



3rd Party Control Protocol for LINUS6.4
- LINUS Control v2.8 -





1. General	3
2. Commands.....	4
2.1. get_device_info	4
2.2. set_ip_static.....	4
2.3. set_snapshot	4
2.4. get_snapshot_folders	4
2.5. get_snapshot_files.....	4
2.6. get_snapshot_last_loaded	5
2.7. set_mute_all	5
2.8. set_channel_mute	5
2.9. get_channel_mute.....	5
2.10. set_channel_gain	5
2.11. get_channel_gain	5
2.12. set_channel_gain_down.....	6
2.13. set_channel_gain_up.....	6
2.14. set_channel_delay	6
2.15. get_channel_delay	6
2.16. set_fall_back_enabled	6
2.17. get_fall_back_enabled	6
2.18. get_fall_back_active	7
2.19. get_fall_back_time	7
2.20. set_fall_back_revert_to_primary_source.....	7
2.21. set_standby.....	7
2.22. get_standby	7
2.23. set_group_clear	7
2.24. set_identify	7
2.25. get_bridge_mode_12	8
2.26. get_bridge_mode_34	8
2.27. get_readback.....	8



1. General

The specifics of the 3rd Party Control Protocol for LINUS6.4, are as follows:

- The devices shall be controlled with UDP Datagrams.
- UDP Communication should be kept to the minimum amount possible to configure the Amplifiers.
- Redundant information should not be sent.
- A UDP message should always contain one command.
- UDP commands should be sent in the JSON format.
- UDP messages should be sent to UDP Port 6791 on the devices.
- SET commands are acknowledged by the amplifiers.
- GET commands will be responded to by the amplifier, always to the IP of the Host where the message originated, and to the originating UDP port on that Host.
- The amplifiers should only be controlled either by the LINUS Control software, -or- the 3rd Party Control Protocol, not both.
- Commands can be applied from device Firmware v1.5.

Note – For example in a typical bar installation, the system would be commissioned with LINUS Control, allowing for easy real-time control of all aspects of the design. When the commissioning engineer wants to commit the design, they would save the system in one or more 'Snapshots' inside the Amplifier. LINUS Control can then be removed, and the system can be remotely controlled by the 3rd Party Control protocol, for instance with a wall remote capable of sending UDP Datagrams. The snapshots can be loaded using this protocol, and changes (such as gains and mutes etc) pushed as and when required. Loading the snapshot once again would take the installation back to the state where the commissioning engineer had left it. Snapshots can be made for different configurations, times of day, areas of operation etc.



2. Commands

2.1.get_device_info

Message Format: {"command":"get_device_info"}

Example Reply: {"command": "get_device_info", "hw_addr": "04:89:5B:6D:E2:DA", "ip_addr": "192.168.1.30", "ip_mask": "255.255.255.0", "model": "LINUS6.4", "serial": "LI24110021", "sw_version": "V1.3", "hw_version": "V1.1", "desc": "LINUS"}

2.2.set_ip_static

Message Format: {"command":"set_ip_static", "ip_address":IP (String), "netmask":Netmask (String)}

Example Message: {"command":"set_ip_static", "ip_address":"192.168.0.5", "netmask":"255.255.255.0"}

Example Reply: {"status": "OK", "ip_address": "192.168.1.30", "netmask": "255.255.255.0", "command": "set_ip_static"}

2.3.set_snapshot

Message Format: {"command":"set_snapshot", "path":folder/file (String)}

Example Message: {"command":"set_snapshot", "path":"20/20 Test"}

Example Reply: {"response": "unknown snapshot", "status": "Error", "command": "set_snapshot"}

Snapshots in LINUS 6.4 are organised as files in folders. There are 20 folders for the 20 Snapshot slots. The 21st folder contains the live state of the amplifier, set from LINUS Control.

2.4.get_snapshot_folders

Message Format: {"command":"get_snapshot_folders"}

Example Message: {"command":"get_snapshot_folders"}

Example Reply: {"folders": ["PresetLib 1.2", "3", "4", "7", "11", "13", "14", "18", "DefaultName #1", "1", "2", "5", "20", "15", "10", "12", "8", "17", "6", "19", "9", "16", "21"], "command": "get_snapshot_folders"}

2.5.get_snapshot_files

Message Format: {"command":"get_snapshot_files", "folder":folder (String)}

Example Message: {"command":"get_snapshot_files", "folder":"1"}

Example Reply: {"files": ["1 test1"], "command": "get_snapshot_files"}



2.6.get_snapshot_last_loaded

Message Format: {"command":"get_snapshot_last_loaded"}

Example Message: {"command":"get_snapshot_last_loaded"}

Example Reply: {"snapshot_last_loaded": "20/20 test", "conform running state": false, "command": "get_snapshot_last_loaded"}

2.7.set_mute_all

Message Format: {"command":"set_mute_all", "enable":true/false (Boolean)}

Example Message: {"command":"set_mute_all", "enable":true}

Example Reply: {"mute": true, "status": "OK", "command": "set_mute_all"}

2.8.set_channel_mute

Message Format: {"command":"set_channel_mute", "channel":Channel number (Number), "enable":true/false (Boolean)}

Example Message: {"command":"set_channel_mute", "channel":3, "enable":true}

Example Reply: {"channel": 3, "mute": true, "status": "OK", "command": "set_channel_mute"}

2.9.get_channel_mute

Message Format: {"command":"get_channel_mute", "channel":Channel number (Number)}

Example Message: {"command":"get_channel_mute", "channel":3}

Example Reply: {"channel": 1, "mute": true, "status": "OK", "command": "get_channel_mute"}

2.10. set_channel_gain

Message Format: {"command":"set_channel_gain", "channel":Channel number (Number), "gain":Gain dB (Number)}

Example Message: {"command":"set_channel_gain", "channel":1, "gain":0}

Example Reply: {"channel": 1, "gain": 0.0, "status": "OK", "command": "set_channel_gain"}

2.11. get_channel_gain

Message Format: {"command":"get_channel_gain", "channel":Channel number (Number)}

Example Message: {"command":"get_channel_gain", "channel":1}

Example Reply: {"channel": 1, "gain": 0.0, "status": "OK", "command": "get_channel_gain"}



2.12. set_channel_gain_down

Message Format: {"command":"set_channel_gain_down", "channel":Channel number (Number), "step":Gain decrement dB (Number)}

Example Message: {"command":"set_channel_gain_down", "channel":1, "step":10}

Example Reply: {"channel": 1, "step": 10.0, "gain": -10.0, "status": "OK", "command": "set_channel_gain_down"}

2.13. set_channel_gain_up

Message Format: {"command":"set_channel_gain_up", "channel":Channel number (Number), "step":Gain increment dB (Number)}

Example Message: {"command":"set_channel_gain_up", "channel":1, "step":10}

Example Reply: {"channel": 1, "step": 10.0, "gain": 0.0, "status": "OK", "command": "set_channel_gain_up"}

2.14. set_channel_delay

Message Format: {"command":"set_channel_delay", "channel":Channel number (Number), "delay":Delay ms (Number)}

Example Message: {"command":"set_channel_delay", "channel":3, "delay":200}

Example Reply: {"channel": 3, "delay": 200.0, "status": "OK", "command": "set_channel_delay"}

2.15. get_channel_delay

Message Format: {"command":"get_channel_delay", "channel":Channel number (Number)}

Example Message: {"command":"get_channel_delay", "channel":1}

Example Reply: {"channel": 1, "delay": 0.0, "status": "OK", "command": "get_channel_delay"}

2.16. set_fall_back_enabled

Message Format: {"command":"set_fall_back_enabled", "enable":true/false (Boolean)}

Example Message: {"command":"set_fall_back_enabled", "enable":true}

Example Reply: {"enabled": true, "status": "OK", "command": "set_fall_back_enabled"}

2.17. get_fall_back_enabled

Message Format: {"command":"get_fall_back_enabled"}

Example Message: {"command":"get_fall_back_enabled"}

Example Reply: {"fall_back_enabled": true, "status": "OK", "command": "get_fall_back_enabled"}



2.18. get_fall_back_active

Message Format: {"command":"get_fall_back_active"}

Example Message: {"command":"get_fall_back_active"}

Example Reply: {"fall_back_active": false, "status": "OK", "command": "get_fall_back_active"}

2.19. get_fall_back_time

Message Format: {"command":"get_fall_back_time"}

Example Message: {"command":"get_fall_back_time"}

Example Reply: {"fall_back_time": 2, "status": "OK", "command": "get_fall_back_time"}

2.20. set_fall_back_revert_to_primary_source

Message Format: {"command":"set_fall_back_revert_to_primary_source"}

Example Message: {"command":"set_fall_back_revert_to_primary_source"}

Example Reply: {"status": "OK", "command": "set_fall_back_revert_to_primary_source"}

2.21. set_standby

Message Format: {"command":"set_standby", "enable":true/false (Boolean)}

Example Message: {"command":"set_standby", "enable":true}

Example Reply: {"standby": true, "status": "OK", "command": "set_standby"}

2.22. get_standby

Message Format: {"command":"get_standby"}

Example Message: {"command":"get_standby"}

Example Reply: {"standby": true, "status": "OK", "command": "get_standby"}

2.23. set_group_clear

Message Format: {"command":"set_group_clear"}

Example Message: {"command":"set_group_clear"}

Example Reply: {"status": "OK", "command": "set_group_clear"}

2.24. set_identify

Message Format: {"command":"set_identify", "enable":true/false (Boolean), "time":Duration [seconds] (Number)}

Example Message: {"command":"set_identify", "enable":true, "time":3}

Example Reply: {"identify": true, "status": "OK", "command": "set_identify"}



2.25. get_bridge_mode_12

Message Format: {"command": "get_bridge_mode_12"}

Example Message: {"command": "get_bridge_mode_12"}

Example Reply: {"bridge_mode_12": false, "status": "OK", "command": "get_bridge_mode_12"}

2.26. get_bridge_mode_34

Message Format: {"command": "get_bridge_mode_34"}

Example Message: {"command": "get_bridge_mode_34"}

Example Reply: {"bridge_mode_34": false, "status": "OK", "command": "get_bridge_mode_34"}

2.27. get_readback

Message Format: {"command": "get_readback"}

Example Message: {"command": "get_readback"}

Example Reply: {"V1": -80.0, "I1": 0.3, "GRRMS1": 0.0, "GRPK1": 0.0, "GRAMP1": 0.0, "S1": -80.0, "Z1": 0.0, "V2": -80.0, "I2": 0.3, "GRRMS2": 0.0, "GRPK2": 0.0, "GRAMP2": 0.0, "S2": -80.0, "Z2": 0.0, "V3": -80.0, "I3": 0.3, "GRRMS3": 0.0, "GRPK3": 0.0, "GRAMP3": 0.0, "S3": -80.0, "Z3": 0.0, "V4": -80.0, "I4": 0.3, "GRRMS4": 0.0, "GRPK4": 0.0, "GRAMP4": 0.0, "S4": -80.0, "Z4": 0.0, "AMPON": false, "OVERTEMP": false, "AVGLIM": false, "TEMP": 33, "SMPS": 235, "AES1LOCK": false, "AES2LOCK": false, "DANTELOCK": false, "AES1FS": "Unknown", "AES2FS": "Unknown", "DANTEFS": "Unknown", "status": "OK", "command": "get_readback"}