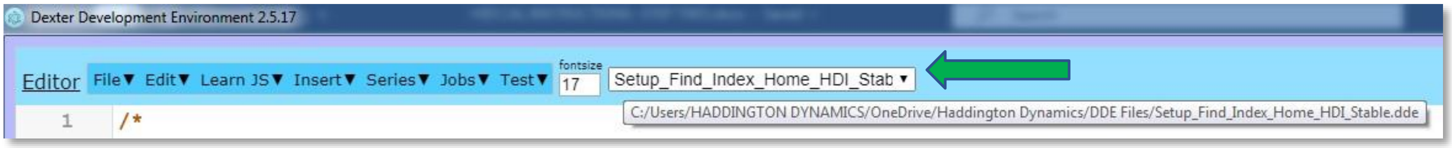
INSTRUCTIONS for HDI CALIBRATION- Step Three

Line up your X on the Base Long & notch on the Base Code Disk

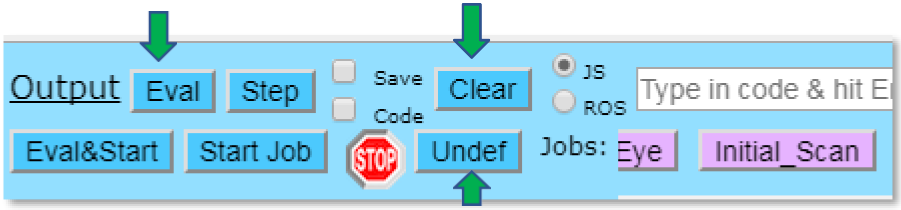
1. A successful Step Two MUST be completed before proceeding with this Step Three.
2. Perform a fresh Reboot of Dexter.
3. Open DDE and make sure the Setup_Find_Index* file below is selected and shown in the window.


Dexter Development Environment 2.5.17
Editor
File▼
Edit▼
Learn JS▼
Insert▼
Series▼
Jobs▼
Test▼
fontsize
17
Setup_Find_Index_Home_HDI_Stab ▾
1

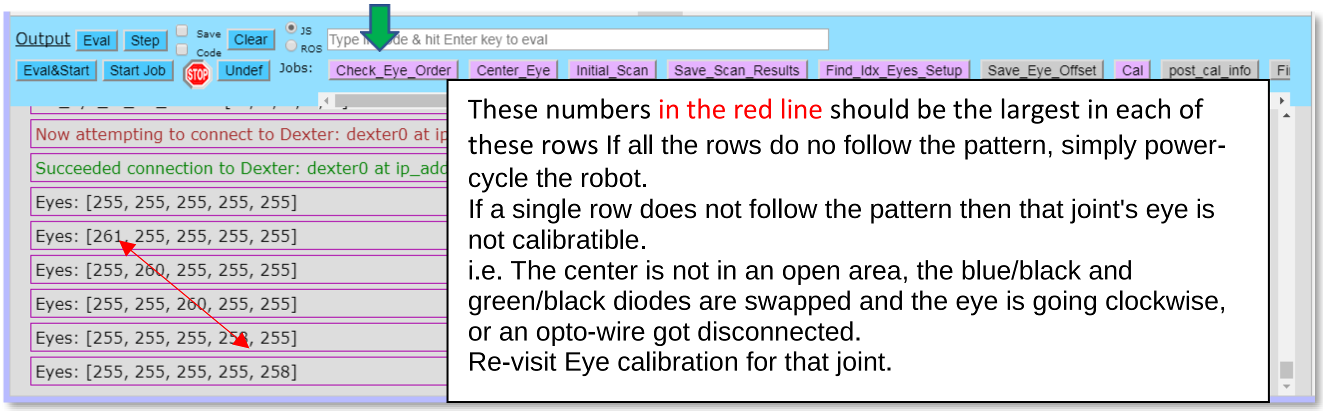
```
/*
```

C:/Users/HADDINGTON DYNAMICS/OneDrive/Haddington Dynamics/DDE Files/Setup_Find_Index_Home_HDI_Stable.dde

then select Undef, Clear & Eval in that order. (Make sure cursor is inside the Editor pane and nothing is highlighted)



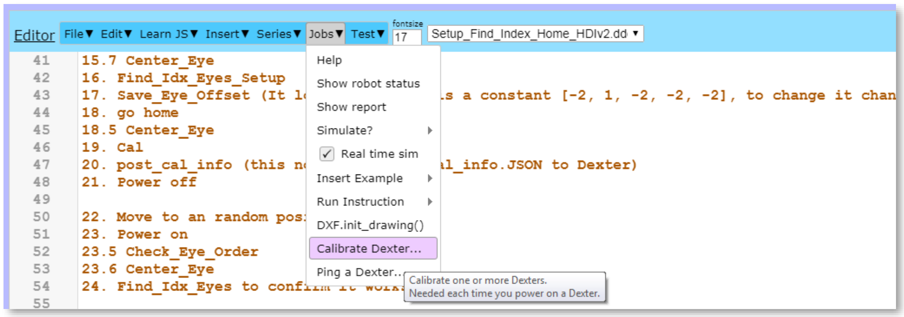
4. Select the Check_Eye_Order box.



```
Now attempting to connect to Dexter: dexter0 at ip
Succeeded connection to Dexter: dexter0 at ip_add
Eyes: [255, 255, 255, 255, 255]
Eyes: [261, 255, 255, 255, 255]
Eyes: [255, 260, 255, 255, 255]
Eyes: [255, 255, 260, 255, 255]
Eyes: [255, 255, 255, 258, 255]
Eyes: [255, 255, 255, 255, 258]
```

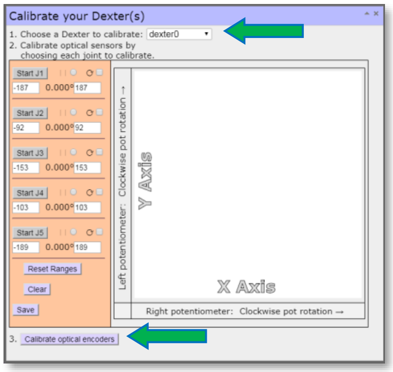
These numbers in the red line should be the largest in each of these rows If all the rows do no follow the pattern, simply power-cycle the robot.
If a single row does not follow the pattern then that joint's eye is not calibratible.
i.e. The center is not in an open area, the blue/black and green/black diodes are swapped and the eye is going clockwise, or an opto-wire got disconnected.
Re-visit Eye calibration for that joint.

5. Next, Select Calibrate Optical Encoders **USE CALIBRATION WINDOW'S CAL FOR THIS**



```
15.7 Center_Eye
16. Find_Idx_Eyes_Setup
17. Save_Eye_Offset (It l                       s a constant [-2, 1, -2, -2, -2], to change it chan
18. go home
18.5 Center_Eye
19. Cal
20. post_cal_info (this n                       al_info.JSON to Dexter)
21. Power off

22. Move to an random pos:
23. Power on
23.5 Check_Eye_Order
23.6 Center_Eye
24. Find_Idx_Eyes to confirm it work:
```

Calibrate one or more Dexters.
Needed each time you power on a Dexter.

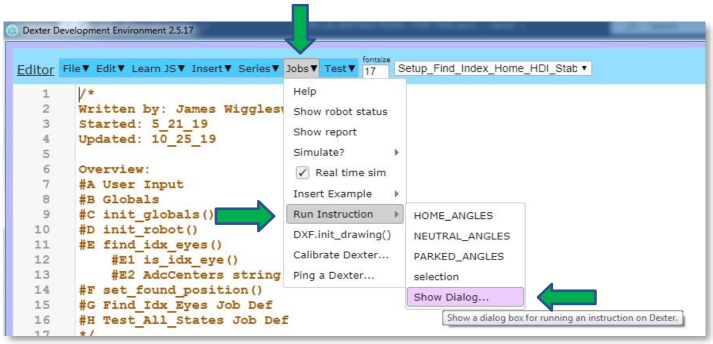


1. Choose a Dexter to calibrate: dexter0
2. Calibrate optical sensors by choosing each joint to calibrate.

-187	0.000°	187
-92	0.000°	92
-153	0.000°	153
-103	0.000°	103
-189	0.000°	189

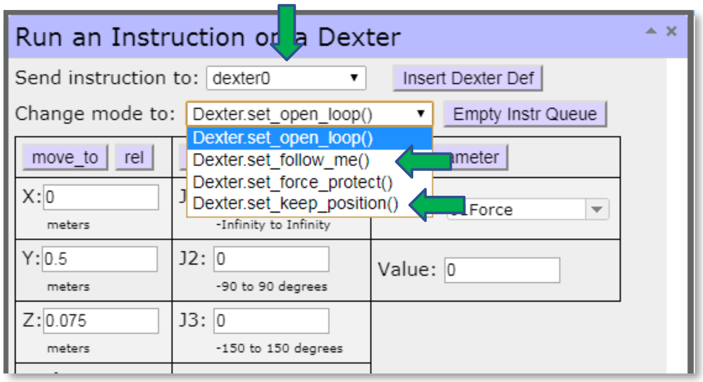
3. Calibrate optical encoders

6. After successful full calibration where Dexter moves each joint in their full range you can check Dexter for smooth joint movements by manually moving all the joints. To do this - Go to the main menu by selecting Jobs/ Run Instruction/Show Dialog.



```
/*
Written by: James Wigglesw
Started: 5_21_19
Updated: 10_25_19

Overview:
#A User Input
#B Globals
#C init_globals()
#D init_robot()
#E find_idx_eyes()
    #E1 is_idx_eye()
    #E2 AdcCenters string
#F set_found_position()
#G Find_Idx_Eyes Job Def
#H Test_All_States Job Def
*/
```

Show a dialog box for running an instruction on Dexter.



Send instruction to: dexter0

Change mode to: Dexter.set_open_loop()

X: 0 meters

J -Infinity to Infinity

Y: 0.5 meters

J2: 0 -90 to 90 degrees

Value: 0

Z: 0.075 meters

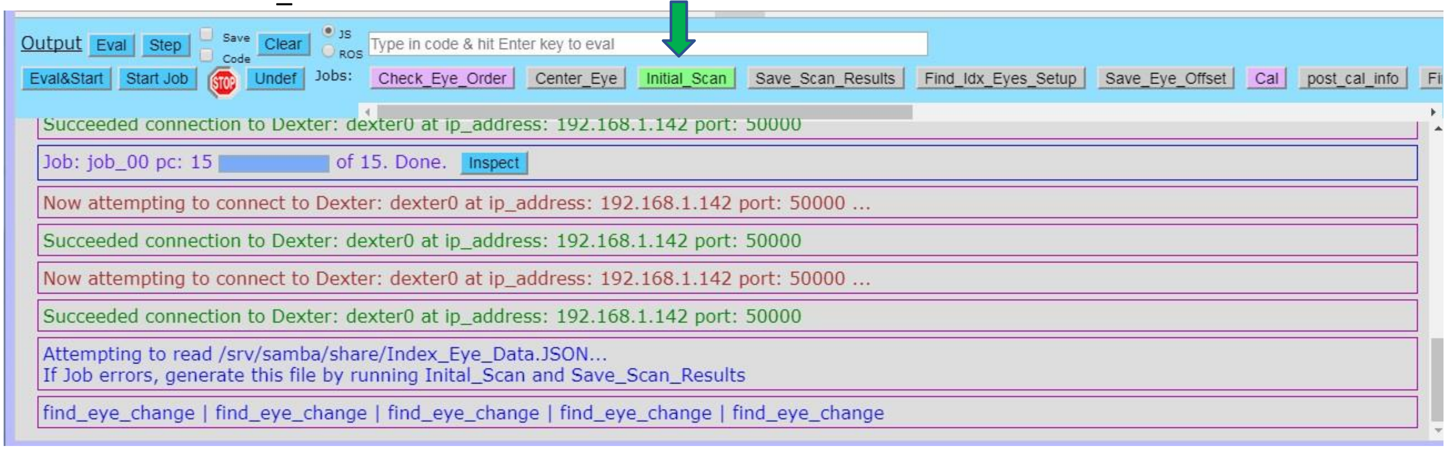
J3: 0 -150 to 150 degrees

With the Run Instruction On a Dexter window open, select Send Instruction to: and select dexter 0 in the drop down box...then select Change Mode to: Dexter.set_Follow_Me(). This will unlock the Dexter so you can manually move around all joints and check out Follow Me mode for smooth operation of all joints.

If the robot does not move freely and smoothly, take steps to review the offending joint. This could be caused by an incomplete movement of the joint during the calibrating of optical encoders from an obstruction or something has changed on the code disk eyes after calibrating optical encoders. Once finished Change Mode to: Dexter.set_Keep_Position()then OpenLoop (InitalScan needs to be done in OpenLoop)

Close Run Instruction window.

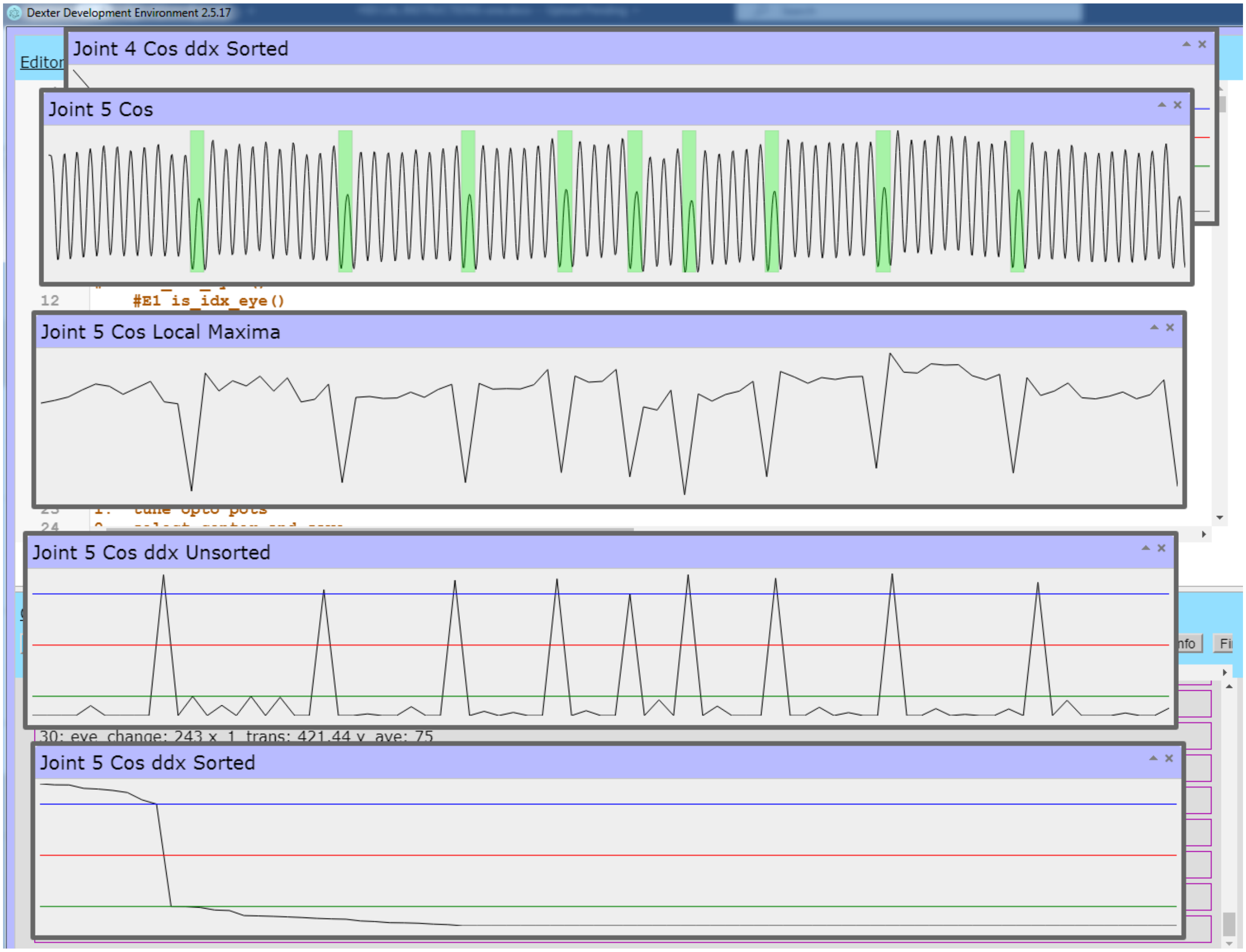
7. Next select Initial_Scan



```
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Job: job_00 pc: 15 of 15. Done.
Now attempting to connect to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000 ...
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Now attempting to connect to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000 ...
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Attempting to read /srv/samba/share/Index_Eye_Data.JSON...
If Job errors, generate this file by running Inital_Scan and Save_Scan_Results
find_eye_change | find_eye_change | find_eye_change | find_eye_change | find_eye_change
```

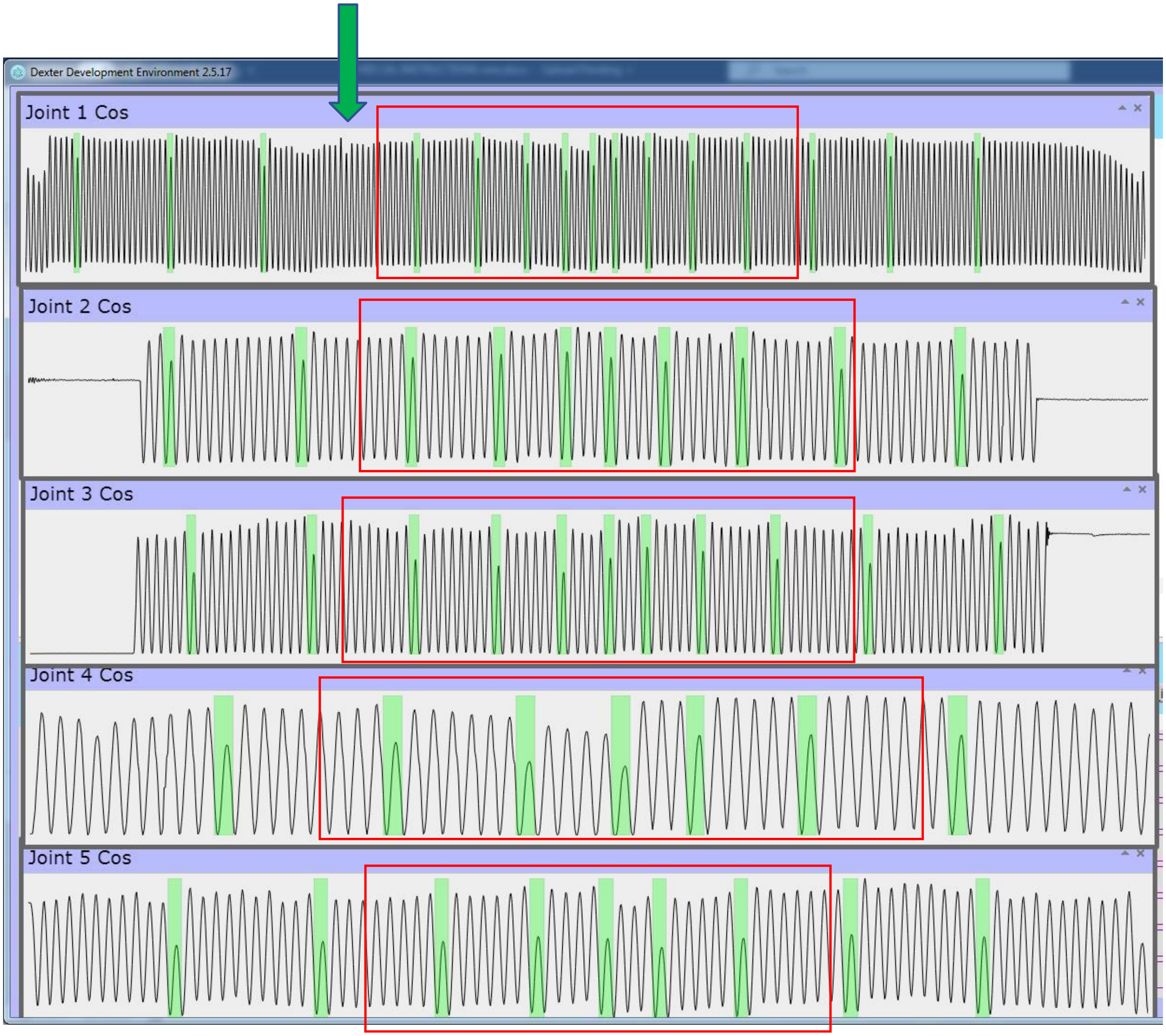
The following scans will show up on the screen stacked on top of each other. There will be 4 windows per joint to review. You can move the windows down to review that joint's series of scans all at once as pictured. The most important window is the one with the green spikes that show the Index pulses highlighted as a Green spike.

Here you need to verify that the code disk for each joint is performing properly. The starting point for each code disk will be the 3 black spikes that show the starting position of each code disk. The green spikes represent the smaller Index Pulse openings on each code disk. Dexter uses these smaller index pulses to locate where each joint is positioned, so this is a critical review step. You need to verify the spike count on each side of the code disk as follows. From the left or right of the three black spikes, the green spikes will show separation that will incrementally increase in even numbers on one side and odd numbers on the other side. Note that some code disk's odd and even sides are opposite of the others. In the window below you can see to the right of the three black spikes that the black spikes increase in odd numbers starting with three spikes, then five spikes, seven spikes etc. Count and verify that these spikes increase by two between each green spike separation. On the other side verify that the spikes start with four spikes then increase by two each time. Once verified, proceed to the next joint's set of scans.



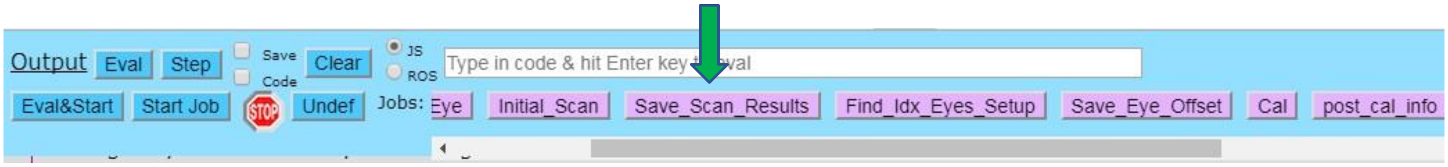
```
#E1 is_idx_eye()
1.  tune opto pots
30: eye change: 243 x 1 trans: 421.44 v ave: 75
```

You may find there is a failure between the green spikes like shown below (Green Arrow). In this graph you can see that there was a failure to recognize the smaller index pulse between the twelve and fourteen separations. Since the separation represents a small index in the code disk it would be difficult to make that opening smaller and ever be able to get that index pulse read and identified with a green spike. However, the location of this failure is far enough away from the center three pulses that if Dexter is positioned close to home position on startup it should never be an issue. Please note the limit area where there can be no failures on each joint below. The area that's needs to be free of failure is highlighted with the red boxes.



*If there is an issue with any joint outside of these areas, review the troubleshooting guide. Extra green spikes are usually the result of something blocking that opening like a hair, or small dust particles you can fix by counting through the indexes and locating the blockage. After mitigating the issue rerun the initial scan and review.

8. When initial scan looks good, Select Save_Scan_Results.

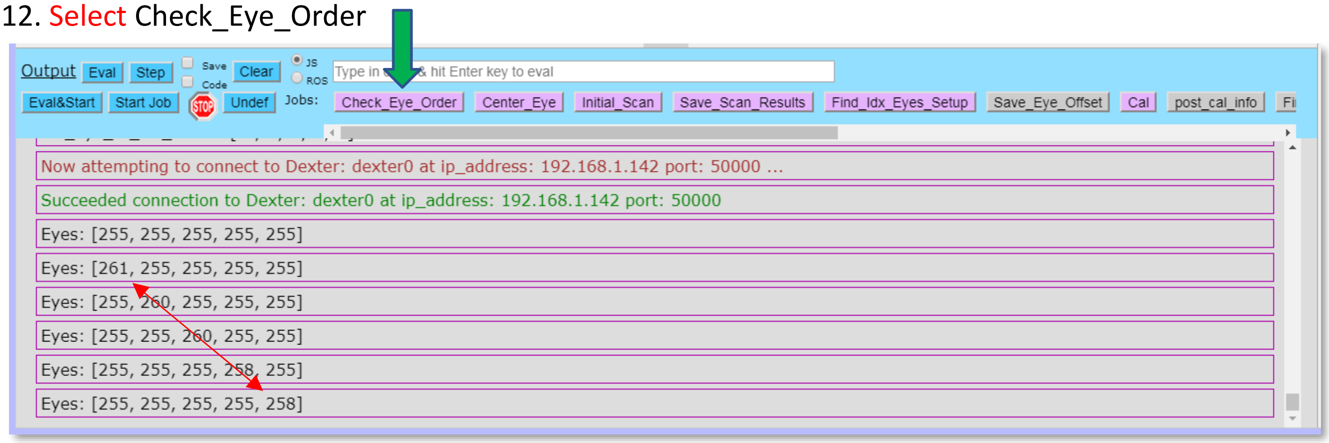


9. Power off

10. Manually move to where you want to define home to be (this will be J1 at X mark and allother joints at our known visual zero point)

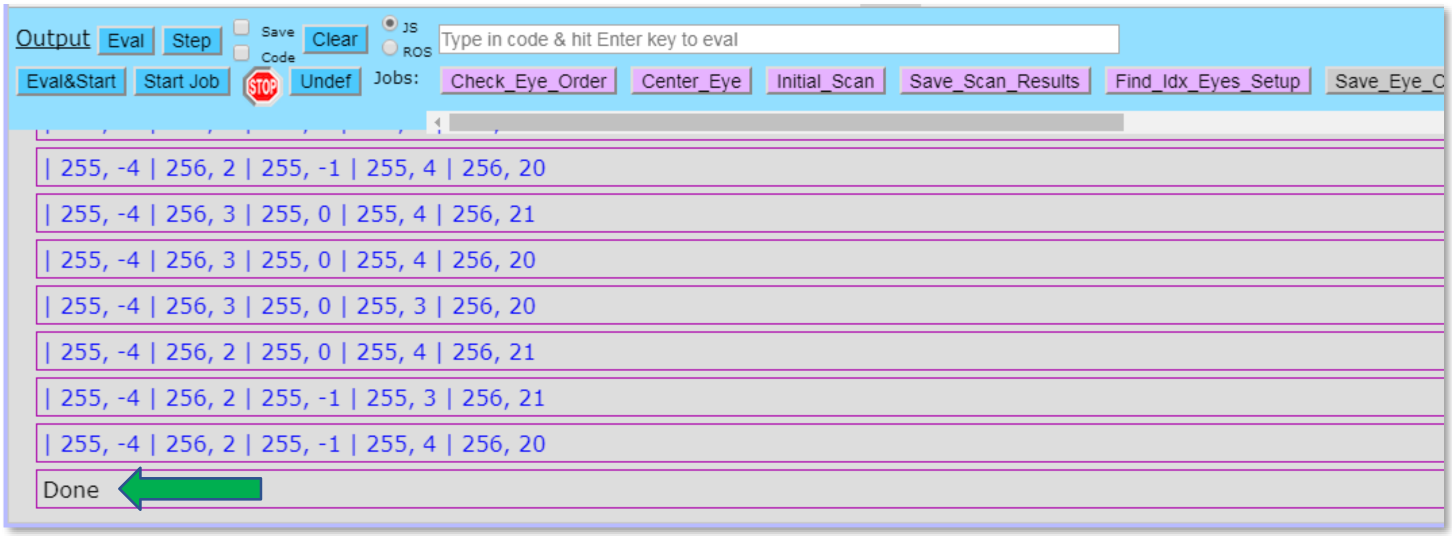
11. Power On

12. Select Check_Eye_Order



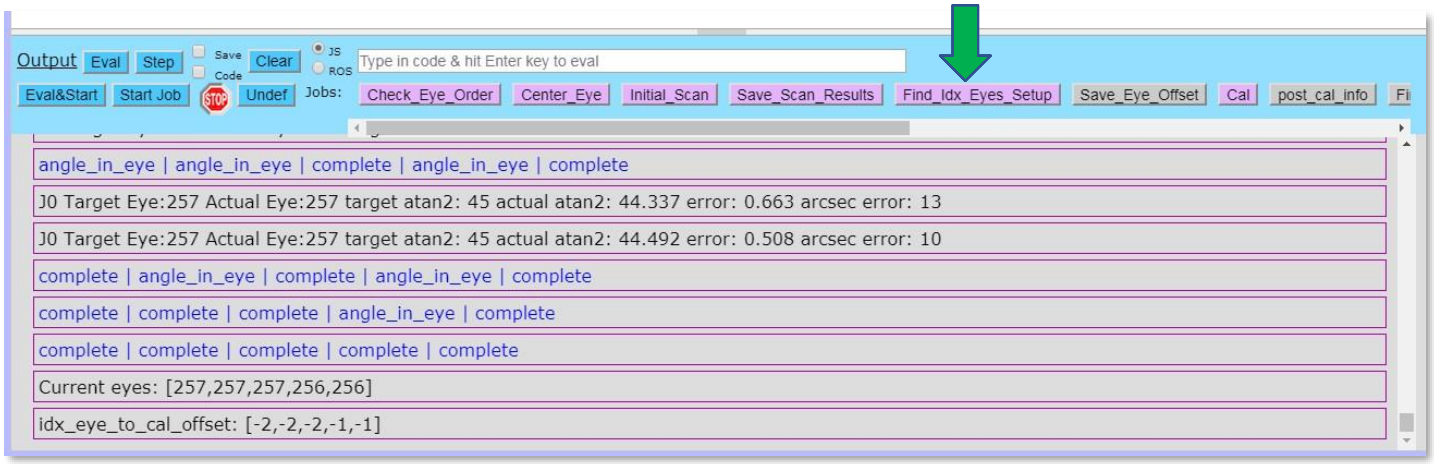
```
Now attempting to connect to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000 ...
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Eyes: [255, 255, 255, 255, 255]
Eyes: [261, 255, 255, 255, 255]
Eyes: [255, 260, 255, 255, 255]
Eyes: [255, 255, 260, 255, 255]
Eyes: [255, 255, 255, 258, 255]
Eyes: [255, 255, 255, 255, 258]
```

13. Select Center_Eye and wait until you see Done at the bottom of the window.



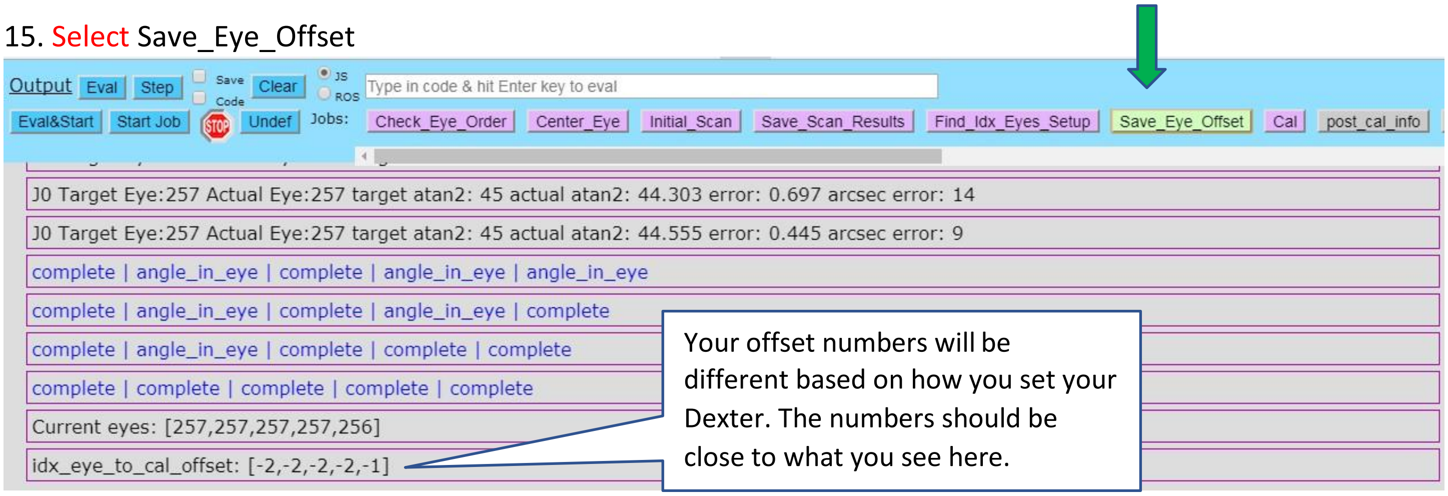
```
| 255, -4 | 256, 2 | 255, -1 | 255, 4 | 256, 20
| 255, -4 | 256, 3 | 255, 0 | 255, 4 | 256, 21
| 255, -4 | 256, 3 | 255, 0 | 255, 4 | 256, 20
| 255, -4 | 256, 3 | 255, 0 | 255, 3 | 256, 20
| 255, -4 | 256, 2 | 255, 0 | 255, 4 | 256, 21
| 255, -4 | 256, 2 | 255, -1 | 255, 3 | 256, 21
| 255, -4 | 256, 2 | 255, -1 | 255, 4 | 256, 20
Done
```

14. Select Find_Idx_Eyes_Setup and wait until you see the offset at the bottom of the window.



```
angle_in_eye | angle_in_eye | complete | angle_in_eye | complete
J0 Target Eye:257 Actual Eye:257 target atan2: 45 actual atan2: 44.337 error: 0.663 arcsec error: 13
J0 Target Eye:257 Actual Eye:257 target atan2: 45 actual atan2: 44.492 error: 0.508 arcsec error: 10
complete | angle_in_eye | complete | angle_in_eye | complete
complete | complete | complete | angle_in_eye | complete
complete | complete | complete | complete | complete
Current eyes: [257,257,257,256,256]
idx_eye_to_cal_offset: [-2,-2,-2,-1,-1]
```

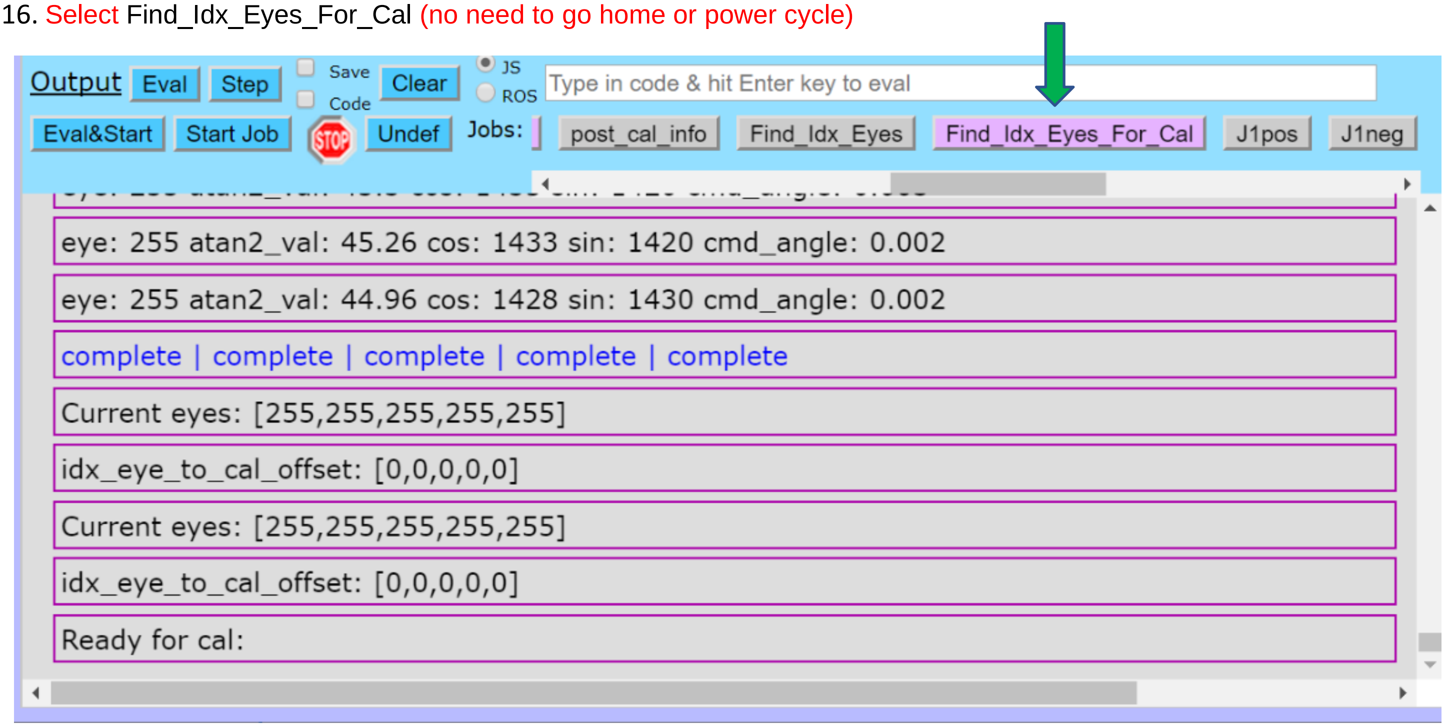
15. Select Save_Eye_Offset



```
J0 Target Eye:257 Actual Eye:257 target atan2: 45 actual atan2: 44.303 error: 0.697 arcsec error: 14
J0 Target Eye:257 Actual Eye:257 target atan2: 45 actual atan2: 44.555 error: 0.445 arcsec error: 9
complete | angle_in_eye | complete | angle_in_eye | angle_in_eye
complete | angle_in_eye | complete | angle_in_eye | complete
complete | angle_in_eye | complete | complete | complete
complete | complete | complete | complete | complete
Current eyes: [257,257,257,257,256]
idx_eye_to_cal_offset: [-2,-2,-2,-2,-1]
```

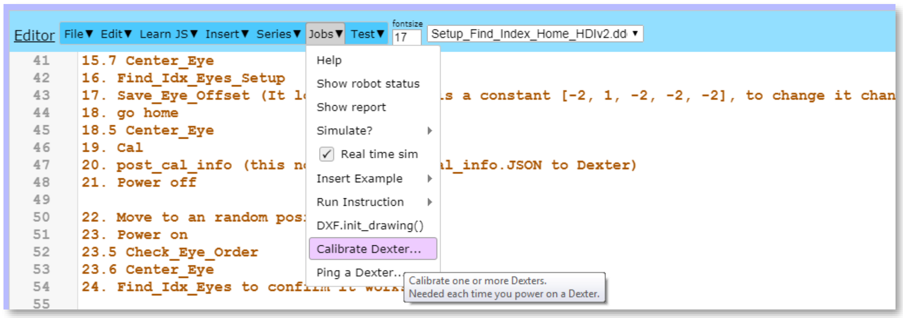
Your offset numbers will be different based on how you set your Dexter. The numbers should be close to what you see here.

16. Select Find_Idx_Eyes_For_Cal (no need to go home or power cycle)



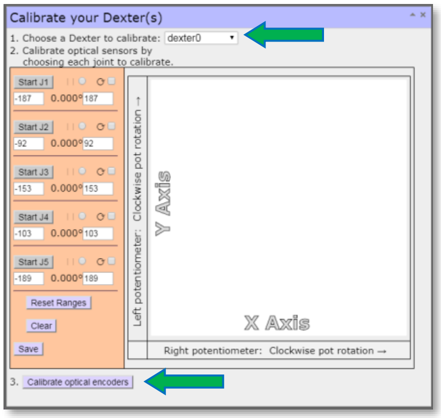
```
eye: 255 atan2_val: 45.26 cos: 1433 sin: 1420 cmd_angle: 0.002
eye: 255 atan2_val: 44.96 cos: 1428 sin: 1430 cmd_angle: 0.002
complete | complete | complete | complete | complete
Current eyes: [255,255,255,255,255]
idx_eye_to_cal_offset: [0,0,0,0,0]
Current eyes: [255,255,255,255,255]
idx_eye_to_cal_offset: [0,0,0,0,0]
Ready for cal:
```

17. Next, Select Calibrate Optical Encoders **USE CALIBRATION WINDOW'S CAL FOR THIS**



```
15.7 Center_Eye
16. Find_Idx_Eyes_Setup
17. Save_Eye_Offset (It l                    s a constant [-2, 1, -2, -2, -2], to change it chan
18. go home
18.5 Center_Eye
19. Cal
20. post_cal_info (this n                   al_info.JSON to Dexter)
21. Power off

22. Move to an random pos
23. Power on
23.5 Check_Eye_Order
23.6 Center_Eye
24. Find_Idx_Eyes to confirm it work
```

Calibrate one or more Dexters.
Needed each time you power on a Dexter.

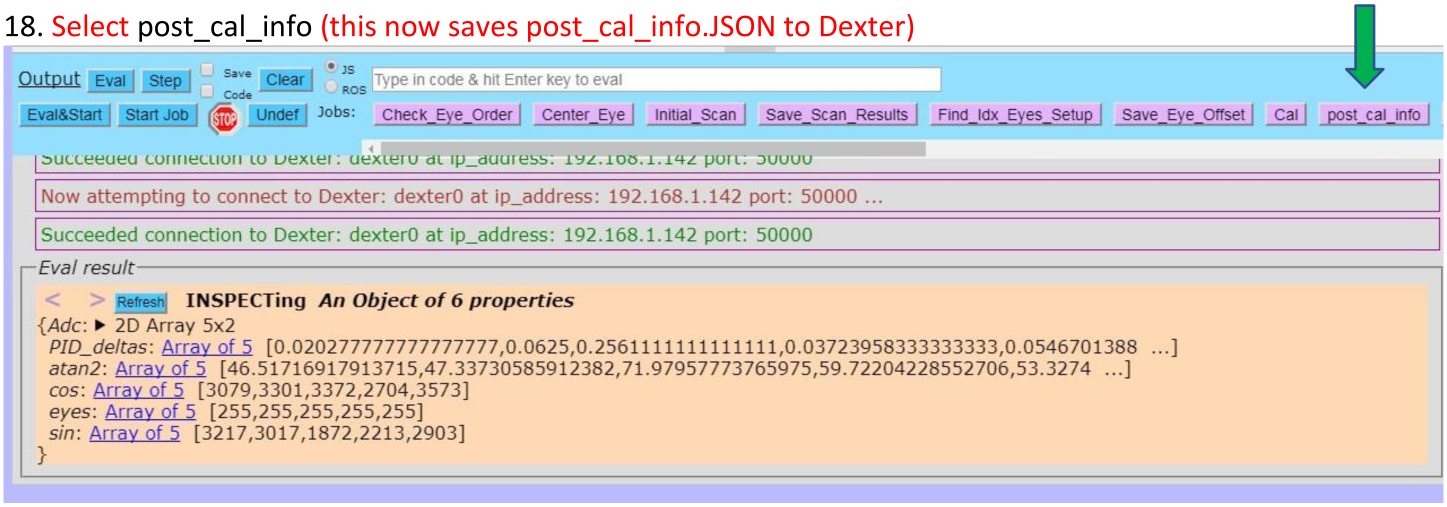

Calibrate your Dexter(s)
Start J1
Start J2
Start J3
Start J4
Start J5
Reset Ranges
Clear
Save
Y Axis
X Axis
Left potentiometer: Clockwise pot rotation →
Right potentiometer: Clockwise pot rotation →
Calibrate optical encoders

1. Choose a Dexter to calibrate: dexter0
2. Calibrate optical sensors by choosing each joint to calibrate.

-187	0.000°	187
-92	0.000°	92
-153	0.000°	153
-103	0.000°	103
-189	0.000°	189

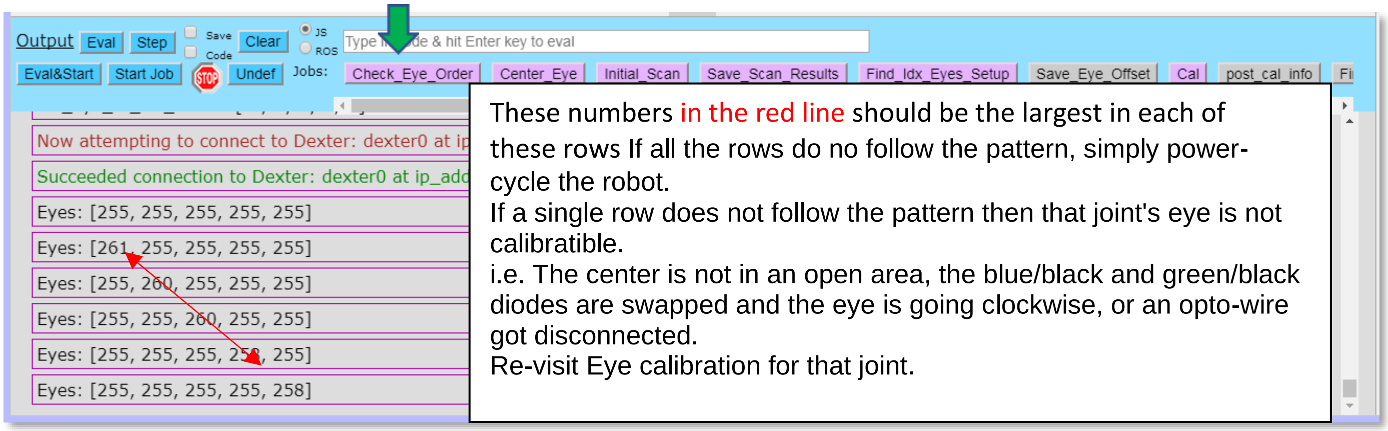
3. Calibrate optical encoders

18. Select post_cal_info (this now saves post_cal_info.JSON to Dexter)



```
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Now attempting to connect to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000 ...
Succeeded connection to Dexter: dexter0 at ip_address: 192.168.1.142 port: 50000
Eval result
INSPECTing  An Object of 6 properties
{Adc: ▶ 2D Array 5x2
 PID_deltas: Array of 5  [0.020277777777777777,0.0625,0.2561111111111111,0.037239583333333333,0.0546701388  ...]
 atan2: Array of 5  [46.51716917913715,47.33730585912382,71.97957773765975,59.72204228552706,53.3274  ...]
 cos: Array of 5  [3079,3301,3372,2704,3573]
 eyes: Array of 5  [255,255,255,255,255]
 sin: Array of 5  [3217,3017,1872,2213,2903]
}
```

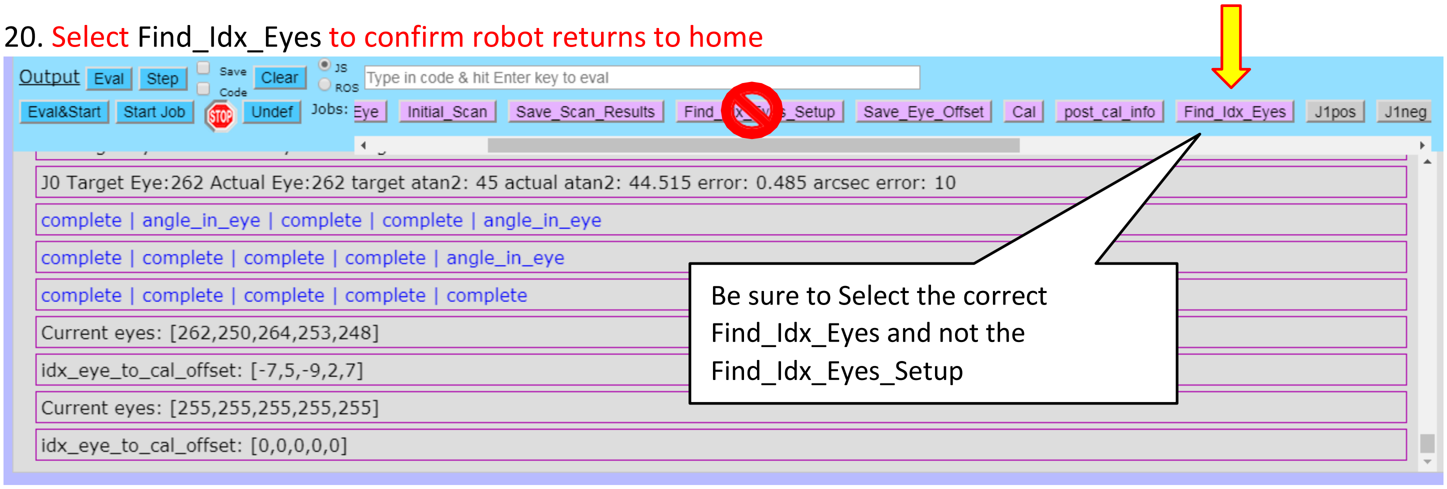
19. Select Check Eye Order



```
Now attempting to connect to Dexter: dexter0 at ip
Succeeded connection to Dexter: dexter0 at ip_add
Eyes: [255, 255, 255, 255, 255]
Eyes: [261, 255, 255, 255, 255]
Eyes: [255, 260, 255, 255, 255]
Eyes: [255, 255, 260, 255, 255]
Eyes: [255, 255, 255, 258, 255]
Eyes: [255, 255, 255, 255, 258]
```

These numbers in the red line should be the largest in each of these rows If all the rows do no follow the pattern, simply power-cycle the robot.
If a single row does not follow the pattern then that joint's eye is not calibratible.
i.e. The center is not in an open area, the blue/black and green/black diodes are swapped and the eye is going clockwise, or an opto-wire got disconnected.
Re-visit Eye calibration for that joint.

20. Select Find_Idx_Eyes to confirm robot returns to home



```
J0 Target Eye:262 Actual Eye:262 target atan2: 45 actual atan2: 44.515 error: 0.485 arcsec error: 10
complete | angle_in_eye | complete | complete | angle_in_eye
complete | complete | complete | complete | angle_in_eye
complete | complete | complete | complete | complete
Current eyes: [262,250,264,253,248]
idx_eye_to_cal_offset: [-7,5,-9,2,7]
Current eyes: [255,255,255,255,255]
idx_eye_to_cal_offset: [0,0,0,0,0]
```

Be sure to Select the correct Find_Idx_Eyes and not the Find_Idx_Eyes_Setup

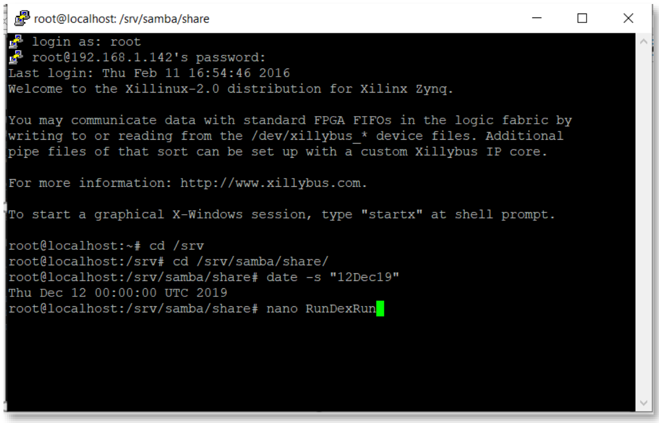
SET UP IS ALMOST COMPLETE

ImportantSee below for Putty file to update RunDexRun

This is the final step to activate the RunDexRun file so the robot can be ready to train. Follow the steps to enable the RunDexRun file on Dexter's hard drive.

Start the PuTTY App

The login is root   the password is klg

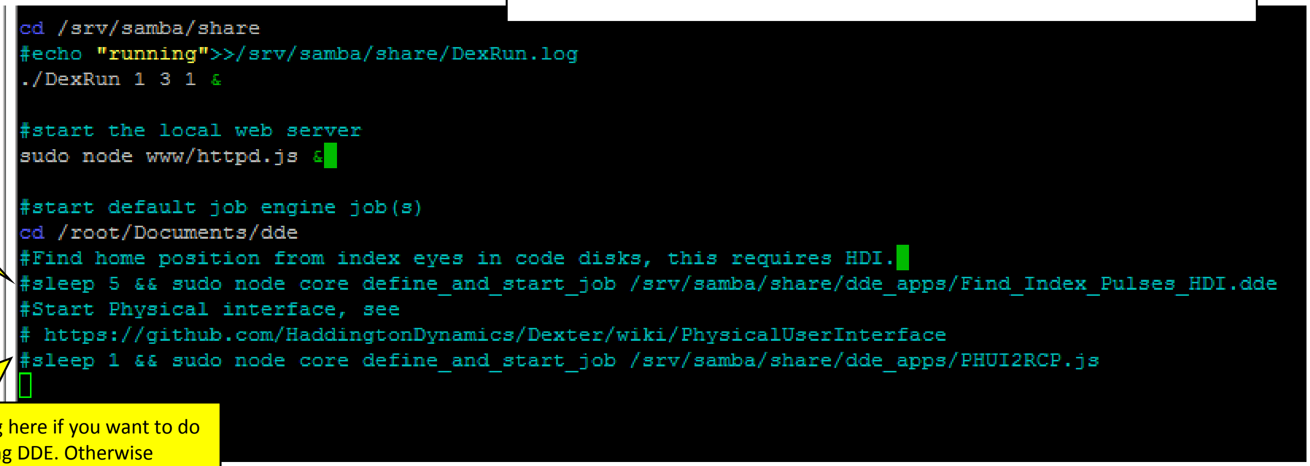


```
login as: root
root@192.168.1.142's password:
Last login: Thu Feb 11 16:54:46 2016
Welcome to the Xillinux-2.0 distribution for Xilinx Zynq.

You may communicate data with standard FPGA FIFOs in the logic fabric by
writing to or reading from the /dev/xillybus_* device files. Additional
pipe files of that sort can be set up with a custom Xillybus IP core.

For more information: http://www.xillybus.com.

To start a graphical X-Windows session, type "startx" at shell prompt.

root@localhost:~# cd /srv
root@localhost:/srv# cd /srv/samba/share/
root@localhost:/srv/samba/share# date -s "12Dec19"
Thu Dec 12 00:00:00 UTC 2019
root@localhost:/srv/samba/share# nano RunDexRun
```

At root@localhost:/srv# type cd /srv/samba/share/
At root@localhost:/srv/samba/share# type nano RunDexRun
This will open the RunDexRun file

Maximize the window and use the arrow keys to navigate to the bottom of the file. Remove the "#" in front of the lines pictured below.
Edit the line by navigating with your arrow keys and deleting the hashtag in front of the word sleep based on your needs specified in yellow below.

After deleting the hashtag hit CTRL O and then Enter then CTRL X to exit.



```
cd /srv/samba/share
#echo "running">>/srv/samba/share/DexRun.log
./DexRun 1 3 1 &

#start the local web server
sudo node www/httpd.js &

#start default job engine job(s)
cd /root/Documents/dde
#Find home position from index eyes in code disks, this requires HDI.
#sleep 5 && sudo node core define_and_start_job /srv/samba/share/dde_apps/Find_Index_Pulses_HDI.dde
#Start Physical interface, see
# https://github.com/HaddingtonDynamics/Dexter/wiki/PhysicalUserInterface
#sleep 1 && sudo node core define_and_start_job /srv/samba/share/dde_apps/PHUI2RCP.js
```

Always remove this hashtag.

Leave this hashtag here if you want to do programming using DDE. Otherwise remove for training on startup (Default).

Power cycle Dexter to enable changes to RunDexRun. It will take almost 3 minutes to reboot and then it is ready for training after the end effector nods at you.

Optional
If you ever need to go through the calibration steps again you will need to disable Find_Index_Pulses_HDI.dde and PHUI2RCP.js in RunDexRun. The robot will not run PHUI on bootup and allow you to redo the calibration. Re- add the "#" in front of the same lines as shown above.

After adding the hashtag hit CTRL O and then Enter then CTRL X to exit.

MaintenanceRun your Dexter through several training series and stress test for at 36 hours. Cone drive lubricant needs to be replaced after 100 hours and again at 2000 hours. Adjust belts as needed.