Introduction

This document describes the L-ISA Processor REST API, which can be used to observe and affect the processor state through remote clients.

Connectivity

The L-ISA Processor REST API is available over the HTTP protocol on port 80: http://*processor-IP-address/endpoint*.

Various functionalities are available over a set of *endpoints* which are part of the URL. The HTTP access method may also vary depending on functionality.

Encoding

The data exchanged between the processor and its clients over this API is encoded in JSON format (with some exceptions). When responding to a request with a JSON data, the server generates the corresponding content type HTTP header: `Content-Type: application/json`.

A client produces the same header in its requests when sending a JSON data, otherwise the server responds with an error.

Errors

Whenever the server accepts and successfully processes a request, it responds with an HTTP status code `200 OK`. In case of an error, the server responds with an error code with no additional data returned in the response body.

Code	Status	Description
200	OK	No error.
400	Bad Request	The server cannot process the received request. The request is malformed or does not contain all the necessary data in it.
403	Forbidden	The request is not allowed at the current state of the processor. For example, the client has not obtained the processor's lock or the client provided an invalid authentication data.
404	Not Found	Unrecognized URL or unrecognized endpoint.
405	Internal Error	The server encountered an internal error while processing the request.

Any HTTP status code (apart from `200 OK`) must be treated as an error by the client.

Endpoints

Processor information

Method	URL path
GET	/api/processor

This endpoint returns the processor information status, including the audio interface and processor hardware status.

On success the server responds with a JSON object containing the following fields:

JSON field name	Type	Description
hw_version	String	Processor type: PR1001 (L-ISA Processor) or PR1003 (L-ISA Processor II)
fw_version	String	Firmware version string
processor_id	String	Processor ID string
display_name	String	Processor display name
mac_address	String	Processor network interface MAC address, for example 01:23:45:67:89:AB. This field is reported only for the L-ISA Processor II.
hardware	Object	Hardware status information (see below)
boot	Object	Cold boot behavior (see below)

The **hardware** field corresponds to the objects returned by specialized endpoint/processor/hardware as described below.

Processor hardware information

Method	URL path
GET	/api/processor/hardware

This endpoint returns the processor hardware status information:

JSON field name	Type	Description
psu1	Boolean	PSU1 health status: `true/false` = healthy/failure
psu2	Boolean	PSU2 health status: `true/false` = healthy/failure
disk	String	Disk health status: `ok`, `fail`, `unknown`
filesystem	String	Filesystem health status: `ok`, `fail`, `unknown`
emergency_key	Boolean	When true, indicates the processor has booted from an emergency key
cpu_fan_speed	Number	CPU fan speed in RPM
case_fan_speed	Number	Case fan speed in RPM
psu_fan_speed	Number	PSU fan speed in RPM
cpu_temperature	Number	CPU temperature in celsius degrees
case_temperature	Number	Case temperature in celsius degrees

Cold boot behavior

Method	URL path
GET	/api/processor/boot

This endpoint returns the processor cold boot behavior information:

JSON field name	Type	Description
cold_boot_behavior	Number	Cold boot behavior code: • `-1`: Boot muted • `-2`: Boot into the last active configuration • `0/.../19`: Boot into the specific configuration
cold_boot_behavior_desc	String	Textual description of the cold boot behavior code

Configurations

Method	URL path
GET	/api/configs

This method returns the array of configuration slots. Each item of the returned array is a JSON object that represents the corresponding configuration. The configuration is stored in the slot that matches the array item index: `(0/.../19)`

Accessing specific configuration

A particular configuration can be accessed by its index `(0/.../19)` put into the URL path:

Method	URL path
GET	/api/configs/*index*

For example: GET http://*processor-IP-address*/api/configs/3 returns the JSON object that represents the configuration in slot 3.

The configuration JSON object has the following fields:

JSON field name	Type	Description
index	Number	Configuration slot index `(0/.../19)`
valid	Boolean	Configuration validity: `true/false` = valid/empty
locked	Boolean	Lock status: `true/false` = locked/unlocked
active	Boolean	Active state: `true/false` = active/inactive
name	String	Configuration name
version	String	Processor software version when the configuration had been stored
timestamp	String	Configuration modification date/time in ISO8601 format (YYYY-MM-DDThh:mm:ss.ss, for example 2022-03-09T11:58:12.05)

For an invalid index, the server returns a `404 Not Found` status code.

Recalling the processor state

Method	URL path
GET	/api/configs/*index*/recall

This endpoint loads the processor state.

If there is a controller currently connected to the processor, the recall fails with a `403 Forbidden` status code response.

Shutdown

Method	URL path
POST	/api/processor/shutdown

This endpoint triggers the processor shutdown. The client must be locked to the processor (via the OSC channel). If not connected the server responds with a `403 Forbidden` status code.

The server expects an empty JSON object in this request coming from a client.

Wake-On-LAN (WoL)

The L-ISA Processor II supports Wake-on-LAN, with no password. Any WoL client that can send the magic packet containing the MAC address of the processor is able to wake the L-ISA Processor II.