

Bill of Materials

This bill of materials (BOM) contains **EVERYTHING** needed to build a setup. From the less memorable screw to the super expensive syringes used for reward delivery.

NOTE

This BOM is destined to the people from the Champalimaud Foundation (CF). Nonetheless, the goal is that people outside CF are also able to order everything and build the setup.

WARNING

This BOM specifies the part numbers used by the lab for these setups, but not every component needs to be of a specific part number.

The only things that need to be of a specific model are the Harp Devices, the Behavior Poke Port Breakout v1.1, the Poke Small v1.1 and the 12V Power Supplies.

It's possible to use a different camera model, but please confirm that the camera can be used in Bonsai (although for some cameras the Bonsai workflow will probably need to be adapted).

Harp

This section contains the setup components that relate to the Harp devices somehow (except for the Camera that has a section of its own).

Devices

The Harp devices are assembled by the Hardware and Software Platform, so place an Agendo request to order them.

WARNING

For people outside of the Champalimaud Foundation, it's possible to order the Harp devices in the [Open Ephys Production Site](#).

It's also possible to assemble the devices in-house with the right equipment since all of them are under an open source license. Some devices (like the Harp SyringePump) must be built in-house since they are not sold externally.

Item	Description	Amount	Observations
Harp Behavior	General-purpose Harp board	1	-
Harp SoundCard	Delivers the auditory stimulus	1	-
Harp Audio Amplifiers	Amplifies the auditory stimulus	2	1 Harp Audio Amplifier per speaker
Harp ClockSynchronizer	Synchronizes the timestamps from every Harp device	1	-
Harp SyringePump	Device used for reward delivery	2	-
Harp CurrentDriver	Drives/controls the LED/laser used in optogenetics	1	Optional

Peripherals

For this part of the BOM, a peripheral is considered to be anything that either interacts with the Harp devices or is needed make the different devices work and/or interact with each other.

Item	Description	Amount	Part Number	Part of the Harp Device kit	Agendo	Observations
Behavior Poke Port Breakout v1.1	Makes the ethernet ports pins from the Harp Behavior available	3	-	✗	✓	Connects to the Harp Behavior
RJ-to-RJ cables	-	3	IM1037	✗	✗	Connects the Harp Behavior to the Behavior Poke Port Breakout v1.1
Poke Small v1.1	Board with infrared beam to detect animal pokes	3	-	✗	✓	Connects to the Behavior Poke Port Breakout v1.1

Item	Description	Amount	Part Number	Part of the Harp Device kit	Agendo	Observations
3.5 mm-stereo-audio-jack-to-bare-wires	-	3	BC-A3ML006F	✗	✗	Connects the Poke Small v1.1 to the Behavior Poke Port Breakout v1.1
5 mm white/blue LED	Placed on the box lid to give cues to the animal	1	-	✗	✗	Connects to the LED0 pins of the Harp Behavior
3 mm green LED	Placed in the central poke to give cues to the animal	1	-	✗	✗	Connects to the LED1 pins of the Harp Behavior
RCA-to-RCA cables	-	2	-	✓	✗	Connects each Harp Audio Amplifier to the Harp SoundCard
Speakers	Deliver the auditory stimulus	2	TODO	✗	✗	1 speaker per Harp Audio Amplifier
Banana Plug	4 mm Triple Contact Plug (Black or Red) - Pack of 10 units	4	557-0100	✗	✗	Connects the speakers to the Harp Audio Amplifiers
10 ml Glass Syringe	Hamilton 1000 Series Gastight Syringes: Luer Lock Syringes,	2	Hamilton 81620	✗	✗	Used for the Harp SyringePump

Item	Description	Amount	Part Number	Part of the Harp Device kit	Agendo	Observations
	TLL Termination					
4-way Stopcock	Pack of 10 units	2	Masterflex 30600-04	✗	✗	Attaches to the end of the syringe (1 per Harp SyringePump)
Nylon Male Luer Fitting	Pack of 25 units	2	Masterflex MFLX45505-31	✗	✗	Attaches to one end of the stopcock (1 per Harp SyringePump)
Nylon Female Luer Fitting	Pack of 25 units	2	Masterflex 45502-00	✗	✗	Attaches to one end of the stopcock (1 per Harp SyringePump)
Spouts	Used in reward delivery	2	B14200 100 BULK	✗	✗	Used in reward delivery. Glued to the physical lateral pokes
Flexible Tubing	Masterflex Tygon E-3603 Non-DEHP Tubing - 15 meters per unit	2	Masterflex 06407-71	✗	✗	Connects the spouts to one of the luer fittings (either male or female)
Mini USB cable	-	5/6*	AK-300130-018-S	✓	✗	Connects the Harp boards to the computer
Micro USB cable	-	1	-	✓	✗	Used to upload sounds to the Harp SoundCard
3.5 mm-stereo-	-	4/5*	TODO	✗	✗	Connects the Harp ClockSynchronizer to

Item	Description	Amount	Part Number	Part of the Harp Device kit	Agendo	Observations
audio-jack-to-jack cable						every other Harp device
12V Power Supply **	AC/DC Wall Mount Adapter 12V 12W	9/10*	VER12US120-JA	✓ ***	✗	1 is used to power the LED strip that illuminates the behavior box, the remaining are used to power the Harp devices
BNC-to-bare-wires conector **		1	810-4605	✗	✗	Connects the LED strip to the power supply

* if the Harp CurrentDriver is being used

** the number of power supplies can be decreased as explained in the subsection [below](#)

*** the power supply used to power the LED strip that illuminates the behavior box must be ordered separately

Reducing the number of power supplies

From the table above, it's pretty noticeable that a lot of power supplies are required, which takes up a lot of space and power outlets. It's possible to use the same power supply for different devices according to their characteristics. The devices can be grouped in the following way:

- Harp Behavior, Harp SoundCard and Harp CurrentDriver*
- V+ of the Harp Audio Amplifiers
- V- of the Harp Audio Amplifiers
- Both Harp SyringePumps

From the list above, it's possible to switch the last line from the previous table with the lines from the table below.

Item	Description	Amount	Part Number	Comes with Harp Device	Agendo	Observations
12V Power Supply	AC/DC Wall Mount Adapter 12V 12W	5	VER12US120-JA	✓ ***	✗	1 is used to power the LED strip that illuminates the behavior box, the remaining are used to power the Harp devices
2-to-1 cables for the 12V power supplies	-	4/3*		✗	✓ ***	-
4-to-1 cables for the 12V power supplies	-	0/1*		✗	✓ ***	-

* if the Harp CurrentDriver is being used

Mechanical Components

This section contains the mechanical components of the setup that were developed and assembled in-house.

⚠️ WARNING

The files for the mechanical components are not currently available online and can't also be ordered, so people outside of the Champalimaud Foundation will have to develop and assemble their own.

Item	Description	Amount	Part Number	Observations
Behavior box	-	1	-	Made of acrylic
Physical pokes	-	3	-	Preferably made of metal, but can also be 3D printed
Speaker holder	-	2	-	Preferably made of metal, but can also be 3D printed
Speaker holder pole	Hollow alluminium tube	2	-	-
Box LED holder	-	1	-	3D printed

Camera

This section contains the hardware needed to setup the camera and fix it to the lid of the behavior box. Click [here](#) to go to the camera configuration instructions.

Item	Description	Amount	Part Number	Observations
FLIR Camera ↗		1	BFS-U3-16S2M-CS	-
Camera Lens ↗		1	A4Z2812CS-MPIR	-
Camera USB cable ↗	USB-A to Micro-B Straight with Thumbscrews, 3 Meters	1	CEI USB3-1-1-2-3M	This cable MUST be connected to a USB 3.0 port for performance
Camera GPIO cable ↗	6 Pin Female Straight Plug (Hirose HR10A-7P-6S) to Flying Leads, 3 Meters	1	CEI MVA-50-3-X-3	Connects to the Harp Behavior to trigger/monitor the camera frames
Tripod Adapter ↗	BFS 30 mm BFLY CM3 Tripod Adapter	1	ACC-01-0003	Used to fix the camera to the Thorlabs poles in the behavior box
Thorlabs 75mm post	Ø12.7 mm Optical Post, SS, M4 Setscrew, M6 Tap, L = 75	1	TR75/M	Connects to the camera

Item	Description	Amount	Part Number	Observations
mm				
Thorlabs 150mm post	Ø12.7 mm Optical Post, SS, M4 Setscrew, M6 Tap, L = 150 mm	1	TR150/M	Fixes the camera structure to the box lid
Thorlabs post clamp	Rotating Clamp for Ø1/2" Posts, 360° Continuously Adjustable, 5 mm Hex	1	SWC/M	Fixes both Thorlabs posts to each other
M6 Screw	15/16 mm	1		Fixes the longer post to the behavior box lid
M6 Setscrew	M6 x 1.0 Stainless Steel Setscrew, 16 mm Long, 25 Pack	1	SS6MS16	Connects the Thorlabs post to the Camera

Others

This section contains the remaining components needed for the setup. None of these components needs to be the exact model present on the list. It just corresponds to the models that have been used by the lab, but feel free to use different ones, as long as they work.

Item	Description	Amount	Part Number	Observations
Computer	Mini PC Blackview MP100 Mini PC AMD Ryzen 7 5825U/16GB/512GB SSD	1	MP100(16+512)-BLACK	-
Screen	-	1	-	-
HDMI cable	-	1	-	-
Keyboard	-	1	-	-
Mouse	-	1	-	-
USB Hub	HUB USB 3.0 TP-Link UH700 7 Ports	1	P004616	-
Power extension with 6 electrical	Must be appropriate to 90° plugs	1		To plug all of the power supplies in a setup

Item	Description	Amount	Part Number	Observations
outlets				
<u>KVM switch</u>		-		Optional: useful in case one wants to use the same screen + mouse + keyboard kit in different computers
12V LED Strip		1		-
Optogenetics Light Source	-	-	-	Optional

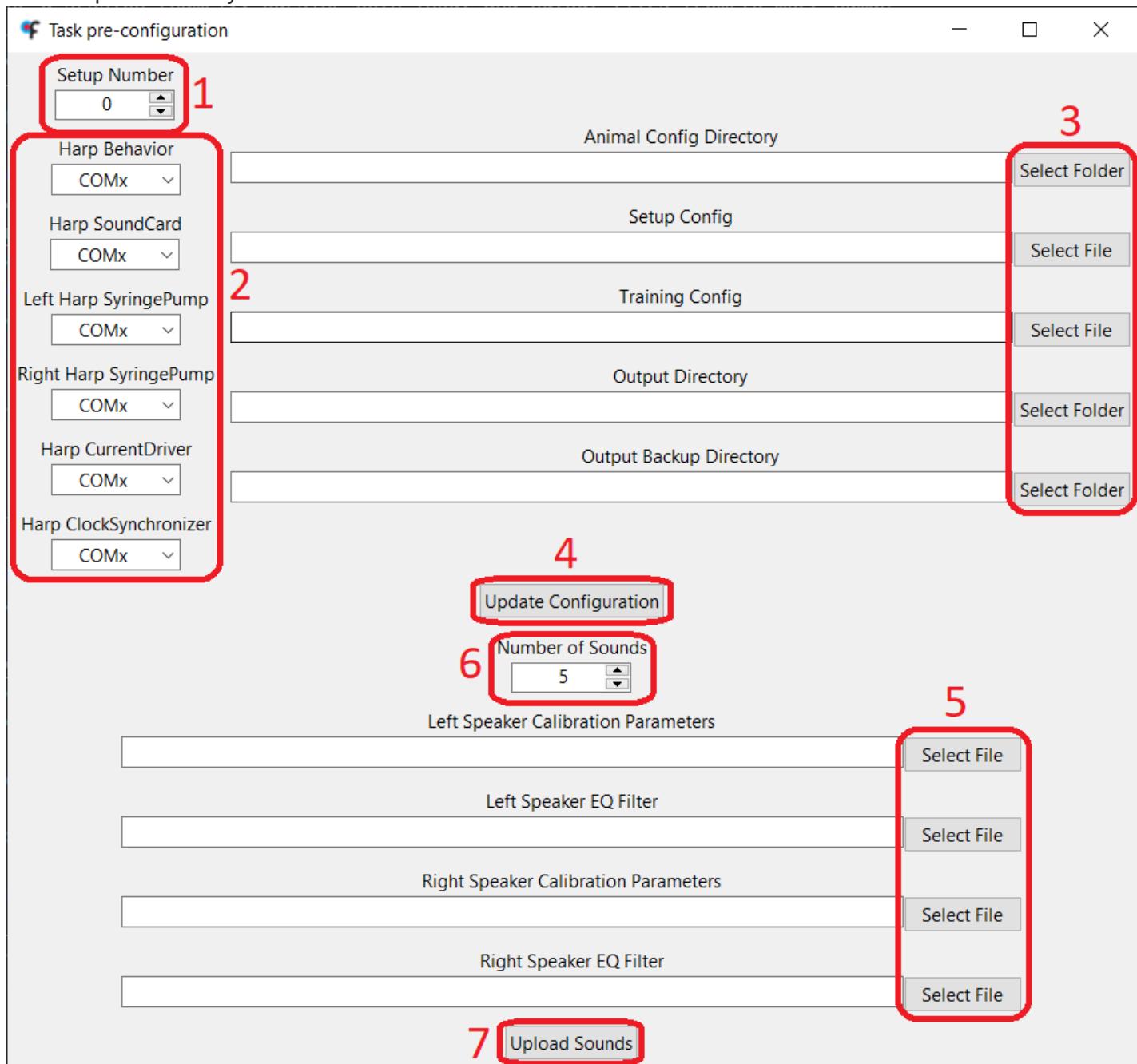
Software Installation

The task's software is composed by the Bonsai workflow, which contains the task logic, and by small Python scripts that perform some operations at the beginning and at the end of the session (namely, parsing both input and output files from/to more human readable formats). In order to facilitate the deployment of the project and with reproducibility in mind, the Bonsai workflow was developed inside a Bonsai environment and the Python scripts uses [uv](#) to create and maintain the Python virtual environment.

To install the task's software, follow the steps:

1. Download the source code from the latest project [release](#) and unzip it. Alternatively, clone the [repository](#).
2. Run `./Setup.cmd` to install the Bonsai environment, the Python environment and FFmpeg and to download the necessary configuration files (`template.yml`, `setup.csv`, `training.csv`) into `./config` directory (in case these don't exist already). An application window will appear which is used to specify the setup number, the COM port of each Harp device and the paths to the configuration files

and output directory.



1. Select the setup number.
2. Specify the serial port that corresponds to each Harp device.
 - The setup doesn't require a Harp CurrentDriver to work. This device is only used in optogenetics sessions, so if this is not going to be the use case, the serial port for this device can be ignored.
3. Select the paths to the different input and output files.
 - The configuration files downloaded to the `./config` directory can be moved elsewhere!
 - The `template.yml` must be inside the animal config directory!
 - The animal config directory is where the `[animal_id].yml` file of each animal will be created inside the corresponding batch folder.
 - The output backup directory is *optional*. Video-related data will not be saved to this directory.

4. Click on the **Update Configuration** button, which will create the `./src/config/config.yml` file, which is required to run the task.
 - Please don't move or delete the `./src/config/config.yml` file! If, for some reason, any of the paths or COM ports need to be changed, re-run the `Setup.cmd` script.
5. To upload the sounds to the Harp SoundCard, start by selecting the files containing calibration parameters and EQ filter of each speaker. These files should have the `.npy` extension and were generated during [speaker calibration](#) procedure.
6. *Optional:* Select the number of sounds to be uploaded to the Harp SoundCard.
7. Click on the **Upload Sounds** button and wait until the process is finished (the button will be enabled again when it happens). The uploaded sounds are 10-second white noises with a sampling rate of 192 kHz.

Camera Drivers

In order to run the Bonsai workflow that contains the task's logic, the camera drivers for the setup's camera must be installed. Instructions to install the drivers for both [FLIR](#) and [Point Grey](#) cameras can be found below.

FLIR

In order to use the FLIR camera to record the sessions, install the [Spinnaker drivers](#). It's mandatory that the computer has the version 1.29.0.5 of the drivers installed, since the version supported by Spinnaker Bonsai Package. After opening the installer, follow the steps below:

1. Click on **Next**. Then, accept the terms and click on **Next** again.
2. Select **Application Development** and click on **Next**.
3. Deselect **GigE Driver** and click on **Next**.
4. Deselect the **I will use GigE Cameras.** checkbox and click on **Next**.
5. Finally, click on **Install**.

Point Grey

In order to use the Point Grey camera to record the sessions, install the FlyCap2 drivers (version 2.11.3.425). After opening the installer, follow the steps below:

1. Select **FlyCapture2_x64.msi** and click on **Install**.
2. Click on **Next**. Then, accept the terms and click on **Next** again, twice.
3. Deselect the **Automatically register with PGR via the internet** checkbox and click on **Next**, twice, again.
4. Click on **Complete**.
5. Select the **I will use USB cameras** checkbox and click on **Next**.
6. Select the **Click to confirm** checkbox and click on **Next**, twice.
7. Finally, click on **Install**.

FFmpeg

The task's code makes use of the FFmpeg software to save the video recordings in the disk, because it allows the video to be recorded with higher framerates, without loss of image quality and by making a better use of the computer's resources than the native [VideoWriter](#) Bonsai node.

FFmpeg can be installed with WinGet by running the following command in the Terminal:

```
winget install Gyan.FFmpeg
```

Harp Devices Setup

In order to use the Harp devices, the USB drivers and each device's firmware must be installed. The installer for the USB drivers can be downloaded [here](#). The [Firmware](#) section contains the instructions for installing each board's firmware.

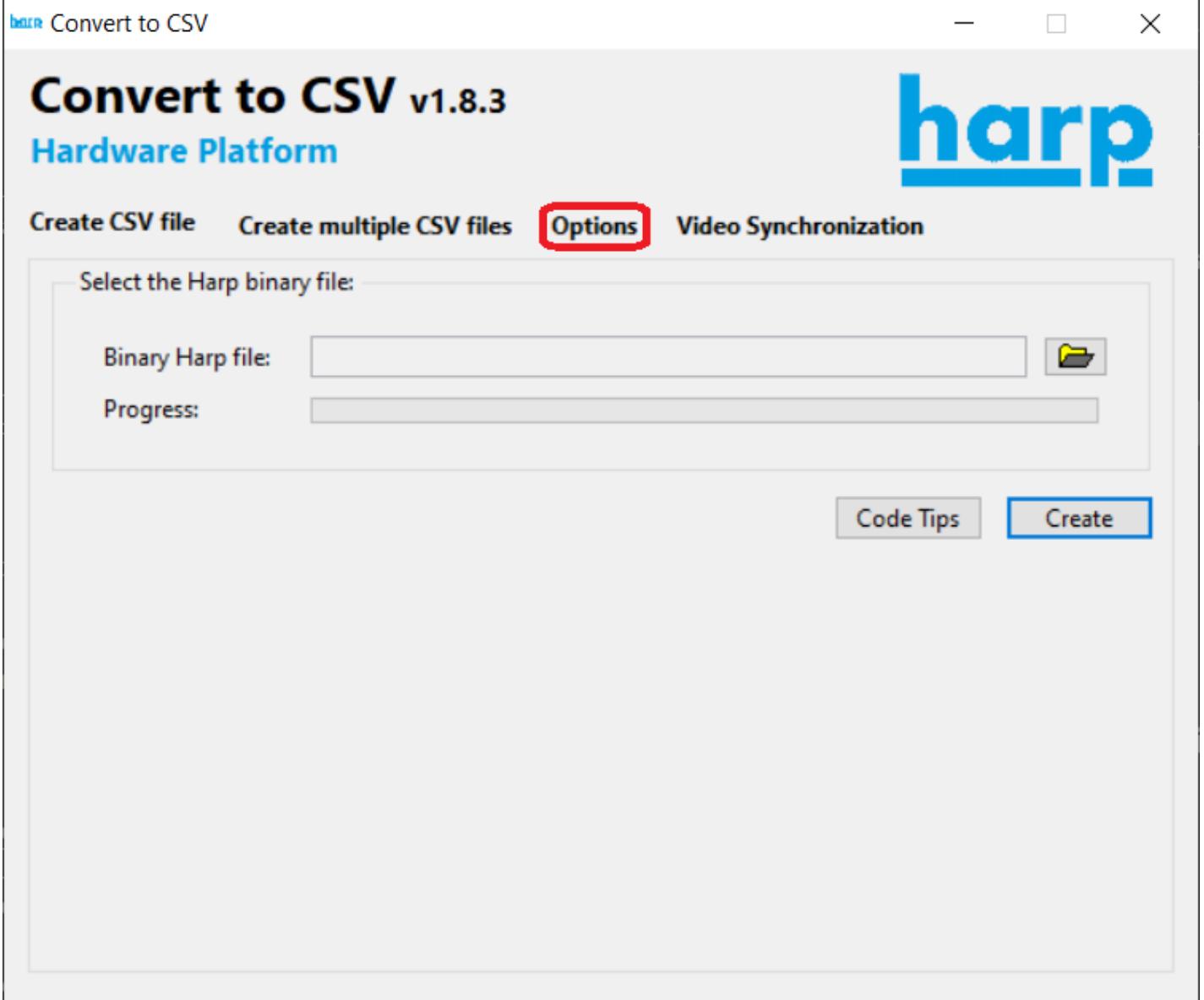
NOTE

The Harp devices are usually delivered with the firmware already installed, but it may be useful to know how to install the firmware (for example: in case a new firmware version that solves a particular bug is released)

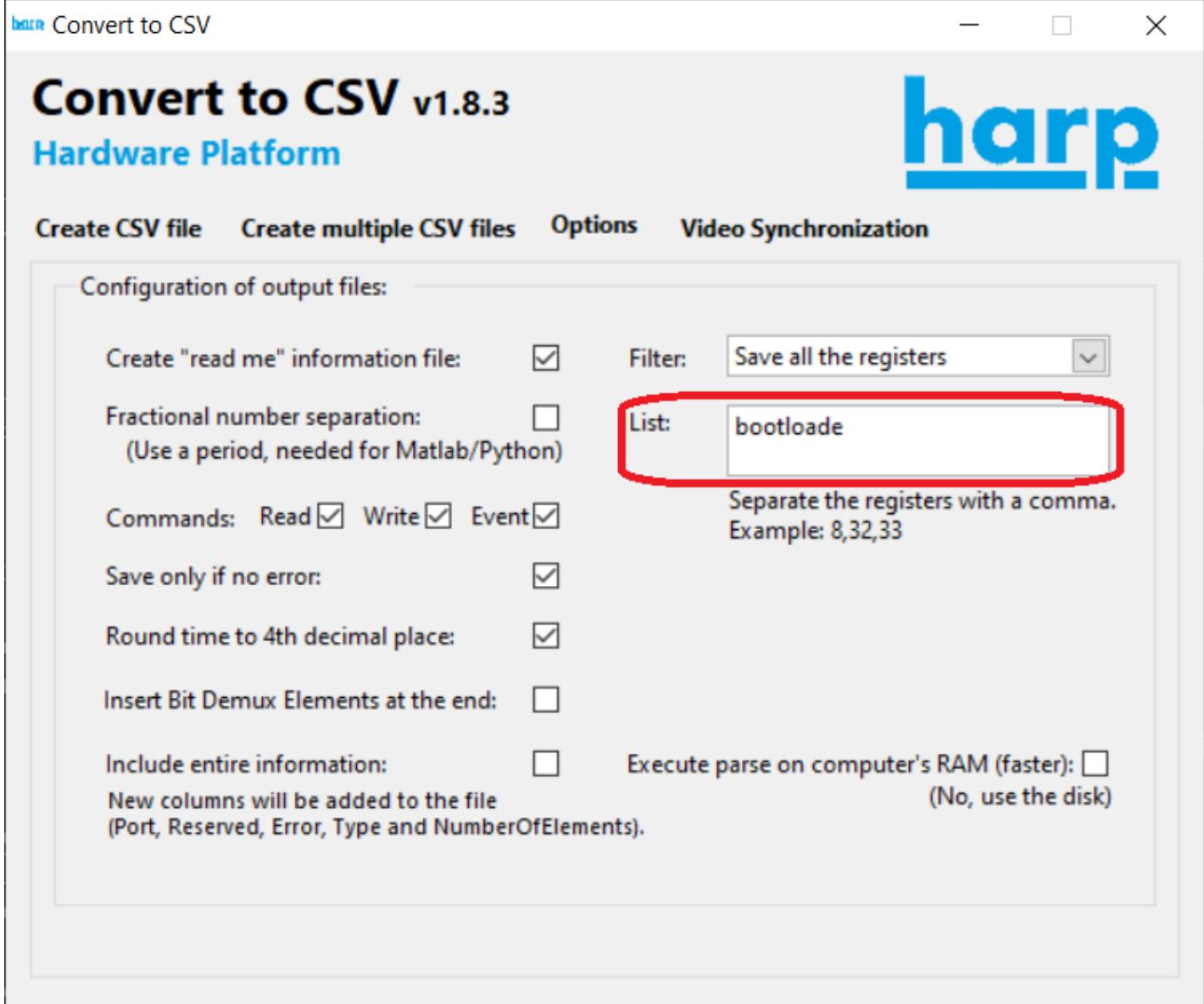
Firmware Installation

If the Harp Convert to CSV GUI is already installed, skip to step 3. If it's not already installed, but other Labview-based Harp board GUI is, skip to step 2 instead.

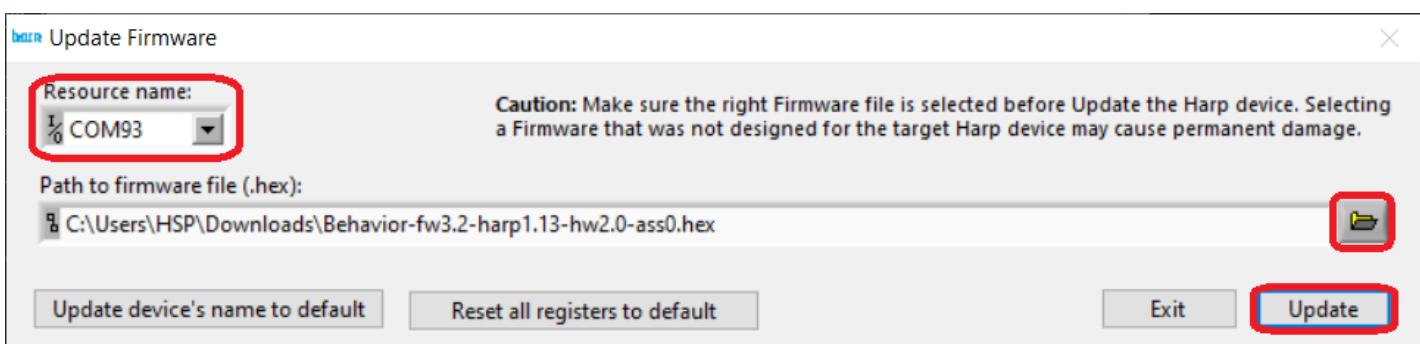
1. Install the [LabView Runtime](#) and reboot the computer.
2. Install the [Harp Convert to CSV GUI](#).
3. Open the application. Then, click on [Options](#).



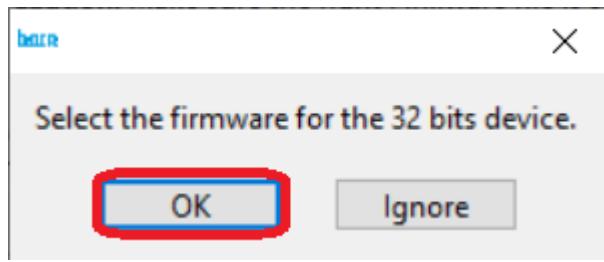
4. Write "bootloader" in the **List** textbox. The **Update Firmware** window should appear.



5. Choose the serial port (COMx) of the board and the corresponding firmware binary (.hex file). Then, click on **Update**.



6. **For the Harp SoundCard:** during the installation of the firmware, select the PIC32 firmware when the application asks for the 32 bits device firmware ([SoundCard.X-fw2.0-harp1.4-hw2.2-ass0.hex](#)).



Firmware Versions

You can find the firmware versions currently being used in the table below.

Device	Firmware Version	Harp Core Version	Hardware Version	PIC32 Firmware
Harp Behavior	v3.2	v1.13	v2.0	-
Harp SoundCard	v2.3 (Custom)	v1.4	v2.2	v2.3 (Custom)
Harp ClockSynchronizer	v1.0	v1.13	v1.0	-
Harp SyringePump	v1.0	v1.13	v1.1	-
Harp CurrentDriver	v0.4	v1.13	v1.1	-

NOTE

The source code for the custom firmware of the Harp SoundCard can be found [here](#).

Calibration Protocols

Some of the hardware used in the task requires calibration. Since the calibration protocols are already written somewhere else, they can be found by clicking in the name of the protocols from the list below:

- [Speaker Calibration ↗](#)
- [Harp SyringePump Calibration ↗](#)
- Optogenetics LED Calibration (*Not available yet*)

Camera Configuration

The setups need a camera to record the sessions and to monitor the task in real time. Since there's a need for the video to be recorded with some specific settings, this section contains instructions explaining how to configure PointGrey and FLIR cameras, which are the two types of cameras used.

Desired Camera Settings

As mentioned above, there is a need for the camera used in the setup to have some specifications, namely:

- a frame rate of 100 fps.
- the ability for the camera to send a pulse to the Harp Behavior every time a frame is being acquired.
- the ability to receive a pulse from the Harp Behavior when the animal is poking in the central port (by making one of the Behavior's digital outputs mimic the central port digital input).

FLIR

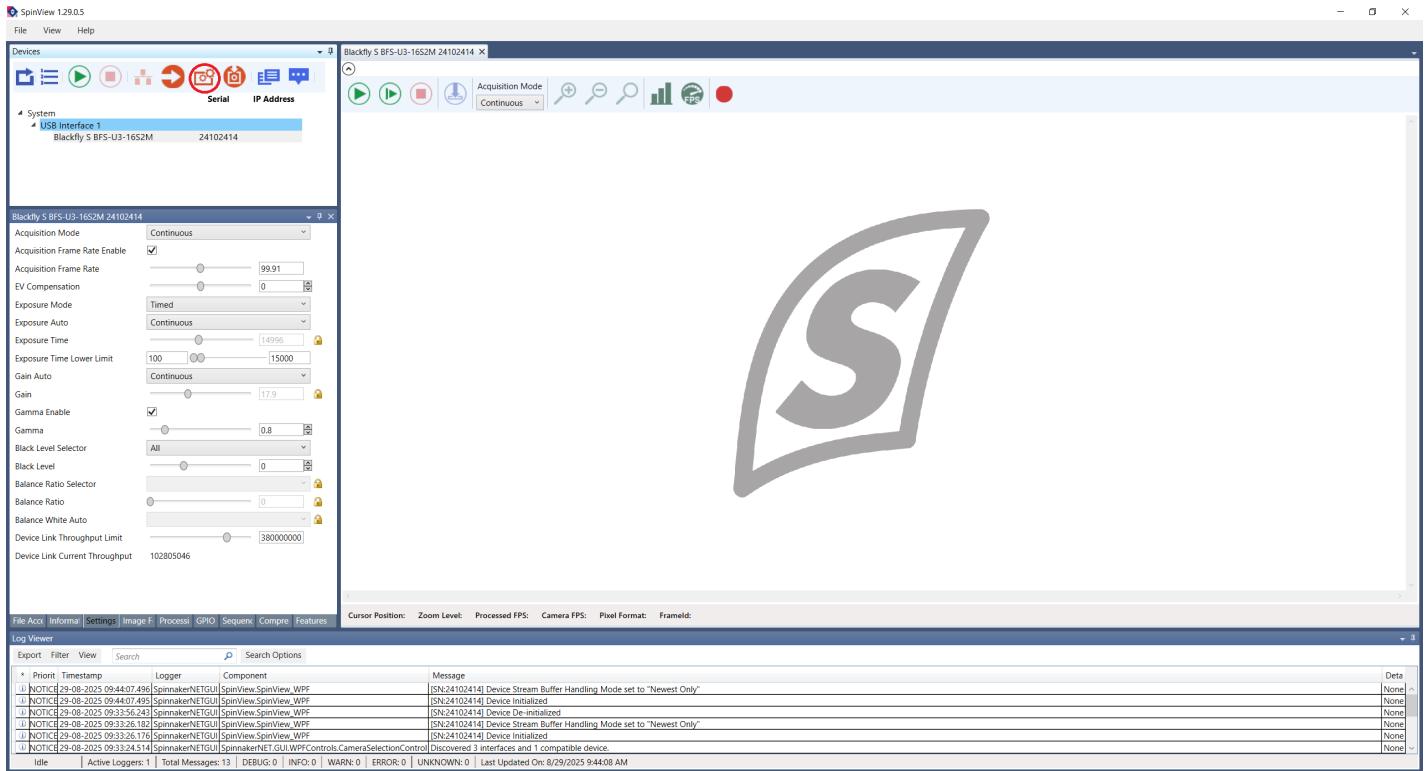
After installing the [Spinnaker Drivers](#), connect the camera to the computer in an USB-3.0 port (connecting it to an USB-2.0 port limits the camera's capabilities, namely the frame rate).

Firmware Update

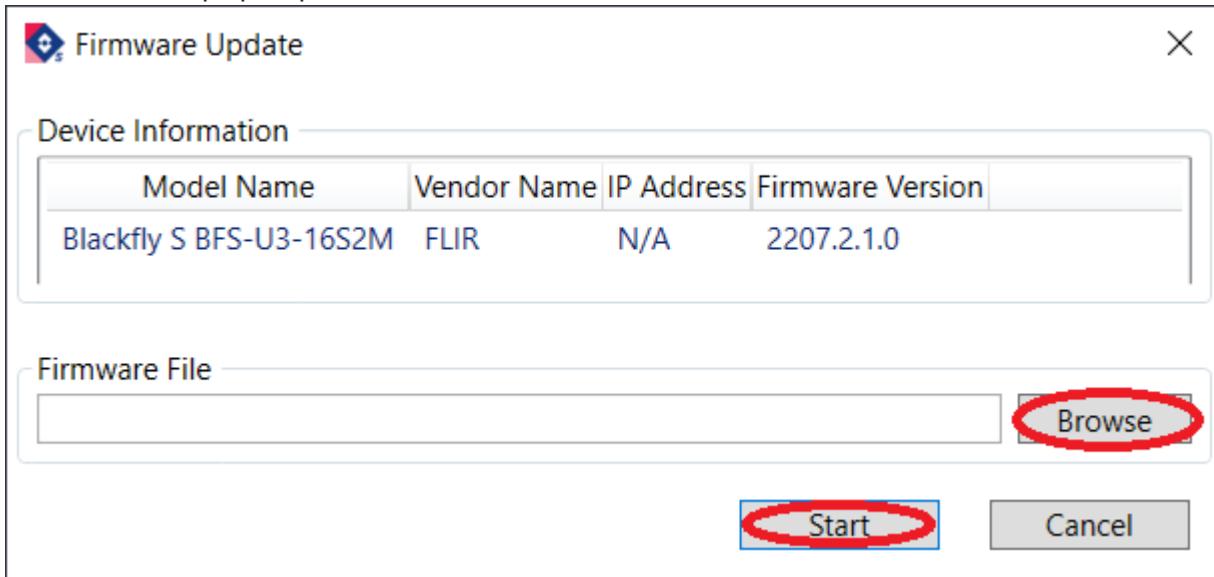
In order to access the state of the GPIO pins in the camera model BFS-U3-16S2M-CS, it's necessary that the firmware version [2207.2.1.0](#) is installed. Follow the instructions to verify the firmware version currently installed and install a different version in case it's necessary.

1. Open the SpinView software.
2. Select the camera to be configured.

3. Click on the Update Device Firmware icon.



4. After a new window pops up, click on its **Browse** button to choose the firmware file. Then, click on



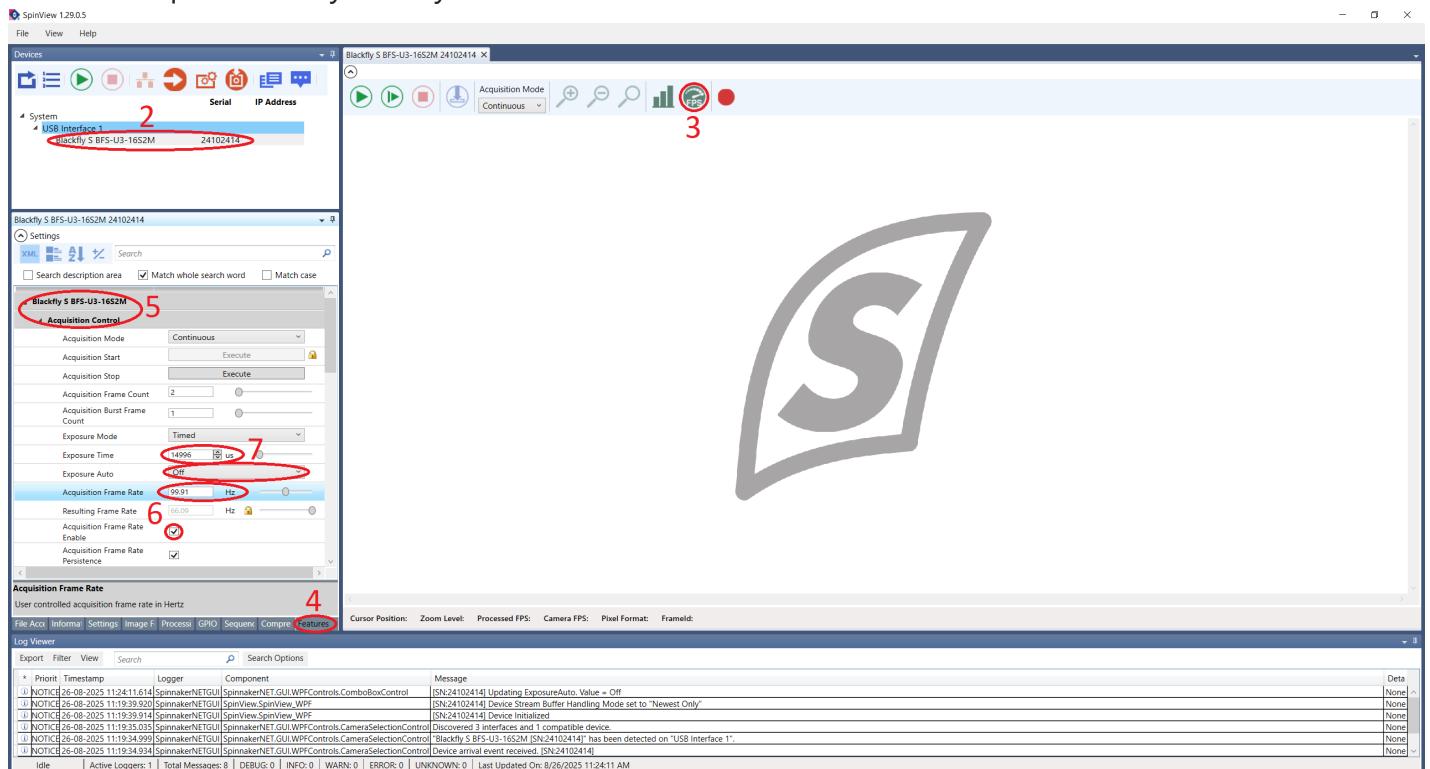
Camera Configuration

After having the correct firmware version installed, follow the instructions to configure the camera.

1. Open the SpinView software.
2. Select the camera to be configured.
3. Click on the **Optimize camera frame rate** button (see the figure below).
4. Click on **Features**.
5. Click on the camera model name and then on **Acquisition Control**.

6. Change the **Acquisition Frame Rate** parameter to the desired value. Confirm that the **Acquisition Frame Rate Enable** setting is checked.

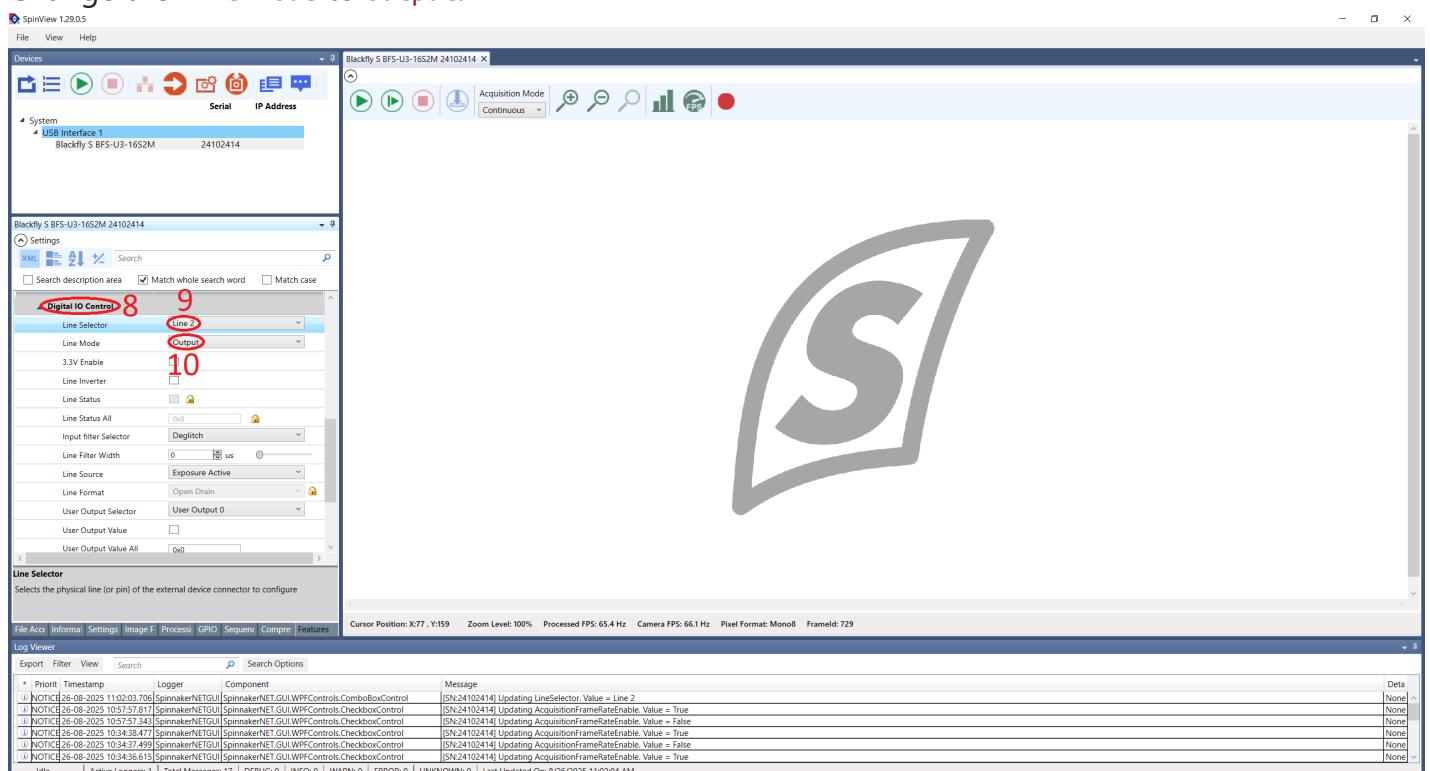
7. Change the **Exposure Auto** setting to **Off** and adjust the **Exposure Time** to the maximum possible so that the acquisition stays stably at 100 FPS.



8. Close the **Acquisition Control** section and open the one named **Digital IO Control**.

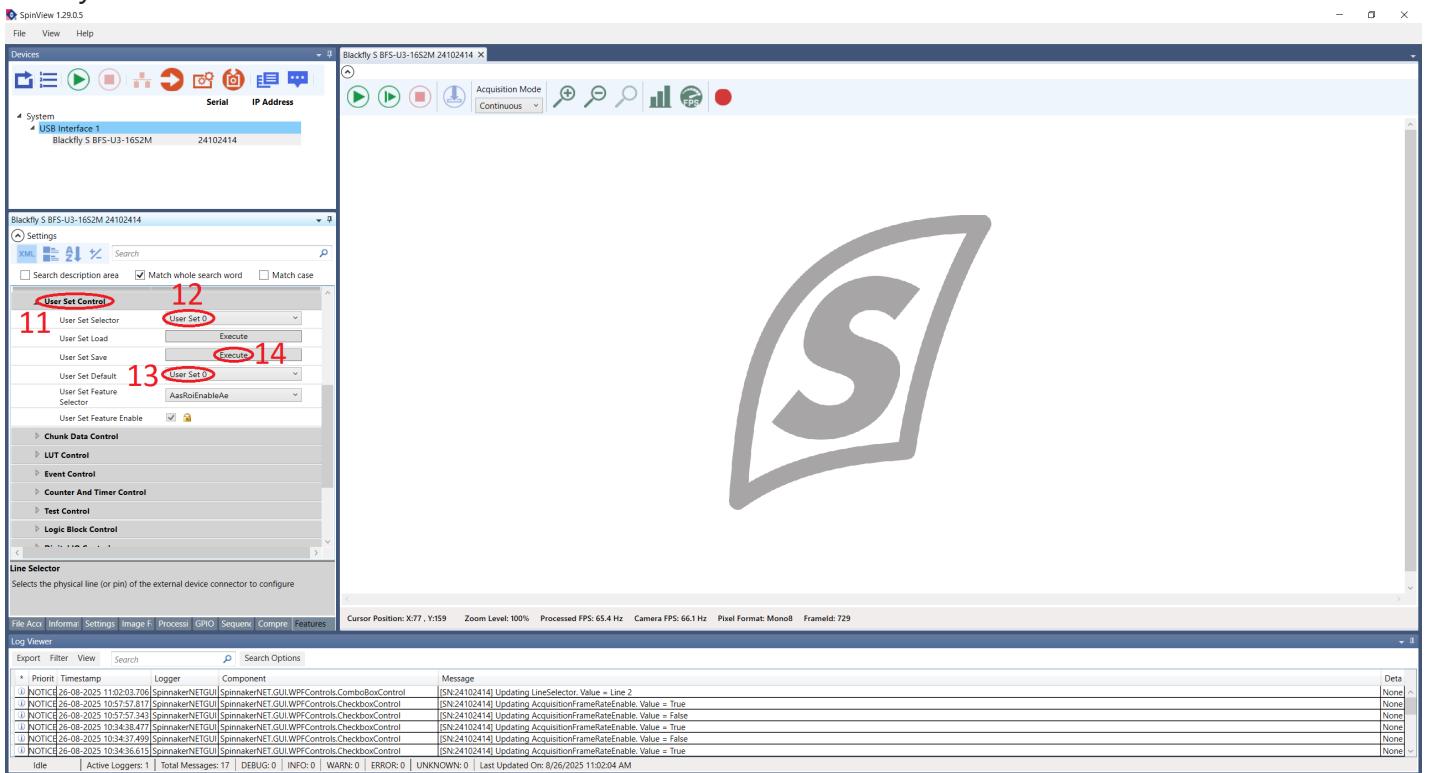
9. Select the **Line 2** in the **Line Selector** parameter.

10. Change the **Line Mode** to **Output**.

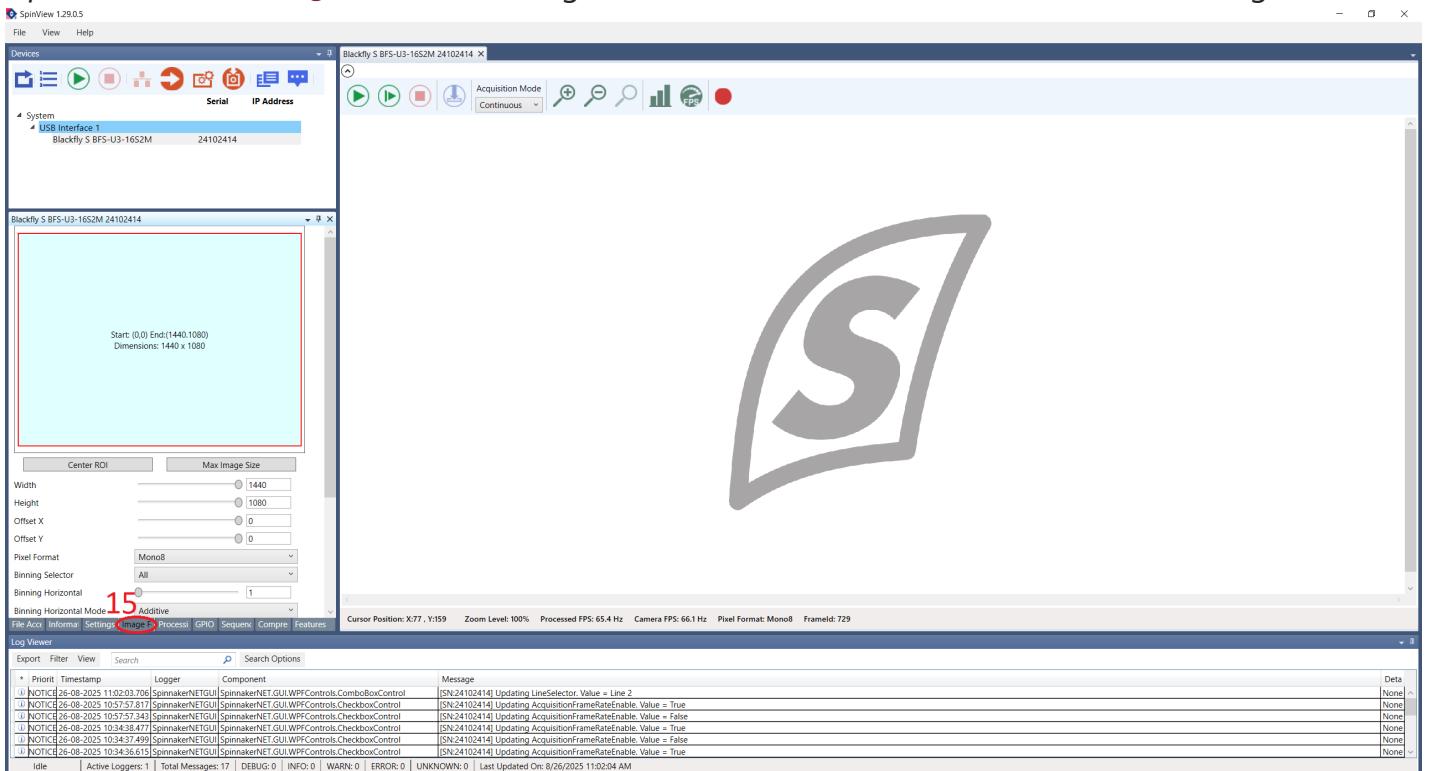


11. Close the **Digital IO Control** section and open the one named **User Set Control**.

12. Select an **User Set Selector** other than **Default**.
13. Select the same User Set in the **User Set Default** parameter.
14. Finally, click on the **User Set Save** button to save the current User Set.



15. Optional: Click on **Image Format** to change the camera resolution and save the User Set again.



Hardware Connections

In order for the camera to be able to communicate with the Harp Behavior, it's necessary that the [camera GPIO cable](#) is connected. The following connections shall be made:

- Line 2 (pin 3) from the camera to the Behavior's DI3 pin.
- Line 0 (pin 2) from the camera to the Behavior's DO0 pin.

! WARNING

Don't forget to connect the camera's grounds (pins 5 and 6) to the Harp Behavior's GND pins!

Blackfly S Cased Models

Diagram	Color ¹	Pin	Line	Function	Description
	Green	1 ²	3	V _{AUX}	Auxiliary Input Voltage (DC)
				GPI	Non-isolated Input
	Black	2	0	OPTOIN	Opto-isolated Input
				VOUT	Camera Power Output
	Red	3 ²	2	GPIO ³	Non-isolated Input/Output
	White	4	1	OPTOOUT ³	Opto-isolated Output
	Blue	5	N/A	Opto GND	Opto-isolated Ground
	Brown	6	N/A	GND	Camera Power Ground

1—GPIO cable assembly wire colors

2—Dual function pin

3—Open drain output, requires pullup resistor

GPIO Cable Color Code

The color code for the [CEI MVA-50-3-X-3](#) camera GPIO cable is as follows:

Pin	Color
1	White
2	Brown
3	Green
4	Yellow
5	Gray

Pin	Color
6	Pink
H	Shield

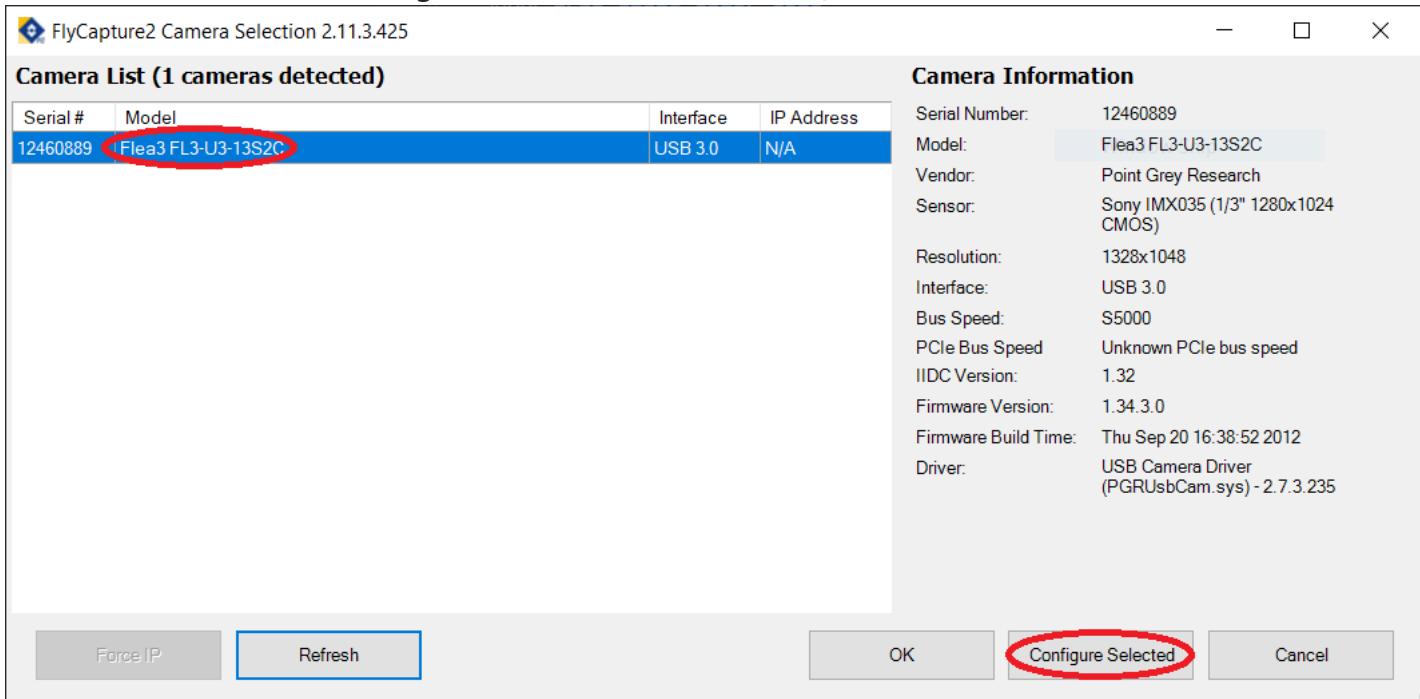
PointGrey

After installing the PointGrey Drivers, connect the camera to the computer in an USB-3.0 port (connecting it to an USB-2.0 port limits the camera's capabilities, namely the frame rate).

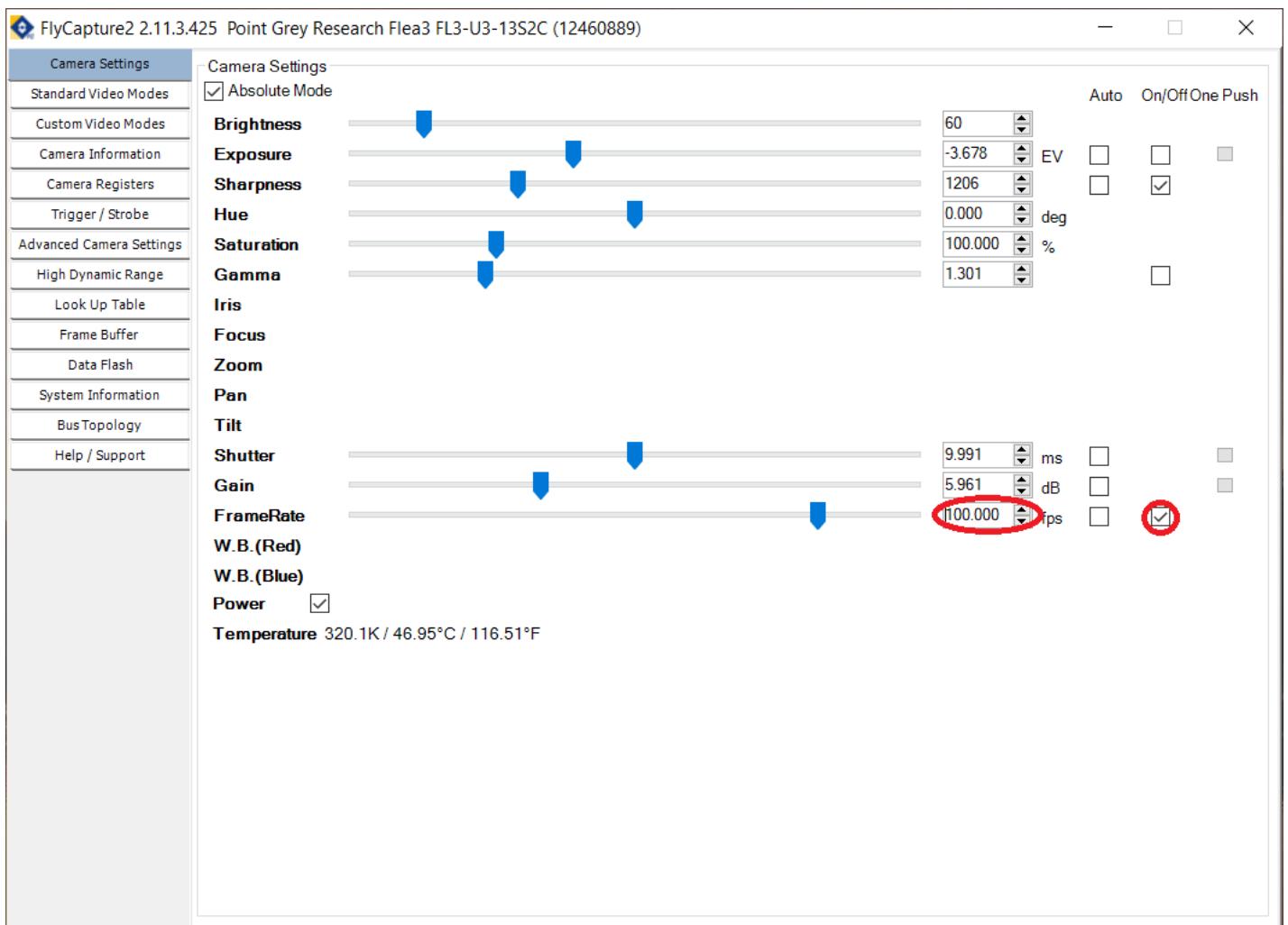
Camera Configuration

To configure a PointGrey camera follow the instructions below.

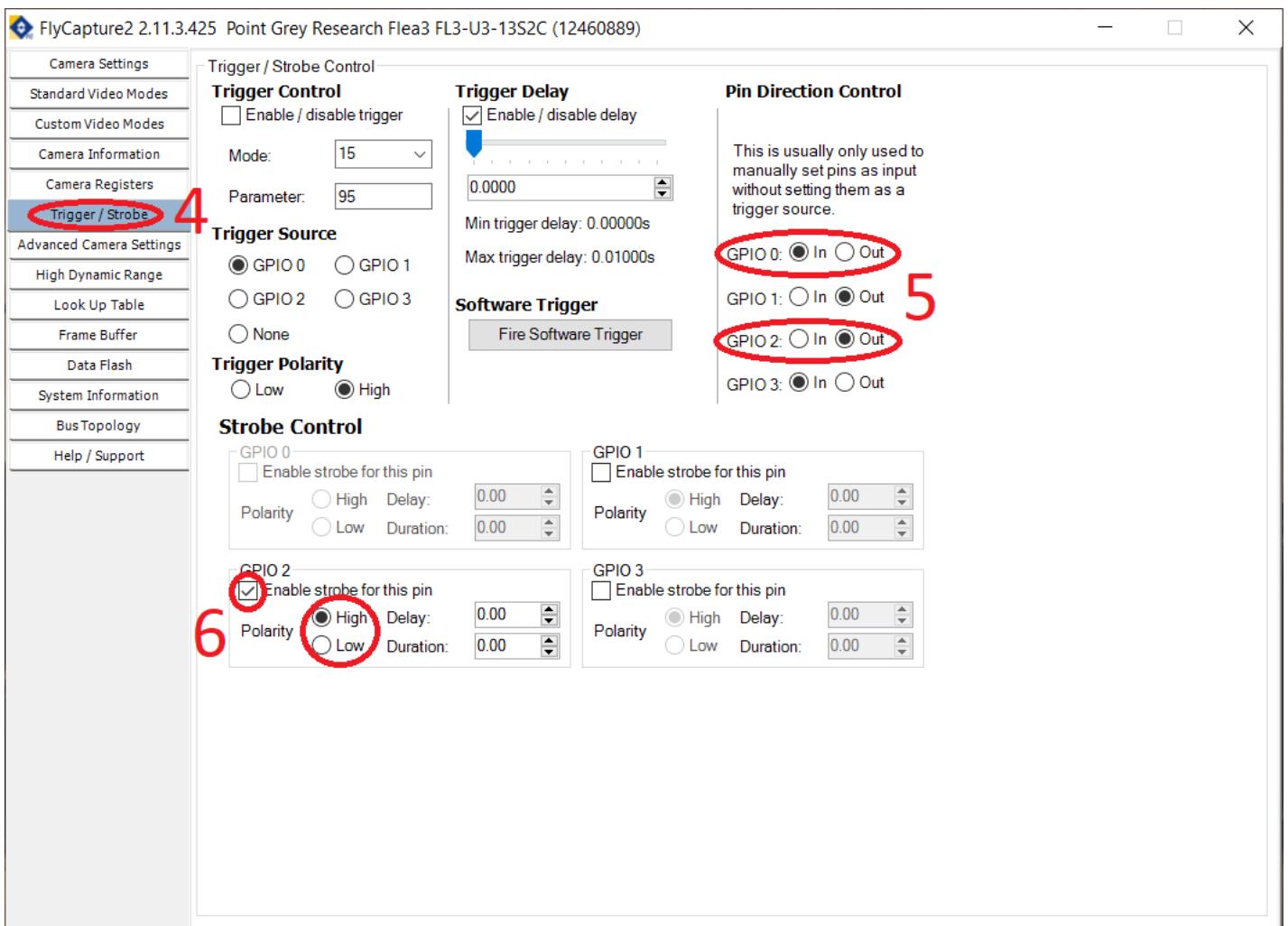
1. Open the Point Grey FlyCap2 software.
2. Select the camera to be configured and click on the **Configure Selected** button.



3. In the **Camera Settings** tab, activate the **On/Off** checkbox from the **FrameRate** row and modify its value.



4. Go to the **Trigger / Strobe** tab.
5. Select the GPIO 0 to be an input and the GPIO 2 to be an output.
6. Under **Strobe Control**, activate the **Enable strobe for this pin** checkbox for GPIO 2 and change the polarity to **High**.

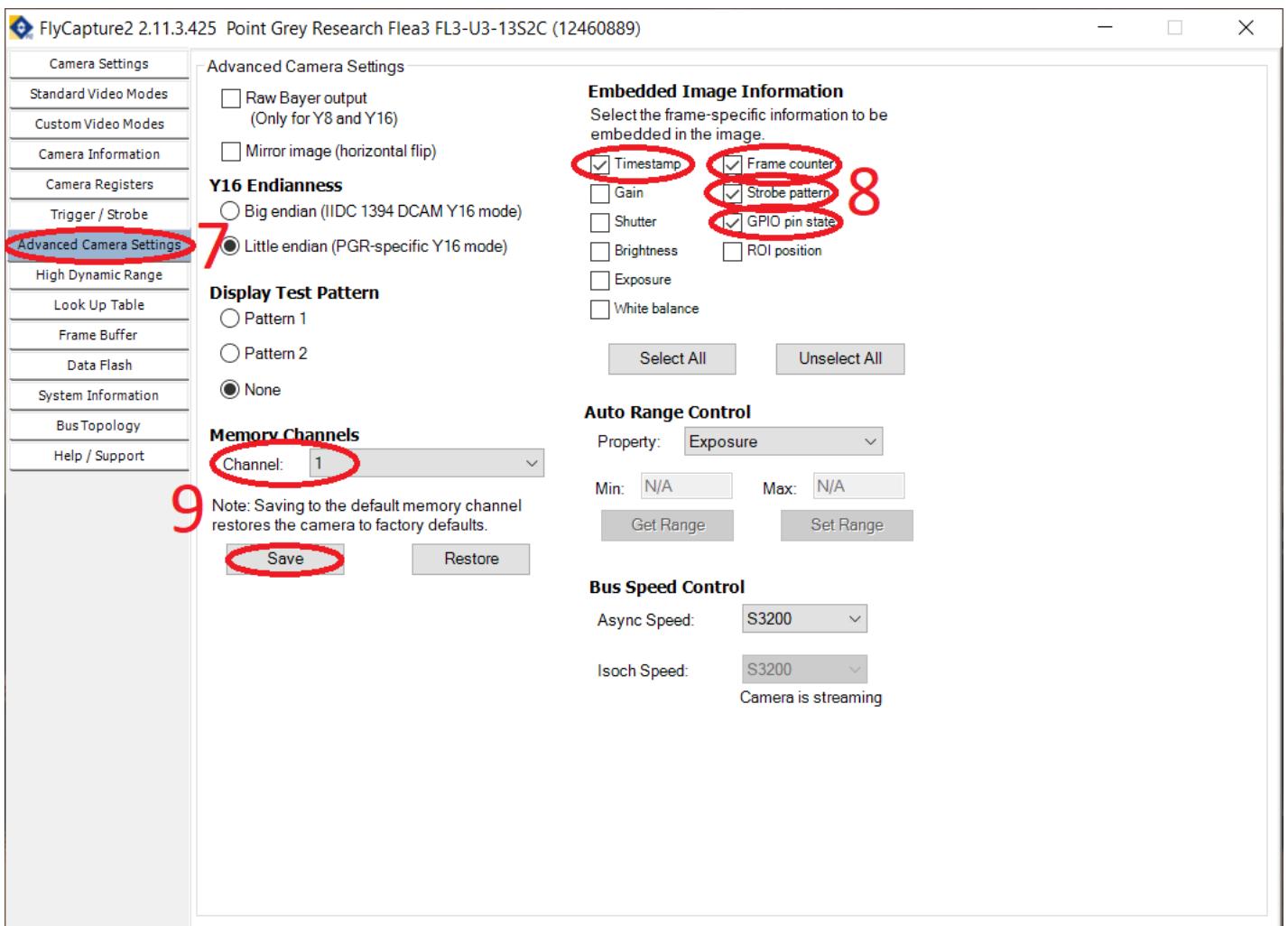


7. Go to the [Advanced Camera Settings](#).

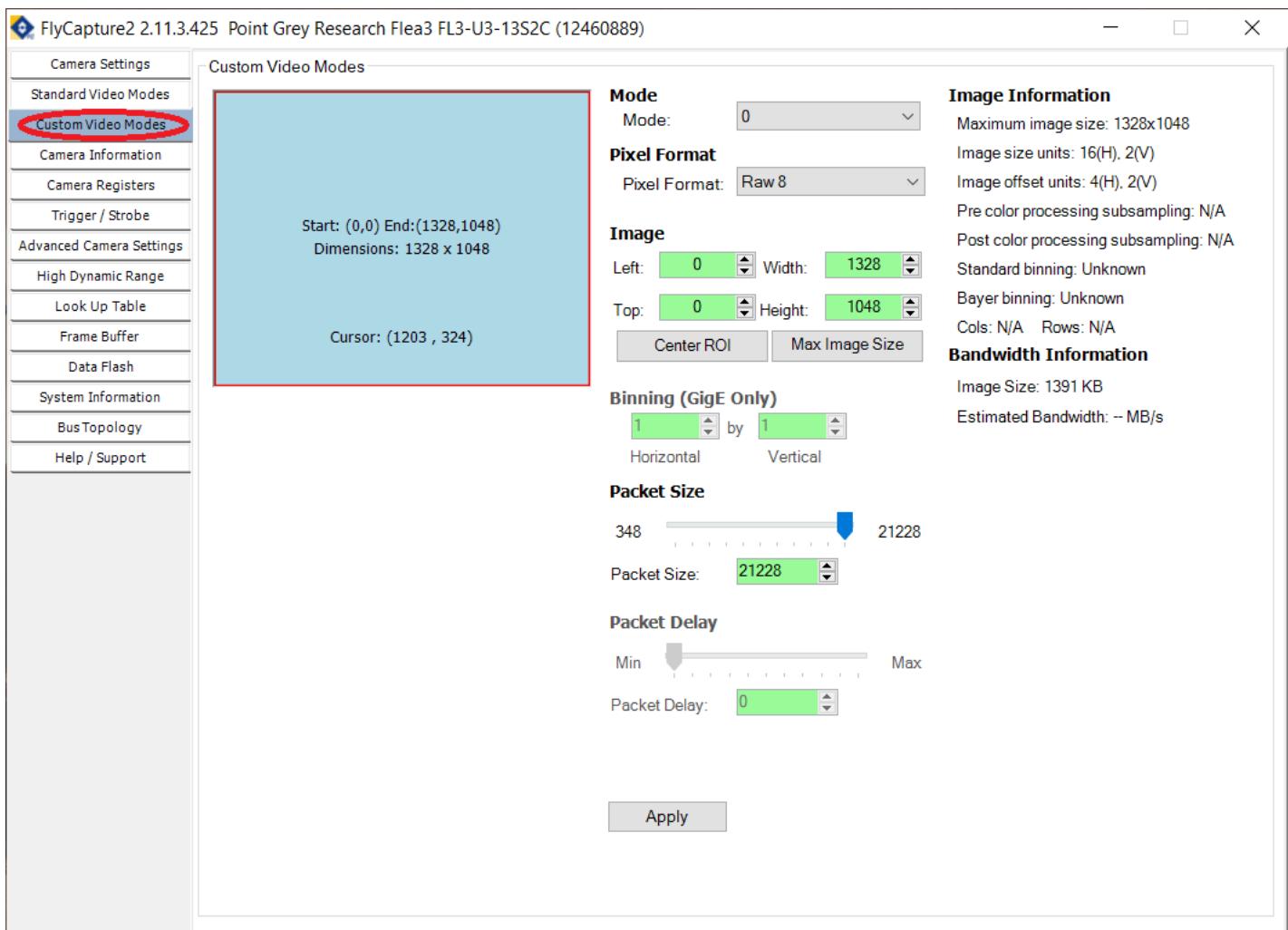
8. In the [Embedded Image Information](#), select the following fields:

- Timestamp
- Frame counter
- Strobe pattern
- GPIO pin state

9. In the [Memory Channels](#) section, select a channel different from [Default](#) and click on [Save](#) to save the current configuration.



10. Optional: Go to the **Custom Video Modes** and modify the camera resolution. Then save the camera configuration again.



Hardware Connections

Since the GPIO cable used by the PointGrey cameras is different from the one used by the FLIR cameras, the connections that need to be made between the camera and the Harp Behavior are also different. The connections should be made as follows:

- Pin 1 (IO0) from the camera to the Behavior's DO0 pin.
- Pin 3 (IO2) from the camera do the Behavior's DI3 pin.

! WARNING

Don't forget to connect the camera's grounds (pins 5 and 6) to the Harp Behavior's GND pins!

Diagram	Pin	Function	Description
	1	IO	Opto-isolated input (default Trigger in)
	2	O1	Opto-isolated output
	3	IO2	Input/Output/serial transmit (TX)
	4	IO3	Input/Output/serial receive (RX)
	5	GND	Ground for bi-directional IO, V_{EXT} , +3.3 V pins
	6	OPTO_GND	Ground for opto-isolated IO pins
	7	V_{EXT}	Allows the camera to be powered externally
	8	+3.3 V	Power external circuitry up to 150 mA

 **NOTE**

In the actual GPIO cable, the color of the cable corresponding to pin 5 might be purple instead of brown!

Task Configuration

After setting up both the hardware and the software needed for the task, there's just one more step to follow before start running the task: configuring it.

Despite the task having a [common structure](#), there are small variations to the task that can be achieved by tweaking innumerous configurations. Additionally, there's a need to input the calibration parameters for different pieces of hardware (speakers, SyringePumps, etc), which vary from setup to setup. Because of this, 3 different configuration files were created:

- [animal.yml](#) - This file contains task configurations that are not level-dependent and/or don't change throughout a session. Some of the parameters present in this file can be updated based on the previous session. Every animal has its own file ([\[animal_name\].yml](#)), so one can think of this file as some sort of ID. A description of each animal-specific setting can be found [here](#).
- [training.csv](#) - Generally, the configurations that can be set in this file are task parameters that can change a lot during the training phase of an animal. Each line of the file is a different training level the animal has to progress to/through. It can be assumed that when the animal reaches the last level, it's ready for the "actual" experiment. A description of each training-specific setting can be found [here](#).
- [setup.csv](#) - The settings that can be found in this file don't usually change the logic of the state machine, but are necessary for the setup to be working correctly (for example: equipment calibration parameters). Each line of this file is a different setup, so it's possible to have a single file stored in a drive containing the configurations for every setup and "point" to that file from the pre-configuration script (the application mentioned in the [Software Installation](#) page). A description of each setup-specific setting can be found [here](#).

Output Directory

It is intended that all data from every animal is saved in a single directory (the output directory). Look at the following example of an output directory.

```
output/
└── batch/
    └── RAT0001/
        └── 241001/
            ├── cam_metadata_143123.csv
            ├── out_RAT0001_241001.csv
            ├── RAT0001_241001_143123.mp4
            ├── config/
            │   ├── animal_143123.yml
            │   ├── setup_143123.json
            │   └── training_143123.csv
            └── events/
```

```

    └── 143123
        ├── behavior
        │   ├── behavior_0.bin
        │   ├── ...
        │   ├── behavior_122.bin
        │   └── device.yml
        ├── soundcard
        ├── left_pump
        ├── right_pump
        └── current_driver
    ├── plots/
    │   ├── block_1.png
    │   ├── block_2.png
    │   └── block_3.png
    └── unparsed_out/
        └── out_143123.json

```

Inside the output directory there is a folder for every animal. Inside each animal's folder there is a folder for every session day, whose name is in the `YYMMDD` format, and inside every session day folder there are different files and folders types:

- `out_[animal]_YYMMDD.csv` - this is the file that contains the final output structure for the session. This file is generated by a Python script that runs after Bonsai closes and that joins the different `out_hhmmss.json` files that can be found inside the `unparsed_out` directory.
- `[animal]_YYMMDD_hhmmss.mp4` - this is the video recorded during the session.
- `cam_metadata_hhmmss.csv` - this is the file containing metadata directly saved from the camera for every frame recorded in `[animal]_YYMMDD_hhmmss.mp4`, so for each frame it's possible to know the camera timestamp, the frame ID and the state of the camera GPIO pins.
- `config/-` - this directory contains a copy of the configuration files used in the session. The files are saved with the names: `animal_hhmmss.yml`, `setup_hhmmss.json` and `training_hhmmss.csv`.
- `events/-` - this directory contains a directory for each time Bonsai was initialized during the session named `hhmmss/`, which, in turn, has a directory for every Harp device (example: `behavior/`), which, in turn, has the binary files that contain every message sent by the device during the session (`[device_name]_X.bin`, where `X` is the register number) and the file containing the `device.yml` (the device metadata file).
- `plots/-` - this directory is where the figures containing plots with some metrics regarding the animal's performance are saved. A figure per block is saved with name `block_X.png` (where `X` is the block number).
- `unparsed_out/-` - this directory contains the raw output structure saved in (a fake) JSON directly from Bonsai and that will be converted into a single CSV file. The files saved follow the name `out_hhmmss.json`.

 **NOTE**

In order to avoid losing or overwriting data in case there's a need to restart the workflow, most of the files and directories are saved with a suffix `hhmmss` corresponding to the time at which the workflow was started.

 **CAUTION**

Don't store any other files in the output directory that are not generated automatically during a session and don't change the names of the files. Some operations - auto-update of configuration files, the numbers in the file names - depend on the files saved by the task's software and on the way they are saved.

If there's a need to manually save other files related to the experiment, please do it somewhere else.

Running the Task

After having everything else setup, the only thing left is to run the task. To do it, follow the steps:

1. In the task's directory, double-click on the `./Run.cmd` script.
2. Answer the initial prompts. Then, the bonsai workflow will open and the task will start with the GUI that allows the user to monitor the task in real-time and interact with some aspects of it.

Initial prompts

The goal of the initial prompts is to ease up launching the task without the need for the user to change the configuration files manually every session, specially for parameters that depend on the data saved in the previous session (like the trial number) and for repetitive operations (like choosing the animal the will perform the task). Below, you can find the prompts with a brief description of what it does and what answers are valid.

1. Hello! :) Let me know who you are, please:

This prompt only accepts letters and spaces and the first character **must** be a capital letter. Examples of valid answers are: `John Doe` or `JD`.

2. Are you training an animal from which batch?

This prompt accepts alphanumeric answers (with underscores, but the underscore can't be the first or last character). An example of a valid answer is: `batch_name`.

3. Which furry friend is going to be joining us?

This prompt accepts an animal ID, which is composed by 2 to 6 capital letters followed by 4 digits: Examples of valid answers are: `ANIMAL0000` or `RAT0055`.

4. Get parameters from last session (fixation time, reaction time, lnp_time and training_level)? [y/n]

This prompt only accepts `y` and `n` as valid answers and only appears in case the animal input in 3. already started training previously. If the answer is affirmative, the values from fixation time, reaction time, LNP time and training level are read from the last trial done by the animal and written in the animal's `animal.yml` file.

5. What is the training level the animal should stop progressing? (You can leave blank if the animal can progress until the last level)

This prompt only accepts a number up to 2 digits.

In addition to the prompts mentioned above, the startup script also looks at the data saved from the last session and updates the session number and the block number automatically.

Testing the Setup

The setup is composed by a lot of hardware (3/4 Harp devices, around half a dozen Harp peripherals and a camera), so it only takes a small issue in hardware for the whole operation to go south.

With this in mind, this section aims to guide the user on going through the hardware and testing it to either ensure everything is working correctly or to rule out possible problems in case the setup is not working as expected.

Quick checks

This section serves as checklist of the most basic stuff a setup should have or should pass through before anything else:

- Verify you have all of the necessary hardware (behavior box, Harp tower, Harp SyringePumps, speakers, camera, computer). A more detailed list with the necessary materials can be found [here](#).
- Configure the setup and the camera (read the [Software Installation](#) and the [Camera Configuration](#) sections if you need to do so).
- Calibrate the speakers.

⚠️ WARNING

The calibration protocol is not up to date, but it can be found [here](#).

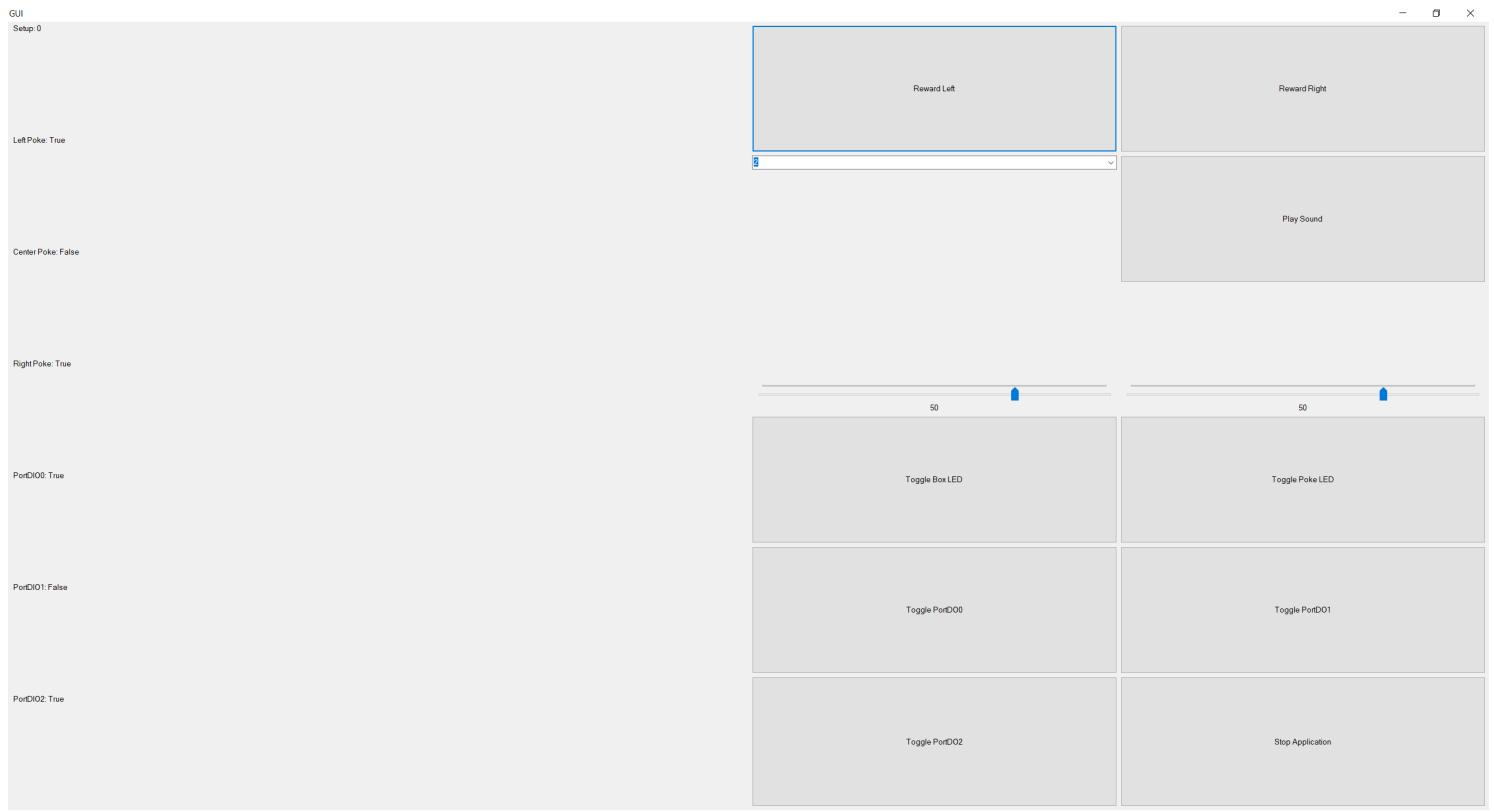
- Configure the task (read the [Task Configuration](#) section).
- Run and test the task without an animal to check everything is working as desired (read the [Running the Task](#) section).

Testing the Harp devices

The Harp devices are the interface between the real world and the computer (and, consequently, the task's logic), specially through the pokes, the speakers and the reward delivery system.

In order to test each of these parts individually, a bonsai workflow named `SetupTesting.bonsai` was created. To open it, follow the instructions:

1. Open `./bonsai/Bonsai.exe`.
2. Click on `Open File` and select the workflow `./src/SetupTesting.bonsai`.
3. Start the execution of the workflow. A GUI similar to the one shown in the figure below should pop-up (if it doesn't, double-click on the `GUI` node).



The left side of the GUI shows the state of the digital inputs (DI) used in the task. The first three are the DI corresponding to the pokes, which are very straight forward. The remaining ones are a second set of DI that are used in the task to time certain events that require timing precision (i.e. fixation time). To test these DI in particular, click on the **Toggle PortDOx** buttons and check whether the state of the corresponding **PortDIOx** changes with it.

The right side of the GUI is more straight forward:

- The **Reward Left/Right** buttons should deliver water, by using the same Harp SyringePump protocol used in the task.
- The **PlaySound** button should play the sound stored in the index indicated in the combobox on the left (Note: index 31 is a silence and is used to stop a sound that is playing) and with the left and right intensities (dB SPL) that are shown in the sliders (this workflow uses the same calibration parameters that would be used in the task).
- The **Toggle [something]** buttons toggle a specific digital output (i.e. the box LED).
- Finally, the **Stop Application** button stops the execution of the workflow.

Testing the Camera

There aren't a lot of requirements for the camera recording. Besides having a clear image and the camera being capturing the region of interest (which in this case is the region containing the pokes, the speakers and the animal interacting with these), the other major requirement is that the frame rate is 100 Hz.

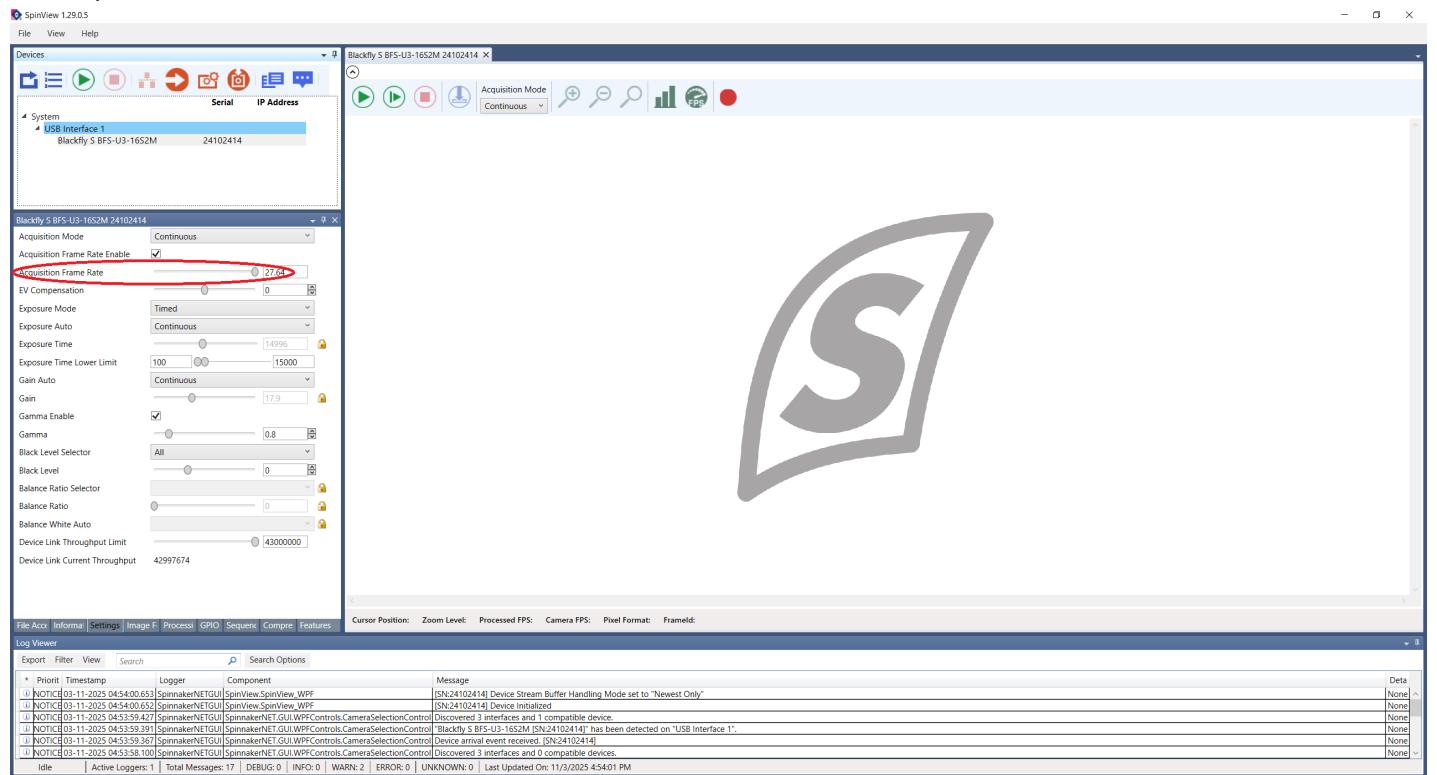
⚠️ WARNING

If the camera hasn't been configured yet, read the [Camera Configuration](#) section before proceeding.

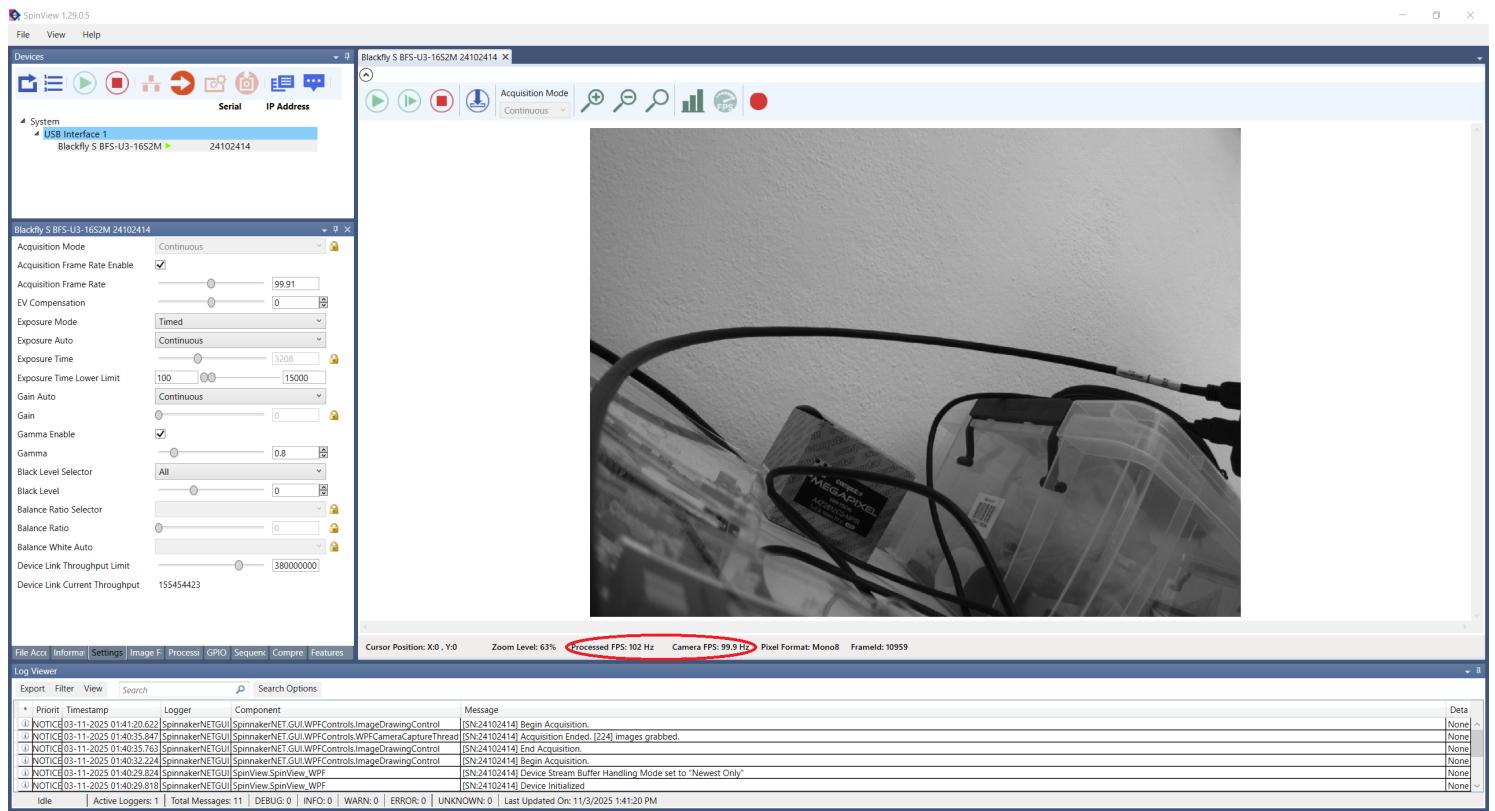
To check if the camera is recording with the desired frame rate, open the respective camera software ([SpinView](#) for the FLIR cameras and [Point Grey FlyCap2](#) for the Point Grey cameras) and take a look at the real frame rate values given by the camera (see the figures below).

If the real frame rate doesn't match the desired value (in this case 100 Hz), it might be due to one of two factors:

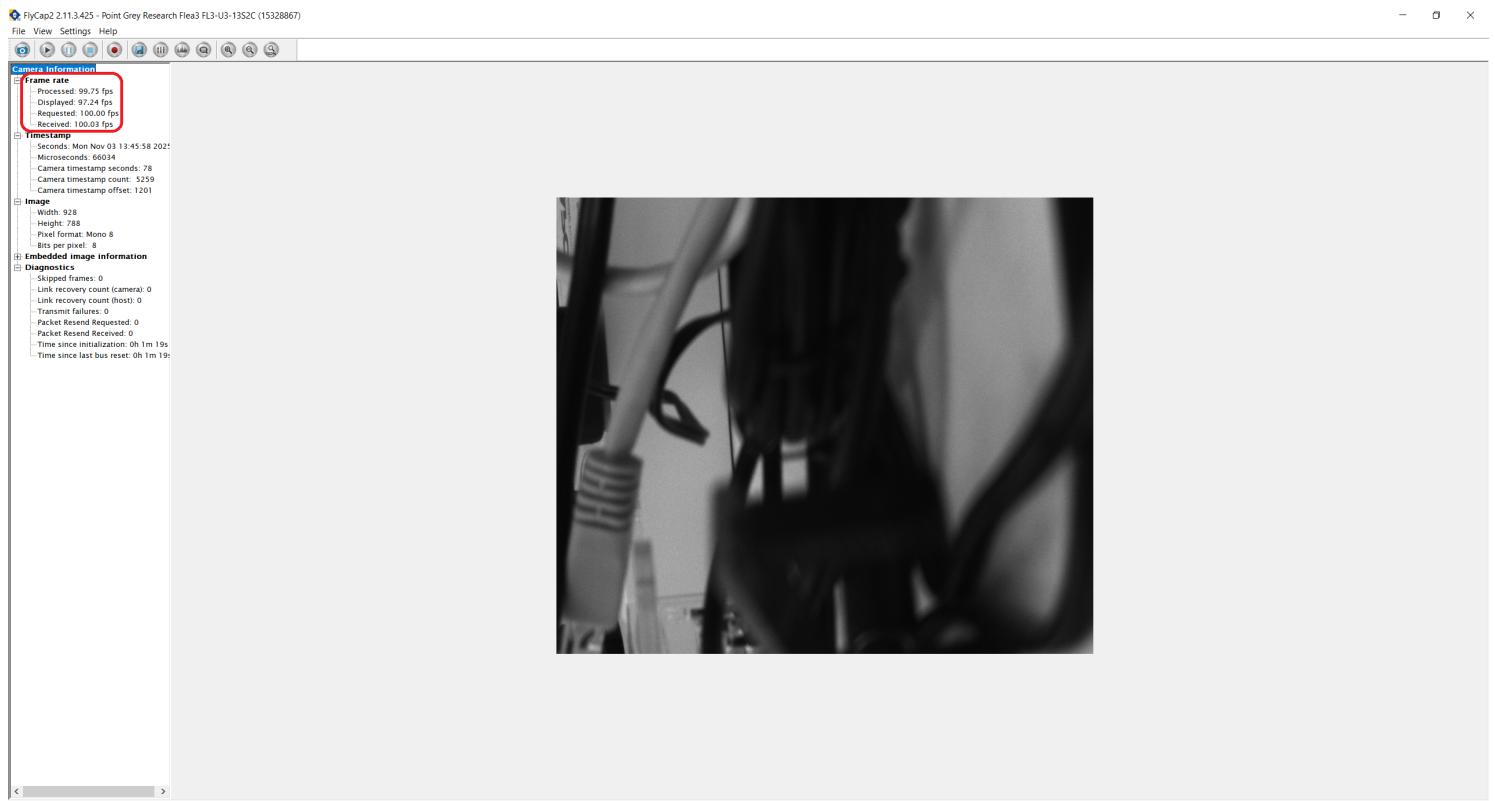
- There isn't enough light reaching the camera. Try to decrease the exposure time (read the [Camera Configuration](#) section again if you need) or try to increase the luminosity of the region of interest.
- The camera is using USB-2.0 instead of USB-3.0. This happens if the camera is connected to a USB-2.0 port of the computer. This might also happen if there's some sort of contact issue between the camera and the computer (when the camera is connected to a USB-3.0 port) - make sure the connections are solid and, as a last resource, switch the cable altogether. If this is the case, the camera's software will cap the maximum frame rate configuration value to 27.64 Hz (see figure below).



SpinView



FlyCap2



Known Issues

This project is still being actively developed and maintained. However, since this project is composed by a lot of small parts, it's normal that some issues are discovered and, by one reason or another, can't be

addressed right away. Here's a list of known issues and how to deal with them for now.

Bonsai Freeze

There's an ongoing issue where the Bonsai might freeze during a session. It's still unclear what's causing the issue, but it has the following side effects:

- Video corruption (the video from the session is basically lost)
- Possible data loss in the JSON out structure