



Gerätehandbuch

IO-Link Master mit EtherCAT-Schnittstelle

CabinetLine

8 Ports

IP 20

**AL1930**

Firmware: 3.1.x

Deutsch

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbemerkung</b>	<b>5</b>
1.1	Rechtliche Hinweise.....	5
1.2	Zweck des Dokuments.....	5
1.3	Zeichenerklärung .....	6
1.4	Änderungshistorie .....	6
<b>2</b>	<b>Sicherheitshinweise</b>	<b>7</b>
2.1	Allgemein .....	7
2.2	Notwendige Vorkenntnisse .....	7
2.3	Sicherheitssymbole auf dem Gerät.....	7
2.4	IT-Sicherheit.....	8
<b>3</b>	<b>Bestimmungsgemäße Verwendung</b>	<b>9</b>
<b>4</b>	<b>Funktion</b>	<b>10</b>
4.1	Kommunikation, Parametrierung, Auswertung .....	11
4.1.1	IO-Link.....	11
4.1.2	EtherCAT.....	11
4.1.3	Internet of Things (IoT) .....	11
4.1.4	Sicherheitsmodus .....	11
4.1.5	Parametrierung.....	11
4.1.6	Optische Signalisierung.....	12
4.2	Digitale Eingänge .....	12
4.3	IO-Link-Versorgung.....	12
<b>5</b>	<b>Montage</b>	<b>13</b>
5.1	Gerät montieren .....	13
<b>6</b>	<b>Elektrischer Anschluss</b>	<b>14</b>
6.1	Hinweise.....	14
6.2	EtherCAT-Ports verbinden .....	15
6.3	IoT-Port .....	15
6.4	IO-Link Ports .....	16
6.4.1	IO-Link Devices für Class-A-Betrieb anschließen.....	16
6.4.2	IO-Link Devices für Class-B-Betrieb anschließen.....	17
6.5	Gerät anschließen.....	18
<b>7</b>	<b>Bedien- und Anzeigeelemente</b>	<b>19</b>
7.1	Übersicht .....	19
7.2	LED-Anzeigen .....	20
7.2.1	Status-LEDs .....	20
7.2.2	EtherCAT-Schnittstelle .....	20
7.2.3	IoT-Port .....	21
7.2.4	Spannungsversorgung .....	21
7.2.5	IO-Link Ports (Class A) .....	21

<b>8</b>	<b>Inbetriebnahme</b>	<b>22</b>
<b>9</b>	<b>Parametrierung</b>	<b>23</b>
9.1	LR DEVICE .....	24
9.1.1	Hinweise.....	25
9.1.2	IoT: IP-Einstellungen konfigurieren .....	26
9.1.3	IoT: Sicherheitsmodus konfigurieren .....	27
9.1.4	IoT: Zugriffsrechte konfigurieren.....	28
9.1.5	IoT: Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren.....	29
9.1.6	Fieldbus: EtherCAT-Schnittstelle konfigurieren .....	30
9.1.7	IO-Link-Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen.....	30
9.1.8	IO-Link-Ports: Betriebsart konfigurieren .....	31
9.1.9	IO-Link-Ports: Gerätevalidierung und Datenspeicherung einstellen .....	32
9.1.10	Info: Geräteinformationen zeigen .....	33
9.1.11	Firmware: Gerät auf Werkseinstellungen rücksetzen .....	33
9.1.12	Firmware: Gerät neu starten.....	33
9.1.13	IO-Link Devices konfigurieren .....	34
9.2	ifm-IoT-Core .....	35
9.2.1	Hinweise für Programmierer.....	36
9.2.2	Erste Schritte.....	41
9.2.3	Allgemeine Funktionen .....	41
9.2.4	IoT: Zugriffsrechte konfigurieren.....	45
9.2.5	IoT: IP-Einstellungen konfigurieren .....	45
9.2.6	IoT: Sicherheitsmodus konfigurieren .....	45
9.2.7	Fieldbus: Fieldbus-Schnittstelle konfigurieren .....	49
9.2.8	IO-Link Ports: Betriebsart Pin 4 (US) einstellen.....	49
9.2.9	IO-Link Ports: Gerätevalidierung und Datenspeicherung konfigurieren.....	50
9.2.10	IO-Link Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren ..	52
9.2.11	IO-Link Ports: Prozessdaten lesen / schreiben.....	52
9.2.12	IO-Link Ports: Port-Events anzeigen .....	55
9.2.13	IO-Link Devices: Auf Parameter zugreifen .....	55
9.2.14	IO-Link Devices: Geräteinformationen lesen und schreiben.....	57
9.2.15	IO-Link Devices: IO-Link Events anzeigen .....	57
9.2.16	Gateway: Gerät rücksetzen, neu starten und lokalisieren .....	57
9.2.17	Gateway: Geräteinformationen lesen .....	58
9.2.18	Gateway: Zustands- und Diagnoseinformationen lesen .....	58
9.2.19	Gateway: Firmware aktualisieren .....	59
9.2.20	Gateway: Anwendungskennung einstellen.....	60
9.2.21	Benachrichtigungen abonnieren.....	61
9.2.22	Web Socket nutzen .....	65
9.2.23	MQTT-Unterstützung.....	67
9.2.24	IoT-Core-Visualizer nutzen.....	71
9.3	EtherCAT .....	78
9.3.1	ESI-Datei installieren .....	78
9.3.2	IO-Link Master in TwinCAT-Projekt einbinden.....	78
9.3.3	IO-Link-Master konfigurieren .....	79
9.3.4	IO-Link-Ports konfigurieren.....	79
9.3.5	Zyklische Prozessdaten konfigurieren .....	80
9.3.6	Zyklische Prozessdaten lesen und schreiben.....	81
9.3.7	Diagnose- und Statusinformationen lesen.....	82
9.3.8	IO-Link Events lesen .....	82
9.3.9	IO-Link Devices konfigurieren .....	83
9.3.10	EtherCAT: Hinweise für Programmierer .....	84

<b>10</b>	<b>Betrieb</b>	<b>87</b>
10.1	Web-based Management nutzen.....	87
<b>11</b>	<b>Wartung, Instandsetzung und Entsorgung</b>	<b>88</b>
11.1	Reinigung .....	88
11.2	Firmware aktualisieren.....	88
11.3	IO-Link Device tauschen.....	88
<b>12</b>	<b>Werkseinstellungen</b>	<b>89</b>
<b>13</b>	<b>Zubehör</b>	<b>90</b>
<b>14</b>	<b>Anhang</b>	<b>91</b>
14.1	Technische Daten .....	92
14.1.1	Einsatzbereich.....	92
14.1.2	Elektrische Daten .....	92
14.1.3	Eingänge / Ausgänge .....	92
14.1.4	Eingänge .....	93
14.1.5	Ausgänge .....	93
14.1.6	Schnittstellen .....	93
14.1.7	Umgebungsbedingungen .....	94
14.1.8	Zulassungen / Prüfungen .....	94
14.1.9	Mechanische Daten.....	94
14.1.10	Elektrischer Anschluss .....	95
14.2	EtherCAT .....	96
14.2.1	Parameterdaten.....	97
14.2.2	Zyklische Daten.....	98
14.2.3	Azyklische Daten.....	99
14.2.4	Events .....	108
14.3	ifm-LoT-Core .....	112
14.3.1	Übersicht: LoT-Profile.....	113
14.3.2	Übersicht: LoT-Typen.....	120
14.3.3	Übersicht: LoT-Dienste.....	121
<b>15</b>	<b>Index</b>	<b>135</b>

# 1 Vorbemerkung

## Inhalt

Rechtliche Hinweise .....	5
Zweck des Dokuments .....	5
Zeichenerklärung .....	6
Änderungshistorie.....	6

14801

## 1.1 Rechtliche Hinweise

1631

© Alle Rechte bei ifm electronic gmbh. Vervielfältigung und Verwertung dieser Anleitung, auch auszugsweise, nur mit Zustimmung der ifm electronic gmbh.

Alle auf unseren Seiten verwendeten Produktnamen, -Bilder, Unternehmen oder sonstige Marken sind Eigentum der jeweiligen Rechteinhaber:

- AS-i ist Eigentum der AS-International Association, (→ [www.as-interface.net](http://www.as-interface.net))
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ [www.can-cia.org](http://www.can-cia.org))
- CODESYS™ ist Eigentum der CODESYS GmbH, Deutschland (→ [www.codesys.com](http://www.codesys.com))
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ [www.odva.org](http://www.odva.org))
- EtherNet/IP® ist Eigentum der → ODVA™
- EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland
- IO-Link® ist Eigentum der → PROFIBUS Nutzerorganisation e.V., Deutschland (→ [www.io-link.com](http://www.io-link.com))
- ISOBUS ist Eigentum der AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ [www.aef-online.org](http://www.aef-online.org))
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ [www.microsoft.com](http://www.microsoft.com))
- Modbus® ist Eigentum der Schneider Electric SE, Frankreich (→ [www.schneider-electric.com](http://www.schneider-electric.com))
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ [www.profibus.com](http://www.profibus.com))
- PROFINET® ist Eigentum der → PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® ist Eigentum der → Microsoft Corporation, USA

## 1.2 Zweck des Dokuments

22044

Dieses Dokument gilt für Geräte des Typs „IO-Link Master mit EtherCAT-Schnittstelle CabinetLine 8 Port IP 20“ (Art.-Nr.: AL1930).

Es ist Bestandteil des Gerätes und enthält Angaben zum korrekten Umgang mit dem Produkt.

- Dieses Dokument vor dem Einsatz des Gerätes lesen.
- Dieses Dokument während der Einsatzdauer des Gerätes aufbewahren.

## 1.3 Zeichenerklärung

15989



### WARNUNG

Warnung vor schweren Personenschäden.  
Tod oder schwere irreversible Verletzungen sind möglich.



### VORSICHT

Warnung vor Personenschäden.  
Leichte reversible Verletzungen sind möglich.



### ACHTUNG

Warnung vor Sachschäden



Wichtiger Hinweis  
Fehlfunktionen oder Störungen sind bei Nichtbeachtung möglich



Information  
Ergänzender Hinweis

► ... Handlungsaufforderung

> ... Reaktion, Ergebnis

→ ... "siehe"

[abc](#) Querverweis

123 Dezimalzahl

0x123 Hexadezimalzahl

0b010 Binärzahl

[...] Bezeichnung von Tasten, Schaltflächen oder Anzeigen

## 1.4 Änderungshistorie

59017

Ausgabe	Thema	Datum
00	Neuerstellung des Dokuments	04 / 2019
01	Korrektur: Technische Daten - Strombelastbarkeit je Ausgang	09 / 2019
02	<ul style="list-style-type: none"> <li>Hinzugefügt: Neue IoT-Core-Funktionen</li> <li>Hinzugefügt: IoT Core Visualizer</li> <li>Korrektur: Beschreibung des IoT-Core-Dienstes getsubscriptioninfo</li> </ul>	10 / 2020
03	Entfernt: ifm IoT Core – DNS-Unterstützung	10 / 2021

## 2 Sicherheitshinweise

### Inhalt

Allgemein .....	7
Notwendige Vorkenntnisse .....	7
Sicherheitssymbole auf dem Gerät .....	7
IT-Sicherheit .....	8

213

### 2.1 Allgemein

2272

- Das beschriebene Gerät wird als Teilkomponente in einem System verbaut. Die Sicherheit dieses Systems liegt in der Verantwortung des Erstellers. Der Systemersteller ist verpflichtet, eine Risikobeurteilung durchzuführen und daraus eine Dokumentation nach den gesetzlichen und normativen Anforderungen für den Betreiber und den Benutzer des Systems zu erstellen und beizulegen. Diese muss alle erforderlichen Informationen und Sicherheitshinweise für Betreiber, Benutzer und ggf. vom Systemersteller autorisiertes Servicepersonal beinhalten.
- Dieses Dokument vor Inbetriebnahme des Produktes lesen und während der Einsatzdauer aufbewahren.
- Das Produkt muss sich uneingeschränkt für die betreffenden Applikationen und Umgebungsbedingungen eignen.
- Das Produkt nur bestimmungsgemäß verwenden (→ **Bestimmungsgemäße Verwendung** (→ S. 9)).
- Die Missachtung von Anwendungshinweisen oder technischen Angaben kann zu Sach- und / oder Personenschäden führen.
- Für Folgen durch Eingriffe in das Gerät oder Fehlgebrauch durch den Betreiber übernimmt der Hersteller keine Haftung und keine Gewährleistung.
- Montage, elektrischer Anschluss, Inbetriebnahme, Programmierung, Konfiguration, Bedienung und Wartung des Produktes darf nur für die jeweilige Tätigkeit ausgebildetes, autorisiertes Fachpersonal durchführen.
- Geräte und Kabel wirksam vor Beschädigung schützen.

### 2.2 Notwendige Vorkenntnisse

22046

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung eines Produkts verursachen kann.

### 2.3 Sicherheitssymbole auf dem Gerät

15021



Allgemeiner Warnhinweis

Hinweise in Kapitel "Elektrischer Anschluss" beachten (→ **Elektrischer Anschluss** (→ S. 14))!

## 2.4 IT-Sicherheit

51595

### **ACHTUNG!**

Bei Betrieb des Gerätes in einer ungeschützten Netzwerkumgebung.

- > Unzulässiger Lese- oder Schreibzugriff auf Daten möglich.
- > Unzulässige Beeinflussung der Gerätefunktion möglich.
- ▶ Zugriffsmöglichkeiten auf das Gerät prüfen und einschränken:
  - Zugriff auf autorisierte Nutzer beschränken.
  - Sichere Verfahren wählen, um sich mit dem Gerät zu verbinden (z. B. VPN).
  - Verschlüsselte Datenübertragung nutzen (z. B. https / TLS).



### 3 Bestimmungsgemäße Verwendung

22671

Der IO-Link-Master dient als Gateway zwischen intelligenten IO-Link-Devices und dem EtherCAT-Netzwerk. Das Gerät ist für den Einsatz im Anlagenbau als Schaltschrankmodul konzipiert.

- Das Gerät nur innerhalb der Grenzen der technischen Daten einsetzen (→ **Technische Daten** (→ S. [92](#))).

## 4 Funktion

### Inhalt

Kommunikation, Parametrierung, Auswertung .....	11
Digitale Eingänge .....	12
IO-Link-Versorgung .....	12

7482

## 4.1 Kommunikation, Parametrierung, Auswertung

Inhalt	
IO-Link .....	11
EtherCAT .....	11
Internet of Things (IoT) .....	11
Sicherheitsmodus .....	11
Parametrierung .....	11
Optische Signalisierung .....	12

7485

### 4.1.1 IO-Link

7773

Das Gerät stellt folgende IO-Link-Funktionen bereit:

- IO-Link Master (IO-Link Revision 1.0 und 1.1)
- 8 IO-Link Ports für den Anschluss von IO-Link Devices
- Bereitstellung von Prozessdaten der angeschlossenen IO-Link Devices für Monitoring-Software LR SMARTOBSERVER (→ [www.ifm.com](http://www.ifm.com))

### 4.1.2 EtherCAT

2259

Das Gerät bietet folgende EtherCAT-Funktionen:

- Bereitstellung der Funktionen eines EtherCAT Slave
- 2-Port-Switch für den Zugriff auf die EtherCAT-Schnittstelle (X21/X22)
- Gateway für Übertragung der Prozess- und Parameterdaten zwischen den angeschlossenen IO-Link Devices und der übergeordneten EtherCAT-Steuerung

### 4.1.3 Internet of Things (IoT)

22408

Das Gerät bietet folgende IoT-Funktionen:

- Gateway für Übertragung der Prozess-, Parameter- und Monitoringdaten zwischen IO-Link Master / IO-Link Devices und der IT-Netzwerkebene
- REST-API für Zugriff auf Prozess- und Parameterdaten
- Unterstützte Protokolle: TCP/IP JSON, MQTT

### 4.1.4 Sicherheitsmodus

34083

Die IoT-Schnittstelle bietet folgende optionale Sicherheitsfunktionen:

- Sicherer Datentransport durch verschlüsselte Verbindung (Secure Layer Transport - TLS)
- Zugriffsschutz per Authentifizierung

### 4.1.5 Parametrierung

7284

Das Gerät bietet folgende Konfigurationsoptionen:

- Parametrierung des IO-Link Masters des AL1930 mit Parametriersoftware LR DEVICE, EtherCAT-Projektierungssoftware oder ifm-IoT-Core-Diensten
- Parametrierung der angeschlossenen IO-Link Devices (Sensoren, Aktuatoren) mit Parametriersoftware LR DEVICE, EtherCAT-Projektierungssoftware oder ifm-IoT-Core-Diensten

- Speicherung von Parametersätzen der angeschlossenen IO-Link Devices für automatische Wiederherstellung (Data Storage)

#### **4.1.6 Optische Signalisierung**

7772

Das Gerät verfügt über folgende optische Anzeigen:

- Status- und Fehleranzeige des Gateways, der EtherCAT-Verbindung und des Systems
- Statusanzeige der Spannungsversorgung
- Status- und Aktivitätsanzeige der Ethernet-Verbindung
- Status-, Fehler- und Kurzschluss-/Überlastanzeige der IO-Link-Ports

### **4.2 Digitale Eingänge**

7584

Das Gerät verfügt über 8 zusätzliche digitale Eingänge (Typ 2 nach EN 61131-2).

Die digitalen Eingänge liegen an Klemme 2 der Ports X01...X08.

Die digitalen Eingänge werden von der Versorgungsspannung US gespeist. Sie beziehen sich auf das Potential von US (Klemme 3).

### **4.3 IO-Link-Versorgung**

7623

Das Gerät verfügt über 8 Versorgungen für IO-Link Devices.

Die IO-Link Ports X01...X08 sind Class-A-Ports.

Jede Versorgung verfügt über eine Kurzschlussüberwachung.

Das Gerät gewährleistet den Brandschutz für angeschlossene IO-Link Devices durch Bereitstellung eines energiebegrenzten Stromkreises an den IO-Link Ports (nach IEC61010-1 und Class 2 nach UL1310).

## 5 Montage

### Inhalt

Gerät montieren.....	13
----------------------	----

22016

### 5.1 Gerät montieren

16676



- ▶ Anlage während der Montage spannungsfrei schalten.  
Das Gerät enthält Bauelemente, die durch elektrostatische Entladung beschädigt oder zerstört werden können.
- ▶ Beim Umgang mit dem Gerät die notwendigen Sicherheitsmaßnahmen gegen elektrostatische Entladung (ESD) beachten.
- ▶ Das Gerät nur an einer geerdeten Tragschiene betreiben.
  
- ▶ Das Gerät in einen Schaltschrank der Mindestschutzart IP 54 einbauen. Der Schaltschrank soll in Übereinstimmung mit den Vorschriften der lokalen und nationalen Bestimmungen installiert werden.
- ▶ Das Gerät senkrecht an einer 35 mm-Profilschiene befestigen.
- ▶ Ausreichend Platz zu Boden und Deckel des Schaltschranks sowie benachbarten Geräten lassen, um Luftzirkulation zu ermöglichen und unzulässige Erwärmung zu vermeiden.

## 6 Elektrischer Anschluss

### Inhalt

Hinweise .....	14
EtherCAT-Ports verbinden .....	15
IoT-Port .....	15
IO-Link Ports .....	16
Gerät anschließen .....	18

22017

### 6.1 Hinweise

23709



Das Gerät darf nur von einer Elektrofachkraft installiert werden.

- Die nationalen und internationalen Vorschriften zur Errichtung elektrotechnischer Anlagen befolgen.

Gerät ist nur für den Betrieb an SELV/PELV-Spannungen geeignet.

- Hinweise zur IO-Link-Beschaltung beachten!

Die IP-Schutzart des Gesamtsystems hängt ab von den Schutzarten der einzelnen Geräte und der genutzten Verbindungselemente.

Für UL-Anwendungen:

- Für den Anschluss des IO-Link Masters nur Kabel im Bereich AWG 26 bis 12 und einem Mindesttemperaturbereich von 75 °C verwenden.

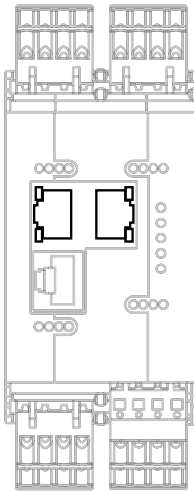
Anschlussbelegung: → **Technische Daten** (→ S. [92](#))

Die Stromkreise sind untereinander und zu berührbaren Oberflächen des Geräts getrennt mit Basisisolierung nach EN61010-1 (Sekundärstromkreis mit maximal 30 V DC, abgeleitet von Netzstromkreis bis 300 V der Überspannungskategorie II).

Die Kommunikationsschnittstellen sind untereinander und zu berührbaren Oberflächen des Geräts getrennt mit Basisisolierung nach EN61010-1 (Sekundärstromkreis mit maximal 30 V DC, abgeleitet von Netzstromkreis bis 300 V der Überspannungskategorie II). Sie sind ausgelegt für Netzwerkumgebung 0 nach IEC TR62102.

## 6.2 EtherCAT-Ports verbinden

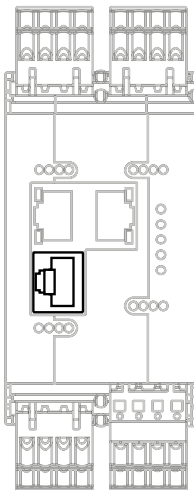
17858



- ▶ Gerät über Buchsen X21 und/oder X22 mit dem EtherCAT-Netzwerk verbinden.
- ▶ Für den Anschluss Steckverbindungen mit mindestens der Schutzart IP 20 verwenden (→ **Zubehör** (→ S. [90](#))).

## 6.3 IoT-Port

17859



- ▶ Gerät über Buchse X23 mit dem IT-Netzwerk verbinden (z. B. Laptop/PC mit Parametriersoftware LR DEVICE, Laptop/PC mit Monitoring-Software LR SMARTOBSERVER, PC/Laptop mit http-Request-fähiger Software).
- ▶ Für den Anschluss Steckverbindungen mit mindestens der Schutzart IP 20 verwenden (→ **Zubehör** (→ S. [90](#))).

## 6.4 IO-Link Ports

10896

Die IO-Link Ports des Geräts erfüllen die Anforderungen der IO-Link-Spezifikation 1.0 bis 1.1.2.

► Hinweise zur IO-Link Beschaltung beachten!



### WARNUNG

Zuführung von Energie in die IO-Link Ports des IO-Link Masters

> Brandgefahr!

► Ein- und Rückspeisung von Energie in die IO-Link Ports verhindern.

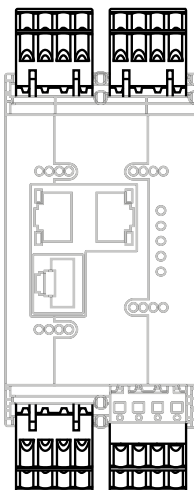
► Vor Inbetriebnahme den korrekten Anschluss der Versorgungsverbindungen prüfen.

### 6.4.1 IO-Link Devices für Class-A-Betrieb anschließen

17532

Hinweise zur Beschaltung:

- Die Stromversorgung der angeschlossenen IO-Link Devices darf ausschließlich über den IO-Link Master erfolgen.
- Die zusätzlichen digitalen Eingänge der IO-Link Ports X01...X08 (Klemme 2) verfügen über ein Typ-2-Verhalten nach Norm EN61131-2. Die angeschlossene Elektronik muss dafür elektrisch ausgelegt sein.



- IO-Link Devices mit den Ports X01...X08 verbinden.
  - Maximale Leitungslänge pro IO-Link Port: 20 m
- Für den Anschluss nur Verbindungsleitungen mit mindestens der Schutzart IP 20 verwenden.

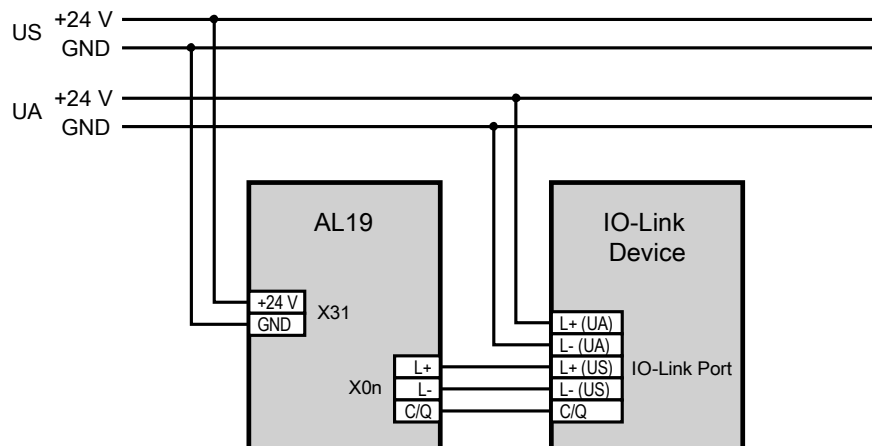


## 6.4.2 IO-Link Devices für Class-B-Betrieb anschließen

12104

Hinweise zur Beschaltung:

- Für den Class-B-Betrieb muss dem IO-Link Device eine zusätzliche Hilfsspannung UA zugeführt werden.
- Anschlussbild:



- Erlaubte maximale Stromstärke für UA: 4 A

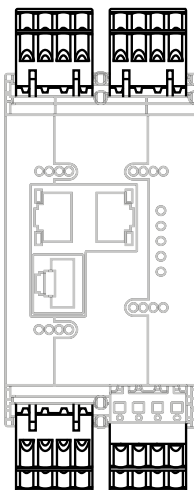


### WARNUNG

Nichtbeachtung der galvanischen Trennung der Stromkreise

> Brandgefahr!

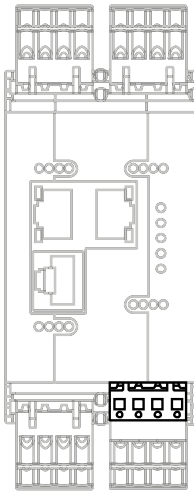
- ▶ Sicherstellen, dass externe Versorgung UA vom Stromkreis des IO-Link Masters galvanisch getrennt ist unter Beachtung von Basisisolierung (nach IEC 61010-1, Sekundärstromkreis mit maximal 30 V DC, abgeleitet von Netzstromkreis bis 300 V der Überspannungskategorie II).
- ▶ Sicherstellen, dass IO-Link Devices und Verbindungstechnik die galvanische Trennung unterstützen.



- ▶ IO-Link Devices mit den Ports X01...X08 verbinden.
  - Maximale Leitungslänge pro IO-Link-Port: 20 m
- ▶ IO-Link Devices anschließen an UA mit 24 V DC (20...30 V SELV/PELV).
- ▶ Für den Anschluss der IO-Link Devices Verbindungsleitungen mit mindestens der Schutzart IP 20 verwenden.

## 6.5 Gerät anschließen

17851



- ▶ Anlage spannungsfrei schalten.
- ▶ IO-Link Master über den Port X31 anschließen an US mit 24 V DC (20...30 V SELV/PELV).
  - Empfohlene maximale Kabellänge: 25 m
- ▶ Für den Anschluss des Geräts Verbindungsleitungen mit mindestens der Schutzart IP 20 verwenden.



Bei Leitungslängen größer 25 m den eintretenden Spannungsabfall und die notwendige minimale Versorgungsspannung von 20 V beachten!

## 7 Bedien- und Anzeigeelemente

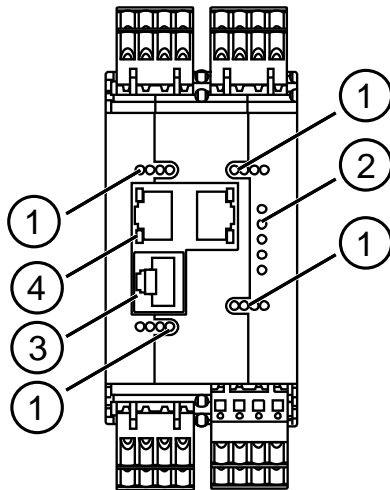
### Inhalt

Übersicht.....	19
LED-Anzeigen .....	20

5440

### 7.1 Übersicht

7685



- ① Status-LEDs IOL und DI des IO-Link-Ports (X01...X08)  
(→ **IO-Link Ports (Class A)** (→ S. [21](#)))
- ② Status-LED PWR der Spannungsversorgung (X31)  
(→ **Spannungsversorgung** (→ S. [21](#)))  
Status-LEDs RDY, RUN und ERR (→ **Status-LEDs**  
(→ S. [20](#)))  
Status-LED IoT des IoT-Ports (X23) (→ **IoT-Port**  
(→ S. [21](#)))
- ③ Status-LEDs LNK und ACT des IoT-Ports (X23)  
(→ **IoT-Port** (→ S. [21](#)))
- ④ Status-LED L/A der EtherCAT-Ports 1 (X21) und 2 (X22) (→ **EtherCAT-Schnittstelle** (→ S. [20](#)))

## 7.2 LED-Anzeigen

22024

Das Gerät verfügt über folgende LED-Anzeigen.

### 7.2.1 Status-LEDs

1986

Die LED mit der Bezeichnung RDY zeigt den Status des Gateways.

Die LED mit der Bezeichnung RUN zeigt den aktuellen Zustand der EtherCAT-Zustandsmaschine.

Die LED mit der Bezeichnung ERR zeigt auftretende Fehler an.

Status-LED			Bedeutung
RDY	grün	ein	Status: OK
		blinkt 5 Hz	Status: Fehler
		blinkt (200 ms ein, 800 ms aus)	Status: Firmware-Update läuft
		aus	Status: Gateway nicht aktiv oder Gateway startet neu
RUN	grün	ein	Gerät im Zustand OPERATIONAL
		blinkt 2,5 Hz	Gerät im Zustand PRE-OPERATIONAL
		blinkt (200 ms ein, 1000 ms aus)	Gerät im Zustand SAFE-OPERATIONAL
		blinkt 10 Hz	Gerät startet und ist noch nicht im Zustand INIT oder Gerät ist im Zustand BOOTSTRAP
		aus	Gerät im Zustand INIT
ERR	rot	ein	Fehler in Anwendungssteuerung
		blinkt 10 Hz	Boot-Fehler
		blinkt (200 ms ein, 200 ms aus, 200 ms ein, 1000 ms aus)	Watchdog-Fehler (EtherCAT oder Prozessdaten)
		blinkt (200 ms ein, 1000 ms aus)	Lokaler Fehler
		blinkt 2,5 Hz	Ungültige Konfiguration
		aus	kein Fehler

### 7.2.2 EtherCAT-Schnittstelle

17852

Jede EtherCAT-Schnittstelle (X21, X22) verfügt über 1 LED mit der Bezeichnung L/A. Die LED zeigt den Status der Ethernet-Verbindung.

Status-LED			Bedeutung
L/A	grün	ein	Ethernet-Verbindung hergestellt
		blinkt	Es werden Daten über die Ethernet-Schnittstelle übertragen.
		aus	keine Ethernet-Verbindung

## 7.2.3 IoT-Port

17722

Der IoT-Port verfügt über 3 LEDs mit der Bezeichnung LNK, ACT und IoT. Die LEDs zeigen den Status der Ethernet-Verbindung und die Geräteidentifizierung.

Status-LED			Bedeutung
LNK	grün	ein	Ethernet-Verbindung hergestellt
		aus	keine Ethernet-Verbindung
ACT	gelb	blinkt	Es werden Daten über die Ethernet-Schnittstelle übertragen.
		aus	keine Datenübertragung
IoT	grün	blinkt	Geräteidentifizierung aktiv

## 7.2.4 Spannungsversorgung

17856

Die Schnittstelle zur Spannungsversorgung (X31) verfügt über die LED mit der Bezeichnung PWR. Die LED zeigt den Status der Spannungsversorgung.

Status-LED			Bedeutung
PWR	grün	ein	Versorgungsspannung $U_s$ liegt an
		aus	keine Versorgungsspannung an oder anliegende Versorgungsspannung zu niedrig

## 7.2.5 IO-Link Ports (Class A)

22029

Jeder IO-Link Port Class A verfügt über 2 LEDs mit der Bezeichnung IOL und DI. Die LEDs zeigen den Status des IO-Link Ports.

Status-LED			Bedeutung
IOL	gelb	aus	Port als DI/DO konfiguriert: Klemme 4 (C/Q) = OFF
		ein	Port als DI/DO konfiguriert: Klemme 4 (C/Q) = ON
	grün	blinkt 1 Hz	Port als IO-Link konfiguriert: kein IO-Link Device gefunden
		blinkt 2 Hz	Port als IO-Link konfiguriert: Zustand PREOPERATE
		ein	Port als IO-Link konfiguriert: Zustand OPERATE
	rot	blinkt 2 Hz	Port-Konfigurationsfehler oder Kurzschluss / Überlast an US
		ein	Übertragungsfehler
DI	gelb	aus	Digitaler Eingang : Klemme 2 = OFF
		ein	Digitaler Eingang: Klemme 2 = ON

## 8 Inbetriebnahme

Durch Einschalten der Versorgungsspannung startet der AL1930 mit den Werkseinstellungen. Die Anzeigelemente signalisieren den aktuellen Betriebszustand (→ **Bedien- und Anzeigeelemente** (→ S. [19](#))).

Um die Parametrierung des AL1930 zu ermöglichen, müssen die IoT-Schnittstelle und / oder die Feldbus-Schnittstelle der Netzwerkumgebung entsprechend konfiguriert werden.

- ▶ Feldbus-Schnittstelle konfigurieren (→ **Feldbus: Feldbus-Schnittstelle konfigurieren** (→ S. [49](#), **"IO-Link Master in TwinCAT-Projekt einbinden"** → S. [78](#))).
- ▶ IoT-Schnittstelle konfigurieren konfigurieren (→ **IoT: IP-Einstellungen konfigurieren** (→ S. [26](#)) oder → **IoT: IP-Einstellungen konfigurieren** (→ S. [45](#))).
- > IoT-/Feldbus-Schnittstelle hat gültige IP-Einstellungen.
- > Anwender kann AL1930 parametrieren.

Weitere Schritte:

- Optional: Firmware des AL1930 aktualisieren (→ **Firmware aktualisieren** (→ S. [88](#))).
- AL1930 parametrieren (→ **Parametrierung** (→ S. [23](#))).

## 9 Parametrierung

### Inhalt

LR DEVICE .....	24
ifm-IoT-Core .....	35
EtherCAT .....	78

22367

## 9.1 LR DEVICE

### Inhalt

Hinweise .....	25
IoT: IP-Einstellungen konfigurieren .....	26
IoT: Sicherheitsmodus konfigurieren .....	27
IoT: Zugriffsrechte konfigurieren .....	28
IoT: Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren .....	29
Fieldbus: EtherCAT-Schnittstelle konfigurieren.....	30
IO-Link-Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen .....	30
IO-Link-Ports: Betriebsart konfigurieren .....	31
IO-Link-Ports: Gerätevalidierung und Datenspeicherung einstellen .....	32
Info: Geräteinformationen zeigen .....	33
Firmware: Gerät auf Werkseinstellungen rücksetzen .....	33
Firmware: Gerät neu starten.....	33
IO-Link Devices konfigurieren .....	34

22822

Im Auslieferungszustand ist der AL1930 mit den Werkseinstellungen konfiguriert  
(→ **Werkseinstellungen** (→ S. [89](#))).

Benötigte Software: LR DEVICE (1.5.0.x oder höher) (Art.-Nr.: QA0011/QA0012)



## 9.1.1 Hinweise

### Inhalt

Offline-Parametrierung .....	25
Rückfallwerte der IO-Link Ports .....	25

22369

### Offline-Parametrierung

22405

Der AL1930 unterstützt die Offline-Parametrierung. Dabei erstellt der Anwender eine Konfiguration für den IO-Link Master und die angeschlossenen IO-Link Devices, ohne mit dem AL1930 verbunden zu sein (OFFLINE-Modus). Die so erstellte Konfiguration kann als Datei (\*.Irp) gespeichert und später auf den AL1930 geladen und aktiviert werden.



Weiter Infos zur Offline-Parametrierung: → Bedienungsanleitung LR DEVICE

### Rückfallwerte der IO-Link Ports

13423



Der AL1930 besitzt keine Failsafe-Funktion für die Ausgänge der IO-Link Ports. Bei einer Unterbrechung der Feldbusverbindung werden die zuletzt genutzten Ausgangswerte geschrieben und als ungültig gekennzeichnet.

## 9.1.2 IoT: IP-Einstellungen konfigurieren

Für den Zugriff auf den IO-Link Master über die IT-Infrastruktur muss der Anwender die IP-Einstellungen der IoT-Schnittstelle einstellen.



Um die IP-Einstellungen mit DHCP zu konfigurieren, muss im IT-Netzwerk ein DHCP-Server aktiv sein. Ist kein DHCP-Server im IT-Netzwerk erreichbar, wird dem IoT-Port mit dem Zeroconfig-Protokoll automatisch eine IP-Adresse zugewiesen (Adressbereich: → **Werkseinstellungen** (→ S. [89](#))).

Um die IP-Einstellungen der IoT-Schnittstelle zu konfigurieren:

- ▶ Menü [IoT] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[DHCP]	DHCP-Client des Geräts aktivieren / deaktivieren	[Static IP]	IP-Einstellungen werden vom Anwender eingestellt
		[DHCP]	IP-Einstellungen werden von einem DHCP-Server im Netzwerk eingestellt.
[IP address]*	IP-Adresse des IoT-Ports	Werkseinstellung: 169.254.X.X	
[Subnet mask]*	Subnetzmaske des Ethernet-Netzwerks	Werkseinstellung: 255.255.0.0	
[Default gateway IP address]*	IP-Adresse des Netzwerk-Gateways	Werkseinstellung: 0.0.0.0	
[MAC address]	MAC-Adresse des IoT-Ports	Wert ist fest eingestellt.	

\* ... nur editierbar, wenn Parameter [DHCP] = [Static IP]

- ▶ Geänderte Werte auf dem Gerät speichern.

### 9.1.3 IoT: Sicherheitsmodus konfigurieren

Die IoT-Schnittstelle des IO-Link Master bietet einen Sicherheitsmodus. Er ermöglicht die sichere Datenübertragung per Transportverschlüsselung sowie die Einschränkung des Zugriffs auf IO-Link Master und IO-Link Devices per Nutzerauthentifizierung.

Um den Sicherheitsmodus zu konfigurieren:

- ▶ Menü [IoT] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[Security mode HTTPS]	Sicherheitsmodus einstellen	[Disabled]	Sicherheitsmodus deaktiviert
		[Enabled]	Sicherheitsmodus aktiviert
[Security password]	Passwort Hinweis: Das gesetzte Passwort wird nicht angezeigt.		

- ▶ Geänderte Werte auf dem Gerät speichern.



Der Sicherheitsmodus schützt nur den Zugriff auf das Gerät über die IoT-Schnittstelle.  
Der Nutzername "administrator" ist nicht änderbar.



Der Sicherheitsmodus kann aktiviert werden, ohne das Passwort zu setzen. Beim Versuch, auf das Gerät zu schreiben, verlangt LR DEVICE die Eingabe und Bestätigung des Passworts.

Nach der Eingabe des Passworts kann der Nutzer uneingeschränkt auf IO-Link Master und angeschlossene IO-Link Devices zugreifen. Eine erneute Abfrage des Passworts erfolgt erst wieder, wenn die aktuelle LR DEVICE-Sitzung beendet wurde (z. B. nach einem Neustart des LR DEVICE).

Um das gesetzte Passwort zu ändern:

- ▶ Mit gültigem Passwort anmelden.
- ▶ In Feld [Security password] das neue Passwort eingeben.
- ▶ Änderungen auf das Gerät schreiben.
- > Neues Passwort ist gesetzt.

## 9.1.4 IoT: Zugriffsrechte konfigurieren

Die Zugriffsrechte regeln, welche Instanz die Parameterdaten, Prozessdaten und Ereignis-/Diagnosemeldungen lesen und / oder schreiben darf.

Um die Zugriffsrechte auf den IO-Link Master zu konfigurieren:

- ▶ Menü [IoT] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[Access Rights]	Zugriffsrechte auf Parameterdaten, Prozessdaten und Ereignis-/Diagnosemeldungen des IO-Link Masters sowie der angeschlossenen IO-Link Devices	[EtherCAT + IoT]*	<ul style="list-style-type: none"> <li>▪ EtherCAT und IoT-Core haben Lese- und Schreibrechte auf Parameter und Prozessdaten</li> <li>▪ EtherCAT und IoT-Core haben Leserechte auf Ereignisse/Alarmer</li> </ul>
		[EtherCAT + IoT (read-only)]	<ul style="list-style-type: none"> <li>▪ EtherCAT hat Lese- und Schreibrechte auf Parameter und Prozessdaten</li> <li>▪ EtherCAT hat Leserechte auf Ereignisse/Alarmer</li> <li>▪ IoT-Core hat Leserechte auf Parameter, Prozessdaten und Ereignisse/Alarmer</li> </ul>
		[IoT only]	<ul style="list-style-type: none"> <li>▪ IoT-Core hat Lese- und Schreibrechte auf Parameter und Prozessdaten</li> <li>▪ IoT hat Leserechte auf Ereignisse/Alarmer</li> <li>▪ EtherCAT hat keine Zugriffsrechte</li> </ul>

\* ... Werkeinstellung

- ▶ Geänderte Werte auf dem Gerät speichern.



Wenn in LR DEVICE und EtherCAT-Projektierungssoftware der Parameter [Access Rights] = [EtherCAT + IoT], dann gelten immer die in der EtherCAT-Projektierungssoftware eingestellten Parameterwerte.

Wenn in LR DEVICE der Parameter [Access Rights] = [IoT only], dann in EtherCAT-Projektierungssoftware den Parameter [Access Rights] = [Keep settings] setzen.

Wenn in LR DEVICE der Parameter [Access Rights] = [EtherCAT + IoT (read-only)], dann ist der Schreibzugriff auf die Gerätekonfiguration über LR DEVICE und IoT-Core-Dienste gesperrt. Um den Schreibzugriff wieder zu ermöglichen, den Parameter über Feldbus-Projektierungssoftware auf [EtherCAT + IoT] setzen.

Änderungen des Parameters [Access Rights] sind erst wirksam nach einem Neustart des IO-Link Masters (→ **Firmware: Gerät neu starten** (→ S. 33)).

## 9.1.5 IoT: Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren

16552

Um die Übertragung von Prozessdaten vom IO-Link Master an LR AGENT oder LR SMARTOBSERVER zu ermöglichen, muss die Schnittstelle entsprechend konfiguriert werden.

- ▶ Menü [IoT] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[IP address LR Agent or SMARTOBSERVER]	IP-Adresse des LR AGENT oder LR SMARTOBSERVER	Werkseinstellung: 255.255.255.255	
[Port LR Agent or SMARTOBSERVER]	Port-Nummer des LR AGENT oder LR SMARTOBSERVER, an die Prozessdaten gesendet werden	0 ... 65535	Werkseinstellung:: 35100
[Interval LR Agent or SMARTOBSERVER]	Zykluszeit für die Übertragung der Prozessdaten zum LR AGENT oder LR SMARTOBSERVER (Wert in ms)	[Off]	keine Übertragung
		500 ... 2147483647	500 ms ... 2147483647 ms
[Application Tag]	Quellenbezeichner des IO-Link Masters in der Struktur des LR AGENT oder LR SMARTOBSERVER (String32)	Werkseinstellung: AL1930	



Nach der Änderung des Parameters [Port LR Agent or SMARTOBSERVER] oder [Application Tag] kann es 120 Sekunden dauern, bis das Gerät erneut eine TCP-Verbindung aufbaut.

Um die Verzögerung zu vermeiden:

- ▶ Nach der Änderung des Parameters den IO-Link Master neu starten.
- ▶ Geänderte Werte auf dem Gerät speichern.

## 9.1.6 Fieldbus: EtherCAT-Schnittstelle konfigurieren

22759

Für die Identifikation des IO-Link Masters in der EtherCAT-Projektierungssoftware kann der Anwender einen Namen vergeben.



Die Adressvergabe für den Feldbus-Port erfolgt über die EtherCAT-Projektierungssoftware.

Um den Feldbus-Port zu konfigurieren:

- ▶ Menü [Fieldbus] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Parameter	Beschreibung	Mögliche Werte
[Hostname]	Name des Geräts im EtherCAT-Netzwerk	z.B. al1xxx
[MAC address]	MAC-Adresse des Geräts	Wert ist fest eingestellt.

- ▶ Geänderte Werte auf dem Gerät speichern.

## 9.1.7 IO-Link-Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen

16551

Der Anwender kann für jeden IO-Link Port separat entscheiden, ob die Prozessdaten der angeschlossenen IO-Link Devices an LR AGENT oder LR SMARTOBSERVER übertragen werden.



Die Übertragung von Prozessdaten setzt voraus, dass die Schnittstelle zu LR AGENT oder LR SMARTOBSERVER korrekt konfiguriert ist (→ **IoT: Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren** (→ S. [29](#))).

Um die Datenübertragung zu aktivieren / deaktivieren:

- ▶ Menü [Port x] wählen (x = 1...8).
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[Transmission to LR Agent or SMARTOBSERVER]	Übertragung von Prozessdaten des angeschlossenen IO-Link Devices an LR AGENT oder LR SMARTOBSERVER	[Disabled]	Prozessdaten nicht übertragen
		[Enabled]	Prozessdaten übertragen

- ▶ Geänderte Werte auf dem Gerät speichern.

## 9.1.8 IO-Link-Ports: Betriebsart konfigurieren

Die IO-Link-Ports X01...X08 des Geräts unterstützen folgende Betriebsarten:

- Deaktiviert: keine Datenübertragung an Klemme 4 (C/Q) des IO-Link-Ports
- Digitaler Eingang (DI): binäres Eingangssignal an Klemme 4 (C/Q) des IO-Link-Ports
- Digitaler Ausgang (DO): binäres Ausgangssignal an Klemme 4 (C/Q) des IO-Link-Ports
- IO-Link: IO-Link-Datentransfer über Klemme 4 (C/Q) des IO-Link-Ports

Der Anwender kann für jeden IO-Link-Port die Betriebsart separat einstellen.

Um die Betriebsart eines IO-Link-Ports einzustellen:

- ▶ Menü [Port x] wählen (x = 1...8).
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[Mode Pin4 US]	Betriebsart des Klemme 4 des Ports	[Disabled]	Port deaktiviert
		[DI]	Betrieb als digitaler Eingang
		[DO]	Betrieb als digitaler Ausgang
		[IO-Link]	Betrieb als IO-Link-Schnittstelle
[Cycle time actual]**	Aktuelle Zykluszeit der Datenübertragung zwischen IO-Link Master und IO-Link Device am Port (Wert in Mikrosekunden)	Parameter nur lesbar	
[Cycle time preset]*	Zykluszeit der Datenübertragung zwischen IO-Link Master und IO-Link Device am Port (Wert in Mikrosekunden)	0	Gerät stellt schnellstmögliche Zykluszeit automatisch ein.
		1	1 Mikrosekunden
		...	...
		132800	132800 Mikrosekunden
[Bitrate]**	Aktuelle Datenrate der Datenübertragung zwischen IO-Link Master und IO-Link Device am Port	Parameter nur lesbar	

\* ... Parameter nur verfügbar, wenn [Mode] = [IO-Link]

\*\* ... Parameter nur sichtbar, wenn IO-Link Device am IO-Link-Port angeschlossen ist.

- ▶ Geänderte Werte auf dem Gerät speichern.

## 9.1.9 IO-Link-Ports: Gerätevalidierung und Datenspeicherung einstellen

Der Anwender kann wählen, wie sich die IO-Link Ports bezüglich der Gerätevalidierung und der Speicherung / Wiederherstellung von Parameterdaten des angeschlossenen IO-Link Devices verhalten soll.

Folgende Optionen sind verfügbar:

Option	Validierung des IO-Link Devices	Sicherung der Parameterwerte	Wiederherstellung der Parameterwerte
[No check and clear]	nein	nein	nein
[Type compatible V1.0 device]	ja, Prüfung auf Kompatibilität zu IO-Link Standard V1.0	nein	nein
[Type compatible V1.1 device]	ja, Prüfung auf Kompatibilität zu IO-Link Standard V1.1	nein	nein
[Type compatible V1.1 device with Backup + Restore]	ja, Prüfung auf Kompatibilität zu IO-Link Standard V1.1 und Baugleichheit (Vendor ID und Device ID)	ja, automatische Sicherung der Parameterwerte; Änderungen der aktuellen Parameterwerte werden gespeichert	ja, Wiederherstellung der Parameterwerte bei Anschluss eines baugleichen IO-Link Devices im Auslieferungszustand
[Type compatible V1.1 device with Restore]	ja, Prüfung auf Kompatibilität zu IO-Link Standard V1.1 und Baugleichheit (Vendor ID und Device ID)	nein, keine automatische Sicherung; Änderungen der aktuellen Parameterwerte werden nicht gespeichert	ja, Wiederherstellung der Parameterwerte bei Anschluss eines baugleichen IO-Link Devices im Auslieferungszustand



Die Optionen gelten nur, wenn der IO-Link Port im Betriebsmodus "IO-Link" ist.

Für Optionen [Type compatible V1.1 device with Backup + Restore] und [Type compatible V1.1 device with Restore]: Bei Änderung der Vendor ID und Device ID im Online-Modus wird der Datenspeicher gelöscht und eine neue Sicherung der Parameterwerte des angeschlossenen IO-Link Devices im IO-Link Master erzeugt.

Um die Gerätevalidierung und die Datenspeicherung zu konfigurieren:

- Menü [Port x] wählen (x = 1...8).
- > Menüseite zeigt die aktuellen Einstellungen.
- Folgende Parameter wie gewünscht einstellen:

Name	Beschreibung	Mögliche Werte	
[Validation / Data Storage]	Unterstützter IO-Link-Standard und Verhalten des IO-Link Masters bei Anschluss eines neuen IO-Link Devices am Port x (x = 1...8)	[No check and clear]	
		[Type compatible V1.0 device]	
		[Type compatible V1.1 device]	
		[Type compatible V1.1 device with Backup + Restore]	
		[Type compatible V1.1 device with Restore]	
[Vendor ID]	ID des Herstellers, der validiert werden soll	0...65535	Werkseinstellung: 0 ifm electronic: 310
[Device ID]	ID des IO-Link Devices, das validiert werden soll	0...16777215	Werkseinstellung: 0

- Geänderte Werte auf dem Gerät speichern.



### 9.1.10 Info: Geräteinformationen zeigen

12218

Um die allgemeine Informationen des ifm IO-Link Masters zu lesen:

- ▶ Menü [Info] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.

Name	Beschreibung	Mögliche Werte
[Product code]	Artikelnummer des IO-Link Masters	AL1930
[Device family]	Gerätefamilie des IO-Link Masters	IO-Link Master
[Vendor]	Hersteller	ifm electronic gmbh
[SW-Revision]	Firmware des IO-Link Masters	
[HW-Revision]	Hardware-Stand der IO-Link Masters	
[Bootloader revision]	Bootloader-Version des IO-Link Masters	
[Serial number]	Seriennummer	

### 9.1.11 Firmware: Gerät auf Werkseinstellungen rücksetzen

7209

Beim Rücksetzen des IO-Link Masters werden alle Parameter auf die Werkseinstellungen gesetzt:

Um das Gerät auf die Werkseinstellungen rückzusetzen:

- ▶ Menü [Firmware] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Auf [Factory Reset] klicken, um das Gerät rückzusetzen.
- > LR DEVICE setzt das Gerät auf die Werkseinstellungen.

### 9.1.12 Firmware: Gerät neu starten

18105

Bei einem Neustart des Geräts bleiben alle Einstellungen erhalten.

Um das AL1930 neu zu starten:

- ▶ Menü [Firmware] wählen.
- > Menüseite zeigt die aktuellen Einstellungen.
- ▶ Auf [Reboot] klicken, um das Gerät neu zu starten.
- > LR DEVICE startet den ifm IO-Link Master neu.

## 9.1.13 IO-Link Devices konfigurieren

Um die an das Gerät angeschlossenen IO-Link Devices mit der Parametriersoftware LR DEVICE zu konfigurieren:

### Voraussetzungen:

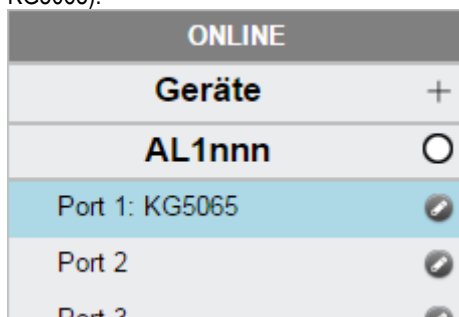
- > IO-Link Master ist korrekt installiert und mit der Parametriersoftware LR DEVICE verbunden.
- > IO-Link Device ist korrekt mit dem AL1930 verbunden.
- > Betriebsart des IO-Link-Ports ist "IO-Link" (→ **IO-Link-Ports: Betriebsart konfigurieren** (→ S. 31)).
- > IoT besitzt Schreibrechte auf dem IO-Link Master (→ **IoT: Zugriffsrechte konfigurieren** (→ S. 28)).

### 1 IO-Link Master wählen

- ▶ LR DEVICE starten.
- ▶ IODD-Datei-Bibliothek aktualisieren  
ODER:  
IODD-Datei des IO-Link Devices manuell importieren.
- ▶ Netzwerk nach Geräten scannen.
- > LR DEVICE erkennt IO-Link Master.

### 2 IO-Link Device hinzufügen

- ▶ Unter [ONLINE]: Auf gewünschten IO-Link Master klicken.
- > LR DEVICE erkennt automatisch die an den IO-Link Master angeschlossenen IO-Link Devices (z.B. ifm Sensor KG5065).



### 3 IO-Link Device konfigurieren

- ▶ Mausklick auf den Port, an dem das IO-Link Device angeschlossen ist.
- > LR DEVICE liest und zeigt die aktuellen Parameterwerte des IO-Link Devices.
- ▶ IO-Link Device konfigurieren.



Informationen über die verfügbaren Parameter des IO-Link Device: → IO Device Description (IODD) des IO-Link Devices

- ▶ Geänderte Konfiguration auf dem IO-Link Device speichern.

## 9.2 ifm-LoT-Core

### Inhalt

Hinweise für Programmierer .....	36
Erste Schritte .....	41
Allgemeine Funktionen .....	41
IoT: Zugriffsrechte konfigurieren .....	45
IoT: IP-Einstellungen konfigurieren .....	45
IoT: Sicherheitsmodus konfigurieren .....	45
Fieldbus: Feldbus-Schnittstelle konfigurieren .....	49
IO-Link Ports: Betriebsart Pin 4 (US) einstellen .....	49
IO-Link Ports: Gerätevalidierung und Datenspeicherung konfigurieren .....	50
IO-Link Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren .....	52
IO-Link Ports: Prozessdaten lesen / schreiben .....	52
IO-Link Ports: Port-Events anzeigen .....	55
IO-Link Devices: Auf Parameter zugreifen .....	55
IO-Link Devices: Geräteinformationen lesen und schreiben .....	57
IO-Link Devices: IO-Link Events anzeigen .....	57
Gateway: Gerät rücksetzen, neu starten und lokalisieren .....	57
Gateway: Geräteinformationen lesen .....	58
Gateway: Zustands- und Diagnoseinformationen lesen .....	58
Gateway: Firmware aktualisieren .....	59
Gateway: Anwendungskennung einstellen .....	60
Benachrichtigungen abonnieren .....	61
Web Socket nutzen .....	65
MQTT-Unterstützung .....	67
IoT-Core-Visualizer nutzen .....	71

31995



Allgemeine Hinweise zum ifm-LoT-Core: → **Hinweise für Programmierer** (→ S. [36](#))

## 9.2.1 Hinweise für Programmierer

### Inhalt

IoT Core: Allgemeine Informationen.....	36
Auf den ifm-IoT-Core zugreifen .....	37
IoT-Core: Diagnosecodes.....	40

10989

## IoT Core: Allgemeine Informationen

50737

Die Gerätefamilie CabinetLine verfügt über einen IoT-Core. Der IoT Core ermöglicht es dem Anwender, den AL1930 über eine REST API aus IT-Netzwerken heraus anzusprechen und in Internet-of-Things-Anwendungen einzubinden.

Auf dem AL1930 ist eine Gerätebeschreibung gespeichert. Diese Gerätebeschreibung ist ein strukturiertes, maschinenlesbares Datenobjekt im JSON-Format. In dieses Datenobjekt werden alle aktuellen Werte von Parametern, Prozessdaten, Diagnosedaten und Geräteinformationen abgebildet. Die Datenwerte können mithilfe von Diensten gelesen und geändert werden.

## Auf den ifm-IoT-Core zugreifen

50741

Der Anwender kann über HTTP-Requests auf den ifm-IoT-Core zugreifen. Folgende Request-Methoden stehen zur Verfügung.

### GET-Request

21300

Mit der GET-Methode kann der Anwender lesend auf einen Datenpunkt zugreifen.

Die Anfrage an den IoT-Core besitzt folgende Syntax:

`http://ip/datapoint/service`

Parameter	Beschreibung
ip	IP-Adresse der IoT-Schnittstelle
data_point	Datenpunkt, auf den zugegriffen werden soll
service	Dienst

Die Rückgabe des IoT-Cores besitzt folgende Syntax:

```
{
  "cid":id,
  "data":{"value":resp_data},
  "code":diag_code
}
```

Parameter	Beschreibung
id	Correlation ID für die Zuordnung von Anfrage und Rückgabe
resp_data	Wert des Datenpunkts; abhängig von Datentyp des Datenpunkts
diag_code	Diagnosecode (→ <b>IoT-Core: Diagnosecodes</b> (→ S. <a href="#">40</a> ))

### Beispiel: GET-Request

52004

Anfrage (per Browser):

`http://192.168.0.250/devicetag/applicationtag/getdata`

Rückgabe:

```
{
  "cid":-1,
  "data":{"value":"AL1930"},
  "code":200
}
```

## POST-Request

Mit der POST-Methode kann der Anwender lesend und schreibend auf einen Datenpunkt zugreifen.

Die Anfrage an den IoT-Core besitzt folgende allgemeine Syntax:

```
{
  "code": "code_id",
  "cid": id,
  "adr": "data_point/service",
  "data": {req_data},
  "auth": {"user": "usr_id", "passwd": "password"}
}
```

Feld	Parameter	Beschreibung
code	code_id	Dienstklasse
		▪ request      Anfrage
		▪ transaction      Transaktion
		▪ event      Ereignis
cid	id	Correlation ID für die paarweise Zuordnung von Anfrage und Rückgabe; vom Nutzer frei vergebbare Kennung
adr	data_point	Datenpunkt des Elemente-Baums, auf den zugegriffen werden soll
	service	auszuführender Dienst (→ <b>Übersicht: IoT-Dienste</b> (→ S. <a href="#">121</a> ))
data*	req_data	Daten, die an IoT-Core gesendet werden (z.B. neue Werte); Syntax abhängig vom Dienst
auth**	usr_id	Nutzername (Base64-codiert); Defaultwert: administrator
	password	Passwort (Base64-codiert)

\* = optional; nur erforderlich bei Diensten, die Daten an den IoT-Core senden (z .B. setdata)

\*\* = optional; nur erforderlich, wenn Sicherheitsmodus aktiviert ist

Die Rückgabe des IoT-Cores besitzt folgende Syntax:

```
{
  "cid": id,
  "data": {resp_data},
  "code": diag_code
}
```

Feld	Parameter	Beschreibung
cid	id	Correlation ID für die Zuordnung von Anfrage und Rückgabe (siehe Request)
data*	resp_data	Werte, die vom IoT-Core zurückgegeben werden; Syntax abhängig vom Dienst
code	diag_code	Diagnosecode (→ <b>IoT-Core: Diagnosecodes</b> (→ S. <a href="#">40</a> ))

\* = optional; nur erforderlich bei Diensten, die Daten vom IoT-Core empfangen (z .B. getdata)

### Beispiel: POST-Request

52006

Anfrage:

```
{  
  "code": "request",  
  "cid": 4711,  
  "adr": "devicetag/applicationtag/getdata"  
}
```

Rückgabe:

```
{  
  "cid": 4711,  
  "data": {"value": "AL1930"},  
  "code": 200  
}
```

**IoT-Core: Diagnosecodes**

22783

Code	Text	Beschreibung
200	OK	Anfrage erfolgreich abgearbeitet
230	OK but needs reboot	Anfrage erfolgreich abgearbeitet; IO-Link Master muss neu gestartet werden
231	OK but block request not finished	Anfrage erfolgreich abgearbeitet; blockweise Anfrage aber noch nicht beendet
232	Data has been accepted, but internally modified	Neue Werte wurden akzeptiert, wurde aber vom IO-Link Master angepasst (Master cycle time)
233	IP settings (of IoT-Port) have been updated. Application needs to reload device. Wait at least 1 second before reloading device.	IP-Einstellungen erfolgreich geändert, IO-Link Master wird neu geladen; mind. 1 Sekunde warten
400	Bad request	Ungültige Anfrage
401	Unauthorized	Nicht-authorisierte Anfrage
403	Forbidden	Verbotene Anfrage
500	Internal Server Error	Interner Fehler
503	Service Unavailable	Dienst nicht verfügbar (z. B. IO-Link Port im falschen Betriebsmodus; kein IO-Link Device an IO-Link Port)
530	The requested data is invalid	Ungültige Prozessdaten
531	IO-Link Error	Fehler in IO-Link Master / Device
532	PLC connected Error	Fehler beim Schreiben der Daten, da IO-Link Master noch mit Feldbus-SPS verbunden ist.



## 9.2.2 Erste Schritte

7118

Um die Gerätebeschreibung des AL1930 zu lesen:

- ▶ Folgende POST-Anfrage an den AL1930 senden:  
{"code": "request", "cid": -1, "adr": "gettree"}
- > AL1930 gibt die Gerätebeschreibung als strukturiertes JSON-Objekt zurück.
- ▶ In der Baumstruktur des JSON-Objekts alle Unterstrukturen und die enthaltenen Datenpunkte identifizieren.
- ▶ Die anwendbaren Dienste für den Zugriff auf Unterstrukturen und darin enthaltene Datenpunkte identifizieren.

## 9.2.3 Allgemeine Funktionen

60471

Das AL1930 besitzt den Typ device (→ **Übersicht: IoT-Typen** (→ S. 120)).

Auf dem Wurzelement vom Typ device können folgende Dienste angewendet werden:

Dienst	Beschreibung
../gettree	Gesamtbaum oder Teilbaum der Gerätebeschreibung (JSON) ausgeben
../getidentity	Geräteinformationen lesen
../getdatamulti	Mehrere Parameterwerte sequentiell lesen
../getelementinfo	Detaillierte Informationen eines Elements lesen
../getsubscriberlist	Liste mit allen aktiven Benachrichtigungsabonnements ausgeben
../querytree	Gerätebeschreibung nach spezifischen Elementen durchsuchen

In Abhängigkeit von den Lese- und Schreibrechten können auf Elemente vom Typ data folgende Dienste angewendet werden:

Dienst	Beschreibung
../getdata	Wert des Elements lesen
../setdata	Wert des Elements schreiben

## Beispiel: Eigenschaften eines Elements lesen

22406

**Aufgabe:** Datentyp und Wertebereich des Parameters accessrights bestimmen.

**Lösung:** Mit Dienst getelementinfo die Eigenschaften des Elements `iotsetup/accessrights` lesen. Die Felder `type` (Datentyp) und `valuation` (Wertebereich) enthalten die benötigten Informationen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "getelementinfo",
  "data": { "adr": "iotsetup/accessrights" }
}
```

- Antwort (Response):

```
{
  "cid": 4711,
  "data": {
    "identifier": "accessrights",
```

```

"type":"data",
"uid":null,
"profiles":["parameter"],
"format":{
  "type":"enum",
  "namespace":"json",
  "encoding":"integer",
  "valuation":{
    "valuelist":{
      "0":"Fieldbus + IoT",
      "1":"Fieldbus + IoT (read-only)",
      "3":"IoT only"}}}},
"code":200
}

```

Der Parameter accessrights hat den Datentyp ENUM mit den gültigen Werten "Fieldbus + IoT", "Fieldbus + IoT (read only)" und "IoT only".

## Beispiel: Teilbaum ausgeben

60247

**Aufgabe:** Alle direkten Unterelemente des Knotens firmware ausgeben.

**Lösung:** Mit Dienst gettree den gewünschten Teilbaum ausgeben (Wurzelknoten: firmware, zu zeigende Unterebenen: 1)

- Anfrage (Request):

```

{
  "code":"request",
  "cid":4711,
  "adr":"gettree",
  "data":{
    "adr":"firmware",
    "level":1}
}

```

- Rückgabe (Response):

```

{
  "cid":4711,
  "data":{
    "identifier":"firmware",
    "type":"structure",
    "profiles":[
      "software","software/uploadablessoftware"],
    "subs":[
      {
        "identifier":"version","type":"data","profiles":["parameter"],
        "format":{"type":"string","namespace":"json","encoding":"UTF-8"}},
      {
        "identifier":"type","type":"data",
        "format":{"type":"string","namespace":"json","encoding":"UTF-8"}},
      {
        "identifier":"install","type":"service"},
      {
        "identifier":"factoryreset","type":"service"},
      {
        "identifier":"signal","type":"service"},
      {
        "identifier":"container","type":"data",
        "format":{"type":"binary","namespace":"json","encoding":"base64"}},

```

```
{
  "identifizier":"reboot","type":"service"}]
},
"code":200
}
```

## Beispiel: Mehrere Parameterwerte des IO-Link Master gleichzeitig lesen

17310

**Aufgabe:** Folgende aktuelle Werte sollen vom IO-Link Master gelesen werden: Temperatur, Seriennummer

**Lösung:** Die aktuellen Parameterwerte mit dem dienst getdatamulti lesen (Datenpunkt Temperatur: /processdatamaster/temperature; Datenpunkt Seriennummer: /deviceinfo/serialnumber)

- Anfrage (Request):

```
{
  "code":"request",
  "cid":4711,
  "adr":"/getdatamulti",
  "data":{"datatosend":["/processdatamaster/temperature","/deviceinfo/serialnumber"]}
}
```

- Rückgabe (Response):

```
{
  "cid":4711,
  "data":{
    "processdatamaster/temperature":{"code":200,"data":44},
    "deviceinfo/serialnumber":{"code":200,"data":"000174210147"}},
  "code":200
}
```

## Beispiel: Gerätebeschreibung durchsuchen

60237

**Aufgabe:** Alle Elemente mit der Bezeichnung "status" und dem Profil "runcontrol" auflisten.

**Lösung:** Mit dem Dienst querytree die Gerätebeschreibung mit den Parametern "status" (name) und "runcontrol" (profile) durchsuchen.

- Anfrage (Request):

```
{
  "cid":4711,
  "adr":"querytree",
  "code":"request",
  "data":{
    "profile":"runcontrol",
    "name":"status"}
}
```

- Antwort (Response):

```
{
  "cid":4711,
  "data":{
    "adrList":[
      "device/connections/mqttConnection/status",
      "device/connections/mqttConnection/mqttCmdChannel/status"]},
  "code":200
}
```

## Speicherdauer einstellen

60243

Der IoT Core bietet die Möglichkeit, die Speicherdauer von Daten und Benachrichtigungen einzustellen. Die Dienste **Dienst: setdata** (→ S. [131](#)) und **Dienst: subscribe** (→ S. [133](#)) verfügen deshalb über den Parameter "duration".

## Beispiel: Benachrichtigungen abonnieren

60200

**Aufgabe:** Es sollen regelmäßig die aktuellen Werte folgender Parameter an einen Netzwerk-Server mit der IP-Adresse 192.168.0.4 verschickt werden:

- Produktname des IO-Link Devices an IO-Link Port X02
- zyklische Eingangsdaten des IO-Link Devices an IO-Link Port X02
- Betriebstemperatur des IO-Link Masters.

Das Abonnement soll nur bis zum nächsten Neustart des IO-Link Masters aktiv sein.

**Lösung:** Mit dem subscribe-Dienst die erforderlichen Daten abonnieren.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/timer[1]/counter/datachanged/subscribe",
  "data": {
    "callback": "http://192.168.0.4:80/temp",
    "datatosend": [
      "/iolinkmaster/port[2]/iolinkdevice/productname",
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/temperature"],
    "duration": "uptime"
  }
}
```

- Antwort (Response):

```
{
  "cid": 4711,
  "code": 200
}
```

## 9.2.4 IoT: Zugriffsrechte konfigurieren

58227

Unterstruktur: `iotsetup`

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../accessrights</code>	Zugriffsrechte auf den IO-Link Master	rw

rw ... lesen und schreiben



Wenn in IoT und EtherCAT-Projektierungssoftware der Parameter [Access Rights] = [EtherCAT + IoT], dann gelten immer die in der EtherCAT-Projektierungssoftware eingestellten Parameterwerte.

Wenn in IoT der Parameter [Access Rights] = [IoT only], dann in EtherCAT-Projektierungssoftware den Parameter [Access Rights] = [Keep settings] setzen.

Wenn in LR DEVICE der Parameter [Access Rigts] = [EtherCAT + IoT (read-only)], dann ist der Schreibzugriff auf die Gerätekonfiguration über LR DEVICE und IoT-Core-Dienste gesperrt. Um den Schreibzugriff wieder zu ermöglichen, den Parameter über Feldbus-Projektierungssoftware auf [EtherCAT + IoT] setzen.

Änderungen des Parameters [Access Rights] sind erst wirksam nach einem Neustart des IO-Link Masters (→ **Firmware: Gerät neu starten** (→ S. [33](#))).

## 9.2.5 IoT: IP-Einstellungen konfigurieren

60465

Unterstruktur: `iotsetup`

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../network/dhcp</code>	Konfiguration der IP-Einstellungen des IoT-Ports	rw
<code>../network/ipaddress</code>	IP-Adresse des IoT-Ports	rw
<code>../network/subnetmask</code>	Subnetzmaske des Netzwerksegments	rw
<code>../network/ipdefaultgateway</code>	IP-Adresse des Netzwerk-Gateways	rw

rw ... lesen und schreiben

Anwendbare Dienste:

Name	Beschreibung
<code>../network/setblock</code>	alle Werte der Unterstruktur blockweise schreiben



Die IP-Parameter in der Unterstruktur network nur blockweise mit dem Dienst setblock ändern!

## 9.2.6 IoT: Sicherheitsmodus konfigurieren

25397

Der Zugriff auf die IoT-Schnittstelle des IO-Link Masters kann mit einem Sicherheitsmodus geschützt werden:

Unterstruktur: `iotsetup`

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../security/securitymode	aktiver Sicherheitsmodus	rw
../security/password	Passwort für Authentifizierung (Base64-codiert)	w

rw ... lesen und schreiben  
w ... nur schreiben



Gültiger Zeichensatz für die Base64-Kodierung / Dekodierung des Passworts: UTF-8  
Online-Werkzeug für Kodierung / Dekodierung: → [www.base64encode.org](http://www.base64encode.org)

## Hinweis: Sicherheitsmodus

17561

Der Sicherheitsmodus erlaubt es, den Zugriff auf den IO-Link Master und die angeschlossenen IO-Link Devices aus dem IT-Netzwerk heraus zu beschränken. Im aktivierten Sicherheitsmodus gelten folgende Einschränkungen:

- Zugriff nur mit Authentifizierung (passwortgeschütztes Nutzerkonto)
- Zugriff nur über sichere https-Verbindung (Transport Layer Security - TLS)



Der Sicherheitsmodus schützt nur den Zugriff auf das Gerät über die IoT-Schnittstelle.  
Standardwert für user ist: administrator  
Das gesetzte Passwort kann nicht mit getdata ausgelesen werden.

Der aktuelle Status der Sicherheitsfunktion kann mit dem Dienst getidentity gelesen werden (→ **Dienst: getidentity** (→ S. [124](#))).

Für die Authentifizierung muss der Anwender den POST-Requests zusätzlich das Feld "auth" mit gültigem Nutzernamen und Passwort übergeben. Nutzernamen und Passwort werden als Base64-codierte Zeichenketten dargestellt (→ **Beispiel: Anfrage mit Authentifikation** (→ S. [47](#))).

Folgende Anfragen können bei aktiviertem Sicherheitsmodus auch ohne Authentifizierung ausgeführt werden:

- /getidentity
- /deviceinfo/vendor/getdata
- /deviceinfo/productcode/getdata

## Beispiel: Sicherheitsmodus aktivieren

34532

**Aufgabe:** Den Sicherheitsmodus der IoT-Schnittstelle des IO-Link Masters aktivieren. Das Passwort "password" setzen (Base64-codiert: cGFzc3dvcmQ=)

**Lösung:** Die Aktivierung besteht aus 2 Schritten:

### 1 Sicherheitsfunktion aktivieren

Mit Dienst setdata den Sicherheitsmodus über den Datenpunkt `iotsetup/security/securitymode` aktivieren.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": -1,
  "adr": "/iotsetup/security/securitymode/setdata",
  "data": {"newvalue": "1"}
}
```

- Antwort (Response):

```
{
  "cid": -1,
  "code": 200
}
```

### 2 Gewünschtes Passwort setzen

Mit Dienst setdata das gewünschte Passwort in den Datenpunkt `iotsetup/security/password` schreiben.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": -1,
  "adr": "/iotsetup/security/password/setdata",
  "data": {"newvalue": "cGFzc3dvcmQ="}
}
```

- Antwort (Response):

```
{
  "cid": -1,
  "code": 200
}
```

## Beispiel: Anfrage mit Authentifikation

51702

**Aufgabe:** Die Temperatur des IO-Link Masters soll gelesen werden. Die Sicherheitsfunktion ist aktiviert (aktuelles Passwort: password).

**Lösung:** Mit Dienst getdata den Datenpunkt `processdatamaster/temperature` lesen. Die Anfrage muss mit https gesendet werden. Nutzernamen und Passwort werden als Base64-codierte Zeichenketten übergeben ("administrator" = "YWRTaW5pc3RyYXRvcg==", "password" = "cGFzc3dvcmQ=")

- Anfrage (Request):

```
{
  "code": "request",
  "cid": -1,
  "adr": "processdatamaster/temperature/getdata",
  "auth": {"user": "YWRTaW5pc3RyYXRvcg==", "passwd": "cGFzc3dvcmQ="}
}
```

- Antwort (Response):

```
{
  "cid":-1,
  "data":{"value":37},
  "code":200
}
```

## Beispiel: Passwort rücksetzen

21577

**Aufgabe:** Das existierende Passwort soll rückgesetzt werden.

**Lösung:** Um ein Passwort rückzusetzen, den Sicherheitsmodus deaktivieren. Für die Deaktivierung ist die Eingabe des Nutzernamens und des Passworts notwendig (Felder "user" und "passwd").

- Anfrage (Request):

```
{
  "code":"request",
  "cid":-1,
  "adr":"iotsetup/security/securitymode/setdata",
  "data":{"newvalue":0},
  "auth":{"user":"YWRtaW5pc3RyYXRvcg==", "passwd":"SW9UNG1mbQ=="}
}
```

- Rückgabe (Response):

```
{
  "cid":-1,
  "code":200
}
```



## 9.2.7 Fieldbus: Feldbus-Schnittstelle konfigurieren

16564

Über die Feldbus-Schnittstelle (Ports X21 / X22) wird der AL1930 in das EtherCAT-Netzwerk integriert.

Unterstruktur: `fieldbussetup`

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../hostname</code>	Bezeichnung des IO-Link Masters in Feldbus-Projekt	rw
<code>../fieldbusfirmware</code>	Firmware-Version des IO-Link Masters	r
<code>../connectionstatus</code>	Status der Verbindung zum EtherCAT-Netzwerk	r

r ... nur lesen

rw ... lesen und schreiben

## 9.2.8 IO-Link Ports: Betriebsart Pin 4 (US) einstellen

56695

Unterstruktur: `iolinkmaster/port[n]` (n = 1...8).

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../mode</code>	Betriebsart Pin 4 des Ports	rw*
<code>../mastercycletime_preset</code>	Zykluszeit der Datenübertragung am IO-Link-Port (Wert in ms)	rw*
<code>../mastercycletime_actual</code>	aktuelle Zykluszeit der Datenübertragung am IO-Link-Port (Wert in ms)	r
<code>../comspeed</code>	Datenübertragungsrate des IO-Link Ports	r

r ... nur lesen

rw ... lesen und schreiben

\* ... nur änderbar, wenn EtherCAT-SPS nicht in Zustand RUNNING

## 9.2.9 IO-Link Ports: Gerätevalidierung und Datenspeicherung konfigurieren

56492

Unterstruktur: `iolinkmaster/port[n]` ( $n = 1 \dots 8$ ).

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../validation_datastorage_mode</code>	Verhalten des IO-Link-Ports beim Anschluss der eines neuen IO-Link Devices	rw*
<code>../validation_vendorid</code>	IO-Link-ID des Herstellers, der validiert werden soll	rw*
<code>../validation_deviceid</code>	IO-Link-ID des Geräts, das validiert werden soll	rw*
<code>../datastorage</code>	Struktur für Data Storage des Ports	rw
<code>../datastorage/maxsize</code>	maximale Größe des Data-Storage-Inhalts (in Bytes)	r
<code>../datastorage/chunksize</code>	Größe eines Datensegments (in Bytes)	r
<code>../datastorage/size</code>	Größe der Data-Storage-Inhalts (in Bytes)	r

r ... nur lesen

rw ... lesen und schreiben

\* ... nur änderbar, wenn sich die EtherCAT-SPS nicht im Zustand RUNNING befindet

Anwendbare Dienste:

Dienst	Beschreibung
<code>../validation_useconnecteddevice</code>	das mit dem IO-Link Port verbundene IO-Link Devices validieren*
<code>../datastorage/getblobdata</code>	Inhalt des Data-Storage-Bereichs lesen
<code>../datastorage/stream_set</code>	Einzelnes Datensegment übertragen*
<code>../datastorage/start_stream_set</code>	Sequentielle Übertragung mehrerer Datensegmente starten*

\* ... nur änderbar, wenn sich die EtherCAT-SPS nicht im Zustand RUNNING befindet

### Beispiel: Datenspeicher eines IO-Link Ports klonen

36055

#### Aufgabe:

Der Datenspeicher des IO-Link Ports X02 von IO-Link Master 1 soll nach IO-Link Master 2 geklont werden.

#### Lösung:

Das Klonen besteht aus 2 Schritten. Im ersten Schritt wird der Datenspeicher des IO-Link Ports von IO-Link Master 1 gelesen. Im zweiten Schritt werden die gelesenen Daten in den Datenspeicher des IO-Link Ports von IO-Link Master 2 gespeichert.

Datenspeicher sichern:

#### 1 Vorbereitungen

- ▶ Segmentgröße des Data Storage lesen ( $h$  = Anzahl der Bytes):  
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/chunksize/getdata"}`  
 Beispiel:  $h = 256$
- ▶ Gesamtgröße des Data-Storage-Bereichs lesen ( $g$  = Anzahl der Bytes):  
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/size/getdata"}`  
 Beispiel:  $g = 550$
- ▶ Anzahl der Leseschritte  $n$  berechnen:  $n$  = erste ganzzahlige Wert, für den gilt:  $g < n \cdot h$   
 Beispiel:  $n = 3$ , da  $550 < 3 \cdot 256$

#### 2 Datenspeicher des IO-Link Ports lesen

- ▶ Segmentweise den Data Storage lesen ("pos" gibt den Byte-Offset an, an dem mit der Länge "length" der Lesevorgang startet).

```
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": 0, "length": h}}
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": h, "length": h}}
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": 2*h, "length": h}}
...
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/getblobdata", "data": {"pos": n*h, "length": h}}
```

Beispiel:

1. Lesebefehl: pos = 0, length = 256
2. Lesebefehl: pos = 256, length = 256
3. Lesebefehl: pos = 512, length = 256

- > Jedes Segment wird als BASE64-String zurückgegeben.
- Segmente verbinden (konkateneren).

Datenspeicher wiederherstellen:

## 1 Vorbereitungen

- Größe des gesicherten Data-Storage-Inhalts ermitteln (n = Anzahl der Bytes).  
Beispiel: n = 550
- Segmentgröße lesen (s = Anzahl der Bytes):  
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/chunksize/getdata"}  
Beispiel: s = 256

## 2 Data-Storage-String übertragen

- Segmentweises Schreiben des Data-Storage-Strings starten ("size" = Größe des gelesenen Data Storage):  
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/start\_stream\_set", "data": {"size": n}}  
Beispiel: size = 550
- Segmentweise Data-Storage-String übertragen ("value" = String-Wert mit Länge s):  
{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/stream\_set", "data": {"value":  
"aWZtfgIAAABBTDf4NXhfY25faXRfdDluMi43Nw..."}}

## 9.2.10 IO-Link Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren

58228

Unterstruktur: `iolinkmaster/port[n]` ( $n = 1 \dots 8$ ).

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../senddatatosmob</code>	Prozessdaten an LR SMARTOBSERVER senden	rw

rw ... lesen und schreiben

## 9.2.11 IO-Link Ports: Prozessdaten lesen / schreiben

60467

Unterstruktur: `iolinkmaster/port[n]` ( $n = 1 \dots 8$ )

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../pin2in</code>	Wert des digitalen Eingangs an Klemme 2 des IO-Link Ports	r
<code>../iolinkdevice/pdin</code>	Wert des IO-Link Eingangs an Klemme 4 des IO-Link Ports	r
<code>../iolinkdevice/pdout</code>	Wert des IO-Link Ausgangs an Klemme 4 des IO-Link Ports	rw*

r = nur lesen

rw = lesen und schreiben

\* = nur änderbar, wenn Feldbus-SPS nicht im Betriebszustand RUNNING

### Beispiel: IO-Link Prozesswert lesen (Betriebsart "IO-Link")

16574

**Aufgabe:** Den aktuellen Messwert des ifm-Temperatursensors TN2531 an IO-Link-Port X2 lesen

**Lösung:** Den Datenpunkt für die Prozess-Eingangsdaten mit dem Dienst `getdata` lesen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/iolinkmaster/port[2]/iolinkdevice/pdin/getdata"
}
```

- Rückgabe (Response):

```
{
  "cid": 4711,
  "data": {"value": "03C9"},
  "code": 200
}
```

Der Rückgabewert wird im Hexadezimalformat dargestellt. Der Rückgabewert enthält neben dem Temperaturwert zusätzliche Informationen (→ IO Device Description (IODD) des Sensors). Der Temperaturwert wird in den Bits 2 bis 15 abgebildet.

$0x03C9 = 0b11111001001$

Temperaturwert:  $0b111110010 = 242$

Daraus folgt: Der aktuelle Temperaturwert ist  $24,2\text{ °C}$ .

## Beispiel: IO-Link Prozesswert schreiben (Betriebsart "IO-Link")

56508

**Aufgabe:** Der Buzzer des DV2500 an IO-Link Port X2 soll eingeschaltet werden. Der DV2500 arbeitet im On/Off-Modus.

**Lösung:** Die IODD des DV2500 zeigt die Struktur des Prozesswerts (→ z. B. LED-Aktivität). Der Buzzer wird über Bit 40 des Prozesswerts geschaltet (OFF = 0, ON = 1).

Um nur den Buzzer zu schalten:

1. Den aktuellen Prozesswert lesen (→ **Beispiel: IO-Link Prozesswert lesen (Betriebsart "IO-Link")** (→ S. 52)).
2. Bit 40 des gelesenen Werts auf 1 setzen.
3. Neuen Prozesswert schreiben.

Beispiel:

Gelesener Prozesswert:

0x0000 0000 004D = 0b0000 0000 0000 0000 0000 0000 0000 0000 0100 1101

Neuer Prozesswert:

0b0000 0001 0000 0000 0000 0000 0000 0000 0000 0100 1101 = 0x0100 0000 004D

- Anfrage (Request):

```
{
"code":"request",
"cid":10,
"adr":"iolinkmaster/port[2]/iolinkdevice/pdout/setdata",
"data":{"newvalue":"01000000004D"}
}
```

- Rückgabe (Response):

```
{
"cid":10,
"code":200
}
```

## Beispiel: Digitalen Ausgangswert schreiben (Betriebsart "DO")

56506

**Aufgabe:** Der digitale Ausgangswert des IO-Link Devices an IO-Link Port X1 soll auf "ON" gesetzt werden. Die Betriebsart des IO-Link Ports ist "Digital Output (DO)".

**Lösung:** Den Wert 1 auf den Datenpunkt pdout schreiben. Der Wert muss als Hexadezimalwert mit der Länge von 1 Byte übergeben werden (OFF = "00", ON = "01").

- Anfrage (Request):

```
{
"code":"request",
"cid":10,
"adr":"iolinkmaster/port[1]/iolinkdevice/pdout/setdata",
"data":{"newvalue":"01"}
}
```

- Rückgabe (Response):

```
{
"cid":10,
"code":200
}
```

## Beispiel: Digitalen Eingangswert lesen (Betriebsart "DI")

56507

**Aufgabe:** Der digitale Eingangswert des IO-Link Devices an IO-Link Port X5 soll gelesen werden. Die Betriebsart des IO-Link Ports ist "Digital Input (DI)".

**Lösung:** Den Prozesswert des Datenpunkts pdin schreiben. Der Prozesswert wird als Hexadezimalwert mit der Länge von 1 Byte zurückgegeben (OFF = "00", ON = "01").

- Anfrage (Request):

```
{  
"code":"request",  
"cid":10,  
"adr":"iolinkmaster/port[5]/iolinkdevice/pdin/getdata"  
}
```

- Rückgabe (Response):

```
{  
"cid":10,  
"data":{"value":"00"},  
"code":200  
}
```

## 9.2.12 IO-Link Ports: Port-Events anzeigen

58237

Unterstruktur: `iolinkmaster/port[n]` ( $n = 1...8$ ).

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../portevent</code>	Anzeige folgender Ereignisse an IO-Link Port n: <ul style="list-style-type: none"> <li>IO-Link Device stecken</li> <li>IO-Link Device ziehen</li> <li>Betriebsart des IO-Link Ports geändert</li> </ul>	r

r ... nur lesen



Ereignisse abonnieren: → **Benachrichtigungen abonnieren** (→ S. [61](#))

## 9.2.13 IO-Link Devices: Auf Parameter zugreifen

58231

Der ifm-IO-T-Core unterstützt die Konfiguration der angeschlossenen IO-Link Devices. Der Zugriff auf einen Parameter erfolgt über IO-Link Index und Subindex (→ IO Device Description (IODD) des Geräts).

Unterstruktur: `iolinkmaster/port[n]/iolinkdevice` ( $n = 1...8$ )

Anwendbare Dienste:

Dienst	Beschreibung
<code>../iolreadacyclic</code>	Parameter eines IO-Link Devices lesen (azyklisch)
<code>../iolwriteacyclic</code>	Parameter eines IO-Link Devices schreiben (azyklisch)

### Beispiel: Parameterwert eines IO-Link Devices lesen

16546

**Aufgabe:** Seriennummer des ifm-Temperatursensors TN2531 an IO-Link-Port X02 lesen

**Lösung:** Die Seriennummer mit dem Dienst `iolreadacyclic` aus dem IO-Link Devices lesen ( Index: 21, Subindex: 0)

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/iolinkmaster/port[2]/iolinkdevice/iolreadacyclic",
  "data": {"index": 21, "subindex": 0}
}
```

- Rückgabe (Response):

```
{
  "cid": 4711,
  "data": {"value": "4730323134323830373130"},
  "code": 200
}
```

Der zurückgegebene Wert wird im Hexadezimalformat dargestellt. Die Konvertierung des HEX-Werts in einen STRING-Wert ergibt: G0214280710

## Beispiel: Parameterwert eines IO-Link Devices ändern

16578

**Aufgabe:** Die Ausgangskonfiguration OUT1 des ifm-Temperatursensors TN2531 an IO-Link-Port X02 auf den Wert "Hnc / Hysteresefunktion, Öffner" setzen.

**Lösung:** Den Parameter [ou1] des Sensors mit dem Dienst iolwritecyclicdata auf den Wert 4 ändern. Auf den Parameter kann über IO-Link Index 580, Subindex 0 zugegriffen werden (→ IO-Link-Beschreibung des Sensors).

- Anfrage (Request):

```
{  
  "code": "request",  
  "cid": 4711,  
  "adr": "/iolinkmaster/port[2]/iolinkdevice/iolwritecyclic",  
  "data": {"index": 580, "subindex": 0, "value": "34"}  
}
```

Der Wert muss im Hexadezimalformat übergeben werden. Die Konvertierung des STRING-Werts in einen HEX-Wert ergibt: 34

- Rückgabe (Response):

```
{"cid": 4711, "code": 200}
```



## 9.2.14 IO-Link Devices: Geräteinformationen lesen und schreiben

58230

Unterstruktur: iolinkmaster/port[n]/iolinkdevice (n = 1...8)

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../status	Status des angeschlossenen IO-Link Devices	r
../vendorid	IO-Link ID des Herstellers	r
../deviceid	IO-Link ID des IO-Link Devices	r
../productname	Produktname des IO-Link Devices	r
../serial	Seriennummer des IO-Link Devices	r
../applicationspecifictag	Anwendungsspezifische Kennung (Application Tag)	rw

r ... nur lesen  
rw ... lesen und schreiben

## 9.2.15 IO-Link Devices: IO-Link Events anzeigen

58238

Unterstruktur: iolinkmaster/port[n]/iolinkdevice (n = 1...8).

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../iolinkevent	Anzeige von IO-Link Events	r

r ... nur lesen



Ereignisse abonnieren: → **Benachrichtigungen abonnieren** (→ S. [61](#))

## 9.2.16 Gateway: Gerät rücksetzen, neu starten und lokalisieren

22761

Unterstruktur: firmware

Anwendbare Dienste:

Name	Beschreibung
../factoryreset	IO-Link Master auf Werkseinstellungen rücksetzen
../reboot	IO-Link Master neu starten
../signal	Blinken der Status-LED auslösen

## 9.2.17 Gateway: Geräteinformationen lesen

12076

Unterstruktur: deviceinfo

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../productcode	Artikelnummer	r
../vendor	Hersteller	r
../devicefamily	Gerätefamilie	r
../hwrevision	Hardware-Revision	r
../serialnumber	Seriennummer	r
../swrevision	Firmware-Version	r
../bootloaderrevision	Bootloader-Version	r
../extensionrevisions	Firmware- und Bootloader-Version	r
../fieldbustype	Feldbus	r

r ... nur lesen

Zusätzliche Informationen über das AL1930 kann mit dem Dienst `getidentity` gelesen werden (→ **Dienst: `getidentity`** (→ S. [124](#))).

## 9.2.18 Gateway: Zustands- und Diagnoseinformationen lesen

60468

Unterstruktur: processdatamaster

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../temperature	Temperatur des IO-Link Masters (Wert in °C)	r
../voltage	aktueller Spannungswert der Geräteversorgung US (Wert in mV)	r
../current	aktueller Stromwert der Geräteversorgung US (Wert in mA)	r
../supervisionstatus	Status der Geräteversorgung US	r

r ... nur lesen

## 9.2.19 Gateway: Firmware aktualisieren

11616

Unterstruktur: firmware

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../version	Software-Version	r
../type	Software-Typ	r
../container	Struktur für die Firmware-Aktualisierung	w
../container/maxsize	maximale Größe des Container-Struktur (in Bytes)	r
../container/chunksize	Größe eines Datensegments (in Bytes)	r
../container/size	Größe des Container-Inhalts (in Bytes)	r

r = nur lesen

w = nur schreiben

Anwendbare Dienste:

Name	Beschreibung
../install	Auf den IO-Link Master übertragene Firmware installieren
../container/stream_set	Einzelnes Datensegment übertragen
../container/start_stream_set	Sequentielle Übertragung mehrerer Datensegmente starten

## Beispiel: Firmware aktualisieren

31160

### Aufgabe:

Die Firmware des Geräts aktualisieren; Größe der Firmware-Datei: 356676 Bytes

### Lösung:

Die Firmware des Geräts wird in Fragmenten (chunks) auf das Gerät übertragen. Die Größe der Fragmente ist abhängig von der Größe des Flashspeichers des IO-Link Masters. Um die Firmware zu übertragen, muss die Firmware-Datei mit BASE64 in eine Zeichenfolge umgewandelt werden.

#### 1 Vorbereitungen

- ▶ Größe der Fragmente ermitteln (g = Anzahl der Bytes):  
{"code": "request", "cid": -1, "adr": "/firmware/container/chunksize/getdata"}
- ▶ Firmware-Datei in einen BASE64-String umwandeln.

#### 2 Übertragung der Firmware starten

- ▶ Übertragung der Firmware mit dem Dienst start\_stream\_set starten (Parameter "size": Größe der Firmware-Datei):  
{"code": "request", "cid": -1, "adr": "/firmware/container/start\_stream\_set", "data": {"size": 356676}}

#### 3 Firmware in den Flashspeicher des IO-Link Masters laden

- ▶ BASE64-String der Firmware-Datei fragmentweise an den IO-Link Master senden (value = String-Wert mit Länge g).  
{"code": "request", "cid": -1, "adr": "/firmware/container/stream\_set", "cid": -1, "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDluMi43Nw..."}}
- ▶ Schritt 3 wiederholen, bis alle Fragmente der Firmware-Datei an IO-Link Master gesendet wurden.
- > IO-Link Master speichert die empfangenen Segmente im Container-Bereich.

#### 4 Firmware installieren

- ▶ Installation der übertragenen Firmware starten.  
{"code": "request", "cid": -1, "adr": "/firmware/install", "data": {}}

## 9.2.20 Gateway: Anwendungskennung einstellen

58232

Der Anwender kann die Bezeichnung des IO-Link Masters einstellen:

Unterstruktur: devicetag

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../applicationtag	Bezeichnung des IO-Link Masters (Application Tag)	rw

rw ... lesen und schreiben



Für die Speicherung des applicationtag stehen auf dem IO-Link Master 32 Byte zu Verfügung. Wird der Speicherbereich beim Schreiben mit setdata überschritten, bricht der IoT Core den Schreibvorgang ab und gibt den Diagnosecode 400 zurück.

Beim Schreiben der Anwendungskennung den unterschiedlichen Speicherbedarf der einzelnen UTF-8-Zeichen beachten:

- Zeichen 0-127: 1 Byte pro Zeichen
- Zeichen >127: mehr als 1 Byte pro Zeichen

### Beispiel: Bezeichnung des IO-Link Master ändern

10987

**Aufgabe:** Die Bezeichnung des IO-Link Masters für die Darstellung im LR SMARTOBSERVER auf AL1930 setzen.

**Lösung:** Den Parameter [Application Tag] mit dem Dienst setdata auf den Wert [AL1930] ändern.

Der Datenpunkt des Parameters [Application Tag] im Gerätebeschreibungsobjekt ist /devicetag/applicationtag.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/devicetag/applicationtag/setdata",
  "data": {"newvalue": "AL1930"}
}
```

- Rückgabe (Response):

```
{"cid": 4711, "code": 200}
```

## 9.2.21 Benachrichtigungen abonnieren

58298

Besitzt ein Datenpunkt das Unterelement datachanged, kann der Anwender Benachrichtigungen über Werte- oder Zustandsänderungen abonnieren. Auslöser für die Benachrichtigungen kann der Ablauf eines Timers oder ein Ereignis sein. Der IoT Core unterstützt die Ausgabe der Benachrichtigungen im CSV- oder im JSON-Format.

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
timer[x]/counter	Timer für das Auslösen einer Benachrichtigung	rw
timer[x]/interval	Zykluszeit der Aktualisierung der abonnierten Werte	rw
iolinkmaster/port[n]/portevent	Anzeige folgender Ereignisse an IO-Link Port n: <ul style="list-style-type: none"> <li>IO-Link Device stecken</li> <li>IO-Link Device ziehen</li> <li>Betriebsart des IO-Link Ports geändert</li> </ul>	rw
iolinkmaster/port[n]/iolinkdevice/iolinkevent	Anzeige von IO-Link Events	rw

r ... nur lesen  
rw ... lesen und schreiben  
x = [1,2]  
n = 1...8

Anwendbare Dienste:

Name	Beschreibung
../datachanged/subscribe	Benachrichtigung anmelden
../datachanged/unsubscribe	Benachrichtigung abmelden
../datachanged/getsubscriptioninfo	Informationen über Benachrichtigung zeigen

Zusätzlich kann der Anwender mit **Dienst: getsubscriberlist** (→ S. [125](#)) alle aktiven Abonnements anzeigen.

### Beispiel: Benachrichtigungen abonnieren

60212

**Aufgabe:** Es sollen regelmäßig die aktuellen Werte folgender Parameter an einen Netzwerk-Server mit der IP-Adresse 192.168.0.4 verschickt werden:

- zyklische Eingangsdaten des IO-Link Devices an IO-Link Port X02
- Betriebstemperatur des IO-Link Masters.

**Lösung:** Mit dem subscribe-Dienst die erforderlichen Daten abonnieren.



Folgende Optionen stehen zusätzlich zur Verfügung:

- per WebSockets (ws://): **Beispiel: WebSockets nutzen** (→ S. [65](#))
- per MQTT (mqtt://): **Beispiel: MQTT-Kommandokanal konfigurieren** (→ S. [69](#))

- Anfrage (Request):

```
{
"code": "request",
"cid": 4711,
"adr": "/timer[1]/counter/datachanged/subscribe",
"data":
```

```
{
  "callback": "http://192.168.0.4:80/temp",
  "datatosend": [
    "/iolinkmaster/port[2]/iolinkdevice/pdin",
    "/processdatamaster/temperature"
  ]
}
```

Zusätzlich muss das Zeitintervall des timer[1] auf einen Wert zwischen 500 ms und 2147483647 ms gesetzt werden.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4712,
  "adr": "/timer[1]/interval/setdata",
  "data": {"newvalue": 500}
}
```

- Rückgabe (Response):

```
{
  "cid": 4712,
  "code": 200
}
```

- Benachrichtigung (JSON)

```
{
  "code": "event",
  "cid": 4711,
  "adr": "",
  "data": {
    "eventno": "6317",
    "srcurl": "/timer[1]/counter/datachanged",
    "payload": {
      "/timer[1]/counter": {"code": 200, "data": 1},
      "/processdatamaster/temperature": {"code": 200, "data": 39},
      "/iolinkmaster/port[2]/iolinkdevice/pdin": {"code": 200, "data": "03B0"}}
  }
}
```

## Beispiel: Abonnement ändern

60263

**Aufgabe:** Das existierende Abonnement (**Beispiel: Benachrichtigungen abonnieren** (→ S. 61)) soll geändert werden. Anstatt der Temperatur des IO-Link Masters soll die anliegende Betriebsspannung übertragen werden.

**Lösung:** Das existierende Abonnement überschreiben. Dafür müssen in der Anfrage die Parameterwerte für "cid" und "callback" gleich denen des existierenden Abonnements sein.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/timer[1]/counter/datachanged/subscribe",
  "data": {
    "callback": "http://192.168.0.4:80/temp",
    "datatosend": [
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/voltage"
    ]
  }
}
```

## Beispiel: Benachrichtigungen im CSV-Format abonnieren

60214

**Aufgabe:** Es sollen alle 2 Sekunden die aktuellen Werte folgender Parameter an einen Netzwerk-Server mit der IP-Adresse 192.168.0.4 verschickt werden:

- zyklische IO-Link-Eingangsdaten des IO-Link Devices an Port X02
- Betriebstemperatur des IO-Link Masters.

Die Daten sollen im CSV-Format (Komma-Separator) übertragen werden.

### Lösung:

- ▶ Mit dem subscribe-Dienst die erforderlichen Daten abonnieren und das Ausgabeformat auf "csv0" setzen.



Daten im CSV-Format können nur per TCP-Protokoll verschickt werden.

- Anfrage (Request):

```
{
  "cid": 1,
  "adr": "/timer[1]/counter/datachanged/subscribe",
  "code": "request",
  "callback": "tcp://192.168.50.59:1883/topic",
  "codec": "csv0",
  "data": {
    "datatosend": [
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/temperature"
    ]
  }
}
```

- ▶ Das Intervall des Timers auf 2 Sekunden setzen:

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4712,
  "adr": "/timer[1]/interval/setdata",
  "data": { "newvalue": 2000 }
}
```

Die zyklische gesendete Benachrichtigung hat folgende Struktur:

/timer[1]/counter/datachanged,6317,200,1,200,39,200,03B0

## Beispiel: Benachrichtigungen abmelden

60265

**Aufgabe:** Das existierende Abonnement (**Beispiel: Benachrichtigungen abonnieren** (→ S. [61](#))) soll gelöscht werden.

**Lösung:** Mit dem Dienst unsubscribe das Abonnement löschen. Dafür muss in der Anfrage der Wert des Parameters "callback" gleich dem des existierenden Abonnements sein.

```
{
"code": "request",
"cid": 4711,
"adr": "/timer[1]/counter/datachanged/unsubscribe",
"data": {
"callback": "http://192.168.0.4:80/temp"
}
}
```

## Beispiel: Abonnements prüfen

60267

**Aufgabe:** Informationen über das existierende Abonnement (**Beispiel: Benachrichtigungen abonnieren** (→ S. [61](#))) anzeigen.

**Lösung:** Mit Dienst getsubscriptioninfo und den Parameterwerten cid, "adr" und "callback" des existierenden Abonnements die Informationen abrufen.

- Anfrage (Request):

```
{
"code": "request",
"cid": 4711,
"adr": "/timer[1]/counter/datachanged/getsubscriptioninfo",
"data": {
"callback": "http://192.168.0.4:80/temp"
}
}
```

- Rückgabe (Response):

```
{
"cid": 4711,
"data": {
"callback": "http://192.168.0.4:80/temp",
"datatosend": [
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"
]},
"code": 200
}
```



## 9.2.22 Web Socket nutzen

60210

Der IoT Core unterstützt die Kommunikation per WebSocket-Protokoll. Mit Web Sockets kann der Anwender einen Full-Duplex-Kommunikationskanal über eine TCP-Verbindung aufbauen.

WebSockets können für folgende Dienste genutzt werden:

- subscribe / unsubscribe



Maximale Anzahl der WebSocket-Verbindungen: 8

Sichere WebSocket-Verbindungen (wss://) werden nicht unterstützt.

Um Benachrichtigungen über eine WebSockets-Verbindung zu übertragen:

- ▶ WebSocket-Verbindung herstellen (z. B. "ws://192.168.0.55:80/websocket")
  - Option 1: ohne Parameter "callback"
- ▶ subscribe-/unsubscribe-Anfrage ohne Parameter "callback" stellen.
- > IoT-Core sendet Benachrichtigungen über existierende WebSocket-Verbindung.
- Option 2: mit Parameter "callback"
- ▶ subscribe-/unsubscribe-Anfragen mit Parameter "callback" ("ws:///myTopic") stellen.
- > IoT-Core sendet Benachrichtigungen über existierende WebSocket-Verbindung an das Topic myTopic.

## Beispiel: WebSockets nutzen

60213

**Aufgabe:** Es sollen regelmäßig die aktuellen Werte folgender Parameter über eine existierende WebSocket-Verbindung an die Datensenkung myTopic verschickt werden:

- Produktname des IO-Link Devices an IO-Link Port X02
- zyklische Eingangsdaten des IO-Link Devices an IO-Link Port X02
- Betriebstemperatur des IO-Link Masters.

**Lösung:** Mit dem subscribe-Dienst die erforderlichen Daten abonnieren.

- Anfrage (Request):

```
{
"code": "request",
"cid": 4711,
"adr": "/timer[1]/counter/datachanged/subscribe",
"data": {
"callback": "ws:///myTopic",
"datatosend": [
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]
}
```

Sollen die Benachrichtigungen über die existierende WebSocket-Verbindung, aber ohne spezielle Datensenkung übertragen werden, kann der callback-Parameter entfallen.

- Anfrage (Request):

```
{
"code": "request",
"cid": 4711,
"adr": "/timer[1]/counter/datachanged/subscribe",
"data": {
```

```
"datatosend":[  
  "/iolinkmaster/port[2]/iolinkdevice/productname",  
  "/iolinkmaster/port[2]/iolinkdevice/pdin",  
  "/processdatamaster/temperature"]}  
}
```

## 9.2.23 MQTT-Unterstützung

60226

Der IoT Core unterstützt das MQTT-Protokoll. Das Protokoll ermöglicht es einem MQTT-Client, über einen MQTT-Broker mit dem IoT Core zu kommunizieren, um Daten anzufordern und zu empfangen. Der IoT Core kann über die MQTT-Verbindung Daten veröffentlichen (publish).

### MQTT-Kommandokanal konfigurieren

60215

Um eine MQTT-Kommunikation zu ermöglichen, muss der Anwender einen MQTT-Kommandoanal aktivieren und konfigurieren.

Unterstruktur: `connections/mqttConnection`

Name	Beschreibung	Zugriff
<code>../type</code>	Typ der Verbindung (MQTT)	r
<code>../status</code>	globaler MQTT-Status	r
<code>../status/preset</code>	Voreinstellung des MQTT-Status; Grundeinstellung: running	r
<code>../MQTTSetup</code>	Unterstruktur für allgemeine MQTT-Einstellungen	w
<code>../MQTTSetup/QoS</code>	Quality of Service der MQTT-Kommunikation <ul style="list-style-type: none"> <li>0: QoS Level 0 - PUBLISH (ohne Bestätigung)</li> <li>1: QoS Level 1 - PUBLISH &gt; PUBREC (Einfache Bestätigung)</li> <li>2: QoS Level 2 - PUBLISH &gt; PUBREC &gt; PUBREL &gt; PUBCOMP (Doppelte Bestätigung)</li> </ul>	rw
<code>../MQTTSetup/version</code>	MQTT-Version	r
<code>../mqttCmdChannel</code>	Unterstruktur des MQTT-Kommandokanals	w
<code>../mqttCmdChannel/type</code>	Type des MQTT-Kommandokanals	r
<code>../mqttCmdChannel/status</code>	Status des MQTT-Kommandokanals	r
<code>../mqttCmdChannel/status/preset</code>	Voreinstellung des MQTT-Kommandokanal-Status; Grundeinstellung: stopped	r
<code>../mqttCmdChannel/mqttCmdChannelSetup</code>	Struktur für Einstellungen des Kommandokanals	w
<code>../mqttCmdChannel/mqttCmdChannelSetup/brokerIP</code>	IP-Adresse des MQTT-Brokers	rw
<code>../mqttCmdChannel/mqttCmdChannelSetup/brokerPort</code>	Portnummer des MQTT-Brokers	rw
<code>../mqttCmdChannel/mqttCmdChannelSetup/cmdTopic</code>	Bezeichnung des MQTT-Topics	rw
<code>../mqttCmdChannel/mqttCmdChannelSetup/defaultReplyTopic</code>	Standard-Antwort-Topic	rw

Anwendbare Dienste:

Name	Beschreibung
<code>../status/start</code>	MQTT aktivieren
<code>../status/stop</code>	MQTT deaktivieren
<code>../status/reset</code>	MQTT rücksetzen
<code>../mqttCmdChannel/status/start</code>	MQTT-Kommandokanal aktivieren
<code>../mqttCmdChannel/status/stop</code>	MQTT-Kommandokanal deaktivieren
<code>../mqttCmdChannel/status/reset</code>	MQTT-Kommandokanal rücksetzen



Hinweise zu den Zuständen einer MQTT-Verbindung: **Hinweis: Verbindungszustände**  
(→ S. [68](#))

Um eine MQTT-Verbindung zu erzeugen, folgende Schritte nacheinander ausführen:



Sicherstellen, dass der MQTT-Broker erreichbar ist und der gewählte Port des MQTT-Brokers für die Datenübertragung freigegeben ist.

Max. Anzahl gleichzeitiger MQTT-Verbindungen: 10

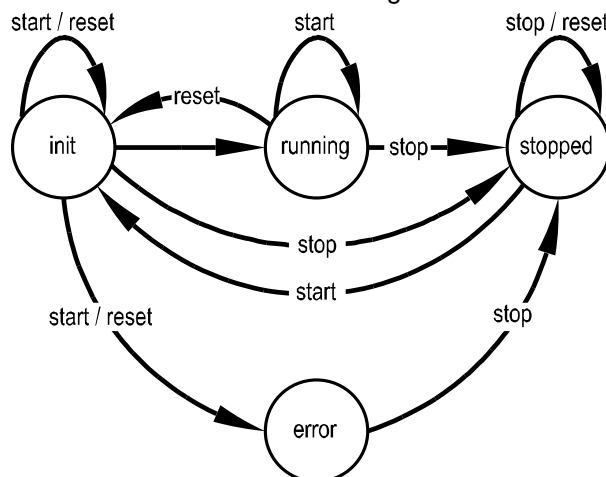
Wildcards "+" und "#" in Topics werden nicht unterstützt.

- ▶ MQTT-Kommandokanal aktivieren.
- ▶ IP-Adresse des MQTT-Brokers setzen.
- ▶ Portnummer des MQTT-Brokers setzen.
- ▶ Topic setzen.
- ▶ Standard-Antwort-Topic setzen.
- > Der Kommandokanal wird mit den gewählten Eigenschaften erzeugt.
- > Der Anwender kann mit dem IoT Core auf das Topic veröffentlichen (publish).
- > MQTT-Clients können das Topic abonnieren (subscribe).

### Hinweis: Verbindungszustände

60231

Das folgende Zustandsdiagramm zeigt den Einfluss der Dienste "start", "stop" und "reset" auf den Zustand einer MQTT-Verbindung:



Nach dem Abschluss der Initialisierung im Zustand "init" wechselt die Verbindung automatisch in den Zustand "running".

Die Verbindung wechselt automatisch in den Zustand "error", wenn mindestens eines der folgenden Ereignisse eintritt:

- kein MQTT-Broker erreichbar

## Beispiel: MQTT-Kommandokanal konfigurieren

**Aufgabe:** MQTT-Kommandokanal konfigurieren und aktivieren (IP-Adresse MQTT-Broker: 192.168.82.100, Port: 1883, Topic: abc).

### Lösung:

► Prüfen, ob MQTT-Broker erreichbar ist und der Port freigegeben wurde.

► Kommandokanal aktivieren

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/start"
}
```

► IP-Adresse des MQTT-Brokers/-Servers einstellen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4712,
  "adr": "/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/brokerIP/setdata"
  "data": {"192.168.82.100"}
}
```

► Portnummer des MQTT-Brokers/-Servers einstellen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4713,
  "adr": "/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/brokerPort/setdata"
  "data": {"1883"}
}
```

► Topic einstellen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4714,
  "adr": "/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/cmdTopic/setdata"
  "data": {"abc"}
}
```

► Standard-Antwort-Topic einstellen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4715,
  "adr": "/connections/mqttConnection/mqttCmdChannel/mqttCmdChannelSetup/defaultReplyTopic/setdata"
  "data": {"xyz"}
}
```

► QoS einstellen.

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4716,
  "adr": "/connections/mqttConnection/MQTTSetup/QoS/setdata",
  "data": {"QoS2"}
}
```

## Beispiel: Temperatur an MQTT-Broker veröffentlichen

17372

**Aufgabe:** Temperatur des IO-Link Masters an einen MQTT-Broker veröffentlichen (IP-Adresse MQTT-Broker: 192.168.82.100, Port: 1883, Topic: abc).

### Lösung:

- Anfrage (Request):

```
{
  "code": "request",
  "cid": -1,
  "adr": "/timer[1]/counter/datachanged/subscribe",
  "data": {
    "callback": "mqtt://192.168.82.100:1883/abc",
    "datatosend": ["processdatamaster/temperature"]
  }
}
```

- Antwort (Response):

```
{
  "cid": -1,
  "code": 200
}
```

## 9.2.24 IoT-Core-Visualizer nutzen

### Inhalt

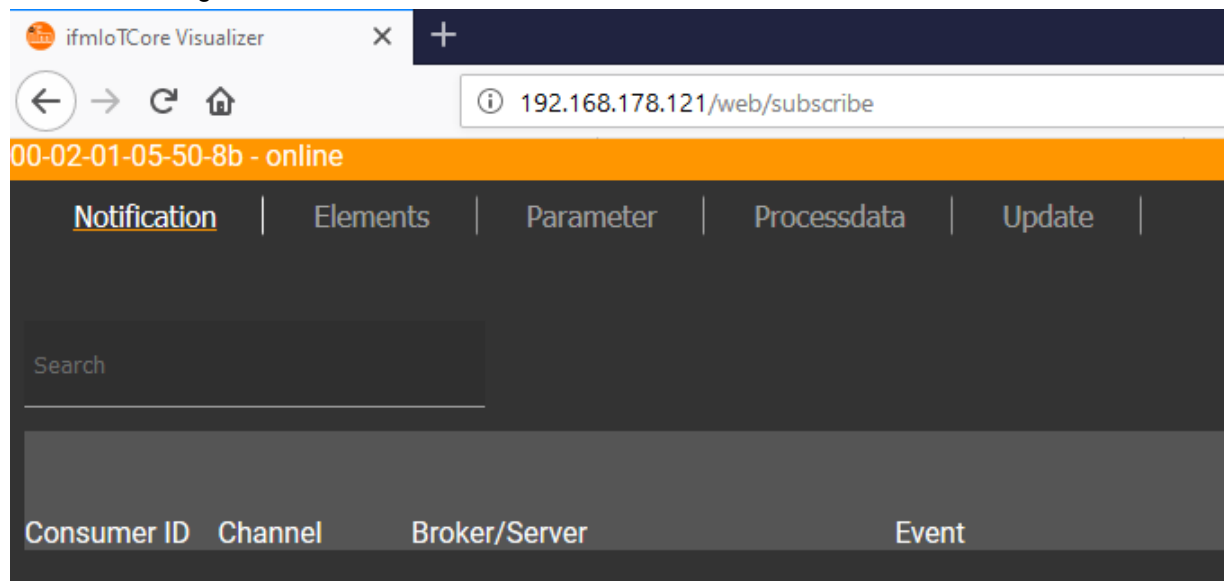
Benachrichtigungen verwalten .....	72
Elemente im Gerätebaum suchen .....	74
IO-Link Master konfigurieren .....	75
Prozessdaten lesen und schreiben .....	76
Firmware aktualisieren .....	77

60270

Der ifm-IO-T-Core-Visualizer des IO-Link Masters bietet eine grafischen Benutzeroberfläche für den Zugriff auf Funktionen des ifm-IO-T-Cores.

Um den IoT-Core-Visualizer zu starten:

- ▶ Webbrowser starten.
- ▶ Folgende Adresse aufrufen: <http://ipaddress/web/subscribe>
- > Browser zeigt IoT-Core-Visualizer:



Über das Navigationsmenü hat der Anwender Zugriff auf folgende Funktionen:

- [Notification]: Benachrichtigungen erstellen und verwalten (subscribe / unsubscribe)
- [Elements]: Elemente in Gerätebeschreibung suchen
- [Parameter]: IO-Link Master parametrieren
- [Processdata]: Prozessdaten lesen und schreiben
- [Update]: Firmware des IO-Link Masters aktualisieren

## Benachrichtigungen verwalten

60474

Die Menüseite ermöglicht es, folgende Funktionen auszuführen

- Benachrichtigungen erstellen
- Aktive Benachrichtigungen anzeigen
- Benachrichtigungen löschen (einzeln, alle)

Voraussetzungen:

- Iot-Core-Visualizer ist gestartet.
- ▶ Auf [Notification] klicken.
- > Menüseite für Verwaltung von Benachrichtigungen erscheint.
- > Menüseite zeigt in Tabelle alle angemeldeten Benachrichtigungen

## Neue Benachrichtigung erstellen

60718

Die Anmeldung neuer Benachrichtigungen erfolgt mithilfe eines Wizards.

Voraussetzungen:

- Menüseite [Notification] ist geöffnet.
- ▶ Am rechten Rand der Tabelle auf [+] klicken.
- > Wizard für die Erstellung von Benachrichtigungen erscheint.

00-02-01-05-50-8b - online

Notification | Elements | Parameter | Processdata | Update

### Add Subscription

1 Events      2 Data      3 Transfer Info

#### Event

Please choose one event, you want to subscribe to.

Search for ... identifier of data element to subscribe to its

- ☐ counter  
00-02-01-05-50-8b/timer[1]/counter/datachanged
- ☐ counter  
00-02-01-05-50-8b/timer[2]/counter/datachanged
- ☐ preset  
00-02-01-05-50-8b/connections/mqttconnection/status/preset/datachanged
- ☐ status  
00-02-01-05-50-8b/connections/mqttconnection/status/datachanged
- ☐ qos  
00-02-01-05-50-8b/connections/mqttconnection/mqttsetup/qos/datachanged
- ☐ preset  
00-02-01-05-50-8b/connections/mqttconnection/mqtcmdchannel/status/preset/datachanged

CANCEL NEXT >



- ▶ Mithilfe des Wizards schrittweise die gewünschten Benachrichtigungsparameter eingeben.
- > Erstellte Benachrichtigungsabonnement wird in Tabelle angezeigt.



Bei zyklischen Benachrichtigungen über timer[1] oder timer[2] muss der Anwender zusätzlich die Intervallzeit des betreffenden Timers setzen.

## Benachrichtigung löschen

60719

Voraussetzungen:

- Menüseite [Notification] ist geöffnet.
- Mindestens eine Benachrichtigung ist aktiv.
- ▶ In Spalte [Unsubscribe] auf [x] klicken.
- > Gewählte Benachrichtigung wird gelöscht (unsubscribe).

## Elemente im Gerätebaum suchen

Die Menüseite [Elements] ermöglicht es, die Gerätebeschreibung nach Elementen mit bestimmten Eigenschaften (status, profile, name) zu durchsuchen und die Ergebnisse auszugeben.

Voraussetzungen:

- IoT-Core-Visualizer ist gestartet.
- ▶ Auf [Elements] klicken.
- > Eingabemaske erscheint.

00-02-01-05-50-8b - online

Notification | **Elements** | Parameter | Processdata | Update

Search for ...

identifier	one identifier to search for
profile	profile(s) to search for. Seperator ','
type	one type to search for

Processdatamaster | Deviceinfo | Timer[1] | Timer[2] | lotsetup | Fieldbussetup | Connections | Iolinkmaster

Firmware | Devicetag

^ 00-02-01-05-50-8b

getidentity	00-02-01-05-50-8b/getidentity	type: service profiles: undefined	Copy URL
gettree	00-02-01-05-50-8b/gettree	type: service profiles: undefined	Copy URL
querytree	00-02-01-05-50-8b/querytree	type: service profiles: undefined	Copy URL

- ▶ In Eingabefeldern [identifier], [profile] und [type] die Suchkriterien des gewünschten Elements eingeben.
- ▶ Auf [Search for ...] klicken.
- > IoT-Core-Visualizer durchsucht Gerätebeschreibung nach Elementen mit gewählten Suchkriterien.
- > Ergebnisliste zeigt alle gefundenen Elemente.

## IO-Link Master konfigurieren

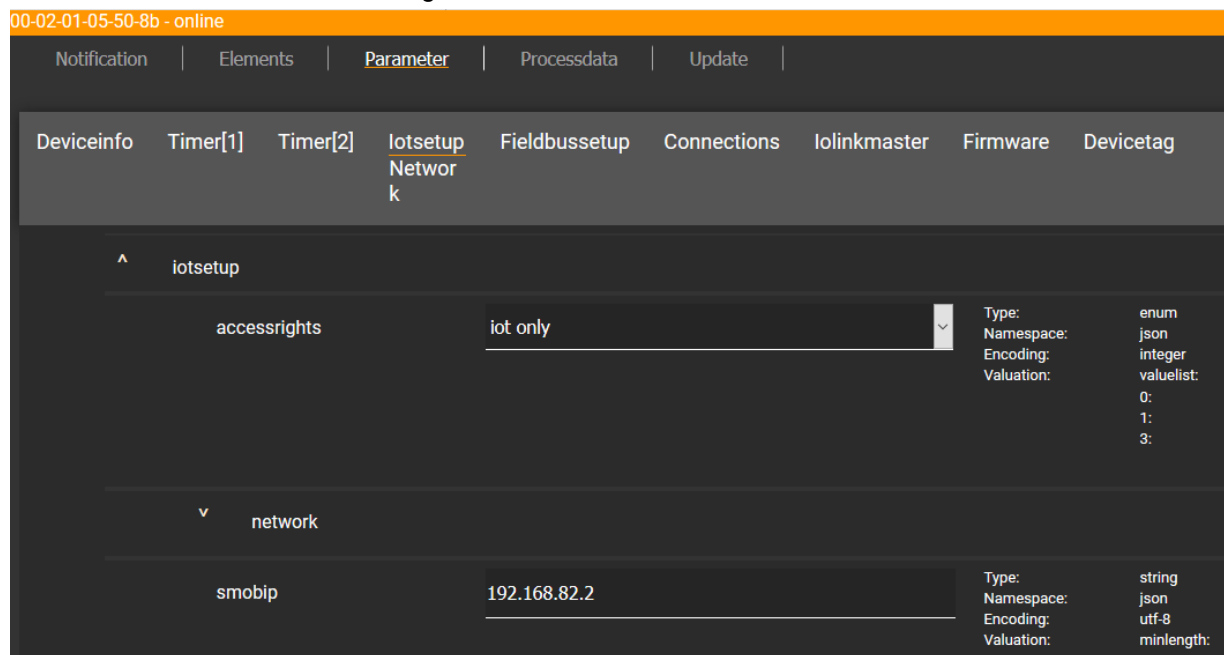
Die Menüseite [Parameter] ermöglicht es, den IO-Link Master zu konfigurieren.

Verfügbare Optionen:

- Einzelne Parameter lesen und schreiben.
- Aktuelle Konfiguration des Geräts sichern und wiederherstellen.

Voraussetzungen:

- Iot-Core-Visualizer ist gestartet.
- ▶ Auf [Parameter] klicken.
- > Menüseite zeigt verfügbare Parameter des IO-Link Masters.
- > Aktuelle Parameterwerte werden angezeigt.
- > Editierbare Parameter können geändert werden.



Um einen Parameter zu ändern:

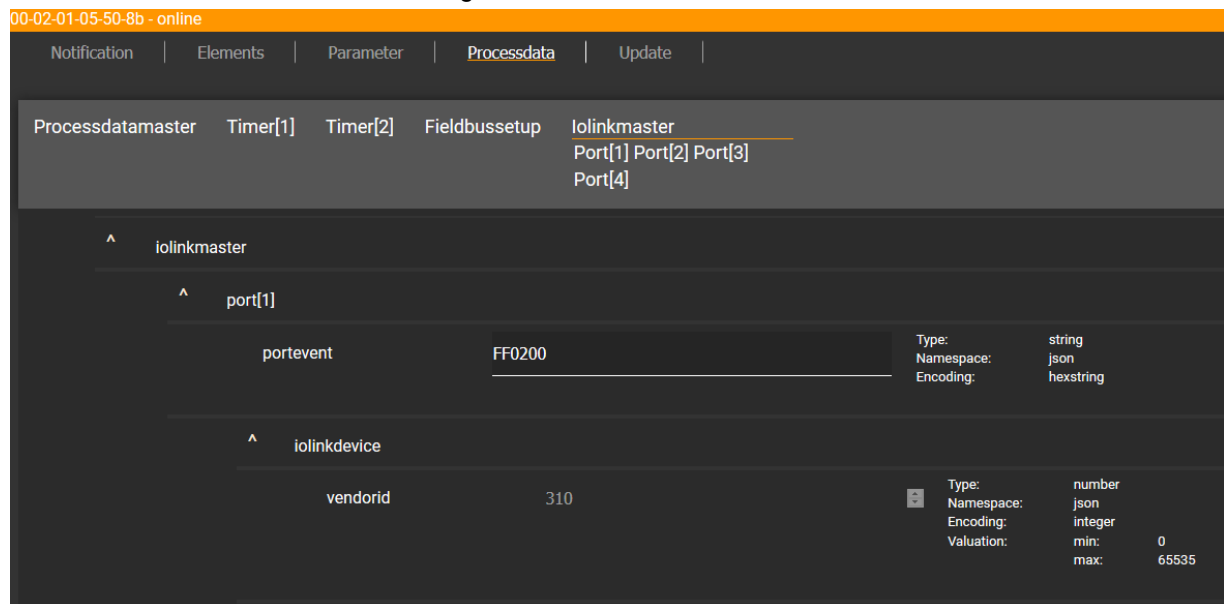
- ▶ In Gerätebeschreibung zum gewünschten Parameter navigieren.
- ▶ Parameterwert ändern.
- ▶ Auf Stift-Symbol klicken, um die Änderung auf dem IO-Link Master zu speichern.
- > Geänderter Parameterwert ist aktiv.
- ▶ Optional: Vorgang wiederholen, um weitere Parameterwerte zu ändern.

## Prozessdaten lesen und schreiben

Die Menüseite ermöglicht es, die Prozessdaten des IO-Link Masters und der angeschlossenen IO-Link Devices zu lesen und zu schreiben.

Voraussetzungen:

- lot-Core-Visualizer ist gestartet.
- ▶ Auf [Processdata] klicken.
- > Menüseite zeigt die Unterstrukturen der Gerätebeschreibung, die Prozessdaten und Events enthalten.
- > Aktuelle Prozesswerte werden angezeigt.
- > Editierbare Prozessdaten können geändert werden.



Um den Wert eines Prozessdatums zu ändern:

- ▶ In Gerätebeschreibung zum gewünschten Prozessdatum navigieren.
- ▶ Prozesswert ändern.
- ▶ Auf Stift-Symbol klicken, um die Änderung auf dem IO-Link Master zu speichern.
- > Geänderter Prozesswert ist aktiv.
- ▶ Optional: Vorgang wiederholen, um weitere Prozesswerte zu ändern.

## Firmware aktualisieren

Die Menüseite [Update] bietet die Möglichkeit, die Firmware des IO-Link Masters zu aktualisieren:

Voraussetzungen:

- lot-Core-Visualizer ist gestartet.
- Auf [Update] klicken.
- > Menüseite zeigt Informationen zur aktuellen Firmware-Version.

00-02-01-05-50-8b - online

Notification	Elements	Parameter	Processdata	Update																
<h3>Firmware</h3> <p>00-02-01-05-50-8b/firmware</p> <table border="1"> <thead> <tr> <th colspan="2">Version:</th> <th>AL1x2x_cn_ei_v3.1.44</th> <th>Container</th> </tr> </thead> <tbody> <tr> <td>Type:</td> <td>firmware</td> <td></td> <td>Max size: 4194304</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Chunk size: 4096</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Size: 0</td> </tr> </tbody> </table> <div> <div>Load software file</div> <div>choose software package</div> </div> <div>Update</div>					Version:		AL1x2x_cn_ei_v3.1.44	Container	Type:	firmware		Max size: 4194304				Chunk size: 4096				Size: 0
Version:		AL1x2x_cn_ei_v3.1.44	Container																	
Type:	firmware		Max size: 4194304																	
			Chunk size: 4096																	
			Size: 0																	

- Auf [Load software file] klicken und neue Firmware-Datei (\*.bin) wählen.
- Auf [Update] klicken, um den Aktualisierungsprozess zu starten.
- > Firmware des IO-Link Masters wird aktualisiert.
- > Bereich zeigt Fortschrittsanzeige.
- > Wenn Aktualisierungsprozess erfolgreich, dann startet der IO-Link Master automatisch neu.

## 9.3 EtherCAT

### Inhalt

ESI-Datei installieren .....	78
IO-Link Master in TwinCAT-Projekt einbinden .....	78
IO-Link-Master konfigurieren .....	79
IO-Link-Ports konfigurieren.....	79
Zyklische Prozessdaten konfigurieren.....	80
Zyklische Prozessdaten lesen und schreiben .....	81
Diagnose- und Statusinformationen lesen .....	82
IO-Link Events lesen .....	82
IO-Link Devices konfigurieren .....	83
EtherCAT: Hinweise für Programmierer .....	84

17299

Das Gerät kann feldbusseitig mit jeder EtherCAT-fähigen Projektierungssoftware konfiguriert werden. Die Angaben in den folgenden Abschnitten beziehen sich auf die EtherCAT-Projektierungssoftware TwinCAT 3.1.

### 9.3.1 ESI-Datei installieren

12762

Für die Abbildung des AL1930 in einer Feldbus-Projektierungssoftware stellt ifm electronic eine ESI-Datei bereit (→ [www.ifm.com](http://www.ifm.com)). In der ESI-Datei sind alle Parameter- und Prozessdaten sowie deren gültige Wertebereiche definiert.

Um die ESI-Datei in die EtherCAT-Projektierungssoftware TwinCAT 3.1 einzubinden:

- ▶ ESI-Datei des Geräts herunterladen.
- ▶ Heruntergeladene Datei in folgendes Unterverzeichnis des TwinCAT-Installationsverzeichnisses kopieren:  
`..\3.1\Config\Io\EtherCAT`
- ▶ TwinCAT starten.
- > TwinCAT lädt die Gerätebeschreibung in den Gerätekatalog.

### 9.3.2 IO-Link Master in TwinCAT-Projekt einbinden

17297

Das Gerät wird als EtherCAT Slave in das TwinCAT-Projekt eingebunden.

#### Voraussetzungen

- > ESI-Datei des IO-Link Masters ist installiert (→ **ESI-Datei installieren** (→ S. 78)).

#### 1 EtherCAT-Projekt erstellen/öffnen

- ▶ EtherCAT-Projektierungssoftware starten.
- ▶ Neues Projekt erstellen.  
 ODER  
 Existierendes Projekt öffnen.

#### 2 EtherCAT-SPS und I/O-Peripherie konfigurieren

- ▶ EtherCAT-SPS und gewünschte I/O-Peripherie wählen und konfigurieren.
- > Projekt enthält EtherCAT-SPS und I/O-Peripherie.

#### 3 IO-Link Master in Projekt einbinden

- ▶ Im Solution Explorer: Rechtsklick auf Gerät, mit dem der IO-Link Master verbunden ist.
- > Kontextmenü erscheint.
- ▶ Im Kontextmenü: [Add new item...] wählen.
- > Fenster [Insert EtherCAT Device] erscheint.
- ▶ Im Gerätebaum unter [ifm electronic] > [ifm IO-Link Master] den IO-Link Master wählen.

- ▶ [OK] klicken, um das gewählte Gerät in das Projekt einzufügen.
- > TwinCAT fügt IO-Link Master zum Projekt hinzu.

#### 4 ID des AL1930 einstellen

- > In Solution Explorer: Doppelklick auf Knoten des IO-Link Masters.
- > Detailansicht zeigt aktuelle Einstellungen.
- > Registerkarte [General] wählen.
- > In Feld [Id] die gewünschte ID des IO-Link Masters wählen.

#### 5 Projekt speichern

- ▶ Projekt speichern.

### 9.3.3 IO-Link-Master konfigurieren

18602

Der Konfiguration des IO-Link Masters erfolgt über das CoE-Interface (→ TwinCAT-Online-Hilfe). Die Konfiguration erfolgt über die folgenden Parameter:

Name	Beschreibung	Referenz
Current Use Case	Zugriffsrechte auf den IO-Link Master (Access Rights)	→ <b>Manufacturer Specific Index (0x2000)</b> (→ S. <a href="#">100</a> )
Reset To Factory	IO-Link Master auf Werkseinstellungen rücksetzen	→ <b>Manufacturer Specific Index (0x2000)</b> (→ S. <a href="#">100</a> )

### 9.3.4 IO-Link-Ports konfigurieren

17959

Der Konfiguration der IO-Link-Ports erfolgt über das CoE-Interface (→ TwinCAT-Online-Hilfe). Der Nutzer kann jeden IO-Link-Port separat konfigurieren. Die Konfiguration erfolgt über die folgenden Parameter:

Name	Beschreibung	Referenz
IO Settings	Konfiguration der IO-Link-Ports X01...X08	→ <b>Port Configuration (0x8000)</b> (→ S. <a href="#">103</a> )
Vendor Specific IO Settings	Herstellerspezifische Einstellungen der IO-Link-Ports X01...X08	→ <b>Port Configuration (0x8000)</b> (→ S. <a href="#">103</a> )



Der AL1930 besitzt keine Failsafe-Funktion für die Ausgänge der IO-Link Ports. Bei einer Unterbrechung der Feldbusverbindung werden die zuletzt genutzten Ausgangswerte geschrieben und als ungültig gekennzeichnet.

### 9.3.5 Zyklische Prozessdaten konfigurieren

Art und Umfang der zyklischen Eingangs- und Ausgangsdaten an den IO-Link-Ports werden über die Feldbus-Module definiert (→ **EtherCAT-Module** (→ S. [97](#))). In den Werkseinstellungen sind alle Slots mit dem Modul "IOL\_4/4\_I/O" konfiguriert.

Um die zyklischen Prozessdaten zu konfigurieren:

#### Voraussetzungen

- > AL1930 ist in das Gerät eingebunden (→ **IO-Link Master in TwinCAT-Projekt einbinden** (→ S. [78](#), "**Beispiel: Eigenschaften eines Elements lesen**" → S. [41](#))).

#### 1 Geräte-Editor öffnen

- ▶ Im Solution Explorer: Doppelklick auf Knoten des AL1930.
- > Fenster zeigt verfügbare Konfigurationsoptionen.
- ▶ Registerkarte [Slots] wählen.
- > Fenster zeigt die aktuelle Konfiguration der zyklischen Daten.

#### 2 Feldbusmodule zuweisen

- ▶ In linker Tabellenhälfte: Slot klicken des gewünschten IO-Link-Ports wählen.
- ▶ In rechter Tabellenhälfte: Gewünschtes Feldbusmodul klicken.
- ▶ Auf [<] klicken, um das gewählte Feldbusmodul dem Slot zuzuweisen.

#### 3 Weitere IO-Link-Ports konfigurieren

- ▶ Optional: Schritt 1 für weitere IO-Link-Ports wiederholen.
- > Zyklische Daten sind den Feldbus-Slots zugeordnet.
- > Der Solution Explorer zeigt die konfigurierten Module als Unterelemente des Geräteknotens.

#### 4 Projekt speichern

- ▶ Projekt speichern.



### 9.3.6 Zyklische Prozessdaten lesen und schreiben



- Um die Gültigkeit der zyklischen Prozessdaten zu prüfen, das PQI-Byte auswerten (→ **Mapping: Port Qualifier (0xF101)** (→ S. 107)).

Auch bei einer Unterbrechung der Feldbusverbindung zeigt das PQI-Byte an, dass die Prozessdaten gültig sind. Dies kann ungewollte Auswirkungen auf den Steuerungsprozess haben.

- Geeignete Maßnahmen treffen, um eine Unterbrechung der Feldbusverbindung zu erkennen.

Die zyklischen Eingangsdaten der IO-Link-Ports X01...X08 sind über folgende Indexgruppe zugänglich:

Name (Index)	Beschreibung	Referenz
IO-Link Inputs (0x6000)	Zyklische Eingangsdaten an den IO-Link-Ports X01...X08	→ <b>IO-Link Inputs (0x6000)</b> (→ S. 102)
IO-Link Outputs (0x7000)	Zyklische Ausgangsdaten an den IO-Link-Ports X01...X08	→ <b>IO-Link Outputs (0x7000)</b> (→ S. 102)



In der Betriebsart "Digital Input (DI)" wird der digitale Prozesswert des IO-Link Ports im 1. Byte der Input-Daten abgebildet (Subindex 0x01).

In der Betriebsart "Digital Output (DO)" wird der digitale Prozesswert des IO-Link Ports im 1. Byte des Output-Daten abgebildet (Subindex 0x01).

Gültige Werte:

- 0x000 = OFF
- 0x001 = ON

Bei der Konfiguration der Feldbus-Slots erzeugt TwinCAT im Solution Explorer automatisch Variablen für die zyklischen Eingangs- und Ausgangsdaten. Sie liegen gruppiert in den Ordnern unterhalb des jeweiligen Feldbus-Moduls. Der Anwender kann die Variablen direkt mit den Elemente einer Globalen Variablenliste (GVL) verknüpfen.

Folgende Variablen werden erzeugt:

Gruppe > Variable	Variable	Beschreibung
[TxPDO]	[input byte n]	Byte n der zyklischen Eingangsdaten des Feldbus-Moduls
[RxPDO]	[output byte m]	Byte m der zyklischen Ausgangsdaten des Feldbus-Moduls

n = 0...(Max. Anzahl der Bytes an konfigurierten Eingangsdaten)-1

m = 0...(Max. Anzahl der Bytes an konfigurierten Ausgangsdaten)-1

### 9.3.7 Diagnose- und Statusinformationen lesen

17961

Diagnose- und Statusinformationen sind über folgende Indexgruppen zugänglich:

Name (Index)	Beschreibung	Referenz
MDP Standard Information (0x1000)	<ul style="list-style-type: none"> <li>Geräteinformationen über IO-Link Master</li> <li>Identity Object</li> <li>Time Stamp</li> </ul>	→ <b>MDP Standard Information (0x1000)</b> (→ S. <a href="#">99</a> )
Port Mode (0x9000)	Informationen über IO-Link Devices an IO-Link-Ports X01...X08	→ <b>Port Mode (0x9000)</b> (→ S. <a href="#">104</a> )
Diagnostics Data (0xA000)	Diagnosedaten der IO-Link-Ports X01...X08	→ <b>Diagnostics Data (0xA000)</b> (→ S. <a href="#">104</a> )
Device Status (0xF000)	<ul style="list-style-type: none"> <li>Status des IO-Link Devices an IO-Link-Port X01...X08</li> <li>Port Qualifier</li> </ul>	→ <b>Device Status / Port Status (0xF000)</b> (→ S. <a href="#">105</a> )

Beim Einbinden des IO-Link Masters in ein EtherCAT-Projekt erzeugt TwinCAT im Solution Explorer automatisch Variablen für Diagnose- und Statusinformationen. Sie liegen gruppiert in Ordnern unterhalb des Geräteknotens. Der Anwender kann die Variablen direkt mit den Elemente einer Globalen Variablenliste (GVL) verknüpfen.

Folgende Variablen werden erzeugt:

Gruppe > Variable	Beschreibung
[TxPDO IO-Link Device Status] > [State of IO-Link Ch.n]	Status des IO-Link Devices an IO-Link-Port X0n
[TxPDO IO-Link Port Qualifier] > [Qualifier of IO-Link Ch.n]	Port Qualifier Bits des IO-Link-Ports X0n
[TxPDO New Diagnosis Message available] > [New Message Available Flag]	Benachrichtigung über neue Diagnosemeldungen

n ... 1...8

### 9.3.8 IO-Link Events lesen

11750

IO-Link Events werden in der "Diagnosis History" gespeichert (→ **Diagnosis History (0x10F3)** (→ S. [108](#))).

Der IO-Link Master speichert maximal 64 Events.

Die einzelnen Events werden in einem Ringpuffer gespeichert. Der AL1930 unterstützt folgende Betriebsmodi für das Schreiben des Pufferspeichers:

- Overwrite Mode: Ist der Puffer voll, wird das älteste Event überschrieben.
- Acknowledge Mode: Events werden erst überschrieben, wenn sie gelesen und bestätigt wurden.

Die Konfiguration erfolgt über den Subindex 0x05. Die Events werden in den Subindizes 0x06...0x46 gespeichert.

### 9.3.9 IO-Link Devices konfigurieren

9031

Der IO-Link Master unterstützt die Konfiguration der angeschlossenen IO-Link Devices aus der EtherCAT-Projektierungssoftware heraus. Der Zugriff auf die Parameter eines IO-Link Devices erfolgt über IO-Link Index und Subindex. Die Anzahl der einstellbaren Parameter ist abhängig vom angeschlossenen IO-Link Device.



Verfügbare Parameter der IO-Link Devices: → IO Device Description (IODD) des IO-Link Device

Der Anwender kann IO-Link Index und Subindex mit folgenden Methoden lesen und schreiben:

- Azyklische Kommunikation (→ **Azyklische Dienste nutzen** (→ S. [84](#)))

## 9.3.10 EtherCAT: Hinweise für Programmierer

### Inhalt

Azyklische Dienste nutzen.....	84
--------------------------------	----

17965

### Azyklische Dienste nutzen

17958

Der AL1930 unterstützt folgende Dienste für azyklische Lese- und Schreibvorgänge:

- AoE - ADS over EtherCAT (→ **ADS over EtherCAT nutzen** (→ S. [84](#)))
- CoE - CANopen over EtherCAT (→ **CANopen over EtherCAT nutzen** (→ S. [85](#)))

### ADS over EtherCAT nutzen

9109

AoE eignet sich für den unterbrechungsfreien Zugriff auf die angeschlossenen IO-Link Devices während der Laufzeit des Geräts. Der Zugriff auf den IO-Link Master mit AoE wird nicht unterstützt.



Die FBs für die Nutzung von AoE sind Bestandteil der TwinCAT-Bibliothek `tc2_system.lib`.

Für die Eingänge der ADS-Funktionsbausteine gelten folgende Regeln:

Eingang	Beschreibung		Mögliche Werte	
PORT	ADS Communication Port = 0x1000 + IO-Link-Portnummer		0x1001 0x1002 ... 0x1008	IO-Link-Port X01 IO-Link-Port X02 ... IO-Link-Port X08
IDXGRP	AoE-Index-Gruppe		0xF302	
IDXOFFS	Index Offset		z. B. Zugriff auf Index 21, Subindex 0: 0x0021°0x00°0x00	
	Bits 0-7:	IO-Link Subindex		
	Bits 8-15:	00000000		
	Bits 16-31:	IO-Link Index		
ERRID	ADS-Fehlercode		z. B. Zugriff auf Parameter des IO-Link Devices verweigert: 0x0700°8023	
	Bits 0-15:	Fehlercode vom IO-Link Device		
	Bits 16-31:	ADS device error = 0x0700		

## CANopen over EtherCAT nutzen

16206

CoE eignet sich für den azyklischen Zugriff auf den IO-Link Master und die angeschlossenen IO-Link Devices. CoE nutzt die Feldbusobjekte "IO-Link Acyclic Command" (→ **IO-Link Acyclic Command (0x3100)** (→ S. 101)). Für jeden IO-Link-Port steht ein separates Feldbusobjekt zur Verfügung.

Um azyklisch per CoE auf das Gerät zuzugreifen, kann der Anwender die folgenden Funktionsblöcke nutzen:

- FB\_EcCoESdoRead: SDO eines EtherCAT-Slaves lesen
- FB\_EcCoeSdoWrite: SDO eines EtherCAT-Slave schreiben



Die FBs für die Nutzung von CoE sind Bestandteil der Funktionsbibliothek Tc2\_EtherCAT.library.

► Funktionsbibliothek Tc2\_EtherCAT.library zum Projekt hinzufügen

Beschreibung der Funktionsblöcke: → Hilfe-Funktion von TwinCAT

Für die Eingänge der CoE-Funktionsblöcke gelten folgende Regeln:

Eingang	Beschreibung	Mögliche Werte	
sNetId	AMS-Net-ID des <Feldbsu>-Masters, an den der IO-Link Master angeschlossen ist	projektabhängig; z.B. 172.16.2.131.2.1	
nSlaveAddr	EtherCAT-Adresse des IO-Link-Ports am IO-Link Master = 0x1000 + IO-Link-Portnummer	0x1001 0x1002 ... 0x1008	IO-Link-Port X01 IO-Link-Port X02 ... IO-Link-Port X08
nSubIndex	IO-Link Subindex des Parameters	geräteabhängig; → IODD	
nIndex	IO-Link Index des Parameters	geräteabhängig; → IODD	

## Prinzip der azyklischen Kommandoabarbeitung

6937

Genereller Ablauf der azyklischen Kommunikation:

### 1 Command Request schreiben

- ▶ Im Command Buffer (0x310n:Subindex 0x01): Gewünschte Befehlsdaten schreiben.
- > Anforderungsdaten werden übertragen.
- > Verarbeitung des Befehls wird gestartet.
- > Anforderungskanal wird geblockt.

### 2 Status prüfen

- ▶ Im Status-Byte (0x310n:Subindex 0x02): Status lesen.
  - Wenn Status == 0xFF: Befehl wird abgearbeitet, Schritt 2 wiederholen.
  - Wenn Status < 0xFF: Befehlsabarbeitung beendet, weiter mit Schritt 3

### 3 Command Response lesen

- ▶ Im Response Buffer (0x310n:Subindex 0x03): Rückgabedaten lesen.
- > Blockierung des Anforderungskanals wird aufgehoben.
- > Nächster Befehl kann gestartet werden.



Der IO-Link Master kann jeweils nur eine CoE-Anfrage verarbeiten. Wird während einer laufenden Anfrage eine weitere CoE-Anfrage gestartet, antwortet das Gerät mit einem Fehler (SDO-Abbruch-Code: 0x06090030).

## 10 Betrieb

### Inhalt

Web-based Management nutzen .....	87
-----------------------------------	----

22368

### 10.1 Web-based Management nutzen

60872

Das Gerät verfügt über einen integrierten Web-Server. Der Web-Server generiert eine Webseite mit folgenden Daten:

- Statusinformationen der Ports
- Zugriff auf Produktseite angeschlossener IO-Link Devices (nur ifm-Geräte)
- Diagnoseinformationen des Geräts
- Versionsinformationen der installierten Firmware-Komponenten

Um auf die Web-Schnittstelle des IO-Link Masters zuzugreifen:

- ▶ IO-Link Master über den IoT-Port mit Laptop / PC verbinden.
- ▶ Optional: IP-Einstellungen des IoT-Schnittstelle prüfen.
- ▶ Webbrowser starten.
- ▶ Im Adressfeld des Webbrowsers der IP-Adresse der IoT-Schnittstelle eingeben und mit [ENTER] bestätigen.
- > Webbrowser zeigt die Webseite mit Status- und Diagnoseinformationen des Geräts.

# 11 Wartung, Instandsetzung und Entsorgung

## Inhalt

Reinigung .....	88
Firmware aktualisieren .....	88
IO-Link Device tauschen .....	88

16306

Der Betrieb des Geräts ist wartungsfrei.

- ▶ Gerät nach dem Gebrauch gemäß den gültigen nationalen Bestimmungen umweltgerecht entsorgen.

## 11.1 Reinigung

7127

- ▶ Die Oberfläche des Geräts bei Bedarf reinigen.
- ▶ Verschmutzungen mit einem weichen, chemisch unbehandelten und trockenen Tuch entfernen.
- ▶ Bei starker Verschmutzung ein feuchtes Tuch verwenden.
- ▶ Für die Reinigung keine ätzenden Reinigungsmittel verwenden!

## 11.2 Firmware aktualisieren

60870

Die Firmware des IO-Link Masters kann über den IoT-Core-Visualizer aktualisiert werden (→ **Firmware aktualisieren** (→ S. [77](#))).

## 11.3 IO-Link Device tauschen

7775

Um ein IO-Link Device zu tauschen:

### Voraussetzung:

- > Neues IO-Link Device ist im Auslieferungszustand (Werkseinstellungen).
- > Neues IO-Link Device unterstützt IO-Link-Standard 1.1 oder höher.

### 1 Datenspeicherung einstellen

- ▶ Folgende Parameter des IO-Link-Ports einstellen
  - Gerätevalidierung und Datenspeicherung auf [Type compatible V1.1 device with Restore] oder [Type compatible V1.1 device with Backup + Restore] setzen.
  - Korrekte Werte für [Vendor ID] und [Device ID] setzen entsprechend der Eigenschaften des IO-Link Devices.
- ▶ Änderungen speichern.

### 2 IO-Link Device tauschen

- ▶ Altes IO-Link Device vom IO-Link Master trennen.
- ▶ Neues IO-Link Device mit dem gleichen IO-Link-Port des AL1930 verbinden.
- > IO-Link Master kopiert Parameterwerte aus dem Datenspeicher auf das neue IO-Link Device.



## 12 Werkseinstellungen

16549

In den Werkseinstellungen besitzt das Gerät folgende Parametereinstellungen:

Parameter	Werkseinstellung
[IP address] (IoT-Schnittstelle)	169.254.X.X
[Subnet mask] (IoT-Schnittstelle)	255.255.0.0
[IP gateway address] (IoT-Schnittstelle)	0.0.0.0
[Host name]	leer
Datenspeicher (Data Storage)	leer

## 13 Zubehör

17853

Zubehörliste des AL1930: → [www.ifm.com](http://www.ifm.com) > Produktseite > Zubehör

## 14 Anhang

### Inhalt

Technische Daten.....	92
EtherCAT .....	96
ifm-IoT-Core .....	112

7156

## 14.1 Technische Daten

### Inhalt

Einsatzbereich .....	92
Elektrische Daten .....	92
Eingänge / Ausgänge .....	92
Eingänge .....	93
Ausgänge .....	93
Schnittstellen .....	93
Umgebungsbedingungen .....	94
Zulassungen / Prüfungen .....	94
Mechanische Daten .....	94
Elektrischer Anschluss .....	95

9011

### 14.1.1 Einsatzbereich

23710

Einsatzbereich	
Applikation	E/A-Module für Schaltschrank
Durchschleiffunktion	Feldbusschnittstelle

### 14.1.2 Elektrische Daten

22819

Elektrische Daten	
Betriebsspannung [V]	20...30 DC; (US; nach SELV/PELV)
Stromaufnahme [mA]	300...3900; (US)
Schutzklasse	III
Sensorversorgung US	
Strombelastbarkeit gesamt [A]	3,6

### 14.1.3 Eingänge / Ausgänge

23711

Ein-/Ausgänge	
Gesamtzahl der Ein- und Ausgänge	16; (konfigurierbar)
Anzahl der Ein- und Ausgänge	Anzahl der digitalen Eingänge: 16; Anzahl der digitalen Ausgänge: 8

## 14.1.4 Eingänge

22820

Eingänge	
Anzahl der digitalen Eingänge	16; (IO-Link Port Class A)
Schaltpegel High [V]	11...30
Schaltpegel Low [V]	0...5
Kurzschlussfest	ja

## 14.1.5 Ausgänge

22821

Ausgänge	
Anzahl der digitalen Ausgänge	8; (IO-Link Port Class A)
Strombelastbarkeit je Ausgang [mA]	300
Kurzschlussfest	ja

## 14.1.6 Schnittstellen

17947

Schnittstellen	
Kommunikationsschnittstelle	Ethernet; IO-Link
Kommunikationsschnittstelle	IO-Link; TCP/IP; EtherCAT
<b>Ethernet</b>	
Übertragungsstandard	10Base-T; 100Base-TX
Übertragungsrate [MBit/s]	10; 100
Protokoll	EtherCAT, Zero config
Werkseinstellungen	MAC-Adresse: siehe Typenschild
<b>IO-Link Master</b>	
Übertragungstyp	COM 1 / COM 2 / COM 3
IO-Link-Revision	V1.1
Anzahl Ports Class A	8
<b>IoT-Schnittstelle</b>	
Übertragungsstandard	10Base-T; 100Base-TX
Übertragungsrate [MBit/s]	10; 100
Protokoll	DCP, DHCP, Auto IP
Werkseinstellungen	<ul style="list-style-type: none"> <li>▪ IP-Adresse: 169.254.X.X</li> <li>▪ Subnetzmaske: 255.255.0.0</li> <li>▪ Gateway IP-Adresse: 0.0.0.0</li> <li>▪ MAC-Adresse: siehe Typenschild</li> </ul>

## 14.1.7 Umgebungsbedingungen

17862

Umgebungsbedingungen	
Einsatzort	Schaltschrank
Umgebungstemperatur [°C]	-25...65
Lagertemperatur [°C]	-25...85
Max. zulässige relative Luftfeuchtigkeit [%]	90, linear abnehmend bis 50% (40 °C)
Max. Höhe über NN [m]	2000
Schutzart	IP 20
Verschmutzungsgrad	2

## 14.1.8 Zulassungen / Prüfungen

22824

Zulassungen / Prüfungen	
EMV	<ul style="list-style-type: none"> <li>▪ EN 61000-6-2</li> <li>▪ EN 61000-6-4</li> </ul>
MTTF [Jahre]	90

## 14.1.9 Mechanische Daten

22825

Mechanische Daten	
Gewicht [g]	330,5
Werkstoffe	Gehäuse: PA

## 14.1.10 Elektrischer Anschluss

Spannungsversorgung IN X31	
Steckverbindung	COMBICON
Anschlussbelegung	1: GND (US) 2: GND (US) 3: + 24 V DC (US) 4: + 24 V DC (US)
Prozessanschluss IO-Link Ports Class A X01...X08	
Steckverbindung	COMBICON
Anschlussbelegung	1: Sensorversorgung (US) L+ 2: DI 3: Sensorversorgung (US) L- 4: C/Q IO-Link
Ethernet IN / OUT X21, X22	
Steckverbindung	RJ-45
IoT X23	
Steckverbindung	RJ-45

## 14.2 EtherCAT

### Inhalt

Parameterdaten.....	97
Zyklische Daten.....	98
Azyklische Daten.....	99
Events.....	108

22433



## 14.2.1 Parameterdaten

### Inhalt

Feldbusparameter .....	97
EtherCAT-Module .....	97
ESI-Datei .....	97

1568

### Feldbusparameter

8953

Die Feldbusparameter enthalten Informationen für die Integration des Geräts in das EtherCAT-Netzwerk:

Name	Beschreibung	Mögliche Werte
EtherCAT-Adresse	"Explicit Device ID" des Geräts	0...65534

### EtherCAT-Module

9461

Modul	Beschreibung	
IOL_In_4Byte	IO-Link aktiviert	4 Byte Eingangsdaten
IOL_In_8Byte		8 Byte Eingangsdaten
IOL_In_16Byte		16 Byte Eingangsdaten
IOL_In_32Byte		32 Byte Eingangsdaten
IOL_Out_4Byte		4 Byte Ausgangsdaten
IOL_Out_8Byte		8 Byte Ausgangsdaten
IOL_Out_16Byte		16 Byte Ausgangsdaten
IOL_Out_32Byte		32 Byte Ausgangsdaten
IOL_4/4_I/O-Bytes		4 Byte Eingangsdaten / 4 Byte Ausgangsdaten (Default)
IOL_8/8_I/O-Bytes		8 Byte Eingangsdaten / 8 Byte Ausgangsdaten
IOL_4/16_I/O-Bytes		4 Byte Eingangsdaten / 16 Byte Ausgangsdaten
IOL_16/4_I/O-Bytes		16 Byte Eingangsdaten / 4 Byte Ausgangsdaten
IOL_16/16_I/O-Bytes		16 Byte Eingangsdaten / 16 Byte Ausgangsdaten
IOL_32/32_I/O-Bytes		32 Byte Eingangsdaten / 32 Byte Ausgangsdaten
Digital_IN	IO-Link deaktiviert	Digitaler Eingang
Digital_OUT		Digitaler Ausgang
Deactivated	deaktiviert	

### ESI-Datei

10277

Für die Abbildung des AL1930 in einer Feldbus-Projektierungssoftware stellt ifm electronic eine ESI-Datei bereit. Die ESI-Datei kann von der ifm-Webseite heruntergeladen werden. In der ESI-Datei sind alle Parameter- und Prozessdaten sowie deren gültige Wertebereiche definiert.

## 14.2.2 Zyklische Daten

### Inhalt

Process Data Objects (PDO).....	98
---------------------------------	----

22429

### Process Data Objects (PDO)

10929

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Name	Beschreibung	Datentyp / Zugriff
0x1C12	RxPDO Assign	Outputs: List of references to RxPDO Mapping 0x160n; one subindex for each module	UINT16
0x1C13	TxPDO Assign	Inputs: List of references to TxPDO Mapping 0x1A0n; one subindex for each module	UINT16
0x160n	RxPDO Mapping	Outputs: List of references to RxPDO Data in Output Area; one index for each module; multiple subindices	UINT32 / rw
0x1A0n	TxPDO Mapping	Inputs: List of references to TxPDO Data in Input Area; one index for each module; multiple subindices	UINT32 / r
0x1A08	TxPDO Mapping of New Msg. Avail.	Inputs: Reference to TxPDO Data of New Message Available in 0x10F3:04 (1 bit) and 31 alignment bits	UINT32 / r
0x1A09	TxPDO Mapping of Timestamp	Inputs: Reference to TxPDO Data of Timestamp in 0x10F8 (64 bit)	UINT64 / r
0x1A81	TxPDO Mapping of Device Status	Inputs: References to TxPDO Data of Device Status in 0xF100:nn (8 bit), for all ports	UINT32 / r
0x1A82	TxPDO Mapping of Port Qualifier	Inputs: References to TxPDO Data of Port Qualifier in 0xF101:0n (8 bit), for all ports	UINT32 / r
0x70n0	Output Area, RxPDO Data	Outputs: RxPDOs (the cyclic data itself) of all modules; one index for each module; multiple subindices for multiple cyclic data objects	Octet String / rw
0x60n0	Input Area, TxPDO Data	Inputs: TxPDOs (the cyclic data itself) of all modules; one index for each module; multiple subindices for multiple cyclic data objects	Octet String / r
0x140n	RxPDO Parameter	Outputs: RxPDO Control for setting outputs valid/invalid; one index for each module; only subindex 8; unused here	Record / rw
0x180n	TxPDO Parameter	Inputs: TxPDO State for telling if inputs are valid/invalid; one index for each module; only subindex 7; unused here	Record / r
0xF100	Device Status	This status byte is included in input data of each module; one subindex for each module	UINT8 / r
0xF101	Port Qualifier	This status byte is included in input data of each module; one subindex for each module	UINT8 / r

## 14.2.3 Azyklische Daten

### Inhalt

Hinweis .....	99
MDP Standard Information (0x1000) .....	99
Manufacturer Specific Index (0x2000) .....	100
IO-Link Acyclic Command (0x3100) .....	101
IO-Link Inputs (0x6000) .....	102
IO-Link Outputs (0x7000) .....	102
Port Configuration (0x8000) .....	103
Port Mode (0x9000) .....	104
Diagnostics Data (0xA000) .....	104
Device Status / Port Status (0xF000) .....	105

22427

### Hinweis

16560

Das Gerät implementiert ein "Modular Device Profile" mit einem "IO-Link Profile" nach ETG.5001.1.

### MDP Standard Information (0x1000)

9043

Identitätsinformationen über das Gerät und aktuelle und verfügbare Prozessdatenkonstellationen

Index	Sub-index	Beschreibung	Mögliche Werte / Referenz	Datentyp / Zugriff
0x1000		Device Type	MDP Profile = 0x184C1389	UINT32 / r
0x1008		Manufacturer Device Name	"IO-Link Master CL EC 8P IP20"	STRING / r
0x1009		Manufacturer Hardware Version	z.B. "AA"	STRING / r
0x100A		Manufacturer Software Version		STRING/ r
0x1018		Identity Object		
	0x1	Vendor ID	0x622	UINT32 / r
	0x2	Product Code	"AL1930"	UINT32 / r
	0x3	Revision Number:		UINT32 / r
	0x4	Serial Number		UINT32 / r
0x10F8		Timestamp (value in ns)		UINT64 / r

r ... nur lesen

rw ... lesen und schreiben

**Manufacturer Specific Index (0x2000)**

11011

## Herstellerspezifische Parameter

Index	Sub-index	Beschreibung	Mögliche Werte		Datentyp / Zugriff
0x2001		Component Name	"EtherCAT IO-Link Gateway"		STRING / r
0x2002		Vendor Name	"ifm electronic"		STRING / r
0x2003		Vendor URL	"www.ifm.com"		STRING / r
0x2004		Order Number	"AL1930"		STRING / r
0x2005		Manufacturing Date			STRING / r
0x2006		QS Date			STRING / r
0x2007		Installation Location	user-defined; max. 20 characters		STRING / rw
0x200A		Equipment ID	user-defined; max. 20 characters		STRING / rw
0x2F00		Reset To Factory	0xA500:	Factory Reset of System + NVMEM	UINT16 / w
			0xA501:	Factory Reset of System	
0x2F01		Device Localization	0x00:	LED RDY blinking for 5 s	UINT8 / w
0x2F02		Current Use Case (Access Rights)	0xA500:	EtherCAT + IoT	UINT16 / rw
			0xA501:	EtherCAT + IoT (read only)	
			0xA502:	IoT (only)	

r ... nur lesen  
 rw ... lesen und schreiben  
 w ... nur schreiben

## IO-Link Acyclic Command (0x3100)

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Description	Mögliche Werte		Datentyp / Zugriff
0x310n	0x01	Command Buffer	Byte 0:	Command (0x0 = Read, 0x1 = Write)	ARRAY_OF_BYTE / rw
			Byte 1:	IO-Link Index, LSB	
			Byte 2:	IO-Link Index, MSB	
			Byte 3:	IO-Link Subindex	
			Byte 4:	For Write only: Length of following data (0x01...0xE8)	
			Byte 5:	For Write only: Data (0x00...0xFF)	
	0x02	Status	0x00:	Command completed: no error, no response data	UINT8 / r
			0x01:	Command completed: no error, response data available	
			0x02:	Command completed: error, no response data	
			0x03:	Command completed: error, response data available	
			0xFF:	Command is executing (pending)	
	0x03	Response Buffer	Byte 0:	Status (see Subindex 0x02)	ARRAY_OF_BYTE / r
			Byte 1:	reserved	
			Byte 2:	If status = 0x1   0x3: Length of the following data	
			Byte 3..m:	If status = 0x1: Read data (0x00...0xE8) If status = 0x3: 1 byte IO-Link Error Code + 1 byte Additional Code	

r ... nur lesen

rw ... lesen und schreiben

## IO-Link Inputs (0x6000)

2260

Eingangsdaten der IO-Link-Ports X01...X08

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Beschreibung	Mögliche Werte	Datentyp / Zugriff
0x60n0		IO-Link Inputs		
	▪ 0x01	Byte 1	je Byte: 0x00...0xFF	pro Byte: UINT8 / r
	...	...		
	▪ 0x20	Byte 32		

r ... nur lesen

## IO-Link Outputs (0x7000)

23089

Ausgangsdaten der IO-Link-Ports X01...X08

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Beschreibung	Mögliche Werte	Datentyp / Zugriff
0x70n0		IO-Link Outputs		
	▪ 0x01	Byte 1	pro Byte: 0x00...0xFF	pro Byte: UINT8 / rw
	...	...		
	▪ 0x20	Byte 32		

rw ... lesen und schreiben

## Port Configuration (0x8000)

Herstellerspezifische Einstellungen der IO-Link-Ports X01...X08

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Beschreibung	Mögliche Werte		Datentyp / Zugriff
0x80n0		Port Configuration			
	0x04	Device ID	0x000000 ... 0xFFFFFFFF		UINT32 / rw
	0x05	Vendor ID	0x0000 ... 0xFFFF		UINT32 / rw
			0x136:	ifm electronic	
	0x20	IO-Link Revision	0x10:	IO-Link Revision 1.0	UINT8 / rw
			0x11:	IO-Link Revision 1.1	
	0x22	Cycle Time			UINT32 / rw
	0x24	Process Data In Length			UINT8 / rw
	0x25	Process Data Out Length			UINT8 / rw
	0x28	Master Control	0x00:	Deactivated	UINT16 / rw
			0x01:	Digital Input	
			0x02:	Digital Output	
			0x03:	IO-Link	
0x80n8		Vendor Specific Port Configuration			
	0x01	Validation ID	0x00:	No check	UINT8 / rw
			0x01:	V1.0 Device, no DS	
			0x02:	V1.1 Device, no DS	
			0x03:	V1.1 Device, Backup + Restore	
			0x04:	V1.1 Device, Restore	
	0x02	Reconfigure	0x00:	No action	UINT8 / rw
			0xFF:	Activate configuration	
	0x03	Byte Swap	0x00:	No action	UINT8 / rw
			0x01:	Byte swap	

rw ... lesen und schreiben

## Port Mode (0x9000)

17300

Aktuelle Werte der angeschlossenen IO-Link Devices

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Beschreibung	Mögliche Werte	Datentyp / Zugriff
0x90n0		Port Mode		
	0x04	Device ID		UINT32 / r
	0x05	Vendor ID		UINT32 / r
	0x20	IO-Link Revision		UINT8 / r
	0x21	Frame Capability		UINT8 / r
	0x22	Cycle Time		UINT8 / r
	0x24	PD In Length		UINT8 / r
	0x25	PD Out Length		UINT8 / r

r ... nur lesen

## Diagnostics Data (0xA000)

16537

Für jeden Port stellt das Gerät folgende Diagnosedaten bereit:

Wahl des IO-Link Ports über n (n = 0: Port X01, n = 1: Port X02,...)

Index	Sub-index	Beschreibung	Mögliche Werte		Datentyp / Zugriff
0xA0n0		Diagnostics Data			
	0x01	IO-Link State	0x00:	INACTIVE	UINT8 / r
			0x01:	DIGINPUT	
			0x02:	DIGOUTPUT	
			0x08:	OPERATE	
			0x09:	STOP	
	0x02	Subindex 0x02: Lost Frames			UINT8 / r

r ... nur lesen



**Device Status / Port Status (0xF000)**

17933

Status des IO-Link Device am Port X01...X08

Index	Sub-index	Beschreibung	Mögliche Werte	Datentyp / Zugriff
0xF000		Module Device Profile		
	▪ 0x01	▪ Module Index Distance	0x0010	UINT16 / r
	▪ 0x02	▪ Maximum number of modules	0x0008	UINT16 / r
0xF030		Configured Module Ident List		
0xF050		Detected Module Ident List		
0xF100		Device Status		UINT8 / r
	▪ 0x01	▪ IO-Link Port X01	→ <b>Mapping: Device Status (0xF100)</b> (→ S. <a href="#">106</a> )	
	...	...		
	▪ 0x0n	▪ IO-Link Port X08		
0xF101		IO-Link Port Qualifier		UINT8 / r
	▪ 0x01	▪ PQI Port X01	→ <b>Mapping: Port Qualifier (0xF101)</b> (→ S. <a href="#">107</a> )	
	...	...		
	▪ 0x0n	▪ PQI Port X08		

r ... nur lesen

**Mapping: Device Status (0xF100)**

8344

Bit							
7	6	5	4	3	2	1	0
Error Code				Port State			

Legende:

- [Error Code] Fehlercode des IO-Link Device
    - 0x0 No error
    - 0x3 Invalid Device ID
    - 0x4 Invalid Vendor ID
    - 0x7 Invalid cycle time
    - 0x8 Invalid length of PD In
    - 0x9 Invalid length of PD Out
    - 0xA No device detected
    - 0xB Supply voltage low or short circuit
    - 0xD Unspecified error
  - [Port State] Zustand des IO-Link Ports
    - 0x0 Deactivated
    - 0x1 Digital Input
    - 0x2 Digital Output
    - 0x3 OP: IO-Link, Operate state
    - 0x4 STOP: IO-Link, not Operate state (fault or no device)
    - 0x5 PreOP: IO-Link, device in PreOP state
- Hinweis: Treten mehrere Fehler gleichzeitig auf, wird nur ein Fehler angezeigt. Die anderen Fehlermeldungen werden unterdrückt.

## Mapping: Port Qualifier (0xF101)

14998

Port Qualifier Information (PQI) enthalten Diagnoseinformationen des IO-Link-Ports. Die PQI wird vom IO-Link Master zusätzlich zu den Prozessdaten an den EtherCAT-Controller gesendet.

Bit							
7	6	5	4	3	2	1	0
PVI	DE	DA	DACT	PVO	--	DI2	DI4

Legende:

- [DI4] Signalstatus des digitalen Eingangs an Klemme 4 (wenn genutzt)
 

0x0	OFF
0x1	ON
- [DI2] Signalstatus des digitalen Eingangs an Klemme 2 (wenn genutzt)
 

0x0	OFF
0x1	ON
- [PVO] Port Validity Output: Gültigkeit der Ausgangsdaten des IO-Link Devices
 

0x0	ungültig
0x1	gültig
- [DACT] Device deactivated: zeigt, ob der IO-Link Port konfiguriert und einsetzbar ist
 

0x0	aktiviert und einsetzbar
0x1	deaktiviert oder nicht verfügbar
- [DA] Device Available: zeigt, ob IO-Link Device erkannt wurde und ob sich das Device in Zustand "Preoperate" oder "Operate" befindet
 

0x0	kein Gerät
0x1	Gerät erkannt
- [DE] Device Error: zeigt, ob ein Fehler oder eine Warnung aufgetreten ist; Hinweis: Anwender muss die Fehlerursache über die azyklischen Dienste separat ermitteln.
 

0x0	kein Fehler
0x1	Fehler
- [PVI] Port Validity Input: Gültigkeit der Eingangsdaten des IO-Link Devices
 

0x0	ungültig
0x1	gültig

## 14.2.4 Events

7110

### Diagnosis History (0x10F3)

11232

Index	Sub-index	Beschreibung	Mögliche Werte		Datentyp / Zugriff			
0x10F3		Diagnosis History						
	0x01	Maximum Messages	0x00...0x0x40		UINT8 / r			
	0x02	Newest Messages	0x00...0x0x40		UINT8 / r			
	0x03	Newest Ack. Message	Override Mode (Subindex 5, Bit 4 = 0):		UINT8 / rw			
			0	Reading: When the message queue will be overwritten, the slave shall set Subindex 0x03 to 0  Writing: the slave will clear all messages, i.e. resetting Subindex 0x02, 0x03, 0x04 and 0x5/Bit 5				
			1...5	Writing: the slave shall return SDO-Abort with codes 0x06090030 (value range of parameter exceeded) or 0x0609003				
			6...0x46	Writing: Subindex 0x03 = Written value without checking				
			0x46...255	Writing: SDO-Abort with codes 0x06090030 or 0x06090031(value of parameter written too high)				
			Acknowledge Mode (Subindex 5, Bit 4 = 1):					
			0	Read: No messages have been acknowledge so far  Writing: All acknowledged messages will be deleted.				
			1...5	Writing: The slave shall return SDO-Abort with codes 0x06090030 (value range of parameter exceeded) or 0x06090032				
			6...0x46	Read: SubIndex of latest acknowledged diagnosis message  Writing: Messages are acknowledged				
			0x46...255	Writing: SDO-Abort with codes 0x06090030 or 0x06090031(value of parameter written too high)				
			0x04	New Message Available		Override Mode:		BOOL / r
						0x0	newest msg. was read	
						0x1	newest msg. was not read	
	Acknowledge Mode:							
	0x0	no acknowledged message						
	0x1	diagnosis message are available which can be acknowledged (Subindex 0x02<->Subindex 0x03)						
	0x05	Flags	Bit 0:	Enable Emergency sending (according to ETG.1000-6)	UINT16 / rw			
				0		default if device does not support Emergency sending		

Index	Sub-index	Beschreibung	Mögliche Werte			Datentyp / Zugriff	
				1	new diagnosis messages shall be sent as Emergency message		
			Bit 1:	Disable info messages			
				0	Info messages are stored in the diagnosis message queue (default)		
				1	Info messages will not be stored in the diagnosis message queue		
			Bit 2:	Disable warning messages			
				0	Warning messages are stored in the diagnosis message queue (default)		
				1	Warning messages will not be stored in the diagnosis message queue		
			Bit 3:	Disable error messages			
				0	Error messages are stored in the message queue (default)		
				1	Error messages will not be stored in the diagnosis message queue		
			Bit 4:	Mode selection for diagnosis history handling			
				0	Overwrite Mode		
				1	Acknowledge Mode		
			Bit 5:	Overwrite/Discard Information			
				Overwrite Mode:			
				1	unacknowledged messages have been overwritten (=buffer overrun) (Subindex 0x03 is set to 0)		
				Acknowledge Mode:			
				1	message buffer is full with unacknowledged messages and a new message is discarded.		
			Bit 6...15	reserved			
	0x06	Diagnosis Message 01	→ Mapping: Diagnosis Message (→ S. <a href="#">110</a> )			OCTET STRING / r	
...							
0x0x46	Diagnosis Message 64						

r ... nur lesen  
r/w ... lesen und schreiben

## Mapping: Diagnosis Message

16940

Parameter	Inhalt	Mögliche Werte			Datentyp / Zugriff
Diag Code	Diagnostic code	Bit 0...15:	0x0000 - 0xDFFF	not used	UINT32 / r
			0xE000 - 0xE7FF	Bit 16...31: Manufacturer specific	
			0xE800	Bit 16...31: Emergency Error Code from DS301 or DS4xxx	
			0xE801 - 0xEDFF	reserved	
			0xEE00 - 0xEFFF	Bit 16...31: Profile specific	
			0xF000 - 0xFFFF	not used	
			0xFF01	New IO-Link Event <ul style="list-style-type: none"> <li>Byte 1: IO-Link Port (0x00 = Port 01, ...)</li> <li>Byte 2: Event Qualifier</li> <li>Byte 3+4: Event Code</li> <li>Byte 5: unused</li> </ul>	
			0xFF02	Port Configuration has failed <ul style="list-style-type: none"> <li>Byte 1...5: unused</li> </ul>	
			0xFF03	IO-Link Device Lost (disconnected) <ul style="list-style-type: none"> <li>Byte 1: IO-Link Port (0x00 = Port 01, ...)</li> </ul>	
			0xFF04	IO-Link Device Fault (validation error or PD length mismatch) <ul style="list-style-type: none"> <li>Byte 1: IO-Link Port (0x00 = Port 01, ...)</li> </ul>	
			0xFF05	IO-Link Device Operating <ul style="list-style-type: none"> <li>Byte 1: IO-Link Port (0x00 = Port 01, ...)</li> <li>Byte 2: Old status code</li> <li>Byte 3: Old info code</li> <li>Byte 4: New status code</li> <li>Byte 5: New info code</li> </ul>	
Flags	Event type	Bit 0...3:	0x00	Info message	UINT16 / r
			0x01	Warning message	
			0x02	Error message	
Text ID	Text ID as reference to Diagnosis text as defined in ESI file	0x0000		no Text ID	UINT16 / r
		else		Text ID reference to ESI file	
Time Stamp	Time Stamp ( value in ns)				UINT64 / r
Flags Parameter 1	Data type Parameter 1	0x0005		UINT8	UINT16 / r
Parameter 1	IO-Link Port	0x01		Port X01	UINT8 / r
		0x02		Port X02	

Parameter	Inhalt	Mögliche Werte			Datentyp / Zugriff
		...		...	
		0x08		Port X08	
Flags Parameter 2	Data type Parameter 2	0x0006		UINT16	UINT16 / r
Parameter 2	Event Code	geräteabhängig (→ IODD-Beschreibung des IO-Link Devices)			UINT16 / r
Flags Parameter 3	Data type Parameter 3	0x0005		UINT8	UINT16 / r
Parameter 3	Event Qualifier	Bit 0..2:	0x1	PHL	UINT8 / r
			0x2	DL	
			0x3	AL	
			0x4	APPL	
		Bit 3:	0x0	Source: Device	
			0x1	Source: Master	
		Bit 4...5:	0x1	Event Type: Info Message	
			0x2	Event Type: Warning Message	
			0x3	Event Error Message	
		Bit 6...7:	0x1	Message Type: Single	
			0x2	Message Type: Going	
			0x3	Message TYPe: Coming	

r ... nur lesen

rw ... lesen und schreiben

## 14.3 ifm-IoT-Core

### Inhalt

Übersicht: IoT-Profil.....	113
Übersicht: IoT-Typen.....	120
Übersicht: IoT-Dienste.....	121

8988



## 14.3.1 Übersicht: IoT-Profile

### Inhalt

Profil: blob .....	113
Profil: deviceinfo .....	114
Profil: devicetag .....	114
Profil: iolinkdevice_full .....	115
Profil: iolinkmaster .....	115
Profil: mqttCmdChannel .....	116
Profil: mqttCmdChannelSetup .....	116
Profil: mqttConnection .....	116
Profil: mqttSetup .....	117
Profil: network .....	117
Profil: parameter .....	118
Profil: processdata .....	118
Profil: runcontrol .....	118
Profil: service .....	118
Profil: software .....	118
Profil: software/uploadedablessoftware .....	119
Profil: timer .....	119

17711

### Profil: blob

9584

Element (identifier)	Eigenschaften	Pflicht	Kommentar
blobname	<ul style="list-style-type: none"> <li>type = data</li> <li>profiles = blob</li> </ul>		kennzeichnet Element als Geräteinformation
../size	type = data	obligatorisch	
../chunksize	type = data	obligatorisch	
../setblobdata	type = service	optional	
../getblobdata	type = service	optional	
../start_stream_set	type = service	optional	
../stream_set	type = service	optional	
../clear	type = service	optional	
../getcrc	type = service	optional	
../getmd5	type = service	optional	
../getdata	type = service	optional	
../setdata	type = service	optional	

## Profil: deviceinfo

17135

Element (identifier)	Eigenschaften	Pflicht	Kommentar
deviceinfo	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = deviceinfo</li> </ul>		kennzeichnet Element als Geräteinformation
../devicename	type = data	optional	
../devicefamily	type = data	optional	
../devicevariant	type = data	optional	
../devicesymbol	type = data	optional	
../deviceicon	type = data	optional	
../serialnumber	type = data	obligatorisch	
../productid	type = data	optional	
../productname	type = data	optional	
../productcode	type = data	obligatorisch	
../producttext	type = data	optional	
../ordernumber	type = data	optional	
../productiondate	type = data	optional	
../productioncode	type = data	optional	
../hwrevision	type = data	obligatorisch	
../swrevision	type = data	obligatorisch	
../bootloaderrevision	type = data	optional	
../vendor	type = data	optional	
../vendortext	type = data	optional	
../vendorurl	type = data	optional	
../vendorlogo	type = data	optional	
../productwebsite	type = data	optional	
../supportcontact	type = data	optional	
../icon	type = data	optional	
../image	type = data	optional	
../standards	type = data	optional	

## Profil: devicetag

17438

Element (identifier)	Eigenschaften	Pflicht	Kommentar
devicetag	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = devicetag</li> </ul>		
../applicationtag	type = data	obligatorisch	
../applicationgroup	type = data	optional	
../machinecode	type = data	optional	
../tenant	type = data	optional	

**Profil: iolinkdevice\_full**

7376

Element (identifier)	Eigenschaften	Pflicht	Kommentar
iolinkdevice	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = iolinkdevice_full</li> </ul>		Struktur eines IO-Link Devices
../vendorid	type = data	obligatorisch	
../deviceid	type = data	obligatorisch	
../productname	type = data	obligatorisch	
../serial	type = data	obligatorisch	
../applicationspecifictag	type = data	obligatorisch	
../pdin	type = data	obligatorisch	
../pdout	type = data	obligatorisch	
../status	type = data	obligatorisch	
../iolreadacyclic	type = data	obligatorisch	
../iolwriteacyclic	type = data	obligatorisch	
../iolinkevent	type = data	obligatorisch	

**Profil: iolinkmaster**

14997

Element (identifier)	Eigenschaften	Pflicht	Kommentar
masterport	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = iolinkmaster</li> </ul>		ausführbarer Dienst
../mode	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../comspeed	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../mastercycletime_actual	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../mastercycletime_preset	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../validation_datastorage_mode	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../validation_vendorid	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../validation_deviceid	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../additionalpins_in	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = processdata</li> </ul>	optional	
../additionalpins_out	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = processdata</li> </ul>	optional	
../portevent	<ul style="list-style-type: none"> <li>type = data</li> </ul>	obligatorisch	
../iolinkdevice	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = iolinkdevice_full</li> </ul>	obligatorisch	

**Profil: mqttCmdChannel**

60217

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
mqttCmdChannel	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = commChannel</li> </ul>		Profil des MQTT-Kommandokanals
../type	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	obligatorisch	Protokolltyp der Schnittstelle
../status	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	obligatorisch	Zustand des MQTT-Kommandokanals (Mögliche Werte: init, running, stopped, error)
../mqttCmdChannelSetup	type = profile		Unterprofil: <b>Profil: mqttCmdChannelSetup</b> (→ S. <a href="#">116</a> )

**Profil: mqttCmdChannelSetup**

60220

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
mqttCmdChannelSetup	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = mqttCmdChannelSetup</li> </ul>		Einstellungen des MQTT-Kommandokanals
../brokerIP	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	optional	
../brokerPort	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	optional	
../cmdTopic	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	optional	
../defaultReplyTopic	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	optional	

**Profil: mqttConnection**

60216

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
mqttConnection	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = commInterface</li> </ul>		MQTT-Verbindung im IoT Core
../type	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	obligatorisch	Protokolltyp der Schnittstelle
../status	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	obligatorisch	globaler Zustand des MQTT (Mögliche Werte: init, running, stopped, error)
../mqttSetup	type = profile		Unterprofil: <b>Profil: mqttSetup</b> (→ S. <a href="#">117</a> )
../mqttCmdChannel	type = profile		Unterprofil: <b>Profil: mqttCmdChannel</b> (→ S. <a href="#">116</a> )

**Profil: mqttSetup**

60218

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
mqttSetup	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = mqttSetup</li> </ul>		Eigenschaften des mqtt-Kommandokanals
../QoS	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = Number</li> </ul>	obligatorisch	Quality of Service der MQTT-Verbindung
../version	<ul style="list-style-type: none"> <li>type = data</li> <li>data type = STRING</li> </ul>	obligatorisch	

**Profil: network**

11179

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
network	<ul style="list-style-type: none"> <li>type = structure</li> <li>profiles = deviceinfo</li> </ul>		kennzeichnet Element als Geräteinformation
../macaddress	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../ipaddress	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../ipv6address	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../subnetmask	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../ipdefaultgateway	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../dhcp	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../ipversion	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../hostname	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../autonegotiation	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../portspeed	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../enablenetwork	type = service	optional	
../disablenetwork	type = service	optional	

## Profil: parameter

16545

Das Profil wird genutzt, um Elemente vom Type data als Parameter zu kennzeichnen (azyklische Daten). Das Profil definiert keine Unterstruktur.

## Profil: processdata

16569

Das Profil wird genutzt, um Elemente vom Type data als Prozessdaten zu kennzeichnen (zyklische Daten). Das Profil definiert keine Unterstruktur.

## Profil: runcontrol

60219

Element (identifier)	Eigenschaften	Pflicht	Kommentar
runcontrol	<ul style="list-style-type: none"> <li>type = profile</li> <li>profile = runcontrol</li> </ul>		Steuerung des MQTT-Kommandokanals
../start	type = service	obligatorisch	<b>Dienst: start</b> (→ S. <a href="#">131</a> )
../stop	type = service	obligatorisch	<b>Dienst: stop</b> (→ S. <a href="#">132</a> )
../reset	type = service	obligatorisch	<b>Dienst: reset</b> (→ S. <a href="#">129</a> )

## Profil: service

16575

Element (identifier)	Eigenschaften	Pflicht	Kommentar
service	<ul style="list-style-type: none"> <li>type = service</li> <li>profile = service</li> </ul>		ausführbarer Dienst

## Profil: software

10999

Element (identifier)	Eigenschaften	Pflicht	Kommentar
software	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = software</li> </ul>		kennzeichnet Element als Software
../version	type = data	obligatorisch	
../type	type = data	obligatorisch	
../status	type = structure	optional	
../diag	type = structure	optional	

**Profil: software/uploadedablessoftware**

12559

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
software	<ul style="list-style-type: none"> <li>type = structure</li> <li>profiles = software/uploadablessoftware</li> </ul>		Software, die über den IoT Core auf das Gerät geladen werden kann
../lastinstall	type = data	optional	
../installhistory	type = data	optional	
../container	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = blob</li> </ul>	obligatorisch	
../preinstall	type = service	optional	
../install	type = service	obligatorisch	
../postinstall	type = service	optional	
../abortinstall	type = service	optional	
../installstatus	type = data	optional	

**Profil: timer**

10997

Element (identifizier)	Eigenschaften	Pflicht	Kommentar
timer	<ul style="list-style-type: none"> <li>type = structure</li> <li>profile = timer</li> </ul>		
../counter	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	obligatorisch	
../interval	<ul style="list-style-type: none"> <li>type = data</li> <li>profile = parameter</li> </ul>	optional	
../start	type = service	optional	
../stop	type = service	optional	

## 14.3.2 Übersicht: IoT-Typen

16547

Der ifm-IoT-Core nutzt folgende Elementtypen:

Name	Beschreibung
structure	Element ist ein Strukturelement (wie ein Ordner im Dateisystem)
service	Element ist ein Dienst, der aus dem Netzwerk heraus angesprochen werden kann
event	Element ist ein Ereignis, das durch die Firmware ausgelöst werden kann und Benachrichtigungen verschickt
data	Element ist ein Datenpunkt
device	Wurzelement, das ein Gerät repräsentiert



### 14.3.3 Übersicht: IoT-Dienste

#### Inhalt

Dienst: factoryreset.....	121
Dienst: getblobdata.....	122
Dienst: getdata .....	122
Dienst: getdatamulti .....	123
Dienst: getelementinfo.....	123
Dienst: getidentity .....	124
Dienst: getsubscriberlist .....	125
Dienst: getsubscriptioninfo .....	126
Dienst: gettree .....	127
Dienst: install .....	128
Dienst: iolreadacyclic.....	128
Dienst: iolwriteacyclic .....	128
Dienst: querytree .....	129
Dienst: reboot .....	129
Dienst: reset .....	129
Dienst: setblock .....	130
Dienst: setdata.....	131
Dienst: signal .....	131
Dienst: start .....	131
Dienst: start_stream_set.....	132
Dienst: stop.....	132
Dienst: stream_set.....	132
Dienst: subscribe .....	133
Dienst: unsubscribe .....	134
Dienst: validation_useconnecteddevice .....	134

17708

#### Dienst: factoryreset

12188

**Name:** factoryreset

**Beschreibung:** Der Dienst setzt die Parameter des Geräts auf die Werkseinstellungen.

**Anfragedaten (Feld "data"):** keine

**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/firmware/factoryreset"
}
```

**Dienst: getblobdata**

41972

**Name:** getblobdata**Beschreibung:** Der Dienst liest ein Binary Large Object (blob).**Anwendbar auf:** datastorage**Anfragedaten (data):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
pos	obligatorisch	number	0	Byte-Position
length	obligatorisch	number	-	Größe des Objekts (Anzahl der Bytes)

**Rückgabedaten (data):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
data	obligatorisch	STRING	0	zu dekodierende Daten (BASE64-codiert)
crc	optional	HEX STRING		CRC der Daten nach der Dekodierung
md5	optional	HEX STRING		MD5-Prüfsumme der Daten nach der Dekodierung

**Dienst: getdata**

12223

**Name:** getdata**Beschreibung:** Dienst liest den Wert eines Datenpunkts und gibt diesen aus.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
value	obligatorisch	STRING	Wert des Elements/Datenpunkts

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "devicetag/applicationtag/getdata"
}
```

**Dienst: getdatamulti**

17964

**Name:** getdatamulti**Beschreibung:** Der Dienst liest sequentiell die Werte mehrerer Datenpunkte und gibt diese aus. Für jeden Datenpunkt werden der Wert und ein Diagnosecode ausgegeben.**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
datatosend	obligatorisch	ARRAY OF STRINGS	Liste von Datenpunkten, die abgefragt werden sollen; Datenpunkte müssen den Dienst getdata unterstützen ("datatosend":["url1","url2",...,"urlx"])

**Rückgabedaten (Feld "data"):** für jeden abgefragten Datenpunkt

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
url	obligatorisch	STRING	Datenpunkt, der abgefragt wurde
code	obligatorisch	INT	Diagnosecode der Abfrage
data	obligatorisch	STRING	Wert des Datenpunkts

**Dienst: getelementinfo**

13342

**Name:** getelementinfo**Beschreibung:** Der Dienst liest die Eigenschaften eines Elements des IoT-Baums.**Anwendbar auf:** Objekte vom Typ device**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
adr	obligatorisch	STRING		URL des Elements, dessen Eigenschaften gelesen werden sollen

**Rückgabedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
identifizier	obligatorisch	STRING		Bezeichner des Elements
type	obligatorisch	STRING		Typ des Elements
format	optional	JSON-Objekt	leer	Format der Daten oder des Serviceinhalts
uid	optional	STRING	leer	
profiles	optional	JSON-Array	leer	
hash	optional	STRING	--	

**Dienst: getidentity**

52381

**Name:** getidentity**Beschreibung:** Der Dienst liest die Geräteinformationen des AL1930 und gibt sie aus.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
iot		device	Gerätebeschreibung als JSON-Objekt
iot.name	obligatorisch	STRING	
iot.uid	optional	STRING	
iot.version	obligatorisch	STRING	
iot.catalogue	optional	ARRAY OF OBJECTS	
iot.deviceclass	optional	ARRAY OF STRING	
iot.serverlist	optional	ARRAY OF OBJECTS	
device	optional		AL1930
device.serialnumber	optional		Seriennummer
device.hwrevision	optional		Hardwarestand
device.swrevision	optional		Softwarestand
device.custom	optional		
security	optional		Sicherheitsoptionen
security.securitymode	optional	ENUM	zeigt, ob Sicherheitsmodus aktiviert ist
security.authscheme	optional	ENUM	zeigt aktives Authentifizierungsschema
security.ispasswordset	optional	BOOL	zeigt, ob ein Passwort gesetzt wurde
security.activeconnection	optional	ENUM	zeigt aktuell genutzte Kommunikationsschnittstelle
			▪ tcp_if    unverschlüsselte http-Verbindung an IoT-Schnittstelle, Port 80
			▪ tls_if    verschlüsselte https-Verbindung an IoT-Schnittstelle, Port 443
			▪ fb_if    unverschlüsselte http-Verbindung an Feldbus-Schnittstelle, Port 80

**Dienst: getsubscriberlist**

60239

**Name:** getsubscriberlist**Beschreibung:** Der Dienst liefert eine Liste mit allen aktiven Abonnements.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):** Array mit jeweils folgenden Daten

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
adr	obligatorisch	STRING	Datenquelle
datatosend	obligatorisch	ARRAY OF STRINGS	Liste mit URLs der abonnierten Datenpunkten
cid	obligatorisch	NUMBER	ID des Abonnements
callbackurl	obligatorisch	STRING	Adresse, an die IoT-Core Ereignisse benachrichtigen soll
duration	obligatorisch	STRING	Speicherdauer des Werts

**Beispiel:**

- Anfrageobjekt:**

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/getsubscriberlist"
}
```

- Rückgabeobjekt:**

```
{
  "cid": 4711,
  "data": [
    {
      "adr": "/timer[1]/counter/datachanged/subscribe",
      "datatosend": ["/iolinkmaster/port[2]/iolinkdevice/pdin"],
      "cid": 1,
      "callbackurl": "http://192.168.0.45:80/temp",
      "duration": "lifetime"
    },
    {
      "adr": "/timer[1]/counter/datachanged/subscribe",
      "datatosend": ["/processdatamaster/temperature", "/processdatamaster/voltage"],
      "cid": 2,
      "callbackurl": "http://192.168.0.44:80/temp",
      "duration": "lifetime"
    }
  ]
  "code": 200
}
```

## Dienst: getsubscriptioninfo

60244

**Name:** getsubscriptioninfo

**Beschreibung:** Der Dienst liefert Informationen über ein existierendes Abonnement (subscribe).



Für die Abfrage müssen folgende Parameter des existierenden Abonnements genutzt werden:

- Wert des Identifiers cid (z. B. 4711)
- Nummer des Timers (z. B. timer[1])
- Name des callback-Topics (z. B. temp)

### Anfragedaten (Feld "data"):

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
callback	obligatorisch	STRING	Adresse, an die IoT-Core Ereignisbenachrichtigungen senden soll; komplette URL: http://ipaddress:port/path

### Rückgabedaten (Feld "data"):

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
subscription	obligatorisch	BOOL	Status des übergebenen Abonnement-Parameter
datatosend	obligatorisch	ARRAY OF STRINGS	Liste mit abonnierten Datenpunkten
cid	obligatorisch	NUMBER	ID der subscribe-Anfrage
callbackurl	obligatorisch	STRING	Adresse, an die IoT-Core Ereignisbenachrichtigungen senden soll; komplette URL: http://ipaddress:port/path

Beispiel:

#### • Anfrageobjekt:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/timer[1]/counter/datachanged/getsubscriptioninfo",
  "data": {
    "callback": "http://192.168.0.44:80/temp"
  }
}
```

#### • Rückgabeobjekt:

```
{
  "cid": 4711,
  "data": {
    "subscription": true,
    "datatosend": [
      "/iolinkmaster/port[2]/iolinkdevice/productname",
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/temperature"
    ],
    "callbackurl": "http://192.168.0.44:80/temp",
    "duration": "lifetime"
  },
  "code": 200
}
```

**Dienst: gettree**

60201

**Name:** gettree

**Beschreibung:** Der Dienst liest die Gerätebeschreibung des IO-Link Masters und gibt sie als JSON-Objekt aus. Die Ausgabe kann auf einen Teilbaum der Gerätebeschreibung begrenzt werden.

**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
adr	optional	STRING	Wurzelement des Teilbaums
level	optional	STRING	max. Ebene, bis zu der der Teilbaum ausgegeben wird <ul style="list-style-type: none"> <li>keine Angabe: alle Ebenen werden angezeigt</li> <li>0: keine Unterelemente anzeigen ("subs")</li> <li>1: Unterelemente anzeigen</li> <li>2: Unterelemente bis zur 2. Ebene anzeigen</li> <li>3: Unterelemente bis zur 3. Ebene anzeigen</li> <li>...</li> <li>20: Unterelemente bis zur 20. Ebene anzeigen</li> </ul>

**Rückgabedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
identifizier	obligatorisch	STRING	Bezeichner des Wurzelements
type	obligatorisch	STRING	Typ des Elements
format	optional	JSON-Objekt	Format des Dateninhalts
uid	optional	STRING	
profiles	optional	JSON-Array	
subs	obligatorisch	JSON-Array	Unterelemente
hash	optional	STRING	

Beispiele:

- die komplette Gerätebeschreibung ausgeben

```
{
  "code": "request",
  "cid": 4,
  "adr": "/gettree"
}
```

- den Teilbaum counter[2] der Gerätebeschreibung bis zur 2. Ebene ausgeben

```
{
  "code": "request",
  "cid": 4,
  "adr": "/gettree"
  "data": {
    "adr": "counter[2]",
    "level": 2
  }
}
```

**Dienst: install**

35379

**Name:** install**Beschreibung:** Der Dienst installiert die in einem Speicherbereich des Geräts gespeicherte Firmware.**Anwendbar auf:** container**Anfragedaten (data):** keine**Rückgabedaten (data):** keine**Dienst: iolreadacyclic**

12222

**Name:** iolreadacyclic**Beschreibung:** Der Dienst liest azyklisch den Parameterwert eines IO-Link Devices. Der Zugriff erfolgt über IO-Link Index und Subindex.**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
index	obligatorisch	NUMBER	IO-Link Index des Parameters
subindex	obligatorisch	NUMBER	IO-Link Subindex des Parameters

**Rückgabedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
value	obligatorisch	STRING	Parameterwert; Wert im Hexadezimalformat

**Dienst: iolwriteacyclic**

11035

**Name:** iolwriteacyclic**Beschreibung:** Der Dienst schreibt azyklisch den Parameterwert eines IO-Link Devices. Der Zugriff erfolgt über IO-Link Index und Subindex.**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
index	obligatorisch	NUMBER	IO-Link Index des Parameters
subindex	obligatorisch	NUMBER	IO-Link Subindex des Parameters
value	obligatorisch	STRING	Neuer Wert des Parameters; Wert im Hexadezimalformat

**Rückgabedaten (Feld "data"):** keine



## Dienst: querytree

60205

**Name:** querytree**Beschreibung:** Der Dienst durchsucht einen Gerätebaum nach den Kriterien profile, type und name und gibt eine Liste aus mit den URLs der gefundenen Elemente. Mindestens eines der Suchkriterien muss angegeben werden. Der Dienst ist nur auf dem Wurzelknoten des Geräts ausführbar.**Anfrage (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
profile	optional	STRING	Profil des gesuchten Elements
type	optional	STRING	Typ des gesuchten Elements
name	optional	STRING	Name des gesuchten Elements

**Rückgabe (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
urlList	obligatorisch	ARRAY	Array mit URLs der gefundenen Elements; URLs sind durch Kommas getrennt

## Dienst: reboot

10986

**Name:** reboot**Beschreibung:** Der Dienst startet das Gerät neu.**Anfragedaten (data):** keine**Rückgabedaten (data):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4,
  "adr": "firmware/reboot"
}
```

## Dienst: reset

60234

**Name:** reset**Beschreibung:** Der Dienst setzt eine Verbindung zurück in den Initialisierungszustand.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/reset"
}
```

## Dienst: setblock

12224

**Name:** setblock**Beschreibung:** Der Dienst setzt die Werte mehrerer Datenpunkte einer Struktur gleichzeitig.**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
datatoset	obligatorisch	ARRAY OF OBJECTS	Liste von Datenpunkten und deren neuen Werten; Datenpunkte müssen den Dienst setdata unterstützen
consistent	optional	BOOL	

**Rückgabedaten (Feld "data"):** keine

Beispiel:

Request:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "iotsetup/network/setblock",
  "data": {
    "datatoset": {
      "ipaddress": "192.168.0.6",
      "subnetmask": "255.255.255.0",
      "ipdefaultgateway": "192.168.0.250",
      "dhcp": 0
    }
  }
}
```

Response:

```
{
  "cid": 4711,
  "code": 233
}
```

**Dienst: setdata**

7159

**Name:** setdata**Beschreibung:** Der Dienst setzt den Wert eines Datenpunkts.**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
newvalue	obligatorisch	STRING	Neuer Wert des Elements/Datenpunkts
duration	obligatorisch	STRING	Dauer der Speicherung des Werts <ul style="list-style-type: none"> <li>lifetime: Wert wird mit IoT Core gespeichert; Wert bleibt gültig auch nach Neuart des Geräts</li> <li>uptime: Wert wird bis zum nächsten Neustart des Geräts gespeichert</li> </ul>

**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "devicetag/applicationtag/setdata",
  "data": {
    "newvalue": "ifm IO-Link master",
    "duration": "lifetime"
  }
}
```

**Dienst: signal**

25406

**Name:** signal**Beschreibung:** Der Dienst löst das Blinken der Status-LEDs des AL1930 aus.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "firmware/signal"
}
```

**Dienst: start**

60232

**Name:** start**Beschreibung:** Der Dienst startet eine Verbindung.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/start"
}
```

**Dienst: start\_stream\_set**

36563

**Name:** start\_stream\_set**Beschreibung:** Der Dienst startet die sequenzielle Übertragung mehrerer Datenfragmente.**Anwendbar auf:** Objekte vom Typ data**Anfragedaten (data):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
size	obligatorisch	STRING		Gesamtgröße der zu übertragenden Daten (Anzahl der Bytes)

**Rückgabedaten (data):** keine**Dienst: stop**

60233

**Name:** stop**Beschreibung:** Der Dienst stoppt eine Verbindung.**Anfragedaten (Feld "data"):** keine**Rückgabedaten (Feld "data"):** keine

Beispiel:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/connections/mqttConnection/MQTTSetup/mqttCmdChannel/status/stop"
}
```

**Dienst: stream\_set**

39175

**Name:** stream\_set**Beschreibung:** Der Dienst überträgt ein Datensegment.**Anwendbar auf:** Objekte vom Typ data**Anfragedaten (data):**

Datenfeld	Pflichtfeld	Datentyp	Default	Beschreibung
value	obligatorisch	BIN (BASE64)	*	Segment der Binardaten (BASE64-codiert)

**Rückgabedaten (data):** keine

**Dienst: subscribe**

60208

**Name:** subscribe

**Beschreibung:** Der Dienst abonniert die Werte von Datenpunkten. Die zu abonnierenden Datenpunkte werden als Liste übergeben. Der IoT Core sendet Änderungen an die in callback definierte Datensenke.



CSV-formatierte Benachrichtigungen können nur mit dem TCP-Protokoll über einen aktivierten und konfigurierten MQTT-Kanals übertragen werden.

**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
callback	obligatorisch	STRING	Adresse, an die IoT-Core die Benachrichtigungen senden soll; URL-Format: <ul style="list-style-type: none"> <li>JSON: http://ipaddress:port/path</li> <li>JSON: ws:///path</li> <li>JSON: mqtt://ipadress:port/topic</li> <li>CSV: tcp://ipaddress:port/path</li> </ul>
datatosend	obligatorisch	ARRAY OF STRINGS	Liste aus URLs von Datenelementen; Elemente müssen getdata unterstützen
codec	optional	STRING	Format der zurückgegebenen Daten <ul style="list-style-type: none"> <li>json: JSON-formatiert</li> <li>csv: CSV mit Standard-Separator (,)</li> <li>csv0: CSV-formatiert mit Komma-Separator (,)</li> <li>csv1: CSV-formatiert mit Semikolon-Separator (;)</li> </ul>
duration	obligatorisch	STRING	Dauer der Speicherung des Werts <ul style="list-style-type: none"> <li>lifetime: Wert wird mit IoT Core gespeichert; Wert bleibt gültig auch nach Neustart des Geräts</li> <li>uptime: Wert wird bis zum nächsten Neustart des Geräts gespeichert</li> <li>once: nur eine Benachrichtigung schicken, Benutzer muss Abonnement direkt wieder abmelden</li> </ul>

**Rückgabedaten (Feld "data"):** keine**Benachrichtigung: JSON**

```
{
  "code": "event",
  "cid": 4711,
  "adr": "",
  "data": {
    "eventno": "EventNo",
    "srcurl": "SrcURL",
    "payload": {
      "eventurl": { "code": "EventStatus", "data": "EventData",
        "datapointurl_1": { "code": "DataStatus_1", "data": "DataValue_1",
          "datapointurl_2": { "code": "DataStatus_2", "data": "DataValue_2",
            ...}}}}
    }
  }
}
```

**Benachrichtigung: CSV**

SrcURL, EventNo, EventStatus, EventData, DataStatus\_1, DataValue\_1, DataStatus\_2, DataValue\_2, ...

- SrcURL: Quelle des Ereignisses (Datenpunkt, auf den subscribe-Kommando aufgeführt wurde)

- EventNo: Ereignisnummer
- EventStatus: Statuscode des Ereignisses
- EventData: Eventdaten
- DataStatus\_1: Statuscode des 1. Elements in Liste datatosend
- DataValue\_1: Wert des 1. Elements in Liste datatosend
- DataStatus\_2: Statuscode des 2. Elements in Liste datatosend
- DataValue\_2: Wert des 2. Elements in Liste datatosend
- ...

## Dienst: unsubscribe

16567

**Name:** unsubscribe

**Beschreibung:** Der Dienst löscht ein bestehendes Abonnement. Das unsubscribe ist erfolgreich, wenn die cid und die Callback-Adresse registriert sind für ein aktives Abonnement (subscribe). Wird im callback der STRING "DELETE" übergeben, löscht der IO-Link Master alle aktiven Abonnements.

**Anfragedaten (Feld "data"):**

Datenfeld	Pflichtfeld	Datentyp	Beschreibung
callback	obligatorisch	STRING	Adresse, an die IoT-Core Ereignisbenachrichtigungen senden soll; kompletter URL: http://ipaddress:port/path

**Rückgabedaten (Feld "data"):** keine

## Dienst: validation\_useconnecteddevice

27933

**Name:** validation\_connecteddevice

**Beschreibung:** Der Dienst prüft, ob Geräte-ID und Hersteller-ID des angeschlossenen IO-Link Devices mit den Datenpunkten ../validation\_vendorid und ../validation\_deviceid übereinstimmen.

**Anwendbar auf:** Objekte vom Typ structure

**Anfragedaten (data):** keine

**Rückgabedaten (data):** keine

# 15 Index

## A

ADS over EtherCAT nutzen.....	84
Allgemein .....	7
Allgemeine Funktionen .....	41
Änderungshistorie.....	6
Anhang.....	91
Auf den ifm-IoT-Core zugreifen .....	37
Ausgänge.....	93
Azyklische Daten .....	99
Azyklische Dienste nutzen.....	84

## B

Bedien- und Anzeigeelemente .....	19
Beispiel .....	
Abonnement ändern .....	62
Abonnements prüfen .....	64
Anfrage mit Authentifikation.....	47
Benachrichtigungen abmelden .....	64
Benachrichtigungen abonnieren .....	44, 61
Benachrichtigungen im CSV-Format abonnieren.....	63
Bezeichnung des IO-Link Master ändern.....	60
Datenspeicher eines IO-Link Ports klonen.....	50
Digitalen Ausgangswert schreiben (Betriebsart .....	53
Digitalen Eingangswert lesen (Betriebsart .....	54
Eigenschaften eines Elements lesen .....	41
Firmware aktualisieren .....	59
Gerätebeschreibung durchsuchen.....	43
GET-Request .....	37
IO-Link Prozesswert lesen (Betriebsart .....	52
IO-Link Prozesswert schreiben (Betriebsart .....	53
Mehrere Parameterwerte des IO-Link Master gleichzeitig lesen.....	43
MQTT-Kommandokanal konfigurieren.....	69
Parameterwert eines IO-Link Devices ändern .....	56
Parameterwert eines IO-Link Devices lesen .....	55
Passwort rücksetzen .....	48
POST-Request .....	39
Sicherheitsmodus aktivieren.....	47
Teilbaum ausgeben.....	42
Temperatur an MQTT-Broker veröffentlichen .....	70
WebSockets nutzen .....	65
Benachrichtigung löschen .....	73
Benachrichtigungen abonnieren.....	61
Benachrichtigungen verwalten .....	72
Bestimmungsgemäße Verwendung .....	9
Betrieb.....	87

## C

CANopen over EtherCAT nutzen .....	85
------------------------------------	----

## D

Device Status / Port Status (0xF000) .....	105
Diagnose- und Statusinformationen lesen .....	82
Diagnosis History (0x10F3) .....	108
Diagnostics Data (0xA000).....	104
Dienst .....	
factoryreset .....	121
getblobdata .....	122
getdata .....	122
getdatamulti.....	123
getelementinfo.....	123

getidentity .....	124
getsubscriberlist .....	125
getsubscriptioninfo .....	126
gettree.....	127
install .....	128
iolreadacyclic.....	128
iolwriteacyclic .....	128
querytree .....	129
reboot .....	129
reset .....	129
setblock .....	130
setdata .....	131
signal .....	131
start .....	131
start_stream_set .....	132
stop .....	132
stream_set .....	132
subscribe.....	133
unsubscribe.....	134
validation_useconnecteddevice.....	134
Digitale Eingänge.....	12

## E

Eingänge.....	93
Eingänge / Ausgänge .....	92
Einsatzbereich .....	92
Elektrische Daten.....	92
Elektrischer Anschluss .....	14, 95
Elemente im Gerätebaum suchen.....	74
Erste Schritte .....	41
ESI-Datei.....	97
ESI-Datei installieren .....	78
EtherCAT .....	11, 78, 96
Hinweise für Programmierer .....	84
EtherCAT-Module .....	97
EtherCAT-Ports verbinden .....	15
EtherCAT-Schnittstelle .....	20
Events .....	108

## F

Feldbusparameter.....	97
Feldbus .....	
EtherCAT-Schnittstelle konfigurieren.....	30
Feldbus-Schnittstelle konfigurieren .....	49
Firmware .....	
Gerät auf Werkseinstellungen rücksetzen .....	33
Gerät neu starten .....	33
Firmware aktualisieren .....	77, 88
Funktion .....	10

## G

Gateway .....	
Anwendungskennung einstellen.....	60
Firmware aktualisieren .....	59
Gerät rücksetzen, neu starten und lokalisieren.....	57
Geräteinformationen lesen .....	58
Zustands- und Diagnoseinformationen lesen .....	58
Gerät anschließen .....	18
Gerät montieren .....	13
GET-Request.....	37

## H

Hinweis .....	99
---------------	----

Sicherheitsmodus .....	46
Verbindungszustände .....	68
Hinweise .....	14, 25
Hinweise für Programmierer .....	36

## I

ifm-IoT-Core .....	35, 112
Inbetriebnahme .....	22
Info .....	
Geräteinformationen zeigen .....	33
Internet of Things (IoT) .....	11
IO-Link .....	11
IO-Link Acyclic Command (0x3100) .....	101
IO-Link Device tauschen .....	88
IO-Link Devices .....	
Auf Parameter zugreifen .....	55
Geräteinformationen lesen und schreiben .....	57
IO-Link Events anzeigen .....	57
IO-Link Devices für Class-A-Betrieb anschließen .....	16
IO-Link Devices für Class-B-Betrieb anschließen .....	17
IO-Link Devices konfigurieren .....	34, 83
IO-Link Events lesen .....	82
IO-Link Inputs (0x6000) .....	102
IO-Link Master in TwinCAT-Projekt einbinden .....	78
IO-Link Master konfigurieren .....	75
IO-Link Outputs (0x7000) .....	102
IO-Link Ports .....	16
Betriebsart Pin 4 (US) einstellen .....	49
Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren .....	52
Gerätevalidierung und Datenspeicherung konfigurieren .....	50
Port-Events anzeigen .....	55
Prozessdaten lesen / schreiben .....	52
IO-Link Ports (Class A) .....	21
IO-Link-Master konfigurieren .....	79
IO-Link-Ports .....	
Betriebsart konfigurieren .....	31
Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen .....	30
Gerätevalidierung und Datenspeicherung einstellen .....	32
IO-Link-Ports konfigurieren .....	79
IO-Link-Versorgung .....	12
IoT .....	
IP-Einstellungen konfigurieren .....	26, 45
Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren .....	29
Sicherheitsmodus konfigurieren .....	27, 45
Zugriffsrechte konfigurieren .....	28, 45
IoT Core .....	
Allgemeine Informationen .....	36
IoT-Core .....	
Diagnosecodes .....	40
IoT-Core-Visualizer nutzen .....	71
IoT-Port .....	15
IT-Sicherheit .....	8

## K

Kommunikation, Parametrierung, Auswertung .....	11
---	----

## L

LED-Anzeigen .....	20
LR DEVICE .....	24

## M

Manufacturer Specific Index (0x2000) .....	100
Mapping .....	
Device Status (0xF100) .....	106
Diagnosis Message .....	110
Port Qualifier (0xF101) .....	107
MDP Standard Information (0x1000) .....	99
Mechanische Daten .....	94
Montage .....	13
MQTT-Kommandokanal konfigurieren .....	67
MQTT-Unterstützung .....	67

## N

Neue Benachrichtigung erstellen .....	72
Notwendige Vorkenntnisse .....	7

## O

Offline-Parametrierung .....	25
Optische Signalisierung .....	12

## P

Parameterdaten .....	97
Parametrierung .....	11, 23
Port Configuration (0x8000) .....	103
Port Mode (0x9000) .....	104
POST-Request .....	38
Prinzip der azyklischen Kommandoabarbeitung .....	86
Process Data Objects (PDO) .....	98
Profil .....	
blob .....	113
deviceinfo .....	114
devicetag .....	114
iolinkdevice_full .....	115
iolinkmaster .....	115
mqttCmdChannel .....	116
mqttCmdChannelSetup .....	116
mqttConnection .....	116
mqttSetup .....	117
network .....	117
parameter .....	118
processdata .....	118
runcontrol .....	118
service .....	118
software .....	118
software/uploadedablessoftware .....	119
timer .....	119
Prozessdaten lesen und schreiben .....	76

## R

Rechtliche Hinweise .....	5
Reinigung .....	88
Rückfallwerte der IO-Link Ports .....	25

## S

Schnittstellen .....	93
Sicherheitshinweise .....	7
Sicherheitsmodus .....	11
Sicherheitssymbole auf dem Gerät .....	7
Spannungsversorgung .....	21
Speicherdauer einstellen .....	44
Status-LEDs .....	20



## T

Technische Daten .....	92
------------------------	----

## U

Übersicht .....	19
IoT-Dienste .....	121
IoT-Profil .....	113
IoT-Typen .....	120
Umgebungsbedingungen .....	94

## V

Vorbemerkung .....	5
--------------------	---

## W

Wartung, Instandsetzung und Entsorgung .....	88
Web Socket nutzen .....	65
Web-based Management nutzen .....	87
Werkseinstellungen .....	89

## Z

Zeichenerklärung .....	6
Zubehör .....	90
Zulassungen / Prüfungen .....	94
Zweck des Dokuments .....	5
Zyklische Daten .....	98
Zyklische Prozessdaten konfigurieren .....	80
Zyklische Prozessdaten lesen und schreiben .....	81