #4, the file will be named ID63P8Ex4_2017-09-03_T16-30.wav.
The file will be saved in C:\Gibb\ and will be segregated into a subfolder based on the experiment. In the example above the file would be saved to the directory C:\Gibb\Experiment4\.

Metz



When the Metz automatic naming format is chosen, the entry fields are "Rat ID", "Day ID", and "Cohort ID". Upon entry, the file will be named based on this data and the date and time of recording. For example, if the recording is made on September 3rd, 2017 at 4:30 pm for Rat #63 on Day #8, belonging to Cohort #4, the file will be named R63D8C4_2017-09-03_T16-30.wav.



Upon data entry, another window will open asking whether the rat belongs to an experimental or control group.
Rather than clicking the buttons, the appropriate group can be selected by using the following keyboard shortcuts:
e = Experimental
c = Control

The file will be saved in C:\Metz\ and will be segregated into either the Experimental or Control subfolder depending on the selection in the second window.

Other

Other is used by labs other than Euston, Gibb, or Metz. The file naming format follows the same method as Euston, but files are saved in C:\Other\.

Continuous vs Timed Acquire

By toggling the Continuous checkbox, the user can determine whether the recording will time out automatically after the number of seconds specified by the Recording Time field, or will run until the the Stop Recording button is pressed.
When Continuous is not toggled, the user can set the length of the recording by typing in the number of seconds in the Recording Time field. Recording is not permitted when an invalid time is entered, as marked by a red field background. A valid time must be a natural number (a whole number greater than 0).

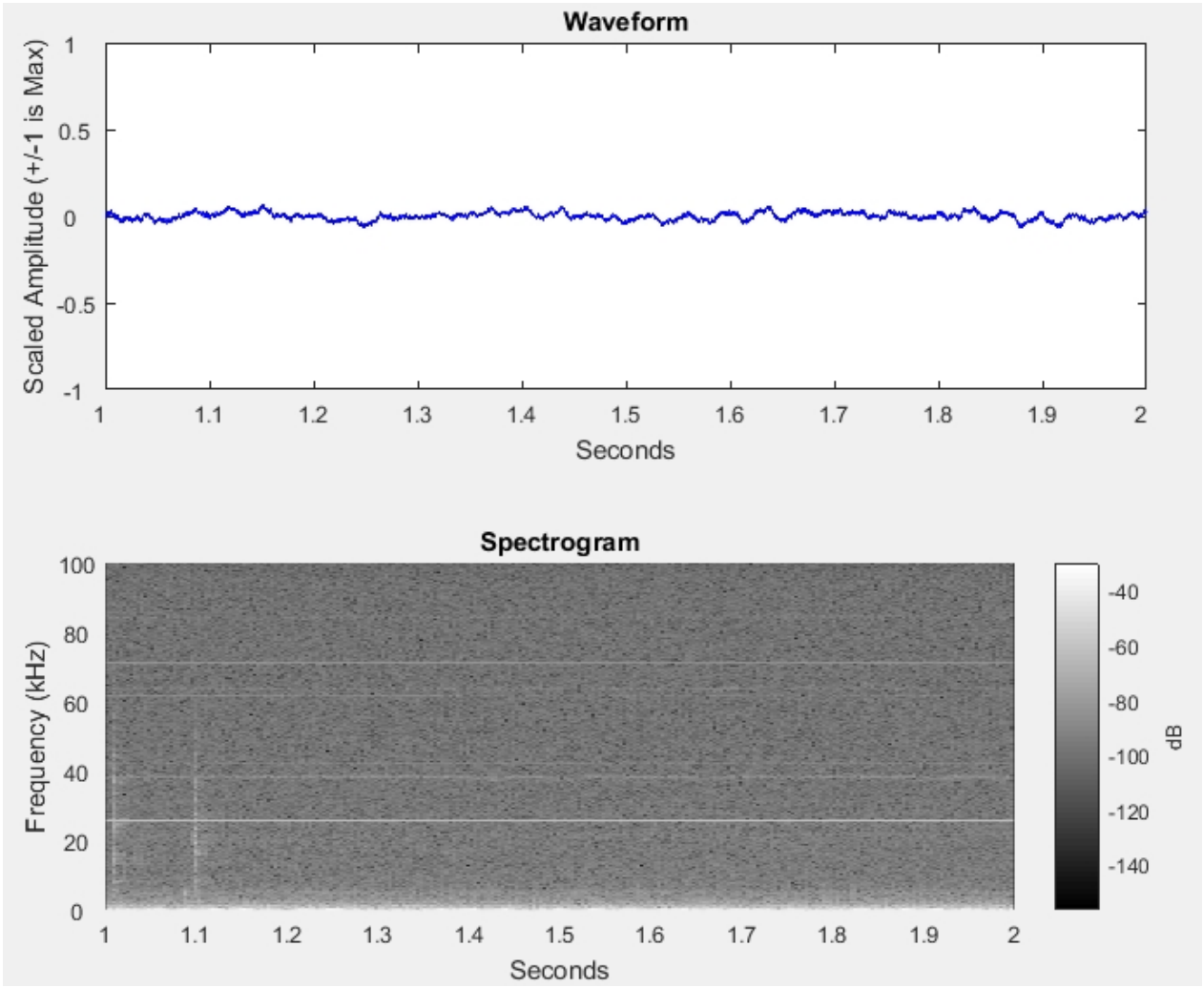
Storage Fidelity

The user has three bit depth options: 32-bit int, 24-bit float, or 16-bit float. All three are compatible with Raven Pro. Higher bit depth settings can record smaller differences between sounds, but require more memory.
Audio files can be downsampled after recording, but never upsampled, so it is recommended that users choose 32-bit unless they are certain that a lower bit depth fulfills their requirements.

Start/Stop

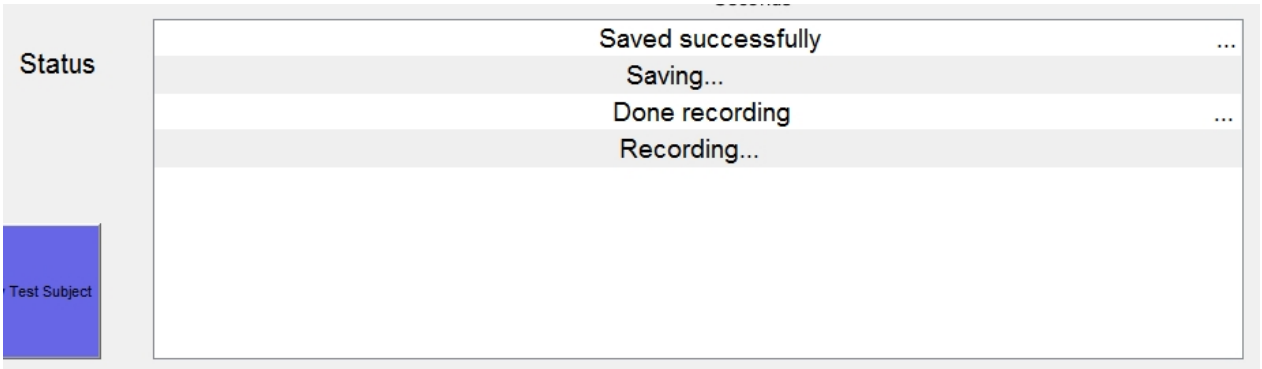
Pressing the green "Start Recording" begins the acquisition of audio, which will display in the Waveform and Spectrogram axes and write "Recording..." in the Status Output. The button will change to the red "Stop Recording" button which can be pressed at any time to halt acquisition, even when the timer has not fully elapsed in the case of non-continuous recording.

Waveform & Spectrogram



Data from the audio recording is displayed to the waveform and spectrogram axes one second at a time to provide live feedback of input.
The y-axis on the waveform plot is scaled to the maximum range of the microphone. If at any point the graph breaches the top or bottom of the graph, clipping is occurring and the settings on the microphone must be adjusted to a lower dB level.
The spectrogram displays vocalizations up to 100 kHz in frequency. Due to limitations in the processor, frequencies above 100 kHz cannot be represented in the recording.

Status Output



The status output field displays information for the user. A typical recording session produces the four lines of output shown above. This is also where warnings and error messages appear.

New Test Subject

After a recording has been completed and saved, pressing the New Test Subject button will reset elements of the program to enable a new recording. After one recording is complete, the program will not allow the user to press "Start Recording" again until it is initialized for the new recording task. The current directory is maintained, but the local file name is erased.

Errors

Following are some problems that you may encounter and error messages that may appear in the Status output. If you discover any glitches or problems with the program that are not addressed in this section, please contact Erica Nordin at erica.nordin@uleth.ca with details about the encountered issue.

"Invalid field prevents record"

Either the recording time or the file name is invalid. Check which one is coloured red and make appropriate changes.

"Transfer rate is too slow"

The program is not saving the audio fast enough to keep pace with data acquisition. Close all programs aside from MATLAB to ensure maximum processing power is diverted to the program.

"RX6 not connected"

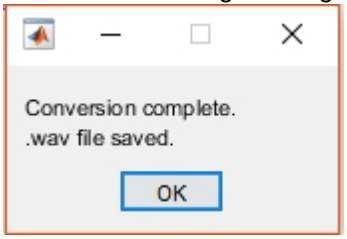
The program is unable to load the appropriate circuits onto the processor. Check that the RX6 processor is turned on and all connecting cords are securely in place. Run the zBUSmon procedure to ensure that the the PC is properly interfacing with the RX6. If the error continues, turn off the RX6 and microphone, restart the PC, and begin the start up sequence from scratch.

"Microphone not connected"

The program is not receiving input from the microphone. Check that the microphone is turned on and all connecting cords are securely in place.

Program or computer crashes during recording or before saving is complete

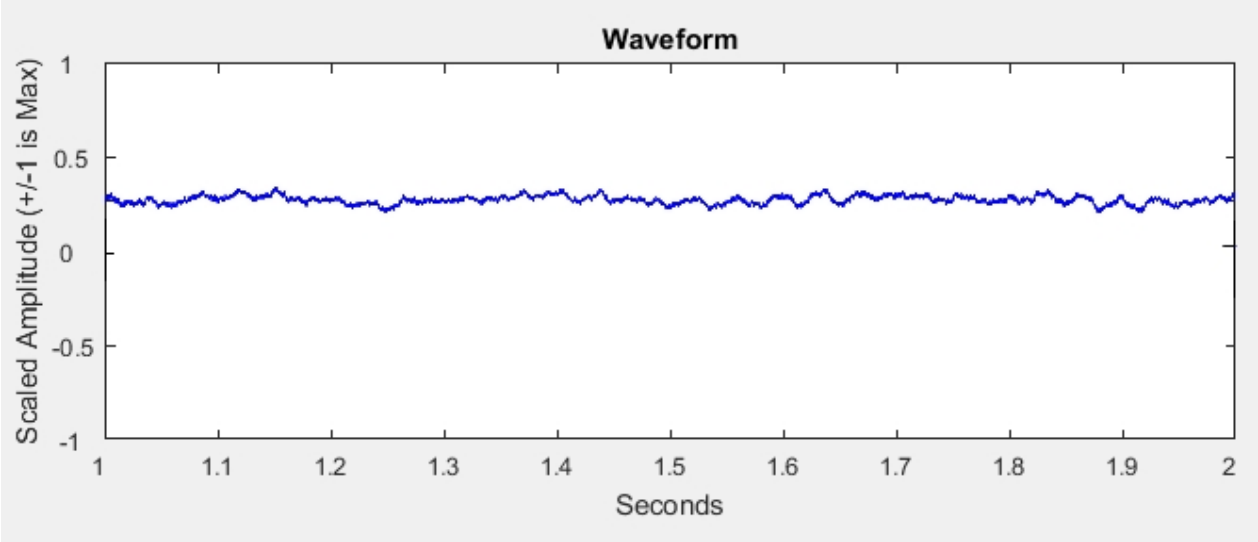
In the case of a crash, the program will not have produced a .wav file. However, it will have produced a binary .F32 file which contains the data recorded up to the crash point and has the same name and location as the .wav file which was supposed to be produced. Open the BinaryToWav program at C:\Ultrasonic Recording Program\BinaryToWav and select your .F32 file when the file selection window appears. Another window will open allowing you to choose your bit depth.
The program will convert the .F32 file into a .wav file with the bit depth appended to the name (e.g. test.F32 will become test_24bit.wav if 24 bit float is selected). The .wav file is saved in the same location as the .F32 file and the following message is displayed upon completion:



To prevent data loss, the program does not automatically delete the .F32 file; this must be done manually by the user once they are confident that they have recovered their data.

Note that the BinaryToWav program assumes that your recording was done with a sampling rate of 195312 Hz, which is standard for the TDT RX6. If your recording was done without this particular processor, BinaryToWav will not convert it properly.

DC offset



DC offset can be recognized by the mean value of the waveform amplitude shifting to be above or below 0. DC offset can result in unexpected audio clipping errors. If you observe DC offset when you begin a recording, stop the recording and reset the RX6 by following the zBUSmon procedure.

PC Boot Failure

You may encounter a critical failure upon booting the computer reading "Attempted switch from DPC" which causes the computer to reboot and start normally on the second boot. This failure has been determined to be an issue with the TDT PCI card. Unfortunately, the PCI card is fundamental to the hardware, and is very expensive to replace. Given that no other issues seem to have surfaced as a result of the card, and the critical failure is immediately rectified by a reboot, the developers have decided to ignore this issue for now. If this problem expands, the PCI card will likely have to be replaced.

Change Bit Depth of Recording

If a recording is completed with a lower bit depth than is necessary, then unfortunately nothing can be done to recover the lost data. However, if the bit depth of a recording is too high (usually due to memory constraints or incompatible programs), the bit depth can be reduced using the DownsamplerForWAVs executable located in C:\Ultrasonic Recording Program\Downsampler. The downsampler supports all of the three bit depth options available in the recording program: 32 bit int can be reduced to 24 or 16 bit float and 24 bit float can be reduced to 16 bit float. 16 bit files cannot be downsampled.
The program will not overwrite the existing WAV file; it will append the new bit depth to the file name and create a new WAV file in the same location.
The existence of the Downsampler allows researchers to record with maximum fidelity (32 bit) and reduce the size of the files later. Thus, it is recommended that the 32 bit setting be chosen when recording.

Fresh Install

If the computer dies, the hard drive is upgraded, or anything else happens to disrupt the executable or the TDT, some or all of the aspects of the recording software and/or hardware will have to undergo a fresh install. Follow the instructions in this section for set-up. Note that depending on the event that necessitated the set-up, all or some of these steps may be necessary. Evaluate what needs to be done before executing all these steps.

TDT Hardware and Software

External Hardware

The PC has a series of hardware extending from it: the fiber optic cables that plug into the computer attach to the TDT processor, which receives audio from the SoundConnect Microphone Power Supply, connected to the collecting microphone. The fiber optic cables should NOT be bent beyond a gentle curve. See http://www.tdt.com/rx6-multifunction-processor.html for details about the processor and https://www.listeninc.com/wp/media/SoundConnectManual_20150430.pdf for details about the microphone power supply.

PCI Card

The TDT processor interfaces with the PC via a specialized PCI card, which the fiber optic cables connect to at the back of the PC.
Note that the currently installed card causes some issues upon booting the computer, and may have to be replaced in the future. See http://www.tdt.com/files/manuals/Sys3Manual/Optibit.pdf for information about the PCI card and its installation.

Associated Software

Upon installation of the PCI card, drivers for the card must be downloaded from http://www.tdt.com/downloads.html. The key programs RPvdsEx and zBUSmon should be included in the driver download; if not, they must be downloaded separately from the TDT website. The processor's microcode may have to be updated when downloading new drivers. See http://www.tdt.com/files/fastfacts/Microcode.pdf for instructions.
Additional ActiveX Controls (also available via the TDT download link) are required to interface with MATLAB. Unlike the drivers, the ActiveX Controls are password protected. The password to enter when prompted during installation is "spider".

RPvdsEx Circuit File

The RPvdsEx program converts .rcx circuits to assembly code and loads the code onto the TDT processor to direct its audio collection. The AcquireAudio file, located at C:\Ultrasonic Recording Program\AcquireAudio.rcx is the software interface between Ultrasonic_Recording_Interface and the processor.

Ultrasonic Recording Program Directory

The C: drive must contain a folder named "Ultrasonic Recording Program" containing a variety of key files. All files except the manual are located in the "Executables and Key Files" folder in the source code repository; the manual is located in the "Manual" folder.

To summarize, the "Ultrasonic Recording Program" directory should contain the following:
1. The folder "GUI", containing the Ultrasonic_Recording_Interface executable and the text files mccExcludedFiles, readme, and requiredMCRProducts
2. The folder "RecoverAudio", containing the BinaryToWav executable and the text files mccExcludedFiles, readme, and requiredMCRProducts
3. The folder "Downsampler" containing the Downsampler executable and the text files mccExcludedFiles, readme, and requiredMCRProducts
4. AcquireAudio.rcx
5. The latest version of the pdf-published manual.

MATLAB Runtime 9.2

In order to run compiled MATLAB programs, MATLAB Runtime 9.2 must be installed on the PC. Unlike most MATLAB features, Runtime does not require either a license or the basic MATLAB package to function.

Modifying the Program

The executables on the recording PC available to the users has been compiled from MATLAB code. In order to change the user interface or its background functionality, the source code must be modified and recompiled.

Source Code

The source code can be retrieved from https://github.com/ericanordin/TDTadaptor. The code can be viewed by anyone, but only the account holder can directly modify this repository; the repository must be cloned or downloaded from GitHub to make changes.
The code is under version control with Git, meaning that different versions of the program throughout the history of development can be retrieved without saving multiple copies of the files. The original program author recommends that version control be maintained if the program is changed. For more information about using Git for version control, see https://git-scm.com/book/en/v2/.

Development Environment

The program was developed using MATLAB R2017a. Attempting to edit the program on older or newer versions of MATLAB may yield issues with built-in MATLAB functions.
The compiled executable requires MATLAB Runtime to function. Runtime versions correspond to regular MATLAB versions, so if changing the development environment from R2017a to a more recent MATLAB version, be sure to change the Runtime version on the recording PC from 9.2 to the appropriate version.

Compiling

Compiling requires a license for the MATLAB Compiler add-on. You may contact MATLAB to purchase this, or track down the PC in the CCBN which already has the compiler installed. At the time that this manual was published, Clifford Donovan (c.donovan@uleth.ca) was the primary user of this PC, located in EP1255. Note that executables produced with a trial version of the Compiler expire when the trial does, so acquiring a trial version will not suffice.

The TDTadaptor root directory contains the file compileScript.m which lists the commands that must be run to produce the executables. These commands must be modified slightly between PCs to reflect the local save locations for the executables.

Probable Modifications

Many aspects of the program can be changed. The author has provided some guidelines for elements that she judges most likely to be adjusted.

Labs

The program contains an enumeration for various labs at the CCBN to enable custom file naming and organization for each one.

The following are the files that may have to be modified when changing the available labs or what the program does with a lab:

1. The enumeration is defined in +Enums\LabName.m
2. The GUI that allows the user to choose the lab is created in +GUIFiles\LabScreen.m
3. The function that builds the save location is +StandardFunctions\makeLabDirectory.m
4. The function that builds the local name for the WAVE file is +StandardFunctions\setNameAuto.m
5. The GUI that allows the user to enter rat data is created in +StandardFunctions\RatScreen.m, and must be modified if the automatic naming procedure is adjusted.

Adding a new lab

A new lab can be introduced by adding the PI's name to the LabName enumeration. In order to allow the user to choose the new lab, LabScreen must be modified to add a push button corresponding to the lab.
If WAVE files are to be saved into subfolders within the lab directories, instructions must be introduced to makeLabDirectory to build the appropriate subfolder.
If the default local file naming convention (the convention used for lab name "Other") is not desired, RatScreen and/or setNameAuto must be modified; see section "Changing automatic naming" for details.

Changing automatic naming

RatScreen allows the user to enter information about the rat. What the entry fields correspond to and which fields are available can be modified from the RatScreen GUI file.
Note that RatScreen currently assumes three entry fields and adjusts what they are labeled as based on the lab name. Changing the labeling can be done easily by changing the SetLabels function within RatScreen; changing the number of entry fields would involve a more in-depth adjustment.
Information entered in RatScreen is imported to setNameAuto to build the local file name. Modify setNameAuto to introduce the appropriate naming conventions for a lab.
Aspects of the naming convention correspond to the labels assigned in RatScreen. Generally speaking, if one of RatScreen and setNameAuto are changed, the other must also be changed.

Changing Recording Bit Depths

Currently, the program is set up for recording Float 16, Float 24, or Int 32 WAVE files. To allow the program to handle other bit depths requires the modification of not just the main program, but also the mini-programs used for modifying files after recording is complete.
The following files must be changed when introducing a new bit depth:

1. The bit depth enumeration is defined in +Enums\SaveFormat.m
2. The main GUI is created in +GUIFiles\RecordScreen.m
3. The Downsampler GUI is created in 'Post-Record Processing\DownsamplerGUI.m'.
4. The BinaryToWav GUI is created in 'Post-Record Processing\BinaryToWavGUI.m'.

Because the program records audio to a binary .F32 file then converts it to a WAVE file, the code governing background recording does not need to be modified when adding new bit depths.

Main program

The new bit depth must be defined in the SaveFormat enumeration, and appropriate scaling and corresponding numerical value defined in SaveFormat's functions.

RecordScreen interacts with SaveFormat via the SelectBitDepth function within the RecordScreen constructor. A new radio button must be added to bitDepthGroup and its interaction with SaveFormat must be defined in SelectBitDepth.

Downsampler

The DownsamplerGUI must be changed to reflect the new bit depth. The appearance of the GUI varies based on the bit depth of the input file to offer only the lower bit depths as options, so the new bit depth must be introduced as a push button to each higher bit depth GUI. In addition, a new GUI must be built corresponding to the new bit depth.

Depending on the nature of the change, 'Post-Record Processing\Downsampler.m' may have to be adjusted with its GUI.

BinaryToWav

A push button corresponding to the new bit depth must be added to the BinaryToWavGUI.

Depending on the nature of the change, 'Post-Record Processing\BinaryToWav.m' may have to be adjusted with its GUI.

Modifying the Manual

The manual was created and can be modified using the HelpNDoc program, which allows the developer to create sections of the manual and export the manual to a variety of formats including PDF and Word. To download HelpNDoc, visit https://www.helpndoc.com/download.
The template used to format the manual is located in 'Manual\HelpNDoc Template\pdf\'. HelpNDoc will not automatically access this format; directions for introducing the format into the HelpNDoc program can be found as 'Manual\HelpNDoc Template\Instructions.txt'.