Additional	Information		
Stock No.	X9751375.01	Order No.	
Product	Interface Instructions for colorSENSOR CFO100 CFO100(100) CFO200 CFO200(100)	B-Order No.	MICRO-EPSILON
Main Operating Instructions		colorSENSOR CF0	O – X9751375

Interface Instructions for the colorSENSOR CFO

Inhalt

REST-A	API communication with the controller	1
1.1 B	Basic informations	1
1.1.1	API Endpoint	1
1.1.2	Examples for manual query	1
1.1.3	Query example	2
1.1.4	Quickstart via API communication	3
1.1.5	REST-API Introduction	4
1.1.5	5.1 General Informations	4
1.1.5	5.2 Response Format and Error Handling	4
1.1.5	5.3 Switching outputs, triggers and hold time settings	5
1.1.5	5.4 Websockets	7
1.1.5	5.5 Networking and Discovery	8
1.2 R	Resources	10
1.2.1	Device Information	10
1.2.	1.1 Get Device Properties	10
1.2.2	Sensor	10
1.2.2	2.1 Retrieve Sensor Samples	10
1.2.2	2.2 Retrieve latest Sensor Sample	13
1.2.2	2.3 Retrieve all Matchers (color groups)	15
1.2.2	2.4 Create a new Matcher (color group)	17
1.2.2	2.5 Remove multiple or all ColorMatchers	21
1.2.2	2.6 Retrieve Matcher (color group) Details	22
1.2.2	2.7 Delete a Matcher (color group)	24
1.2.2	2.8 Update the Matcher (color group) Configuration	24
1.2.2	2.9 Retrieve ColorDetectables	28
1.2.2	2.10 Remove multiple or all ColorDetectables	29
1.2.2	2.11 Create ColorDetectables	29
1.2.2	2.12 Delete ColorDetectable)	31
1.2.2	2.13 Modify ColorDetectable	31
1.2.2	2.14 Get ColorDetectable	33
1.2.2	2.15 Switch current Detection Profile	34
1.2.2	2.16 Create DetectionProfiles	35
1.2.2	2.17 Remove multiple or all DetectionProfiles	43
1.2.2	2.18 Retrieve DetectionProfiles	43
1.2.2	2.19 Delete DetectionProfile	48
1.2.2	2.20 Modify DetectionProfile	48
1.2.2	2.21 Get DetectionProfile	57
1.2.2	2.22 Start Autogain Procedure	62

1.2	2.2.23	Query custom White Reference	64
1.2	2.2.24	Sample a custom White Reference	65
1.2	2.2.25	Lese Aktionsauslöser aus	66
1.2	2.2.26	Remove multiple or all Action Triggers	67
1.2	2.2.27	Create ActionTriggers	67
1.2	2.2.28	Delete ActionTrigger	68
1.2	2.2.29	Modify ActionTriger	69
1.2	2.2.30	Get ActionTrigger	70
1.2	2.2.31	Get sensoric Capabilities	71
1.2	2.2.32	Retrieve Colorspaces	74
1.2	2.2.33	Get Colorspace	74
1.2.3	Setting	gs	76
1.2	2.3.1	Export Settings	76
1.2	2.3.2	Upload Settings	76
1.2	2.3.3	Import Settings	77
1.2	2.3.4	Reset Settings	77
1.2.4	Systen	n	77
1.2	2.4.1	Request System Settings	77
1.2	2.4.2	Modify System Settings	78
1.2	2.4.3	Reset to Factory Firmware and Settings	78
1.2	2.4.4	Initiate Reboot	79
1.2	2.4.5	Get time settings	79
1.2	2.4.6	Change time settings	79
1.2	2.4.7	Retrieve supported Timezones.	80
1.2.5	Netwo	rk	81
1.2	2.5.1	Reset Network Settings	81
1.2	2.5.2	Retrieve status and configuration of all network interfaces	81
1.2	2.5.3	Retrieve status and configuration of a single network interface	83
1.2	2.5.4	Modify IPv4 and/or IPv6 address configuration	84
1.2.6	Periph	erals	88
1.2	2.6.1	Get Keypad Information	88
1.2	2.6.2	Modify Keypad	89
1.2	2.6.3	Retrieve user interactions on the keypad	90
1.2	2.6.4	Retrieve a list of available input elements on the keypad	90
1.2	2.6.5	Simulate a user interaction on the keypad	91
1.2	2.6.6	Get Output Configuration	91
1.2	2.6.7	Modify Output Configuration	92
1.2	2.6.8	Get current interface configuration	92
1.2	2.6.9	Modify interface configuration	93

1.2.6.10	Get trigger source event statistics	94
1.2.6.11	Get current interface configuration	94
1.2.6.12	Modify interface configuration	95
1.2.7 Actio	ons	96
1.2.7.1	Action descriptions	96
1.2.7.2	Action "enable_switching_output"	96
1.2.7.3	Action "teach_single"	96
1.2.7.4	Action "keylock"	96
1.2.7.5	Action "run_autogain"	97
1.2.7.6	Action "remove_all_detectables"	97
1.2.7.7	Action "remove_all_matchers"	97
1.2.7.8	Retrieve Actions	97
1.2.7.9	Get Action	97
1.2.7.10	Software-triggered Actions	98
1.2.8 Defa	aults	103
1.2.8.1	Retrieve DefaultMapValues	104
1.2.8.2	Create DefaultMapValues	105
1.2.8.3	Remove multiple oder all DefaultMapValues	106
1.2.8.4	Modify DefaultMapValue	106
1.2.8.5	Delete DefaultMapValue	108
1.2.8.6	Get DefaultMapValue	108
1.2.9 Firm	ware	109
1.2.9.1	Get Firmware Information	109
1.2.9.2	Firmware Image Upload	109
1.2.9.3	Upload Firmware	110
1.2.9.4	Firmware Upload methods	110
1.2.9.5	Error handling	111
1.2.9.6	Get firmware image upload	112
1.2.9.7	Delete firmware image upload	113
1.2.9.8	Switch to new Firmware	113
1.2.9.9	Upload Chunk of Firmware	113
1.2.9.10	Get Recovery Firmware Information	114
1.2.9.11	Upgrade Recovery Firmware	115
1.2.9.12	Get Firmware Settings	115
1.2.9.13	/firmware/settings	116
1.2.9.14	Get Firmware Status	116
1.2.10 A	ccess Control	117
1.2.10.1	Users	117
1.2.10.2	Roles	117

1.2.	10.3	Auswertung der Zugriffsberechtigungen	117
1.2.	10.4	HTTP Responses	117
1.2.	10.5	Inspect Access Control Scopes and Actions	118
1.2.	10.6	Login into an account	118
1.2.	10.7	Retrieve Information about the currently used Credentials	120
1.2.	10.8	Logout / Invalidate any current credentials	121
1.2.	10.9	Create User	121
1.2.	10.10	Remove multiple or all Users	122
1.2.	10.11	Retrieve User	122
1.2.	10.12	Delete User	123
1.2.	10.13	Modify User	123
1.2.	10.14	Get User	124
1.2.	10.15	Retrieve AccessRoles	125
1.3 R	EST-AF	PI Type Reference	126
1.3.1	Acces	sAction	126
1.3.2	Acces	sControlFeatures	126
1.3.3	Acces	sPermission	126
1.3.4	Acces	sRole	126
1.3.5	Acces	sScope	127
1.3.6	Action	l	127
1.3.7	Action	EnableSwitchingOutput	127
1.3.8	Action	KeyLock	127
1.3.9	Action	RemoveAllDetectables	128
1.3.10	Acti	onRemoveAllMatchers	128
1.3.11	Acti	onResultEnableSwitchingOutput	128
1.3.12	Acti	onResultKeyLock	130
1.3.13	Acti	onResultRemoveAllDetectables	130
1.3.14	Acti	onResultRemoveAllMatchers	130
1.3.15	Acti	onResultRunAutogain	131
1.3.16	Acti	onResultTeachDetectable	131
1.3.17	Acti	onRunAutogain	132
1.3.18	Acti	onTeachDetectable	132
1.3.19	Acti	onTrigger	133
1.3.20	Alia	s	134
1.3.21		plificationLevel	
1.3.22	Any	Action	134
1.3.23	Any	ActionResult	134
1.3.24	Auto	ogainSettings	134
1.3.25	Ave	rageSampleCount	135

1.3.26	BaseColorTolerance	135
1.3.27	BaseSerialSettings	135
1.3.28	BoxColorTolerance	135
1.3.29	ChromaticityCoordinate	135
1.3.30	ColorDetectable	135
1.3.31	ColorDetectionResult	136
1.3.32	ColorDetectionResultList	139
1.3.33	ColorDetectionResultOrNil	139
1.3.34	ColorMatcher	139
1.3.35	ColorMatchingResult	141
1.3.36	ColorRepresentations	142
1.3.37	Colorspace	142
1.3.38	ColorspaceAxis	143
1.3.39	ColorspaceID	143
1.3.40	ColorspaceToleranceMap	143
1.3.41	ColorTolerance	144
1.3.42	CompensationSettings	144
1.3.43	CorrectedColor	145
1.3.44	CurrentDetectionProfileID	145
1.3.45	CurrentSwitchingOutputsState	145
1.3.46	CylinderColorTolerance	145
1.3.47	Standardwertepaare	145
1.3.48	DetectionProfile	146
1.3.49	DeviceInformation	150
1.3.50	DeviceSerialNumber	151
1.3.51	DeviceVendorKey	151
1.3.52	DeviceVendorName	151
1.3.53	Error	151
1.3.54	FirmwareBuildId	151
1.3.55	FirmwareImageFile	152
1.3.56	FirmwareImageSize	152
1.3.57	FirmwareImageUpload	152
1.3.58	FirmwareInformation	153
1.3.59	FirmwareRecoveryInformation	153
1.3.60	FirmwareRunningInformation	153
1.3.61	FirmwareSettings	154
1.3.62	FirmwareVersion	154
1.3.63	HashDigest	154
1.3.64	HoldTime	154

1.3.65	Hostname	155
1.3.66	InfiniteColorTolerance	155
1.3.67	InputsState	155
1.3.68	InterfaceRS232	155
1.3.69	InterfaceUSB	155
1.3.70	KeypadEvent	156
1.3.71	KeypadEventInput	156
1.3.72	KeypadEventName	156
1.3.73	KeypadIndicator	156
1.3.74	KeypadInformation	157
1.3.75	KeypadInputButton	157
1.3.76	LoginInformation	158
1.3.77	MacAddress	158
1.3.78	NetworkAddressConfigurationIPv4	159
1.3.79	NetworkAddressConfigurationIPv4DHCP	159
1.3.80	NetworkAddressConfigurationIPv4Static	159
1.3.81	NetworkAddressConfigurationIPv6	159
1.3.82	NetworkAddressConfigurationIPv6Auto	159
1.3.83	NetworkAddressConfigurationIPv6DHCP	159
1.3.84	NetworkAddressConfigurationIPv6Static	159
1.3.85	NetworkAddressIPv4	160
1.3.86	NetworkAddressIPv6	160
1.3.87	NetworkInterfaceAddressConfigurationInformation	160
1.3.88	NetworkInterfaceAddressConfigurationIPv4	161
1.3.89	NetworkInterfaceAddressConfigurationIPv6	161
1.3.90	NetworkInterfaceAddressConfigurationState	161
1.3.91	NetworkInterfaceAddressesIPv4	163
1.3.92	NetworkInterfaceAddressesIPv6	163
1.3.93	NetworkInterfaceAddressFamilyInformationIPv4	163
1.3.94	NetworkInterfaceAddressFamilyInformationIPv6	164
1.3.95	NetworkInterfaceAddressFamilyStateIPv4	165
1.3.96	NetworkInterfaceAddressFamilyStateIPv6	165
1.3.97	NetworkInterfaceAddressIPv4	166
1.3.98	NetworkInterfaceAddressIPv6	166
1.3.99	NetworkInterfaceInformation	166
1.3.100	NetworkInterfaceName	168
1.3.101	NetworkInterfaceStaticData	168
1.3.102	NormalizationConstant	168
1.3.103	ReleaseChannel	169

	1.3.104	SampleRate	169
	1.3.105	SamplingSettings	169
	1.3.106	SensorCapabilities	170
	1.3.107	SerialElizaSettings	176
	1.3.108	SerialModbusSettings	177
	1.3.109	SignalColor	177
	1.3.110	SphereColorTolerance	177
	1.3.111	SupportedTimezones	177
	1.3.112	SwitchingOutputDriver	177
	1.3.113	SwitchingOutputs	177
	1.3.114	SwitchingOutputsWritable	178
	1.3.115	SystemSettings	178
	1.3.116	SystemTimeSettings	178
	1.3.117	Timestamp	179
	1.3.118	TimestampBackendUptime	179
	1.3.119	ToleranceShapeName	179
	1.3.120	TransformedColor	179
	1.3.121	TriggerEvent	179
	1.3.122	TriggerEventName	179
	1.3.123	TriggerSource	180
	1.3.124	TriggerSourcesStatus	180
	1.3.125	User	181
	1.3.126	UUID	181
	1.3.127	WantedSwitchingOutputsState	181
	1.3.128	WhiteReference	182
	1.3.129	WrappedNetworkInterfaceAddressIPv4	182
	1.3.130	WrappedNetworkInterfaceAddressIPv6	182
2	Termin	al Documentation	183
2	2.1 P	reface	183
	2.1.1	Connection Details	183
	2.1.2	Syntax	183
	2.1.3	Output Formats & Message Parsing	184
	2.1.4	Nomenclature	184
	2.1.5	Common Patterns	184
	2.1.6	Differences compared to the REST API	184
2	2.2 C	ommand Reference	186
	2.2.1	Access Command	186
	2.2.2	Device Command	186
	2.2.3	Firmware Command	186

	2.2.4	Help Command1	86
	2.2.5	Keypad Command1	87
	2.2.6	Matcher Command	87
	2.2.7	Network Command	89
	2.2.8	Repeat Command	89
	2.2.9	Sample Command1	90
	2.2.10	Sensor Command1	90
	2.2.11	SET Command 1	90
	2.2.12	System Command1	91
3		us Documentation 1	
	3.1 lı	ntroduction 1	92
		Quickstart1	
	3.3	Supported colorSENSOR Features1	94
	3.4	Data Types and Register Addressing1	
	3.4.1	Data Types and Modbus Functions	
	3.4.2	Register Addresses	
	3.4.3	Simple Data Types	
	3.5	Session State, Concurrency and Multiple Interfaces	
	3.5.1	Functions1	97
	3.5.	.1.1 Autogain Procedure API Endpoint: /sensor/detection- ofiles/current/autogain1	107
	3.5.		191
		ofiles/current/white-reference1	197
	3.5.	.1.3 Add Color to Color Table API Endpoint: /sensor/matchers 1	197
	3.5.	1.4 Manage Color Positions of a Color Group API Endpoint: /sensor/matchers1	97
	3.5.	.1.5 Read Sensor Capabilities API Endpoint: /sensor/capabilities	198
	3.5.	.1.6 Get Current Sample API Endpoint: /sensor/samples/current	198
	3.5.		
	pro	ofiles/current1	199
	3.5.	1.8 Clear Color Table API Endpoint: /sensor/matchers	199
	3.5.	1.9 Switching Outputs Driver API Endpoint: /peripherals/outputs	199
	3.5.	.1.10 Firmware Version API Endpoint: /firmware	199
	3.5.	1.11 Settings Reset API Endpoint: /settings	200
	3.5.	.1.12 Factory Reset API Endpoint: /system/factory-reset	200
	3.5.	1.13 Reboot the Device API Endpoint: /system/reboot	200
	3.5.	.1.14 Upgrade Recovery Firmware API Endpoint: /system/factory-reset2	200
	3.5.	1.15 RS232 Interface Configuration API Endpoint: /peripherals/rs232	200
	3.5.	.1.16 USB Interface Configuration API Endpoint: /peripherals/usb	200
	3.5.	.1.17 Data Format Test API Endpoint: None	201

1 REST-API communication with the controller

1.1 Basic informations

The HTTP-based API 1 enables the controller configuration or current states to be queried and changed.

It is the primary interface, via which all details of the controller are accessible. For example, the integrated web application, RS232, USB, MEDAQULib and Modbus also communicate with the controller exclusively via the API.

The API is REST-like and supports the following HTTP requests to distinguish actions with different effects:

- GET

 Status of a resource (e.g. a current sample); GET queries are free of side effects and are sometimes saved in cache memory.

- POST

Create a new resource (e.g. teach in a new colour or colour group). Erzeugen einer neuen
 Ressource (z.B. eine neue Farbe oder Farbgruppe einlernen).

- PUT

• Changing a resource (e.g. changing a tolerance in a colour group / matcher

- DELETE

- Deleting, resetting or emptying a resource (e.g. deleting a colour or resetting to the factory setting).
 - 1) Version 1.5.10

1.1.1 API Endpoint

The API endpoints are to be understood here as an addition to the basic URL (http://). As an example, the browser call (GET-request) for the colourSENSOR CFO to query the initial configuration is mentioned here:

http://169.254.168.150/api/peripherals/outputs

Basis-URL | API-Endpunkt

http://{sensor}/api

sensor	Hostname or IP adress of he controller		
String, required			

1.1.2 Examples for manual query

The following manual queries are only intended to illustrate the exemplary use of the API.

For integration into own tools, most programming languages and development environments typically provide HTTP programming interfaces for easy access.

The following recommended tools for API-querying are not mandatory, but can be used.

Environment	Tool recommendation
Command line (CLI)	cURL
Windows PowerShell	Invoke-RestMethod
Chrome/Chromium	Postman

1.1.3 Query example

},

Every web browser uses GET requests to retrieve content. With tools, it is easy to send requests with other HTTP verbs as well. In application to the colourSENSOR CFO, for example, the request results in

```
curl -X GET http://169.254.168.150/api/sensor/samples/current
the following response in form of a JSON-object:
 "data": {
  "timestamp": 3145601368.0,
  "uuid": "0f721d83-f3c0-4584-8913-a51a5b842784",
  "transformed color": {
   "values": [ 99.953887939453125, -0.0064074993133544922,
0.017380714416503906 ]
  },
  "corrected color": {
   "values": [ 0.79777300357818604, 0.74252212047576904,
0.28755432367324829 ]
  },
  "detection": {
   "matcher": null,
   "output pattern": { "states": [ false, false, false ] },
   "distances": [ null, null, null ]
  },
  "representations": {
   "RGB": [ 0.9994870320649919, 0.99951960105113213, 0.99927028464270895 ]
  },
  "inputs": {
   "trigger 0 down": true,
   "trigger 1 down": true,
   "trigger_2_up": false,
   "trigger_0_up": false,
   "trigger 3 down": true,
   "trigger 3 up": false,
   "trigger 2 down": true,
   "trigger 1 up": false
  }
```

```
"errors": []
}
```

1.1.4 Quickstart via API communication

The following sequence of requests configures the sensor for the detection of a wanted color.

The request examples below are sent via the curl command (examples prefixed with \$) or the Windows Powershell (prefixed with >).

The same requests can be used in all other programming languages or environments supporting HTTP requests, as well.

1. Clear all settings:

```
$ curl -X DELETE http://sensor/api/settings
```

- > Invoke-RestMethod -Method DELETE -Uri http://<sensorIP>/api/settings
 - 2. Place a neutral white color target in front of your sensor's optics.
 - 3. Adjust the sensor settings to the optical setup by initiating the *Autogain* procedure:

```
$ curl -X POST http://<sensorIP>/api/sensor/detection-profiles/current/au-
togain
```

- > Invoke-RestMethod -Method POST -Uri http://<sensorIP>/api/sensor/detection-profiles/current/autogain
 - 4. Place your wanted target object in front of the sensor's optics.
 - 5. Teach this color:

```
$ curl -X POST http://<sensorIP>/api/sensor/detectables
> Invoke-RestMethod -Method POST -Uri http://<sensorIP>/api/sensor/detecta-
```

- 6. The sensor should now pull up its first output pin as long as the target is placed in front of it.
 - The first switching output pin is assigned to the first color, by default. This is configurable.
- 7. Request the current sample data and pick only the color values:

```
# hint: "jq" is a separate tool for querying JSON datasets
$ curl http://<sensorIP>/api/sensor/samples/current | jq .data.trans-
formed_color.values
> (Invoke-RestMethod -Method GET -Uri http://<sensorIP>/api/sensor/sam-
ples/current).data.transformed color
```

Result:

```
[ 101.23353576660156, 7.889449596405029, -42.49897003173828 ]
```

1.1.5 REST-API Introduction

1.1.5.1 General Informations

Collections (arrays or lists)

- can be gueried via GET request
- some collections allow to be filtered via query parameters, outlined in the resource documentation below
- some collections allow POST requests to create new items in the collection
 - properties of new items can usually be set in the POST request body
 - a failure to create a new item due to malformed data is indicated with a 400 HTTP status code
 - a failure to create a new item caused by too many items in the collection is indicated by a 422 HTTP status code
 - some collections can be depleted by sending a DELETE request to them supporting
 the same query parameters as GET requests if any
 - a DELETE request on the collection always returns a positive HTTP status code like 200 or 204 with no regard for the number of items that have been deleted (and even if no items have been deleted as a result of the request)

Collection Items

- Individual collection items typically can be accessed via a URL path like /collection-path/item-id (e.g. /api/access/users/barbara, where /users is the collection and barbara the item id).
 - If no item with that specific id exists, either because it has been deleted in the meantime in response to a user request or (in case of size- or time-limited collections) the item was removed, the request will be terminated with a 404 HTTP status code.
 - The data of some collections items can be updated with PUT requests where the
 body contains a JSON object with the data that should be updated. Partial updates
 are supported as well. Changing read-only attributes may be terminated with a 400
 HTTP status code. The response for a successful update contains the state of the resource after the modification.
 - The unique id of an item (most often a uuid) is invariable.
 - Some collection items can be removed with a DELETE request.
- Some collections support more than one URL-compatible item id field. The matcher, detectable and detection profile collection items for example, can also be retrieved by their respective alias field.

Data Formatting Conventions

- Identifiers are in American English
- Compound words in resource urls are hyphenated (e.g. white-reference)
- Compound words in JSON objects or data accepted as multipart/form-data uses snake case (e.g. white reference)

1.1.5.2 Response Format and Error Handling

The API will return a response containing a body with a JSON object for any given request to any known endpoint of the following form:

colorSENSOR CFO 4

{

```
"errors": [ ...errors ],

"data": { ...data }
```

data contains the actual response payload that has been requested whereas errors contains any errors that the API encountered while processing the request. An error object found in the errors array has the following form:

```
"message": String,
"mapping": String | null,
"code": String
```

Attribute	Туре	Content
message	String	The message property contains a human-readable english description of the error. Its primary target audience is the developer working with the API and should guide them on their way to resolve the error.
mapping	String / null	The mapping property contains a valid JavaScript expression as a string that evaluates to the property on the submitted or provided object where the error was encountered. The mapping may be null if no such connection can be made. If it is set an expression like foo[1].bar would refer to the bar property in the second item of an array named foo found on the root object.
code	String	The code property can be used as an identifier to distinguish between different error types. This may be helpful if you want to display localized error messages (where you could use code as the key to your translation dictionary) or when used in code to change the behaviour of your application in case of an error. The code is a string representing an error class hierarchy where each error class is delimited by a dot. This is helpful when you want to start of with little set of error translations and become more exact with the descriptions at a later point in time. So instead of translating a code like LPLC.validation.non_negative_float you could just translate LPLC.validation with 'Please check your input' and add a translation for float errors later on.

Error codes are documented for each resource below but the following are the most common ones:

- LPLC.validation
- LPLC.validation.missing input
- LPLC.validation.readonly
- LPLC.validation.non negative float
- LPLC.validation.positive integer
- LPLC.validation.smaller integer
- LPLC.validation.single character
- LPLC.validation.string
- LPLC.validation.boolean
- LPLC.format.encoding.utf8
- LPLC.format.malformed.json

1.1.5.3 Switching outputs, triggers and hold time settings

Every sample period ends with the selection of the most appropriate matcher ("group of colors"). This matcher is applied to the switching outputs under certain conditions. Relevant configuration for this behavior are the following settings:

- optional triggered update of switching outputs (see action-triggers)
- matcher attributes hold_time and reset_output_after_hold_time_expired

The following situations and actions are used in the behavior specification below:

- no active hold time: The hold_time attribute of the most recently applied matcher was zero.
 Thus there is currently no hold time configured until another matcher with a non-zero hold time is applied.
- hold time is expired: The most recently applied matcher had a hold time greater than zero, but this hold time elapsed since this matcher was applied. Thus an hold time was previously active, but it expired in the meantime.
- reset_output_after_hold_time_expired is on/off: When a matcher is applied (see below), then
 the attribute reset_output_after_hold_time_expired is memorized until another matcher is selected. The on/off state refers to this memorized value.
- *unchanged detected matcher*: The most recently applied matcher and the currently detected matcher are the same.
- *new detected matcher*: The most recently applied matcher and the currently detected matcher are not the same.
- action *do nothing*: The switching outputs and the currently memorized hold time settings stay unchanged.
- action apply new matcher: Memorize the matcher attributes hold_time and reset_output_after_hold_time_expired as current hold time settings and set the switching outputs as specified in the output pattern of the matcher.
- action apply 'no match': Memorize the detection profile attribute non_matching_hold_time as
 the current hold time. Memorize false for reset_output_after_hold_time_expired. Set the
 switching outputs as specified in the detection profile attribute non_matching_output.

The behaviour with and without *triggered updates of switching outputs* differs significantly. Thus both situations are specified separately below.

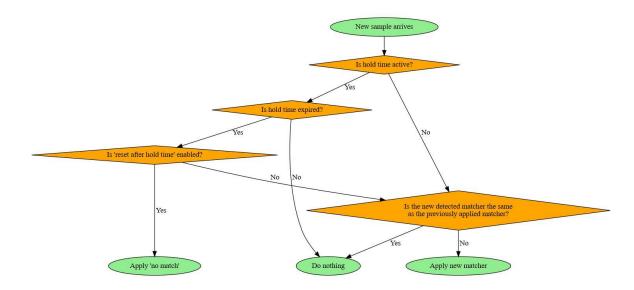
Disabled triggered update of switching outputs

Triggered updates are disabled if none of the trigger events (e.g. a rising edge of the first input line) is configured with the enable_switching_output action (see /api/actions).

The following table lists the specific behavior based on the currently active hold time settings and depending on the currently detected matcher.

Currently active hold time settings	Action for unchanged detected matcher	Action for new detected matcher
no active hold time	do nothing	apply new matcher
hold time is expired; reset_output_after_hold_time_expired is off	do nothing	apply new matcher
hold time is expired; reset_output_after_hold_time_expired is on	apply no match	apply no match
hold time is not expired	do nothing	do nothing

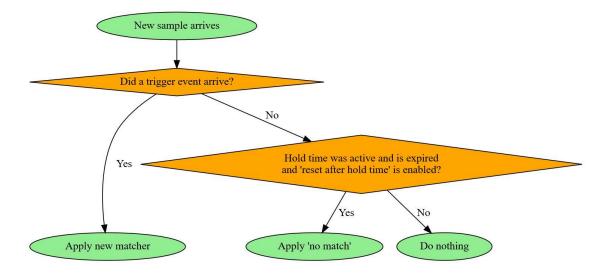
The following flowchart visualizes the decisions and actions when applying a matcher. It is a different representation of the table above.



Enabled triggered update of switching outputs

Triggered updates are enabled if any of the trigger events (e.g. a rising edge of the first input line) is configured with the *enable_switching_output* action (see /api/actions)

The following flowchart visualizes the decisions and actions when applying a matcher.



1.1.5.4 Websockets

The websocket provided by the API can be used to develop highly interactive frontends for the sensors that work without polling the endpoints of the REST-like resources.

Aside from a stream of samples that the sensors pushes to the websocket it also broadcasts information about events like newly taught colors or a change of the sensors configuration.

The websocket is developed as a complement to the resources and not as a replacement. The primary interface to control the sensor is and always will be the resources outlined below.

Requests to any of the endpoints under /websocket do not reset the session timeout.

Overview

You can access the websocket on the <code>/websocket/notifications/websocket</code> path. The data transmitted on this channel resembles a stream of information. In order to easily identify relevant packages, every payload transmitted through the socket is encapsulated in a JSON object containing the following properties:

- id
 - The unique identifier of this particular websocket packet
- source
 - The source this packet originates from or the reason for its transmission (e.g. detection profile.matcher)
- timestamp
 - The sensors uptime when the packet is sent over the websocket.
- payload
 - An object containing an event property (with values like changed or created), and an optional unid and data attribute that represent the object described by the event from the source.

One of the fields <code>added_items</code>, <code>changed_items</code> and <code>removed_items</code> may be present if <code>source</code> refers to a collection. In this case the value of this field is a list of identifiers belonging to the affected items of the collection. This allows clients to synchronize their data model without requesting the full collection after each collection-related notification. If a collection-related notification does not contain any of the fields above, then the scope of changes is unspecific and thus a full retrieval of the collection may be necessary.

Fallback

In order to support older Browsers the /websocket/notifications endpoint is compatible with the SockJS Client-Bibliothek library that implements fallback techniques like XHR Streaming, JSONP, Long Polling and others.

1.1.5.5 Networking and Discovery

Network discovery

The sensor announces itself in the local network via the following protocols:

- zeroconf / avahi broadcasts
- SSDP

The SSDP protocol allows discovery of the sensors via the Windows network neighborhood. The zeroconf protocol allows discovery on Linux, MacOS and mobile devices.

Automatically assigned link-local address

The sensor is reachable via its explicitly configured IPv4 and IPv6 addresses as well as via its automatically configured link-local address. This address belongs to the subnet "fe80::/10" with the local address part being based on the MAC address of the sensor (see "EUI-64"). The link-local address of the sensor is usable in all networks independent of the sensor configuration. Thus it provides a stable address under all circumstances.

Link-local addresses in general need to be suffixed with the local network identifier.

Examples

- Windows: fe80::1234:56ff:fe78:90ab%0
- Linux / MacOS / Android: fe80::1234:56ff:fe78:90ab%eth0

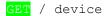
The network identifier (suffix after "%") in the examples above need to be adjusted to the local setup of the device connecting to the sensor.

1.2 Resources

1.2.1 Device Information

Some constant properties describe the individual device itself and include information about the model and vendor.

1.2.1.1 Get Device Properties



Return invariable properties of the device.

Response

Code	Body	application/json	
200	Properties (object)	DeviceInformation	
	data	id	Serial Number
	DeviceInformation, required	DeviceSerialNumber, required	
		model_name	human-readable name of the de-
		string, required	vice model
		model_key	unique id of the device model
		string, required	
		variant	indicates a special series of a
		any of string or null, required	model
		vendor_key	Unique key identifying the organ-
		DeviceVendorKey, required	ization distributing this device
		vendor_name	Name of vendor of this device
		DeviceVendorName, required	
		device_id	Deprecated: use "id" instead.
		DeviceSerialNumber, optional, Deprecated	
		model	Deprecated: use "model_name"
		string, optional, Deprecated	instead.
		vendor	Deprecated: use "vendor_name"
		DeviceVendorName, optional, Deprecated	instead.
4XX	errors	Error[]	
	Array of Error, required		
		code	machine-readable unique error
		string, optional	code
		mapping	a reference to the parameter that
		string, optional	caused the error
		message	human-readable error descrip-
		string, optional	tion

Example

1.2.2 Sensor

Query and modify all details of sensoric configuration and operation.

1.2.2.1 Retrieve Sensor Samples

GET / sensor / samples

Returns a list of samples from the color detection.

When no additional query parameters are passed the collection contains samples from the past. You can activate sample-streaming with the stream query parameter. In this case only new samples will be returned, as they become available.

Samples are implemented as ring buffer. Old samples will be removed from the collection as new samples are added.

Request

iequesi	
Query Parameters	
stream number, one of [0, 1], de-	Controls whether or not stream-mode is activated.
fault: 0, optional	When streaming is activated only the sensor will contionously transmit new samples to the client. The number of samples that are transmitted can be controlled with the stream_count query parameter.
	When streaming is deactivated (which is the default), only past samples are returned.
stream_count integer , default: 0, optional	Determines how many samples should be transmitted before the connection is terminated when stream-mode has been activated. The default is to stream indefinitely.
format string, one of [json, csv], default: json, optional	Determines the output format of the samples when stream-mode has been activated. If csv is selected the first transmitted line are the column headers. Headers are based on the default JSON representation and use the syntax also used by JavaScript. Given the ex-
	ample representations.RGB[0] the value for this header would refer to the first item in the RGB representation.
delimiter string , default: , , minimum	Determines the column delimiter when CSV has been selected as output-format.
length: 1, maximum length: 1, optional	If you want to use a semicolon as delimiter be sure to url-encode it first (%3B). Otherwise it's interpreted as query parameter separator.
	Be aware that even though unicode characters are allowed by the API you should restrict yourself to one-byte characters as most tools will fail to use delimiters that use two or more bytes.

Response

Code	Body	application/json				
200	Properties (object)	Data				
	data	samples	ColorDetectionRe-			
	object, re-	Array of Color-	sult[]			
	quired	DetectionResult,				
		required				
			uuid	unique identifier (UUID)		
			UUID (string), pat-	as defined by RFC 4122,		
			tern: ^[a-f0-9-	ITU-T Rec. X.667, and		
] +\$, required,	ISO/IEC 9834-8		
			read-only			
			timestamp	The timestamp (given in		
			TimestampBa-	microseconds) is based		
			<u>ckendUptime</u>	on the uptime of the inter-		
			(number), required	nal analog sensor		
				backend. It may get reset to zero under specific		
				conditions.		
			corrected color	Representation of a color		
			CorrectedColor, re-	in the colorspace XYZ.		
			quired	in the deletapade X12.		
			1			
				values	Location in a color-	
				Array of number, mini-	space	
				mum items: 3, maximum	*	
				items: 3, required		
			transformed_color	A color represented by a		
			TransformedColor,	coordinate in the color-		
			required	space. The array indices		
				of the values property		
				match the order of the		

Code	Body	application/json				
	•			colorspace.axes property of currently used detection profile.		
				values Array of number, minimum items: 3, maximum items: 3, required	Location in a color- space	
			representations ColorRepresentati-	Pre-calculcated visual representations of a color		
			ons, required	RGB Array of number, minimum items: 3, required	RGB color array representing the axes r, g, and b in that order. Values are floats between 0 and 1.	
			inputs InputsState, required	The state of all inputs during a given period is specified by a list of possible events combined with a boolean value indicating, if the given event occurred within the period.		
				// boolean, required	The boolean value indicates whether the named input event occurred during the last period.	
			detection ColorMatchingResult, required	After each sampling period the retrieved color value is compared to the stored detectables (color positions). Detectables are ignored, if the tolerance shape of their corresponding matcher does not encompass the current sample. Finally the closes suitable detectable is selected as the winner of the color matching operation. The corresponding matcher determines the state of the sensor for the duration of the next sampling period.		
				matcher any of <u>UUID</u> (string) or null, optional, Deprecated	Deprecated: use "cho- sen_matcher_id" instead	
				chosen_matcher_id any of <u>UUID</u> (string) or null, required distances Array of any of number or	unique identifier of the selected matcher Distance between the sample's color	
				null, required	position and the selected matcher's closest color position along the three axes of the color space. The array contains three 'null' values, if no suitable matcher was found for the current color sample.	
				output_pattern CurrentSwitchingOutputs- State, required	Currently active state of the Switch-	

Code	Body	application/json				
					ing Outputs. Be- ware that this may deviate from the specified output states of the cur- rent best matcher, since settings like triggered input or hold time influence update process for the Switching Out- puts.	
					states Array of any of boolean or null, re- quired	List of True/False values de- scribing the current states of the Switch- ing Outputs
			signal_level number, required	The signal level indicates the usage of the internal ADC sampling range. This		
	errors Array of Error, required	Error[]				
		code String, optional mapping	machine-readable unique error code a reference to the			
		String, optional	parameter that caused the error			
		message String, optional	human-readable er- ror description			

1.2.2.2 Retrieve latest Sensor Sample

GET / sensor / samples / current

Returns the latest sample.

Be aware that the same sample may be returned for successive requests if no new samples arrived meanwhile. While the sensor is performing auto-gain or is over-saturated, it will return the last valid sample that was processed.

The result is empty (null), while the sensor is processing a configuration change request.

Response

Code	Body	application/json		
200	Properties			
	(object)			
	data	ColorDetection-		
	Any of Col-	Result		
	orDetection-			
	Result or			
	null, re-			
	quired			
		uuid	unique identifier (UUID) as	
		UUID (string), pat-	defined by RFC 4122, ITU-T	
		tern: ^[a-f0-9-	Rec. X.667, and ISO/IEC	
] +\$, required,	9834-8	
		read-only		
		timestamp	The timestamp (given in mi-	
		TimestampBa-	croseconds) is based on the	
		ckendUptime	uptime of the internal analog	
		(number), requi-	sensor backend. It may get	
		red	reset to zero under specific	
			conditions.	

Code	Body	application/json			
		corrected_color CorrectedColor, required	Representation of a color in the colorspace XYZ.		
			values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
		transformed_co- lor TransformedCo- lor, required	A color represented by a co- ordinate in the colorspace. The array indices of the val- ues property match the order of the colorspace.axes property of currently used de- tection profile.		
			values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
		representations ColorRepresenta- tions, required	Pre-calculcated visual repre- sentations of a color suitable for rendering		
			RGB Array of number, minimum items: 3, maximum items: 3, required	RGB color array representing the axes r, g, and b in that or- der. Values are floats between 0 and 1.	
		inputs InputsState, re- quired	The state of all inputs during a given period is specified by a list of possible events combined with a boolean value indicating, if the given event occurred within the period.		
			// boolean, required	The boolean value indicates whether the named input event occurred during the last period.	
		detection ColorMatchingRe- sult, required	After each sampling period the retrieved color value is compared to the stored detectables (color positions). Detectables are ignored, if the tolerance shape of their corresponding matcher does not encompass the current sample. Finally the closes suitable detectable is selected as the winner of the color matching operation. The corresponding matcher determines the state of the sensor for the duration of the next sampling period.	Deprecated: use "cho-	
			any of <u>UUID</u> (string) or null, optional, Deprecated	sen_matcher_id" instead	
			chosen_matcher_id any of <u>UUID</u> (string) or null, required	unique identifier of the selected matcher	
			distances Array of any of number or null, required	Distance between the sample's color position and the selected matcher's closest color position along the three axes of the color space. The array contains three 'null' values, if no suitable matcher was found for the current color sample.	
			output_pattern CurrentSwitchingOutputs- State, required	Currently active state of the Switching Outputs. Beware that this may deviate from the speci- fied output states of the current best matcher, since settings like	

Code	Body	application/json			
				triggered input or hold time in-	
				fluence update process for the	
				Switching Outputs.	
				states	List of True/False
				Array of any of boolean or null,	values describing
				required	the current states
					of the Switching
					Outputs
		signal_level	The signal level indicates the		
		number, required	usage of the internal ADC		
			sampling range. This		
	errors	Error[]			
	Array of Er-				
	ror, re-				
	quired				
		code	machine-readable unique er-		
		String, optional	ror code		
		mapping	a reference to the parameter		
		String, optional	that caused the error		
		message	human-readable error		
		String, optional	description		

1.2.2.3 Retrieve all Matchers (color groups)

GET / sensor / matchers

Matchers (color groups) describe a color detection result. A Matcher contains details regarding the wanted behaviour of the Switching Outputs and a list of *Detectables* (color positions).

The most basic setup of a colorsensor could involve only a single color group containing all *positive* detection results. All samples that are not matched by this color group would indicate a problem of the monitored real-world process.

A more advanced usage of Matchers could additionally include a Matcher for the different acceptable background colors between real target objects (e.g. the color of the conveyor belt). Thus the Switching Outputs of the sensor could indicate whether a *positive*, a *neutral* or a *negative* real-world event was sampled.

Of course, every Matcher may also simply contain exactly one color position, in order to allow finegrained classification of the target's appearance.

The tolerance field of a matcher describes the shape and the dimensions of the part of the color-space that is covered by this matcher. Only color positions within this space may cause a match for this matcher. A tolerance is specified by a shape and a dictionary of limits. Both attributes need to be specified. An empty dictionary of limits is interpreted as the default limits for the requested shape.

Request

Query Parameters	
profile_id	Filter ColorMatchers by the given Detection Profile ID. Only ColorMatchers that are part of
String, optional	the given Detection Profile will be returned.

Response

Code	Body	application/json			
200	Properties				
	(object)				
	data	data			
	Object, re-				
	quired				
		matchers	ColorMatcher[]		
		Array of Col-			
		orMatcher, re-			
		quired			

Cada	Dade	annia ation /ia an	I	T	T
Code	Body	application/json		(11115)	
			uuid	unique identifier (UUID)	
			UUID (string), pat-	as defined by RFC 4122,	
			tern: ^[a-f0-9-]+\$,	ITU-T Rec. X.667, and	
			required, read-only		
			alias	A numerical value that	
			Alias (integer), re-	can be used to address	
			quired, read-only	an item in a collection. If	
				an alias is specified	
				alongside an uuid attrib-	
				ute, that alias can be	
				used as an alternative to	
				address the item in URLs	
				and other protocols like	
				Modbus or serial inter-	
				faces.	
			name	human-readable name of	
			String, required	the matcher	
			tolerance	Specification of a geo-	
			Any of InfiniteColor-	·	
			Tolerance, Sphere-	mensions in the current	
			ColorTolerance,	colorspaces.	
			CylinderColorToler-		
			ance or BoxColor-		
			Tolerance, re-		
			quired		
			7-11-0-1		
				InfiniteColorTolerance	
				limits	limits
				Object, required	IIIIII
				shape	Name of the geo-
				ToleranceShapeName	metrical shape of
				(string), required	the tolerance. The
					supported toler-
					ance shapes can
					be retrieved via
					/api/sen-
					sor/capabili-
					ties .
				SphereColorTolerance	
				limits	limits
				Object, required	radius
					Numer, required
				shape	Name of the geo-
				ToleranceShapeName	metrical shape of
				(string), required	the tolerance. The
				(Stillig), required	supported toler-
					ance shapes can
					be retrieved via
					-
					/api/sen-
					sor/capabili-
				Outine de monte de la company	ties.
ļ				CylinderColorTolerance	line it e
				limits	limits
				Object, required	radius
					Number, required
					half_height
ļ					Number, required
				shape	Name of the geo-
				ToleranceShapeName	metrical shape of
				(string), required	the tolerance. The
					supported toler-
					ance shapes can
					be retrieved via
					/api/sen-
					sor/capabili-
					ties
				BoxColorTolerance	
				limits	limits
				Object, required	half_edges
				, , = -	Array of number, minimum items:
	i .	1	1	1	
					3, maximum items: 3, required

Code	Body	application/json			
	,			shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported toler- ance shapes can be retrieved via /api/sensor/capabilities.
			output_pattern WantedSwit- chingOutputsState, required	The combination of tristate values describes a logical state of the switching outputs of the sensor. The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.	
				uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
				states Array of any of boolean or null, required	List of True/False/Null values de- scribing the wanted states of the Switching Outputs
			hold_time HoldTime (num- ber), maximum: 3153600000, re- quired	Minimum duration (in seconds) of a matcher's output setup being applied after detection.	
			reset_output_af- ter_hold_time_ex- pired Boolean, default: false, required	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering inputs and wish to reset the outputs afterwards.	
			signal_color Any of string or null, required	A custom color name. How and what color will be displayed is defined by the client.	
	errors Array of Er- ror, re- quired	Error []			
		code String, optional mapping String, optional	machine-readable unique error code a reference to the parameter that caused the error		
		message String, optional	human-readable er- ror description		

1.2.2.4 Create a new Matcher (color group)

POST / sensor / matchers

Stores a new matcher (color group) on the sensor. In order to add colors to it, use the /api/sen-sor/detectables endpoint.

Request

Body	application/json	
Properties		
(ColorMat-		
cher)		
	uuid	unique identifier (UUID) as defined
	UUID (string), pattern: ^ [a-f0-	by RFC 4122, ITU-T Rec. X.667,
	9-]+\$, required, read-only	and ISO/IEC 9834-8

Body	application/json		
Jouy	alias	A numerical value that can be used	
	Alias (integer), required,	to address an item in a collection. If	
	read-only	an alias is specified alongside an	
	read-only	uuid attribute, that alias can be	
		used as an alternative to address	
		the item in URLs and other proto-	
		cols like Modbus or serial inter-	
		faces.	
	name	human-readable name of the	
	String, required	matcher	
	tolerance	Specification of a geometric shape	
	Any of InfiniteColorTolerance,	and its dimensions in the current	
	SphereColorTolerance, Cylin-	colorspaces.	
	derColorTolerance or Box-		
	ColorTolerance, required		
	· ·		
		InfiniteColorTolerance	
		limits	limits
		Object, required	
	+	shape	Name of the geometrical shape of the
		ToleranceShapeName (string), re-	tolerance. The supported tolerance
		quired	shapes can be retrieved via /api/sen-
			sor/capabilities .
		SphereColorTolerance	
1			
		limits	limits
			limits radius
		limits Object, required	radius
		Object, required	radius Numer, required
		Object, required shape	radius Numer, required Name of the geometrical shape of the
		Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance
		Object, required shape	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance
		Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Object, required shape ToleranceShapeName (string), required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Shape ToleranceShapeName (string), required CylinderColorTolerance	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.
		Shape ToleranceShapeName (string), required CylinderColorTolerance	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), re-	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-
		Shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities . limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities .
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3,
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3,
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3, maximum items: 3, required
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits Object, required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3, maximum items: 3, required Name of the geometrical shape of the
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits Object, required shape ToleranceShapeName (string), required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3, maximum items: 3, required Name of the geometrical shape of the tolerance. The supported tolerance
		shape ToleranceShapeName (string), required CylinderColorTolerance limits Object, required shape ToleranceShapeName (string), required BoxColorTolerance limits Object, required	radius Numer, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits radius Number, required half_height Number, required Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities. limits half_edges Array of number, minimum items: 3, maximum items: 3, required Name of the geometrical shape of the

Body	application/json		
	output_pattern WantedSwitchingOutputs- State, required	The combination of tristate values describes a logical state of the switching outputs of the sensor. The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.	
		uuid UUID (string), pattern: ^ [a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
		states Array of any of boolean or null, required	List of True/False/Null values describing the wanted states of the Switching Outputs
	hold_time HoldTime (number), maxi- mum: 3153600000, required	Minimum duration (in seconds) of a matcher's output setup being ap- plied after detection.	
	reset_output_af- ter_hold_time_expired Boolean, default: false, re- quired	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering inputs and wish to reset the outputs afterwards.	
	signal_color Any of string or null, required	A custom color name. How and what color will be displayed is defined by the client.	

Example

```
"uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
  "alias": 6,
  "name": "clean bottle cap",
  "tolerance": {
    "limits": {
      "radius": 2,
      "half height": 4
    "shape": "cylinder"
  "output_pattern": {
    "uuid": "ladc74e2-96ac-4761-b9e6-2d93e02d9244",
    "states": [
     true,
      false,
      false
    ]
  } ,
  "hold_time": 0,
  "reset_output_after_hold_time_expired": false,
"signal_color": null
}
```

Response

Code	Body	application/json			
200	Properties				
400	(object)				
	data <u>ColorMat-</u> <u>cher</u> , requi- red	A matcher represents a distinguished detection result and the wanted behaviour of the sensor whenever it is encountered.			
			ColorMatcher[]		

Code	Body	application/json			
-	Douy		uuid	unique identifier (UUID)	
			UUID (string), pat-	as defined by RFC 4122,	
			tern: ^ [a-f0-9-]+\$, required, read-only	ITU-T Rec. X.667, and ISO/IEC 9834-8	
			alias	A numerical value that	
			Alias (integer), re-	can be used to address	
			quired, read-only	an item in a collection. If an alias is specified	
				alongside an uuid attrib-	
				ute, that alias can be	
				used as an alternative to address the item in URLs	
				and other protocols like	
				Modbus or serial inter- faces.	
			name	human-readable name of	
			String, required	the matcher	
			tolerance Any of InfiniteColor-	Specification of a geo- metric shape and its di-	
			Tolerance, Sphere-	mensions in the current	
			ColorTolerance,	colorspaces.	
			CylinderColorToler- ance or BoxColor-		
			Tolerance, re-		
			quired	1.5 % 0 : 7 :	
				InfiniteColorTolerance limits	limits
				Object, required	
				shape	Name of the geo-
				ToleranceShapeName (string), required	metrical shape of the tolerance. The
				(5.111.19), 1.04	supported toler-
					ance shapes can
					be retrieved via /api/sen-
					sor/capabili-
				SphereColorTolerance	ties .
				limits	limits
				Object, required	radius
				shape	Numer, required Name of the geo-
				ToleranceShapeName	metrical shape of
				(string), required	the tolerance. The
					supported toler- ance shapes can
					be retrieved via
					/api/sen- sor/capabili-
					ties .
				CylinderColorTolerance	limita
				limits Object, required	limits radius
					Number, required
					half_height Number, required
				shape	Name of the geo-
				ToleranceShapeName	metrical shape of
				(string), required	the tolerance. The supported toler-
					ance shapes can
					be retrieved via /api/sen-
					/api/sen- sor/capabili-
					ties.
				BoxColorTolerance	limita
				limits Object, required	limits half_edges
				, , ,	Array of number,
					minimum items: 3, maximum items: 3,
					required
		i	1	1	<u> </u>

Code	Body	application/json				
				shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sen-sor/capabilities.	
			output_pattern WantedSwit- chingOutputsState, required	The combination of tri- state values describes a logical state of the switch- ing outputs of the sensor. The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.		
				uuid <u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU- T Rec. X.667, and ISO/IEC 9834-8	
				states Array of any of boolean or null, required	List of True/False/Null val- ues describing the wanted states of the Switching Out- puts	
			hold_time HoldTime (num- ber), maximum: 3153600000, re- quired	Minimum duration (in seconds) of a matcher's output setup being applied after detection.		
			reset_output_af- ter_hold_time_ex- pired Boolean, default: false, required	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering inputs and wish to reset the outputs afterwards.		
			signal_color Any of string or null, required	A custom color name. How and what color will be displayed is defined by the client.		
	errors Array of Er- ror, required	Error []				
		code	machine-readable			
		String, optional	unique error code			
		mapping String, optional	a reference to the parameter that caused the error			
		message	human-readable er-			
		String, optional	ror description			
			lowing error codes			
		LPLC.validatio	n.collection_siz	e_exceeded		

1.2.2.5 Remove multiple or all ColorMatchers

DELETE / sensor / matchers

Remove a selection of ColorMatchers either based on a given filter argument (if supported for this collection) or remove all ColorMatchers from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DELETE request against an empty collection or for a filtered DELETE request against a collection without ColorMatchers matching the filter.

Response

Code	
204	The empty response indicates success

1.2.2.6 Retrieve Matcher (color group) Details

GET / sensor / matchers / {itemId}

Returns the current configuration of a matcher.

Request

Path Variables	
itemId	
String, required	

Response

Code		application/json			
200	Properties (ob-				
400	ject)				
	data	A matcher repre-			
	ColorMatcher,	sents a distinguished			
	required	detection result and			
		the wanted behav-			
		iour of the sensor			
		whenever it is en-			
		countered.			
			ColorMatcher[]		
			uuid	unique identifier (UUID)	
			UUID (string), pat-	as defined by RFC 4122,	
			tern: ^ [a-f0-9-]+\$,		
			required, read-	ISO/IEC 9834-8	
			only		
			alias	A numerical value that	
			Alias (integer), re-	can be used to address	
			quired, read-only	an item in a collection. If	
			94	an alias is specified	
				alongside an uuid attrib-	
				ute, that alias can be	
				used as an alternative to	
				address the item in URLs	
				and other protocols like	
				Modbus or serial inter-	
				faces.	
			name	human-readable name of	
			String, required	the matcher	
			tolerance	Specification of a geomet-	
			Any of Infinit-	ric shape and its dimen-	
			eColorTolerance.	sions in the current color-	
			SphereColorToler-	spaces.	
			ance, Cylinder-	spaces.	
			ColorTolerance or		
			BoxColorToler-		
			ance, required		
			ance, required		
				InfiniteColorTolerance	
				limits	limits
				Object, required	
				shape	Name of the geometrical
				ToleranceShapeName	shape of the tolerance. The
				(string), required	supported tolerance shapes
					can be retrieved via
					/api/sensor/capabil-
					ities .
				SphereColorTolerance	
				limits	limits
	•	•			

Code	Body	application/icon		Ī	<u> </u>
Coue	Body	application/json		Object, required	radius
				,	Numer, required
				shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via
					/api/sensor/capabil- ities.
				CylinderColorTolerance	
				limits Object, required	limits radius
					Number, required half_height Number, required
				shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabil- ities.
				BoxColorTolerance	icies.
				limits	limits
				Object, required	half_edges Array of number, minimum items: 3, maximum items: 3, required
				shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabil- ities.
			output_pattern	The combination of tri-	
			WantedSwit-	state values describes a	
			chingOutputs- State, required	logical state of the switching outputs of the sensor. The states true or	
				false cause the output to go up or down. The	
				state null keeps the previous state of the output unchanged.	
				uuid UUID (string), pattern: ^ [a-f0-9-]+\$, required, read-only states Array of any of boolean or null, required	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8 List of True/False/Null values describing the wanted states of the Switching Outputs
			hold_time HoldTime (num- ber), maximum: 3153600000, re- quired	Minimum duration (in sec- onds) of a matcher's out- put setup being applied after detection.	
			reset_output_af- ter_hold_time_ex- pired Boolean, default: false, required	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering inputs and wish to reset the outputs afterwards.	
			signal_color Any of string or null, required	A custom color name. How and what color will be displayed is defined by the client.	
	errors Array of Error, required	Error []			
		code String, optional	machine-readable unique error code		

Code	Body	application/json			
		mapping	a reference to the		
		String, optional	parameter that		
			caused the error		
		message	human-readable		
		String, optional	error description		
		May return the following error codes			
		LPLC.not found.collection.item			

1.2.2.7 Delete a Matcher (color group)

DELETE / sensor / matchers / {itemId}

Deletes the matcher and all associated detectables.

Request

Path variables	
itemId	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LPIC.not found.collection.item

1.2.2.8 Update the Matcher (color group) Configuration

PUT / sensor / matchers / {itemId}

Update the matcher with a new configuration.

Request

Path Variables	
itemId	
String, required	

Body	application/json		
Properties (Color Mat- cher)			
	uuid UUID (string), pattern: ^ [a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
	alias Alias (integer), required, read- only	A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.	
	name	human-readable name of the matcher	
	String, required tolerance Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or Box- ColorTolerance, required	Specification of a geometric shape and its dimensions in the current colorspaces.	
		InfiniteColorTolerance	
		limits Object, required	limits
		shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tol- erance shapes can be retrieved via /api/sensor/capabili- ties.
		SphereColorTolerance	
		limits	limits

Body	application/json		
		Object, required	radius Numer, required
		shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tol- erance shapes can be retrieved via /api/sensor/capabili-
			ties.
		CylinderColorTolerance	P 9
		limits Object, required	limits radius Number, required half_height Number, required
		shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tol- erance shapes can be retrieved via /api/sensor/capabili- ties .
		BoxColorTolerance	
		limits Object, required	limits half_edges Array of number, minimum items: 3, maximum items: 3, required
		shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tol- erance shapes can be retrieved via /api/sensor/capabili- ties .
	output_pattern WantedSwitchingOutputs- State, required	The combination of tristate values describes a logical state of the switching outputs of the sensor. The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.	
		uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
		states Array of any of boolean or null, required	List of True/False/Null values de- scribing the wanted states of the Switching Outputs
	hold_time HoldTime (number), maxi- mum: 3153600000, required	Minimum duration (in seconds) of a matcher's output setup being applied after detection.	
	reset_output_af- ter_hold_time_expired Boolean, default: false, re- quired	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering inputs and wish to reset the outputs afterwards.	
	signal_color Any of string or null, required	A custom color name. How and what color will be displayed is defined by the client.	

Examples

```
"uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
"alias": 6,
"name": "clean bottle cap",
"tolerance": {
    "limits": {
        "radius": 2,
        "half_height": 4
    },
    "shape": "cylinder"
},
"output_pattern": {
    "uuid": "ladc74e2-96ac-4761-b9e6-2d93e02d9244",
    "states": [
```

```
true,
    false,
    false
]
},
"hold_time": 0,
"reset_output_after_hold_time_expired": false,
"signal_color": null
}
```

Response

Ticope		T	T	T	T
Code	Body	application/json			
200	Properties (ob-				
400	ject)				
	jootj				
404					
	data	A matcher repre-			
	ColorMatcher,	sents a distin-			
	required	guished detection			
	required				
		result and the			
		wanted behaviour of			
		the sensor whenever			
		it is encountered.			
		it is chesantered.	O a la mM a la la an		
			ColorMatcher		
			uuid	unique identifier (UUID) as	
			<u>UUID</u> (string), pattern:	defined by RFC 4122, ITU-T	
			^ [a-f0-9-]+\$, re-	Rec. X.667, and ISO/IEC	
				9834-8	
			quired, read-only		
			alias	A numerical value that can be	
			Alias (integer), re-	used to address an item in a	
			quired, read-only	collection. If an alias is speci-	
			7-11-04, 1044 01119	fied alongside an uuid attrib-	
1				ute, that alias can be used as	
				an alternative to address the	
				item in URLs and other proto-	
				cols like Modbus or serial in-	
				terfaces.	
			name	human-readable name of the	
			String, required	matcher	
			tolerance	Specification of a geometric	
			Any of InfiniteColor-	shape and its dimensions in	
			Tolerance, Sphere-	the current colorspaces.	
			ColorTolerance, Cylin-		
			derColorTolerance or		
			BoxColorTolerance,		
			required		
				InfiniteColorTolerance	
-					P 9
				limits	limits
				Object, required	
				shape	Name of the geomet-
				ToleranceShapeName	rical shape of the tol-
				(string), required	erance. The sup-
					ported tolerance
					shapes can be re-
					trieved via
					· ·
					/api/sensor/ca-
	<u></u>				pabilities.
				SphereColorTolerance	
	1			limits	limits
				Object, required	radius
					Numer, required
				shape	Name of the geomet-
				ToleranceShapeName	rical shape of the tol-
				(string), required	erance. The sup-
				(Surrig), required	
					ported tolerance
					shapes can be re-
					trieved via
					/api/sensor/ca-
	ļ				pabilities.
				CylinderColorTolerance	
				limits	limits
L	I .	1	l		

Code	Body	application/json		T	1
Code	Body	application/json		Object, required	radius
				Object, required	Number, required
					half_height
					Number, required
				shape	Name of the geomet-
				ToleranceShapeName	rical shape of the tol-
				(string), required	erance. The sup-
					ported tolerance shapes can be re-
					trieved via
					/api/sensor/ca-
					pabilities .
				BoxColorTolerance	
				limits	limits
				Object, required	half_edges
					Array of number, mini-
					mum items: 3, maxi- mum items: 3, re-
					auired
				shape	Name of the geomet-
				ToleranceShapeName	rical shape of the tol-
				(string), required	erance. The sup-
					ported tolerance
					shapes can be re-
					trieved via /api/sensor/ca-
					/api/sensor/ca- pabilities .
			output_pattern	The combination of tristate	haniiiries .
			WantedSwitchingOut-	values describes a logical	
			putsState, required	state of the switching outputs	
				of the sensor.	
				The states true or false	
				cause the output to go up or	
				down. The state null keeps	
				the previous state of the output unchanged.	
				uuid	unique identifier
				UUID (string), pattern: ^ [a-	(UUID) as defined by
				f0-9-]+\$, required , read-only	RFC 4122, ITU-T Rec.
					X.667, and ISO/IEC
					9834-8
				states	List of True/False/Null
				Array of any of boolean or null, required	values describing the wanted states of the
				ridii, required	Switching Outputs
			hold time	Minimum duration (in sec-	
			HoldTime (number),	onds) of a matcher's output	
			maximum:	setup being applied after de-	
<u> </u>		-	3153600000, required	tection.	
			reset_output_af- ter_hold_time_ex-	Controls if the output should be reset after the hold time	
			pired	passed. This is helpful if you	
			Boolean, default:	only sample by triggering in-	
			false, required	puts and wish to reset the	
			-	outputs afterwards.	
			signal_color	A custom color name. How	
			Any of string or null,	and what color will be dis-	
			required	played is defined by the cli-	
-	errors	Error []		ent.	
	Array of Error,				
	required				
		code	machine-readable		
		String, optional	unique error code		
		mapping	a reference to the pa-		
		String, optional	rameter that caused		
 		maccago	the error human-readable error		
		message String, optional	description		
		May return the folio		1	
		LPLC.not_found.			

1.2.2.9 Retrieve ColorDetectables

GET / sensor / detectables

Detectables describe positions in the currently selected colorspace. Each detectable is part of a Matcher. Every Matcher may contain zero or more Detectables.

Detectables are used to determine the most suitable Matcher for a sampled color. This closest match defines the result of a sampling period and thus the behaviour of the sensor during the next sampling period.

Request

Query Parameters	
matcher_id	Filter detectables by the given Matcher ID. Only Detectables
<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	that are part of the given Matcher will be returned.
profile_id	Filter ColorDetectables by the given Detection Profile ID.
String, optional	Only ColorDetectables that are part of the given Detection
	Profile will be returned.

Response

Code	Body	application/json			
200	Properties				
	(object)				
	data	detectables			
	Object, requi-	Array of Color-			
	red	Detectable, re-			
		quired			
			ColorDetectable[]		
			uuid	unique identifier (UUID) as de-	
			UUID (string), pattern:	fined by RFC 4122, ITU-T Rec.	
			^[a-f0-9-]+\$, re-	X.667, and ISO/IEC 9834-8	
			quired, read-only		
			alias	A numerical value that can be	
			Alias (integer), re-	used to address an item in a	
			quired, read-only	collection. If an alias is speci-	
				fied alongside an uuid attribute,	
				that alias can be used as an al-	
				ternative to address the item in	
				URLs and other protocols like	
				Modbus or serial interfaces.	
			matcher_id	reference to the Matcher con-	
			UUID (string), pattern:	taining this Detectable	
			^[a-f0-9-]+\$, re-		
			quired, read-only		
			color	A color represented by a coor-	
			TransformedColor, re-	dinate in the colorspace. The	
			quired	array indices of the values	
				property match the order of the	
				colorspace.axes property of	
				currently used detection profile.	
				color	
				values	Location in a color-
				Array of number, minimum	space
				items: 3, maximum items: 3, re-	
				quired	
			representations	Pre-calculcated visual repre-	
			ColorRepresentations,	sentations of a color suitable	
-	1		optional, read-only	for rendering	
	-			representations	DOD 1
				RGB	RGB color array rep-
				Array of number, minimum	resenting the axes r,
				items: 3, maximum items: 3, re-	g, and b in that order.
				quired	Values are floats
	Errors	Error[]			between 0 and 1.
	Array of Error,	LITOTE			
	required				
	required	code	machine-readable		
		String, optional	unique error code		
		July, optional	unique en or code		

mapping String, optional	a reference to the pa- rameter that caused the error	
message	human-readable error	
String, optional	description	

1.2.2.10 Remove multiple or all ColorDetectables

DELETE / sensor / detectables

Remove a selection of ColorDetectables either based on a given filter argument (if supported for this collection) or remove all ColorDetectables from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DELETE request against an empty collection or for a filtered DELETE request against a collection without ColorDetectables matching the filter.

Request

Query Parameters		
matcher_id Remove only detectables with the given <i>Matcher</i> ID.		
<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only		
profile id	Filter ColorDetectables by the given Detection Profile ID.	
String, optional	Only ColorDetectables that are part of the given Detection	
	Profile will be returned.	

Response

Code	
204	The empty response indicates success.

1.2.2.11 Create ColorDetectables

POST / sensor / detectables

Create a new ColorDetectable.

All supported data attributes in the body of the request are optional.

Request

Body	application/json		
Properties	uuid	unique identifier (UUID) as	
(ColorDe-	<u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, re-	defined by RFC 4122, ITU-T	
tectable)	quired, read-only	Rec. X.667, and ISO/IEC	
		<u>9834-8</u>	
	alias	A numerical value that can be	
	Alias (integer), required, read-only	used to address an item in a	
		collection. If an alias is speci-	
		fied alongside an uuid attrib-	
		ute, that alias can be used as	
		an alternative to address the	
		item in URLs and other proto-	
		cols like Modbus or serial in-	
		terfaces.	
	matcher_id	reference to the Matcher con-	
	<u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, re-	taining this Detectable	
	quired, read-only		
	color	A color represented by a co-	
	<u>TransformedColor</u> , required	ordinate in the colorspace.	
		The array indices of the val-	
		ues property match the order	
		of the colorspace.axes	
		property of currently used de-	
		tection profile.	
		color	
		values	Location in a colorspace

	Array of number, minimum items: 3, maximum items: 3,	
	required	
representations	Pre-calculcated visual repre-	
ColorRepresentations, optional, read-only	sentations of a color suitable	
	for rendering	
	representations	
	RGB	RGB color array representing
	Array of number, minimum	the axes r, g, and b in that or-
	items: 3, maximum items: 3,	der. Values are floats
	required	between 0 and 1.

Examples

```
"uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5", "color": {
    "values": [
     0.476731,
      0.381263,
      0.128475
    ]
  },
  "representations": {
    "RGB": [
     0.396114,
     0.479113,
     0.552308
    ]
  }
}
```

Response

Code	Body	application/json	
200 400	Properties (<u>object)</u>		
	data ColorDetectable, required	A detectable represents the numeric position in a colorspace. It is connected to a <i>Matcher</i> .	
		ColorDetectable	
		uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
		alias Alias (integer), required, read-only	A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.
		matcher_id UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	reference to the Matcher containing this Detectable
		color TransformedColor, required	A color represented by a coordinate in the colorspace. The array indices of the values property

Code	Body	application/json		
			match the order of	
			the color-	
			space.axes prop-	
			erty of currently	
			used detection pro-	
			file.	
			color	
			values	Location in a co-
			Array of number,	lorspace
			minimum items: 3,	
			maximum items: 3,	
			required	
		representations	Pre-calculcated vis-	
		ColorRepresentations, optional, read-only	ual representations	
			of a color suitable	
			for rendering	
			representations	
			RGB	RGB color array
			Array of number,	representing the
			minimum items: 3,	axes r, g, and b
			maximum items: 3,	in that order. Va-
			required	lues are floats
				between 0 and 1.
	errors Array of Error, required	Error[]		
		code	machine-readable	
		String, optional	unique error code	
		Mapping	a reference to the	
		String, optional	parameter that	
			caused the error	
		Message	human-readable er-	
		String, optional	ror description	
		May return the following error codes		
		LPLC.validation.collection size	exceeded	

1.2.2.12 Delete ColorDetectable)

DELETE / sensor / detectable / {itemId}

Deletes a single ColorDetectable.

Request

Path Variables	
itemId	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LCOL.samples.unavailable

1.2.2.13 Modify ColorDetectable

PUT / sensor / detectable / {itemId}

Modifies a single ColorDetectable.

Request

rioquosi	
Path Variables	
itemId	
String, required	

Body application/json		
-----------------------	--	--

Properties	uuid	unique identifier (UUID) as defined by RFC 4122,	
(ColorDe-	<u>UUID</u> (string), pattern:	ITU-T Rec. X.667, and ISO/IEC 9834-8	
tectable)	^[a-f0-9-]+\$, re-		
	quired, read-only		
	alias	A numerical value that can be used to address an	
	Alias (integer), required,	item in a collection. If an alias is specified alongside	
	read-only	an uuid attribute, that alias can be used as an alter-	
		native to address the item in URLs and other proto-	
		cols like Modbus or serial interfaces.	
	matcher_id	reference to the <i>Matcher</i> containing this Detectable	
	<u>UUID</u> (string), pattern:		
	^[a-f0-9-]+\$, re-		
	quired, read-only		
	color	A color represented by a coordinate in the color-	
	<u>TransformedColor</u> , requi-	space. The array indices of the values property	
	red	match the order of the colorspace.axes prop-	
		erty of currently used detection profile.	
		color	
		values	Location in a colorspace
		Array of number, minimum items: 3, maximum	
		items: 3, required	
	representations	Pre-calculcated visual representations of a color	
	ColorRepresentations,	suitable for rendering	
	optional, read-only		
		representations	
		RGB	RGB color array represent-
		Array of number, minimum items: 3, maximum	ing the axes r, g, and b in
		items: 3, required	that order. Values are floats
			between 0 and 1.

Examples

```
"uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
 "alias": 2,
 "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
   "values": [
     0.476731,
     0.381263,
     0.128475
   ]
  },
  "representations": {
   "RGB": [
     0.396114,
     0.479113,
     0.552308
    ]
  }
}
```

Response

Code	Body	application/json		
200 400 404	Properties (object)			
	data ColorDetectable, required	A detectable represents the numeric position in a colorspace. It is connected to a <i>Matcher</i> .		
		ColorDetectable		
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	

Code	Body	application/json		
		alias	A numerical value	
		Alias (integer), required, read-only	that can be used to	
			address an item in a	
			collection. If an alias	
			is specified along-	
			side an uuid attrib-	
			ute, that alias can be	
			used as an alterna-	
			tive to address the	
			item in URLs and	
			other protocols like	
			Modbus or serial in-	
			terfaces.	
		matcher_id	reference to the	
		<u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$,	Matcher containing	
		required, read-only	this Detectable	
		color	A color represented	
		TransformedColor, required	by a coordinate in	
		, , , , , , , , , , , , , , , , ,	the colorspace. The	
			array indices of the	
			values property match the order of	
			the color-	
			space.axes prop-	
			erty of currently	
			used detection pro-	
			file.	
			color	
			values	Location in a co-
			Array of number,	lorspace
			minimum items: 3,	'
			maximum items: 3,	
			required	
		representations	Pre-calculcated vis-	
		ColorRepresentations, optional, read-only		
		Colornepresentations, optional, read-only	of a color suitable	
			for rendering	
			representations	707
			RGB	RGB color array
			Array of number,	representing the
			minimum items: 3,	axes r, g, and b
			maximum items: 3,	in that order. Va-
			required	lues are floats
				between 0 and 1.
	errors	Error[]		
	Array of Error, required			
		code	machine-readable	
		String, optional	unique error code	
		mapping	a reference to the	
		String, optional	parameter that	
		Samig, optional	caused the error	
-	+	maccaga	human-readable er-	
		message		
	Ba A A' A'	String, optional	ror description	
	May return the following e			
	LCOL.samples.unavaila	ble		

1.2.2.14 Get ColorDetectable

GET / sensor / detectable / {itemId}

Returns a single ColorDetectable.

Request

Path Variables	
itemId	
String, required	

Response

Code	Body	application/json	

200	Properties (object)			
	data ColorDetectable, required	A detectable represents the numeric position in a colorspace. It is connected to a		
	Ocioi Detectable, Tequired	Matcher.		
		ColorDetectable		
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T	
		alias	Rec. X.667, and ISO/IEC 9834-8 A numerical value	
		Alias (integer), required , read-only	that can be used to address an item in a collection. If an alias	
			is specified along- side an uuid attrib- ute, that alias can be	
			used as an alterna- tive to address the item in URLs and	
			other protocols like Modbus or serial in- terfaces.	
		matcher_id UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	reference to the Matcher containing this Detectable	
		color TransformedColor, required	A color represented by a coordinate in the colorspace. The	
			array indices of the values property match the order of	
			the color- space.axes prop- erty of currently	
			used detection pro- file.	
			values	Location in a co-
			Array of number, minimum items: 3, maximum items: 3, required	lorspace
		representations ColorRepresentations, optional, read-only	Pre-calculcated vis- ual representations of a color suitable for rendering	
			representations	
			RGB Array of number, minimum items: 3, maximum items: 3, required	RGB color array representing the axes r, g, and b in that order. Va- lues are floats
				between 0 and 1.
	erros Array of Error, required	Error[]	machine-readable	
			unique error code	
		String, optional mapping	a reference to the	
		String, optional	parameter that caused the error	
		message	human-readable er-	
		String, optional	ror description	
	May return the following erro	r codesn	10. dooonphon	
	LCOL.samples.unavailabl			

1.2.2.15 Switch current Detection Profile

PUT / sensor / detection-profiles

Only one of the available Detection Profiles is active at a given time. Write a new Detection Profile ID to the <code>current_profile_id</code> field in order to change the currently used profile.

Request

Body	application/json
Examples	a014e415-0fec-4734-ac3f-30da0a5f3899
Example	

Response

Code	Body	application/json	
200	Properties (object)		
	data CurrentDetectionProfileID (string), pattern: ^[a-f0-9-]+\$, required, read-only	The sensor can store multiple Detection Profiles, but it can only apply one at a time. The field current_profile_id contains the UUID of the Detection Profile that is currently used by the sensor for its operation. It allows to use the shortcut API endpoint /api/sensor/detection-profiles/current instead of specifying a Detection Profile by its UUID.	
	errors Array of Error, required	Error[]	
		code	machine-readable unique
		String, optional	error code
		mapping	a reference to the parameter
		String, optional	that caused the error
		message	human-readable error
		String, optional	description

1.2.2.16 Create DetectionProfiles

POST / sensor / detection-profiles

Create a new DetectionProfile.

All supported data attributes in the body of the request are optional.

Request

Body	application/json			
Properties	uuid	unique identifier (UUID) as defined by RFC		
(Detection-	UUID (string), pattern:	4122, ITU-T Rec. X.667, and ISO/IEC 9834-8		
Profile)	^[a-f0-9-]+\$, re-			
	quired, read-only			
	aliFas	A numerical value that can be used to address		
	Alias (integer), re-	an item in a collection. If an alias is specified		
	quired, read-only	alongside an uuid attribute, that alias can be		
		used as an alternative to address the item in		
		URLs and other protocols like Modbus or se-		
		rial interfaces.		
	name	Human-readable name of the Detection Profile		
	String, required			
	colorspace	A colorspace describes the numeric conver-		
	Colorspace, required	sion of colors under certain circumstances.		
		Different standardized colorspaces are suitable		
		for different detection tasks.		
		colorspace		
		name		
		String, required		
		space_id	Unique name of a	
		ColorspaceID,required	colorspace	
		axes	ColorspaceAxis[]	
		Array of ColorspaceAxis, minimum items: 3,		
		maximum items: 3, required		
			id	Unique name

Body	application/json			
Bouy	αρμιισαιιστήμεστι		String, required	
			label	Human-
			String, required	readable
				name
			minimum	lowest ex-
			Number, required	pected value of a color
				along this axis
				under usual
				circum-
				stances
			maximum	highest ex-
			Number, required	pected value of a color
				along this axis
				under usual
				circum-
				stances
	non_matching_output WantedSwitch-	This state of the Switching Outputs is applied,		
	ingOutputsState, re-	if the currently sample color does not belong to any of the stored <i>Matchers</i> .		
	quired	to any of the stored wateriers.		
	<u> </u>	non_matching_output		
		uuid	unique identifier	
		<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re-	(UUID) as defined by	
		quired, read-only	RFC 4122, ITU-T	
			Rec. X.667, and ISO/IEC 9834-8	
		states	List of	
		Array of any of boolean or null, required	True/False/Null val-	
			ues describing the	
			wanted states of the	
	non_match-	Minimum duration (in accords) of the	Switching Outputs	
	ing hold time	Minimum duration (in seconds) of the non matching output state being applied to		
	HoldTime (number),	the Switching Outputs of the sensor. This pro-		
	maximum 3153600000,	longing of a potential non matching event may		
	required	be useful, if the processing period of a con-		
		nected actor exceeds the sampling period of		
	compensation_set-	the sensor. The compensation settings of a Detection Pro-		
	tings	file describe the configuration of internal sen-		
	CompensationSettings,	sor components related to the stabilization and		
	required	compensation algorithms.		
		These values can be determined by issuing a		
		POST request against /api/sensor/detec-		
		tion-profiles/current/autogain. The result is a suitable set of compensation set-		
		tings for this sensor under the current circum-		
		stances.		
		The content of this data object is not meant to		
		be manipulated by regular users. It should be		
		handled as is (stored, transmitted and applied without modification or introspection).		
		compensation settings		
	sampling_settings	Sampling Settings describe all details of the		
	SamplingSettings, re-	sampling process.		
	quired	Its attributes may be queried and inspected		
		(e.g. in order to retrieve the current sample		
		rate). Most values stored within the Sampling Set-		
		tings should not be modified directly. The re-		
		lated API endpoint /api/sensor/detec-		
		tion-profiles/current/autogain		
		should be used instead.		
		The only modifiable attribute within the Sampling Settings is the <i>averages</i> value. It is safe		
		to change it, even though the default values		
		calculated during an autogain operation		
		should be optimal for most detection tasks.		
		sampling_settings		
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		

Dadu	annlication/icon		T
Body	application/json	led intensity	relative intensity of
		led_intensity	relative intensity of
		Number, minimum: 0, maximum: 1, required	the internal emitter
			during the light
			phase
		base_sample_rate	The base sample
		SampleRate (number), minimum: 0.01, re-	rate determines the
		quired	duration of a sam-
		· ·	pling period.
			After each sampling
			period, the gathered
			data is processed
			and a new detection
			result is calculated
			(e.g. the most suita-
			ble Matcher for the
			given sample). This
			may affect the state
			of the Switching Out-
			puts or trigger con-
			figured actions. Thus
			the base sample rate
			defines the maxi-
			mum rate of
			changes for the
			Switching Outputs.
			See also the effec-
		offestive semants not	tive sample rate.
		effective_sample_rate	The effective sample
		SampleRate (number), minimum: 0.01, re-	rate is the numeric
		quired	product of the base
			sample rate and the
			number of averages.
			It determines the
			minimum duration
			that a target needs
			to be sampled in or-
			der to determine its
			visual appearance
			correctly.
			With the default
			value of average set
			to one, this value is
			equal to the base
			sample rate.
		minimum_wanted_sample_rate	This informational
		SampleRate (number), minimum: 0.01, re-	value represents the
		quired	sample rate that was
			requested during the
			most recent Au-
			togain operation.
			The effective sample
			rate may deviate
			from the wanted
			sample rate, if the
			requested sample
			rate was not achiev-
			able due to limita-
			tions of the sensor
			(e.g. exceeding the
			supported sample
			rate) or due to the
			environment (e.g.
			not enough light,
			thus a slower ampli-
			fication with higher
			gain was neces-
			sary).
		sample_light_phase	defines if the sensor
		Boolean, required	should periodically
		boolean, required	activate the internal
	i	1	emitter for sampling
		a consider all and a six	
		sample_dark_phase Boolean, required	defines if the sensor should periodically

Body	application/json		
			deactivate the inter-
			nal emitter for sam-
			pling
		averages	Number of previous
		AverageSampleCount (integer), minimum: 1,	samples to be aver-
		required	aged for every sam-
			pling result. A rolling
			averaging algorithm
			is applied to the
			samples.
		amplification	The amplification
		AmplificationLevel (integer), required	level specifies the in-
			ternal configuration
			of an amplifier. This
			value is not meant to
			be manipulated by
			regular users. It
			should be handled
			as is (stored, trans-
			mitted and applied
			without modification
	white reference	The White Reference attribute is used for indi-	or introspection).
	_		
	Array of number, re-	cating a custom color balancing.	
	quired	Its content is subject to internal use. Thus it should not be accessed directly, but only	
		through the related API endpoints (e.g.	
		/api/sensor/detection-pro-	
		files/{itemId}/white-reference).	
	normalization_con-	Normalization constants are related to the	
	stant	White Reference.	
	Array of number, re-	Its content is subject to internal use. Thus it	
	quired	should not be accessed directly, but only	
		through the related API endpoints (e.g.	
		/api/sensor/detection-pro-	
		files/{itemId}/white-reference).	

Examples

```
"name": "#0",
"uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
"alias": 1,
"non_matching_hold_time": 0,
"colorspace": {
    "name": "L*a*b*",
   "axes": [
        "id": "L",
"label": "L*",
        "minimum": 0,
         "maximum": 100
      } ,
        "id": "a",
"label": "a*",
"minimum": -500,
         "maximum": 500
      },
        "id": "b",
"label": "b*",
"minimum": -200,
"maximum": 200
      }
   ],
```

```
"space id": "Lab"
  },
  "compensation_settings": {
    "monitor integration": {
      "control": 0.32499998807907104,
      "references": [
       0.7283520102500916,
        0.7442666888237,
        0.7066696286201477
      ]
    },
    "use calibration_samples": true
  "normalization constant": [
   237.4935277662995,
    242.62655153828055,
   587.8264132734112
  ],
  "white reference": [
    95.047,
    100,
    108.883
 ],
  "non_matching_output": {
    "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
    "states": [
     true,
     true,
      true,
      true,
      true,
      true,
      true,
      true
   ]
  },
  "sampling_settings": {
   "led intensity": 1,
    "amplification": 1,
    "sample light phase": true,
    "minimum wanted sample rate": 1000,
    "averages": 1,
    "base sample rate": 1000,
    "sample_dark_phase": true,
    "effective sample rate": 1000
  }
}
```

Response

Code	Body	application/json		
200	Properties			
400	(object)			
	data	A Detection Profile con-		
	Detection-	tains a complete set of		
	Profile, re-	sensor settings for a		
	quired	given detection task.		
		Multiple profiles can be		
		stored in order to switch		
		easily between different		
		detection task or for the		
		incremental development		
		of a refined profile.		
		Some attributes of a De-		
		tection Profile expose in-		

Code	Body	application/json			
	1	ternal details of the sen-		1	
1		sor, that should be deter-			
		mined indirectly via other			
		means. These attributes			
		are described only super-			
	1	ficially, since they should			
	1	be handled as is without			
	1	changing their value or			
		structure.			
<u> </u>	 	DetectionProfile			1
<u> </u>	 		Lunique identifier (LILID) se defined by DEC	-	+
	1	uuid	unique identifier (UUID) as defined by RFC		
	1	<u>UUID</u> (string), pattern:	4122, ITU-T Rec. X.667, and ISO/IEC 9834-		
	1	^[a-f0-9-]+\$, re-	8		
	ļ	quired, read-only			<u> </u>
		alias	A numerical value that can be used to ad-		
		Alias (integer), required,	dress an item in a collection. If an alias is		
		read-only	specified alongside an uuid attribute, that		
			alias can be used as an alternative to ad-		
			dress the item in URLs and other protocols		
<u>L</u>	<u> </u>		like Modbus or serial interfaces.		<u></u>
		name	Human-readable name of the Detection		
	1	String, required	Profile		
		colorspace	A colorspace describes the numeric con-		
	1	Colorspace, required	version of colors under certain circum-		
	1		stances. Different standardized colorspaces		
	1		are suitable for different detection tasks.		
—	 	 	colorspace		
					1
	1		name		
-	1		String, required	Haining or a second	1
	1		space_id	Unique name of a	
	ļ		ColorspaceID, required	colorspace	
	1		axes	ColorspaceAxis[]	
	1		Array of ColorspaceAxis, minimum items: 3,		
			maximum items: 3, required		
1				id	Unique
1	•				
1]			String, required	name
				String, required	
				label	name
					name Human- readable
				String, required	name Human- readable name
				label String, required minimum	name Human- readable name lowest
				String, required	name Human- readable name lowest ex-
				label String, required minimum	name Human- readable name lowest ex- pected
				label String, required minimum	name Human- readable name lowest ex- pected value of
				label String, required minimum	name Human- readable name lowest ex- pected value of a color
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis under
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis under usual
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum-
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances
				label String, required minimum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum-
				label String, required minimum Number, required	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex-
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual
				label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- tances
		non matching output	This state of the Switching Outputs is an	label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual
		non_matching_output	This state of the Switching Outputs is applied if the gurrently complex color does not	label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- tances
		WantedSwitch-	plied, if the currently sample color does not	label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- tances
		WantedSwitch- ingOutputsState, re-		label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- talong
		WantedSwitch-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> .	label String, required minimum Number, required maximum	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- talong
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output	label String, required minimum Number, required maximum Number, required	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- talong
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output uuid	label String, required minimum Number, required maximum Number, required unique identifier	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- tances
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output uuid UUID (string), pattern: ^[a-f0-9-]+\$, re-	Iabel String, required minimum Number, required maximum Number, required unique identifier (UUID) as defined by	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- tances
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output uuid	label String, required minimum Number, required maximum Number, required unique identifier	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- stances
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output uuid UUID (string), pattern: ^[a-f0-9-]+\$, re-	Iabel String, required minimum Number, required maximum Number, required unique identifier (UUID) as defined by	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- stances
		WantedSwitch- ingOutputsState, re-	plied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output uuid UUID (string), pattern: ^[a-f0-9-]+\$, re-	Iabel String, required minimum Number, required maximum Number, required unique identifier (UUID) as defined by RFC 4122, ITU-T	name Human- readable name lowest ex- pected value of a color along this axis under usual circum- stances highest ex- pected value of a color along this axis under usual circum- stances

Code	Body	application/json			
Code	Body		states Array of any of boolean or null, required Minimum duration (in seconds) of the	List of True/False/Null val- ues describing the wanted states of the Switching Outputs	
		non_match- ing_hold_time HoldTime (number), max- imum 3153600000, re- quired	non_matching_output state being applied to the Switching Outputs of the sensor. This prolonging of a potential non matching event may be useful, if the processing period of a connected actor exceeds the sampling period of the sensor.		
		compensation_settings CompensationSettings, required	The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms. These values can be determined by issuing a POST request against /api/sen-sor/detection-profiles/cur-rent/autogain. The result is a suitable set of compensation settings for this sensor under the current circumstances. The content of this data object is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or intro-		
		sampling settings SamplingSettings, requi-	spection). compensation_settings Sampling Settings describe all details of the sampling process.		
		red	Its attributes may be queried and inspected (e.g. in order to retrieve the current sample rate). Most values stored within the Sampling Settings should not be modified directly. The related API endpoint /api/sensor/detection-profiles/current/autogain should be used instead. The only modifiable attribute within the Sampling Settings is the averages value. It is safe to change it, even though the default values calculated during an autogain operation should be optimal for most detection tasks.		
			sampling_settings		
			led_intensity Number, minimum: 0, maximum: 1, required	relative intensity of the internal emitter during the light phase	
			base_sample_rate SampleRate (number), minimum: 0.01, re- quired	The base sample rate determines the duration of a sampling period. After each sampling period, the gathered data is processed and a new detection result is calculated (e.g. the most suitable <i>Matcher</i> for the given sample). This may affect the state of the Switching Outputs or trigger configured actions. Thus the base sample rate defines the maximum rate of changes for the Switching Outputs. See also the effective sample rate.	

Code	Body	application/json		
			effective_sample_rate SampleRate (number), minimum: 0.01, required	The effective sample rate is the numeric product of the base sample rate and the number of averages. It determines the
				minimum duration that a target needs to be sampled in order to determine its vis- ual appearance cor-
			minimum_wanted_sample_rate	rectly. With the default value of average set to one, this value is equal to the base sample rate. This informational
			SampleRate (number), minimum: 0.01, required	value represents the sample rate that was requested during the most recent Autogain operation. The effective sample rate may deviate from the wanted sample rate,
				if the requested sample rate was not achievable due to limitations of the sensor (e.g. exceeding the supported sample rate) or due to the environment (e.g. not enough light, thus a slower amplification with higher gain was necessary).
			sample_light_phase Boolean, required	defines if the sensor should periodically activate the internal emitter for sampling
			sample_dark_phase Boolean, required	defines if the sensor should periodically deactivate the inter- nal emitter for sam- pling
			averages AverageSampleCount (integer), minimum: 1, required	Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.
			amplification AmplificationLevel (integer), required	The amplification level specifies the internal configuration of an amplifier. This value is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).
		white_reference Array of number, re- quired	The White Reference attribute is used for indicating a custom color balancing. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).	

Code	Body	application/json		
	300,	normalization_constant Array of number, re- quired	Normalization constants are related to the White Reference. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-pro-	
	errors Array of Error, re- quired	Error[]	<pre>files/{itemId}/white-reference).</pre>	
		code String, optional	machine-readable unique error code	
		mapping String, optional	a reference to the parameter that caused the error	
		message String, optional	human-readable error description	
		May return the following LPLC.validation.coll	error codes Lection size exceeded	

1.2.2.17 Remove multiple or all DetectionProfiles

DELETE / sensor / detection-profiles

Remove a selection of DetectionProfiles either based on a given filter argument (if supported for this collection) or remove all DetectionProfiles from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DELETE request against an empty collection or for a filtered DELETE request against a collection without DetectionProfiles matching the filter.

Response

Code	Body	application/json	
200	Properties (ob-		
204	ject)		
	errors	Error[]	
	Array of Error, re-		
	quired		
		code String, optional	machine-readable unique error code
		mapping	a reference to the parameter that caused the error
		String, optional	a following to the parameter that sadded the offer
		message	human-readable error description
		String, optional	·
	data	data	
	Object, required		
		current_profile_id	The sensor can store multiple Detection Profiles, but it can only
		CurrentDetectionProfileID	apply one at a time. The field current profile id contains
		(string), pattern: ^[a-f0-	the UUID of the Detection Profile that is currently used by the sen-
		9-]+\$, required , read-only	sor for its operation. It allows to use the shortcut API endpoint
			/api/sensor/detection-profiles/current instead of
			specifying a Detection Profile by its UUID.

1.2.2.18 Retrieve DetectionProfiles

GET / sensor / detection-profiles

Retrieves a list of available DetectionProfiles

Response

Code	Body	application/json		
200	Proper-			
	ties (ob-			
	ject)			
	data	data		

				1	T	
Code	Body	application/json				
	Object,					
	requi-					
	red					
		detection-profiles	DetectionProfile[]			
		Array of Detection-				
		Profile, required				
			uuid	unique identifier (UUID) as		
			UUID (string), pattern:	defined by RFC 4122, ITU-T		
			^[a-f0-9-]+\$, re-	Rec. X.667, and ISO/IEC		
			quired, read-only	9834-8		
			alias	A numerical value that can		
			Alias (integer), required ,	be used to address an item		
			read-only	in a collection. If an alias is		
			read-only	specified alongside an uuid		
				attribute, that alias can be		
				used as an alternative to ad-		
				dress the item in URLs and		
				other protocols like Modbus		
				or serial interfaces.		
			name	Human-readable name of		
ļ			String, required	the Detection Profile		
			colorspace	A colorspace describes the		
	1		Colorspace, required	numeric conversion of colors		
	1			under certain circumstances.		
				Different standardized color-		
				spaces are suitable for diffe-		
				rent detection tasks.		<u> </u>
				colorspace		
				name		
				String, required		
				space id	Unique name	
				ColorspaceID, required	of a color-	
	1				space	
				axes	Color-	
	1			Array of ColorspaceAxis,	spaceAxis[]	
				minimum items: 3, maximum	spaceAxis[]	
				items: 3, required	1.1	1 between
					id	Unique
					String, requi-	name
					red	
					label	Human-
					String, requi-	readable
					red	name
					minimum	lowest ex-
					Number, re-	pected
					quired	value of a
						color along
						this axis
						under
						usual cir-
						cum-
	1					stances
	1				maximum	highest ex-
	1				Number, re-	pected
					quired	value of a
	1				-1	color along
						this axis
	1					under
						usual cir-
	1					cum-
	1					stances
 	-	+	non matching cutrus	This state of the Switching		Sidilles
			non_matching_output WantedSwitch-	This state of the Switching		
	1			Outputs is applied, if the cur-		
	1		ingOutputsState, re-	rently sample color does not		
	1		quired	belong to any of the stored		
ļ				Matchers.		
				non_matching_output		ļ
				uuid	unique identi-	
	1			UUID (string), pattern: ^ [a-	fier (UUID) as	
1	1			f0-9-]+\$, required, read-	defined by	
				only	RFC 4122,	
1	1			_	ITU-T Rec.	
	1	1	1	<u> </u>		

Code	Body	application/json				
					X.667, and ISO/IEC 9834-	
				states Array of any of boolean or null, required	Eist of True/False/Null values describing the wanted states of the	
					Switching Out- puts	
			non_match- ing_hold_time HoldTime (number), maximum 3153600000, required	Minimum duration (in seconds) of the non_matching_output state being applied to the Switching Outputs of the sensor. This prolonging of a potential non matching event may be useful, if the processing period of a connected actor exceeds the sampling period of the sensor.		
			compensation_settings CompensationSettings, required	The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms. These values can be determined by issuing a POST request against /api/sensor/detection-profiles/current/autogain. The result is a suitable set of compensation settings for this sensor under the current circumstances. The content of this data object is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).		
			sampling_settings SamplingSettings, re- quired	compensation_settings Sampling Settings describe all details of the sampling process. Its attributes may be queried and inspected (e.g. in order to retrieve the current sam- ple rate). Most values stored within the Sampling Settings should not be modified directly. The related API endpoint /api/sensor/detec- tion-profiles/cur- rent/autogain should be used instead. The only modifiable attribute within the Sampling Settings is the averages value. It is safe to change it, even though the default values calculated during an au- togain operation should be optimal for most detection tasks. sampling settings		
				led_intensity Number, minimum: 0, maximum: 1, required	relative inten- sity of the in- ternal emitter	

Code	Body	application/json			
Code	Body	αρριισαιιστή 3011		during the light	
				phase	
			base_sample_rate	The base sam-	
			SampleRate (number), mini-	ple rate deter-	
			mum: 0.01, required	mines the du-	
			, ,	ration of a	
				sampling pe-	
				riod.	
				After each	
				sampling pe-	
				riod, the gath-	
				ered data is	
				processed and a new detec-	
				tion result is	
				calculated	
				(e.g. the most	
				suitable	
				Matcher for the	
				given sample).	
				This may affect	
				the state of the	
				Switching Out-	
				puts or trigger	
				configured ac-	
				tions. Thus the	
				base sample rate defines	
				the maximum	
				rate of	
				changes for	
				the Switching	
				Outputs.	
				See also the	
				effective sam-	
				ple rate.	
			effective_sample_rate	The effective	
			SampleRate (number), mini-	sample rate is	
			mum: 0.01, required	the numeric	
				product of the	
				base sample	
				rate and the number of av-	
				erages.	
				It determines	
				the minimum	
				duration that a	
				target needs to	
				be sampled in	
				order to deter-	
				mine its visual	
				appearance	
				correctly.	
				With the de-	
				fault value of	
				average set to one, this value	
				is equal to the	
				base sample	
				rate.	
			minimum_wanted_sam-	This informa-	
			ple_rate	tional value	
			SampleRate (number), mini-	represents the	
			mum: 0.01, required	sample rate	
				that was re-	
				quested dur-	
				ing the most	
				recent Au-	
				togain opera-	
				tion. The effec-	
				tive sample	
				rate may devi- ate from the	
1				ate Irom the	

Cada	Dadu	annliastian/issn	1	T	
Code	Body	application/json			wanted sam-
					ple rate, if the
					requested
					sample rate
					was not
					achievable
					due to limita-
					tions of the
					sensor (e.g.
					exceeding the
					supported
					sample rate)
					or due to the
					environment
					(e.g. not
					enough light,
					thus a slower
					amplification
					with higher
					gain was nec-
ļ					essary).
				sample_light_phase	defines if the
				Boolean, required	sensor should
					periodically
					activate the in-
					ternal emitter
-				sample deukb	for sampling defines if the
				sample_dark_phase	
				Boolean, required	sensor should
					periodically deactivate the
					internal emitter
					for sampling
-				averages	Number of
				AverageSampleCount (inte-	previous sam-
				ger), minimum: 1, required	ples to be av-
				gor), minimum. 1, required	eraged for
					every sam-
					pling result. A
					rolling avera-
					ging algorithm
					is applied to
					the samples.
				amplification	The amplifica-
				AmplificationLevel (integer),	tion level spec-
				required	ifies the inter-
					nal configura-
					tion of an am-
					plifier. This
					value is not
					meant to be
					manipulated
					by regular us-
					ers. It should
					be handled as
					is (stored,
					transmitted
					and applied
					without modifi- cation or intro-
					spection).
-			white reference	The White Reference attrib-	speciforij.
			Array of number, re-	ute is used for indicating a	
			quired	custom color balancing.	
			4411.04	Its content is subject to inter-	
				nal use. Thus it should not	
				be accessed directly, but	
				only through the related API	
				endpoints (e.g. /api/sen-	
				sor/detection-pro-	
				files/{itemId}/white-	
	<u> </u>			reference).	
		1	1	1	l

Code	Body	application/json			
			normalization_constant Array of number, re- quired	Normalization constants are related to the White Reference. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/whitereference).	
		current_profile_id CurrentDetection- ProfileID (string), pattern: ^[a-f0- 9-]+\$, required, read-only	The sensor can store multiple Detection Profiles, but it can only apply one at a time. The field current_profile_id contains the UUID of the Detection Profile that is currently used by the sensor for its operation. It allows to use the shortcut API endpoint /api/sensor/detection-profiles/current instead of specifying a Detection Profile by its UUID.		
	errors Array of Error, re- quired	Error[]			
		code String, optional mapping	machine-readable unique error code a reference to the pa-		
		String, optional message	rameter that caused the error human-readable error		
		String, optional	description		

1.2.2.19 Delete DetectionProfile

DELETE / sensor / detection-profiles / {itemId}

Deletes a single DetectionProfile.

Request

Path Variables	
itemId	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LPLC.not found.collection.item

1.2.2.20 Modify DetectionProfile

PUT / sensor / detection-profiles / {itemId}

Modifies a single DetectionProfile.

Request

Path Variables	
itemId	

String, required

Body	application/json			
Properties	uuid	unique identifier (UUID) as defined by RFC		
(Detection-	UUID (string), pattern:	4122, ITU-T Rec. X.667, and ISO/IEC 9834-8		
Profile)	^[a-f0-9-]+\$, re-			
,	quired, read-only			
	alias	A numerical value that can be used to ad-		
	Alias (integer), required,	dress an item in a collection. If an alias is		
	read-only	specified alongside an uuid attribute, that		
		alias can be used as an alternative to ad-		
		dress the item in URLs and other protocols		
		like Modbus or serial interfaces.		
	name	Human-readable name of the Detection Pro-		
	String, required	file		
	colorspace	A colorspace describes the numeric conver-		
	Colorspace, required	sion of colors under certain circumstances.		
		Different standardized colorspaces are sui-		
		table for different detection tasks.		
		colorspace		
		name		
		String, required		
		space id	Unique name of a	
		ColorspaceID, required	colorspace	
		axes	ColorspaceAxis[]	
		Array of ColorspaceAxis, minimum items: 3,	σοιοισμασεπλίδ[]	
		maximum items: 3, required		
		maximum items. o, required	id	Unique name
			String, required	Onique name
			label	Human-
				readable
			String, required	
				name
			minimum	lowest ex-
			Number, required	pected value
				of a color
				along this axis
				under usual
				circum-
				stances
			maximum	highest ex-
			Number, required	pected value
				of a color
				along this axis
				under usual
				circum-
	man matchin :	This state of the Outletties Outletties		stances
	non_matching_output	This state of the Switching Outputs is applied if the surrently completed as potentials.		
	WantedSwitch-	plied, if the currently sample color does not		
	ingOutputsState, re-	belong to any of the stored Matchers.		
	quired	land matching output		
	-	non_matching_output	iaa talaate	
		uuid	unique identifier	
		<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re-	(UUID) as defined by	
		quired, read-only	RFC 4122, ITU-T	
			Rec. X.667, and	
	-		ISO/IEC 9834-8	
		states	List of	
		Array of any of boolean or null, required	True/False/Null val-	
			ues describing the	
			wanted states of the	
			Switching Outputs	
	non_match-	Minimum duration (in seconds) of the		
	ing_hold_time	non_matching_output state being applied to		
	HoldTime (number), max-	the Switching Outputs of the sensor. This		
	imum 3153600000, re-	prolonging of a potential non matching event		
	quired	may be useful, if the processing period of a		
		connected actor exceeds the sampling pe-		
			Ī	l
		riod of the sensor.		
	compensation_settings	The compensation settings of a Detection		
	CompensationSettings,	The compensation settings of a Detection Profile describe the configuration of internal		
		The compensation settings of a Detection		

	1		T	
Body	application/json			
		These values can be determined by issuing		
		a POST request against /api/sensor/de-		
		tection-profiles/current/autogain.		
		The result is a suitable set of compensation		
		settings for this sensor under the current cir-		
		cumstances.		
		The content of this data object is not meant		
		to be manipulated by regular users. It should		
		be handled as is (stored, transmitted and ap-		
		plied without modification or introspection).		
		,		
		compensation settings		
	sampling_settings	Sampling Settings describe all details of the		
	SamplingSettings, requi-	sampling process.		
	red	Its attributes may be queried and inspected		
		(e.g. in order to retrieve the current sample		
		rate).		
		Most values stored within the Sampling Set-		
		tings should not be modified directly. The re-		
		<pre>lated API endpoint /api/sensor/detec-</pre>		
		tion-profiles/current/autogain		
		should be used instead.		
		The only modifiable attribute within the Sam-		
		pling Settings is the averages value. It is safe		
		to change it, even though the default values		
		calculated during an autogain operation		
		should be optimal for most detection tasks.		
		sampling_settings		
			rolativa intonsity of	
		led_intensity	relative intensity of	
		Number, minimum: 0, maximum: 1, required	the internal emitter	
			during the light	
			phase	
		base_sample_rate	The base sample	
		SampleRate (number), minimum: 0.01, re-	rate determines the	
		quired	duration of a sam-	
			pling period.	
			After each sampling	
			period, the gathered	
			data is processed	
			and a new detection	
			result is calculated	
			(e.g. the most suita-	
			ble <i>Matcher</i> for the	
			given sample). This	
			. ,	
			may affect the state	
			of the Switching Out-	
			puts or trigger con-	
			figured actions. Thus	
			the base sample rate	
			defines the maxi-	
			mum rate of	
			changes for the	
			Switching Outputs.	
			See also the effec-	
			tive sample rate.	
		effective_sample_rate	The effective sample	
		SampleRate (number), minimum: 0.01, re-	rate is the numeric	
		quired	product of the base	
			sample rate and the	
			number of averages.	
			It determines the	
			minimum duration	
			that a target needs	
			to be sampled in or-	
			der to determine its	
			visual appearance	
			correctly.	
			With the default	
			value of average set	
			to one, this value is	
			equal to the base	
			sample rate.	
L	1			

Body	application/json		
Dody		minimum_wanted_sample_rate	This informational
		SampleRate (number), minimum: 0.01, re-	value represents the
		quired	sample rate that was
			requested during the
			most recent Au-
			togain operation.
			The effective sample
			rate may deviate
			from the wanted
			sample rate, if the
			requested sample
			rate was not achiev-
			able due to limita-
			tions of the sensor
			(e.g. exceeding the
			supported sample
			rate) or due to the
			environment (e.g.
			not enough light,
			thus a slower ampli-
1			fication with higher
			gain was neces-
			sary).
		sample_light_phase	defines if the sensor
		Boolean, required	should periodically
		· '	activate the internal
			emitter for sampling
		sample_dark_phase	defines if the sensor
		Boolean, required	should periodically
		Doorean, required	deactivate the inter-
			I I
1			nal emitter for sam-
-	<u> </u>		pling
		averages	Number of previous
		AverageSampleCount (integer), minimum: 1,	samples to be aver-
		required	aged for every sam-
			pling result. A rolling
			averaging algorithm
			is applied to the
			samples.
		amplification	The amplification
		AmplificationLevel (integer), required	level specifies the in-
		,	ternal configuration
			of an amplifier. This
			value is not meant to
			be manipulated by
			regular users. It should be handled
			as is (stored, trans-
			mitted and applied
			without modification
			or introspection).
	white_reference	The White Reference attribute is used for in-	
	Array of number, re-	dicating a custom color balancing.	
	quired	Its content is subject to internal use. Thus it	
		should not be accessed directly, but only	
		through the related API endpoints (e.g.	
		/api/sensor/detection-pro-	
		files/{itemId}/white-reference).	
	normalization_constant	Normalization constants are related to the	
	Array of number, re-	White Reference.	
	quired	Its content is subject to internal use. Thus it	
	7011001	should not be accessed directly, but only	
		through the related API endpoints (e.g.	
		/api/sensor/detection-pro-	
		files/{itemId}/white-reference).	

Examples

```
{
   "name": "#0",
   "uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
```

```
"alias": 1,
"non_matching_hold_time": 0,
"colorspace": {
  "name": "L*a*b*",
  "axes": [
    {
      "id": "L",
      "label": "L*",
      "minimum": 0,
      "maximum": 100
    },
      "id": "a",
      "label": "a*",
      "minimum": -500,
      "maximum": 500
    },
      "id": "b",
      "label": "b*",
      "minimum": -200,
      "maximum": 200
    }
  ],
  "space_id": "Lab"
},
"compensation_settings": {
  "monitor integration": {
    "control": 0.32499998807907104,
    "references": [
      0.7283520102500916,
      0.7442666888237,
      0.7066696286201477
    ]
  } ,
  "use_calibration_samples": true
"normalization constant": [
 237.4935277662995,
  242.62655153828055,
 587.8264132734112
"white reference": [
 95.047,
 100,
 108.883
"non matching output": {
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
   true,
    true,
   true,
    true,
   true,
   true,
    true,
    true
  ]
"sampling_settings": {
  "led_intensity": 1,
  "amplification": 1,
```

```
"sample_light_phase": true,
"minimum_wanted_sample_rate": 1000,
"averages": 1,
"base_sample_rate": 1000,
"sample_dark_phase": true,
"effective_sample_rate": 1000
}
```

Response

Code	Body	application/json			
200	Properties (object)	application/json			
400	Properties (object)				
400					
404	data	A Detection Profile			
		contains a complete			
	DetectionProfile, required	set of sensor set-			
		tings for a given de-			
		tection task.			
		Multiple profiles can			
		be stored in order to			
		switch easily be-			
		tween different de-			
		tection task or for			
		the incremental de-			
		velopment of a re-			
		fined profile.			
		Some attributes of a			
		Detection Profile ex-			
		pose internal details			
		of the sensor, that			
		should be deter-			
		mined indirectly via			
		other means. These			
		attributes are de-			
		scribed only superfi-			
		cially, since they			
		should be handled			
		as is without chang-			
		ing their value or			
<u> </u>		structure.			
		DetectionProfile			
		uuid	unique identifier (UUID) as		
		<u>UUID</u> (string), pat-	defined by RFC 4122, ITU-T		
		tern: ^[a-f0-9-	Rec. X.667, and ISO/IEC 9834-8		
]+\$, required,	3004-0		
		read-only alias	A numerical value that can		
		Alias (integer), re-	be used to address an item		
		quired, read-only	in a collection. If an alias is		
		quirou, rodu-only	specified alongside an uuid		
			attribute, that alias can be		
			used as an alternative to		
			address the item in URLs		
			and other protocols like		
			Modbus or serial interfaces.		
		name	Human-readable name of		
		String, required	the Detection Profile		
		colorspace	A colorspace describes the		
		Colorspace, requi-	numeric conversion of col-		
		red	ors under certain circum-		
			stances. Different standar-		
			dized colorspaces are sui-		
			table for different detection tasks.		
			colorspace		
—			name		
			String, required		
			space_id	Unique name	
			ColorspaceID, required	of a color-	
				space	
				<u> </u>	

Code	Body	application/json			
	•	·	axes Array of ColorspaceAxis, minimum items: 3, maxi- mum items: 3, required	Color- spaceAxis[]	
				id String, requi- red	Unique name
				String, required	Human- readable name
				minimum Number, re- quired	lowest expected value of a color along this axis under usual cir- cum- stances
				maximum Number, re- quired	highest expected value of a color along this axis under usual cir- cum- stances
		non_matching_out- put WantedSwitch- ingOutputsState, re- quired	This state of the Switching Outputs is applied, if the currently sample color does not belong to any of the stored <i>Matchers</i> . non_matching_output		
			uid UUID (string), pattern: ^[a-f0-9-]+\$, required, readonly	unique identi- fier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834- 8	
			states Array of any of boolean or null, required	List of True/False/Null values describ- ing the wanted states of the Switching Out- puts	
		non_match- ing_hold_time HoldTime (number), maximum 3153600000, re- quired	Minimum duration (in seconds) of the non_matching_output state being applied to the Switching Outputs of the sensor. This prolonging of a potential non matching event may be useful, if the processing period of a connected actor exceeds the sampling period of the sensor.		
		compensation_set- tings CompensationSet- tings, required	The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms. These values can be determined by issuing a postule of a Detection of the compensation of the compens		

Code	Body	application/json			
Code	Войу	application/json	quest against /api/sen-		
			sor/detection-pro-		
			files/current/au-		
			togain. The result is a suit-		
			able set of compensation		
			settings for this sensor un-		
			der the current circum-		
			stances.		
			The content of this data ob-		
			ject is not meant to be ma-		
			nipulated by regular users.		
			It should be handled as is		
			(stored, transmitted and ap-		
			plied without modification		
			or introspection). compensation_settings		
		sampling_settings	Sampling Settings describe		
		SamplingSettings,	all details of the sampling		
		required	process.		
		required	Its attributes may be que-		
			ried and inspected (e.g. in		
			order to retrieve the current		
			sample rate).		
			Most values stored within		
			the Sampling Settings		
			should not be modified di-		
			rectly. The related API end-		
			<pre>point /api/sensor/de-</pre>		
			tection-profiles/cur-		
			rent/autogain should be		
			used instead.		
			The only modifiable attrib-		
			ute within the Sampling Set-		
			tings is the averages value.		
			It is safe to change it, even		
			though the default values		
			calculated during an au- togain operation should be		
			optimal for most detection		
			tasks.		
			sampling_settings		
			led_intensity	relative inten-	
			Number, minimum: 0, maxi-	sity of the in-	
			mum: 1, required	ternal emitter	
			-	during the light	
				phase	
			base_sample_rate	The base sam-	
			SampleRate (number), mini-	ple rate deter-	
			mum: 0.01, required	mines the du-	
				ration of a	
				sampling pe-	
				riod. After each	
				Aπer each sampling pe-	
				riod, the gath-	
				ered data is	
				processed and	
				a new detec-	
				tion result is	
				calculated	
				(e.g. the most	
				suitable	
				Matcher for the	
				given sample).	
				This may affect	
				the state of the	
				Switching Out-	
				puts or trigger	
				configured ac- tions. Thus the	
				base sample	
				rate defines	
				the maximum	
				and maximum	

Code	Body	application/json			
Couc	Body	иррпоилоп/јооп		rate of	
				changes for	
				the Switching	
				Outputs.	
				See also the	
				effective sam-	
				ple rate.	
			ffective_sample_rate	The effective	
			ampleRate (number), mini- num: 0.01, required	sample rate is the numeric	
		m	ium: 0.01, required	product of the	
				base sample	
				rate and the	
				number of av-	
				erages.	
				It determines	
				the minimum	
				duration that a	
				target needs to	
				be sampled in order to deter-	
				order to deter- mine its visual	
				appearance	
				correctly.	
				With the de-	
				fault value of	
				average set to	
				one, this value	
				is equal to the	
				base sample	
				rate.	
			inimum_wanted_sam-	This informa-	
			le_rate ampleRate (number), mini-	tional value represents the	
			num: 0.01, required	sample rate	
			idili. 0.01, required	that was re-	
				quested dur-	
				ing the most	
				recent Au-	
				togain opera-	
				tion. The effec-	
				tive sample	
				rate may devi- ate from the	
				wanted sam-	
				ple rate, if the	
				requested	
				sample rate	
				was not	
				achievable	
				due to limita-	
				tions of the	
				sensor (e.g. exceeding the	
				supported	
				sample rate)	
				or due to the	
				environment	
				(e.g. not	
				enough light,	
				thus a slower	
				amplification	
				with higher gain was nec-	
				essary).	
		Q.S	ample_light_phase	defines if the	
		Bo	oolean, required	sensor should	
				periodically	
				activate the in-	
				ternal emitter	
				for sampling	
			ample_dark_phase	defines if the	
	<u> </u>	Вс	oolean, required	sensor should	

Code	Body	application/json		
2000				periodically deactivate the internal emitter
			averages	for sampling Number of
			AverageSampleCount (integer), minimum: 1, required	previous sam- ples to be av- eraged for every sam- pling result. A rolling avera- ging algorithm
				is applied to the samples.
			amplification AmplificationLevel (integer), required	The amplification level specifies the internal configuration of an amplifier. This value is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted
		white vefevence	The White Reference attrib-	and applied without modification or intro- spection).
		white_reference Array of number, re- quired	ute is used for indicating a custom color balancing. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/whitereference).	
		normalization_con- stant Array of number, re- quired	Normalization constants are related to the White Reference. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/whitereference).	
	errors Array of Error, required	Error[]		
		code String, optional	machine-readable unique error code	
		mapping	a reference to the parame-	
		String, optional message	ter that caused the error human-readable error	
		String, optional	description	
		May return the follow LPLC.not found.co		
		_		<u>. </u>

1.2.2.21 Get DetectionProfile

GET / sensor / detection-profiles / {itemId}

Returns a single DetectionProfile.

Request

Path Variables	
itemId	
String, required	

Response

Code	Body	application/json			
200	Properties (ob-				
	ject)				
	data	A Detection Profile contains a			
	<u>DetectionProfile</u> ,	complete set of sensor settings			
	required	for a given detection task.			
		Multiple profiles can be stored			
		in order to switch easily be-			
		tween different detection task or for the incremental develop-			
		ment of a refined profile.			
		Some attributes of a Detection			
		Profile expose internal details of			
		the sensor, that should be de-			
		termined indirectly via other			
		means. These attributes are de-			
		scribed only superficially, since			
		they should be handled as is			
		without changing their value or			
		structure.			
-		DetectionProfile uuid	unique identifier (UUID) as		
		UUID (string), pattern: ^ [a-	defined by RFC 4122, ITU-T		
		f0-9-]+\$, required , read-only	Rec. X.667, and ISO/IEC		
		10 5 j. v, regained, road only	9834-8		
		alias	A numerical value that can		
		Alias (integer), required, read-	be used to address an item		
		only	in a collection. If an alias is		
			specified alongside an uuid attribute, that alias can be		
			used as an alternative to		
			address the item in URLs		
			and other protocols like		
			Modbus or serial interfaces.		
		name	Human-readable name of		
		String, required	the Detection Profile		
		colorspace	A colorspace describes the		
		Colorspace, required	numeric conversion of col- ors under certain circum-		
			stances. Different standar-		
			dized colorspaces are sui-		
			table for different detection		
			tasks.		
			colorspace		
			name String required		
-			String, required space_id	Unique name	
			ColorspaceID, required	of a color-	
				space	
			axes	Color-	
			Array of ColorspaceAxis,	spaceAxis[]	
			minimum items: 3, maxi-		
			mum items: 3, required	: al	Unio
				id String, requi-	Unique name
				red	Hame
				label	Human-
				String, requi-	readable
				red	name
				minimum	lowest
				Number, re- quired	expected value of a
				quireu	color
L	1	1	l	Î.	1 00.01

non_matching_output WantedSwitchingOutputsState,	This state of the Switching Outputs is applied, if the	maximum Number, re- quired	along this axis under usual cir- cum- stances highest expected value of a color along this axis under usual cir- cum- stances
required	currently sample color does not belong to any of the stored <i>Matchers</i> .		
	non_matching_output uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, readonly	unique identi- fier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834- 8	
	states Array of any of boolean or null, required	List of True/False/Null values describ- ing the wanted states of the Switching Out- puts	
non_matching_hold_time HoldTime (number), maximum 3153600000, required	Minimum duration (in seconds) of the non_match-ing_output state being applied to the Switching Outputs of the sensor. This prolonging of a potential non matching event may be useful, if the processing period of a connected actor exceeds the sampling period of the sensor.		
compensation_settings CompensationSettings, required	The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms. These values can be determined by issuing a POST request against /api/sensor/detection-profiles/current/autogain. The result is a suitable set of compensation settings for this sensor under the current circumstances. The content of this data object is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).		

 	,	
sampling_settings SamplingSettings, required	Sampling Settings describe all details of the sampling process. Its attributes may be queried and inspected (e.g. in order to retrieve the current sample rate). Most values stored within the Sampling Settings should not be modified directly. The related API endpoint /api/sensor/detection-profiles/current/autogain should be used instead. The only modifiable attribute within the Sampling Settings is the averages value. It is safe to change it, even though the default values calculated during an autogain operation should be optimal for most detection tasks.	
	sampling_settings	
	led intensity Number, minimum: 0, maximum: 1, required	relative inten- sity of the in- ternal emitter during the light phase
	base_sample_rate SampleRate (number), minimum: 0.01, required	The base sample rate determines the duration of a sampling period. After each sampling period, the gathered data is processed and a new detection result is calculated (e.g. the most suitable Matcher for the given sample). This may affect the state of the Switching Outputs or trigger configured actions. Thus the base sample rate defines the maximum rate of changes for the Switching Outputs. See also the effective sample rate.
	effective_sample_rate SampleRate (number), minimum: 0.01, required	The effective sample rate is the numeric product of the base sample rate and the number of averages. It determines the minimum

			duration that a
			target needs to
			be sampled in
			order to deter-
			mine its visual
			appearance
			correctly.
			With the de-
			fault value of
			average set to
			one, this value
			is equal to the
			base sample
			rate.
		minimum_wanted_sam-	This informa-
		ple_rate	tional value
		SampleRate (number), mini-	represents the
		mum: 0.01, required	sample rate
			that was re-
			quested dur-
1			ing the most
			recent Au-
1			togain opera-
1			
			tion. The effec-
1			tive sample
			rate may devi-
			ate from the
1			wanted sam-
			ple rate, if the
			requested
			sample rate
			was not
			achievable
			due to limita-
			tions of the
			sensor (e.g.
			exceeding the
			supported
			sample rate)
			or due to the
			environment
			(e.g. not
			enough light,
			thus a slower
			amplification
			with higher
			gain was nec-
			essary).
I		sample_light_phase	defines if the
1		Boolean, required	sensor should
1			periodically
1			activate the in-
			ternal emitter
1			for sampling
-		sample_dark_phase	defines if the
1			sensor should
		Boolean, required	
1			periodically
			deactivate the
1			internal emitter
			for sampling
		averages	Number of
1		AverageSampleCount (inte-	previous sam-
1		ger), minimum: 1, required	ples to be av-
1			eraged for
1			every sam-
1			pling result. A
1			rolling avera-
			ging algorithm
1			is applied to
			the samples.
1		amplification	The amplifica-
1		AmplificationLevel (integer),	tion level spec-
1		required	ifies the inter-
	<u>'</u>		I

	1	T		
				nal configura-
				tion of an am-
				plifier. This
				value is not
				meant to be
				manipulated
				by regular us-
				ers. It should
				be handled as
				is (stored,
				transmitted
				and applied
				without modifi-
				cation or intro-
				spection).
		white_reference	The White Reference attrib-	
		Array of number, required	ute is used for indicating a	
			custom color balancing.	
			Its content is subject to in-	
			ternal use. Thus it should	
			not be accessed directly,	
			but only through the related	
			API endpoints (e.g.	
			/api/sensor/detec-	
			tion-pro-	
			files/{itemId}/white-	
			reference).	
		normalization_constant	Normalization constants are	
		Array of number, required	related to the White Refer-	
			ence.	
			Its content is subject to in-	
			ternal use. Thus it should	
			not be accessed directly,	
			but only through the related	
			API endpoints (e.g.	
			/api/sensor/detec-	
			tion-pro-	
			files/{itemId}/white-	
	orrors	Error[]	reference).	
	errors Array of Error, re-	Error[]		
	auired			
	quireu	code	machine-readable unique	
		String, optional	error code	
		mapping	a reference to the parame-	
		String, optional	ter that caused the error	
		message	human-readable error	
		String, optional	description	
		May return the following error		
		LPLC.not found.collectio		
L	I	TITC. HOC_TOURIG. COTTECCTO	11 • 1 00111	

1.2.2.22 Start Autogain Procedure

POST / sensor / detection-profiles / {itemId} / autogain

Execute the autogain procedure in order to determine suitable sampling properties for the current environment. The resulting sampling setup is applied automatically. These new settings are in effect as soon as the response is sent.

The autogain procedure initiates a dynamic recalibration of the internal emitter and all compensation processes. It results in quick changes or flashing of the internal emitter (if enabled). The operation is usually finished within 15 seconds. The response is sent after all related activities are completed. Later requests for a sample will return values based on the adjusted sampling settings.

Request

Path Variables	
itemId	
String, required	
Body	application/json

Properties (AutogainSettings)	
level Number, default: 0.8, minimum: 0.01, maximum: 1, optional	Target value for the auto-gain procedure
minimum_sample_rate SampleRate (number), minimum: 0.02, optional	Desired sample rate (the default is the current sample rate)
enable_internal_emitter Boolean, default: true, optional	controls the power of the internal light source
enable_ambient_light_compensation Boolean, default: true, optional	Control the ambient light compensation procedure. This setting is only relevant if <code>enable_internal_emitter</code> is set to true. The ambient light compensation leads to a pulsed usage of the internal light emitter. Samples are collected for alternating light and dark phases. This allows to calculate a color sample of the target excluding any optical interference from external light sources. You should not disable ambient light compensation unless the optical path is perfectly isolated. Otherwise external light will inevitably interfere with the color sampling.
averages AverageSampleCount (integer), minimum: 1, optional	Anzahl zu mittelnder vorheriger Stichproben für jedes Stichprobenergebnis.

Examples

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
```

Response

Code	Body	application/json	
200	Properties (object)		
400	data sampling_settings SamplingSettings, required	Sampling Settings describe all details of the sampling process. Its attributes may be queried and inspected (e.g. in order to retrieve the current sample rate). Most values stored within the Sampling Settings should not be modified directly. The related API endpoint /api/sensor/detection-profiles/current/autogain should be used instead. The only modifiable attribute within the Sampling Settings is the averages value. It is safe to change it, even though the default values calculated during an autogain operation should be optimal for	
-		most detection tasks.	
		SamplingSettings led_intensity Number, minimum: 0, maximum: 1, required	relative intensity of the internal emitter during the light phase
		base_sample_rate SampleRate (number), minimum: 0.01, required	The base sample rate determines the duration of a sampling period. After each sampling period, the gathered data is processed and a new detection result is calculated (e.g. the most suitable <i>Matcher</i> for the given sample). This may affect the state of the Switching Outputs or trigger configured actions. Thus the base sample rate defines the maximum rate of changes for the Switching Outputs. See also the <i>effective sample rate</i> .
		effective_sample_rate SampleRate (number), minimum: 0.01, required	The effective sample rate is the numeric product of the base sample rate and the number of averages.

Code	Body	application/json	
5046			It determines the minimum duration that a target needs to be sampled in order to determine its visual appearance correctly. With the default value of average set to one, this value is equal to the base sample rate.
		minimum_wanted_sample_rate SampleRate (number), minimum: 0.01, required	This informational value represents the sample rate that was requested during the most recent <i>Autogain</i> operation. The effective sample rate may deviate from the wanted sample rate, if the requested sample rate was not achievable due to limitations of the sensor (e.g. exceeding the supported sample rate) or due to the environment (e.g. not enough light, thus a slower amplification with higher gain was necessary).
		sample_light_phase Boolean, required	defines if the sensor should periodi- cally activate the internal emitter for sampling
		sample_dark_phase Boolean, required	defines if the sensor should periodi- cally deactivate the internal emitter for sampling
		averages AverageSampleCount (integer), minimum: 1, required	Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.
		amplification AmplificationLevel (integer), required	The amplification level specifies the internal configuration of an amplifier. This value is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).
	compensation_settings CompensationSettings, required	The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms. These values can be determined by issuing a POST request against /api/sensor/detection-profiles/current/autogain. The result is a suitable set of compensation settings for this sensor under the current circumstances. The content of this data object is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).	
	errors Array of Error, required	Error[]	
	ransy or airor, required	code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description
	May return the following error con LCOL.autogain LCOL.autogain.invalid_targ LCOL.autogain.invalid_samp LPLC.validation.boolean	et_level	

1.2.2.23 Query custom White Reference

GET / sensor / detection-profiles / {itemId} / white-reference

Verify the existence of a custom White Reference. A successful response (HTTP Status 200) indicates that a custom White Reference is in use. The *not found* response (HTTP Status 404) indicates that the factory default White Reference is used instead.

The detailed content of the response is not relevant. Instead the related normalization_constants field of the Detection Profile is adjusted based on the current White Reference.

Request

Path Variables	
itemId	
String, required	

Response

Code	A custom White Reference i	A custom White Reference is in use.		
200	Body	application/json		
404				
	Properties (object)			
	data Array of number, required	The White Reference attribute is used for indicating a custom color balancing. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/whitereference).		
	errors	Error[]		
	Array of Error, required		and the same debter and the same	
		Code	machine-readable unique	
		String, optional	error code	
		mapping	a reference to the param-	
		String, optional	eter that caused the error	
		message	human-readable error	
		String, optional	description	

1.2.2.24 Sample a custom White Reference

POST / sensor / detection-profiles / {itemId} / white-reference

Apply a custom White Reference for the color handling of the sensor. The currently sampled color is used for calculating the White Reference. You should pick a neutral white target for this action.

Please note that the change of the White Reference is not in effect immediately. Thus you should wait for three seconds, before sampling new values.

Request

Path Variables	
itemId	
String, required	

Response

Code	Body	application/json	
200	Properties (object)		
406			
	data Array of number, required	The White Reference attribute is used for indicating a custom color balancing. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).	
	errors	Error[]	

Array of Error, required		
	code	machine-readable
	String, optional	unique error code
	mapping	a reference to the pa-
	String, optional	rameter that caused the
		error
	message	human-readable error
	String, optional	description
May return the following e		
LCOL.white_reference	.too_dark	

1.2.2.25 Lese Aktionsauslöser aus

GET / sensor / action-triggers

The sensor can be programmed to react on specific external or internal events. The available actions can be either triggered via trigger input lines or via API requests. This allows customized behaviour, e.g. teaching colors via an external button.

Multiple ActionTrigger items can be created. Each ActionTrigger assigns one or more actions to a specific event (see trigger_sources in /api/sensor/capabilities). Multiple ActionTriggers may refer to the same event (see order of execution below for details).

The actions assigned to an *ActionTrigger* are evaluated separately. Thus it is possible to specify the same action (even with the same parameters) multiple times. The list of actions for an *ActionTrigger* may be empty.

The actions within a single *ActionTrigger* are executed successively according to the order of the list items. The order of execution among multiple *ActionTrigger* items is undefined.

Trigger Events describing a state *change* (e.g. trigger_0_edge_rising) are emitted only once at the moment of the state change. Thus attached actions are executed only once for every state change.

Trigger Events describing a *state* (e.g. trigger_0_level_low) are emitted continuously as long as the state is active. The actions of an *ActionTrigger* attached to such a Trigger Event are executed periodically. After every execution of such an *ActionTrigger* further executions are skipped for a period of one second or until the next state change (whichever comes first). The only exception for this periodically executed actions is the *enable_switching_output* action. If this action is attached to a *state*, then it is re-evaluated whenever the hold time of the currently detected macher expires (i.e. for matchers with hold time zero: in every sample period).

See actions for a list of supported actions.

Request

Query Parameters	
event	Filter ActionTriggers by the given event name (e.g. trig-
<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	ger_0_edge_rising).

Response

Code	Body	application/json			
200	Properties (object)				
	data	data			
	Object, required				
		action-triggers	ActionTrigger[]		
		Array of ActionTrig-			
		ger, required			
			uuid	unique identifier (UUID) as de-	
			UUID (string), pat-	fined by RFC 4122, ITU-T Rec.	
			tern: ^[a-f0-9-	X.667, and ISO/IEC 9834-8	

]+\$, required , read-only		
		event TriggerEventName (string), required	Any of the event names provided by /api/sensor/capabilities (attribute trigger sources) is allowed.	
		actions Array of Action, required	List of actions to be executed af-	
			Action[] name String, required	Unique name of the action
			arguments Object, required	arguments
	erros Array of Error, re- quired	Error[]		
		code String, optional	machine-readable unique error code	
		mapping String, optional	a reference to the parameter that caused the error	
		message String, optional	human-readable error description	

1.2.2.26 Remove multiple or all Action Triggers

DELETE / sensor / action-triggers

Remove a selection of ActionTriggers either based on a given filter argument (if supported for this collection) or remove all ActionTriggers from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DELETE request against an empty collection or for a filtered DELETE request against a collection without ActionTriggers matching the filter.

Request

Query Parameters	
event	Delete all ActionTriggers assigned to a given event name
<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	(e.g. trigger_0_edge_rising).

Response

Code	
204	The empty response indicates success

1.2.2.27 Create ActionTriggers

POST / sensor / action-triggers

Create a new ActionTrigger.

All supported data attributes in the body of the request are optional.

Request

Body	application/json		
Properties (Action Trigger)	uuid	unique identifier (UUID) as de-	
	<u>UUID</u> (string), pattern:	fined by RFC 4122, ITU-T Rec.	
	^[a-f0-9-]+\$, required,	X.667, and <u>ISO/IEC 9834-8</u>	
	read-only		
	event	Any of the event names pro-	
	TriggerEventName (string),	<pre>vided by /api/sensor/capa-</pre>	
	required	bilities (attribute trig-	
		ger sources) is allowed.	
	actions	List of actions to be executed	
	Array of Action, required	after the given event.	
		Action[]	

nai	me	Unique name of the action
Str	ring, required	
arç	guments	arguments
Ob	oject, required	-

Examples

Response:

Code	Body	application/json		
200 100	Properties (object)			
	data ActionTrigger, required	An Action Trigger assigns a given set of actions with an event. At the end of each sample period, all events are evaluated. All corresponding actions are executed afterwards.		
		ActionTrigger		
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re- quired, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
		event	Any of the event names pro-	
		TriggerEventName	vided by /api/sensor/ca-	
		(string), required	pabilities (attribute trig-	
		(),	ger sources) is allowed.	
		actions	List of actions to be executed	
		Array of Action, required	after the given event.	
			Action[]	
			name	Unique
			String, required	name o the ac- tion
			arguments	argu-
			Object, required	ments
	erros Array of Error, required	Error[]		
		code	machine-readable unique er-	
		String, optional	ror code	
		mapping	a reference to the parameter	
		String, optional	that caused the error	
		message	human-readable error	
		String, optional	description	
	May return the following error of LPLC.validation.collection	codes		

1.2.2.28 Delete ActionTrigger

DELETE / sensor / action-triggers / {itemId}

Deletes a single ActionTrigger.

Request

Path Variables

itemId	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LPLC.not found.collection.item

1.2.2.29 Modify ActionTriger

PUT / sensor / action-triggers / {itemId}

Modifies a single ActionTrigger.

Request

Path Variables		
itemId		
String, required		
Body	application/json	
Properties (Action Trigger)		
uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
event TriggerEventName (string), required	Any of the event names provided by /api/sensor/capabilities (attribute trigger sources) is allowed.	
actions Array of Action, required	List of actions to be executed after the given event.	
	Action[] name String, required	Unique name of the action
	arguments Object, required	arguments

Examples

Response

Code	Body	application/json		
200	Properties (object)			
400				
404				
	data	An Action Trigger assigns a given set		
	ActionTrigger, required	of actions with an event.		
		At the end of each sample period, all		
		events are evaluated. All correspond-		
		ing actions are executed afterwards.		
		ActionTrigger		
		uuid	unique identifier (UUID) as de-	
		<u>UUID</u> (string), pattern: ^ [a-f0-9-	fined by RFC 4122, ITU-T Rec.	
] +\$, required, read-only	X.667, and ISO/IEC 9834-8	

	event	Any of the event names pro-	
	TriggerEventName (string), required	<pre>vided by /api/sensor/capa-</pre>	
		bilities (attribute trig-	
		ger sources) is allowed.	
	actions	List of actions to be executed	
	Array of Action, required	after the given event.	
		Action[]	
		name	Unique
		String, required	name of
			the action
		arguments	arguments
		Object, required	
errors Array of Error, required	Error[]		
	code	machine-readable unique error	
	String, optional	code	
	mapping	a reference to the parameter	
	String, optional	that caused the error	
	message	human-readable error descrip-	
	String, optional	tion	
May return the following			
LPLC.not_found.coll	ection.item		

1.2.2.30 Get ActionTrigger

GET / sensor / action-triggers / {itemId}

Returns a single ActionTrigger

Request

l e	
Path Variables	
itemId	
String, required	

Response

Code	Body	application/json		
200	Properties (object)			
	data ActionTrigger, required	An Action Trigger assigns a given set of actions with an event. At the end of each sample period, all events are evaluated. All corresponding actions are executed afterwards.		
		ActionTrigger		
		uuid <u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
		event TriggerEventName (string), required	Any of the event names provided by /api/sensor/capabilities (attribute trigger_sources) is allowed.	
		actions	List of actions to be executed	
		Array of Action, required	after the given event.	
			Action[]	
			name String, required	Unique name of the action
			arguments Object, required	arguments
	erros Array of Error, required	Error[]	, , ,	
		code	machine-readable unique er-	
		String, optional	ror code	
		mapping String, optional	a reference to the parameter that caused the error	
		message	human-readable error	
		String, optional	description	

May return the following error codes	
LPLC.not found.collection.item	

1.2.2.31 Get sensoric Capabilities

GET / sensor / capabilities

Response

Code		application/json			
200	Properties (ob- ject)				
	data	Provide access to the			
	SensorCapabi-	sensoric details sup-			
	lities, required	ported by this device			
	ntico, regained	(e.g. colorspaces, input			
		and output lines,).			
		SensorCapabilities			
		maximum sample rate	the maximum sample		
		Integer, required	rate the sensor sup-		
		integer, required	ports		
		tolerances	List of tolerance		
		Array of ColorTolerance	specifications sup-		
		(union), required	ported by the sensor		
			InfiniteColorToler-		
			ance	P 1	
			limits	limits	
			Object, required		
			shape	Name of the geometrical	
			ToleranceShap-	shape of the tolerance. The	
			eName (string), re-	supported tolerance shapes	
			quired	can be retrieved via	
				/api/sensor/capabili-	
				ties .	
			SphereColorToler- ance		
			limits	limits	
			Object, required	radius	
				Numer, required	
			shape	Name of the geometrical	
			ToleranceShap-	shape of the tolerance. The	
			eName (string), re-	supported tolerance shapes	
			quired	can be retrieved via	
			quiica	/api/sensor/capabili-	
				ties.	
			CylinderColorTole-	0100.	
			rance		
			limits	limits	
			Object, required	radius	
			Object, required	Number, required	
				half height	
				Number, required	
			shape	Name of the geometrical	†
			ToleranceShape-	shape of the tolerance. The	
			Name (string), requi-	supported tolerance shapes	
			red	can be retrieved via	
			150	/api/sensor/capabili-	
				ties.	
			BoxColorTolerance	C±C5.	
			limits	limits	+
			Object, required	half_edges	
			Object, required	Array of number, minimum	
				items: 3, maximum items: 3,	
				required	
			shana	Name of the geometrical	
			shape Toloropochopo		
			ToleranceShape-	shape of the tolerance. The	
			Name (string), requi-	supported tolerance shapes	
			red	can be retrieved via	
				/api/sensor/capabili-	
]	ties .	

Code	Body	application/json			
	•	output drivers	List of supported		
		Array of Switch-	electrical output driv-		
		ingOutputDriver (string),	ers		
		required trigger sources	Beinhaltet die Liste		
		Array of TriggerSource,	verfügbarer Auslöse-		
		required	quellen mit ihrem da-		
		'	zugehörigen Auslöse-		
			fall. Auslösefälle kön-		
			nen zum Ausführen		
			bestimmter Aktionen		
			automatisiert werden. TriggerSource[]		
			name	Name of the trigger input	
			String, required	Name of the tingger input	
			events	TriggerEvent []	
			Array of TriggerEvent,		
			required		
				name	
				TriggerEventName (string),	
-		output_pin_count	Number of available	required	
		Integer, required	switching output lines		
		Actions	Deprecated: use		
		Array of Action, re-	/api/actions in-		
		quired, Deprecated	stead		
			Action[]		
			name	Unique name of the action	
			String, required	orguments.	
			arguments Object, required	arguments	
		colorspaces	List of supported co-		
		Array of Colorspace, required	lorspaces.		
			Colorspace[]		
			name		
			String, required		
			space_id ColorspaceID, requi- red	Unique name of a colorspace	
			axes	ColorspaceAxis[]	
			Array of Colorspace-	-	
			Axis, minimum items:		
			3, maximum items: 3,		
			required	id	Unique
				String, required	name
				label	Human-
				String, required	readable
					name
				minimum Number required	lowest ex-
				Number, required	pected
					value of a color along
					this axis
					under
					usual cir-
					cum-
		<u> </u>		maximum	stances
				maximum Number, required	highest ex- pected
					1 200.00
					value of a
					color along
					color along this axis
					color along this axis under
					color along this axis under usual cir-
					color along this axis under usual cir- cum-
		colorspace toler-	The evaluation of tol-		color along this axis under usual cir-
		colorspace_toler- ance_maps	The evaluation of tolerances against posi-		color along this axis under usual cir- cum-
					color along this axis under usual cir- cum-

	T = -		T		1
Code	Body	application/json			
			depends on the cur-		
			rently configured col-		
			orspace. For example		
			the tolerance attribute		
			"half height" refers to		
			the brightness-re-		
			lated axis of a color-		
			space (e.g. "L*" for		
			the "Lab*" color-		
			space) and is used		
			for the height of the		
			cylindrical tolerance		
			shape and the first		
			edge of the box toler-		
			ance shape.		
			The hue-related at-		
			tributes (e.g. "a" and		
			"b" for the "Lab*" col-		
			orspace) are used for		
			. ,		
			the "radius" of a cylin- der tolerance shape		
			'		
			and the second and		
			third edges of the		
			box tolerance shape.		
			The colorspace_toler-		
			ance_maps define		
			these relationships		
			between colorspaces		
			and tolerances.		
			ColorspaceTole-		
			ranceMap[]		
				Unique name of a coloronace	
			colorspace_id	Unique name of a colorspace	
			ColorspaceID		
			(string), required		
			tolerance_shape	Name of the geometrical	
			ToleranceShap-	shape of the tolerance. The	
			eName (string), re-	supported tolerance shapes	
			quired	can be retrieved via	
				/api/sensor/capabili-	
				ties .	
			limits_axes_map	limits_axes_map	
			Object, required		
				half_height	
				Array of string, optional	
				half_edges	
				Array of string, optional	
				radius	
				Array of string, optional	
		settings_categories	List of categories that	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
		Array of string, required	can be selected dur-		
			ing import to control		
			which settings should		
			be applied. See the		
			documentation for		
			the POST request to		
	1	<u> </u>	`/api/seetings.		-
		maximum_detecta-	Maximum number of		
		bles_count	color positions (De-		
		Integer, required	tectable) to be stored		
			in a detection profile.		
		maximum_match-	Maximum number of		
		ers_count	detection results		
		Integer, required	(Matcher) be stored		
			in a detection profile.		
	errors	Error[]			
	Array of Error,				
	required				
		code	machine-readable		
		String, optional	unique error code		
	†	mapping	a reference to the pa-		İ
		String optional			
		String, optional	rameter that caused the error		

Code	Body	application/json		
		message	human-readable error	
		String, optional	description	

1.2.2.32 Retrieve Colorspaces

GET / sensor / colorspaces

Retrieves a list of available Colorspaces.

Response

Code	Body	application/json			
200	Properties (ob- ject)				
	data Object, required	data			
	, , ,	colorspaces Array of Color- space, required	Colorspace[]		
			name String, required		
			space_id ColorspaceID, re- quired	Unique name of a colorspace	
			axes Array of Color- spaceAxis, mini- mum items: 3, maximum items: 3, required	ColorspaceAxis[]	
				id String, required	Unique name
				label String, required	Human-readable name
				minimum Number, requi- red	lowest expected value of a color along this axis under usual circumstances
				maximum Number, requi- red	highest expected value of a color along this axis un- der usual circum- stances
	errors Array of Error, required	Error[]			
		code String, optional	machine-readable unique error code		
		mapping String, optional	a reference to the parameter that caused the error		
		message String, optional	human-readable error description		

1.2.2.33 Get Colorspace

GET / sensor / colorspaces / {space_id}

Returns a single Colorspace.

Request

Path Variables	
space_id	
String, required	

Response

Code	Body	application/json		
200	Properties (object)			
	data	A colorspace describes the numeric		
	Colorspace, required	conversion of colors under certain		
		circumstances. Different standar-		
		dized colorspaces are suitable for		
		different detection tasks.		
		colorspace		
		name		
		String, required		
		space_id	Unique name of a	
		ColorspaceID, required	colorspace	
		axes	ColorspaceAxis[]	
		Array of ColorspaceAxis, minimum		
		items: 3, maximum items: 3, re-		
		quired		
			id	Unique name
			String, required	
			label	Human-readable
			String, required	name
			minimum	lowest expected
			Number, required	value of a color
				along this axis
				under usual cir-
				cumstances
			maximum	highest expected
			Number, required	value of a color
				along this axis
				under usual cir-
				cumstances
	errors	Error[]		
	Array of Error, required			
		code	machine-readable	
		String, optional	unique error code	
		mapping	a reference to the	
		String, optional	parameter that	
			caused the error	
		message	human-readable er-	
		String, optional	ror description	
		May return the following error code		
		LPLC.not_found.collection.it	em	

1.2.3 **Settings**

Management of all device settings

1.2.3.1 Export Settings

GET / settings

Export the complete configuration of the device.

Response

Code		
200	Textual representation of the complete device configuration. This configuration export can be	
	uploaded to the same or another sensor without modifications.	
	The configuration data is encoded as Base64. The Base64 encoding is supposed to signal, that	
	the configuration data dump is not meant to be manipulated or inspected automatically. You	
	may not rely on a specific internal structure as it may change over time without further notice.	
	Body	text/plain

1.2.3.2 Upload Settings

POST / settings

Replace the device configuration with the one being uploaded.

If you only want to partially import the settings you can do so by specifying one or more import categories. If you don't specify at least one import category the default is to import all of them.

Import Category Key	Will import
import_category_access	Users, roles and permissions
import_category_firmware	Firmware settings (like the branch, but not the firmware it-
	self)
import_category_keybad	Keypad settings
import_category_network	Network configuration
import_category_outputs	Output driver
import_category_sensor	Matchers (color groups), detectables (colors), colorspace,
	sample configuration
import_category_system	System settings (like timezone and hostname)

A machine-readable list of import categories is returned by the /api/sensor/capabilities end $point \ under \ the \ {\tt settings_categories} \ key.$

Settings exported from older firmwares will automatically be migrated to the new format required by the sensor. In case a migration fails the response will contain the LPLC.migration.execution failed error code. Settings from more recent firmwares than the one used on the sensor MAY fail on import, if the settings format is no longer compatible. In that case the response will contain the LPLC.migration.future version error code.

Request

Body	multipart/form-data
Properties (object)	
Settings_file	The settings file containing a Base64 encoded configuration
File, required	dump. See GET request for /api/settings.
/^import_category+/	Import only a specific subset of the configuration.
Any, optional	

Response

Code	
204	The empty response indicates success.
400	May return the following error codes
	LPLC.format.encoding.utf8
	LPLC.format.malformed.base64
	LPLC.validation.missing input
	LPLC.format.malformed.json

LPLC.format.malformed.json.not dict
LPLC.migration.future version
LPLC.migration.execution failed

1.2.3.3 Import Settings

PUT / settings

Replace the complete device configuration with the uploaded configuration dump.

The content to be uploaded can be retrieved via a GET request on /api/settings. This Base64 encoded configuration dump is expected as the request body.

Settings exported from older firmwares will automatically be migrated to the new format required by the sensor. In case a migration fails the response will contain the LPLC.migration.execution_failed error code. Settings from more recent firmwares than the one used on the sensor MAY fail on import, if the settings format is no longer compatible. In that case the response will contain the LPLC.migration.future version error code.

Request

_	· ·	
Γ	Body	text/plain
L	,	10717/010111

Response

Code		
204	The empty response indicates success.	
400	May return the following error codes	
	LPLC.format.encoding.utf8	
	LPLC.format.malformed.base64	
	LPLC.format.malformed.json	
	LPLC.format.malformed.json.not_dict	
	LPLC.migration.future_version	
	LPLC.migration.execution_failed	

1.2.3.4 Reset Settings

DELETE / settings

Reset the device configuration to the factory defaults.

Response

Code	
204	The empty response indicates success

1.2.4 System

Manage the device's system settings:

1.2.4.1 Request System Settings

GET / system

Response

Code	Body	application/json	
200	Properties (ob-		
	ject)		
	data	SystemSettings	
	SystemSettings,		
	required		
		hostname	Human-readable name identifying
		Hostname, pattern: ^ (?: [a-zA-Z0-9] (?: [a-zA-	the device in the network
		$Z0-9-$ * [a-zA-Z0-9])?\.)*[a-zA-Z0-	
		9] (?:[a-zA-Z0-9\-]*[a-zA-Z0-9]) ?\$, optional	
		uptime	The current system uptime in sec-
		any of number or null, optional, read-only	onds. Though highly unlikely can be

		nil in case the system reported an invalid value.
errors Array of Error, required	Error []	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description

1.2.4.2 Modify System Settings



PUT / system

Request

Body	application/json
Properties (SystemSettings)	
hostname	Human-readable name identifying the device in the network
Hostname, pattern: ^ (?: [a-zA-Z0-9] (?: [a-zA-Z0-9\-	
]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-	
9\-]*[a-zA-Z0-9])?\$, optional	
uptime	The current system uptime in seconds. Though highly un-
any of number or null, optional, read-only	likely can be nil in case the system reported an invalid value.

Examples

```
"hostname": "cfo-7454232361"
```

Response

Code	Body	application/json	
200 400 500	Properties (object)		
	data SystemSettings, required	SystemSettings	
		hostname Hostname, pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9])?\$, optional	Human-readable name identifying the device in the network
		uptime any of number or null, optional, read-only	The current system uptime in seconds. Though highly unlikely can be nil in case the system reported an invalid value.
	errors Array of Error, required	Error []	
		code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description
	May return the follo	owing error codes	

1.2.4.3 Reset to Factory Firmware and Settings

POST / system / factory-reset

Reset the sensor's firmware to its factory default and initiate a reboot. After completion the sensor will use its original ("recovery") firmware and all settings are reset to their defaults. The recovery firmware can be upgraded via "/firmware/upgrade-from-current".

In case you **only** want to reset the settings it is sufficient to send a **DELETE** request to the /api/settings **endpoint**.

Response

Code	
204	The empty response indicates success
500	
	May return the following error codes
	LPLC.system.action failed

1.2.4.4 Initiate Reboot

POST / system / reboot

Reboots the device.

The software-triggered reboot is the more polite method to shutdown the sensor compared to unplugging the power supply. However the latter is safe as well.

Response

Code	
204	The empty response indicates success
500	
	May return the following error codes
	LPLC.system.action failed

1.2.4.5 Get time settings

GET / system / time

Response

Code	Body	application/json	
200	Properties (object)		
	data	SystemTimeSettings	
	SystemTimeSettings, required		
		now	current time from the per-
		Timestamp (string), optional	spective of the sensor
		timezone	currently configured time-
		String, optional	zone
		ntp_servers	one or more network time
		Array of string, optional	servers
		default_ntp_servers	preconfigured network time
		Array of string, optional, reaed-only	servers
	errors	Error []	
	Array of Error, required		
		code	machine-readable unique
		String, optional	error code
		mapping	a reference to the parameter
		String, optional	that caused the error
		message	human-readable error
		String, optional	description

1.2.4.6 Change time settings

PUT / system / time

Request

Body	application/json
Properties (SystemTimeSettings)	

now	current time from the perspective of the sensor
Timestamp (string), optional	
timezone	currently configured timezone
String, optional	
ntp servers	one or more network time servers
Array of string, optional	
default_ntp_servers	preconfigured network time servers
Array of string, optional, reaed-only	

Examples

```
{
  "now": "2018-01-24T15:45:15.694004+01:00",
  "timezone": "Europe/Berlin",
  "ntp_servers": [
     "pool.ntp.org"
],
  "default_ntp_servers": [
     "pool.ntp.org"
]
}
```

Response

Code	Body	application/json	
200	Properties (object)	11	
400			
500			
	data	SystemTimeSettings	
	SystemTimeSettings, required		
		now	current time from the per-
		Timestamp (string), optio-	spective of the sensor
		nal	
		timezone	currently configured time-
		String, optional	zone
		ntp_servers	one or more network time
		Array of string, optional	servers
		default_ntp_servers	preconfigured network time
		Array of string, optional, re-	servers
		aed-only	
	errors	Error []	
	Array of Error, required		
		code	machine-readable unique
		String, optional	error code
		mapping	a reference to the parame-
		String, optional	ter that caused the error
		message	human-readable error
		String, optional	description
	May return the following error codes		
	LPLC.system.action_failed		
	LPLC.validation.readonly		
	LPLC.validation.readonly		

1.2.4.7 Retrieve supported Timezones.

GET / system / time / zones

The device contains knowledge about an exhaustive list of officially standardized timezones. The sensor should be configured either with a local timezone or with UTC.

Response

Code	Body	application/json	
200	Properties (object)		
	data	Data	
	Object, required		
		Timezone_names	List of timezones supported by the device

	Array of string, required	
errors	Error []	
Array of Error, required		
	code	machine-readable unique error code
	String, optional	
	mapping	a reference to the parameter that caused the error
	String, optional	
	message	human-readable error description
	String, optional	·

1.2.5 Network

The services of the sensor device are accessible via network connections. The network interfaces of the device can be configured for all standard compliant network setups.

1.2.5.1 Reset Network Settings

DELETE / network

Reset networking settings to factory defaults

Response

Code	
204	The empty response indicates success.

1.2.5.2 Retrieve status and configuration of all network interfaces

GET / network / interfaces

Returns a list of network interfaces.

Each network interface has a MAC-Address (hardware_address), a unique interface name (iface) and indicates the current physical connection status (has link).

Both IPv4 (ipv4) and IPv6 (ipv6) are supported, both with their current configuration (address configurations) and actual interface addresses (current addresses).

Interface addresses are in CIDR notation starting with the interface address, followed by slash and ending with a decimal number representing the subnet mask (IPv4) or prefix length (IPv6).

The collection itself is read-only as no new interfaces can be added. You can change the individual interface configuration with PUT requests to a specific interface resource.

Response

Code	Body	application/json			
200	Properties (object)				
	data Array of NetworkInter- faceAddressConfigu- rationState, required	NetworkInter- faceAddressConfigurati- onState[]			
		ipv4 NetworkInter- faceAddressFa- milyStateIPv4, optional	IPv4 Network address configura- tion		
			ipv4		
			address_configurations Array of any of Net-workAddress Configuration- IPv4Static or Net-workAddressConfigurationIPv4DHCP, optional	NetworkAddressConfigurationIPv4Static[]	
				method	Config- uration

Setting one of Sett	Code	Body	application/json			
address Addr	Joue	2049	apphounon/jour			
Address Address Pleva Pleva Address Pleva						
Pwd Network					quired	
NetworkAddressConfigurations					addraga	
Saleway NetworkAddressIPv4 (string), required address NetworkAddressIPv4 (string), optional NetworkAddressIPv4 (string), one of static, others, others string, one of static, others, others string, one of static, others, others string, one of static, others, othe						
gateway NetworkAddressIPv4 (string), optional gateway (string), optional gateway (string), optional gateway (string), optional gateway (string), one of static, dhop], required gateway for out-going method string, one of static, dhop], required gateway						work
Gateway Getault gate Getault g						-
Sateway NetworkAddressIPV4 (string), optional Sateway NetworkAddressIPV4 (string), optional Sateway NetworkAddressConfiguration Taraffic Sateway Sat						
Ipv6						
NetworkAddressPvd (string), optional gate way for out-gigging traffic					gateway	
NetworkAddressConfiguration Pv6Pt method Configuration Pv6Static Pv6 Network NetworkAddressConfiguration Pv6Static NetworkAddressConfiguration Pv6Static NetworkAddressConfiguration Pv6Static NetworkAddressConfiguration Pv6Static NetworkAddressPv6 NetworkAddressConfiguration Pv6DHCP NetworkAddressConfiguration Pv6DH					NetworkAddressIPv4	gate-
NetworkAddressConfiguration					(string), optional	
NetworkAddressConfiguration NetworkAddressConfiguration NetworkInter-faceAddressFare Network Net						going
Ipv6					National Address Occupie	traffic
Ipv6 Ipv6 IPv6 Network ating, one of [static, dhcp], required Ipv6 Ip						
Ipv6 NetworkInter- faceAddressFa- milyStateIPv6, optional ipv6 address configurations Ipv6 Array of any of NetworkAddressConfigurations Ipv6 AddressConfigurations Ipv6 AddressConfigurations Ipv6 AddressConfigurations Ipv6 AddressConfigurations Ipv6 AddressConfigurations Ipv6 Ipv					method	
Ipv6 NetworkInter- address configurations Ipv6 Ip						
Ipv6 NetworkInter-faceAddressFa-milyState[Pv6, optional ipv6 address_configuration ipv6 address_configurations Array of any of NetworkAddressConfigurations Array of any of Network AddressConfiguration[Pv6Static.] Network-AddressConfiguration[Pv6DHCP or Network-AddressConfiguration[Pv6DHCP or Network-AddressConfiguration[Pv6DHCP or Network-AddressConfiguration[Pv6DHCP or Network-AddressConfiguration] Network-AddressConfiguration[Pv6DHCP] NetworkInterfaceAddress_siPv6 (string), required Pv6DHCP or Network-AddressIPv6 (string), optional NetworkAddressIPv6 (string), optional Optional NetworkAddressConfiguration[Pv6DHCP] NetworkAddressConfiguration[Pv6DHCP] NetworkAddressConfiguration[Pv6DHCP] method Configuration[Pv6DHCP] Configuration[Pv6DHCP] Configuration[Pv6DHCP] Configuration[Pv6DHCP] Configuration[Pv6DHCP] method Configuration[Pv6DHCP] Configuration[Pv6						used
Ipv6 NetworkInter-faceAddressFa-milyState Pv6, optional Ipv6 Address configuration Ipv6 Address configuration Ipv6 Address configuration Ipv6 Address configuration Pv6Static Ipv6 Address configuration Pv6Static Ipv6 Address configuration Pv6Static Ipv6 Address configuration Pv6DHCP or Network Address configuration Pv6Auto, optional Ipv6 Isvatic, dhcp], required Ipv6						
NetworkAddressConfiguration						-
tion pv6 address_configurations pv8 address_configurations Array of any of NetworkAddressConfigurationIPv8Static, Network N						
milyStateIPv6, optional ipv6 address_configurations Array of any of NetworkAddressConfigurationIPv6Static, Network- AddressConfigurationIPv6Static, Network- AddressConfigurationIPv6DHCD or Network- AddressConfigurationIPv6DHCD or Network- AddressConfigurationIPv6Auto, optional method String, one of (static, dhcpl, required sused for the address- NetworkInterfaceAddress- IPv6 String), required IPv6 networkInterfaceAddress- Inct NetworkAddressIPv6 String), required default Qate NetworkAddressIPv6 String), optional out- going traffic NetworkAddressConfigurationIPv6DHCP[] method ConfigurationIPv6DHCP[] method ConfigurationIPv6DHCP[]						
address_configurations Array of any of NetworkAddressConfigurationIPv6Static[] Array of any of NetworkAddressConfigurationIPv6Static, Network AddressConfigurationIPv6DHCP or Network-AddressConfigurationIPv6AddressConfigurationIPv6AddressConfigurationIPv6AddressConfigurationIPv6AddressConfiguration method string, one of [static, dhcp], required address Address NetworkInterfaceAddressiplv6 (string), required gateway NetworkAddressIPv6 (string), optional gateway NetworkAddressIPv6 (string), optional NetworkAddressConfiguration under the addressiplv6 (string), optional Outpain the addressiplv6 (string) the addressiplv6 (st						
tions Array of any of NetworkAddressConfigurationIPv6Static, Network AddressConfigurationIPv6DHCP or Network AddressConfigurationIPv6Auto, optional method string, one of [static, dhcp], required string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional page way NetworkAddressIPv6 (string), optional string, one of [static, dhcp], required address IPv6 (string), required page way NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] NetworkAddressConfigurationIPv6DHCP[] method Configuration	<u> </u>				NetworkAddressConfigu-	
workAddressConfigurationIPv6Static, Network AddressConfigurationIPv6DHCP or Network AddressConfigurationIPv6Auto, optional method string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCPI NetworkAddressConfigurationIPv6DHCPI method Configuration way for out-going traffic				tions		
Network-AddressConfiguration Pv6DHCP or Network-AddressConfiguration Pv6Auto, optional				workAddressConfig-		
AddressConfigurationIPv6DHCP or Network AddressConfigurationIPv6Auto, optional method string, one of [static, dhcp], required address NetworkInterfaceAddressiPv6 (string), required gateway NetworkAddressIPv6 (string), optional gateway NetworkAddressConfigurationIPv6DHCP[] method Configuration method used dress. IPv6 sdring), required gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method Configuration Configuration Configuration Configuration				uration/Pv6Static,		
Network-AddressConfigura- tionIPv6Auto, op- tional method String, one of startic, dhcp], re- quired string, re- quired string, required		i e		Network-		
AddressConfigurationIPv6Auto, optional method string, one of [static, dhcp], required address address NetworkInterfaceAddressiPv6 (string), required gateway NetworkAddressIPv6 (string), optional patents NetworkAddressConfiguration NetworkAddressConfigurationIPv6DHCP[] nethod Configuration nethod used for the ad- dress. IPv6 nethod gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method Configuration				AddressConfigura-		
method string, one of static, dhcp], required method used for the address lPv6 string), required lPv6 notation mothod dress. lPv6 string), required mothod				AddressConfigura- tionIPv6DHCP or		
method configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura-		
string, one of [static, dhcp], required uration method used for the address. address IPv6 IPv				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-		
Static, dhcp], required method used for the address IPv6 networkInterfaceAddressilPv6 (string), required method used for the address. IPv6 networkInterfaceAddressilPv6 network address in CIDR notation notation default gateway MetworkAddressIPv6 (string), optional method NetworkAddressConfigurationIPv6DHCP[] method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-		
Quired used for the addess IPv6 NetworkInterfaceAddressin CIDR notation				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-		
address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional gate-way for out-going traffic NetworkAddressConfiguration Pv6DHCP[] method ad-dress IPv6 net-work ad-dress in CIDR nota-tion default gate-way for out-going traffic				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of	uration
address IPv6 NetworkInterfaceAddressiPv6 (string), required work address in CIDR notation				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], re-	uration method used
NetworkInterfaceAddressin CIDR notation				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], re-	uration method used for the
SIPv6 (string), required work address in CIDR notation gateway MetworkAddressIPv6 (string), optional work gateway for outgoing traffic NetworkAddressConfigurationIPv6DHCP[] method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required	uration method used for the ad- dress.
address in CIDR notation gateway NetworkAddressIPv6 (string), optional NetworkAddressConfiguration NetworkAddressConfigurationIPv6DHCP[]				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address	uration method used for the ad- dress. IPv6
in CIDR notation gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method in CIDR notation default gate- way for out- going traffic NetworkAddressConfigurationIPv6DHCP[]				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddres-	uration method used for the ad- dress. IPv6 net- work
gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method notation default gate- way for out- going traffic NetworkAddressConfigurationIPv6DHCP[]				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddres-	uration method used for the ad- dress. IPv6 net- work ad-
gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method default gate- way for out- going traffic NetworkAddressConfigurationIPv6DHCP[]				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddres-	uration method used for the ad- dress. IPv6 net- work ad- dress
NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[] method Qateway for outgoing traffic NetworkAddressConfigurationIPv6DHCP[] method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddres-	uration method used for the address. IPv6 network address in CIDR nota-
out- going traffic NetworkAddressConfigu- rationIPv6DHCP[] method Config- uration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddres- sIPv6 (string), required	uration method used for the address. IPv6 network address in CIDR notation
going traffic NetworkAddressConfigurationIPv6DHCP[] method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6	uration method used for the address. IPv6 network address in CIDR notation default gate-
traffic NetworkAddressConfigurationIPv6DHCP[] method traffic NetworkAddressConfigurationIPv6DHCP[] uration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6	uration method used for the address. IPv6 network address in CIDR notation default gateway for
rationIPv6DHCP[] method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6	uration method used for the address. IPv6 network address in CIDR notation default gateway for outgoing
method Configuration				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional	uration method used for the address. IPv6 network address in CIDR notation default gateway for outgoing
				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional	uration method used for the address. IPv6 network address in CIDR notation default gateway for outgoing
method				AddressConfigura- tionIPv6DHCP or Network- AddressConfigura- tionIPv6Auto, op-	string, one of [static, dhcp], required address NetworkInterfaceAddressIPv6 (string), required gateway NetworkAddressIPv6 (string), optional NetworkAddressConfigurationIPv6DHCP[]	uration method used for the address. IPv6 network address in CIDR notation default gateway for outgoing traffic

Code	Body	application/json			
				string, one of [static, dhcp, auto], required	used for the ad- dress.
				NetworkAddressConfigu- rationIPv6Auto[]	
				method string, one of [static, dhcp, auto], required	Configuration method used for the address.
	errors Array of Error, required	Error []			
		code String, optional	machine-readable unique error code		
		mapping String, optional	a reference to the parameter that caused the error		
		message String, optional	human-readable er- ror description		

1.2.5.3 Retrieve status and configuration of a single network interface

GET / network / interfaces /{name}

Returns information, current status, address configuration, and current addresses for a single interface.

Request

Path Variables	
name	
String, required	

Response

Code	Body	application/json			
200	Properties (object)				
	data Array of NetworkInterfaceAddressConfigurationState, required	NetworkInter- faceAddressConfi- gurationState[]			
	4	ipv4 NetworkInter- faceAddressFa- milyStateIPv4, opti- onal	IPv4 Network address configuration		
			ipv4		
			address_configurations Array of any of Network-AddressConfiguration-IPv4Static or Network-AddressConfiguration-IPv4DHCP, optional	NetworkAddressConfigurationIPv4Static[]	
				method string, one of [static, dhcp], re- quired	Configuration method used for the address.
				address NetworkInter- faceAddressIPv4 (string), required	IPv4 network address in CIDR notation

	mapping String, optional message	a reference to the parameter that caused the error human-readable error		
errors Array of Error, re- quired	code String, optional	machine-readable unique error code		
			gurationIPv6Auto[] method string, one of [static, dhcp, au to], required	Configuration method used for the ad- dress.
			[static, dhcp, au to], required NetworkAddressConfi-	for the ad- dress.
			gurationIPv6DHCP[] method string, one of	Configuration method used
			gateway NetworkAddressIPv6 (string), optional NetworkAddressConfi-	default gate- way for out- going traffic
			address NetworkInter- faceAddressIPv6 (string), required	IPv6 network address in CIDR notation
			method string, one of [static, dhcp], re- quired	Configuration method used for the address.
		Array of any of Network-AddressConfiguration-IPv6Static, Network-AddressConfiguration-IPv6DHCP or Network-AddressConfiguration-IPv6Auto, optional	gurationIPv6Static[]	
	onal	ipv6 address_configura-	NetworkAddressConfi-	
	NetworkInter- faceAddressFa- milyStateIPv6, opti-	IPv6 Network address configuration		
			method string, one of [static, dhcp], re- quired	Configuration method used for the ad- dress.
			NetworkAddressConfi- gurationIPv4DHCP[]	
			gateway NetworkAddressIPv4 (string), optional	default gate- way for out- going traffic

1.2.5.4 Modify IPv4 and/or IPv6 address configuration

PUT / network / interfaces / {name}

Interfaces are readonly except for their IPv6 and IPv4 address configurations.

You can set new address configurations by providing the <code>address_configurations</code> key in the respective IP address family object (<code>ipv4</code> or <code>ipv6</code>). <code>address_configurations</code> will replace any existing configuration with the new configuration. If you only want to add a new configuration be sure to submit any existing ones as well.

The address_configurations list only affects their respective address family. If you only want to alter the IPv4 address configuration it is sufficient to set the <code>ipv4.address_configurations</code> key. Any other address family will remain unaffected as long as you do not alter their own <code>address_configuration</code> list as well.

The response to a network configuration change request is returned *before* the new configuration is applied. This ensures that the caller receives an acknowledgment from the API before the network connection may get lost due to the changed configuration. The reconfiguration of the new network setup happens in the background shortly after the response is emitted. Thus the API may close existing connection and will not respond to further requests for a few seconds. Please note that only one configuration change may be requested at a time. Thus the API will delay the response to a second request until all internal processes for the first request are finished. Such a response to a quick subsequent request may be delayed by up to 20 seconds. If too many parallel requests are competing for network setup changes, the HTTP status response 423 (*Locked*) will be returned after a timeout of 20 seconds.

Please Note: Even though the API supports multiple address configurations for each address family only the first will be applied at the moment. This is a pending feature. Therefore you should only provide one address configuration item per address family.

Request

Path Variables	
name	
String, required	

Body	application/json		
Properties (NetworkInter-			
faceInformation)			
iface	unique name describing a network		
<u>NetworkInterfaceName</u>	interface		
(string), pattern: ^[a-z0-			
9-]+\$, required, read-			
only			
hardware_address	unique hardware address of a net-		
MacAddress (string), pat-	work interface		
tern: ^([a-f0-			
9]{2}:){5}[a-f0-			
9] {2}\$, required , read- only			
has link	current physical connection status		
boolean, required , read-	(whether a cable is plugged in or		
only	not)		
ipv4	IPv4 Network address configura-		
NetworkInterfaceAddress-	tion		
FamilyStateIPv4, optional	tion		
1 armiyotaten v-, optionar	ipv4		
	address configurations	NetworkAddressConfiguratio-	
	Array of any of Network-	nlPv4Static[]	
	AddressConfigurationIPv4Static or	in violation	
	NetworkAddressConfiguration-		
	IPv4DHCP, optional		
	, ,	method	Configuration
		string, one of [static, dhcp], re-	method used for
		quired	the address.
		address	IPv4 network ad-
		NetworkInterfaceAddressIPv4	dress in CIDR
		(string), required	notation
		gateway	default gateway
		NetworkAddressIPv4 (string), optio-	for outgoing traf-
		nal	fic

Body	application/json		
		NetworkAddressConfiguratio- nlPv4DHCP[]	
		method string, one of [static, dhcp], re- quired	Configuration method used for the address.
ipv6 NetworkInterfaceAddress- FamilyStateIPv6, optional	IPv6 Network address configuration		
	ipv6		
	address_configurations Array of any of Network- AddressConfigurationIPv6Static, NetworkAddressConfiguration- IPv6DHCP or Network- AddressConfigurationIPv6Auto, optional	NetworkAddressConfiguratio- nIPv6Static[]	
		method string, one of [static, dhcp], re- quired	Configuration method used for the address.
		address NetworkInterfaceAddressIPv6 (string), required	IPv6 network ad- dress in CIDR notation
		gateway NetworkAddressIPv6 (string), optio- nal	default gateway for outgoing traf- fic
		NetworkAddressConfiguratio- nlPv6DHCP[]	
		method string, one of [static, dhcp, auto], required	Configuration method used for the address.
		NetworkAddressConfiguratio- nlPv6Auto[]	
		method string, one of [static, dhcp, auto], required	Configuration method used for the address.

Examples

Remove all IPv6 address configurations

```
{
  "ipv6": {
     "address_configurations": []
  }
}
```

Replace existing IPv4 configuration with DHCP

Set static and dynamic IPv4 configuration

{

Response

Respon Code	Body	application/json			
200	Properties (object)	',			
400					
404					
423					
	data	NetworkInter-			
	Array of NetworkIn-	faceAddressConfigurati-			
	terfaceAddressCon-	onState[]			
	figurationState, re-				
	quired				
		ipv4	IPv4 Network		
		NetworkInter-	address configuration		
		faceAddressFa-			
		milyStateIPv4, optional			
			ipv4		
			address_configura-	NetworkAddressCon-	
			tions	figurationIPv4Static[]	
			Array of any of Net-		
			workAddressConfigu-		
			rationIPv4Static or		
			NetworkAddressCon-		
			figurationIPv4DHCP, optional		
			οριισται	method	Configura-
				string, one of	tion method
				[static, dhcp], re-	used for the
				quired	address.
				address	IPv4 network
				NetworkInter-	address in
				faceAddressIPv4	CIDR nota-
				(string), required	tion
				gateway	default gate-
				NetworkAddressIPv4	way for out-
				(string), optiona	going traffic
				NetworkAddressCon-	
				figurationIPv4DHCP[]	
				method	Configura-
				string, one of	tion method
				[static, dhcp], re-	used for the
				quired	address.
		ipv6	IPv6 Network		
		NetworkInter-	address configuration		
		faceAddressFa-			
		milyStateIPv6, optional			
			ipv6	A1	
			address_configura-	NetworkAddressCon-	
			tions	figurationIPv6Static[]	
			Array of any of Net-		
			workAddressConfigu- rationIPv6Static, Net-		
			workAddressConfigu-		
			rationIPv6DHCP or		
			NetworkAddressCon-		
			figurationIPv6Auto,		
			optional		
	L		οριισται	<u> </u>	l

			method	Configura-
			string, one of	tion method
			[static, dhcp], re-	used for the address.
			quired	
			address	IPv6 network
			NetworkInter-	address in
			faceAddressIPv6	CIDR nota-
			(string), required	tion
			gateway	default gate-
			NetworkAddressIPv6	way for out-
			(string), optional NetworkAddressCon-	going traffic
			figurationIPv6DHCP[]	0 "
			method	Configura- tion method
			string, one of	used for the
			[static, dhcp, au	address.
			to], required	address.
			NetworkAddressCon-	
			figurationIPv6Auto[]	0 "
			method	Configura-
			string, one of	tion method
			[static, dhcp, au	used for the
			to], required	address.
errors	Error []			
Array of Error, re-				
quired		and the second of the second		
	code	machine-readable		
	String, optional	unique error code		
	mapping	a reference to the pa-		
	String, optional	rameter that caused		
		the error		
	message	human-readable error		
	String, optional	description		
	May return the following	error codes		
	LPLC.validation LPLC.not found.coll	action item		
l	Trrc.not_tound.coll	ection.item		

1.2.6 Peripherals

1.2.6.1 Get Keypad Information

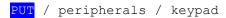
GET / peripherals / keypad

Response

Code	Body	application/json	
200	Properties (Keypad-		
	Information)		
	data	Describe the current state of	
	KeypadInformation,	the keypad as well as access	
	required	to visualization data.	
		KeypadInformation	
		locked	Boolean flag indicating the state of the key lock (true
		boolean, required	-> locked, false -> unlocked).
			All keypad inputs are ignored while the lock is ac-
			tive.
		clear_matcher_before_teach	The boolean flag controls whether multiple detecta-
		boolean, required	bles can be stored for a matcher via keypad-based
			teach operations. A value of true implies, that a
			teach operation always removes all existing detecta-
			bles from the currently selected matcher before
			adding the new detectable. With a value of false pre-
			viously existing detectables are not deleted before a
			new one is added.
		visualization_url	The visualization resource location can be used for
		any of string or null, optional,	providing a virtual keypad interface.
		read-only	Its URL may start with a scheme (e.g. http or https)
			for a full URL including hostname or it may start with

		a slash, indicating a path provided by the device it- self.
		This attribute cannot be modified.
errors Array of Error, required	Error []	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description

1.2.6.2 Modify Keypad



Modify basic states of the keypad.

Request

Body	application/json	
Properties (KeypadInformation)		
	locked	Boolean flag indicating the state of the key lock (true -> locked,
	boolean, required	false -> unlocked).
		All keypad inputs are ignored while the lock is active.
	clear_matcher_before_teach	The boolean flag controls whether multiple detectables can be
	boolean, required	stored for a matcher via keypad-based teach operations. A value of true implies, that a teach operation always removes all existing detectables from the currently selected matcher before adding the new detectable. With a value of false previously existing detectables are not deleted before a new one is added.
	visualization_url	The visualization resource location can be used for providing a vir-
	any of string or null, optional, read-	tual keypad interface.
	only	Its URL may start with a scheme (e.g. http or https) for a full URL in-
		cluding hostname or it may start with a slash, indicating a path pro-
		vided by the device itself.
		This attribute cannot be modified.

Examples

```
{
  "locked": true,
  "clear_matcher_before_teach": false,
  "visualization_url": "/media/keypad-image.svg"
}
```

Response

Code	Body	application/json	
200 400	Properties (KeypadInformation)		
	data KeypadInformation, required	Describe the current state of the keypad as well as access to visualization data. KeypadInformation	
		locked boolean, required	Boolean flag indicating the state of the key lock (true -> locked, false -> unlocked). All keypad inputs are ignored while the lock is active.
		clear_matcher_before_teach boolean, required	The boolean flag controls whether multiple detectables can be stored for a matcher via keypad-based teach operations. A value of true implies, that a teach operation always removes all existing detectables from the currently selected matcher before adding the new detectable.

		With a value of false previously ex-
		isting detectables are not deleted before a new one is added.
	visualization_url any of string or null, optional, read- only	The visualization resource location can be used for providing a virtual keypad interface. Its URL may start with a scheme (e.g. http or https) for a full URL including hostname or it may start with a slash, indicating a path provided by the device itself. This attribute cannot be modified.
errors Array of Error, required	Error []	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description
May return the following of LPLC. validation	error codes	

1.2.6.3 Retrieve user interactions on the keypad

GET / peripherals / keypad / events

Return a list of keypad events.

This collection is implemented as a ring-buffer meaning that older Events will be removed once new events are added.

Response

Code	Body	application/json	
	Properties (object)		
	events	KeypadEvent[]	
	Array of KeypadEvent,		
	required		
		source	The usual source of events is inputs.
		string, required	
		name	Name of a keypad input (button) that may
		KeypadEventInput (string), required	trigger events.
		event	Input peripherals can trigger different
		KeypadEventName (string), required	events.
		timestamp	The timestamp is given in milliseconds and
		integer, minimum: 0, required	should be monotonic increasing.

1.2.6.4 Retrieve a list of available input elements on the keypad

GET / peripherals / keypad / inputs

Returns a list of keypad input elements.

Every keypad input element represents a physical button on the keypad.

Response

Code	Body	application/json			
200	Properties (object)				
	data object, required	data			
		inputs Array of KeypadInput- Button, required	KeypadInputButton[]		
			name KeypadEventInput (string), required	Name of a keypad input (button) that may trigger events.	

		capabilities Array of object, required	object[]	
			name KeypadEventName (string), required	Input peripherals can trigger different events.
			url string, required	The event can be triggered externally by submitting a POST request against this resource.
	errors Array of Error, required	Error []		
		code String, optional	machine-readable unique error code	
		mapping String, optional	a reference to the pa- rameter that caused the error	
		message String, optional	human-readable error description	

1.2.6.5 Simulate a user interaction on the keypad

POST / peripherals / keypad / inputs / {name} / {event}

Simulates a button-press by externally triggering the given event for the input.

See the collection of keypad inputs for a list or URLs available for triggering events.

Request

Path Variables	
name	
String, required	
event	
String, required	

Response

Code		
204	The empty response indicates success	
404		
	May return the following error codes	
	LPLC.resource.unspecified	
	LPLC.resource.invalid	
	LPLC.illegal_request	

1.2.6.6 Get Output Configuration

GET / peripherals / outputs

Response

Code	Body	application/json	
200	Properties (object)		
	data SwitchingOutputs, required	Eletrical output lines can drive external actors in different electrical modes.	
		SwitchingOutputs	
		output_driver SwitchingOutputDriver (string), required	The Output Driver defines the electrical behaviour of the switching outputs. The supported output drivers can be retrieved via /api/sensor/capabilities.
		count	Number of available output lines
		integer, required	

errors	Error []	
Array of Error, required		
	code	machine-readable unique error
	String, optional	code
	mapping	a reference to the parameter that
	String, optional	caused the error
	message	human-readable error descrip-
	String, optional	tion

1.2.6.7 Modify Output Configuration

PUT / peripherals / outputs

Request

Body	application/json
Properties (SwitchingOutputsWritable)	
output_driver	The Output Driver defines the electrical behaviour of the
SwitchingOutputDriver (string), required	switching outputs. The supported output drivers can be re-
	trieved via /api/sensor/capabilities.

Response

Code	Body	application/json	
200 400	Properties (object)		
	data SwitchingOutputs, required	Eletrical output lines can drive external actors in different electrical modes.	
		SwitchingOutputs	
		output_driver SwitchingOutputDriver (string), required	The Output Driver defines the electrical behaviour of the switching outputs. The supported output drivers can be retrieved via /api/sensor/capabilities.
		count integer, required	Number of available output lines
	errors Array of Error, required	Error []	
		code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error descrip- tion
	May return the following error LPLC.validation	codes	

1.2.6.8 Get current interface configuration

GET / peripherals / rs232

Response

Code	Body	application/json	
200 400	Properties (object)		
	data InterfaceRS232, required	InterfaceRS232	
		protocol any of <u>SerialModbusSettings</u> or SerialElizaSettings, required	SerialModbusSettings
			type string, one of [none, eliza, modbus], default: eliza, required
			slave_id any of number or null, required
			frame_format

		<pre>string, one of [rtu, ascii], default: rtu, required</pre>
		SerialElizaSettings
		type string, one of [none, eliza, modbus], default: eliza, required
	baud_rate number, one of [9600, 19200, 115200], re- quired	
errors Array of Error, required	Error []	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description

1.2.6.9 Modify interface configuration

PUT / peripherals / rs232

Request

Body	application/json	
Properties (InterfaceRS232)		
protocol	InterfaceRS232	
any of <u>SerialModbusSettings</u> or		
SerialElizaSettings, required		
	protocol	SerialModbusSettings
	any of SerialModbusSettings or Seri-	
	alElizaSettings, required	
		type
		string, one of [none, eliza, modbus],
		default: eliza, required
		slave_id
		any of number or null, required
		frame_format
		string, one of [rtu, ascii], default: rtu,
		required
		SerialElizaSettings
		type
		string, one of [none, eliza, modbus],
		default: eliza, required
	baud rate	
	number, one of	
	[9600, 19200, 115200], required	

Request

Code	Body	application/json	
200 400	Properties (object)		
	data InterfaceRS232, required	InterfaceRS232	
		protocol any of <u>SerialModbusSettings</u> or <u>SerialE-lizaSettings</u> , required	SerialModbusSettings
			type string, one of [none, eliza, mod- bus], default: eliza, required
			slave_id any of number or null, required
			<pre>frame_format string, one of [rtu, ascii], default: rtu, required</pre>
			SerialElizaSettings
			type

		<pre>string, one of [none, eliza, mod- bus], default: eliza, required</pre>
	baud_rate number, one of [9600, 19200, 115200], required	
errors Array of Error, required	Error []	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description
May return the following LPLC.validation	error codes	

1.2.6.10 Get trigger source event statistics

GET / peripherals / trigger-sources

Response

Code	Body	application/json		
200	Properties (object)			
	data TriggerSourcesStatus, required	The sensor has a number of input lines that can be used as trigger sources. The event counters are updated periodically (approximately every second).		
		TriggerSourcesStatus		
		trigger_sources Array of object, required	object[]	
			name string, required	
			event_counters Object, required	event_counters
				edge_falling Number, required
				edge_rising Number, required
				level_low Number, required
		errors Array of Error, required	Error []	,
			code String, optional	machine-readable unique error code
			mapping String, optional	a reference to the pa- rameter that caused the error
_			message String, optional	human-readable error description

1.2.6.11 Get current interface configuration

GET / peripherals / usb

Response

Code	Body	application/json	
200	Properties (object)		
	data InterfaceUSB, required	InterfaceUSB	
		protocol Any of SerialModbusSettings or SerialElizaSettings, required	SerialModbusSettings
			<pre>type string, one of [none, eliza, modbus], de- fault: eliza, required</pre>
			slave_id

		any of number or null, required
		frame_format
		string, one of [rtu, ascii], default: rtu,
		required
		SerialElizaSettings
		type
		string, one of [none, eliza, modbus], de-
		fault: eliza, required
errors	Error []	
Array of Error, required		
	code	machine-readable unique error code
	String, optional	
	mapping	a reference to the parameter that caused the
	String, optional	error
	message	human-readable error description
	String, optional	

1.2.6.12 Modify interface configuration

PUT / peripherals / usb

Request

Body application/json	
Properties (InterfaceUSB)	
protocol any of <u>SerialModbusSettings</u> or <u>SerialElizaSettings</u> , required	SerialModbusSettings
	type
	string, one of [none, eliza, modbus], default: eliza, required
	slave_id
	any of number or null, required
	frame_format
	string, one of [rtu, ascii], default: rtu, required
	SerialElizaSettings
	type string, one of [none, eliza, modbus], default: eliza, required

Response

Code	Body	application/json	
200 400	Properties (object)		
	data InterfaceUSB, required	InterfaceUSB	
		protocol Any of <u>SerialModbusSettings</u> or <u>SerialElizaSettings</u> , required	SerialModbusSettings
			type string, one of [none, eliza, modbus], default: eliza, required
			slave_id any of number or null, required
			<pre>frame_format string, one of [rtu, ascii], default: rtu, required</pre>
			SerialElizaSettings
			<pre>type string, one of [none, eliza, modbus], default: eliza, required</pre>
	errors Array of Error, required	Error []	
		code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description
	May return the following LPLC.validation	g error codes	

1.2.7 Actions

The sensor can be programmed to react on specific external or internal events. The available actions can be either triggered via trigger input lines or via API requests. This allows customized behaviour, e.g. teaching colors via an external button.

This endpoint provides details for all available actions. Additionally actions can be executed for one-time operations.

See action-triggers if you want to connect trigger input line events with specific actions for repeated operations.

1.2.7.1 Action descriptions

Every action accepts a a distinct set of optional or mandatory arguments. They are summarized in the argument field of each action in the collection. The detailed description of their meaning and the specification of each Action's behaviour is described below.

1.2.7.2 Action "enable_switching_output"

The behavior of the switching outputs changes significantly while this action is configured for at least one *trigger event*. See the introduction chapter about *Switching outputs, triggers and hold time settings* for a detailed specification of the different behaviors.

Arguments: none

1.2.7.3 Action "teach single"

Sample a new detectable whenever the action is executed.

Arguments:

- matcher_id (optional, default: null): the UUID of a matcher or null. In case of null the
 new detectable is assigned to a matcher based on matcher_output_pattern. In case of
 the ID belonging to an existing matcher, the new detectable is assigned to this matcher. In
 case of an ID that does not belong to an existing matcher, a matcher with this ID is created
 and assigned to the new detectable.
- matcher_output_pattern (optional, default: null): output pattern of the target matcher. This field is only considered, if matcher_id is null. In case of matcher_output_pattern being null, a new matcher is created whenever the action is executed. Otherwise the matcher_output_pattern is expected to be a dictionary containing a states field. This states field is supposed to contain a list of boolean output states. If a matcher with this output pattern already exists, then the new detectable is added to this matcher. If no matcher with such an output pattern exists, then a new matcher for this output pattern is created before adding the new detectable to it.
- remove_matcher_detectables_before (optional, default: true): the boolean value specifies whether detectables belonging to the selected matcher should be removed right before the new detectable is added.

1.2.7.4 Action "keylock"

Control the state of the keypad locking.

A typical approach could be to connect the rising edge of a trigger input to this action with the parameter locked being true and the falling edge of the same trigger input with false. Thus the locking state of the keypad would follow the level of the trigger input.

Arguments:

• locked (required): the boolean value specifies the wanted target state of the keypad locking. The true value locks the keypad. The false value releases the lock.

1.2.7.5 Action "run_autogain"

Start an autogain procedure.

Probably the action "remove_all_detectables" should be executed afterwards, since the color values may not be accurate anymore due to changed sampling settings.

Arguments: none

1.2.7.6 Action "remove all detectables"

Clear the detectables collection.

Arguments: none

1.2.7.7 Action "remove_all_matchers"

Clear the matchers collection. This also removes all detectables.

Arguments: none

1.2.7.8 Retrieve Actions

GET / actions

Retrieves a list of available Actions

Response

Code	Body	application/json		
200	Properties (object)			
	data	data		
	object, required			
		actions	Action []	
		Array of Action, required		
			name	Unique name of the
			string, required	action
			arguments	arguments
			object, required	
	errors	Error []		
	Array of Error, required			
		code	machine-readable unique	
		String, optional	error code	
		mapping	a reference to the parame-	
		String, optional	ter that caused the error	
		message	human-readable error	
		String, optional	description	

1.2.7.9 Get Action

GET / actions / {itemId}

Returns a single Action.

Request

Path Variables	
itemId	
String, required	

Response

Code	Body	application/json	
200	Properties (object)		
	data	The sensor allows the connection of events with ac-	
	Action, required	tions. Actions can be related to the sensor operations	
		or the information handled by the sensor (e.g. the list	
		of stored detectables).	
		The Action consists of a unique name and a set of op-	
		tional arguments.	
		The list of available Actions and their possible argu-	
		ments can be retrieved via /api/actions.	
		Action	
		name	Unique name of the action
		string, required	
		arguments	Arguments
		object, required	
	errors	Error []	
	Array of Error, required		
		code	machine-readable unique
		String, optional	error code
		mapping	a reference to the parameter
		String, optional	that caused the error
		message	human-readable error
		String, optional	description
		May return the following error codes	
		LPLC.not_found.collection.item	

1.2.7.10 Software-triggered Actions

POST / actions / {itemId} / execute

Execute the given action once. This provides access to all operations that can be connected to trigger input line events.

Additional parameters (if required by the action) can be provided in the body of the request. For example the *keylock* action could be executed by providing a dictionary containing the key *locked* with the wanted boolean target state. See the *arguments* field of each action in the collection above.

The content of the response depends on the specific action that was executed.

The example actions with their respective parameters and responses may not be supported by all sensors. The Actions collection contains the authoritative list of supported actions for each sensor.

Request

Path Variables		
itemId		
String, required		

Body	application/json		
Properties (AnyAc-			
tion (union))			
ActionEnableSwit-	Apply the <i>output_pattern</i> of the		
chingOutput	currently detected matcher to		
ActionEnableSwit-	the switching outputs of the		
chingOutput, optio-	sensor.		
nal			
	ActionEnableSwitchingOut-		
	put		
	name	Unique name of	
	string, required	the action	
	arguments	arguments	

Body	application/json			
	object, required			
ActionTeachDe-	Add the currently sampled			
tectable ActionTeachDe-	color as a Detectable to the selected matcher.			
tectable, optional	Scieded materier.			
/	ActionTeachDetectable			
	arguments	arguments		
	object, required	matches id	The new Detectable is	
		matcher_id <u>UUID</u> (string), pat-	The new Detectable is assigned to the	
		tern: ^[a-f0-9-	Matcher identified by	
]+\$, optional,	this UUID. In case this	
		read-only	matcher UUID (and "matcher_output_pat-	
			tern") is undefined, a	
			new matcher is cre-	
			ated.	
		matcher_out- put_pattern:	Pattern of the switch- ing outputs to be used	
		object, optional	when selecting the tar-	
			get matcher for the	
			new detectable. A suit-	
			able matcher is cre- ated, if no matcher	
			with the specified pat-	
			tern is found. This field	
			is ignored, if	
			"matcher_id" is not null. If no pattern is	
			defined (an no	
			"matcher_id"), then a	
			new matcher is cre- ated whenever the	
			corresponding action	
			is executed.	
			matcher_output_pat-	
			tern:	List of True/False
			Array of any of bool-	values describing
			ean or null, required	the wanted states
				of the Switching
			remove_matcher_de-	Outputs Remove all De-
			tectables_before	tectables belong-
			boolean , default:	ing to the config-
			true, optional	ured Matcher be-
				fore attaching the new Detectable.
	name	Unique name of		. 10 W Detectable.
	string, required	the action		
ActionKeyLock	Change the <i>locked</i> state of the			
ActionKeyLock, optional	keypad. This allows or disallows local access to the sen-			
<u> </u>	sor via the keypad.			
	ActionKeyLock			
	arguments object, required	arguments		
	object, required	locked	Target state of the	
		boolean, required	keypad locking.	
	name	Unique name of		
	string, required	the action		
ActionRunAutogain	Start an automatic adjustment			
ActionRunAutogain,	of the optiocal sensor setup.			
optional	See /api/sensor/detec-			
	tion-profiles/cur- rent/autogain for de-			
	tails.			
	ActionRunAutogain			
	name	Unique name of		
	string, required arguments	the action arguments		
	u gumento	argamento	1	1

Body	application/json		
	object, required		
ActionRemoveAll-	Remove all stored Detectables		
Detectables	belonging to any Matcher.		
ActionRemoveAllDe-			
tectables, optional			
	ActionRemoveAllDetectables		
	name	Unique name of	
	string, required	the action	
	arguments	arguments	
	object, required		
ActionRemoveAll-	Remove all stored Matchers		
Matchers	(including the related detecta-		
ActionRemoveAll-	bles).		
Matchers, optional			
	ActionRemoveAllMatchers		
	name	Unique name of	
	string, required	the action	
	arguments	arguments	
	object, required		

Response

Code	Body	application/json			
200	Properties (object)				
204	, , ,				
	data	ActionResultEnableSwit-			
	any of ActionResultEna-	chingOutput			
	bleSwitchingOutput, Ac-				
	tionResultTeachDetecta-				
	ble, ActionResultKey-				
	Lock, ActionRe- sultRunAutogain, Action-				
	ResultRemoveAllDetecta-				
	bles (string) or ActionRe-				
	sultRemoveAllMatchers				
	(string), required				
	· · · · · · · · · · · · · · · · · · ·	uuid	unique identifier (UUID)		
		<u>UUID</u> (string), pattern:	as defined by RFC		
		^[a-f0-9-]+\$, re-	4122, ITU-T Rec. X.667,		
		quired, read-only	and <u>ISO/IEC 9834-8</u>		
		timestamp	The timestamp (given		
		TimestampBackendUp-	in microseconds) is		
		time (number), required	based on the uptime of		
			the internal analog sen-		
			sor backend. It may get reset to zero under spe-		
			cific conditions.		
		corrected color	Representation of a		
		CorrectedColor, required	color in the colorspace		
			XYZ.		
			values	Location in a	
			Array of number, mini-	colorspace	
			mum items: 3, maxi-		
			mum items: 3, required		
		transformed_color	A color represented by		
		TransformedColor, requi-	a coordinate in the col-		
		red	orspace. The array indices of the values		
			property match the or-		
			der of the color-		
			space.axes property		
			of currently used detec-		
			tion profile.		
			values	Location in a	
			Array of number, mini-	colorspace	
			mum items: 3, maxi-		
			mum items: 3, required		
		representations	Pre-calculcated visual		
		ColorRepresentations, re-	representations of a		
		quired	color suitable for ren-		
		1	dering		

Code	Body	application/json		
	200,		RGB Array of number, minimum items: 3, maximum items: 3, required	RGB color array representing the axes r, g, and b in that order. Values are floats between 0 and 1.
		inputs InputsState, required	The state of all inputs during a given period is specified by a list of possible events combined with a boolean value indicating, if the given event occurred within the period.	
			// boolean, required	The boolean value indicates whether the named input event occurred during the last period.
		detection ColorMatchingResult, required	After each sampling period the retrieved color value is compared to the stored detectables (color positions). Detectables are ignored, if the tolerance shape of their corresponding matcher does not encompass the current sample. Finally the closes suitable detectable is selected as the winner of the color matching operation. The corresponding matcher determines the state of the sensor for the duration of the next sampling period.	
			detection matcher any of <u>UUID</u> (string) or	Deprecated: use "cho-
			null, optional, Deprecated chosen_matcher_id any of UUID (string) or	sen_matcher_id" instead unique identifier of the selected
			null, required distances Array of any of number or null, required	matcher Distance be- tween the sam- ple's color posi- tion and the se- lected matcher's closest color position along the three axes of the color space. The array con- tains three 'null' values, if no suitable matcher was found for the current color sample. Currently active
			output_pattern CurrentSwitchingOut- putsState, required	state of the Switching Out- puts. Beware

Codo	Pody	application/icon	1		
Code	Body	application/json		that this may deviate from the specified output states of the current best matcher, since settings like triggered input or hold time influence update process for the Switching Outputs. states Array of any of boolean or null, required	List of True/False values de- scribing the wanted
					states of the Switching Outputs
		signal_level number, required	The signal level indicates the usage of the internal ADC sampling range. This		,
		ActionResultTeachDe- tectable			
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re- quired, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8		
		alias Alias (integer), required, read-only	A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.		
		matcher_id UUID (string), pattern: ^[a-f0-9-]+\$, re- quired, read-only	reference to the Matcher containing this Detectable		
		color <u>TransformedColor</u> , required	A color represented by a coordinate in the col- orspace. The array indi- ces of the values property match the or- der of the color- space.axes property of currently used detec- tion profile.		
			color values	Location in a	
			Array of number, minimum items: 3, maximum items: 3, required	colorspace	
		representations ColorRepresentations, optional, read-only	Pre-calculcated visual representations of a color suitable for ren- dering		
		ActionResultKeyLock locked boolean, required ActionResultRunAutogain	New state of the key- pad locking.		
		level	Target value for the auto-gain procedure		

Code	Body	application/json		
		Number, default: 0.8,		
		minimum: 0.01, maxi-		
		mum: 1, optional		
		minimum_sample_rate	Desired sample rate	
		SampleRate (number),	(the default is the cur-	
		minimum: 0.02, optional	rent sample rate)	
		enable_internal_emitter	controls the power of	
		Boolean, default: true,	the internal light source	
		optional		
		enable_ambi-	Control the ambient	
		ent_light_compensation	light compensation pro-	
		Boolean, default: true,	cedure. This setting is	
		optional	only relevant if ena-	
			ble_inter-	
			nal_emitter is set to true. The ambient light	
			compensation leads to	
			a pulsed usage of the	
			internal light emitter.	
			Samples are collected	
			for alternating light and	
			dark phases. This al-	
			lows to calculate a	
			color sample of the tar-	
			get excluding any opti-	
			cal interference from	
			external light sources.	
			You should not disable	
			ambient light compen-	
			sation unless the opti-	
			cal path is perfectly iso-	
			lated. Otherwise exter-	
			nal light will inevitably	
			interfere with the color	
			sampling.	
		averages	Number of previous	
		AverageSampleCount (integer), minimum: 1, op-	samples to be aver- aged for every sam-	
		tional	pling result. A rolling	
		tional	averaging algorithm is	
			applied to the samples.	
	errors	Error []	applied to the equiples.	
	Array of Error, required			
		code	machine-readable	
		String, optional	unique error code	
		mapping	a reference to the pa-	
		String, optional	rameter that caused the	
		5, 1	error	
		message	human-readable error	
		String, optional	description	

1.2.8 Defaults

Collection of defaults and settings for specific tasks. Apart from the custom user-defined values this API also returns factory defaults. Defaults are implicitly applied during specific actions like the creation of matchers or when executing certain behaviours.

Client applications can use this API endpoint to store settings that are independent from their current session or the client itself. Type and validation checks are the responsibility of the client application.

Be aware that defaults may be applied at runtime (like the creation of a matcher) and thus an invalid default value will break the application at a later point in time. Mind the notes below to prevent such problems when operating with the defaults API.

Default values are not subject to any kind of validation, but are handled as raw data. An invalid hold_time (negative, string instead of a number) for a matcher will become effective during the creation of a new matcher and only if the request for creating the matcher did not contain

a hold_time field. In such a case the creation of a matcher would fail. It is therefore paramount to properly validate default values.

- The fields object_type and key are yours to choose. This allows applications to store sessionand client-independent data (e.g. an interface theme, color scheme, etc.). In order to avoid name-collisions with internal default-fields you should prefix the object_type or key field with x- (e.g. x-theme instead of theme). The API will never use fields internally that start with x-.
- The API resolves defaults with the following steps. Applications should implement the same behaviour, when resolving default values:
 - 1. Check if an element in the defaults collection matches both object type and key
 - 2. In case it does: use this value
 - 3. In case it does not: use the value from factory defaults

1.2.8.1 Retrieve DefaultMapValues

GET / defaults

Returns two collections of *DefaultMapValue* objects. defaults contains all custom defaults and factory_defaults all those that are part of the factory settings. The latter can't be changed but custom defaults take precedence over factory defaults.

Response

Code	Body	application/json		
200	Properties (object)			
	data	data		
	object, required	1	5 () 14)/ 1 5	
		defaults Array of DefaultsMapValue, required	DefaultsMapValue[]	
			uuid	unique identifier
			<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re- quired , read-only	(UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC
			quirou, rodu omy	9834-8
			object_type string, required, read- only	Name of the object the default is meant for
			key string, required, read-	name of the object's property
			value any, required	Actual default value for the object's property
		factory defaults	DefaultsMapValue[]	City
		Array of <u>DefaultsMapValue</u> , required	Dela dita Map Value []	
			uuid UUID (string), pattern: ^ [a-f0-9-]+\$, re- quired, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
			object_type string, required, read- only	Name of the object the default is meant for
			key string, required, read- only	name of the object's property
			value any, required	Actual default value for the object's property
	errors Array of Error, required	Error[]		
		code	machine-readable	
		String, optional	unique error code	
		mapping String, optional	a reference to the pa- rameter that caused the error	

Code	Body	application/json		
		message	human-readable error	
		String, optional	description	

1.2.8.2 Create DefaultMapValues

```
POST / defaults
```

All valid attributes for a PUT request of a defaults object are allowed. The attributes <code>object_type</code>, <code>key</code> and <code>value</code> are required. The API ensures that only one combination of <code>object_type</code> and <code>key</code> is present at a time. A POST request therefore doesn't necessarily increase the number of elements in the collection.

Request

Body	application/json
Properties (DefaultsMapValue)	
uuid	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec.
<u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	X.667, and ISO/IEC 9834-8
object_type	Name of the object the default is meant for
string, required , read-only	
key	name of the object's property
string, required , read-only	
value	Actual default value for the object's property
any, required	

Examples

```
Matcher: Tolerance
```

```
{
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object_type": "matcher",
  "key": "tolerance",
  "value": {
      "shape": "sphere",
      "limits": {
            "radius": 4
       }
    }
}
```

Matcher: Hold Time

```
{
  "uuid": "55b35901-lea6-4b3d-864a-60af15a9b0c5",
  "object_type": "matcher",
  "key": "hold_time",
  "value": 0
}
```

Matcher: reset output after Hold Time expiry

```
"uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset_output_after_hold_time_expired",
  "value": false
}
```

Autogain: number of samples used for averaging

```
{
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object_type": "autogain",
  "key": "averages",
  "value": 0
```

Response

Code	Body	application/json	
200 400	Properties (object)		
	data DefaultsMapValue, required	DefaultsMapValue	
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, re- quired, read-only	unique identifier (UUID) as defined by RFC 4122, ITU- T Rec. X.667, and ISO/IEC 9834-8
		object_type string, required, read-only	Name of the object the default is meant for
		key string, required , read-only	name of the object's property
		value any, required	Actual default value for the object's property
	errors Array of Error, required	Error[]	
		code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description
	May return the following error control LPLC.validation.collection_size_ex		

1.2.8.3 Remove multiple oder all DefaultMapValues

DELETE / defaults

Remove a selection of DefaultMapValues either based on a given filter argument (if supported for this collection) or remove all DefaultMapValues from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DE-LETE request against an empty collection or for a filtered DELETE request against a collection without DefaultMapValues matching the filter.

Response

Code	
204	The empty response indicates success

1.2.8.4 Modify DefaultMapValue

PUT / defaults / {itemId}

Modify the default's value. The fields unid, object type and key are invariable.

Request

rioquoot		
PathVariables		
itemId		
String, required		
Body	application/json	
Properties (object)		
data	DefaultsMapValue	
DefaultsMapValue, required		

uuid	unique identifier (UUID) as defined by
<u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, re-	RFC 4122, ITU-T Rec. X.667, and
quired, read-only	ISO/IEC 9834-8
object_type	Name of the object the default is meant
string, required , read-only	for
key	name of the object's property
string, required, read-only	
value	Actual default value for the object's
any, required	property

Examples

```
Matcher: Tolerance
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object type": "matcher",
  "key": "tolerance",
  "value": {
    "shape": "sphere",
    "limits": {
      "radius": 4
  }
}
Matcher: Hold Time
  "uuid": "55b35901-1ea6-4b3d-864a-60af15a9b0c5",
  "object type": "matcher",
  "key": "hold time",
  "value": 0
}
Matcher: reset output after Hold Time expiry
  "uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset output after hold time expired",
  "value": false
}
Autogain: number of samples used for averaging
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object type": "autogain",
  "key": "averages",
  "value": 0
}
```

Response

Code	Body	application/json	
200	Properties (object)		
400			
404			
	data	DefaultsMapValue	
	DefaultsMapValue, required		

	uuid UUID (string), pattern: ^ [a-f0-	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
	9-]+\$, required , read-only	
	object_type string, required, read-only	Name of the object the default is meant for
	key string, required, read-only	name of the object's property
	value any, required	Actual default value for the object's property
errors Array of Error, required	Error[]	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description
May return the following LPLC.not found.collection		

1.2.8.5 Delete DefaultMapValue

DELETE / defaults / {itemId}

Deletes a single DefaultMapValue

Request

PathVariables	
itemId	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LPLC not found collection item

1.2.8.6 Get DefaultMapValue

GET / defaults / {itemId}

Returns a single DefaultMapValue.

Request

PathVariables	
itemId	
String, required	

Response

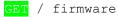
Code	Body	application/json	
200	Properties (object)		
	data <u>DefaultsMapValue</u> , required	DefaultsMapValue	
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
		object_type	Name of the object the de-
		string, required , read-only	fault is meant for
		key string, required, read-only	name of the object's prop- erty
		value	Actual default value for the
		any, required	object's property
	errors Array of Error, required	Error[]	

	code	machine-readable unique
	String, optional	error code
	mapping	a reference to the parameter
	String, optional	that caused the error
	message	human-readable error
	String, optional	description
May return the following error codes		
	und.collection.item	

1.2.9 Firmware

The firmware is stored on the device and controls all of its aspects. It can be upgraded and safely be reset to the factory defaults.

1.2.9.1 Get Firmware Information



Returns information about the currently running firmware.

Response

Code	Body	application/json	
200	Properties (object)		
	data	Information describing a firmware version.	
	FirmwareInformation, requi-		
	red		
		FirmwareInformation	
		id	unique ID of the currently running
		FirmwareBuildId (string), pattern: ^ [a-f0-	firmware image
		9]+\$, required	
		channel	Describes the kind of a publication
		ReleaseChannel (string), one of [sta-	Releases on the stable channel
		ble, feature], default: stable, required	are generally considered well-tested
			and are recommended for use in
			production.
			Releases on the feature add new
			features but haven't been tested as
			much as a stable release. Feature
			releases can but should only be
			used in production with careful consideration.
		created on	time this firmware build was created
		Timestamp (string), required	time this inniwate balla was created
		name	human-readable name of this re-
		string, required	lease
		notes	Release notes formatted as mark-
		string, required	down
		version	version of a firmware
		FirmwareVersion (string), required	
		works with	compatible device models (see
		Array of string, required	model key in /api/device)
	errors	Error[]	
	Array of Error, required		
		code	machine-readable unique error code
		String, optional	
		mapping	a reference to the parameter that
		String, optional	caused the error
		message	human-readable error description
		String, optional	

1.2.9.2 Firmware Image Upload

GET / firmware / images

Returns the list of all active partial or complete firmware uploads.

Response

Code Body application/json	
----------------------------	--

200	Properties (object)		
	data	FirmwareImageUpload []	
	Array of FirmwareImageU-		
	pload, required		
		uuid	Unique ID of a firmware upload
		<u>UUID</u> (string), pattern: ^[a-f0-	
		9-]+\$, required , read-only	
		build_id	unique ID of the currently running firmware
		HashDigest (string), pattern: ^ [a-	image
		f0-9]+\$, required	
		status	Current status of the firmware upload
		string, one of [incomplete,	
		complete, invalid_signa-	Incomplete
		ture, processing_failure,	
		malformed_content, de-	the number of bytes received is lower than
		vice_mismatch], required	the number of bytes that have been an-
			nounced complete
			the firmware upload is complete and the
			new firmware can be applied invalid_signa-
			ture
			the firmware checksum didn't match the ex-
			pected value processing_failure
			an internal undefined error occurred while
			processing the firmware malformed_content
			the uploaded firmware image uses an unex-
			pected format or misses essential infor-
			mation device_mismatch
			the firmware image can not be applied to this device
	<u> </u>	uploaded size	number of uploaded bytes
		integer, minimum: 0, required	Turnber of uploaded bytes
		expected size	expected total number of bytes for the firm-
		integer, minimum: 1, required	ware image
		max chunk size	maximum size for a data chunk uploaded to
		integer, minimum: 1, required	the device
—	errors	Error[]	and dovido
	Array of Error, required		
	, indy or Error, required	code	machine-readable unique error code
		String, optional	masimis rodddolo dingdo onor oodd
		mapping	a reference to the parameter that caused the
		String, optional	error
		message	human-readable error description
		String, optional	Transaction of the decomposition
L		g, optional	l .

1.2.9.3 Upload Firmware

POST / firmware / images

Upload a new firmware for an upgrade in separate chunks or as a single form-based file upload.

1.2.9.4 Firmware Upload methods

Two different approaches are available for the upload of a firmware image. The client selects the wanted method by using the associated request format:

- * `multipart` format: Upload the complete image in a single request.
- * `JSON` body: Successively upload single blocks of the firmware image.

File-based upload ("multipart"):

The file based approach is simple to use and preferable for most situations. It requires the use of the multipart/form-data request format. The request transmits the full firmware image file to the API. The API's response to this request is emitted as soon as all related operations are finished.

The request format multipart/form-data is commonly used for file based HTML forms. Thus it is also possible to use this firmware upload method with a simple HTML form even without any client side code.

Chunk-based upload (JSON body):

The chunk based approach requires more effort on the client side. This approach may be helpful if you want to achieve advanced flow control or status indications during the firmware upload. Use a JSON formatted request body if you want to use this method.

The initial POST request creates and returns a firmware upload entity (FirmwareImageUpload). You may use its UUID for uploading the chunks of the firmware image via subsequent POST requests to /api/firmware/images/UPLOAD_UUID/upload. The firmware upload can be finalized and applied by a POST request to /api/firmware/images/UPLOAD_UUID/apply.

1.2.9.5 Error handling

In case of an non-recoverable error the API will return a 400 (Bad Request) HTTP status code as early as possible. If the <code>apply</code> parameter has been set to a positive value the status code will be 424 (Failed Dependency).

Request

Body	multipart/form-data	application/json
Properties(object)		
firmware file FirmwareImageFile (file), required	The actual binary firmware image file. Please note that a filename (with an arbitrary value) needs to be supplied (in technical terms: the Content-Disposition header of this part of the request needs to have a name and a filename field).	
apply integer, one of [0, 1] , default: 0, required	Whether to apply the firmware once it has been received and validated. If this field is set to 1 the firmware will be applied at once, otherwise the API returns the firmware details to allow the application to send a separate request to apply the firmware.	

Response

Code	Body	application/json	
200	Properties (object)		
400			
424			
	data	A fully or partially uploaded	
	FirmwareImageUpload, requi-	firmware image to be used for	
	red	upgrading the firmware	
		FirmwareImageUpload []	
		uuid	Unique ID of a firmware upload
		UUID (string), pattern: ^ [a-	
		f0-9-]+\$, required, read-	
		only	
		build_id	unique ID of the currently running firmware im-
		HashDigest (string), pattern:	age
		^[a-f0-9]+\$, required	
		status	Current status of the firmware upload
		string, one of [incomplete,	incomplete
		complete, invalid_sig-	the number of bytes received is lower than the
		nature, pro-	number of bytes that have been announced
		cessing_failure, mal-	complete
		formed_content, de-	the firmware upload is complete and the new
		vice_mismatch], required	firmware can be applied invalid_signature

		the firmware checksum didn't match the expected value processing failure
		an internal undefined error occurred while pro-
		cessing the firmware malformed_content '
		the uploaded firmware image uses an unex-
		pected format or misses essential information
		device_mismatch
		the firmware image can not be applied to this
		device
	uploaded_size	number of uploaded bytes
	integer, minimum: 0, required	and the desired section of the section for the Con-
	expected_size	expected total number of bytes for the firm- ware image
	integer, minimum: 1, required max chunk size	maximum size for a data chunk uploaded to
	integer, minimum: 1, required	the device
errors	Error[]	THO GOVICO
Array of Error, required		
	code	machine-readable unique error code
	String, optional	·
	mapping	a reference to the parameter that caused the
	String, optional	error
	message	human-readable error description
	String, optional	
May return the following en		
LPLC.validation.missi		
LPLC.validation.strin		
LPLC.validation.non_n		
LPLC.validation.posit LPLC.validation.small		
LPLC.format.malformed		
LPLC.internal error	wp 1000	

1.2.9.6 Get firmware image upload

GET / firmware / images / {itemId}

Returns a single firmware image upload.

Request

Path Variables	
itemId	
String, required	

Response

Code	Body	application/json	
200	Properties (object)		
	data <u>FirmwareImageUpload</u> , required	A fully or partially uploaded firm- ware image to be used for upgrad- ing the firmware	
		FirmwareImageUpload []	
		uuid <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	Unique ID of a firmware upload
		build_id HashDigest (string), pattern: ^[a- f0-9]+\$, required	unique ID of the currently running firmware image
		status string, one of [incomplete, complete, invalid_signa- ture, processing_failure, malformed_content, de- vice_mismatch], required	Current status of the firmware upload incomplete the number of bytes received is lower than the number of bytes that have been announced complete the firmware upload is complete and the new firmware can be applied invalid_signature the firmware checksum didn't match the expected value processing_failure an internal undefined error occurred while processing the firmware malformed_content the uploaded firmware image uses an unexpected format or misses essential information device_mismatch

		the firmware image can not be applied to this device
	uploaded_size integer, minimum: 0, required	number of uploaded bytes
	expected_size integer, minimum: 1, required	expected total number of bytes for the firm- ware image
	max_chunk_size integer, minimum: 1, required	maximum size for a data chunk uploaded to the device
errors Array of Error, required	Error[]	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description
May return the following LPLC.not_found.colle		

1.2.9.7 Delete firmware image upload

GET / firmware / images / {itemId}

Deletes a single firmware image upload.

Request

Path Variables	
itemId	
String, required	

Response

Code		
204	The empty response indicates success.	
	May return the following error codes	
	LPLC.not_found.collection.item	

1.2.9.8 Switch to new Firmware

POST / firmware / images / {itemId} / apply

Applies the firmware to the device thus overwriting the current system image followed by a reboot of the device.

Request

1 to quoot	
Path Variables	
itemId	
String, required	

Response

Code	
204	The empty response indicates success.
400	
424	
500	
	May return the following error codes
	LPLC.format.malformed.upload
	LPLC.internal error

1.2.9.9 Upload Chunk of Firmware

POST / firmware / images / {itemId} / upload

Uploads a data chunck (see max_chunk_size) for this firmware image. The Content-Range is send by the client and used by the server to determine where the chunk is inserted into the final image. Uploads must happen synchronous so that every chunks start address is defined as (last chunk end address + 1).

New chunks can be uploaded as long as the firmware status is reported as incomplete. All other status indicate an either successful or defective firmware upload. In case of a permanent failure all subsequent chunk uploads will be terminated with a HTTP 400 (Bad Request) status code.

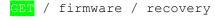
Request

Path Variables	
itemId	
String, required	
Request Headers	
Content-Range	Defines where the chunk is positioned in the firmware image
string, pattern: ^bytes\s+\d+-\d+/\d+\$, required	file.

Response

Code		
204	The empty response indicates success.	
400		
	May return the following error codes	
	LPLC.format.malformed.upload	
	LPLC.header.content range.conflicts	
	LPLC.header.content range.invalid	
	LPLC.header.content range.missing	
	LPLC.upload.missing chunk	
	LPLC.payload too big	

1.2.9.10 Get Recovery Firmware Information



Returns information about the current recovery firmware.

Response

Code	Body	application/json	
200	Properties (object)	-	
	data FirmwareInformation, required	Information describing a firmware version.	
		FirmwareInformation	
		id FirmwareBuildId (string), pattern: ^[a-f0-9]+\$, required	unique ID of the currently running firmware image
		channel ReleaseChannel (string), one of [stable, feature] , de- fault: stable, required	Describes the kind of a publication Releases on the stable channel are generally considered well-tested and are recommended for use in production. Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.
		created_on <u>Timestamp</u> (string), re- quired	time this firmware build was created
		name string, required	human-readable name of this release

	notes string, required	Release notes formatted as markdown
	version FirmwareVersion (string), required	version of a firmware
	works_with Array of string, required	compatible device models (see model_key in /api/device)
errors Array of Error, required	Error[]	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description

1.2.9.11 Upgrade Recovery Firmware

POST / firmware / recovery / upgrade-from-current

Replaces the stored recovery image with the current system firmware. This is helpful in case you want to update the recovery image to a more recent version.

The factory image merely contains the actual firmware. It does not store the sensors configuration or settings.

The update process will take several minutes.

Response

Code	
204	The empty response indicates success
500	

1.2.9.12 Get Firmware Settings

GET / firmware / settings

Returns current settings regarding the firmware and possible upgrades.

Response

Code	Body	application/json	
200	Properties (object)		
	data FirmwareSettings, required	Settings related to the device's firmware and upgrades.	
		FirmwareSettings	
		release_channel ReleaseChannel (string), one of [stable, feature], default: stable, required	Describes the kind of a publication Releases on the stable channel are generally considered well-tested and are recommended for use in production. Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.
	errors Array of Error, required	Error[]	
		code String, optional	machine-readable unique error code

	mapping	a reference to the parameter
	String, optional	that caused the error
	message	human-readable error
	String, optional	description

1.2.9.13 /firmware/settings

PUT / firmware / settings

Request

Body	application/json
Properties (FirmwareSettings)	
release_channel	Describes the kind of a publication
ReleaseChannel (string), one of [stable, feature] , default: stable, required	Releases on the stable channel are generally considered well-tested and are recommended for use in production. Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.

Examples

```
{
   "release_channel": "stable"
}
```

Response

Code	Body	application/json	
200 400	Properties (object)		
	data FirmwareSettings, required	Settings related to the device's firmware and upgrades.	
		FirmwareSettings	
	errors	release_channel ReleaseChannel (string), one of [stable, feature] , default: stable, required Error[]	Describes the kind of a publication Releases on the stable channel are generally considered well-tested and are recommended for use in production. Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.
	Array of Error, re- quired		
		code String, optional	machine-readable unique error code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description
	May return the followard LPLC.validation	owing error codes	

1.2.9.14 Get Firmware Status

GET / firmware / settings

Returns information about the currently running firmware version.

Response

Code	Body	application/json	
200	Properties (object)		
	data	Information describing the currently running firmware.	

FirmwareRunningInformation, required		
	FirmwareRunningInformation	
	build_id <u>FirmwareBuildId</u> (string), pattern: ^[a- f0-9]+\$, required	unique ID of the currently running firmware image
	source_url any of string or null, optional	Absolute base URL of a firmware repository delivering firmware images suitable for this device
	version FirmwareVersion (string), required	version of a firmware
errors Array of Error, required	Error[]	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error description

1.2.10 Access Control

Manage access to data and settings of the sensor.

1.2.10.1 Users

Users are the identities that are allowed to access the API.

A password is required for authenticating a user during login.

1.2.10.2 Roles

Roles describe a set of permissions. Each user may belong to multiple roles.

The role named anonymous is special: it cannot be assigned to users. Instead it describes the set of permissions that are granted to every unauthenticated as well as authenticated request. Thus this special role can be considered the minimum set of permissions that is open for everyone.

1.2.10.3 Auswertung der Zugriffsberechtigungen

All actions can be executed without authentication by default as long as no user account has been created.

If at least one user exists, access control is applied by the API. Thus permissions are checked before an incoming request is processed.

Authentication is conducted via the HTTP Basic Authentication Scheme.

Authorisation for a given action (e.g. *view* or *edit*) targeted at a specific API endpoint is verified as follows:

- Which kind of action is requested by the user: view (GET) or edit (POST, PUT, DELETE)?
- To which access scope does the target API endpoint belong (e.g. network)?
- Which roles are assigned to the authenticated user (e.g. operator)?
- Does at least one of the permissions of any of these roles belong to the requested *scope* and contain the requested *action*?

The last of the questions above decides, whether a request is processed or rejected.

1.2.10.4 HTTP Responses

The following additional HTTP responses can be emitted while access control is active:

HTTP 401 Unauthorized

is returned if the request requires authentication, but no credentials were supplied or the given credentials were rejected (e.g. unknown user or wrong password). Web applications interfacing the API may want to use the X-WWW-Authenticate-Scheme-Disable header (see below) in order to prevent the user's browser from intercepting this error response.

HTTP 403 Forbidden

is returned if the given credentials were valid, but the associated user is not allowed to request the given action in the target scope.

The client may specify the X-WWW-Authenticate-Scheme-Disable HTTP header in any request. The content of this header is expected to be a comma-separated list of authentication schemes (see RFC 7235). These authentication schemes will not be advertised by the API as part of the WWW-Authenticate header in its response. The following example header content is suitable for preventing a browser from interfering with authentication related responses: -WWW-Authenticate-Scheme-Disable: Basic, Digest.

1.2.10.5 Inspect Access Control Scopes and Actions



GET / access

Inspect the available aspects of the access control setup.

Response

Code	Body	application/json	
200	Properties (object)		
	data AccessControlFeatures, required	AccessControlFeatues	
		actions Array of AccessAction (string), required	Available actions that can be allowed or denied via permissions.
		scopes Array of AccessScope (string), required	Available scopes that can be accessed with the different actions.
	errors Array of Error, required	Error[]	
		code String, optional	machine-readable unique er- ror code
		mapping String, optional	a reference to the parameter that caused the error
		message String, optional	human-readable error description

1.2.10.6 Login into an account



This endpoint can be used to create a session for the current user agent or testing credentials.

If the caller provided valid credentials and didn't set the session timeout to 0 the response will contain a Set-Cookie header that contains a session token used for future authentication and which is automatically handled by XMLHttpRequest and fetch.

Once a session token has been issued to the user agent any subsequent request to any of the API endpoints will reset the session timeout to the value provided by session timeout as long as the session did not already expire at the point in time when the request was sent to the API. Therefore if the session timeout was set to 15 minutes and a request was made every 10 minutes the session would be valid indefinitely. This does not apply if the endpoint is explicitly excluded from resetting the session timeout (like GET /api/access/login).

To test credentials the client may send them along with <code>session_timeout</code> set to 0. The response status code will indicate if the credentials are valid but omits the <code>Set-Cookie</code> header thus retaining the currently used session token.

Be aware that the API may start to rate-limit the endpoint if too many invalid credentials have been send to it. Make sure that you implement some kind of user feedback in case of responses with HTTP status code 429 like deactivating the login form and/or displaying the remaining time until a new login may be attempted.

Request

Body	application/json
Properties (object)	
username string, required	The name of the user that should be authenticated
password string, required	The password of the user that should be authenticated
session_timeout number , default: 1200, optional	The lifetime of the session on the server-side in seconds. Passing 0 will prevent the API from setting the Set-Cookie header and allows for checking credentials without creating a new session.

Response

Code	T		-	
200	Authentication with the provided			
403	credentials was successful			
429	ordernials was successful			
	Body	application/json		
	Properties (object)	принашенную н		
	data	Describes the cur-		
	LoginInformation, required	rently active login		
	<u>====</u> , ======	provided by the user		
		agent		
		LoginInformation		
		logged in user	The currently logged in user.	
		any of User or null,	Is null if the credentials	
		required	didn't match any known user	
			or have expired.	
			User	
			name	unique name identi-
			string, pattern: ^ [\w-] +\$,	fying an account
			required, read-only	
			password	Password assigned
			string, optional	to this account (only
				writable; never re-
				turned in responses).
				Either a password
				or a pass-
				word_hash needs
				to be supplied when
				creating a new user
				or changing a pass-
				word.
			password_hash	Password hash as-
			HashDigest (string), pattern:	signed to this ac-
			^[a-f0-9]+\$, optional	count. Either a
				password or a
				password_hash
				needs to be supplied when creating a new
				user or changing a
				password.
			roles	The roles assigned
			Array of string, optional	to an account define
				its set of permis-
				sions.
		session timeout	Number of seconds this ses-	
		any of number or	sion has left before expiring.	
		null, required	Is null if the provided cre-	
			dentials could not be	

		matched to any active sessions, if the session expired or if the supplied authentication mechanism does not support sessions (e.g. HTTP Authentication).	
errors Array of Error, required	Error[]	,	
	code String, optional	machine-readable unique er- ror code	
	mapping String, optional	a reference to the parameter that caused the error	
	message String, optional	human-readable error description	

1.2.10.7 Retrieve Information about the currently used Credentials

GET / access / login

Response

Code	Body	application/json		
200	Properties (object)			
	data LoginInformation, required	Describes the cur- rently active login pro- vided by the user agent		
		LoginInformation		
		logged_in_user any of <u>User</u> or null, re- quired	The currently logged in user. Is null if the credentials didn't match any known user or have expired.	
			User	
			name string, pattern: ^[\w-]+\$, re- quired, read-only	unique name identi- fying an account
			password string, optional	Password assigned to this account (only writable; never returned in responses). Either a password or a password hash needs to be supplied when creating a new user or changing a password.
			password_hash HashDigest (string), pattern: ^[a-f0-9]+\$, optional	Password hash assigned to this account. Either a password or a password_hash needs to be supplied when creating a new user or changing a password.
			roles Array of string, optional	The roles assigned to an account define its set of permissions.
		session_timeout any of number or null, required	Number of seconds this session has left before expiring. Is null if the provided credentials could not be matched to any active sessions, if the session expired or if the supplied authentication mechanism does not support sessions (e.g. HTTP Authentication).	
	errors Array of Error, required	Error[]		

code String, optional	machine-readable unique error code
mapping String, optional	a reference to the parameter that caused the error
message String, optional	human-readable error description

1.2.10.8 Logout / Invalidate any current credentials

DELETE / access / login

Response

Code	
204	The provided credentials were successfully invalidated.
400	
403	

1.2.10.9 Create User

POST / access / users

Create a new User.

All supported data attributes in the body of the request are optional.

Request

Body	application/json
Properties (User)	
name	unique name identifying an account
string, pattern: $\lceil \sqrt{w} - \rceil + \$$, required , read-only	
password	Password assigned to this account (only writable; never re-
string, optional	turned in responses). Either a password or a pass-
	word hash needs to be supplied when creating a new user
	or changing a password.
password_hash	Password hash assigned to this account. Either a pass-
HashDigest (string), pattern: ^[a-f0-9]+\$, optional	word or a password hash needs to be supplied when cre-
	ating a new user or changing a password.
roles	The roles assigned to an account define its set of permis-
Array of string, optional	sions.

Examples

```
{
   "name": "alice"
}
```

Response

Code	Body	application/json	
200 400	Properties (object)		
	data User, required	User	
		name	unique name identifying an ac-
		string, pattern: ^ [\w-] +\$, required, read-only	count
		password string, optional	Password assigned to this account (only writable; never returned in responses). Either a password or a pass-word_hash needs to be supplied when creating a new user or changing a password.
		password_hash HashDigest (string), pattern: ^[a-f0-9]+\$, optional	Password hash assigned to this account. Either a password or a password hash needs to be

		supplied when creating a new user or changing a password.
	roles Array of string, optional	The roles assigned to an account define its set of permissions.
errors Array of Error, required	Error[]	
	code String, optional	machine-readable unique error code
	mapping String, optional	a reference to the parameter that caused the error
	message String, optional	human-readable error descrip- tion
	lowing error codes n.collection_size_exceeded	

1.2.10.10Remove multiple or all Users

DELETE / access / users

Remove a selection of Users either based on a given filter argument (if supported for this collection) or remove all Users from the collection.

All delete requests result in an empty success response (204). This is even valid for a non-filtered DE-LETE request against an empty collection or for a filtered DELETE request against a collection without Users matching the filter.

Response

Code	
204	The empty response indicates success

1.2.10.11Retrieve User

GET / access / users

Retrieves a list of available Users

Response

Code	Body	application/json		
200	Properties (object)			
	data Object, required	data		
		users Array of <u>User</u> , re- quired	User[]	
			name string, pattern: ^ [\w-]+\$, required, read-only	unique name identifying an ac- count
			password string, optional	Password assigned to this account (only writable; never returned in responses). Either a password or a pass-word_hash needs to be supplied when creating a new user or changing a password.
			password_hash HashDigest (string), pattern: ^[a-f0-9]+\$, optional	Password hash assigned to this account. Either a password or a password_hash needs to be supplied when creating a new user or changing a password.
			roles Array of string, optional	The roles assigned to an account define its set of permissions.
	errors Array of Error, required	Error[]		

	code	machine-readable unique error	
	String, optional	code	
	mapping	a reference to the parameter	
	String, optional	that caused the error	
	message	human-readable error descrip-	
	String, optional	tion	

1.2.10.12Delete User

```
DELETE / access / users / {name}
```

Deletes a single User.

Request

Path Variables	
name	
String, required	

Response

Code	
204	The empty response indicates success
	May return the following error codes
	LPLC.not found.collection.item

1.2.10.13Modify User

```
PUT / access / users / {name}
```

Modifies a single User.

Request

Path Variables	
name	
String, required	
Body	application/json
Properties (User)	
name	unique name identifying an account
string, pattern: ^ [\w-]+\$, required, read-only	
password	Password assigned to this account (only writable; never re-
string, optional	turned in responses). Either a password or a pass-
	word hash needs to be supplied when creating a new user
	or changing a password.
password_hash	Password hash assigned to this account. Either a pass-
HashDigest (string), pattern: ^[a-f0-9]+\$, optional	word or a password hash needs to be supplied when cre-
	ating a new user or changing a password.
roles	The roles assigned to an account define its set of permis-
Array of string, optional	sions.

Examples

```
{
   "name": "alice"
}
```

Response

Code	Body	application/json	
200	Properties (object)		
400			
404			
	data	User[]	
	User, required	-	
		name	unique name identifying an account

	string, pattern: ^ [\w-] +\$, re-	
	quired, read-only	
	password	Password assigned to this account (only
	string, optional	writable; never returned in responses). Ei-
		ther a password or a password_hash
		needs to be supplied when creating a
		new user or changing a password.
	password_hash	Password hash assigned to this account.
	HashDigest (string),	Either a password or a password_hash
	pattern: ^[a-f0-	needs to be supplied when creating a
	9]+\$, optional	new user or changing a password.
	roles	The roles assigned to an account define
	Array of string, op- tional	its set of permissions.
errors	Error[]	
Array of Error, required		
	code	machine-readable unique error code
	String, optional	
	mapping	a reference to the parameter that caused
	String, optional	the error
	message String, optional	human-readable error description
May return the following error co	odes	

1.2.10.14Get User

GET / access / users /{name}

Returns a single User.

Request

Path Variables	
name	
String, required	

Response

Code	Body	application/json	
200	Properties (object)		
	data	User[]	
	User, required		
		name	unique name identifying an account
		string, pattern:	
		^ [$\w-$] +\$, re-	
		quired, read-only	
		password	Password assigned to this account (only
		string, optional	writable; never returned in responses). Ei-
			ther a password or a password hash
			needs to be supplied when creating a
			new user or changing a password.
		password_hash	Password hash assigned to this account.
		HashDigest (string),	Either a password or a password hash
		pattern: ^[a-f0-	needs to be supplied when creating a
		9]+\$, optional	new user or changing a password.
		roles	The roles assigned to an account define
		Array of string, op-	its set of permissions.
		tional	
	errors	Error[]	
	Array of Error, required	code	machine readable unique error and
		String, optional	machine-readable unique error code
		mapping	a reference to the parameter that caused
		String, optional	the error
		message	human-readable error description
		String, optional	Trainan-readable error description
	May return the following error	ouring, optional	
	codes		

LPLC.not_found.collec-	
tion.item	

1.2.10.15Retrieve AccessRoles

GET / access / roles

Retrieves a list of available AccessRoles.

Response

Code	Body	application/json			
200	Properties (object)				
	data object, required	data			
		roles Array of Access- Role, required	AccessRole[]		
			id string, required		
			permissions Array of Ac- cessPermission, required	AccessPermission[]	
				scope AccessScope (string), required	Every API endpoint belongs to an ac- cess control scope. A scope combined with a number of ac- tions forms a permis- sion.
				actions Array of Acces- sAction (string), required	
	errors Array of Error, required	Error[]			
		code String, optional	machine-readable unique error code		
		mapping String, optional	a reference to the parameter that caused the error		
		message String, optional	human-readable error description		

1.3 REST-API Type Reference

1.3.1 AccessAction

Type Information

AccessAction (string)

Examples:

view

1.3.2 AccessControlFeatures

Properties

actions	Available actions that can be allowed or denied via permissions.
Array of AccessAction (string), required	
scopes	Available scopes that can be accessed with the different actions.
Array of AccessScope (string), required	

Examples

```
{
  "actions": [
    "view",
    "edit"
],
  "scopes": [
    "access",
    "miscellaneous",
    "network",
    "notify",
    "peripherals",
    "sensor",
    "settings",
    "system"
]
}
```

1.3.3 AccessPermission

A permission defines a set of allowed actions in a specific access control scope.

Properties

scope Array of AccessScope (string), required	Every API endpoint belongs to an access control scope. A scope combined with a number of actions forms a permission.
actions Array of AccessAction (string), required	

1.3.4 AccessRole

A role describes a set of permissions. Each user may belong to multiple roles.

Properties

id		
string, required		
permissions	AccessPermission[]	
Array of AccessPermission, required		
	scope	Every API endpoint belongs to an access
	AccessScope (string), required	control scope. A scope combined with a
		number of actions forms a permission.
	actions	

Array of AccessAction (string),	
required	

1.3.5 AccessScope

Every API endpoint belongs to an access control scope. A scope combined with a number of actions forms a permission.

Type Information

AccessScope (string)

Examples

network

1.3.6 Action

The sensor allows the connection of events with actions. Actions can be related to the sensor operations or the information handled by the sensor (e.g. the list of stored detectables).

The Action consists of a unique name and a set of optional arguments.

The list of available Actions and their possible arguments can be retrieved via /api/actions.

Properties

name	Unique name of the action
string, required	
arguments	arguments
object, required	

1.3.7 ActionEnableSwitchingOutput

Apply the *output pattern* of the currently detected matcher to the switching outputs of the sensor.

Properties

name	Unique name of the action
string, required	
arguments	arguments
object, required	

1.3.8 ActionKeyLock

Change the *locked* state of the keypad. This allows or disallows local access to the sensor via the keypad.

Properties

arguments object, required	arguments
name	Unique name of the action
string, required	

Examples

```
{
  "name": "keylock",
  "arguments": {
     "locked": true
  }
}
```

1.3.9 ActionRemoveAllDetectables

Remove all stored Detectables belonging to any Matcher.

Properties

name	Unique name of the action
string, required	
arguments	arguments
object, required	

1.3.10 ActionRemoveAllMatchers

Remove all stored Matchers (including the related detectables).

Properties

name	Unique name of the action
string, required	
arguments	arguments
object, required	

1.3.11 ActionResultEnableSwitchingOutput

After each sampling period a Detection Result is determined based on the currently sampled color and the contents of the color storage (matchers and detectables).

In addition to the sampled color, the Detection Result includes transitions and events on all input lines during the last sample period, as well as the state of the switching outputs during the following sampling period.

Properties

uuid UUID (string), pattern: ^ [a- f0-9-]+\$, required, read- only timestamp TimestampBackendUptime (number), required	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8 The timestamp (given in microseconds) is based on the uptime of the internal analog sensor backend. It may get reset to zero under specific conditions.		
corrected_color CorrectedColor, required	Representation of a color in the colorspace XYZ.		
	values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
transformed_color TransformedColor, required	A color represented by a co- ordinate in the colorspace. The array indices of the values property match the order of the color- space.axes property of currently used detection profile.		
	values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
representations ColorRepresentations, required	Pre-calculcated visual repre- sentations of a color suitable for rendering		
	Array of number, minimum items: 3, maximum items: 3, required	RGB color array represent- ing the axes r, g, and b in that order. Values are floats between 0 and 1.	
inputs InputsState, required	The state of all inputs during a given period is specified by a list of possible events combined with a boolean		

	value indication if the street		
	value indicating, if the given		
	event occurred within the period.		
		The boolean value indicates	
	//		
	boolean, required	whether the named input	
		event occurred during the	
		last period.	
detection	After each sampling period		
ColorMatchingResult, requi-	the retrieved color value is		
red	compared to the stored de-		
	tectables (color positions).		
	Detectables are ignored, if		
	the tolerance shape of their		
	corresponding matcher		
	does not encompass the		
	current sample. Finally the		
	closes suitable detectable is		
	selected as the winner of the		
	color matching operation.		
	The corresponding matcher		
	determines the state of the		
	sensor for the duration of		
	the next sampling period.		
	matcher	Deprecated: use "cho-	
	any of <u>UUID</u> (string) or null,	sen_matcher_id" instead	
	optional, Deprecated		
	chosen matcher id	unique identifier of the se-	
	any of UUID (string) or null,	lected matcher	
	required		
	distances	Distance between the sam-	
	Array of any of number or	ple's color position and the	
	null, required	selected matcher's closest	
		color position along the	
		three axes of the color	
		space.	
		The array contains three	
		'null' values, if no suitable	
		matcher was found for the	
		current color sample.	
	output_pattern	Currently active state of the	
	CurrentSwitchingOutputs-	Switching Outputs. Beware	
	State, required	that this may deviate from	
	, ,	the specified output states	
		of the current best matcher,	
		since settings like triggered	
		input or hold time influence	
		update process for the	
		Switching Outputs	
		states	List of True/False values de-
		Array of any of boolean or	scribing the current states of
		null, required	the Switching Outputs
signal_level	Der Signalpegel zeigt die	/	<u> </u>
number, required	Verwendung des internen		
, "	ADC-Abtastbereichs an		
	1	1	

Examples

```
5.271166801452637,
    -32.290863037109375
  ]
} ,
"representations": {
  "RGB": [
    0.6569499359485452,
    0.7560762577592035,
    0.9910401649653352
  ]
},
"inputs": {
  "trigger_0_level_low": true,
  "trigger_0_edge_falling": false,
"trigger_0_edge_rising": false,
"trigger_0_level_high": false
"detection": {
  "chosen matcher id": null,
  "distances": [
    null,
    null,
    null
  ],
  "output pattern": {
    "states": [
      true,
       true,
       true
    ]
  }
} ,
"signal_level": 0.7
```

1.3.12 ActionResultKeyLock

Properties

locked	New state of the keypad locking.
boolean, required	

Examples

```
{
    "locked": true
```

1.3.13 ActionResultRemoveAllDetectables

The response is empty and returns HTTP status 204.

Type Information

ActionResultRemoveAllDetectables (string)

1.3.14 ActionResultRemoveAllMatchers

The response is empty and returns HTTP status 204.

Type Information

ActionResultRemoveAllDetectables (string)

1.3.15 ActionResultRunAutogain

Optional settings for customizing the Autogain procedure.

Properties

level	Target value for the auto-gain procedure	
Number, default: 0.8, minimum: 0.01, maximum: 1, optional		
minimum_sample_rate	Desired sample rate (the default is the current sample rate)	
SampleRate (number), minimum: 0.02, optional		
enable_internal_emitter	controls the power of the internal light source	
Boolean, default: true, optional		
enable_ambient_light_compensation Boolean, default: true, optional	Control the ambient light compensation procedure. This setting is only relevant if enable_internal_emitter is set to true. The ambient light compensation leads to a pulsed usage of the internal light emitter. Samples are collected for alternating light and dark phases. This allows to calculate a color sample of the target excluding any optical interference from external light sources. You should not disable ambient light compensation unless the optical path is perfectly isolated. Otherwise external light will inevitably interfere with the color sampling.	
averages AverageSampleCount (integer), minimum: 1, optional	Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.	

Examples

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
```

1.3.16 ActionResultTeachDetectable

A detectable represents the numeric position in a colorspace. It is connected to a *Matcher*.

Properties

uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
alias Alias (integer), required, read-only	A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.	
matcher_id UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
color <u>TransformedColor</u> , required	A color represented by a coordinate in the colorspace. The array indices of the values property match the order of the colorspace.axes property of currently used detection profile.	
	color	
	values Array of number, minimum items: 3, maximum items: 3, required	Location in a co- lorspace
representations ColorRepresentations, optional, read- only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	
	representations	

RGB	RGB color array
Array of number, minimum items: 3, maximum items: 3,	representing the
required	axes r, g, and b in
	that order. Values
	are floats between
	0 and 1.

Examples

```
"uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
     0.476731,
      0.381263,
      0.128475
    ]
  },
  "representations": {
    "RGB": [
     0.396114,
     0.479113,
      0.552308
    ]
  }
}
```

1.3.17 ActionRunAutogain

Start an automatic adjustment of the optiocal sensor setup. See /api/sensor/detection-profiles/current/autogain for details.

Properties

name	Unique name of the action
string, required	
arguments	arguments
object, required	

1.3.18 ActionTeachDetectable

Add the currently sampled color as a Detectable to the selected matcher.

Properties

arguments object, requi- red	arguments		
	matcher_id <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, optional, read-only	The new Detectable is assigned to the Matcher identified by this UUID. In case this matcher UUID (and "matcher_output_pattern") is undefined, a new matcher is created.	
	matcher_output_pattern: object, optional	Pattern of the switching outputs to be used when selecting the target matcher for the new detectable. A suitable matcher is created, if no matcher with the specified pattern is found. This field is ignored, if "matcher_id" is not null. If no pattern is defined (an no "matcher_id"), then a new matcher is created whenever the corresponding action is executed.	

		matcher_output_pattern:	
		states Array of any of boolean or null, required	List of True/False values describing the current states of the Switching Outputs
	remove_matcher_detectables_before boolean, default: true, optional	Remove all Detectables belong- ing to the configured Matcher be- fore attaching the new Detecta- ble.	
name string, required	Unique name of the action		

Examples

```
{
  "name": "teach_single",
  "arguments": {
    "matcher_id": "3f26aff4-8650-42a0-b319-51776c443fbc",
    "remove_matcher_detectables_before": false
  }
}
```

1.3.19 ActionTrigger

An Action Trigger assigns a given set of actions with an event.

At the end of each sample period, all events are evaluated. All corresponding actions are executed afterwards.

Properties

op ooo		
uuid	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and	
UUID (string), pat-	ISO/IEC 9834-8	
tern: ^[a-f0-9-		
] +\$, required,		
read-only		
event	Any of the event names provided by /api/sensor/capabilities	
TriggerEventName	(attribute trigger_sources) is allowed.	
(string), required		
actions	List of actions to be executed after the given event.	
Array of Action, re-		
quired		
	Action[]	
	name	Unique name of the action
	String, required	
	arguments	arguments
	Object, required	

Examples

1.3.20 Alias

A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.

Type Information

Alias (integer)

Examples

4

1.3.21 AmplificationLevel

The amplification level specifies the internal configuration of an amplifier. This value is not meant to be manipulated by regular users. It should be handled *as is* (stored, transmitted and applied without modification or introspection).

Type Information

AmplificationLevel (integer)

1.3.22 AnyAction

Type Information

any of <u>ActionEnableSwitchingOutput</u>, <u>ActionTeachDetectable</u>, <u>ActionKeyLock</u>, <u>ActionRunAutogain</u>, <u>ActionRemoveAllDetectables</u> or <u>ActionRemoveAllMatchers</u>

1.3.23 AnyActionResult

Type Information

any of <u>ActionResultEnableSwitchingOutput</u>, <u>ActionResultTeachDetectable</u>, <u>ActionResultKeyLock</u>, <u>ActionResultRemoveAllDetectables</u> (string) or <u>ActionResultRemoveAllMatchers</u> (string)

1.3.24 AutogainSettings

Optional settings for customizing the Autogain procedure.

Properties

level Number, default: 0.8, minimum: 0.01, maximum: 1, op-	Target value for the auto-gain procedure
tional	
minimum_sample_rate SampleRate (number), minimum: 0.02, optional	Desired sample rate (the default is the current sample rate)
enable_internal_emitter Boolean, default: true, optional	controls the power of the internal light source
enable_ambient_light_compensation Boolean, default: true, optional	Control the ambient light compensation procedure. This setting is only relevant if <code>enable_internal_emitter</code> is set to true. The ambient light compensation leads to a pulsed usage of the internal light emitter. Samples are collected for alternating light and dark phases. This allows to calculate a color sample of the target excluding any optical interference from external light sources. You should not disable ambient light compensation unless the optical path is perfectly isolated. Otherwise external light will inevitably interfere with the color sampling.

averages	Number of previous samples to be averaged for every sam-
AverageSampleCount (integer), minimum: 1, optional	pling result. A rolling averaging algorithm is applied to the
	samples.

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
```

1.3.25 AverageSampleCount

Number of previous samples to be averaged for every sampling result. A rolling averaging algorithm is applied to the samples.

Type Information

AverageSampleCount (integer), minimum: 1

1.3.26 BaseColorTolerance

Properties

shape	Name of the geometrical shape of the tolerance. The supported tolerance shapes
ToleranceShapeName (string), re-	can be retrieved via /api/sensor/capabilities.
quired	
limits	limits
object, required	

1.3.27 BaseSerialSettings

Properties

```
Type
string, one of [none, eliza, modbus], default: eliza, required
```

1.3.28 BoxColorTolerance

Properties

limits	limits
Object, required	half_edges
	Array of number, minimum items: 3, maximum items: 3, required
shape	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via
ToleranceShapeName	/api/sensor/capabilities.
(string), required	

1.3.29 ChromaticityCoordinate

Location in a colorspace

Type Information

Array of number, minimum items: 3, maximum items: 3

1.3.30 ColorDetectable

A detectable represents the numeric position in a colorspace. It is connected to a *Matcher*.

Properties

uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
alias Alias (integer), required, read-only	A numerical value that can be used to address an item in a collection. If an alias is specified alongside an uuid attribute, that alias can be used as an alternative to address the item in URLs and other protocols like Modbus or serial interfaces.	
matcher_id <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, required, read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
color <u>TransformedColor</u> , required	A color represented by a coordinate in the colorspace. The array indices of the values property match the order of the colorspace. axes property of currently used detection profile.	
	color values Array of number, minimum items: 3, maximum items: 3, required	Location in a co- lorspace
representations ColorRepresentations, optional, read- only	Pre-calculcated visual representations of a color suitable for rendering	
	representations	
	RGB Array of number, minimum items: 3, maximum items: 3, required	RGB color array representing the axes r, g, and b in that order. Values are floats between 0 and 1.

```
"uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
      0.476731,
      0.381263,
      0.128475
    ]
 },
  "representations": {
    "RGB": [
     0.396114,
     0.479113,
      0.552308
    ]
  }
}
```

1.3.31 ColorDetectionResult

After each sampling period a Detection Result is determined based on the currently sampled color and the contents of the color storage (matchers and detectables).

In addition to the sampled color, the Detection Result includes transitions and events on all input lines during the last sample period, as well as the state of the switching outputs during the following sampling period.

Properties

uuid	unique identifier (UUID) as
UUID (string), pattern: ^ [a-	defined by RFC 4122, ITU-T
f0-9-]+\$, required, read-	Rec. X.667, and ISO/IEC
only	9834-8

r	I	I	
timestamp TimestampBackendUptime (number), required	The timestamp (given in microseconds) is based on the uptime of the internal analog sensor backend. It may get reset to zero under specific conditions.		
corrected_color CorrectedColor, required	Representation of a color in the colorspace XYZ.		
	values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
transformed_color TransformedColor, required	A color represented by a co- ordinate in the colorspace. The array indices of the values property match the order of the color- space.axes property of currently used detection		
	profile. values Array of number, minimum items: 3, maximum items: 3, required	Location in a colorspace	
representations ColorRepresentations, required	Pre-calculcated visual repre- sentations of a color suitable for rendering		
	RGB Array of number, minimum items: 3, maximum items: 3, required	RGB color array represent- ing the axes r, g, and b in that order. Values are floats between 0 and 1.	
inputs InputsState, required	The state of all inputs during a given period is specified by a list of possible events combined with a boolean value indicating, if the given event occurred within the period.		
	// boolean, required	The boolean value indicates whether the named input event occurred during the last period.	
detection ColorMatchingResult, required	After each sampling period the retrieved color value is compared to the stored detectables (color positions). Detectables are ignored, if the tolerance shape of their corresponding matcher does not encompass the current sample. Finally the closes suitable detectable is selected as the winner of the color matching operation. The corresponding matcher determines the state of the sensor for the duration of the next sampling period.		
	matcher any of <u>UUID</u> (string) or null, optional, Deprecated	Deprecated: use "cho- sen_matcher_id" instead	
	chosen_matcher_id any of <u>UUID</u> (string) or null, required	unique identifier of the se- lected matcher	
	distances Array of any of number or null, required	Distance between the sample's color position and the selected matcher's closest color position along the three axes of the color space. The array contains three 'null' values, if no suitable	

	output_pattern CurrentSwitchingOutputs- State, required	matcher was found for the current color sample. Currently active state of the Switching Outputs. Beware that this may deviate from the specified output states of the current best matcher, since settings like triggered input or hold time influence update process for the Switching Outputs.	
		states Array of any of boolean or null, required	List of True/False values de- scribing the current states of the Switching Outputs
signal_level number, required	The signal level indicates the usage of the internal ADC sampling range. This		

```
"uuid": "4575656f-efe4-4a7d-862c-9660c15cdf4e",
"timestamp": 12455148861,
"corrected color": {
  "values": [
    0.419219434261322,
    0.4271118938922882,
    0.18753691017627716
  ]
},
"transformed_color": {
  "values": [
    78.10789489746094,
    5.271166801452637,
    -32.290863037109375
  ]
},
"representations": {
  "RGB": [
    0.6569499359485452,
    0.7560762577592035,
    0.9910401649653352
"trigger_0_level_low": true,
"trigger_0_edge_falling": false,
  "trigger_0_edge_rising": false,
  "trigger_0_level_high": false
},
"detection": {
  "chosen matcher id": null,
  "distances": [
    null,
    null,
    null
  ],
  "output_pattern": {
    "states": [
      true,
      true,
      true
    ]
  }
```

```
},
"signal_level": 0.7
}
```

1.3.32 ColorDetectionResultList

Type Information

Array of ColorDetectionResult

1.3.33 ColorDetectionResultOrNil

Type Information

any of ColorDetectionResult or null

1.3.34 ColorMatcher

A matcher represents a distinguished detection result and the wanted behaviour of the sensor whenever it is encountered.

Properties

uuid	unique identifier (UUID) as	
<u>UUID</u> (string), pattern:	defined by RFC 4122, ITU-T	
^ [a-f0-9-]+\$, re-	Rec. X.667, and ISO/IEC	
quired, read-only	9834-8	
alias	A numerical value that can be	
Alias (integer), re-	used to address an item in a	
quired, read-only	collection. If an alias is speci-	
	fied alongside an uuid attrib-	
	ute, that alias can be used as	
	an alternative to address the	
	item in URLs and other proto-	
	cols like Modbus or serial in-	
	terfaces.	
name	human-readable name of the	
String, required	matcher	
tolerance	Specification of a geometric	
Any of InfiniteColor-	shape and its dimensions in	
Tolerance, Sphere-	the current colorspaces.	
ColorTolerance, Cylin-	·	
derColorTolerance or		
BoxColorTolerance, re-		
quired		
	InfiniteColorTolerance	
	limits	limits
	Object, required	
	shape	Name of the geometrical shape of the tolerance. The supported
	ToleranceShapeName	tolerance shapes can be retrieved via /api/sensor/capa-
	(string), required	bilities.
	SphereColorTolerance	
	limits	limits
	Object, required	radius
		Numer, required
	shape	Name of the geometrical shape of the tolerance. The supported
	ToleranceShapeName	tolerance shapes can be retrieved via /api/sensor/capa-
	(string), required	bilities.
	CylinderColorTolerance	
	limits	limits
	Object, required	radius
		Number, required
		half_height
		Number, required

	shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.
	BoxColorTolerance	
	limits Object, required	limits half_edges Array of number, minimum items: 3, maximum items: 3, required
	shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.
output_pattern WantedSwitchingOut- putsState, required	The combination of tristate values describes a logical state of the switching outputs of the sensor. The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.	
	uuid <u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, required, read-only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
	states Array of any of boolean or null, required	List of True/False/Null values describing the wanted states of the Switching Outputs
hold time HoldTime (number), maximum: 3153600000, required	Minimum duration (in seconds) of a matcher's output setup being applied after detection.	
reset_output_af- ter_nold_time_expired Boolean, default: false, required	Controls if the output should be reset after the hold time passed. This is helpful if you only sample by triggering in- puts and wish to reset the outputs afterwards.	
signal_color Any of string or null, re- quired	A custom color name. How and what color will be displayed is defined by the client.	

```
"uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
 "alias": 6,
 "name": "clean bottle cap",
 "tolerance": {
   "limits": {
     "radius": 2,
     "half height": 4
   "shape": "cylinder"
  "output pattern": {
   "uuid": "ladc74e2-96ac-4761-b9e6-2d93e02d9244",
   "states": [
     true,
     false,
      false
   ]
 "hold_time": 0,
 "reset output after hold time expired": false,
 "signal color": null
}
```

1.3.35 ColorMatchingResult

After each sampling period the retrieved color value is compared to the stored detectables (color positions). Detectables are ignored, if the tolerance shape of their corresponding matcher does not encompass the current sample. Finally the closes suitable detectable is selected as the winner of the color matching operation. The corresponding matcher determines the state of the sensor for the duration of the next sampling period.

Properties

matcher any of <u>UUID</u> (string) or null, optional, Depre- cated	Deprecated: use "chosen_matcher_id" instead	
chosen_matcher_id any of <u>UUID</u> (string) or null, required	unique identifier of the selected matcher	
distances Array of any of number or null, required	Distance between the sample's color position and the selected matcher's closest color position along the three axes of the color space. The array contains three 'null' values, if no suitable matcher was found for the current color sample.	
output_pattern CurrentSwitchingOut- putsState, required	Currently active state of the Switching Outputs. Beware that this may deviate from the specified output states of the current best matcher, since settings like <i>triggered input</i> or <i>hold time</i> influence update process for the Switching Outputs.	
	states Array of any of boolean or null, required	List of True/False values describing the current states of the Switching Outputs

Examples

```
No Match
```

```
{
  "chosen_matcher_id": null,
  "distances": [
    null,
    null,
    null
],
  "output_pattern": {
    "states": [
        true,
        true
    ]
  }
}
```

Suitable Match

```
"chosen_matcher_id": "4575656f-efe4-4a7d-862c-9660c15cdf4e",
"distances": [
   1.4,
   0.3,
   null
],
"output_pattern": {
   "states": [
       true,
       false,
       true
```

```
}
}
```

1.3.36 ColorRepresentations

Pre-calculcated visual representations of a color suitable for rendering

Properties

RGB	RGB color array representing the axes r, g, and b in that or-
Array of number, minimum items: 3, maximum items: 3, re-	der. Values are floats between 0 and 1.
quired	

Examples

```
{
  "RGB": [
    0.3197475,
    0.754686,
    0.216748
]
}
```

1.3.37 Colorspace

A colorspace describes the numeric conversion of colors under certain circumstances. Different standardized colorspaces are suitable for different detection tasks.

Properties

. opolitioo		
name String, required		
space_id ColorspaceID, required	Unique name of a colorspace	
Array of ColorspaceAxis, minimum items: 3, maxi- mum items: 3, required	ColorspaceAxis[]	
•	id String, required	Unique name
	label String, required	Human-readable name
	minimum Number, required	lowest expected value of a color along this axis under usual circumstances
	maximum Number, required	highest expected value of a color along this axis under usual circumstances

Examples

```
{
    "id": "b",
    "label": "b*",
    "minimum": -200,
    "maximum": 200
}
]
```

1.3.38 ColorspaceAxis

Properties

id	Unique name
String, required	
label	Human-readable name
String, required	
minimum	lowest expected value of a color along this axis under usual circumstances
Number, required	
maximum	highest expected value of a color along this axis under usual circumstances
Number, required	

1.3.39 ColorspaceID

Unique name of a colorspace.

Type Information

ColorspaceID (string)

1.3.40 ColorspaceToleranceMap

Specify the usage of the axes of each colorspace for non-trivial tolerance shapes. See "color-space_tolerance_maps" below "/capabilities" for more details.

Properties

colorspace id	Unique name of a colorspace
ColorspaceID (string), required	oriique name et a eciciopace
tolerance_shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.
limits_axes_map Object, required	limits_axes_map
	half_height Array of string, optional
	half_edges Array of string, optional
	radius Array of string, optional

Examples

```
"colorspace_id": "Lab",
  "tolerance_shape": "cylindrical",
  "limits_axes_map": {
    "half_height": [
       "L"
    ],
    "radius": [
       "a",
       "b"
    ]
}
```

}

1.3.41 ColorTolerance

Specification of a geometric shape and its dimensions in the current colorspaces.

Type Information

any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance

Examples

```
Infinite
  "shape": "infinite",
  "limits": {}
Sphere
  "shape": "sphere",
  "limits": {
    "radius": 2
Cylidner
  "shape": "cylinder",
  "limits": {
    "radius": 2,
    "half height": 4
}
Box
  "shape": "box",
  "limits": {
    "half edges": [
      4,
      2,
    ]
```

1.3.42 CompensationSettings

The compensation settings of a Detection Profile describe the configuration of internal sensor components related to the stabilization and compensation algorithms.

These values can be determined by issuing a POST request against /api/sensor/detection-profiles/current/autogain. The result is a suitable set of compensation settings for this sensor under the current circumstances.

The content of this data object is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).

1.3.43 CorrectedColor

Representation of a color in the colorspace XYZ.

Properties

I I	
values	Location in a colorspace
Array of number, minimum items: 3, maximum items: 3, re-	
quired	

1.3.44 CurrentDetectionProfileID

The sensor can store multiple Detection Profiles, but it can only apply one at a time. The field <code>current_profile_id</code> contains the UUID of the Detection Profile that is currently used by the sensor for its operation. It allows to use the shortcut API endpoint <code>/api/sensor/detection-profiles/current</code> instead of specifying a Detection Profile by its UUID.

Type Information

CurrentDetectionProfileID (string), pattern: ^[a-f0-9-]+\$

Examples

a014e415-0fec-4734-ac3f-30da0a5f3899

1.3.45 CurrentSwitchingOutputsState

Currently active state of the Switching Outputs. Beware that this may deviate from the specified output states of the current best matcher, since settings like *triggered input* or *hold time* influence update process for the Switching Outputs.

Properties

states	List of True/False values describing the current states of the
Array of any of boolean or null, required	Switching Outputs

1.3.46 CylinderColorTolerance

Properties

limits	limits
object, required	
	radius
	number, required
	half_height
	number, required
shape	Name of the geometrical shape of the tolerance. The sup-
ToleranceShapeName (string), required	ported tolerance shapes can be retrieved via /api/sen-
	sor/capabilities.

1.3.47 Standardwertepaare

Properties

uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, readonly	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
object_type	Name of the object the default is meant for
string, required , read-only	
key	name of the object's property
string, required, read-only	
value	Actual default value for the object's property
any, required	

```
Matcher: Tolerance
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object_type": "matcher",
  "key": "tolerance",
  "value": {
    "shape": "sphere",
    "limits": {
      "radius": 4
  }
}
Matcher: Hold Time
  "uuid": "55b35901-1ea6-4b3d-864a-60af15a9b0c5",
  "object_type": "matcher",
  "key": "hold time",
  "value": 0
}
Matcher: reset output after Hod Time expiry
  "uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset output_after_hold_time_expired",
  "value": false
}
Autogain: number of samples used for averaging
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object type": "autogain",
  "key": "averages",
  "value": 0
```

1.3.48 DetectionProfile

A Detection Profile contains a complete set of sensor settings for a given detection task.

Multiple profiles can be stored in order to switch easily between different detection tasks or for the incremental development of a refined profile.

Some attributes of a Detection Profile expose internal details of the sensor that should be determined indirectly via other means. These attributes are described only superficially, since they should be handled as is without changing their value or structure.

Properties

uuid <u>UUID</u> (string), pattern: ^ [a- f0-9-]+\$, required, read- only	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8	
alias Alias (integer), required,	A numerical value that can be used to address an item in a col-	
read-only	lection. If an alias is specified	

	T		T
	alongside an uuid attribute, that		
	alias can be used as an alterna-		
	tive to address the item in URLs		
	and other protocols like Modbus		
	or serial interfaces.		
name	Human-readable name of the		
String, required	Detection Profile		
colorspace	A colorspace describes the nu-		
Colorspace, required	meric conversion of colors under certain circumstances. Different		
	standardized colorspaces are suitable for different detection		
	tasks.		
	colorspace		
	name		
	String, required		
	space_id	Unique name of a colorspace	
	ColorspaceID, required	Offique flame of a colorspace	
	axes	ColorspaceAxis[]	
	Array of ColorspaceAxis, mini-	ColorspaceAxis[]	
	mum items: 3, maximum items:		
	3, required		
	o, required	id	Unique name
		String, required	- Inque nume
		label	Human-readable
		String, required	name
		minimum	lowest expected
		Number, required	value of a color along
			this axis under usual
			circumstances
		maximum	highest expected
		Number, required	value of a color along
		, ,	this axis under usual
			circumstances
non_matching_output	This state of the Switching Out-		
WantedSwitch-	puts is applied, if the currently		
ingOutputsState, required	sample color does not belong to		
	any of the stored <i>Matchers</i> .		
	non_matching_output		
	uuid	unique identifier (UUID) as de-	
	UUID (string), pattern: ^ [a-f0-	fined by RFC 4122, ITU-T Rec.	
	9-]+\$, required , read-only	X.667, and ISO/IEC 9834-8	
	states	List of True/False/Null values	
	Array of any of boolean or null,	describing the wanted states of	
	required	the Switching Outputs	
non_matching_hold_time	Minimum duration (in seconds)		
HoldTime (number), maxi-	of the non_matching_output state		
mum 3153600000, required	being applied to the Switching		
	Outputs of the sensor. This pro-		
	longing of a potential non match-		
	ing event may be useful, if the		
	processing period of a con-		
	nected actor exceeds the sam-		
	pling period of the sensor.		
compensation_settings	The compensation settings of a		
CompensationSettings, re-	Detection Profile describe the		
quired	configuration of internal sensor		
	components related to the stabi-		
	lization and compensation algo-		
	rithms.		
	These values can be determined		
	by issuing a POST request		
	against /api/sensor/detection-		
	profiles/current/autogain. The re-		
	sult is a suitable set of compen-		
		İ	
	sation settings for this sensor un-		
	der the current circumstances.		
	der the current circumstances. The content of this data object is		
	der the current circumstances. The content of this data object is not meant to be manipulated by		
	der the current circumstances. The content of this data object is		

	and applied without modification		
	or introspection).		
	compensation settings		
sampling_settings	Sampling Settings describe all		
SamplingSettings, required	details of the sampling process.		
SamplingSettings, required	Its attributes may be queried and		
	inspected (e.g. in order to re-		
	trieve the current sample rate).		
	Most values stored within the		
	Sampling Settings should not be		
	modified directly. The related API		
	endpoint /api/sensor/de-		
	tection-profiles/cur-		
	rent/autogain should be		
	used instead.		
	The only modifiable attribute		
	within the Sampling Settings is		
	the averages value. It is safe to		
	change it, even though the de-		
	fault values calculated during an		
	autogain operation should be		
	optimal for most detection tasks.		
	sampling_settings		
	led intensity	relative intensity of the internal	
	Number, minimum: 0, maximum:	emitter during the light phase	
	1, required	control during the light phase	
		The base sample rate deter-	
	base_sample_rate	· •	
	SampleRate (number), mini-	mines the duration of a sam-	
	mum: 0.01, required	pling period.	
		After each sampling period,	
		the gathered data is processed	
		and a new detection result is	
		calculated (e.g. the most suita-	
		ble <i>Matcher</i> for the given sam-	
		ple). This may affect the state	
		of the Switching Outputs or	
		trigger configured actions.	
		Thus the base sample rate de-	
		fines the maximum rate of	
		changes for the Switching Out-	
		puts.	
		See also the effective sample	
		rate.	
	offective comple vete		
	effective_sample_rate	The effective sample rate is the	
	SampleRate (number), mini-	numeric product of the base	
	mum: 0.01, required	sample rate and the number of	
		averages.	
		It determines the minimum du-	
		ration that a target needs to be	
		sampled in order to determine	
		its visual appearance correctly.	
		With the default value of aver-	
		age set to one, this value is	
		equal to the base sample rate.	
	minimum_wanted_sample_rate	This informational value repre-	
	SampleRate (number), mini-	sents the sample rate that was	
	mum: 0.01, required	requested during the most re-	
		cent <i>Autogain</i> operation. The	
		effective sample rate may devi-	
		ate from the wanted sample	
		rate, if the requested sample	
		rate was not achievable due to	
		limitations of the sensor (e.g.	
		exceeding the supported sam-	
		ple rate) or due to the environ-	
		' /	
		ment (e.g. not enough light,	
		thus a slower amplification with	
	i	higher gain was necessary).	
		1 61 1611	
	sample_light_phase	defines if the sensor should	
	sample_light_phase Boolean, required	defines if the sensor should periodically activate the inter- nal emitter for sampling	

	sample_dark_phase Boolean, required averages AverageSampleCount (integer),	defines if the sensor should periodically deactivate the in- ternal emitter for sampling Number of previous samples to be averaged for every sam-	
	minimum: 1, required	pling result. A rolling averaging algorithm is applied to the samples.	
	amplification AmplificationLevel (integer), required	The amplification level specifies the internal configuration of an amplifier. This value is not meant to be manipulated by regular users. It should be handled as is (stored, transmitted and applied without modification or introspection).	
white_reference Array of number, required	The White Reference attribute is used for indicating a custom color balancing. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).		
normalization_constant Array of number, required	Normalization constants are related to the White Reference. Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).		

```
"name": "#0",
"uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
"alias": 1,
"non_matching_hold_time": 0,
"colorspace": {
  "name": "L*a*b*",
  "axes": [
    {
      "id": "L",
      "label": "L*",
      "minimum": 0,
      "maximum": 100
    } ,
      "id": "a",
      "label": "a*",
      "minimum": -500,
      "maximum": 500
    } ,
      "id": "b",
      "label": "b*",
      "minimum": -200,
      "maximum": 200
    }
  ],
  "space id": "Lab"
},
```

```
"compensation settings": {
  "monitor integration": {
    "control": 0.32499998807907104,
    "references": [
      0.7283520102500916,
      0.7442666888237,
      0.7066696286201477
    ]
  },
  "use calibration_samples": true
"normalization_constant": [
 237.4935277662995,
  242.62655153828055,
  587.8264132734112
"white reference": [
 95.047,
  100,
 108.883
],
"non matching output": {
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
   true,
   true,
   true,
   true,
    true,
    true,
   true,
   true
 ]
} ,
"sampling_settings": {
 "led_intensity": 1,
  "amplification": 1,
  "sample light phase": true,
  "minimum wanted sample rate": 1000,
  "averages": 1,
  "base sample rate": 1000,
  "sample_dark_phase": true,
  "effective sample rate": 1000
}
```

1.3.49 DeviceInformation

Properties

}

id	Serial Number
DeviceSerialNumber, required	
model_name	human-readable name of the device model
string, required	
model_key	unique id of the device model
string, required	
variant	indicates a special series of a model
any of string or null, required	
vendor_key	Unique key identifying the organization distributing this device
DeviceVendorKey, required	
vendor_name	Name of vendor of this device
DeviceVendorName, required	
device id	Deprecated: use "id" instead.
DeviceSerialNumber, optional, Deprecated	

model	Deprecated: use "model_name" instead.
string, optional, Deprecated	
vendor	Deprecated: use "vendor_name" instead.
DeviceVendorName, optional, Deprecated	_

```
{
  "vendor_name": "Micro-Epsilon Eltrotec GmbH",
  "vendor_key": "eltrotec",
  "variant": null,
  "model_key": "me_cfo_100",
  "model_name": "CFO100",
  "id": "7454228060"
}
```

1.3.50 DeviceSerialNumber

Serial Number.

Type Information

DeviceSerialNumber (string)

1.3.51 DeviceVendorKey

Unique key identifying the organization distributing this device.

Type Information

DeviceVendorKey (string)

Examples

acme

1.3.52 DeviceVendorName

Name of vendor of this device.

Type Information

<u>DeviceVendorName</u> (string)

Examples

Acme Corporation

1.3.53 Error

List of error indicators that are both machine-parseable and human-readable

Properties

1-0-0-1110-0	
code	machine-readable unique error code
string, optional	
mapping	a reference to the parameter that caused the error
string, optional	
message human-readable error description	
string, optional	•

1.3.54 FirmwareBuildld

Unique ID of the currently running firmware image.

Type Information

```
FirmwareBuildId (string), pattern: ^[a-f0-9-]+$
```

Examples

d985c28e03a4eb39132c02affeb29e71

1.3.55 FirmwareImageFile

Type Information

FirmwareImageFile (file)

1.3.56 FirmwareImageSize

Size of the firmware image in bytes.

Type Information

FirmwareImageSize (integer), minimum: 1, maximum: 1073741824

1.3.57 FirmwareImageUpload

A fully or partially uploaded firmware image to be used for upgrading the firmware.

Properties

uuid	Unique ID of a firmware upload
<u>UUID</u> (string), pattern: ^ [a-	
f0-9-]+\$, required , read-	
only	
build_id	unique ID of the currently running firmware image
HashDigest (string), pattern:	
^[a-f0-9]+\$, required	
status	Current status of the firmware upload
string, one of [incom-	incomplete
plete, complete, in-	the number of bytes received is lower than the number of bytes that have been announced
valid_signature, pro-	complete
cessing_failure, mal-	the firmware upload is complete and the new firmware can be applied invalid_signature
formed_content, de-	the firmware checksum didn't match the expected value processing_failure
<pre>vice_mismatch], re- quired</pre>	an internal undefined error occurred while processing the firmware malformed_content
quired	the uploaded firmware image uses an unexpected format or misses essential information de-
	vice_mismatch
unlanded size	the firmware image can not be applied to this device
uploaded_size	number of uploaded bytes
integer, minimum: 0, re-	
quired	avacated total number of hytee for the firmware image
expected_size integer, minimum: 1, re-	expected total number of bytes for the firmware image
quired	
max chunk size	maximum size for a data chunk uploaded to the device
integer, minimum: 1, re-	maximum size for a data churik uproaded to the device
auired	
quired	<u> </u>

Examples

```
"uuid": "78b40d5e-e82c-45a9-8842-9481f889f790",
"build_id": "e943ce84dbe474bc4d163b44c90070b105fd66bb",
"expected_size": 335544320,
"max_chunk_size": 1048576,
"status": "incomplete",
"uploaded size": 24117248
```

}

1.3.58 FirmwareInformation

Information describing a firmware version.

Properties

1 100011100	
id	unique ID of the currently running firmware image
FirmwareBuildId (string), pat-	
tern: ^ [a-f0-9]+\$, required	
channel	Describes the kind of a publication
ReleaseChannel (string), one of [stable, feature], default:	Releases on the stable channel are generally considered well-tested and are recommended for use in production.
stable, required	Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.
created on	time this firmware build was created
Timestamp (string), required	
name	human-readable name of this release
string, required	
notes	Release notes formatted as markdown
string, required	
version	version of a firmware
FirmwareVersion (string), re-	
quired	
works_with	compatible device models (see model key in /api/device)
Array of string, required	,

1.3.59 FirmwareRecoveryInformation

Type Information

FirmwareRecoveryInformation (string)

Examples

```
"created on": "2018-02-13T05:40:39+01:00",
  "name": "CFO",
  "id": "4fab356b5014b5cc82efc4a81bfefbfcdc2d9165",
  "version": "1.3.16",
  "channel": "stable",
  "works with": [
    "me cfo 100",
    "me cfo 200"
  ],
 "notes": "# Release 1.3.16 (2018-02-13 - CFO) \n\n## Veröffentlichungshin-
weise\n\nWartungsrelease für CFO-Sensoren.\n\n\## Änderun-
gen\nkeine\n\n## Fehlerkorrekturen\n* Hochladen von Konfigurationsdateien
mit mehr als 70 Farben ermöglicht\n* Announcierung des korrekten Hostnamen
via avahi/zeroconf\n* SSDP: Kommunikation via IPv6-Link-Local-Adresse er-
möglicht\n* SSDP: auch die Auto-Konfigurations-IP (via RFC3927) unter
\"CurrentAddresses\" announcieren\n* SSDP: nach Konfigurationsänderungen an
neue IP-Addressen binden"
```

1.3.60 FirmwareRunningInformation

Information describing the currently running firmware.

Properties

	•	
ſ	build_id	unique ID of the currently running firmware image
	FirmwareBuildId (string), pattern: ^[a-f0-9]+\$, re-	
	quired	

source_url	Absolute base URL of a firmware repository delivering firmware
any of string or null, optional	images suitable for this device
version	version of a firmware
FirmwareVersion (string), required	

```
"build_id": "4fab356b5014b5cc82efc4a81bfefbfcdc2d9165",
   "source_url": null,
   "version": "1.3.16"
}
```

1.3.61 FirmwareSettings

Settings related to the device's firmware and upgrades.

Properties

channel	Describes the kind of a publication	
ReleaseChannel (string), one	Releases on the stable channel are generally considered well-tested and are recom-	
of [stable, feature], default:	mended for use in production.	
stable, required	Releases on the feature add new features but haven't been tested as much as a stable	
	release. Feature releases can but should only be used in production with careful consider-	
	ation.	

Examples

```
{
   "release_channel": "stable"
}
```

1.3.62 FirmwareVersion

Version of a firmware

Type Information

FirmwareVersion (string)

Examples

```
v2.3.42
```

1.3.63 HashDigest

Unique identifier (hexadecimal digest string).

Type Information

```
HashDigest (string), pattern: ^[a-f0-9]+$
```

Examples

```
d985c28e03a4eb39132c02affeb29e71
```

1.3.64 HoldTime

Minimum duration (in seconds) of a matcher's output setup being applied after detection.

Type Information

HoldTime (number), minimum: 0, maximum: 3153600000

1.3.65 Hostname

Human-readable name identifying the device in the network.

Type Information

Hostname (string), pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-]*[a-zA-Z0-9])?\.

1.3.66 InfiniteColorTolerance

Properties

limits	limits	
Object, required		
shape	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via	
ToleranceShapeName	/api/sensor/capabilities.	
(string), required		

1.3.67 InputsState

The state of all inputs during a given period is specified by a list of possible events combined with a boolean value indicating, if the given event occurred within the period.

Properties

//	The boolean value indicates whether the named input event occurred during the last period.
boolean, required	

1.3.68 InterfaceRS232

Properties

protocol any of <u>SerialModbusSettings</u> or <u>SerialElizaSettings</u> , re-	SerialModbusSettings
quired	
	type
	string, one of [none, eliza, modbus], default: eliza, re-
	quired
	slave_id
	any of number or null, required
	frame_format
	string, one of [rtu, ascii], default: rtu, required
	SerialElizaSettings
	type
	string, one of [none, eliza, modbus], default: eliza, required
baud rate	quireu
number, one of [9600, 19200, 115200], required	

1.3.69 InterfaceUSB

Properties

protocol any of SerialModbusSettings or SerialElizaSettings, re-	SerialModbusSettings
quired	
	type
	<pre>string, one of [none, eliza, modbus], default: eliza, re- quired</pre>
	slave_id
	any of number or null, required
	frame_format
	string, one of [rtu, ascii], default: rtu, required
	SerialElizaSettings
	type string, one of [none, eliza, modbus], default: eliza, required
baud rate number, one of [9600, 19200, 115200], required	

1.3.70 KeypadEvent

A keypad event represents a single press or release event of a button at a specific time.

Properties

source	The usual source of events is inputs.	
string, required	·	
name	Name of a keypad input (button) that may trigger events.	
KeypadEventInput (string), required		
event	Input peripherals can trigger different events.	
KeypadEventName (string), required		
timestamp The timestamp is given in milliseconds and should be monotonic increa		
integer, minimum: 0, required		

Examples

Intensity Button pressed

```
"source": "inputs",
   "name": "intensity",
   "event": "down",
   "timestamp": 6403500
}
```

Intensity Button released

```
{
  "source": "inputs",
  "name": "intensity",
  "event": "up",
  "timestamp": 6405800
}
```

1.3.71 KeypadEventInput

Name of a keypad input (button) that may trigger events.

Type Information

KeypadEventInput (string)

1.3.72 KeypadEventName

Input peripherals can trigger different events.

Type Information

KeypadEventName (string)

1.3.73 KeypadIndicator

The keypad features multiple LEDs as visual indicators.

The indicators may be lit, blinking or off.

Properties

name	Name of the indicator	
string, required		

type string, required	The type describes the possible modes of visualization for this indica-	
	tor.	
animation	The visual state of each indicator is de-	
Array of object, required	scribed by an infinite loop of animation	
	steps.	
	object []	
	enabled	Visual status of the indicator (on or off)
	boolean, required	
	color	Name or description of a color
	string, optional	·
	duration	Duration (in seconds) of this part of the
	number, required	looping animation.

1.3.74 KeypadInformation

Describe the current state of the keypad as well as access to visualization data.

Properties

locked boolean, required	Boolean flag indicating the state of the key lock (true -> locked, false -> unlocked). All keypad inputs are ignored while the lock is active.	
clear_matcher_before_teach boolean, required	The boolean flag controls whether multiple detectables can be stored for a matcher via keypad-based teach operations. A value of true implies, that a teach operation always removes all existing detectables from the currently selected matcher before adding the new detectable. With a value of false previously existing detectables are not deleted before a new one is added.	
visualization_url any of string or null, optional, read- only	The visualization resource location can be used for providing a virtual keypad interface. Its URL may start with a scheme (e.g. http or https) for a full URL including hostname or it may start with a slash, indicating a path provided by the device itself. This attribute cannot be modified.	

Examples

```
{
  "locked": true,
  "clear_matcher_before_teach": false,
  "visualization_url": "/media/keypad-image.svg"
}
```

1.3.75 KeypadInputButton

The keypad contains several inputs (buttons) that may generate events.

Properties

name	Name of a keypad input (button) that may trigger events.	

KeypadEventInput (string), required		
capabilities Array of object, required	object[]	
Tiray or object, required	name KeypadEventName (string), required	Input peripherals can trigger different events.
	url string, required	The event can be triggered externally by submitting a POST request against this resource.

1.3.76 LoginInformation

Describes the currently active login provided by the user agent.

Properties

logged_in_user	The currently logged in user.	
any of <u>User</u> or null, re-	Is null if the credentials	
quired	didn't match any known user	
	or have expired.	
	User	
	name	unique name identifying an account
	string, pattern: ^ [\w-] +\$, re-	
	quired, read-only	
	password	Password assigned to this account (only writable; never re-
	string, optional	turned in responses). Either a password or a password hash needs to be supplied when creating a new user or changing a password.
	password_hash	Password hash assigned to this account. Either a password or
	HashDigest (string), pattern:	a password hash needs to be supplied when creating a new
	^[a-f0-9]+\$, optional	user or changing a password.
	roles	The roles assigned to an account define its set of permissions.
	Array of string, optional	·
session_timeout	Number of seconds this ses-	
any of number or null,	sion has left before expiring.	
required	Is null if the provided cre-	
	dentials could not be matched	
	to any active sessions, if the	
	session expired or if the sup-	
	plied authentication mecha-	
	nism does not support ses-	
	sions (e.g. HTTP Authentica-	
	tion).	

1.3.77 MacAddress

Unique hardware address of a network interface.

Type Information

```
MacAddress (string), pattern: ^([a-f0-9]{2}:){5}[a-f0-9]{2}$
```

00:01:2e:7a:dc:23

1.3.78 NetworkAddressConfigurationIPv4

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	

1.3.79 NetworkAddressConfigurationIPv4DHCP

The Dynamic Host Configuration Protocol requires a router distributing leases on request.

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	

1.3.80 NetworkAddressConfigurationIPv4Static

Static address configuration does not depend on network infrastructure.

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	
address	IPv4 network address in CIDR notation
NetworkInterfaceAddressIPv4 (string), required	
gateway	default gateway for outgoing traffic
NetworkAddressIPv4 (string), optional	

1.3.81 NetworkAddressConfigurationIPv6

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	

1.3.82 NetworkAddressConfigurationIPv6Auto

Stateless address autoconfiguration (SLAAC) solely relies on the Neighbourhood Discovery Protocol. SLAAC is only available for IPv6.

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	

1.3.83 NetworkAddressConfigurationIPv6DHCP

The Dynamic Host Configuration Protocol requires a router distributing leases on request.

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	

1.3.84 NetworkAddressConfigurationIPv6Static

Static address configuration does not depend on network infrastructure.

Properties

method	Configuration method used for the address.
string, one of [static, dhcp], required	
address	IPv6 network address in CIDR notation
NetworkInterfaceAddressIPv6 (string), required	

gateway	default gateway for outgoing traffic
NetworkAddressIPv4 (string), optional	

1.3.85 NetworkAddressIPv4

Type Information

NetworkAddressIPv4 (string)

Examples

192.168.1.100

1.3.86 NetworkAddressIPv6

Type Information

NetworkAddressIPv6 (string)

Examples

fda0:576b:c643:100:40f:10ff:fe02:e6f

1.3.87 NetworkInterfaceAddressConfigurationInformation

Properties

ipv4 NetworkInterfaceAddressFamilyInfor-	IPv4 Network address configuration		
mationIPv4, required			
	ipv4	111111111111111111111111111111111111111	
	address_configurations Array of any of Network- AddressConfiguration- IPv4Static or Network- AddressConfiguration- IPv4DHCP, optional	NetworkAddressConfiguratio- nIPv4Static[]	
		method string, one of [static, dhcp], required	Configura- tion method used for the address.
		Address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR nota- tion
		gateway NetworkAddressIPv4 (string), optional	default gate- way for out- going traffic
		NetworkAddressConfiguratio- nIPv4DHCP[]	
		method string, one of [static, dhcp], required	Configura- tion method used for the address.
	current_addresses Array of WrappedNetworkInter-faceAddressIPv4, required	WrappedNetworkInter- faceAddressIPv4[]	
		address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR nota- tion
ipv6 NetworkInterfaceAddressFamilyInfor- mationIPv6, required	IPv6 Network address configuration		
	ipv6		

address_configurations Array of any of Network- AddressConfiguration- IPv6Static, Network- AddressConfiguration- IPv6DHCP or Network- AddressConfigurationIPv6Auto, optional	NetworkAddressConfiguratio- nlPv6Static[]	
	method string, one of [static, dhcp], required	Configura- tion method used for the address.
	address NetworkInterfaceAddressIPv6 (string), required	IPv6 network address in CIDR nota- tion
	gateway NetworkAddressIPv6 (string), optional	default gate- way for out- going traffic
	NetworkAddressConfiguratio- nIPv6DHCP[]	
	method string, one of [static, dhcp, auto], required	Configura- tion method used for the address.
	NetworkAddressConfiguratio- nIPv6Auto[]	
	method string, one of [static, dhcp, auto], required	Configuration method used for the address.
current_addresses Array of WrappedNetworkInter-faceAddressIPv6, required	WrappedNetworkInter- faceAddressIPv6[]	
	address NetworkInterfaceAddressIPv6 (string), required	IPv6 network address in CIDR nota- tion

1.3.88 NetworkInterfaceAddressConfigurationIPv4

Type Information

Array of any of NetworkAddressConfigurationIPv4Static or NetworkAddressConfigurationIPv4DHCP

1.3.89 NetworkInterfaceAddressConfigurationIPv6

Type Information

 $\label{eq:configuration} \mbox{Array of any of $\underline{\mbox{NetworkAddressConfigurationIPv6DHCP}}$ or $\underline{\mbox{NetworkAddressConfigurationIPv6Auto}}$$

1.3.90 NetworkInterfaceAddressConfigurationState

Configurable network address configuration of a network interface

Properties

ipv4	IPv4 Network address configura-	
NetworkInterfaceAddressFa-	tion	
milyStateIPv4, optional		
	ipv4	

	address_configurations Array of any of Network- AddressConfigurationIPv4Static or NetworkAddressConfigura- tionIPv4DHCP, optional	NetworkAddressConfiguratio- nlPv4Static[] method string, one of [static, dhcp],	Configuration method used
		required	for the ad- dress.
		address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR nota- tion
		gateway <u>NetworkAddressIPv4</u> (string), op- tional	default gate- way for out- going traffic
		NetworkAddressConfiguratio- nlPv4DHCP[]	
		<pre>method string, one of [static, dhcp], required</pre>	Configuration method used for the address.
ipv6 NetworkInterfaceAddressFa- milyStateIPv6, optional	IPv6 Network address configuration		
	ipv6		
	address_configurations Array of any of Network- AddressConfigurationIPv6Static, NetworkAddressConfiguration- IPv6DHCP or Network- AddressConfigurationIPv6Auto, optional	NetworkAddressConfiguratio- nIPv6Static[]	
		method string, one of [static, dhcp], required	Configuration method used for the address.
		address NetworkInterfaceAddressIPv6 (string), required	IPv6 network address in CIDR nota- tion
		gateway NetworkAddressIPv6 (string), optional	default gate- way for out- going traffic
		NetworkAddressConfiguratio- nIPv6DHCP[]	
		method string, one of [static, dhcp, auto], re- quired	Configuration method used for the address.
		NetworkAddressConfiguratio- nIPv6Auto[]	
		method string, one of [static, dhcp, auto], required	Configuration method used for the address.

Remove all IPv6 address configurations

```
{
  "ipv6": {
     "address_configurations": []
   }
}
```

Replace existing IPv4 configuration with DHCP

Set static and dynamic IPv4 configuration

1.3.91 NetworkInterfaceAddressesIPv4

Properties

current_addresses	WrappedNetworkInterfaceAddressIPv4[]	
Array of WrappedNetworkInter-		
faceAddressIPv4, required		
-	address	IPv4 network address in CIDR nota-
	NetworkInterfaceAddressIPv4 (string), re-	tion
	quired	

1.3.92 NetworkInterfaceAddressesIPv6

Properties

current_addresses	WrappedNetworkInterfaceAddressIPv6[]	
Array of WrappedNetworkInter-		
faceAddressIPv6, required		
	address	IPv6 network address in CIDR nota-
	NetworkInterfaceAddressIPv6 (string), re-	tion
	quired	

1.3.93 NetworkInterfaceAddressFamilyInformationIPv4

IPv4 Network address configuration

Properties

i ropertico		
address_configurations	NetworkAddressConfigurationIPv4Sta-	
Array of any of Network-	tic[]	
AddressConfigurationIPv4Static or		
NetworkAddressConfiguration-		
IPv4DHCP, optional		

	method string, one of [static, dhcp], re-	Configuration method used for the address.
	quired address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR notation
	gateway NetworkAddressIPv4 (string), optional	default gateway for outgoing traffic
	NetworkAddressConfiguratio- nIPv4DHCP[]	
	method string, one of [static, dhcp], re- quired	Configuration method used for the address.
current_addresses Array of WrappedNetworkInter-faceAddressIPv4, required	WrappedNetworkInterfaceAddres-sIPv4[]	
	address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR notation

1.3.94 NetworkInterfaceAddressFamilyInformationIPv6

IPv6 Network address configuration.

Properties

address_configurations	NetworkAddressConfigurationIPv6Sta-	
Array of any of Network- AddressConfigurationIPv6Static,	tic[]	
NetworkAddressConfiguration-		
IPv6DHCP or NetworkAddressCon-		
figurationIPv6Auto, optional		
,		
	method	Configuration method used for the ad-
	<pre>string, one of [static, dhcp], re- quired</pre>	dress.
	address	IPv6 network address in CIDR notation
	NetworkInterfaceAddressIPv6 (string),	
	required	
	gateway	default gateway for outgoing traffic
	NetworkAddressIPv6 (string), optional	
	NetworkAddressConfiguratio- nIPv6DHCP[]	
	method	Configuration method used for the ad-
	string, one of	dress.
	[static, dhcp, auto], required	
	NetworkAddressConfiguratio-	
	nIPv6Auto[]	
	method	Configuration method used for the ad-
	string, one of	dress.
	[static, dhcp, auto], required	

current_addresses Array of WrappedNetworkInter- faceAddressIPv6, required	WrappedNetworkInterfaceAddres-sIPv6[]	
	address NetworkInterfaceAddressIPv6 (string), required	IPv6 network address in CIDR notation

1.3.95 NetworkInterfaceAddressFamilyStateIPv4

IPv4 Network address configuration.

Properties

address_configurations Array of any of Network- AddressConfigurationIPv4Static or NetworkAddressConfiguration- IPv4DHCP, optional	NetworkAddressConfigurationIPv4Static[]	
	method string, one of [static, dhcp], re- quired	Configuration method used for the address.
	address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR notation
	gateway NetworkAddressIPv4 (string), optional	Standard Gateway für abgehenden Ver- kehr
	NetworkAddressConfiguratio- nIPv4DHCP[]	
	method string, one of [static, dhcp], required	Configuration method used for the address.

1.3.96 NetworkInterfaceAddressFamilyStateIPv6

IPv6 Network address configuration.

Properties

address_configurations	NetworkAddressConfigurationIPv6Sta-	
Array of any of Network-	tic[]	
AddressConfigurationIPv6Static,		

NetworkAddressConfiguration-		
IPv6DHCP or NetworkAddressCon-		
figurationIPv6Auto, optional		
	method	Configuration mathed wood for the ad
	***************************************	Configuration method used for the ad-
	string, one of [static, dhcp], re-	dress.
	quired	
	address	IPv6 network address in CIDR notation
	NetworkInterfaceAddressIPv6 (string),	
	required	
	gateway	default gateway for outgoing traffic
	NetworkAddressIPv6 (string), optional	
	NetworkAddressConfiguratio-	
	nlPv6DHCP[]	
	method	Configuration method used for the ad-
	string, one of	dress.
	[static, dhcp, auto], required	
	NetworkAddressConfiguratio-	
	nlPv6Auto[]	
	method	Configuration method used for the ad-
	string, one of	dress.
	[static, dhcp, auto], required	

1.3.97 NetworkInterfaceAddressIPv4

IPv4 network address in CIDR notation.

Type Information

NetworkInterfaceAddressIPv4 (string)

Examples

192.168.1.100/24

1.3.98 NetworkInterfaceAddressIPv6

IPv6 network address in CIDR notation.

Type Information

NetworkInterfaceAddressIPv6 (string)

Examples

fda0:576b:c643:100:40f:10ff:fe02:e6f/64

1.3.99 NetworkInterfaceInformation

Description of the currently active addresses of the interface and its configuration.

Properties

iface NetworkInterfaceName (string),	unique name describing a net- work interface	
pattern: ^[a-z0-9-]+\$, re- quired, read-only		
hardware_address	unique hardware address of a	
<u>MacAddress</u> (string), pattern: ^([a-f0-9]{2}:){5}[a-f0-	network interface	
9] {2}\$, required, read-only		
has_link	current physical connection sta-	
boolean, required , read-only	tus (whether a cable is plugged in or not)	
ipv4	IPv4 Network address configura-	
NetworkInterfaceAddressFa- milyStateIPv4, optional	tion	
	ipv4	

	address_configurations Array of any of Network- AddressConfigurationIPv4Static or NetworkAddressConfigura- tionIPv4DHCP, optional	NetworkAddressConfiguratio- nIPv4Static[]	
		<pre>method string, one of [static, dhcp], required</pre>	Configuration method used for the address.
		address NetworkInterfaceAddressIPv4 (string), required	IPv4 network address in CIDR nota- tion
		gateway <u>NetworkAddressIPv4</u> (string), optional	default gate- way for out- going traffic
		NetworkAddressConfiguratio- nIPv4DHCP[]	
		method string, one of [static, dhcp], required	Configuration method used for the ad- dress.
ipv6 NetworkInterfaceAddressFa- milyStateIPv6, optional	IPv6 Network address configuration		
	ipv6		
	address_configurations Array of any of Network- AddressConfigurationIPv6Static, NetworkAddressConfiguration- IPv6DHCP or Network- AddressConfigurationIPv6Auto, optional	NetworkAddressConfiguratio- nIPv6Static[]	
		method string, one of [static, dhcp], required	Configuration method used for the address.
		Address NetworkInterfaceAddressIPv6 (string), required	IPv6 network address in CIDR nota- tion
		gateway <u>NetworkAddressIPv6</u> (string), optional	default gate- way for out- going traffic
		NetworkAddressConfiguratio- nIPv6DHCP[]	
		method string, one of [static, dhcp, auto], re- quired	Configuration method used for the address.
		NetworkAddressConfiguratio- nlPv6Auto[]	
		method string, one of [static, dhcp, auto], re- quired	Configuration method used for the address.

Remove all IPv6 address configurations

```
{
  "ipv6": {
     "address_configurations": []
   }
}
```

Replace existing IPv4 configuration with DHCP

Set static and dynamic IPv4 configuration

1.3.100 NetworkInterfaceName

Unique name describing a network interface.

Type Information

NetworkInterfaceName (string), pattern: ^[a-z0-9-]+\$

Examples

eth0

1.3.101 NetworkInterfaceStaticData

Properties

iface	unique name describing a network interface	
NetworkInterfaceName (string), pattern: ^[a-z0-9-]+\$, re-		
quired, read-only		
hardware_address	unique hardware address of a network interface	
MacAddress (string), pattern: ^ ([a-f0-9]{2}:) {5}[a-		
f0-9] {2}\$, required, read-only		
has_link	current physical connection status (whether a cable is	
boolean, required , read-only	plugged in or not)	

1.3.102 NormalizationConstant

Normalization constants are related to the White Reference.

Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).

Type Information

Array of number

1.3.103 ReleaseChannel

Describes the kind of a publication

Releases on the stable channel are generally considered well-tested and are recommended for use in production.

Releases on the feature add new features but haven't been tested as much as a stable release. Feature releases can but should only be used in production with careful consideration.

Type Information

ReleaseChannel (string), one of [stable, feature], default: stable

Examples

stable

1.3.104 SampleRate

Type Information

SampleRate (number), minimum: 0.01

1.3.105 SamplingSettings

Sampling Settings describe all details of the sampling process.

Its attributes may be queried and inspected (e.g. in order to retrieve the current sample rate).

Most values stored within the Sampling Settings should not be modified directly. The related API endpoint /api/sensor/detection-profiles/current/autogain should be used instead.

The only modifiable attribute within the Sampling Settings is the *averages* value. It is safe to change it, even though the default values calculated during an autogain operation should be optimal for most detection tasks.

Properties

led_intensity Number, minimum: 0, maximum: 1, required	relative intensity of the internal emitter during the light phase
base_sample_rate SampleRate (number), minimum: 0.01, required	The base sample rate determines the duration of a sampling period. After each sampling period, the gathered data is processed and a new detection result is calculated (e.g. the most suitable <i>Matcher</i> for the given sample). This may affect the state of the Switching Outputs or trigger configured actions. Thus the base sample rate defines the maximum rate of changes for the Switching Outputs. See also the <i>effective sample rate</i> .
effective_sample_rate SampleRate (number), minimum: 0.01, required	The effective sample rate is the numeric product of the base sample rate and the number of averages. It determines the minimum duration that a target needs to be sampled in order to determine its visual appearance correctly. With the default value of average set to one, this value is equal to the base sample rate.
minimum_wanted_sample_rate SampleRate (number), minimum: 0.01, required	This informational value represents the sample rate that was requested during the most recent <i>Autogain</i> operation. The effective sample rate may deviate from the wanted sample rate, if the requested sample rate was not achievable due to limitations of the sensor (e.g. exceeding the supported sample rate) or due to the environment (e.g. not enough light, thus a slower amplification with higher gain was necessary).
sample_light_phase Boolean, required	defines if the sensor should periodically activate the internal emitter for sampling

sample_dark_phase	defines if the sensor should periodically deactivate the internal emitter for sampling
Boolean, required	
averages	Number of previous samples to be averaged for every sampling result. A rolling avera-
AverageSampleCount (integer), min-	ging algorithm is applied to the samples.
imum: 1, required	
amplification	The amplification level specifies the internal configuration of an amplifier. This value is
AmplificationLevel (integer), requi-	not meant to be manipulated by regular users. It should be handled as is (stored, trans-
red	mitted and applied without modification or introspection).

```
{
  "led_intensity": 0.7,
  "base_sample_rate": 1000,
  "effective_sample_rate": 1000,
  "minimum_wanted_sample_rate": 1000,
  "sample_light_phase": true,
  "sample_dark_phase": true,
  "averages": 1,
  "amplification": 5
}
```

1.3.106 Sensor Capabilities

Provide access to the sensoric details supported by this device (e.g. colorspaces, input and output lines, ...).

Properties

maximum_sample_rate Integer, required	the maximum sample rate the sensor supports		
tolerances Array of ColorTolerance	List of tolerance specifica- tions supported by the sen-		
(union), required	sor		
	InfiniteColorTolerance		
	limits Object, required	limits	
	shape ToleranceShapeName	Name of the geometrical shape of the tolerance. The	
	(string), required	supported tolerance shapes	
		can be retrieved via	
		/api/sensor/capabili- ties.	
	SphereColorTolerance		
	limits Object, required	limits radius Numer, required	
	shape	Name of the geometrical	
	ToleranceShapeName	shape of the tolerance. The	
	(string), required	supported tolerance shapes	
	(09), 109400	can be retrieved via	
		/api/sensor/capabili- ties.	
	CylinderColorTolerance		
	limits	limits	
	Object, required	radius	
		Number, required	
		half_height	
		Number, required	
	shape	Name of the geometrical	
	ToleranceShapeName	shape of the tolerance. The	
	(string), required	supported tolerance shapes can be retrieved via	
		/api/sensor/capabili- ties.	
	BoxColorTolerance		
	limits	limits	
	Object, required	half_edges	

		Array of number, minimum	
		items: 3, maximum items: 3,	
	- In a second	required	
	shape ToleranceShapeName	Name of the geometrical shape of the tolerance. The	
	(string), required	supported tolerance shapes	
	(stillg), required	can be retrieved via	
		/api/sensor/capabili-	
		ties.	
output drivers	List of supported electrical		
Array of Switch-	output drivers		
ingOutputDriver (string),			
required			
trigger_sources	Beinhaltet die Liste verfügba-		
Array of TriggerSource,	rer Auslösequellen mit ihrem		
required	dazugehörigen Auslösefall. Auslösefälle können zum		
	Ausführen bestimmter Aktio-		
	nen automatisiert werden.		
	TriggerSource[]		
	name	Name of the trigger input	
	String, required	a.no or the trigger input	
	events	TriggerEvent []	
	Array of TriggerEvent, re-	990.2.0111 []	
	quired		
		name	
		TriggerEventName (string), re-	
		quired	
output_pin_count	Number of available switch-		
Integer, required	ing output lines		
Actions	Deprecated: use /api/actions		
Array of Action, re-	instead		
quired, Deprecated			
	Action[]		
	name	Unique name of the action	
	String, required		
	arguments Object, required	arguments	
colorspaces	List of supported color-		
Array of Colorspace, re-	spaces.		
quired	Spaces.		
	Colorspace[]		
	name		
	String, required		
	space_id	Unique name of a colorspace	
	ColorspaceID, required		
	axes	ColorspaceAxis[]	
	Array of ColorspaceAxis, min-		
	imum items: 3, maximum		
	items: 3, required		Liebana a
		id Ctring required	Unique name
		String, required	Human-readable name
		label	Human-readable name
		String, required minimum	lowest expected value of a
		Number, required	color along this axis under
			usual circumstances
		maximum	highest expected value of a
		Number, required	color along this axis under
			usual circumstances
colorspace_toler-	The evaluation of tolerances		
ance_maps	against positions of detecta-		
Array of ColorspaceTol-	bles depends on the cur-		
eranceMap, required	rently configured colorspace.		
	For example the tolerance at-		
	tribute "half_height" refers to		
	the brightness-related axis of		
	a colorspace (e.g. "L*" for the		
	"Lab*" colorspace) and is used for the height of the cy-		
	lindrical tolerance shape and		
	the first edge of the box toler-		
	ance shape.		

	The hue-related attributes (e.g. "a" and "b" for the "Lab*" colorspace) are used for the "radius" of a cylinder toler- ance shape and the second and third edges of the box		
	tolerance shape.		
	The colorspace_toler-		
	ance_maps define these rela-		
	tionships between color-		
	spaces and tolerances.		
	ColorspaceToleranceMap[]		
	colorspace_id ColorspaceID (string), re- quired	Unique name of a colorspace	
	tolerance_shape ToleranceShapeName (string), required	Name of the geometrical shape of the tolerance. The supported tolerance shapes	
	(can be retrieved via /api/sensor/capabili- ties .	
	limits_axes_map Object, required	limits_axes_map	
		half_height Array of string, optional	
		half_edges Array of string, optional	
		radius Array of string, optional	
settings_categories Array of string, required	List of categories that can be selected during import to control which settings should be applied. See the documentation for the POST request to `/api/seetings.		
maximum_detecta-	Maximum number of color		
bles_count	positions (Detectable) to be		
Integer, required	stored in a detection profile.		
maximum_match-	Maximum number of detec-		
ers_count	tion results (Matcher) be		
Integer, required	stored in a detection profile.		

Examples

```
"output_pin_count": 8,
"tolerances": [
  {
    "shape": "infinite",
    "limits": {}
  },
  {
    "shape": "sphere",
    "limits": {
       "radius": 2
     }
  },
    "shape": "cylinder",
"limits": {
       "half_height": 4,
       "radius": 2
  },
     "shape": "box",
     "limits": {
```

```
"half edges": [
        4,
        2,
        2
      ]
    }
  }
],
"actions": [
  {
    "name": "enable switching output",
    "arguments": {}
  },
    "name": "teach_single",
    "arguments": {}
  }
],
"maximum sample rate": 20000,
"maximum detectables count": 256,
"maximum matchers count": 256,
"trigger_sources": [
  {
    "name": "trigger_0",
    "events": [
      {
        "name": "trigger 0 level high"
      },
      {
        "name": "trigger 0 level low"
      },
        "name": "trigger_0_edge_rising"
      },
      {
        "name": "trigger_0_edge_falling"
    ]
  } ,
    "name": "trigger_1",
    "events": [
      {
        "name": "trigger 1 level_high"
      },
      {
        "name": "trigger 1 level low"
      },
      {
        "name": "trigger_1_edge_rising"
      },
      {
        "name": "trigger 1 edge falling"
    ]
  } ,
    "name": "trigger 2",
    "events": [
        "name": "trigger_2_level_high"
      },
```

```
"name": "trigger 2 level low"
      } ,
      {
        "name": "trigger_2_edge_rising"
      },
      {
        "name": "trigger_2_edge_falling"
    ]
  },
    "name": "trigger 3",
    "events": [
      {
        "name": "trigger 3 level high"
      },
      {
        "name": "trigger 3 level low"
      } ,
      {
        "name": "trigger 3 edge rising"
      },
      {
        "name": "trigger_3_edge_falling"
    ]
  }
],
"colorspaces": [
  {
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
        "id": "a",
        "label": "a*",
        "minimum": -500,
        "maximum": 500
      },
      {
        "id": "b",
        "label": "b*",
        "minimum": -200,
        "maximum": 200
      }
    ],
    "name": "L*a*b*",
    "space_id": "Lab"
  },
  {
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
        "id": "u",
```

```
"label": "u*",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "v",
      "label": "v*",
      "minimum": 0,
      "maximum": 100
    }
  ],
  "name": "L*u*v*",
  "space id": "Luv"
},
  "axes": [
    {
      "id": "X",
      "label": "X",
      "minimum": 0,
      "maximum": 120
    },
    {
      "id": "Y",
"label": "Y",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "Z",
      "label": "Z",
      "minimum": 0,
      "maximum": 120
   }
  ],
  "name": "XYZ",
  "space id": "XYZ"
},
{
  "axes": [
    {
      "id": "x",
      "label": "x",
      "minimum": 0,
      "maximum": 1
    },
    {
      "id": "y",
      "label": "y",
      "minimum": 0,
      "maximum": 1
    },
    {
      "id": "Y",
      "label": "Y",
      "minimum": 0,
      "maximum": 100
    }
  "name": "xyY",
  "space_id": "xyY"
},
{
```

```
"axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
      {
        "id": "u",
        "label": "u'",
        "minimum": 0,
        "maximum": 1
      },
        "id": "v",
        "label": "v'",
        "minimum": 0,
        "maximum": 1
      }
    ],
    "name": "L*u'v'",
    "space id": "uvL"
],
"output_drivers": [
 "off",
  "npn",
  "pnp",
  "push-pull"
],
"colorspace tolerance maps": [
  {"colorspace id": "Lab", "tolerance shape": "box",
  "limits_axes_map": {"half_edges": ["L", "a", "b"]}},
  {"colorspace id": "Lab", "tolerance_shape": "cylinder",
   "limits_axes_map": {"half_height": ["L"], "radius": ["a", "b"]}},
  {"colorspace_id": "Luv", "tolerance_shape": "box",
   "limits axes map": {"half edges": ["L", "u", "v"]}},
  {"colorspace id": "Luv", "tolerance shape": "box",
   "limits axes map": {"half height": ["L"], "radius": ["u", "v"]}}
],
"settings categories": [
 "access",
  "defaults",
  "emitters",
  "firmware",
  "keypad",
  "network",
  "outputs",
  "peripherals",
  "sensor",
  "system"
1
```

1.3.107 SerialElizaSettings

Properties

}

```
type
string, one of [none, eliza, modbus], default: eliza, required
```

1.3.108 SerialModbusSettings

Properties

type	
string, one of [none, eliza, modbus], default: eliza, required slave id	
any of number or null, required	
frame_format	
string, one of [rtu, ascii], default: rtu, required	

1.3.109 SignalColor

A custom color name. How and what color will be displayed is defined by the client.

Type Information

any of string or null

1.3.110 SphereColorTolerance

Properties

limits	limits
Object, required	
	radius
	number, required
shape	Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via
ToleranceShapeName	/api/sensor/capabilities.
(string), required	

1.3.111 SupportedTimezones

List of timezones supported by the device.

Type Information

Array of string

Examples

```
[
  "Africa/Casablanca",
  "Antarctica/Troll",
  "Europe/Berlin",
  "UTC"
]
```

1.3.112 SwitchingOutputDriver

The Output Driver defines the electrical behaviour of the switching outputs. The supported output drivers can be retrieved via /api/sensor/capabilities.

Type Information

<u>SwitchingOutputDriver</u> (string)

Examples

```
push-pull
```

1.3.113 SwitchingOutputs

Eletrical output lines can drive external actors in different electrical modes.

Properties

output_driver The Output Driver defines the electrical behavior	
SwitchingOutputDriver (string), required	switching outputs. The supported output drivers can be re-
	trieved via /api/sensor/capabilities.
count	Number of available output lines
integer, required	

Examples

```
{
  "count": 8,
  "output_driver": "push-pull"
}
```

1.3.114 SwitchingOutputsWritable

Eletrical output lines can drive external actors in different electrical modes.

Properties

output_driver The Output Driver defines the electrical behaviour		
SwitchingOutputDriver (string), required	switching outputs. The supported output drivers can be re-	
	trieved via /api/sensor/capabilities.	

1.3.115 SystemSettings

Properties

hostname	Human-readable name identifying the device in the network
Hostname, pattern: ^ (?: [a-zA-Z0-9] (?: [a-zA-Z0-	
9\-]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-	
zA-Z0-9\-] * [a-zA-Z0-9]) ?\$, optional	
uptime	The current system uptime in seconds. Though highly unlikely
any of number or null, optional, read-only	can be nil in case the system reported an invalid value.

Examples

```
{
  "hostname": "cfo-7454232361"
}
```

1.3.116 SystemTimeSettings

Properties

now	current time from the perspective of the sensor
Timestamp (string), optional	
timezone	currently configured timezone
String, optional	
ntp servers	one or more network time servers
Array of string, optional	
default ntp servers	preconfigured network time servers
Array of string, optional, reaed-only	

Examples

```
{
  "now": "2018-01-24T15:45:15.694004+01:00",
  "timezone": "Europe/Berlin",
  "ntp_servers": [
      "pool.ntp.org"
],
```

```
"default_ntp_servers": [
    "pool.ntp.org"
]
```

1.3.117 Timestamp

Timestamp (Format: ISO 8601)

Type Information

Timestamp (string)

Examples

```
2018-01-24T14:04:26+01:00
```

1.3.118 TimestampBackendUptime

The timestamp (given in microseconds) is based on the uptime of the internal analog sensor backend. It may get reset to zero under specific conditions.

Type Information

TimestampBackendUptime (number), minimum: 0

1.3.119 ToleranceShapeName

Name of the geometrical shape of the tolerance. The supported tolerance shapes can be retrieved via /api/sensor/capabilities.

Type Information

ToleranceShapeName (string)

1.3.120 TransformedColor

A color represented by a coordinate in the colorspace. The array indices of the values property match the order of the colorspace.axes property of currently used detection profile.

1.3.121 TriggerEvent

Trigger Events can be emitted by their trigger source. Actions can be attached to a Trigger Event (see /api/sensor/action-triggers).

Properties

name

TriggerEventName (string), required

1.3.122 TriggerEventName

Type Information

TriggerEventName (string)

Examples

```
trigger 0 level high
```

1.3.123 TriggerSource

Each Trigger Source is a peripheral input with the ability to emit one or more Trigger Events.

Properties

name String, required	Name des Auslöseeingangs
events Array of TriggerEvent, required	TriggerEvent []
	name TriggerEventName (string), required

Examples

1.3.124 TriggerSourcesStatus

The sensor has a number of input lines that can be used as trigger sources. The event counters are updated periodically (approximately every second).

Properties

trigger_sources Array of object, required	object[]		
	name		
	string, required		
	event_counters	event_counters	
	Object, required		
		edge_falling	
		Number, required	
		edge_rising	
		Number, required	
		level_low	
		Number, required	

Examples

```
}
}

},
{
    "name": "trigger_1",
    "event_counters": {
        "edge_falling": 0,
        "edge_rising": 0,
        "level_low": 35124832,
        "level_high": 0
    }
}

}
```

1.3.125 User

Properties

name	unique name identifying an account
string, pattern: ^ [\w-] +\$, required,	
read-only	
password	Password assigned to this account (only writable; never returned in responses).
string, optional	Either a password or a password hash needs to be supplied when creating a
	new user or changing a password.
password_hash	Password hash assigned to this account. Either a password or a pass-
HashDigest (string), pattern: ^ [a-f0-	word hash needs to be supplied when creating a new user or changing a pass-
9]+\$, optional	word.
roles	The roles assigned to an account define its set of permissions.
Array of string, optional	

Examples

```
{
   "name": "alice"
}
```

1.3.126 UUID

Unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8.

Type Information

```
UUID (string), pattern: ^ [a-f0-9-] +$
```

Examples

```
a014e415-0fec-4734-ac3f-30da0a5f3899
```

1.3.127 WantedSwitchingOutputsState

The combination of tristate values describes a logical state of the switching outputs of the sensor.

The states true or false cause the output to go up or down. The state null keeps the previous state of the output unchanged.

Properties

uuid UUID (string), pattern: ^[a-f0-9-]+\$, required, readonly	unique identifier (UUID) as defined by RFC 4122, ITU-T Rec. X.667, and ISO/IEC 9834-8
states Array of any of boolean or null, required	List of True/False/Null values describing the wanted states of the Switching Outputs

Examples

```
{
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
    true,
    false,
    false,
    false,
    false,
    false,
    false,
    false,
    false,
    false,
    false
]
```

1.3.128 WhiteReference

The White Reference attribute is used for indicating a custom color balancing.

Its content is subject to internal use. Thus it should not be accessed directly, but only through the related API endpoints (e.g. /api/sensor/detection-profiles/{itemId}/white-reference).

Type Information

Array of number

1.3.129 WrappedNetworkInterfaceAddressIPv4

Network addresses (IPv4) in CIDR notation.

Properties

address	IPv4 network address in CIDR notation
NetworkInterfaceAddressIPv4 (string), required	

1.3.130 WrappedNetworkInterfaceAddressIPv6

Network addresses (IPv6) in CIDR notation.

Properties

address	IPv6 network address in CIDR notation
NetworkInterfaceAddressIPv6 (string), required	

2 Terminal Documentation

The text-based terminal is accessible via the following interfaces of the sensor:

- RS-232 (SYS connector pins)
- USB (optional)

2.1 Preface

2.1.1 Connection Details

The sensor uses the following configuration to communicate via the serial interface:

Baud Rate 19200 (RS-232)

Data Bits 8

Stop Bits 1

Parity None

Line Feed 0x0A/LF/\n

Encoding UTF-8

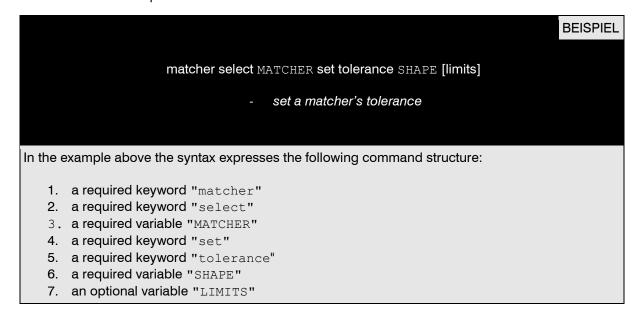
2.1.2 Syntax

The serial console uses color, markup, and font weight to outline how commands can and should be used. As not all terminals support color and font formatting rules you may only see the described markup in your terminal.

Output is formatted as following:

- lowercase letters indicate a keyword
- uppercase letters indicate a variable
- square brackets indicate an optional keyword or variable
- the [...] character sequence indicates an unlimited number of variables
- optional keywords and variables are colored in grey
- required variables are bold & white

Please mind that all input is case-sensitive. matcher select is not the same as MATCHER SELECT.



2.1.3 Output Formats & Message Parsing

When used in automation it's recommended to use the JSON output format. You can switch between output formats with the set output-format command. The available formats are json and human. Please be advised that human output format is subject to change and is currently not recommended for parsing.

Once a command is executed it will output one or more data packets. Independent of the output format these packets end with a specific two-byte sequence. The first byte indicates if the packet was generated by a successful command and is either 0x20 in case of success or 0x07 in case of a failure. The second byte marks the end of the packet and is guaranteed to be 0x00, the ASCII NUL character.

2.1.4 Nomenclature

The serial interface uses a set of names and identifiers that are closely modelled after the sensors REST API.

UUID

UUID is short for "Universally unique identifier" and is a 36-characters long character sequence with 5 alpha-numeric groups separated by dashes. Most collections use a UUID as unique keys for their items. UUIDs used by the serial interface and the underlying REST API are UUIDv4 as specified in RFC 4122.

Detectable position

The commands for adding and editing a detectable use a parameter called the position.

This value refers to the color property in the REST API and describes the three-dimensional location of the color in the currently activate colorspace. The format regex is

```
^(\d+(?:\.\d*)?),(\d+(?:\.\d*)?), (\d+(?:\.\d*)?)$ (Example: 3.14,7.6,8).
```

2.1.5 Common Patterns

Some tasks in the serial interface share common or similar behaviours. These recurring patterns are outlined here.

Collections and the select keyword:

Collections are a set of items of a specific type that the sensor controls. Most collections handled by the sensor use a UUID as unique key, by which they can be referenced.

In the context of the serial interface collections can easily be spotted by the use of the "select" keyword followed by a variable as part of the command syntax. Even though collections usually use a UUID or another unique key as identifier, the variable can take other arguments. You may also use a list index, as defined by the corresponding list command for this collection or the special index '-1' to reference the last item of this type that you've created.

Property Commands using show [PROPERTY]:

Most commands that display detailed information on an object (may it be a collection item or any other resource) using the the show keyword support the output of any of the attributes individually. The help command for the command will show you the allowed choices for the property value.

2.1.6 Differences compared to the REST API

The serial protocol is based on the REST API but there are some differences that are outlined in this section.

- There is no feature parity. The REST API is the primary configuration method for the sensor. Not all features implemented in the REST API are available via the serial interface.
- Some commands like samples stream implement parameters that don't match the corresponding API endpoint. This may be the case if the behaviour controlled by these parameters is specific to the HTTP or serial protocol.
- While the REST API uses plural resources names like matchers the serial protocol uses singular command names (in this case matcher).
- There is no general /defaults API endpoint implementation available on the serial interface. Instead commands like matcher default set hold_time implement a more finegrained access

2.2 Command Reference

2.2.1 Access Command

Handle security related tasks like logins and session management.

Supported Subcommands

```
access login USERNAME [PASSWORD]
```

- Login with username and password to access privileged commands.

```
access logout
```

- Logout and invalidate the current session.

```
access session
```

- Show current session information.

2.2.2 Device Command

Supported Subcommands

```
device [show] [PROPERTY]

[PROPERTY]: id | model_key | model_name | variant | vendor_key |
    vendor name
```

2.2.3 Firmware Command

The sensor firmware provides the connectivity of the sensor and its services, as well as the functionality of sensor backend.

Supported Subcommands

```
firmware [version]
```

- Show the current firmware version.

```
firmware recovery [show] [PROPERTY]
```

- Show information about the recovery firmware.

```
[PROPERTY]: channel | created_on | id | name | version
```

firmware recovery restore

- Restore the system from the recovery image. All settings are reset to their defaults.

```
firmware recovery upgrade
```

- Store the currently running firmware image as a recovery image. You may use this operation after verifying a successful firmware upgrade.

2.2.4 Help Command

List and describe all available commands.

Supported Subcommands

```
help [COMMAND]
```

2.2.5 Keypad Command

The keypad provides local access to most basic sensor actions.

Supported Subcommands

```
keypad [show] [PROPERTY]
```

- Show the keypad status. [PROPERTY]: lock

keypad lock [STATE]

- Change the state of the keypad lock mechanism.

2.2.6 Matcher Command

A matcher specifies the sensor behaviour based on the sensor input.

Supported Subcommands

Matcher

A matcher contains multiple detectables (colors) and the desired sensor behaviour (e.g. out put states and hold time) that should be applied when one of the colors is detected.

```
matcher[list]
```

- Show the list of configured matchers for the detection profile.

```
matcher add [OUTPUT PATTERN]
```

- Add a matcher to the detection profile.

```
matcher remove all
```

- Remove all matchers.

```
matcher select MATCHER [show] [PROPERTY]
```

- Show detailed information about a matcher.

```
[PROPERTY]: hold_time | name | num_detectables | output_pat-
tern | reset_output_after_hold_time_expired | signal_color |
tolerance | uuid
```

matcher select **MATCHER** remove

- Remove a single matcher.

```
matcher select MATCHER set name NAME
```

- Set a matcher's name.

```
matcher select MATCHER set hold time DURATION
```

Set a matcher's hold time.

```
matcher select MATCHER set output_pattern BITMASK
```

- Set a matcher's output bitmask.

```
matcher select MATCHER set tolerance SHAPE [LIMITS]
```

- Set a matcher's tolerance.

```
SHAPE: One of the shapes defined by the API.

box / cylinder / infinite / sphere
```

[LIMITS]: A string describing the limits of the tolerance (e.g. "2r" for a sphere, "4h/6r" for a cylinder, or "1/2/3" for a box). Optional for infinite, but required for every other shape.

Matcher Defaults

Whenever a new matcher is created a few properties are set to predefined default values. You can change these defaults to your liking in order to reduce the number of changes needed afterwards.

```
matcher default [show] [PROPERTY]
```

Display matcher default values.
 [PROPERTY]: hold time | tolerance

matcher default set hold_time DURATION

- Set the default hold time for new matchers.

matcher default set tolerance SHAPE [LIMITS]

- Set the default tolerance for new matchers.

SHAPE: One of the shapes defined by the API.

box | cylinder | infinite | sphere

[LIMITS]: A string describing the limits of the tolerance (e.g. "2r" for a sphere,

"4h/6r" for a cylinder, or "1/2/3" for a box). Optional for infinite, but re-

quired for every other shape.

Detectable

Multiple detectables can belong to a matcher. Each detectable represents a position within the currently active colorspace.

```
matcher select MATCHER detectable [list]
```

- List all detectables belonging to a matcher.

```
matcher select MATCHER detectable remove all
```

- Remove the detectables belonging to a matcher.

```
matcher select MATCHER detectable add [POSITION]
```

- Add a detectable to a matcher. Sample the current detectable, if no position is given.

[POSITION]: A position in the current color space. Expected format: 23.918,6,17.29113

matcher select MATCHER detectable select DETECTABLE [show] [PROPERTY]

- show detailed information about a detectable [PROPERTY]: matcher | position | rgb | uuid

matcher select MATCHER detectable select DETECTABLE remove

- Remove a single detectable from a matcher.

matcher select ${\tt MATCHER}$ detectable select ${\tt DETECTABLE}$ set position ${\tt PO-SITION}$

Modify the detectable's position in the colorspace.

[POSITION]: Eine Position in dem aktuellen Farbraum. Erwartetes Format: 23.918,6,17.29113

2.2.7 Network Command

The network configuration allows the use of network-based sensor features (e.g. API or the web interface).

Supported Subcommands

```
network [list]
```

- Show the connection state and active addresses of all network interfaces.

network reset

- Reset the network configuration to its factory default.

```
network select INTERFACE [show] [PROPERTY]
```

- Show the connection state and active addresses of a network interface.

[PROPERTY]: ipv4_addresses | ipv4_config | ipv6_addresses | ipv6_config | link | mac | name

```
network select INTERFACE set ipv4 dhcp
```

- Request a dynamically assigned IP address (via DHCP) for a network interface.

```
network select INTERFACE set ipv4 static ADDRESS [GATEWAY]
```

- Define a static IPv4 address for the network interface.

```
network select INTERFACE set ipv4 disabled
```

- Disable IPv4 connectivity for the network interface.

```
network select INTERFACE set ipv6 auto
```

- Enable IPv6 state-less auto network configuration (SLAAC) for the interface.

```
\verb"network" select {\bf INTERFACE} set ipv6 dhcp
```

- Request a dynamically assigned IP address (via DHCPv6) for a network interface.

```
network select INTERFACE set ipv6 static ADDRESS [GATEWAY]
```

- Define a static IPv6 address for the network interface.

```
network select INTERFACE set ipv6 disabled
```

- Disable IPv6 connectivity for the network interface.

2.2.8 Repeat Command

Conveniently execute a command multiple times (e.g. following changes of the color sampling results).

Supported Subcommands

```
repeat REPETITIONS DELAY [ARGUMENTS [...]]
```

- Repeat a single command for a number of times with a given delay. You may stop execution with CTRL-C.

REPETITIONS: number of repetitions, 0 for infinite

DELAY: delay in seconds, 0 for no delay

2.2.9 Sample Command

Request sample results from the sensor.

Supported Subcommands

```
sample [show] [PROPERTY]
```

Show the current color sample.
 [PROPERTY]: color | detection | output_pattern | timestamp | trigger

```
sample stream [COUNT] [FREQUENCY]
```

 Retrieve a continuous stream of color samples from the sensor [COUNT]: number of records to retrieve, default 0 for infinite [FREQUENCY]: speed of samples in hertz, default infinite

2.2.10 Sensor Command

Sensor settings influence the sampling and processing of sensor signals. Changed settings may invalidate previously sampled detectables.

Supported Subcommands

```
sensor colorspace [show]
```

- Show the currently configured colorspace.

```
sensor colorspace list
```

- List available colorspaces.

```
sensor colorspace set COLORSPACE
```

Switch to a different colorspace.

```
sensor autogain [SAMPLE RATE] [TARGET LEVEL]
```

- Perform the autogain procedure in order to adjust the sensor to the current optical environment (distance, light intensity, target appearance).

```
sensor white-reference reset
```

Reset the white reference to the factory default. The factory default works well
with a commonly used optical path (fiber and optics). Use the factory default if a
proper white reference target is not available.

```
sensor white-reference sample
```

- Sample a new white reference from the current target. The target should be neutral white. This may improve the calculation of absolute color values within the given colorspace.

2.2.11 SET Command

Change properties of the console interface.

Supported Subcommands

set echo STATE

 enable/disable any output of prompts or typed text STATE: off | on

set output-format FORMAT

 switch the response output format FORMAT: human | json

2.2.12 System Command

Interact with the system hosting the sensor.

Supported Subcommands

```
system settings reset
```

- Reset all settings to their factory defaults.

```
system hostname [show]
```

- Show the system's hostname.

```
\verb|system| hostname| set | \textbf{HOSTNAME}|
```

- Define the hostname of the system.

```
system timezones [list]
```

- List all timezones supported by the system

```
system time [show] [PROPERTY]
```

 Show the system's current time settings [PROPERTY]: now | timezone

```
system time set now TIME
```

- Set the system's current time in ISO8601 format.

```
system time set timezone TIMEZONE
```

- Set the system's timezone

```
system reboot
```

- Reboot the device

3 Modbus Documentation

Modbus protocol is only available for the colorSENSOR CFOXXX(100) Option 100.

3.1 Introduction

The modbus protocol is a single-master protocol. Data is exchanged over a serial or via network (TCP/IP) interface. The Controller acts as a Modbus slave: It responds to requests from a master.

The Modbus protocol allows partial access to the most relevant features of the controller. Internally it uses the HTTP-based API of the controller for all operations.

The colorSENSOR Modbus interface supports the following protocol features:

- Transport via TCP (IPv4 and IPv6)
- Transport via RS232 and USB using RTU (default) or ASCII format
- Serial baudrates: 9600, 19200 (default), 115200

The Modbus slave address (relevant only for serial connections) is configurable. By default the colorSENSOR CFO is not bound to a specific address, but responds to every packet.

The full set of supported commands is available as a JSON dump. This structured dataset is supposed to ease the generation of a vendor-specific Modbus mapping for the controller.

3.2 Quickstart

The following configuration details and hints should ease the first steps with the controllers Modbus protocol implementation:

- Connect to the sensors Modbus protocol via RS232 (Baudrate: 19200), USB or TCP (port 502).
- Use Big-Endian (byte-order and word-order) when interpreting data in Modbus responses.
- Consider the 1-based addressing scheme when accessing registers. For example a documented address of 501 is transmitted over the wire as 500. Most Modbus client implementations will apply this translation implicitly. Only very few implementations use the on-wire address instead. In this case the documented address needs to be decremented for these specific clients.
- Retrieve the Input Registers from 500 up to 508 via a Modbus request. These registers contain fixed values in different formats (e.g. float, 32 bit and 64 bit integer). Ensure that your client implementation interpretes these values properly according to their documented value (see the register content documentation). In case of misinterpretations you may need to adjust the endianness or the address offset of your client implementation.

3.3 Supported colorSENSOR Features

The Modbus interface of the colorsensors provides most features of the following API endpoints:

- /defaults (only matcher-releated defaults)
- /device
- /firmware (only status retrieval; no upgrade)
- /firmware/recovery
- /firmware/recvery/upgrade-from-current
- /sensor/samples/current
- /sensor/matchers
- /sensor/detectable
- /sensor/detection-profiles
- /sensor/detection-profiles/autogain
- /sensor/detection-profiles/white-reference
- /sensor/capabilities
- /system
- /system/factory-reset
- /system/reboot
- /peripherals/outputs
- /peripherals/rs232
- /peripherals/usb
- /settings

The following API endpoints are not supported due to the volatile nature of their data or their complexity (hard to express within the modbus protocol):

- /access
- /action-triggers
- /actions
- /firmware/images
- /firmware/settings
- /network/interfaces/
- /peripherals/keypad
- /peripherals/trigger-sources
- /system/time
- /system/time/zones

3.4 Data Types and Register Addressing

3.4.1 Data Types and Modbus Functions

The Modbus protocol specifies different functions for accessing and manipulating values.

The following functions (and their respective function codes) are used for the different types of data:

Function	Code	Function name
Read-only bits	2	Read Discrete I
Writable bits	1	Read Coils
	5	Write single coils
	15	Write multiple coils
Read-only words	4	Read input registers
Writable words	3	Read Multiple Holding Registers
	6	Write Single Holding Register
	16	Write Multiple Holding Register
	23	Read/Write Multiple Registers

3.4.2 Register Addresses

The addressing of data via the Modbus protocol is not strictly specified. Different implementations use a variety of name schemes and offsets. The relevant details of this Modbus implementation are:

- All addresses written in this documentation are register offsets relative to the specific Modbus function.
- All addresses are 1-based. This approach is used by most Modbus implementations.

For example the register for the float test value is documented as a read-only word at address 501. This address could also be written as 30501 (based on a traditional Modbus addressing scheme mapping the functions to specific address ranges). The content of this register can be retrieved with the Read Input Registers function (function identifier "4"). The internal address of this value (as used for the on-wire format of Modbus) is 500 (due to the 1-based register addressing). This internal address is only used by very few Modbus client implementations. Most implementations use the 1-based address, instead.

Clients without support for address offsets may need to decrement every address (as documented here) when assembling the Mod-bus data frame.

3.4.3 Simple Data Types

The Modbus specification describes simple data types (bits and 16-bit-words). Additionally the following data types are used by the Modbus implementation of the colorSENSOR:

- Float values: two registers (32 bit), IEEE-754, big-endian word-order and byte-order.
- Integer values with 32 bit (two registers) or 64 bit (four registers): big-endian word-order and byteorder.
- Strings: the first word contains the length; all following bytes contain the ASCII characters. Each
 "word" register (after the length) contains two characters (first: upper, second: lower byte).Reading past the end of the string length is allowed and returns null bytes.Thus usually a trailing null
 byte is present at at the end of the string.But you may not rely on this, as the trailing null byte is
 missing, if the string uses exactlythe maximum number of allowed characters for this string.
- Bytes: a raw byte array is used for binary data transfers. Each register contains two characters (first: upper, second: lower byte). Reading past the end of the binary data is allowed and returns null bytes. The length of the raw data should be handled via a separate register.

Bitmask: 16 bit words are used to represent or manipulate boolean fields. Each bit represents a
single boolean value. The description of each bitmask data field maps bit positions to the boolean state described by this bit. A value of zero is considered to be "false" (not active). A value of
one is true. The bit positions start with zero with the least significant bit.

3.5 Session State, Concurrency and Multiple Interfaces

Multiple interfaces of the sensor can communicate via the modbus protocol. Each hardware interface (e.g. RS232, USB) manages its own state. This is relevant for stateful operations (e.g. access to a collection), that require a sequence of read or write requests. The Ethernet interface accepts TCP connections. Each connection tracks its own state for the duration of the connection.

3.5.1 Functions

3.5.1.1 Autogain Procedure API Endpoint: /sensor/detection-profiles/current/autogain

Execute the autogain procedure in order to determine suitable sampling properties for the current optical environment. The resulting sampling setup is applied automatically. These new settings are in effect as soon as the response is sent. The success or failure of an autogain procedure can be verified as soon as the autogain is running flag is cleared.

Address	Type	Operation	Description		FC
00020	Bit	write	Start an autogain procedure		5, 15
00302 Bitmask		itmask read	Status of the	most recently started autogain procedure	4
			Position	Description	
			0	Is still running	
			1	Finished successfully	
			2	Failed: Target is too dark	
00410	Float	read / write	Minimum war	nted sample rate	3, 4, 6, 16
00412	Float	read / write	Target analog input level		3, 4, 6, 16
00414	Uint16	read / write	Number of samples used for averaging		3, 4, 6, 16
00415	Bitmask	read / write		for autogain procedure	3, 4, 6, 16
			Default value: 65535		
			Position	Description	
			0	Enable internal emitter	
			1	Enable ambient tight compensation	
00416	Bitmask	read / write	Override defa	ult autogain settings with custom values	3, 4, 6, 16
			Position	Description	
			0	Overwrite minimum wanted sample rate	
			1	Overwrite target analog input level	
			2	Overwrite number of samples used for	
				averaging	

3.5.1.2 White reference API Endpoint: /sensor/detection-profiles/current/white-reference

The white reference is used for calculating accurate color positions in the colorspaces. The factory default white reference is suitable for a special set of sensor and optics. A custom white reference can be sampled. A reference white target is recommended for this.

Address	Туре	Operation	Description	FC
00021	Bit	write	Reset the custom white reference	5
00022	Bit	write	Sample a custom white reference	5

3.5.1.3 Add Color to Color Table API Endpoint: /sensor/matchers

Address	Type	Operation	Description	FC
00024	Bit		Create a new matcher and assign the current color position to it (as a detectable).	5,15
00451	Uint16	read	Retrieve the identifier of the most recently created matcher.	4

3.5.1.4 Manage Color Positions of a Color Group API Endpoint: /sensor/matchers

Each color group (matcher) may refer to one or more color positions (detectables).

Address	Туре	Operation	Description	FC
00025	Bit	write	Add a new detectable to an existing matcher (color group).	5
00026	Bit	write	Delete all detectables of an existing matcher (color group).	5
00027	Bit	read	Indicate whether the currently selected matcher exists.	1
00311	Uint16	read	Current number of detectables (color positions) assigned to the matcher.	4

00450	Uint16	read / write	Specify the matcher (color group) when adding or removing detecta-	3, 6
			bles (color positions).	

3.5.1.5 Read Sensor Capabilities API Endpoint: /sensor/capabilities

Inspect the available features of the controller.

Address	Type	Operation	Description	Description			
00300	Uint16	read	Number of available switching outputs		4		
00301 Bitmask		read	Colorspaces su	pported by the sensor	4		
			Position	Description			
			0	XYZ			
			1	L*a*b*			
			2	xyY			
			3	L*u*v*			
			4	L*u'v'			
00303	Bitmask	read	Available tolera	nce shapes	4		
					Position	Description	
			0	Infinite (classification)			
			1	Sphere			
			2	Cylinder			
			3	Box			
00304	Bitmask	read	Available switch	ning output drivers	4		
			Position	Description			
			0	Disabled			
			1	NPN			
			2	PNP			
			3	Push-Pull			
00305	Float	read	Maximum samp	ole rate	4		
00307	Uint16	read	Maximum numb	per of detectables	4		
00308	Uint16	read	Maximum numb	per of matchers	4		

3.5.1.6 Get Current Sample API Endpoint: /sensor/samples/current

Retrieve the latest color detection sample. A single read operation covering the complete memory range of the sample is guaranteed to be consistent. Multiple read operations in series will probably result in a combination of values from the different samples gathered during the time between the first and the last request.

Address	Type	Operation	Description	FC
00150	Uint64	read	Timestamp of the current sample	4
00154	Float	read	Signal level of the current sample	4
00156	Float	read	Representation of the color in the XYZ colorspace (X)	4
00158	Float	read	Representation of the color in the XYZ colorspace (Y)	4
00160	Float	read	Representation of the color in the XYZ colorspace (Z)	4
00162	Float	read	Representation of the color in the currently active colorspace L	4
00164	Float	read	Representation of the color in the currently active colorspace a	4
00166	Float	read	Representation of the color in the currently active colorspace b	4
00168	Float	read	Representation of the color as RGB values red (between 0.0 and 1.0)	4
00170	Float	read	Representation of the color as RGB values green (between 0.0 and 1.0)	4
00172	Float	read	Representation of the color as RGB values blue (between 0.0 and 1.0)	4
00174	Uint16	read	Inputs with a high level event during the last sample period (bit 0 -> IN0)	4
00175	Uint16	read	Inputs with a low level event during the last sample period (bit 0 -> IN0)	4
00176	Uint16	read	Inputs with a rising edge event during the last sample period (bit 0 -> IN0)	4
00177	Uint16	read	Inputs with a falling edge event during the last sample period (bit 0 -> IN0)	4
00178	Uint16	read	ID of the closest matcher in range of the last sample's color position. The value 65535 is returned if the sampled color position was not in range of any of the available matchers.	4
00179	Uint16	read	Currently active state of the Switching Outputs (bit 0 -> OUT0)	4
00180	Float	read	Distance (based on the axes of the currently configured colorspace) between the last sampled color position and the closest suitable matcher (if any). A negative value (-1) indicates that no matcher is in range. Distance 1	4
00182	Float	read	Distance (based on the axes of the currently configured colorspace) between the last sampled color position and the closest suitable matcher (if any). A negative value (-1) indicates that no matcher is in range. Distance 2	4
00184	Float	read	Distance (based on the axes of the currently configured colorspace) between the last sampled color position and the closest suitable matcher (if any). A negative value (-1) indicates that no matcher is in range. Distance 3	4

3.5.1.7 Status of the Color Table API Endpoint: /sensor/dtection-profiles/current

Retrieve the current usage of the color table.

Address	Туре	Operation	Description	FC
00309	Uint16	read	Current number of matchers (color groups) stored in the color table	4
00310	Uint16	read	Current number of detectables (color positions) stored in the color table	4

3.5.1.8 Clear Color Table API Endpoint: /sensor/matchers

Delete all colors that are stored in the color table.

Address	Type	Operation	Description	FC
00023	Bit	read	Remove all stored colors	5, 15

3.5.1.9 Switching Outputs Driver API Endpoint: /peripherals/outputs

Electrical output lines can drive external actors in different electrical modes. The currently active mode can be retrieved and changed.

Address	Type	Operation	Description	Description		
00400	Uint16	read / write	Retrieve and change the	Retrieve and change the current switching output driver.		
			Description	Values		
			off	0		
			npn	1		
			pnp	2		
			push-pull	3		

3.5.1.10 Firmware Version API Endpoint: /firmware

Read information about the firmware.

	Address	Type	Operation	Beschreibung	FC
Ī	00100	Uint16	read	Firmware Version (Major: X.0.0)	4
	00101	Uint16	read	Firmware Version (Major: 0.X.0)	
	00102	Uint16	read	Firmware Version (Major: 0.0.X)	

3.5.1.11 Device Information API Endpoint: /device

Read information about the device.

Address	Туре	Operation	Description	FC
00103	String	read	Device serial	4
00114	String	read	Vendor of device	
00123	String	read	Device model	
00132	String	read	Device variant	

3.5.1.12 Configure access lock API Endpoint

Lock or unlock certain methods of accessing the sensor.

Address	Type	Operation	Description		FC
00460	Bitmask	read / write	Lock or unlo	ck certain access actions	3, 4, 6, 16
			Position	Description	
			0	Lock keypad (ignore any keypress	
				event)	
			1	Reject read access for the API	
			2	Reject write access for the API	

3.5.1.13 Manage API users API Endpoint: /access/user

Manage the user accounts used by the HTPI API.

Address	Туре	Operation	Description	FC
00028	Bit	write	Remove all existing API users (i.e. disable API access	1, 5, 15
			control).	
00461	Uint16	read	Number of API users	4

3.5.1.14 Settings Reset API Endpoint: /settings

Reset the controller settings to their factory defaults.

Address	Туре	Operation	Description	FC
00006	Bit	write	Reset all settings	5

3.5.1.15 Factory Reset API Endpoint: /system/factory-reset

Reset the controller firmware to its factory default and initiate a reboot.

Address	Туре	Operation	Description	FC
00002	Bit	write	Trigger a factory reset of the firmware and the settings	5

3.5.1.16 Reboot the Device API Endpoint: /system/reboot

Trigger a reboot of all controller components.

Address	Type	Operation	Description	FC
00001	Bit	write	Trigger a reboot	5

3.5.1.17 Upgrade Recovery Firmware API Endpoint: /system/factory-reset

Replace the stored recovery image with the current system firmware. This is helpful if you want to update the recovery image to a more recent firmware version.

Address	Type	Operation	Description	FC
00003	Bit	write	Upgrade the recovery firmware to the currently running firmware version	5

3.5.1.18 RS232 Interface Configuration API Endpoint: /peripherals/rs232

Inspect or change the settings address of the controller for the RS232 interface. Some settings refer to the Modbus slave protocol. The Modbus slave ID is used for serial communication if more than one Modbus device is connected to the same bus. The frame format may be changed according to the needs of the Modbus master.

Address	Type	Operation	Description		FC
00430	Uint16	read / write	Baud rate of RS	232 interface	3,4,6,16
			Werte	Description	
			9600	0	
			19200	1	
			115200	2	
00431	Uint16	read / write	Protocol to be used for the RS232 interface		3,4,6,16
			Values	Description	
			eliza	0	
			modbus	1	
00432	Uint16	read / write	Slave ID to be u	sed for the Modbus protocol (1247)	3,4,6,16
00433	Uint16	read / write	Frame format to	be used for the Modbus protocol possible values	3,4,6,16
			Values	Description	
			rtu	0	
			ascii	1	

3.5.1.19 USB Interface Configuration API Endpoint: /peripherals/usb

Inspect or change the settings address of the controller for the USB interface. Some settings refer to the Modbus slave protocol. The Modbus slave ID is used for serial communication if more than one Modbus device is connected to the same bus. The frame format may be changed according to the needs of the Modbus master.

Adress	Type	Operation	Description		FC	
00440	Uint16	read / write	Protocol to be used for	Protocol to be used for the USB interface possible values		
			Values	Description		
			eliza	0		
			modbus	1		

00441	Uint16	read / write	Slave ID to be used for	3, 4, 6, 16	
00442	Uint16	read / write	Frame format to be use	3, 4, 6, 16	
			Values	Description	
			rtu	0	
			ascii	1	1

3.5.1.20 Data Format Test API Endpoint: None

Some registers respond with specified fixed values in order to allow clients to verify the correctness of the configured data format easily.

Address	Туре	Operation	Description	FC
00500	Uint16	read	A 16 bit integer value containing the number 1234.	4
00501	Float	read	A float value containing the number -1.0.	4
00503	Uint32	read	A 32 bit integer value containing the number 12345678.	4
00505	Uint64	read	A 64 bit integer value containing the number 123456789012.	4

