



Endpoints Specification (Swagger/OpenAPI)

Team #3, SpaceY

Zifan Si

Jianqing Liu

Mike Chen

Xiaotian Lou

January 14, 2026

1 Revision History

Date	Version	Notes
Jan. 14, 2026	Rev 0	Initial Draft

Contents

1 Revision History	i
2 REST API Communication Protocol (OpenAPI)	1
2.1 Transport and Data Format	1
2.2 Servers / Deployment Targets	1
2.3 Communication Model	1
2.4 Endpoint Reference	1
2.5 Schemas (JSON Data Types)	2
2.5.1 StatusResponse	2
2.5.2 BoundingBox (nullable)	2
2.5.3 ManualMoveRequest	2
2.5.4 ManualMoveToRequest	2
2.5.5 EmptyResponse	2
2.6 Example Request/Response Flows	3

2 REST API Communication Protocol (OpenAPI)

This project provides an HTTP REST API for controlling and monitoring the gimbal system. The API contract is documented using an OpenAPI specification and is viewable in Swagger Editor (<https://editor.swagger.io/>). The API is used by the front-end/controller to issue commands and query current system state.

2.1 Transport and Data Format

- **Transport:** HTTP
- **Style:** REST-style endpoints under /api/*
- **Content Type:** application/json
- **Encoding:** JSON request/response bodies

2.2 Servers / Deployment Targets

Swagger UI provides local server presets for development:

- **Frontend development default:** http://localhost:5000
- **Flask default (port 80):** http://localhost

2.3 Communication Model

The REST API follows a **request-response** model:

1. The client initiates requests to the back-end service.
2. The server responds with an HTTP status code and a JSON response body.
3. Action endpoints (e.g., arm/disarm) may return an empty response body and rely on the HTTP status code to indicate success.

2.4 Endpoint Reference

Table 1 summarizes the endpoints.

Method	Path	Purpose
POST	/api/status	Get current system status.
POST	/api/manual_move	Move the gimbal in a direction (relative control).
POST	/api/manual_move_to	Move the gimbal to absolute angles (pan/tilt targets).
POST	/api/arm	Arm tracking (enable tracking loop).
POST	/api/disarm	Disarm tracking (disable tracking loop).

Table 1: REST API endpoints (Swagger/OpenAPI documented).

2.5 Schemas (JSON Data Types)

Swagger UI defines the following schemas.

2.5.1 StatusResponse

Returned by POST /api/status.

- **armed** (required): tracking armed state
- **tilt** (required): current tilt angle
- **pan** (required): current pan angle
- **preview** (required): preview-related state
- **bbox** (nullable): optional bounding box of detected target

2.5.2 BoundingBox (nullable)

- **pts_s** (required): points/coordinate representation
- **conf** (required): confidence score
- **left** (required): left coordinate
- **top** (required): top coordinate
- **width** (required): width
- **height** (required): height

2.5.3 ManualMoveRequest

Request body for POST /api/manual_move.

- **direction** (required): direction command (e.g., up/down/left/right)

2.5.4 ManualMoveToRequest

Request body for POST /api/manual_move_to.

- **tilt** (required): target tilt angle
- **pan** (required): target pan angle

2.5.5 EmptyResponse

Used for action endpoints (e.g., /api/arm, /api/disarm) where a success status code indicates completion and no response fields are required.

2.6 Example Request/Response Flows

Get Status

```
POST /api/status  
Content-Type: application/json
```

Manual Move (Directional)

```
POST /api/manual_move  
Content-Type: application/json  
  
{"direction":"left"}
```

Manual Move (Absolute Angles)

```
POST /api/manual_move_to  
Content-Type: application/json  
  
{"tilt":10.0,"pan":-5.0}
```

Arm / Disarm Tracking

```
POST /api/arm  
POST /api/disarm
```