Blimp Platform Report Requirements

February 5, 2026

1 Purpose and Acceptance Standard

This document defines the required contents and quality standard for a two-part submission:

1. **Part 1: Technical Handoff Package**
2. **Part 2: Team Management Report**

Acceptance Standard

The submission will be considered complete only if:

> A new lab member can rebuild the platform and run the required demos using only the lab GitHub repository and the submitted documentation, with no additional undocumented steps or personal assistance from other lab members.

2 Part 1: Technical Report Requirements

The Technical Report must include all design files for every part, complete electronics documentation, complete code and configuration documentation, and clear instructions to run the system. Any tested but abandoned configuration must be documented, including the reason it was discarded.

2.1 Lab GitHub Requirements

T1. All code must be in the lab GitHub repository.
All code, scripts, configuration files, and documentation must be pushed to the lab GitHub repository. Personal repositories, personal zip packages, and local-only folders are not acceptable. The platform must be runnable by cloning the lab repository.

T2. Repository organization and navigation.
The repository must be organized so a new person can locate and run all components without asking you. At minimum include the following top-level structure (or provide an equivalent structure with a clearly documented directory map in `README.md`):

- `README.md` quick start and directory map
- `docs/` detailed build and software documentation
- `src/` source code
- `configs/` configuration files
- `scripts/` setup and run scripts
- `tests/` acceptance tests and utilities
- `cad/` all mechanical design files
- `electronics/` wiring, pin maps, and interface documentation

T3. Reproducible setup from a clean system.
A lab member must be able to go from a clean Raspberry Pi setup to a working demo using written instructions and a small number of commands.

- Specify Raspberry Pi model, OS image version, kernel (if relevant), and Python version
- Provide pinned dependencies using `requirements.txt` or an equivalent lock file
- Document all interface enablement steps (for example I2C and camera interface)
- Provide setup and run scripts such as `setup.sh` and `run_demo.sh`, or equivalent Makefile targets

T4. Release tag.
Create a tagged release in the lab GitHub repository (for example `platform_report_YYYY_MM`) that corresponds to the documented setup and demos. The report must reference this tag.

2.2 Mechanical Design Requirements

T5. Complete mechanical design package for every part.
Provide all design files for every mechanical part on the platform. For each part include:

- Native CAD file
- STEP export
- STL export if 3D printed
- Print settings: material, layer height, infill, supports, orientation
- Assembly notes: fasteners, installation order, photos, and common failure points

T6. Assembly instructions with photos.
Provide step-by-step assembly instructions, including how parts align, how wires are routed, and how sensors are mounted. Photos must be included for non-obvious steps.

2.3 Electronics and Wiring Requirements

T7. Electronics bill of materials.
Provide a complete electronics BOM with vendor, part numbers, quantity, and acceptable substitutions.

T8. Wiring diagram and pin map.
Provide a wiring diagram and a pin map that includes power distribution and all signal connections.

T9. Power architecture and safety.
Document the power tree (battery, regulators, fuses, switch), expected current draw by subsystem, and safety considerations.

T10. Interfaces and addresses.
Document all hardware interfaces and key parameters, including I2C addresses, UART baud rates, SPI configuration, PWM mapping, connector types, and any required kernel overlays or interface settings.

T11. Mounting and reliability notes.
Document how sensors and electronics are mechanically secured, including brackets, adhesives, fasteners, and vibration isolation. Any reliability issues observed during tests and the implemented fix must be recorded.

2.4 Software and Operation Requirements

T12. Software stack documentation.
Document the software stack, including:

- Repository structure and module purpose
- Dependencies and installation steps
- Configuration parameters and where they are stored
- Logging format and where logs are saved

T13. How the system works.
Provide a clear explanation of how the system works:

- Architecture diagram: sensors, compute, control, actuation
- Control loop description: estimation, filtering, controller, actuation
- Vision autonomy pipeline description: inputs, outputs, algorithms, limitations
- Gate control logic: state machine and interfaces

T14. How to run each capability.
Document where to find each component and how to run it. At minimum include commands for:

- Sensor bring-up and verification
- Hover control demo
- Camera bring-up and streaming verification
- Vision autonomy demo
- Cage gate actuation demo

Include expected outputs and an explicit method to confirm pass.

T15. Validation evidence and acceptance tests.
Any claim that something works must be backed by evidence and a written acceptance test procedure.

- Provide acceptance tests with pass/fail criteria for hover, gate actuation, cameras, and reliability
- Provide logs, plots, and or short recorded evidence tied to a specific commit hash
- Provide safety checklist and test environment notes

T16. Known issues and troubleshooting guide.
Provide a prioritized list of open issues and a troubleshooting guide organized by symptom, including hardware blockers and integration issues.

T17. Decisions log for tested but abandoned configurations.
Any configuration that was tested but abandoned must be documented in a decisions log table. Each entry must include:

- What was tried: hardware, wiring, firmware, algorithm, settings
- How it was tested
- Results and evidence
- Reason for discarding
- What would be needed to revisit

3 Part 2: Team Management Report Requirements

The Team Management Report must document roles, assigned tasks, milestone planning, and day-to-day working cadence.

M1. Team roster, roles, and ownership.
List every team member and their role. Provide an ownership map for mechanical, electrical, controls, autonomy, testing, documentation, and integration. Identify who is responsible for integration and final merges.

M2. Assigned tasks and current status.
Provide a task list organized by subsystem with:

- Owner for each task
- Expected output for each task
- Current status
- Links to related GitHub issues and commits

M3. Weekly milestones plan versus actual.
For each week in the reporting period include:

- Planned milestone
- Delivered output
- Evidence: commit hash, test log, photo, or demo reference
- Blockers and how they were addressed

M4. Planned daily working hours and cadence.
Provide the planned day-to-day working hours for each team member and the planned meeting cadence. Include how blockers are escalated and expected response times.

M5. Process and quality control.
Document the process that will be used going forward:

- Definition of done and test gate required before declaring a feature complete
- Code review and merge policy
- Documentation review policy
- Hardware change control process for CAD and wiring revisions

4 Appendix: Required Tables and Templates

This appendix lists the minimum tables that must be included in the submission. You may add more as needed.

A. Decisions Log (Tested and Abandoned Configurations)

Configuration Tried	How Tested	Result	Evidence	Reason Discarded	Revisit Notes

B. Acceptance Tests Summary

Test	Procedure Summary	Pass Criteria	Evidence (Commit, Log)