fly-group-activity-monitor (flyGrAM) Installation

Nicolas Mei, Jamie Catalano, Kavin Nuñez

Overview:

Below is a set of functions and scripts to do real time fly activity quantitation with the added option to control optogenetic stimulation. FlyGrAM allows for quantitation of fly motion (i.e. the number of flies moving in a given frame) during various experimental paradigms. Importantly, this movement quantitation is done in real-time (or near real-time).

Features:

1. Can communicate with an arduino to control the onset time, duration, frequency, and pulse width of LED diodes for optogenetic experiments.
2. Can directly write out 'raw' (corrected for camera lens barrel distortions) collected video during an experiment to '.avi' format for subsequent analysis with other more sophisticated tracking software packages (i.e. C-Trax, Jabba, etc...).
3. Can produce real time activity plots for 4 differents regions of interest (ROI).

Required Software:

1. Python 2.7 64-bit
2. OpenCV (version 3.1) with python bindings RC 3.0+
3. FFMPEG 64-bit Zeranoe build (version: 2015-06-05 git-7be0f48 or 2017-12-15-2d131fc)

Recommended Computer Hardware:

1. 4-core processor (higher clock rate is better) - will ensure fast FFMPEG encoding
2. 16-32 GB RAM
3. High quality USB camera that supports at least 30 fps that is supported by OpenCV
4. Arduino (Uno: http://store.arduino.cc/product/A000066)

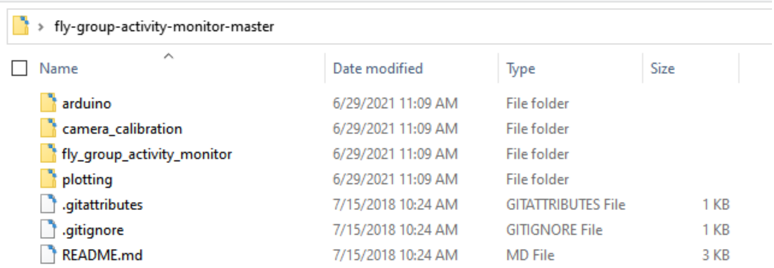
FlyGrAM Installation Instructions:

Download our software:

1. Download the latest flyGrAM software from Github (https://github.com/kaunlab/FlyGrAM-fly_group_activity_monitor)
    a. fly_activity_experiment_manager.py
    b. fly_group_activity_monitor_gui.py
    c. roi.py

2. Unzip flyGrAM zipped folder to a desired location

3. Check that all files are located in the correct places within the “fly-group-activity-monitor-master” folder



fly-group-activity-monitor-master

Name	Date modified	Type	Size
arduino	6/29/2021 11:09 AM	File folder	
camera_calibration	6/29/2021 11:09 AM	File folder	
fly_group_activity_monitor	6/29/2021 11:09 AM	File folder	
plotting	6/29/2021 11:09 AM	File folder	
.gitattributes	7/15/2018 10:24 AM	GITATTRIBUTES File	1 KB
.gitignore	7/15/2018 10:24 AM	GITIGNORE File	1 KB
README.md	7/15/2018 10:24 AM	MD File	3 KB

a. fly-group-activity-monitor-master/arduino/Opto-blink_and_Solenoids/ Opto-blink_and_Solenoids.ino

b. fly-group-activity-monitor-master/camera_calibration/ checkerboard_10mm.png

c. fly-group-activity-monitor-master/fly_group_activity_monitor/
   i. fly_activity_experiment_manager.py
   ii. fly_group_activity_monitor_gui.py
   iii. roi.py

d. fly-group-activity-monitor-master/plotting/
   i. fly_activity_plotting.py
   ii. flygram_analysis.py

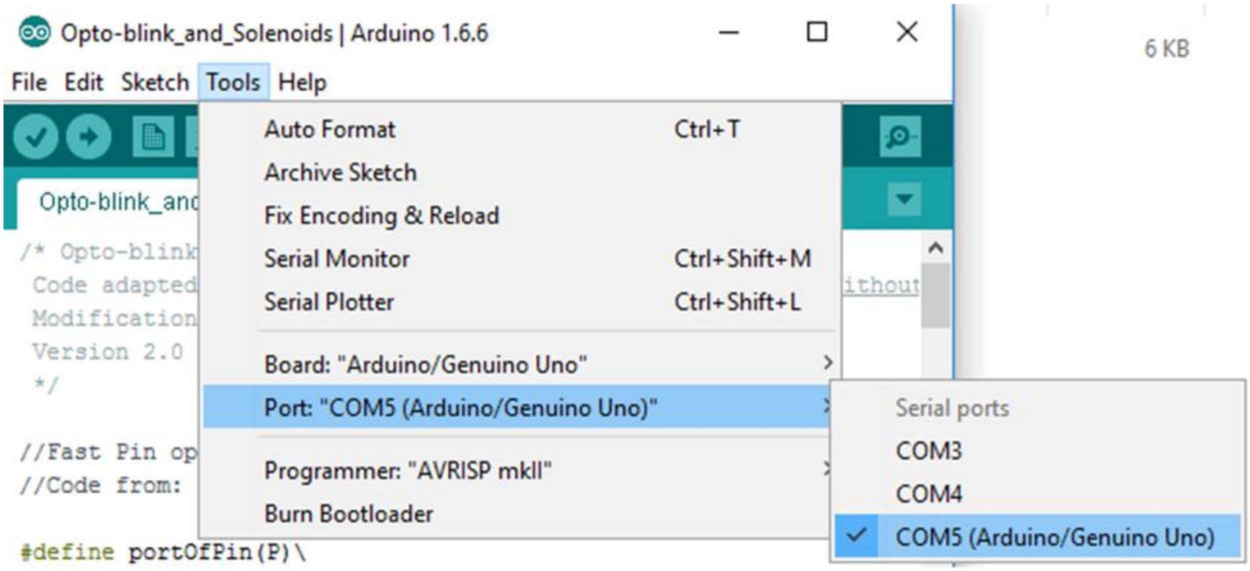
Setting up the Arduino:

4. Download the free and open-source Arduino IDE (integrated development environment) (https://www.arduino.cc/en/Main/Software)

5. Plug in Arduino

6. Using the Arduino IDE open up the .ino file located in the unzipped flyGrAM software folder. The folder path should look something like:
"C:\fly_gram_unzipped_location\fly-group-activity-monitor-master\arduino\Opto-blink_and_Solenoids\"

7. After opening up the .ino file click the "Tools" submenu and make sure that the Arduino is detected by the computer. This can be determined be looking at the "Port" and determining if "(Arduino/Genuino Uno)" immediately follows a COM port designation. Be sure to remember the COM port number! It will be used later!


Opto-blink_and_Solenoids | Arduino 1.6.6
File Edit Sketch Tools Help
Opto-blink_and
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Board: "Arduino/Genuino Uno"
Port: "COM5 (Arduino/Genuino Uno)"
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3
COM4
✓ COM5 (Arduino/Genuino Uno)
6 KB
ithout

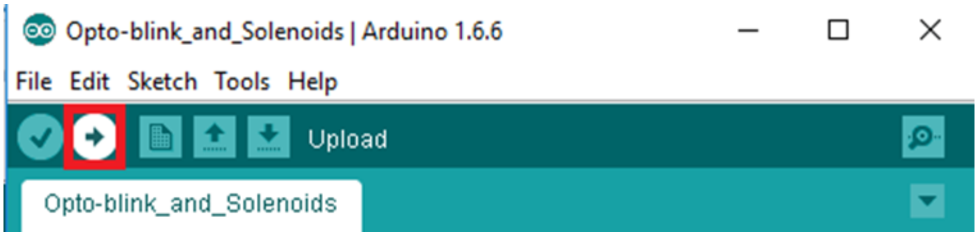
```
/* Opto-blink
 Code adapted
 Modification
 Version 2.0
 */

//Fast Pin op
//Code from:

#define portOfPin(P)\
```

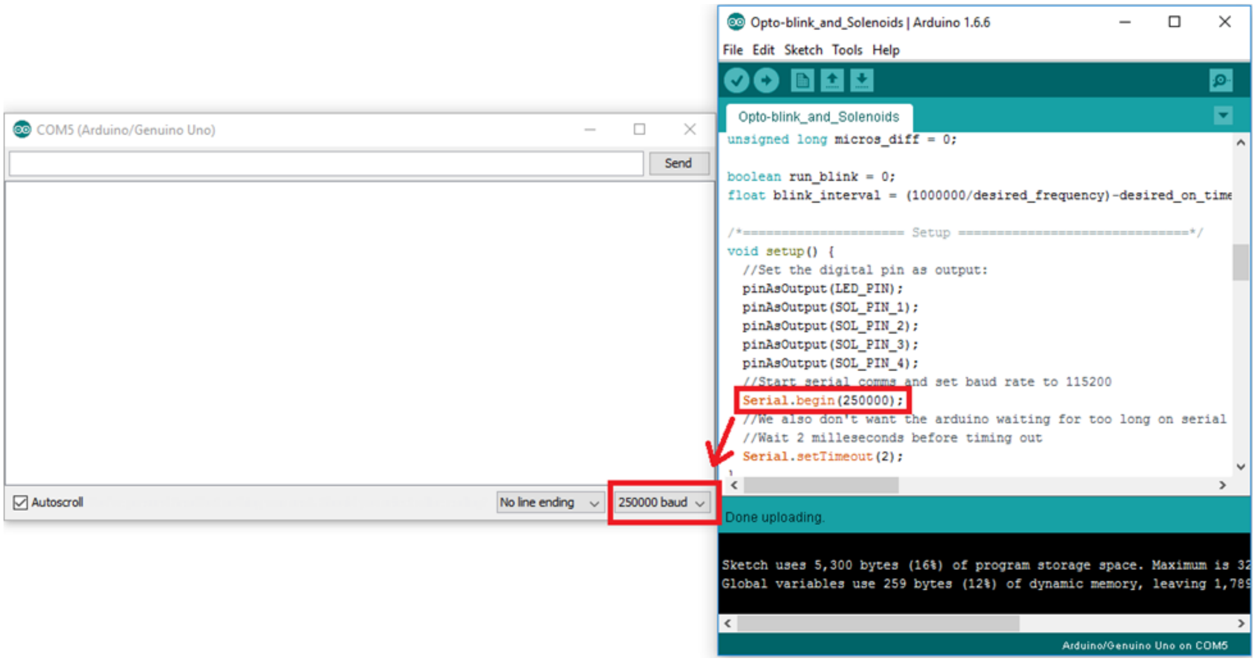
Note: Depending on how many devices are connected to the computer, the COM port which the Arduino will be detected on could vary. Furthermore, the COM port of the arduino may also change after plugging in other devices.

8. Upload the .ino code to the Arduino by clicking the upload button. If the code compiles and upload correctly you should see the message “Done Uploading” near the bottom of the IDE screen.


Opto-blink_and_Solenoids | Arduino 1.6.6
File Edit Sketch Tools Help
Upload
Opto-blink_and_Solenoids

9. To further test that the upload process was successful. Open up a “Serial Monitor” instance which can be found under the “Tools” submenu.

10. In the new “Serial Monitor” window. First ensure that the baud rate is set equal to the “Serial.begin(baud_rate);” found in the Arduino code.


COM5 (Arduino/Genuino Uno)
Send
☑ Autoscroll
No line ending
250000 baud
Opto-blink_and_Solenoids | Arduino 1.6.6
File Edit Sketch Tools Help
Opto-blink_and_Solenoids
Arduino/Genuino Uno on COM5

```
unsigned long micros_diff = 0;

boolean run_blink = 0;
float blink_interval = (1000000/desired_frequency)-desired_on_time

/*===================== Setup ==============================*/
void setup() {
  //Set the digital pin as output:
  pinAsOutput(LED_PIN);
  pinAsOutput(SOL_PIN_1);
  pinAsOutput(SOL_PIN_2);
  pinAsOutput(SOL_PIN_3);
  pinAsOutput(SOL_PIN_4);
  //Start serial comms and set baud rate to 115200
  Serial.begin(250000);
  //We also don't want the arduino waiting for too long on serial
  //Wait 2 milleseconds before timing out
  Serial.setTimeout(2);
}
```

Done uploading.

```
Sketch uses 5,300 bytes (16%) of program storage space. Maximum is 32
Global variables use 259 bytes (12%) of dynamic memory, leaving 1,789
```

11. In the serial monitor menu type in "[5,50]" and then press "Enter" to start a 5 Hz 50 ms pulse width LED blink. Other frequencies and intensities of blinking can be used by typing in "[desired_frequency, desired_led_on_pulse_width]". (Note: the Arduino does not check input so it is possible to enter bogus parameters like: "[50,25]" which will just result in the LEDs being always on)

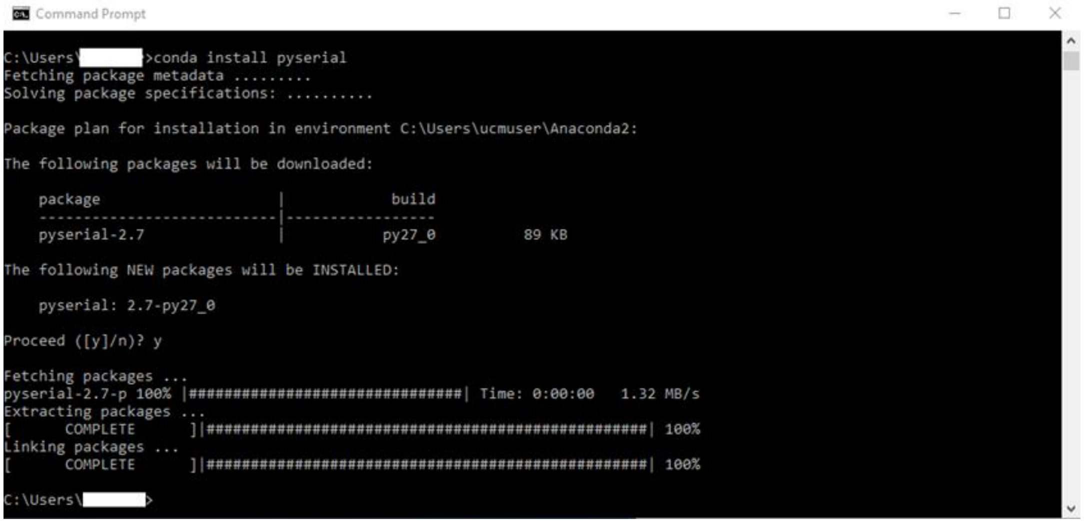
Setting up Python and OpenCV support:

12. Download and install Python Anaconda 2.7 64-bit (https://repo.anaconda.com/archive/)
    a. Anaconda2-5.0.0-Windows-x86_64.exe

13. Now open a command prompt (as administrator if possible) and type: *"conda update conda".* Say yes to the command prompt asking if you would like to update the conda environment.
    a. computers 'command prompt', NOT python or anaconda 'command prompt'

14. Now type *"conda update anaconda".* Say yes to the command prompt asking if you would like to update.

15. Finally type in *"conda install pyserial"* and okay the prompt asking if you would like to install the "NEW package". You should see the following:



```
C:\Users\[redacted]>conda install pyserial
Fetching package metadata .........
Solving package specifications: ..........

Package plan for installation in environment C:\Users\ucmuser\Anaconda2:

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    pyserial-2.7               |           py27_0          89 KB

The following NEW packages will be INSTALLED:

    pyserial: 2.7-py27_0

Proceed ([y]/n)? y

Fetching packages ...
pyserial-2.7-p 100% |###############################| Time: 0:00:00   1.32 MB/s
Extracting packages ...
[      COMPLETE      ]|##################################################| 100%
Linking packages ...
[      COMPLETE      ]|##################################################| 100%

C:\Users\[redacted]>
```

a. **IMPORTANT**: You need pyserial version 2.7. If your computer installed a newer version. Then do the following:
b. In command prompt, type in: *"pip install pyserial* == *2.7"*

16. Download OpenCV for windows (version 3.1) (http://opencv.org/downloads.html)
    a. Sourceforge link: https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.1.0/opencv-3.1.0.exe/download

17. Running the downloaded OpenCV file will open up an unzip location prompt. Extract the OpenCV folder to a desired location (the location doesn't really matter. Desktop is recommended since it is easy to access)

18. In the unzipped OpenCV folder go to: \opencv\build\python\2.7\x64 (if you extracted the OpenCV folder to the desktop the path should look something like: "C:\Users\user\Desktop\opencv\build\python\2.7\x64"). This folder should contain a file called "cv2.pyd". "Ctrl+C" this file.

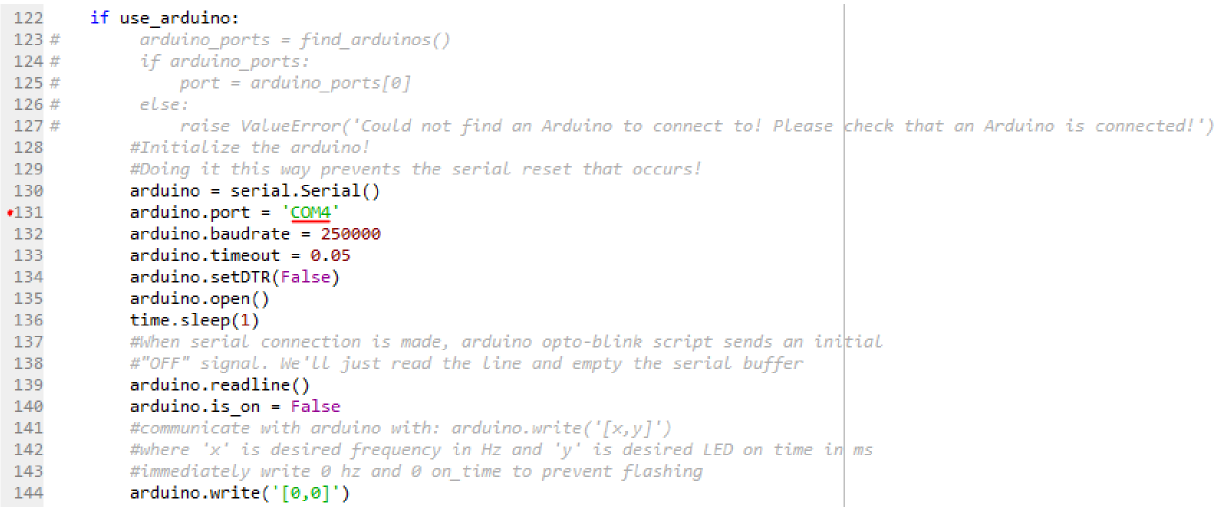
19. Copy the "cv2.pyd" file to the "site-packages" folder of your Anaconda Python distribution.
    a. If you installed Anaconda Python for all users then the path to site-packages should look something like: "C:\Anaconda2\Lib\site-packages"
    b. If you installed Anaconda Python only for yourself then the path to site-packages should look something like: "C:\Users\your_user_name\Anaconda2\Lib\site-packages"

Connecting Arduino ‘COM’ to Python

20. Start up Spyder. If using Windows, make sure to start the program as an administrator! (an IDE that is included in the Anaconda Python install package and that should have already been installed)

21. Once Spyder has started up, use it to open “fly_group_activity_monitor_master.py” located in the fly-group-activity-monitor-master folder that was downloaded and extracted. The folder path to the .py file should look something like: “\fly-group-activity-monitor-master\fly_group_activity_monitor\”

22. Change Line 131 “arduino.port = ‘COM’” to include the port number where Arduino is connected (see step 6). (e.g. “arduino.port = ‘COM5’”)



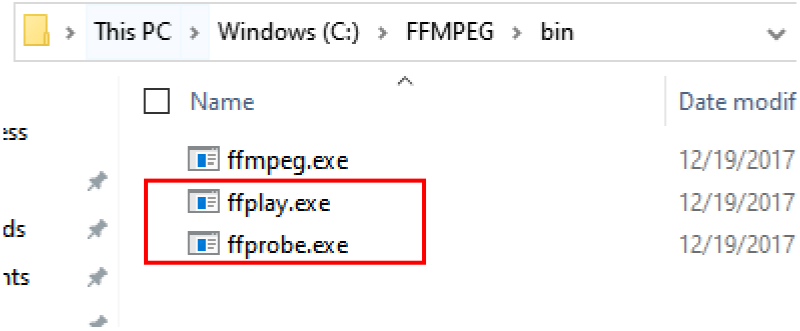
```
122     if use_arduino:
123 #        arduino_ports = find_arduinos()
124 #        if arduino_ports:
125 #            port = arduino_ports[0]
126 #        else:
127 #            raise ValueError('Could not find an Arduino to connect to! Please check that an Arduino is connected!')
128         #Initialize the arduino!
129         #Doing it this way prevents the serial reset that occurs!
130         arduino = serial.Serial()
131         arduino.port = 'COM4'
132         arduino.baudrate = 250000
133         arduino.timeout = 0.05
134         arduino.setDTR(False)
135         arduino.open()
136         time.sleep(1)
137         #When serial connection is made, arduino opto-blink script sends an initial
138         #"OFF" signal. We'll just read the line and empty the serial buffer
139         arduino.readline()
140         arduino.is_on = False
141         #communicate with arduino with: arduino.write('[x,y]')
142         #where 'x' is desired frequency in Hz and 'y' is desired LED on time in ms
143         #immediately write 0 hz and 0 on_time to prevent flashing
144         arduino.write('[0,0]')
```

23. Save changes to “fly_group_activity_monitor_master.py” and close out of Spyder

Setting up FFMPEG

24. Use directions from https://www.wikihow.com/Install-FFmpeg-on-Windows to download and install ffmpeg 64-bit static (version: 2015-06-05 git-7be0f48) for the relevant OS.
    a. (https://ffmpeg.org/download.html#build-linux)
    b. (https://ffmpeg.org/download.html#build-windows)
    c. (https://ffmpeg.org/download.html#build-mac)

25. Extract the downloaded compressed ffmpeg static file to the Desktop. Rename the extracted folder to “FFMPEG”. Finally, move this renamed folder to the “C:” drive. The path to the ffmpeg executable (as well as ffplay and ffprobe) should be found in “C:\FFMPEG\bin\” if the above steps have been done correctly.
    a. Builds we use: 2015-06-05 git-7be0f48 & ffmpeg-20171215-2d131fc-win64-static
    b. See images below to confirm your FFMPEG has the necessary files

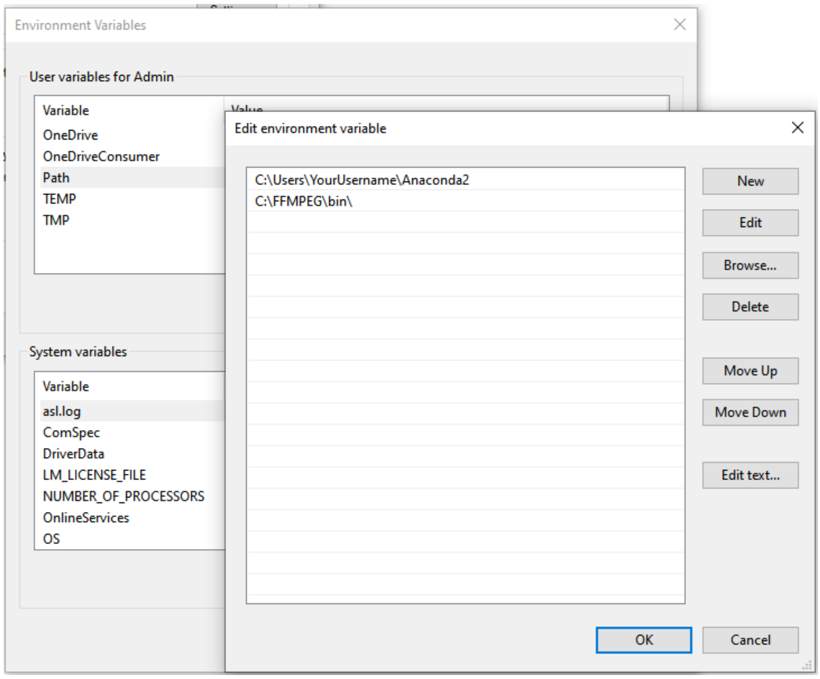


This PC › Windows (C:) › FFMPEG › bin

Name	Date modif
ffmpeg.exe	12/19/2017
ffplay.exe	12/19/2017
ffprobe.exe	12/19/2017

Set Up Environment Variables

26. On computer, go to “edit system environment variables” control panel
27. Within the “Advanced tab” click on “Environment Variables”
28. Within “User variables for Admin” for variable name “Path” add
    a. C:\Users\____\Anaconda2
    b. C:\FFMPEG\bin\



Environment Variables

User variables for Admin

Variable	Value
OneDrive	
OneDriveConsumer	
Path	
TEMP	
TMP	

System variables

Variable
asl.log
ComSpec
DriverData
LM_LICENSE_FILE
NUMBER_OF_PROCESSORS
OnlineServices
OS

Edit environment variable

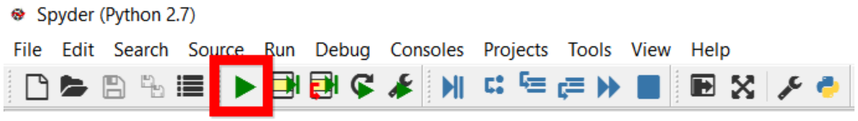
C:\Users\YourUsername\Anaconda2
C:\FFMPEG\bin\

Camera calibration and testing that everything works:

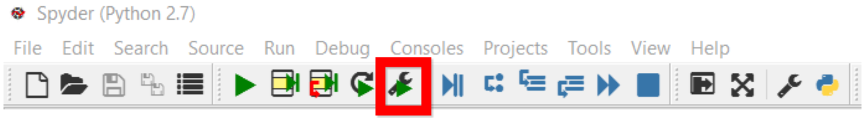
29. Start up Spyder. If using Windows, make sure to start the program as an administrator!

30. Once Spyder has started up, use it to open “fly_group_activity_monitor_gui.py” located in the fly-group-activity-monitor-master folder that was downloaded and extracted. The folder path to the .py file should look something like: “\fly-group-activity-monitor-master\fly_group_activity_monitor\”

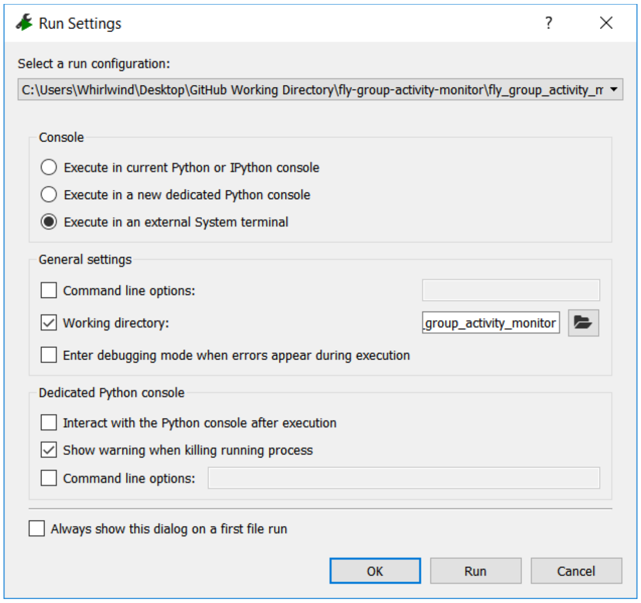
31. Once the “fly_group_activity_monitor_gui.py” file has been opened press the “Play” button located in the top menu.



   a. If this is is not your first time running a python file then you’ll need to click the “Run Settings” button which looks like:



   b. If this is the first time running a python file you should see the following menu and run the file and you should see a user interface pop up


Run Settings

Select a run configuration:

C:\Users\Whirlwind\Desktop\GitHub Working Directory\fly-group-activity-monitor\fly_group_activity_m

Console

○ Execute in current Python or IPython console

○ Execute in a new dedicated Python console

● Execute in an external System terminal

General settings

□ Command line options:

☑ Working directory: group_activity_monitor

□ Enter debugging mode when errors appear during execution

Dedicated Python console

□ Interact with the Python console after execution

☑ Show warning when killing running process

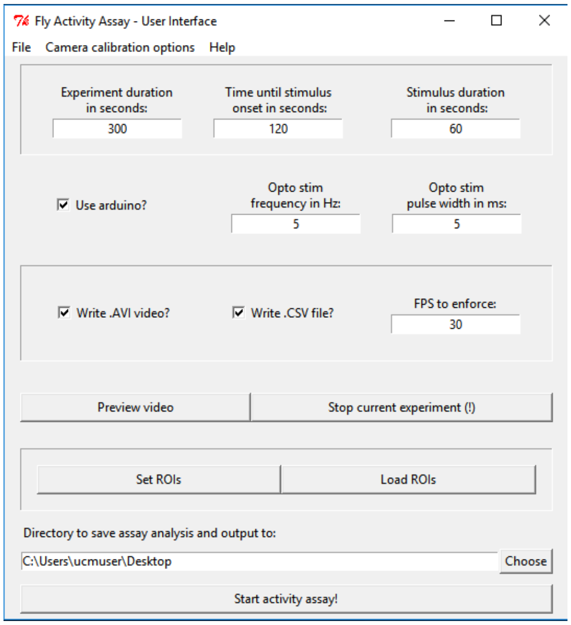
□ Command line options:

□ Always show this dialog on a first file run

32. Make sure to remember what the current settings are as you may want to revert your run settings for other python programs.

33. The settings for flyGrAM should use *“Execute in an external System terminal”* found under the “Console” subheading.

34. If successfully run, you should see the following user interface pop up


Fly Activity Assay - User Interface
File
Camera calibration options
Help
Preview video
Stop current experiment (!)
Set ROIs
Load ROIs
Choose
Start activity assay!

Experiment duration in seconds: 300

Time until stimulus onset in seconds: 120

Stimulus duration in seconds: 60

☑ Use arduino?

Opto stim frequency in Hz: 5

Opto stim pulse width in ms: 5

☑ Write .AVI video?

☑ Write .CSV file?

FPS to enforce: 30

Directory to save assay analysis and output to: C:\Users\ucmuser\Desktop

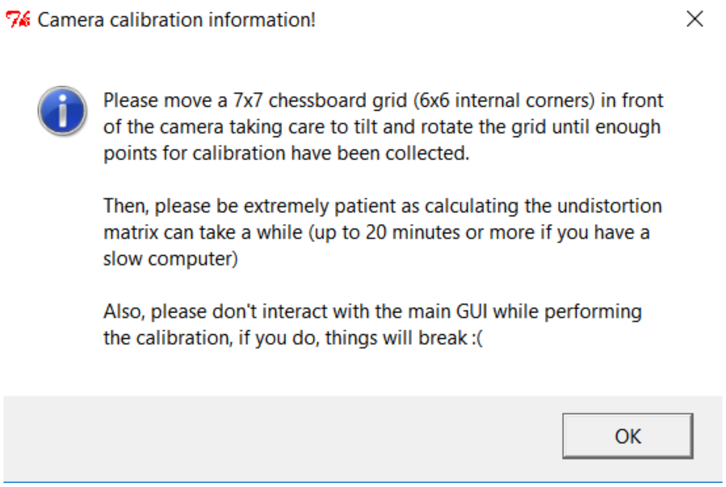
35. First, check that the camera is operational. To do so, click the “Preview video” button on the interface. (Note: the video window may take some time to show up. DO NOT click the “Preview video” button multiple times)
    a. If the video that shows up is from a laptop webcam or another camera. You will need to go into your “device manager” and “disable” the camera. You can always go back and re-enable it later via the “device manager”

Note that the preview video that is initially taken will feature some distortion (straight edges in the video will look curved). To correct for this distortion a camera calibration will need to be run.

36. Print off the 10mm checkerboard found in “\fly-group-activity-monitor\camera_calibration” and make sure to mount it on a solid flat surface. Make sure there are no wrinkles or distortions of the chessboard!

IMPORTANT: Before running the camera calibration, please make sure that any infrared (IR) filter on the camera lens has been taken off and any IR back-illumination has been turned off. It is also critical that the position of the camera as well as its focus has been optimized. If the distance of the camera from the arenas or the lens focus is changed, then the calibration will need to be performed again.

37. To run the calibration, select the “calibrate camera” option in the menu of the flyGrAM user interface. Read the directions that pop up and make sure to save the resulting calibration file. (Note: If the directions are not informative enough see: https://www.youtube.com/watch?v=kizO_s-YUDU for a video example of a similar calibration process)



Please move a 7x7 chessboard grid (6x6 internal corners) in front of the camera taking care to tilt and rotate the grid until enough points for calibration have been collected.

Then, please be extremely patient as calculating the undistortion matrix can take a while (up to 20 minutes or more if you have a slow computer)

Also, please don't interact with the main GUI while performing the calibration, if you do, things will break :(

IMPORTANT: To get a good calibration, it is important to move the checkerboard at a slow but steady rate, taking care to rotate and translate the board so that it ends up at many different positions and angles. Calibration points will show as a rainbow colored array overlaid on the checkerboard. However, if the camera can’t detect all vertices of the checkerboard then only the previous good image will be shown.

NOTE: After collecting enough calibration points, the main “Fly Activity Assay - User Interface” window will likely show “(Not Responding)”. This is normal. Wait until the Python console shows “RMS, camera_matrix, distortion coefficients, mean error, and reprojection error” before resuming interaction with the user interface.

Troubleshooting

Problem: Command prompt issues

Solution: Make sure you run command prompt as administrator. (Right click on icon, select ‘run as administrator’.)

Problem: ‘conda’ not recognized as internal or external command

Solution: This is likely an environment variable issue. See steps 26-28 above.

Additional Support:
https://stackoverflow.com/questions/44515769/conda-is-not-recognized-as-internal-or-external-command

Problem: matplotlib related errors

Solution: You most likely need a different version of matplotlib. Run command prompt as administrator. Type in *‘pip install matplotlib* == *“version #”’*