PS0802 RS232 ASCII PROTOCOL

|  |  |
| --- | --- |
| Version | V1.0 |
| Date | 2019-06-10 |
| Draft by | Farrell Tan |

目录

[PS0802 RS232 ASCII PROTOCOL 0](#_Toc11662087)

[1 Serial communication protocol format 2](#_Toc11662088)

[2 Routing command 3](#_Toc11662089)

[2.1 Video routing 3](#_Toc11662090)

[3 Audio Configuration 4](#_Toc11662091)

[3.1 Binding Audio Source to Input 4](#_Toc11662092)

[3.2 Mute Configuration 5](#_Toc11662093)

[3.3 Volume Configuration 6](#_Toc11662094)

[4 Video Output Blanking Configuration 6](#_Toc11662095)

[5 Video Output Freeze Configuration 7](#_Toc11662096)

[6 Video Output HDCP Configuration 7](#_Toc11662097)

[7 Output Signal format 8](#_Toc11662098)

[8 Input Signal format 9](#_Toc11662099)

[9 I/O Configuration 9](#_Toc11662100)

[~~10~~ ~~EDID~~ 10](#_Toc11662101)

[~~10.1~~ ~~Read EDID~~ 10](#_Toc11662102)

[~~10.2~~ ~~Write EDID~~ 11](#_Toc11662103)

[~~10.3~~ ~~EDID MODE Configuration~~ 12](#_Toc11662104)

[11 System command 13](#_Toc11662105)

[11.1 Device size 13](#_Toc11662106)

[11.2 Device IP 14](#_Toc11662107)

[11.3 System Reset 14](#_Toc11662108)

[11.4 Software Version (Read only) 15](#_Toc11662109)

# Serial communication protocol format

Baud Rate: 57600

Data bits: 8

Parity: None

Stop bits: 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (N bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET/GET | Space | The target that handles this command. | Space | Command type | [Parameter1] [Parameter2] | ⮠  This is ASCII carriage return 0x0d |

**Notes:**

Space is the ASCII character 0x20

⮠ Represents the ASCII character 0x0d

All Return messages are always terminated by CR/LF, the ASCII characters 0x0d 0x0a

All items shown in square brackets, [], are optional.

Any SET command that contains leading zeroes should not include the leading zeros in any response message.

**The value ranges for certain commands are not given, please state and minimum and maximum values for each command that uses a numerical value range.**

# Routing command

## Video routing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | INx/INxx/INxxx  x is the input port number  For example, IN1/IN01/IN001 | Space | VIDEO | ALL | ⮠  This is ASCII carriage return 0x0d |

1. Set video route: Input port-x/xx/xxx switch to HDMI/HDBST output

For example, SET video route: Input port 1 switch to all output ports

Send: SET IN1 VIDEO ALL ⮠ Receive: IN1 VIDEO ALL

NOTE:

1. We can switch one input port to one output port, or to more than one output ports at a time.

# Audio Configuration

## Binding Audio Source to Input

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | INx/INxx/INxxx  x is the input port number  For example, IN1/IN01/IN001 | Space | AUDIOIN | EMBEDDED  L/R1  L/R2  L/R3  L/R4  L/R5  L/R6  L/R7  L/R8 | ⮠  This is ASCII carriage return 0x0d |

A: Binging the audio source to input:

For example: Binging the L/R1 to input 1

Send: SET IN1 AUDIOIN L/R1⮠ Receive: IN1 AUDIOIN L/R1⮠

For example: Binging the EMBEDDED to input 1

Send: SET IN1 AUDIOIN EMBEDDED ⮠ Receive: IN1 AUDIOIN EMBEDDED⮠

For example: Get the input 1 audio binding status:

Send: GET IN1 AUDIOIN ⮠ Receive: IN1 AUDIOIN EMBEDDED⮠

NOTE:

1. When switch to input 1, then the audio will select L/R1;
2. When the input audio binging to EMBEDDED, if the input video if DVI, then will no audio;

## Mute Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | LINE  SPEAKER  MIC | Space | MUTE | ON  OFF | ⮠  This is ASCII carriage return 0x0d |

A: Mute/Unmute the audio

For example: Mute the line

Send: SET LINE MUTE ON⮠ Receive: LINE MUE ON⮠

For example: Mute the Speaker

Send: SET SPEAKER MUTE ON⮠ Receive: SPEAKER MUE ON⮠

For example: Mute the MIC

Send: SET MIC MUTE ON⮠ Receive: MIC MUE ON⮠

For example: Get the MIC Mute status

Send: GET MIC MUTE ⮠ Receive: MIC MUE ON⮠

NOTE:

1. LINE: The input audio;
2. SPEAKER: Speaker output;
3. MIC: Microphone input；

## Volume Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | SPEAKER  MIC | Space | VOLUME |  | ⮠  This is ASCII carriage return 0x0d |

A: Mute/Unmute the audio

For example: SET the Speaker volume

Send: SET SPEAKER VOLUME 50⮠ Receive: SPEAKER VOLUME 50⮠

For example: Get the MIC volume

Send: GET MIC VOLUME ⮠ Receive: MIC VOLUME 50⮠

NOTE:

1. SPEAKER: Speaker output;
2. MIC: Microphone input；

# Video Output Blanking Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | OUT | Space | BLANK | ON  OFF | ⮠  This is ASCII carriage return 0x0d |

A: Blank video output on/off

For example: Blank video output on

Send: SET OUT BLANK ON⮠ Receive: OUT BLANK ON⮠

For example: Get the Blank status

Send: GET OUT BLANK ⮠ Receive: OUT BLANK OFF ⮠

NOTE:

1. When Blank on, then the output will show blank video
2. When Blank off, then the output will show input video

# Video Output Freeze Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | OUT | Space | FREEZE | ON  OFF | ⮠  This is ASCII carriage return 0x0d |

A: Freeze on/off

For example: video output Freeze on

Send: SET OUT FREEZE ON⮠ Receive: OUT FREEZE ON⮠

For example: Get the FREEZE status

Send: GET OUT FREEZE ⮠ Receive: OUT FREEZE ON ⮠

# Video Output HDCP Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | OUT | Space | HDCP | AUTO  1.4  2.2  OFF | ⮠  This is ASCII carriage return 0x0d |

A: Freeze on/off

For example: video output Freeze on

Send: SET OUTHDCP AUTO⮠ Receive: OUT HDCP AUTO⮠

For example: Get the FREEZE status

Send: GET OUT HDCP⮠ Receive: OUT HDCP AUTO ⮠

# Output Signal format

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (N bytes) | Command parameters  (0 or N bytes) | Command tail  (1 byte) |
| SET/GET | Space | OUT | Space | OUT-SIGNAL | Null (0 byte)  or  TYPE@Resolution (N bytes) | ⮠  This is ASCII carriage return 0x0d |

1. GET output signal format of output board:

Send: GET OUT OUT-SIGNAL⮠ Receive: OUT OUT-SIGNAL UHD-HDMI@3840x2160p60⮠

Note: Output resolution list:

1024x768p60

1280x720p50

1280x720p60

1360x768p60

1600x1200p60

1920x1080p50

1920x1080p60

1920x1200p60

3840x2160p30

3840x2160p50

3840x2160p60

AUTO

# Input Signal format

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (N bytes) | Command parameters  (0 or N bytes) | Command tail  (1 byte) |
| GET | Space | INx/INxx/INxxx  x is the input port number  For example, IN1/IN01/IN001 | Space | IN-SIGNAL | Null (0 byte)  or  TYPE@Resolution (N bytes) | ⮠  This is ASCII carriage return 0x0d |

1. GET output signal format of output board:

Send: GET IN1 IN-SIGNAL⮠ Receive: IN1 IN-SIGNAL UHD-HDMI@4K2Kp60⮠

# I/O Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (N bytes) | Command parameters  (0 or N bytes) | Command tail  (1 byte) |
| GET/SET | Space | Ix/Ixx/Ixxx  x is the input port number  For example, IN1/IN01/IN001 | Space | LEVEL | HIGH  LOW | ⮠  This is ASCII carriage return 0x0d |

1. GET I/O Status:

Send: GET I1 LEVEL⮠ Receive: I1 LEVEL HIGH⮠

1. SET Ox LEVEL:

Send: SET O1 LEVEL HIGH Receive: O1 LEVEL HIGH⮠

# ~~EDID~~

## ~~Read EDID~~

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ~~Operation type~~  ~~(3 byte)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Target~~  ~~(N bytes)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Command type~~  ~~(10 bytes)~~ | ~~Command parameters~~  ~~(N bytes)~~ | ~~Command tail~~  ~~(1 byte)~~ |
| ~~GET~~ | ~~Space~~ | ~~OUTx/OUTxx/OUTxxx~~  ~~x is the output port number~~  ~~For example, OUT1/OUT01/OUT001~~ | ~~Space~~ | ~~EDID~~ | ~~Send: (4/5 byte)~~  ~~PART1~~  ~~PART2~~  ~~……~~  ~~PART16~~  ~~Receive: (53/54 bytes)~~  ~~PART1 d1 d2 … d16~~  ~~PART2 d1 d2 … d16~~  ~~……~~  ~~PART16 d1 d2 … d16~~ | ~~⮠~~  ~~This is ASCII carriage return 0x0d~~ |

1. ~~GET (Read) EDID data from output port~~

~~For example, GET (Read) EDID data from output port 1 (Read 16 times)~~

~~Send: GET OUT1 EDID PART1⮠ Receive: OUT1 EDID PART1 00 FF FF FF FF FF FF 00 63 18 22 00 66 00 00 00~~

~~Send: GET OUT1 EDID PART2⮠ Receive: OUT1 EDID PART2 05 1C 01 03 80 59 32 78 0A EE 91 A3 54 4C 99 26~~

~~Send: GET OUT1 EDID PART3⮠ Receive: OUT1 EDID PART3 0F 50 54 01 08 00 81 C0 81 C0 81 00 81 80 95 00~~

~~Send: GET OUT1 EDID PART4⮠ Receive: OUT1 EDID PART4 A9 C0 B3 00 01 01 08 E8 00 30 F2 70 5A 80 B0 58~~

~~Send: GET OUT1 EDID PART5⮠ Receive: OUT1 EDID PART5 8A 00 C4 8E 21 00 00 1E 02 3A 80 18 71 38 2D 40~~

~~Send: GET OUT1 EDID PART6⮠ Receive: OUT1 EDID PART6 58 2C 45 00 50 1D 74 00 00 1E 00 00 00 FD 00 17~~

~~Send: GET OUT1 EDID PART7⮠ Receive: OUT1 EDID PART7 3C 0F 88 3C 00 0A 20 20 20 20 20 20 00 00 00 FC~~

~~Send: GET OUT1 EDID PART8⮠ Receive: OUT1 EDID PART8 00 48 44 4D 49 0A 20 20 20 20 20 20 20 20 01 16~~

~~Send: GET OUT1 EDID PART9⮠ Receive: OUT1 EDID PART9 02 03 40 F1 55 61 10 1F 04 13 05 14 20 21 22 5D~~

~~Send: GET OUT1 EDID PART10⮠ Receive: OUT1 EDID PART10 5E 5F 60 65 66 07 12 03 16 01 23 09 07 07 83 01~~

~~Send: GET OUT1 EDID PART11⮠ Receive: OUT1 EDID PART11 00 00 6E 03 0C 00 30 00 B8 3C 21 10 80 01 02 03~~

~~Send: GET OUT1 EDID PART12⮠ Receive: OUT1 EDID PART12 04 67 D8 5D C4 01 78 80 03 E2 00 4F E3 0F 01 E0~~

~~Send: GET OUT1 EDID PART13⮠ Receive: OUT1 EDID PART13 01 1D 80 D0 72 1C 16 20 10 2C 25 80 50 1D 74 00~~

~~Send: GET OUT1 EDID PART14⮠ Receive: OUT1 EDID PART14 00 9E 66 21 56 AA 51 00 1E 30 46 8F 33 00 50 1D~~

~~Send: GET OUT1 EDID PART15⮠ Receive: OUT1 EDID PART15 74 00 00 1E 00 00 00 00 00 00 00 00 00 00 00 00~~

~~Send: GET OUT1 EDID PART16 Receive: OUT1 EDID PART16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A1~~

~~NOTE:~~

1. ~~EDID totally have 256 bytes, so we need to read 16 times and 16 bytes will be read per time.~~

## ~~Write EDID~~

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ~~Operation type~~  ~~(3 byte)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Target~~  ~~(N bytes)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Command type~~  ~~(10 bytes)~~ | ~~Command parameters~~  ~~(53/54 bytes)~~ | ~~Command tail~~  ~~(1 byte)~~ |
| ~~SET~~ | ~~Space~~ | ~~INx/INxx/INxxx~~  ~~x is the input port number~~  ~~For example, IN1/IN01/IN001~~ | ~~Space~~ | ~~EDID~~ | ~~PART1 d1 d2 … d16~~  ~~PART2 d1 d2 … d16~~  ~~……~~  ~~PART16 d1 d2 … d16~~ | ~~⮠~~  ~~This is ASCII carriage return 0x0d~~ |

1. ~~SET (Write) EDID data to input port~~

~~For example, SET (Write) EDID data to input port 1 (Write 16 times)~~

~~Send: SET IN1 EDID PART1 00 FF FF FF FF FF FF 00 63 18 22 00 66 00 00 00⮠ Receive: IN1 EDID PART1 00 FF FF FF FF FF FF 00 63 18 22 00 66 00 00 00~~

~~Send: SET IN1 EDID PART2 05 1C 01 03 80 59 32 78 0A EE 91 A3 54 4C 99 26⮠ Receive: IN1 EDID PART2 05 1C 01 03 80 59 32 78 0A EE 91 A3 54 4C 99 26~~

~~Send: SET IN1 EDID PART3 0F 50 54 01 08 00 81 C0 81 C0 81 00 81 80 95 00⮠ Receive: IN1 EDID PART3 0F 50 54 01 08 00 81 C0 81 C0 81 00 81 80 95 00~~

~~Send: SET IN1 EDID PART4 A9 C0 B3 00 01 01 08 E8 00 30 F2 70 5A 80 B0 58⮠ Receive: IN1 EDID PART4 A9 C0 B3 00 01 01 08 E8 00 30 F2 70 5A 80 B0 58~~

~~Send: SET IN1 EDID PART5 8A 00 C4 8E 21 00 00 1E 02 3A 80 18 71 38 2D 40⮠ Receive: IN1 EDID PART5 8A 00 C4 8E 21 00 00 1E 02 3A 80 18 71 38 2D 40~~

~~Send: SET IN1 EDID PART6 58 2C 45 00 50 1D 74 00 00 1E 00 00 00 FD 00 17⮠ Receive: IN1 EDID PART6 58 2C 45 00 50 1D 74 00 00 1E 00 00 00 FD 00 17~~

~~Send: SET IN1 EDID PART7 3C 0F 88 3C 00 0A 20 20 20 20 20 20 00 00 00 FC⮠ Receive: IN1 EDID PART7 3C 0F 88 3C 00 0A 20 20 20 20 20 20 00 00 00 FC~~

~~Send: SET IN1 EDID PART8 00 48 44 4D 49 0A 20 20 20 20 20 20 20 20 01 16⮠ Receive: IN1 EDID PART8 00 48 44 4D 49 0A 20 20 20 20 20 20 20 20 01 16~~

~~Send: SET IN1 EDID PART9 02 03 40 F1 55 61 10 1F 04 13 05 14 20 21 22 5D⮠ Receive: IN1 EDID PART9 02 03 40 F1 55 61 10 1F 04 13 05 14 20 21 22 5D~~

~~Send: SET IN1 EDID PART10 5E 5F 60 65 66 07 12 03 16 01 23 09 07 07 83 01⮠ Receive: IN1 EDID PART10 5E 5F 60 65 66 07 12 03 16 01 23 09 07 07 83 01~~

~~Send: SET IN1 EDID PART11 00 00 6E 03 0C 00 30 00 B8 3C 21 10 80 01 02 03⮠ Receive: IN1 EDID PART11 00 00 6E 03 0C 00 30 00 B8 3C 21 10 80 01 02 03~~

~~Send: SET IN1 EDID PART12 04 67 D8 5D C4 01 78 80 03 E2 00 4F E3 0F 01 E0⮠ Receive: IN1 EDID PART12 04 67 D8 5D C4 01 78 80 03 E2 00 4F E3 0F 01 E0~~

~~Send: SET IN1 EDID PART13 01 1D 80 D0 72 1C 16 20 10 2C 25 80 50 1D 74 00⮠ Receive: IN1 EDID PART13 01 1D 80 D0 72 1C 16 20 10 2C 25 80 50 1D 74 00~~

~~Send: SET IN1 EDID PART14 00 9E 66 21 56 AA 51 00 1E 30 46 8F 33 00 50 1D⮠ Receive: IN1 EDID PART14 00 9E 66 21 56 AA 51 00 1E 30 46 8F 33 00 50 1D~~

~~Send: SET IN1 EDID PART15 74 00 00 1E 00 00 00 00 00 00 00 00 00 00 00 00⮠ Receive: IN1 EDID PART15 74 00 00 1E 00 00 00 00 00 00 00 00 00 00 00 00~~

~~Send: SET IN1 EDID PART16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A1⮠ Receive: IN1 EDID PART16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A1~~

~~NOTE:~~

1. ~~EDID totally have 256 bytes, so we need to write 16 times and 16 bytes will be write per time.~~

## ~~EDID MODE Configuration~~

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ~~Operation type~~  ~~(3 byte)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Target~~  ~~(N bytes)~~ | ~~Spacer~~  ~~(1 byte)~~ | ~~Command type~~  ~~(10 bytes)~~ | ~~Command parameters~~  ~~(53/54 bytes)~~ | ~~Command tail~~  ~~(1 byte)~~ |
| ~~SET~~ | ~~Space~~ | ~~SYS~~ | ~~Space~~ | ~~EDIDMODE~~ | ~~4K60~~  ~~4K30~~  ~~1080P60~~  ~~720P60~~  ~~1920x1200P60~~  ~~MANUAL~~  ~~AUTO~~ | ~~⮠~~  ~~This is ASCII carriage return 0x0d~~ |

~~SET EDID Mode:~~

~~Send: SET SYS EDIDMODE 4K60⮠ Receive: SYS EDIDMODE 4K60⮠~~

# System command

## Device size

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (0/3-7 bytes) | Command tail  (1 byte) |
| SET/GET | Space | SYS | Space | SIZE | a b  a is input ports number  b is output ports number | ⮠  This is ASCII carriage return 0x0d |

1. GET the device size:

For example, GET the device size (16x16)

Send: GET SYS SIZE⮠ Receive: SYS SIZE 16 16

SET the device size:

For example, SET the device size to 8x16

Send: SET SYS SIZE 8 16⮠ Receive: SYS SIZE 8 16

## Device IP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET/GET | Space | SYS | Space | IP | mode,address,mask,gateway | ⮠ This is ASCII carriage return 0x0d |

1. GET the device size:

For example, GET the device IP (MAC: D8B04CB947DF )

Send: GET SYS IP⮠ Receive: SYS IP D8B04CB947DF DHCP,192.168.0.119,255.255.255.0,192.168.0.1

Send: GET SYS IP⮠ Receive: SYS IP D8B04CB947DF STATIC,192.168.0.222,255.255.255.0,192.168.0.24

1. SET the device IP:

For example, Set the device IP to STATIC 192.168.1.1

Send: SET SYS IP STATIC,192.168.0.222,255.255.255.0,192.168.0.1⮠

Receive: SYS IP STATIC,192.168.0.222,255.255.255.0,192.168.0.1

For example, Set the device IP to HDCP (auto obtain)

Send: SET SYS IP DHCP⮠

Receive: SYS IP DHCP

## System Reset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (N bytes) | Command tail  (1 byte) |
| SET | Space | SYS | Space | RESET | ALL | ⮠  This is ASCII carriage return 0x0d |

1. SET (Reset) the device :

For example, Set (Reset) the device

Send: SET SYS RESET ALL⮠ Receive: SYS RESET ALL

## Software Version (Read only)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation type  (3 byte) | Spacer  (1 byte) | Target  (N bytes) | Spacer  (1 byte) | Command type  (10 bytes) | Command parameters  (0 or 1/2/3 bytes) | Command tail  (1 byte) |
| GET | Space | SYS | Space | VERSION | Send: Null (0 byte)  Receive: N bytes | ⮠  This is ASCII carriage return 0x0d |

1. Get the software version of output board:

For example, Get the software version of output port 1

Send: GET SYS VERSION⮠ Receive: SYS VERSION 2019/01/01-12:00:00