



LMP SH-IP & HRNL-25W MQTT 3.1.1 PROTOCOL GUIDE

Public protocol integration guide

DATASHEET

MQTT

v1.1.193 | BUILD 193

www.xichtee.com
office@xichtee.com
Company ID: 56695284
Tax ID / VAT: 2122403635



Document control

DOCUMENT	VALUE
Title	LMP SH-IP & HRNL-25W MQTT 3.1.1 Protocol Guide
Document revision	1.0 · first draft
Firmware baseline	v1.1.193 · build 193
Issued	2026-08-21 · Europe/Bratislava
Audience	Automation, monitoring and control-system integrators

Purpose

This guide documents the implemented MQTT 3.1.1 client: broker connection, exact topics, retained status and availability, accepted command documents, state semantics, recovery and Cluster authority.

NOTE This is a public guide to the implemented product interface. Verify the running firmware version before relying on version-specific fields or commands.



DATASHEET

LMP SH-IP & HRNL-25W

MQTT 3.1.1 PROTOCOL GUIDE

Contents

Document control	2
Contents	3
01 · Overview	4
02 · Configure the client	5
03 · Topic model	6
04 · PUBLISH anatomy	7
05 · Retained status	8
06 · Control commands	10
07 · Rule and preset configuration	11
08 · Working client examples	13
09 · Session and recovery	14
10 · Cluster and security	15
11 · Reference	16



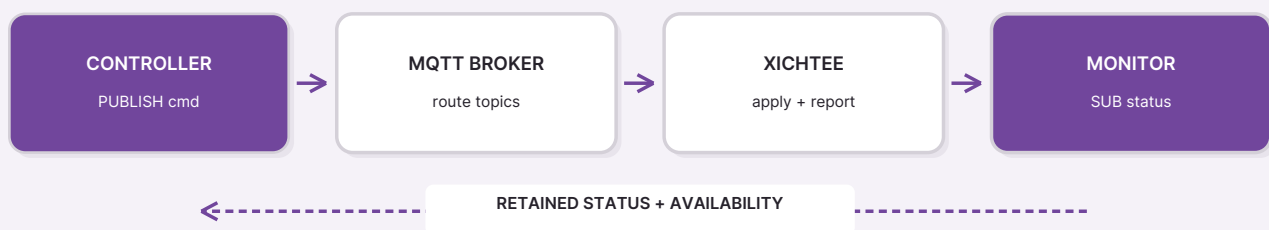
01 • Overview

Broker-mediated control and telemetry

The device is an MQTT client. It opens a TCP connection to a configured broker, subscribes to one device command topic and one prefix-wide command topic, then publishes retained availability and a retained JSON status document.

MQTT APPLICATION PATH

The broker separates command publishers from the device and from status consumers.



Commands use QoS 0 and no application reply; status is the observable readback.

ITEM	IMPLEMENTED CONTRACT
Protocol	MQTT 3.1.1, protocol level 4, over plain IPv4 TCP.
Role	Device-initiated client connection to one numeric broker IPv4 address.
Session	Clean Session; optional username/password; generated client identifier.
Delivery	QoS 0 for subscriptions, status, availability, Last Will and commands.
Security	No TLS, certificates, DNS hostname, WebSockets, MQTT 5 or IPv6.
Licence	An active MQTT licence and one available shared network socket are required.

NOTE QoS 0 means best-effort delivery. A broker-level successful publish is not proof that a command was valid, authorised or applied.



02 · Configure the client

Set the broker in Network Config

Enable the MQTT Client, enter the broker's numeric IPv4 address, port, keepalive and a topic prefix, then save. Network-settings changes restart the device. Optional credentials are sent only when configured.

SETTING	VALUE / GUIDANCE
Enable	MQTT licence active and Enable MQTT Client selected.
Broker IP	Numeric IPv4 address; 0.0.0.0 disables the client. DNS names are not supported.
Broker Port	1–65535; stored default 1883.
Keepalive	5–600 seconds; stored default 30 seconds.
Topic Prefix	Required; up to 63 bytes. Default xichtee.
Username	Optional; up to 31 bytes.
Password	Optional; up to 63 bytes. Configure a username when broker authentication is used.
Socket budget	MQTT consumes one shared runtime network socket.

Prefix rules

Use a short, stable prefix without MQTT wildcard characters and without leading or trailing slashes. Separate production environments with different prefixes or broker ACL namespaces.

IMPORTANT The connection is unencrypted. Broker credentials, commands and state are visible to any party able to observe the TCP path. Use an isolated trusted network or an external TLS/VPN/broker-bridge boundary.



03 · Topic model

Exact topic tree

Let `<prefix>` be the configured prefix and `<uuid>` the device UUID shown in the web interface. The UUID branch is device-specific; the all branch broadcasts to every matching XICHTEE client using that prefix.

MQTT TOPIC NAMESPACE

PUB is device → broker. SUB is broker → device. Retained applies only to device publications.

`<prefix>`

NODE

configured namespace

— `<uuid>`

NODE

one device

— `status`

PUB

retained JSON · about every 1 s

— `availability`

PUB

retained online / offline Last Will

— `cmd`

SUB

device command · publish non-retained

— `all`

NODE

every device sharing prefix

— `cmd`

SUB

broadcast command · publish non-retained

Prefer `<prefix>/<uuid>/cmd`. Use `<prefix>/all/cmd` only for deliberate fleet-wide actions.

TOPIC	DIRECTION	RETAIN	PAYLOAD
<code>//status</code>	Device publishes	Yes	JSON object
<code>//availability</code>	Device publishes	Yes	online / offline
<code>//cmd</code>	Device subscribes	No	JSON or key=value
<code>/all/cmd</code>	Device subscribes	No	JSON or key=value

IMPORTANT Never retain a command. A retained command can execute again whenever the device reconnects and resubscribes.



04 • PUBLISH anatomy

Validated QoS-0 command packet

This worked packet publishes a compact JSON command to demo/all/cmd. It is shown to make topic length, Remaining Length and the retain flag unambiguous; production applications should use a standards-compliant MQTT client library.

MQTT PUBLISH • APPLY_PRESET

57 bytes total: 2 fixed-header bytes + 2 topic-length bytes + 12 topic bytes + 41 payload bytes.

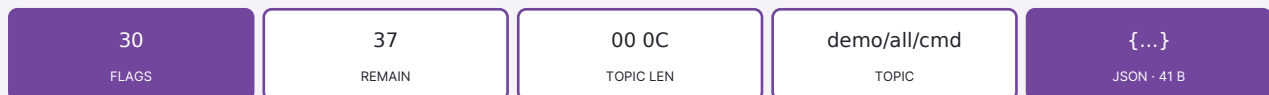


30 = PUBLISH, DUP 0, QoS 0, RETAIN 0.

37 hex = 55 remaining bytes after the fixed header.

00 0C = 12-byte UTF-8 topic; the final 41 bytes are UTF-8 JSON.

VALIDATED LOGICAL FIELDS



All labels and examples are vector text and remain searchable in the PDF.

```
30 37 00 0C
64 65 6D 6F 2F 61 6C 6C 2F 63 6D 64
7B 22 63 6D 64 22 3A 22 61 70 70 6C 79 5F 70 72
65 73 65 74 22 2C 22 73 65 67 22 3A 30 2C 22 70
72 65 73 65 74 22 3A 33 7D
```

NOTE The complete broker-to-device MQTT packet is limited to 512 bytes, including fixed header, Remaining Length, topic and payload. Keep command documents compact.



05 • Retained status

One JSON snapshot per active device

The device publishes retained status immediately after an accepted change and approximately once per second while connected. Arrays follow active-segment order; zero-based indices in commands align with array index 0.

STATUS OBJECT FAMILIES

A retained document is a snapshot, not proof of a currently live connection.

RUNTIME

ip · mode · seg · swap · uptime_s · fw · features

AUX PORT 1

mode1_kind · gpi_state · temp_c · temp_status

AUX PORT 2

mode2_kind · gpi_state · temp_c · temp_status

SEGMENTS

counts[] · colors[][] in active segment order

PRESETS

preset_audio[10] with channels, thresholds and three colours

RULES

rules_active[] · five-bit mask per active segment

Use availability plus a client-side status-age limit before treating retained state as current.

FIELD	TYPE / MEANING
ip	Current IPv4 address string.
mode / seg	Zero-based active-segment mode / active segment count.
swap	Boolean physical segment-order setting.
uptime_s / fw	Integer uptime reading / firmware version string.
features	Unsigned licensed-feature bitmask; treat as informational.
mode1_kind / mode2_kind	gpi or sensor.
modeN_gpi_state	Boolean in GPI mode; otherwise null.
modeN_temp_c	Celsius number when temp status is ok; otherwise null.
modeN_temp_status	disabled, starting, ok, disconnected, crc_error, invalid or stale.

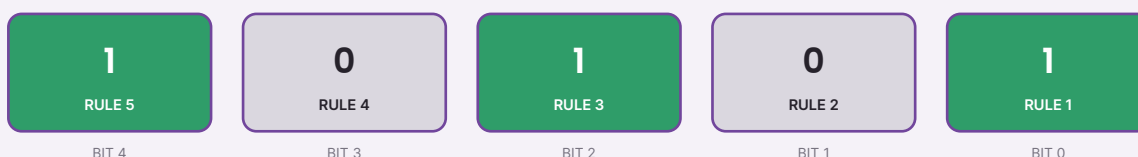


Status arrays

FIELD	SHAPE	MEANING
counts	[integer × seg]	Configured LED count for each active segment.
colors	[[r,g,b] × seg]	Stored/manual segment colour; each component 0–255.
rules_active	[mask × seg]	Unsigned 0–31 mask; bit 0 Rule 1 through bit 4 Rule 5.
preset_audio	[object × 10]	Exactly one Audio Monitor object for each preset.
preset_audio[].ch	[left,right]	Two channel numbers, each 1–32.
preset_audio[].th	[low,high]	Ordered thresholds, each 0–100.
preset_audio[].c	[lowRGB,midRGB,highRGB]	Three RGB triplets, components 0–255.

RULES_ACTIVE MASK - EXAMPLE VALUE 21 / 0x15

Bit 0 is Rule 1. A set bit reports Active; it does not prove that rule currently wins LED priority.



21 = 16 + 4 + 1 → Rules 5, 3 and 1 are Active; Rule 1 is the visible winner on that segment.

```
{ "ch": [1,2], "th": [65,85],  
  "c": [[0,255,0],[255,140,0],[255,0,0]] }
```

NOTE colors is not necessarily the instantaneous rendered LED colour while a rule or animation is active. The status object also does not identify the current Cluster leader.



06 • Control commands

Compact JSON is recommended

Publish commands with QoS 0 and RETAIN false. JSON must begin directly with {. A key/value compatibility form such as `cmd=set_rule&seg=0&idx=1&key=0&val=1` is also accepted. All indices below are zero-based.

COMMAND	MINIMAL JSON	EFFECT
reboot	<code>{"cmd":"reboot"}</code>	Immediate software restart.
reboot alias	<code>{"reboot":1}</code>	Same immediate restart.
apply_preset	<code>{"cmd":"apply_preset","seg":0,"preset":3}</code>	Copies preset 4 base RGB into segment 1 manual colour.
set_rule Active	<code>{"cmd":"set_rule","seg":0,"idx":1,"key":0,"val":1}</code>	Requests Segment 1 Rule 2 On.
legacy mode	<code>{"mode":1}</code>	Selects zero-based active-segment mode, capped by licence.
legacy colour	<code>{"seg":0,"r":0,"g":255,"b":0}</code>	Sets all three manual RGB components for Segment 1.

Command semantics

RULE	DETAIL
No application ACK	There is no response or error topic. Observe status or another public readback.
apply_preset	Copies the preset's base RGB only; it does not activate a rule or independently start an effect.
Runtime Active	Rule Active changes are not persisted.
Configuration save	Preset, rule configuration, mode and colour save after about 120 seconds from the latest change.
Temporary busy state	A command can be dropped rather than queued while another configuration operation owns the model.
Reboot timing	Do not reboot immediately after a configuration command that still needs the delayed save.
Broadcast risk	A command on <code>/all/cmd</code> can affect or reboot every matching device.

IMPORTANT A maintained Follow GPI/Logic input owns Rule Active. MQTT Active writes are ignored until that ownership is released.



07 • Rule and preset configuration

set_rule

Select a licensed segment with seg or type, and Rule 1–5 with idx or rule values 0–4. Either send one key/value pair or the named direct fields.

KEY	DIRECT FIELD	VALUE
0	active	Boolean request.
1	preset_id / preset	0–9.
2	tsl_index	–1 disables; 0–9999.
3	tsl_bit	0–3.
4	osc_path	Up to 63 bytes.
5	osc_match_mode	0 Trigger, 1 Exact, 2 Range, 3 String.
6 / 7	osc_arg_min / osc_arg_max	Floating-point A / B.
8	osc_arg_str	Up to 31 bytes.
9	gpi_source	0 Disabled, 1–2 GPI, 3–12 Logic outputs 1–10.

set_preset selectors and core fields

Select preset 1–10 with idx or id values 0–9. All other fields are optional.

FIELDS	VALUE
r, g, b	Non-negative; values above 255 clamp to 255.
name	Non-empty; up to 31 bytes.
sa / ea	Start/stop effect: 0 Instant, 1 Wipe, 2 Blink, 3 Fade, 4 Breathing, 5 Strobe, 6 Chase, 7 Phone Ring, 10–19 Custom 1–10.
ca	Constant: 0 Static, 1 Wipe, 2 Blink, 3 Fade, 4 Breathing, 5 Strobe, 6 Chase, 7 Phone Ring, 8 Morse, 9 Audio Monitor, 10–19 Custom.
sd / ed	0 left-to-right, 1 right-to-left.
cd	0 left-to-right, 1 right-to-left, 2 centre-out / two-channel.
st / et / ct / ct2	Non-negative milliseconds.
ctx	Morse text: A–Z, 0–9 and spaces; maximum 64 characters; lower-case normalised.



Audio Monitor preset fields

FIELDS	VALUE / NORMALISATION
amch / amch2	Left/right channel; clamps to 1–32.
amt1 / amt2	Thresholds clamp to 0–100 and are reordered low/high.
amc1 / amc2 / amc3	Low/mid/high colour as exact #RRGGBB or RRGGBB.
amc1r/g/b	Low RGB component aliases; each clamps to 0–255.
amc2r/g/b	Middle RGB component aliases; each clamps to 0–255.
amc3r/g/b	High RGB component aliases; each clamps to 0–255.
Precedence	Explicit component aliases override the corresponding hex colour component.

Examples

```
{ "cmd": "set_rule", "seg": 0, "idx": 0, "active": 1 }
{ "cmd": "set_rule", "seg": 0, "idx": 0, "preset": 4,
  "osc_path": "/studio/program", "osc_match_mode": 0 }
{ "cmd": "set_preset", "idx": 4, "name": "PROGRAM",
  "r": 255, "g": 0, "b": 0, "ca": 0 }
{ "cmd": "set_preset", "idx": 5, "ca": 9, "amch": 1, "amch2": 2,
  "amt1": 65, "amt2": 85, "amc1": "#00FF00",
  "amc2": "#FF8C00", "amc3": "#FF0000" }
```

NOTE Do not combine an unrelated command family with legacy mode/colour fields in the same payload. Send one clear intent per PUBLISH.



08 · Working client examples

Observe availability and status

The commands below use Mosquitto command-line clients and a documentation-only broker address. Replace the broker, prefix and UUID. Use a protected client configuration or broker credential mechanism when authentication is required.

```
# Terminal 1 · observe retained availability and status
mosquitto_sub -h 192.0.2.20 -p 1883 -v -q 0 \
-t 'demo/+/availability' -t 'demo/+/status'

# Terminal 2 · device-specific Rule 1 On, never retained
mosquitto_pub -h 192.0.2.20 -p 1883 -q 0 \
-t 'demo/replace-with-device-uuid/cmd' \
-m '{"cmd":"set_rule","seg":0,"idx":0,"active":1}'
```

Validated broadcast packet

```
mosquitto_pub -h 192.0.2.20 -p 1883 -q 0 \
-t 'demo/all/cmd' \
-m '{"cmd":"apply_preset","seg":0,"preset":3}'
```

VERIFY	EXPECTED OBSERVATION
availability	retained online while the session is established; retained offline after the broker publishes the Last Will.
status freshness	New status about every second; reject snapshots older than the controller's chosen freshness limit.
rules_active	The matching segment mask changes only if the Active command was accepted and not input-owned.
apply_preset	The segment entry in colors changes to the selected preset's base RGB.

IMPORTANT Do not put passwords directly on a command line or in scripts. They can be exposed through history, logs and process inspection.

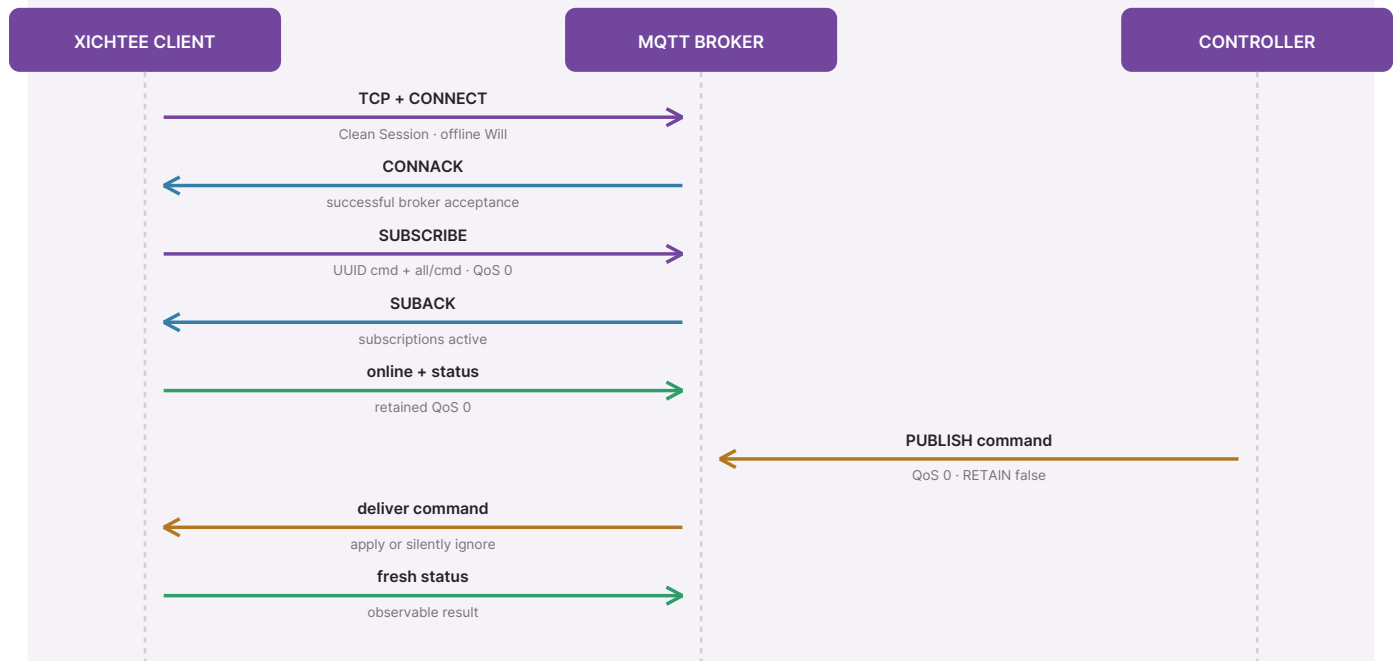


09 • Session and recovery

Connect, subscribe, publish, recover

MQTT 3.1.1 SESSION

The device reconnects automatically after transport, framing, acknowledgement or keepalive failure.



Idle sessions use PINGREQ/PINGRESP; loss causes close and a clean reconnect.

CONDITION	BEHAVIOUR / CLIENT GUIDANCE
Normal reconnect	Automatic retry after approximately 3 seconds.
Connect / MQTT acknowledgement stall	Session is closed after a bounded wait; it reconnects cleanly.
Missing PINGRESP	Connection closes and reconnects; status traffic does not hide a failed keepalive.
Malformed, oversized or incomplete packet	Session is discarded rather than continuing with uncertain framing.
Retained offline	Broker Last Will indicates an ungraceful disconnect; still apply status freshness checks.
Reconnect	Clean Session resubscribes. Any retained command is dangerous because it can replay.



10 · Cluster and security

Broker online is not control authority

CLUSTER WRITE OWNERSHIP

Publish commands only after a separate Cluster-status read identifies the ready leader.

STANDALONE / OFF

MQTT commands are accepted subject to normal validation.

PRIMARY MASTER

Commands are accepted while it is the configured active master.

FOLLOWER

MQTT remains read-capable and may publish followed state; every command is ignored.

BACKUP STANDBY

Do not expect a live/fresh MQTT session until the backup is promoted.

BACKUP LEADER

Commands are accepted only while the ready elected backup owns control.

FAILOVER

Discover the new leader, wait for its MQTT session, then reassert controller intent.

Neither online availability nor mqtt_ok proves Cluster leadership.

Broker and network controls

CONTROL	RECOMMENDATION
Network boundary	Keep plain MQTT inside a trusted isolated network or protect it with a VPN/TLS bridge.
Broker ACL	Allow the device to publish UUID status/availability and subscribe only to its UUID and approved all command topic.
Broadcast topic	Deny publisher access unless fleet-wide commands are explicitly required.
Credentials	Use unique least-privilege broker credentials and never expose them in documents, source or logs.
Monitoring	Alert on offline, stale status, repeated reconnects and unexpected UUID/prefix combinations.
Writer ownership	Use one authoritative application writer per rule or configuration object.

IMPORTANT A Follower can be MQTT-online while deliberately refusing commands. Treat Cluster authority and MQTT transport health as separate signals.



11 • Reference

Quick reference

TOPIC	VALUE
Protocol	MQTT 3.1.1 · Clean Session · plain IPv4 TCP
Default broker settings	Port 1883 · keepalive 30 s · prefix xichtee
QoS	0 for all implemented publish and subscribe operations
Status	//status · retained JSON · about 1 s
Availability	//availability · retained online/offline
Device command	//cmd · publish non-retained
Broadcast command	/all/cmd · publish non-retained
Incoming packet limit	512 bytes including MQTT framing, topic and payload
Application response	None; confirm through status or another public readback
Rule Active	Runtime only; not persisted

Standard reference

OASIS MQTT Version 3.1.1 with Errata 01:

<https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html>

Compatibility

This guide applies to LMPSH-IP and HRNL-25W firmware v1.1.193 (build 193). It documents XICHTEE topics, JSON fields, commands, retained-state interpretation and Cluster behavior in addition to MQTT 3.1.1 framing.

IMPORTANT The MQTT transport is not encrypted. Do not route it across an untrusted network without an external trusted security boundary.