



3rd Party Control Protocol for LINUS6.4

- LINUS Control v2.9.4 -





■目次

1. 概要.....	3
2. コマンド	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	4
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	5
2.13. set_channel_gain_up.....	5
2.14. set_channel_delay.....	5
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	6
2.19. get_fall_back_time.....	6
2.20. set_fall_back_revert_to_primary_source	6
2.21. set_standby	6
2.22. get_standby.....	6
2.23. set_group_clear	7
2.24. set_identify	7
2.25. get_bridge_mode_12.....	7
2.26. get_bridge_mode_34.....	7
2.27. get_readback.....	7



1. 概要

LINUS6.4 のサードパーティー制御プロトコルの仕様は以下のとおりです。

- 各デバイスは UDP データグラムで制御されます。
- アンプの設定に必要な最小限の UDP 通信にとどめるべきです。
- リダンダント（冗長）の情報は送信しないでください。
- 1つの UDP メッセージには、常に 1つのコマンドのみとしてください。
- UDP コマンドは JSON 形式で送信される必要があります。
- UDP メッセージは、各デバイスの UDP ポート 6791 に送信される必要があります。
- SET コマンドは、アンプ側で応答（確認）されます。
- GET コマンドは、メッセージを送信したホストの IP アドレスと、そのホストの UDP ポートに対して応答が返されます。
- LINUS Control ソフトウェアとサードパーティー制御プロトコルは併用できません。どちらか一方のみで制御してください。
- これらのコマンドは、ファームウェア v1.5 以降のデバイスで使用可能です。

注意 例えば、一般的な商業設備への導入では、設計のあらゆる側面をリアルタイムで簡単にコントロールできるように LINUS Control でシステムの検証を行います。検証するエンジニアは、設計を確定したい場合、システムを 1つ以上の「スナップショット」としてアンプ内部に保存します。そうすることで LINUS Control 用の PC を取り外し、例えば UDP データグラムを送信できるウォールリモートコントローラーを使用して、サードパーティー制御プロトコルによってシステムをリモートコントロールすることが可能になります。このプロトコルを使用してスナップショットをロードしたり、必要に応じて変更（ゲインやミュート設定など）を適用したりすることができます。スナップショットをもう一度ロードすると、システムは設定をしたエンジニアが保存した状態に戻ります。スナップショットは、異なる構成や時間帯、運用エリアなどに合わせて作成できます。



2. コマンド

2.1. get_device_info

メッセージ形式 : {"command": "get_device_info"}

応答例 : {"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

メッセージ形式 : {"command": "set_ip_static", "ip_address": IP (文字列), "netmask": Netmask (文字列)}

メッセージ例 : {"command": "set_ip_static", "ip_address": "192.168.0.5", "netmask": "255.255.255.0"}

応答例 : {"status": "OK", "ip_address": "192.168.1.30", "netmask": "255.255.255.0", "command": "set_ip_static"}

2.3. set_snapshot

メッセージ形式 : {"command": "set_snapshot", "path": folder/file (文字列)}

メッセージ例 : {"command": "set_snapshot", "path": "20/20 Test"}

応答例 : {"response": "unknown snapshot", "status": "Error", "command": "set_snapshot"}

LINUS 6.4 のスナップショットは、フォルダー内のファイルとして構成されます。20 のスナップショットスロットに対して 20 個のフォルダーがあります。21 番目のフォルダーには、LINUS Control から設定されたアンプの現在の状態が格納されます。

注意 : “path” (パス) 引数に含まれる空白文字は削除したりトリミングしてはいけません。“path”引数は必ず、最初にフォルダ用スロット番号、次にファイル用スロット番号、その後にスペース (空白) 1 文字から始まります。これに続いてスナップショット名が続きますが、この名前は末尾に空白文字を追加して全体を 20 文字にパディングしなければなりません。

2.4. get_snapshot_folders

メッセージ形式 : {"command": "get_snapshot_folders"}

メッセージ例 : {"command": "get_snapshot_folders"}

応答例 : {"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

メッセージ形式 : {"command": "get_snapshot_files", "folder": folder (文字列)}

メッセージ例 : {"command": "get_snapshot_files", "folder": "1"}

応答例 : {"files": ["1 test1", "1", "command": "get_snapshot_files"}

2.6. get_snapshot_last_loaded

メッセージ形式 : {"command": "get_snapshot_last_loaded"}

メッセージ例 : {"command": "get_snapshot_last_loaded"}

応答例 : {"snapshot_last_loaded": "20/20 test", "conform running state": false, "command": "get_snapshot_last_loaded"}



"get_snapshot_last_loaded"}

2.7. set_mute_all

メッセージ形式 : {"command":"set_mute_all", "enable":true/false (ブール)}

メッセージ例 : {"command":"set_mute_all", "enable":true}

応答例 : {"mute": true, "status": "OK", "command": "set_mute_all"}

2.8. set_channel_mute

メッセージ形式 : {"command":"set_channel_mute", "channel":Channel number (数値), "enable":true/false (ブール)}

メッセージ例 : {"command":"set_channel_mute", "channel":3, "enable":true}

応答例 : {"channel": 3, "mute": true, "status": "OK", "command": "set_channel_mute"}

2.9. get_channel_mute

メッセージ形式 : {"command":"get_channel_mute", "channel":Channel number (数値)}

メッセージ例 : {"command":"get_channel_mute", "channel":3}

応答例 : {"channel": 1, "mute": true, "status": "OK", "command": "get_channel_mute"}

2.10. set_channel_gain

メッセージ形式 : {"command":"set_channel_gain", "channel":Channel number (数値), "gain":Gain dB (数値)}

メッセージ例 : {"command":"set_channel_gain", "channel":1, "gain":0}

応答例 : {"channel": 1, "gain": 0.0, "status": "OK", "command": "set_channel_gain"}

2.11. get_channel_gain

メッセージ形式 : {"command":"get_channel_gain", "channel":Channel number (数値)}

メッセージ例 : {"command":"get_channel_gain", "channel":1}

応答例 : {"channel": 1, "gain": 0.0, "status": "OK", "command": "get_channel_gain"}

2.12. set_channel_gain_down

メッセージ形式 : {"command":"set_channel_gain_down", "channel":Channel number (数値), "step":Gain decrement dB (数値)}

メッセージ例 : {"command":"set_channel_gain_down", "channel":1, "step":10}

応答例 : {"channel": 1, "step": 10.0, "gain": -10.0, "status": "OK", "command": "set_channel_gain_down"}

2.13. set_channel_gain_up

メッセージ形式 : {"command":"set_channel_gain_up", "channel":Channel number (数値), "step":Gain increment dB (数値)}

メッセージ例 : {"command":"set_channel_gain_up", "channel":1, "step":10}

応答例 : {"channel": 1, "step": 10.0, "gain": 0.0, "status": "OK", "command": "set_channel_gain_up"}

2.14. set_channel_delay

メッセージ形式 : {"command":"set_channel_delay", "channel":Channel number (数値), "delay":Delay ms (数値)}



メッセージ例 : {"command": "set_channel_delay", "channel": 3, "delay": 200}

応答例 : {"channel": 3, "delay": 200.0, "status": "OK", "command": "set_channel_delay"}

2.15. get_channel_delay

メッセージ形式 : {"command": "get_channel_delay", "channel": Channel number (数値)}

メッセージ例 : {"command": "get_channel_delay", "channel": 1}

応答例 : {"channel": 1, "delay": 0.0, "status": "OK", "command": "get_channel_delay"}

2.16. set_fall_back_enabled

メッセージ形式 : {"command": "set_fall_back_enabled", "enable": true/false (ブール)}

メッセージ例 : {"command": "set_fall_back_enabled", "enable": true}

応答例 : {"enabled": true, "status": "OK", "command": "set_fall_back_enabled"}

2.17. get_fall_back_enabled

メッセージ形式 : {"command": "get_fall_back_enabled"}

メッセージ例 : {"command": "get_fall_back_enabled"}

応答例 : {"fall_back_enabled": true, "status": "OK", "command": "get_fall_back_enabled"}

2.18. get_fall_back_active

メッセージ形式 : {"command": "get_fall_back_active"}

メッセージ例 : {"command": "get_fall_back_active"}

応答例 : {"fall_back_active": false, "status": "OK", "command": "get_fall_back_active"}

2.19. get_fall_back_time

メッセージ形式 : {"command": "get_fall_back_time"}

メッセージ例 : {"command": "get_fall_back_time"}

応答例 : {"fall_back_time": 2, "status": "OK", "command": "get_fall_back_time"}

2.20. set_fall_back_revert_to_primary_source

メッセージ形式 : {"command": "set_fall_back_revert_to_primary_source"}

メッセージ例 : {"command": "set_fall_back_revert_to_primary_source"}

応答例 : {"status": "OK", "command": "set_fall_back_revert_to_primary_source"}

2.21. set_standby

メッセージ形式 : {"command": "set_standby", "enable": true/false (ブール)}

メッセージ例 : {"command": "set_standby", "enable": true}

応答例 : {"standby": true, "status": "OK", "command": "set_standby"}

2.22. get_standby

メッセージ形式 : {"command": "get_standby"}

メッセージ例 : {"command": "get_standby"}

応答例 : {"standby": true, "status": "OK", "command": "get_standby"}



2.23. set_group_clear

メッセージ形式 : {"command": "set_group_clear"}

メッセージ例 : {"command": "set_group_clear"}

応答例 : {"status": "OK", "command": "set_group_clear"}

2.24. set_identify

メッセージ形式 : {"command": "set_identify", "enable": true/false (ブール), "time": Duration [seconds] (数値)}

メッセージ例 : {"command": "set_identify", "enable": true, "time": 3}

応答例 : {"identify": true, "status": "OK", "command": "set_identify"}

2.25. get_bridge_mode_12

メッセージ形式 : {"command": "get_bridge_mode_12"}

メッセージ例 : {"command": "get_bridge_mode_12"}

応答例 : {"bridge_mode_12": false, "status": "OK", "command": "get_bridge_mode_12"}

2.26. get_bridge_mode_34

メッセージ形式 : {"command": "get_bridge_mode_34"}

メッセージ例 : {"command": "get_bridge_mode_34"}

応答例 : {"bridge_mode_34": false, "status": "OK", "command": "get_bridge_mode_34"}

2.27. get_readback

メッセージ形式 : {"command": "get_readback"}

メッセージ例 : {"command": "get_readback"}

応答例 : {"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"}

CODA

C O D A A U D I O



202512

●この製品を安全にお使いいただくために、設置・運用には十分な安全対策を行ってください。●商品写真やイラストは、実際の商品と一部異なる場合があります。●掲載内容は発行時のもので、予告なく変更されることがあります。変更により発生したいかなる損害に対しても、弊社は責任を負いかねます。●記載されている商品名、会社名等は各社の登録商標、または商標です。



ヒビノインターサウンド株式会社

〒105-0022 東京都港区海岸2-7-70

E-mail: info@hibino-intersound.co.jp

<https://www.hibino-intersound.co.jp/>