VZX-8 8-Zone Audio Processor

Open Interface Protocol Documentation

Table of Contents

Preface

The zone audio processors can be controlled via a protocol called **Open Interface**. The interface can be used to `get`, `set`, `subscribe`, and `unsubscribe` to parameters of a zone audio processor. These parameters can be accessed through different protocols and ports:

- **TCP/IP** (port 8989)
- **JSONRPC/HTTP/IP** (port 8990)
- **OCA/TCP/IP** (ports 55555 for unsecure, 55556 for secure)

> **Device Access:** The device can be reached either through its IP address or through its service name, which is printed on the label on the back of the device (e.g., `VZX-abcdef.local`). For HTTP endpoints, this would be `http://VZX-abcdef.local:8990/v1/jsonrpc`.
>
> **Note:** Resolving via service name is only possible on systems where an mDNS resolver is in place. As a fallback, you can always use IP addresses.

> **Important:** The Open Interface must be enabled first via the device's web interface before it can be used.

Available Parameters

The following parameters are accessible through the Open Interface:

Device Parameters

Parameter Name	Read	Write	Subscribe	Type	Description
`device/name`	✓	✓	✓	String	Device name (max 30 characters)
`device/ready`	✓		✓	Boolean	Device ready state (read-only)
`device/error`	✓		✓	Boolean	Collected error state (read-only)
`device/softreset`		✓		Boolean	Trigger device reboot (write `true`)

Zone Parameters

Parameter Name	Read	Write	Subscribe	Type	Description
`device/zone/[1-8]/name`	✓	✓	✓	String	Zone name
`device/zone/[1-8]/level`	✓	✓	✓	Number	Zone output level in dB
`device/zone/[1-8]/mute`	✓	✓	✓	Boolean	Zone mute state

Action Parameters

Start / Stop configured Play Message, Mute Zone or Set GPO actions.

Parameter Name	Read	Write	Subscribe	Type	Description
`device/action/[1-20]/active`	✓	✓	✓	Boolean	Action activation state

Device State Flags (Read-Only)

Device state flags provide detailed system status information. All state flags are **read-only** and can be subscribed to for real-time monitoring.

Parameter Name	Read	Write	Subscribe	Type	Description
`device/stateflag/watchdog-reset`	✓		✓	Boolean	Watchdog reset occurred
`device/stateflag/control-board-revision`	✓		✓	Boolean	Control board revision flag
`device/stateflag/control-board-variant`	✓		✓	Boolean	Control board variant flag
`device/stateflag/audio-board-variant`	✓		✓	Boolean	Audio board variant flag
`device/stateflag/frontboard-revision`	✓		✓	Boolean	Frontboard revision flag
`device/stateflag/frontboard-variant`	✓		✓	Boolean	Frontboard variant flag
`device/stateflag/cpu-temperature-high`	✓		✓	Boolean	CPU temperature high warning
`device/stateflag/cpu-temperature-low`	✓		✓	Boolean	CPU temperature low warning
`device/stateflag/zynq-reset`	✓		✓	Boolean	ZYNQ processor reset flag
`device/stateflag/zynq-errors`	✓		✓	Boolean	ZYNQ processor error flag
`device/stateflag/adc-reference-voltage`	✓		✓	Boolean	ADC reference voltage flag
`device/stateflag/smps-power-good`	✓		✓	Boolean	SMPS power good status
`device/stateflag/smps-temperature-high`	✓		✓	Boolean	SMPS temperature high warning

Parameter Name	Read	Write	Subscribe	Type	Description
device/stateflag/smps-temperature-low	✓		✓	Boolean	SMPS temperature low warning
device/stateflag/fan-failure	✓		✓	Boolean	Fan failure detected
device/stateflag/pg-3-v3-audio	✓		✓	Boolean	3.3V audio power good
device/stateflag/pg-5-v	✓		✓	Boolean	5V power good
device/stateflag/pg-12-v	✓		✓	Boolean	12V power good
device/stateflag/pg-24-v	✓		✓	Boolean	24V power good
device/stateflag/pg-48-v-phantom	✓		✓	Boolean	48V phantom power good
device/stateflag/ext-fuse-trigger	✓		✓	Boolean	External fuse triggered
device/stateflag/vcs08-error	✓		✓	Boolean	VCS-8 hardware error
device/stateflag/vwp-error	✓		✓	Boolean	VWP hardware error
device/stateflag/can-bus-error	✓		✓	Boolean	CAN bus communication error
device/stateflag/audio-mute	✓		✓	Boolean	Hardware audio mute active
device/stateflag/vcs-1-error	✓		✓	Boolean	VCS-8 #1 error
device/stateflag/vcs-2-error	✓		✓	Boolean	VCS-8 #2 error
device/stateflag/vcs-3-error	✓		✓	Boolean	VCS-8 #3 error

Parameter Name	Read	Write	Subscribe	Type	Description
device/stateflag/vcs-4-error	✓		✓	Boolean	VCS-8 #4 error
device/stateflag/vcs-5-error	✓		✓	Boolean	VCS-8 #5 error
device/stateflag/vcs-6-error	✓		✓	Boolean	VCS-8 #6 error
device/stateflag/vcs-7-error	✓		✓	Boolean	VCS-8 #7 error
device/stateflag/vcs-8-error	✓		✓	Boolean	VCS-8 #8 error
device/stateflag/vwp-1-error	✓		✓	Boolean	VWP #1 error
device/stateflag/vwp-2-error	✓		✓	Boolean	VWP #2 error
device/stateflag/vwp-3-error	✓		✓	Boolean	VWP #3 error
device/stateflag/vwp-4-error	✓		✓	Boolean	VWP #4 error
device/stateflag/vwp-5-error	✓		✓	Boolean	VWP #5 error
device/stateflag/vwp-6-error	✓		✓	Boolean	VWP #6 error
device/stateflag/vwp-7-error	✓		✓	Boolean	VWP #7 error
device/stateflag/vwp-8-error	✓		✓	Boolean	VWP #8 error
device/stateflag/vwp-9-error	✓		✓	Boolean	VWP #9 error
device/stateflag/vwp-10-error	✓		✓	Boolean	VWP #10 error

Parameter Name	Read	Write	Subscribe	Type	Description
device/stateflag/vwp-11-error	✓		✓	Boolean	VWP #11 error
device/stateflag/vwp-12-error	✓		✓	Boolean	VWP #12 error
device/stateflag/vwp-13-error	✓		✓	Boolean	VWP #13 error
device/stateflag/vwp-14-error	✓		✓	Boolean	VWP #14 error
device/stateflag/vwp-15-error	✓		✓	Boolean	VWP #15 error
device/stateflag/vwp-16-error	✓		✓	Boolean	VWP #16 error
device/stateflag/player-1-error	✓		✓	Boolean	Media player 1 error
device/stateflag/player-2-error	✓		✓	Boolean	Media player 2 error
device/stateflag/player-3-error	✓		✓	Boolean	Media player 3 error
device/stateflag/player-4-error	✓		✓	Boolean	Media player 4 error
device/stateflag/player-5-error	✓		✓	Boolean	Media player 5 error
device/stateflag/player-6-error	✓		✓	Boolean	Media player 6 error
device/stateflag/player-7-error	✓		✓	Boolean	Media player 7 error
device/stateflag/player-8-error	✓		✓	Boolean	Media player 8 error
device/stateflag/high-cpu-error	✓		✓	Boolean	High CPU usage error

Parameter Name	Read	Write	Subscribe	Type	Description
`device/stateflag/high-flash-error`	✓		✓	Boolean	High flash usage error
`device/stateflag/high-memory-error`	✓		✓	Boolean	High memory usage error
`device/stateflag/dsp-memory-error`	✓		✓	Boolean	DSP memory error
`device/stateflag/dsp-watchdog-error`	✓		✓	Boolean	DSP watchdog error

Data Types

The following data types are used in the Open Interface protocol:

- **Boolean:** Either `true` or `false` (no quotes in TCP/IP, JSON boolean in JSONRPC)
- **String:** Must be enclosed in double quotes. To include a quote character, escape it with a backslash (e.g., `"a string with a\" inside"`)
- **Number:** Numeric values must use a period (`.`) as the decimal separator (e.g., `-42.0`)
- **Enum:** String values from a predefined set, must be enclosed in double quotes

> **Value Clamping:** Numeric values outside the acceptable range will be automatically clamped to the valid range. No error response will be issued in this case.

TCP/IP Protocol

The TCP/IP protocol utilizes port **8989** for communication.

Protocol Characteristics

- Both commands and responses are delimited by newline characters (`\n` or `\r\n`)
- The protocol supports both Unix-style (`\n` , 0x0a) and Windows-style (`\r\n` , 0x0d 0x0a) line endings
- Every command results in a corresponding response message
- Subscriptions trigger additional asynchronous messages
- A persistent connection is required for subscriptions
- Invalid commands generate error messages (responses beginning with `error`)

Commands

- `get <parameter-path>` - Retrieve current value
- `set <parameter-path> <value>` - Set new value
- `subscribe <parameter-path>` - Subscribe to value changes
- `unsubscribe <parameter-path>` - Unsubscribe from value changes

Examples

Retrieve Device Name

```
Client → Server: get device/name\n
Server → Client: device/name "VZX-deadbeef"\n
```

Set Device Name

```
Client → Server: set device/name "myname"\n
Server → Client: ok\n
```

Mute Zone 1

```
Client → Server: set device/zone/1/mute true\n
Server → Client: ok\n
```

Set Zone 1 Level

```
Client → Server: set device/zone/1/level -42.0\n
Server → Client: ok\n
```

Subscribe to Zone 1 Mute

```
Client → Server: subscribe device/zone/1/mute\n
Server → Client: ok\n

// Whenever the mute changes, the server responds:
Server → Client: device/zone/1/mute true\n
```

Read State Flag

```
Client → Server: get device/stateflag/cpu-temperature-high\n
Server → Client: device/stateflag/cpu-temperature-high false\n
```

Subscribe to State Flag

```
Client → Server: subscribe device/stateflag/fan-failure\n
Server → Client: ok\n

// When the fan failure state changes:
Server → Client: device/stateflag/fan-failure true\n
```

JSONRPC/HTTP/IP Protocol

The JSONRPC protocol utilizes port **8990** for communication. It follows the JSON-RPC 2.0 standard.

Protocol Characteristics

- Endpoint: `http://<device-ip>:8990/v1/jsonrpc`
- HTTP POST method with JSON payload
- Subscriptions are **not possible** through this protocol (HTTP is not persistent)
- Each request must include a unique `id` field

Methods

- `get` - Retrieve current value
- `set` - Set new value

Examples

Retrieve Device Name

```
Request:
POST http://VZX-abcdef.local:8990/v1/jsonrpc
Content-Type: application/json

{
  "jsonrpc": "2.0",
  "method": "get",
  "params": {
    "key": "device/name"
  },
  "id": "random-id"
}

Response:
{
  "jsonrpc": "2.0",
  "result": "VZX-deadbeef",
  "id": "random-id"
}
```

Mute Zone 1

Request:

```
{
  "jsonrpc": "2.0",
  "method": "set",
  "params": {
    "key": "device/zone/1/mute",
    "value": true
  },
  "id": "another-random-id"
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": "ok",
  "id": "another-random-id"
}
```

Set Zone 1 Level

Request:

```
{
  "jsonrpc": "2.0",
  "method": "set",
  "params": {
    "key": "device/zone/1/level",
    "value": -42.0
  },
  "id": "another-random-id2"
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": "ok",
  "id": "another-random-id2"
}
```

Read State Flag

Request:

```
{
  "jsonrpc": "2.0",
  "method": "get",
  "params": {
    "key": "device/stateflag/smps-power-good"
  },
  "id": "check-power"
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": "check-power"
}
```

This document describes the Open Interface protocol for VZX-8 8-Zone Audio Processor.
For additional support, refer to the device's web interface or contact technical support.