Willow 1.0 User Guide

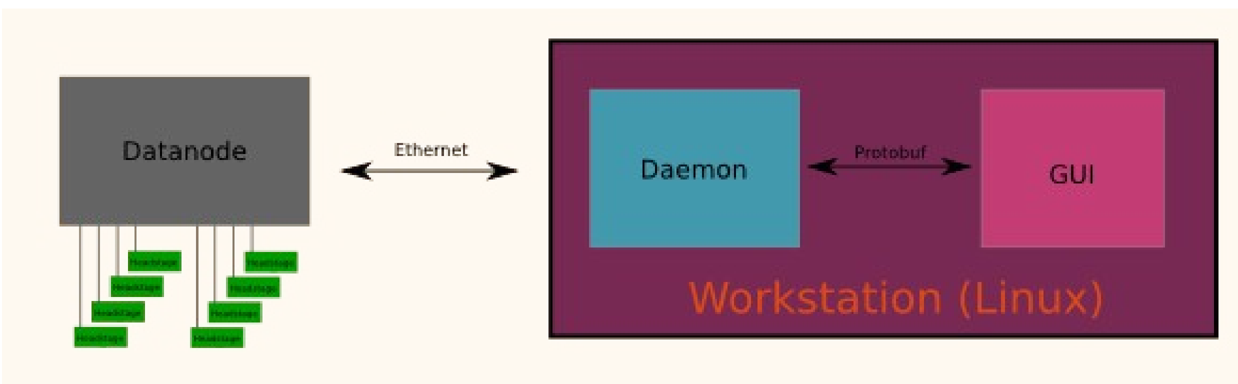
About Willow

Willow is an electrophysiology system developed by LeafLabs in collaboration with MIT's Synthetic Neurobiology Group. The system is based on the Intan RHD chips, and utilizes a scalable, direct-to-disk architecture to efficiently record 1024 channels of neural data.

This document explains the setup and usage of the Willow system, as of the 1.0 release, Spring 2015. The latest release of Willow is supported by the website:

www.scalablephysiology.org/willow

System Overview



The Willow system is comprised of both hardware and software components. On the hardware side, there is the *datanode*, which is the main data acquisition and storage computer. The datanode collects data from the *headstages*, which are modular amplifier and digitizer boards. Each headstage acquires from 128 channels, and each datanode can collect from as many as 8 headstages. The datanode is driven by the *workstation* - a computer running Linux**. The datanode and the workstation communicate over a high-speed ethernet connection.

The workstation itself runs a multi-layered software stack. The *daemon* (leafysd) handles the low-level communication with the datanode, data collection, routing, and I/O. The daemon exposes a control interface based on Google's *Protocol Buffers* (protobuf). The *GUI software* (willowgui) wraps this control interface in some higher-level functions that allow the user to record, stream, an analyze data with the convenience of a graphical user interface.

***As of Willow 1.0, Linux is the only supported operating system.*

Software/Networking Setup

If you purchased a pre-configured Willow system from LeafLabs, or if your workstation has already been set up for Willow, you may skip to the next section: Hardware Connections.

This guide assumes a apt-based Linux distribution (Debian, Ubuntu, Mint, etc.) with standard packages like git and Python 2.7 installed.

First, download the daemon source code:

$ git clone https://github.com/leaflabs/leafysd.git
$ cd leafysd

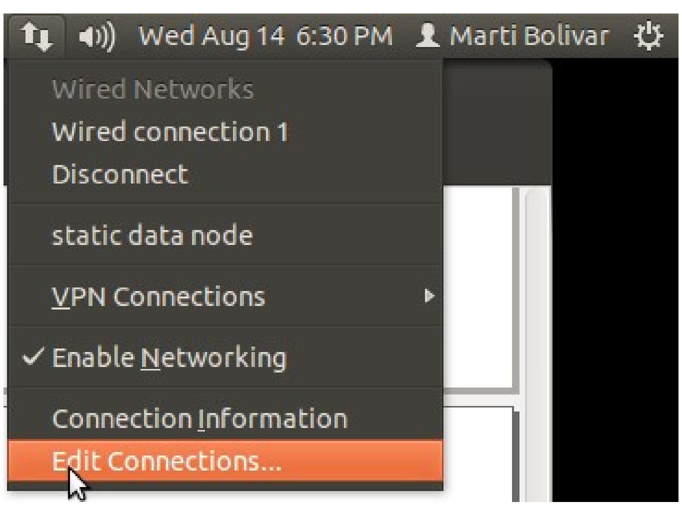
Open leaysd/README.txt, and install all mandatory dependencies. Then, compile the daemon using scons:

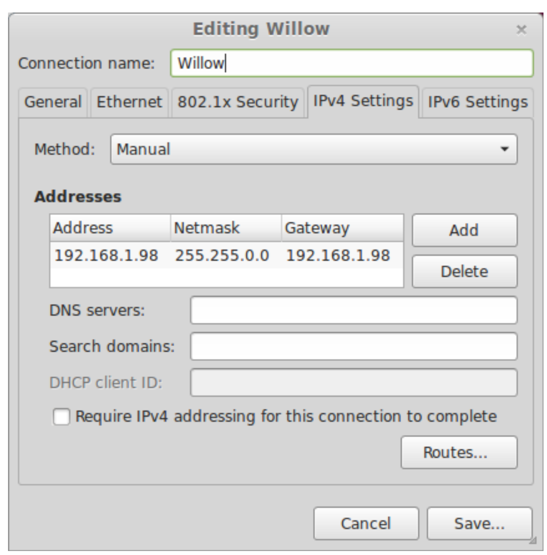
$ cd leafysd
$ scons EXTRA_FLAGS='-O2'

The resulting executable should be in build/. Next, set up your network interface for Willow. Start Network Connection Editor, either from a terminal:

$ nm-connection editor

or (on Ubuntu) by clicking "Edit Connections" in the Network Connections dropdown in the menu bar. Add a new Ethernet connection, and call it "Willow". In the IPv4 Settings tab, set the "Method" to "Manual", then add a new address with Address=192.168.1.98, Netmask=255.255.0.0, and Gateway=192.168.1.98. Click "Save", and close the Connection Editor.





Next, download the GUI software:

$ git clone https://github.com/leaflabs/willowgui.git
$ cd willowgui

and open src/config.json in a text editor:

$ gedit src/config.json

In your editor, find the "daemonDir" parameter, and change its value to point to the location of the daemon install (top-level leafysd directory). For example:

```
"daemonDir" : {
   "type" : "str",
   "description" : "Daemon Directory",
   "value" : "/home/chrono/neuro/willow1.0/leafysd"
}
```

Similarly, modify the "dataDir" parameter to point to an existing directory where data files will be saved by default. Leave the other configuration parameters in their default state for now – you can modify them later from within the GUI. Save your modified config.json in src/.

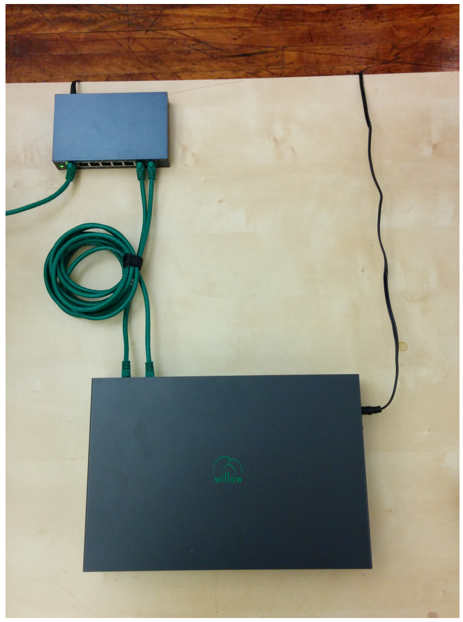
This completes the setup process for the Willow software.

Hardware Connections

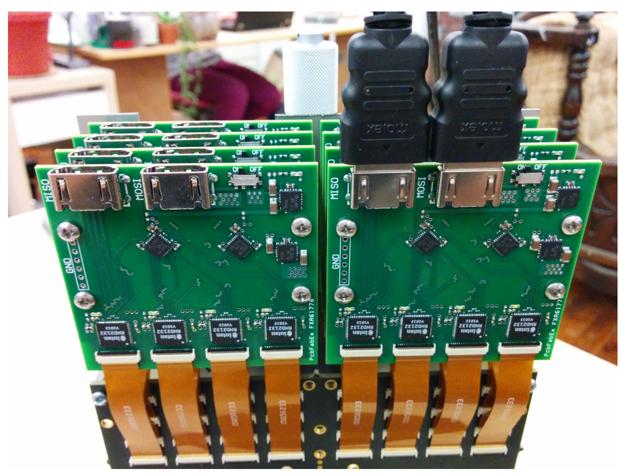
Follow these instructions to set up the Willow Hardware:

1. Connect the Willow Power Supply to port labeled 12VDC on right side of Datanode.

2. Connect TCP/UDP ports to network switch using two Ethernet cables.

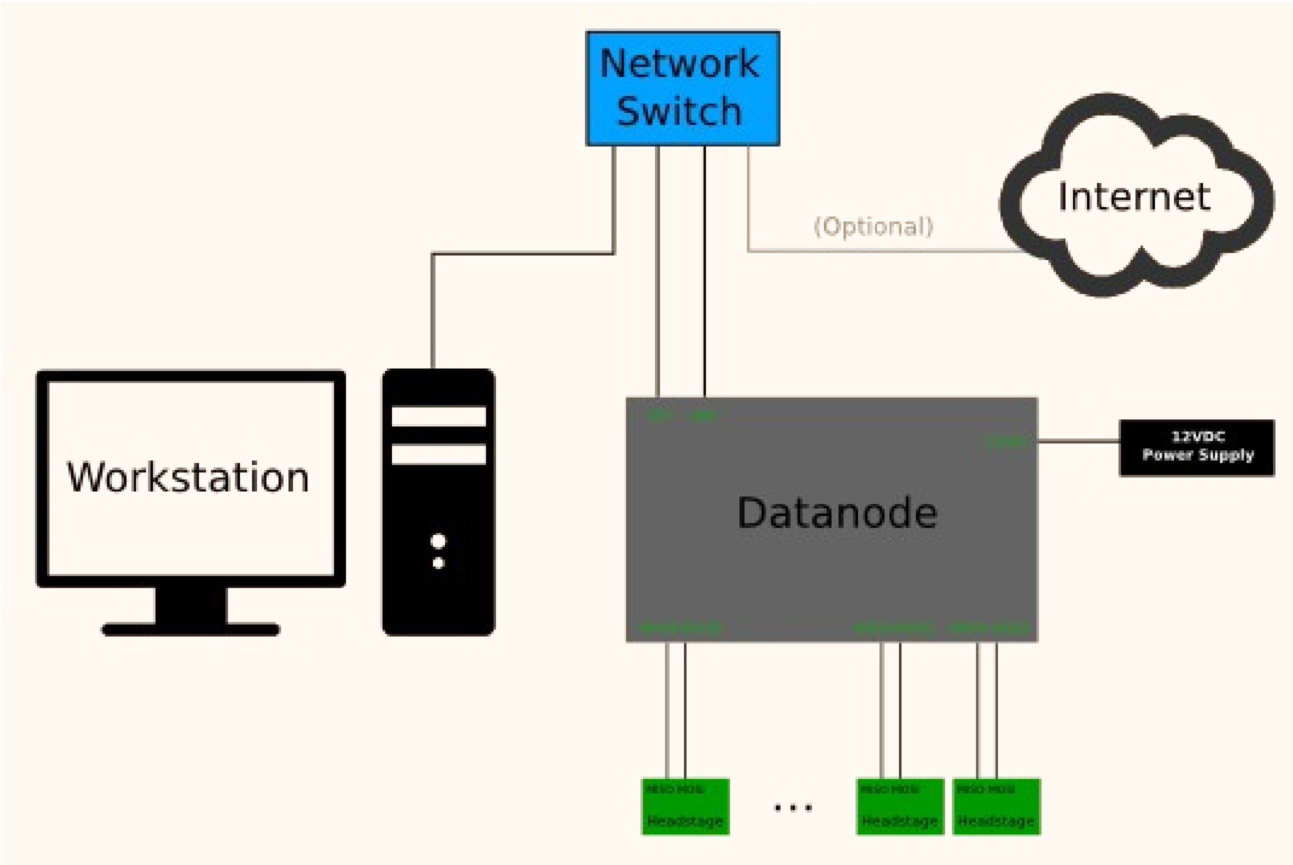


3. Connect workstation to the same network switch via Ethernet cable.
4. Connect Datanode to Headstages via HDMI cables. Connect any Datanode MISO/MOSI paired-ports to respective MISO/MOSI paired-ports on any Headstage. Repeat desired number of paired-port connections between Datanode and Headstages, as needed.



5. Slide power switch on side of Datanode to ON position to power Datanode. Wait until white LEDs on Headstages flash twice (takes about 15 seconds).

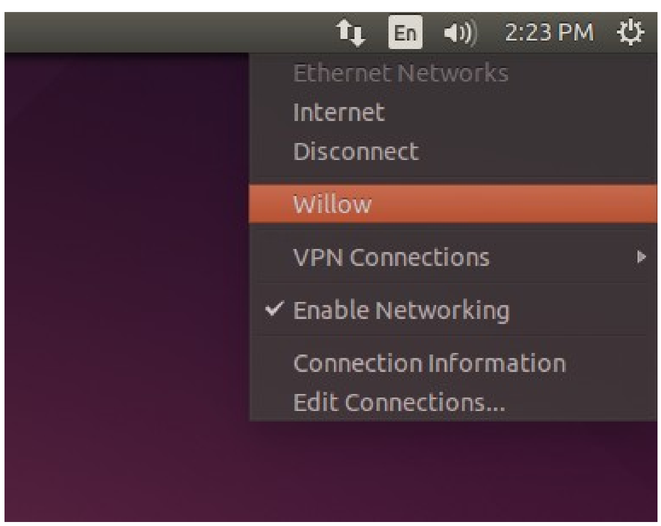
At this point, the connection topology should look as in the figure below, and you are ready to start the GUI software.



GUI Startup

Follow these instructions to start up the GUI software:

1. Select "Willow" from your Network Connections.



2. Expand the MTU for the current network interface. A utility script to accomplish this is included in leafysd/util:

   $ cd leafysd/util
   $ ./expand_eth_buffers.sh <interface>

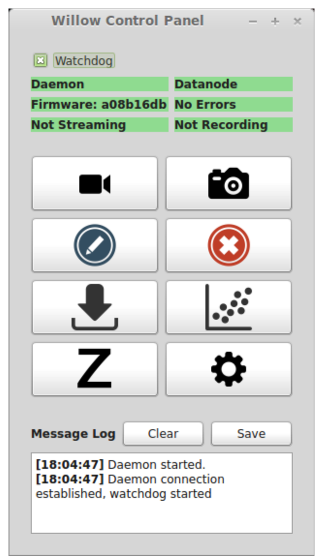
Where <interface> is the name of your network interface (default is eth0). Note that this script requires sudo priveleges.

Note: If you're using a pre-configured Willow workstation, this happens automatically upon startup.

3. To start the GUI:

   $ cd willowgui
   $ src/main.py

This should bring up the main GUI window (shown below, right). If set up is correct, the six status bars at the top will display green indicating no errors.



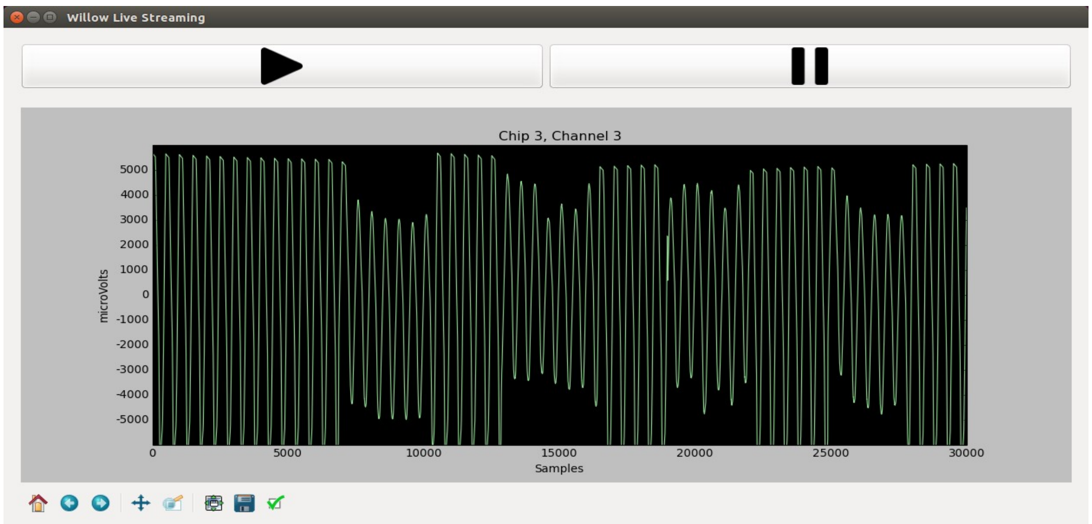
Note: On a pre-configured Willow workstation, you can start the GUI by clicking the "Willow GUI" icon on the Unity Launcher side panel (above left).

Basic GUI Usage

This section is offered as a quick-start guide to basic GUI usage. For more in-depth documentation, please refer to the user guide distributed with the GUI repo (willowgui/docs/user_guide).

Stream live data: 

Click this button to launch a *Stream Window*. A dialog box will appear in which channel number, plotting range, and frame rate can be entered. After clicking “OK”, the *Stream Window* will appear. Start/stop the stream by clicking the “play” and “pause” buttons.



Take a snapshot: 

Click this button to take a *snapshot*, which is a short (1-10 second) sampling of all 1024 channels, stored in an HDF5 data file. A dialog box will appear in which number of samples to collect and the target filename can be entered. By default, the snapshot will be 1 second long (30000 samples) and have a timestamp-specific filename of the form:

s**napshot_YYYYMMDD-hhmmss.h5**.

Selecting “Plot When Finished” to open the snapshot data in a Plot Window (see below).

Start recording: 

Click this button to start recording to the Datanode. **WARNING**: Each recording will start at the beginning of the disk, overwriting any previously recorded experiments. To ensure that no data is lost, make certain that any important experiments have been transferred off the Datanode (see *Transfer Experiment* below) before recording. While recording, the recording label in the status bar will turn red and list the current disk usage.

Stop recording: 

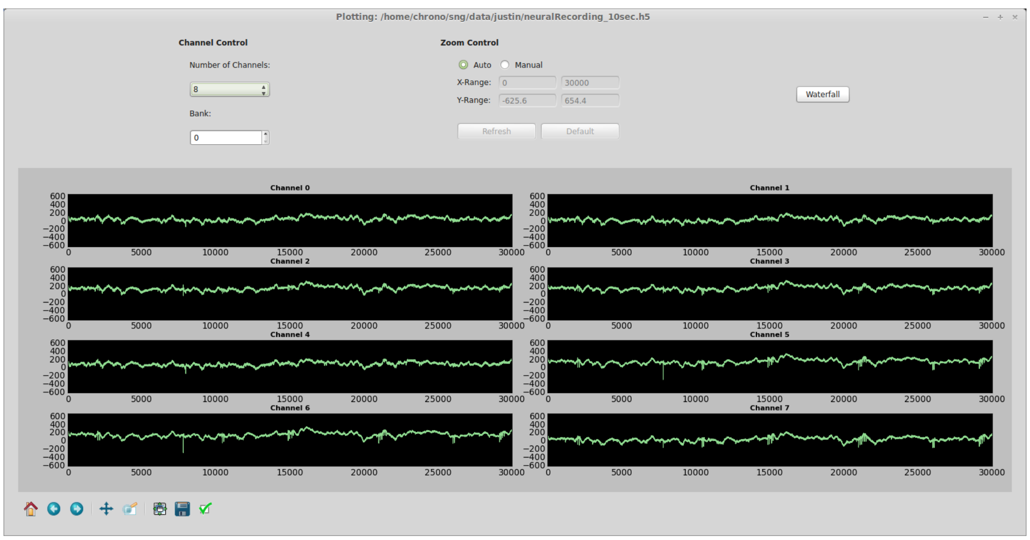
Click this button to stop recording. The 'recording' status bar will turn back to its green “Not Recording” state.

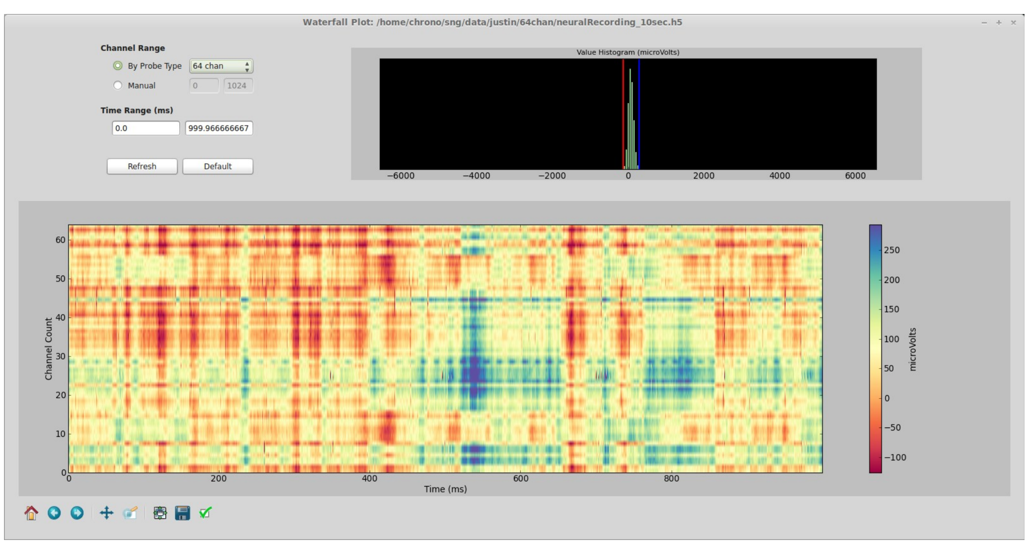
Transfer experiment: 

Click this button to transfer an experiment (i.e. previous recording) to the workstation. A dialog box will appear in which the length of the experiment to transfer (or entire recording), and the target filename can be selected. Select “Name Automatically” to name the file with the UNIX time-stamp from when the recording was started.

Plot data: 

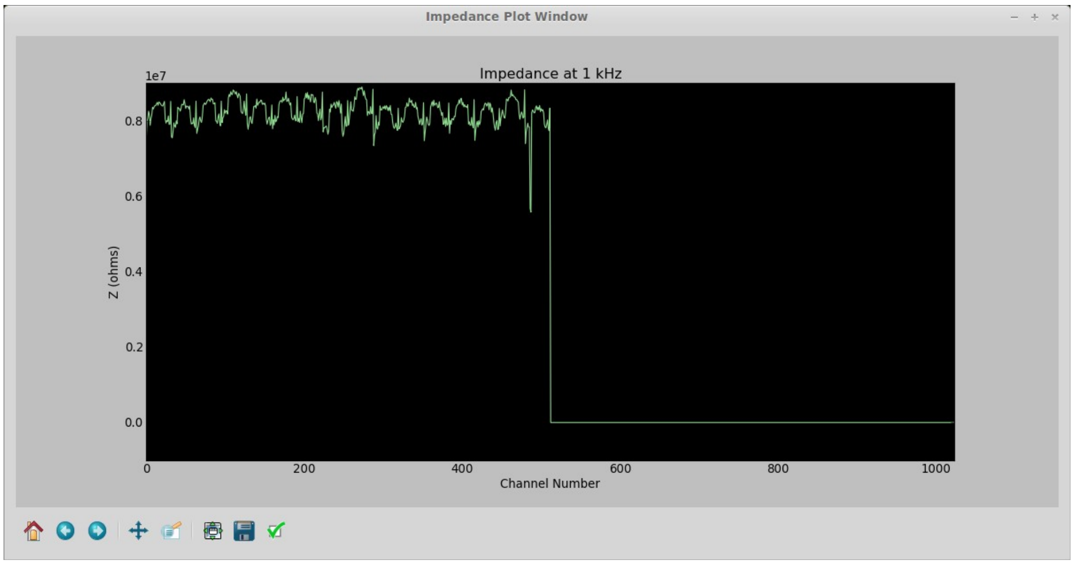
Click this button to plot data from a previously acquired recording or snapshot. A file browser will appear in which the desired file can be selected. A subsequent dialog box will appear in which the amount of data desired for import can be selected. Upon selection, a *Plot Window* will open in which channel traces can be viewed as line plots. Click “Waterfall” near the top-right of the Plot Window to open a *Waterfall Plot* – a 2D spectrogram-like visualization with channel count on the *y*-axis, and time on the *x*-axis.





Measure Impedance: Z

Click this button to measure the impedance of the probe connected to the headstages. A dialog will appear where you can choose to measure the impedance of a single channel, or of all channels in sequence. If you choose “Single Channel”, the process will take about 10 seconds, and the result will be posted on the message log. If you choose “All Channels”, the loop will run for about 3 minutes, and the result will be saved in a time-stamped .npy file, also reported in the message log. In “All Channels” mode, you also have the option to “Plot When Finished”, which will bring up a plot window showing the impedance results across all channels (see Figure 10). All impedance measurements are taken at f = 1 kHz.



Configure Settings:

Click this button to set the preferences for the GUI software. Global parameters like the default data directory, memory settings, and the datanode storage capacity can be modified from this dialog. Changes will persist after you close the GUI.