Return to: README - PlanktoScope Software Task Force
All dates in this document are listed in the UTC time zone.
The recurring meeting link is meet.google.com/vyj-ouqv-vdg

2024-06-06

For meetings starting in June 2024, please go to 2024-05 to 2024-08 Software Meetings instead.

2024-05-30

In attendance: Ethan, Oumayma, Melissa, Satoshi, Morgan, Adam

Work package updates + discussion

Refer to: Software Work Packages

- Hardware Drivers (Ethan):
  - Updates from Oumayma: made some changes on the two active PRs device-backend#35 and device-backend#36
  - Ethan: next work after this PR will be to switch from RPi.GPIO library (including in the shush package, if we can't remove the shush package) to a different GPIO library for RPi5 compatibility
- Backend APIs (Ethan):
  - Updates from Melissa
  - TODO for Ethan: review PR 402
  - M: will be focusing on putting together a mid-term report, while E reviews the PR; no work on the MQTT docs planned this week otherwise
- OS (Ethan):
  - (no updates)
- Docs (Ethan):
  - (no updates)
- Release Engineering (Ethan):
  - Using the 64-bit OS for official builds (tracked in PR 416): I did some basic testing, but I still need to do a bit more testing before I merge this PR as part of v2024.0.0-beta.0
  - The other blocker for v2024.0.0-beta.0 is to change the ISO button group to a slider with a step size of 50. This is a very small change, I've just been occupied with other things.

    - PR 416: I figured out how to automatically run our OS setup scripts on GitHub Actions on each commit (working builds for a subset of the setup scripts are at https://github.com/ethanjli/rpi-forklift-demo); once I finish implementing GItHub Actions builds for PlanktoScope OS, it'll resolve issue #42
        - This makes it faster & much easier for me to build SD card images for development, testing, and releases. This also makes it feasible for me to also provide OS builds with the graphical desktop, for use by developers.
        - As a byproduct, I've now also created some GitHub Actions to make it easier to build RPi SD card images on GitHub (ethanjli/cached-download-action, ethanjli/pigrow-action, ethanjli/pinspawn-action, ethanjli/piqemu-action, ethanjli/pishrink-action) which enable other people to build custom SD card images on our SD card images. These actions also enable us to automatically run some "smoke tests" to ensure that the system boots correctly, the programs all start correctly, etc., so that we don't have to think about that when we're doing our own manual testing.
        - My next steps are to fix the remaining build errors, make downloading of Docker containers much faster (by doing it outside the VM), and to make the GitHub Actions workflow add version information which is added by the install.planktoscope.community/distro.sh script but is not yet added in GitHub Actions.
- GUI (Loïc):
    - A: T & L have been working on a mockup of the new GUI, it looks impressive. They figured out how to incorporate some Vue stuff into Node-RED - it was very powerful, e.g. checklists, sliders, etc. It seemed to make many things much more possible. Also d3.js integration is possible this way. There's a good plan for the next iteration of the dashboard. Having T & L together for 3 days has been great for progress.
    - E: I'm really excited to hear that we're bringing regular Javascript files (via Vue) into the project
- Any other work packages with updates, questions, or action items to discuss?

Other updates + discussion

- Any other topics to discuss?
    - S: just a new version, PlanktoScope-Ver2.1JP2, and sent some to colleagues around Japan. Just started to test it.

2024-05-23

In attendance: Loïc, Melissa, Oumayma, Ethan, Adam, Thibaut
(this meeting will be limited to a duration of 90 minutes)

Work package updates + discussion

Refer to: Software Work Packages

- GUI (Loïc):
    - Update of the Preview page
        - (discussion about the two layout prototypes)
        - E: seems like there are some limitations in Node-RED which will make mobile-responsive layout tricky to implement…
        - T: it might be okay to force phones to use landscape mode (e.g. control panel on the right), since phones are a low priority for us
        - T: let's go with the horizontal-friendly layout
        - (discussion about adjustments to the panels on the first layout, including adding a volume setting under the pump adjustment panel)
        - (discussion about supporting preset values)
    - Next step: work with Thibaut on the frontend
    - Won't be able to attend next week's meeting.
- Backend APIs (Ethan):
    - Updates from Melissa
        - (review of MQTT docs changes)
- Hardware Drivers (Ethan):
    - Updates from Oumayma
        - Working on PRs on GitHub
        - Had problems with the autoformatter, resolved some errors. Still have errors with the type-checker
        - PR 36:
            - (discussed type-checking complaints; TODO for Ethan: resolve those complaints)
            - (discussed motor-releasing behavior at shutdown, when the Python hardware controller quits and thus makes the focus process and stepper process quit)
            - Next steps: update main.py for the two processes; and then have Ethan do a PR review/test
        - PR 35 was on hold (needed input about whether the changes affect electronics boards, e.g. longevity of the LED); next step is to test changes and delete remaining unnecessary methods

- Docs (Ethan):
    - PR 415: review is needed; will follow up separately
- OS (Ethan):
    - (no updates)
- Release Engineering (Ethan):
    - (no updates)
    - Next step: make at least one or two changes needed for the beta pre-release
- Any other work packages with updates, questions, or action items to discuss?

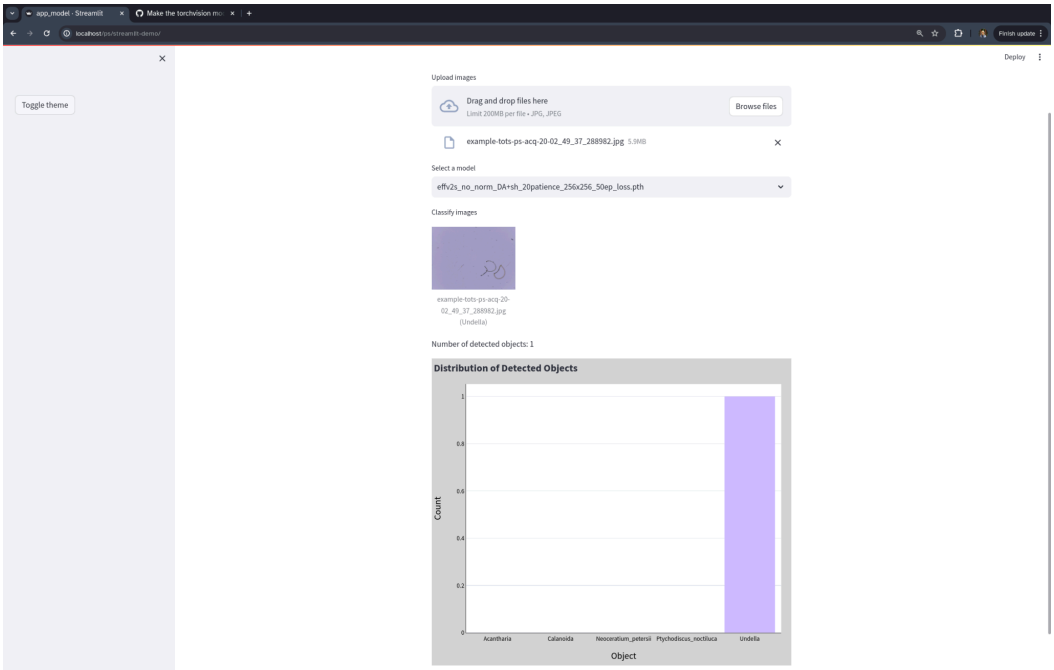
2024-05-16

In attendance: Thibaut, Loïc, Ethan, Oumayma, Melissa, Adam, Wassim, Satoshi

Work package updates + discussion

Refer to: Software Work Packages

- Targeted ML-based classification (Wassim):
    - Update from Ethan: I merged PR streamlit-classification-app#3 to make the demo work without errors. Here's what it looks like:



    - T: could the code underlying this be in a Jupyter notebook? Would it be easy for other people to modify? e.g. I'd like to see the classification scores, visualize different things, etc.

        - W: this demo app uses streamlit (Python framework for developing web applications/interfaces); basically you can use templates or components to add things.
    - W: mussel dataset work: have started experimenting with a specific species in the dataset, did some tests; results look promising. Still have to test other models, training strategies, different hyperparameters and data augmentation results.
        - W: there's a high imbalance between the three classes in the dataset. I started with a small portion of the dominant class to avoid dealing with the class imbalance at the start. I have a dataset subset with balanced classes. Started doing experiments on the dataset. Good evaluation scores so far. Will test other models, other training strategies; will test attention mechanisms to see what happens.
- Hardware Drivers (Ethan):
    - Updates from Oumayma
        - O: we abandoned the idea of modifying the code remotely. We've been working on the RPi. We had a problem with the command as we shared on Slack.
            - E: run this command:

```
Unset
wget -O - https://install.planktoscope.community/distro.sh | sh -s -- -v master -H planktoscopehat
```

            - E: try using this SD card image as the base: https://downloads.raspberrypi.com/raspios_oldstable_armhf/images/raspios_oldstable_armhf-2024-03-12/2024-03-12-raspios-bullseye-armhf.img.xz
        - PR 34
        - O: Created PR 35 & PR 36, haven't been able to test on the RPi yet.
- Backend APIs (Ethan):
    - Updates from Melissa
        - PR 29: I have updated the mqtt reconnection method but have not tested it.
        - Diagrams/PR 402
            - M: Updated the docs, but many tests don't work on GitHub
            - high-priority TODO for E: fix the GitHub Actions workflow on the PlanktoScope repo for running checks on pull requests made from branches in external repos (as opposed to the PlanktoScope/PlanktoScope repo)

        - M: worked with Jeremy on Monday to determine the range of values; tested different things on Node-RED dashboard, were able to determine ranges of values. Will need to be revised.
        - (feedback from E)
        - T: I would be interested to hear about what changes (if any) we should make to the design of the API
            - E: e.g. reorganize the structure of the topics into devices (light, pump, focus) which are self-contained - so that all topics related to (e.g.) the light are under light/, instead of having them be split between actuator/light and status/light
- Docs (Ethan):
    - Follow-up on a DM request I received from Thibaut to remove the FAQ's question about pricing: I temporarily removed that question pending a deeper discussion and decision-making process. Can we find a satisfactory way to restore the question and provide an answer which better communicates the value FairScope brings to justify its price vs. the cost of bare parts from the v2.1 BOM?
        - T: let's discuss this in another meeting.
        - T: I'd like to avoid situation where customers avoid being exposed to different prices. It's hard because it's linked to the cost of materials, but also the cost for manufacturing, access to manufacturing equipment.
        - T: also there's a question about the cost of the HAT for v2.5: do we use the price provided by FairScope for this, or some other price?
        - T: I think a higher priority for work than communicating the price in the FAQ is to improve the BOM for v2.5
        - TODO for E: make a PR which addresses these concerns and follow-up on Slack with Thibaut - no need to discuss in another meeting unless new questions come up which need discussion in a meeting
- Release Engineering (Ethan):
    - (No progress from Ethan)
    - Any testing of v2024.0.0-alpha.2? Any issues noticed?
        - T: regarding Satoshi's experience with the segmenter (seeing dirt being outputted by the segmenter) - I'd be interested to see the raw images and the segmented objects so I can see what changed
            - S: I'm repairing my PlanktoScope-JP2 so I no longer have any working machines at the moment which can analyze things. (Repairment will finish soon, so might be possible to test again next week)

        - T: Jeremy has been testing it and was fairly happy with it. We should also have him give some feedback. I tested this week and it was working pretty well; it was very satisfying to have the previous bugs fixed. I haven't had time to specifically test quality, optics, sharpness parameters.
        - T: I think it's fine to go into beta to expose this to more people, even before continuing to iterate on image quality parameters.
        - Conclusion: the main precondition for going into beta is for Ethan to fix a few things which people requested to be fixed (in previous meetings or on Slack)
- OS (Ethan):
    - Merged PR 411: the `machine-name` binary (for generating a machine name and corresponding wifi SSID from the RPi's serial number) is now provided in the OS by Forklift rather than by the OS setup scripts; this is a low-level implementation detail which doesn't need discussion.
- GUI (Loïc):
    - L: during the past two weeks I've been trying to find many solutions to fix a big problem I have which prevented me from hiding/showing groups without reloading the page. Right now the only way to e.g. change the displayed groups when changing the sampling gear setting is to force the page to refresh. I've been in discussions to get help from other people using Node-RED.
    - Updates on Dashboard :
        - New version of "Sample" page
            - L: I've been redesigning the Sample Info page to be visually structured more like a regular form. Not sure about the current layout.
            - L: right now the page feels too bare to me. Would like some feedback on the current design.
            - S: for heavy users, I like the format of the current version of the Node-RED dashboard where we can see all data at a single glance without having to scroll. In this new design we must scroll a lot to see everything. It's a bit of a waste of time. When we run the PlanktoScope on-board, we have very little time - scrolling around will be a problem.
            - L: the previous design was more compact, but Thibaut wanted something more like a survey - if I do that, it's more complicated.
            - A: the visual design looks clean, I like that. Not sure what else we can add.
            - (feedback from E)

        - T: I think we can reduce the spacing between components of the page. That could be done to make the page more compact.
        - T: we should first have a good understanding of the field needed, and the information type for a specific field (input text vs date picker), and the logic for showing/hiding a fiefld depending on what's selected. That's been established, and now we're transitioning to a phase which is more about the visual look. I'd like to be very intentional about that, and emphasizing the functionality over the visual appeal later.
        - T: e.g. Project name and Station ID can be combined side-by-side into a single line. e.g. sampling gear can come afterwards.
        - T: similarly, the "global sample information" title is maybe unnecessary because the things under it could just fall under the "Sample" page header. But I think these are less important concerns.
        - T: I think you can just move things around to make the page more compact.
        - T: Maybe there's another way to show/hide elements without a page refresh?
        - S: Keep the "Sampling gear" stable. After the first sampling, save the "sampling gear" setting for future samples - we want to keep our own default. i.e. have a similar behavior for sampling gear as the behavior we currently have for white balance.
- Begin of "Welcome" and "Feedback" pages
    - "Feedback form" page
        - T: would be nice to know if the page is connected to the internet or not, so that the form can be sent. Maybe the form should be disabled when there's no internet
        - E: maybe when there's no internet, we can replace the "send" button with a "save to send later" button
        - T: or we can provide some indication on the homepage's link to the Feedback form page regarding internet connectivity
        - S: is sending feedback from a PlanktoScope the only way to send feedback, or will there be an option to send feedback from a web browser on other computers?
        - T: we can see that Melissa and Oumayma are having trouble coding on the machine; and Loic doesn't have access to the machine. Could we have a machine

connected to the internet which can be used remotely by people for testing? And this could enable other people to play with the dashboard.
        - (response from E)
        - TODO for E: start building developer environment SD card images which come with the full graphical desktop
    - L: mainly I did this page because I was stuck on some other things - this page is not a high priority
- "Welcome" page
    - L: this page is also a low priority, I just did it so I could make forward progress
    - L: I implemented logic to show this page only before the "validate" button (the submit button) is pressed on the page; the page is hidden in all future startups
    - (E: various concerns)
    - T: the goal of collecting this info about users is so FairScope knows that the machine has actually been brought online. And to enter some info that won't be needed each time the machine is used. But seeing this prototype makes me agree with the concerns that E shared. We need to provide a way for the user to modify the info later.
    - T: maybe let's focus on the other pages (e.g. preview, segmentation) which are clear and sure and finalize them, and we can address these auxiliary pages (like this page) later. These will require a lot of careful thinking/design about how the pages will fit into the workflow
- Segmentation page
    - L: investigated the possibility of adjusting the order of the segmentation queue. But when we launch the queue, the page refreshes and loses its state of the selected elements.
- L: I haven't been working on acquisition and preview pages because they're complex. Last time we were working on this page, you (T) requested something different from the design/layout of the current Node-RED version, and you wanted to see if we could make a different design. But I think users like the current layout with controls on the right. I need to talk with someone to get input for the design of the preview.
    - L: for Acquisition page, will try to implement in Node-RED to see if it's feasible. That's my next step

        - L: for Preview page, I need more input on the design of the page.
        - T: I think having the controls on the right of the Preview page is fine.
        - L: when we were discussing the Preview page, you (T) said you don't really want to stay with the controls on the right; but if the current design is fine for you, I can proceed with the current design
        - T: let's come up with a good design which looks good on the Preview and Acquisition pages. The idea of your internship is to propose new potential designs for the graphical interface. Maybe instead of proposing one solution, just play with different options which come to mind, and don't spend weeks on it. Just propose some ideas of options, and we can discuss this.
        - T: you can have different prototypes for a Preview page which explore different aspects of the problem.
        - L: I'd like to get confirmation about what information I need to put in the controls on the Preview page - which input elements/buttons are needed?
        - T: sounds like you need more context from me about how the machine is used; (explanation of some context about how image focusing is done by the user on the machine; coarse vs. fine focusing); maybe look at other microscope control software projects to see how they did their GUI design
        - A: https://micro-manager.org/ could be a useful program to look at to see how they did their GUI design
- Any other work packages with updates, questions, or action items to discuss?

Other updates + discussion

2024-05-09

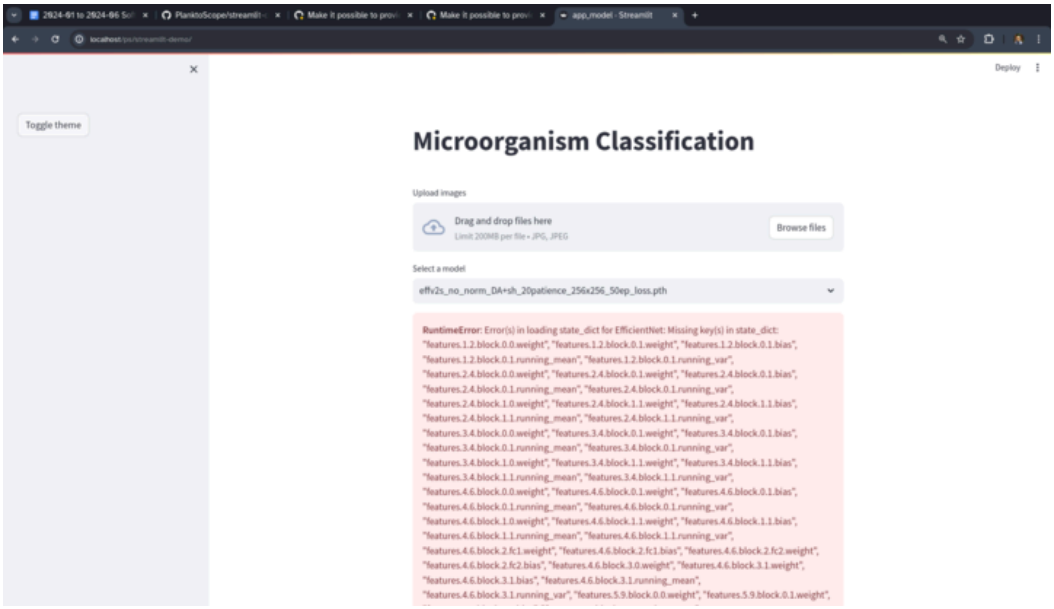
In attendance: Ethan, Satoshi, Adam

Work package updates + discussion

Refer to: Software Work Packages

- Targeted ML-based classification (Wassim):
    - Update from Ethan: I merged PR streamlit-classification-app#2 so that the Forklift package can use pretrained models, and I tested it using the pretrained model weights file Wassim mentioned last meeting. Here's what it looks like (i.e.

we're making progress!):



- Hardware Drivers (Ethan):
    - Updates from Oumayma
        - PR 34
- Backend APIs (Ethan):
    - Updates from Melissa
        - Diagrams/PR 402
- Docs (Ethan):
    - Follow-up on a DM request I received from Thibaut to remove the FAQ's question about pricing: discussion & decision-making
        - E: (explained my proposed compromise to change our answer to the question, without removing the question entirely)
        - S: in my opinion, it depends on the situation. Makes sense for Thibaut to ask this question since he's leading FairScope. But in my case, I also receive questions very often in Japan about the cost of the PlanktoScope. I would be happy if we can leave this question in the FAQ. But I think it's okay to delete the question if Thibaut wants to delete it.
        - A: I'd be fine with both.
- Release Engineering (Ethan):
    - (No progress from Ethan)
    - Any testing of v2024.0.0-alpha.2? Any issues noticed?
        - S: I just tested alpha.2; haven't noticed any problems, it works well. I checked the segmentation compared with v2023.9.0, and it's almost comparable. alpha.2 has a bit more noise in the segmented objects compared to v2023.9.0 because the flow cell is dirty and v2023.9.0 was able to delete the noise/dirt while alpha.2 was not able to delete the

        noise/dirt; it was easy for me to recognize those false positives in the segmented dataset (they looked like circles).
        - S: segmentation in alpha.2 was slightly faster than in v2023.9.0
- OS (Ethan):
    - Forklift: it will soon be possible to manage/update the installation of machine-name using Forklift rather than in the initial setup scripts
- GUI (Loïc):
    - Updates
- Any other work packages with updates, questions, or action items to discuss?

Other updates + discussion

- Any other topics to discuss?
    - S: built PlanktoScope-JP v2.1; revised/improved some design features (e.g. AC adapter). Planning to take some project from my institute to test more PlanktoScope in our institute to monitor plankton around Japan. I'll have presentation next Monday.
    - E: attended OHS this past weekend; nothing much to share (didn't really have much discussion with people about PlanktoScope or Forklift)
    - A: I've been documenting the darkfield setup, trying to figure out how to integrate it, trying to find the right LED. Also working on the passive plankton concentrator (cyclonic design, like the Dyson vacuum: continuous centrifugation), been doing some 3D-printing associated with that. Resubmitting my Pyrocystis paper - doing a bunch of new experiments; need Manu to review it, but he's travelling for the next two months.
    - (discussion about for-profit vs nonprofit entity for PlanktoScope in the US)
        - A: I talked to some people, still unsure what will work best for our current situation. It seems like a lot of people regret doing nonprofit. I would've liked to meet with the people from law school doing that business class. We'll probably have to decide without Manu, he doesn't have a good-enough idea to make the choice. I had thought this choice would've been more straightforward
        - Next step will be to talk with Manu about envisioned scope/activities of this entity
        - (more discussion)
        - S: even now, many people want to use the PlanktoScope - not only scientists, but also community level (students, fishermen, aquaculture). In Japan, it's difficult for us - many aquaculture ponds or cages all around Japan. Many people are getting annoyed with plankton. There will be so many people.

- S: For now, only a few people know the PlanktoScope. When I talk about the machine, almost everyone are interested in it. As more people know about the machine, more people will join the community. I've only been back in Japan for less than one year, but I've met so many people who are interested. In Japan, there are just two options: DIY building of PlanktoScope; but usually people ask me (because we communicate in Japanese); or buy FairScope's PlanktoScope. But in Japan, we usually must buy overseas machine and through some intermediate agency; for FairScope, we're using a company as an intermediate distributor who buys PlanktoScopes from FairScope and sells them to us. Technical support from that intermediary is fine, we can work well. I asked them to be an intermediary because they know well about Zooplankton and EcoTaxa (since they also work with UVP). Maybe half of the people would want to do DIY and half would buy from intermediary, not sure.
- S: There's a good protocol from Fabien, we can translate it with Google. I'm planning to make a Japanese-language version of the protocol. I haven't worked hard on that because I think the protocol could be a good chance for people to practice English. I just sent my v2.6 PlanktoScope to my colleagues, and she just tested it yesterday. With Fabien's protocol and some a few notes from me, she was able to make it work well for her.
- S: Right now we just have a small community less than 10 people, so I'm able to handle all this myself, and we all know each other. But if/when the community grows, I will not be able to handle everything.
- E: sounds like maybe a potential area of need for a nonprofit could be related to the challenges with growing a community
- S: another thing which is challenging is identifying species after taking images. I think we should take money if someone wants someone else to identify the plankton for them. It takes a lot of work and knowledge. It should be one of the central issues for PlanktoScope. We somehow take some money to support our activities, so it might make sense to have some service people can pay to label objects for them.
- S: we can never identify all of the plankton with ML/DL, we still need humans as the foundation.
- E: a nonprofit research organization could be a direction for the US-based nonprofit
- A: this would be a good thing to talk to the Law School people about, since they have experience with all these kinds of nonprofits, and they've helped other entities who've gone nonprofit and entities who've gone for-profit

        - A: let's try to set up a meeting with them and Thibaut next week - I'll cc you on the emails
        - S: I'd love to stay updated on your progress - it will be helpful to hear about your experiences as I think about what we'll do next in Japan
- Review of actionables/TODOs for next week:

2024-05-02

In attendance: Oumayma, Adam, Morgan, Melissa, Thibaut, Jeremy, Satoshi, Ethan, Loïc, Wassim

Introductions

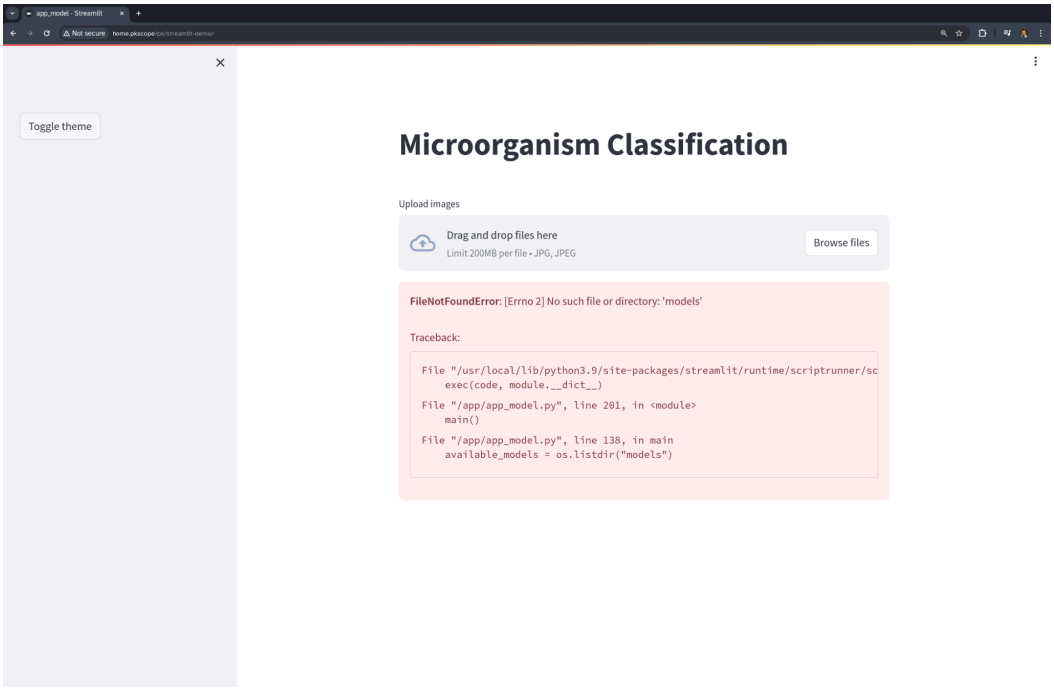
- Jeremy: recently hired at FairScope; biologist (PhD in marine biology, microbiology): interaction between phytoplankton & parasites; microscopy & flow cytometry. PlanktoScope R&D, improvement of the machine.
    - T: Jeremy is a fixed position at the company (here for the long-term). The goal is to publish a paper eventually with you (Adam)
- Satoshi: biological oceanographer, working on phytoplankton & zooplankton, using imaging systems (incl. flow cytometry & planktoscope); working at Japan's governmental fisheries research agency. End-user of PlanktoScope
- (Ethan)
- Adam: Prakash Lab; PhD in biochemistry; Pyrocystis, ML, etc.

Work package updates + discussion

Refer to: Software Work Packages

- GUI (Loïc):
    - Follow-up from last meeting, feedback from Jeremy about the sample metadata input form. Had time to discuss with Thibaut, he agreed. We'll put things on pause for the moment to focus on the principal pages, and later add them to the planktoscope. For now I won't work on any of that (sample metadata, templates/campaigns, user switching camera to preview, statistics, etc.) - these are not priorities yet.
    - Restarted the sample page - Thibaut wanted something more like a regular form for users (scientists & non-scientists) to understand.
    - T: the logic about the order of inputs to be filled, and switching between different sampling gear types, and positioning of information; a more visual approach. We don't need to have everything condensed in a single no-scrolling page; we can

have a longer page which is easier to fill, e.g. because it's easier to understand visually.
    - L: my work is still a work-in-progress, nothing interesting to show today.
- Targeted ML-based classification (Wassim):
    - Update from Ethan: [streamlit-classification-app PR 1] adds automatic building of the Docker container in GitHub Actions, and a Forklift package to deploy it on a PlanktoScope (see the URL in the screenshot):



    - Caveat: we can't really install Pytorch for armv7, so this app is running on a 64-bit version of the PlanktoScope OS without the Python hardware controller
    - Note for `wassim@fairscope.com`: currently the pretrained models are not downloadable by any URL I'm aware of (I assume you have some URLs of models on Google Drive or something), so the app shows the error in the screenshot above if it's built on GitHub Actions as a Docker Container. If you can give me a URL of a small pretrained model which I can download using `curl`, then I can download it into the Docker container (as part of the GitHub Actions workflow) so that the demo app has a model to run:
        - W: [effv2s_no_norm_DA+sh_20patience_256x256_50ep_loss.pth]
        - TODO for E: try embedding this model into the Docker container image
    - W: we're still annotating data on EcoTaxa; we have the 3 larval stages (D veliger, umbo veliger and pedi veliger) for each of the three species (mussel: Mytilus galloprovincialis, Mytilus edulis & oyster: Crassostrea gigas). The other intern Laetitia at FairScope will take charge of that. But we're able to start experimenting on the dataset of larvae.
- Release Engineering (Ethan):

- Update from Ethan: I tried setting up v2024.0.0-alpha.2 on the 64-bit version of Raspberry Pi OS bullseye (we were previously stuck on 32-bit because raspimjpeg was compiled for 32-bit), and the Python hardware controller seemed to work fine (and there should be a significant speedup in the segmenter, according to previous measurements). I want to keep building 32-bit SD card images until we delete raspimjpeg entirely (in v2024.1.0) - but should we start building and uploading 64-bit SD card images with v2024.0.0-beta.0, e.g. as `planktoscope-v2024.0.0-beta.0+adafruithat.arm64.img.gz` and `planktoscope-v2024.0.0-beta.0+adafruithat.armv7.img.gz` (and also the equivalents for planktoscopehat)? If so, how do we want to communicate to users which SD card image they should download?
    - A: the only reason to stay with 32-bit is for raspimjpeg?
    - E: also to have a longer testing period for 64-bit OS
    - T: if we agree that 64-bit version with picamera2 makes more sense for the beta and it simplifies things to just do 64-bit, let's just do that
    - S: I agree. Having many versions will make things more complicated for end-users. We already have various software versions - if we only provide a few versions, that will be simpler. If we want to move to 64-bit, let's just do that directly.
    - (consensus: just go directly to 64-bit build for v2024.0.0-beta.0, and if someone has problems they can run the setup scripts on 32-bit RPi OS or I can make a 32-bit SD card image for them)
    - M: the Google Coral device works with TensorFlow Lite; it won't be used by the segmenter, but it'll be used by other models. Don't know if it'll have any interactions with 64-bit environment.
- Any findings/feedback from testing v2024.0.0-alpha.2?
    - Any testing? Any issues noticed?
        - J: have tested it a bit. It's the only version I was using up to now. Main thing is that the quality of the preview stream is…when the image is full of stuff, it's a white image; if just a few things, it's clear. If many objects are in the FOV, it's less and less clear. Preview stream quality is not enough. We tried to do some improvements, didn't get enough improvements.
        - J: PlanktoScope isn't made to observe culture in flasks, but I tried to do that, sampling from the bottom of a flask where all particles had sedimented. Tried to focus on a specific cell which isn't moving, but couldn't focus effectively because the cell was too small. While using the preview, it wasn't enough to do focusing - it was too pixelated. Can we have a way to change preview stream settings with three quality levels (low, medium, high)?

        - T: (summary of J's comments): hard to do focus because preview stream is too pixelated; and image quality decreases when many objects are in FOV
        - E: for v2024.0.0-alpha.2, I increased frame quality setting by a lot, and I also increased the number of pixels in the image; probably the only way to improve image quality beyond alpha.2 is to increase the number of pixels even more
        - J: I'm not sure I tried the v2024.0.0-alpha.2 version - it might have been alpha.1
        - E: ok - try v2024.0.0-alpha.2 and let's see if the preview stream quality improvements there are good enough or if I need to increase number of pixels in the stream frames
        - J: why is exposure time so high? With a shorter exposure time, we could remove the need for stop-flow imaging.
            - T: picamera2 library allows Satoshi to try testing out the global shutter camera
            - E: hardware limits us to at least 100 us, for the Pi Camera Module v2
        - S: I've tested global shutter camera with alpha.1 (haven't had time to test alpha.2 yet). Will probably have some to test alpha.2 next week. When I tested global shutter camera with alpha.1, I felt that it required more power/resources - the software was not working as smoothly.
        - E: we'll leave one or two weeks for nesting
- Follow-up on FL's feedback about having an ISO slider, e.g. with increments of 50 (ISO 100, 150, 200, 250, 300, 350, etc.) - do we have approval to make that change for v2024.0.0-beta.0? Is there anything else we need to consider regarding this change?
    - S: I agree with FL about changing ISO to a slider; if we just have a slider, it can be hard to select an exact value
    - E: I'll try to have increment of 50 for slider, rather than increment of 1
- T: for the beta or alpha, did we have anything with hardware.json becoming empty?
    - E: I haven't seen this before; the fact that Maggi got this problem on a PlanktoScope which had worked for a long time suggests that there's a problem in the Node-RED dashboard.
    - T: currently we're storing data into .json files; Node-RED could do that itself with the global variables in Node-RED - to automatically synchronize global variables with files. Instead of using our

current method, we'll try to move towards that standard approach for saving values.
        - E: it could be complicated/difficult to fix this in the current Node-RED dashboard. Maybe wait to see if more users experience this problem.
        - T: we'll need to figure out how to reproduce this problem.
- Hardware Drivers (Ethan):
    - Updates from Oumayma
        - Currently working on refactoring light.py; currently it has two types of LEDs. One is controlled by I2C bus, other is PWM. Thibaut proposed that we keep just one type, to keep the code more organized and shorter. For that I also need to follow structure of stopflow.py as a model; haven't really started to modify the code yet. And I should use that as a model for the code of focus.py.
            - E: maybe use camera/hardware.py and camera/mqtt.py as the model, instead of stopflow.py
            - E: will need PWM LED driver for the adafruithat
            - Haven't tested these changes yet - had problem with connecting to the PlanktoScope at the same time as connecting it to the internet, haven't resolved the problem yet
                - Now testing with the recent version, we can't maintain connection to the PlanktoScope. We tried it on both my computer and Melissa's computer, neither worked. The problem goes away when switching back from alpha.2 to alpha.1
                - TODO for O: send a message on Slack about problems, E will follow up for troubleshooting.
                - E: for now, just download files from PlanktoScope to your local computer in order to push changes to GitHub.
                    - TODO for O: send a Slack message for E to follow up on this
            - T: in planktoscopehat version, we're using "LED 1" to control the illumination LED. Make sure that's the PWM LED.
                - (discussion)
                - T: adafruithat LED is digital pin, not PWM pin
                - E: it looks like you're not using the PWM LED right now, so you can delete that. We'll need a digital (on/off) LED driver for the adafruithat PlanktoScope.
                - T: fairscope won't support any work for the adafruithat PlanktoScope

            - E: I'll make the digital LED driver, then
        - PR 28
            - O: this PR should be closed
        - PR 34
            - E: Next step will be to test the code (test on alpha.1 if needed), which should be possible even without internet
    - Updates from Melissa
        - PR 29
            - M: I have many problems with it, but I didn't work on it this week. Have to look at logs on the PlanktoScope.
- Backend APIs (Ethan):
    - Updates from Melissa
        - Diagrams/PR 402
            - M: I tried to make a new PR (because I did things very differently) to add descriptions about the changes I made, but the changes went to last week's PR.
            - E: it's fine to continue using the old PR since the changes are to the same MQTT page.
            - Next steps: continue adding info about allowed range of values, etc. Also play around with Node-RED to provide concrete examples to test out the changes Oumayma is making - different MQTT messages which can be sent to Python directly from Node-RED; debug panel to help check what's being sent. That can be the next way to do not just documentation but also testing.
    - Updates from Ethan:
        - At the end of last week I attended a workshop (https://depts.washington.edu/machines/scienceautomation/) on building a community & open-source hardware+software ecosystem for lab automation. Workshop activities included trying out a CNC multi-tool system (Jubilee) which has been used as a liquid-handling robot and for other science things; the interface for controlling it is a Python API used in Jupyter Notebooks, and it was a really great way to prototype automation experiments. I think having a Python API (e.g. a Python client for the MQTT API) will be the right long-term approach to enable scientists-who-know-Python to automate their PlanktoScopes or do other advanced/custom things.
            - T: this will simplify development of software on the PlanktoScope - it should be easier to develop and evolve things on the PlanktoScope. Jupyter Notebooks could simplify things. Scott

wanted to just do Jupyter Notebook on the camera itself, Node-RED was an unnecessary layer of complexity for him. This could also be nice for data visualization. And this could be an alternative to the existing way we do segmentation, or a way to test new features.

- Any other work packages with updates, questions, or action items to discuss?
    - T: For anyone who wants to stay, can we go through the picamera2 docs and look at parameters related to image quality, and what parameters we can tune?
        - O: last time I saw parameters in the camera/hardware.py

2024-04-25

In attendance: Ethan, Oumayma, Melissa, Loïc, Thibaut, Morgan, Fabien

Work package updates + discussion

Refer to: Software Work Packages

- Release Engineering (Ethan):
    - Updates on blockers for v2024.0.0-beta.0 prerelease:
        - Merged device-backend PR 32 (disable subtraction of masks between consecutive frames by default; the user can enable a Forklift feature flag to re-enable the old behavior) and device-backend PR 33 (increase camera preview from 640x480 to 960x720, compared to the raspimjpeg camera preview which was 800x600; and increase max quality of each frame)
        - Merged PR 399 (which integrates the device-backend changes with some other improvements), and did some light testing.
        - Published a v2024.0.0-alpha.2 prerelease last night (instead of a v2024.0.0-beta.0 prerelease). Due to the potential impact of the changes made since v2024.0.0-alpha.1 (and, in particular, because the requested change to segmenter behavior is a backwards-incompatible change which I think deserves some more testing before we release to all users for testing), this next prerelease will be a v2024.0.0-alpha.2; we can give it one week (or two weeks, if none of us have time for testing this upcoming week) for testing with advanced users and, if we don't identify any more blockers for a beta prerelease, we'll promote it to v2024.0.0-beta.0 for everyone to test.
    - O: M and I downloaded the SD card image but are not yet sure what needs to be tested.

        - T: we can give you a proper training on operating the planktoscope. You should follow up with Jeremy.
    - T: quality of acquired images?
        - E: (test again with the new ISO calibration; maybe try changing the sharpness adjustment setting)
    - (regarding the issues Jody encountered in https://planktoscope.slack.com/archives/C019Q3PT5T3/p1713568135261089)
        - T: we weren't sure what to use between the old OS version before your work vs. v2023.9.0 plus some additional fixes. Right now we're using the old version with an updated value for the LED brightness.
- Targeted ML-based classification (Wassim):
    - We have no points to discuss this week
- Docs (Ethan) & Backend APIs (Ethan):
    - Merged PR 398 (some small additions + updates to the OS-related pages of the technical reference docs; and renamed "PlanktoScope Software Distribution" to "PlanktoScope OS"; and restored the sidebar link to https://docs-edge.planktoscope.community/reference/software/interfaces/mqtt/ for the old MQTT API reference page describing the current MQTT API - Melissa will rewrite this page so that we have an accurate and comprehensive reference of what the MQTT API looks like right now, before we start changing the API)
    - Updates/questions from Melissa + discussion
        - M: Here is a link to architecture script for each component, which I've edited with a mermaid real-time editor that can be accessed via notion https://www.notion.so/Update-of-week-dfb603d4a108430790796298be99a14e
            - M: T told me to add diagrams to the documentation
            - M: I tried to push things to GitHub but there's some error with GitHub Actions. Initially I tried doing things in Markdown and using a script to generate diagrams. So for now I just uploaded the images.
            - (feedback from E and T about the diagrams)
- Hardware Drivers (Ethan):
    - Device-backend PR 29 from Melissa
        - E: GH Actions errors: https://github.com/PlanktoScope/device-backend/actions/runs/8785732151/job/24106917484?pr=29; I can add a PR review to fix issues.
        - M: I haven't tested this code - I don't know how to test it
            - TODO for E: improve the onboarding experience to help new developers understand how to test the code

        - TODO for E: follow up with M and O on Slack to schedule a video on how to test code changes
        - O: is there another way to do testing of code changes, not on a PlanktoScope?
        - TODO for O + M: test the “Remote - SSH” plugin in VSCode (prefer that approach rather than my low-quality poethepoet scp command)
    - Updates/questions from Oumayma + discussion
        - (follow-up on GitHub Actions problems with Docker container builds; resolved now)
        - O: will make another PR for the refactoring of stepper.py
        - O: device-backend PR 28
            - O: wasn’t sure how to test changes.
        - O: I tested the new poethepoet command in WSL, and I had an error of “scp: unexpected filename”. I tested with planktoscope.local and home.pkscope
        - O: I’m also working on metadata in the camera subpackage, working with Thibaut on the Metadata Compilation file
    - Update from Ethan: device-backend PR 33 adds an `scp` poethepoet command; see the hardware controller’s README for usage instructions (Usage -> Development section) and prerequisites for that command to work.
- GUI (Loïc):
    - Updates on node-red dashboard :
        - Issue with a map integration on sample page (worldmap node-red)
            - Existing options are not immediately ready to use, they’ll require changes (if they can work at all)
            - E: if we need to do our own HTML+Javascript, https://openseadragon.github.io/ might be useful
            - FL: when I make scientific figures, I just plot coastlines as a series of points in an x-y graph. https://www.ngdc.noaa.gov/mgg/shorelines/
        - Feedback of a user (Jeremy @ FairScope) to make templates/presets for sample metadata and project settings
            - (discussion with E & FL)
            - Next step: talk with Thibaut for review/approval
        - Feedback to add a page where a user can see all segmentation and modify metadata. It adds a problem if we modify metadata after segmentation - it’s already exported for EcoTaxa.

    - FL: if metadata is modified after segmentation, we need to re-export the EcoTaxa export archive with an updated metadata TSV table.
- Preview page: can we rotate the camera? Also, can we add mirror mode to swap image? e.g. adding CSS to the MJPEG stream.
    - FL: good idea, not sure how useful that will be. Initially the PlanktoScope was supposed to be turned on its side - which made the view of the camera mirror what we see on-screen. When we rotated the PlanktoScope differently, then we had to rotate the camera to keep the field-of-view dimensions correct.
    - FL: I don't think mirroring will help much. It does look weird for people to see their samples go from down to up or from left to right in the image - so rotating will be useful. The orientation of the camera is fixed; changing orientation in the camera preview will change the UI layout
    - E: rotating camera preview stream so particles flow from up to down may help new users
    - FL: experienced users don't care too much, but it will help new users
- Thibaut had idea to add a page to see a summary of statistics of each sample (the info to be displayed was not specified, so unsure what we would display)
    - FL: for metadata inspection, this will be extremely important - trained users make many errors in entering metadata! Not necessarily *all* metadata, but having a global overview across samples can help detect anomalies.
    - FL: if there's a possibility to update and correct metadata, then it doesn't matter when the user checks the global overview of metadata across all datasets. A "metadata editing" page should provide this global overview of metadata.
- Adding videos/photos to the documentation of how to place the PlanktoScope, etc.
    - E: maybe just photos/images - not videos
    - FL: usually the protocols in protocols.io are extracted and sent to planktoscope as PDF file. It could also be exported as a JSON file - so you could add an interactive protocol.
    - E: if we can reuse an existing renderer for the protocols.io JSON files, it could be easy to integrate. If we have to write our own renderer, that's a big project.

- The different PlanktoScope hardware versions all have different boxes and hardware inside. For the new version of the PlanktoScope hardware, Thibaut wants to keep the placement of modules and boards and everything the same - so users can upgrade hardware without changing the wooden box. Can we do in-place software upgrades
    - E: yes, this is what I'm working on.
    - (decision: we will include this into the GUI wireframe/prototype, and we can hide that in the GUI until in-place upgrade functionality is added)
- Homepage (layout + content)
    - FL: I'm used to having a shutdown button on the homepage. It's nice to have a shutdown button which is easily accessible
    - FL: when you first start a PlanktoScope, the first thing you do is adjust the optics so you get a view of the sample, before you even think about filling in metadata - not run a sample!
        - FL: it's annoying that the light isn't on at the start, and instead we have to start the light ourselves.
    - (consensus: instead of "run a sample" button, "preview a sample" button - and instead of going to metadata page, go to the sample preview page)
    - FL: it would be nice to have a "shutdown" button in the navigation sidebar button on the left. The button should have a confirmation dialog, and maybe also be locked.
    - FL + E: need for some kind of quick access to shut down the PlanktoScope, e.g. like https://github.com/bluerobotics/BlueOS/blob/master/doc/dashboard.png (though we can have a different design)
    - L: to implement this with Node-RED dashboard, we can have a "shutdown" page in the navigation sidebar
- Sample metadata entry page
    - Once we fix the hardware clock, show the UTC time to help the user understand that they must use UTC time. Maybe also a button to click to fill in the sampling date/time from the current UTC time?
    - Rename the "Test" button to something else, e.g. "No metadata" or "Demo/Test" or "Quick Test"
        - (consensus: "Demo" or "Demo/Test")
    - Latitude/longitude: split the different formats into two different columns and separate them

        - Sampling date & time should be specific to the sample type. Also, what are the specific semantics of “sampling date & time”? e.g. when a Niskin bottle was closed, or after it came back on deck? Also, maybe we do need to record sampling end time for the “Net (horizontal)” sample type?
    - Navigation footer element
        - FL: for pages at the end (i.e. no start or no end), have a disabled button for the direction we can’t go (instead of hiding that button) to make things more intuitive
    - Overall design?
        - FL: a lot of space is used for a small number of things - e.g. the big buttons like “global sample information” take a lot of space but are filled with white everywhere
- OS (Ethan):
    - Merged PR 397 (discussed last week); no discussion needed.
- Any other work packages with updates, questions, or action items to discuss?
    - T: There’s ongoing work on oyster larvae; I can share some data now (https://ecotaxa.obs-vlfr.fr/prj/12053)
    - T: I’ve been restructuring the Metadata Compilation spreadsheet. In a few weeks maybe we can go through this and discuss together with Fabien, Oumayma, etc., and be able to provide info to Melissa for documentation
    - FL: I’m ready to test something with the PlanktoScope!

Other updates + discussion

- Any other topics to discuss?
- Review of actionables/TODOs for next week:

2024-04-18

In attendance: Adam, Oumayma, Melissa, Ethan, Loïc, Wassim

Work package updates + discussion

Refer to: Software Work Packages

- GUI (Loïc):
    - Updates on wireframe prototype (figma link)
    - Presentation of the beginning on node-red: just trying to transfer the wireframe to Node-RED to see if the wireframe’s design is possible to implement in Node-RED

        - Home page
            - Question for T: any stats to show besides internet connectivity and disk space remaining?
        - Sample page: blocked, need input from someone on what metadata needs to be entered into this page
            - A: would be great to have a button to just save the current image in Node-RED
            - E: yes, like a “game screenshots” in video games - but for the microscope
        - Preview page: where should the “pump by one field-of-view” buttons go? It might turn out that Node-RED makes it hard to have them in the corner of the camera preview
        - Hardware settings page: will follow up with E later about what needs to be added to this page
- Targeted ML-based classification (Wassim):
    - Resource (GPU & RAM) allocation for training ML models: Google Colab’s resources are insufficient, we need more RAM; is it possible to get GPU resources, or buy a FairScope machine, or rent some cloud compute resources? This needs input from T
        - A: a professional Google Colab subscription might be sufficient for more RAM and better GPU.
    - I’ve started reading about larval stages; thought about attention mechanisms in NN architectures. Tried to implement some attention mechanisms; will try various approaches. Then we can select an approach which meets our needs.
    - Started a small Streamlit-based web interface with a pre-trained model (for 5 classes) which outputs predictions. At the bottom of the page we get a graph showing distribution of each class.
        - E: L, how would we integrate something like this from Node-RED if we deployed it on a PlanktoScope?
            - L: will be simplest to keep these kinds of apps external, and then we link to them from Node-RED somehow (either direct links from Node-RED or a link to an “apps” page with a list of links to each app)
        - E: I’d like to have a prototype where we deploy this app to run on a PlanktoScope, via Forklift. Next step for W: get this into a Docker container
- Hardware Drivers (Ethan):
    - O (Question): branching on Git?
        - E: feature branches + PRs. Smaller feature branches, rather than long-lived ones. Merge in small steps, one branch per module.

- O (Question): Regarding testing procedures, are we planning to implement unit testing to validate individual components of our codebase, or will we primarily rely on integration testing directly on the Planktoscope to verify the functionality of the system as a whole?
    - E: (thoughts about testing)
- O(Question): It seems that much of the functionality implemented in imager is now being transitioned to mqtt.py in imagernew.Could you confirm whether the code in imager is still actively used, or if the plan is to deprecate it in favor of mqtt.py in imagernew?
    - E: we'll delete imager after the v2024.0.0 release, and rename "imagernew" to "imager"
- Updates from Oumayma / discussion
    - O: we tried to use MobaXTerm, had some problems with connecting to the PlanktoScope. The problems went away automatically. We tested both beta and alpha versions (Thibaut gave us two SD cards with the alpha & beta versions on them), had same problems with connectivity (both Wi-Fi hotspot and Ethernet cable). It said it was connected, but when we tried to connect over SSH in MobaXTerm (to planktoscope.local), we couldn't. But when we retried on Monday, we didn't have this problem - the problem disappeared by itself.
    - O: then we tried to install VSCode on the Raspberry Pi. We'll stick with using nano to modify the code. We also looked for other alternatives (e.g. git hooks to automatically copy code to the Raspberry Pi when we commit)
        - E: VSCode on your computer might be able to edit files on a remote compute via SSH
        - L: there's an extension for that. Just use SSH URL with password, and then you can access remote files
        - W: the extension is called "Remote - SSH". I've used it before too.
        - TODO for E: new idea: as part of developer workflow, I can add a poe command to copy the files to a specified Raspberry Pi
    - O: I tried out using pip deptree to generate a graph of dependencies. But it was too messy to be readable.
    - O: then I started modifying the main.py code. Also had to look at mqtt.py and imagernew and stepper.py. Before editing them I have to refactor stepper.py and split it into pump.py and focus.py
    - O: for handling of process lifecycles, I implemented something to just automatically restart the process a limited number of times, within the MQTT process. Afterwards, an error is logged and the process terminates.

            - E: for now, let's just let things die and not try to auto-restart them (due to reasons of complexity)
        - O: I made a plan on the refactoring: (https://miro.com/app/board/uXjVKS7RuaY=/)
            - E: besides needing to remove the "auto-restart" from this plan, everything looks good to me
            - TODO for O: get the changes on a branch in github
- Backend APIs (Ethan):
    - Updates from Melissa / discussion:
        - M: what documentation should I add to GitHub?
        - E: just focus on MQTT API documentation for now
    - M: I started working on code for automatic reconnection to the MQTT server
        - E: no need to limit the number of attempts to reconnect, can just keep trying forever
        - E: this looks good. What's the test plan?
    - M: what are the next priorities?
        - E: start drafting a design document in https://github.com/PlanktoScope/proposals
- OS (Ethan):
    - E: with PR 397 (and the PRs linked to from it) which I'll merge today, all OS config files are now managed in Forklift (except a few system service definition files to integrate Forklift with the OS), so we now have a working mechanism to update almost everything PlanktoScope-specific (besides the hardware controller and the Node-RED dashboard, until we containerize them; and the bootloader configuration to enable hardware kernel drivers for things like I2C, because that's lower-priority and requires slightly more careful design) without re-flashing the SD card, just by running `forklift pallet switch github.com/PlanktoScope/pallet-standard@branch-name_or_git_tag_name` and rebooting. This is part of the work I need to record a video talk about tomorrow for the Open Hardware Summit, which is why I prioritized it for my work this past week.
    - E: a nice byproduct of this change is that if a user causes changes/customization to any OS config files in certain directories (e.g. for wifi settings), the changed files are all collected in a single directory - which makes it easier to see what changed. And that directory can just be copied onto a fresh SD card installation (followed by a reboot) to restore the changes/customizations. It'll make my work simpler & faster when I'm experimenting with OS configurations (e.g. to make our autohotspot work on RPi OS 12 for RPi 5 compatibility), but most users don't need to know about this.

        - E: this also makes it possible to add a “powerwash”/”reset” feature to reset the SD card to its initial state while preserving all user/application data; or we can add a “total factory reset” feature which also resets all user/application data (which is like re-flashing the SD card, but faster). But to keep things simple, let's just keep users on the current “re-flash the SD card if you have any problems” workflow which we use right now, at least until we hear complaints about that?
- Release Engineering (Ethan):
    - E: No updates, last week’s requested changes before beta are still on my backlog. Will try to take care of them early next week.
- Docs (Ethan):
    - E: PR 396 fixed problems with setup instructions for the project docs site, and Melissa tested the docs and was able to get local live preview working. The testing process highlighted the need for a proper onboarding guide which maps out which folders/repositories each part of the project is in, since there were mismatched assumptions about which guide Melissa should be following.
        - Melissa: any additional feedback for improvements needed to the docs, or documentation which needs to be included in a proper on-boarding guide?
    - E: No other progress from me, I've been too preoccupied with other deadlines. TODO is still to update PlanktoScope Software Documentation; also still need to write a proper on-boarding guide
- Any other work packages with updates, questions, or action items to discuss?

Other updates + discussion

- Any other topics to discuss?
- Review of actionables/TODOs for next week:

2024-04-11

In attendance: Thibaut, Wassim, Morgan, Oumayma, Melissa, Loïc, Ethan

Work package updates + discussion

Refer to: Software Work Packages

- Release Engineering (Ethan):
    - Are we ready to promote v2024.0.0-alpha.1 to v2024.0.0-beta.0 after this meeting? In other words: 1) are we confident enough that the software is stable enough that it’s fine to ask everyone to test it out, and 2) do we think that we probably won’t learn anything new about potential problems by waiting another

week or two for additional testing among people in the #6-dev-software channel on Slack (whether that's because we're confident in the thoroughness of our testing so far, or because we don't have enough people/time to do further testing)?

- FL: I didn't got time yet to go through and Pierre only got time to quickly test it. I would definitively give it a try if possible but the current agenda is impossible until the end of next week (including 2 days of planktoscope formation+ 2 days of data analysis with master students). From the first tests done so far it seems that the camera
- T: Things to do before promoting to beta:
    - Remove the segmenter's functionality to subtract previous frame's segmentation from current frame's segmentation as a default settings. It's trying to prevent a situation which never happens, and it creates an artifact. We can go straight to beta.0 with this change.
    - Improve streaming quality to enable a decent focusing set since it's very hard right now to focus with this degraded streaming quality. Previously, it was easier to get good focus because we had enough "pixels" on the streaming to see objects correctly. Right now it's hard to get good focus because streaming is lower quality or something has changed. Would be good to increase stream quality for the beta release - start with increasing the dimensions of the preview stream.
- E: Things to before promoting to beta:
    - Change the calculation of image gain from ISO setting
- T: things to do either before beta or during beta:
    - Improve image quality since image quality is lower than before. (e.g. Developer Journal : https://docs.google.com/presentation/d/1gYuVzc7xeFSY3GsvmjYbsxB9uyJD3K4Kk4L9F38S1-4/edit?usp=drivesdk)
    - See if we still want to disable sharpness adjustment instead of keeping the default value of minimal sharpness adjustment
    - Ensure that we have input-validation logic (for both the optic config page and the fluidic acq page) to check that the pump volume is always a positive number

- Is it fine to deprecate the USB backup feature and to deprecate Portainer? i.e. we'll plan to remove them from the default SD card images in v2024.1.0 (though it'll still be reasonably straightforward for advanced users to re-add Portainer as a third-party app)

        - FL: it was bugged since a while and nobody told me it was working again; technically getting the possibility to do a full backup easily would be useful; no need or more complicated thing but getting a “full backup copy” of whatever is in the data folder would be needed (but could be maybe implemented within the gallery “save to usb”)
        - (we will deprecate these things)
- Docs (Ethan):
    - Discussion/decision for Thibaut to review (forgot to cover this last week): is it fine if we just call our SD card images the “PlanktoScope OS”? This would simplify/clarify the language I’ve been using previously (“the PlanktoScope software distribution”) to the way we integrate all of the various software-related things (backend, frontend, autohotspot, file gallery, network configs, etc.) together for end-users. This change would be implemented in our docs and future release notes.
        - FL: yes!
        - TL: sounds good!
    - Updates from Ethan about our process & documentation for software development processes and documentation writing/publishing processes, and any follow-up discussion:
        - PlanktoScope Software Development Processes is now updated
    - TODO for E: update PlanktoScope Software Documentation
    - M: couldn’t get poetry setup to work on ubuntu (for the docs site) - the problem was package dependencies
        - TODO for E: try to install poetry deps for docs site on a fresh ubuntu VM to figure out what is missing
        - TODO for M: try again to install poetry deps, send Ethan the error messages which poetry prints
- Hardware drivers (Ethan):
    - Updates from Oumayma on documenting the existing backend, and any follow-up discussion
        - O: I’ve been looking through the code which controls each component and studying libraries I’m not familiar with (e.g. loguru). For light.py, I think the thing which makes it so long is that there are so many exceptions which are handled; the main goal would be to verify if all of this error-handling is needed
            - EL: error-handling is important for robustness. Maybe a good first step is to split the hardware driver from the MQTT API handling
        - O: I looked into isolation of failures between different processes - main.py.

        - (discussion about possibility of automatically restarting crashed processes)
    - O: Tried figuring out development environment, couldn't get poetry to work on your own computers. Next step will be to try using MobaXTerm to remotely access files on the PlanktoScope.
        - E: If MobaXTerm doesn't work, can just use System File Browser and Cockpit from the PlanktoScope's home page
- Backend APIs (Ethan):
    - Updates from Melissa from documenting the existing MQTT API and any ideas for design changes, and follow-up discussion
        - M: I looked at the docs you gave me last week to understand how Node-RED communicates with Python backend. By reading the mqtt.py code, I wrote some ideas for optimizing the code:
            - Automatic reconnection to MQTT broker when disconnection is detected
            - Add flow control to prevent loss of messages (e.g. by setting an appropriate quality-of-service parameter in the MQTT broker)
            - For reporting the state of the LED: add a status/led topic
            - Let's combine the advantages of MQTT for async communication with the advantages of HTTP for request-response interaction. e.g. for status of modules, we can use HTTP. For statuses, we don't need to get that info very often so we can just use HTTP
    - Discussion of MQTT vs. HTTP for request-response-style interactions between backend & frontend:
        - Examples of motivating use-cases for request-response-style interactions:
            - Reading & updating the state of hardware drivers:
                - e.g. send a request to read the state of the LED, receive a response with the state of the LED; send a request to update the state of the LED, receive a response about whether that request succeeded (e.g. to update the state of the LED toggle in Node-RED to be consistent with the state of the backend)
                - e.g. when the Node-RED dashboard starts, it should ask the backend what the name of the camera is (instead of assuming the backend starts after the Node-RED dashboard, and assuming the backend will announce the name of the camera at startup). Other API clients also need to be able to ask for the name of the camera.

        - Reading & updating the state of the queue of datasets which the segmenter is actively processing
        - Mutual exclusion & isolation between RPC-style function calls made by different clients:
            - e.g. while the backend is processing an image capture request from client A (which might have one set of capture parameters), it should wait to finish that before handling an image capture request from client B sent right after client A's request (which might have a different set of capture parameters, e.g. PNG output instead of JPEG). Client A should receive a response with the filename of the captured image corresponding to client A's request; client B should receive a response with the filename of the captured image corresponding to client B's request.
            - e.g. while the backend is still running an image acquisition started by a request from client A, the backend needs clear & simple semantics for rejecting any requests from client B to start another image acquisition, and only sending that information to client B (so that client A won't receive a "error: image acquisition already started!" message when client B tries to start an acquisition while client A is watching the progress of the ongoing acquisition)
- Options:
    - Option 1: continue using MQTT for push-notification-style parts of the API; switch to HTTP for request-response-style parts of the API
    - Option 2: continue using MQTT for the entire API; use HTTP-inspired design (e.g. separating response codes from error messages; naming routes by resources; making actions orthogonal to resources and/or specifying verbs on resources; making clients provide a unique request ID for each request, and including that request ID with the response; making certain classes of requests idempotent) for request-response-style parts of the MQTT API
    - Any other options we should consider?
- Trade-offs between options, to inform decision-making:
    - Trade-offs for overall simplicity/maintainability of the API's design; how important are they for us?

    - HTTP structures us into consistent pattern - good for simplicity/maintainability
    - But it adds software complexity
- Trade-offs for implementation in the Node-RED GUI; how important are they for us?
    - L: should be simple to do HTTP. Should be simple either way. No clear advantages of one approach over the other.
- Trade-offs for implementation in the Python backend; how important are they for us?
- Trade-offs for integration with third-party software (both hardware drivers to be deployed on the PlanktoScope, and API clients); how important are they for us?
    - E: the existing MQTT API made it harder/more complex for me to write an API client for the PlanktoScope in a rigorous/robust way, esp. for request-response-style interactions
    - T: My dream would be to have other modules easily added on top of what already exists. Would be good to have dashboard not just on the machine, but also in the cloud. Would be good to have several modules. Not just light, pump, focus, camera; but also agitator, heater, day/night cycle - everything in the same architecture, with simple Python code using the same easy template.
        - E: I think there aren't clear strengths of one approach over the other for integration of modules.
- Trade-offs for deployment of the segmenter outside the PlanktoScope; how important are they to us?
    - MQTT requires us to run an MQTT broker - makes deployment harder & more complex, esp. on regular people's laptops.
    - T: I'm against running the segmenter outside the PlanktoScope because it's hard for people to install. It'd be better to expose the segmetner's functionality as an installable library to use in Jupyter Notebook.
- Any performance-related trade-offs; how important are they for us?
    - MQTT bandwidth usage and latency are lower than HTTP. This is relevant if we want to send messages over a network - and probably more relevant over satellite internet. We can work around this by having an MQTT

server as a “proxy” for HTTP for requests over satellite internet, but this adds significant software complexity.
        - MQTT is more energy efficient, but probably negligible compared to running the pump
    - Any trade-offs for “developer experience”; how important are they for us?
        - T: how easy will it be to implement a new communication channel between the user and a specific hardware module?
            - E: in python, we’d probably use a wrapper for MQTT which we have to maintain; for HTTP, there might be an existing wrapper we can reuse
        - E: curl + web browsers make it easy to test and troubleshoot HTTP APIs; no similar tool exists for MQTT in general. But I could finish writing the planktoscope CLI to make a command-line alternative to curl for our specific API (this adds some complexity)
    - Any other trade-offs; how important are they for us?
- Tentative decision:
    - One option: we could try implementing MQTT and HTTP APIs for the light module, and see which is better
    - Second option: just stick with MQTT and be careful about how we design request-response-style interaction, and we can decide later if we want to add HTTP APIs
        - L: deciding later might involve a lot more work
        - E: I’d like to structure the ongoing Python rewrite so that it’d be simple to add an HTTP API later (i.e. for each hardware device we have hardware driver modules, and then we have an MQTT API module, and then we can add an HTTP API module later if needed)
    - (consensus: second option)
- TODO for T: go through the metadata compilation that Laurent had started, and we can discuss in two weeks - preferably with Fabien

- GUI (Loïc):
    - Updates on specs document (Specifications : PlanktoScope GUI redesign v1.1)
    - Follow-up discussions:
        - Planned architecture for modularity/extensibility in Node-RED (e.g. for developers writing third-party apps meant to integrate with the GUI)

        - T: iterative design of the dashboard. Loïc's work will just be a first step. It's nice to have something modular. My suggestion here is that we should have one tab that demonstrates the big functionalities that are not visible by the user but is meant to structure different functions of communication from Node-RED to the backend. So a first square where you know how to spend commands to the LED, and another to send commands to the pump.
        - T: for having other apps within Node-RED, it's easy to implement either directly in Node-RED or through some kind of wrapper (like the gallery's iframe)
        - (discussion)
        - Consensus: structure for modularity for changes made within Node-RED should be that any changes made by other people should be done in a separate tab, to make merging easier.
    - Setup wizard: UX & scope for this rewrite?
        - T: it'd be valuable to have a first iteration on specific high-priority pages, and then move to specific other pages with testing to clarify the needs. Iterative process.
        - L: I will include iterative design of the setup wizard as part of my rewrite.
- Updates on wireframe prototype
    - Simplified the home page; added some monitoring info (e.g. disk space, usage statistics)
    - TODO for L & T: go over the metadata fields which need to be on the sample info page
    - Preview page
        - Discussion of focus adjustment UI elements
            - E: what if we use the same structure/layout/pattern of UI elements for the focus as for the pump? That way, we would have the same pattern consistent across both the focus and the pump, and understanding how to operate the focus could make it intuitive to understand how to operate the pump.
        - Discussion of where to put advanced camera settings (e.g. white balance), which is currently in the hardware settings page
            - T: let's try to put camera preview on the hardware settings page to help with white balance calibration
    - Segmentation page

            - E: Showing a live estimate of the number of objects in the dataset? Could be in the statistics page, if that page will also show ongoing segmentations
            - T: Would be good to give a sense of what's coming out in the ongoing segmentation page. Could be in the statistics page.
            - T: for the "segmented" column, maybe start with just "yes/no" instead of a count of number of times the dataset was already segmented
            - E: maybe also leave "remaining time" estimate off of the first iteration of the new GUI, since that requires some implementation work in the Python segmenter
        - Hardware settings page
            - TODO for L: send E a message on Slack to discuss read-only machine info which we'll need to display
        - Added a navigation bar at the bottom of the pages
    - Update on review of feedback survey responses
        - T: we've received 3 new responses, I haven't had time to see who answered
- OS (Ethan):
    - Update on Forklift: I've been making progress on moving deployment of various systemd services and OS config files from our OS setup scripts into management by Forklift. WIP branch on our main repo (not yet a PR): feature/forklift-overlays
    - For the rest of the month I will continue to be focusing on implementing some missing features in Forklift (and managing PlanktoScope OS files + apps via Forklift) for my PlanktoScope-related work, because I need to prepare a 10-minute talk about it for early next month, and because I need something to talk about in my Prakash Lab group meeting early next month and I have no other results to present 🤡
- Any other work packages with updates, questions, or action items to discuss?
    - T: I received a USAF resolution target for optical characterization (e.g. with other lenses).

Other updates + discussion

- Any other topics to discuss?
- Review of actionables/TODOs for next week:

2024-04-04

In attendance: Loïc, Morgan, Oumayma Elbez, Melissa Djadoun, Ethan, Wassim, Satoshi, Thibaut, Adam

Introductions

- Oumayma: 2nd-year soft eng. student at IMT Atlantique. Joined FairScope this week, here for 4 months. Will work on the Python backend; refactoring it, etc.
- Melissa: 1st-year Master's student for embedded systems at Univ. of Brittany. Will work on MQTT API.

Work package updates + discussion

Refer to: Software Work Packages
(Thibaut will be late to this meeting, so we'll jump around in the agenda depending on which topics need Thibaut's input vs. what can be discussed without him)

- GUI (Loïc):
    - Progress on specs document (Specifications : PlanktoScope GUI redesign v1.1)
        - Read and answered comments on the document from Ethan & Fabien Lombard
        - Re-did the requirements section (which had "current" & "new" features) by merging the "current" and "new" subsections into a single list
            - L: Regarding Fabien's request for segmenter settings in the GUI: discussed together with Thibaut about a potential design. But there might be many settings, so it's a bit complicated. We'll leave design of a GUI to set segmenter settings for a future iteration of the GUI.
        - Moved the task prioritization into a separate spreadsheet (linked to from the document)
    - Progress on wireframe prototype (linked to from the end of the specs page)
        - (discussion of display of the number of output images from the segmenter for each segmented dataset)
        - Camera settings will be moved to the hardware settings page
        - New "segmentation statistics" page
        - Blocked on meetings with Thibaut (and maybe other people) to get more input on how certain things should be designed
        - L: please view the wireframe and add comments/feedback (but don't edit it)

        - (brainstorming of what to do with display of acquisition IDs, sample IDs, project IDs, etc. on the segmentation page)
    - **Potentially-detailed discussion** with Thibaut of some questions from the specs document
        - Planned architecture for modularity/extensibility in Node-RED (e.g. for developers writing third-party apps meant to integrate with the GUI)
        - Which would be simpler in the GUI to interact with the Python backend: HTTP request/response + MQTT notifications, or MQTT request/response + MQTT notifications?
            - M: discussed briefly with Thibaut yesterday, I generally prefer MQTT
        - Setup wizard: UX & scope?
        - (decision: we'll discuss this next week)
        - TODO for E: include these topics on the agenda for next week
    - Update on review of feedback survey responses (need Thibaut for this discussion too)
- Release engineering (Ethan):
    - Has anyone had a chance to test v2024.0.0-alpha.1 yet? Any problems discovered yet?
        - O: We tested it yesterday on a PlanktoScope. No problems identified, just brainstormed ideas for some functionalities to add in the next version.
        - E+L: Thibaut also told me he tested it yesterday and everything worked
        - T: picamera2 worked very well. Didn't test very intensively yet though - e.g. playing around with sample metadata settings.
    - So far, has anyone seen any need to delay our timeline for proceeding to beta testing (current plan is to do it after next week's software meeting)?
        - (no)
        - T: we can fix the ethernet wi-fi internet sharing bug during beta testing
        - T: on this call, only S, O, M, and me can do testing. O+M: let's do some testing this upcoming week. So by next week's meeting we'll have some good insight on the testing of the alpha.1 version of the software.
        - S: I tested the alpha.1 version with both 2.1 and 2.6 hardware just saving blank images, and it works well. My previously-mentioned concern about low file-size has been resolved by alpha.1. I think we can probably go into beta.
        - S: I'll be on a research cruise Friday of next week from April 12 to April 30 - so I may be unable to join next week.

    - Discussion (with Thibaut & Fabien?): need for quantitative comparison between the imagers, to characterize the performance of the new imager before we go into a stable release? Is it fine to proceed to beta testing before we do this?
        - T: during beta-testing, it would be nice to have a single not-moving FOV with some bubbles or specimens, and take photos with the old library vs. the new library, and make sure they're the same after segmentation. I will pass this to Jeremy (another FairScope member - full-time, recently finished PhD at Roscoff; now a "postdoc" with FairScope; he'll be focused on use of the PlanktoScope). We can use the alpha version of the software for this test. Also compare the size of JPEG files between imager and imagernew
        - S: yes, we should check whether the segmented images would change as a result of switching to imagernew
- Docs (Ethan):
    - Update: implemented various small fixes/improvements discussed at last meeting (no need to discuss): PR 386
    - Update on new technical reference docs pages (Architecture: OS; Functionalities: Camera Settings, Sample Imaging, Image Segmentation; Subsystems: Installation & Startup; Release Process): PR 390, PR 391
        - Discussion with Thibaut: I realized that the software delivered by our SD card images actually fits a reasonable definition of "operating system". Over the past year I've been referring to the software set up by our automated setup/installation scripts and delivered by our SD card images as "the PlanktoScope software distribution", which feels unwieldy and ambiguous. Maybe it would be more clear+concise to just call it the "PlanktoScope OS"? I've tried out that term in https://docs-edge.planktoscope.community/reference/software/architecture/os/ .
    - **Potentially-detailed discussion** with Melissa & Oumayma about processes + documentation for software development workflow: Best practices/workflow to edit on the python code - from the PlanktoScope itself via System file manager or some other way ?
        - O: I tried to download poetry on my computer and set up the python environment and dependencies, but I had a problem with it - I had some errors I'm working on solving. Errors: typing the commands that were on GitHub. The installation succeeded, but when I tried using poetry to install the development environment, there were some errors. Using a Windows computer with regular Intel process.
        - M: I'm reading the docs, mainly the PlanktoScope software guidelines. Starting with documenting the MQTT topics. I want to know how many

topics will be in the platform, want to know more about the hardware, etc.
        - E: for you, the “software development” process will be writing docs - see https://github.com/PlanktoScope/PlanktoScope/tree/master/documentation#usage
- Hardware drivers (Ethan):
    - Update from Ethan: fixed some small issues reported by Tanguy on Slack last week (no need to discuss): PR 385
    - Potential update from Thibaut: default pixel size for v2.1/v2.3 hardware (based on the old Pi Camera v2 module)?
        - Potential TODO for Ethan to update `hardware.json` config files?
        - T: didn’t get time to do this. Not really a high-priority item, but I might do it when I play around with optics in the next few weeks.
    - **Potentially-detailed discussion** with Melissa & Oumayma (& Thibaut?): what is our plan for rewriting the Python backend?
        - T: this will mostly be done by O. When the three of us (me, M, O) were discussing together, we talked about having them work together on the refactor, and also have M work together with L to ensure the refactor works well with the Node-RED rewrite. We want to have a template we can re-use for different modules. Split the pump and the focus into two separate Python scripts. Also simplify the light module. And then (with M) improve the way we do communication over MQTT: what are the messages, the variables, the units, the ranges of variables, etc.
        - T: another idea to discuss is: in the future development of the PlanktoScope hardware (v2.6, v2.7, etc., and a field-specific version, and a lab-specific version, and an autonomous version), it would be make sense to have loops that are not only asking the pump and camera to work, but also things like the bubbler, UV light, agitator device, etc. So I was wondering about having a MQTT command to just take a single picture from the camera; and the control loop can happen from Node-RED - which could then control the pump, the camera, etc. Then Node-RED could expose an automation interface for advanced users.
            - E: yes, agree. We’ll have a driver layer (including the picamera2-based camera) fully exposed over MQTT, and we’ll separate the “domain logic” into a layer over that
            - T: we could have O first focused on learning the code via the driver layer, or something? Would be good to give O and M a visual way to learn and understand the architecture of the software.

        - M: yesterday we started sketching out the architecture of frontend vs. backend, but we haven't gotten around to sketching the architecture inside the backend
        - TODO for E: update PlanktoScope Software Documentation, and remove some sections which I've transferred into the project docs site. @M+O: this might be a useful starting point for your work on documenting the software architecture
        - TODO for E: sketch a diagram of the software architecture (incl. Python backend architecture) and start some tech ref pages on the project docs site
        - T: after the refactor, we can work on some specific functionalities, and support L on the frontend
- Backend APIs (Ethan):
    - **Potentially-detailed discussion** with Melissa & Oumayma (& Thibaut?): what is our plan for documenting & redesigning the MQTT API?
        - Refer to: PlanktoScope Software Documentation (Ethan's draft documentation of the MQTT API)
        - First step will be to document everything
        - T: from my point of view, we first describe what's actually being used before changing the Python or Node-RED's use of the APIs; make sure to document everything correctly
        - E: need to document differences between the APIs in the adafruithat version and the planktoscopehat version of the v2024.0.0-alpha.1 version of the software. Would like to make the APIs & Python backends converge so that APIs are the same and we just have different drivers depending on a config variable.
        - T: start by describing correctly not only how things should look, but also range for values, units, etc.
        - M: let's review the data sent over MQTT. Don't need to send data every time if the data hasn't changed. Need to reduce the number of topics.
            - E: yes, would be good to consolidate some topics; others may need to be split out
- Any other work packages with updates, questions, or action items to discuss?

Other updates + discussion

- Any other topics to discuss?
    - Ethan/Adam/Thibaut: Manu wants to meet with us re:starting a US entity - do we want to schedule a time to meet together, or will we wait to see how serious he is about that?

        - A: I'm at Autodesk today
        - E: I'll propose to Manu that we meet tomorrow instead
- Review of actionables for next week:
    - O+M: work on understanding what exists, documenting it, and writing down any desired changes to discuss next week.
    - E: a few more docs pages

2024-03-28

In attendance: Ethan, Satoshi, Loïc, Adam, Thibaut, Fabien

Work package updates + discussion

Refer to: Software Work Packages

- Release engineering (Ethan)
    - v2024.0.0-alpha.1 prerelease has been tagged! It fixed some small problems I discovered on the planktoscopehat SD card image after I tested on a PlanktoScope with v2.6 hardware.
        - F: I've asked Pierre to test this, but I'm not sure when he'll be able to - next week he will be sailing somewhere around Western France with PlanktoScope. I personally won't have time right now for testing.
        - S: Are there any differences between the images from last week vs. the alpha.1 images? I've noticed a difference in size between raw images from last year vs. this year. In the old version it was 5 MB, but in the new version it's just 206 kB
            - E: yes - imagernew on planktoscopehat version; "restart segmenter" button; and "pscopehat" -> "planktoscopehat"; and JPEG quality 85 -> 95
            - F: raw image filesize also depends on the exposure level of the image and how clean the background is.
        - S: tomorrow I'll test the new SD card images
        - E: one thing I'd like people testing the new SD card images to check is whether they have to use different ISO/white balance settings with the new imager.
        - Decision: we'll check back in 2 weeks to see how much testing has been done and decide whether to stay in alpha longer or proceed to beta
    - Follow-up on action item from last week: any feedback on the changelog?
        - F: I read it, no edits suggested

        - Tanguy requested a patch to v2023.9.0 to fix some issues he's identified. Is FairScope currently trying to ship the v2023.9.0 software to customers, rather than staying on v2.3 and waiting for v2024.0.0??
            - T: version numbers are a bit confusing to me. I never know which numbers belong to which versions. The last release on GitHub is v2023.9.0; that's the one I'd like to use for production. The known bug from raspimjpeg is annoying, that's the motivation for v2024.0.0; but in some cases it could be nice to expose customers to v2023.9.0 and ask them to follow the procedure to overcome the problem, in order to benefit from the new features added by v2023.9.0. Mostly we're providing the v2.3 software from two years ago. The main problem is raspimjpeg, and also that the hardware settings aren't correct by default - so each time we have to modify them or create a custom OS image specific to the hardware settings we want for v2.6 hardware - we need to ensure we have the correct focal length and objective lens and pixel size calibration.
            - T: following your reply on Slack, we decided to stay on v2.3 software for now. I think it's valuable to find a way to get this current alpha to a beta soon, so that we can test with customers, test in production, and get feedback. If v2024.0.0-alpha.0 can progress to beta in 2 weeks, that would be ideal.
- GUIs (Loïc): any updates, questions, or action items?
    - Refer to: Specifications : PlanktoScope GUI redesign v1.1
        - L: Ethan added some comments to this document, maybe we (@Thibaut) can discuss (this is a TODO)
    - (screenshare of wireframe mockups in Figma for the new GUI design)
        - L: I started with a homepage and some basic info and a "getting started" button, and big buttons for navigating to important pages by scrolling down, and smaller buttons for less important pages. Navigation icon-only sidebar on the left (hamburger menu adds text labels to the icon by expanding the sidebar).
        - L: logs page
            - E: unified logs panel (one log for all python processes), or a separate log for each process?
                - T: I think it could be nice to have just one log; it could also be reasonable to have a different log panel for each Python process
                - L: I could prototype a separate log for each process
                - A: that seems very useful

        - T: maybe we could have separate log panels for exploring, and a single button to download all the logs. For now we can just have controller vs. segmenter logs separate, and later we can decide whether to split up the logs of the various individual controller modules
- L: hardware settings page is mostly placeholder form elements right now, and we can also show some specific information about specific hardware components in the PlanktoScope.
    - T: the hardware info would be stored in EEPROM in the PlanktoScope HAT; and Node-RED would display the hardware info
    - F: what is the plan for backwards-compatibility with PlanktoScopes which lack the EEPROM or the relevant info? Could those things cause compatibility issues?
        - E: we'll have to show "unknown"
    - T: the vision is to provide to EcoTaxa the hardware configuration as metadata; incl. LED model, flowcell, etc.
- L: sample metadata page is a very rough draft - not sure I understand the requirements for this fully yet.
    - T: Loic will focus on the GUI layout & ergonomics; other students will focus on the Python backend rewrite.
    - T: for location, all timestamps should be shown in UTC, rather than local timezones.
    - F: I agree. The GPS module provides UTC times everywhere, too. With UTC time, latitude, and longitude, you can directly calculate where the sun is in the sky (e.g. solar noon, dawn, dusk)
    - F: In my point of view, there should be a sample metadata page which visualizes metadata about a sample, and visualizes the metadata which have been entered. In practice, if I ask 10 scientists to enter the same latitude, longitude, volume filtered, etc., I regularly have ~10-15% rate of data entry errors. There is a great interest to let scientists visualize their metadata and have the possibility to correct their metadata. If this page is just about visualizing their metadata, we do eventually need a way to correct the metadata.
    - L: in the frontend and/or backend we can have a function to check the validity of each input element.
    - F: even when the input format is correct, people have entered numbers incorrectly (e.g. 10.2 instead of 11.2)

            - F & E: having a way to edit/correct the metadata later would be helpful
            - T: this page is just for pre-acquisition stuff so far.
            - T: for editing metadata afterwards, we just need to edit the metadata.json file
                - E: also may need to update the file integrity checksum for metadata.json in the integrity file
                - T: the segmenter checks integrity of the images, at least
        - L: I still need some more information before I can finish the wireframes.
        - T: we got some responses in the feedback form, will need to review that feedback for ideas. Most people who filled out the form are happy with the current GUI.
        - T: once the current alpha is stabilized, it'll be useful to work on deployment and documentation, and the version of the machine we're working on will be stable this year and we'll improve various annoyances in the current version.
        - L: I will probably keep the new GUI similar to the previous GUI, for users who find the previous GUI intuitive
- New products (Thibaut): any updates, questions, or action items?
    - T: we're working on v2.7 hardware. Ideally it'd be linked to the release of the new software. For the hardware, we're writing up specs on the HAT to incorporate what's in the v2.6 hardware but also support transition to RPi 5. Mostly the difference will be for the HAT to power/control the bubbler directly. The pump and LED will probably not be connected directly to the HAT but instead to a peripheral PCB, to simplify wiring. And some improvements to electronics.
    - T: we may also allow different fraction sizes, both for smaller objects (e.g. HABs) and larger objects (e.g. oyster larvae)
    - F: question about wiring of bubbler: our current bubblers are pretty fragile. Will bubblers still be replaceable/interchangeable?
        - T: we may switch to some other bubbler - the current one is very cheap ($5), and we aren't using most of its provided functionalities.
    - F: timeline of release of v2.7 hardware is around November. It will be an incremental improvement, rather than a dramatic change
    - Use a action organizers such as Trello or Notion to distribute actions to keep track of the development FairScope/PlanktoScope
        - T: this is something we'll do internally at FairScope.
        - E: for software we'll stay on GitHub, we can link/cross-reference/copy things as needed.

- Docs (Ethan): no updates this week; planning to add some more technical reference pages this week
    - Since we're making v2.6 the default hardware config for the planktoscopehat SD card image, I'd like to add v2.6 CAD files, fabrication files, and assembly instructions to our GitHub repo and online docs site, as part of the v2024.0.0 release. Who should I work with on this - would it be Tanguy?
        - T: sounds fine to me; it's Tanguy's work to make documentation accessible. We do need to be able to edit it correctly and release it iteratively (to add more content, etc.). There's a separation between the docs software on GitHub and our internal processes. We have to find a way here. One way could be to have that internally to FairScope (a website we could edit easily). Then it would be one place for software, another place for hardware.
        - E: I'd be fine to start with a simple/stupid approach for uploading hardware docs with the v2024.0.0 docs, but just as a way to discover what the best process would be for us.
        - T: operation of the machine is not really documented - it really is the protocol which Fabien has written with Pierre. It would make sense to have that on GitHub, or should that stay separate? For Fabien it's probably easier not to have to deal with GitHub. GitHub is specifically for the software…
        - F: The nice thing about protocols.io is that it kind of works like GitHub; it allows versioning of protocols. I would prefer to continue to make new versions on protocols.io - so that when people land on old protocols they get notified of new protocol versions; this is not necessarily the case on GitHub. This week I had some trouble trying to direct someone to a page on the PlanktoScope - there are many versions, and it's hard to find the list of materials. Need to go deep in the GitHub to start to find something. It gets confusing because there's the old one, unofficial one, official one. Navigating through things is very complicated. We start to use people to follow protocols.io.
        - E: yes, we just include copies of PDF things (e.g. from protocols.io) in the docs site; we can continue doing that, and we can do that for v2.6 hardware assembly instructions.
        - F: protocols.io shows/includes metadata for protocols - including a DOI, and direct link to PDF and HTML exports.
            - E: yes, that's what I used.
        - E: I'd like to keep protocols.io around as a separate authority of information. Our software docs site will always only describe the latest

version of the software, while protocols.io is a great reference for info about older versions of software.
        - F: protocols.io will also provide other information on operation, e.g. pointing out differences from capabilities/usage of FlowCam
        - Decision: for now FairScope will have its own docs site with v2.6 hardware documentation, and PlanktoScope project docs site will just include a link to the FairScope website. We will revisit this in the future.
        - TODO for E: add a GitHub Issues item to track integration of v2.6 hardware files/docs into our Git repo
    - T: Can we delete https://planktoscope.readthedocs.io/en/latest/ ?
        - TODO for E: migrate v2.1 hardware assembly docs from https://github.com/KonkArLab/PlanktoScope/tree/master/docs, and do the same for the "Get your kit" page from planktoscope.org
- Image acquisition (Ethan): no updates this week; no action items allocated to this week, but next step will be to continue incrementally rewriting the hardware controller
    - S: I tested the global shutter camera and it works well!
    - T & E: (discussion about how to proceed with rewriting the other hardware modules; decision: E will work together with incoming students who will be rewriting the hardware modules; the new `camera` and `imagernew` subpackages are representative of how I will try to make code organization clearer)
    - T & E: (discussion about working with incoming students to document and redesign the MQTT API; this should have a design document)
    - T: there are two incoming students who will work on backend, they'll attend the meeting next week. TODO for E: add agenda items to discuss these things next week.
    - T: Verification of the hardware.json : 12mm lens -> 0.75um and 16mm lens -> 0.88um
    - F & E: (discussion of camera settings & image metadata newly exposed by the picamera2 library)
    - T: I've set some target to improve the optics, and to test different camera settings with the picamera2-based imager, and to test different lenses. Current lenses are $5 each, so I ordered lenses with the same spec but better quality (~$100 each). This may break compatibility with other datasets, but may be useful for certain applications. Also looking into dependence of optimal white balance gains vs. lenses, to be sure we have a good way to calibrate in the factory. Will present results after I get results.
    - TODO for T: check the pixel size calibration for v2.1/v2.3 hardware (using the old Pi Camera v2 module)

- OS (Ethan): no updates this week; no action items allocated to this week, but next step will be to continue working on the migration to Raspberry Pi 12 (bookworm) and 64-bit OS
    - T: Non-standard installation of the Latest stable isn't working from Raspberry Pi OS (64-bit) - Desktop using the documentation (x Command failed (exit code 128): git clone --mirror --quiet --filter=blob:none https://github.com/PlanktoScope/PlanktoScope /tmp/tmp.BtV3EZF0Eu)
        - E: yes, we can't install on 64-bit RPi OS yet. For now it has to be the RPi OS 11 (bullseye) 32-bit OS
        - TODO for E: in https://docs.planktoscope.community/setup/software/nonstandard-install/, make the information about the required base OS (i.e. bullseye/11, and 32-bit) more visually emphasized (e.g. with bold, or adding a warning box)
    - TODO for E: improve the documentation's explanation of master vs. beta vs. stable branches
    - TODO for E: remove balenaEtcher instructions from the non-standard install docs and from the standard install docs
- Any other work packages with updates, questions, or action items to discuss?
    - T: now there are many repos on GitHub and it's hard for me to navigate.
    - TODO for E: add a GitHub issue for developer documentation (either on the docs site or in repos) to make repos easier to navigate and find whatever we're looking for, and with better organization of code. Maybe add a README in the PlanktoScope repo's software directory with links for how to navigate the PlanktoScope repos on GitHub.

Other updates + discussion

- Any other topics to discuss?
- Review of actionables for next week:
    - T + E: (discussion of how we keep track of action items & TODOs)
    - (various TODOs listed above, and various things we've been working on which we plan to keep working on)

2024-03-21

In attendance: Ethan, Adam, Loic, Satoshi, Wassim, Thibaut

Work package updates + discussion

- GUIs (Loïc)
    - L: No work on GUI this week. Was working on the English translation of the specification doc: Specifications : PlanktoScope GUI redesign v1.1. Feel free to review and add comments - if there's anything you'd like to add to the specifications, add a comment on the document for that.
    - L: shared the UX improvements feedback survey (https://e9135syb3ri.typeform.com/to/tncY6Pyg), open for anyone to give feedback. It will guide the specifications and the design of the next GUI. We sent this survey to all FairScope customers.
        - TODO for everyone: fill out the survey when you have time!
- Targeted ML-based classification (Wassim)
    - W: Today: trying out YOLOv9-c (previously we'd tested YOLOv8-nano, which is a smaller nano), with fine-tuning on oysters dataset. YOLO-v9 was overfitting compared to YOLO-v8 (also fine-tuned on the oysters dataset). We need to mix multiple objects in the same dataset - to make the training set more representative of real-world datasets. Right now our training set is just at most one object per image.
    - W: I intend to test some Pytorch models. Plan to get EcoTaxa datasets tomorrow, try some classification tasks, note the training time & inference time. This will help us to compare approaches.
- Image acquisition (Ethan)
    - device-backend PR 19 (picamera2) + PR 380 major progress update:
        - Ethan: I finished the rewrite and tested everything (without plankton samples) on an adafruithat-based PlanktoScope. I need someone to do basic testing on a pscopehat-based PlanktoScope (i.e. can you change camera settings? can you perform image acquisition? are there any obvious problems with the camera preview?), and then we can merge these PRs and do more extensive testing afterwards. SD card images for testing: https://github.com/PlanktoScope/PlanktoScope/pull/380#issuecomment-2011401427
            - S: I probably can
            - TODO for E: get a pscopehat-based PlanktoScope for testing
        - Ethan: I've removed the cap on MJPEG stream framerate, so it adapts to the maximum rate supported by the camera + the RPi + your shutter speed setting + your network connection + your web browser (the network connection should usually be the bottleneck). This will make the camera preview smoother, but it may also increase CPU usage on client devices (e.g. phones). Is this a concern?

        - (no concerns)
    - Ethan: I also changed the filename format of saved images (and thus of object IDs for EcoTaxa) from `{hours}_{minutes}_{seconds}_{microseconds}.jpg` to `{index}_{year}-{month}-{day}_{hours}-{minutes}-{seconds}-{microseconds}.jpg` ; are there any concerns about this change?
        - (no concerns)
        - T: let's leave index out
        - TODO for E: remove index from device-backend PR 19
    - Ethan: what JPEG quality level should we use for saving images? raspimjpeg config file suggests we've been using 80 (for a range from 0 to 100), and that setting was probably shared between saved images and the MJPEG stream; picamera2 library defaults to 90 (for image files, and for a range from 0 to 95), and the MJPEG stream quality is controlled separately
        - Thibaut: disk usage doesn't matter so much. We do want to maximize quality for now. If this causes problems in the future, we could optimize. Let's go for better quality. We'll need to ensure the segmenter doesn't crash.
        - Decision: let's go do max quality with JPEG for now, and test it out with real samples.
    - Ethan: I'm making an executive decision that all new Python files in the hardware controller need to match a standard code style (the command to automatically format files is in the readme) and to pass type-checking and other linting checks (for code complexity, possible errors, etc.), to help with code correctness and maintainability; starting with device-backend PR 19, these checks are automatically run/enforced in the CI as part of the PR merge requirements. Thus, as we replace each hardware-control Python module with rewritten versions (the rewritten versions must pass all checks), we will incrementally bring the codebase into adherence with the standard style and these automatic checks.
- Release engineering (Ethan)
    - Ethan: Moved all remaining bugfixes out of the v2024.0.0 milestone except for the picamera2 migration, since we now understand that the raspimjpeg camera timeout error is a very big problem in v2023.9.0. The other remaining tasks for v2024.0.0 are documentation improvements, which I can do during beta testing for v2024.0.0, or I can delay them for the next release if I run out of time.
    - Ethan: So after we merge PR 380, I'll release v2023.9.0-alpha.1 for more extensive testing of the new imager, and we'll go into feature-freeze for beta (i.e. only smaller bugfixes can be added afterwards). After we've had time to test

v2023.9.0-alpha.1, we could do v2023.9.0-beta.0 as a general beta since v2023.9.0 is so broken due to raspimjpeg. We won't make a decision on v2023.9.0-beta.0 until after we've gotten testing feedback on v2023.9.0-alpha.1, but does anyone already have any concerns/questions with this proposed plan?

- Ethan: what is our plan/timeline for totally deleting raspimjpeg and the old imager module? I propose doing it after the stable release of v2024.0.0. Thoughts?
    - (no strong opinions)
- T: white balances may need to be recalibrated for picamera2. Fabien has made an iterative loop to adjust white balance. We've removed the IR filter, so things may look somewhat more pink in the previous manually-calibrated white-balance values. It would be useful to have some calibration for the white balance. This would be good to have an intern do.
    - T: Tanguy tried doing the iterative optimization of white balance gains over 4 loops. We were looking at the saturation level in the resulting images and trying to minimize that. We were able to get a low saturation (<5% is good, <2% is perfect)
    - T: ISP in the RPi may be able to enable the camera compensate for the removal of the IR filter.
- Ethan: Drafted a deprecation notice for the PlanktoScope changelog & release notes of v2024.0.0-alpha.1, need review/feedback for any revisions: https://github.com/PlanktoScope/PlanktoScope/blob/159fac7153a7bd27fb02f40358ad70823ad83cd3/software/CHANGELOG.md#deprecated
    - T: Let's not use "pkscope" or "pscope" as abbreviations for "PlanktoScope". "pkscope" is fine for the SSID. Let's use "planktoscopehat" instead of "adafruithat" for all future SD card releases
    - S: I still want to work with the adafruithat
        - E: my plan is to have the Python backend provide support for adafruithat and planktoscopehat, and we'll have one version of the new Node-RED dashboard which doesn't care what HAT it's running on.
    - TODO for E: update the SD card setup scripts (and the non-standard install docs) for this.

- Docs (Ethan)
    - Ethan: Cleaned up (and added some more context on the left side to) Software Work Packages
        - S: I may be able join the automated sampling project
            - T: it's not a current project, we'll progressively work on it. But we plan to start this project eventually. If you already have some ideas before FairScope starts working on it, that'd be great.

        - S: I could also help with the Docs or Public communications work packages.
    - T: I'd also like to translate another document from French to English: specs document for PlanktoScope HAT v2.0
- OS (Ethan)
    - Ethan: No work & no updates this past week
- New products (Tanguy/Thibaut)
    - Thibaut: PlanktoScope HAT v2.0 will be designed for use with the RPi 5 (which already has a cooling fan module). We're not yet sure how we'll integrate with the RPi's hardware RTC support. We want to control the bubbler, add better control of the LED, figure out EEPROM.
    - T: we were looking for driver boards/chips to control the pump. Right now the library we use (SlushEngine) is made specifically for one Trinamic motion controller chip, so we only have one source of driver boards/chips right now. It'd be great if we could have a Python library which can control more boards.
        - E: should be fine to choose a different driver chip, assuming it has a Python library which we've tested and confirmed to provide the required functionalities.
- Any other work packages to discuss?

Other updates + discussion

- How do we feel about our experiment with organizing the meeting agenda around work packages: do we want to continue this experiment? Are there any modifications we want to try out for next week?
    - T: this meeting had lots of updates, not much about actions. Could be good to take 10 minutes at the end or during the call to distribute some actions between now and next week. e.g. reviewing the changelog could be one action. We can construct actionables throughout the meeting, and compile a list at the end
- Review of actionables for next week:
    - TODO for everyone: review the changelog: https://github.com/PlanktoScope/PlanktoScope/blob/ca470275fbc6360235b78cdebbb503213e565f0a/software/CHANGELOG.md#unreleased
    - TODO for everyone: fill out Loic's survey at https://e9135syb3ri.typeform.com/to/tncY6Pyg
    - E: publish the v2024.0.0-alpha.1 release, make announcement on #software channel
    - E: might add more tech ref docs

Proposal review

Refer to the Proposals project board on GitHub, and our description of the proposals process. (no time to go over proposals today)

Review of proposals in the Final Comments column:

- (none)

Review of proposals in the Under Review column:

- Laurent: Proposal 290 (still on hold, Ethan's too busy to do an iteration of edits/improvements to the design document for this proposal)
    - Ethan: after working on device-backend PR 19 (for picamera2), I identified a bunch of changes I'd like to make to the hardware config file: adding some more fields, removing some fields, changing the representations of values for some fields. I also identified some questions we'll have to think about: https://github.com/PlanktoScope/PlanktoScope/issues/290#issuecomment-2011217230

Review of proposals in the Submitted column:

- (none)

Next steps for proposals in the Draft column:

- Laurent: Proposal 292 (still on hold, not a high priority for us)
- Ethan: Proposal 307 (still on hold, Ethan's been too busy to update the design document based on what we've learned over the past month of testing)
- Ethan: Proposal 320 (new proposal)
- Ethan: Proposal 377 (new proposal)

TODO: post an update about our decisions from this meeting, to https://github.com/PlanktoScope/PlanktoScope/issues/282

2024-03-14

Note: this meeting is at 8:30 AM PDT (one hour later than in previous weeks in the US) / 16:30 CET (same time as before in Europe) today because the US has already transitioned to Daylight Savings Time, but Europe does not transition until the end of the month.

In attendance: Ethan, Loïc, Satoshi, Wassim, Thibaut

Individual updates + follow-up discussion

(note for ML-related updates: if anything needs to be changed in the PlanktoScope software to support ML-related activities, please add agenda items for those things! Any other updates related to ML should be added to 2024-01 to 2024-06 AI Meetings to be discussed in tomorrow's ML meeting instead of this meeting)

- Satoshi: had good pictures but problems with segmentation on the recent cruise (in East China Sea, near Okinawa): for diatom chains (*Climacodinium*) with very transparent cells: after segmentation, the diatom chains are split up into many smaller objects by the segmenter
    - E: workaround for this would be to increase the value of the "dilate" parameter in the segmenter
    - T: what lenses did you use?
    - S: I tried on 25 mm with 16 mm, and 25 mm with 12 mm; both hardware v2.1 and hardware v2.6
    - T: it would be helpful if you can upload the raw images to share, which may help other people test out other segmentation algorithms
- Thibaut: planning of priorities for possible tasks to give to the students who will soon be working at FairScope
    - T: refer to 2024-02 Software Work Package Brainstorming
    - T: these are work packages for the software, in medium-term or long-term. Each needs one person to be in charge. Our draft is already in good shape, so let's discuss who will lead which work packages.
    - T: we'll have new people joining in 2 weeks, let's think of subjects they can be involved in
    - (we did work on the document linked above)
    - (we decided to move ML-based data exploration & visualization into the "planned projects (not yet a priority)" category, since we aren't prioritizing anyone to lead those projects yet)
        - W: those can be future internships
    - T: for now the highest priorities are to get our existing software into a more solid place
    - E: What is the meaning of "team lead"?
        - E: ensuring that priority work items have people working on them
        - T: leaders gathering in these weekly sync meetings, providing updates on work packages; communicating with other teams. Non-lead team members don't necessarily need to give updates to other teams. Members of teams can communicate internally among themselves, help do work, meet in other meetings; and they may need to coordinate

    - (we got side-tracked and didn't finish this topic yet)
    - (we returned to this topic and identified active vs. inactive work packages, where active work packages are indicated in green, and we finished assigning leaders to the active work packages)
- Loic's dev updates
    - I've been working on a personal repository on GitHub
    - E: let's move it to the PlanktoScope organization on GitHub so that it's easier to find (fine to keep it a private repository, or to make it public and add a note in the readme that it's still an early experiment)
    - (showed demo of defining a custom node to list files in a directory, and showed some graphing nodes)
    - Summary: I've been trying various things to understand how Node-RED works
    - T: regarding this new version of node-red-dashboard: it seems to be easy to navigate and is still actively maintained (unlike the previous version)?
    - L: different nodes, but everything works approximately the same way
    - L: in a meeting with T, we discussed what we wanted for the next version.
        - T: this is the first month of a 6-month internship. First month will be about defining the spec of the next GUI. The first meeting was for me to share my vision for that. The idea now is to get initial feedback from other people in the community (devs, super-users) on what they'd like to have for big functionalities.
    - E: are we doing a total rewrite?
        - L: the node-red-dashboard and node-red-dashboard-2 are on different URLs, so they don't overlap. I can adapt the past version to the new version.
        - T: so you (Ethan) shouldn't spend time on making any changes to the existing Node-RED dashboard. We might keep the embedded filebrowser at first, and eventually work on improving it, etc.
    - S: what is the intended use of the graph nodes?
        - L: that's just an example of what we can do with Node-RED, not something we've decided to use for the next version. For now I'm just trying out things.
- Ethan's dev updates
    - Fixed the problem Wassim was stuck on with the picamera2 implementation: https://github.com/PlanktoScope/device-backend/pull/19#issuecomment-1993002421 . Now that the blocking problem is solved, does Wassim have time to continue working on this pull request, or do we need me to finish this pull request?
        - W: did you manage to see the camera preview on the dashboard?
            - E: yes

    - T: ideally we finish picamera2 migration before working on segmentation. This could be a task an incoming student could start working on?
    - W: many of the TODOs in the current version of the picamera2 module are already replicated. In my last push to the PR, I finished all methods of the imager process. The same is true for the picamera class. If we merge those changes together, that should fix
    - W: we already started preparing our end-of-studies internships, so going back to picamera2 will be a challenge for me
    - E: let's just have Wassim push up any unpushed work, and I'll take over fully
    - W: I just reviewed the code for the PlanktoScope HAT and the Adafruit HAT; I found that we don't have the same level of progress in both of them. Everything is complete in PlanktoScope HAT, but some missing methods on Adafruit HAT
    - TODO for E: merge my fixes for Adafruit HAT imagernew into PlanktoScope HAT imagernew, and then merge the remaining methods from PlanktoScope HAT imagernew to Adafruit HAT imagernew
- Email correspondence from Katie Crider / Margaret Mulholland (PlanktoScopes for HABs) about running the segmenter on an HPC cluster. To support this, we may need to refactor the Python segmenter so that we also have the option to launch it as a script from the command-line (with folders to process specified via command-line args), as an alternative to launching it as a server/worker which requires an MQTT broker and a Node-RED dashboard or MQTT client to generate commands for the segmenter. This change would greatly simplify running the segmenter on other computers for batch processing; the proposed change is now tracked as issue 378. How much priority (if any) do we want to give to this, compared to other tasks in our v2024.0.0 milestone?
    - T: if we do some work on the segmenter, why not re-do it from scratch?
    - T: I think other work will be higher priorities in the segmenter work package. I personally would prefer to prioritize trying other segmentation algorithms. Maybe we can leave this for incoming students to work on.
    - E: I personally can't commit the time to rewrite the entire segmenter from scratch - that's above my risk tolerance for code rewrites, given that I am just volunteering on this project and it's not my main PhD project. I can do incremental rewrites.
    - S: it would be useful for me to have this - because I want to do automatic sampling every hour, and it would be nice to have automatic command-line pipeline. But not a high priority
    - (consensus: we will put this on the backlog, but we won't prioritize it)

    - Docs site: in the “Technical Reference” section, created tech specs pages for the hardware and the software. There’s some missing (or potentially incorrect?) information in the hardware specs - would be helpful if people can review it and send me any additional information or corrections (e.g. via Slack DMs).
        - T: I’ll pass this to Tanguy
    - Added new draft proposals (see next section of agenda)
- T: we also have internal FairScope software meetings in French. How to coordinate with you (Ethan) and the rest of the community?
    - T: maybe we can use this work package structure to structure the next software meeting (where each leader has a few minutes to give updates on their blocks)
    - T: we’ll keep FairScope meeting notes in French. But we need to figure out how to have the English meetings not just be updates from us, but also input & decision-making.
    - T: we haven’t talked to you (Ethan) yet about the specs I discussed with Loic. And the vision of what the user will be doing on the software. The user needs to take high-quality images with good metadata; different modalities for use of the machine. Make it simple, nice, fluent, quick; but also powerful and complete when needed. Files related to the users, files related to the protocol (e.g. the user uses the machine differently at different times). Would be nice if people could define their own protocols to use the machine more easily. It’s about the vision as well, but we need some time to brainstorm that correctly.
    - T: 1-hour weekly meetings are nice, but maybe not enough time to do brainstorming
    - S: I’d be interested to work on the community area for the PlanktoScope project, but I’m busy with my other work so I can’t commit to doing much work on PlanktoScope
    - TODO for E: for next week’s meeting, I’ll try out structuring the meeting agenda based on the work package areas
        - T: only include things for the “green” (i.e. active) work packages
    - T: I’ll give you access to the FairScope software meeting notes:
      Cahier des charges
        - E: I can read things via Google Translate
        - Cahier des charges
        - TODO for E: review this document
- Other TODOs before next meeting:
    - T & L: English summary of Cahier des charges. Discuss how to request open-ended input from other people (E, S, Fabien)
    - (discussion of some questions we might want to ask in a survey)
    - TODO for E: add space in agenda for Tanguy to present about the planning about the next product

        - T: David is working on enabling plastic flowcells to be 3D-printed with transparent resin, and 3D-print a better sample intake container
- T: there's also going to soon be a HAT compatible with RPi 5. We'll have a full-time intern on that. For now we're testing the intern with a small PCB which will be used to mount the LED more securely. We'll also use that for interfacing with the pump and the bubbler.
    - S: RPi 5 is starting to be sold in Japan now!

2024-03-07

(this was originally planned as a meeting for 2024-02-29, but it was rescheduled to 2024-03-07 because Ethan didn't wake up in time for the meeting on 2024-02-29)
In attendance: Ethan, Adam, Morgan, Fabien, Loic, Thibaut (joined partway into the meeting).

Individual updates + follow-up discussion

- Morgan: tested different model sizes on Raspberry Pi 5 with TPU. It took 5-34 minutes; with TPU, N (nano) model is usable; 1000px images can't be used because the Google Coral on USB just broke (it disconnected, had to reconnect USB manually or reboot RPi; sometimes it works, sometimes not - because of inadequate power over USB). Will test Google Coral on PCIe. YOLOv8 on CPU worked. Hailo-8 (spelling?) TPU; it also accepts models beyond TF-lite - e.g. Keras; it's directly on PCIe.
- Debrief of discussions among Adam+Ethan+Thibaut from OSM24
    - Things we learned at the conference about unsolved problems which users have (e.g. a unified database/platform for data from all instruments; data visualization functionality beyond what EcoTaxa provides)
        - Adam: highest priority is the segmenter. Some PlanktoScopes were sold with a 2 GB RPi, which crashes the segmenter (this happened to me at OSM).
        - Fabien: having an external segmenter will help, but also important to make sure we can run the segmenter on-board.
        - Fabien: I heard you also encountered the raspimjpeg crashing problems a lot at OSM. I've seen it on 1/3rd of my PlanktoScopes
        - Ethan: (recap of request for unified dashboard, interoperable, dataviz, etc., from Margie and others)
        - Fabien: initially EcoTaxa was only designed to sort images for export to other workflows for dataviz. Downstream calculations are dependent on experimental protocols, so people have to be pretty involved in such calculations (example visualization notebook: https://planktoscope.slack.com/files/U0149AST9CL/F06NTFPGJ9F/lab_i

nstructions-answers.html). We'll try to keep EcoTaxa focused on sorting images, to keep a manageable scope.
    - Long-term software development roadmap
    - Plan for software refactoring to improve simplicity & maintainability
    - Draft of a map of software-related work/activities, dividing it into working groups and/or work packages: 2024-02 Software Work Package Brainstorming
        - T: two other students will start to work on software. Maybe we can prepare figure out possible priorities for things they could do.
        - T: I've designated Loic as leader of the GUI work package, and Morgan for the on-board EcoTaxa equivalent
    - Tentative discussions about setting up a non-profit (based in the US?) which would be responsible for various parts of the existing PlanktoScope software, and also for new software (e.g. a cloud platform??)
- Ethan's dev updates:
    - I now have a working prototype for running the segmenter on a separate computer: https://github.com/PlanktoScope/pallet-segmenter . This includes a stripped-down Node-RED dashboard app which only has the page for the segmenter, and an instance of the MQTT broker which the segmenter needs to function (and which bridges the segmenter with the Node-RED dashboard).
        - A dataset (which I had collected on R/V Sikuliaq in the Arctic last year) of 400 raw images which previously took ~2 h to process on the RPi4 now takes ~6 min on my laptop (AMD Ryzen 7840U CPU, with base frequency of 3.3 GHz) in power-saving mode!
            - This is a ~20x speedup, which makes it feel kind of absurd to run the segmenter on the RPi whenever a perfectly good laptop with Docker (and maybe also an x86-based CPU) is also available…
            - This also suggests that real-time segmentation is feasible if we off-load segmentation to a separate computer with enough CPU power, since the PlanktoScope had taken 20 minutes to do the full stop-flow acquisition of the dataset.
            - Fabien: another improvement for real-time segmentation is that instead of doing a running median of three raw images, instead do one median every 50 frames, keep that background every 10 images, and then generate a new one. This is what FlowCam is doing - it's fast because they don't regenerate background images for every single frame. Median image calculation is a performance bottleneck, lots of I/O to RAM, etc.
            - TODO for Ethan: add a GitHub tracking issue on background calc performance, as part of the throughput tracking issue

- Opinions requested: should we change the “Segmenter” page in the main Node-RED dashboard to just embed this segmenter-only dashboard page (like what we do for the File Gallery), or should we leave the existing Node-RED dashboard as-is? For now, I’m leaning towards the latter just to keep the UI layout less weird, but the former option would make it easy to just clean up the segmenter once and have the new version available to both the adafruit and pscopehat versions of the Node-RED dashboard…
    - Fabien: I have no concerns, as long as there’s a direct button to go to that specific page directly from the dashboard (incl. the sidebar)
    - E: yes, we would integrate it like how we integrate the file gallery.
    - F: what improvement will it provide to do this?
    - T: we might as well leave the dashboard as-is.
    - (rough consensus: only make a change if it’s annoying to E in the future)
- T: it’s great that the segmenter exists. Seems like more resources are needed to improve the segmenter, make it more configurable from the dashboard by advanced users, etc.
    - E: splitting the segmenter into its own embedded app would make it easy to make a simple-user version of the segmenter and an advanced-user version of the segmenter, and swap them out independently of the rest of the dashboard
    - T: in addition to Loic, we’ll have some more interns working on GUI improvements, e.g. for the segmenter, and for image acquisition (e.g. autofocus)
    - F: focus can be a problem sometimes - I tried to focus too much on one specific object, which left me away from the middle of the flowcell. Best option is to have a semi-wet flowcell and get both the front and rear plane equally blurred.
    - T: could be nice to have flexibility for the dashboard and software to have these kinds of small functions easy to implement
    - F: also one problem is when the flowcell is tilted and not parallel to the camera. Making it easier to focus can be helpful, but it can be a trap for people which makes them too confident in the automatic stuff but with worse data quality because they didn’t check things like flowcell alignment.
- Pre-proposal discussion: right now the segmenter uses “simple_threshold” rather than “adaptive_threshold”, with 127 as the threshold value. This is why the segmenter started failing on datasets collected after a month on the Sikuliaq

(because the raw image brightness gradually decreased over the span of 1 month due to some kind of gradual hardware failure) - refer to Ethan 2024-01 to 2024-06 for sample images. Do we want to continue using "simple_threshold" (but maybe make the threshold parameter adjustable by the user, and maybe even estimate a recommended threshold value for a dataset with some simple histogram approach, or else provide live GUI feedback to the user in the "Optic Configuration" page showing the live brightness histogram of the camera preview vs. the fixed threshold value of 127?), or do we want to switch to "adaptive_threshold", or do we want to do something else entirely?

- Fabien: I'm surprised you didn't try to increase the ISO - frog-in-the-pot problem.
- Fabien: I think the hardware failure is probably with the LED.
- Fabien: in the long term, there's a lack of tools for adjusting imaging parameters. If there were a live calculation and UI feedback of mean R/G/B values in the images, that would've given you feedback about the brightness of the images. That would've allowed you to identify the problem earlier. And getting that information isn't that complicated - without having to generate a histogram.
- Fabien: this would also help people to adjust their white balance - after all, all the white balance calibration and ISO adjustment and scotch tape is targeting the mean R/G/B parameters. Optimal mean value is ~245 when the LED is on.
- TODO for Ethan: add a GitHub feature request (or draft a proposal) for this.
- Fabien: as for thresholding, I got to the simplest solution of simple thresholding and figured out how unstable it was; then I moved to adaptive threshold. Advantage of adaptive thresholding is that it allows together to segment stuff that could be transparent - transparent objects are saved because they look for gradient of change in the image.
- E: I think the PlanktoScope's adaptive thresholding is global (e.g. Otsu's method)
- F: I tried some other thresholding strategies, they didn't work that well (e.g. canny edge, sobel)
- A: if we have better quality control of image acquisition (e.g. with feedback on R/G/B mean values), that should be enough to address the problem.
- F: I also had this issue with the LED getting dimmer and dimmer and dimmer!

- Does anyone have questions/comments about: segmenter performance measurements on 32-bit vs. 64-bit OS and RPi4 vs. RPi5 vs. x86_64

    - F: the 32-bit vs. 64-bit OS change - is that just a pure performance improvement for everyone? vs. RPi5 requires a hardware change, generates versioning issues
    - E: right
    - M: RPi4 & 5 are same size & same OS; we can have 64-bits on either.
    - A: is it possible to run segmenter on 2 GB of RAM?
    - F: it was possible until we moved to the HQ camera; that broke our ability to run on 2 GB of RAM, because 5x image size in RAM for the 3-frame median was too much
    - E: Thibaut will just have to send 4 GB RAM RPi replacements to people
- If you have any concerns about the following changes, please provide feedback: device-backend PR 20
    - F: one problem besides this is that in EcoTaxa in the same project, you cannot have two objects with the exact same name; or two acquisitions with the exact same name; or two samples with the exact same name. Sample IDs need to be unique names, and acquisition IDs need to be unique names. Usually we have the sample name which is repeated in the acquisition ID. The image name usually is supposed to have a name which includes the sample ID and acquisition ID. Technically, also the TSV file is supposed to have a unique name as well. If we do have that, then when we load stuff in EcoTaxa, we can just have it import everything in a folder; or update everything except things we've already imported. Our lack of TSV files prevents us from doing that, and instead we have to import samples individually.
    - F: because the PlanktoScope has a broken system timekeeping across reboots, people have had problems importing in the same project because two images were acquired coincidentally at the same system time.
    - Morgan: RPi5 has RTC
        - E: yeah, that'll help
    - F: we don't necessarily need the timestamp, but it was basically inserted to just ensure that we don't get two people who produce two "sample 1"/"acquisition 1", and to disambiguate between those things.
    - E: we could add timestamp to the end of the EcoTaxa zip filename
    - TODO for E: check with Thibaut if he's fine with me doing this.
    - (Thibaut joined the meeting later, so we returned to this discussion topic)
    - F: within EcoTaxa, the acquisition ID..basically you can't have two acquisition IDs with the same name; so that's why we put project ID and sample ID into the acquisition ID for EcoTaxa. Similar story with sample ID (so we put project ID into the sample ID for EcoTaxa).

        - T: I don't think the date is so good because the dates are often incorrect
        - F: ideally it should be the date of sampling itself - we want to be able to get back whatever sample we collected on a certain day. If the timestamp is instead for the time of image acquisition or of segmentation, that's not so useful. But timestamps are worse than useless if the times are incorrect.
        - E: existence of multiple possible timestamps can be confusing if we don't specify which one in the filename…probably simpler to just not have timestamps in the filenames
        - (rough consensus on this)
    - No need for discussion: device-backend PR 22, device-backend PR 21
    - My priority for this upcoming week will be to make progress on device-backend PR 19 (picamera2 migration).
- Morgan: Loic is here to work on the GUI
- Loic: we'll keep Node-RED for now. This week I was searching for frameworks as alternatives to Node-RED, e.g. openHAB (Java, simpler than Node-RED for prototyping). But when you try to make something more difficult, it's really hard. And if we move to an alternative, we have to do more work on e.g. connecting to Python.
    - Loic: I found a new library for the dashboard (node-red-dashboard-v2). I started to try it out, it's better for the design, more responsive, more customizable. I'll continue trying it out.
    - Loic: I looked at integrating custom nodes on the new dashboard, it's really simple - I just do it in JS on my personal project, pack it with npm, and send it to Node-RED.

2024-02-16

This is an in-person meeting between Thibaut, Adam, and Ethan

Long-term plan

- Some software-related goals for in-person work:
    - Clarify the long-term roadmap for how we will evolve the software, and for new major functionalities we will add
    - Identify some options for how the new frontend interns will approach their work
- (discussion + document work: 2024-02-16 Software Work Package Brainstorming)

2024-02-08

This meeting is planned to be 1 hour long (instead of the usual 30 min long) because we will have a deeper discussion.
In attendance: Ethan, Thibaut, Adam, Fabien, Wassim, Morgan

Individual updates + follow-up discussions

- Ethan's dev updates:
    - v2024.0.0-alpha.0 prerelease (for OSM booth demo)
- Fabien:
    - Tried to reproduce previously-reported bugs (loss of all metadata after trying to calibrate the unit) on a unit. Have not yet been able to reproduce those bugs.
    - A: some dead links on the splash page, e.g. Portainer and some other links, and some documentation
    - TODO for E: look into what Adam reported
    - T: bugs also related to the global architecture in the Node-RED flows. Need to clean the processes related to user interaction and files+metadata linked to acquisition/users/etc. Will need to look into metadata, make sure we agree on that, and work on the GUI.
- Morgan: (presentation of planned work/goals)
    - MLops dev, here to integrate ML & detection into the PlanktoScope. Working on migration to RPi 5. Goal is to detect specific species/stages (targeted classification rather than general classification).
    - F: object detection/classification without a separate step for background subtraction and segmentation is harder and requires a better training set.
    - T: improving segmentation is also a topic which needs discussion
    - A: for my pyrocystis classification system, my classifier was trained on segmented images, and my object detector was trained on raw images
    - F: yes, that method works well on lab cultures where you can get lots of data; harder in environmental samples with rare objects (esp. those which are rare on EcoTaxa)
    - A: more recent neural object detectors are more robust with low sample counts - training doesn't need to be started from scratch, we can just fine-tune
    - F: yeah, we had different goals which made those less suitable
- Any updates related to FairScope @ OSM24?
    - (this was skipped without review)
- Any other updates?
    - F: there will be improvements to EcoTaxa; summary exports
    - E: I used the EcoTaxa API for the first time this past week

- T: we should also improve the documentation for the API exposed by the backend
- E: had issue with segmenter on last 200 datasets
- F: look at the intermediate results saved by the segmenter
- E: possible cause might be progressive dimming of raw images over the course of the expedition (following the standard protocol, I kept ISO at 100 the whole time). LED was always on throughout the expedition.
- F: intense use of LED - I've also seen dimming of raw images over extended operations. There's too much noise in the images because everything's too dim. There should be options in the segmenter to get more freedom about that, but it'll require tuning segmentation parameters to recover some nice behavior on example images. Be aware that I had tuned current options on good images. Here, I'd say you should tune the parameters to make the segmenter more aggressive about cleaning noise in the image. Segmenter assumes noise is 1-3 pixels, but not more; here, your noise is more than 3 pixels. The noise could be part of the objects, so your objects are probably also overestimated in size. Probably need to adjust erode/dilate parameters.
- A: In one of the new machines, I've noticed that the light looks much more dim. Maybe swap with a new LED and see if that solves the problem. In that case, I had to increase ISO and exposure time.
- F: swapping out the LED would be a good way to test if the problem is with the LED or with the camera sensor.
- T: we reduced the intensity of the LED in the last software release. Could be good to make this adjustable by the user
- F: that would also be a metadata field to capture. For now, first we need to test whether the failure was in the LED or in the camera sensor. If it's the LED, we'll need to start treating it as a consumable.
- TODO for E or A: locate the dim PlanktoScope and try to swap the LED with a not-dim PlanktoScope, to diagnose the problem. Overdriving LEDs will reduce their lifetime, and we were overdriving the LEDs. Normally LEDs are rated for very long lives.
- F: if overdriving of LEDs was the cause of the problem, the fix to reduce default LED brightness in v2023.9.0 may already solve the problem.

Discussion about metadata

Refer to Metadata Compilation
Discussion:

- T: goal for this is to look through each row of the spreadsheet and discuss it and document it.

- F: renaming fields would have some backwards-compatibility issues.
- E: can rename fields internally in the PlanktoScope software, but not rename externally-visible fields exported to EcoTaxa
- (decision: no renaming of fields in the EcoTaxa export TSV file)
- T: looks like the spreadsheet needs more work before we can go over it
- TODO for T: prepare the metadata spreadsheet for a next follow-up

Proposal review

(no change to existing proposals over the past week)
Refer to the Proposals project board on GitHub, and our description of the proposals process.

(skipped because the meeting ran out of time)

Review of proposals in the Final Comments column:

- (none)

Review of proposals in the Under Review column:

- Laurent: Proposal 290
    - Any updates from our discussions about metadata today?

Review of proposals in the Submitted column:

- (none)

Next steps for proposals in the Draft column:

- Laurent: Proposal 292 (no change)
- Ethan: Proposal 307 (no change)

TODO: post an update about our decisions from this meeting, to https://github.com/PlanktoScope/PlanktoScope/issues/282

2024-02-01

In attendance: Ethan, Adam, Morgan Coulm, Thibaut, Thibaut's brother (Fabien Pollina)

Introductions

- Fabien Pollina: works at manufacturing company; marketing & management. Interested in project, helping out with FairScope strategy.

- Morgan: student; apprentice. MLops, integration of models. Trying to use detection models on Raspberry Pi 4 & 5.
- Adam: Prakash lab; biochemist. Started working on PlanktoScope with Thibaut. Recently getting involved with FairScope company, PlanktoScope improvements
- (Ethan)

Individual updates + follow-up discussions

- Ethan:
    - Merged PRs 365 & 363
        - Thibaut (re: 365): nice to know that these small bugs are being taken care of. This change was merged already.
        - (discussion of the process for making changes to the Node-RED dashboard)
        - Thibaut (re: 363): another apprentice will work for 2 years on software development, will be fluent in English and be able to do UX and a bit of software development (in order to be able to define needs from the community). This way we can define a good POV on the need for the overall GUI. If there is nothing existing, what do we need to have? Possibility to write something new from scratch.
    - Also merged PRs 364 & 361 (no feedback needed)
    - Discussions with Khuong Huynh
        - T: the student is at Trondheim Univ. in Norway; next to a company named CFEED producing both algae and copepods, and maybe fish larvae to feed salmon. The company wants a better way to monitor their culture. I'm planning to visit them.
        - T: this person's efforts might overlap with your (Morgan) efforts. Might be good to connect on strategies, applications.
        - TODO for E: set up a groupchat between Khuong, Morgan, and Wassim
    - Follow-up with Wassim
        - E: will try to unblock Wassim by testing/troubleshooting his current code to figure out why it's not working
    - Planning to make an alpha prerelease of v2024.0.0 for use at OSM FairScope booth
- Thibaut : PlanktoScopes functionalities for OSM : WIfi Sharing +
    - T: plan is to have one or two iPads to access PlanktoScope interface, and/or invite people to connect from their phone. Would also like to have internet access on those iPads to access docs & web. I'd like to connect my personal phone to PlanktoScope and share my network with it, and let the PlanktoScope pass internet access via wifi to my computer.

    - E: recommend testing stability of the Wi-Fi network connection, since I've seen disconnection+reconnection behavior on my own computer.
    - T: give me an SD card image to test next week, and I'll do testing.
    - TODO for E: prepare an SD card image (v2024.0.0-alpha.0)
    - A: for ipads, will you still be editing wpa_supplicant file to connect to wifi?
- Morgan's goal outside the ML
    - T: Morgan will explore capacity to start to use targeted classification on RPi 5. Look for certain species, certain stage of given species, try to see if we can find it correctly in the environment. Will start with oyster larvae, mussel larvae. We have a nearby partner studying them. Morgan is starting by testing Pi, studying models. Right now this is a half classwork half FairScope work arrangement.
    - T: let's include Adam in the groupchat between Morgan, Khuong, and Wassim, since Adam has also done some ML work with PlanktoScope data.
    - M: I've trained one model on one species.
    - T: you can present more of that next week - to talk about what you will do.
    - TODO for E: add this to the agenda for next week
- Any follow-ups on the metadata discussions on Slack from this past week? e.g. documenting the exported metadata more thoroughly
    - T: next week will include discussion with Fabien Lombard on metadata - meeting will be 1 hour long.
    - TODO for E: add this to the agenda for next week
- Increasing streaming quality?
    - T: this might also include latency
    - E: could be possible after picamera2 migration to have URL query params to specify the resolution of the camera preview stream
- FairScope booth & OSM
    - T: back drape for the booth will show a collage of Plankton. Will have people passing the booth put a marker on the map to guess the source of the sample in the collage. Will need to make a form on FairScope's website with a map, some personal info (to gather data on users). Need to order the drape. How will we make this map input form work?
        - E: low-tech approach?
        - M: I've done something similar to the requested map input form functionality in the past.
        - FP: maybe use Google Maps?
- A: visited the Blue Robotics people. They had very advanced software; were using Vue(?); lots of proxy capabilities, addons. In 2 hours we got our Pi connected to their Pi, to get 4G access to the PlanktoScope. This enables control of a buoy a few hundred meters off-shore.

    - A: everything is open, at https://github.com/bluerobotics/BlueOS . Nice parallel: they have a strategy where they offer a product at a price point 10X less than existing solutions. Now they're super successful - impressive drone, everything running off RPi & Arduinos with custom HATs
    - E: interesting that they're using Docker too
    - A: yeah, they think we're on the right track besides lack of support for node-red-dashboard. They have 3 people working on the software.
    - A: they're very nice and open to talking, eager to help us avoid some of the mistakes they had made. I was very impressed at how much their software and grown, how much functionality it had, how well everything integrated. All the nice capabilities were already built-in. Nice to see a company working smoothly through development.
    - T: I set up a call with Tony from Blue Robotics on Monday.
    - A: he's returning to Hawaii, bringing the PlanktoScope with him for more testing. We can either connect the PlanktoScope to their 4G modem via Wi-Fi or Ethernet.
    - T: how are they enabling remote access to the software via 4G?
    - A: dashboard isn't accessed locally, it's always accessed through the internet. they're using TEAL (worldwide 4G connectivity network designed for IoT devices), which has contracts with cell network providers worldwide. 0.08 cents per MB or something.
    - TODO for E: check how Blue OS provides access to the dashboard (what is their network architecture?)
    - T: partnership will be very beneficial for hardware & software
    - A: someone from the company will be at OSM with a BlueBoat; not sure if they'll have much of a BlueROV presence there
- T: FairScope workspace setup
    - T: this is between the offices and the production space
    - T: Morgan is in Brest right now

Proposal review

(no change to existing proposals over the past week)

2024-01-24

(this meeting did not occur because only Ethan was in attendance)
In attendance:

Individual updates + follow-up discussions

- Ethan:
    - Need feedback on issue #355

Proposal review

(no change to existing proposals over the past week)

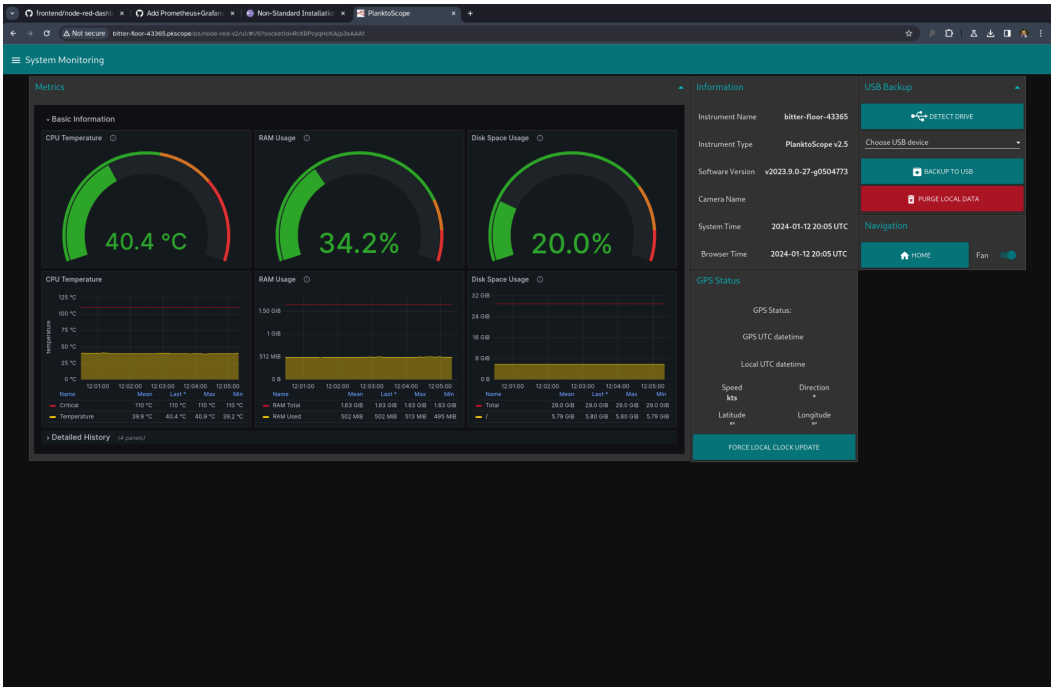
2024-01-18

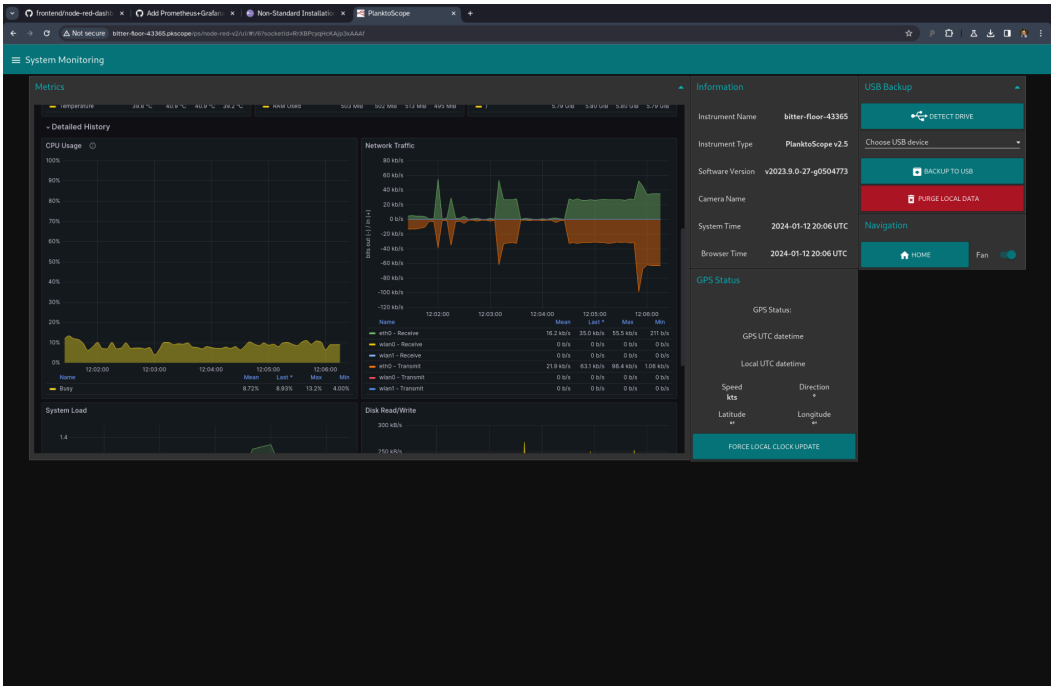
In attendance: Satoshi, Ethan

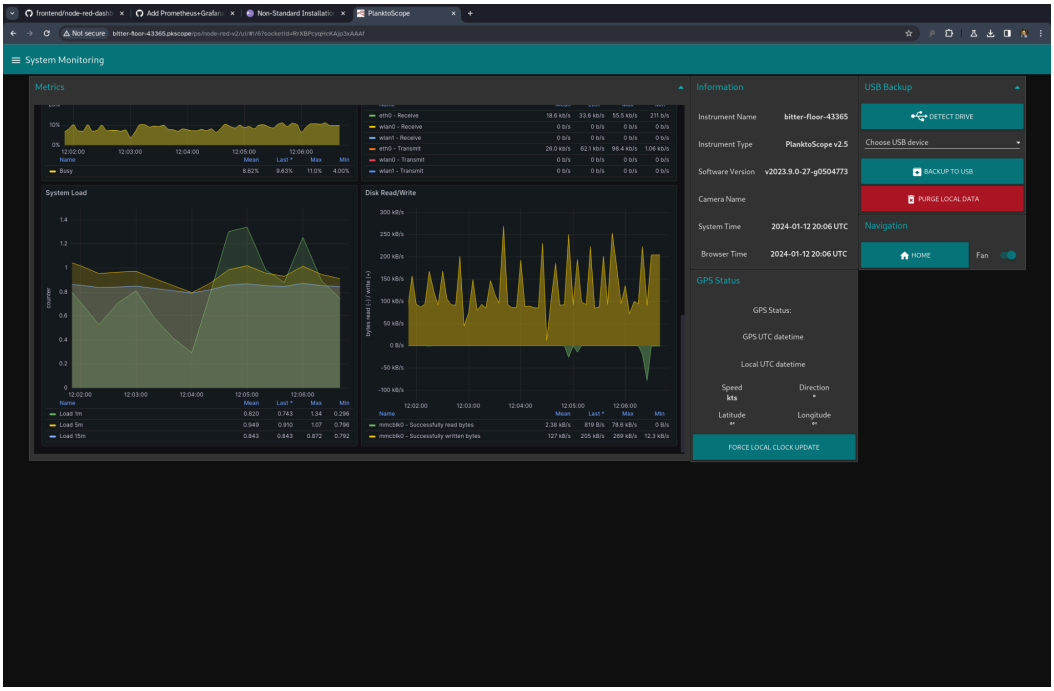
Individual updates + follow-up discussions

- Ethan:
    - Merged PR 351: now internet connection sharing works with a phone connected by USB tethering (if connected), and also with one Ethernet-to-USB adapter (if connected). PlanktoScope browser app access doesn't generally work from a phone connected by USB tethering though, because mDNS appears to be blocked by the phone.
        - Might have temporarily broken something though: if the PlanktoScope has no internet but my laptop is connected to both the internet and the PlanktoScope, the PlanktoScope blocks my laptop from getting internet until I disconnect and reconnect it. Will need to investigate.

- Merged PR 346. Screenshot:







- CPU usage information has been moved to the Detailed History section because I think it's not critical info to ensure system stability, and because the gauge is broken if the RPi's system time is incorrect.
    - S: that seems fine to me
    - S: the orange and red zones on CPU temperature here are better than what we had previously

        - This PR also adds a feature to the System Monitoring page of the Node-RED dashboard to compare the web browser's time (i.e. the system time of the client device accessing the Node-RED dashboard) with the RPi's system time; if the times are off by more than 1 minute, then a button is shown which the user can click on to set the RPi's system time to match the web browser's time.
            - This provides an easy way to fix the RPi's time, without having to use Cockpit, if/when GPS is unavailable and no hardware RTC exists (which is true of all v2.1 PlanktoScopes).
    - Merged PR 348; together with PR 324 (merged last week), total boot time has been reduced by ~1 minute, down to ~30 sec total. Hopefully this makes boot nearly as fast as in the v2.3 software.
    - (almost ready to merge) PR 352: the segmenter will be delivered/deployed as a Docker container image. This is part of the work needed for adding Raspberry Pi 5 support (via an incremental migration to the 64-bit OS since piwheels is unable to build the opencv and scikit-image packages for RPiOS 12).
    - Merged PR 350: now shell scripts (e.g. for autohotspot behavior) are managed as part of the Forklift pallet, making it easier to develop/test changes to those scripts
- Asynchronous update from Fabien on teaching with 10 planktoscopes:
    - Started the teaching with new version; half of the planktoscope were touched by the timeout error and loss of all metadata (hat, lens, calibration etc), I switched back to the old version. Will try to see if I can repeat the bug, but it was REALLY present (about 40% of machines touched, seems to be quite dependant on the machine, will see why it came from, to note : all machines were ok initially, only after calibration white balance/ pixel size/ pump rate did the planktoscope start to fail. I suspect that this comes from gestion of "calibration" data (eg. errazing pump rate per step and replacing it with new value) but didn't get time to try to repeat the bug for the moment
        - E: loss of all metadata is interesting, have not seen that before. Would like to see what the filesystem looks like there (e.g. are the metadata files missing, or is the software just unable to load the metadata files?).
        - (discussion about raspimjpeg timeout problems and picamera2 replacement)
            - E: TODO: send check-in message to Wassim
        - E: maybe it happens if the planktoscope is kept on for long periods of time? It's never happened to me when I try running image acquisition immediately after boot.

        - E: possible workaround would be to restart raspimjpeg after each acquisition, but then it loses all the camera settings (e.g. white balance and exposure control stuff) so it's not an easy workaround
        - E TODO: see if I can reproduce the "loss of all metadata" problem by changing hardware settings (e.g. pump steps/mL)
- A: T and I were working on a grant submission. We talked to two people from MBARI and it seemed like they really structured it towards us. We should get something for next week so we can plan the New Orleans trip.
    - E TODO: find jasmine today to do oceans conference registration+payment
    - A TODO: follow up with T to confirm flight dates for A and E, and (stretch goal) to find time for a meeting this upcoming week.

Proposal review

Refer to the Proposals project board on GitHub, and our description of the proposals process.

(skipped because the meeting ran out of time)

Review of proposals in the Final Comments column:

- (none)

Review of proposals in the Under Review column:

- Laurent: Proposal 290
    - Waiting until after February to give @tpollina time to provide input

Review of proposals in the Submitted column:

- (none)

Next steps for proposals in the Draft column:

- Laurent: Proposal 292
- Ethan: Proposal 307

TODO: post an update about our decisions from this meeting, to https://github.com/PlanktoScope/PlanktoScope/issues/282

2024-01-11

In attendance: Satoshi, Ethan, Adam

Individual updates + follow-up discussions

- Ethan:
  - Currently working on issue 252/PR 346. Preview:

    - E: (described the plan for this dashboard, asked for feedback on level of detail and what is visible in the screenshot above)
    - A: it looks good to me, not too much [detail]
    - S: it's fine to me

- E:

  - E: TODO: resize these graphs down so that the Detailed History row is the same height as the Basic Info row
- Details which don't need discussion: Merged PR 324, PR 341, PR 342
- Satoshi: talked to a company in Japan to resell PlanktoScope (Seabreath). They talked about things related to electromagnetic compliance, power supply mains compliance in Japan.
- (discussion between E and A about when the next FairScope USA planning meeting will be)
  - E: Thibaut isn't available for a meeting today
  - A: I'll call Thibaut later today

Proposal review

Refer to the Proposals project board on GitHub, and our description of the proposals process.

(skipped because the meeting ran out of time)

Review of proposals in the Final Comments column:

- (none)

Review of proposals in the Under Review column:

- Laurent: Proposal 290
  - A: I don't have much of an opinion. It seemed rather complicated.

    - E: TODO ask Thibaut for feedback/input, if he doesn't have any then I'll go ahead and make some executive decisions in/for prototyping which I'll document on this proposal.

Review of proposals in the Submitted column:

- (none)

Next steps for proposals in the Draft column:

- Laurent: Proposal 292
- Ethan: Proposal 307
    - E: tested USB tethering for internet access, it works great. WIll go ahead and add documentation about USB tethering anyways, and will add router configuration for USB tethering anyways. Proposal 307 then will just be about any further changes to make beyond USB tethering (e.g. whether to include a Wi-Fi hotspot) for use-cases where keeping a phone always connected all the time by USB to the Raspberry Pi isn't feasible

TODO: post an update about our decisions from this meeting, to https://github.com/PlanktoScope/PlanktoScope/issues/282

2024-01-04

In attendance: Ethan, Adam

Individual updates + follow-up discussions

- Anything to discuss about the v2023.9.0 release?
    - Adam: just installed it, did a purge of my current machine in lab yesterday. Was hoping one of the other units would be around, but probably Manu took a new version to Peru. I've been helping a high school student who was using the PlanktoScope. It seemed to work fine for basic usage.
    - A: can make a document going over pros & cons. Previously I had mainly been using the PlanktoScope in demos, e.g. in Santa Cruz. Could be nice to have an option to increase the preview stream resolution - good for demos and as an educational tool. Just live feed streaming and pump/focus adjustment rather than image acquisition, for people to see what's in the sample. Right now I just increase the window size. On my end I changed the raspimjpeg settings in the software, but normally it's easy enough to just zoom in on the preview stream in the browser interface.
        - TODO for Ethan: make a GitHub issues feature request.

        - Adam: a small and simple solution could just be to have another page with a larger camera preview display
        - E: or we could have a button on the optic configuration page to make the preview bigger
        - A: that could be good.
- Ethan:
    - started looking at what changes will be needed for the software to be able to run on Raspberry Pi OS 12 (bookworm), which is needed to run on the Raspberry Pi 5
    - did a bit of testing for proposal 307 (see below)
    - started to think about what to prioritize for upcoming v2024.0.0 release, and what scope of changes to aim for: https://github.com/PlanktoScope/PlanktoScope/milestone/4 (probably need to move some things out of that). Migration to RPi OS 12 (bookworm) will probably have to wait until v2024.1.0 or later, due to the magnitude of changes we'll need to make (which should probably be gradually rolled out across two or three releases, rather than changing everything in a single release). Key question: should we wait for the picamera2 migration to be finished before releasing v2024.0.0, in the hope that it will fix our mysterious raspimjpeg "camera timeout" problems?
- Discussion about https://github.com/PlanktoScope/PlanktoScope/issues/213 and directory naming
    - A: at some point we'll want to no be totally dependent on EcoTaxa as part of the UX, and migrate towards having support for our own platform that would look like Foldscope's Microcosmos. If/when we do a PlanktoScope workshop, it'd be nice to get some feedback on how people are using the data they collect (e.g. EcoTaxa upload vs. other downstream uses).
- Invitation from Tobias to speak at LIBRE hub video series
    - A: I chatted with Tobias a bit, I'll check with Thibaut. Should be easy enough to do this talk.
- (any other discussions)

Proposal review

Refer to the Proposals project board on GitHub, and our description of the proposals process.

(skipped because the meeting ran out of time)

Review of proposals in the Final Comments column:

- (none)

Review of proposals in the Under Review column:

- Laurent: Proposal 290
    - Adam: will take a look tonight

Review of proposals in the Submitted column:

- (none)

Next steps for proposals in the Draft column:

- Laurent: Proposal 292
- Ethan: Proposal 307
    - Ethan: tested out USB tethering from a phone (Option C), it worked for sharing internet but not for accessing pkscope.local. Leaning towards Option C and away from Option B (Ethernet)
    - Adam: I agree about not emphasizing Ethernet.
    - Adam: a way to navigate captive portals will be important to have. TODO: try out these options and demo them in a future meeting.

TODO: post an update about our decisions from this meeting, to https://github.com/PlanktoScope/PlanktoScope/issues/282