RJ-08 SERIES
SAS4 HIGH DENSITY JBOD
USER GUIDE

Version 1.0

2025/11/7

1 Table of Contents

2 Introduction

2.1 Overview

This manual introduces **IW-RJ-460-08 (SAS4)** and **IW-RJ-472-08 (SAS4)** Just-a-Bunch-of-Disks (JBOD) enclosures. These models are next-generation updates of the well-known RJ-460-08 (4U/60-bay) and RJ-472-08 (4U/72-bay) platforms, redesigned around Broadcom SAS4x48 expanders to deliver 24G SAS throughput while remaining fully backward-compatible with 12G SAS3 and 6G SATA drives and hosts.

This document is intended for storage administrators, system integrators, and OEM partners who deploy, operate, or service high-density SAS JBODs in data-center and enterprise environments.

Model	***Form factor & bays***	***SAS domain layout***	***Typical external host ports***	***Typical expansion port***
IW-RJ-460-08 SAS4	4U, 60 bays	1 Bridge + 2 Edge expanders (tree topology)	up to 4× SFF-8674	1× SFF-8674
IW-RJ-472-08 SAS4	4U, 72 bays	1 Bridge + 3 Edge expanders (tree topology)	up to 4× SFF-8674	1× SFF-8674

Notes

• Drive backplanes are organized identically to prior SAS3 models: RJ-460 uses three expanders (bridge + two edges); RJ-472 uses four (bridge + three edges). This preserves host mapping, zoning options, and management behavior familiar from the SAS3 generation.

2.2 What's New in the SAS4 Generation

- Broadcom SAS4x48 expander silicon
    - 24G SAS link rate per lane (2× SAS3 bandwidth).
    - Higher device fan-out with improved fairness and arbitration.
- Updated cabling

  - Internal: MCIO 8i/4i harnesses between bridge and edge expanders for low-loss, 24G-ready routing.
  - External: SFF-8674 host/expansion ports on the bridge module.
  - On board edge expander (good airflow for cooling)

- Seamless command compatibility

  The CLI command set and workflows remain the same as the SAS3 versions (help, rev, showmfg, sasaddr, date, showlogs/clearlogs, showpost, sesinfo, sensor, disk, console, bpsys, ipconfig, fdl/tftpfdl, user). Administrators can reuse existing procedures and scripts with minimal or no changes.

- Backward compatibility

  Operates with 12G SAS3 HBAs/RAIDs and 6G SATA drives; mixed-rate operation follows SAS negotiation rules.

2.3 Key Features

- High throughput: 24G SAS link speed; compatible with 12G SAS3 and 6G SATA drives.
- Dual-domain support: Optional second expander domain for path redundancy and high availability.
- Hot-swappable FRUs: CRPS power supplies, fan modules, expanders and drive carriers; service without powering down the enclosure.
- Smart cooling: Multi-level fan speed control with thermal monitoring and audible/LED alarms.
- Flexible zoning: Divide bays into independent groups to serve multiple hosts.
- Cascading: Expansion port enables enclosure-to-enclosure scaling.
- Network & console management: 1G Ethernet (HTTP/Telnet/IPMI) and RS-232 console with a proven, scriptable CLI; SES enclosure services for integration with host utilities and RAID/HBA GUIs.
- Module hot-swap capability may vary by configuration and service policy.

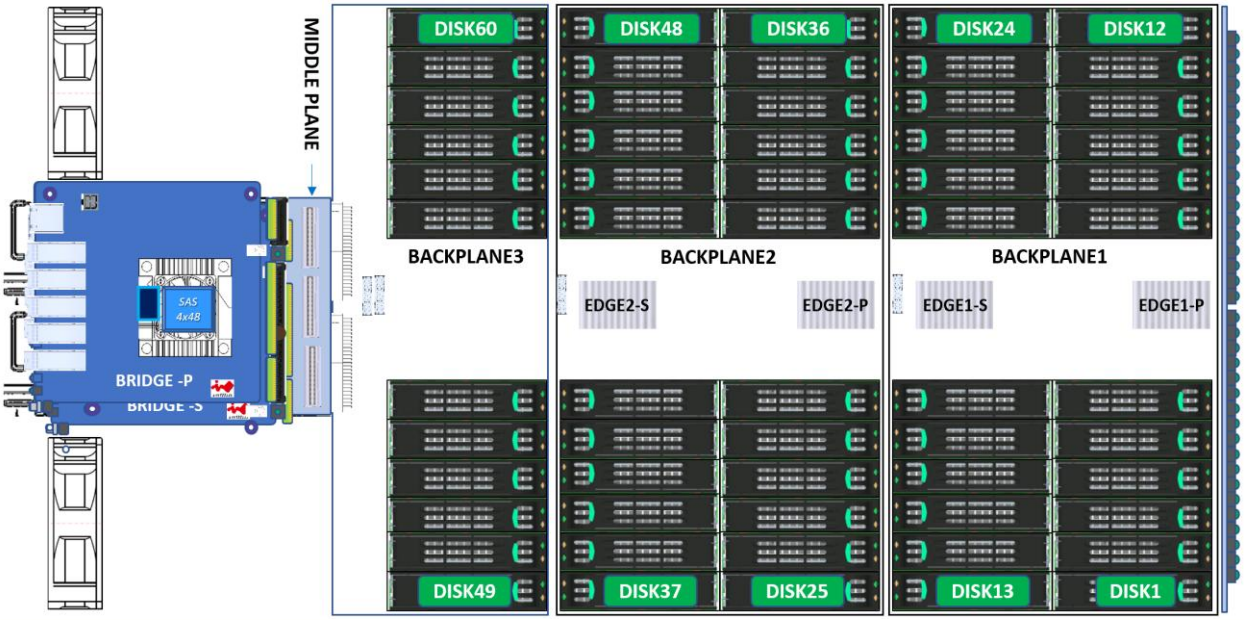

FIGURE 1 RJ-460 INTERNAL BACKPLANES

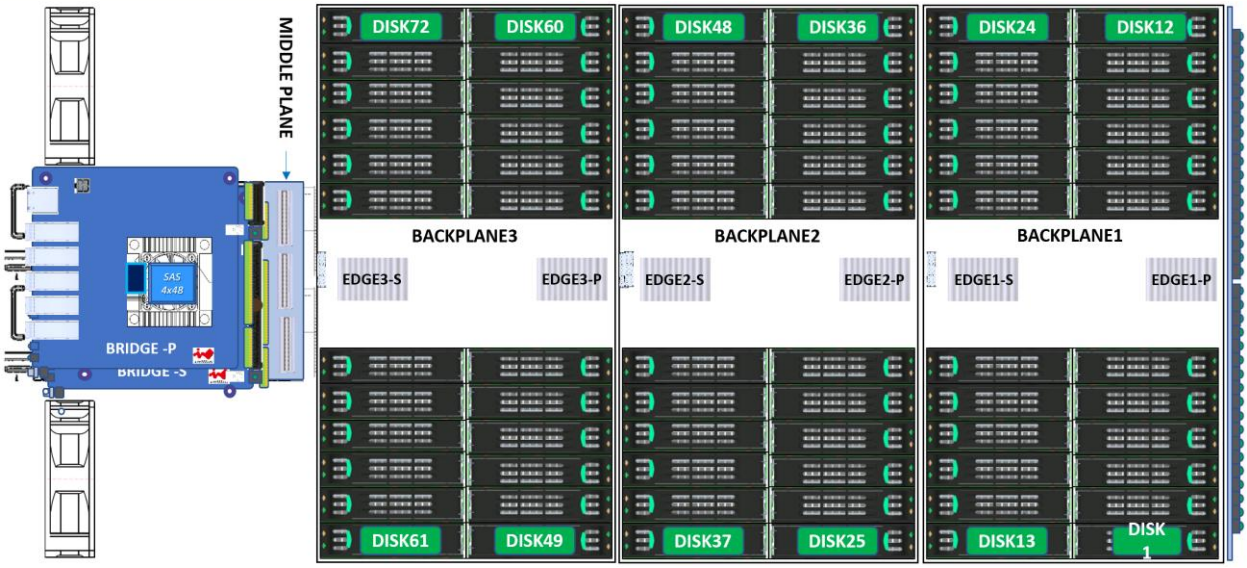

FIGURE 2 RJ-472 INTERNAL BACKPLANES

2.4 SAS Topology

Tree topology reduces the number of cascade levels (or hops), avoiding or reducing the unfair arbitration that can affect the far-end devices in cascade topology. RJ-460 and RJ-472 use tree topology to reduce the number of cascade levels, avoiding or reducing the unfair arbitration. There are 3/4 expanders to extend the number of devices in a RJ-460/RJ-472 SAS domain. Bridge expander is the root and two edge expanders cascaded. Figure 3 and 4 depicts the topology.

The Bridge expander terminates the external host and expansion ports and links downstream to Edge expanders over MCIO 8i/4i.

- RJ-460 SAS4: Bridge → Edge1 (bays 1–24) and Edge2 (bays 25–48); remaining bays are attached at bridge level (49–60), preserving the legacy slot mapping.
- RJ-472 SAS4: Bridge → Edge1 (bays 1–24), Edge2 (25–48), Edge3 (49–72), matching the established layout of the SAS3 predecessor.

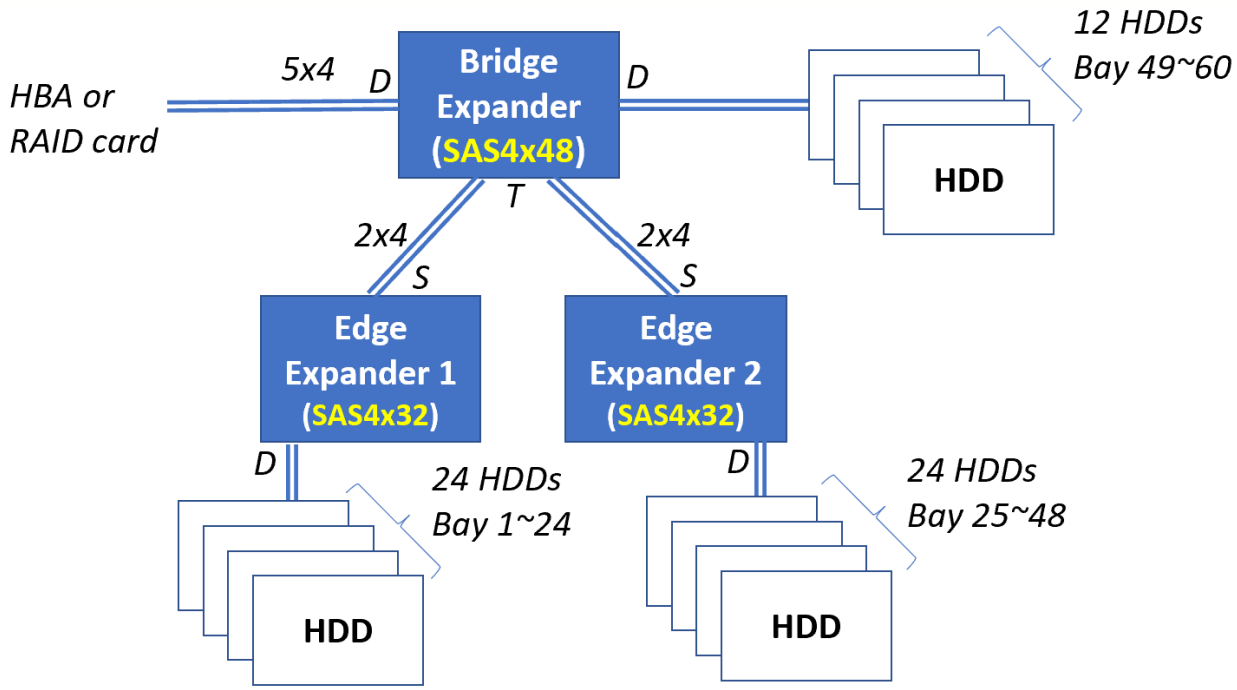

FIGURE 3 RJ-460 SAS4 TOPOLOGY

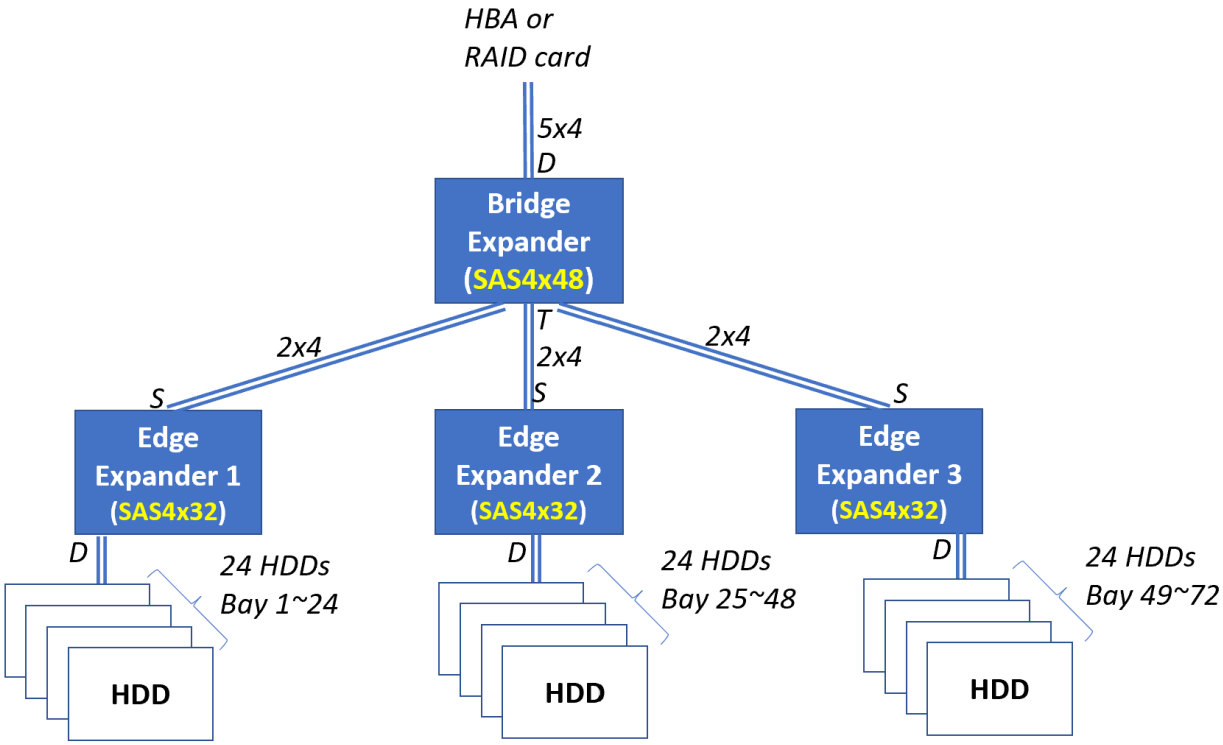

FIGURE 4 RJ-472 SAS4 TOPOLOGY

3 System Interface

3.1 Overview

RJ-460 (SAS4) and RJ-472 (SAS4) expose the same controls and indicators as the prior SAS3 models, with updated 24G SAS cabling: external ports use SFF-8674 and internal expander links use MCIO 8i/4i. Modules are arranged as: front panel (buttons/LEDs), rear (fan FRUs, bridge expander module, CRPS PSUs), and internal (edge expanders and backplanes).

3.2 Front Panel

Figure 6 shows the buttons, LEDs detail on front panel.

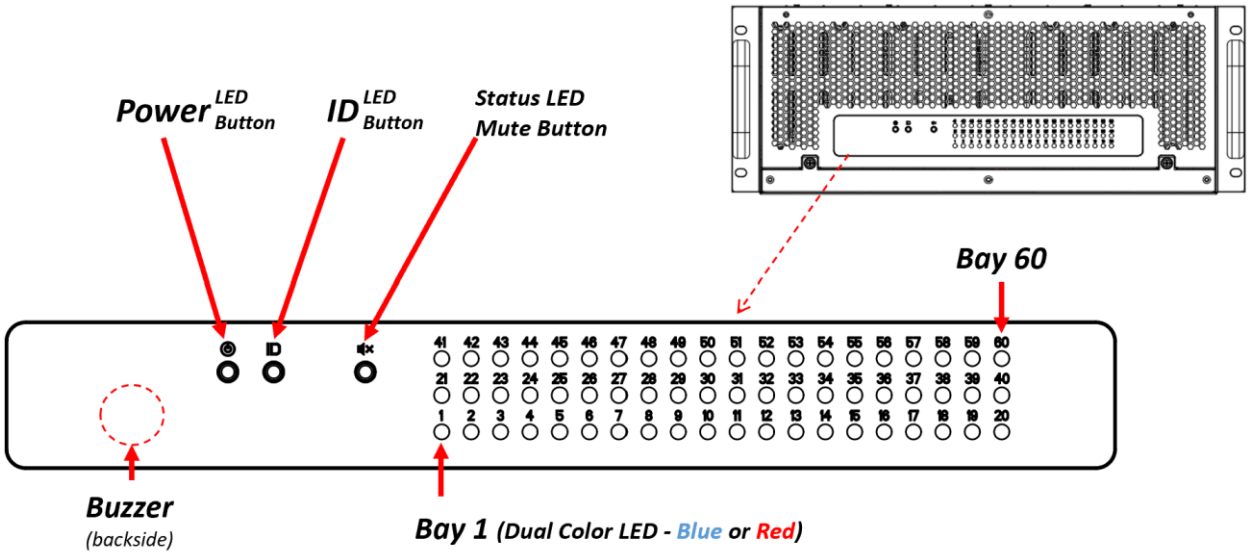

FIGURE 5 RJ-460 SAS4 FRONT PANEL DETAIL

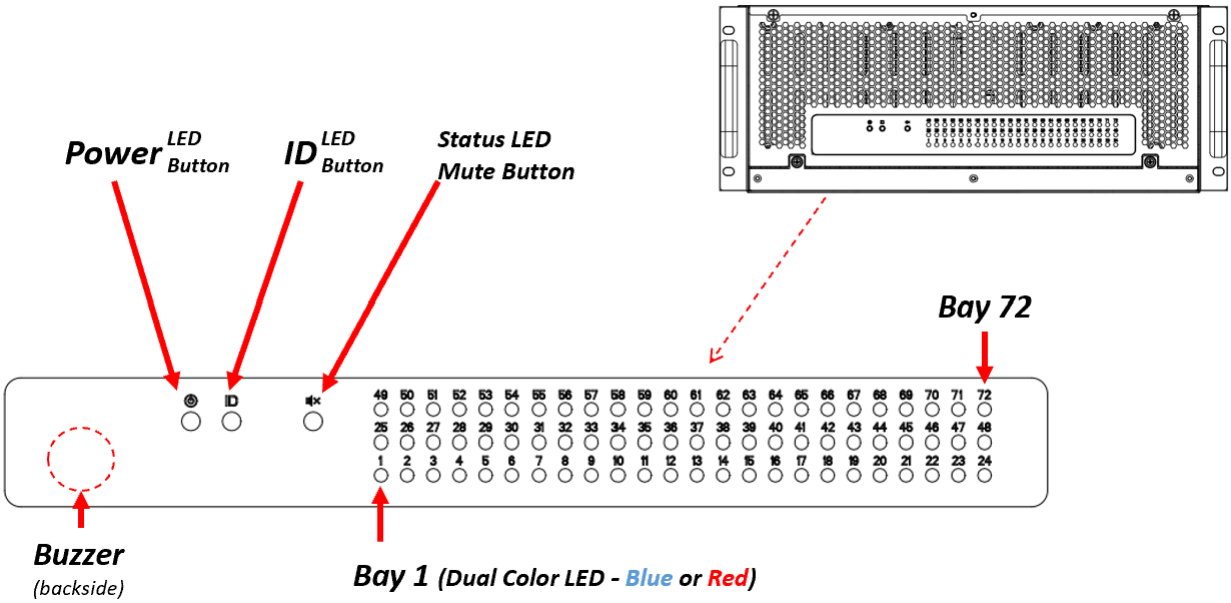

FIGURE 6 RJ-472 SAS4 FRONT PANEL DETAIL

3.2.1 LED

The system is equipped with LEDs to provide status information for indicating component current state.

Power LED (blue color)

- Power is applied to the system(PSU). LED is illuminated when system is in operating.

Bay LED (dual color)

- Indicates hard disk current state. There are dual color LED that blue is used for operational and red is used for warning, error or failure.
  - Blue LED steady ON – hard disk is installed and powered
  - Blue LED in 4 Hz blinking – hard disk is accessed or in activity
  - Red LED steady ON – hard disk is failed
  - Red LED in 0.5 Hz blinking – hard disk is identified (LOCATE)
  - Red LED in 1 Hz blinking – hard disk is in RAID rebuild state

Status LED (red color)

- System current status
  - Steady ON – System overheats
  - 1 Hz blinking – System fans in error condition

ID LED (blue color)

- Push button and turn on to identify the enclosure

3.2.2 Button Control

There are three push-buttons on front panel and two push-buttons on rear side (on bridge expander module).

Power button

- The button is for applying or removing power from the power supply to the system
  - Turn on – Press and hold for more than 3 seconds to power on system
  - Turn off – Press and hold for more than 3 seconds to shutdown system

ID button

- Press the button to toggle the ID LED ON and OFF

Mute button

- Press the button to toggle the buzzer beeping or silence

3.2.3 Advance Button Control

Feature	***Description***
Auto Power-ON	Press **ID + Power** buttons simultaneously to toggle this feature. Power LED blinks when this feature is engaged.
Cooling Mode	Press and hole **ID** button for more than **5** seconds to change mode in turns. There are three fan cooling modes to setup - **quiet, normal, aggressive**. Default is normal mode. ID LED blinks when cooling mode is engaged.
Mute Mode	Press and hold **Mute** button for more than ***5*** seconds to toggle mute feature. Status LED blinks when feature engaged.

3.2.4 Buzzer

The buzzer is to notify system warning states

System Status	***Description***
Fan Error	0.25s beep and 1.75s silence repeatedly. (slow beeps)

Thermal Error	0.25s beep, 0.25 silence, 0.25 beep and 1.25s silence repeatedly. (fast beeps)
Fan & Thermal Error	Alter 'Fan Error' and 'Thermal Error' repeatedly
PSU Error	continuous beep

3.3 Rear Panel

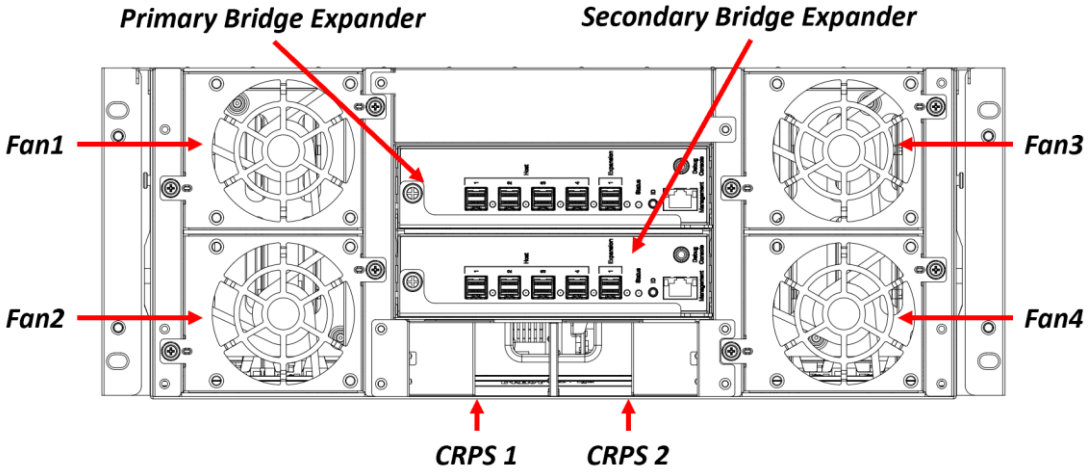

FIGURE 7 BACKSIDE DETAIL

Fan modules, bridge expanders and PSU modules can be plug-in/out on the backside (rear-side). Figure 7 depicts the location of the components.

3.4 Bridge Expander Module

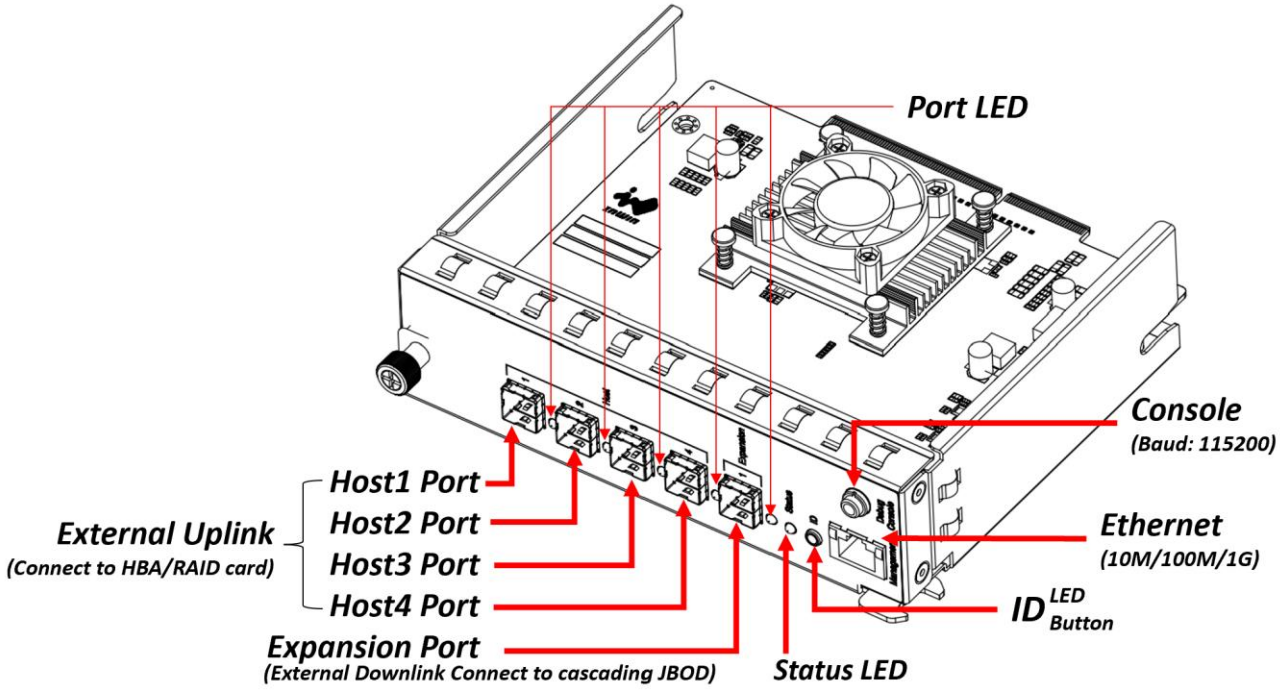

FIGURE 8 BRIDGE EXPANDER MODULE

Bridge expander module (Figure 8) is the main component that controls and manages the RJ-460/RJ-472 SAS4 JBOD system. Users can use ethernet or RS-232 console cable to connect to the bridge module using telnet or some console applications to command it by CLI.

Host port (SFF-8674)

- External uplink port
- RAID or HBA card connect to the bridge module
- There are 4 ports (Host1, Host2, Host3 and Host4) for host connection

Expansion port (SFF-8674)

- External downlink port to cascade another expander

Status LED

- Works like front panel status LED
- Indicates the system current state
    - Steady ON – the system overheats
    - 1 Hz blinking – the system fans in error condition

Port LED

- There are Host1, Host2, Host3, Host4 and Expansion port LEDs
- Illuminated LED Indicates host port connection (from RAID or HBA) or expansion expander is physically connected

ID LED

- Works synchronously with front panel ID LED
- Illuminate to identify the chassis

ID button

- Works synchronously with front panel ID button
- Press to toggle the ID LED ON/OFF

UART Console

- The RS-232 port on the Bridge Expander module provides a serial connection for users to manage the JBOD system with terminal application such as Tera Term, Hyper Term…etc. To manage the JBOD system through the serial port, configure the terminal console as per the settings below to connect it.

Parameter	***Value***
Baud Rate	115200
Data	8 bit
Parity	None
Stop Bit	1 bit
Flow Control	None

Ethernet

- Support 10M/100M/1G bps ethernet PHY
- Support TCP/IP protocol
- Support Telnet CLI
- Support Webpages (http). Provide an account password security setting interface. IP address could be check from CLI then view the webpages for system information on a web browser. Figure 9~13 are the examples of the web pages.

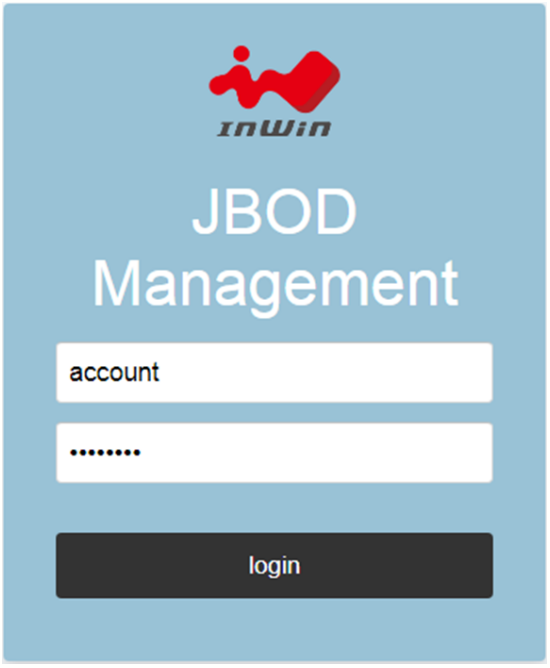

FIGURE 9 VERSION PAGE

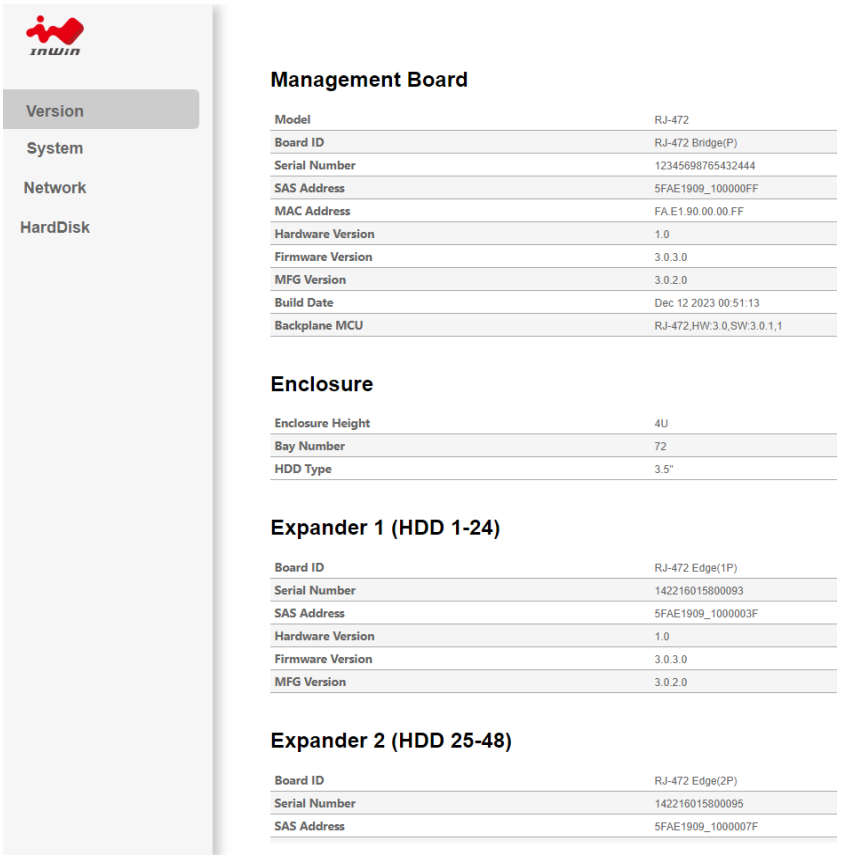

FIGURE 10 VERSION PAGE

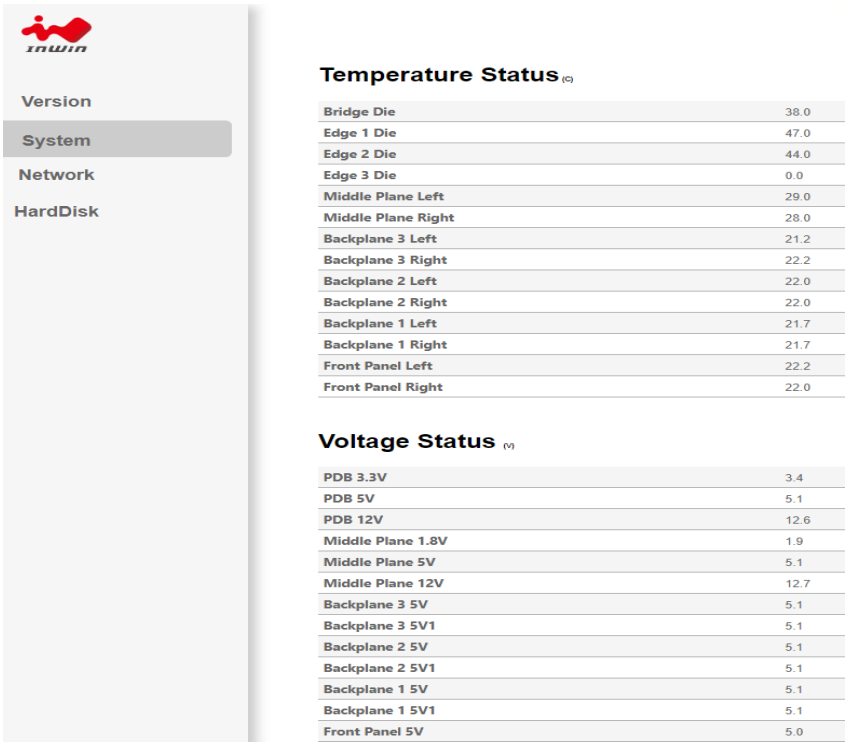

FIGURE 11 SYSTEM PAGE

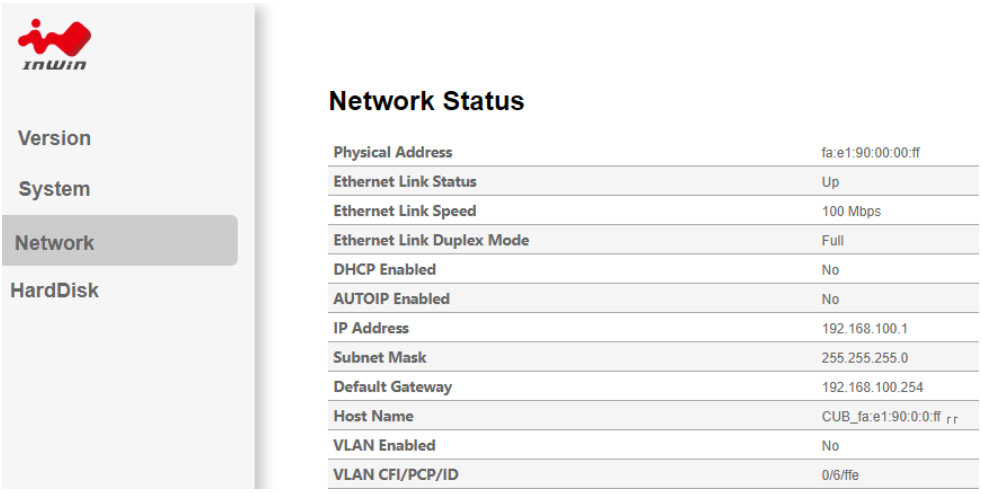

FIGURE 12 NETWORK PAGE

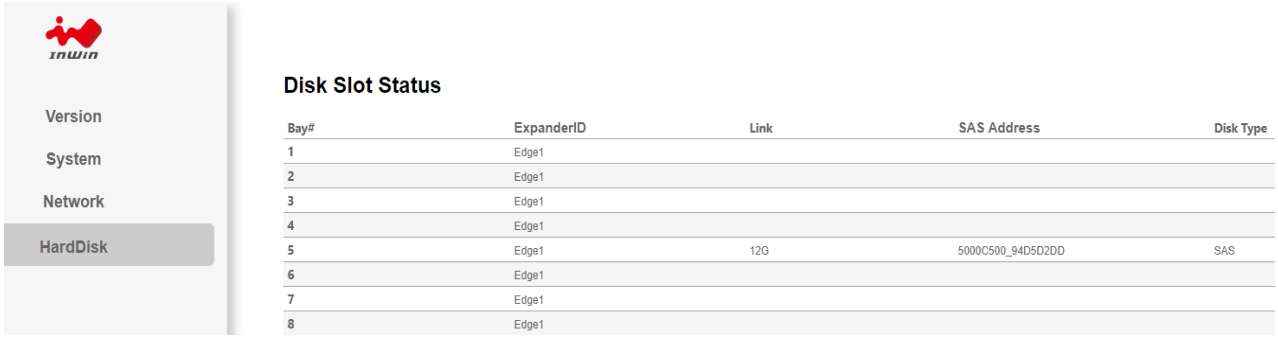

FIGURE 13 HARD DISK PAGE

3.5 Internal modules

- Edge expander board (SAS4)

  Edge expanders are in 24-bay board and connect to the bridge over MCIO 8i/4i internal cables in the SAS4 design (SAS3 manuals show SFF-8643; behavior is unchanged). They are managed through the bridge CLI (Ethernet or RS-232).

- Slot mapping
  - RJ-460: bridge + two edge expanders (tree topology). Bays 1–24 on Edge1, 25–48 on Edge2, 49–60 on Bridge.
  - RJ-472: bridge + three edge expanders. Bays 1–24 on Edge1, 25–48 on Edge2, 49–72 on Edge3
- Internal cables: use MCIO 8i/4i harnesses between bridge and edge expanders.

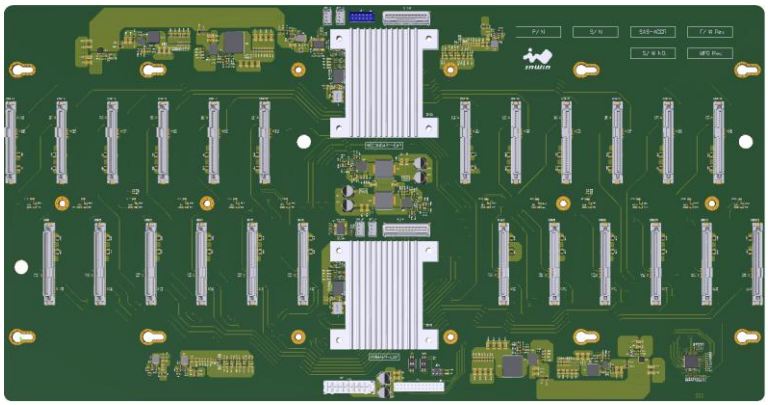

FIGURE 14 EDGE EXPANDER BOARD

3.6 Fan modules & CRPS PSUs

Hot-swappable FRUs located on the rear; faults illuminate the front Status LED and trigger the buzzer (can be muted from the front panel)

4 Command Line Interface (CLI)

The RJ-460/RJ-472 SAS4 JBOD system provide flexible and useful commands to monitor and manage hard disk, expanders and backplane system. Users can use telnet or console applications to manage the system with CLI commands.

4.1 help

List all command list and the usage

Syntax

```
help [Command]
 - X means a mandatory argument described as x
 - [X] means an optional argument
 - <X> means replace argument with value
 - X(D) means a decimal value
 - X(H) means a hexadecimal value
 - X(*) means special formatting is specified
 - X(s) means a decimal value in seconds
 - X(ms) means a decimal value in milliseconds
 - opt1|opt2|opt3 means literally specify
   "opt1","opt2", or "opt3"
 - X1 X2 ... means a variable number
   of arguments
```

Example

```
Bridge >help

================================================================================

    Description             Command

================================================================================

    Show Sensors            sensor

    Remote Command 3        edge3

    Remote Command 2        edge2

    Remote Command 1        edge1

    Remote Command 0        bridge
                        .
                        .
                        .

 Bridge >help fdl

================================================================================

    Description             Command
```

```
================================================================================

    File download           fdl <BufferID(H)> <Offset(H)> <Erase(*)>
                             - Erase is replaced with "Y" or "N"
```

4.2 rev

Show vendor ID, product ID, active firmware and firmware version information

Syntax

```
rev
```

Example

```
Bridge >rev

================================================================================
Hardware Revision Information:-
================================================================================

Vendor ID                       :IN-WIN
Product ID                      :RJ-460 Bridge(P)
Product Revision Level          :Minor = 0x00, Unit = 0x00
Component Type                  :Sas4x
Component ID                    :0x0240 (Bond Option :48)
Component Revision Level        :1      (B0)
Security Mode                   :Hard Secure

================================================================================
Firmware Revision Information:-
================================================================================

Active Firmware: Firmware Copy 1

Boot Image:
  Location     :   SPI NOR CS
  Revision     :   1.0.1.0
  Platform Name:   IN-WIN SAS4 Expander System
  Version Name :   SAS4xFW-01.00.01.00 12/27/24
  Firmware Family: 64  OemFamily: 0
Firmware Copy 1:
  Location     :   SPI NOR CS
  Revision     :   4.0.0.0
  Platform Name:   JBOD-H SAS4 Expander System
  Version Name :   SAS4xFW-04.00.00.00 08/26/25
  Firmware Family: 64  OemFamily: 0
Firmware Copy 2:
  Location     :   SPI NOR CS
  Revision     :   4.0.1.0
  Platform Name:   JBOD-H SAS4 Expander System
  Version Name :   SAS4xFW-04.00.01.00 08/26/25
  Firmware Family: 64  OemFamily: 0
================================================================================
SDK Revision:-
================================================================================

  HAL Revision: 0.9.2.0
  SPI Revision: 0x14001800
  SES Revision: 0.9.2.0
  SCE Revision: 0.9.2.0

  SerDes M0 FW: 0x41b0
```

```
================================================================================
Current Toolchain Revision:-

================================================================================
ARM v.5060750
```

4.3 showmfg

Show manufacture configuration version information

Syntax

```
showmfg
```

Example

```
Bridge >showmfg
================================================================================
Manufacturing Image Version Information:-
================================================================================
Product ID :                    17
Platform ID:                    7
Associated Firmware Revision:   4:0:0:0
Mfg Revision:                   4:0:0:0
Product Name:                   RJ-460 Bridge(P)
Platform Name:                  SAS4 BOARD
Mfg Image Creation time stamp:  6/10/25 18:22 (mm/dd/yy hr:min)
```

4.4 sasaddr

display expander's SAS address

Syntax

```
sasaddr
```

Example

```
Bridge >sasaddr

 Expander SAS Addresses -

  SxP Port 0 SAS Address: 0x5FAE1970000005FD
  SxP Port 1 SAS Address: 0x5FAE1970000005FF
  SxP Port 2 SAS Address: 0x0000000000000000
  SxP Port 3 SAS Address: 0x0000000000000000
```

4.5 date

Show or set current date time. There are no RTC in system. System always starts up at incorrect date. System log will record event with datetime. User can set datetime and broadcast the new value to other expanders by 'date send'

Syntax

```
date [ set <newdate(*)> | send [<SasAddrHi(H)> <SasAddrLow(H)>] |
                          recv [<SasAddrHi(H)> <SasAddrLow(H)>] ]
 - no arguments, output local expander date/time and uptime
```

```
 - 'set <newdate(*)>' Set local expander date/time using format for <newdate> of
   YYYYMMDDHHMMSS
 - 'send [sas_addr]' If sas_addr set then set the remote expander's clock using
the
   local expander's current clock value.  If sas_addr not specified then set
clock
   for all expanders in the domain.
 - 'recv [sas_addr]' If sas_addr set then fetch the remote expander's current
clock
   and output the result.  If sas_addr not specified then get and output the
current
   clock for all expanders in the domain.
```

Example

```
Bridge >date

Tue Jan  9 13:41:08 2024

Current expander uptime is 0 days, 3 hours, 41 minutes, 8 seconds



 Bridge >date set 20240111125000

Expander date/time set to Thu Jan 11 12:50:00 2024




 Bridge >date send
Result sending SET TIMESTAMP to 5fae19091000003d:  SUCCESS

Result sending SET TIMESTAMP to 5fae19091000007d:  SUCCESS

Result sending SET TIMESTAMP to 5fae1909100000bd:  SUCCESS


 Bridge >edge1 date
>>>edge1:


Thu Jan 11 12:51:17 2024

Current expander uptime is 0 days, 3 hours, 44 minutes, 7 seconds
```

4.6 showlogs

Show the current system logs

Syntax

```
showlogs [hex|detail|default]
```

Example

```
 Bridge >showlogs
<Feb 04 2023 10:25:50.548>:SCSI:A SCSI Command Descriptor Block

<Feb 04 2023 10:25:50.672>:SCSI:A SCSI Command Descriptor Block
```

```
<Feb 04 2023 10:31:16.154>:SCSI:A SCSI Command Descriptor Block

<Feb 04 2023 10:31:16.155>:SCSI:A SCSI Command Descriptor Block

<Feb 04 2023 10:31:16.156>:SCSI:A SCSI Command Descriptor Block
…
```

4.7 clearlogs

Clear the system logs

Syntax

```
clearlogs
```

Example

```
 Bridge >clearlogs


 Bridge >showlogs


 Bridge >
```

4.8 showpost

Show POST (Power-On-Self-Test) status of expander

Syntax

```
showpost
```

Example

```
Bridge >showpost

POST status of expander on-chip devices


================================================================================

On-Chip Device            POST Status

================================================================================

Smart Serial             :PASS
Watchdog Timer           :PASS
Timer 1                  :PASS
Timer 2                  :PASS
Timer 3                  :PASS
Timer 4                  :PASS
PLLs                     :PASS
Interrupt                :PASS
Phy Loopback             :PASS
ISTWI Channel 00         :PASS
ISTWI Channel 01         :PASS
ISTWI Channel 02         :PASS
ISTWI Channel 06         :PASS
ISTWI Channel 07         :PASS
True Random Num Generator:PASS
```

```
Internal RAM Test         :PASS
Forced ECC Error Test     :PASS
SXP Port Buffers Test     :PASS
```

4.9 sesinfo

Display SES-2 devices information

Syntax

```
sesinfo
```

Example

```
Bridge >sesinfo
Show SES Enclosure Elements

Current Fan Level(1~7): 1 1 1 1
Fan01           FanSpd:  2700   SpdCode:     1  Fail:     0
Fan02           FanSpd:  2700   SpdCode:     1  Fail:     0
Fan03           FanSpd:  2700   SpdCode:     1  Fail:     0
Fan04           FanSpd:  2700   SpdCode:     1  Fail:     0
BridgeDie.C   Fail: 0 Tempt: 43     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
MP Left.C     Fail: 0 Tempt: 29     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
MP Right.C    Fail: 0 Tempt: 29     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP3 Left.C    Fail: 0 Tempt: 27     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP3 Right.C   Fail: 0 Tempt: 27     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP2 Left.C    Fail: 0 Tempt: 28     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP2 Right.C   Fail: 0 Tempt: 28     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP1 Left.C    Fail: 0 Tempt: 28     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
BP1 Right.C   Fail: 0 Tempt: 28     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
FP Left.C     Fail: 0 Tempt: 29     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
FP Right.C    Fail: 0 Tempt: 28     UTWarn: 0     UTFail: 0    OTWarn: 0
OTFail: 0
PdbVoltage3.3V          330
PdbVoltage 5V           500
PdbVoltage 12V          1200
MPVoltage 5V            510
MPVoltage 12V           1190
BP3 Voltage 5V          510
BP2 Voltage 5V          510
BP1 Voltage 5V          510
PowerSupplyElement00    Install: 1, Fail: 0
PowerSupplyElement01    Install: 1, Fail: 0
```

4.10 sensor

Display backplane system sensor information

Syntax

```
sensor
```

Example

```
PRIMARY> sensor

BP Description: JS4MCU,HW:4.0,SW:4.0.0,1
MCU UID: 56B5
Status: 5(SMBus)        00(BP)  00(FAN)
Expander Installed: 03(MP)      00(BP1) 00(BP2) 00(BP3)
CRPS: 00(Status)        03(Installed)
Temperature: (C)
        ExpDie: 41, 44, 41, 0
        MP:  28.2, 28.5
        BP3: 27.7, 27.7
        BP2: 27.7, 28.2
        BP1: 27.7, 28.0
        FP:  29.0, 28.0
Voltage: (V)
        PDB:  3.3,  5.0, 12.0
        MP:   5.1,  5.1, 11.9
        BP3:  5.1,  4.9
        BP2:  5.1,  5.1
        BP1:  5.1,  5.1
        FP:   5.1
Fan RPM:
 3780, 3780, 4200, 4200,    0,    0,
Fan PWM:
    0,    0,   54,   43,    1,    1,
```

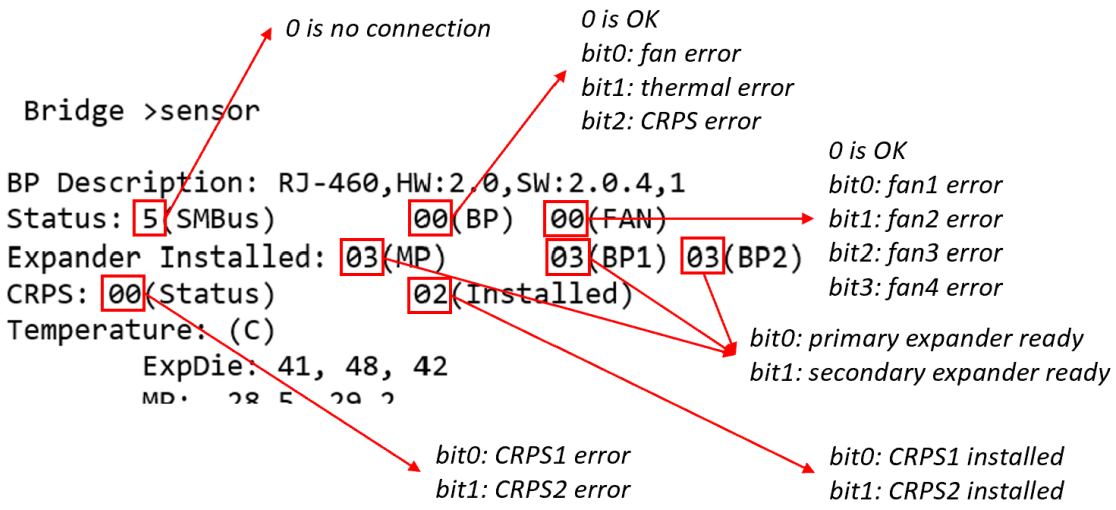

FIGURE 15 SENSOR INFORMATION DETAIL

4.11 disk

Display 60 or 72 bays disk PHY information

Syntax

```
disk
```

Example

```
 Bridge >disk
                PHY                                 ZONE      CONN CONN MAP
HDD  DEV        CNG                      ROUTE ZONE CTRL CONN ELEM PHY  PHY
No   TYPE NLR   CNT  ATTACHED SAS ADDR TYPE  GRP  BUS  TYPE INDX LINK ID

  1  END 22.5G 0x03 5000C500_94D5CA01    D  0x01 0x04 0x20 0x01 0x00 000
  2  END 22.5G 0x03 5000C500_94D5D01D    D  0x01 0x04 0x20 0x02 0x00 001
  3  END 22.5G 0x03 5000C500_94D5D7AD    D  0x01 0x04 0x20 0x03 0x00 002
  4  END 22.5G 0x03 5000C500_94D5D449    D  0x01 0x04 0x20 0x04 0x00 003
  5  END 22.5G 0x03 5000C500_94D5C8BD    D  0x01 0x04 0x20 0x05 0x00 004
  6  END 22.5G 0x03 5000C500_94D5C7E5    D  0x01 0x04 0x20 0x06 0x00 005
  7  END 22.5G 0x03 5000C500_94D5C6F1    D  0x01 0x04 0x20 0x07 0x00 006
  8  END 22.5G 0x03 5000C500_94D5D54D    D  0x01 0x04 0x20 0x08 0x00 007
  9  END 22.5G 0x03 5000C500_94D5C80D    D  0x01 0x04 0x20 0x09 0x00 008
 10  END 22.5G 0x03 5000C500_94D5D2CD    D  0x01 0x04 0x20 0x0A 0x00 009
 11  END 22.5G 0x03 5000C500_94D5D60D    D  0x01 0x04 0x20 0x0B 0x00 010
 12  END 22.5G 0x03 5000C500_94D5D575    D  0x01 0x04 0x20 0x0C 0x00 011
 13  END 22.5G 0x03 5000C500_94D5D035    D  0x01 0x04 0x20 0x0D 0x00 012
 14  END 22.5G 0x03 5000C500_94D5C915    D  0x01 0x04 0x20 0x0E 0x00 013
 15  END 22.5G 0x03 5000C500_94D5D4E1    D  0x01 0x04 0x20 0x0F 0x00 014
 16  END 22.5G 0x03 5000C500_94D5D195    D  0x01 0x04 0x20 0x10 0x00 015
 17  END 22.5G 0x03 5000C500_94D5D215    D  0x01 0x04 0x20 0x11 0x00 016
 18  END 22.5G 0x03 5000C500_94D5CAA1    D  0x01 0x04 0x20 0x12 0x00 017
 19  END 22.5G 0x03 5000C500_94D5D72D    D  0x01 0x04 0x20 0x13 0x00 018
 20  END 22.5G 0x03 5000C500_94D5D365    D  0x01 0x04 0x20 0x14 0x00 019
 21  END   12G 0x03 5000C500_94D5CA8D    D  0x01 0x04 0x20 0x15 0x00 020
 22  END   12G 0x03 5000C500_94D5CF25    D  0x01 0x04 0x20 0x16 0x00 021
 23  END   12G 0x03 5000C500_94D5D21D    D  0x01 0x04 0x20 0x17 0x00 022
 24  END   12G 0x03 5000C500_94D5C941    D  0x01 0x04 0x20 0x18 0x00 023
 25  END   12G 0x03 5000C500_94D5D6F5    D  0x01 0x04 0x20 0x19 0x00 000
 26  END   12G 0x03 5000C500_94D5D7A5    D  0x01 0x04 0x20 0x1A 0x00 001
 27  END   12G 0x03 5000C500_94D5CD49    D  0x01 0x04 0x20 0x1B 0x00 002
 28  END   12G 0x03 5000C500_94D5CA11    D  0x01 0x04 0x20 0x1C 0x00 003
 29  END   12G 0x03 5000C500_94D5D1D1    D  0x01 0x04 0x20 0x1D 0x00 004
 30  END   12G 0x03 5000C500_94D5D111    D  0x01 0x04 0x20 0x1E 0x00 005
 31  END   12G 0x03 5000C500_94D5CF75    D  0x01 0x04 0x20 0x1F 0x00 006
 32  END   12G 0x03 5000C500_94D5CBE9    D  0x01 0x04 0x20 0x20 0x00 007
 33  END   12G 0x03 5000C500_94D5D2BD    D  0x01 0x04 0x20 0x21 0x00 008
 34  END   12G 0x03 5000C500_94D5D775    D  0x01 0x04 0x20 0x22 0x00 009
 35  END   12G 0x03 5000C500_94D5CA45    D  0x01 0x04 0x20 0x23 0x00 010
 36  END   12G 0x03 5000C500_94D5CB15    D  0x01 0x04 0x20 0x24 0x00 011
 37  END   12G 0x03 5000C500_94D5C76D    D  0x01 0x04 0x20 0x25 0x00 012
 38  END   12G 0x03 5000C500_94D5D09D    D  0x01 0x04 0x20 0x26 0x00 013
 39  END   12G 0x03 5000C500_94D5D379    D  0x01 0x04 0x20 0x27 0x00 014
 40  END   12G 0x03 5000C500_94D5CF71    D  0x01 0x04 0x20 0x28 0x00 015
 41  END   12G 0x03 5000C500_94D5D569    D  0x01 0x04 0x20 0x29 0x00 016
 42  END   12G 0x03 5000C500_94D5CF79    D  0x01 0x04 0x20 0x2A 0x00 017
 43  END   12G 0x03 5000C500_94D5CB01    D  0x01 0x04 0x20 0x2B 0x00 018
 44  END   12G 0x03 5000C500_94D5D095    D  0x01 0x04 0x20 0x2C 0x00 019
 45  END   12G 0x03 5000C500_94D5D4BD    D  0x01 0x04 0x20 0x2D 0x00 020
 46  END   12G 0x03 5000C500_94D5D011    D  0x01 0x04 0x20 0x2E 0x00 021
 47  END   12G 0x03 5000C500_94D5CF61    D  0x01 0x04 0x20 0x2F 0x00 022
 48  END   12G 0x03 5000C500_94D5D1A1    D  0x01 0x04 0x20 0x30 0x00 023
 49  END   12G 0x03 5000C500_94D5CBB9    D  0x01 0x04 0x20 0x31 0x00 000
 50  END   12G 0x03 5000C500_94D5D591    D  0x01 0x04 0x20 0x32 0x00 001
 51  END   12G 0x03 5000C500_94D5CE01    D  0x01 0x04 0x20 0x33 0x00 002
 52  END   12G 0x03 5000C500_94D5CC81    D  0x01 0x04 0x20 0x34 0x00 003
 53  END   12G 0x03 5000C500_94D5C9D1    D  0x01 0x04 0x20 0x35 0x00 004
 54  END   12G 0x03 5000C500_94D5CC99    D  0x01 0x04 0x20 0x36 0x00 005
 55  END   12G 0x03 5000C500_94D5D48D    D  0x01 0x04 0x20 0x37 0x00 006
 56  END   12G 0x03 5000C500_94D5CBE1    D  0x01 0x04 0x20 0x38 0x00 007
 57  END   12G 0x03 5000C500_94D5CBD9    D  0x01 0x04 0x20 0x39 0x00 008
 58  END   12G 0x03 5000C500_94D5CCF1    D  0x01 0x04 0x20 0x3A 0x00 009
 59  END   12G 0x03 5000C500_94D5CFC1    D  0x01 0x04 0x20 0x3B 0x00 010
 60  END   12G 0x03 5000C500_94D5D725    D  0x01 0x04 0x20 0x3C 0x00 011
 61  END   12G 0x03 5000C500_94D5D3B5    D  0x01 0x04 0x20 0x3D 0x00 012
```

```
 62  END   12G 0x03 5000C500_94D5CBF9    D  0x01 0x04 0x20 0x3E 0x00 013
 63  END   12G 0x03 5000C500_94D5D7FD    D  0x01 0x04 0x20 0x3F 0x00 014
 64  END   12G 0x03 5000C500_94D5D4C5    D  0x01 0x04 0x20 0x40 0x00 015
```

4.12 console

Switch RS-232 UART control to bridge, edge1, edge2 or edge3 expander.

Syntax

```
console <0|1|2|3>
```

Example

```
 Bridge >console 1
Switch Smart Console to Edge1

 Edge1 >
 Edge1 >console 0
Switch Smart Console to Bridge


 Bridge >
 Bridge > console 2
Switch Smart Console to Edge2


 Edge2 >
 Edge2 >console 0
Switch Smart Console to Bridge



 Bridge >
```

4.13 bpsys

Display backplane system control parameters that are saved in persistent storage (flash)

Syntax

```
bpsys
```

Example

```
 Bridge >bpsys
BP System NVRAM Config:
        Fan Speed Profile: Normal
        Auto Power UP: 0
        Mute Enable: 0
        Fan Control Temperature Base: 45
        Zone Config: 0 (Disable)

BP System Running Status:
        Test Enable: 0
        ID LED Enable: 0
```

bpsys power <0|1>

set backplane system auto power-on and startup after power suppliers ready. Default value 0 requires users to press power button for more than 3 seconds to power up system.

Syntax

```
bpsys power <0|1>
```

Example

```
 Bridge >bpsys power 1
Set Auto Power UP: 1
```

bpsys default 0

reset backplane system configuration to default

Syntax

```
bpsys default 0
```

Example

```
 Bridge >bpsys default 0
Reset bpsys to default configuration!
Reboot system to use the new setting


 Bridge >reset

Performing POST for Smart Serial
…
```

bpsys fan <0|1|2>

Set backplane system fan speed profile mode. There are 3 modes for fan speed profile. Default value is 1 (Normal)

Mode ID	***Name***	***Description***
0	Quiet	8 fan RPM levels – 34%, 44%, 54%, 63%, 71%, 79%, 87%, 95%
1	Normal	8 fan RPM levels – 41%, 49%, 57%, 65%, 73%, 81%, 89%, 97%
2	Aggressive	8 fan RPM levels – 45%, 53%, 61%, 69%, 77%, 85%, 93%, 100%

Syntax

```
bpsys fan <0|1|2>
```

Example

```
 Bridge >bpsys fan 0
Set Fan Speed Profile: 0
```

bpsys thermal <37~52>

Sets the temperature (°C) at which the fan controller drives the fans to the highest level. The controller samples all enclosure thermal sensors and uses the max reading (expander die temp is excluded) to map into the 8 fan levels. Default = 45°C.

Syntax

```
bpsys thermal <37~52>
```

Example

```
 Bridge >bpsys thermal 45
Set Fan Control Max Temperature (37C~52C): 45
```

How it works

- The value defines the top of the fan curve. If the max sensor reaches this value, fans go to the highest SES level for the selected profile. Lower values make fans ramp up earlier; higher values delay ramp.
- Example Profile = Normal (levels: 41, 49, 57, 65, 73, 81, 89, 97%): with max sensor 40.1°C
  - bpsys thermal 42 ⇒ highest level (~97%)
  - bpsys thermal 45 ⇒ 2nd-highest level (~89%). (Normal profile levels: 41, 49, 57, 65, 73, 81, 89, 97%.)

When to adjust

- Partially populated chassis / higher ambient: choose 42–44°C to ramp earlier and keep drives/canisters cooler. (Earlier ramp = more airflow/noise)
- Fully populated with strong airflow or noise-sensitive: use 45–47°C to reduce fan speed when temps are moderate. (Later ramp = lower noise, watch temperatures.)

Persistence & visibility

The setting is saved in NVRAM and shown by bpsys as *Fan Control Temperature Base*.

Related commands

- **bpsys fan <0|1|2>** selects the profile (Quiet/Normal/Aggressive) which defines the 8 RPM levels; bpsys thermal only shifts where those levels engage.
- sensor shows current temperatures; use it to verify the max reading you're targeting. (Section reference in the manual.)
- **bpsys** prints the current thermal base along with other saved settings.

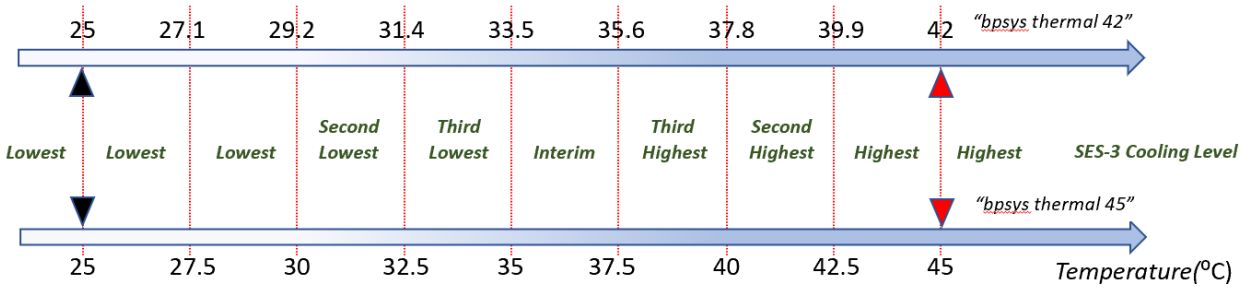

FIGURE 16 BPSYS THERMAL 42 VS 45 COMPARISON

bpsys zone <0|1|2>

Set PHY-base zone mode. There are 2 zone mode, 0 – disable, 1 – enable. RJ-460 also have mode 2. Default (0) is zone disabled. Refer to Figure 17 for RJ-472 enable mode (3 zones – zone1 is bay1-24, zone2 is bay25-48, zone3 is bay49-72). Figure 18 for RJ-460 enable mode 1 and Figure 19 for RJ-460 enable mode 2

Syntax

```
bpsys zone <0|1|2>
```

Example

```
 Bridge >bpsys zone 1
Set Zone Profile: 1
```

3 Zones Configuration – 24/24/24

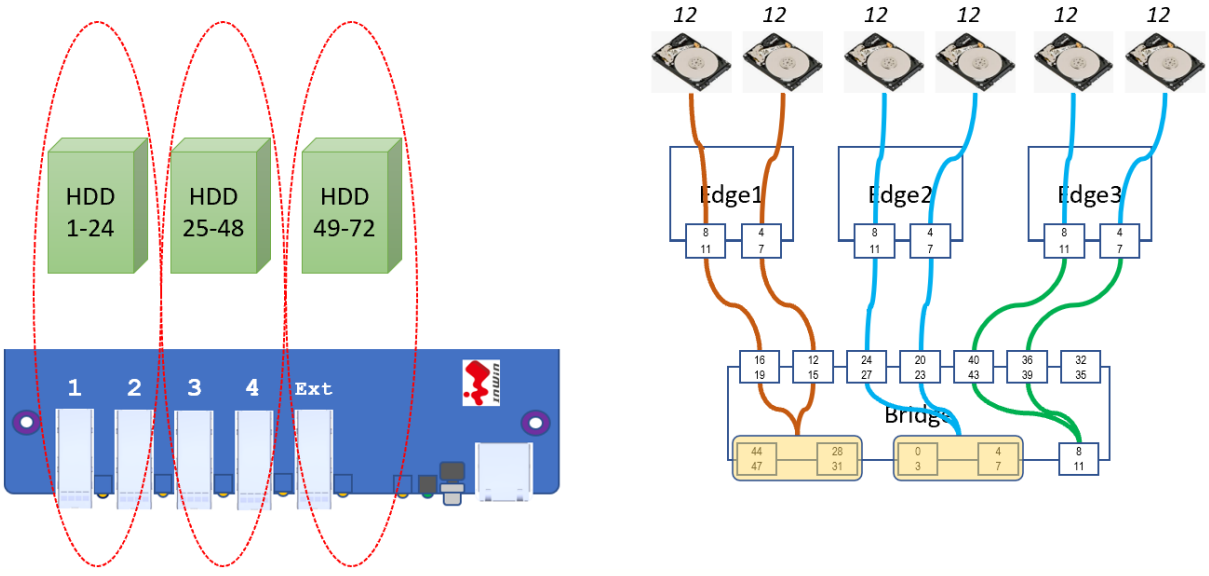

FIGURE 17 RJ-472 ZONES ENABLE CONFIGURATION

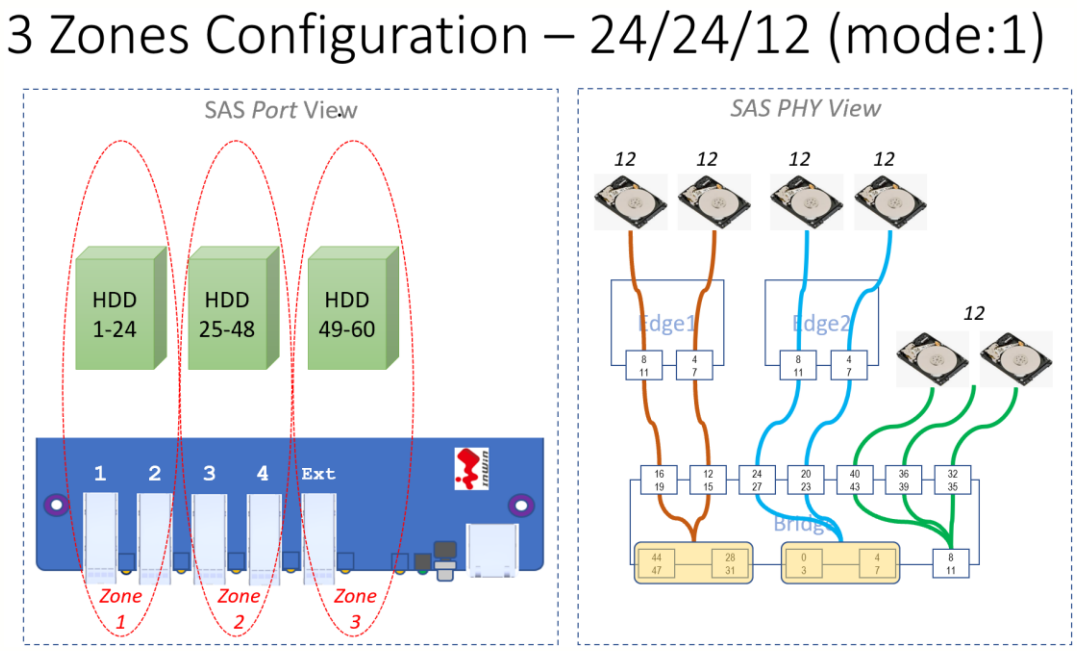

FIGURE 18 RJ-460 ZONES ENABLE MODE 1 CONFIGURATION

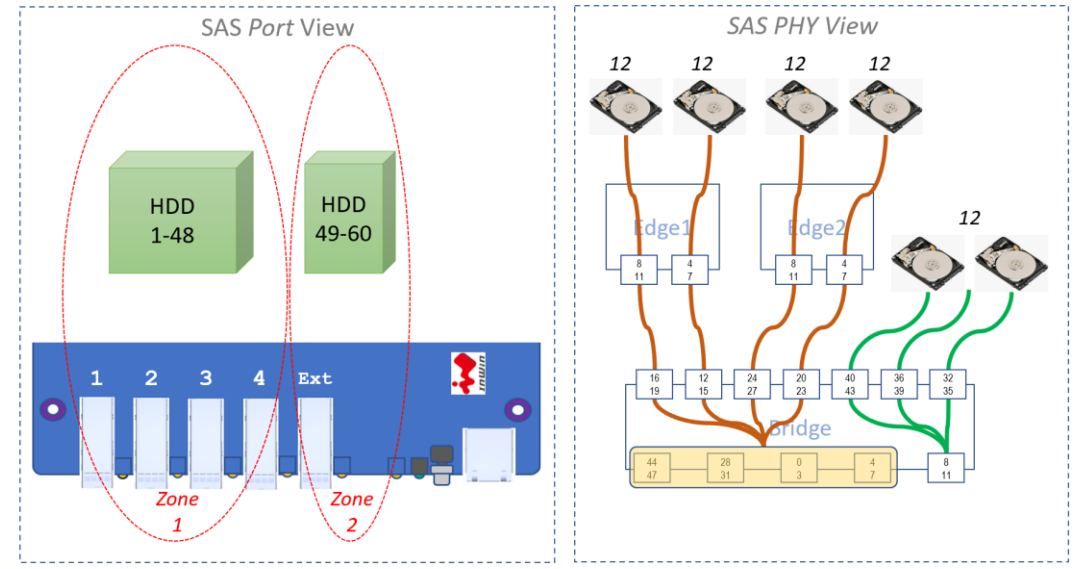

FIGURE 19 RJ-460 ZONES ENABLE MODE 2 CONFIGURATION

bpsys mute <0|1>

Set backplane system buzzer to mute mode. Default (0) is to enable beeping when system occurs some errors.

Syntax

```
bpsys mute <0|1>
```

Example

```
 Bridge >bpsys mute 1
Set Mute Enable: 1
```

bpsys id <0|1>

Turn backplane system ID LED on or off. The value is not saved to storage. Default (0) is to turn off ID LED. This command enable user to turn on ID LED from remote.

Syntax

```
bpsys id <0|1>
```

Example

```
 Bridge >bpsys id 1
ID LED Enable: 1
```

bpsys test <0|1>

Set backplane system to test mode that blinks all LEDs and play buzzer for test. The value is not saved to storage. Default (0) is to quit the test mode.

Syntax

```
bpsys test <0|1>
```

Example

```
 Bridge >bpsys test 1
Test Enable: 1
```

bpsys reset <1>

Make backplane system to reset all backplanes, expanders, front panel, middle plane and PDB. The command will power down all expanders, front panel, BP1, BP2, BP3. Then reset middle plan and PDB modules. System will then power on BP1, BP2, BP3 and front panel in few seconds, and wait another few seconds to power on expanders and HDDs.

Syntax

```
bpsys reset 1
```

Example

```
 Bridge >bpsys reset 1
Reset MCU System


 Bridge >
Performing POST for Smart Serial
…
```

4.14 ipconfig

Show or configure network interface IP address for RJ-472-08 bridge board.

ipconfig [verbose]

Display bridge expander module ethernet port configuration that are saved in persistent storage (flash). Default setting is static IP address (192.168.100.1).

Syntax

```
ipconfig [verbose]
```

Example

```
Bridge >ipconfig

================================================================================
    Physical Address . . . . . . . . . : FA-E1-90-00-00-FF
    IPv6 Enabled . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . : No
    AUTOIP Enabled . . . . . . . . . . : No
    DHCPv6 Enabled . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . : 192.168.100.1
    Subnet Mask  . . . . . . . . . . . : 255.255.255.0
    Default Gateway  . . . . . . . . . : 192.168.100.254
    IPv6 Address . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . : Up
    Ethernet Link Speed  . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . : Full
================================================================================



 Bridge >ipconfig verbose

================================================================================
    Physical Address . . . . . . . . . : FA-E1-90-00-05-BF
    IPv6 Enabled . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . : No
    AUTOIP Enabled . . . . . . . . . . : No
    DHCPv6 Enabled . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . : 192.168.100.1
    Subnet Mask  . . . . . . . . . . . : 255.255.255.0
    Default Gateway  . . . . . . . . . : 192.168.100.254
    IPv6 Address . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . : Up
```

```
    Ethernet Link Speed  . . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . . : Full
    Ethernet speed/duplex initial setting: 0
    Host Name  . . . . . . . . . . . . . : SAS4X_fa:e1:90:0:5:bf
    VLAN Enabled . . . . . . . . . . . . : No
    VLAN CFI/PCP/ID  . . . . . . . . . . : 0/6/ffe
    WakeLan Magic Packet Wakeup Enabled  : No
    WakeLan Unicast Packet Wakeup Enabled: No
=============================================================================
```

ipconfig dhcp

configure ethernet port IP address to DHCP mode.

Syntax

```
ipconfig dhcp
```

Example

```
 Bridge >ipconfig dhcp

 Please reboot for new setting



 Bridge >reset

…
 Bridge >ipconfig

=============================================================================
    Physical Address . . . . . . . . . . : FA-E1-90-00-00-FF
    IPv6 Enabled . . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . . : Yes
    AUTOIP Enabled . . . . . . . . . . . : No
    DHCPv6 Enabled . . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . . : 0.0.0.0
    Subnet Mask  . . . . . . . . . . . . : 0.0.0.0
    Default Gateway  . . . . . . . . . . : 0.0.0.0
    IPv6 Address . . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . . : Up
    Ethernet Link Speed  . . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . . : Full
=============================================================================
```

ipconfig static <ip_addr> <netmask> <gateway>

Configure ethernet port to static mode and assign the IP address, netmask and gateway.

Syntax

```
ipconfig static <IP_addr> <Netmask> <Gateway>
```

Example

```
 Bridge >ipconfig static 192.168.100.1 255.255.255.0 192.168.100.254
```

```
 Please reboot for new setting



 Bridge >reset


…
 Bridge >ipconfig

================================================================================
    Physical Address . . . . . . . . . . : FA-E1-90-00-00-FF
    IPv6 Enabled . . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . . : No
    AUTOIP Enabled . . . . . . . . . . . : No
    DHCPv6 Enabled . . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . . : 192.168.100.1
    Subnet Mask  . . . . . . . . . . . . : 255.255.255.0
    Default Gateway  . . . . . . . . . . : 192.168.100.254
    IPv6 Address . . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . . : Up
    Ethernet Link Speed  . . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . . : Full
================================================================================
```

ipconfig auto

Configure ethernet port to auto IP mode.

Syntax

```
ipconfig auto
```

Example

```
 Bridge >ipconfig auto

 Please reboot for new setting



 Bridge >reset


…
Bridge >ipconfig

================================================================================
    Physical Address . . . . . . . . . . : FA-E1-90-00-00-FF
    IPv6 Enabled . . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . . : No
    AUTOIP Enabled . . . . . . . . . . . : Yes
    DHCPv6 Enabled . . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . . : 169.254.104.78
    Subnet Mask  . . . . . . . . . . . . : 255.255.0.0
    Default Gateway  . . . . . . . . . . : 0.0.0.0
    IPv6 Address . . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . . : Up
    Ethernet Link Speed  . . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . . : Full
================================================================================
```

ipconfig dhcp_auto

Configure ethernet port to DHCP and auto IP mode that system will try to get IP address from DHCP. If no DHCP server exist, system will use auto IP mode.

Syntax

```
ipconfig dhcp_auto
```

Example

```
 Bridge >ipconfig dhcp_auto

 Please reboot for new setting



 Bridge >reset


…
Bridge >ipconfig

================================================================================
    Physical Address . . . . . . . . . : FA-E1-90-00-00-FF
    IPv6 Enabled . . . . . . . . . . . : No
    DHCP Enabled . . . . . . . . . . . : Yes
    AUTOIP Enabled . . . . . . . . . . : Yes
    DHCPv6 Enabled . . . . . . . . . . : No
    IP Address . . . . . . . . . . . . : 169.254.104.78
    Subnet Mask  . . . . . . . . . . . : 255.255.0.0
    Default Gateway  . . . . . . . . . : 0.0.0.0
    IPv6 Address . . . . . . . . . . . :
0000:0000:0000:0000:0000:0000:0000:0000
    Ethernet Link Status . . . . . . . : Up
    Ethernet Link Speed  . . . . . . . : 1000 Mbps
    Ethernet Link Duplex Mode  . . . . : Full
================================================================================
```

4.15 fdl

fdl <0|1> <0> <y>

Download firmware or manufacture config (MFG) image from console or telnet. The download protocol is XMODEM, refer to Figure 18

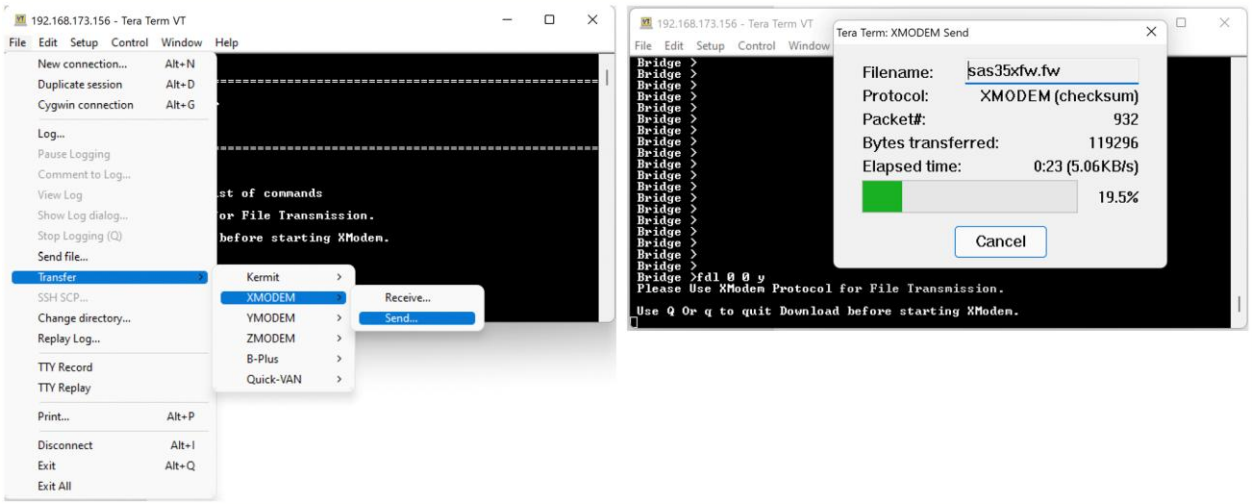

FIGURE 20 FIRMWARE UPGRADE BY XMODEM

Syntax – Download Firmware

```
fdl 0 0 y
```

Syntax – Download MFG

```
fdl 1 0 y
```

Example

```
Bridge >fdl 0 0 y
Please Use XModem Protocol for File Transmission.

Use Q Or q to quit Download before starting XModem.


Received 612736Bytes
Buffer Download Complete
       Start downloading to Edge Expanders ...
       Edge1 Download Successfully!
       Edge2 Download Successfully!
       Edge3 Download Successfully!



Bridge >fdl 1 0 y
Please Use XModem Protocol for File Transmission.

Use Q Or q to quit Download before starting XModem.


Received 49152Bytes
Buffer Download Complete
       Start downloading to Edge Expanders ...
       Edge1 Download Successfully!
       Edge2 Download Successfully!
       Edge3 Download Successfully!
```

4.16 tftpfdl

tftpfdl <0|1> <0> <y> <tftp_server_ip> <filename>

To upgrade firmware through TFTP, open a TFTP server (Figure 19) on your console host and specify the path in TFTP server to the directory where the firmware is stored. Please be sure the TFTP server is reachable to the JBOD system.

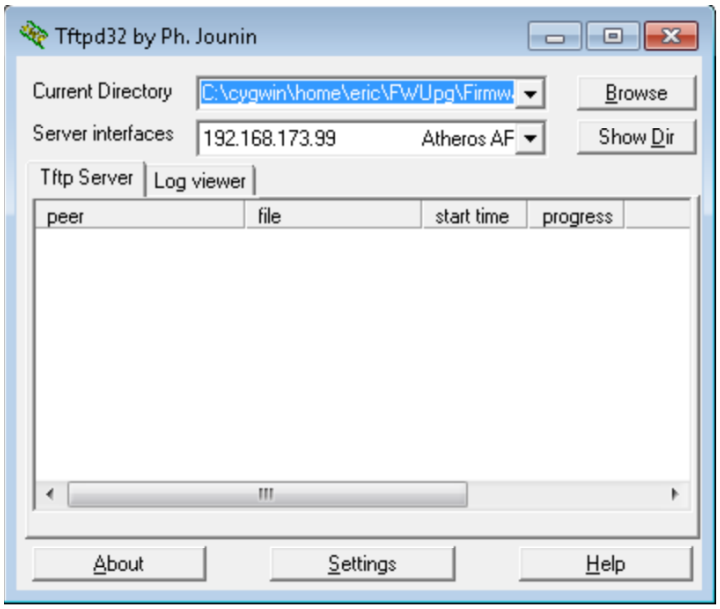

FIGURE 21 TFTP SERVER FOR UPGRADE

Syntax – Download Firmware through TFTP

```
tftpfdl 0 0 y <tftp_server_ip> <fw_filename>
```

Syntax – Download MFG through TFTP

```
tftpfdl 1 0 y <tftp_server_ip> <mfg_filename>
```

Example

```
Bridge >tftpfdl 0 0 y 192.168.173.139 RJ472-3.0.1.fw

 Received 612776Bytes
 Firmware Download Complete
        Start downloading to Edge Expanders ...
        Edge1 Download Successfully!
        Edge2 Download Successfully!
        Edge3 Download Successfully!
```

4.17 user

Assign or show user account and password for WEB, TELNET and IPMI (v1.5) login

The login session will be closed after 5 minutes (timeout) if no activities

The default user account is '**account**'

The default user password is '**password**'

Syntax

```
user [<account> <password>]
```

Example

```
PRIMARY>user

ACCOUNT: account
PASSWORD: password



PRIMARY>user test 123

New ACCOUNT: test
New PASSWORD: 123
```

5 Appendix

5.1 Web REST API

Overview

RJ-460/472 provide a lightweight REST service for programmatic monitoring over HTTP. The API is session-based (login first) and returns JSON. This appendix summarizes the URL format, resources, and example curl usage

Authentication

Before any API call, create a login session with the web server:

```
curl -X POST -d "xser=<account>&xassword=<password>&login=login"
http://<host>/login
```

Use the same account/password set by the CLI user command. If there is no web activity, the session times out after ~300 seconds and you must log in again.

Base URL & Path Structure

```
http://<host>:<port>/api/<SAS_Address>/<item_group>/[id]
```

- host: Expander IP (default 192.168.100.1). port: 80 (fixed).
- SAS_Address: Target bridge expander SAS address in uppercase hex.
- item_group: Resource collection (see A.4).
- id (optional): Zero-based index into the collection.

Responses are JSON.

How to get the SAS address

```
curl -s http://192.168.100.1/version.asp | grep Bridge | cut -d '|' -f 3 | tr -d
'_'
```

Resources

Item Group	*Description*	*ID Behavior*
version	FW/MFG/board/serial for bridge/edges	id optional (0…); without id returns all boards
element	Counts of objects: driverslot, fan, voltage, temperature, powersupply	no id
driverslot	Per-bay link rate, SAS address, type	id = bay index
fan	Per-fan rpm, pwm, status	id = fan index
voltage	Named voltage readings	id = sensor index
temperature	Named temperature readings	id = sensor index
powersupply	PSU presence/status	id = PSU index

* Tip: Use /element first to learn the valid index ranges for your chassis.

Example - Query resources

```
curl -X GET http://192.168.100.1/api/5FAE1970000005FF/version
curl -X GET http://192.168.100.1/api/5FAE1970000005FF/element
curl -X GET http://192.168.100.1/api/5FAE1970000005FF/fan
curl -X GET http://192.168.100.1/api/5FAE1970000005FF/fan/0
```

Example – Response ‘version’

```
{
  "bridge":{"board":"RJ-472 Bridge(P)","serial":"LAB123",
            "sasaddr":"5FAE1909100000FF","FW":"4.0.2.0","MFG":"4.0.2.0"},
  "edge1":{ "...": "..." },
  "edge2":{ "...": "..." },
  "edge3":{ "...": "..." }
}
```

Example – Response ‘fan’

```
{ "data":[ {"name":"Fan01","rpm":7920,"pwm":64,"status":"OK"}, ... ] }
```

Example – Response ‘fan/0’

```
{ "name":"Fan01","rpm":8100,"pwm":64,"status":"OK" }
```

Operational notes

- Always login first; repeat after the session idle timeout (~300 s).

- The API is read-only and intended for monitoring/integration scripts. (Control and configuration remain via CLI)
- Keep <SAS_Address> uppercase and ensure you're querying the bridge expander

5.2 IPMI over LAN

Overview

RJ-460 / RJ-472 expose a read-only IPMI interface over Ethernet for automation and monitoring with tools like ipmitool. The IPMI view mirrors the system's CLI/SES sensors: fans (status), temperatures, and key voltages; it complements (but doesn't replace) the richer CLI/WEB/REST data. Ethernet and IP setup for the bridge module follow the same network configuration used by CLI/WEB.

Network prerequisites

- Connect to the Bridge module's Ethernet port. Link is 10/100/1000 Mbps, Full Duplex.
- Configure an IP using the CLI:
  - ipconfig auto (Auto-IP) or ipconfig dhcp_auto (DHCP, fallback to Auto-IP). Reboot to apply.
  - Use ipconfig to verify address, mask, gateway and link status.

Quick start

Read FRU, SDR, and Sensors from a management host (replace IP, account, password):

```
ipmitool -I lan -H 192.168.100.11 -U <account> -P <password> fru
ipmitool -I lan -H 192.168.100.11 -U <account> -P <password> sdr
ipmitool -I lan -H 192.168.100.11 -U <account> -P <password> sensor
```

Typical FRU fields (example):

- Board Mfg/Product/Serial/Part Number
- Board Extra entries such as SAS-<addr>, build date, SW-<ver>, MFG-<ver> (software & manufacturing image revisions).

What you'll see in IPMI

Temperatures (°C)

Sensors include Bridge Die, Edge1/2/3 Die, and zone probes (MP L/R, BP1/2/3 L/R, FP L/R). Values match the CLI sensor page (e.g., Bridge/Edges plus MP/BP/FP zones), but IPMI returns the value as a single byte raw hex that equals ˚C (e.g., 0x2c → 44 ˚C).

Voltages (V)

PDB 3.3/5/12 V rails, MP/BP/FP 5 V rails are exposed. IPMI shows a raw byte in 0.1 V units (e.g., 0x32 → 5.0 V, 0x78 → 12.0 V). These rails correspond to the CLI sensor voltage list (PDB, MP, BPx, FP).

Fans

Fan01…Fan04 appear as discrete status (OK/Fail) in IPMI; RPM is not provided over IPMI. For RPM & PWM, use CLI sensor or the REST API /fan endpoint, which returns per-fan rpm/pwm/status.

IPMI object	***What it represents***	***Raw → Eng. units***
Bridge Die, EdgeX Die	Expander die temps	raw_hex (e.g., 0x2c) = °C (44 °C)
MP.*, BPx.*, FP.*	Zone temps (Main Plane, Backplanes, Front Panel)	same as above (°C)
PDB 3.3V/5.0V/12V, MP 5V/12V, BPx 5V, FP 5V	Rails	raw_decimal / 10 = Volts (e.g., 0x32=50→5.0 V)
Fan01..Fan04	Fan health	0x00 = OK (status only; use REST/CLI for RPM)

5.3 Logical ID Assignment

What is the Logical ID

The Logical ID (LID) is the enclosure's persistent identifier used by hosts/RAID stacks to recognize the same JBOD across re-cabling, firmware updates, or expander replacements. It follows SAS designator conventions—use 16 hex digits and start with 5 (NAA 5h, IEEE Registered Extended). When a chassis has dual domains (primary/secondary expanders), both domains must expose the same LID so the host sees a single enclosure.
In factory default, the running LID is derived from the backplane MCU unique ID; the manufacture page shows both the stored/manual value and the current (running) value in parentheses. "manufacture lid 0000000000000000" restores the default behavior.

Command - manufacture

All commands are entered at the bridge CLI prompt. Prepend edge1, edge2, edge3 (RJ-472) to run the same command on an edge expander.

```
Bridge >manufacture
Model ID: 0
Hardware Revision: 1
Primary: 1
Serial Number: 14225200912132
SAS Address: 5FAE1970000005FF
Logical ID:  0000000000000000(5FAE1970005FF025)
Edge SAS:    0000000000000000
MAC Address: FAE1900005FF
SMP FECE0: 0

Parameters 0: FF FF FF FF FF FF FF FF
Parameters 1: FF FF FF FF FF FF FF FF
Zone 1:          00 00 00 00 00 00 00 00
Zone 2:          00 00 00 00 00 00 00 00
Zone 3:          00 00 00 00 00 00 00 00
Zone 4:          00 00 00 00 00 00 00 00
Zone 5:          00 00 00 00 00 00 00 00
```

Typical fields:

Model ID, Hardware Revision, Primary (1 = primary domain)

Serial Number, SAS Address

Logical ID: <stored_value>(<running_value>)

Edge SAS, MAC Address, vendor parameters/zone bytes

Logical ID: 0000000000000000 (5FAE1970005FF025)

Here, 000…0 is the stored/manual LID (0 = use default), while the value in parentheses is the active LID generated from the backplane MCU unique ID.

Set a manual Logical ID

manufacture lid <16-hex>

<16-hex> must begin with 5 (e.g., 5000000000000001).

The new LID is saved to NVRAM.

Use the same command under each edge context to keep domains consistent

```
Bridge > manufacture lid 5000000000000001
Bridge > edge1 manufacture lid 5000000000000001
Bridge > edge2 manufacture lid 5000000000000001
Bridge > edge3 manufacture lid 5000000000000001   (RJ-472 only)
```

Restore default (auto-derived) LID

```
Bridge > manufacture lid 0000000000000000
Bridge > edge1 manufacture lid 0000000000000000
Bridge > edge2 manufacture lid 0000000000000000
Bridge > edge3 manufacture lid 0000000000000000   (RJ-472 only)
```

Recommended workflow (primary + secondary domains)

1. Pick a unique LID for the enclosure (example):
   5000000000000001 ← 16 hex digits, starts with 5.
2. Program the Bridge (primary domain):
   Bridge > manufacture lid 5000000000000001
3. Program all Edge expanders (primary domain):
   Bridge > edge1 manufacture lid 5000000000000001
   Bridge > edge2 manufacture lid 5000000000000001
   Bridge > edge3 manufacture lid 5000000000000001 (RJ-472 only)
4. Redo step 2-3 to secondary domain if you want the secondary domain share the same LID
5. Power cycle the enclosure.
   A full power cycle ensures the new LID is advertised by all domains
6. Verify after reboot:
   Bridge >manufacture
   Bridge >edge1 manufacture
   Bridge >edge2 manufacture
   Bridge >edge3 manufacture (RJ-472 only)
7. Confirm that stored and running values both show 5000000000000001.

Validation tips

Host view (SES/OS tools): After the power cycle, the OS/RAID utility should report a single enclosure whose Enclosure Logical Identifier equals your LID.

sasaddr sanity check: Use sasaddr to review expander SAS addresses for each domain. LID is independent from the physical SAS addresses, but should remain consistent across domains.

Avoid duplicates: Never reuse the same LID on two different chassis in the same fabric; duplicates can confuse multipath/SES and cause enclosure re-enumeration.

Naming scheme suggestions

Serial-based: ‘5’ + 15 hex digits derived from the chassis serial (hash or zero-padded).

Fleet scheme: Reserve a block (e.g., 5000000000xxxxxx) and assign per-rack/row to guarantee uniqueness at scale.

Change control: Record programmed LIDs in your asset DB alongside serials and SAS addresses.