

Пошаговые процедуры

Тест 1 — Запуск ПО в Docker

Цель: убедиться, что стек поднимается без ошибок. Требуется изначальное подключение к интернету в аудитории.

Шаги:

1. В корне проекта выполнить `docker compose up -d --build`.
2. Подождать 30–60 с. Выполнить `docker compose ps`.
3. Открыть в браузере `http://localhost:8001/` → проверить `/docs`
4. Если ошибки в логах — записать их.

Ожидаемый результат: все контейнеры UP, `/docs` отвечает 200.

Тест 2 — Регистрация / Авторизация (JWT)

Цель: проверка регистрации и получения JWT.

Шаги:

1. Перейти по кнопке «Войти», и затем на вкладку «Регистрация», заполнить форму валидными данными. Нажать «Регистрация».
2. Перейти на вкладку «Войти», ввести учётные данные.
3. В БД (psql) проверить запись в таблице `users`.

Ожидаемый результат: регистрация успешна (201), вход успешен (200), запись в БД.

Тест 3 — Загрузка облака в БД

Цель: загрузить файл облака через UI и сохранить эксперимент.

Шаги:

1. Перейти на страницу «Создание облака точек», заполнить метаданные, загрузить `scan_sample.txt`.
2. Нажать «Сохранить эксперимент».
3. Через API/DB проверить запись метаданных и наличие файла в `scans/`.

Ожидаемый результат: запись в БД + файл в папке `scans/`.

Тест 4 — Визуализация облака точек

Цель: проверить отображение в 3D-вьюере.

Шаги:

1. Открыть страницу «Просмотр облака точек».

2. Дождаться рендера, проверить вращение/зум/панораму.
3. Оценить на наличие артефактов.

Ожидаемый результат: облако визуализируется корректно, без JS-ошибок.

Тест 5 — Подключение к ЦХД и синхронизация

Цель: проверить синхронизацию с ЦХД.

Шаги:

1. Открыть страницу «Подключение к ЦХД», ввести данные и подключиться.
2. Нажать «Синхронизация».
3. Проверить, что метаданные и файлы доступны в ЦХД; данные на облаке и в локальной бд совпадают

Ожидаемый результат: файлы доступны, файлы совпадают.

Тест 6 — SSH подключение к плате

Цель: проверить удалённый доступ и SFTP. Для работы с платой необходимо предварительно отключиться от wi-fi аудитории и подключиться к wi-fi платы.

Шаги:

1. Перейти на страницу «Снять облако точек»
2. Нажать в UI «Проверить соединение» — дождаться пинга.
3. Дождаться положительного пинга с лидара

Ожидаемый результат: ssh возвращает положительный ответ в консоли в браузере

Тест 7 — Проверка LiDAR (lidar.py)

Цель: проверить работу датчика.

Шаги:

1. Подключиться к плате при помощи нажатия кнопки «Подключиться к плате»
2. Дождаться положительного результата операции в консоли
3. Перейти в раздел параметры съемки
4. Нажать на кнопку «Проверка лидара»
5. Дождаться выполнения проверки съемки лидара (не более 30 секунд)
6. Проверить результаты лога съемки лидара

Ожидаемый результат: скрипт завершается успешно, данные в диапазоне.

Тест 8 — Проверка двигателя (engine.py)

Цель: проверить поворот платформы.

Шаги:

1. На плате/через UI запустить «Проверка двигателя».
 2. Дождаться выполнения проверки двигателя, визуально убедиться, что двигатель повернулся на 240 градусов и вернулся в исходное положение
- Ожидаемый результат:** лог успешной проверки двигателя
-

Тест 9 — Съёмка облака (вариант 1)

Параметры: шаг = 5°, длительность = 1s, угол 90 градусов

Шаги:

1. Установить заданные параметры.
 2. Нажать «Запуск съёмки». Дождаться завершения.
 3. Визуально убедиться, что характер съёмки соответствует заданным параметрам
- Ожидаемый результат:** корректное выполнение процесса съёмки облака точек
-

Тест 10 — Съёмка облака (вариант 2)

Параметры: шаг = 2°, длительность = 3s, угол 90 градусов

Шаги:

1. Установить заданные параметры.
 2. Нажать «Запуск съёмки». Дождаться завершения.
 3. Визуально убедиться, что характер съёмки соответствует заданным параметрам
- Ожидаемый результат:** корректное выполнение процесса съёмки облака точек
-

Тест 11 — Скачивание облака (SFTP)

Цель: проверить скачивание и корректность файла облака точек. Так как будем проверять скаченный в установленное место файл

Шаги:

1. После успешной съёмки облака точек (Тест 10) нажать на кнопку «Скачать облако точек»
 2. Проверить скаченный файл в папке /scans в корне проекта (если запуск выполнялся через uvicorn) либо в папке /downloads (если запуск выполнялся через docker) куда все данные скачиваются на локальном компьютере
 3. Проверить содержимое файла, что в нем действительно было снято облако точек
- Ожидаемый результат:** файл с записанным облаком точек
-

Тест 12 — Полный рабочий цикл (E2E)

Цель: проверить end-to-end процесс. Так как визуализатору и ЦХД необходимо подключение к интернету, то после съёмки облака точек будет выполнено отключение от wi-fi платы и подключение к wi-fi с интернетом

Шаги: последовательно выполнить:

- регистрация
- авторизация
- подключение к плате
- съёмка облака точек
- скачивание файла
- загрузка в локальную БД облака точек
- визуализация
- загрузка локального облака точек в ЦХД.

Ожидаемый результат: все этапы выполняются, ПО выполняет свой функционал, при просмотре данные из ЦХД имеют правильную визуализацию.

Тест 13 — Временные характеристики запуска системы

(Покрывает: время загрузки платы, время запуска ПО, время запуска установки)

Цель: Определить производительность системы при холодном старте.

Шаги:

1. Полностью обесточить Raspberry Pi. Подготовить секундомер.
2. Подать питание и запустить секундомер (**Время запуска установки**).
3. Остановить отсечку времени, когда плата станет доступна по пингу/SSH (**Время загрузки платы**).
4. Подключиться по SSH и выполнить команду запуска контейнеров (docker compose up). Засечь время до появления сообщения Uvicorn running on... или доступности Swagger UI (**Время запуска ПО**).

Ожидаемый результат:

- Время загрузки платы: ≤ 60 сек.
- Время запуска ПО: ≤ 40 сек.
- Общее время готовности: ≤ 2 мин.

Тест 14 — Временные характеристики диагностических процедур

(Покрывает: время проверки двигателя лидара, время проверки подключений)

Цель: Убедиться, что pre-flight проверки не занимают много времени у оператора.

Шаги:

1. В веб-интерфейсе нажать кнопку «Проверить соединение». Засечь время до появления статуса «Подключено» или галочки.
2. Нажать кнопку «Проверка двигателя» (или «Проверка лидара», в зависимости от того, что тестирует мотор самого датчика). Засечь время от нажатия до получения статуса ОК в логге.

Ожидаемый результат:

- Время проверки подключений: ≤ 5 сек.
- Время проверки двигателя: ≤ 10 сек (без учета времени на физический тест, только отклик системы).

Тест 15 — Точность механического поворота на 240 градусов

(Покрывает: точность поворота на 240 градусов)

Цель: Подтвердить точность позиционирования платформы.

Оборудование: Транспортир или заранее нанесенная разметка на основании платформы (метки 0° и 240°).

Шаги:

1. Выставить платформу в нулевое положение.
2. В интерфейсе или через скрипт `engine.py` задать команду поворота ровно на 240 градусов.
3. После остановки двигателя визуально или инструментом замерить отклонение риски на поворотной части от метки 240° на основании.

Ожидаемый результат: Отклонение не превышает 1–2 градуса.

Тест 16 — Границы диапазона измерения (Min/Max)

(Покрывает: минимальное расстояние съемки, максимальное расстояние съемки)

Цель: Проверить заявленные характеристики RPLIDAR A1 (0.15м – 12м).

Шаги:

1. Разместить плоский объект (лист картона) на расстоянии 14–15 см от лидара.
2. Запустить короткую съемку. Проверить полученный файл: наличие точек с дистанцией ~0.15м.
3. Обеспечить наличие стены или объекта на расстоянии 11–12 метров (коридор).
4. Запустить съемку. Проверить файл: наличие точек с дистанцией ~12м.

Ожидаемый результат: Лидар регистрирует препятствия на границах заявленного диапазона.

Тест 17 — Анализ потока данных (Скорость и Качество)

(Покрывает: скорость съемки точек, процент точек с не максимальным качеством)

Цель: Оценить эффективность сбора данных.

Шаги:

1. Запустить съемку длительностью ровно 10 секунд (через скрипт или UI).
2. Открыть полученный файл данных (TXT/CSV).
3. **Скорость:** Разделить общее количество строк (точек) на 10 сек.
4. **Качество:** Если формат данных RPLIDAR содержит колонку Quality (обычно 0–15 или 0–255), подсчитать количество строк, где `Quality < Max_Quality`. Вычислить процент от общего числа.

Ожидаемый результат:

- Скорость съемки: близка к 8000 точек/сек (для RPLIDAR A1).
- Процент точек с низким качеством: не более 10–15% (зависит от освещенности и отражающей способности поверхностей в помещении).

Тест 18 — Дальность беспроводной связи

(Покрывает: дальность беспроводной связи)

Цель: Определить рабочий радиус действия Wi-Fi модуля Raspberry Pi 3B+.

Шаги:

1. Установить установку стационарно.
2. Оператору с ноутбуком (подключенным к Wi-Fi установки) запустить в консоли команду `ping <IP_ADDRESS_RPI> -t`.
3. Медленно удаляться от установки, следя за временем отклика (ms) и потерей пакетов.

4. Зафиксировать расстояние, на котором начнутся регулярные потери пакетов (Timeouts).

Ожидаемый результат: Устойчивая связь на расстоянии не менее 10 метров в условиях прямой видимости (встроенная антенна RPi 3B+).