

ExoBoot control - Semantic Notes

From Semantic_Notes (Exo Actuator dev.)

- flexsea is installed on .venv on laptop and on global env (dephy created one on Rpi)
-

"TEST" is Backed up on box (2/5/25) - from Rpi

- C:\Users\maxsj\OneDrive\Desktop\RESEARCH\TEST\ZZ_Exo_Actuator <-Exo Actuator work
 - TEST is also name of file on RPi (backed up to BOX) <-Linux venv
 - NOT "ZZ_Exo_Actuator" folder in \Desktop\Research\ZZ_Exo_Actuator ← OLD!!
- C:\Users\maxsj\OneDrive\Desktop\RESEARCH\TEST ← .venv used for Actpack development
 - FlexSEA location

GET MATLAB - need to view files in Xiangyu controller

("C:\Users\maxsj\OneDrive\Desktop\RESEARCH\Xiangyu controller\Calibration\exoTorqueCalcCal.m")

Going to Use Rpi or lab desktop - leaning towards Rpi (to be consistant with Xiangyu)

Currently Rpi 4 is setup (Rpi 5 is NOT setup)

When moving code from desktop PC to Rpi - USE BOX

~~~~~

9/2/25

Created New folder on PC desktop called "**Exoboot\_dev**" containing resources to help code the controller

created .venv (selected to install dependencies for Actuator-Package-develop (actpack github fork)

- base is Python 3.11.9(Microsoft Store)

'pip install flexsea' (per Dephy) into .venv

"C:\Users\maxsj\OneDrive\Desktop\Exoboot\_dev\.venv\Lib\site-packages\flexsea-12.0.4.dist-info"

FlexSea Version installed on ->

"C:\Users\maxsj\OneDrive\Desktop\RESEARCH\.venv\Lib\site-packages\flexsea-9.1.0.dist-info"

-> FlexSea version on "Exoboot\_dev" and in "RESEARCH" are DIFFERENT VERSIONS! (important?)

-flexsea version on Rpi is 11.0.6

PROMPT (in agent mode in CoPilot - Claude Sonnet 3.7) \_ \_ \_ \_ \_

"I need help writing code for my experiment. My experiment involves testing human perception of the rise and fall time parameters in a powered ankle exoskeleton. This experiment is based heavily off of other research conducted by 'Peng'. Peng looked at human perception into changes into actuation timing in the torque profile, his paper (which I am following closely and just changing the parameters) can be viewed in Peng\_research(actuation\_timing\_perception).pdf. Both Peng and I are using the Dephy Exoboot powered ankle exoskeleton. The user guides and resources for the Dephy ExoBoot can be viewed in the 'User\_Guides' folder. The controller that Peng uses for his experiment/research paper can be viewed in the 'Peng\_controller(actuation\_timing\_perception)' folder. As I stated before the protocol for Peng's experiment and mine are very similar, I am just changing the parameters in question. The protocol for my specific experiment can be seen in experiment\_protocol.docx attachment. However there is an issue, the Dephy ExoBoot is controlled in python via the FlexSEA python package and Peng's code used an old version of FlexSEA to run his experiment and that version of FlexSea is no longer supported. The version of FlexSEA that can be seen on the folder 'Actuator-Package-develop' (this is a clone of the Github repo for FlexSea, created by Dephy). I need to rewrite his controller codebase using the NEW version of FlexSea (and I also want to create a simple GUI to record the participant responses and change the torque profile rise and fall time parameters accordingly, not with an android application like Peng did). Additional resources can be seen in EXAMPLE\_actpack\_trial.py where this is preliminary code I have written for the Dephy Actpack, which is the actuator portion of the ExoBoot- just to play around with (Keep in mind that this code does not work at all for the ExoBoot). Another resource is EXAMPLE\_torque\_profile\_graphV2.py where I have made a simple graph program to visualize the affects of the torque curve under different rise and fall time parameters. With this information and context, please develop me my controller codebase."

-maybe add context about how this will be ran on the RPi 4 (Ubuntu OS)

-maybe see how I can use Claude Opus 4.1 (Claude's new flagship model) in Copilot terminal

ERROR: Response hit its length limit

Ran again and it gave output of codebase!!!!

-installed dependencies from output code

Instructions and output code given- move code (not code base bc of OS .venv differences) to Rpi and TEST

~~~~~

Dr. Acosta says that Xiangyu uses pre compiled code (he had to recompile every time he changed the code in his controller

- Xiangyu Both used precompiled code and terminal argument for python for his controller

Looking into how to compile now:

-Relates to ActPack adapter, STM32, STM32 controller (per email)

- Email has resource "How to upload/change ExoBoot firmware"

Xiangyu uses OLD python API ["fx_plan_stack" API and "FlexSEA-Rigid" actuator package library] to control the ExoBoot (which is no longer supported) - I am using the FlexSEA actuator package (which is the newer supported version to control the ExoBoot

-In the README.md file in the Xiangyu Actuator-package folder it says that "The repo contains the FlexSEA-Rigid Actuator Package library and sample programs for C/C++, Python, MATLAB and SIMULINK. These scripts are accompanied by and are dependent upon C based libraries for both Windows (.dll) and Unix(.so)."

-In the README.md file for my Actuator-Package it does not say this - *so I might not have to compile* ---and on the [FlexSEA API reference for the Device class](#) it says "This class is essentially a Python wrapper around lower-level C/C++ code that has been pre-compiled into a shared library. These library files are stored on S3 and downloaded lazily. They are referenced by a semantic version string specified in the firmwareVersion constructor argument."

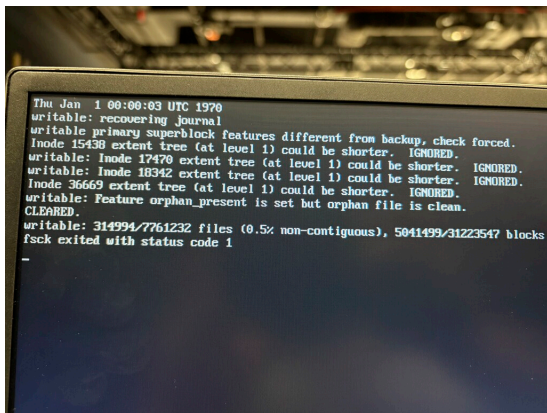
****HOWEVER**, in the [FlexSEA quickstart documentation](#) it states for the documentation for the Device State "side: Either "left" or "right", if applicable; None otherwise. Requires firmware >= v10.0.0." - it also states that "Firmware version 7.2.0 is the latest long-term support release. FASTER customers should not update beyond this version unless given explicit instruction to do so from Dephy."

- for the firmware on the Actpack controller (standalone actuator for the Actpack) the only firmware that worked (even out of all of the available listed ones) was 7.2.0

NEED TO CONNECT ExoBoot TO RPI AND SEE WHICH FIRMWARE VERSIONS IT GETS BACK

9/9

Moved equipment to lab, setting up Rpi



Issue w/ presumed micro SD card

Moved over to Rpi 5 from Rpi 4

Reformatted micro SD card

-Installed Ubuntu Desktop 25.04 on card (using Raspberry Pi Imager v1.6.9)

Ubuntu Setup:

Full Name: Max M

Username: auburn

Computer: rpi-lab-prod

Pass: Auburn123

.

Connected to wifi

Next Step: download vscode, log into box (via firefox) and bookmark, install flexsea per Dephy (Ubuntu)

- just download a compiler and then install via 'pip'

.

-Installed VS Code (logged into Github for CoPilot - and installed extensions)

-installed linux usb libraries (sudo apt-get install libusb-1.0-0-dev) [?]

- adding current user to dialout group (sudo usermod -a -G dialout auburn) → “auburn” is username - machine rebooted

- (/dev/ttyACM*) ← should return an error until Exoboot is plugged in

```
sudo apt update && sudo apt upgrade
sudo apt install libusb-1.0-0 libusb-1.0-0-dev
sudo usermod -a -G dialout "$USER"
```

.

“Exoboot_dev” folder (on PC) uploaded to BOX folder “Exoboot_dev_COMBINED”

-uploaded everything but .venv as this will change based on the OS

-“Exoboot_dev_Rpi” folder created on Rpi

-“Exoboot_dev_COMBINED” copied over from BOX to “Exoboot_dev_Rpi” (9/11)

.

Because my Ubuntu version is weird about installing Python 3.11 via “sudo apt install ...” I had to build it from source following Dephys steps (below)

```
1  # Python dependencies
2  sudo apt install build-essential gdb lcov pkg-config \
3      libbz2-dev libffi-dev libgdbm-dev libgdbm-compat-dev liblzma-dev \
4      libncurses5-dev libreadline6-dev libsqlite3-dev libssl-dev \
5      lzma lzma-dev tk-dev uuid-dev zlib1g-dev
6
7  # Clone the python repo
8  git clone --depth 1 --branch v3.11.3 https://github.com/python/cpython.git
```

Need to install git via (sudo apt install git) in order to clone repo [line 8]

.

see if I can make .venv through vs code after that and then see if installing flexsea (via pip) also works (easiest option)

Python 3.11.3 installed via source (looks good)

Made .venv from VSCode popup (had issues - no pip) - installed pip (sudo apt install python3 pip)

STILL WORKING ON INSTALLING FLEXSEA

-tried to do some git stuff through copilot and it fucked up

=====

Went off the rails on agent mode and ended up making an entirely new codebase in the “Exoboot_dev” desktop folder - entirely new python codebase with all of the bells whistles for Github. Tons of professional files and formatting that I do not need for this experiment BUT IT MIGHT WORK)still need to test

Found out that I can just make a github repo and drag and drop files into it like a shared drive.....(this may be a better solution for moving files & I also have my (Exoboot_dev_COMBINED) OG controller & experiment saved to BOX (thank god) - so I can revert back from versions as Git is confusing and I already 'corrupted' my old controller that I was familiar with

.

TODO: follow steps on Exoboot_dev GitHub on Rpi and see what happens

-if fucked make decision to trace steps back to OG controller (on BOX) or copilot my way through changes

===== 9/16

https://github.com/MaxM-1/Exoboot_dev_hub

.

cloned repo onto Rpi

- ran into issue with deadsnakes/ppa when running install_linux.sh - fixed - issue with .venv - working on now - installed requirements from requirements.txt in folder
- 1)Removed problematic deadsnakes PPA - This repository didn't support Ubuntu 25 (plucky)
- 2)Updated package lists - All repositories now work correctly
- 3)Configured Python environment - Created a virtual environment with Python 3.13.3
- 4)Installed all dependencies - FlexSEA package, project requirements, and the exoboot-perception package



What's Now Available:

- Virtual environment at .venv with Python 3.13.3
- FlexSEA package installed in editable mode
- Project dependencies from requirements.txt
- Exoboot perception experiment package installed
- Data directories created: data, results, settings, logs

.

flexsea still not installed in .venv

- installed via pip on .venv [flexsea version 12.0.4] - (restart vscode, error highlighting removed)

=====

9/18

turned on batteries - ensured full charge-put then into exoboos - put on exoboos- connected to Rpi with usb cable

-issue with getting firmware version (no such file exists where it is trying to pull from - could be bc old vibe code)

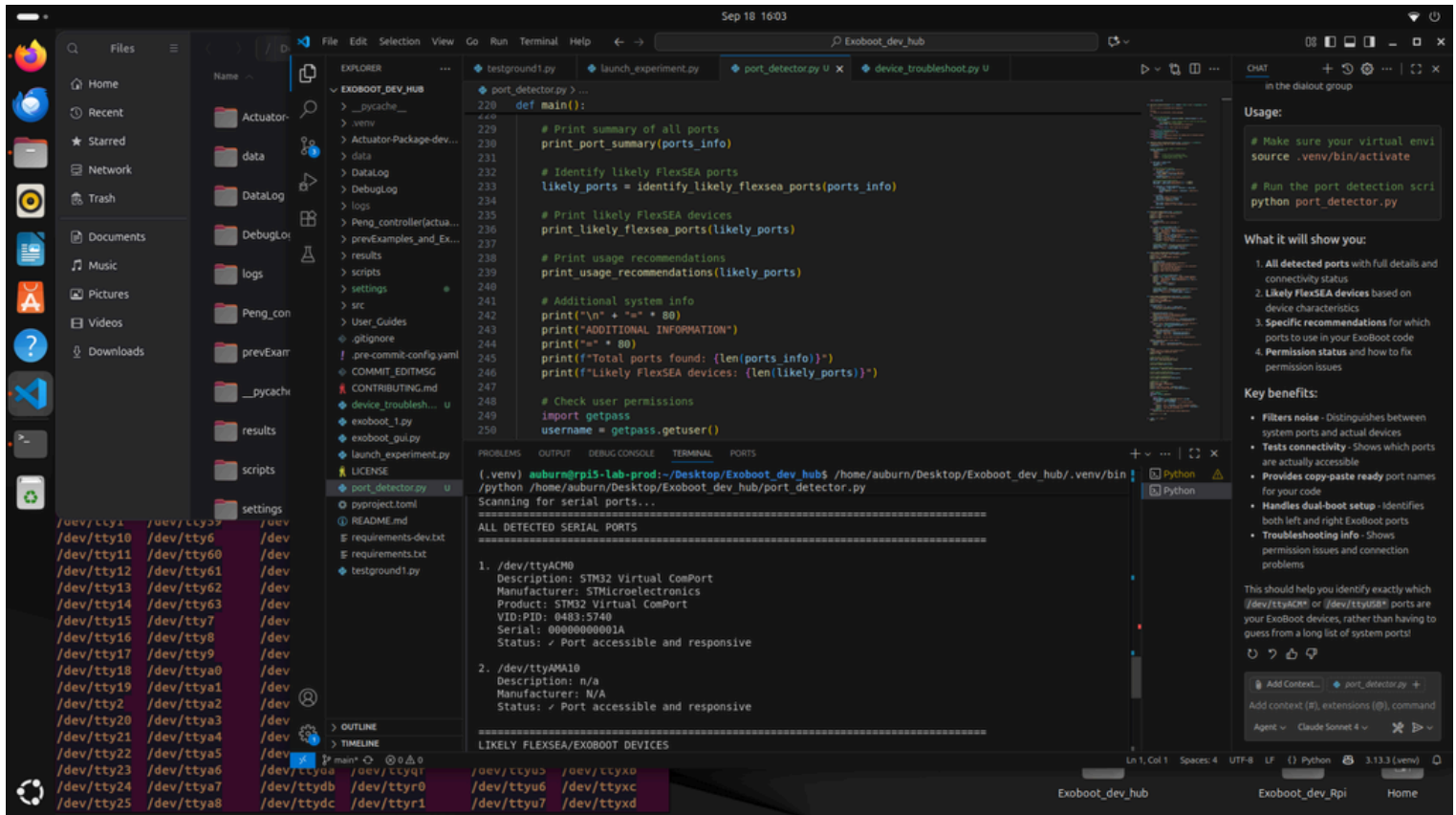
--made new file for troubleshooting and it seemed to work - ran it and used 7.2.0 as input and code says it was connected - after that then i ran launch_experiment again and only /dev/tty0 showed up (see below)

Issue with ports - only showing 1 in gui - crashed when I connected both to same port from menu

-ran ls /dev/tty* and it gave me ALL of the /dev/tty[...] ports

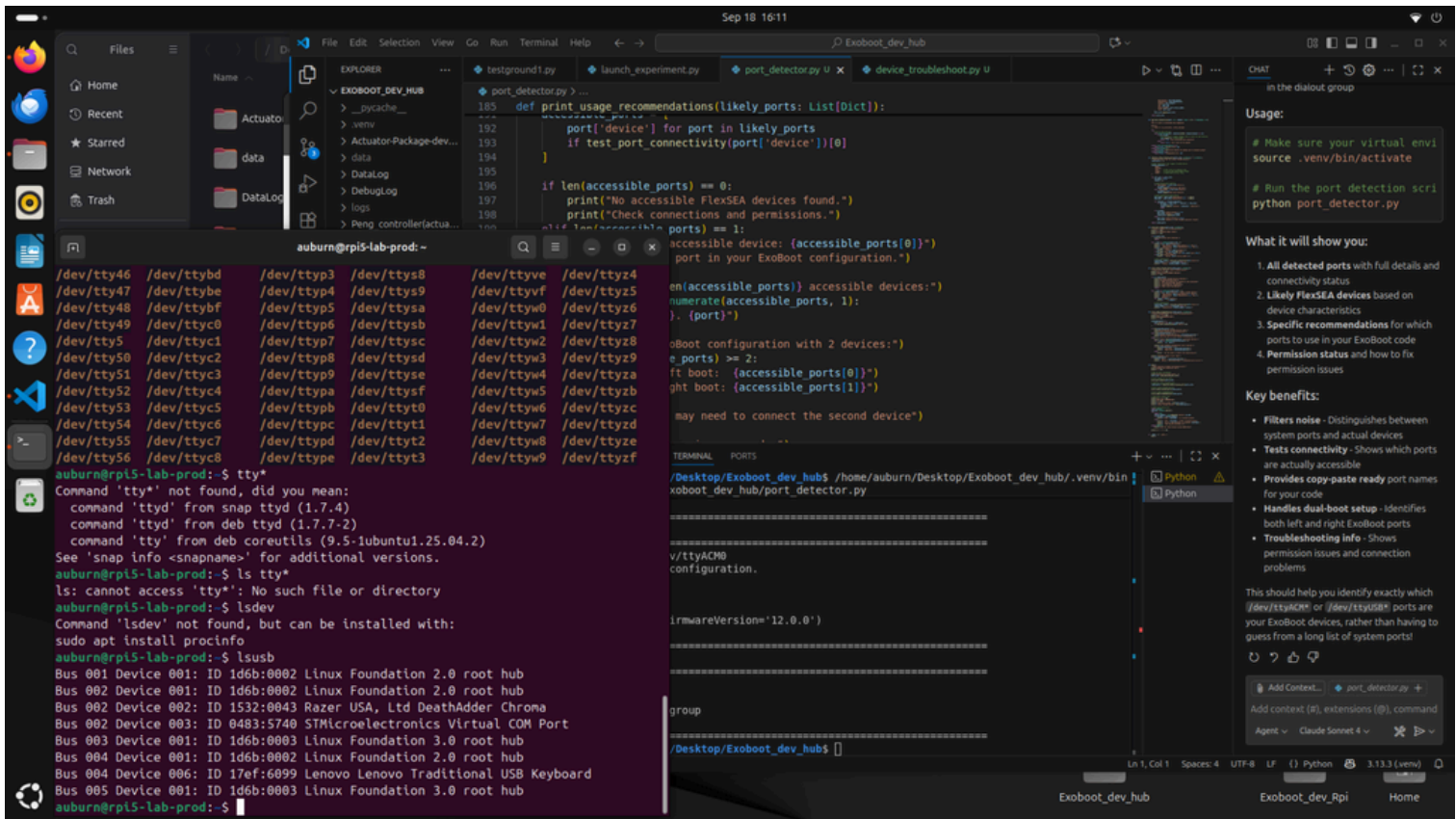
- made new code to troubleshoot ports "port_detector.py"

only seems to be detecting /dev/ttyACM0



detects 2 ports but only seems to be detecting 1 likely Dephy device port /dev/ttyACM0

Ran command in terminal "lsusb" - only showing one "STMicroelectronics Vrtual COM Port - I have a Lenovo keyboard connected and a Razer mouse also connected

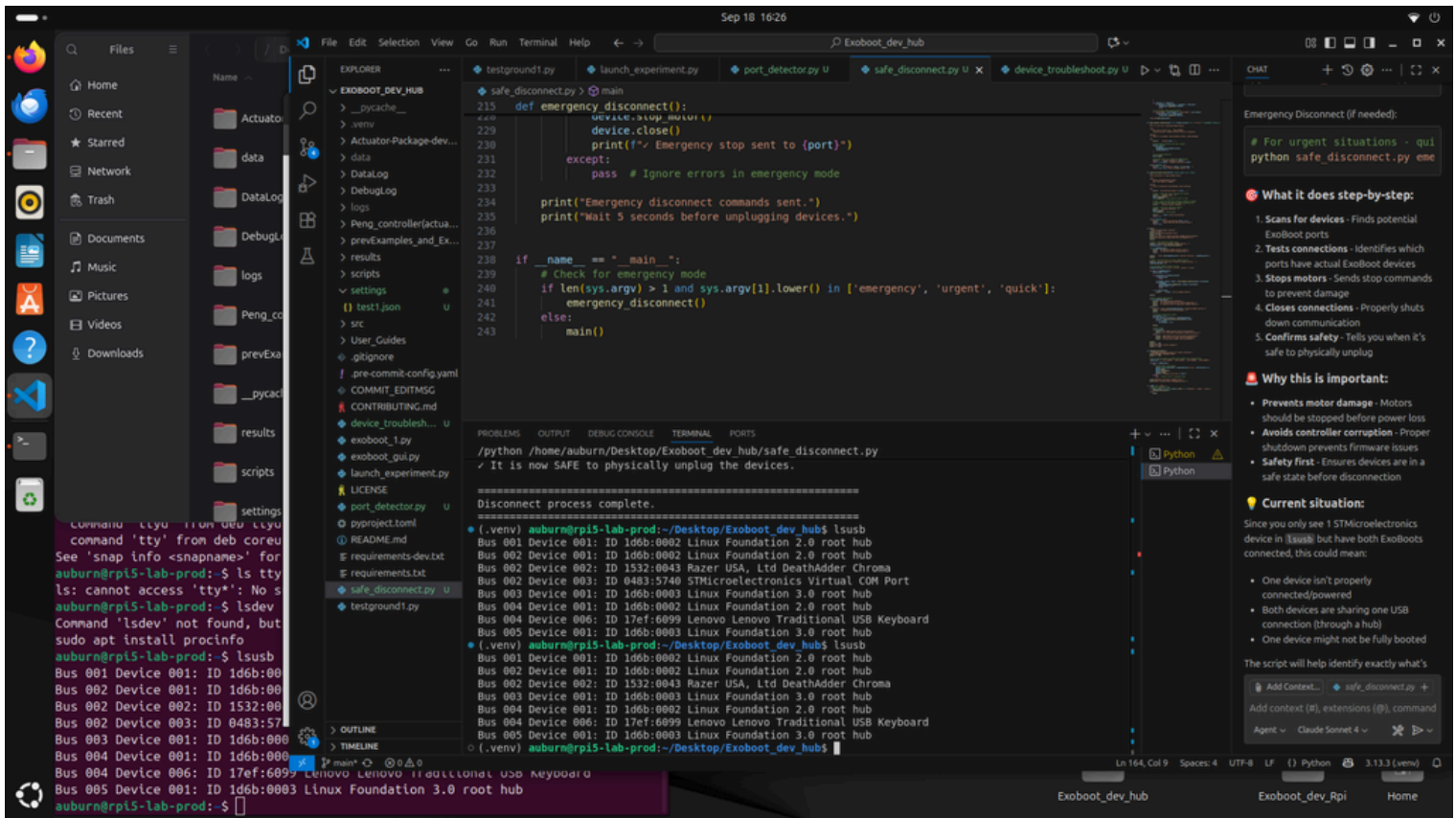


going to try to disconnect and reconnect - had CHET make me code file for disconnect - tried multiple firmware versions -

Try multiple common firmware versions

```
firmware_versions = ["12.0.0", "11.0.0", "10.0.0", "9.0.0"]
```

7.2.0 is what was used to connect not any of these and the code ran - says it is now safe to disconnect



before and after unplugging exoboosts

LETS REDO THIS

- plugged right exoboot into BOTTOM blue USB port on rpi5 → ran 'lsusb' and STMMicroelectronics COM port' showed up in Bus 002 Device 004
 - (.venv) auburn@rpi5-lab-prod:~/Desktop/Exoboot_dev_hub\$ lsusb

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 002: ID 1532:0043 Razer USA, Ltd DeathAdder Chroma

Bus 002 Device 004: ID 0483:5740 STMMicroelectronics Virtual COM Port

Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 004 Device 006: ID 17ef:6099 Lenovo Lenovo Traditional USB Keyboard

Bus 005 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
- plugged left one into TOP blue USB port on rpi5 → ran 'lsusb' and Nothing seemed to change
 - (.venv) auburn@rpi5-lab-prod:~/Desktop/Exoboot_dev_hub\$ lsusb

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 002: ID 1532:0043 Razer USA, Ltd DeathAdder Chroma

Bus 002 Device 004: ID 0483:5740 STMMicroelectronics Virtual COM Port

Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 004 Device 006: ID 17ef:6099 Lenovo Lenovo Traditional USB Keyboard

Bus 005 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

- nothing seemed to change

Testing out ports and boots individually

- Left plugged into TOP alone - NOTHING - Unplugged
- Left plugged into BOTTOM alone - NOTHING ?!?!?!?! - Unplugged
- Right plugged into BOTTOM alone - RECOGNIZED - Unplugged
- Right plugged into TOP alone - RECOGNIZED - Unplugged

-switching USB cabled from Right to Left exo [Both Exo's remain on during this] - cables switched

- Left (w/ switched cable) plugged into BOTTOM - NOTHING - Unplugged
- Left (w/ switched cable) plugged into TOP - NOTHING - Unplugged
- Right (w/ switched cable) plugged into BOTTOM - RECOGNIZED - Unplugged
- Right (w/ switched cable) plugged into TOP - RECOGNIZED - Unplugged

→ thus issue seems to be with the Left Exoboot ONLY

Turned BOTH exo's off - waited a few mins - Turned BOTH exo's on

-left plugged into bottom (first) - RECOGNIZED

-right plugged into top (second) - RECOGNIZED

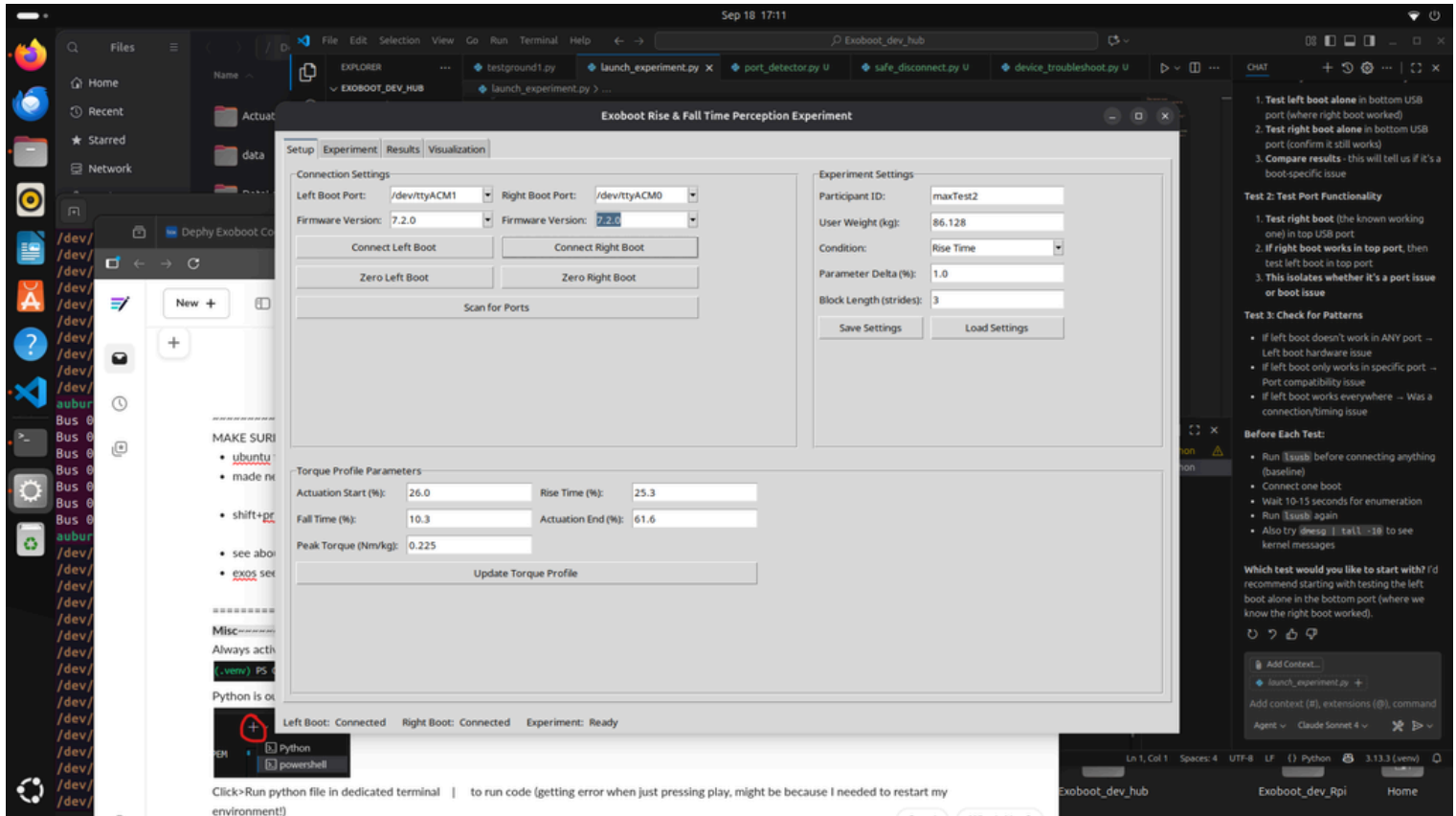
COOKING AND IN BUSINESS - running launch_experiment

The screenshot shows a Linux terminal window with the 'Exoboot Rise & Fall Time Perception Experiment' application running. The application has a graphical interface with several sections:

- Connection Settings:** Left Boot Port: /dev/ttyACM1, Right Boot Port: /dev/ttyACM0, Firmware Version: 10.1.2. Buttons for 'Connect Left Boot', 'Connect Right Boot', 'Zero Left Boot', 'Zero Right Boot', and 'Scan for Ports' are visible.
- Experiment Settings:** Participant ID: P01, User Weight (kg): 70, Condition: Rise Time, Parameter Delta (%): 2.0, Block Length (strides): 3. Buttons for 'Save Settings' and 'Load Settings' are visible.
- Torque Profile Parameters:** Actuation Start (%): 26.0, Rise Time (%): 25.3, Fall Time (%): 10.3, Actuation End (%): 61.6, Peak Torque (Nm/kg): 0.225. A button for 'Update Torque Profile' is visible.

The terminal output shows the results of the experiment, including device IDs and connection status. The status bar at the bottom indicates 'Left Boot: Not Connected', 'Right Boot: Not Connected', and 'Experiment: Ready'.

connected boots



zeroed both boots

adjusted and saved settings

- put in weight (86.1ishKgs)

torque profile kept with default and ready to go

RUNNING EXPERIMENT TRIAL 2 | I am simulating walking in place

Live notices:

- Earlier, Same and Later buttons are weird - sometimes update - cant really tell - cant really tell a noticeable difference
- no live data

Experiment stopped notices:

- - cant scroll up any farther but gives this error

- - scrolled down and was given this error
- .

RESULTS TAB

- 'saved results' - said it was saving to a .csv
- 'export to CSV' - said it was exported to a csv
- .

VISUALIZATION TAB

- 'generate plot' - gave output below

and this is what it gave in the terminal:

misc:

- 'data' file is empty
- 'DataLog' is not -

-
- has regular Dephy output for device

in folder settings

EXITED OUT OF TK WINDOW:

- says that the left and right boot have disconnected successfully
- when I try to run testground1.py (print hello) it does not show up in the terminal

~ ~ ~ ~ ~

json files in results

Uploading Some Files (that might be under gitignore - just being safe) To Box (ones that were 'updated today')

.

ran experiment at 5:20ish

.

vscode recognized changes (probably stuff that is not under gitignore)

Pushed to main via just clicking in vscode - once i did this they saved to the file folder on desktop

~~~~~

MAKE SURE TO SAVE FILES LOCALLY ON RPI - autosave is off

- ubuntu file explorer - look into how to show date time
- made new file called pychache
- shift+prntscrn in ubuntu takes screenshot and copies to clipboard
- see about downloading the "read the docs from Dephy and putting into resources in codebase folder"
- exos seem to tighten when first turned on
- Upload Dephy userGuides (and others??) to that google AI notebook thing

=====

9/23

Note: from savannah - the exoboot data output makes a new file roughly every 10 mins

From Dr. Acosta Meeting:

Boot not adjusting (MOVING):

Get recording of the what the boot is doing - ask savannah as she has code to get torque angle relationship (torque and ankle angle - savannah has code for it) - see what is happening with that data and see if the boot is running savannah code (could be firmware issue)

→ Got Matlab code from Savannah - uploaded to box

Creating new file that does the same thing only in Python (also installed matplotlib, pandas, etc for graphing)

Ran file and it plotted a BLANK plot - looked at Excel file and the ank torque column was 0 for all values...

.

also just looked at safe\_disconnect file and it uses out of data firmware 12.0.0 - need to change

MAYBE make a doc that contains a lean version of this document and add it to the code base to be used as a resource

TODO: make code that just manually moves the boot to see if commands are correct

-made basic file to check connectivity

~~~~~

Put on right exo and turned on

-right when I turned it on I felt it tighten

Ran 'basic_exoboot_connect.py' (had to fix right vs left side code problem) and code gave me error:

" Error reading data from Right Exoboot: 'dict' object has no attribute 'accelx' "

The error indicates that the data object returned by self.device.read() is a dictionary, not an object with attributes. Let me check the current read_data method in the ExoBootController and fix it:

Read basic_exoboot_connect.py

Now let me look at the read_data method in the ExoBootController class to fix the dictionary access issue:

Searched text for def read_data (**/exoboot_1.py), 2 results

Read exoboot_1.py, lines 227 to 267

I can see the issue! The self.device.read() is returning a dictionary, but the code is trying to access it as if it has attributes (like data.accelx). I need to change it to use dictionary key access instead. Let me fix this:

exoboot_1.py+10-10

Perfect! I've fixed the issue. The problem was that `self.device.read()` returns a dictionary with keys like 'accelx', 'gyroz', etc., but the code was trying to access them as attributes (like `data.accelx`).

The fix changes all the data access from:

`data.accelx` → `data['accelx']`

`data.gyroz` → `data['gyroz']`

`data.ank_ang` → `data['ank_ang']`

etc.

Now when you run your basic connection test, it should properly read the data from the ExoBoot without the 'dict' object has no attribute 'accelx' error. The script should now successfully:

Connect to the Right ExoBoot (as you configured)

Read sensor data properly

Display the ankle angle, gyro data, and motor current values

Disconnect cleanly

Try running your `basic_exoboot_connect.py` script again and it should work properly now!

.Trying to fix -below is proposed change to `exoboot_1` that I accepted

Update IMU data

`self.current_time = time.time() * 1000 # ms`

`self.accelx = data['accelx']`

`self.accely = data['accely']`

`self.accelz = data['accelz']`

`self.gyrox = data['gyrox']`

`self.gyroy = data['gyroy']`

`self.gyroz = data['gyroz']`

Update ankle and motor data

`self.ankle_angle = data['ank_ang']`

`self.motor_angle = data['mot_ang']`

`self.ankle_velocity = data['ank_vel']`

`self.motor_current = data['mot_cur']`

.

Now it is streaming data in the terminal!! (shown below)

Time: 1758667222090ms, Ankle Angle: 5054.0, Gyro Z: -100.00, Motor Current: 1072mA

Time: 1758667222191ms, Ankle Angle: 5399.0, Gyro Z: 119.00, Motor Current: 530mA

Right Heel Strike Detected! Num: 7, Expected Duration: 236 ms

-time for some more in depth experiments

.

trial exp 1:

I stomped my foot 3 times (hard) but the data stream said that it detected 7 heelstrikes

-when I was careful to not move my foot it detected no heel strike (maybe because it is detecting gyroZ movement and not actual force)

.

it is also logging data even though should_log is false (for exp. above datalog file is ...17h43m...)

- looked at file and looks populated HOWEVER ank_torque is not populating

.

trial exp 2: starting at max plantar flexion and moving from max. plantarflexion to max. dorsiflexion 3 times ending back to max plantar flexion

(...17h53m....) - ROUGH visual inspection of .csv and it looks accurate (ish) to what I did

prompted to create new file that mapped ankle angle (used Matlab code as guide) - result is below

-used matlab mapping factor

.

prompted to fix - says the issue is likely the matlab mapping factor - and made other changes

You're absolutely right! That flat line at ~12.5 degrees doesn't look like ankle movement at all. The issue is likely with the mapping factor conversion from the MATLAB code. Let me create a new version that just plots the raw ankle angle data without any MATLAB-specific conversions: ankle_angle_plotter.py+33-34

Perfect! I've completely rewritten the script to avoid any MATLAB-specific conversions. Here's what the new version does:

Key Changes:

1)No MATLAB conversions: Uses raw ankle angle data directly from the CSV

2)Two plots:

2.1)Top plot: Raw ankle angle data in encoder units

2.2)Bottom plot: Movement relative to mean position (easier to see ankle motion)

3)Minimal filtering: Only removes zeros and extreme outliers

4)Better statistics: Shows raw data range, mean, and variation

What You'll See Now:

Raw encoder values: The actual values from the ankle angle sensor

Movement patterns: The bottom plot will clearly show when you moved your ankle up and down

Real variation: If you were moving your ankle to max/min positions, you should see clear changes in the data

.

.

this is output now (for exp 2 plantar - dorsi flexion file trial)

Graphed datalog file (ank_ang) in libreoffice and this was the output:

LOOKS GOOD - meaning it is just the python file that graphs it that is wrong!

Prompted to fix:

"What I Fixed:

1Removed all filtering: LibreOffice plots ALL the data points, so now Python does too

2Single plot: Simplified to one clean plot that matches LibreOffice layout

3Thicker line: Used linewidth=2 to make it more visible like LibreOffice

4Proper Y-axis scale: Set the range to 0-9000 to match what you see in LibreOffice

5Cleaner formatting: Matches the LibreOffice appearance

Why It Was Different Before:

- The original code was filtering out data points and using very thin lines
- It was also creating multiple subplots which made it look different
- The filtering was removing some of the smooth data that LibreOffice was showing

Now when you run the script, you should see:

- Clear, smooth ankle flexion cycles (like in LibreOffice)
- Data ranging from ~2000 to ~8000 (matching your LibreOffice plot)
- The same clean, readable format that shows your ankle movements clearly"

. This is the output now - LOOKS GOOD! (ANKLE ANGLE OVER TIME file IS VALIDATED!!!)

Prompted to take lessons learned from this and apply it anywhere else in the codebase

Summary of Changes Made:

✓ torque_angle_analyzer.py - MAJOR FIX

Removed all MATLAB conversions (mapping factor, unit conversions)

Now uses raw data: Raw torque (mNm) vs Raw ankle angle (encoder units)

Removed excessive filtering - plots all data points like the successful ankle plotter

Updated labels and statistics to reflect raw units

✓ exoboot_1.py - ALREADY FIXED

Fixed the dictionary access issue (data['accelx'] instead of data.accelx)

✓ Other Python files - CHECKED

Verified other analysis files are already using raw data correctly

No additional fixes needed

✓ Documentation - CREATED

Created DATA_ANALYSIS_FIXES.md to document the key lessons learned

Provides guidance for future data analysis work

Key Principle Applied:

Use RAW sensor data directly - The same approach that made your ankle angle plot perfect:

No MATLAB-specific conversions

No excessive filtering

Plot all data points

Use raw sensor units

Now when you run torque_angle_analyzer.py, it should show:

Raw torque values in mNm (instead of converted Nm)

Raw ankle angle in encoder units (instead of converted degrees)

All data points without filtering (like LibreOffice shows)

Realistic patterns that match your actual sensor data

.CHANGES ACCEPTED - made new .md file with changes, lessons learned, useful in future

DELETED Savannahs Matlab script (only caused issues obviously)

.....

Ran torque_angle_analyzer.py again and it still shows just a flat line - ank_torque (or whatever) is giving back 0 in output

NEXT TO LOOK INTO

→ → changes pushed to github (commit name is ' . ') & DataLog .csv files backedup to BOX

=====

MAKE CHANGES in RED & generate more docs of resources[as talked about](put in Github) and then write prompt to fill update copilot on on Thursday

=====

~~~~~

.noticed that in Xiangyus calibration that his 'right boot cal .excel' has different columns than mine

~~~~~=====FOR SURE TODO~~~~~=====

.

lookup what each value in datalog output .csv file means (unit) and compile that somewhere

-maybe adjust ankle_angle_plotter.py to plot any column value vs time (1column) given a column name and a unit as an input (will have to get dephy units to keep track of)

-make doc combining savannas exp guide and dephy batery doc to create process for getting exo and batteries setup

=====

Misc~~~~~

Always activate virtual environment and make sure that I am working out of it:

shift+prntscrn in ubuntu takes screenshot and copies to clipboard

Python is output of code | powershell is how to manipulate .venv folder | press "+" if needed

Click>Run python file in dedicated terminal | to run code (getting error when just pressing play, might be because I needed to restart my environment!)