

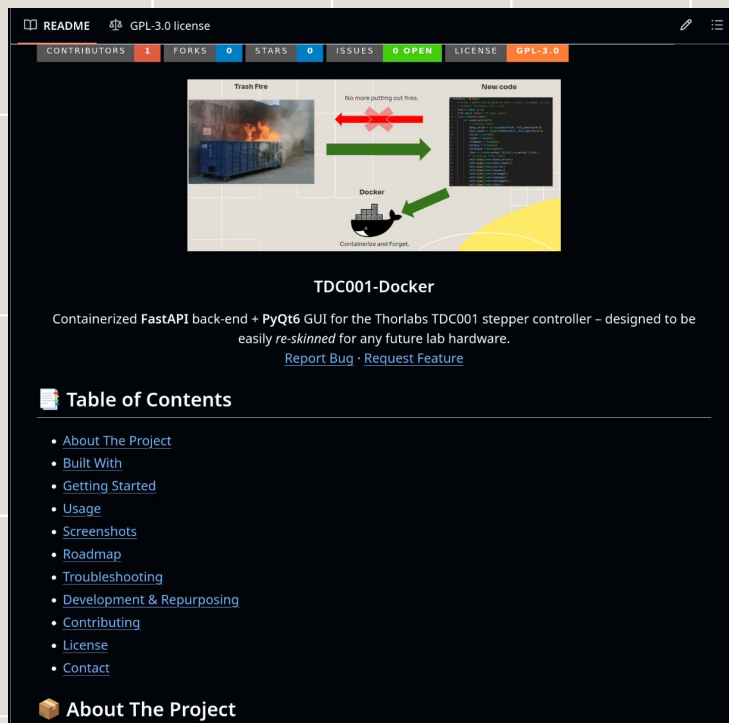


Docker Updates

By Jacob Lazarchik, Made for July 16th 2025

Major Stable Release - TDC001 Docker

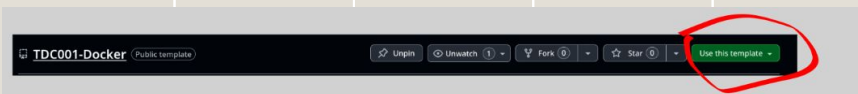
Front Page of the Project on Github



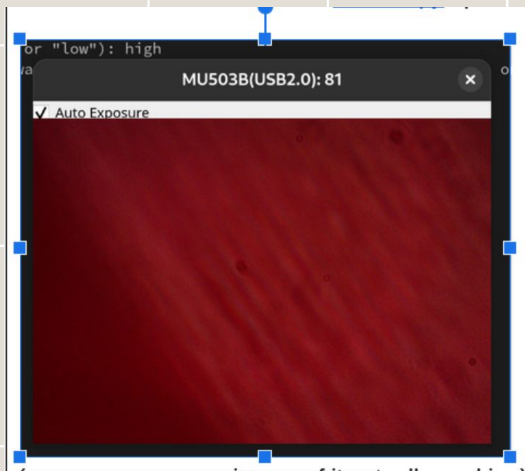
- The shutdown, save, load, and startup scripts for the backend and frontend images have been completed.
- Fully updated readme front page for the project is complete
- Features a fully working pyqt frontend and fastapi backend
- As it stands, it is *mostly* stable release code and able to be copied to many future projects.
- Fully cross platform as well (scripts for all 3 major OSes)
- <https://github.com/JacobGitz/TDC001-Docker>

Beginning of Amscope-Docker

Made the new Repo from Template



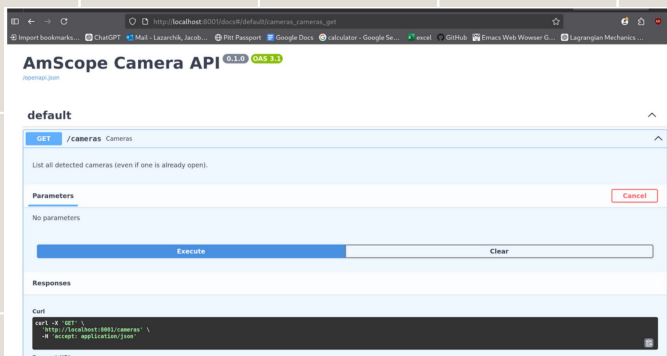
Kai's Original Code Running in Linux



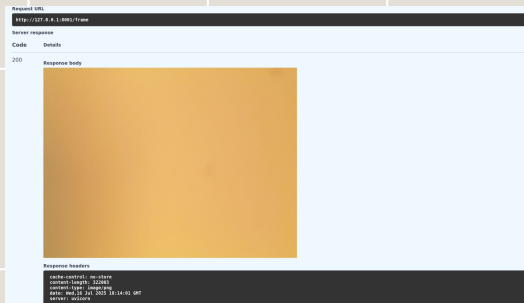
- Kai's original amscope repo was over-connected—too many frontend↔backend links in `qt.py`.
- Porting it to Linux was painful, but I got it running, then I scrapped it.
- In ~1 hour I wrote a lean FastAPI backend, squashed a few bugs, and it ran smoothly.
- Wrapped it in a container; backend started up and broadcasted without a hitch.
- **This exposed fundamental architectural flaws in my TDC001-Docker approach, which I now need to redesign for future projects.**
- <https://github.com/JacobGitz/Amscope-Docker/tree/dev>

The Fundamental Problem - Parallel

Amscope Camera Fastapi Docs Page (Dockerized)



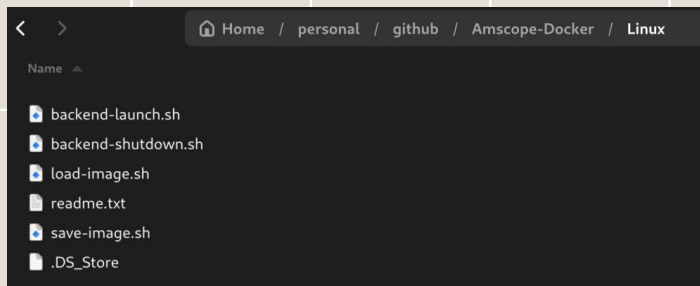
/Frame (getting a frame from a camera successfully)



- The TDC001 setup lets several stepper controllers attach to one computer, **but each Docker container can command only one controller at a time.**
- To stream images **from two cameras simultaneously, you must spin up two separate Docker containers.**
- Also have to **assign a unique port to each** container to avoid network clashes.
- **Bind each container to a specific AmScope camera** (rather than “any available” camera on a desktop) so experiments always know which camera is controlled by **which** container.

Possible Solution

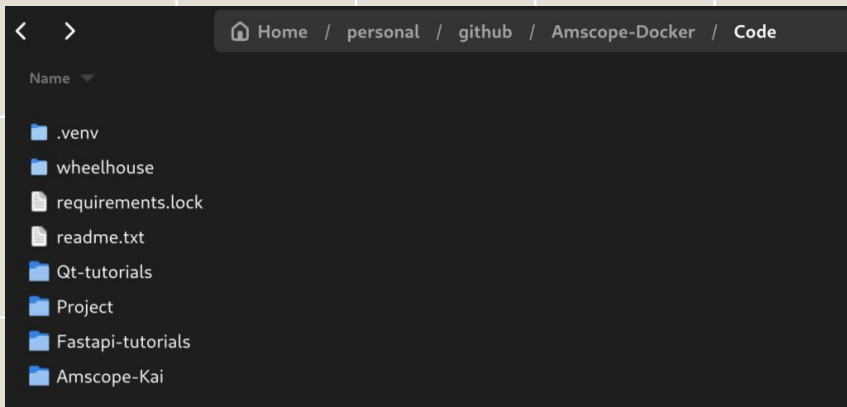
The /Linux Helper Script Directory



- Comment out the /Connect and /Disconnect endpoints (allowing one container to connect to many devices)
- Use Pyvisa (or something else) to identify the unique serial identifier for each device on the system
- Have the container automatically bind to the particular camera we hard-code
- Do this for four or so docker images, one for each camera in the lab
- Ensure each docker image has a unique port and name:tag combination to distinguish the cameras
- **Make this process into some sort of setup script to streamline it**

Other Notable Bugfix - Directories

/Code Directory



- Originally, the “Project” directory, which I renamed, was called “TDC001-project”
- This presented a big problem when moving over to the new Amscope project, as it was confusing. I decided to name it to “Project” instead
- However, upon doing this, all helper scripts broke in /Linux, /Windows, and /MacOS!
- **Possibly have docker-compose files be auto detected or just keep the original name** (I really don't want to)
- Fixing the hardcoded file path issue will be a likely focus that **I will merge into the original TDC001-Docker template, but all helper scripts need fixed**

Remaining Roadmap

Serial Identification

- Make functions that return the unique serial identifiers for cameras
- Use this to hard code what device the container initializes
- Ensure the API tells the user if the required device is connected or not

Fix Helpers

- fix the issue where changing the name of a directory or moving the docker compose files breaks all setup scripts
- Ensure all directories are generalized in name

Setup Script

- make a modified build+save script
- automatically changes the port, image name, hard coded serial device, and anything else easily and in a reusable way.
- Think hard about this

Main Page

- Rewrite my entire landing page for github for this project, ensuring it matches what is in this repo

Major Release

- Once it seems somewhat ready, I will push everything to my main branch, and amscope docker will be complete

Move to Original

- all useful code will be moved to the dev branch of my original TDC001 repo
- This will be the staging for the next major release, which I will have to change the documentation and reflect these changes most likely