Contents

Nexus Ultimate Prop Controller MQTT Integration Guide v1.2

This guide documents MQTT behavior for systems added from the web UI (`Config -> Integrations -> Add Integration System -> mqtt`).

UI Add Defaults (MQTT)

When you add a new MQTT system from the UI, the stored defaults are:

- `type: "mqtt"`
- `enabled: true`
- `base: ""`
- `subscribe_cmd: true`
- `timeout_ms: 2000`
- `rate_limit_ms: 0`
- `publish.events: true`
- `publish.inputs: true`
- `publish.analog: true`
- `publish.state: true`
- `publish.errors: true`
- `publish.actions: true`
- `publish.health: true`

If `base` is blank, effective base is:

`nexusPropControllers/{room_prefix}/props/{device_name}`

Default MQTT Input Address Map (What Firmware Listens To)

`subscribe_cmd=true` is enabled by default, so command subscriptions are active.

Subscription Filters

Case	Subscribed Topic Filter
Always (primary)	`<effective_base>/cmd/#`
Additional enabled MQTT systems with unique base	`<system_base>/cmd/#`

Notes: - If a system `base` is empty, it resolves to `<effective_base>`. - If multiple systems resolve to the same base, that subscription is effectively the same filter. - Manual routes (`integrations.manual_routes` with `transport: "mqtt"`) are subscribed as explicit topics from `manual_routes.path` and matched against full inbound topics (no base-prefix rewrite). - Outbound `mqtt` actions publish to the explicit configured target topic (no base-prefix rewrite).

Inbound Command Mapping

All received command messages are accepted on subscribed `.../cmd/#` topics and then mapped as follows:

Match Rule	Topic Form	Payload Form	Result
Envelope payload (preferred)	Any subscribed `.../cmd/...` topic	`{"action":"...","target":"...","payload":"..."}`	`runAction(action,target,payload)`
Path encoded A	`.../cmd/action/<action>/<target>`	Raw payload string	`runAction(action,target,raw_payload)`
Path encoded B	`.../cmd/<action>/<target>`	Raw payload string	`runAction(action,target,raw_payload)`
Path encoded C	`.../api/action/<action>/<target>`	Raw payload string	`runAction(action,target,raw_payload)`
Legacy command-router payload	Any subscribed `.../cmd/...` topic	`{"cmd":"...", ...}`	Routed through legacy command router

Explicit Examples

Topic	Payload	Runtime Interpretation
`<base>/cmd/action`	`{"action":"set_state","target":"armed","payload":"1"}`	set state to `armed`
`<base>/cmd/action/set_var/puzzle`	`1`	`set_var("puzzle","1")`
`<base>/cmd/operator_control/Reset`	‘`(empty)` | operator_control(“Reset”,“”) | | /cmd/relay/Solenoid | trigger | relay(”Solenoid”,“trigger”) | | /cmd/anything | {“cmd”:“relay”,“id”:“Solenoid”,“value”:true}‘	legacy command-router relay command

Default MQTT Output Address Map (What Firmware Publishes)

With default publish flags from UI add, these topics are published.

Connect-Time Publications

Topic	Retain	Payload
`<base>/sys/online`	Yes	`online`
`<base>/sys/info`	Yes	JSON with firmware version
`<room_prefix>/<device_name>/online`	No	device IP string

For additional enabled MQTT systems with a different resolved base, `<sys_base>/sys/online` and `<sys_base>/sys/info` are mirrored too.

Periodic Publications

Topic	Retain	Interval	Payload
`<base>/state`	Yes	~15s	state snapshot JSON
`<base>/sys/health`	No	~5s	health JSON (`heap`, `uptime`)

Event Publications

Topic Pattern	Retain	Payload
`<base>/events/<event_type>`	No	event JSON/text payload

Explicit Event Type List

These event type suffixes are emitted by firmware code paths:

- `action`
- `actuator`
- `analog`
- `analog_trigger`
- `color`
- `espnow_event`
- `gesture`

- i2c
- input
- input_edge
- logic
- rfid
- rfid_collection
- rfid_data
- rfid_recover
- rfid_rule
- route
- sensor_trigger
- sequence
- sequence_learn
- system
- test

Publish-Flag Routing Map

Event Type	Controlling Publish Flag
input	publish.inputs
analog	publish.analog
action, sequence	publish.actions
error (when emitted as an event type)	publish.errors
input_edge and all other listed event types	publish.events

Periodic/state channels (not events/<type> routing): - <base>/state uses publish.state. - <base>/sys/health uses publish.health.

Scope Notes

- This guide is strictly for UI-added MQTT integrations.
- Manual routes can add additional behavior, but are not part of default UI-add behavior.