Weiver/WeiverPlayer API Manual

[Introduction]

Three methods to control Weiver/WeiverPlayer

Method 1. In case of WeiverPlayer, control by key pad

Method 2. Control through WeiverEx program

Method 3. Control by using API

Method 1 & Method 2 are generally used.

< Method 3 >

- Method 3 uses UDP packets to control the command.
- Method 3 (using API) is available on Weiver 1.0, Weiver Player 1.0, Weiver 2.0, Weiver Player 2.0
- Only playback function is supported, but recording function is not supported.
- HD Radio™ is only available on Weiver 2.0, Weiver Player 2.0.

[Weiver/WeiverPlayer API operating process]

- WeiverEx software is controlled using UDP packets by IP connections with the Weiver/WeiverPlayer software.

- API is for running Weiver/WeiverPlayer without the WeiverEx program, so API example program and packet information are open to the public.

Reference source can be downloaded from Lumantek website.

The following is Weiver/WeiverPlayer API packet structure. (Refer to 'WvPlayerRemoteAPI.h')

```
#define  WEIVER_PLAYER_ID                    0x057b50b7
#define  WEIVER_PLAYER_API_RX_PORT           50504

#define  REMOTE_STRING_SUCCESS _T("Received")
#define  REMOTE_STRING_INVALIDE_PARAM _T("Invalid Param")

typedef enum _WvPlayerRemoteCMD_e
{
        WV_PLAYER_REMOTE_CMD_SET_STRING,
        WV_PLAYER_REMOTE_CMD_MAX
}WvPlayerRemoteCMD_t;

typedef struct _WvPlayerRemotePacket_t
{
        unsigned int u4_ID;// WEIVER_PLAYER_ID
        unsigned int u4_Reserved;//WV_PLAYER_REMOTE_CMD_SET_STRING
        unsigned int u4_DataSize;
        unsigned char u1a_Data[1000];
}WvPlayerRemotePacket_t, *WvPlayerRemotePacket_tp;
```

UDP Port : 50504 is left open to support API in Weiver/Weiver Player. So when you receive data from the WvPlayerRemotePacket structure, Weiver/WeiverPlayer performs the command.

The released source is an example of a Window program. (Applicable to other OSs)

The Example Program (WeiverPlayerRemoteAPI.exe) makes five assumptions prior to explaining command list.

Assumption 1. IP of Weiver/WeiverPlayer is 192.168.100.1
Assumption 2. The port number opened on Weiver/Weiver Player for API support is 50504. (Fixed)
Assumption 3. The port number open for UDP reception is 50000.(Variable)
Assumption 4. Each character byte has a size of 2 bytes.
Assumption 5. Time Out occurs when a response is not received after UDP transmission.

[Data Packet Structure]

connect 50000

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	26	L"connect 50000"
Size(Byte)	4	4	4	13x2

disconnect

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	20	L"disconnect"
Size(Byte)	4	4	4	10x2

rescandisk

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	20	L"rescandisk"
Size(Byte)	4	4	4	10x2

openfile test.wpj

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	34	L"openfile test.wpj"
Size(Byte)	4	4	4	17x2

get openfile

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get openfile"
Size(Byte)	4	4	4	12x2

play start

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	20	L"play start"
Size(Byte)	4	4	4	10x2

play stop

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	18	L"play stop"
Size(Byte)	4	4	4	9x2

play pause

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	20	L"play pause"
Size(Byte)	4	4	4	10x2

play resume

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	22	L" play resume"
Size(Byte)	4	4	4	11x2

get play_status

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	30	L"get play_status"
Size(Byte)	4	4	4	15x2

set freq 400000000

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	36	L"set freq 400000000"
Size(Byte)	4	4	4	18x2

set freq 400000000 hz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	42	L"set freq 400000000 hz"
Size(Byte)	4	4	4	21x2

set freq 400000 khz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	38	L"set freq 400000 khz"
Size(Byte)	4	4	4	19x2

set freq 400 mhz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	32	L"set freq 400 mhz"
Size(Byte)	4	4	4	16x2

get freq

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	16	L"get freq"
Size(Byte)	4	4	4	8x2

get gain_min

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get gain_min"
Size(Byte)	4	4	4	12x2

get gain_max

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get gain_max"
Size(Byte)	4	4	4	12x2

get gain

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	16	L"get gain"
Size(Byte)	4	4	4	8x2

get rf_out_level 1

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	36	L"get rf_out_level 1"
Size(Byte)	4	4	4	18x2

set gain 100

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"set gain 100"
Size(Byte)	4	4	4	12x2

set gain -100

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	26	L"set gain -100"
Size(Byte)	4	4	4	13x2

get play_time_total

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	38	L"get play_time_total"
Size(Byte)	4	4	4	19x2

get play_time_current

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	42	L"get play_time_current"
Size(Byte)	4	4	4	21x2

set play_time_current 5

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	46	L"set play_time_current 5"
Size(Byte)	4	4	4	23x2

get play_section

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	32	L"get play_section"
Size(Byte)	4	4	4	16x2

set play_section 10 20

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	44	L"set play_section 10 20"
Size(Byte)	4	4	4	22x2

get spec_inv

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get spec_inv"
Size(Byte)	4	4	4	12x2

set spec_inv off

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	32	L"set spec_inv off"
Size(Byte)	4	4	4	16x2

set spec_inv on

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	30	L"set spec_inv on"
Size(Byte)	4	4	4	15x2

get sample_rate

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	30	L"get sample_rate"
Size(Byte)	4	4	4	15x2

set sample_rate 10000000

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	48	L"set sample_rate 10000000"
Size(Byte)	4	4	4	24x2

set sample_rate 10000000 hz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	54	L"set sample_rate 10000000 hz"
Size(Byte)	4	4	4	27x2

set sample_rate 10 mhz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	44	L"set sample_rate 10 mhz"
Size(Byte)	4	4	4	22x2

set sample_rate 10000 khz

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	54	L"set sample_rate 10000 khz"
Size(Byte)	4	4	4	27x2

set trigger_out_level low

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	50	L"set trigger_out_level low"
Size(Byte)	4	4	4	25x2

set trigger_out_level high

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	52	L"set trigger_out_level low"
Size(Byte)	4	4	4	26x2

get trigger_out_level

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	42	L"get trigger_out_level"
Size(Byte)	4	4	4	21x2

set extclk off

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	28	L"set extclk off"
Size(Byte)	4	4	4	14x2

set extclk on

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	26	L"set extclk on"
Size(Byte)	4	4	4	13x2

get extclk

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	20	L"get extclk"
Size(Byte)	4	4	4	10x2

get trigger_in

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	28	L"get trigger_in"
Size(Byte)	4	4	4	14x2

set trigger_in off

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	36	L"set trigger_in off"
Size(Byte)	4	4	4	18x2

set trigger_in on

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	34	L"set trigger_in on"
Size(Byte)	4	4	4	17x2

set current_directory d:₩

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	50	L"set current_directory d:₩"
Size(Byte)	4	4	4	25x2

set current_directory d:₩WeiverData

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	70	L"set current_directory d:₩WeiverData"
Size(Byte)	4	4	4	35x2

get current_directory

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	42	L"get current_directory"
Size(Byte)	4	4	4	21x2

get file_count

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	28	L"get file_count"
Size(Byte)	4	4	4	14x2

get filename 0

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	28	L"get filename 0"
Size(Byte)	4	4	4	14x2

get folder_count

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	32	L"get folder_count"
Size(Byte)	4	4	4	16x2

get foldername 0

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	32	L"get foldername 0"
Size(Byte)	4	4	4	16x2

set power -200

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	28	L"set power -200"
Size(Byte)	4	4	4	14x2

get power

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	18	L"get power"
Size(Byte)	4	4	4	9x2

get power_min

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	26	L"get power_min"
Size(Byte)	4	4	4	13x2

get power_max

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	26	L"get power_max"
Size(Byte)	4	4	4	13x2

get freq_min

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get freq_min"
Size(Byte)	4	4	4	12x2

get freq_max

Type	ID	Reserved	Data Size	Data
Data	0x057b50b7	0	24	L"get freq_max"
Size(Byte)	4	4	4	12x2

Command Description

Command	connect <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command for setting a connection with the equipment. <parameter> optional port number - For example, the connect 50000 command sends UDP packets to the 50000 port number when the Weiver appliance sends a response. - If the equipment is under control with WeiverEx or HD Radio™ S/W, the connect command fails. - If the connect command fails, send the disconnect command and then send the connect command. - You have to designate the optional incoming port number that is left open.

Command	disconnect
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to disconnect the equipment. - RF transmission will automatically stop when RF transmission is in progress.

Command	rescandisk
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to update the disk drive connected to the equipment.

Command	openfile <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to set the file to play on the equipment. <parameter> optional file name - Go to the folder that contains the applicable file prior to executing openfile command. - When it comes to command for moving folders, refer to 'set current directory' - openfile command will fail if it is in play. - Refer to the command for play to stop playing. - Weiver file is automatically set to the center frequency and power at the time of recording. - For Weiver file, refer to 'set freq' command and 'set gain' command for adjusting central frequency and power. - HD Radio™ FM file is automatically set to 87.9 MHz and the output power is automatically set to the original power of the file. - HD Radio™ AM file is automatically set to 830KHz and the output power is automatically set to the original power of the file. - When it comes to adjusting central frequency and power of HD Radio™ files, refer to the 'set freq' command and the 'set power' command.

Command	get openfile
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the name of the file opened

Command	play start play stop play pause play resume
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	play start : Start RF transmission. play stop : Stop RF transmission. play pause : Pause RF transmission. play resume : Resume RF transmission. Make it to use play related commands after ‘openfile’.

Command	get play_status
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the play status of equipment

Command	set freq <parameter> set freq <parameter> hz set freq <parameter> khz set freq <parameter> mhz
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to set the frequency of the equipment <parameter> optional frequency - 'openfile' command automatically sets the sample/center frequency of the file. So you don't have to change them. - For HD Radio™, FM files are automatically set to 87.9 MHz and AM files to 830 kHz. Set the frequency using 'set freq' command.

Command	get freq
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the frequency of the equipment

Command	set gain <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	Command to adjust the RF transmission power of the equipment <parameter> optional 'gain' value. gain +10.0 dB applied, <parameter> will be 100 gain -12.3 dB applied, <parameter> will be -123. HD Radio™ is not applicable. As for HD Radio™, refer to 'set power' command

Command	get gain get gain_min get gain_max
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	get gain : Restore gain of RF transmission. get gain_min : Restore minimum gain of RF transmission. get gain_max : Restore maximum gain of RF transmission. If restoration value is '-123', it is '-12.3 dB'. HD Radio™ file is not applicable. As for HD Radio™ file, refer to 'get power' command

Command	get rf_out_level <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	Command to restore the RF output power. <parameter> optional time value(second). Time value should be less than total time. If restoration value is -123, it is -12.3 dBm. As for total time of the file, refer to 'get play_time_total' command. HD Radio™ is not applicable. As for HD Radio™ file, refer to 'get power' command.

Command	set play_time_current <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to set the starting position of the file to play in seconds. <parameter> optional time value(second).

Command	get play_time_total get play_time_current
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	get play_time_total : Restore the size of the file currently playing in seconds. get play_time_current : Restore the position of the file currently playing in seconds.

Command	set play_section <parameter1> <parameter2>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	command to set the interval repetition <parameter1> Optional time value(second) to start the interval repetition. <parameter2> Optional time value(second) to stop the interval repetition.

Command	get play_section
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	Command to restore the interval repetition time(second)

Command	set spec_inv off set spec_inv on
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	set spec_inv off : transmission without inverted left and right. set spec_inv on : transmission with inverted left and right.

Command	get spec_inv
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	get spec_inv : Command to restore the value of inversion.

Command	set sample_rate <parameter> set sample_rate <parameter> hz set sample_rate <parameter> khz set sample_rate <parameter> mhz
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	Command to adjust the sample frequency <parameter> sample frequency - 'openfile' command automatically sets the sample/center frequency of the file. So you don't have to change them.

Command	get sample_rate
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : X
Description	Command to restore the sample frequency

Command	set trigger_out_level low set trigger_out_level high get trigger_out_level
Support	Weiver 1.0 : X Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to control the port power of 'Trigger Out' set trigger_out_level low : Low Level(0V) is outputted. set trigger_out_level high : High Level(3.3V) is outputted. get trigger_out_level : Restore the port status value of 'Trigger Out'.

Command	set extclk off set extclk on get extclk
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to set the 10 MHz synchronization of the equipment to internal/external. set extclk off : Use the internal 10 MHz of the equipment set extclk on : Use the external 10 MHz of the equipment get extclk : Restore the 10 MHz synchronization state value

Command	set trigger_in off set trigger_in on get trigger_in
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to set the operation of multiple equipment according to the input level of the 'Trigger In Port'. set trigger_in off : Operate regardless of the input level of 'Trigger In Port'. set trigger_in on : Operate when the input level of the Trigger In Port is High (3.3v). get trigger_in : Restore operation according to synchronization.

Command	set current_directory <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to designate the folder where the files are located. <parameter> The absolute path of the folder. - You have to change the folder before using the 'openfile' command.

Command	get current_directory
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the current folder of the equipment

Command	get file_count
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore Weiver file and the number of HD Radio™ files in the current folder.

Command	get filename <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore Weiver file and HD Radio™ file names of those indexes among the list of the files retrieved from the current folder. <parameter> Designate index in the file list - Index should be smaller than the value restored 'from get file_count'.

Command	get folder_count
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the number of subfolders in the current folder.

Command	get foldername <parameter>
Support	Weiver 1.0 : O Weiver Player 1.0 : O Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	Command to restore the name of the subfolder of the current folder. <parameter> Designate the index of the folder - Should be smaller than the value restored from 'get folder_count'

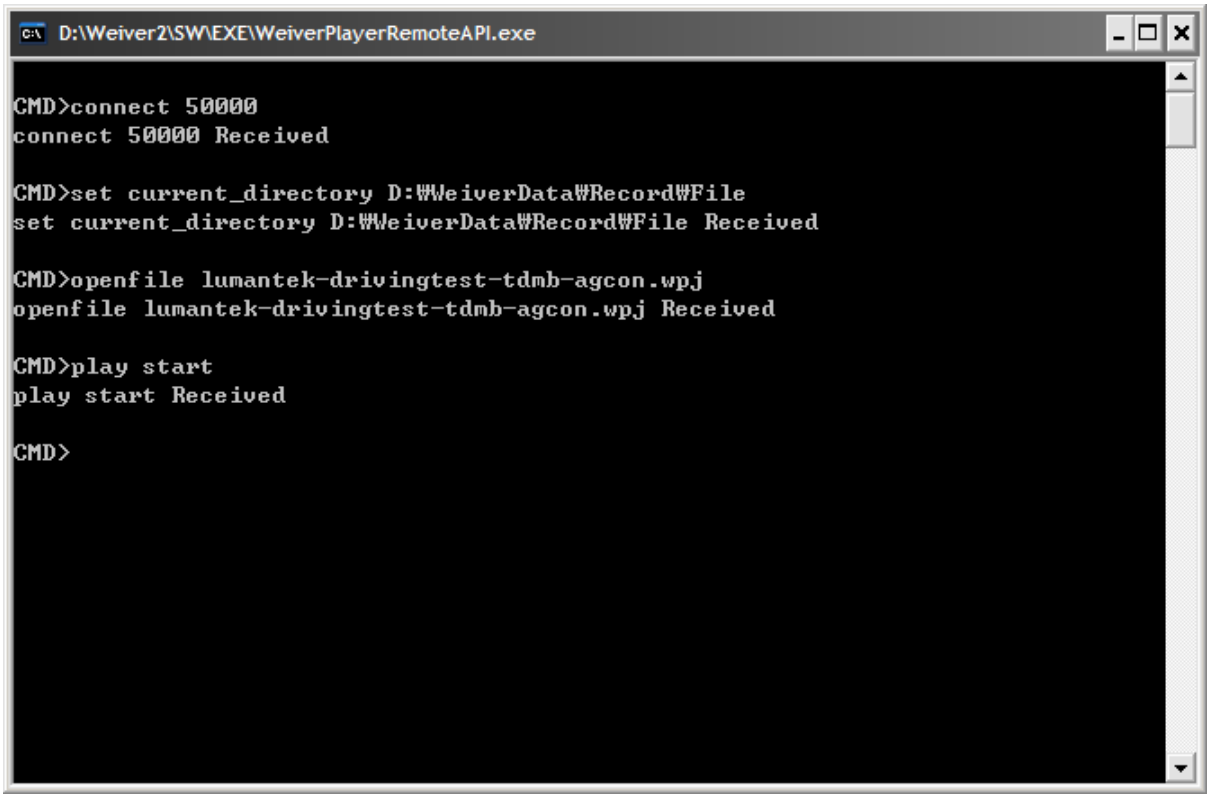
Command	set power <parameter>
Support	Weiver 1.0 : X Weiver Player 1.0 : X Weiver 2.0 : X Weiver Player 2.0 : X HD Radio™ : O
Description	Command to adjust the RF output power of HD Radio™ <parameter> optional power value - power -20.5 dBm applied, <parameter> is -205. - gain -43.2 dBm applied, <parameter>is -432.

Command	get power get power_min get power_max
Support	Weiver 1.0 : X Weiver Player 1.0 : X Weiver 2.0 : X Weiver Player 2.0 : X HD Radio™ : O
Description	get power : Restore the RF output power of HD Radio™ get power_min : Restore the minimum RF output power of HD Radio™ get power_max : Restore the maximum RF output power of HD Radio™ - If restoration value is -205, output power is -20.5 dBm. - If restoration value is -423, output power is -43.2 dBm.

Command	get freq_min get freq_max
Support	Weiver 1.0 : X Weiver Player 1.0 : X Weiver 2.0 : O Weiver Player 2.0 : O HD Radio™ : O
Description	get freq_min : Restore the minimum frequency value of the equipment. get freq_max : Restore the maximum frequency value of the equipment.

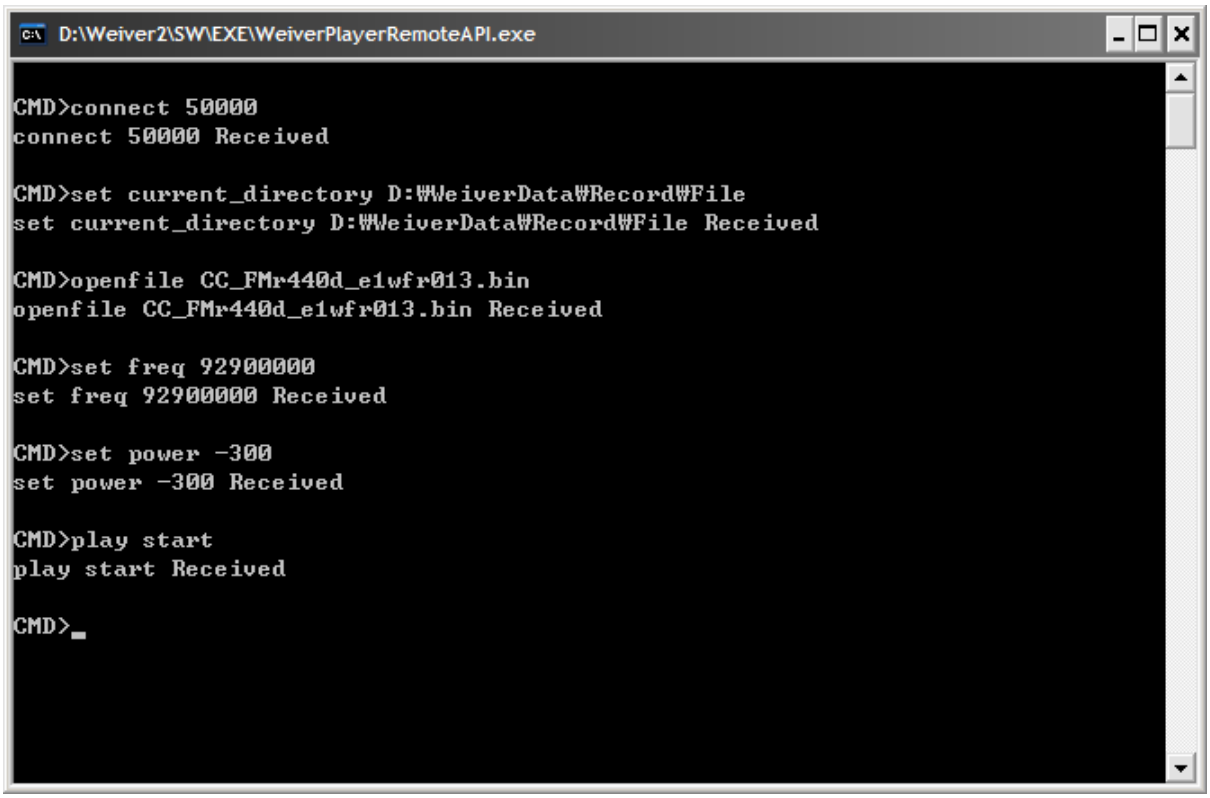
[Example : When playing Weiver file]

connect 50000
set current_directory D:₩WeiverData₩Record₩File
openfile lumantek-drivingtest-tdmb-agcon.wpj
play start



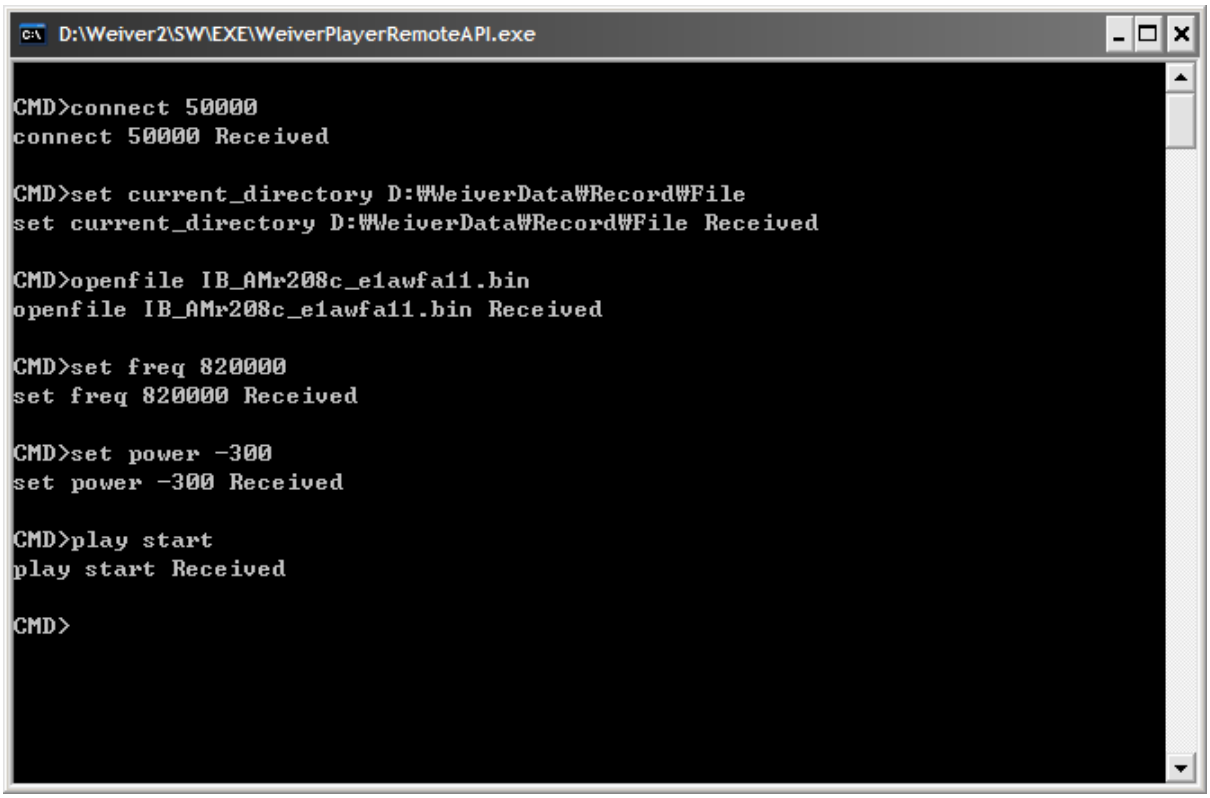
[Example : When playing HD Radio™ FM file]

connect 50000
set current_directory D:₩WeiverData₩Record₩File
openfile CC_FMr440d_e1wfr013.bin
set freq 92900000
set power -300
play start



[Example : When playing HD Radio™ AM file]

connect 50000
set current_directory D:₩WeiverData₩Record₩File
openfile IB_AMr208c_e1awfa11.bin
set freq 820000
set power -300
play start



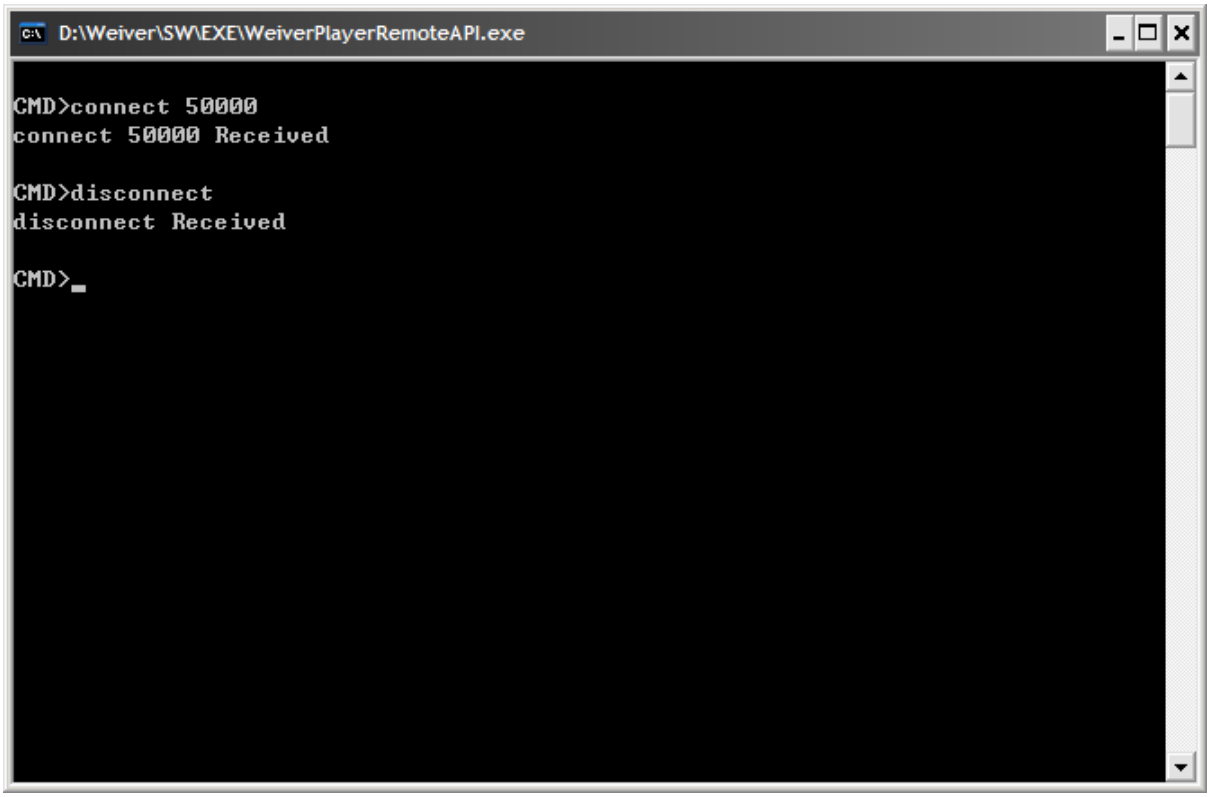
[Example of command use]
connect, disconnect Command

How to use)
"connect" port number : Port number is the number received after sending packet.
disconnect : Disconnects connection.

Explanation)
Answer can be received when UDP port number is sent along "connect" command.
In the program example of UDP port number, "u2_RecvPortFromWeiverPlayer" variable should be set to same value. UDP port can be changed.
Device's receiving UDP port number is set to "50504".

Device stops when "disconnect" command is sent.

Example)
1. connect 50000,
2. disconnect

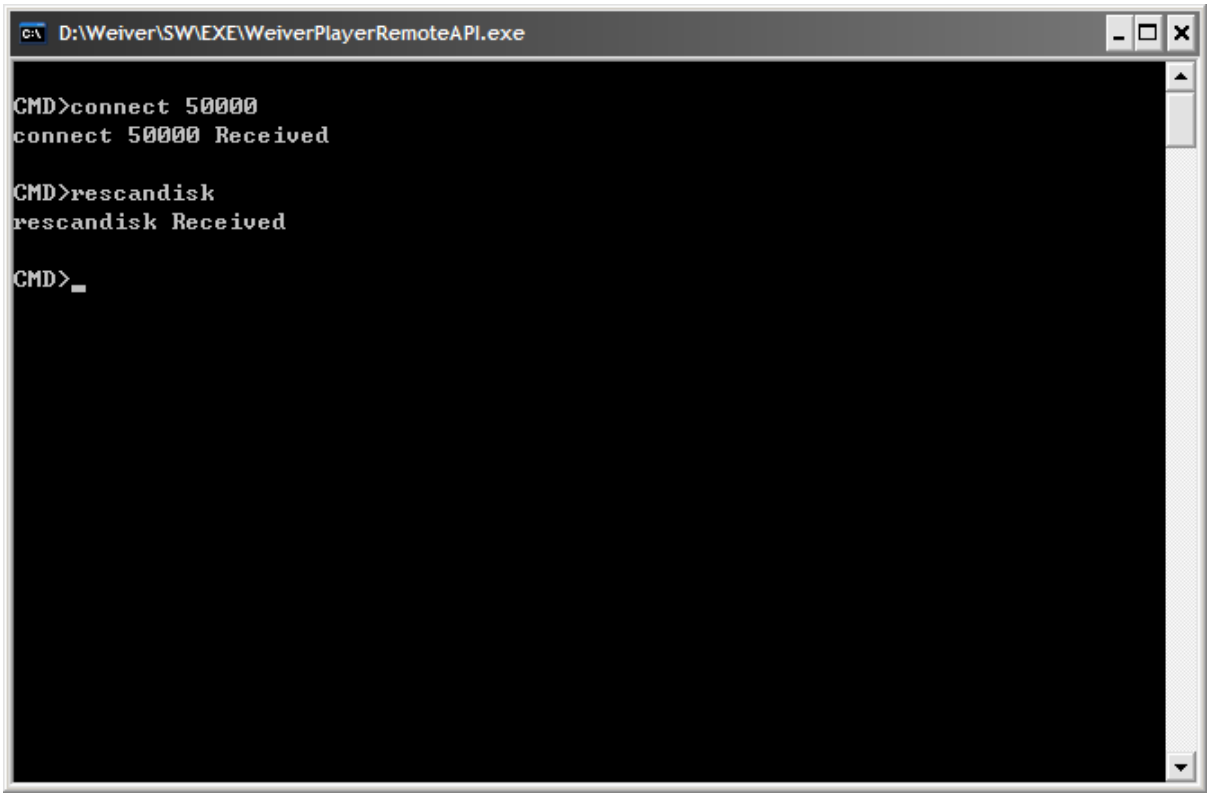


rescandisk Command

How to use)
Disk Drive in the device refreshes when “rescandisk” command is sent.
When disk drive from e-SATA is not visible, try refreshing using “rescandisk” command.

Explanation)
Used when disk drive does not automatically renew.

Example)
1. rescandisk

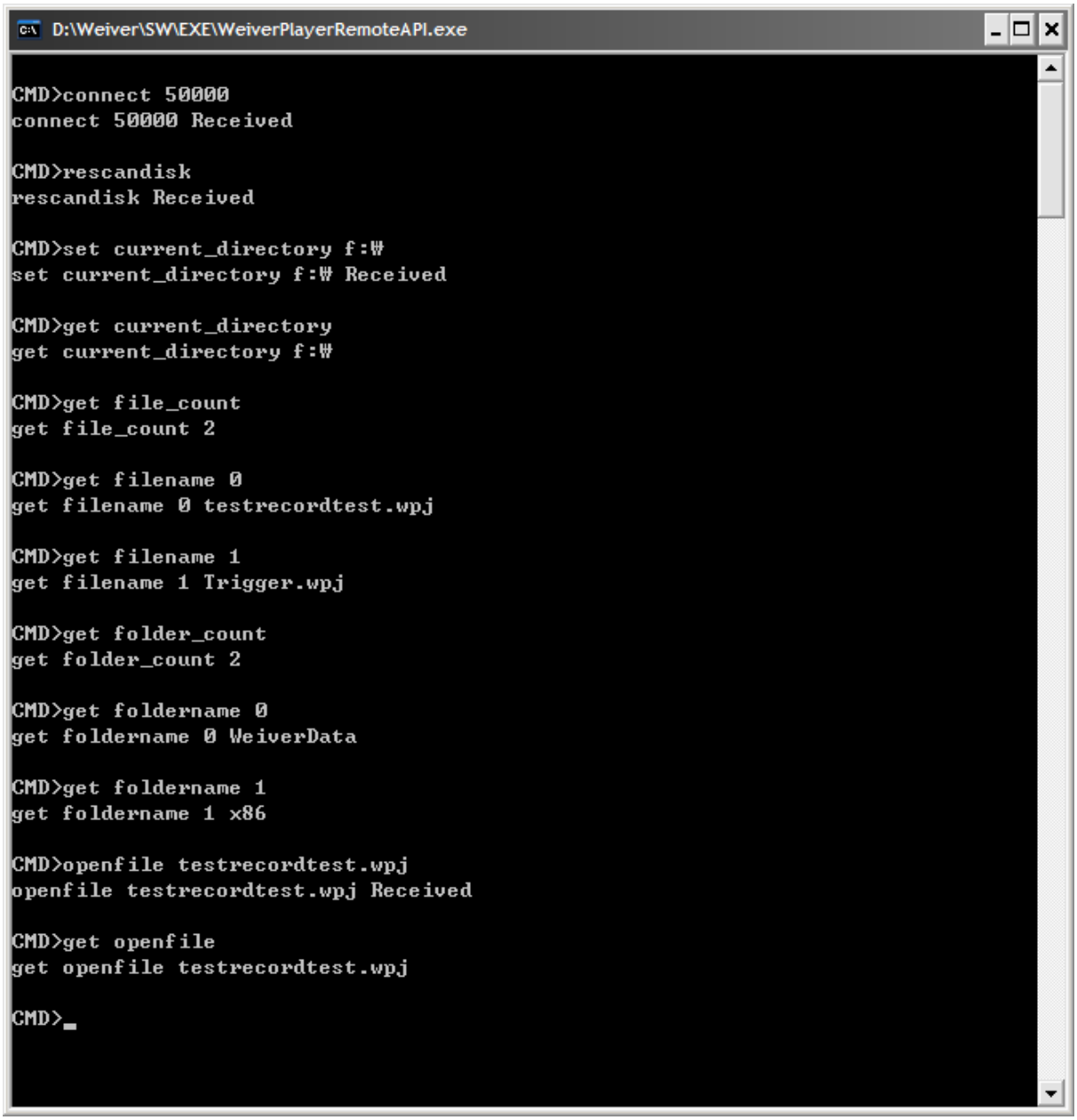


set current_directory, get current_directory Command
get file_count, get filename, get folder_count, get foldername Command
openfile, get openfile Command

How to use)
set current_directory : Directory means disk drive of the device.
get file_count : Prints the number of files in wpj format.
“get filename” index : Index range can have [file quantity -1].
get folder_count : Prints number of folders in “set current_directory”.
“get foldername” index : Index range can have [folder quantity -1].
“openfile” file name : Files in wpj format.

Explanation)
Assign xxx.wpj file that needs to be played.
Play must stop before using “openfile” command.

Example)
1. set current_directory f:₩
2. get current_directory
3. get file_count
4. get filename 0
5. get filename 1
6. get folder_count
7. get foldername 0
8. get foldername 1
9. openfile testrecordtest.wpj
10. get openfile

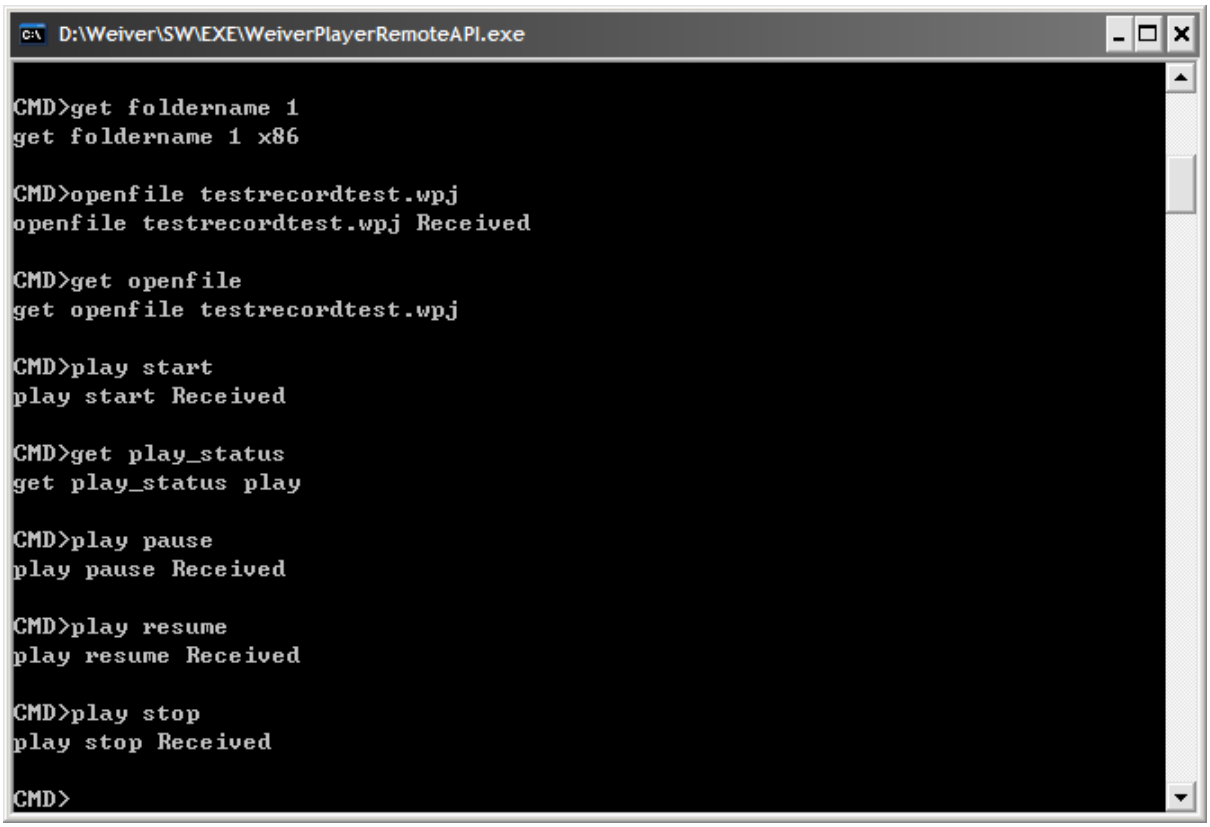

D:\Weiver\SW\EXE\WeiverPlayerRemoteAPI.exe
CMD>connect 50000
connect 50000 Received
CMD>rescandisk
rescandisk Received
CMD>set current_directory f:\
set current_directory f:\ Received
CMD>get current_directory
get current_directory f:\
CMD>get file_count
get file_count 2
CMD>get filename 0
get filename 0 testrecordtest.wpj
CMD>get filename 1
get filename 1 Trigger.wpj
CMD>get folder_count
get folder_count 2
CMD>get foldername 0
get foldername 0 WeiverData
CMD>get foldername 1
get foldername 1 x86
CMD>openfile testrecordtest.wpj
openfile testrecordtest.wpj Received
CMD>get openfile
get openfile testrecordtest.wpj
CMD>_

play start, play stop, play pause, play resume, get play_status Command

How to use)
play start, play stop, play pause, play resume, get play_status

Explanation)
Use “play start, play stop, play pause, play resume” commands after using “openfile” command and selecting file.
“play start” starts play.
“play stop” stops play.
“play pause” pauses play.
“play resume” resumes stopped play.
“get play_status” reads current status.

Example)
1. play start
2. get play_status
2. play pause
3. play resume
4. play stop

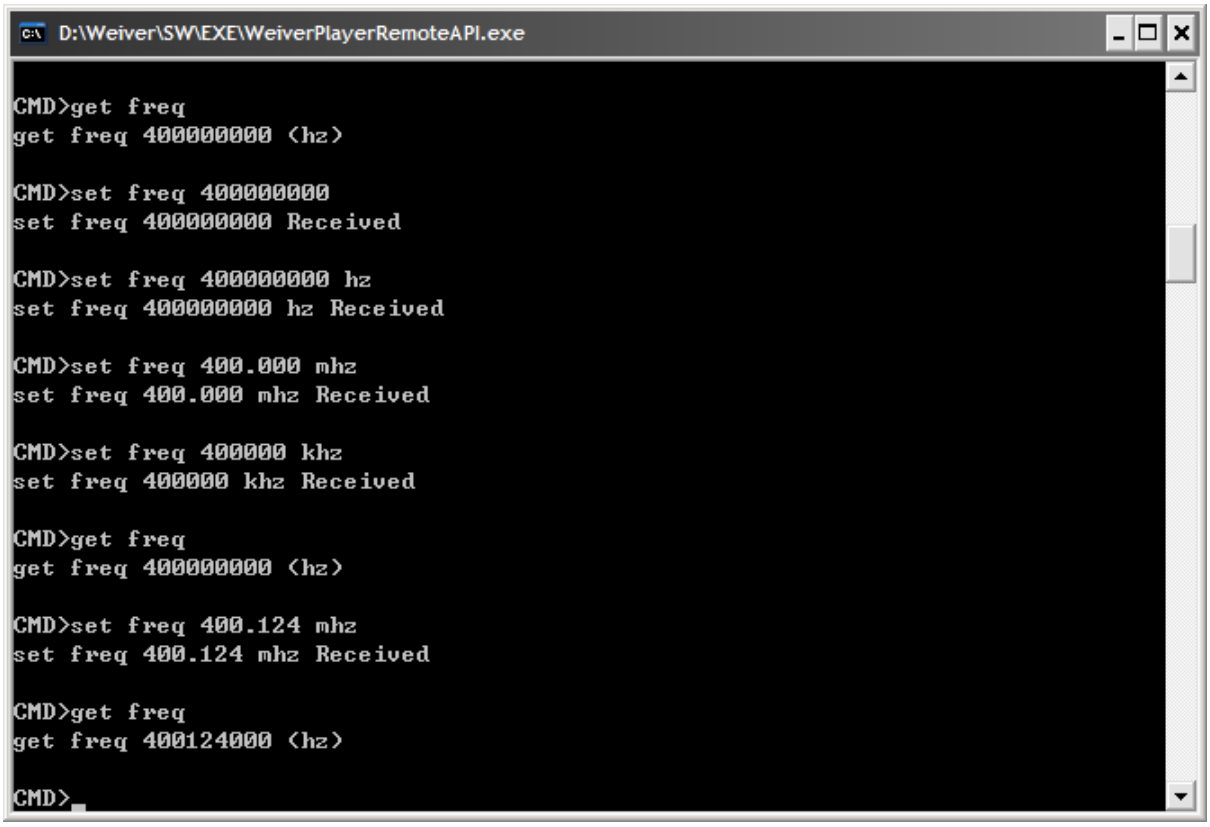


set freq, get freq Command

How to use)
"set freq" frequency
Weiver 1.0 : Frequency range is 50000000(50MHz) ~ 1000000000(1GHz).
Weiver 2.0 : Refer to "get freq_min, get freq_max" commands for frequency.

Explanation)
To change frequency, use "openfile" command to select file and use "set freq" command. Unit of measure is Hz, KHz, MHz.

Example)
1. set freq 400000000 (400MHz set)
2. get freq
3. set freq 400 MHz (400MHz set)
4. set freq 400000 KHz (400MHz set)
5. set freq 400000000 Hz (400MHz set)

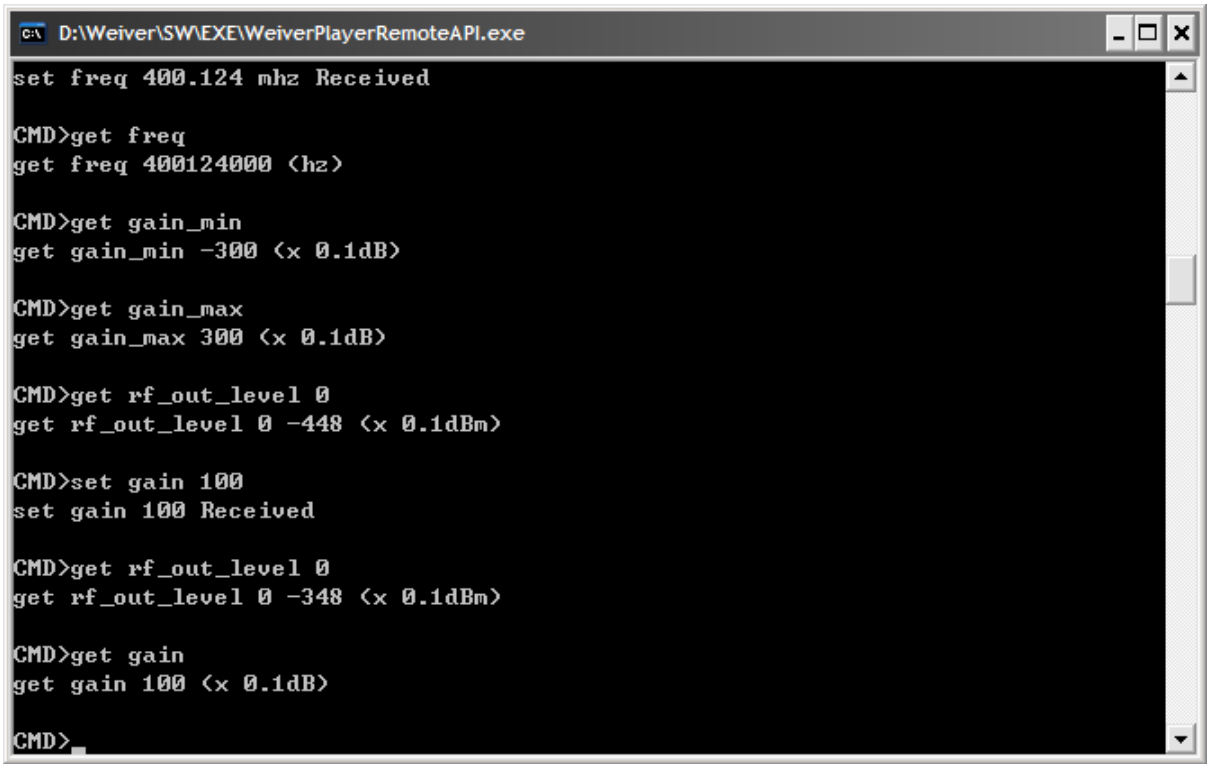


get gain_min, get gain_max, get rf_out_level 0, get gain, set gain Command

How to use)
Used when applying more Gain to play file after using "openfile" command.
Range of Gain is normally -30dB ~ +30dB. After selecting the play file, the range of Gain of the file can be checked using "get gain_min, get gain_max" commands.
"get rf_out_level" time : Range of time is file's total play time.

Explanation)
Unit of measure is 0.1 dB.
For example, "set gain 100" must be inserted to give +10dB Gain.

Example)
1. get gain_min
2. get gain_max
3. get rf_out_level 0
4. set gain 100
5. get rf_out_level 0
6. get gain

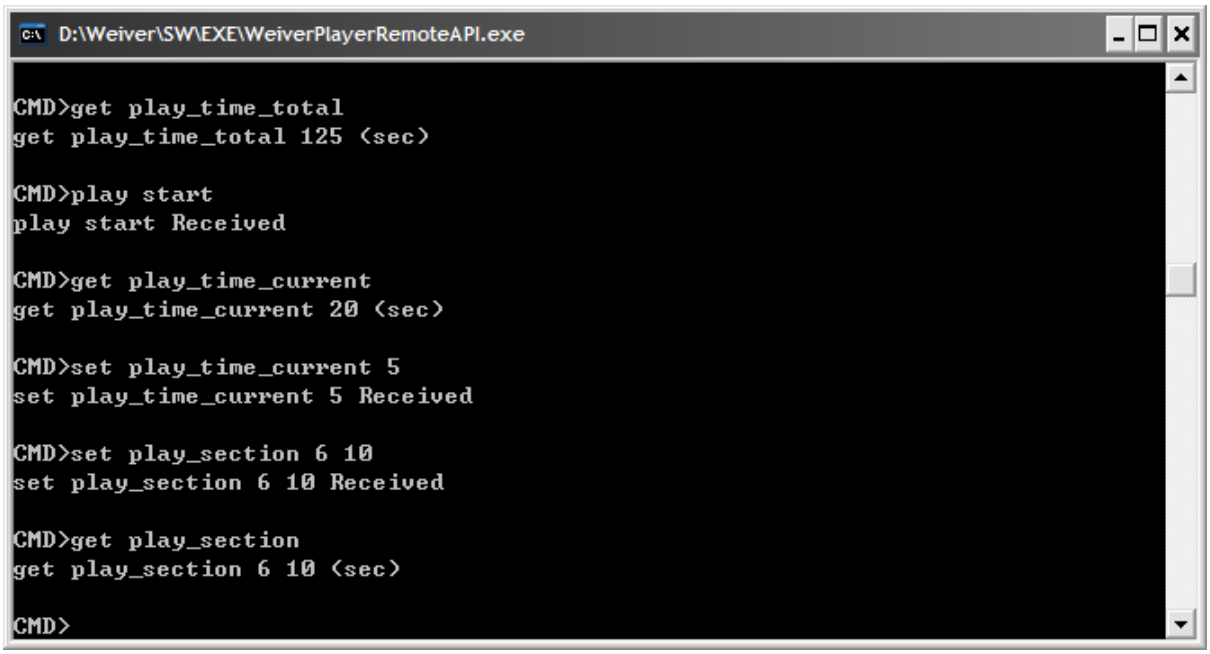


get play_time_total, get play_time_current, set play_time_current,
get play_section, set play_section Command

How to use)
"play time" related command uses "openfile" command to select files. Then the position of the file can be moved and play or get "time" information.
"get play_time_total" command shows total play time in units of seconds.
"set play_time_current" command allows continuous loop play after setting start and end time. Start time must be smaller number than end time, and end time must be smaller than total play time.

Explanation)
Unit of measure is "second."
In this case, file playable time refers to 630 seconds.
"pp 5" moves the file position to "5 seconds point" before playing.
"playrepeat 6 10" loop plays between 6 sec and 10 sec range.

Example)
1. get play_time_total
2. get play_time_current
3. set play_time_current 5
4. set play_section 6 10
5. get play_section

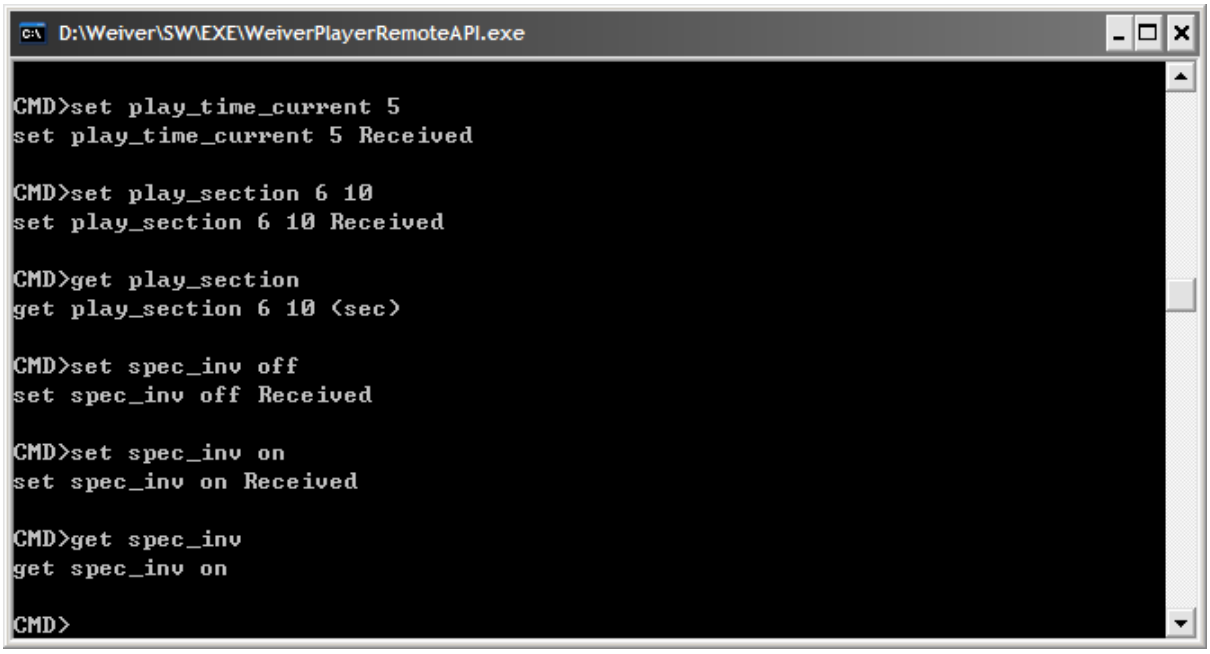


set spec_inv off/on, get spec_inv Command

How to use)
"set spec_inv off" outputs normal RF spectrum of play file.
"set spec_inv on" outputs reverted RF spectrum of play file.

Explanation)
Use "set spec_inv" command in case spectrum must be outputted in reverse.

Example)
1. set spec_inv off
2. set spec_inv on



set sample_rate, get sample_rate Command

How to use)

set sample_rate

Weiver 1.0 : Sample frequency ranges 5000000(5MHz) ~ 30000000(30MHz).

Weiver 2.0 : Sample frequency ranges 1250000(1.25MHz) ~ 70000000(70MHz).

Sample frequency refers to I/Q Rate. In case of BW 8MHz, I/Q Rate is 10MHz. In case of BW 24MHz, I/Q Rate is 30MHz.

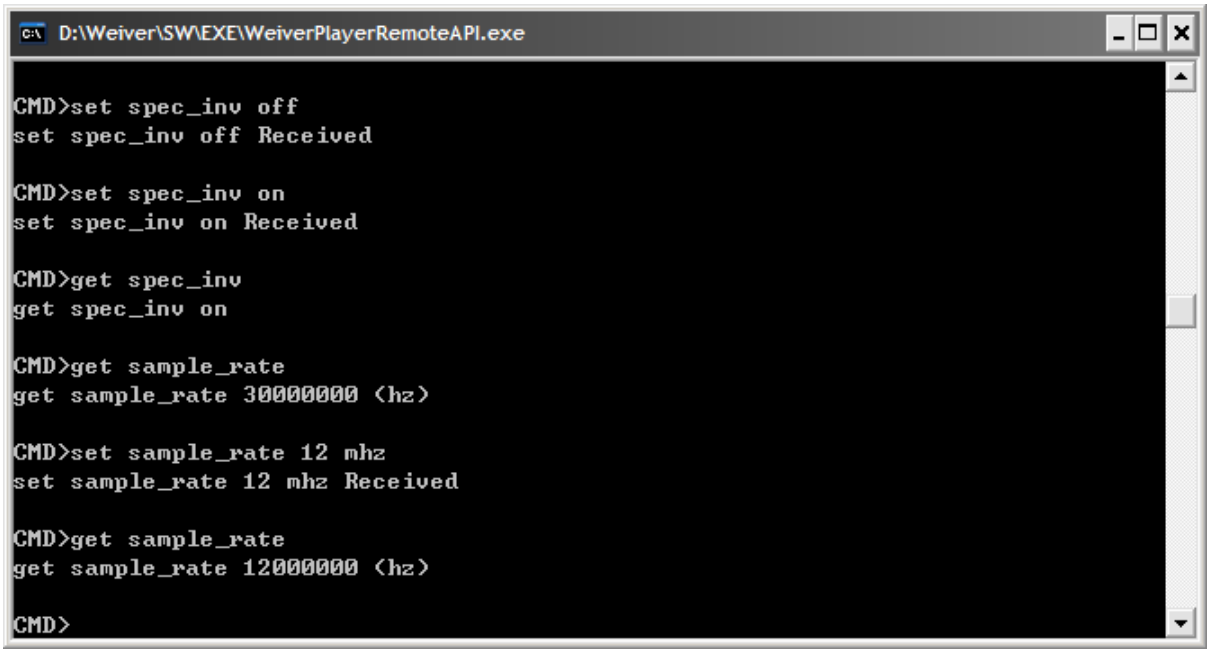
Explanation)

Sample frequency is automatically selected when file is selected using "openfile" command.

Rate can be changed by manually configuring "set sample_rate" command. Unit of measure is "Hz".

Example)

1. set sample_rate 12000000, set sample_rate 12 mhz, set sample_rate 12000 khz
2. get sample_rate

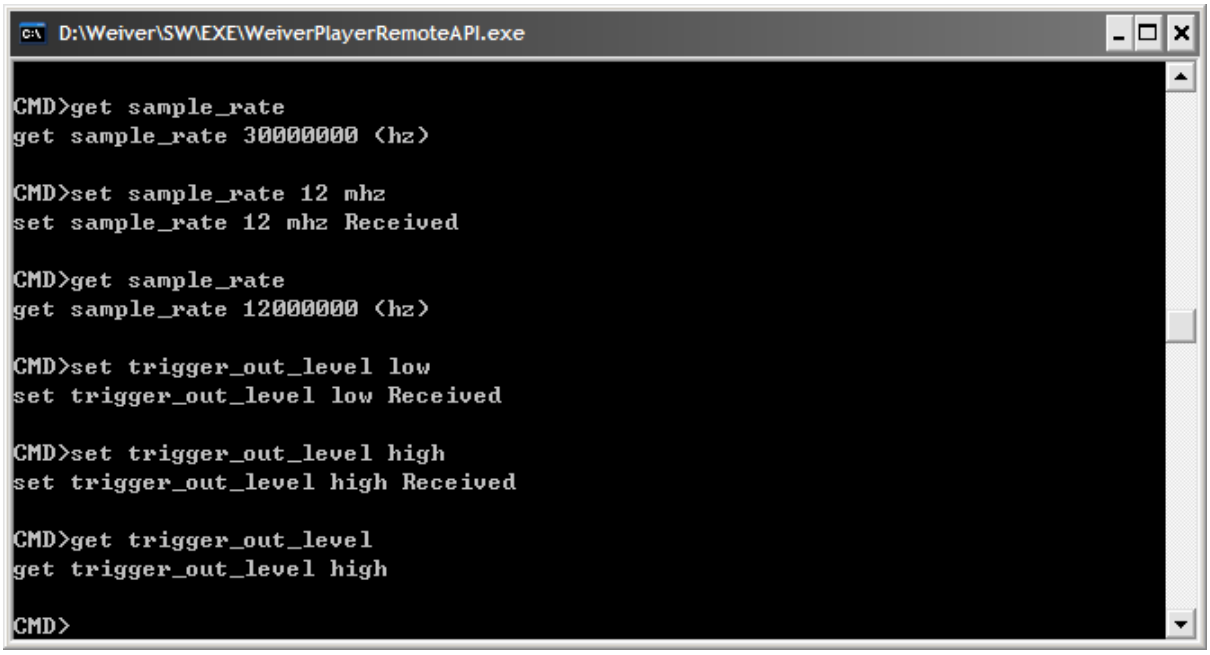


set trigger_out_level, get trigger_out_level Command

How to use)
"Trigger Out" port is available on models WeiverPlayer 1.0, Weiver 2.0, WeiverPlayer.
set trigger_out_level low: LOW Level(0V) signal is outputted from "Trigger Out" port.
set trigger_out_level high: HIGH Level(3.3V) signal is outputted from "Trigger Out" port.

Explanation)
Output level of "Trigger Out" port can be configured.

Example)
1. set trigger_out_level low
2. set trigger_out_level high

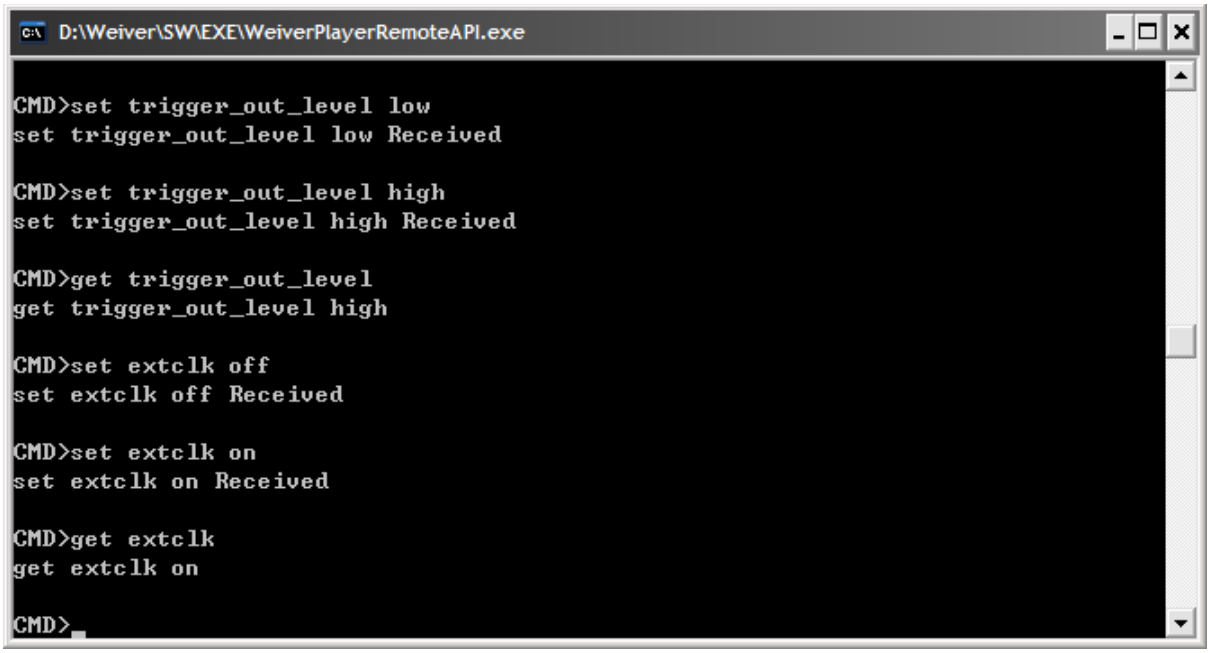


set extclk, get extclk Command

How to use)
You can synchronize device's action clock to external 10MHz input.
set extclk off: Uses internal 10MHz.
set extclk on: Synchronizes with external 10MHz input.

Explanation)
Generally, "set extclk off" is used.

Example)
1. set extclk off
2. set extclk on
3. get extclk

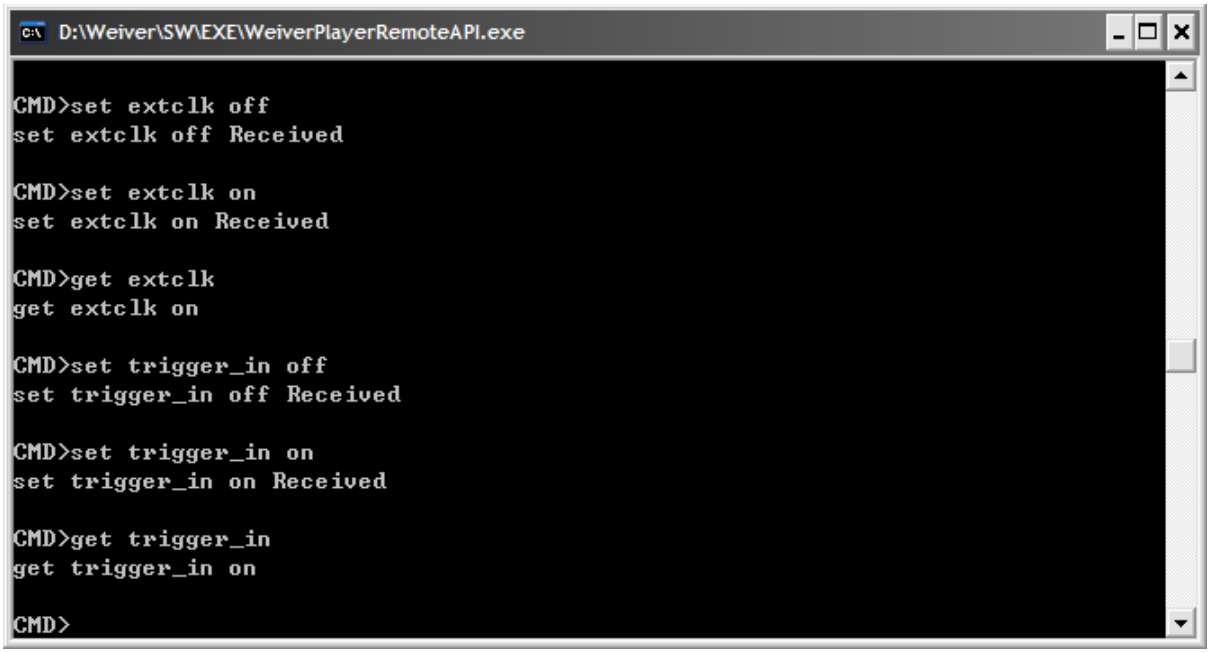


set trigger_in, get trigger_in Command

How to use)
set trigger_in off: RF sent is irrelevant to input level from "trigger In Port."
set trigger_in on: RF is sent when the input level from "trigger In Port" is HIGH(3.3V).

Explanation)
In case play is started with configured "set trigger_in", play is dependent to input level from "trigger In port." More than one Weiver 2.0 device can be synchronized and played..

Example)
1. set trigger_in off
2. set trigger_in on
3. get trigger_in



[WeiverPlayerRemoteAPI Source]

```
CWinApp theApp;
using namespace std;
const unsigned short u2_WeiverPlayerPort = WEIVER_PLAYER_API_RX_PORT;
const unsigned short u2_RecvPortFromWeiverPlayer = 50000;
const char s1a_WeiverPlayerIP[] = "192.168.100.1";
//const char s1a_WeiverPlayerIP[] = "127.0.0.1";
//const char s1a_WeiverPlayerIP[] = "192.168.10.15";

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
	int nRetCode = 0;

	if (!AfxWinInit(::GetModuleHandle(NULL), NULL, ::GetCommandLine(), 0))
	{
		_tprintf(_T("AfxWinInit failed.₩n"));
		nRetCode = 1;
	}
	else
	{
		wchar_t wCmdLine[1000];
		wchar_t wSendCmd[1000];

		WSADATA wsaData;
		WSAStartup(MAKEWORD(2,2), &wsaData);

		while(nRetCode == 0)
		{
			memset(wCmdLine, 0, 1000);
			memset(wSendCmd, 0, 1000);

			wprintf(_T("₩nCMD>"));

			_getws_s(wCmdLine, 1000);

			if(wcsncmp(wCmdLine, _T("exit"), wcslen(_T("exit"))) == 0)
			{
				nRetCode = 1;
			}
			else
			{
				SendToStringCMD((char*)wCmdLine, 2*wcslen(wCmdLine));
			}
			wprintf(_T("₩n"));
		}

		WSACleanup();

	}

	return nRetCode;
```

```
}

void SendToStringCMD(char *s1p_Data, unsigned int u4_Size)
{
	SOCKET h_Socket;
	SOCKADDR_IN Addr;

	int SentBytes;

	WvPlayerRemotePacket_t t_Packet;
	WvPlayerRemotePacket_tp tp_Packet;

	char *pSendData = (char *)&t_Packet;


	char   message[2000];
	memset(message, 0, 2000);


	int clntAddrSize = sizeof(Addr);

	h_Socket = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);

	memset(&Addr, 0, sizeof(Addr));
	Addr.sin_family = AF_INET;
	Addr.sin_port = htons(u2_RecvPortFromWeiverPlayer);
	Addr.sin_addr.s_addr = htonl(INADDR_ANY);


	struct timeval timeout;
	timeout.tv_sec = 2;//2 seconds
	timeout.tv_usec = 0;
	int optlen = sizeof(timeout);

	setsockopt(h_Socket, SOL_SOCKET, SO_RCVTIMEO, (const char *)&timeout, optlen);
	if(bind(h_Socket, (SOCKADDR*)&Addr, sizeof(Addr)) == SOCKET_ERROR)
	{
		wprintf(_T("bind() Error"));
		closesocket(h_Socket);
		return;
	}


	t_Packet.u4_ID = WEIVER_PLAYER_ID;
	t_Packet.u4_Reserved = WV_PLAYER_REMOTE_CMD_SET_STRING;
	t_Packet.u4_DataSize = u4_Size;

	memset(t_Packet.u1a_Data, 0, 1000);
	memcpy(t_Packet.u1a_Data, s1p_Data, u4_Size);

	memset(&Addr, 0, sizeof(Addr));
	Addr.sin_family = AF_INET;
```

```
	Addr.sin_port = htons(u2_WeiverPlayerPort);
	Addr.sin_addr.s_addr = inet_addr(s1a_WeiverPlayerIP);

	SentBytes = sendto(h_Socket
				,pSendData
				,4/*sizeof(t_Packet.u4_ID)*/
				+ 4/*sizeof(t_Packet.u4_Reserved)*/
				+ 4/*sizeof(t_Packet.u4_DataSize)*/ + u4_Size
				,0, (SOCKADDR*)&Addr, sizeof(SOCKADDR_IN));

	if(SentBytes == SOCKET_ERROR)
	{
		wprintf(_T("₩nSend Fail"));
	}
	else
	{
		memset(&Addr, 0, sizeof(Addr));

		int length = recvfrom(h_Socket
				, message, 2000, 0, (SOCKADDR*)&Addr, &clntAddrSize);

		if(length == -1)
		{
			wprintf(_T("Time Out"));
		}
		else
		{
			tp_Packet = (WvPlayerRemotePacket_tp)(message);

			wprintf(_T("%s"), tp_Packet->u1a_Data);
		}
	}

	closesocket(h_Socket);
}
```