Neuroscope Development Notes

Building a Running Program in General

You will need QT, Cerebus, HDF5, LibNeurosuite, and Neuroscope to build a running program.

C++ Compiler

You need a C++ compiler, and you need to know where it is.
Microsoft Visual Studio, clang, gcc have all worked fine.

CMake

This tool is used to build libraries, if you are building libraries from scratch. If the Windows and Mac OSX libraries work for you, you don't need CMake.
Downlaod Cmake from: https://cmake.org/
I also use the GUI version.

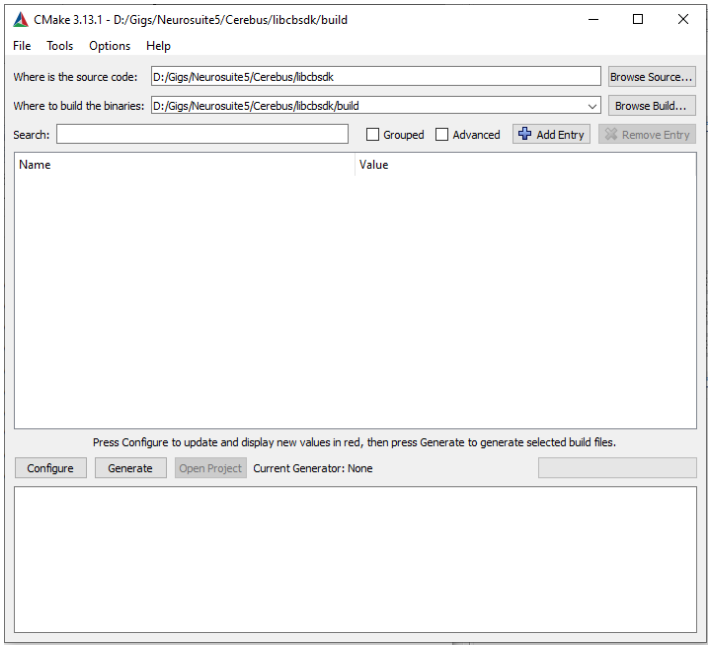
QT

- On the Mac (OSX), attempt to get the latest version of QT, say QT 5.12 or newer. Also install XCODE. Older versions of QT are not compatible with some OSX constraints.
- On Umbuntu or Linux Mint or similar, try to install QT 5.5.1. Older versions likely won't work, and versions 5.6.3 and newer use a Chromium component that Umbuntu may not want.
- I have not tried other versions of Linux, but I'd guess that you may find success using the latest version of QT, say QT5.13 as of May 2020.
- Windows needs 5.5.1 or newer.

For all operating systems, install Qt Creator, since we will use it to build the projects.

Cerebus

If you are not using Mac OSX or Windows, then clone
https://github.com/CompSciEng/libcbsdk.
You may want to get it from Florian Franzen's page, but I copied it in case it gets moved.
Run the Cmake GUI and point to the libcbsdk inside Cerebus. Point also to the build folder.



Click **Configure**, and I hope everything is OK. If not, fill in the missing information and click C**onfigure** again. Then click **Generate**.
On my Mac (OSX), I had the following settings.

- Cerebus CMake
- CMAKE_BUILD_TYPE Release
- CMAKE_EXECUTABLE_FORMAT MACHO
- CMAKE_INSTALL_PREFIX /usr/local
- Qt5Concurrent_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Concurrent
- Qt5Core_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Core
- QtXml_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Xml

My file was found at:
Cerebus/libcbsdk/build/src/release/libcbsdk.dylib

The main output you need is a library called: **libcbsdk t**hat has an extension appropriate for your operating system. (e.g. Mac = dylib, Windows=dll) Write down where yours is located.

HDF5

From the official site, download HDF5. https://www.hdfgroup.org/solutions/hdf5/
You may want HDF5View to help diagnose issues, so I'd grab that also.
They will give you instructions on how to build the libraries. If you only want release libraries, you may be able to find pre-built libraries, and then you can ignore the rest of this section. I wanted debug libraries for Windows, so I had to build them.

Download, unzip, and look for the folder that holds HDF5config.cmake. For **Windows**, from that folder in a command window, type:
ctest -s HDF5config.cmake,BUILD_GENERATOR=VS201764,--enabledshared-shared –enable-cxx –with-zlib –enable-**debug** -vv -O hdf5.log

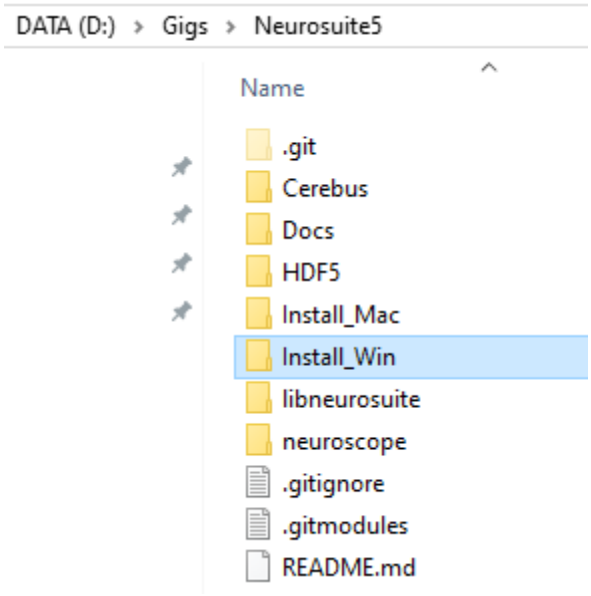
Find the debug libraries that you just built and copy them someplace else. Now rerun this script for the release libraries:
ctest -s HDF5config.cmake,BUILD_GENERATOR=VS201764,--enabledshared-shared –enable-cxx –with-zlib –enable-**production** -vv -O hdf5.log

If you did not move them, they get overwritten.
You now have the HDF5 libraries you need.

LibNeurosuite
Neuroscope

Clone these to your local location.
You will use QtCreator to build these files.

On my Windows box, I use the following folder structure.



QtCreator

I could not get cmake to build neuroscope, so I used QtCreator, which is a free app from Qt.
You need to install QtCreator, if you have not already done so. Then configure your compiler of choice. I used clang for Mac OSX, and maybe Visual Studio 2017 for Windows. I think I used gcc for Linux and maybe also for one pass on the Mac.

The two project files are found on my system at:
Neuroscope5/neuroscope/neuroscope/neuroscope.pro
Neuroscope5/libneurosuite/libneurosuite/libneurosuite.pro
Load them both into QtCreator, and make neuroscope the master project.

Build libneurosuite and then neuroscope and see if they run.
You likely need to change the folder locations for a few libraries.

I STOPPED WRITING HERE, so below are just raw notes that will be re-written.

Bundle Application Files

1. Copy neuroscope.app to ./Install_Mac folder.
   I put it in a folder called DoNotShip, since it may be huge

2. ~/Qt/5.12.2/clang_64/bin/macdeployqt neuroscope.app
3. Set up your app folder like this:
neuroscope.app
Contents
Frameworks
folder:Cerebus dylib
folder:libneuroscope dylib
QT files
info.plist
MacOS
neuroscope
PkgInfo
PlugIns
Resources
icon file
4. copy the custom dylibs inside Framework.
/Cerebus/libcbsdk/build/src/release/libcbsdk.dylib
/libneurosuite/build-libne....-Release/*.dylib
5. modify how neuroscope finds the libraries.
A. Go to the executable inside MacOS
B. otool -L neuroscope
This is optional, but shows the libraries and locations.
I found the @rpath directory this way.
C. // install_name_tool -change liblibneurosuite.1.dylib
@executable_path/../Frameworks/libneurosuite/liblibneurosuite.1.dylib neuroscope
install_name_tool -change @rpath/liblibneurosuite.1.dylib
@executable_path/../Frameworks/libneurosuite/liblibneurosuite.1.dylib neuroscope
D. install_name_tool -change @rpath/libcbsdk.dylib
@executable_path/../Frameworks/Cerebus/libcbsdk.dylib neuroscope
6. Double click on neuroscope.app and make sure it runs.

Make a dmg file:
1. Run Disk Utility.
2. File -> New Image -> Image from Folder take Install_Mac.
3. SaveAs Neuroscope.dmg likely in folder above Install_Mac.
4. Move the dmg inside Install_Mac.
5. Move or delete the neuroscope.app file. It has files that are way too large.
6. Github likely stores it at:
https://github.com/CompSciEng/Neurosuite5/blob/AddNWB/Install_Mac/neuroscope.dmg

NWB Location Files
Each NWB file will have a partnering XML file that specifies where the data is found.
For example, YutaMouse41-150903.nwb has an associated file named: YutaMouse41-150903_loc.xml.
We added the _loc to the end of the file base name, since Neuroscope already makes an XML file that they call the “parameter” file.
Here is a sample.

<parameters creator="neuroscope-1.2.3" version="1.0" >
<nwb_dataset_name>/processing/ecephys/LFP/lfp/data</nwb_dataset_name>

<nwb_sampling_name>/processing/ecephys/LFP/lfp/starting_time</nwb_sampling_name>
<nwb_voltage_electrodes>/processing/ecephys/LFP/lfp/electrodes</nwb_voltage_electrodes>

<nwb_voltage_electrodes_shanks>general/extracellular_ephys/electrodes/shank_electrode_number</nwb_voltage_electrodes_shanks>
</parameters>

The specification document indicates that the voltage should be found in:
acquisition/all_lfp/all_lfp/data
Let us adapt the shorthand that LFP = acquisition/all_lfp/all_lfp/.
Thus, the voltage would be found in LFP/data.

The color integers would be found in LFP/electrodes.
These integers would gives us the indices of:
general/extracellular/electrodes/shank.

Cerebus CMake
CMAKE_BUILD_TYPE Release
CMAKE_EXECUTABLE_FORMAT MACHO
CMAKE_INSTALL_PREFIX /usr/local
Qt5Concurrent_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Concurrent
Qt5Core_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Core
QtXml_DIR /Users/robertmoore/Qt/5.12.2/clang_64/lib/cmake/Qt5Xml

My file was found at:
Cerebus/libcbsdk/build/src/release/libcbsdk.dylib

HDF5 libraries were built last May 2019 using cmake or a script.
I think I built them with debug, moved the files, and then built with release, or vice versa.
This way, you get both debug and release. CMake can do this, but it was not clear.
./Neuroscope/HDF5 2/CMake-hdf5-1.10.5/build/bin/ see .a and dylib files

chmod 755 FixPaths.sh

$ pandoc --from docbook --to html --output myDoc.html index.docbook.txt