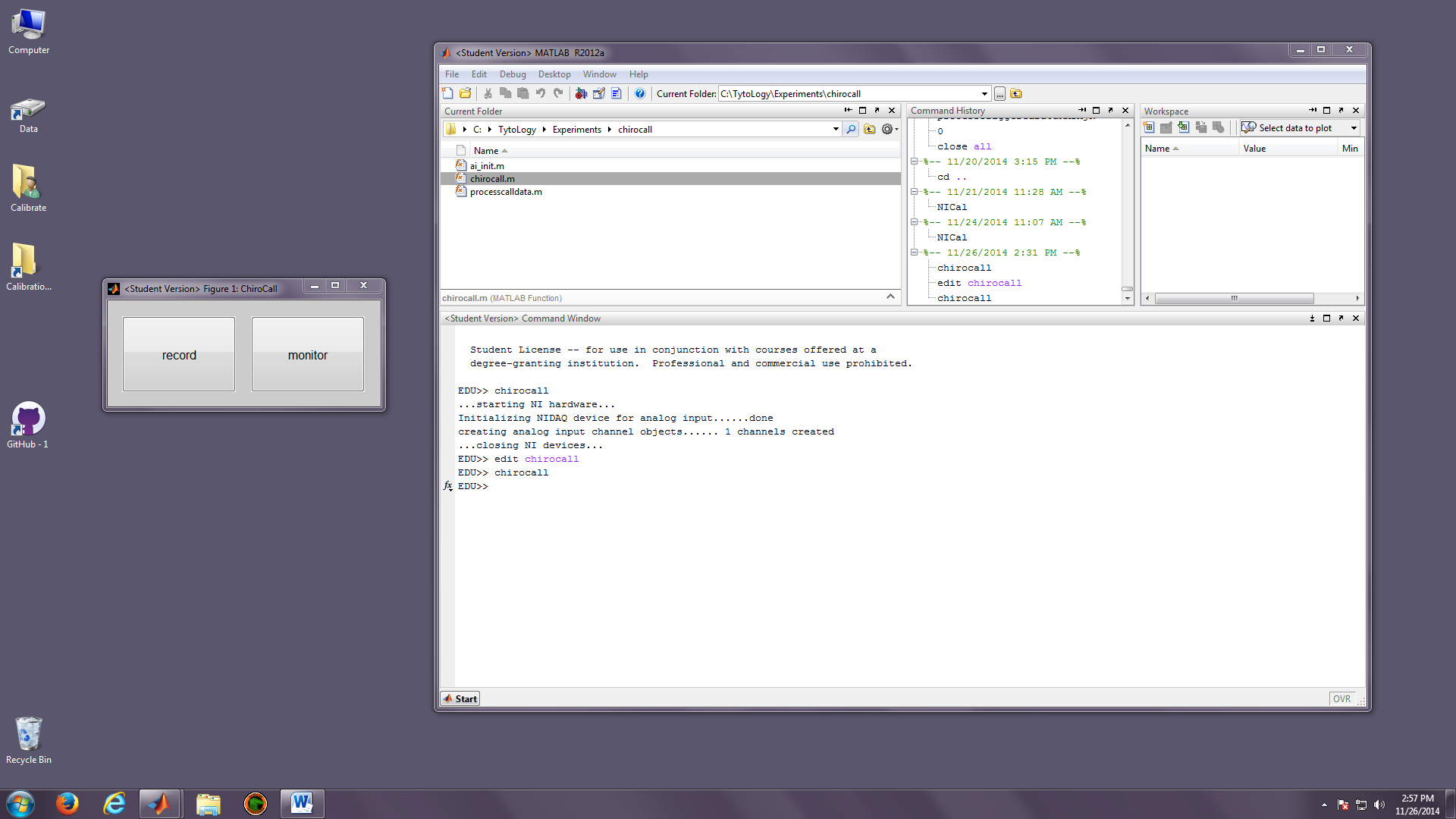
Using chirocall

Chirocall is a simple program that records 1 (or 2) channels of data to disk.

When launched from matlab (add C:\TytoLogy\Experiments\chirocall to your path or simply change to the chirocall directory), chirocall opens a simple, 2-button GUI (***G****raphical* ***U****ser* ***I****nterface*):

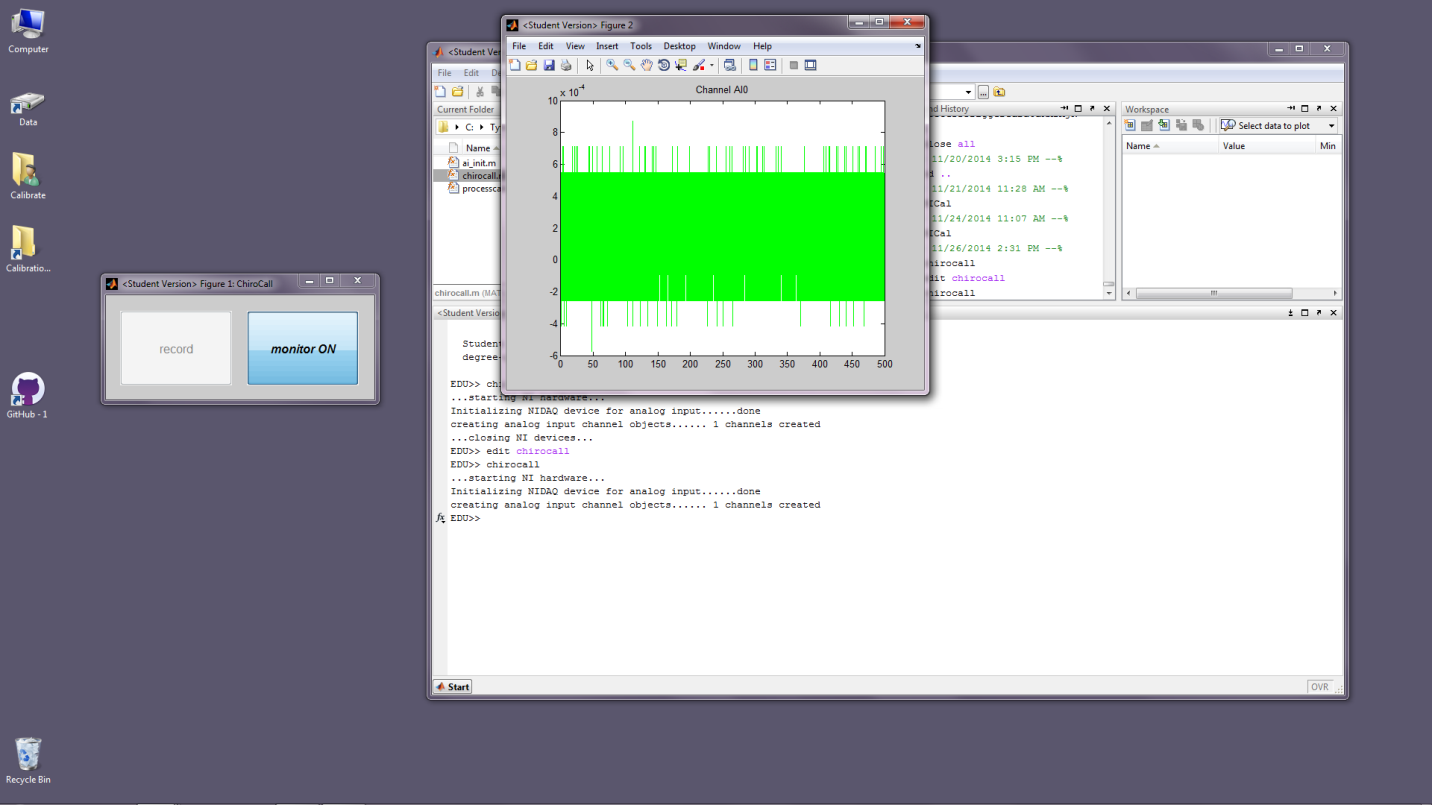


# Monitoring Input Channels:

To display incoming signals without recording them to disk, simply click on the “Monitor” button.

Once clicked, the National Instruments driver and system will be initialized (this might take a few seconds) and a figure will be created to show the incoming data in 0.5 second-long snapshots.

If more than one channel is being collected (see “**Setting Number of Input Channels**” to learn how to change the number of channels that will be recorded), 2 plots will be shown.

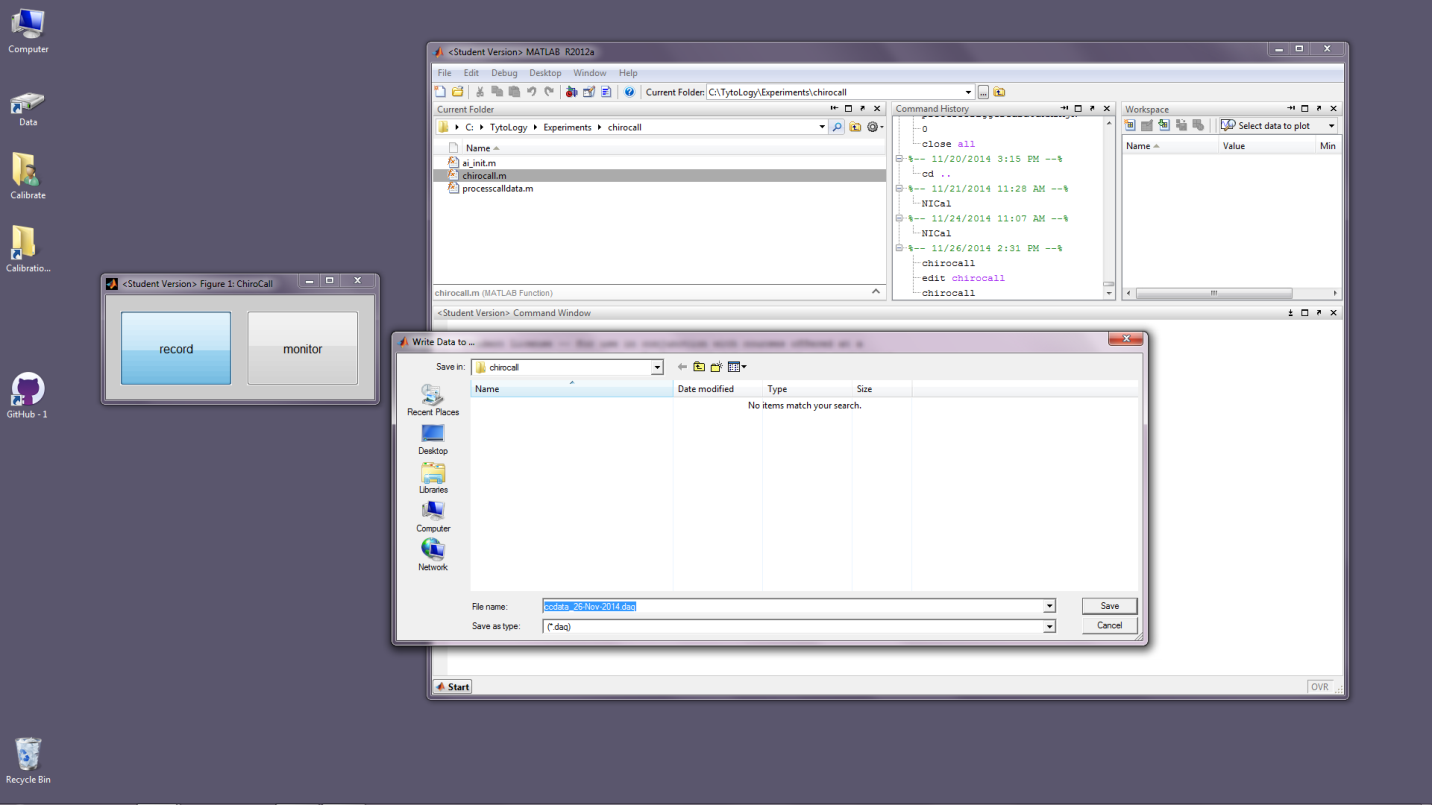


To stop monitoring channels, click the “Monitor” button (now named “monitor ON”) again.

# Recording data to disk

Recording data is initiated by clicking the “Record” button in the GUI.

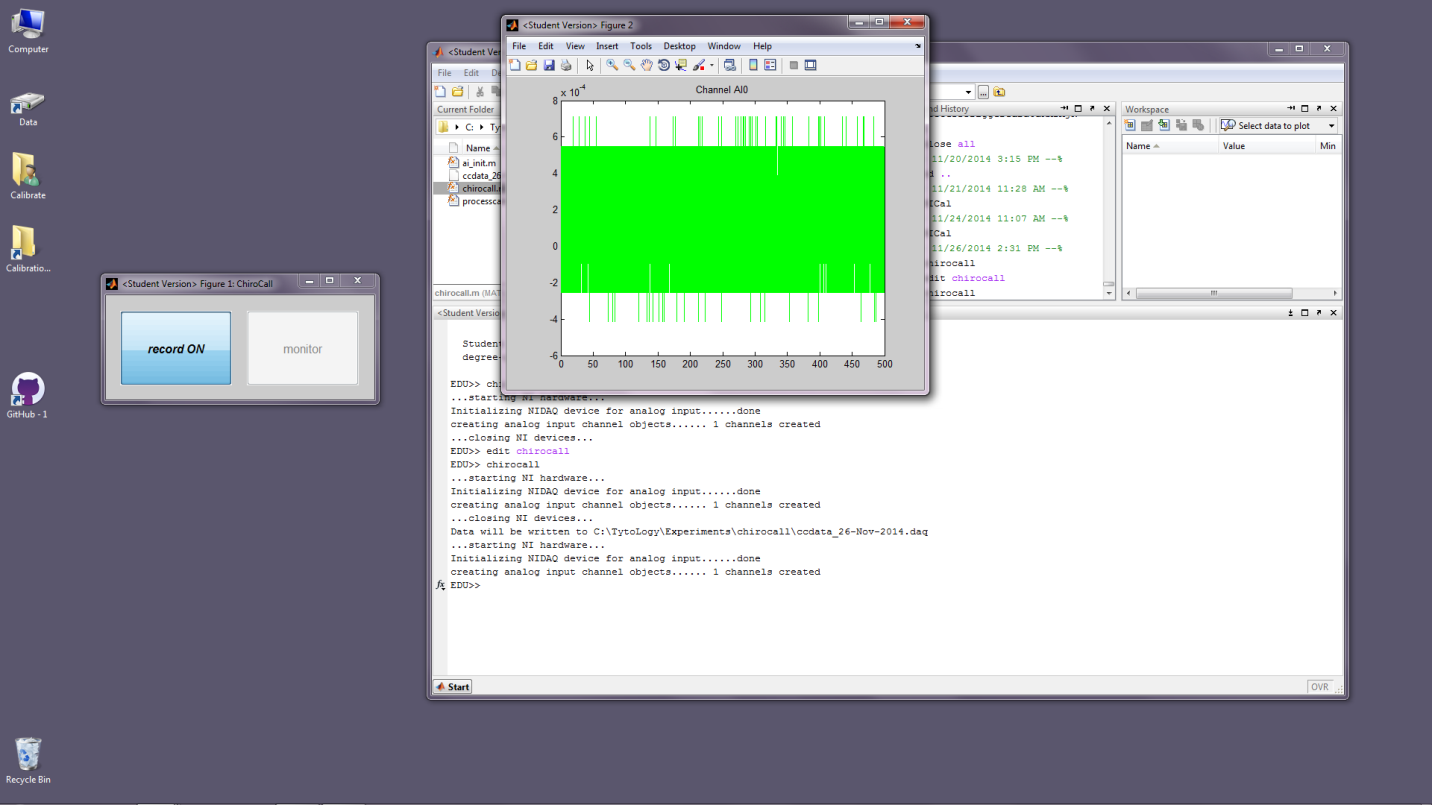
When clicked, a panel will ask for a file to which the data will be recorded.



Select the directory and file name you wish and click “Save”.

If you wish to cancel the recording, click “Cancel”.

After clicking “Save”, data at the analog input channels will be streamed directly to the chosen file and snippets of the data will be displayed in a fashion identical to that in Monitor mode.



To halt data recording, simply click the “Record” button (now labeled “record ON”) again.

Data can be processed/converted to *wav* format using the *processcalldata* function (see “**Processing Data**”.

Alternatively, raw access to the data is possible using the Matlab *daqread* function (see Matlab Data Acquisition Toolbox documentation for more details).

# Setting Number of Input Channels:

The default number of input channels is 1, corresponding to AI0 on the BNC-2110 I/O box),.

To change the number of input channels, edit the chirocall.m file’s “Global Constants” setting:

%------------------------------------------------------------------------

%------------------------------------------------------------------------

% Global Constants

%------------------------------------------------------------------------

%------------------------------------------------------------------------

%------------------------------------------------------------------

% to record from 1 channel (AI1), set H.Nchannels to 1

% to record from 2 channels (AI1 and AI2), set H.Nchannels to 2

%------------------------------------------------------------------

H.Nchannels = 1;

Simply change the value assigned to H.Nchannels to 2 (or back to 1). Settings other than 1 or 2 will give unpredictable and possibly catastrophic results.

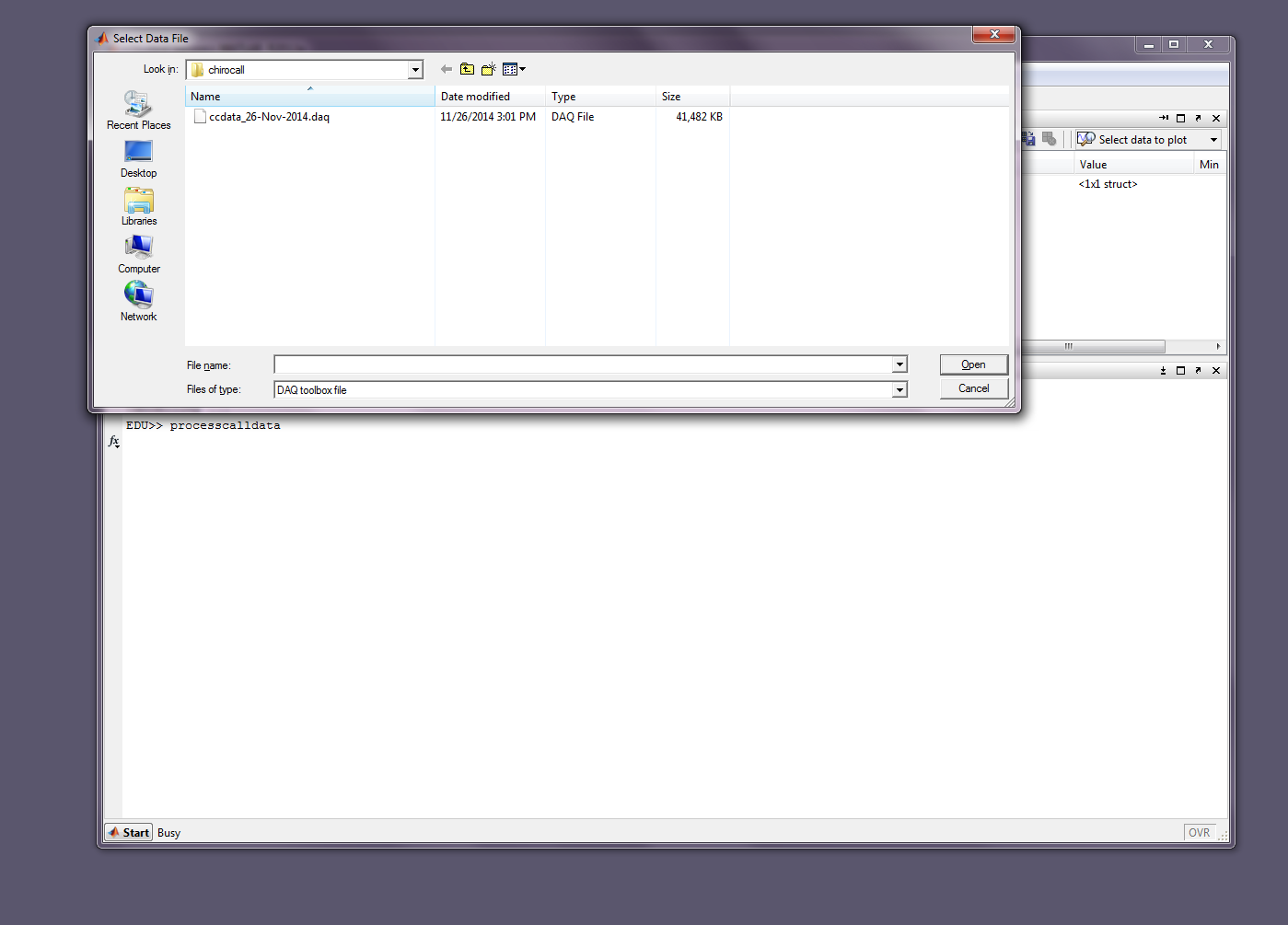
# Processing Call Data

The *processcalldata* function (located in the chirocall directory) can be used to break the continuous data file into shorter “chunks” and store those chunks in *wav* format. The *wav* files can then be read by programs such as Avisoft or Audacity for further processing and analysis.

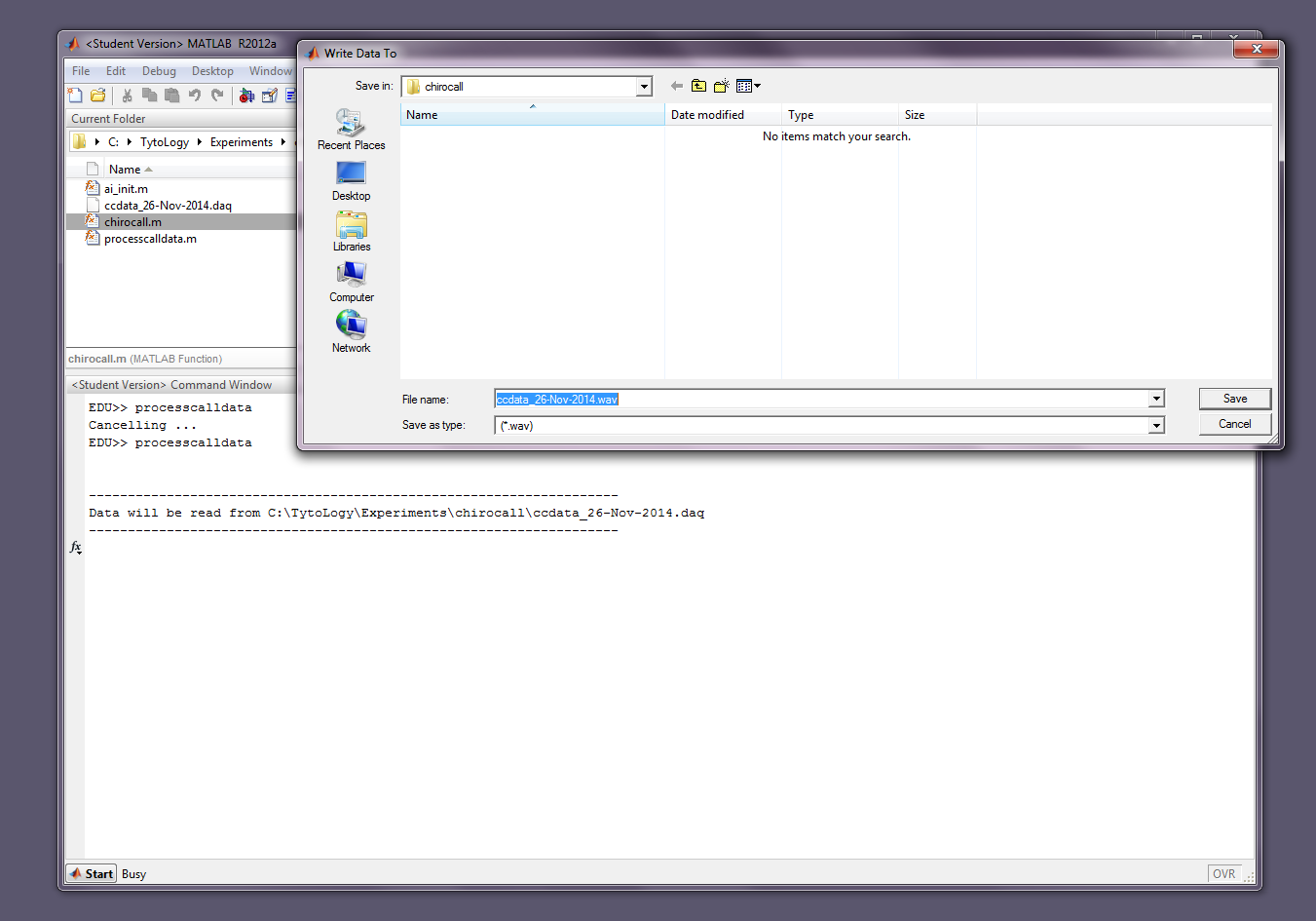
To start processing data, type *processcalldata* at the Matlab command prompt:

>> processcalldata

A panel will ask you to select the *.daq* file to process:



After you select the file, some information about the file will be displayed on the Matlab window and a panel will ask for an output file name.



Because the data will be broken up into shorter duration wav files, the file name you provide for output will become the “base” of the .wav file names. Each chunk of data will have an index number appended to this base.

For example, in this case, the output file selected was

C:\Users\Calibrate\Desktop\tmp\ccdata\_26-Nov-2014.wav

The output wav files will be:

C:\Users\Calibrate\Desktop\tmp\ccdata\_26-Nov-2014\_1.wav

C:\Users\Calibrate\Desktop\tmp\ccdata\_26-Nov-2014\_2.wav

C:\Users\Calibrate\Desktop\tmp\ccdata\_26-Nov-2014\_3.wav

etc.

After selecting the output file name, the program will ask for display some information about the .daq file selected as input:

--------------------------------------------------------------------

File ccdata\_26-Nov-2014.wav has:

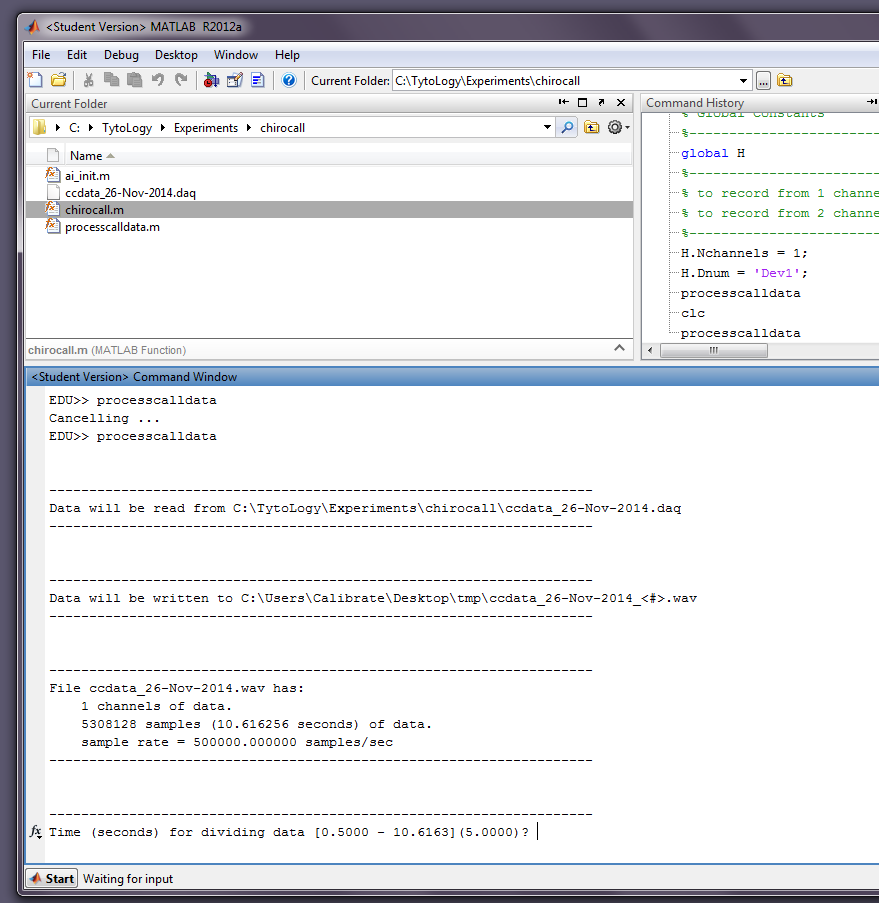
1 channels of data.

5308128 samples (10.616256 seconds) of data.

sample rate = 500000.000000 samples/sec

--------------------------------------------------------------------

*processcalldata* will then request a time value (in seconds) for each “chunk” of the continuous data stream that was recorded.



Minimum value is set to 0.5 seconds, and maximum is the total duration of the recorded file. The default is set to 50% of the total duration of the file you selected for input, rounded down to the nearest second.

In this example, the total file duration was 10.6163 seconds, so the range of “chunk” times is 0.5 to 10.6163 seconds, with a default value of 5.0 seconds.

Press <enter> to select the default, or enter your desired value and then press <enter>.

For this example, 2 seconds was selected as the chunk time.

The program will then compute the chunk time “windows” and display them:

--------------------------------------------------------------------

Time (seconds) for dividing data [0.5000 - 10.6163](5.0000)? 2

Data will be divided into 6 "chunks"

Chunk times (seconds):

Chunk 1: 0.00 - 2.00

Chunk 2: 2.00 - 4.00

Chunk 3: 4.00 - 6.00

Chunk 4: 6.00 - 8.00

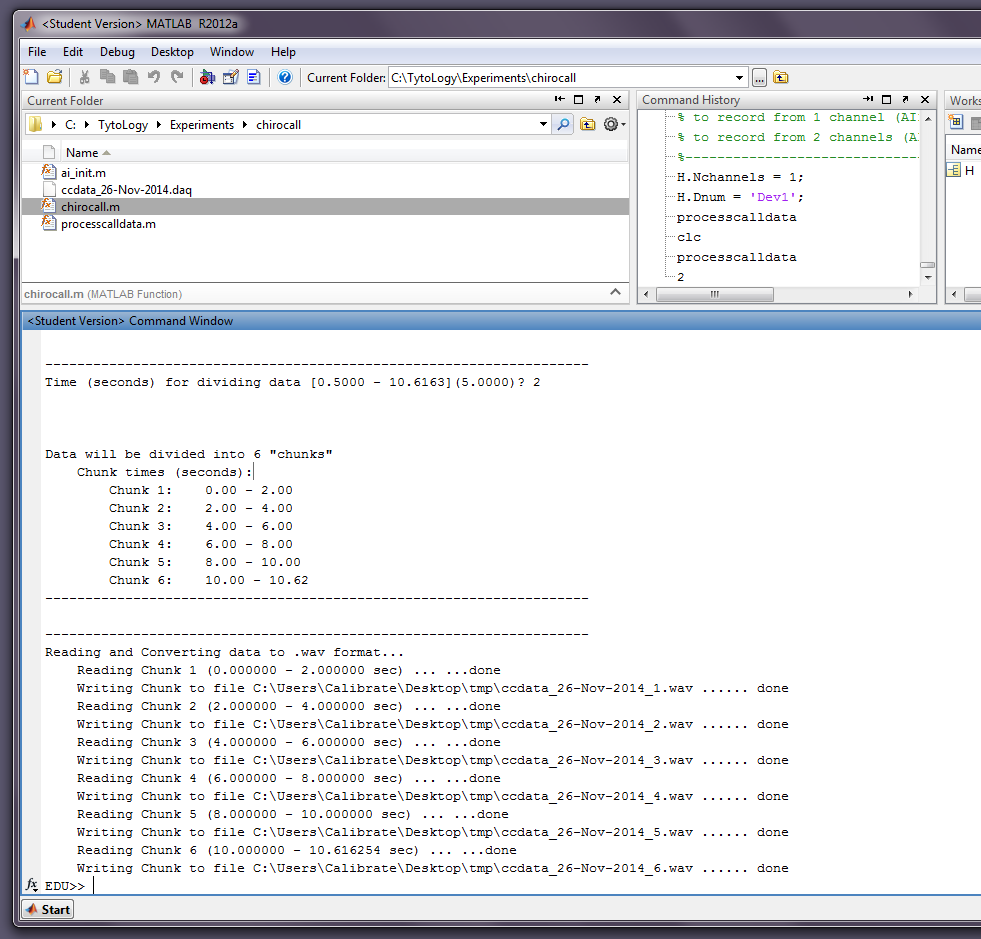
Chunk 5: 8.00 - 10.00

Chunk 6: 10.00 - 10.62

--------------------------------------------------------------------

Note that chunk 6 is the “remainder” of the data after dividing the 10.6163 seconds of data into 2 second chunks.

The program then reads in the chunks of data from the .daq file, normalizes the data to +/- 0.95 V, and exports the chunks to .wav files. Normalization is done due to a +/- 1V limit in .wav format; values outside this range would otherwise be clipped, leading to bad things…



After all the chunks are written to .wav files, the program is finished.