

# Contents

<b>Nexus Ultimate Prop Controller OSC Integration Guide v1.1</b>	<b>1</b>
UI Add Defaults (OSC)	1
Default OSC Input Address Map (What Firmware Listens To)	1
Default OSC Output Address Map (What Firmware Sends)	2
Action-Trigger Payload Tokening	2
Explicit Defaults Summary (UI Add)	3

## Nexus Ultimate Prop Controller OSC Integration Guide v1.1

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

### UI Add Defaults (OSC)

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

- type: "osc"
- enabled: true
- host: ""
- port: 8000
- path: "/cue"
- subscribe\_cmd: true
- subscribe\_address: ""
- var: ""
- listen\_port: 0
- timeout\_ms: 2000

Effective listen port resolution:

- If listen\_port > 0: listen on listen\_port.
- Else: listen on port.
- Default UI-add result: listen on UDP 8000.

### Default OSC Input Address Map (What Firmware Listens To)

subscribe\_cmd=true is enabled by default, so inbound command parsing is active.

### Address-to-Action Parsing Map

When first argument is not an envelope JSON, OSC address is parsed using these patterns:

Address Pattern	Parsed Action	Parsed Target
/api/action/<action>/<target>	<action>	<target>
/api/action/<action>	<action>	empty
/cmd/action/<action>/<target>	<action>	<target>
/cmd/action/<action>	<action>	empty
/action/<action>/<target>	<action>	<target>
/cmd/<action>/<target>	<action>	<target>
/action/<action>	<action>	empty
/cmd/<action>	<action>	empty

First OSC argument becomes runtime payload in these path-encoded forms.

## Envelope Payload Map

If first OSC argument is JSON containing `action`, the payload itself controls dispatch:

Address	Arg0 JSON	Result
Any address (commonly <code>/api/action</code> )	<code>{"action":"...", "target":"...", "payload":"..."}runAction(action,target,payload)</code>	

## First-Argument Type Decoding Map

OSC decoder reads first argument and converts it to action payload string:

OSC Type Tag	Payload String Passed to Runtime
<code>i</code>	integer as string
<code>f</code>	float as string (4 decimals)
<code>s</code>	raw string
<code>T</code>	<code>true</code>
<code>F</code>	<code>false</code>
no arg / unsupported type	1

## Default Variable-Binding Listener State

Variable binding only activates if both are configured:

- `subscribe_address`
- `var`

UI defaults set both to empty, so variable-binding is off by default.

## Default OSC Output Address Map (What Firmware Sends)

No automatic periodic OSC telemetry is sent by default after UI add.

Outbound OSC traffic is sent only when runtime executes an `osc` integration action.

## Outbound Destination Map

Field	Default / Behavior
Destination host	<code>system.host</code> (must be numeric IP for outbound send path)
Destination port	<code>system.port</code> (default 8000)
OSC address	<code>system.path</code> (default <code>/cue</code> ) unless overridden by payload address
OSC args	payload args array, else payload value, else raw payload fallback

## Explicit Outbound Payload Forms

Action Payload	Sent OSC Message
<code>{"address":"/cue/fire", "args":["go",1,true]}</code>	Address <code>/cue/fire</code> , args <code>"go", 1, true</code>
<code>{"value":1}</code>	Address from <code>system.path</code> , args <code>[1]</code>
<code>"/cue/fire"</code>	Address <code>/cue/fire</code> , no args
<code>"go"</code>	Address from <code>system.path</code> , args <code>["go"]</code>

## Action-Trigger Payload Tokening

When an OSC/manual-route trigger fires and you need to pass its inbound value into an action step (for example `set_var`, `mqtt`, `http_post`, or other payload-bearing actions), use any of:

- `<value>`
- `<event_value>`

- <payload>
- <event\_payload>

All four resolve to the same inbound event value.

### **Explicit Defaults Summary (UI Add)**

- Inbound enabled: Yes (`subscribe_cmd=true`).
- Inbound address parser enabled: Yes (patterns listed above).
- Inbound variable binding enabled: No (empty `subscribe_address` and `var`).
- Outbound automatic stream: No.
- Outbound action-driven OSC send: Yes.