3 IO-Link Ports

Application Manual

www.pinetek-networks.com

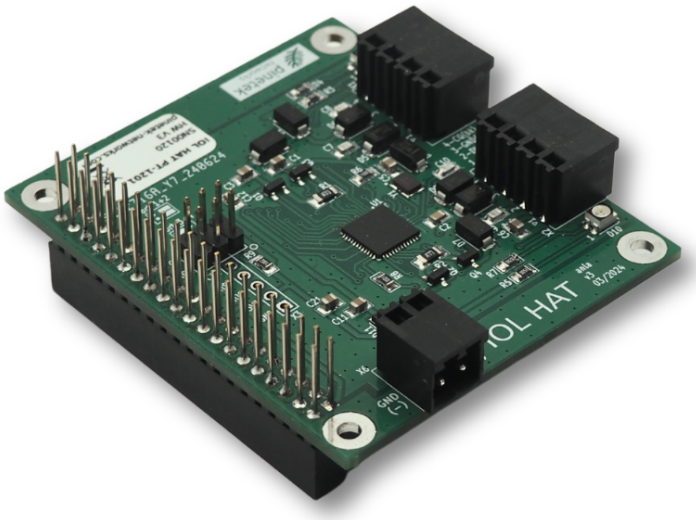

Table of Contents

3 IO-Link Ports

<port> is the **0-based** port index: 0 … 3.

These endpoints control port mode and power, handle IODD files, and provide access to process data and device parameters.

Port Status and Control

Status

`GET /<port>/device/status`

Returns the current status of an IO-Link port.

Response

```
{
  "port_mode": 2,
  "port_mode_string": "sdci",
  "port_mode_string_loc": "...",
  "port_mode_string_loc_long": "...",
  "power": 1,
  "power_string": "on",
  "power_string_loc": "...",
  "power_string_loc_long": "...",
  "power_on_enable": true,
  "power_off_enable": false,
  "comm": 1,
  "comm_string": "active",
  "comm_string_loc": "...",
  "comm_string_loc_long": "...",
  "error": 0,
  "error_string": "no_error",
  "error_string_loc": "...",
  "error_string_loc_long": "...",
  "error_data": 0,
  "error_data_string": "...",
  "error_data_string_loc": "...",
  "error_data_string_loc_long": "...",
  "vendor_id": 1234,
  "device_id": 5678
}
```

Field	Type	Description
port_mode	int	Active port mode (see mode table below)
port_mode_string	string	Mode as string (e.g. sdci, di, do, off)
power	int	1 = power on, 0 = power off
power_on_enable	bool	Whether power-on is currently allowed
power_off_enable	bool	Whether power-off is currently allowed
comm	int	1 = communication active
comm_string	string	Communication status string
error / error_data	int	Error codes; 0 = no error

Field	Type	Description
`vendor_id`	int	Connected device vendor ID
`device_id`	int	Connected device device ID

Set Mode

`GET /<port>/device/setmode/<mode>`

Sets the IO-Link port operating mode.

Response: Error envelope

<mode>	String	Description
`0`	`sdci`	IO-Link (SDCI) communication — default
`1`	`di`	Digital Input
`2`	`do`	Digital Output
`3`	`off`	Port disabled

Power On

`GET /<port>/device/poweron`

Enables 24 V supply power on the port.

Response: Error envelope

Power Off

`GET /<port>/device/poweroff`

Disables supply power on the port.

Response: Error envelope

Set Power

`GET /<port>/device/power/<state>`

Convenience alternative to the dedicated `poweron` / `poweroff` endpoints.

- `<state>` = `1`: power on
- `<state>` = `0`: power off

Response: Error envelope

IODD Handling

IODD Info

`GET /<port>/device/iodd`

Returns IODD file information for the device connected to the port.

Only meaningful when the port is in IO-Link (`sdci`) mode; returns an error otherwise.

Response

```
{
  "error": 0,
  "iodd_name": "...",
  "iodd_language": "en",
  "vendor_id": 1234,
  "vendor_name": "Vendor Inc.",
  "device_id": 5678,
  "device_count": 2,
  "device_options": [
    { "device_name": "Model-A", "device_symbol": "..." }
  ],
  "device_selected": 0
}
```

Field	Description
`iodd_name`	Name of the active IODD file
`iodd_language`	Language code of the IODD
`vendor_id` / `vendor_name`	Device vendor information
`device_id`	IO-Link device ID
`device_count`	Number of device variants in the IODD
`device_options`	Array of available device variants
`device_selected`	Index of the currently active variant

IODD Files

`GET /<port>/device/iodd/files`

Returns a list of all IODD files uploaded for this port. Useful when an IODD ZIP contains multiple device definitions.

Response: Array of IODD file descriptors (structure depends on stored files).

Selected IODD

`GET /<port>/device/iodd/selected`

Returns the index of the currently active IODD variant.

Select IODD

`GET /<port>/device/iodd/selected/set/<index>`

Selects the active IODD variant by index.

Response: Error envelope

Upload IODD

`POST|PUT /<port>/upload/<filename>`

Uploads an IODD file (ZIP or plain XML) for the port. `<filename>` is used as the stored filename on the device.

Content-Type: `application/octet-stream`

Response: Error envelope

Process Data

PD Info

`GET /<port>/pd-info`

Returns process-data variable descriptors derived from the loaded IODD. This is static metadata; it does not trigger IO-Link communication.

Response

```
{
  "error": { "error": 0, "..." },
  "info": {
    "pd_in": [ { "name": "...", "datatype": "...", "..." } ],
    "pd_out": [ { "name": "...", "datatype": "...", "..." } ]
  }
}
```

Use the `pd_in` / `pd_out` descriptor arrays to understand the structure before reading or writing process data.

Write PD

`POST|PUT /<port>/pd`

Writes PD-out values to the device and returns the current PD-in and PD-out values.

Request body

The `value` array must match the order and count of variables returned by `pd-info` for `pd_out`.

```
{
  "value": [
    { "value": "1" },
    { "value": "250" }
  ]
}
```

Response

```
{
  "error": { "error": 0, "..." },
  "value": {
    "pd_in":  [ { "name": "...", "value": "...", "..." } ],
    "pd_out": [ { "name": "...", "value": "...", "..." } ]
  }
}
```

Parameters

Parameter Tree

`GET /<port>/parameters`

Returns the full parameter tree from the IODD in a menu/sub-menu structure with variable descriptors. Useful for building a parameter browser UI.

Response

```
{
  "menu_count": 3,
  "menus": [ { "..." } ]
}
```

Get Parameter

`GET /<port>/parameter/get/<index>`

`GET /<port>/parameter/get/<index>/<subindex>` (for record parameters)

Reads a single parameter by IODD index.

Response

```
{
  "error": { "error": 0, "..." },
  "info": {
    "name": "Temperature",
    "index": 96,
    "subindex": 0,
    "datatype": "Float32",
    "access": "ro"
  },
  "value": { "value": 23.4 }
}
```

Field	Description
`info.name`	Parameter name from IODD
`info.index`	IODD parameter index
`info.subindex`	Sub-index (0 for scalar parameters)
`info.datatype`	Data type string (e.g. `Float32`, `UInt8`)
`info.access`	Access level: `ro`, `rw`, or `wo`
`value.value`	Current parameter value

Set Parameter

`POST|PUT /<port>/parameter/set/<index>`

`POST|PUT /<port>/parameter/set/<index>/<subindex>` (for record parameters)

Writes a single parameter by IODD index.

Request body

```
{ "value": "42" }
```

Response: Same structure as `Get Parameter`, reflecting the written-back value.