

Blue Laser (473 nm)

Violet Laser (405 nm)

Green Laser (561 nm)

Clean up filters

Mirrors

M1

M2

M3

M4

D1

D2

Camera

Dichroic Filter / Beamsplitter

Bandpass Filter

Fiber Bundle

Objective Lens

Fiber bundle stage adjust- focal distance from camera

Fiber bundle stage adjust – left/right in image

Fiber bundle stage adjust height

D3

Multisite Photometry System Schematic

# Table of Contents

For faster navigation, go to View > Check Navigation Pane under Show > Headings Tab and you can jump to any section.

Single Color – Blue and Violet Laser, GCaMP Excitation

1. Background
2. Hardware connections
3. HCImage Settings
4. Laser and Fiber Bundle Alignment
5. Synchronizing with 5CSRTT Protocol
6. Arduino Code and Setup
7. Data Collection Protocol
8. RAID Tower
9. Raw Data Processing
10. A

Dual Color – Blue, Green and Violet Laser, GCaMP and rGeco Excitation

What do the controls do?

# 1. Background

The Multisite Photometry System is designed for collecting photometry data from multiple brain regions simultaneous from the same animal during behavior tasks.

The system utilizes two lasers, one at a blue wavelength (473 nm), for exciting GCaMP, and a violet wavelength (405 nm) to control for background fluorescence and motion artifacts. 405 nm is near the isosbestic wavelength of GCaMP, where the fluorescence signal from GCaMP is the same in the presence of or without Ca2+, so the signal obtained from the violet laser can be used as a control.

The two lasers are directed using a series of mirrors and lenses to a photometry fiber patch cord bundle, which is connected to an animal performing a behavior task. The lasers will excite GCaMP, and the resulting fluorescence will travel back through the fiber bundle, and the fluorescence is captured using a high-speed CMOS camera that captures images of the fiber bundle. A 4X objective is placed in front of the fiber bundle to help focus the camera FOV on the fiber bundle. The GCaMP fluorescence is quantified as the averaged pixel values of the fiber bundle in the images.

The timing of the system is controlled by an Arduino microcontroller. The Arduino receives triggers from MedPC, which indicates when the individual trials of the behavior task begin, which is when the system needs to start collecting photometry data. The Arduino then triggers both the lasers and camera to start pulsing and capturing images, respectively.

The two lasers are multiplexed at 10 Hz (pulsed alternately) to form a 20 Hz pulse train. Once triggered, camera captures an image every time the lasers are pulsed at 20 Hz, for a total duration of 10 seconds per trial. The lasers are continuously pulsed, even while the camera is not collecting data, to prevent the possibility for the animal to use the laser pulses as a timing mechanism to perform better at the 5CSRTT task.

The image datasets are then converted into quantified photometry data using MATLAB.

Camera

Pulse Sequence

473 nm

405 nm

473 nm

405 nm

CMOS Camera

4x

AF- 488

# 2. Hardware Connections

CMOS Camera

473 nm

405 nm

Arduino Microcontroller

MedPC Trigger Output

12

7

4

2

GND

BNC

BNC

BNC

Laptop

USB

1. Arduino

* Pin 2 – MedPC Trigger Input
* Pin 4 – Camera Trigger Output -> CMOS Camera, requires 2 Pin to BNC Cable and a BNC to BNC adapter
* Pin 7 – 405 nm Laser Trigger Output -> 405 nm Laser Control Box, requires 2 pin to BNC Cable
* Pin 12 – 473 nm Laser Trigger Output -> 473 nm Laser Control Box, requires 2 pin to BNC Cable
* Ground – 2 lasers and MedPC Trigger Box need to be grounded, only 2 grounds on Arduino. Use 2 pin input to 1 pin output adapter to link two of the ground wires together

1. MedPC Trigger Output Box

* Positive wire goes to Arduino Pin 2
* Negative wire goes to Ground

1. 473 nm Blue Laser

* Requires 2 pin to BNC Cable for Trigger from Arduino Pin 12
* Power Supply

1. 405 nm Violet Laser

* Requires 2 pin to BNC Cable for Trigger from Arduino Pin 7
* Power Supply

1. CMOS Camera

* Requires 2 pin to BNC Cable for Trigger from Arduino Pin 4
* USB Cable connection to Laptop
* Huge Power supply

1. Laptop

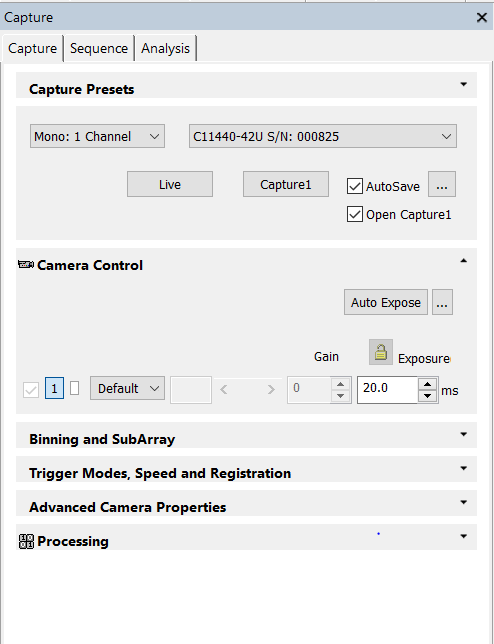
* USB Cable from camera
* Power Supply

# 3. HCImage Settings

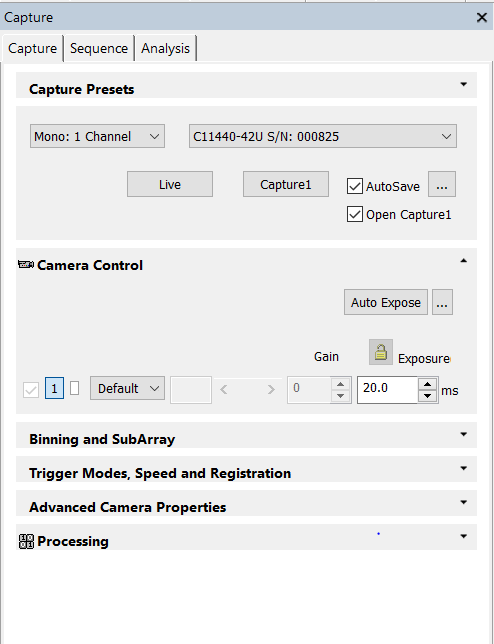
The photometry system camera utilizes the Hamamatsu software, HCImage, for control over image acquisition. Because of the complexity of the triggering system and the high number of images being acquired during a full 30-minute behavior session, there are multiple settings that need to be set prior to image acquisition. Luckily, HCImage can save these settings as presets, so once the settings are optimized, they can be saved for future use. This section will go through step by step the necessary settings required to optimize image acquisition.

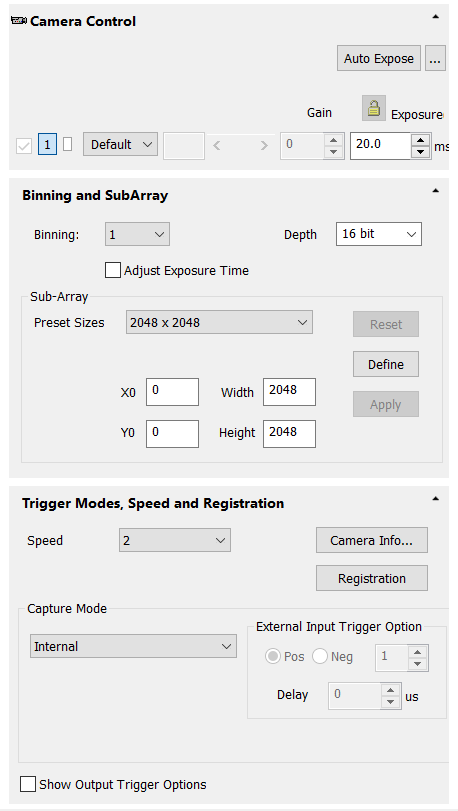
When HCImage is first opened, you will see on the left window all the available settings for the camera, if it is connected. The two tabs that you need are **Capture** and **Sequence**, as shown below. To make it easier, I will just list numerically all the settings that need to be adjusted under both Tabs.

**Capture Settings**

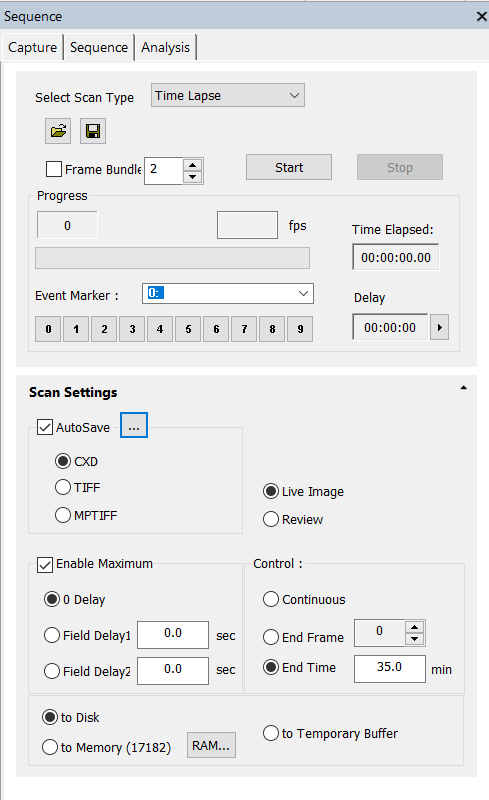
Capture Settings control the basic settings for the camera when capturing an image, such as exposure time, image size and resolution, and trigger mode.

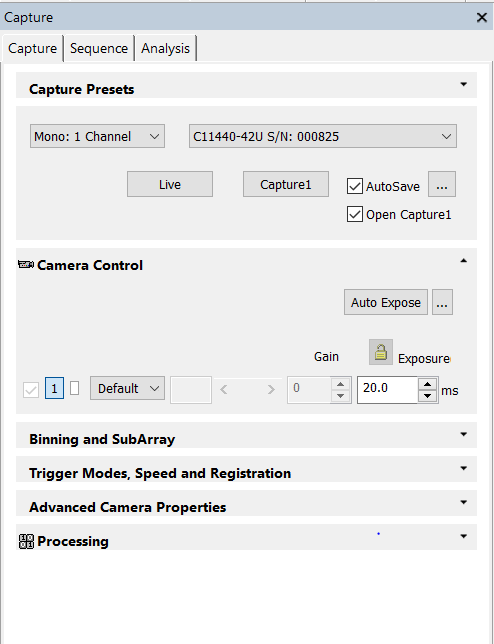
1. **Camera Control** – This controls exposure time for the camera. DO NOT USE THE AUTO EXPOSURE. Set the exposure time manually. Usually, if the camera is acquiring at 20 Hz, a 20 ms exposure time is enough and will not exceed the frequency of the triggers. You should test this with a test trial to make sure that you don’t miss any images.

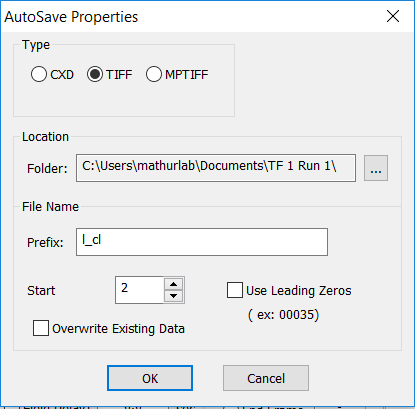
Also, click the  button next to Auto Expose and make sure that Darkfield is selected, so the camera’s exposure setting is optimized for capturing images in the dark.

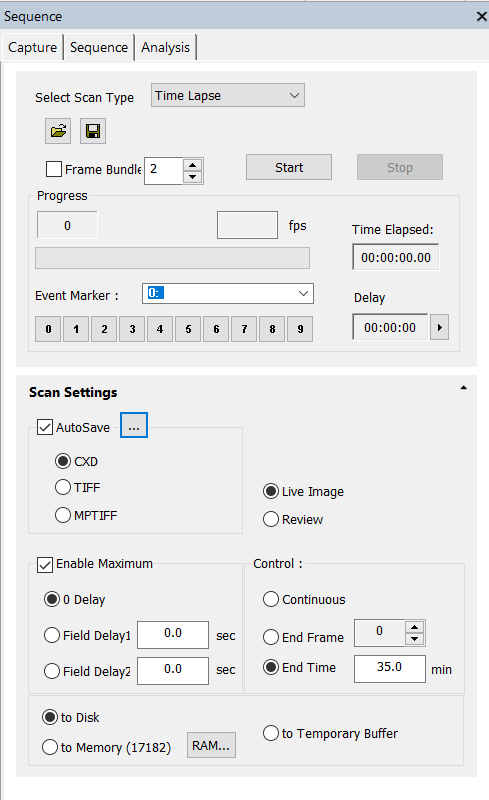
1. **Binning and SubArray** – This determines both the resolution depth of the image and the size of the images being captured. The depth of the image should always be 16 bit. The full image size that the camera is capable of capturing is 2048 x 2048 pixels. However, photometry datasets for a single behavior session exceed 10,000 images, and with full sized images, the size of the datasets would be greater than 20 GB. The photometry bundle is capable of fitting in a 424 x 424 sized image, which will shrink the datasets to a more manageable 6-7 GB. To adjust the image size, change X0, Y0, Width and Height under SubArray settings. Width and Height will crop the image down to the size chosen, so set Width and Height to 424 x 424. X0 and Y0 refers to the X and Y offset if the image is cropped. This needs to be adjusted through trial and error, based on the location of the fiber bundle in the full-sized image. When the settings are optimized, click on Apply to save the Binning and Subarrray settings.
2. **Trigger Modes, Speed and Registration** – This determines the trigger settings for the camera. For the photometry system, we need the camera to recognize an External Edge Trigger for a Positive Edge. This means the camera will trigger and capture an image every time it receives a rising edge voltage step. The camera speed should be set to 2.
3. **Save as a preset** – The last two settings, Advanced Camera Properties and Processing, are fine set to the default. To save all the changes previous mentioned, click on the Capture Presets dropdown and save the preset as a new setting. **THIS WILL ONLY SAVE THE SETTINGS FROM THE CAPTURE TAB, SO YOU STILL NEED TO INPUT SEQUENCE SETTINGS PRIOR TO EVERY BEHAVIOR SESSION.**

**Sequence Settings**

Sequence settings allows us to automate the image acquisition process, so that the camera will continuously capture images every time it receives a trigger. **These settings are not saved in the Capture Settings preset so they need to be selected every time HCImage is opened!**

1. **Scan Type** – Time Lapse. This setting allows the camera to continue capturing images until a specified time duration has elapsed. When used with a trigger, the time elapsed timer will continue counting even when the camera is not being triggered.
2. **Scan Settings** – Make sure that you have selected Time Lapse as the Scan Type before changing these settings. Make sure to change the image type to TIFF and select Autosave. Choose Live Image, check Enable Maximum, and select to Disk. For Scan setting Control, choose End Time and enter 35 mins. If the setting is set to a different unit, like seconds, first enter 60 seconds, which will change the units to minutes. 35 minutes exceeds the length of a behavior session, so the camera will not stop collecting data before the session ends.
3. **Autosave Settings** – Click on the  next to the AutoSave checkbox. Most of these settings are not presets and need to be changed prior to every behavior session. However, the file prefix can be set prior to the start of a new behavior cohort. This prefix determines the name of the individual images being saved. The usual convention is the name the images with the location of the fiber(s) in the fiber bundle that are connected to the animal, and the brain region being imaged. For example, in the example shown on the right, the prefix is l\_cl, where l stands for the leftmost fiber in the bundle, and cl stands for claustrum, the region being imaged. For location, usually left and right is specific enough to distinguish the fibers because at most, we’ve only connected 2 photometry cords onto an animal. However, any naming convention can be used if it is consistently used throughout an entire cohort.





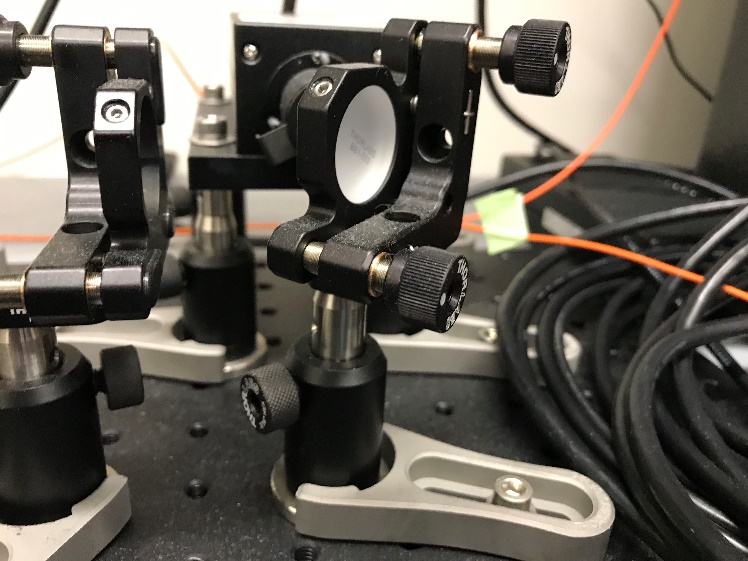
# 4. Laser and Fiber Bundle Alignment

Before using the Multisite system, all the lasers need to be aligned so that the beams are directed to the fiber photometry bundle. Also, the fiber bundle needs to be adjusted to the optimal focal distance from the camera. The laser beam diameter is smaller than the diameter of the fiber bundle, so the beam will at most optimally illuminate 2-3 of the fibers.

The lasers are directed using a series of mirrors and filters. Just a couple of distinctions to note:

* Mirrors will reflect all light
* Dichroic filters, also sometimes called beamsplitters, will pass light within certain wavelength band(s), and reflect all other light
* Bandpass filters will pass light within certain bands, and attenuate all other wavelengths
* Cleanup filters will filter out any wavelengths not within the specific band. These are place in front of the laser beam path to clean up the laser beam wavelength. These do not need to be adjusted for angles, the laser just pass through the filter. We have one for every laser.

Our system contains 4 mirrors (labeled M1 – M4), 3 dichroic filters (labeled D1 – D3) and 1 bandpass filter.

* D1 reflects blue light and passes all other wavelengths
* D2 reflects violet light and passes all other wavelengths
* D3 reflects blue and violet light and passes all other wavelengths
* The bandpass filter passes green light within the emission spectrum of GCaMP6f and attenuates all other wavelengths

Fine-tune adjustment knobs. One will move the beam up/down, other one will move it left/right. Not going to lie, I can never remember which one does which. You can figure it out with the live camera stream 😊

Post

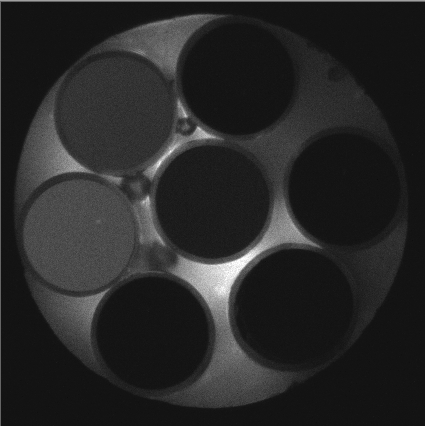
Post holder adjustment knob. Unscrew to move the post up/down or to rotate mirror/lens

To align the laser beams, the two-bounce method is used for each laser to allow for fine-tuned control over the laser beam in all 3 dimensions. In theory, one mirror/filter is used to align the beam in the XY axis and adjust it up/down and left/right, and the other mirror/filter is used to make sure the beam is traveling in a straight path. Each mirror/filter has two knobs that allow for movement in one axis if you twist it. For a more detailed description of the two-bounce method, here’s a link to a guide:

[https://www.edmundoptics.com/resources/applicat3ion-notes/lasers/simplifying-laser-alignment/](https://www.edmundoptics.com/resources/application-notes/lasers/simplifying-laser-alignment/)

Alignment steps

1. For both the focal distance adjustment of the fiber bundle and the laser alignment, the CMOS camera can be used with continuous live streaming to see what the images of the bundle look like.
2. Make sure that all lights are off. Turn on the Hamamatsu camera and open up HCImage. Go to the Capture Settings tab on the left side window and click on Live to start the livestream
3. Turn on the lasers. Unplug the trigger BNC cable, so that the laser fires a continuous beam instead of pulses.
4. Check that the camera trigger setting is set to Internal. Adjust the contrast settings until you can see the fiber bundle in your image.
5. Check the clarity of the fiber bundle image. Adjust the knob that moves the fiber bundle closer/farther away from the objective. Adjust until the fiber bundle image is perfectly in focus. An example is shown in the image below.
6. Turn on only the blue laser. Check if the beam path is aiming relatively close to the fiber bundle. If you can’t see the laser dot on the fiber bundle at all, or if you can tell the beam is extremely out of alignment, go to step 3A for larger adjustments. Otherwise, go to step 4.
7. Large adjustments to the laser path requires drastic changes to the mirror/filter angles. For the blue laser, adjust M1 and D1. For the violet laser, adjust M2 and D2. I would suggest adjusting only M1 and M2 because adjusting the dichroics may affect the path of both laser beams, which will make alignment more difficult. Because the mirrors are the first bounce for the laser beams, they are isolated from one another, and so any changes will only affect one of the laser beams.
8. Loosen the large knob on the post holder of the mirror/filter. You can now adjust the silver post holding the mirror/filter. You can move the height of the post or change the mirror angle. Set it so that the laser beam is directed close to the fiber bundle. PLEASE BE CAREFUL DOING THIS, THE LASER BEAM CAN BE REFLECTED IN ANY DIRECTION.
9. Tighten the knob and make sure the post is secure.
10. Now look at the Hamamatsu camera stream and see where the laser beam is focused. The individual fibers are taped with labels that correspond to its location in the camera image. As seen in the image on the right, the two fibers being used for this image are outlined, blue for the GCaMP signal fiber, red for the control fluorophore fiber. The laser beam here is focused directly in between the two fibers at the top left portion of the fiber bundle.



1. Adjust the laser beam position by looking at the camera stream to see where the signal is strongest. For the blue laser, adjust M1 and D1. For the violet laser, adjust M2 and D2. I would suggest adjusting only M1 and M2 because adjusting the dichroics may affect the path of both laser beams, which will make alignment more difficult. Because the mirrors are the first bounce for both laser beams, they are isolated from one another, and so any changes will only affect the path of one of the laser beams. Adjust by using the two knobs on the mirror/lens of choice and seeing which direction the beam moves in the image. Any of the 6 fibers can be used, although preferably choose fibers that are close to each other because of the laser beam radius constraint.
2. Check the output from the photometry cords using the light meter and compare the output between the two fibers. Ideally, the control fluorophore fiber should have lower output to compensate for the higher signal strength from fluorophore compared to the GCaMP signal. I usually set the actual fiber for recording to around 0.2-0.3 mW and the stationary fiber to around 0.1 mW. If the laser beam is focused at the two fibers, but the power output is too low or high, adjust the power knob on the laser module.
3. Repeat steps 3-7 with the violet laser. Make sure that the output is similar for both the blue and violet laser. The violet can be lower in power if needed because of the higher background noise signal for violet.

# 5. Synchronizing with 5CSRTT Protocol

5CSRTT Stage 3 Trial Timeline

Time (s)

ITI

Cue

0

5

6

11

Response Period

Lasers/Camera begin collecting data

10

Camera stops, Lasers continue pulsing

Laser stops pulsing for 110 ms to reset trigger, proceed to next trial

The Multisite photometry system is optimized for collecting photometry data during the final stage of the 5-choice serial reaction time task (5CSRTT), Stage 3.

* Each 30-minute session consists of a maximum of 100 trials, whichever occurs first.
* The basic timeline of a Stage 3 is shown above.
* Each trial begins with a 5 second ITI period
* The response period is 6 seconds, where the 5-choice cue light is on for the 1st second
* The photometry system begins collecting images at the start of the ITI and collects 10s of images
* If the mouse responds correctly, the response period ends and a reward is dispensed. There is an unlimited time period for the mouse to retrieve the reward, and a 5s period for eating before moving on to the next trial.
* If the mouse responds incorrectly, the house lights go off and the mouse must go through a 5s timeout period. If the mouse responds again during this time, another 5s is added to the timeout duration.
* If the mouse omits, the trial goes through the 11 sec time interval without a response and proceeds to the next trial
* If the mouse responds prematurely during the ITI interval, the trial will restart, including the MedPC trial timer. Photometry data cannot be used for premature trials, because there is no way to align the data to the trial.
* The mouse cannot physically complete a trial in less than or equal to 10 seconds, because of time and movement constraints. There are 2 scenarios where the trial will take 10s. The 1st is if the mouse immediately responds incorrectly when the cue comes on. The time would be 5s ITI + 5s timeout for an incorrect response = 10s. The other scenario would be if the mouse responded correctly when the cue comes on, and then retrieved the reward immediately. In that scenario, the time it takes to complete the trial would be the 5s ITI + 5s delay after reward retrieval = 10s. This is why the camera only collects 10s of photometry data for each trial.
* The lasers are continuously pulsed, even when the camera is not collecting images. This is to account for the possibility the mouse may use the lasers as a timing mechanism. To make sure the laser stays in alignment with the start of the trial, it is briefly off for 110 ms (reason for this duration is explained in Section 6, related to Arduino code) at the end of a trial, and then retriggered at the start of the next trial.

5CSRTT Stage 3 Trial – Example Correct Trial Timeline

ITI

Cue

0

5

6

8

Response Period

Lasers/Camera begin collecting data

7

Correct Response

5s delay ends, lasers stop pulsing for 110 ms, proceed to next trial

Reward Period

Reward Retrieved, 5s delay for eating pellet begins

10

13

5CSRTT Stage 3 Trial – Example Incorrect Trial Timeline

ITI

Cue

0

5

6

Response Period

Lasers/Camera begin collecting data

7

Incorrect Response,

5s timeout period begins

5s timeout ends, lasers stop pulsing for 110 ms,

Proceed to next trial

Timeout Period

10

12

Camera stops, Lasers continue pulsing

ITI

ITI

0

6

11

Response Period

Lasers/Camera begin collecting data

10

Camera stops, Lasers continue pulsing

Laser stops pulsing for 110 ms to reset trigger, proceed to next trial

5s Timeout Period

5s timeout period ends, trial restarts

Premature Response, 5s timeout period begins

17

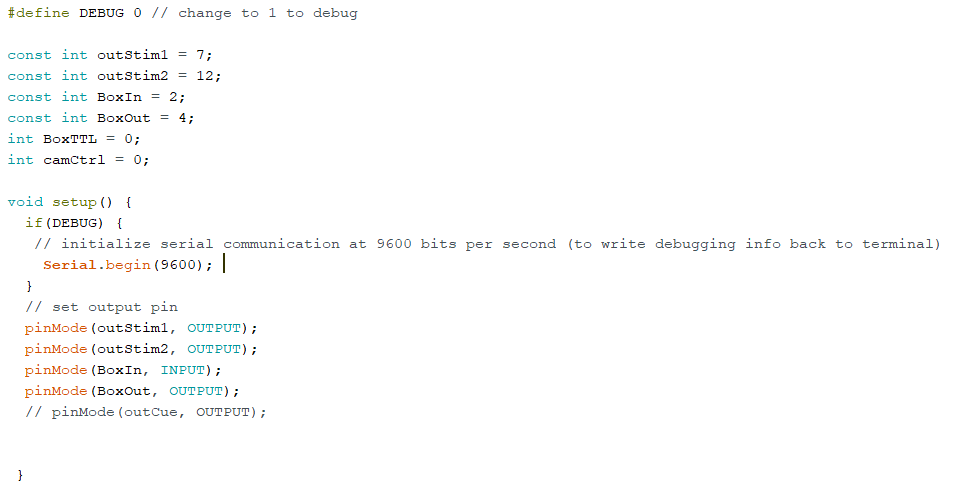
12

1

Cue

# 6. Arduino Code and Setup

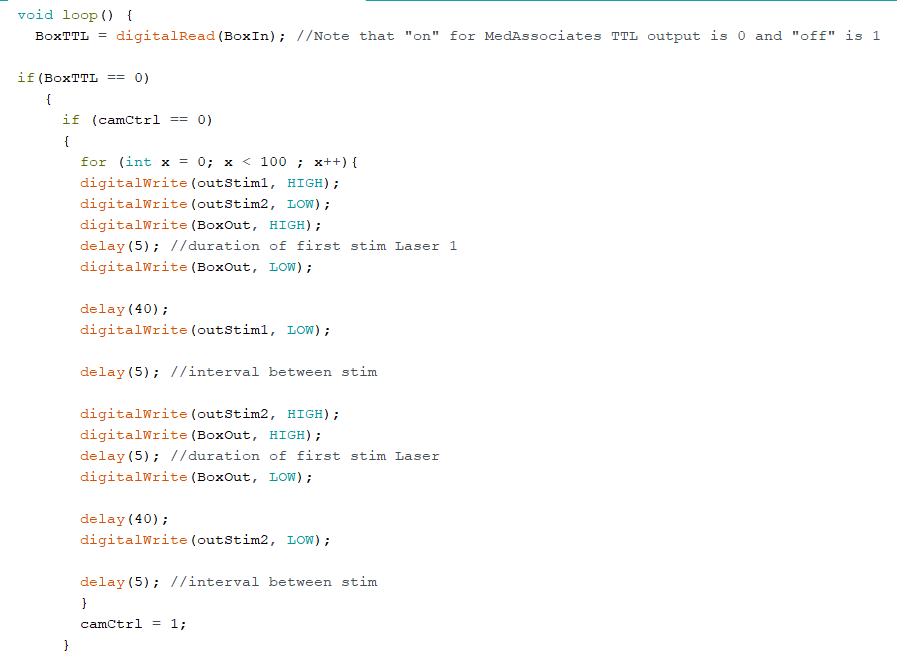
The synchronization between MedPC and the Photometry System is controlled by an Arduino microcontroller. The Arduino takes the input from MedPC, which triggers at the beginning of every trial at the start of the ITI. The Arduino every other aspect of the acquisition, including frequency of the laser pulses and camera acquisition, the duration of each individual pulse, and the time duration the camera captures per trial.

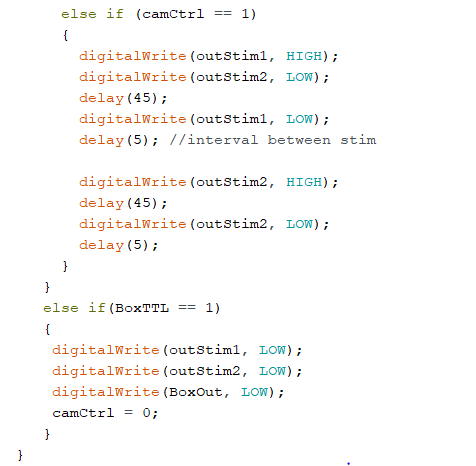
The first segment of the code declares all the constants and variables required for the code, which sets the input and output pins on the Arduino. The pins are:

* Pin 2 – MedPC Trigger Input, const int BoxIn
* Pin 4 – Camera Trigger Output, const int BoxOut -> CMOS Camera, requires 2 Pin to BNC Cable and a BNC to BNC adapter
* Pin 7 – 405 nm Laser Trigger Output, const int outStim1 -> 405 nm Laser Control Box, requires 2 pin to BNC Cable
* Pin 12 – 473 nm Laser Trigger Output, const int outStim2 -> 473 nm Laser Control Box, requires 2 pin to BNC Cable
* Ground – 2 lasers and MedPC Trigger Box need to be grounded, only 2 grounds on Arduino. Use 2 pin input to 1 pin output adapter to link two of the ground wires together

The last two variables, int BoxTTL and int camCtrl, are logic variables that are used later in the code for controlling the timing of the laser pulses and the camera. In the beginning, they are both initialized as 0.

Every Arduino code has both a void setup and a void loop. Basically, an Arduino code loops through a set of commands indefinitely. That is the void loop, which will contain bulk of the commands. The void setup initializes variables and starts the serial, which determines how fast the Arduino writes and displays information. Because our code is not writing any information at a high-speed rate, a serial rate of 9600 is sufficient. The void setup just determines the initial setup right when the code is uploaded because once the code is uploaded, it will immediately start the void loop. The crucial part of the void setup is declaring the Arduino pins as inputs and outputs. BoxIn is an INPUT, while BoxOut, outStim1 and outStim2 are all OUTPUTS.





The void loop will control the actual Arduino function, by looping through every line in the void loop over and over again sequentially. At the start of every loop iteration, BoxTTL reads the input from Pin 2, BoxIn, which is the TTL input from MedPC. There are two different if statements in the loop, that basically split the code into 3 different states. This is because the lasers have to be pulsed continuously until 10ms before the end of the trial, while the camera only collects 10s of data. This is controlled using the variable camCtrl to control the state of the camera.

In the outer if statement, the code checks if BoxTTL is 0 or 1. The MedPC TTL output is inverted, so the trigger is on at 0, and is off at 1. If BoxTTL = 0, the code then proceeds onto the second if statement. The second statement checks the value of camCtrl, which determines the state of the camera. If camCtrl = 0, the camera has not obtained images from the current trial, so the Arduino will trigger the camera to start the camera. If camCtrl = 1, the camera has obtained the full set of images from the current trial and will not trigger. Below, the void loop is outlined step by step, going through the 3 different state scenarios.

State 1 – BoxTTL = 0 , Camera = 0. Duration : 10 s

Arduino will trigger the lasers and the camera. The lasers are being pulsed alternately at 20 Hz (50 ms pulses), so each pulse lasts for a duration of 45 ms with a 5s pause. The camera trigger is much shorter, because the camera only needs a rising edge to trigger, with the Hamamatsu software controlling all other acquisition settings. The camera has a 5ms pulse trigger, so the sequence goes:

Violet laser on, Camera trig on -> 5 ms -> Camera trigger off -> 40 ms -> Violet Laser off -> 5 ms ->

Blue laser on, Camera trig on -> 5ms -> Camera trig off -> 40 ms -> Blue laser off -> 5ms

This sequence loops for 100 times, so 100 ms \* 100 = 10 s of camera images.

Once this sequence is complete, the state of camCtrl changes to 1. This makes it impossible to trigger the camera again until camCtrl is reset back to 0. The void loop will then return to the beginning and check the state of BoxTTL and camCtrl again.

State 2 – BoxTTL = 0, Camera = 1. Duration: 100 ms

The void loop then starts back at the beginning and reads the TTL output from MedPC, then enters the state where the MedPC Trigger is still on, but the camCtrl is set to 1. Here, the lasers will continue to pulse 45 ms on, 5 ms off, but the camera is not triggered. The loop will continually repeat this state until the end of a trial.

State 3 – BoxTTL = 1, Camera = 1. Duration: 110 ms

At the end of a trial, the MedPC TTL will turn off for 110 ms. This is the set duration because if the TTL is off when the Arduino is still in the middle of State 2, it will miss the trigger to turn off the lasers and reset camCtrl. 110 ms is longer than the duration of State 2, so the Arduino will have time to return to the beginning of the void loop and read the TTL output and recognize that it is off. When it does, the loop will go to the 3rd state, where both BoxTTL = 0 and camCtrl = 1. This will reset the lasers by turning them off, and then resets camCtrl to 0, allowing the camera to capture images again for the next trial.

Check TTL state: 0 or 1?

Check Camera state: 0 or 1?

Pulse violet laser, capture image

Pulse blue laser, capture image

Repeat 100 x = 10 s of images

Set Camera state to 1

Pulse violet laser

Pulse blue laser

Reset Camera state to 0

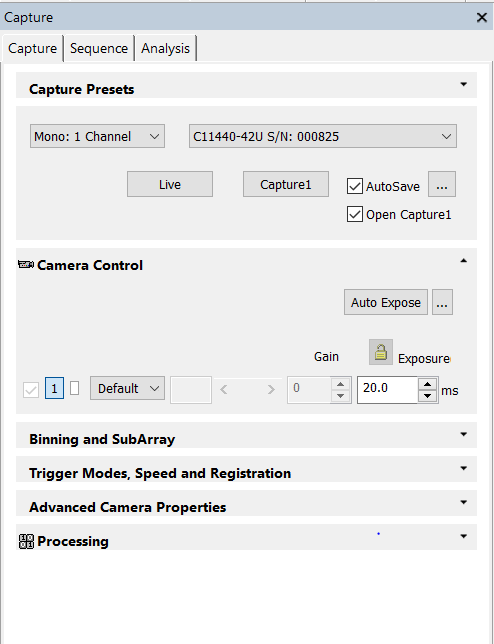
1

0

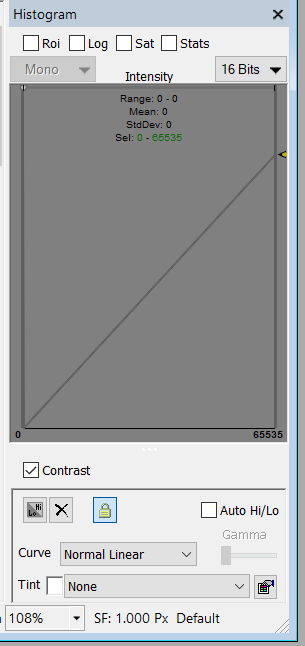
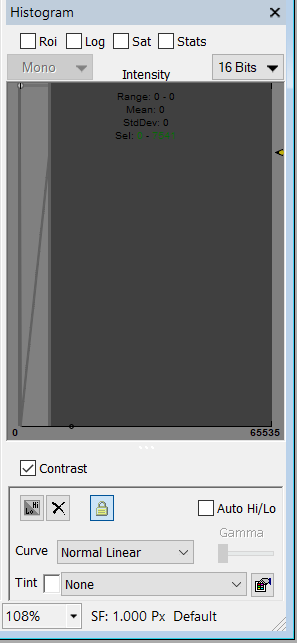
0

1

# 7. Data Collection Protocol

1. Turn on the lasers and camera. Wait at least 5 minutes before beginning a behavior protocol to let the lasers warm up and stabilize.
2. Open up HCImage. Open the preset with the correct Capture settings. For more details, go back to [Section 3 on HCImage Settings](#_3.__HCImage) for the full list.
3. Open Window Task Manager with Ctrl+Alt+Delete, and go to the Details Tab. Find HCImage on the list, right click on it and go to Set Priority -> High. This will prioritize all computational processing towards HCImage and lower the probability that the computer skips an image.
4. Go to the Sequence Tab.
5. Make sure that the Scan Type is set to Time Lapse. Change the image type from CXD to TIFF.
6. Click on the  next to the AutoSave checkbox.
7. Create a new folder for each behavior session. The naming protocol should be

AnimalName Run #. For example, for animal TF 1, the first run of the day, the of the folder should be TF 1 Run 1.

1. Make sure that the folder is on the SSD Hard Drive (C:\) drive. This drive writes at a faster speed and can handle the high speed required to save the images being collected. Usually I just create each behavior session folder in Documents.
2. Click okay, then click on Start to begin the collection sequence.
3. Change the contrast in the image display by clicking on the checkbox marked Contrast under the Histogram, as shown in the images below. Compress the intensity range by clicking on the gray line and the right-hand side of the Histogram and dragging it to the left. This adjustment does not actually change how the image is captured, it just changes the contrast of the image display.
4. Make sure that the correction photometry fiber is placed into the tube of green fluorophore.
5. Load the MedPC protocol on the computer.
6. Initiate and start the program! Make sure that the photometry system starts collecting when the 1st trial starts.
7. After the behavior session is over, make sure to transfer the folder over to the RAID tower. Each behavior session folder should be sorted into Folders named for the date the session was completed. For more details on folder naming and sorting, go to Section 8, RAID Tower and 9, Data Processing.
8. Make sure that the # of images collected is correct for the # of trials completed. If any images were missed, the computer slowed down and did not save it in time, so close any background programs and make sure that all the computing power is going to HCImage.
9. When you move the datasets over to the RAID tower, make sure to organize the datasets. Place everything into an outer folder with the cohort name. Then, name the internal folders by date in the format DD-MM-YY.

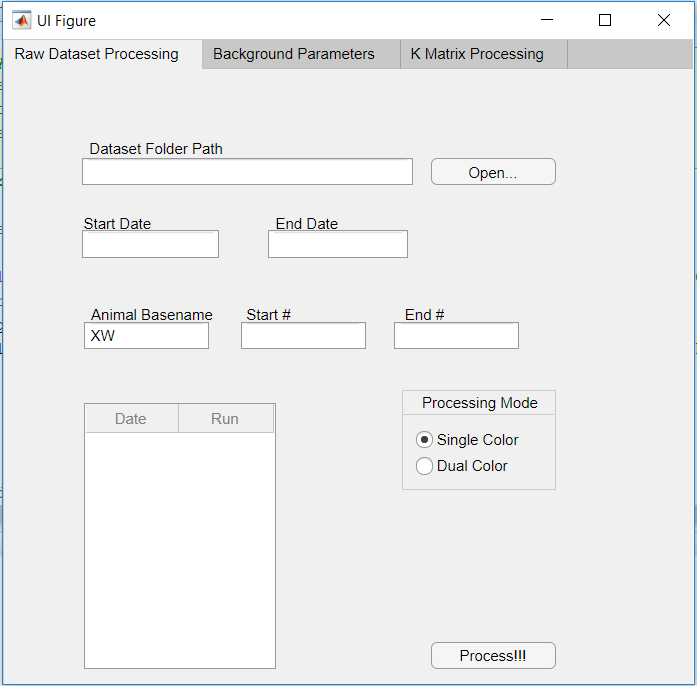
# 8. RAID Tower

# 9. Raw Data Processing

As you can imagine, data processing for the multisite system is computationally intensive, since each dataset for a behavior session is at least 10,000+ images. A Matlab GUI (Graphical User Interface) has been written to expedite the entire process, so all that once a ROI is defined, the program will automatically process and convert the images to raw photometry signal for multiple datasets that you can choose.

The GUI has been converted to a Matlab App for convenience, and can be accessed by going to the **Apps tab in Matlab, then -> Multisite Dataset Process**

Step by Step instructions for using the Program

1. Open the Multisite Dataset Process App. Go to the second tab, Background Parameters.
2. Set the parameters for the datasets that you need to process.
3. The Image prefix is the same image prefix used in the Autosave properties during data collection in HCImage. Don’t forget the \ !
4. Define the names of the specific region you are looking at for ROI 1 and 2. If the system is only being used to look at one region, put NA for ROI 2.
5. Define the # of Images that will be collected per trial. **This is the total number of images collected during a trial divided by 2, because half is the actual GCaMP signal and half is the isosbestic!**
6. Define the image size. This should not change, because a 424 x 424 image is the smallest we can use to capture the entire fiber bundle.
7. Define the max image dataset size. Look at the datasets that you are interested in processing and determine what the largest dataset is. This is important because the processing code will not look at any images beyond this number. It’s okay to set this above the max number of images per dataset you will obtain. Any datasets being processed that has less than this dataset size will be labeled INC, but this does not affect further processing.
8. Go back to the Raw Processing tab and click on . Select the base folder where all the datasets are located (DO NOT GO INTO THE DATE FOLDERS. JUST THE BASE IS FINE).
9. Type in the Starting date and End date you want to process, in the format DD-MM-YY. If you’re only processing one day of data, use the same date for both start and end.
10. Enter the Animal cohort name and the start and end # for the animals being processed. For example, if the datasets being processed are from TF 1, TF 3, and TF 7, then enter TF for Animal basename, 1 for Start # and 7 for End #.
11. Make sure that you are using Single Color!
12. Click on Process to begin!
13. An image will show up, this is the first image from the first dataset that was loaded. Double click to set the ROI for the 1st ROI. Do the same for the other ROIs. If the default position of the ROI is not exactly where the fiber is in the image, drag the circle around using the mouse, and then double click when it is in the right positon. Make sure to check the heading of the image for information which ROI is being defined.
14. Let the program run!

