

5. Control

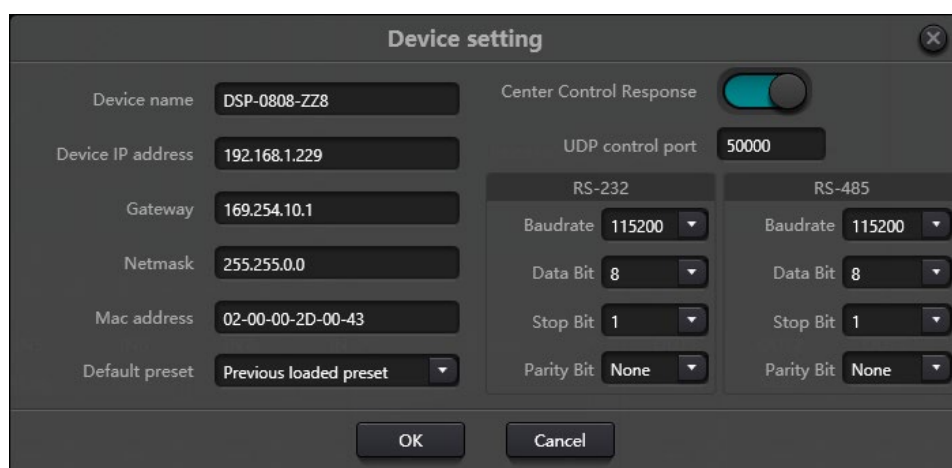
5.1 External Control Programmer

External control programmer supports UDP and RS232 and controls protocols covering all controls parameters of processor, including parameter controls, parameter acquisition and preset calling.

When UDP controls are used, the default port is 50000. Ports can be set in "Device Setting" via upper computer software.

When using RS232 controls are used, the default baud rate is 115200, digit bits is 8, and stop bit is 1, no parity bit. Similarly, they can be set in "Device Setting" . **The interval between messages shall keep more than 100mS for RS232 sending.**

If central control needs reply, please turn on central control reply switch in "Device Setting" .



5.2 Control Protocol

Because of historical reasons, the latest control protocol adopts variable-length and is fully compatible with old fixed-length control protocols. In protocols, the fourth byte is used to distinguish versions. 0- indicates V1 version (previous versions) and 1- indicates V2 version (current protocol version).

The difference between V1 and V2 is V1 may control all processing module parameters, but one command can only control one parameter. If a parameter is needed to control continuous multiple channels, then V2 version shall be used. In other words, in condition that users need to press one key on key panel to trigger GPIO output high-/low-level of devices, or send a command via RS232/RS485, then V2 version will be the best choice.

Software coding rules (total 12 bytes)

byte1	byte2	byte3	byte4	byte5~132
0xb3	Message Type	Length	Version No.	Data

V1:

Information types (byte2): There are three information types including 0x21 (parameter controls), 0x22 (parameter acquisition) and 0x13 (scenario switch).

Length (byte3): invalid.

0x21 (parameter control):

At this point, Databyte5~12 are respectively

byte 5~6	byte 7~8	byte 9~10	byte 11~12
Module ID	Parameter Type	Parameter 1	Parameter 2

Please refer to Appendix A to get the distribution of **Module ID** (byte5~6).

Please refer to Appendix B for **Parameter types** (byte7~8).

When **Parameter 1** (byte9~10) has only one parameter, then only parameter 1 is valid, such as control compressor switch.

Parameter 2 (byte11~12) only valid when there are two parameters, such as control output channel 1 mute. Parameter value 1 shall be filled in input channel number from 0. Parameter value 2 shall be filled in 1 (mute).

Exception: Matrix routing has three parameters. The first one is input channel number, the second one is output channel number, and the third one is routing switch. At this point, byte9 of parameter value 1 shall be filled in input channel numbers, byte 10 shall be filled in output channel number, and parameter 2 shall be filled in routing switch.

0x22 (Parameter Acquisition):

Parameter acquisition rules are the same with parameter controls. The difference between them is values acquired shall be filled in parameter 1 and parameter 2.

0x13 (Scenario Switch):

Users only need to fill scenario numbers (0~15) in byte5 and 0 in byte6~12.

Note: Central control command of V1 version can acquire code through software menu bar of PC. For customized development, please use this protocol rule.

V2:

Message types (byte2): There are three message types (byte2) including 0x21 (parameter controls) , 0x22 (parameter acquisition) , 0x13 (scenario switch) , 0x74 (other controls) and 0x6e (Dante routing) .

Length (byte3): Fill in corresponding data section length based on information type. The length can be longer when actual sending is carried out. Total data volume can be got through adding 4 byte header information to data length.

1. Parameter Control (0x21)

At this point, the formats of data section are as follows.

byte5	byte6	byte7	byte8	byte9~72
Input/Output	Start Channel	End Channel	Parameter Type	Parameter Value

byte5: It indicate control input or output channel, 0x2- input channels and 0x1-output channels

byte6-7: They indicate start and end channel numbers. Channel numbers start from 0.

byte8: This kind of parameter is the same with V1 version. Please refer to Appendix B.

byte9-40: Fill in parameter values of start to end channels. It shall be filled in from the ninth byte. Each parameter value shall take two bytes.

2. Parameter Acquisition (0x22)

Data section format is the same with parameter controls. Parameter values may not be filled in. acquired parameters will be filled in this position.

3. Scenario Switch (0x13)

byte5: Fill in scenario numbers (0-15).

byte6-8: Fill in 0.

4. Other Controls (0x74)

Other controls include but not limited to GPIO, RS232, RS485 and central control replies. The protocol formats are as follows.

GPIO:

byte5	byte6	byte7	byte8	byte9	byte10	byte11	byte12
Control Type	Data Length	Reserved	Reserved	GPIO Direction	Start GPIO	End GPIO	Value

The controlling type for byte5 is 1.

The data length of byte6 is fixed as four bytes.

Byte9 GPIO direction, set input or output. Value 0 indicates input, and value 1 indicates output.

Byte10-11 start GPIO and end GPIO. DSP devices totally have eight GPIOs, which are indicated with number 0-7.

Byte12 is determined according to byte9GPIO direction. The field shall be filled in high (1) /low (0) level for output settings. The field is a return field to read GPIO level value on devices for input settings.

RS232/RS485:

byte5	byte6	byte7	byte8	byte9-132
Control Type	Data Length	Reserved	Reserved	Data

Byte5 is 2 when controlling type RS232, and 3 for RS485.

The data length of byte6 refers to data length that shall be sent via RS232/485 currently.

Byte9-132 shall be filled in data sent by RS232/485.

Central control replies:

byte5	byte6	byte7	byte8	byte9
Control Type	Data Length	Reserved	Reserved	Reply Switch

Byte5 controlling type is 4.

The data length of byte6 is 1.

When byte9 is 1, it means turning on central control replies switch; and 0 means turning off replies.

5. Dante Routing (0x6e)

Data section formats are:

byte5	byte6	byte7	byte8	byte9-24	byte25-40
Dante Channel No.	Routing Switch	Reserved	Reserved	Subscribed channel name	Subscribed device name

Byte5 Dante channel numbers. The difference is, Dante channel numbers start from 1.

Byte6 Dante channel, subscribe/cancel subscription of appointed channels of Dante devices indicated with byte25-49. The appointed channels are indicated with byte9-24 channel names.

5.3 Serial Port-to-UDP (RS232 To UDP)

DSP devices support RS232 translating into UDP. The protocol formats are as follows.

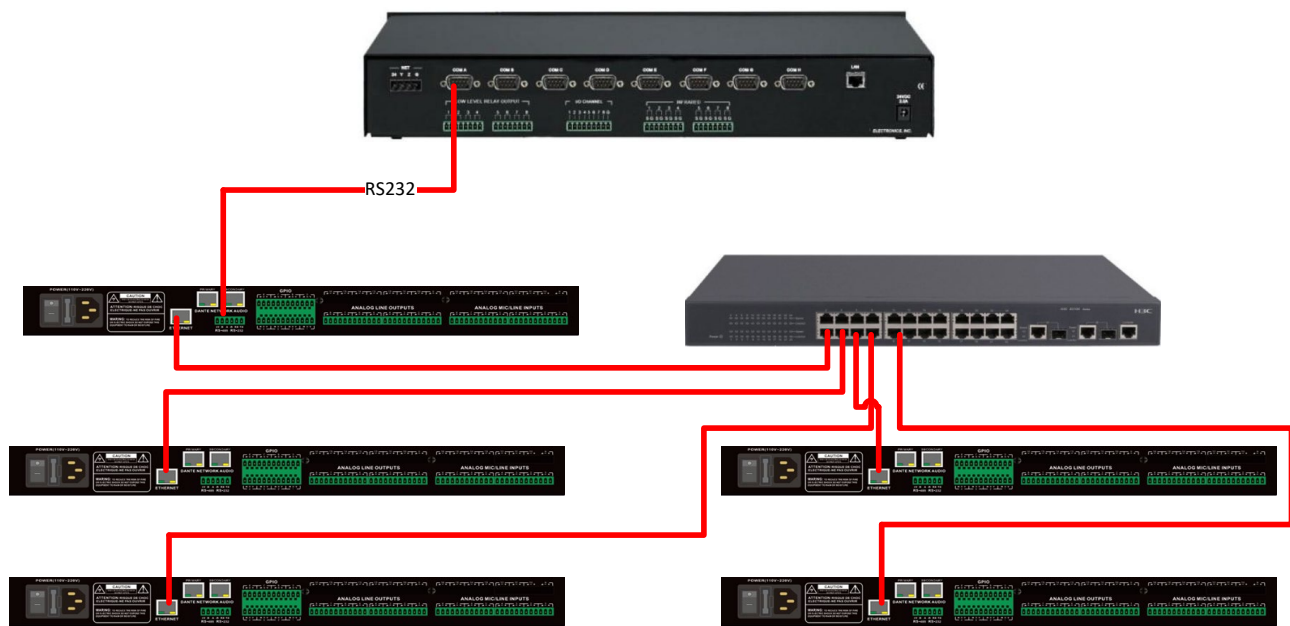
4bytes prefix	4bytes	2bytes	1byte	1byte	128bytes
UDP:	IP Address	Port	Data Length	Reserved	Data

After receiving the protocol format data packet, RS232 send data in the protocol to appointed IP addresses and devices at ports.

For example, when sending data " HELLO DSP to device port 50000 of device " 192.168.10.22" , the protocol commands are as follows.

4 bytes prefix	4 bytes	2 bytes	1 byte	1 byte	128 bytes
0x3a504455 (' :PDU')	0x1610A8C0	0xC350	0x09	0x00	" HELLO DSP"

Application scenario: The function can be applied in scenarios when many central control hosts have no network port. As shown in the figure, central control hosts translate network commands through serial ports to control any network device.



5.4 ASCII control command

Preface:

Concentrate:

1, because the starting bit of the channel is 0, it can be understood that the 0-3 channel corresponds to the IN1-4 software display channel, and 0-3 is only for example. The actual number of channels shall prevail the device model.

2. In the function on/off settings, 1 is open and 0 is off; For example, set:output#mute#0-3#1. The last 1 stands for "turn on mute".

5.4.1 Input volume control and acquisition

set:input#gain#0-3#1

(Setting: Enter #Gain #Channel Number # to 1dB)

get:input#gain#0-3 --> get:input#gain#0-3#1#1#1#1

(Get: Input #Gain#Channel Number) Return Information Example: get:input#gain#0-3#1#1#1#1#1(Input 1-4 channel volume is 1/1/1/1 in turn)

5.4.2 Output volume control and acquisition

set:output#gain#0-3#1

get:output#gain#0-3 --> get:output#gain#0-3#1#1#1#1#1

5.4.3 Phantom power supply control and acquisition

set:input#phant#0-3#1

get:input#phant#0-3 --> get:input#phant#0-3#1#1#1#1#1

5.4.4 Input Mute Control and Acquisition

set:input#mute#0-3#1

get:input#mute#0-3 --> get:input#mute#0-3#1#1#1#1#1

5.4.5 Output mute control and acquisition

set:output#mute#0-3#1

get:output#mute#0-3 -> get:output#mute#0-3#1#1#1#1#1

5.4.6 Control and acquisition of sensitivity

set:input#sens#0-3#1 (for 3db, for second gear)

get:input#sens#0-3 -> get:input#sens#0-3#1#1#1#1#1

5.4.7 Control acquisition of matrices: multiple inputs control a single output, and one input controls multiple outputs

set:mixer#switch#0#0-3#1 (set input 1 to output 1 and 4 routes to open)

set:mixer#switch#0-3#0#1 (set input 1 to 4 to output 1 route open)

set:mixer#gain#0-3#0#1 (set input 1 to 4 to output 1 route gain 1db)

get:mixer#switch#0-3#0(0-3:input,0:output) -> get:mixer#switch#0-3#0#1#0#1#1

5.4.8 Invocation and Saving of Scenarios

scene: toggle #3 (scene call, pc is displayed as scene 4)

scene:save#3 (save scene)

5.4.9 Input level acquisition

get:input#level#0-3 -> get:input#level#0-3 #-105.4#-102.5#-105.2#-104.8(dbfs)

5.4.10 Output level acquisition

get:output#level#0-3 -> get:output#level#0-3#-56.0#-40.8#-43.6#-46.4

5.4.11 System Mute Control and Acquisition

set:sysctl#mute#1 (on system mute)

get:sysctl#mute -> get:sysctl#mute#1

5.4.12 Settings and getting the channel name of the input and output

set:input#name#0#1

get:input#name#0-3 -> get:input#name#0-3#IN1#IN2#IN3#IN4

5.4.13 Input and output inverting control and acquisition

set:input#phase#0-3#1

set:output#phase#0-3#1

get:input#phase#0-3 -> get:input#phase#0-3#1#1#1#1

get:output#phase#0-3 -> get:output#phase#0-3#1#1#1#1

5.4.14 Input and output step control and acquisition

set:input#step#0-3#10

set:output#step#0-3#10

5.4.15 Input and output link control and acquisition

set:input#link#0-3#1

set:output#link#0-3#1

get:input#link#0-3 -> get:input#link#0-3#1#1#1#1

get:output#link#0-3 -> get:output#link#0-3#1#1#1#1

5.4.16 Control and acquisition of signal generators

set:input#type#0-3#1

get:input#type#0-3 -> get:input#type#0-3#1#1#1#1

5.4.17 Restore factory settings control

set:refactory

5.4.18 Scene Reset Control

set:rescene

5.4.19 Get the name of the scene that is set arbitrarily

set:scene#name#0-3#pre1 (PC only supports UTF-8 encoding, Chinese ANSI code will become garbled)

get:scene#name#0-3 (scene number: 0-15) -> get:scene#name#0-3#pre1#pre1#pre1#pre1#pre1

5.4.20 Module name:input,output|mixer

item 名:

(input)mute,gain,sens,phant,type,freq,name,phase,step,link,level

(output)mute,gain,name,step,link,level

(mixer)switch,gain

(scene)toggle,save,name

(sysctl)mute

(rescene)

(refactory)

5.4.21 Format the instruction:

set: Module name #itemname #Start Channel-End Channel # Parameter value

For example

set:input#mute#0-3#0/1

5.4.22 Get the instruction format

get: Module name #itemname #Start Channel- End Channel

For example

get:input#mute#0-3