AiBridge Technical Specification v7.12.1

1.0 Overview

AiBridge is an intelligent telescope control interface designed to eliminate the disconnect between diverse control protocols and hardware in the field of astronomical observation. This software runs on ESP32-based hardware and integrates both the traditional LX200 protocol and the latest ASCOM Alpaca standard. By doing so, it acts as a strategically important bridge, enabling seamless operation of traditional telescope mounts by modern control applications such as SkySafari, NINA, and Stellarium.

1.1 Product Information

Item	**Details**
Product Name	AiBridge - AI-Controllable Telescope Interface
Version	7.12.1
Copyright Holder	Copyright (c) 2025 Nishioka Sadahiko / 西岡定彦
License	MIT License
Contact Email	nishioka.sst@gmail.com
Official Website	https://nskikaku.blog.fc2.com/

GitHub Repo	https://github.com/OnStepNinja

1.2 System Architecture

AiBridge acts as an integrated control hub, receiving connections from various clients over Wi-Fi and converting them into serial commands interpretable by the telescope mount. Its architecture is composed of interactions between the following main components:

1. Client Applications: Astronomy control software such as SkySafari, NINA, Stellarium, etc., connect to AiBridge via Wi-Fi. Either the LX200 protocol (TCP) or ASCOM Alpaca protocol (HTTP) is used for the connection.
2. AiBridge Main Unit: The core ESP32 module of the system. Built-in servers—a Web server (HTTP), a TCP server (LX200), and a UDP listener (Alpaca Discovery)—constantly await requests from each protocol and internally interpret/convert received commands.
3. Telescope Mount: Equatorial or Alt-Az mounts compatible with serial communication such as OnStep. AiBridge sends converted serial commands via UART to the mount for physical control.
4. External Devices: AiBridge also provides proxy (relay) functionality for controlling other Alpaca-compliant devices (focusers, cameras, domes, etc.) on the same network.

Essentially, AiBridge acts as a real-time translation engine, parsing commands received via diverse high-level protocols such as HTTP/Alpaca or TCP/LX200 and converting them into a unified serial command stream that the telescope mount can interpret. The hardware specs described below physically realize this flexible architecture.

2.0 Hardware Specifications

Full utilization of AiBridge software requires correct wiring of the ESP32 module's GPIO pins and proper configuration of serial ports (UART). Notably, input pins specified as INPUT_PULLUP (GPIO39, 33, 32) are internally pulled up at the firmware level, making external pull-up resistors unnecessary and simplifying circuit design. This section provides the technological foundations required for hardware implementation, specifying details to ensure stable system operation.

2.1 GPIO Pin Assignments

Constant Name	GPIO Pin No.	Function	Type/Characteristic
FILE_MANAGER_LED_PIN	25	File Manager enabled state LED	OUTPUT
FILE_MANAGER_ENABLE_PIN	39	File Manager enable button	INPUT_PULLUP
CUSTOM_UI_SELECT_PIN	33	Custom UI select	INPUT_PULLUP
SERIALONSTEP_SELECT_PIN	32	UART switch for OnStep connection	INPUT_PULLUP
DHT_PIN	5	DHT22/11 temperature/humidity sensor	INPUT
SWITCH_0_PIN	27	Alpaca Switch output 1	OUTPUT
SWITCH_1_PIN	14	Alpaca Switch output 2	OUTPUT
SWITCH_2_PIN	12	Alpaca Switch output 3	OUTPUT

SWITCH_3_PIN	13	Alpaca Switch output 4	OUTPUT
DEBUG_TX	19	Serial TX for debugging	OUTPUT
DEBUG_RX	18	Serial RX for debugging	INPUT

2.2 Serial Ports (UART) Configuration

AiBridge configures serial ports for different purposes.

- OnStep Connection Port (SerialOnStep): Main port for communication with the telescope mount. The physical UART used is dynamically switched based on the input level at GPIO32 at startup—enabling greater hardware design flexibility.
    - GPIO32 HIGH: UART0 (TX: GPIO1, RX: GPIO3) is used.
    - GPIO32 LOW: UART2 (TX: GPIO17, RX: GPIO16) is used.
    - Baud rate: 9600 baud
- Debug Port (SerialDebug): Dedicated port for real-time monitoring of system status and error messages during development or troubleshooting.
    - UART Used: UART1 (TX: GPIO19, RX: GPIO18)
    - Baud rate: 115200 baud

Once physical connections are complete, logical network configuration becomes important.

3.0 Network Settings and Protocols

AiBridge's network functions define its operational flexibility. Simultaneous support for Access Point (AP) mode and Station (STA) mode (WIFI_AP_STA) enables both direct connection in Wi-Fi-less observation fields and seamless integration with home networks using a single device.

3.1 Wi-Fi Modes

AiBridge operates in WIFI_AP_STA mode, providing both AP and STA functions simultaneously.

- Access Point (AP) Mode
    - Purpose: Used where no available Wi-Fi networks exist (e.g., outdoor observing sites) to connect directly from a smartphone or PC.
    - SSID: AiBridge5000_AP
    - Password: password
    - IP Address: 192.168.4.1 (static)
- Station (STA) Mode
    - Purpose: Used to connect AiBridge to an existing home Wi-Fi network for operation on the same LAN with other devices.
    - IP Address Assignment: Automatic via DHCP.
    - Settings Method: Write the destination network's SSID and password in aibridge_net_cfg.dat on SPIFFS (internal flash memory). This file can be safely edited via the file management system through a web browser.

3.2 Supported Protocols

AiBridge supports two major protocols widely used in astronomy.

- LX200 Protocol
    - Port: Services provided on TCP ports 9999 and 9998.
    - Features: By having two dedicated ports, allows multiple client applications (for example, two instances of SkySafari on different devices) to connect concurrently.
- ASCOM Alpaca Protocol
    - Port: Communication is on HTTP (port 80), adopting the RESTful API format for strong compatibility with modern software.
    - Device Discovery: Listens on UDP port 32227 for discovery requests, enabling Alpaca-compliant software like NINA/Stellarium to automatically find AiBridge on the network.

Detailed specification of the supported HTTP APIs follows.

4.0 API Specification

AiBridge APIs are structured into three distinct layers, each assigned a specific role. First, a "Direct Control API" for controlling OnStep mounts directly connected to AiBridge via serial. Second, the "Alpaca Device API" which standardizes AiBridge itself as an Alpaca device per ASCOM standards. Third, an "External Device Proxy

API" that acts as a relay for other Alpaca devices on the network. This section systematically explains these API endpoints based on openapi_v7.11.json.

4.1 OnStep Telescope Direct Control API

A set of proprietary APIs for simple control of OnStep mounts directly connected to AiBridge. Coordinates use HMS/DMS format (HH:MM:SS, ±DD:MM:SS).

- GET /api/status
  - Function: Acquires the telescope's current RA/Dec and detailed operational status.
  - Example Response: {"status":"success", "data":{"ra":"12:30:45","dec":"+45:30:00","state":"Tracking",...
- GET /api/goto
  - Function: Automatically slews the telescope to the specified RA/Dec.
  - Parameters: ra, dec
  - Example Response: {"status":"success", "message":"GOTO command sent", ...}
- GET /api/command
  - Function: Sends raw LX200 command directly.
  - Parameters: cmd
- GET /api/stop
  - Function: Emergency stops all telescope movement.

4.2 ASCOM Alpaca Device API

AiBridge exposes the following three types of virtual devices to the network as per ASCOM Alpaca standard. Note: The coordinate format is decimal (RA: hours, Dec: degrees).

4.2.1 Management

Endpoints that provide device info and detection. GET /management/v1/configureddevices returns a JSON list of all Alpaca devices (Telescope, ObservingConditions, Switch) available on AiBridge.

4.2.2 Telescope

Endpoints for Alpaca-standard telescope control.

- Coordinates/Status:

    - GET /api/v1/telescope/0/rightascension
    - GET /api/v1/telescope/0/declination
    - GET /api/v1/telescope/0/tracking
    - GET /api/v1/telescope/0/slewing
- Actions:
    - PUT /api/v1/telescope/0/slewtocoordinatesasync
- Autoguide:
    - GET /api/v1/telescope/0/canpulseguide
    - PUT /api/v1/telescope/0/pulseguide
    - GET /api/v1/telescope/0/ispulseguiding

Note: For broad client compatibility, core action endpoints such as slewtocoordinatesasync and abortslew accept both PUT and GET requests, but PUT is the standard as per Alpaca.

4.2.3 Observing Conditions

Publishes info from a DHT22/DHT11 temp/humidity sensor attached to GPIO5.

- GET /api/v1/observingconditions/0/temperature: Retrieve ambient temperature (C).
- GET /api/v1/observingconditions/0/humidity: Retrieve ambient relative humidity (%).
- Error Handling: If the sensor is disconnected or reading fails, returns the error value -999.0.

4.2.4 Switch

APIs to control the 4 relay outputs; each switch corresponds directly to a designated GPIO pin.

- GET /api/v1/switch/0/maxswitch: Returns total number of switches (4).
- GET /api/v1/switch/0/getswitch: Gets ON/OFF state for a specified switch (Id: 0-3).
- PUT /api/v1/switch/0/setswitch: Sets the state of a specified switch.

Switch Index	**GPIO Pin**
Switch 0	GPIO27

Switch 1	GPIO14
Switch 2	GPIO12
Switch 3	GPIO13

4.3 External Alpaca Device Proxy API

Elevates AiBridge from a simple interface to a comprehensive control hub for the entire observation system.

- Device Discovery: GET /api/external/discover endpoint sends a UDP broadcast, scans the network, and returns a list of responsive Alpaca devices with IP addresses, port numbers, and device info (model, type, etc.). The user does not need to know device IPs in advance.
- General Proxy: GET /api/external/alpaca (in some docs also aliased as /api/external/command) is a highly flexible universal endpoint for proxying any property or method of any Alpaca device on the network. By specifying parameters such as target (IP), device (device type), property (property/method name), one can communicate with and control any Alpaca device (telescope, focuser, camera, etc.) and return results to the client.

5.0 Functional Specifications

In addition to programmatic APIs, AiBridge implements physical operations and intelligent automation functions, elevating the device from a mere protocol converter to an autonomous, field-ready observation instrument.

5.1 File Management System

AiBridge is equipped with file management functionality for operating internal storage (SPIFFS) via web browser.

- Purpose:
    1. Edit Wi-Fi connection settings (aibridge_net_cfg.dat).

2. Upload custom user interfaces (index2.html).
3. Format SPIFFS and check usage status.

- Security Mechanisms: Strict security mechanisms involving physical interaction are implemented to prevent unauthorized access:
    1. File Manager access is limited to the first 30 minutes after system startup.
    2. A physical button (GPIO39) must be pressed within this time to enable access.
    3. Once the button is pressed, access is granted for only 10 minutes.
    4. Access state can be visually confirmed by the unit's LED (GPIO25) lighting up. After 10 minutes, the LED turns off and access is disabled again.

5.2 Telescope Status Monitoring

AiBridge internally monitors the state of connected telescopes using an intelligent system.

- Monitoring Logic: Every 3 seconds, current coordinates are read and compared with previous ones to calculate motion speeds in RA/Dec.
- Status Judgment: Based on speed ratio (compared to the sidereal rate: 15.041 arcsec/s), telescope operation is automatically classified as:
    - Tracking: speed ratio < 0.4 (less than x0.4 sidereal rate)
    - Guiding: speed ratio 0.4 or more, less than 5.0
    - Slewing: speed ratio >= 5.0 (x5 or more)

With this, the Alpaca API's slewing property can report accurate status to clients even if the underlying OnStep controller lacks native high-level status flags.

5.3 Custom UI

Allows users to replace the standard control panel (AiBridge Console) with their own custom UI.

- Activation Requirements:
    1. Upload a custom HTML file named index2.html into SPIFFS via the file management system.
    2. At system startup, ensure GPIO33 pin is set LOW (connected to GND).
- Behavior: When both conditions are met, accessing AiBridge's root URL (/) displays the uploaded index2.html instead of the standard console. If index2.html does not exist, the system automatically falls back to the standard console for safety.

5.4 Location Name

Enhances identification when operating multiple AiBridge devices on the same network.

- Setup: Enter any name (e.g., "MainScope", "GuideScope") into the "Location Name" field of the Wi-Fi setup screen in the file manager.
- Behavior: The entered name is automatically prefixed with "Ai-" (e.g., "Ai-MainScope") and published on the network as the Alpaca device name.
- Advantage: This enables software like NINA to show, for example, "Ai-MainScope Telescope" and "Ai-GuideScope Telescope," making it instantly clear which device is being operated and preventing operational mistakes.