



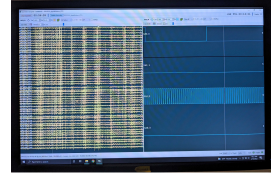
Feb 12, 2024

Version 4

Building a SpikeGLX Rig with camera: Chronic recoverable Neuropixels in mice V.4

DOI

<https://dx.doi.org/10.17504/protocols.io.kxygxzzxkv8j/v4>



Emily A Aery Jones¹

¹Stanford University



Emily Aery Jones

University of Maryland, Baltimore

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.kxygxzzxkv8j/v4>

Protocol Citation: Emily A Aery Jones 2024. Building a SpikeGLX Rig with camera: Chronic recoverable Neuropixels in mice. [protocols.io https://dx.doi.org/10.17504/protocols.io.kxygxzzxkv8j/v4](https://dx.doi.org/10.17504/protocols.io.kxygxzzxkv8j/v4) Version created by **Emily Aery Jones**

License: This is an open access protocol distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited



Protocol status: Working

We use this protocol and it's working

Created: September 14, 2023

Last Modified: February 12, 2024

Protocol Integer ID: 87805

Keywords: electrophysiology, camera, Neuropixels, freely moving recording, spatial navigation, neuropixels recording system, spikeglx rig with camera, chronic recoverable neuropixel, chronic recoverable neuropixels in mice, spikeglx software, neuropixel, spikeglx rig, camera, video frame, first recording, mice, record

Abstract

This protocol collection explains how to build a low-cost, lightweight system to implant 1 Neuropixels 1.0 probe or 2 Neuropixels 2.0 probes into mice, record during freely moving behavior, then recover the probes for future use. This protocol first describes how to build a Neuropixels recording system using a National Instruments acquisition system and SpikeGLX software, from parts acquisition through your first recording. It also describes how to integrate a camera, including code and wiring to record and synchronize video frames. See full collection for more details.

Guidelines

The acquisition set up portion of this protocol is intended to be an unofficial, streamlined version of the official documentation, but not a replacement for it.



Materials

Acquisition system and computer:

Parts list:

https://github.com/emilyasterjones/X_maze/blob/main/Build/Neuropixels%20Acquisition%20System%20Parts.xls
[X](#)

Minimum requirements: <https://open-ephys.github.io/gui-docs/User-Manual/Plugins/Neuropixels-PXI.html#hardware-requirements>

Camera:

Parts list: https://github.com/emilyasterjones/X_maze/blob/main/Build/Mako%20G%20Camera%20Design.xlsx

Consumables:

- 12V power supply (Jameco, 170245)
- 2.1mm barrel jack (Digikey, CP-037A-ND)
- BNC to wire leads (Digikey, 501-1030-ND)
- jumper wires (Sparkfun, 11709)
- resistors (Sparkfun, 10969)
- breadboard (Sparkfun, 12615)
- heatshrink tubing (Amazon, B0722HN8SW)
- solder

Equipment:

- soldering iron, heat gun, and accessories (wire cutters, soldering clamp, solder, light)

Before start

Set aside a few days to learn how the Neuropixels and SpikeGLX systems work

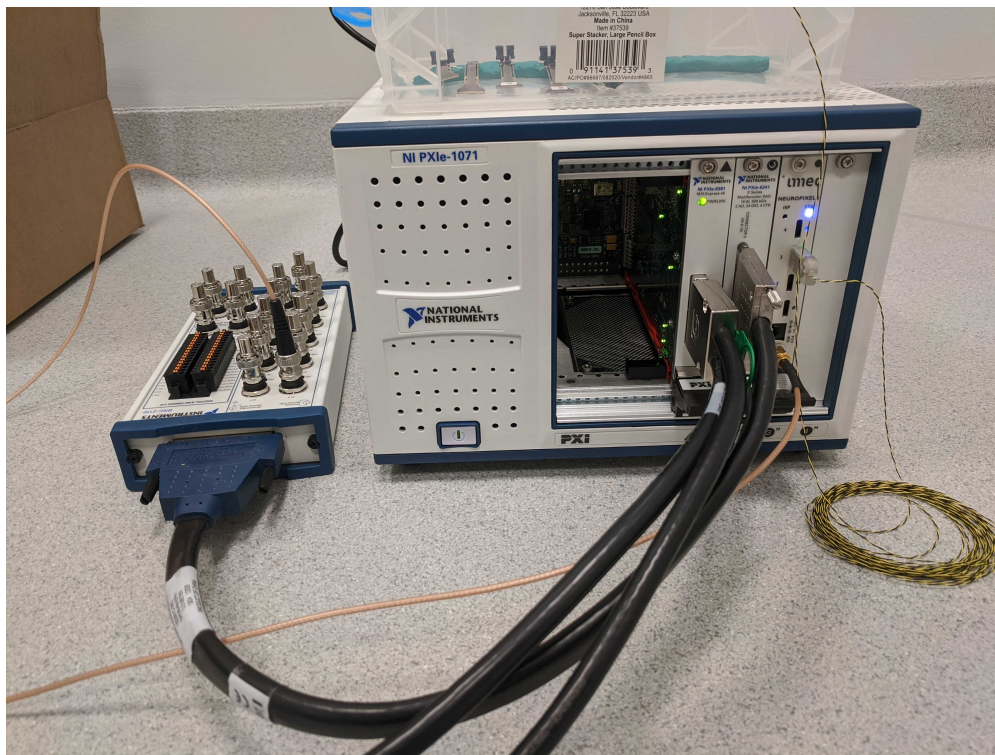
- Read the latest Neuropixels manual: <https://www.neuropixels.org/support>
- Join the Neuropixels community Slack to ask for help:
https://neuropixelsgroup.slack.com/join/shared_invite/zt-3z0jaogn-iHLPiWBD8XC13JeTMPggIQ#/shared-invite/email
- Watch the latest Neuropixels course online: <https://github.com/cortex-lab/neuropixels/wiki>
- Read the SpikeGLX manual: <https://github.com/billkarsh/SpikeGLX/blob/master/Markdown/UserManual.md>

Select a machine vision camera configuration:

- You will need a camera that you can manually wire to send synchronizing pulses to the NIDAQ system in order to synchronize video frames with neural data later.
- You will need a camera, lens, power supply, mount, I/O cable with bare ends, data cable, tripod mount, and computer.
- Supplier: Allied Vision cameras work with NIDAQ systems. This protocol explains how to wire an Allied Vision camera for this purpose.
- Camera: Get a high enough resolution & frame rate for your project. Some cameras can adjust their settings so you can capture lower resolution at higher frame rates and vice versa. C(S)-mount refers to the lens attachment. C-mount cameras can take C or CS lenses, but CS-mount cameras can only take CS lenses.
- Lens: The greater the focal length, the greater the zoom. Use this tool (<https://www.baslerweb.com/en/products/tools/lens-selector>) to estimate lens size based on camera to object distance and the size of the field. E.g. open field camera might use a 6mm lens, while a pupil camera might use a 24mm lens.
- Power supply: whatever the camera needs. You may be able to power the camera through the data cable.
- I/O cable with bare ends: This sends signals to the camera to start/stop capturing and receives signals of frame times.
- Data cable: GigE (Cat6 cable) & USB are the most common. GigE is generally better over distances >5m.
- Mounting: Attach to 80/20 by screwing into a tripod mount.
- Computer: If you don't need real-time control, then just get a tower with enough SSD space to store your videos. If using GigE, make sure you get one that has PCIe cards to fit the Intel network adapter, and also get a PCIe Desktop Adapter.

Set up acquisition hardware

- 1 Load the PCIe module into your computer
- 2 Load the PXIe modules and the IMEC module into the NIDAQ chassis
- 3 Connect the PXIe modules to the BNC breakout board and the computer. Connect the IMEC SMA to the BNC board (analog slot 0). Cover remaining unused BNC slots with BNC terminators to prevent TTLs from leaking across channels.
- 4 Plug in a probe, connected to a headstage, connected to a cable, into a port on the IMEC module.



NIDAQ chassis (right) has 3 modules, left to right: (1) connect to PCIe in the computer, (2) connect to synchronization board, and (3) connect to headstage. The imec module (rightmost) is here connected to a probe (probe box above) and has a cable connecting it to the synchronization board (left), which receives TTL pulses from all hardware to synchronize data streams from multiple sources (e.g. synchronizing video frames to neural data).

- 5 Boot the chassis, then the computer.



Note

Always boot the chassis before booting the computer.

Set computer settings

- 6 Get your machine connected to the internet and install any institute-specific software.
- 7 Run any Windows updates. These can break things sometimes. If Windows updates that occur after the NIDAQ set up break your SpikeGLX, you can roll them back like this:
<https://www.howtogeek.com/235474/how-to-roll-back-builds-and-uninstall-updates-on-windows-10/>
- 8 Change your power settings so your computer doesn't sleep:
<https://github.com/billkarsh/SpikeGLX/blob/master/Markdown/UserManual.md#screen-saver-and-power-settings>.

Disable hibernation: on Windows machines, open Command Prompt as an administrator. Run the command:

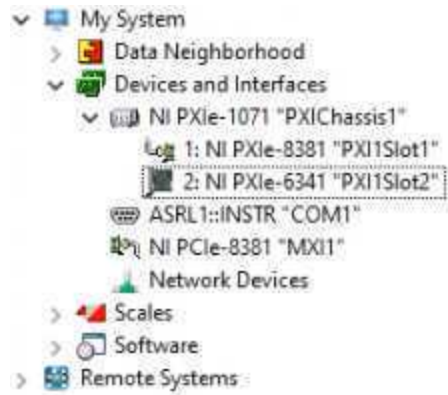
```
powercfg -h off
```

This will prevent the computer from creating hiberfil.sys, a system file used to store the current state of the computer when it hibernates, which on acquisition computers can become massive (hundreds of GB).

If your computer still sleeps or hibernates, install Caffeine:
<https://www.zhornsoftware.co.uk/caffeine/>.

Install National Instruments drivers

- 9 Install PXI Platform services: <https://www.ni.com/en-us/support/downloads/drivers/download.pxi-platform-services.html>
- 10 Install DAQmx: <http://www.ni.com/en-us/support/downloads/drivers.html>
- 11 Run NI MAX.



The left-hand panel should look like this if everything ran correctly, with no warnings about missing drivers.

Install Enclustra drivers

- 12 Download the Enclustra drivers:
https://billkarsh.github.io/SpikeGLX/Support/Enclustra_Win10.zip. Unzip them anywhere.
- 13 Open Device Manager. Under "Other Devices", right-click the PXI device and select "Update". Select browse from my computer and select the folder where you unzipped the Enclustra drivers.
- 14 The PXI device should now appear under "Enclustra" in the Device Manager.

Install SpikeGLX

- 15 Create the directory C:/SpikeGLX
- 16 Download the latest release: <https://billkarsh.github.io/SpikeGLX/#latest-application-downloads>. Unzip and place the the whole thing, still inside the release folder, into your new SpikeGLX folder.
- 17 Copy all the Calibration files that you received via email when you purchased the probes into the Calibration folder in SpikeGLX. Note that some of them have typos from when IMEC sent them. The correct format is [probe #]/[probe #][ADCCalibration or gainCalValues]. I.e. each probe should have its own directory named after it, with 2 files inside with the probe number once, underscore, and the file type.

**Note**

If you need a calibration file, email neuropixels.info@imec.be with the probe #

Update firmware

- 18 Launch SpikeGLX. Go to Tools/Update Imec Firmware.
- 19 Select the slot your IMEC module is in.
- 20 Check the box and select the file(s) for a BS and a BSC update. The files for this version of SpikeGLX are in the folder named Firmware that is included in your SpikeGLX release. Click Update.
- 21 Power cycle by shutting down the computer, then the NIDAQ, and waiting for a few seconds before rebooting.

Test the probe and headstage

- 22 Launch Spike GLX. Tools/BIST Test or Tools/HST Test.
- 23 Select the slot your IMEC module is in, the port your headstage is in, and dock 1.
- 24 For HST, plug in the headstage test device. Note as of April 2021, this isn't working in the newest IMEC updates.

First acquisition

- 25 Launch SpikeGLX. File/New Acquisition.
- 26 Check the boxes for enable IMEC and enable NIDAQ.



- 27 Select the slot your module is in and click Add. Then uncheck the boxes except for the row with the port your headstage is plugged into and dock 1 (dock 2 is for 2.0 probes). Click detect.
- 28 From here, follow the SpikeGLX user manual. Briefly: set your probe configuration (e.g. which reference to use, which block of channels to record):
<https://github.com/billkarsh/SpikeGLX/blob/master/Markdown/UserManual.md#im-setup---configuring-imec-probes> in the "IM Setup" tab and the name of your file in the "Save" tab. Click "Run" to start streaming, and "Enable recording" to start recording to a file.

Calibrate the headstage clock

- 29 Calculate the true sample rate and determine how the headstage and computer clocks map. Do this for every new headstage. First, launch SpikeGLX and detect the headstage, as described above.
- 30 In the "NI Setup" tab, specify 0:7 analog (XA) and 0:7 digital (XD).
- 31 Go to the Sync tab of the Configuration to set up the sync pulse.
 1. Select "Itec slot x" under Square Wave Source, where x is the slot of the Itec card in the PXle chassis. This produces a 1 Hz square wave with 50% duty cycle. The sync output will be visible on the last channel of the "imec0 probe", SY0, bit #6.
 2. Input channel should be Analog 0.
 3. Measured period should read "1.00000..." and Threshold should read "1.100."
 4. Check "Use next run for calibration".
 5. Set calibration time: 20 min minimum, 40 min is ideal.
- 32 Click Run to do a Calibration run.
- 33 When Calibration run is done recording, should get a popup dialog with the measured sample rate. Check "Yes" to apply the new sample rates.
- 34 The Calibration file is now in C:/SpikeGLX/Release_<#>/SpikeGLX/_Calibration. You will have:
 - 1 file for _imec, with 1 entry per headstage in the file - if you've calibrated with that headstage, you're calibrated with any probe attached to it.
 - 1 file for _nidaq
- 35 During subsequent acquisitions, SpikeGLX should automatically use your most recent Calibration files. You can also choose them manually in the Sync tab. In the Sync tab,

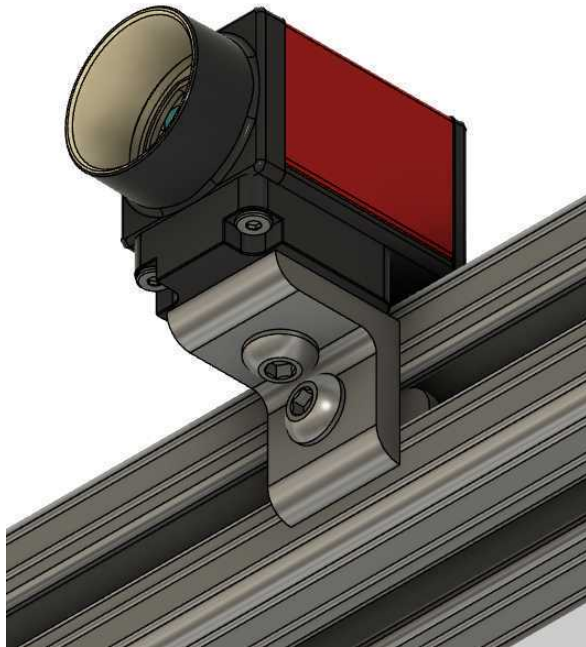
make sure "Use next run for calibration" is unchecked and keep the sync pulse enabled to use the calibrated clock.

Note

If you update SpikeGLX, be sure to copy the Calibration files to the new Release folder

Set up the camera

- 36 Plug the I/O cable and the data cable into the camera. Screw tripod adapter onto base, then attach the adapter to an 80/20 L-bracket mounted on 80/20 frame or to Thorlabs components mounted on an air table, depending on your setup.



Allied Vision Mako G mounted to a tripod adapter and an L-bracket to affix it to an 80/20 frame.



Same as previous figure, implemented

Note

Allied Vision camera tripod mounts attach to

- 37 Set up the camera acquisition computer and acquisition software as described by your camera's manufacturer. This should be a separate computer from the SpikeGLX acquisition computer. For Allied Vision cameras connecting via GigE or FLIR cameras connecting via USB3, you can follow hardware instructions here:

https://github.com/emilyasterjones/X_maze/README.md

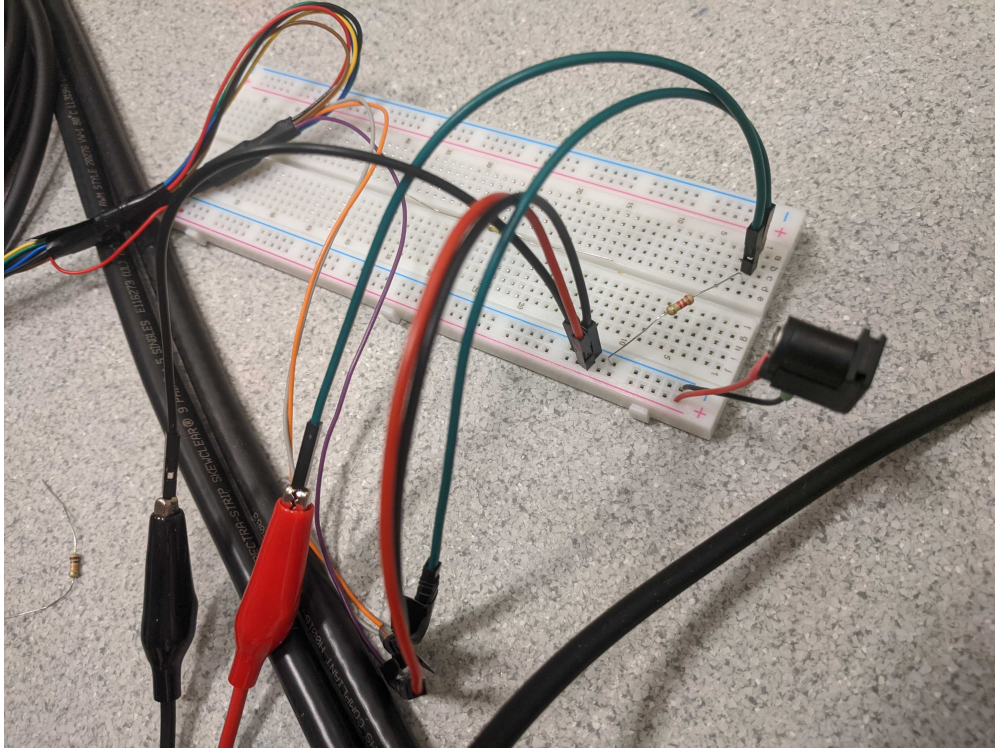
Wire I/O cable to send TTL pulses from camera to SpikeGLX

- 38 To synchronize your video data with your neural data, you'll need to send TTL pulses from the camera to the BNC breakout board. To do so, you'll need to wire the GPIO (Hirose) cable to an external power and to the BNC board. First, look up the correct pinout for your Hirose I/O cable. Allied Vision: https://cdn.alliedvision.com/fileadmin/content/documents/products/accessories/cable/datasheet/Acc_Cable_DataSheet_K7600xxx_AVT_IOcable_V4.1.1_en.pdf. FLIR: <http://softwareservices.flir.com/BFS-U3-23S3/latest/Model/public/DigitalIOControl.html> (change BFS-U3-23S3 to whatever model you have).

- 39 Wire thusly, to a breadboard first while you troubleshoot, then solder and protect with heatshrink tubing. Use the block diagrams starting on page 136 of this document as a guide:

https://cdn.alliedvision.com/fileadmin/content/documents/products/cameras/Mako/techman/Mako_TechMan_en.pdf

- output power & camera power ⇒ power
- camera ground & BNC ground ⇒ ground
- camera output & BNC input ⇒ resistor ⇒ ground



Example wiring on a breadboard of the I/O to bare ends cable.

Note

Test multiple resistors to see which once gives the best TTL pulse.

- 40 Set camera settings thusly:

Brightness | ROI | Autofunction ROI | **Trigger IO** | Color | All

Trigger - Input

Trigger

Source: FixedRate Activ.: RisingEdge

Selector: FrameStart Mode: On

SW Trigger

Acquisition

Mode: Continuous FrameCount: 1

Exp. Mode: Timed

Quick Settings: Trigger Mode

Edge Level Bulk

Quick Settings: Trigger Mode

Edge Level Bulk

Output

SyncOutSource: Polarity: Status:

SyncOut1: Exposing ☒ ☐

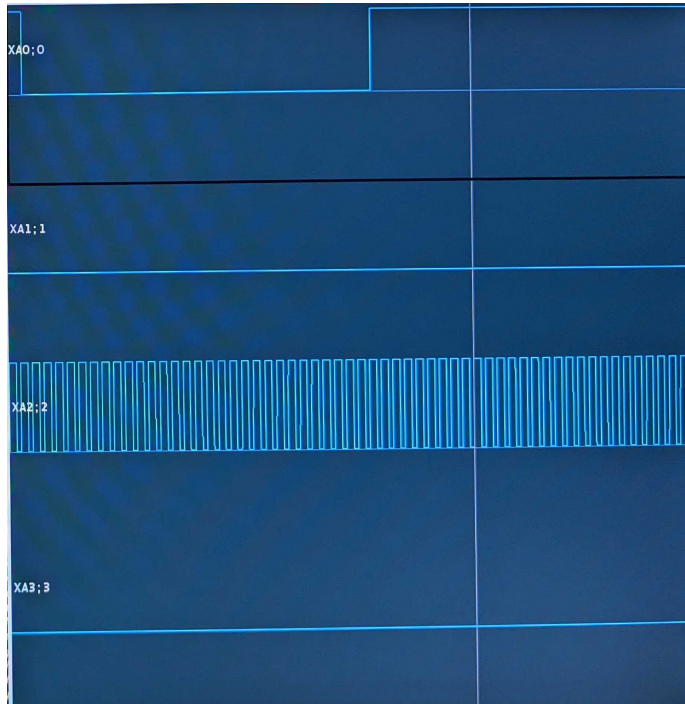
SyncOut2: Exposing ☐ ☐

SyncOut3: Exposing ☐ ☐

SyncLevels: 0

Note: unlike in the figure above, don't check the box for 'polarity'.

- 41 Run SpikeGLX, then hit Run on Vimba Viewer. If it worked, you should see an upward pulse for the duration of the exposure.



TTL pulses received once per frame (up for start of frame, down for end of frame) from a 60fps video capture on channel A2.

Note

Check out other actions you can take over GigE:

https://cdn.alliedvision.com/fileadmin/content/documents/products/cameras/various/features/GigE_Features_Reference.pdf

Acquire video

- 42 **Allied Vision:** Neither SpikeGLX nor Allied Vision have accompanying video acquisition software (Allied Vision's Vimba Viewer allows streaming but not recording of video). To acquire video, use a script to connect to the camera, define its acquisition settings including those for sending TTL pulses, acquire each frame and save it.

One simple option is to install the following:

https://github.com/emilyasterjones/X_maze#gige-vimba-camera-acquisition-instructions

and then run this script:

https://github.com/emilyasterjones/X_maze/tree/main/Capture/Camera. Several camera config files are included; you can create your own by opening Vimba Viewer, searching for and setting relevant variables, then saving this configuration to a file. In particular, adjust the frame rate, gain, and exposure time to suit your project.



FLIR: Depending on the frame rate you require, you may be able to acquire video just using SpinView, FLIR's video streaming and recording software. If you need higher throughput, or want more control over video acquisition (e.g. saving a file with timestamps per frame), use pySpinCapture: <https://github.com/neurojak/pySpinCapture>. An example script to record from a single color camera using the GPU is at https://github.com/emilyasterjones/X_maze/tree/main/Capture/Camera.