4.4 Remote control interfaces

4.4.1 Network interface

Many of unit functions could be controlled using different protocols based on network connection. The unit's network controller supports Static mode for IP protocol. The IP settings could be changed using Control Software.

Default TCP/IP settings

Parameter	Value
Mode	Static
Address	169.254.10.227
Mask	255.255.0.0
Gateway	169.254.10.1
UDP Port	50000

Unit could be controlled via UDP connection using API. Before attempting to use UDP control, please ensure that both the unit and the PC are connected to the same network. Use IP address of the unit and the UDP port number, the default port is 50000.

4.4.2. RS232 and RS485 interfaces

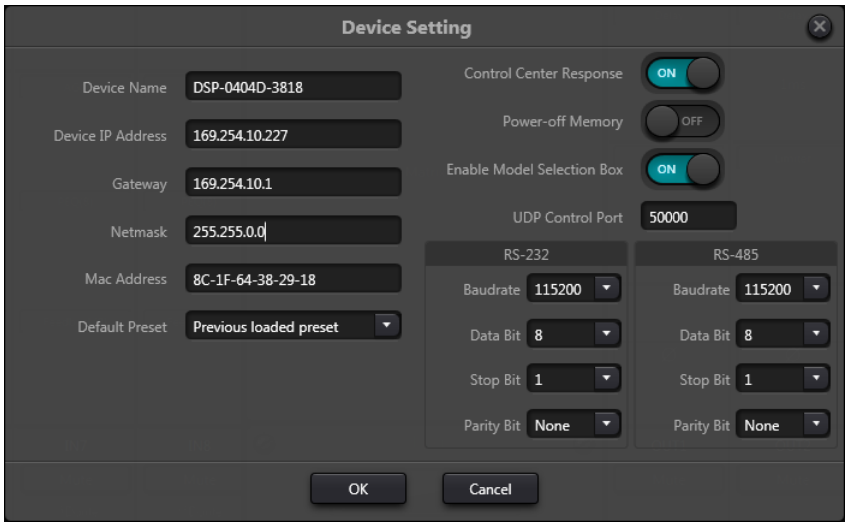
The RS232 and RS485 ports allows to access unit's functions using its API. It also ensures possibility to easy control from a third party control system. When sending several commands please keep at least 100 ms interval between them.

Default RS232 and RS485 ports settings

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

4.4.3. Changing control interface settings

Please use Aberman aDSP Audio System Control Software to set IP, RS232/485 and other settings of the unit according your needs. Click on **Settings** it the top menu and select **Device Settings** item. If you need to have a feedback from the unit please turn on the **Control Center Response** switch.



5. API Protocol

Advanced Programming Interface ensures possibilities of controlling the unit by an external third party control system using IP or RS232/RS485 connection and command line. Available commands and their parameters depend on the API versions which could be different between different versions of firmware. The control command should be sent in ASCII or HEX codes and their structure should be followed the next format guidance.

Please keep in mind that numbering starts from 0. For example, the channel 1 number is 0, the channel number 2 is 1 and etc. When the value for an element's parameter is set to 1 this means this parameter is switched on, while 0 value means that parameter is switched off.

5.1 ASCII format API

The ASCII API is pretty simple and easy to use. There are only two types of commands:
set – to change an element's parameter
get – to request an element's parameter

After command type add colon symbol '**:**' then '**input**', '**output**', or other system component to control. After that element name and parameters divided by sharp symbol '**#**'. Please find details of the available names below:
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

Input volume control and request
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)

Output volume control and request
set:output#gain#0-3#1
get:output#gain#0-3 --> get:output#gain#0-3#1#1#1#1

Phantom power supply control and request
set:input#phant#0-3#1
get:input#phant#0-3 --> get:input#phant#0-3#1#1#1#1

Input Mute Control and Request
set:input#mute#0-3#1
get:input#mute#0-3 --> get:input#mute#0-3#1#1#1#1

Output mute control and request
set:output#mute#0-3#1
get:output#mute#0-3 -> get:output#mute#0-3#1#1#1#1

Control and request 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

Control request 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

Invocation and Saving of Presets (Scenarios)
scene: toggle #3 (scene call, pc is displayed as scene 4)
scene:save#3 (save scene)

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

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

System Mute Control and Request
set:sysctl#mute#1 (on system mute)
get:sysctl#mute -> get:sysctl#mute#1

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

Input and output inverting control and request
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

Input and output step control and request
set:input#step#0-3#10
set:output#step#0-3#10

Input and output link control and request
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

Control and request of signal generators
set:input#type#0-3#1
get:input#type#0-3 -> get:input#type#0-3#1#1#1#1

Restore factory settings control
set:refactory

Scene Reset Control
set:rescene

Get the name of the scene that is set arbitrarily
set:scene#name#0-3#pre1 (PC only supports UTF-8 encoding, Chinese ANSII code will become garbled)
get:scene#name#0-3 (scene number: 0-15) ->
get:scene#name#03#pre1#pre1#pre1#pre1#pre1

5.2 HEX format API version V1

At the moment, there are two versions of the HEX API to support old and new generation of aDSP series units. Both API versions could be used at the same time. The latest Control Protocol (V2) adopts commands with variable-length. This means that one command could be used to control several parameters at once. The old version (V1) with fixed length of the command could control only one parameter per time. To indicate which one of API version is being used the fourth byte (**Version**) should be set to 0 for V1 and to 1 for V2.

For the V1 version the full length of command is fixed to 12 bytes.

Control Command Format

Start	Type	Length	Version	Module ID	Parameter Type	Parameter 1	Parameter 2
1 byte	1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	2 byte
0xB3	…	0x00	0x00	…	…	…	…

Start - each command always starts from **0xB3** byte

Type – three types of command are available for V1:

0x13 – switching preset (scenarios), set number in Module ID (0 ~ 15), 0 for other bytes
0x21 – parameter control (set), fill **Module ID** and **Parameters** Bytes to control an element **0x22** – parameter request (get), the returned values will be in **Parameter 1** and **Parameter 2** bytes

Length – always set to 0, not used for V1

Version – always set to 0 for V1

Module ID – describes an DSP element, refer to unit's configuration

Parameter Type – describes parameter of the DSP element, refer to unit's configuration

Parameter 1 – value of the parameter, valid when parameter has only one control variable

Parameter 2 – value of the parameter, valid when parameter has two variables to control

Note: there is one exception! The Matrix routing has three variables – input channel, output channel and routing switch state. The **first byte** of **Parameter 1** is used for input channel, the **second byte** is for output channel and **Parameter 2** is used for routing switch state description.

5.3 HEX format API version V2

The V2 version is much more complex that is why we do recommend using automatic command generation function embedded into the Aberman aDSP Audio System Control Software.
Click on **Help** in the top menu, and then select **Command Capture…** menu item in the drop list. The Command Capture window will show command for each function that will be used in the Control Software. The Command Source field will show control element description for easy debugging.

