Lessons learned from 15 years of open source robotics

Infra for Drones

whoami

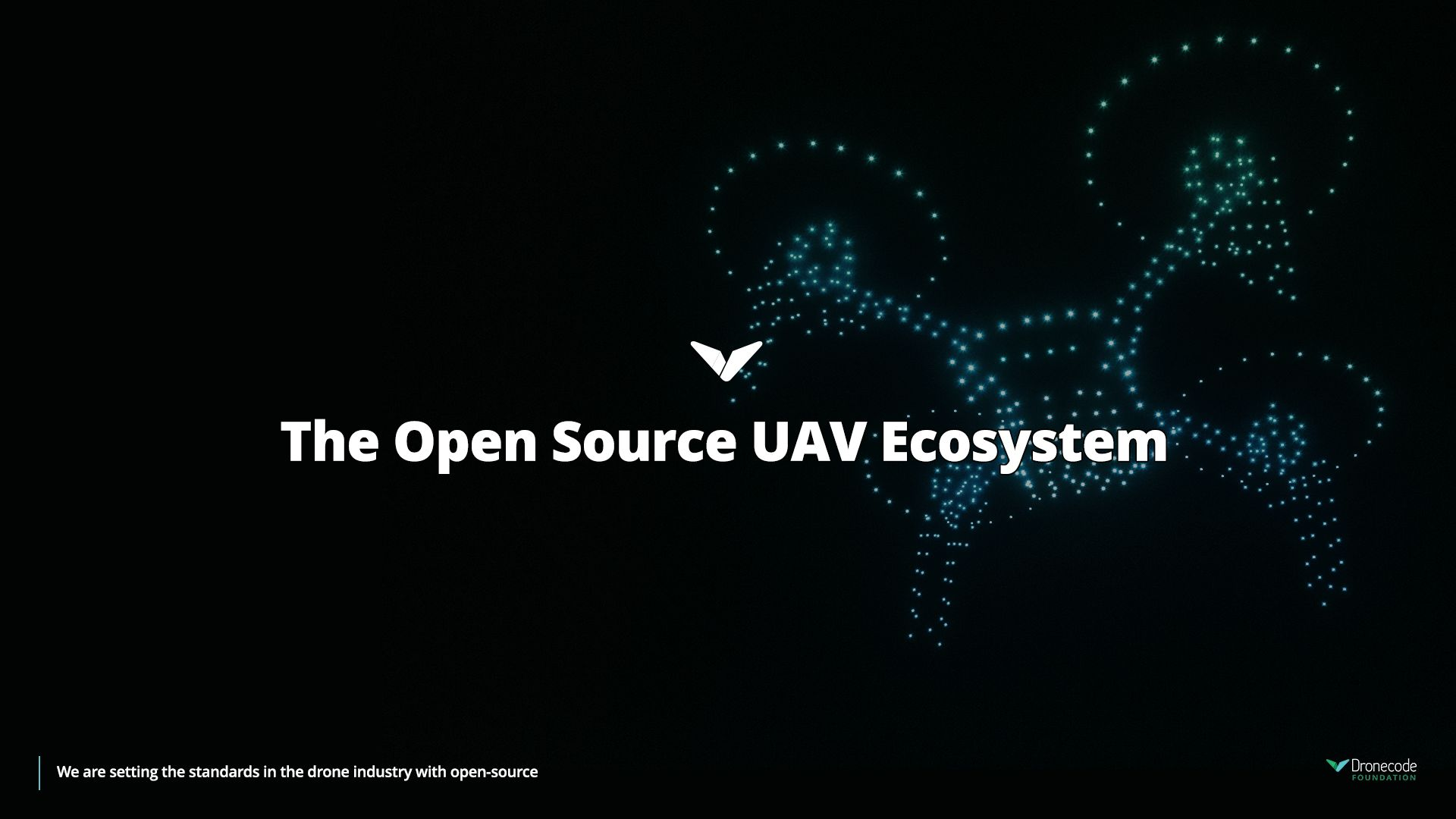
Ramón Roche
General Manager
Dronecode / Linux Foundation

(Still) an individual contributor

10+ years working in aerial robotics

Co-Lead ROS Aerial Robotics CWG

Co-Lead Space Grade Linux SIG



The Open Source UAV Ecosystem

We are setting the standards in the drone industry with open-source

Dronecode FOUNDATION

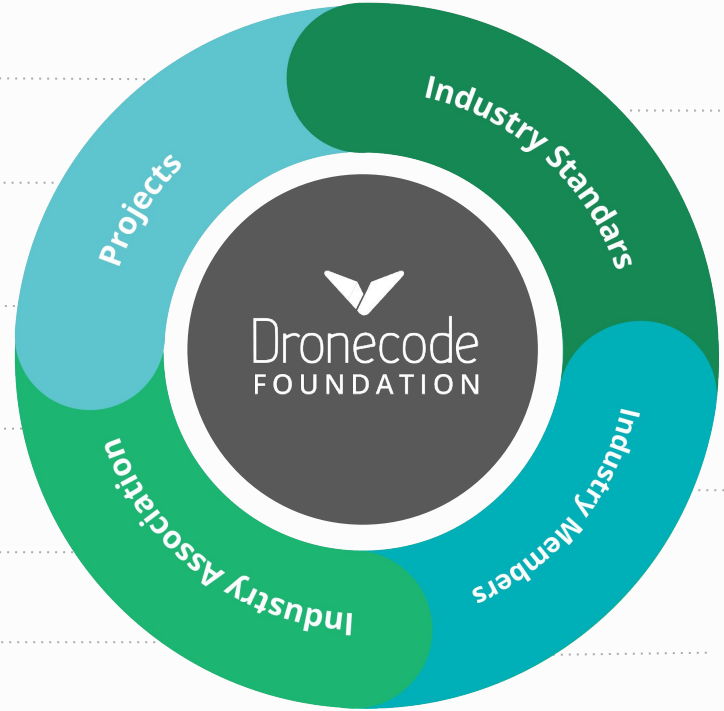
What the hell is Dronecode Foundation

The Dronecode Foundation, marked its 10th anniversary in 2024, we are part of the Linux Foundation.

In simple words, we are a non-profit formed to safeguard open source projects, promote collaboration, standardization, and support the commercial adoption of the open source projects.

- Neutral home for open source projects
- Promoting open collaboration
- Help set standards to accelerating innovation
- Supporting commercial and research use

The Dronecode Ecosystem



QGroundControl

MAVSDK

PX4 autopilot

OpenCV

THE LINUX FOUNDATION

open source robotics alliance

pixhawk

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

Auterion

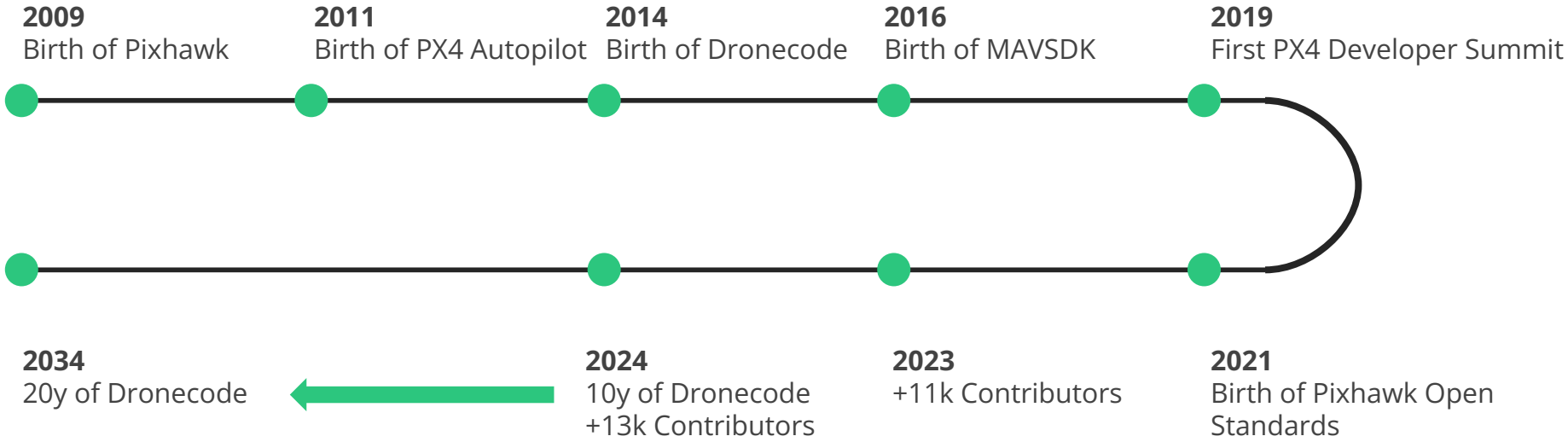
NXP

We are setting the standards in the drone industry with open-source

Member Community

These are the member companies that play an integral role in shaping the future of the industry.

Brief Timeline

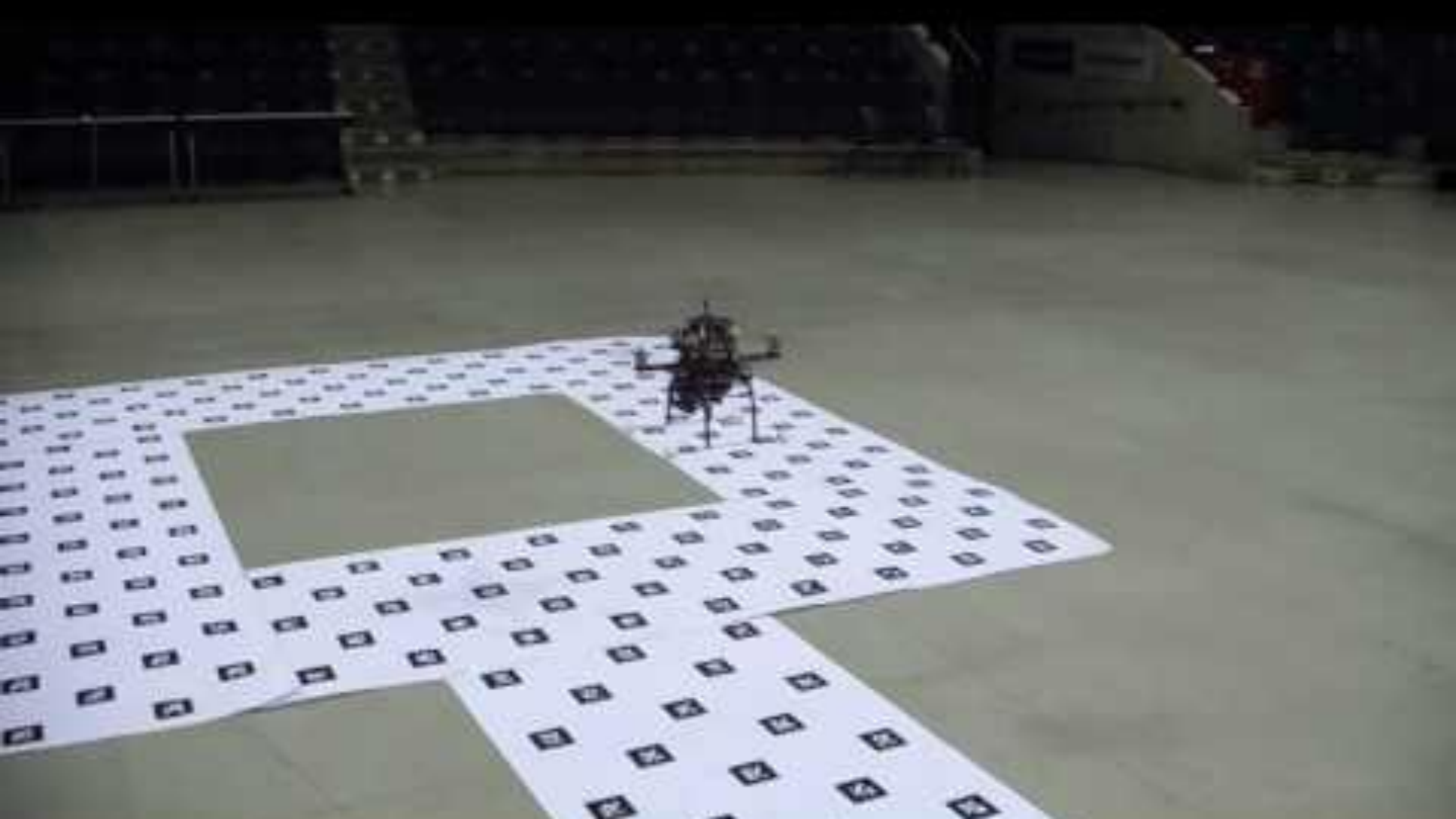


Putting things into perspective

- $1B Project Value
- 59.72M Total Lines of Code
- 13,307 Total Unique Contributors

- 1,900 Contributors in 2024
- 100+ Git Repositories
- 20 Dronecode Members
- 5 Top-Level Open Source Projects

2009



2011


4x speed

Today

PX4 BASED OMNICOPTER USING THE NEW DYNAMIC MIXING SYSTEM


DroneUp
FAA
CERTIFIED
OPERATION
STAND CLEAR


wingtra

How are these varied applications possible with the same codebase?

PX4 Autopilot - ✅ The Answer

Autonomy Stack originally developed for Aerial Robotics, primarily Multi Rotors, over time extended to support Fixed-Wing, VTOL, and Over & Under Surface Vehicles.

Main Characteristics:

- Runs **realtime** on top of Apache NuttX RTOS
- 100% C++ based
- **Modular** architecture with a **DDS-compatible middleware** (uORB)
- Modules are fully **parallelized**, and **thread safe**
- Great hardware support
- Support for custom builds, **trim what you don't need**
- More than **1M** vehicles using PX4
- More than **13k** developers

- **Flight Modes** provide a set of helpers to control autonomy
- **Flight Tasks** allowing developers to extend flight modes
- **Parameter** database exposing functionality back to users
- **Events interface** giving developers a system-wide API for notifications
- **Control allocation** translates thrust and torque commands into actuator commands which control motors and servos
  - Controllers do not require special handling for airframe geometry
- Native **ROS 2 Support** through DDS

PX4 Autopilot - Hardware Support

- Support for more than 80 boards from 30+ manufacturers
- Drivers for more than 100+ sensors
  - IMU, Baro, Actuators, GPS, INS, CAN, UWB... etc.
- Main Architectures Supported
  - STM32 - STMicro
  - iMX - NXP Semiconductors
  - RISCV-V

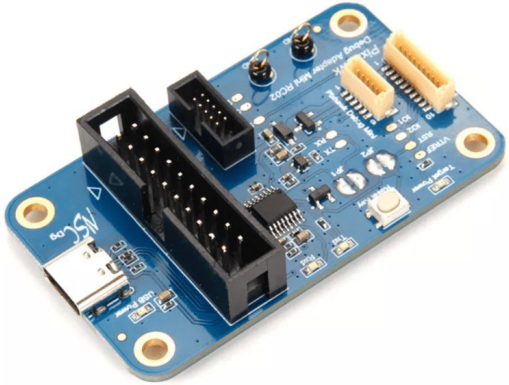
Pixhawk Hardware

Open Hardware & Open Standards

Started as a flight controller open hardware project with the first versions of Pixhawk's

Evolved into an Open Standard for

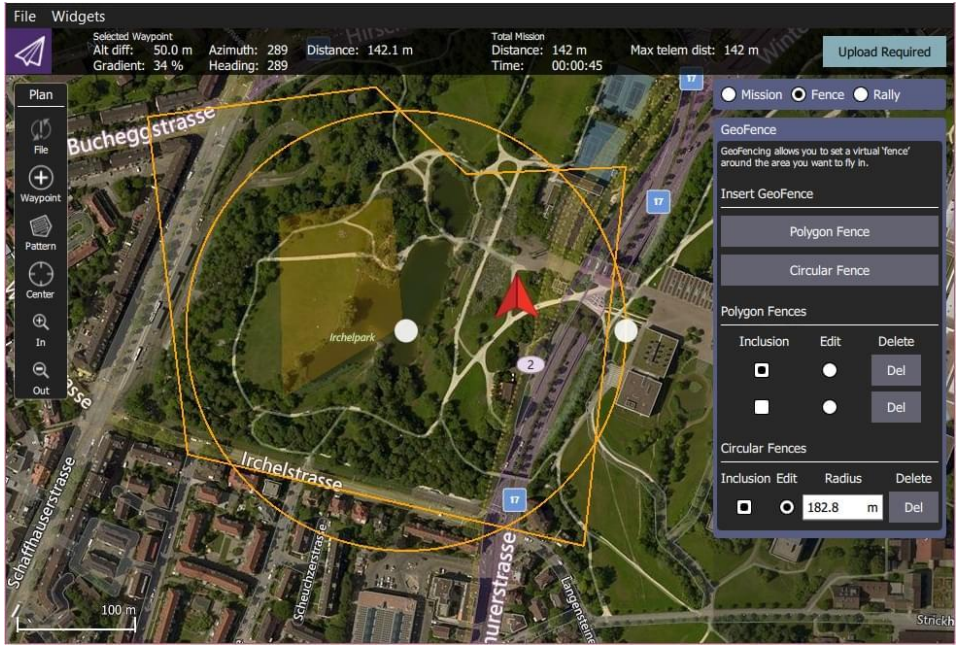
- Flight controllers
- Payloads (Gimbals)
- Smart Batteries
- Connectors
- Debuggers



PX4 Autopilot + QGroundControl - Autonomous Missions

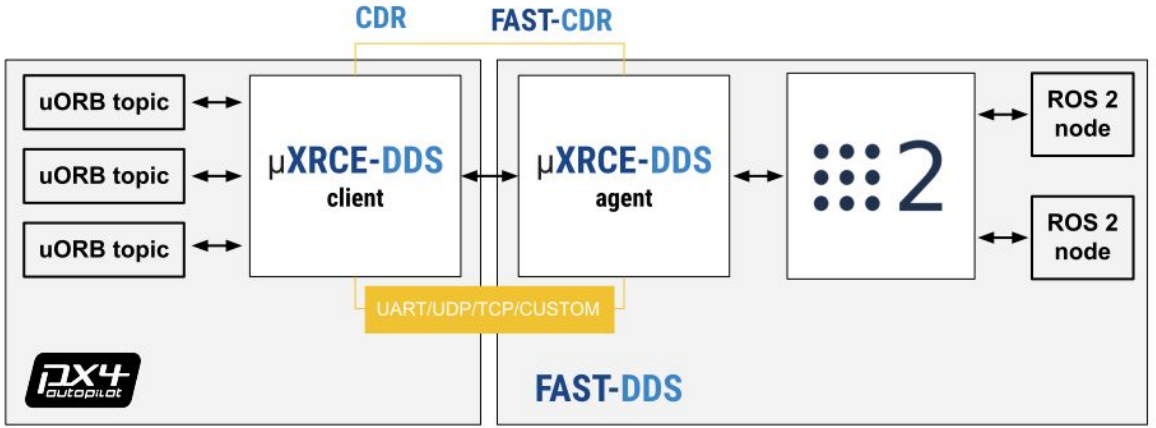
Define waypoints with customizable actions that allow you to control the behavior of vehicles.

Standardized mission protocol trough MAVLink



PX4 Autopilot - ROS 2 Support

- Thanks to uORB middleware we can communicate directly with the ROS 2 middleware (XRCE-DDS based)
- PX4 internal modules can share data with ROS 2 nodes
- Agent / Client approach
- Ethernet and Serial support
- ROS 2 QoS Supported
- We are ready for the switch to Zenoh!

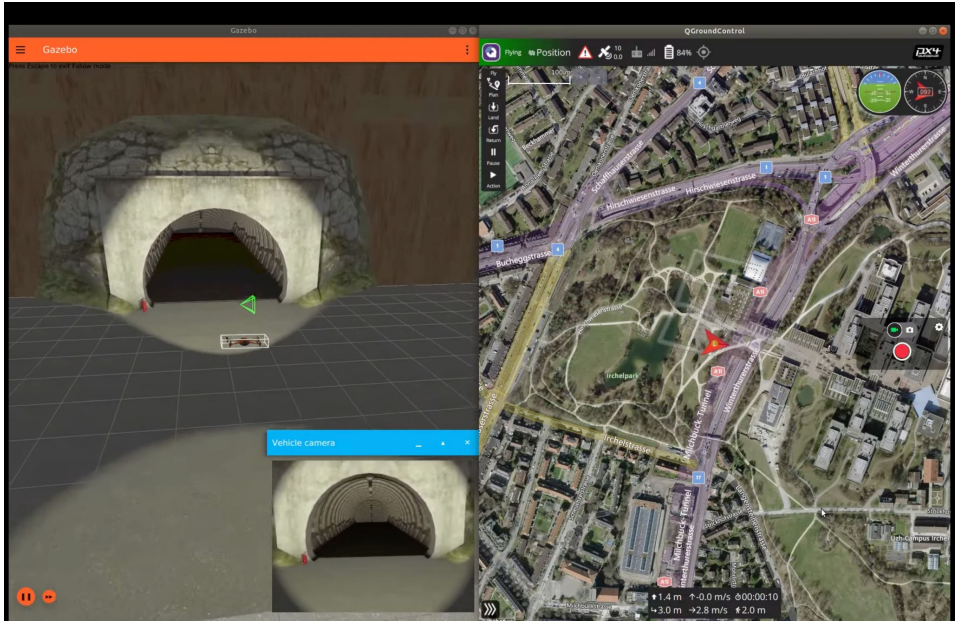


PX4 Autopilot - Simulation

Gazebo is our default simulation agent, we support both classic and modern gazebo, with multiple worlds and models to choose from.

Other Simulation Engines Supported

- Gazebo Classic
- AirSim
- Flight Gear
- jMAVSim



Gazebo Simulation Docs

How to even begin testing PX4?

Let’s split the problem into tiny problems

Developer Testing

- Unit Tests
- Integration Tests / Simulation Tests
- Hardware In the Loop Tests

End User Testing

- Flashing / Installing
- Setting Up
- Tuning
- Manual Flight
- Autonomous Flight
- Validation of Testing

How to even begin testing PX4?

Developer Testing

- Unit Tests
- Integration Tests / Simulation Tests
- Hardware In the Loop Tests

Can be run in CI

End User Testing

- Flashing / Installing
- Setting Up
- Tuning
- Manual Flight
- Autonomous Flight
- Validation of Testing

How to even begin testing PX4?

Developer Testing

- Unit Tests
- Integration Tests / Simulation Tests
- Hardware In the Loop Tests

End User Testing

- Flashing / Installing
- Setting Up
- Tuning
- Manual Flight
- Autonomous Flight
- Validation of Testing

Needs humans in the loop

Tests in the Cloud

*or at least initiated by the cloud

What can be run in CI?

Developer Testing

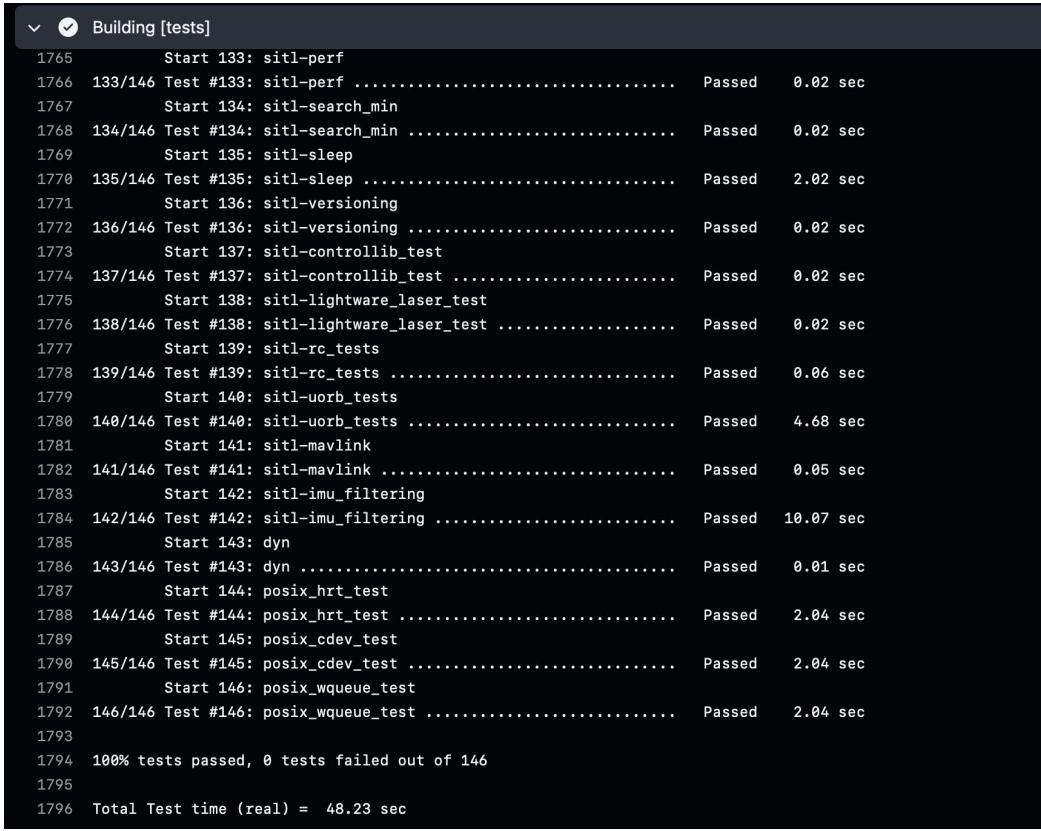
- Unit Tests
- Integration Tests / Simulation Tests
- Hardware In the Loop Tests*

Builds

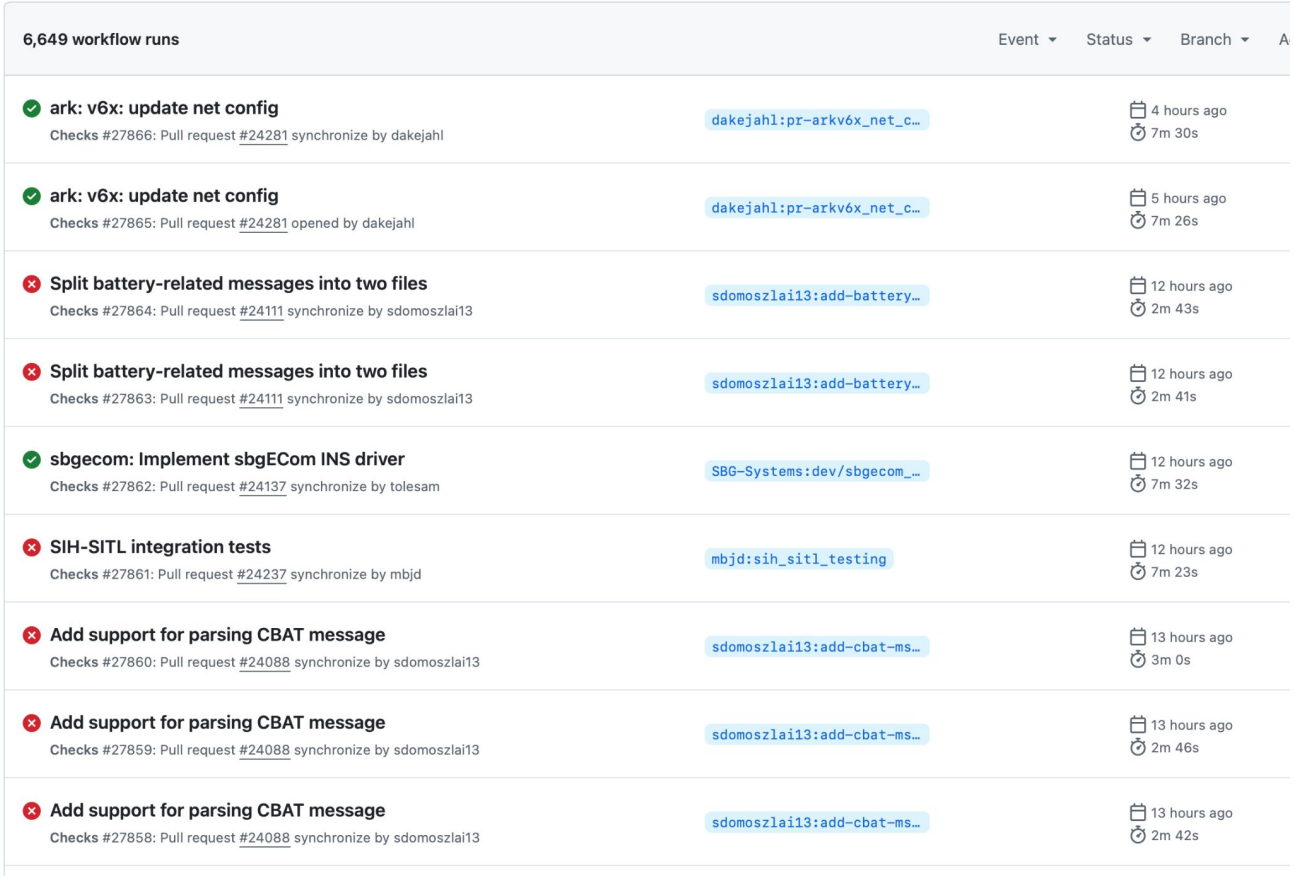
- Releases
- Pull Requests

Unit Tests

- We run unit tests as frequently as possible
- Our coverage is less than ideal < 60%
- It’s super hard to incentivise contributors to write more
- Tests run in around 7mins
- You can select to run some or all

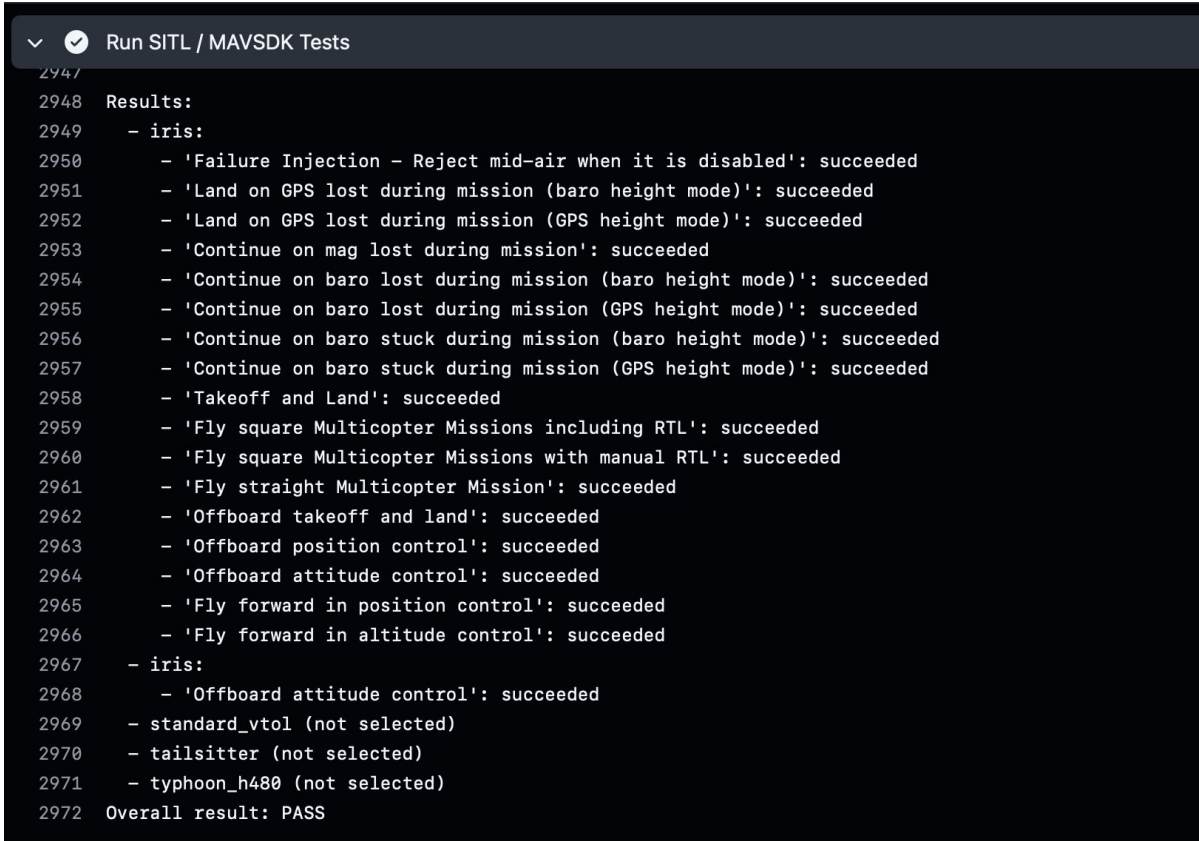


Unit Tests


6,649 workflow runs
Event
Status
Branch
ark: v6x: update net config
Checks #27866: Pull request #24281 synchronize by dakejahl
dakejahl:pr-arkv6x_net_c…
4 hours ago
7m 30s
ark: v6x: update net config
Checks #27865: Pull request #24281 opened by dakejahl
dakejahl:pr-arkv6x_net_c…
5 hours ago
7m 26s
Split battery-related messages into two files
Checks #27864: Pull request #24111 synchronize by sdomoszlai13
sdomoszlai13:add-battery…
12 hours ago
2m 43s
Split battery-related messages into two files
Checks #27863: Pull request #24111 synchronize by sdomoszlai13
sdomoszlai13:add-battery…
12 hours ago
2m 41s
sbgecom: Implement sbgECom INS driver
Checks #27862: Pull request #24137 synchronize by tolesam
SBG-Systems:dev/sbgecom_…
12 hours ago
7m 32s
SIH-SITL integration tests
Checks #27861: Pull request #24237 synchronize by mbjd
mbjd:sih_sitl_testing
12 hours ago
7m 23s
Add support for parsing CBAT message
Checks #27860: Pull request #24088 synchronize by sdomoszlai13
sdomoszlai13:add-cbat-ms…
13 hours ago
3m 0s
Add support for parsing CBAT message
Checks #27859: Pull request #24088 synchronize by sdomoszlai13
sdomoszlai13:add-cbat-ms…
13 hours ago
2m 46s
Add support for parsing CBAT message
Checks #27858: Pull request #24088 synchronize by sdomoszlai13
sdomoszlai13:add-cbat-ms…
13 hours ago
2m 42s

Integration Tests

- Builds and runs PX4 in simulation mode
- Controls PX4 via API calls using the SDK
- Happens faster than realtime
- Tests are written in C++
- The most fragile of our tests
  - A python script is orchestrating the whole thing
  - Frequent timeouts between process comms



Integration Tests

Triggered via pull request 5 hours ago

dakejahl synchronize #24281 `dakejahl:pr-arkv6x_net_c…`

Status	Total duration	Artifacts
Success	**14m 13s**	–

sitl_tests.yml

on: pull_request

Matrix: build

- Testing PX4 iris 8m 41s
- Testing PX4 standard... 13m 26s
- Testing PX4 tailsitter 13m 35s

Integration Tests

Each test launches a new PX4 and Gazebo instance

Test Runner

- Failure Injection
- Land if GPS Lost
- Takeoff and Land
- Fly a square
- Fly straight

Faster than Realtime Execution

PX4 ↔ Gazebo

Simulation Engine

Integration Tests

mrpollo opened on Nov 18, 2024 · edited by mrpollo

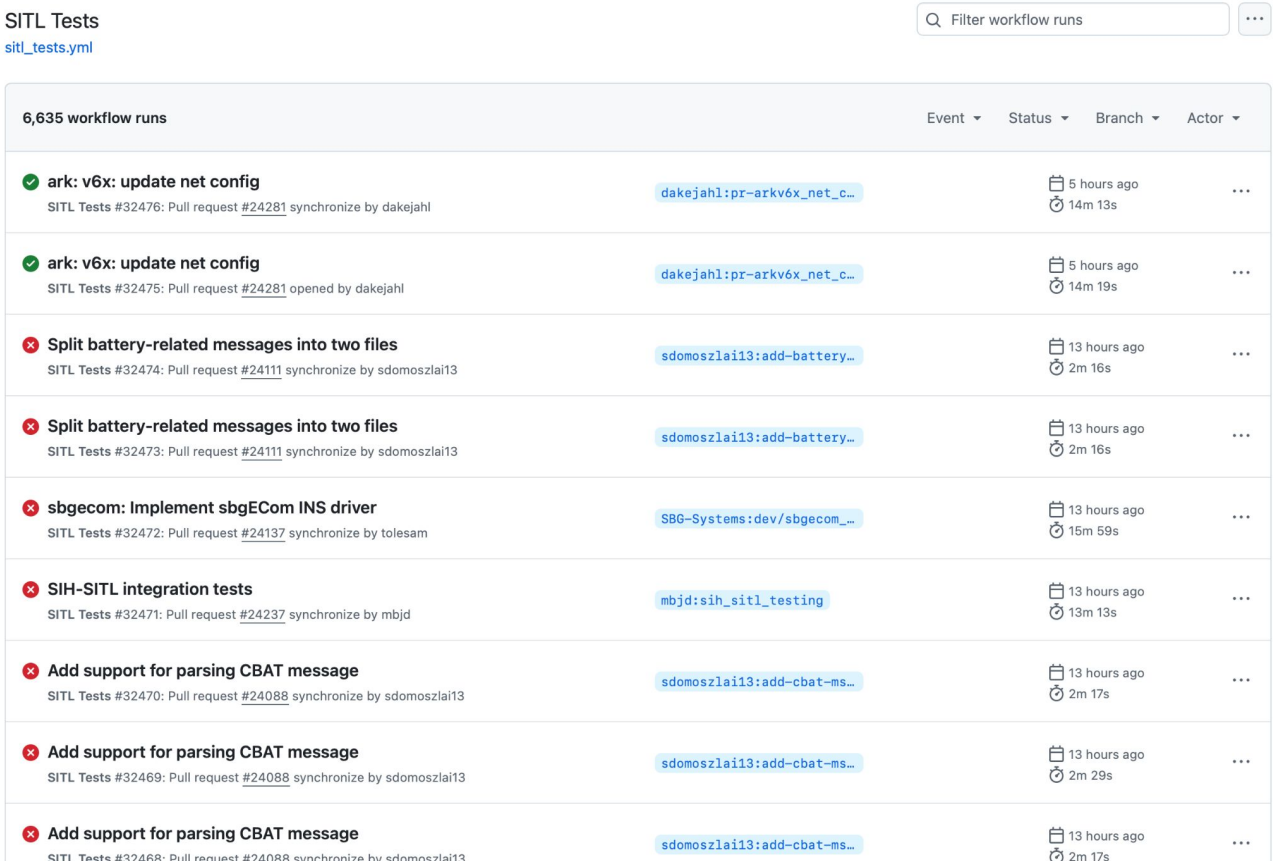
Edits

We need to improve the current SITL tests.

Stage 1

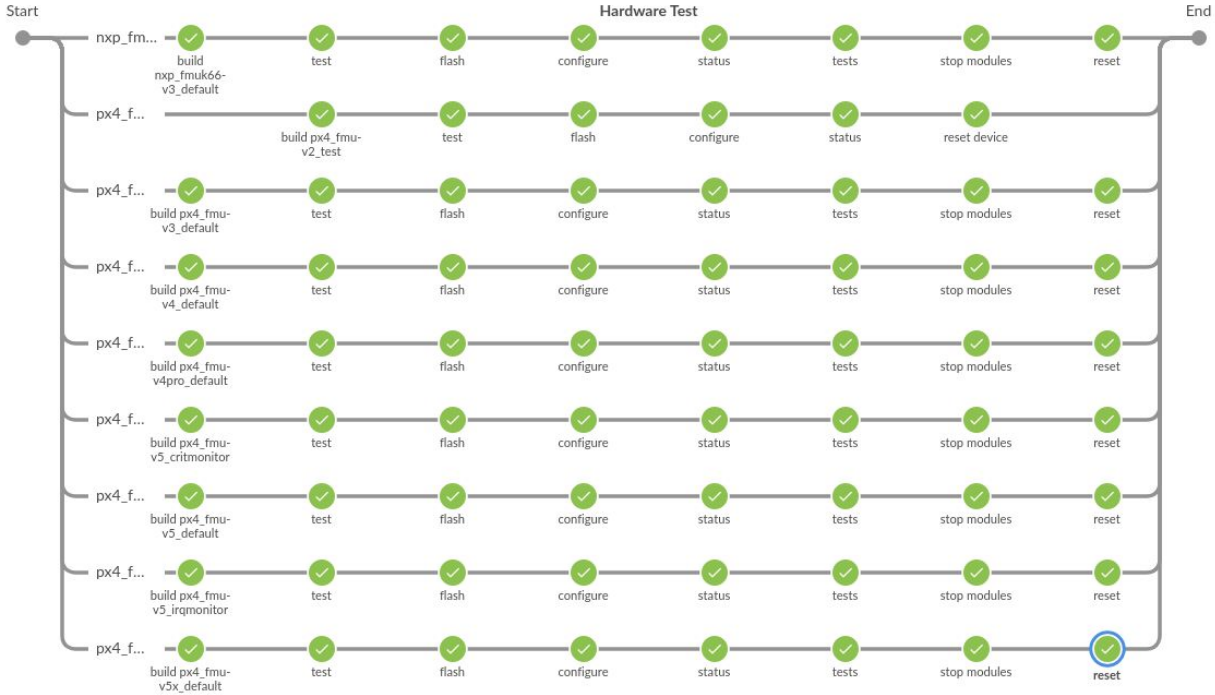
- [ ] Improve specific failing tests
  - [ ] (vtol) Fly VTOL mission with wind change (ramped_up_wind.world)
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11940615234/job/33283663787#step:19:3117
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11946116827/job/33299931473#step:19:3089
  - [ ] (vtol) RTL direct home with approaches
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11941237714/job/33285631449#step:19:2169
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11943817019/job/33293577322#step:19:2172
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11963888081/job/33355197712?pr=24020#step:19:2401
  - [ ] (vtol) RTL direct home not as rally point
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11904739730/job/33174054458?pr=23958#step:19:2461
  - [ ] (VTOL) RTL direct Mission Landing
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11980160836/job/33403865204#step:19:4046
  - [ ] (trailsitter) RTL direct Mission Land
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11941448644/job/33286304485#step:19:1446
  - [ ] (tailsitter) RTL direct rally without approaches
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11944522384/job/33295599685#step:19:2865
  - [ ] (tailsitter) RTL direct rally with approaches
    - https://github.com/PX4/PX4-Autopilot/actions/runs/11950993944/job/33313676674?pr=24011#step:19:3231
    - https://github.com/PX4/PX4-Autopilot/actions/runs/12006914556/job/33466430076?pr=23961#step:19:3272

Integration Tests



Hardware In The Loop

- Linux Desktop
  - Jenkins Slave
- USB Hub
- FTDI to Hardware
- Flashes and Runs Tests
- 20+ Hardware Boards
- Lives in one of the maintainers house
- No longer running
  - Too fragile, Timeouts, interrupts
  - We have plans for a new version
  - Not all supported hardware was found



Hardware In The Loop

Builds builds builds

Context

- Our primary development environment supported is Ubuntu LTS, currently 24.04
- Our GCC toolchain is arm cross build gcc
  - v9.3.1 (arm-none-eabi-gcc)
- Dependencies can be installed by a helper script
  - ./Tools/setup/ubuntu.sh

Example

$ make px4_fmu-v6x

- You always need to specify the hardware target
- We support more than 80 boards

When do we need to build?

- On every PR so we can guarantee the code at least builds
  - but also so we can fly it (when needed)
- On releases so we can produce the release artifacts

Builds builds builds

How do we test builds for all hardware?

Step 1

We run a python script that finds all **board definitions**, and grabs some **metadata** so we can build.

Step 2

Group hardware by **architecture**, and by manufacturer. Currently around 20+ groups

Step 3

Build each, and upload artifacts

```
❯ ./Tools/ci/generate_board_targets_json.py --groups --verbose
======================
= scanning for boards =
======================
excluding label test (nxp_fmuk66-v3_test)
excluding board modalai_voxl2 (modalai_voxl2_default)
excluding label test (cuav_x7pro_test)
excluding label test (cubepilot_cubeorangeplus_test)
excluding label test (cubepilot_cubeorange_test)
excluding label test (px4_fmu-v5x_test)
excluding board px4_ros2 (px4_ros2_default)
excluding label replay (px4_sitl_replay)
excluding label nolockstep (px4_sitl_nolockstep)
excluding label test (px4_sitl_test)
excluding label test (px4_fmu-v4pro_test)
excluding label test (px4_fmu-v4_test)
excluding label stackcheck (px4_fmu-v5_stackcheck)
excluding label test (px4_fmu-v5_test)
============================
= Boards found in ./boards =
============================
{'base': {'container': 'px4io/px4-dev-base-focal:2021-09-08',
          'manufacturers': {'px4': ['airframe_metadata',
                                    'parameters_metadata',
                                    'extract_events',
                                    'px4_sitl_allyes',
                                    'px4_sitl_default',
                                    'px4_sitl_zenoh']}},
 'nuttx': {'container': 'px4io/px4-dev-nuttx-focal:2022-08-12',
           'manufacturers': {'flywoo': ['flywoo_gn-f405_default'],
                             'omnibus': ['omnibus_f4sd_icm20608g',
                                         'omnibus_f4sd_default'],
                             'bitcraze': ['bitcraze_crazyflie_default',
                                          'bitcraze_crazyflie21_default'],
                             'nxp': ['nxp_fmuk66-v3_default',
                                     'nxp_fmuk66-v3_socketcan',
```

Builds builds builds

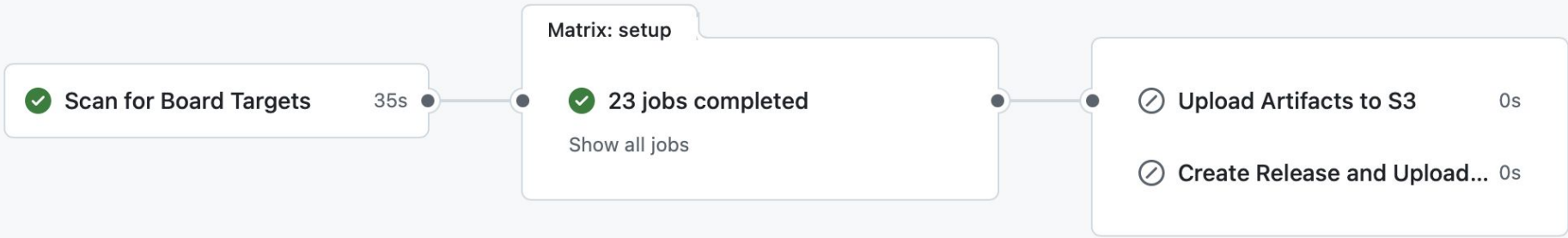
Triggered via pull request 4 hours ago

dakejahl synchronize #24281 `dakejahl:pr-arkv6x_net_c…`

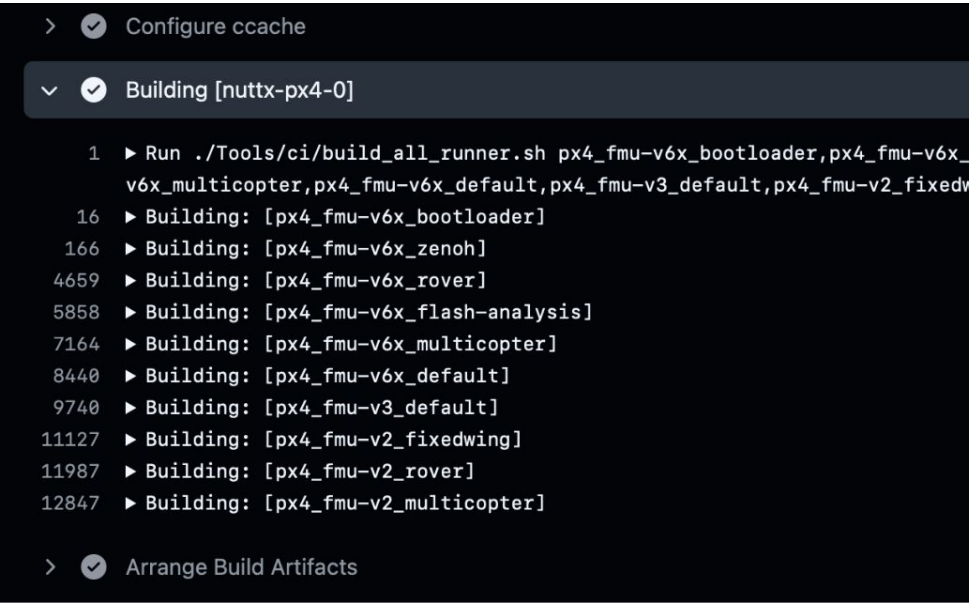
Status	Total duration	Artifacts
Success	**16m 30s**	**23**

build_all_targets.yml

on: pull_request



Builds builds builds



Build Group [nuttx-px4-0] summary

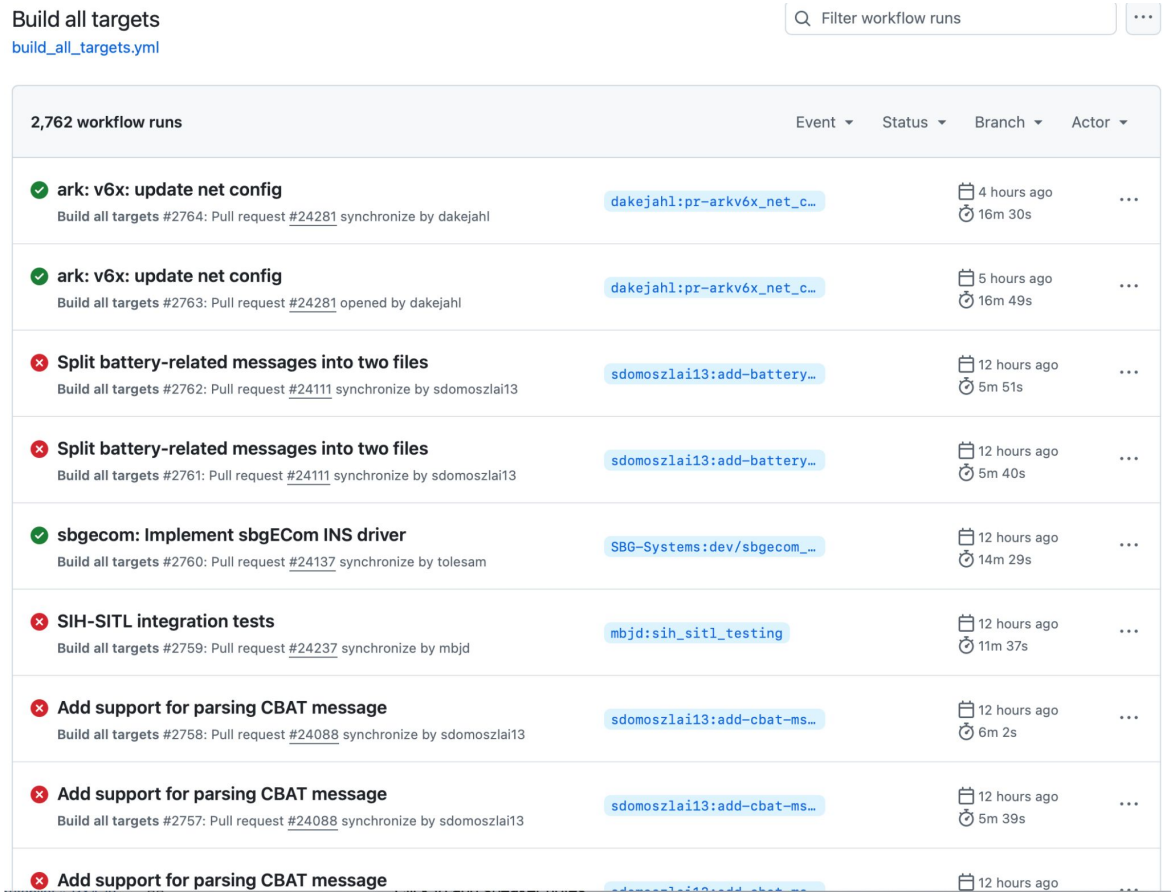
Build Times

- **px4_fmu-v6x_bootloader** - 0h 0m 58s elapsed
- **px4_fmu-v6x_zenoh** - 0h 2m 3s elapsed
- **px4_fmu-v6x_rover** - 0h 0m 55s elapsed
- **px4_fmu-v6x_flash-analysis** - 0h 0m 51s elapsed
- **px4_fmu-v6x_multicopter** - 0h 0m 51s elapsed
- **px4_fmu-v6x_default** - 0h 0m 36s elapsed
- **px4_fmu-v3_default** - 0h 1m 33s elapsed
- **px4_fmu-v2_fixedwing** - 0h 0m 45s elapsed
- **px4_fmu-v2_rover** - 0h 0m 31s elapsed
- **px4_fmu-v2_multicopter** - 0h 0m 32s elapsed

Job summary generated at run-time

Builds builds builds

- Running on self hosted runners
- Hosted in AWS
- Using RunsOn
- < 20 min total run time



Summary of CI

ark: v6x: update net config #24281

Draft **dakejahl** wants to merge 2 commits into `PX4:main` from `dakejahl:pr-arkv6x_net_config`

Conversation 5 | Commits 2 | Checks 57 | Files changed 1

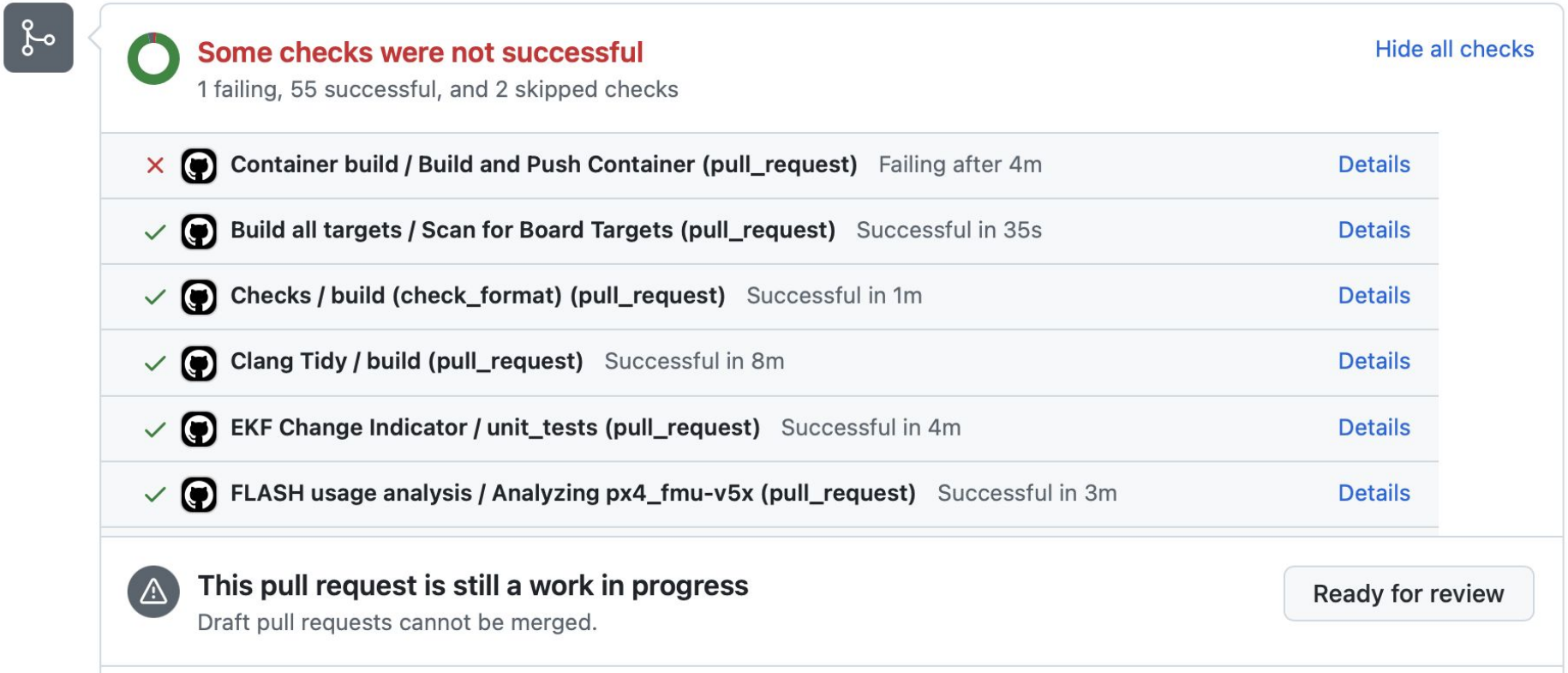
dakejahl commented 6 hours ago • edited — Member

Updated ARK V6X Kconfig options to match our production line settings

New defaults

```
INFO  [netman] DEVICE=eth0
INFO  [netman] BOOTPROTO=static
INFO  [netman] NETMASK=255.255.255.0
INFO  [netman] IPADDR=192.168.0.4
INFO  [netman] ROUTER=192.168.0.1
INFO  [netman] DNS=192.168.0.254
```

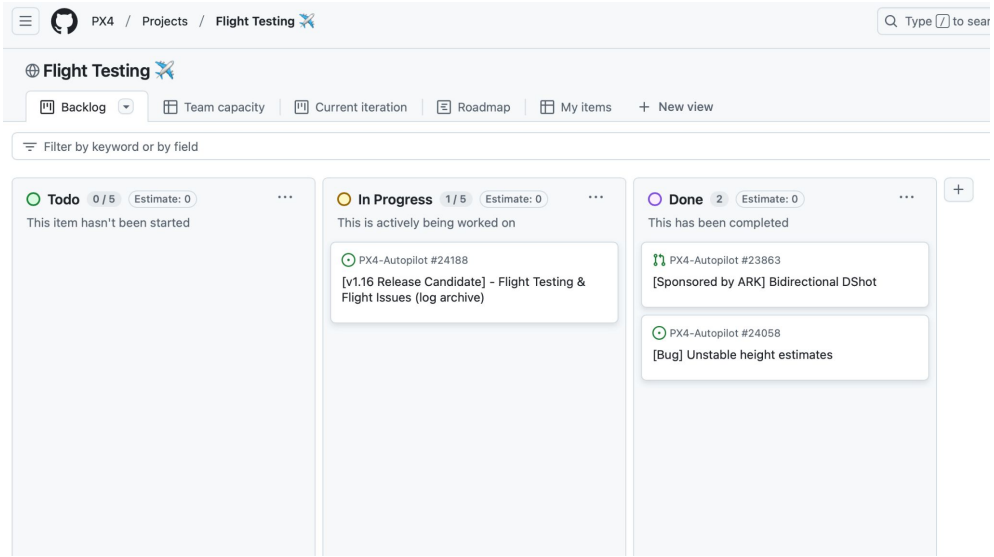
Summary of CI



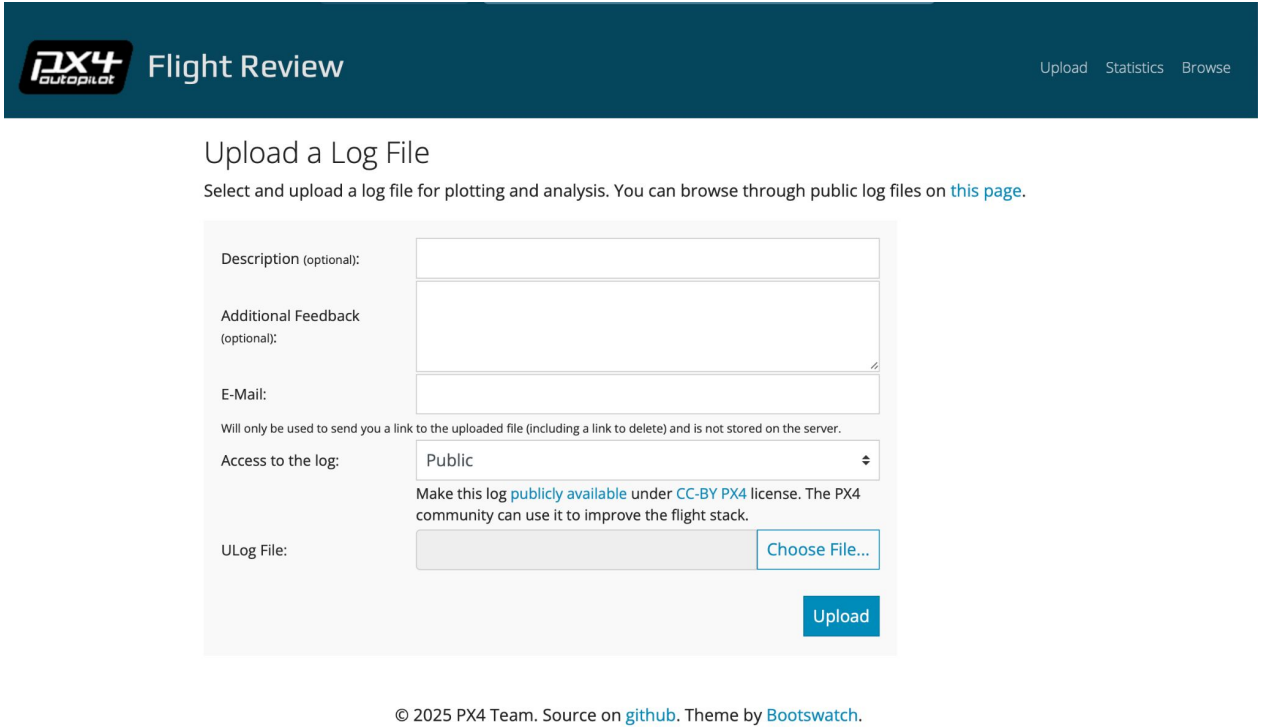
Human In the Loop Tests

End User Testing - Pilots Flying Hardware!

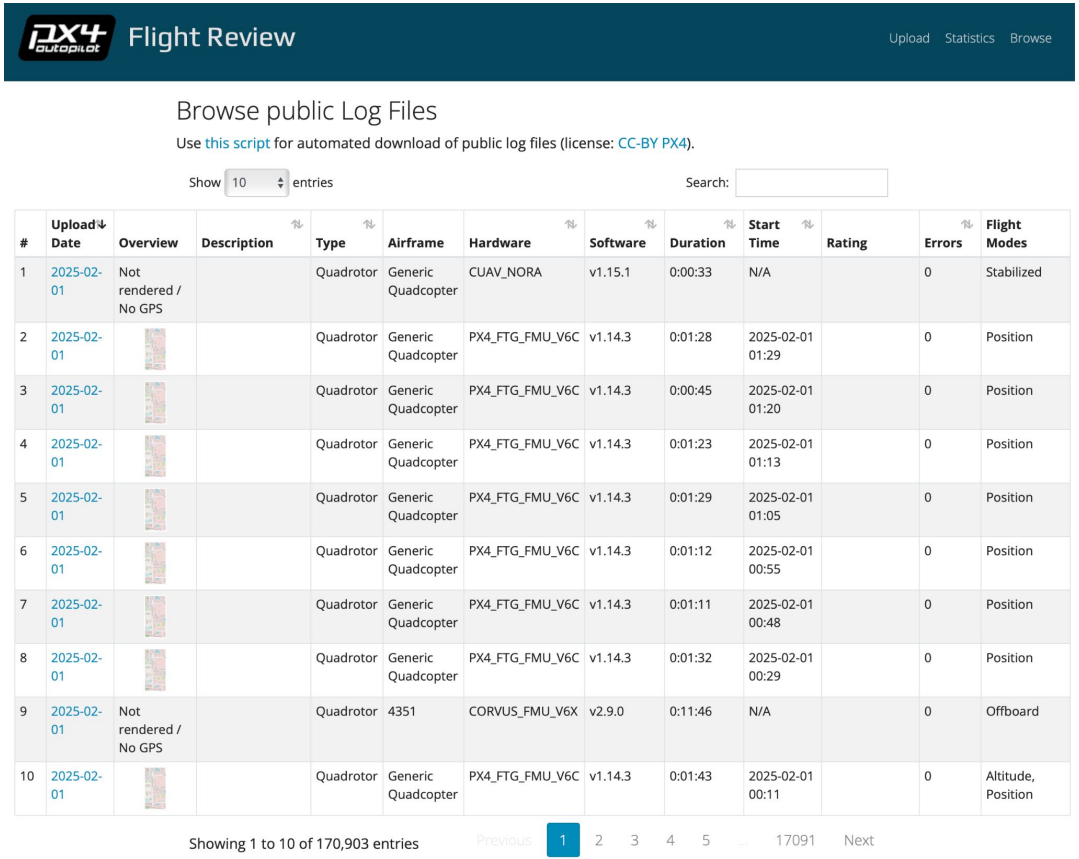
- We rely on our community for tests flights
- One of our member companies is providing recurrent tests
- Other community members also pitch in
- We wrote “Test Cards” with instructions for pilots to follow
- Github Project for tracking of PRs or Issues that require testing
- Pilots upload logs from flights to our servers



PX4 Log Review



PX4 Log Review



Flight Review

Upload Statistics Browse

Browse public Log Files

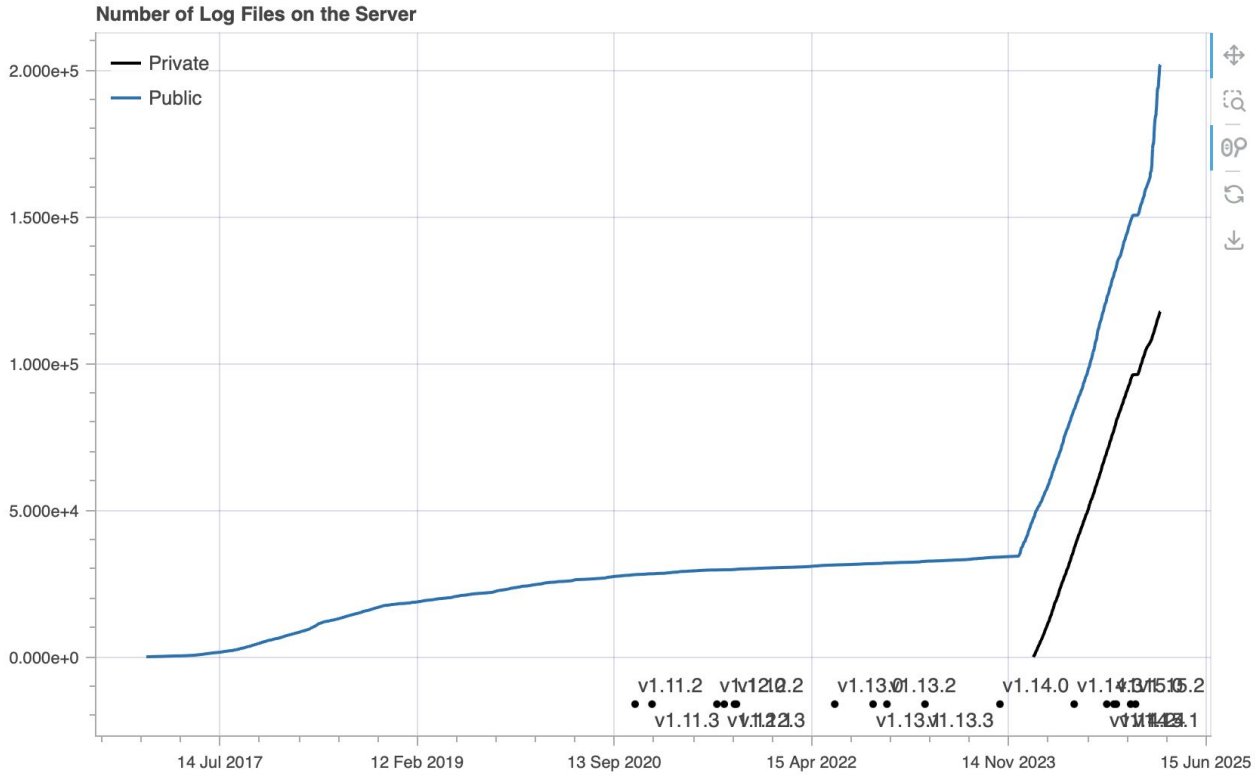
Use this script for automated download of public log files (license: CC-BY PX4).

Show 10 entries Search:

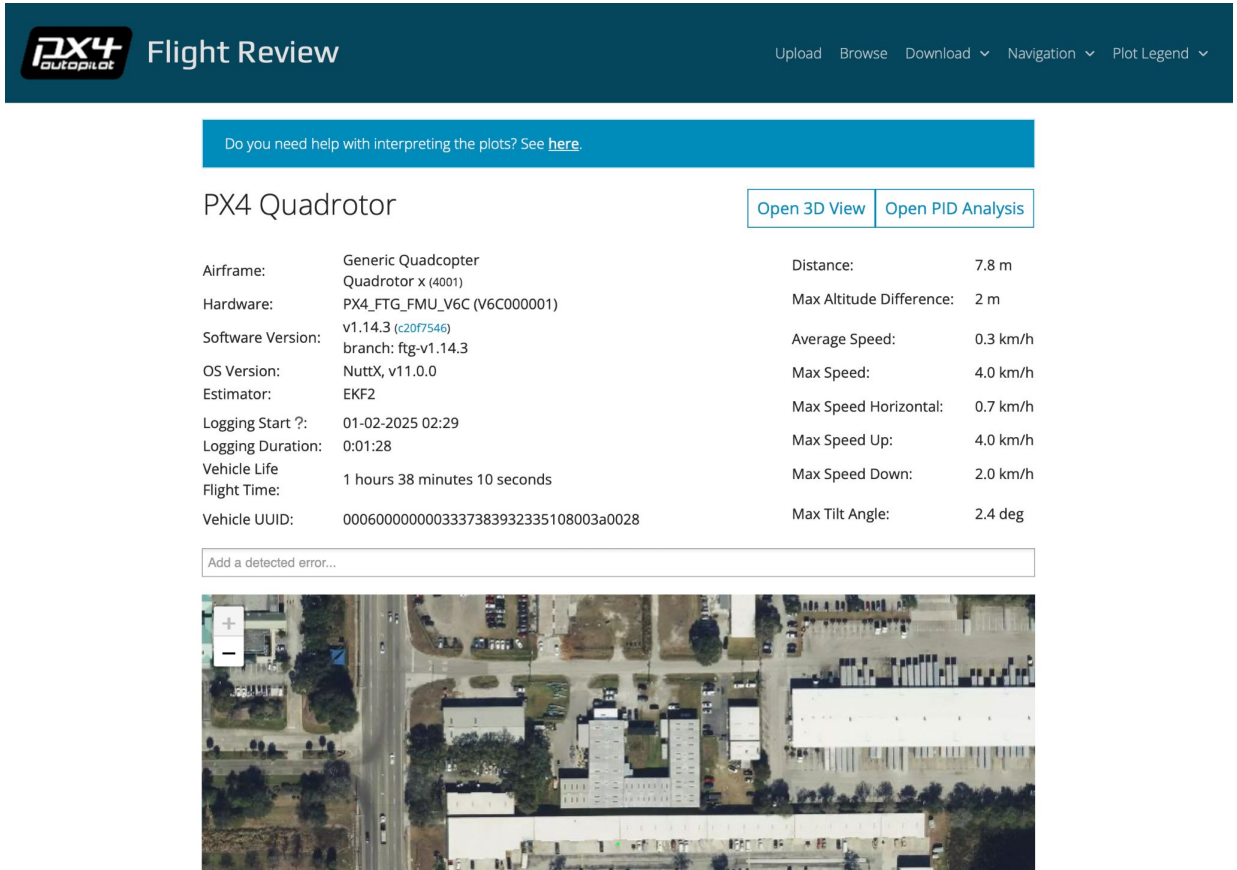
#	Upload Date	Overview	Description	Type	Airframe	Hardware	Software	Duration	Start Time	Rating	Errors	Flight Modes
1	2025-02-01	Not rendered / No GPS		Quadrotor	Generic Quadcopter	CUAV_NORA	v1.15.1	0:00:33	N/A		0	Stabilized
2	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:28	2025-02-01 01:29		0	Position
3	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:00:45	2025-02-01 01:20		0	Position
4	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:23	2025-02-01 01:13		0	Position
5	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:29	2025-02-01 01:05		0	Position
6	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:12	2025-02-01 00:55		0	Position
7	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:11	2025-02-01 00:48		0	Position
8	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:32	2025-02-01 00:29		0	Position
9	2025-02-01	Not rendered / No GPS		Quadrotor	4351	CORVUS_FMU_V6X	v2.9.0	0:11:46	N/A		0	Offboard
10	2025-02-01			Quadrotor	Generic Quadcopter	PX4_FTG_FMU_V6C	v1.14.3	0:01:43	2025-02-01 00:11		0	Altitude, Position

Showing 1 to 10 of 170,903 entries Previous 1 2 3 4 5 ... 17091 Next

PX4 Log Review

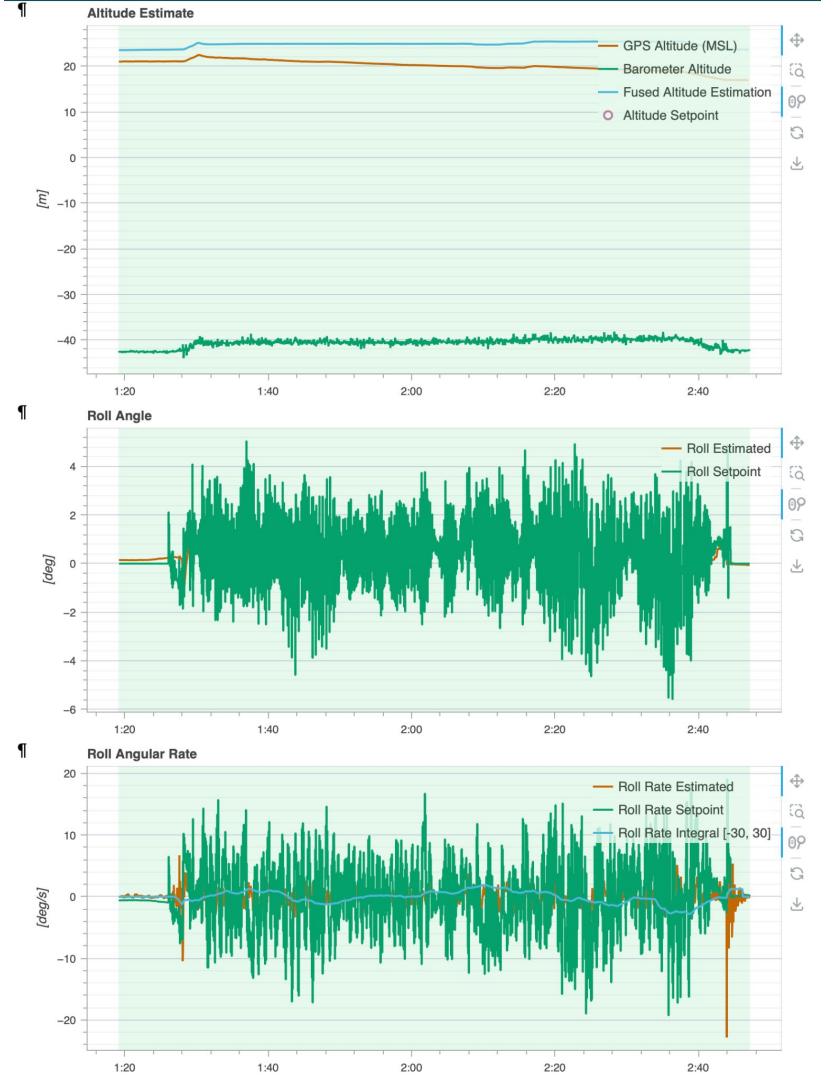


PX4 Log Review



PX4 Log Review

- PID Tracking Performance
- Vibration
- Actuator Controls
- Acceleration Power Spectral Density
- Raw Acceleration
- Raw High-rate IMU Data Plots
- Actuator Outputs
- GPS Uncertainty
- GPS Noise & Jamming
- Thrust and Magnetic Field
- Sampling Regularity of Sensor Data
- Logged Messages
- etc..



How does this work?

afwilkin 2 days ago — Member

Following the tests outlined here

Arm and Take-off
Video: https://www.loom.com/share/6ab405b7b71d426d8a55fa5b50d906de
Log: https://review.px4.io/plot_app?log=1809ab53-0576-4952-83f9-96762e040c66

Flight

Stabilized:

Video: https://www.loom.com/share/6ab405b7b71d426d8a55fa5b50d906de
Log: https://review.px4.io/plot_app?log=1809ab53-0576-4952-83f9-96762e040c66

Altitude:

Video: https://www.loom.com/share/35cc4a5185074605b092a9548852556a?sid=3b0b36de-7eb3-4010-9cb2-84272b707644
Log: https://review.px4.io/plot_app?log=5161578f-2dd4-45b5-aacc-bfc18d68b1fc

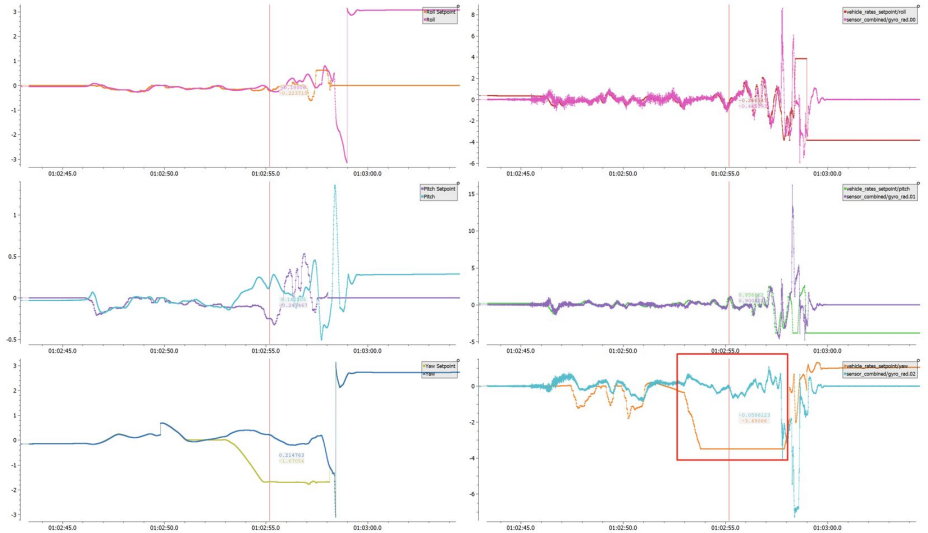
Position

Video: https://www.loom.com/share/ebb87c86ba1c4ca59a090509b13493af?sid=f41d0e77-c455-4ce3-a6a9-68b8f61735ec
Log: https://review.px4.io/plot_app?log=1ded9991-d23a-4409-9c23-e5fa144621e7

Landing
Video: https://www.loom.com/share/2fe0d6627fc742d189a0a98e27732082?sid=c54ebc9f-447e-4abf-875a-355a7c609141
Logs:
https://review.px4.io/plot_app?log=335c5220-8ed4-478d-9153-0e7efbeb9836
https://review.px4.io/plot_app?log=6317f09e-1f3d-4110-8760-360ac2f590a1

MaEtUgR 2 days ago · edited by MaEtUgR — Edits — Member

@afwilkin I had a look at your original log. RC input data looks fine. Translation to setpoints also looks high level ok, tracking is not good and I think this is because the yaw rate and hence also the yaw goes unstable:



Could it be that your vehicle's motor mounts on the arm are not rotated the right way or even rotated slightly the wrong way? That could explain the lack of yaw authority even if the propeller turn direction and motor assignment is correct. For reference the motors should be ever so slightly (2-5°) tilted like indicated on my drawing (assuming props turn in default "cleaning" direction otherwise everything is the opposite way):

Conclusion

Learn More & Get involved

GitHub / Docs / Forums / Discord / Weekly Calls

- Github: PX4, Pixhawk, MAVLink, MAVSDK, QGroundControl
- Documentation
- Forums
- Discord
- Calendar

Enjoy Brussels

Thank You