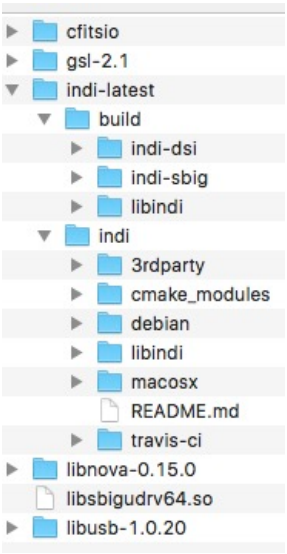
1. I followed these instructions to build INDI from source and get it installed on the Mac in the proper location: indilib.org/forum/general/210-howto-buil...st-libindi-ekos.html

Do this at the command line:



mkdir ~/Projects
cd ~/Projects
git clone github.com/indilib/indi.git

Before you can build, follow the instructions in the Mac OS X readme file in the download from the git. This includes downloading libnova-0.15.0, libusb-1.0.19,cfit3380, and gsl-2.1, then placing the unzipped folders at the same level as your git download, and then running configure and make from the command line in the gsl folder. See the screenshot to the right.

Also before you build, double check that CMakeLists.txt in the libindi folder is correct. Open the file and make sure it says this, if not, change it.

```
set (indiclient_SRCS
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/basedevice.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/baseclient.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/indiproperty.cpp
${CMAKE_CURRENT_SOURCE_DIR}/base64.c
)

set (indiclientqt_SRCS
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/basedevice.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/baseclientqt.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/indiproperty.cpp
${CMAKE_CURRENT_SOURCE_DIR}/base64.c
)
```

Now go back to the Projects directory and follow these instructions.

mkdir -p build/libindi
cd build/libindi
cmake -DCMAKE_INSTALL_PREFIX=/usr/local -DCMAKE_BUILD_TYPE=Debug ~/Projects/indi/libindi
sudo make install

If you want the 3rd Party drivers like SBIG or DSI, you can try to build and install them or just copy them from somewhere else if you can find them.

Check that the binary files are all in /usr/local/bin and that the driver xml files are in /usr/local/share/indi

2. Make sure you have home-brew installed. brew.sh

3. From here, I followed Sean Houghton's Instructions to install a variety of dependencies needed using home-brew:
gist.github.com/seanhoughton/1b2649a2a0ef904d79f9

brew install qt5 --with-dbus

brew tap homebrew/science
brew install cfitsio
brew install cmake
brew install eigen
brew install gettext
brew install astrometry-net

brew tap haraldf/kf5
brew install haraldf/kf5/kf5-kplotting
brew install haraldf/kf5/kf5-kxmlgui
brew install haraldf/kf5/kf5-knewstuff
brew install haraldf/kf5/k5f-kdoctools
brew install haraldf/kf5/kf5-knotifications
brew install haraldf/kf5/kf5-kcrash

4. We can now use the latest version of KStars, rather than the one from Sean Houghton. I picked an empty folder on my hard drive to download the latest version of kstars-bleeding. I executed this from the command line in that folder. You might need a github account and password for this step:

git clone git://anongit.kde.org/kstars.git

5. Then I ran the following commands to export useful directory paths and use cmake to build the Xcode project (also on his page):

mkdir kstars-build
cd kstars-build
export PATH=$PATH:$(brew --prefix gettext)/bin
export Qt5_DIR=$(brew --prefix qt5)
export Qt5DBus_DIR=$Qt5_DIR
export Qt5Test_DIR=$Qt5_DIR
export Qt5Network_DIR=$Qt5_DIR
export ECM_DIR=$(brew --prefix kf5-extra-cmake-modules)/share/ECM

6. Now you have a choice, if you would like to create an xcode project where you can edit the code and make an app bundle, run the following command:
cmake -DCMAKE_INSTALL_PREFIX=~/usr/local -G Xcode ../kstars

If you would prefer a command line version and don't want to use xcode at all, then do this command instead:
cmake -DCMAKE_INSTALL_PREFIX=~/usr/local ../kstars

7a. If you chose to create an XCode project, you can go and get it in the build directory and double click, Xcode should open it. If that is the case then follow these directions to get an app:

a. First copy the entire data directory in the KStars folder from the git to a folder you create called ~/Library/Application Support/kstars
b. In Xcode, Near the top left, it should say "all build." Change that to kstars.
c. Go to "schemes" in the product menu and edit scheme
d. Select "Release" and uncheck "debug executable"
e. Hit the run button at the top left. KStars should run successfully. Do any tests you need to do. Edit code if you like.
f. Then close XCode.
g. In the KStars build folder, you should find a folder called KStars and in there one called "Release" In that folder is kstars.app
h. Copy and paste KStars.app wherever you like.

7b. If instead you chose to just get the command line version, then follow these instructions instead:

a. After running the cmake command, do sudo make install
b. Then run this command to symlink the data directory to ~/Library/Application Support/kstars

   ln -s /usr/local/share/kstars/* ~/Library/Application\ Support/kstars/

8. If you would like gsc so that the simulators can have nice stars to work with, then install gsc following these insructions:

http://www.indilib.org/support/tutorials/139-indi-library-on-raspberry-pi.html

a. Find a folder on your hard drive in a terminal window and use these commands:

```
mkdir ~/gsc
cd ~/gsc
wget -O bincats_GSC_1.2.tar.gz http://cdsarc.u-strasbg.fr/viz-bin/nph-Cat/tar.gz?bincats/GSC_1.2
```

```
tar –xvzf bincats_GSC_1.2.tar.gz
cd src
make
mv gsc.exe gsc
sudo cp gsc /usr/local/bin/
cd ..
cp ~/gsc/bin/regions.* /gsc
```

I had a couple of problems. First, it needed both of the regions files in a subfolder called bin inside the final gsc folder. Second, I had to copy all the folders that began with N and S to the final gsc folder (but NOT in the bin subfolder). Third, I had trouble getting the environment variable permanent, so you will note I made the final line install to /gsc not ~/gsc