



Gerätehandbuch

IO-Link Master mit Modbus TCP-Schnittstelle

CabinetLine

8 Ports

IP 20

AL1940

Firmware: 3.1.x

Deutsch

Inhaltsverzeichnis

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

8	Inbetriebnahme	22
9	Parametrierung	23
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: IP-Einstellungen konfigurieren	30
9.1.7	Fieldbus: Länge der Prozessdaten einstellen.....	31
9.1.8	IO-Link-Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen.....	31
9.1.9	IO-Link-Ports: Betriebsart konfigurieren	32
9.1.10	IO-Link-Ports: Gerätevalidierung und Datenspeicherung einstellen	33
9.1.11	IO-Link-Ports: Rückfallwerte einstellen.....	34
9.1.12	Info: Geräteinformationen zeigen	34
9.1.13	Firmware: Gerät auf Werkseinstellungen rücksetzen	35
9.1.14	Firmware: Gerät neu starten.....	35
9.1.15	IO-Link Devices konfigurieren	35
9.2	ifm-IoT-Core	37
9.2.1	Hinweise für Programmierer.....	38
9.2.2	Erste Schritte	43
9.2.3	Allgemeine Funktionen	43
9.2.4	IoT: Zugriffsrechte konfigurieren.....	48
9.2.5	IoT: IP-Einstellungen konfigurieren	48
9.2.6	IoT: Schnittstelle zu LR AGENT und LR SMARTOBSERVER konfigurieren.....	49
9.2.7	IoT: Sicherheitsmodus konfigurieren	49
9.2.8	Fieldbus: IP-Einstellungen konfigurieren	52
9.2.9	IO-Link Ports: Betriebsart Pin 4 (US) einstellen.....	52
9.2.10	IO-Link Ports: Gerätevalidierung und Datenspeicherung konfigurieren.....	53
9.2.11	IO-Link Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren	55
9.2.12	IO-Link Ports: Prozessdaten lesen / schreiben.....	55
9.2.13	IO-Link Ports: Port-Events anzeigen	58
9.2.14	IO-Link Devices: Auf Parameter zugreifen	58
9.2.15	IO-Link Devices: Geräteinformationen lesen und schreiben.....	60
9.2.16	IO-Link Devices: IO-Link Events anzeigen	60
9.2.17	Gateway: Gerät rücksetzen, neu starten und lokalisieren	60
9.2.18	Gateway: Geräteinformationen lesen	61
9.2.19	Gateway: Zustands- und Diagnoseinformationen lesen	61
9.2.20	Gateway: Firmware aktualisieren	62
9.2.21	Gateway: Anwendungskennung einstellen.....	63
9.2.22	Benachrichtigungen abonnieren	64
9.2.23	Web Socket nutzen	68
9.2.24	MQTT-Unterstützung.....	70
9.2.25	IoT-Core-Visualizer nutzen	74
9.3	Modbus TCP	81
9.3.1	AL1940 in Modbus-Projekt einbinden.....	81
9.3.2	IO-Link Master konfigurieren	82
9.3.3	IO-Link Ports konfigurieren.....	83
9.3.4	Eingangsdaten mehrerer IO-Link Ports lesen.....	84
9.3.5	Eingangsdaten einzelner IO-Link Ports lesen.....	85
9.3.6	Ausgangsdaten mehrerer IO-Link Ports schreiben.....	86
9.3.7	Ausgangsdaten einzelner IO-Link Ports schreiben.....	87
9.3.8	Diagnoseinformationen und Events lesen	87
9.3.9	Geräteinformationen lesen	89
9.3.10	IO-Link Master steuern	89
9.3.11	IO-Link Devices konfigurieren	89
9.3.12	Modbus TCP: Hinweise für Programmierer	90

10	Betrieb	92
10.1	Web-based Management nutzen	92
11	Wartung, Instandsetzung und Entsorgung	93
11.1	Reinigung	93
11.2	Firmware aktualisieren	93
11.3	IO-Link Device tauschen	93
12	Werkseinstellungen	94
13	Zubehör	95
14	Anhang	96
14.1	Technische Daten	97
14.1.1	Einsatzbereich	97
14.1.2	Elektrische Daten	97
14.1.3	Eingänge / Ausgänge	97
14.1.4	Eingänge	98
14.1.5	Ausgänge	98
14.1.6	Schnittstellen	98
14.1.7	Umgebungsbedingungen	99
14.1.8	Zulassungen / Prüfungen	99
14.1.9	Mechanische Daten	99
14.1.10	Elektrischer Anschluss	100
14.2	Modbus TCP	101
14.2.1	Register	102
14.2.2	Azyklische Kommandos	119
14.3	ifm-IoT-Core	128
14.3.1	Übersicht: IoT-Profile	129
14.3.2	Übersicht: IoT-Typen	136
14.3.3	Übersicht: IoT-Dienste	137
15	Index	151

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)
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ www.can-cia.org)
- CODESYS™ ist Eigentum der CODESYS GmbH, Deutschland (→ www.codesys.com)
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ 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)
- ISOBUS ist Eigentum der AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ www.aef-online.org)
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ www.microsoft.com)
- Modbus® ist Eigentum der Schneider Electric SE, Frankreich (→ www.schneider-electric.com)
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ 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 Modbus TCP-Schnittstelle CabinetLine 8 Port IP 20“ (Art.-Nr.: AL1940).

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

51760

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"> Hinzugefügt: Neue IoT-Core-Funktionen Hinzugefügt: IoT Core Visualizer Korrektur: Beschreibung SENS PWR und AUX PWR Korrektur: Beschreibung des IoT-Core-Dienstes getsubscriptioninfo 	10 / 2020

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 Modbus TCP-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. [97](#))).

4 Funktion

Inhalt

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

7482

4.1 Kommunikation, Parametrierung, Auswertung

Inhalt

IO-Link	11
Modbus TCP	11
Internet of Things (IoT)	11
Sicherheitsmodus	11
Parametrierung	12
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)

4.1.2 Modbus TCP

2259

Das Gerät bietet folgende Modbus TCP-Funktionen:

- Bereitstellung der Funktionen eines Modbus TCP Slave
- 2-Port-Switch für den Zugriff auf die Modbus TCP-Schnittstelle (X21/X22)
- Gateway für Übertragung der Prozess- und Parameterdaten zwischen den angeschlossenen IO-Link Devices und der übergeordneten Modbus TCP-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 AL1940 mit Parametriersoftware LR DEVICE, Modbus TCP-Projektierungssoftware oder ifm-LoT-Core-Diensten
- Parametrierung der angeschlossenen IO-Link Devices (Sensoren, Aktuatoren) mit Parametriersoftware LR DEVICE, Modbus TCP-Projektierungssoftware oder ifm-LoT-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 Modbus TCP-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
Modbus TCP-Ports verbinden	14
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.

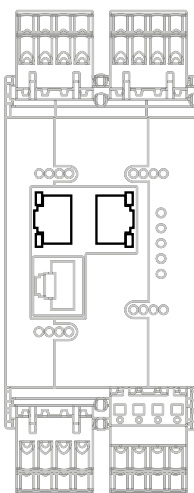
Anschlussbelegung: → **Technische Daten** (→ S. [97](#))

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 Modbus TCP-Ports verbinden

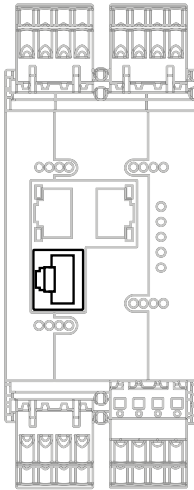
17858



- Gerät über Buchsen X21 und/oder X22 mit dem Modbus TCP-Netzwerk verbinden.
- Für den Anschluss Steckverbindungen mit mindestens der Schutzart IP 20 verwenden (→ **Zubehör** (→ S. [95](#))).

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. [95](#))).

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.

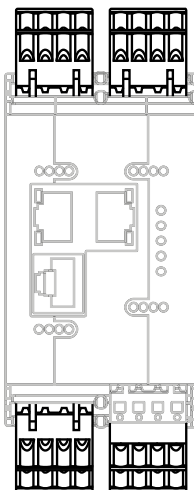
► Vor Inbetriebnahme den korrekten Anschluss der Versorgungsverbindungen prüfen.

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.



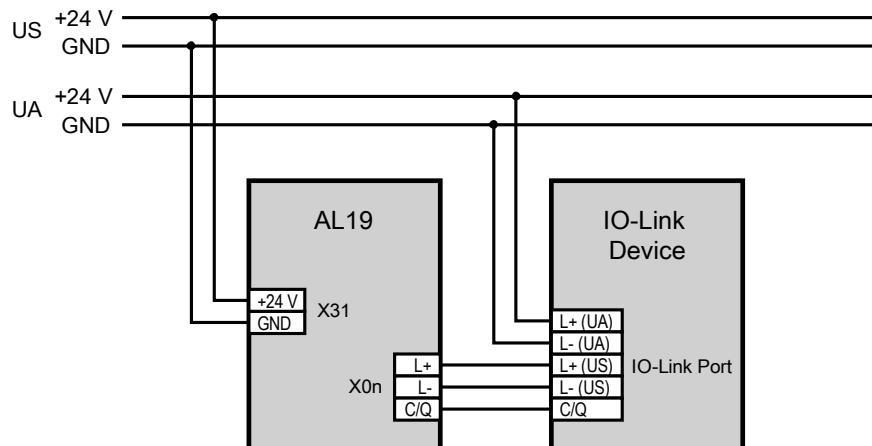
- IO-Link Devices mit den Ports X01...X08 verbinden.
 - Maximale Leitungslänge pro IO-Link Port: 20 m
- Für den Anschluss nur Verbindungsleitungen mit mindestens der Schutzart IP 20 verwenden.

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

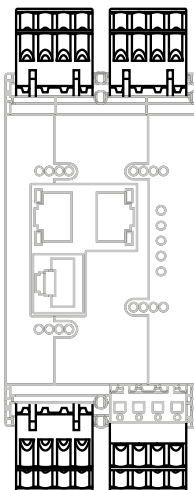


WARNUNG

Nichtbeachtung der galvanischen Trennung der Stromkreise

> Brandgefahr!

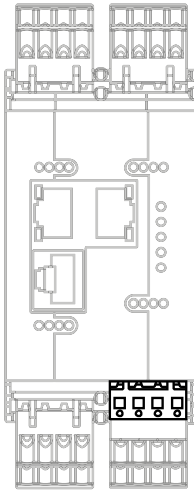
- ▶ Sicherstellen, dass externe Versorgung UA vom Stromkreis des IO-Link Masters galvanisch getrennt ist unter Beachtung von Basisisolierung (nach IEC 61010-1, Sekundärstromkreis mit maximal 30 V DC, abgeleitet von Netzstromkreis bis 300 V der Überspannungskategorie II).
- ▶ Sicherstellen, dass IO-Link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

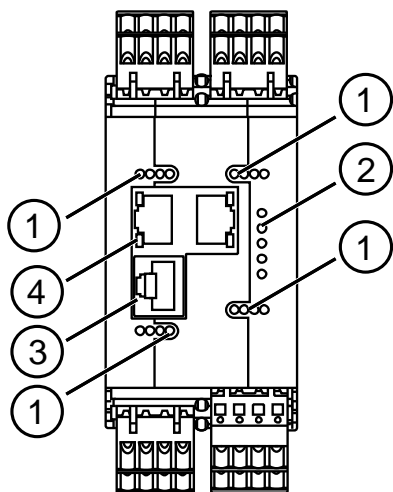
Inhalt

Überblick.....	19
LED-Anzeigen	20

5440

7.1 Überblick

10998



- 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 LNK und ACT der Modbus
TCP-Ports 1 (X21) und 2 (X22) (→ **Ethernet-Ports**
(→ S. [20](#)))

7.2 LED-Anzeigen

22024

Das Gerät verfügt über folgende LED-Anzeigen.

7.2.1 Status-LEDs

11748

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

Die LED mit der Bezeichnung RUN zeigt den aktuellen Zustand der Modbus TCP-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
ERR	rot	ein	Kommunikationsfehler
		blinkt 10 Hz	Boot-Fehler
		blinkt (200 ms ein, 200 ms aus, 200 ms ein, 1000 ms aus)	Watchdog-Fehler (Modbus TCP oder Prozessdaten)
		blinkt (200 ms ein, 1000 ms aus)	Lokaler Fehler
		blinkt 2,5 Hz	Ungültige Konfiguration
		aus	kein Fehler
RUN	grün	ein	Verbindung hergestellt
		blinkt 1 Hz	Bereit, aber noch nicht konfiguriert
		blinkt 5 Hz	Auf Verbindung wartend
		aus	Nicht bereit

7.2.2 Ethernet-Ports

22027

Jeder Ethernet-Port verfügt über 2 LEDs mit der Bezeichnung LNK und ACT. Die LEDs zeigen den Status der Ethernet-Verbindung.

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

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 AL1940 mit den Werkseinstellungen. Die Anzeigelemente signalisieren den aktuellen Betriebszustand (→ **Bedien- und Anzeigeelemente** (→ S. [19](#))).

Um die Parametrierung des AL1940 zu ermöglichen, müssen die IoT-Schnittstelle und / oder die Feldbus-Schnittstelle der Netzwerkumgebung entsprechend konfiguriert werden.

- ▶ Feldbus-Schnittstelle konfigurieren (→ **Feldbus: IP-Einstellungen konfigurieren** (→ S. [30](#)) oder → **Feldbus: IP-Einstellungen konfigurieren** (→ S. [52](#))).
- ▶ IoT-Schnittstelle konfigurieren konfigurieren (→ **IoT: IP-Einstellungen konfigurieren** (→ S. [26](#)) oder → **IoT: IP-Einstellungen konfigurieren** (→ S. [48](#))).
- > IoT-/Feldbus-Schnittstelle hat gültige IP-Einstellungen.
- > Anwender kann AL1940 parametrieren.

Weitere Schritte:

- Optional: Firmware des AL1940 aktualisieren (→ **Firmware aktualisieren** (→ S. [93](#))).
- AL1940 parametrieren (→ **Parametrierung** (→ S. [23](#))).

9 Parametrierung

Inhalt

LR DEVICE.....	24
ifm-IoT-Core	37
Modbus TCP.....	81

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: IP-Einstellungen konfigurieren	30
Fieldbus: Länge der Prozessdaten einstellen	31
IO-Link-Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen	31
IO-Link-Ports: Betriebsart konfigurieren	32
IO-Link-Ports: Gerätevalidierung und Datenspeicherung einstellen	33
IO-Link-Ports: Rückfallwerte einstellen	34
Info: Geräteinformationen zeigen	34
Firmware: Gerät auf Werkseinstellungen rücksetzen	35
Firmware: Gerät neu starten	35
IO-Link Devices konfigurieren	35

22822

Im Auslieferungszustand ist der AL1940 mit den Werkseinstellungen konfiguriert
(→ **Werkseinstellungen** (→ S. [94](#))).

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

9.1.1 Hinweise

Inhalt

Offline-Parametrierung	25
Parametrierung mit LR DEVICE	25

22369

Offline-Parametrierung

22405

Der AL1940 unterstützt die Offline-Parametrierung. Dabei erstellt der Anwender eine Konfiguration für den IO-Link Master und die angeschlossenen IO-Link Devices, ohne mit dem AL1940 verbunden zu sein (OFFLINE-Modus). Die so erstellte Konfiguration kann als Datei (*.lrp) gespeichert und später auf den AL1940 geladen und aktiviert werden.



Weiter Infos zur Offline-Parametrierung: → Bedienungsanleitung LR DEVICE

Parametrierung mit LR DEVICE

10924

Die Parametrierung des AL1940 mit dem LR DEVICE ist nur möglich über die IoT-Schnittstelle X23.

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. [94](#))).

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	[Modbus TCP + IoT]*	<ul style="list-style-type: none"> ▪ Modbus TCP und IoT-Core haben Lese- und Schreibrechte auf Parameter und Prozessdaten ▪ Modbus TCP und IoT-Core haben Leserechte auf Ereignisse/Alarmer
		[Modbus TCP + IoT (read-only)]	<ul style="list-style-type: none"> ▪ Modbus TCP hat Lese- und Schreibrechte auf Parameter und Prozessdaten ▪ Modbus TCP hat Leserechte auf Ereignisse/Alarmer ▪ IoT-Core hat Leserechte auf Parameter, Prozessdaten und Ereignisse/Alarmer
		[IoT only]	<ul style="list-style-type: none"> ▪ IoT-Core hat Lese- und Schreibrechte auf Parameter und Prozessdaten ▪ IoT hat Leserechte auf Ereignisse/Alarmer ▪ Modbus TCP hat keine Zugriffsrechte

* ... Werkeinstellung

- ▶ Geänderte Werte auf dem Gerät speichern.



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

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

Wenn in LR DEVICE der Parameter [Access Rights] = [Modbus TCP + 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 [Modbus TCP + IoT] setzen.

Änderungen des Parameters [Access Rights] sind erst wirksam nach einem Neustart des IO-Link Masters (→ **Firmware: Gerät neu starten** (→ S. 35)).

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: AL1940	



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: IP-Einstellungen konfigurieren



Die Konfiguration der IP-Einstellungen des Fieldbus-Ports ist nur über LR DEVICE und IoT möglich.

Um die IP-Einstellungen der Modbus TCP-Schnittstelle zu konfigurieren:

- ▶ Menü [Fieldbus] 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-Parameter werden vom Anwender eingestellt
		[DHCP]	IP-Parameter werden von einem DHCP-Server im Netzwerk eingestellt.
		[BOOTP]	IP-Parameter werden über das Bootstrap Protocol (BOOTP) eingestellt
[IP address]*	IP-Adresse des Modbus TCP-Ports	Werkseinstellung: 192.168.1.250	
[Subnet mask]*	Subnetzmaske des IP-Netzwerks	Werkseinstellung: 255.255.255.0	
[Default gateway IP address]*	IP-Adresse des Gateways	Werkseinstellung: 0.0.0.0	
[MAC address]	MAC-Adresse der Modbus TCP-Schnittstelle	Wert ist fest eingestellt.	
[Fieldbus firmware]	Firmware-Version des Modbus TCP-Stacks	z.B. 2.6.0.5	
[connectiontimeout]	max. Wert für Verbindungszeitüberschreitung (Wert in Millisekunden)	1 ... 300000	

* ... parameter can only be edited if parameter [DHCP] = [Static IP]

- ▶ Geänderte Werte auf dem Gerät speichern.

9.1.7 Fieldbus: Länge der Prozessdaten einstellen

17306

Um die Länge der zu übertragenen Prozessdaten und der Anordnung der Bytes einzustellen:

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

Name	Beschreibung	Mögliche Werte	
[Process data length]	Länge der Prozess-Eingangsdaten und Prozess-Ausgangsdaten pro IO-Link Port	2 Bytes Input 2 Bytes Output	2 Bytes Eingangsdaten, 2 Bytes Ausgangsdaten
		4 Bytes Input 4 Bytes Output	4 Bytes Eingangsdaten, 4 Bytes Ausgangsdaten
		8 Bytes Input 8 Bytes Output	8 Bytes Eingangsdaten, 8 Bytes Ausgangsdaten
		16 Bytes Input 16 Bytes Output	16 Bytes Eingangsdaten, 16 Bytes Ausgangsdaten
		32 Bytes Input 32 Bytes Output	32 Bytes Eingangsdaten, 32 Bytes Ausgangsdaten
[Swap]	Anordnung der Bytes in Prozessdaten	off	als Array of Bytes
		on	als Integer16-Wert; bei Aktualisierung der Prozessdaten werden Bytes wortweise getauscht (Eingangsdaten und Ausgangsdaten)

- ▶ Geänderte Werte auf dem Gerät speichern.

9.1.8 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.

9.1.9 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.

9.1.10 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.

9.1.11 IO-Link-Ports: Rückfallwerte einstellen

11752

Für den Fall der Unterbrechung der Modbus TCP-Verbindung können den Ausgängen der IO-Link Ports sichere Rückfallwerte zugewiesen werden.

Um die Rückfallwerte der 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	
[Fail-safe digital out]	Rückfallwert für Ausgang (Betriebsart "DO")	Reset	Wert rücksetzen (LOW)
		Old	alten Wert halten
		Set	Wert setzen (HIGH)
[Fail-safe IO-Link]	Rückfallwert für Ausgang (Betriebsart "IO-Link")	Off	kein Rückfallwert
		Reset	Wert rücksetzen
		Old	alten Wert halten
		Pattern	Muster ausgeben

- ▶ Geänderte Werte auf dem Gerät speichern.

9.1.12 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	AL1940
[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	

9.1.13 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.

9.1.14 Firmware: Gerät neu starten

18105

Bei einem Neustart des Geräts bleiben alle Einstellungen erhalten.

Um das AL1940 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.

9.1.15 IO-Link Devices konfigurieren

11033

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

Voraussetzungen:

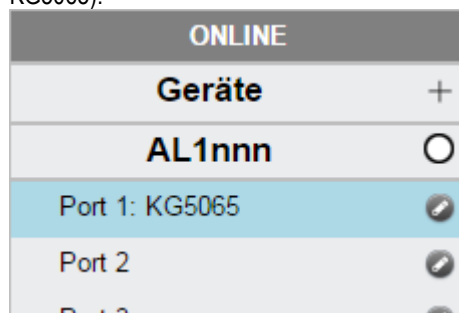
- > IO-Link Master ist korrekt installiert und mit der Parametriersoftware LR DEVICE verbunden.
- > IO-Link Device ist korrekt mit dem AL1940 verbunden.
- > Betriebsart des IO-Link-Ports ist "IO-Link" (→ **IO-Link-Ports: Betriebsart konfigurieren** (→ S. 32)).
- > IoT besitzt Schreibrechte auf dem IO-Link 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	38
Erste Schritte	43
Allgemeine Funktionen	43
IoT: Zugriffsrechte konfigurieren	48
IoT: IP-Einstellungen konfigurieren	48
IoT: Schnittstelle zu LR AGENT und LR SMARTOBSERVER konfigurieren	49
IoT: Sicherheitsmodus konfigurieren	49
Fieldbus: IP-Einstellungen konfigurieren	52
IO-Link Ports: Betriebsart Pin 4 (US) einstellen	52
IO-Link Ports: Gerätevalidierung und Datenspeicherung konfigurieren	53
IO-Link Ports: Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren	55
IO-Link Ports: Prozessdaten lesen / schreiben	55
IO-Link Ports: Port-Events anzeigen	58
IO-Link Devices: Auf Parameter zugreifen	58
IO-Link Devices: Geräteinformationen lesen und schreiben	60
IO-Link Devices: IO-Link Events anzeigen	60
Gateway: Gerät rücksetzen, neu starten und lokalisieren	60
Gateway: Geräteinformationen lesen	61
Gateway: Zustands- und Diagnoseinformationen lesen	61
Gateway: Firmware aktualisieren	62
Gateway: Anwendungskennung einstellen	63
Benachrichtigungen abonnieren	64
Web Socket nutzen	68
MQTT-Unterstützung	70
IoT-Core-Visualizer nutzen	74

31995



Allgemeine Hinweise zum ifm-LoT-Core: → **Hinweise für Programmierer** (→ S. [38](#))

9.2.1 Hinweise für Programmierer

Inhalt

IoT Core: Allgemeine Informationen.....	38
Auf den ifm-IoT-Core zugreifen	39
IoT-Core: Diagnosecodes.....	42

10989

IoT Core: Allgemeine Informationen

50737

Die Gerätefamilie CabinetLine verfügt über einen IoT-Core. Der IoT Core ermöglicht es dem Anwender, den AL1940 über eine REST API aus IT-Netzwerken heraus anzusprechen und in Internet-of-Things-Anwendungen einzubinden.

Auf dem AL1940 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 (→ IoT-Core: Diagnosecodes (→ S. 42))

Beispiel: GET-Request

52004

Anfrage (per Browser):

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

Rückgabe:

```
{
  "cid":-1,
  "data":{"value":"AL1940"},
  "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 (→ Übersicht: IoT-Dienste (→ S. 137))
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 (→ IoT-Core: Diagnosecodes (→ S. 42))

* = 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": "AL1940"},  
  "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 AL1940 zu lesen:

- ▶ Folgende POST-Anfrage an den AL1940 senden:
{"code": "request", "cid": -1, "adr": "gettree"}
- > AL1940 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 AL1940 besitzt den Typ device (→ **Übersicht: IoT-Typen** (→ S. [136](#))).

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": {
    "identifizier": "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"
        }
      }
    }
  }
}
```

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": {
    "identifizier": "firmware",

```

```

"type":"structure",
"profiles":[
"software","software/uploadablesoftware"],
"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"}},
{
"identifier":"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
}
```

DNS-Unterstützung

60202

Der IoT Core unterstützt den Dienst Domain Name System (DNS). Anstatt der konkreten IP-Adresse kann in Anfragen der entsprechende Hostname genutzt werden. DNS kann sowohl für die Adressierung des IoT Cores als auch für die Adressierung von Netzwerkteilnehmern genutzt werden.

Beispiel: DNS-Unterstützung nutzen

60242

Beispiel 1: gettree

Synonyme Anfragen:

- `http://192.168.23.70:8080/gettree`
- `http://example.org:8080/gettree`

Beispiel 2: subscribe

Synonyme Anfragen:

- mit IP-Adresse

```
{
  "cid": 11
  "code": 10,
  "adr": "setasync/datachanged/subscribe",
  "data":{
    "datatosend":["setasync"],
    "callback":"192.168.23.70:8080/dump"}
}
```

- mit Hostname

```
{
  "cid": 11
  "code": 10,
  "adr": "setasync/datachanged/subscribe",
  "data":{
    "datatosend":["setasync"],
    "callback":"http://example.com:8080/dump"}
}
```

Speicherdauer einstellen

60243

Der IoT Core bietet die Möglichkeit, die Speicherdauer von Daten und Benachrichtigungen einzustellen. Die Dienste **Dienst: setdata** (→ S. [147](#)) und **Dienst: subscribe** (→ S. [149](#)) 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
../accessrights	Zugriffsrechte auf den IO-Link Master	rw

rw ... lesen und schreiben



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

Wenn in IoT der Parameter [Access Rights] = [IoT only], dann in Modbus TCP-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. [35](#))).

9.2.5 IoT: IP-Einstellungen konfigurieren

60465

Unterstruktur: iotsetup

Verfügbare Datenpunkte:

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

rw ... lesen und schreiben

Anwendbare Dienste:

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



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

9.2.6 IoT: Schnittstelle zu LR AGENT und LR SMARTOBSERVER konfigurieren

22410

Unterstruktur: `iotsetup`

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
<code>../smobip</code>	IP-Adresse des LR AGENT oder LR SMARTOBSERVER	rw
<code>../smobport</code>	Portnummer des LR AGENT oder LR SMARTOBSERVER	rw
<code>../smobinterval</code>	Zykluszeit für Übertragung der Prozessdaten zum LR AGENT oder LR SMARTOBSERVER (Wert in ms)	rw

rw ... lesen und schreiben

9.2.7 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
<code>../security/securitymode</code>	aktiver Sicherheitsmodus	rw
<code>../security/password</code>	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

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. 140)).

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. 51)).

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
}
```

9.2.8 Fieldbus: IP-Einstellungen konfigurieren

60469

Unterstruktur: fieldbussetup

Verfügbare Datenpunkte:

Name	Beschreibung	Zugriff
../fieldbusfirmware	Firmware-Version des IO-Link Masters	r
../network/macaddress	MAC-Adresse des Feldbus-Ports	r
../network/ipaddress	IP-Adresse des Feldbus-Ports	rw*
../network/subnetmask	Subnetzmaske des Netzwerksegments	rw*
../network/ipdefaultgateway	IP-Adresse des Netzwerk-Gateways	rw*
../network/dhcp	DHCP-Client des Geräts aktivieren / deaktivieren	rw
../connectionstatus	Status der Verbindung zum Modbus TCP-Netzwerk	r

r ... read only

rw ... read and write

* ... nur änderbar, wenn sich die Modbus TCP-SPS nicht im Zustand RUNNING befindet

Anwendbare Dienste:

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



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

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

56695

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

Verfügbare Datenpunkte:

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

r ... nur lesen

rw ... lesen und schreiben

* ... nur änderbar, wenn Modbus TCP-SPS nicht in Zustand RUNNING

9.2.10 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 Modbus TCP-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 Modbus TCP-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. Lesenebefehl: pos = 512, length = 256

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

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":
"aWZtfgIAAABBTDF4NXhfY25faXRfdIuMi43Nw..."}}

9.2.11 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

9.2.12 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 °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. 55)).
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  
}
```

9.2.13 IO-Link Ports: Port-Events anzeigen

58237

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

Verfügbare Datenpunkte:

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

r ... nur lesen



Ereignisse abonnieren: → **Benachrichtigungen abonnieren** (→ S. [64](#))

9.2.14 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` ($n = 1 \dots 8$)

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}
```

9.2.15 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

9.2.16 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. [64](#))

9.2.17 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

9.2.18 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 AL1940 kann mit dem Dienst `getidentity` gelesen werden (→ **Dienst: `getidentity`** (→ S. [140](#))).

9.2.19 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

9.2.20 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": {}}

9.2.21 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 AL1940 setzen.

Lösung: Den Parameter [Application Tag] mit dem Dienst setdata auf den Wert [AL1940] ändern.

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

- Anfrage (Request):

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/devicetag/applicationtag/setdata",
  "data": {"newvalue": "AL1940"}
}
```

- Rückgabe (Response):

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

9.2.22 Benachrichtigungen abonnieren

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"> IO-Link Device stecken IO-Link Device ziehen Betriebsart des IO-Link Ports geändert 	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. [141](#)) alle aktiven Abonnements anzeigen.

Beispiel: Benachrichtigungen abonnieren

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. [68](#))
- per MQTT (mqtt://): **Beispiel: MQTT-Kommandokanal konfigurieren** (→ S. [72](#))

- 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. 65)) 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. 65)) 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. 65)) 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
}
```

9.2.23 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 Datensinke 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 Datensinke 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"]}  
}
```

9.2.24 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"> 0: QoS Level 0 - PUBLISH (ohne Bestätigung) 1: QoS Level 1 - PUBLISH > PUBREC (Einfache Bestätigung) 2: QoS Level 2 - PUBLISH > PUBREC > PUBREL > PUBCOMP (Doppelte Bestätigung) 	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. [71](#))

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

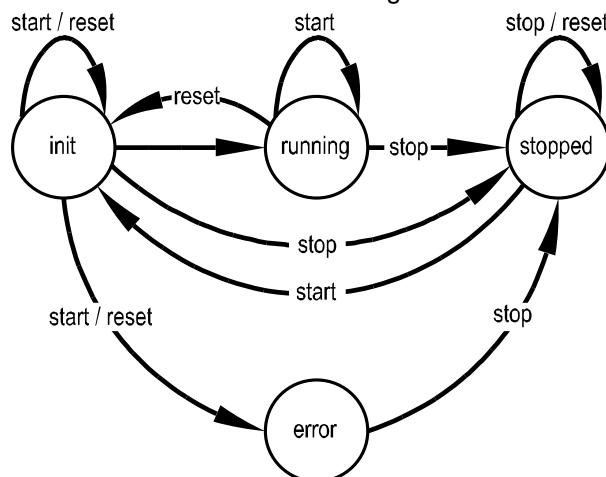
Wildcards "+" und "#" in Topics werden nicht unterstützt.

- ▶ 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
}
```

9.2.25 IoT-Core-Visualizer nutzen

Inhalt

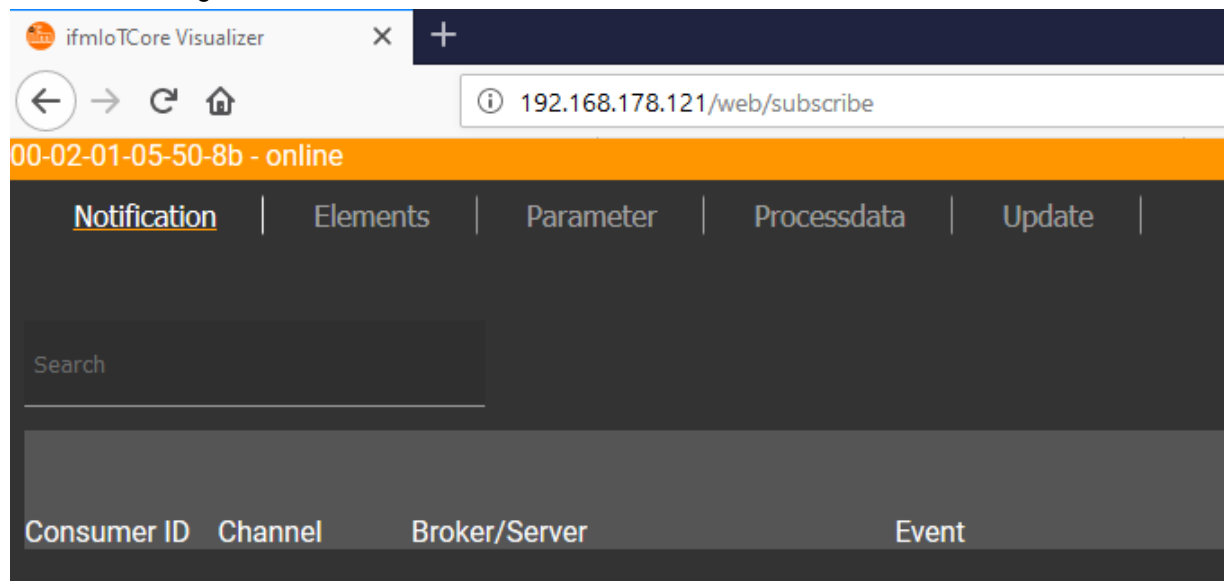
Benachrichtigungen verwalten	75
Elemente im Gerätebaum suchen	77
IO-Link Master konfigurieren	78
Prozessdaten lesen und schreiben	79
Firmware aktualisieren	80

60270

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

- lot-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 ...

- ☐ 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 ...

identifizier

profile

type

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 [identifizier], [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

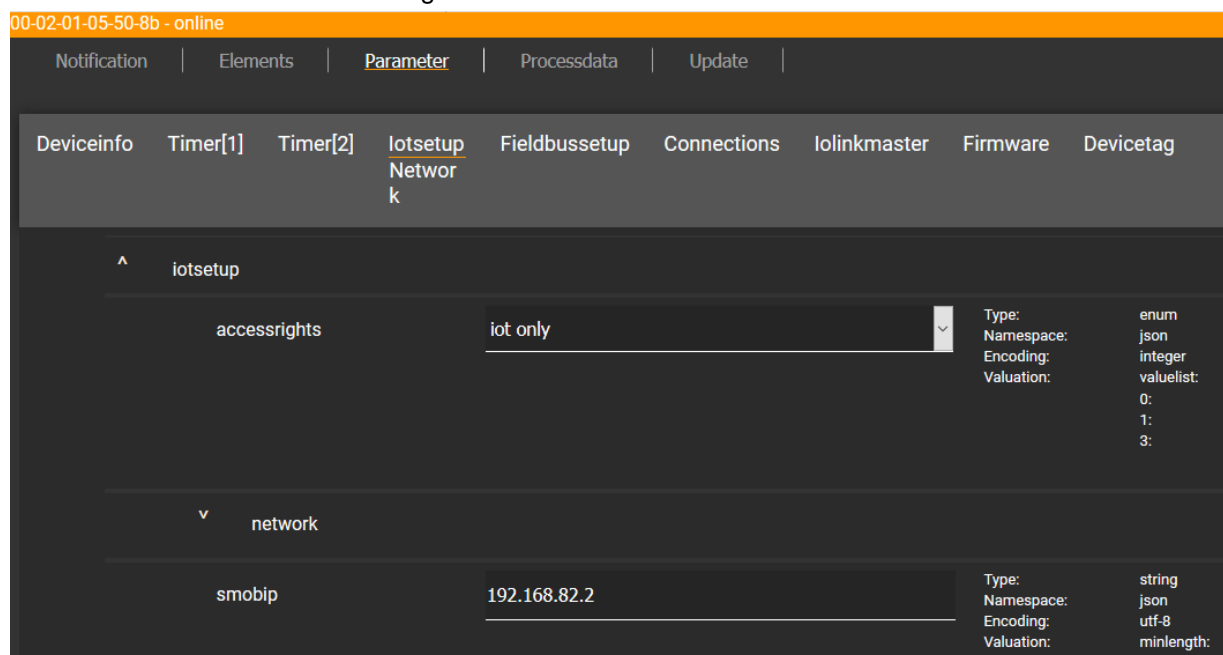
Die Menüseite [Parameter] ermöglicht es, den IO-Link Master zu konfigurieren.

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:

- Iot-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.

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

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

Processdatamaster | Timer[1] | Timer[2] | Fieldbussetup | Iolinkmaster
Port[1] Port[2] Port[3]
Port[4]

^ Iolinkmaster

^ port[1]

portevent	FF0200	Type:	string
		Namespace:	json
		Encoding:	hexstring

^ Iolinkdevice

vendorid	310	Type:	number
		Namespace:	json
		Encoding:	integer
		Valuation:	min: 0
			max: 65535

Um den Wert eines Prozessdatums zu ändern:

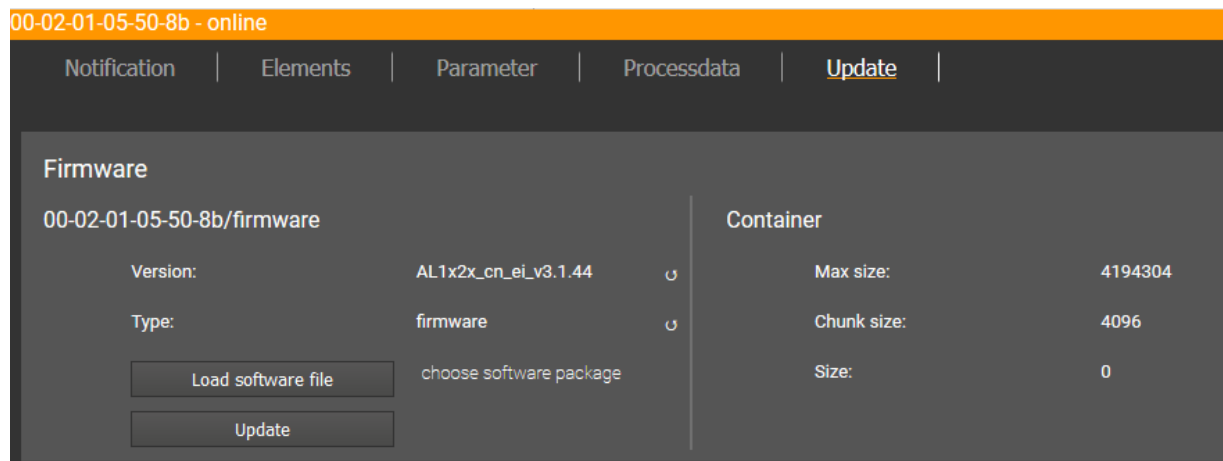
- ▶ In Gerätebeschreibung zum gewünschten Prozessdatum navigieren.
- ▶ Prozesswert ändern.
- ▶ Auf Stift-Symbol klicken, um die Änderung auf dem IO-Link Master zu speichern.
- > Geänderter Prozesswert ist aktiv.
- ▶ Optional: Vorgang wiederholen, um weitere Prozesswerte zu ändern.

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.



- 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 Modbus TCP

Inhalt

AL1940 in Modbus-Projekt einbinden	81
IO-Link Master konfigurieren	82
IO-Link Ports konfigurieren	83
Eingangsdaten mehrerer IO-Link Ports lesen	84
Eingangsdaten einzelner IO-Link Ports lesen	85
Ausgangsdaten mehrerer IO-Link Ports schreiben	86
Ausgangsdaten einzelner IO-Link Ports schreiben	87
Diagnoseinformationen und Events lesen	87
Geräteinformationen lesen	89
IO-Link Master steuern	89
IO-Link Devices konfigurieren	89
Modbus TCP: Hinweise für Programmierer	90

11614

Das Gerät kann felddbusseitig mit jeder Modbus TCP-fähigen Projektierungssoftware konfiguriert werden.

9.3.1 AL1940 in Modbus-Projekt einbinden

11754

Das AL1940 bietet die Funktionalität eines Modbus-TCP-Slaves. Der Anwender kann den IO-Link Master über das Profil eines generischen Modbus-TCP-Slaves in ein Felddbus-Projekt einbinden.

Die Konfiguration des IO-Link Masters, der IO-Link Ports und der Prozessdaten erfolgt über die Modbus-Register des AL1940.

Beispiel: IO-Link Master in eine CODESYS-Projekt einbinden

9612



Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!

- Modbus Master:
→ Online-Hilfe > Felddbusunterstützung > Modbus Configurator > Modbus Master
- Modbus Slave Device:
→ Online-Hilfe > Felddbusunterstützung > Modbus Configurator > Modbus Slave Device

Aufgabe: IO-Link Master in eine CODESYS-Projekt einbinden

Hardware:

- AC14 DL als Modbus-TCP-Master
- AL1940 als Modbus-TCP-Slave

Lösung:

Vorbereitung:

- CODESYS-Projekt mit AC14 DL erstellen.

1 Modbus-TCP-Master erstellen

- Im Gerätebaum: Rechtsklick auf X8
- > Kontextmenü erscheint.
- Im Kontextmenü: [Gerät anhängen...] wählen.
- > Dialogfenster erscheint.
- Folgende Einstellungen wählen:
1. [Vendor]: [ifm electronic] wählen.

2. [Gerät]: [Modbus_TCP_Master] wählen.
 3. [Name]: Einen eindeutigen Namen eingeben.
 - ▶ Auf [Gerät anhängen] klicken.
 - > Gerätebaum zeigt Modbus-TCP-Master als Unterknoten der Schnittstelle X8.
- 2 Modbus-TCP-Slave (AL1940) erstellen**
- ▶ Im Gerätebaum: Rechtsklick auf Knoten des hinzugefügten Modbus-TCP-Masters
 - ▶ Im Kontextmenü: [Gerät anhängen] wählen.
 - > Dialogfenster erscheint.
 - ▶ Folgende Einstellungen wählen:
 1. [Vendor]: [ifm electronic] wählen.
 2. [Gerät]: [Modbus_TCP_Slave] wählen.
 3. [Name]: Einen eindeutigen Namen eingeben
 - ▶ Auf [Gerät anhängen] klicken.
 - > Gerätebaum zeigt AL1940 als Unterknoten des Modbus-TCP-Masters.
- 3 Modbus-TCP-Slave konfigurieren**
- ▶ In folgenden Registerkarten die Parameter wie gewünscht einstellen:
 1. [General]: IP-Adresse und Unit-ID einstellen
 2. [Modbus Slave Channel]: Modbus-Register hinzufügen
 3. [Modbus TCPSlave I/O Mapping]: Werte der Modbus-Register den Variablen zuweisen

9.3.2 IO-Link Master konfigurieren

51664

Registerbereich für die Konfiguration des IO-Link Masters: → **Configuration Area** (→ S. [103](#))

Der Bereich beinhaltet folgende Daten:

- Zugriffsrechte auf den IO-Link Master
- Datenlänge der IO-Link-Eingangs- und Ausgangsdaten aller IO-Link Ports
- Ausrichtung der Bytes in einem Datenwort



- ▶ Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!
- ▶ Beim gleichzeitigen Schreiben mehrerer Register auf die korrekte Länge der übergebenen Daten achten!

Register	Inhalt	Zugriff
8998	Access Rights; Process Data Length	r/w
8999	Byte Swap	r/w

r/w ... Lesen und Schreiben

9.3.3 IO-Link Ports konfigurieren

Registerbereich für die Konfiguration der IO-Link Ports: → **Configuration Area** (→ S. [103](#))

Der Bereich beinhaltet folgende Daten:

- Betriebsart der IO-Link Ports
- Gerätevalidierung und Datenspeicherung
- Rückfallwerte der Ausgänge



- ▶ Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!
- ▶ Beim gleichzeitigen Schreiben mehrerer Register auf die korrekte Länge der übergebenen Daten achten!

Register	Inhalt	Zugriff
9000	Port X01: Port Configuration	r/w
9006	Port X02: Port Configuration	r/w
9012	Port X03: Port Configuration	r/w
9018	Port X04: Port Configuration	r/w
9024	Port X05: Port Configuration	r/w
9030	Port X06: Port Configuration	r/w
9036	Port X07: Port Configuration	r/w
9042	Port X08: Port Configuration	r/w

r/w ... Lesen und Schreiben

Zusätzlich kann der Anwender über folgende azyklische Kommandos die IO-Link Ports des AL1940 einstellen:

- "Set Mode": → **Kommando 0x10 – Set Mode** (→ S. [120](#))
- "Set Validation ID / Data Storage": → **Kommando 0x20 – Set Validation ID / Data Storage** (→ S. [122](#))
- "Set Fail-safe Data Pattern": → **Kommando 0x30 – Set Fail-safe Data Pattern** (→ S. [124](#))

Die Kommandos nutzen die Ablaufmechanismen des azyklischen Kommandokanals (→ **Azyklische Dienste nutzen** (→ S. [91](#))).

9.3.4 Eingangsdaten mehrerer IO-Link Ports lesen

Registerbereich für den kompakten Zugriff auf die Eingangsdaten der IO-Link Ports X01...X04 und X05...X08: → **Input Data** (→ S. [108](#))

Der Bereich enthält folgende Daten:

- zusammengefasste digitale Eingänge - Klemme 2 / Klemme 4 (DI)
- Statusinformationen der IO-Link Ports
- Statusinformationen der IO-Link Devices
- zusammengefasste IO-Link-Eingangsdaten



- Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!

Der Parameter "Invalid Data" zeigt, ob die gelesenen IO-Link Eingangsdaten gültig sind.

- Zusammen mit den Eingangsdaten auch die dazugehörigen Statusinformationen der IO-Link Devices lesen und auswerten!

Register	Inhalt	Zugriff
197	Port X01...X04: Digital Input - Klemme 2 / Klemme 4 (DI)	r
198	Port X01...X04: Status Information IO-Link Ports	r
199	Port X01...X04: Status Information IO-Link Devices	r
200	Port X01...X04: Compact Input Data - IO-Link (4n Bytes)	r
297	Port X05...X08: Digital Input - Klemme 2 / Klemme 4 (DI)	r
298	Port X05...X08: Status Information IO-Link Ports	r
299	Port X05...X08: Status Information IO-Link Devices	r
300	Port X05...X08: Compact Input Data - IO-Link (4n Bytes)	r

r ... nur Lesen

n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ **Configuration Area** (→ S. [103](#)))

9.3.5 Eingangsdaten einzelner IO-Link Ports lesen

Registerbereich für den separaten Zugriff auf Eingangsdaten der einzelnen IO-Link Ports: → **Single Port Access** (→ S. [114](#))

Der Bereich enthält für jeden IO-Link Port X01...X08 folgende Daten:

- Digitale Eingangsdaten an Klemme 2 / Klemme 4 (DI)
- Statusinformationen des IO-Link Ports
- Diagnose- und Statusinformationen des angeschlossenen IO-Link Devices
- IO-Link Eingangsdaten



- Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!

Der Parameter "Invalid Data" zeigt, ob die gelesenen IO-Link Eingangsdaten gültig sind.

- Zusammen mit den Eingangsdaten des Ports auch die dazugehörigen Diagnoseinformationen lesen und auswerten!

Register	Inhalt	Zugriff
1000	Port X01: Digital Input - Pin 2 / Pin 4 (DI)	r
1001	Port X01: Diagnostic + Status Data	r
1002	Port X01: Input Data - IO-Link (n Bytes)	r
2000	Port X02: Digital Input - Pin 2 / Pin 4 (DI)	r
2001	Port X02: Diagnostic + Status Data	r
2002	Port X02: Input Data - IO-Link (n Bytes)	r
3000	Port X03: Digital Input - Pin 2 / Pin 4 (DI)	r
3001	Port X03: Diagnostic + Status Data	r
3002	Port X03: Input Data - IO-Link (n Bytes)	r
4000	Port X04: Digital Input - Pin 2 / Pin 4 (DI)	r
4001	Port X04: Diagnostic + Status Data	r
4002	Port X04: Input Data - IO-Link (n Bytes)	r
5000	Port X05: Digital Input - Pin 2 / Pin 4 (DI)	r
5001	Port X05: Diagnostic + Status Data	r
5002	Port X05: Input Data - IO-Link (n Bytes)	r
6000	Port X06: Digital Input - Pin 2 / Pin 4 (DI)	r
6001	Port X06: Diagnostic + Status Data	r
6002	Port X06: Input Data - IO-Link (n Bytes)	r
7000	Port X07: Digital Input - Pin 2 / Pin 4 (DI)	r
7001	Port X07: Diagnostic + Status Data	r
7002	Port X07: Input Data - IO-Link (n Bytes)	r
8000	Port X08: Digital Input - Pin 2 / Pin 4 (DI)	r
8001	Port X08: Diagnostic + Status Data	r
8002	Port X08: Input Data - IO-Link (n Bytes)	r

r ... nur Lesen

n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ **Configuration Area** (→ S. [103](#)))

9.3.6 Ausgangsdaten mehrerer IO-Link Ports schreiben

Registerbereich für den kompakten Zugriff auf die Ausgangsdaten der IO-Link Ports X01...X04 und X05...X08: → **Output Data** (→ S. [112](#))

Der Bereich enthält folgende Daten:

- Digitale Ausgangsdaten an Klemme 4 (DO)
- IO-Link Ausgangsdaten



- Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!

Der IO-Link Master schreibt in "Compact Output Data" nur die Ausgänge, welche von den übergebenen Ausgangsdaten komplett abgedeckt sind.

Beispiel: Die konfigurierte Prozessdatenlänge ist 4 Bytes. Wenn an Register 600 insgesamt 5 Worte an Ausgangsdaten übergeben werden, schreibt der IO-Link Master die Ausgänge X01 (Worte 1+2) und X02 (Worte 3+4). Der Ausgang X03 wird nicht geschrieben.

- Beim Schreiben der IO-Link Ausgänge auf die korrekte Länge der Ausgangsdaten achten!

Die Ausgangsdaten sind ungültig in folgenden Situationen:

- kein Ethernet-Kabel gesteckt
- SPS hat Verbindung beendet
- Verbindung zur SPS hat Zeitüberschreitung (Timeout)

Register	Inhalt	Zugriff
599	Port X01...X04: Digital Output - Pin 4 (DO)	r/w
600	Port X01...X04: Compact Output Data IO-Link (4n Bytes)	r/w
699	Port X05...X08: Digital Output - Klemme 4 (DO)	r/w
700	Port X05...X08: Compact Output Data IO-Link (4n Bytes)	r/w

r/w ... Lesen und Schreiben

n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ **Configuration Area** (→ S. [103](#)))

9.3.7 Ausgangsdaten einzelner IO-Link Ports schreiben

Registerbereich für den separaten Zugriff auf Ausgangsdaten der einzelnen IO-Link Ports: → **Single Port Access** (→ S. [114](#))

Der Bereich enthält für jeden IO-Link Port X01...X08 folgende Daten:

- Digitale Ausgangsdaten an Klemme 4 (DO)
- IO-Link Ausgangsdaten



- Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!

Wenn der Anwender versucht, mehr als 34 Byte Ausgangsdaten an einen einzelnen Port zu schreiben, bricht der IO-Link Master die Ausführung des Befehls ab und gibt den Fehlercode "2 ILLEGAL ADDRESS" zurück.

- Beim Schreiben von Ausgängen sicherstellen, dass die Länge der übergebenen Ausgangsdaten mit der konfigurierten Prozessdatenlänge übereinstimmt.

Die Ausgangsdaten sind ungültig in folgenden Situationen:

- kein Ethernet-Kabel gesteckt
- SPS hat Verbindung beendet
- Zeitüberschreitung bei Kommunikation mit SPS (Timeout)

Register	Inhalt	Zugriff
1100	Port X01: Digital Output - Pin 4 (DO)	r/w
1101	Port X01: Output Data IO-Link (n Bytes)	r/w
2100	Port X02: Digital Output - Pin 4 (DO)	r/w
2101	Port X02: Output Data IO-Link (n Bytes)	r/w
3100	Port X03: Digital Output - Pin 4 (DO)	r/w
3101	Port X03: Output Data IO-Link (n Bytes)	r/w
4100	Port X04: Digital Output - Pin 4 (DO)	r/w
4101	Port X04: Output Data IO-Link (n Bytes)	r/w
5100	Port X05: Digital Output - Pin 4 (DO)	r/w
5101	Port X05: Output Data IO-Link (n Bytes)	r/w
6100	Port X06: Digital Output - Pin 4 (DO)	r/w
6101	Port X06: Output Data IO-Link (n Bytes)	r/w
7100	Port X07: Digital Output - Pin 4 (DO)	r/w
7101	Port X07: Output Data IO-Link (n Bytes)	r/w
8100	Port X08: Digital Output - Pin 4 (DO)	r/w
8101	Port X08: Output Data IO-Link (n Bytes)	r/w

r/w ... Lesen und Schreiben

n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ **Configuration Area** (→ S. [103](#)))

9.3.8 Diagnoseinformationen und Events lesen

Registerbereich für den Zugriff auf Diagnoseinformationen der IO-Link Ports X01...X08: → **Diagnostic Data** (→ S. [105](#))

Der Bereich beinhaltet folgende Daten:

- Status-/Fehler-Flags für Port-Konfiguration
- Vendor ID / Device ID der angeschlossenen IO-Link Devices
- Events und zugehörige Event Codes



► Allgemeine Regeln für den Zugriff auf die Modbus-Register beachten (→ **Regeln für den Zugriff auf Modbus-Register** (→ S. [90](#)))!



Es werden maximal 3 Events pro IO-Link Port angezeigt.

Einmalige Ereignisse werden nach mindestens 10 s gelöscht (Event Single Shot).

Auftretende Ereignisse zeigen den Zeitpunkt des Fehlerseintritts an (Event appears). Wenn die Fehlerursache verschwindet, wird dies durch ein weiteres Ereignis angezeigt (Event disappears). Beide Ereignistypen treten immer paarweise auf.

Register	Inhalt	Zugriff
30	Port X01: Diagnostic Data	r
40	Port X02: Diagnostic Data	r
50	Port X03: Diagnostic Data	r
60	Port X04: Diagnostic Data	r
70	Port X05: Diagnostic Data	r
80	Port X06: Diagnostic Data	r
90	Port X07: Diagnostic Data	r
100	Port X08: Diagnostic Data	r

r ... nur Lesen

Zusätzliche Diagnose- und Statusinformationen werden in folgenden Registerbereichen bereitgestellt:

- Bereich "Input Data": → **Eingangsdaten mehrerer IO-Link Ports lesen** (→ S. [84](#))
- Bereich "Single Port Access": → **Eingangsdaten mehrerer IO-Link Ports lesen** (→ S. [84](#))

9.3.9 Geräteinformationen lesen

11039

Der Anwender kann Geräteinformationen mit dem FC43 lesen.

Der AL1940 unterstützt folgende Datensätze ("Read Device ID code"):

- Basic Device Identification (0x01): enthaltene Datenobjekte: → Modbus TCP-Spezifikation
- Regular Device Identification (0x02): enthaltene Datenobjekte: → Modbus TCP-Spezifikation
- Specific Device Identification (0x04): enthaltene Datenobjekte:

Objekt-ID	Objektname / Beschreibug	Datentyp	Mögliche Werte
0x00	VendorName	ASCII String	ifm electronic
0x01	ProductCode	ASCII String	1940
0x02	MajorMinorRevision	ASCII String	z.B. V1.001
0x03	VendorUrl	ASCII String	www.ifm.com
0x04	ProductName	ASCII String	IO-Link Master CL MOD 8P IP20
0x05	ModelName	ASCII String	1940
0x06	UserApplicationName	ASCII String	MODBUS IO-Link Master

9.3.10 IO-Link Master steuern

23382

Der Anwender kann mit folgenden azyklischen Kommandos den IO-Link Master steuern:

- "Reboot": → **Kommando 0x40 – Reboot** (→ S. [126](#))
- "Factory Reset": → **Kommando 0x50 – Factory Reset** (→ S. [127](#))

Die Kommandos nutzen die Ablaufmechanismen des azyklischen Kommandokanals (→ **Azyklische Dienste nutzen** (→ S. [91](#))).

9.3.11 IO-Link Devices konfigurieren

9031

Der IO-Link Master unterstützt die Konfiguration der angeschlossenen IO-Link Devices aus der Modbus TCP-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. [91](#)))

9.3.12 Modbus TCP: Hinweise für Programmierer

Inhalt

Regeln für den Zugriff auf Modbus-Register	90
Unterstützte Function Codes	90
Hinweis: Exception Codes	91
Azyklische Dienste nutzen	91

17965

Regeln für den Zugriff auf Modbus-Register

34351

Für den Zugriff auf die Modbus-Register gelten folgende allgemeine Regeln:

- Um Modbus-Register zu lesen oder zu schreiben, nur die gültigen Function Codes nutzen (→ **Unterstützte Function Codes** (→ S. [90](#))).
- Nach jedem Lese- oder Schreibzugriff die Gültigkeit der übertragenen Daten prüfen (→ **Mapping: PQI** (→ S. [115](#))).

Unterstützte Function Codes

13676

Der AL1940 unterstützt folgende Function Codes für den lesenden und/oder schreibenden Zugriff auf die Modbus-Register:

Function Code	Funktionsname / Beschreibung
03 (0x03)	Read Multiple Registers / Mehrere zusammenhängende Register lesen
04 (0x04)	Read Input Register / Eingangsregister lesen
06 (0x06)	Write Single Register / Einzelnes Register schreiben
16 (0x10)	Write Multiple Registers / Mehrere zusammenhängende Register schreiben
23 (0x17)	Read/Write Multiple Registers / Mehrere zusammenhängende Register lesen/schreiben
43 (0x2B)	Read Device Identification / Geräteinformationen lesen



Detaillierte Informationen zu Function Codes: → MODBUS-TCP-Spezifikation

Hinweis: Exception Codes

16557

Bei der fehlerfreien Abarbeitung einer Function-Code-Anfrage hat die Response-Nachricht folgenden Inhalt:

- Response Function Code: Request Function Code
- Response Data: angeforderte Daten

Tritt ein Fehler auf während des Zugriffs auf die Register, antwortet der AL1940 mit einem Error Code anstelle des Function Codes. Die Response-Nachricht hat folgenden Inhalt:

- Response Function Code: Error Code (= Request Function Code + 0x80)
- Response Data: Exception Code

Folgende Exception Codes sind verfügbar:

Exception Code	Name	Beschreibung
0x1	ILLEGAL FUNCTION	Ungültiger Function Code (Modbus-Funktion nicht implementiert)
0x2	ILLEGAL DATA ADDRESS	Ungültige Datenadresse (ungültiges Adresse oder Länge)
0x3	ILLEGAL DATA VALUE	Ungültiger Datenwert (ungültige Parameter; falsche Registeranzahl)
0x4	SERVER DEVICE FAILURE	Nicht behebbarer Fehler im Modbus Server während Abarbeitung

Azyklische Dienste nutzen

11046

Der AL1940 verfügt über eine Kommandoschnittstelle für die Ausführung azyklischer Kommandos. Ein azyklisches Kommando besteht aus einer Anforderung (Request) und einer Antwort (Response).

Register	Inhalt	Zugriff
500	Command Request Channel (Fieldbus PLC >>> IO-Link Master) (44 Bytes)	r/w
0	Command Response Channel (IO-Link Master >>> Fieldbus PLC) (44 Bytes)	r

Struktur des azyklischen Kommandokanals: → **Acyclic Command Channel** (→ S. [116](#))

Genereller Ablauf der azyklischen Kommunikation:

1 Command Request schreiben

- ▶ Im Anforderungskanal: Gewünschte Daten schreiben (außer [User ID]).
- ▶ Gewünschte [User ID] schreiben.
- > Geänderte [User ID] signalisiert neuen Befehl.
- > Im Antwortkanal: Register werden auf 0 gesetzt.
- > Azyklischer Kommandokanal wird blockiert.
- > Verarbeitung des Befehls wird gestartet.

2 Status prüfen

- ▶ Im Antwortkanal: Register [Command Status] prüfen.
- > Wenn [Command Status] <> 0: Weiter mit Schritt 3
- > Wenn [Command Status] == 0: Schritt 2 wiederholen.

3 Command Response lesen

- ▶ Im Antwortkanal: zurückgegebene Nutzdaten lesen.
- > Azyklischer Kommandokanal wird freigegeben.

10 Betrieb

Inhalt

Web-based Management nutzen	92
-----------------------------------	----

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.....	93
Firmware aktualisieren	93
IO-Link Device tauschen	93

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. [80](#))).

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 AL1940 verbinden.
- > IO-Link Master kopiert Parameterwerte aus dem Datenspeicher auf das neue IO-Link Device.

12 Werkseinstellungen

16542

In den Werkseinstellungen besitzt das Gerät folgende Parametereinstellungen:

Parameter	Werkseinstellung
[IP address] (Modbus TCP)	192.168.1.250
[Subnet mask] (Modbus TCP)	255.255.255.0
[IP gateway address] (Modbus TCP)	0.0.0.0
[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
[Modbus TCP name]	leer
Datenspeicher (Data Storage)	leer

13 Zubehör

17853

Zubehörliste des AL1940: → www.ifm.com > Produktseite > Zubehör

14 Anhang

Inhalt

Technische Daten.....	97
Modbus TCP.....	101
ifm-IoT-Core	128

7156

14.1 Technische Daten

Inhalt

Einsatzbereich	97
Elektrische Daten	97
Eingänge / Ausgänge	97
Eingänge.....	98
Ausgänge.....	98
Schnittstellen	98
Umgebungsbedingungen	99
Zulassungen / Prüfungen	99
Mechanische Daten	99
Elektrischer Anschluss	100

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

10921

Schnittstellen	
Kommunikationsschnittstelle	Ethernet; IO-Link
Kommunikationsschnittstelle	IO-Link; TCP/IP; Modbus TCP
Ethernet	
Übertragungsstandard	10Base-T; 100Base-TX
Übertragungsrate [MBit/s]	10; 100
Protokoll	TCP/IP; Modbus TCP
Werkseinstellungen	<ul style="list-style-type: none"> IP-Adresse: 192.168.1.250 Subnetzmaske: 255.255.255.0 Gateway IP-Adresse: 0.0.0.0 MAC-Adresse: siehe Typenschild
IO-Link Master	
Übertragungstyp	COM 1 / COM 2 / COM 3
IO-Link-Revision	V1.1
Anzahl Ports Class A	8
IoT-Schnittstelle	
Übertragungsstandard	10Base-T; 100Base-TX
Übertragungsrate [MBit/s]	10; 100
Protokoll	DCP, DCHP, Auto IP
Werkseinstellungen	<ul style="list-style-type: none"> IP-Adresse: 169.254.X.X Subnetzmaske: 255.255.0.0 Gateway IP-Adresse: 0.0.0.0 MAC-Adresse: siehe Typenschild

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"> ▪ EN 61000-6-2 ▪ EN 61000-6-4
MTTF [Jahre]	90

14.1.9 Mechanische Daten

22825

Mechanische Daten	
Gewicht [g]	329,5
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 Modbus TCP

Inhalt

Register	102
Azyklische Kommandos	119

22433

14.2.1 Register

Inhalt

Configuration Area.....	103
Diagnostic Data	105
Input Data	108
Output Data	112
Single Port Access	114
Acyclic Command Channel	116

18637

Der AL1940 speichert Konfigurationsdaten, Prozessdaten und Status-/Diagnosedaten in Modbus-Registern.

Configuration Area

8912

Register	Inhalt	
	Bits 8-15	Bits 0-7
8998	Access Rights	Process Data Length
8999	reserviert	Byte Swap
9000	Port X01: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9006	Port X02: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9012	Port X03: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9018	Port X04: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9024	Port X05: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9030	Port X06: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9036	Port X07: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	
9042	Port X08: Port Configuration (→ Mapping: Port-Konfiguration (→ S. 104))	

Legende:

- [Access Rights] Zugriffsrechte auf die Parameterdaten, Prozessdaten und die Events/Diagnosemeldungen des IO-Link Masters sowie der angeschlossenen IO-Link-Devices. 1 Byte
 - 0x00 Modbus TCP + IoT
 - 0x01 Modbus TCP + IoT (ro)
 - 0x02 Modbus TCP only
 - 0x03 Keep setting (default)
- [Process Data Length] Länge der Prozess-Eingangsdaten und Prozess-Ausgangsdaten 1 Byte
 - 0x00 2 Bytes Input / 2 Bytes Output Data
 - Input Data: 14 Words
 - Output Data: 10 Words
 - 0x01 4 Bytes Input / 4 Bytes Output Data
 - Input Data: 22 Words
 - Output Data: 18 Words
 - 0x02 8 Bytes Input / 8 Bytes Output Data
 - Input Data: 38 Words
 - Output Data: 34 Words
 - 0x03 16 Bytes Input / 16 Bytes Output Data
 - Input Data: 70 Words
 - Output Data: 66 Words
 - 0x04 32 Bytes Input / 32 Bytes Output Data
 - Input Data: 134 Words
 - Output Data: 130 Words
- [Byte Swap] Anordnung der Bytes in Prozessdaten 1 Byte
 - 0x00 als Array of Bytes
 - 0x01 als Integer16-Wert; bei Aktualisierung der Prozessdaten werden Bytes wortweise getauscht (Eingangsdaten und Ausgangsdaten)
- [Port Configuration] Konfiguration des IO-Link Ports 12 Byte → **Mapping: Port-Konfiguration** (→ S. [104](#))

Mapping: Port-Konfiguration

18639

Bits 8-15	Bits 0-7
Port Mode	Master Cycle Time
reserviert	Validation ID
Vendor ID	
reserviert	Device ID (MSB)
Device ID	Device ID (LSB)
Failsafe Mode -- IO-Link	Failsafe Mode -- Pin 4 (DO)

Legende:

▪ [Port Mode]	Betriebsart des IO-Link-Ports	1 Byte	0x00	deaktiviert
			0x01	digitaler Eingang (DI)
			0x02	digitaler Ausgang (DO)
			0x03	IO-Link
▪ [Master Cycle Time]	Zykluszeit der Datenübertragung zwischen IO-Link Master und IO-Link Device	1 Byte	0x00	As fast as possible
			0x01	2 Millisekunden
			0x02	4 Millisekunden
			0x03	8 Millisekunden
			0x04	16 Millisekunden
			0x05	32 Millisekunden
			0x06	64 Millisekunden
			0x07	128 Millisekunden
▪ [Validation ID]	Unterstützter IO-Link-Standard und Verhalten des IO-Link Masters bei Anschluss neuer IO-Link Devices am IO-Link-Port	1 Byte	0x00	keine Validierung
			0x01	V1.0 device
			0x02	V1.1 device
			0x03	V1.1 device with Backup + Restore
			0x04	V1.1 device with Backup
▪ [Vendor ID]	Vendor ID des Herstellers des Geräts am IO-Link-Port	2 Byte	0x0000...0xFFFF	
▪ [Device ID]	Device ID des Geräts am IO-Link-Port Device ID = 0x123456	3 Byte	pro Byte: 0x00...0xFF	
	▪ Device ID (MSB) = 0x12			
	▪ Device ID = 0x34			
	▪ Device ID (LSB) = 0x56			
▪ [Failsafe Mode -- IO-Link]	Rückfall-Modus für Ausgangsdaten bei Unterbrechung der Modbus TCP-Verbindung	1 Byte	0x00	No Failsafe
			0x01	Failsafe Reset Value
			0x02	Failsafe Old Value
			0x03	Failsafe with Pattern
▪ [Failsafe Mode -- Pin 4 (DO)]	Rückfallwert für Betriebsart "Digitaler Ausgang (DO)"	1 Byte	0x00	Failsafe Reset Value
			0x01	Failsafe Old Value
			0x02	Failsafe Set Value

Diagnostic Data

11746

Register	Inhalt	
	Bits 8-15	Bits 0-7
30	reserviert	Port X01: → Mapping: Diagnostics (→ S. 106)
31	Port X01: Vendor ID	
32	reserviert	Port X01: Device ID (MSB)
33	Port X01: Device ID	Port X01: Device ID (LSB)
34...39	Port X01: Events (→ Mapping: Events (→ S. 107))	
40	Port X02: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
50	Port X03: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
60	Port X04: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
70	Port X05: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
80	Port X06: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
90	Port X07: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	
100	Port X08: Diagnostic Data (Mapping: → Port X01 - Register 30...39)	

Legende:

- [Vendor ID] Vendor ID des Herstellers des Geräts am IO-Link-Port 2 Byte 0x0000...0xFFFF
- [Device ID] Device ID des Geräts am IO-Link-Port 3 Byte pro Byte: 0x00...0xFF
 Device ID = 0x123456
 - Device ID (MSB) = 0x12
 - Device ID = 0x34
 - Device ID (LSB) = 0x56

Mapping: Diagnostics

17305

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserviert	Wrong Length PD OUT	Wrong Length PD IN	Wrong Cycle Time	Wrong Vendor ID/ Device ID	reserviert	reserviert	IOL Mode

Legende:

- [IOL Mode] Betriebsart des IO-Link Ports 1 Bit 0x0 sonst.
0x1 IO-Link
- [Wrong Vendor ID/ Device ID] Prüfung, ob aktuelle und konfigurierte Vendor ID und Device ID übereinstimmen 1 Bit 0x0 OK
0x1 keine Übereinstimmung
- [Wrong Cycle Time] Prüfung, ob aktuelle und konfigurierte Zykluszeit übereinstimmen 1 Bit 0x0 OK
0x1 keine Übereinstimmung
- [Wrong Length PD IN] Prüfung, ob Größe der empfangenen Eingangsdaten mit der konfigurierten Größe übereinstimmen 1 Bit 0x0 OK
0x1 konfigurierte Größe zu klein
- [Wrong Length PD OUT] Prüfung, ob Größe der gesendeten Ausgangsdaten mit der vom IO-Link Device erwarteten Größe übereinstimmen 1 Bit 0x0 OK
0x1 konfigurierte Größe zu klein

13674

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserviert								Event 1: Mode		Event 1: Type		Event 1: Src		Event 1: Instance	
Event 1: Code															
reserviert								Event 2: Mode		Event 2: Type		Event 2: Src		Event 2: Instance	
Event 2: Code															
reserviert								Event 3: Mode		Event 3: Type		Event 3: Src		Event 3: Instance	
Event 3: Code															

Legende:

- | | | | | |
|-----------------------|-------------------------------------|--------|---|-------------------------|
| ▪ [Event m: Mode] | Mode: Modus des Ereignisses | 2 Bit | 0x0 | reserviert |
| | | | 0x1 | Einmaliges Ereignis |
| | | | 0x2 | Ereignis verschwunden |
| | | | 0x3 | Ereignis erschienen |
| ▪ [Event m: Type] | Type: Kategorie des Ereignisses | 2 Bit | 0x0 | reserviert |
| | | | 0x1 | Benachrichtigung |
| | | | 0x2 | Warnung |
| | | | 0x3 | Fehler |
| ▪ [Event m: Src] | Source: Quelle des Ereignisses | 1 Bit | 0x0 | IO-Link Device |
| | | | 0x1 | IO-Link Master |
| ▪ [Event m: Instance] | Type: Auslöser des Ereignisses | 2 Bit | 0x0 | Unbekannt |
| | | | 0x1...0x3 | reserviert |
| | | | 0x4 | Application / Anwendung |
| | | | 0x5...0x7 | reserviert |
| ▪ [Event m: Code] | Code: Ereignis-Code; geräteabhängig | 2 Byte | → IODD-Beschreibung des IO-Link Devices | |

Input Data

17312

Register	Inhalt	
	Bits 8-15	Bits 0-7
197	Port X01...X04: Digital Input - Pin 2 / Pin 4 (DI) (→ Mapping: Digitale Eingangsdaten (→ S. 109))	
198	Port X01...X04: Diagnostic Information (→ Mapping: Diagnoseinformationen (→ S. 110))	
199	Port X01...X04: Status Information IO-Link Ports (→ Mapping: Statusinformationen IO-Link Ports (→ S. 111))	
200	Port X01...X04: Compact Input Block (4n Bytes) (→ Mapping: Compact Input Block (→ S. 111))	
297	Port X05...X08: Digital Input - Pin 2 / Pin 4 (DI) (→ Mapping: Digitale Eingangsdaten (→ S. 109))	
298	Port X05...X08: Diagnostic Information (→ Mapping: Diagnoseinformationen (→ S. 110))	
299	Port X05...X08: Status Information IO-Link Ports (→ Mapping: Statusinformationen IO-Link Ports (→ S. 111))	
300	Port X05...X08: Compact Input Block (4n Bytes) (→ Mapping: Compact Input Block (→ S. 111))	

Legende:

- [Digital Input - Pin 2 / Pin 4 (DI)] Digitale Eingangsdaten Pin 2 / Pin 4 (Betriebsart DO) von 4 IO-Link Ports 2 Bytes
- [Diagnostic Informationen] Diagnoseinformationen 2 Bytes
- [Status Information IO-Link Ports] Statusinformationen der IO-Link Ports 2 Bytes
- [Compact Input Block (4n Bytes)] Eingangsdaten (Betriebsart IO-Link) von 4 IO-Link Ports 4n Bytes pro Byte: 0x00...0xFF
n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ **Configuration Area** (→ S. [103](#)))

11098

Register 197:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X04: Klemme 2	X03: Klemme 2	X02: Klemme 2	X01: Klemme 2	res.	res.	res.	res.	X04: Klemme 4	X03: Klemme 4	X02: Klemme 4	X01: Klemme 4

Register 297:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X08: Klemme 2	X07: Klemme 2	X06: Klemme 2	X05: Klemme 2	res.	res.	res.	res.	X08: Klemme 4	X07: Klemme 4	X06: Klemme 4	X05: Klemme 4

Legende:

- | | | | | |
|-----------|---|-------|-----|------|
| ▪ [Pin 4] | Signalpegel an Pin 4 des IO-Link-Ports (DI) | 1 Bit | 0x0 | LOW |
| | | | 0x1 | HIGH |
| ▪ [Pin 2] | Signalpegel an Pin 2 des IO-Link-Ports (wenn genutzt) | 1 Bit | 0x0 | LOW |
| | | | 0x1 | HIGH |

Mapping: Diagnoseinformationen

Register 198:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X04: SC/ OL	X03: SC/ OL	X02: SC/ OL	X01: SC/ OL	res.	res.	res.	res.	res.	res.	SENS PWR	AUX PWR

Register 298:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X08: SC/ OL	X07: SC/ OL	X06: SC/ OL	X05: SC/ OL	res.	res.	res.	res.	res.	res.	SENS PWR	AUX PWR

Legende:

- [SC/OL]

▪ [SENS PWR]

Short Circuit / Overload: zeigt Auftreten eines Kurzschlusses oder einer Überspannung am IO-Link-Port

Sensor Power: zeigt Status der Versorgungsspannung US

1 Bit

1 Bit

0x0 fehlerfrei

0x1 Kurzschluss oder Überspannung

0x0 US nicht verfügbar

0x1 US verfügbar

0x0 UA nicht verfügbar

0x1 UA verfügbar

Mapping: Statusinformationen IO-Link Ports

16455

Register 199:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X04: Data invalid	X03: Data invalid	X02: Data Invalid	X01: Data invalid	res.	res.	res.	res.	X04: Dev Not Conn	X03: Dev Not Conn	X02: Dev Not Conn	X01: Dev Not Conn

Register 299:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	X08: Data invalid	X07: Data invalid	X06: Data Invalid	X05: Data invalid	res.	res.	res.	res.	X08: Dev Not Conn	X07: Dev Not Conn	X06: Dev Not Conn	X05: Dev Not Conn

Legende:

- [Data invalid] zeigt Status der Prozess-Eingangsdaten am IO-Link-Port 1 Bit 0x0 Daten gültig
0x1 Daten ungültig
- [Dev Not Conn] Device Not Connected: zeigt Verbindung des Geräts am IO-Link-Port 1 Bit 0x0 verfügbar
0x1 nicht verfügbar

Mapping: Compact Input Block

34037

IO-Link Port	Registerbereiche				
	2 Bytes/Port (n = 2)	4 Bytes/Port (n = 4)	8 Bytes/Port (n = 8)	16 Bytes/Port (n = 16)	32 Bytes/Port (n = 32)
Port X01	200	200...201	200...203	200...207	200...215
Port X02	201	202...203	204...207	208...215	216...231
Port X03	202	204...205	208...211	216...223	232...247
Port X04	203	206...207	212...215	224...231	248...263
Port X05	300	300...301	300...303	300...307	300...315
Port X06	301	302...303	304...307	308...315	316...331
Port X07	302	304...305	308...311	316...323	332...347
Port X08	303	306...307	312...315	324...331	348...363

Mapping: Digitale Ausgangsdaten

7264

Register 599:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	X04: Klemme 4	X03: Klemme 4	X02: Klemme 4	X01: Klemme 4

Register 699:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	X08: Klemme 4	X07: Klemme 4	X06: Klemme 4	X05: Klemme 4

Legende:

- [Klemme 4] Signalpegel an Klemme 4 des IO-Link-Ports (DO) 1 Bit 0x0 LOW
0x1 HIGH

Mapping: Compact Output Block

52976

IO-Link Port	Registerbereiche				
	2 Bytes/Port (n = 2)	4 Bytes/Port (n = 4)	8 Bytes/Port (n = 8)	16 Bytes/Port (n = 16)	32 Bytes/Port (n = 32)
Port X01	600	600...601	600...603	600...607	600...615
Port X02	601	602...603	604...607	608...615	616...631
Port X03	602	604...605	608...611	616...623	632...647
Port X04	603	606...607	612...615	624...631	648...663
Port X05	700	700...701	700...703	700...707	700...715
Port X06	701	702...703	704...707	708...715	716...731
Port X07	702	704...705	708...711	716...723	732...747
Port X08	703	706...707	712...715	724...731	748...763

Single Port Access

11408

Register	Inhalt	
	Bits 8-15	Bits 0-7
1000	Port X01: Digital Data - Klemme 2	Port X01: Digital Input - Klemme 4 (DI)
1001	Port X01: → Mapping: Statusinformationen (→ S. 115)	Port X01: → Mapping: PQI (→ S. 115)
1002	Port X01: Input Data IO-Link (n Bytes)	
1100	reserviert	Port X01: Digital Output - Klemme 4 (DO)
1101	Port X01: Output Data IO-Link (n Bytes)	
2000	Port X02: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
3000	Port X03: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
4000	Port X04: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
5000	Port X05: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
6000	Port X06: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
7000	Port X07: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	
8000	Port X08: Single Port Access (Mapping: → Port X01 - Register 1000...1101)	

Legende:

- | | | | | |
|-----------------------------------|---|---------|-----------------------|------|
| ▪ [Digital Input - Klemme 2] | Signalpegel Klemme 2 | 1 Byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Digital Input - Klemme 4 (DI)] | Signalpegel Klemme 4 (Betriebsart DI) | 1 Byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Input Data IO-Link (n Bytes)] | Eingangsdaten (Betriebsart IO-Link) (n Bytes)
n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ Configuration Area (→ S. 103)) | n Bytes | pro Byte: 0x00...0xFF | |
| ▪ [Digital Output - Klemme 4(DO)] | Signalpegel Klemme4 (Betriebsart DO) | 1 Byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Output Data IO-Link (n Bytes)] | Ausgangsdaten (Betriebsart IO-Link) (n Bytes)
n = [2,4,8,16,32]; wird bestimmt durch Parameter [Process Data Length] (→ Configuration Area (→ S. 103)) | n Bytes | pro Byte: 0x00...0xFF | |

Mapping: Statusinformationen

11610

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserviert	reserviert	reserviert	reserviert	reserviert	SC / OL	SENS PWR	AUX PWR

Legende:

- [SC / OL] Short Circuit / Overload: zeigt Auftreten eines Kurzschlusses oder einer Überspannung am IO-Link-Port

1 Bit	0x0	fehlerfrei
	0x1	Kurzschluss oder Überspannung
- [SENS PWR] Sensor Power: zeigt Status der Versorgungsspannung US

1 Bit	0x0	kein Fehler
	0x1	Fehler
- [AUX PWR] Auxiliary Power: zeigt Status der Versorgungsspannung UA

1 Bit	0x0	UA liegt an und kein Fehler
	0x1	Fehler

Mapping: PQI

21509

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserviert	Wrong Length PD OUT	Wrong Length PD IN	Wrong Cycle Time	Wrong VID/DID	Invalid Data	Dev Not Conn	IOL Mode

Legende:

- [IOL Mode] Betriebsart des IO-Link Ports

1 Bit	0x0	sonst.
	0x1	IO-Link
- [Dev Not Conn] Verbindung zwischen IO-Link Device und IO-Link Port

1 Bit	0x0	verbunden.
	0x1	nicht verbunden
- [Invalid Data] Status der Prozess-Eingangsdaten am IO-Link-Port

1 Bit	0x0	gültige Daten
	0x1	ungültige Daten
- [Wrong VID/DID] Prüfung, ob aktuelle und konfigurierte Vendor ID und Device ID übereinstimmen

1 Bit	0x0	OK
	0x1	keine Übereinstimmung
- [Wrong Cycle Time] Prüfung, ob aktuelle und konfigurierte Zykluszeit übereinstimmen

1 Bit	0x0	OK
	0x1	keine Übereinstimmung
- [Wrong Length PD IN] Prüfung, ob Größe der empfangene Eingangsdaten mit der konfigurierten Größe übereinstimmen

1 Bit	0x0	OK
	0x1	konfigurierte Größe zu klein
- [Wrong Length PD OUT] Prüfung, ob Größe der gesendeten Ausgangsdaten mit der vom IO-Link Device erwarteten Größe übereinstimmen

1 Bit	0x0	OK
	0x1	konfigurierte Größe zu klein

Acyclic Command Channel

17311

Für die azyklische Übertragung von Daten stehen folgende Modbus-Register zur Verfügung:

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	Command Request Channel (→ Anforderungskanal (Request) (→ S. 116))	
0	Command Response Channel (→ Antwortkanal (Response) (→ S. 117))	

Legende:

- [Command Request Channel] Bereich für die Übertragung von Kommando-Anforderung (Feldbus-SPS >>> IO-Link Master) 44 Bytes
- [Command Response Channel] Bereich für die Übertragung der Kommando-Anwort (IO-Link Master >>> Feldbus-SPS) 44 Bytes

Anforderungskanal (Request)

10893

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	Port No.	
501	Index	
502	Subindex	
503	Command	User ID
504	Data Length (Number of Bytes)	
505	Data (Byte 0)	Data (Byte 1)
...
521	Data (Byte 32)	Data (Byte 33)

Legende:

- [Port No.] Nummer des IO-Link Ports 1 Word 0x0001 Port X01
0x0002 Port X02
...
0x0008 Port X08
- [Index] Index des IO-Link Objekts 1 Word 0x0000...0xFFFF
- [Subindex] Subindex des IO-Link Objekts 1 Word 0x0000...0xFFFF
- [Command] Kommandonummer 1 Byte 0x01 Lesen
0x02 Schreiben
- [User ID] ID zur Identifikation des Kommandos 1 Byte 0x00...0xFF
- [Data Length (Number of Bytes)] Anzahl der Bytes, die relevante Nutzdaten enthalten (wird nur ausgewertet für Command = 0x02) 1 Word 0x0000 0 Bytes
...
0x0022 34 Bytes
- [Data (Byte n)] Nutzdaten n Bytes pro Byte: 0x00 ... 0xFF

Antwortkanal (Response)

6916

Register	Inhalt	
	Bits 8-15	Bits 0-7
0	Port No.	
1	Index	
2	Subindex	
3	Command	User ID
4	Result	
5	Data Length (Number of Bytes)	
6	Data (Byte 0) / Error Code	Data (Byte 1) / Additional Code
...
21	Data (Byte 30)	Data (Byte 31)

Legende:

▪ [Port No.]	Nummer des IO-Link Ports	1 Word	0x0001	Port X01
			0x0002	Port X02
		
			0x0008	Port X08
▪ [Index]	Index des IO-Link Objekts	1 Word	0x0000...0xFFFF	
▪ [Subindex]	Subindex des IO-Link Objekts	1 Word	0x0000...0xFFFF	
▪ [Command]	Kommandonummer	1 Byte	0x01	Lesen
			0x02	Schreiben
▪ [User ID]	reflektierte User ID aus Anforderungskanal	1 Byte	0x00...0xFF	
▪ [Result]	Status der Kommandoabarbeitung	1 Word	0x0000	OK
			0x000F	OK, aber Datenlänge zu groß (nur mit [Command] = 0x02)
			0x00FF	Fehler
		
▪ [Data Length (Number of Bytes)]	Anzahl der Bytes, die relevante Nutzdaten enthalten	1 Word	0x0000	0 Bytes
		
			0x0020	32 Bytes
▪ [Error Code]	Fehlercode	1 Byte	→ Error Codes (→ S. 118)	
▪ [Additional Code]	zusätzliche Fehlercodes	1 Byte	→ Additional Codes (→ S. 118)	
▪ [Data (Byte n)]	Nutzdaten (Byte n)	n Bytes	pro Byte: 0x00...0xFF	

Error Codes

15475

Fehlercode	Bedeutung
0x71	Dienst nicht verfügbar (unbekannte Befehl wurde an den IO-Link Port gesendet)
0x72	Port gesperrt (ein anderer azyklischer Prozess greift auf den IO-Link Port zu)
0x73	Verboten (Zugriffsrechte verbieten Befehlsausführung)
0x74	Ungültige Daten (falscher Parameter wurde im Befehl gesendet)
0x76	Falscher Port (falsche Port-Nummer)
0x77	Falsche Port-Funktion (falsche Port-Funktion oder falscher Parameter wurde an das Gerät gesendet)
0x78	Ungültige Länge (eingestellte Länge ist > 0x20)
0x80	Fehler in der Geräte-Applikation; Additional Code beachten (→ Additional Codes (→ S. 118))

Additional Codes

34072

Code	Name	Beschreibung
0x00	APP_DEV	Fehler in der Geräte-Applikation – keine Details
0x11	IDX_NOTAVAIL	Index nicht verfügbar
0x12	SUBIDX_NOTAVAIL	Subindex nicht verfügbar
0x20	SERV_NOTAVAIL	Dienst vorübergehend nicht verfügbar
0x21	SERV_NOTAVAIL_LOCTRL	Dienst vorübergehend nicht verfügbar – Lokal
0x22	SERV_NOTAVAIL_DEVCTRL	Dienst vorübergehend nicht verfügbar – Device
0x23	IDX_NOT_WRITEABLE	Zugriff verweigert
0x30	PAR_VALOUTOFRNG	Parameterwert außerhalb des gültigen Bereichs
0x31	PAR_VALGTLM	Parameterwert oberhalb des Grenzwerts
0x32	PAR_VALLTLM	Parameterwert unterhalb des Grenzwerts
0x33	VAL_LENVERRUN	Parameterlänge überschritten
0x34	VAL_LENUNDRUN	Parameterlänge unterschritten
0x35	FUNC_NOTAVAIL	Funktion nicht verfügbar
0x36	FUNC_UNAVAILTEMP	Funktion vorübergehend nicht verfügbar
0x40	PAR_SETINVALID	Ungültiger Parametersatz
0x41	PAR_SETINCONSIST	Inkonsistenter Parametersatz
0x82	APP_DEVNOTRDY	Applikation nicht bereit



Additional Error Codes nur gültig, wenn Error Code = 0x80 (→ **Error Codes** (→ S. [118](#)))

14.2.2 Azyklische Kommandos

Inhalt

Kommando 0x10 – Set Mode	120
Kommando 0x20 – Set Validation ID / Data Storage	122
Kommando 0x30 – Set Fail-safe Data Pattern	124
Kommando 0x40 – Reboot	126
Kommando 0x50 – Factory Reset	127

22631

Kommando 0x10 – Set Mode

23461

Der Befehl ändert die Betriebsart eines IO-Link-Ports des AL1940.



Korrespondierender Parameter: [Port Mode] (→ **Mapping: Port-Konfiguration** (→ S. [104](#)))

Kommandoanforderung

12221

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	Port No.	
501	reserviert	
502	reserviert	
503	0x10	User ID
504	reserviert	
505	Target Mode	
506 ... 521	reserviert	

Legende:

- | | | | | |
|-----------------|-------------------------------------|--------|-------------|------------------------|
| ▪ [Port No.] | Nummer des IO-Link Ports | 1 Word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0008 | Port X08 |
| ▪ [User ID] | ID zur Identifikation des Kommandos | 1 Byte | 0x00...0xFF | |
| ▪ [Target Mode] | Betriebsart des IO-Link Ports | 1 Word | 0x0000 | deaktiviert |
| | | | 0x0001 | digitaler Eingang (DI) |
| | | | 0x0002 | digitaler Ausgang (DO) |
| | | | 0x0003 | IO-Link |

Kommando 0x20 – Set Validation ID / Data Storage

23462

Der Befehl setzt das Verhalten des IO-Link Masters beim Anschluss eines neuen IO-Link Devices an einem IO-Link-Ports des Geräts.



Korrespondierender Parameter: [Validation ID] (→ **Mapping: Port-Konfiguration** (→ S. [104](#)))

Kommandoanforderung

14272

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	Port No.	
501	reserviert	
502	reserviert	
503	0x20	User ID
504	reserviert	
505	Validation ID	
506 ... 521	reserviert	

Legende:

- | | | | | |
|-------------------|--|--------|-------------|------------------------------|
| ▪ [Port No.] | Nummer des IO-Link Ports | 1 Word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0008 | Port X08 |
| ▪ [User ID] | ID zur Identifikation des Kommandos | 1 Byte | 0x00...0xFF | |
| ▪ [Validation ID] | Unterstützter IO-Link-Standard und Verhalten des IO-Link Masters bei Anschluss neuer IO-Link Devices am IO-Link-Port | 1 Word | 0x0000 | keine Validierung |
| | | | 0x0001 | V1.0 device |
| | | | 0x0002 | V1.1 device |
| | | | 0x0003 | V1.1 device Backup + Restore |
| | | | 0x0004 | V1.1 device Backup |

Kommandoantwort

10919

Register	Inhalt	
	Bits 8-15	Bits 0-7
0	Port No.	
1	reserviert	
2	reserviert	
3	0x10	User ID
4	Result	
5	Data Length (Number of Bytes)	
6	reserviert / Error Code	Validation ID / Additional Code
7 ... 21	reserviert	

Legende:

▪ [Port No.]	Nummer des IO-Link Ports	1 Word	0x0001	Port X01
			0x0002	Port X02
		
			0x0008	Port X08
▪ [User ID]	reflektierte User ID aus Anforderungskanal	1 Byte	0x00...0xFF	
▪ [Result]	Status der Kommandoabarbeitung	1 Byte	0x00	OK
			0xFF	Fehler
▪ [Data Length (Number of Bytes)]	Anzahl der Bytes, die relevante Nutzdaten enthalten	1 Word	0x0001	1 Byte
			0x0002	2 Bytes
▪ [Validation ID]	Unterstützter IO-Link-Standard und Verhalten des IO-Link Masters bei Anschluss neuer IO-Link Devices am IO-Link-Port	1 Byte	0x00	keine Validierung
			0x01	V1.0 device
			0x02	V1.1 device
			0x03	V1.1 device Backup + Restore
			0x04	V1.1 device Backup
▪ [Error Code]	Fehlercode	1 Byte	→ Error Codes (→ S. 118)	
▪ [Additional Code]	zusätzliche Fehlercodes	1 Byte	→ Additional Codes (→ S. 118)	

Kommando 0x30 – Set Fail-safe Data Pattern

23464

Der Befehl setzt das Verhalten des Ausgänge bei einer Unterbrechung der Modbus TCP-Verbindung und die entsprechenden Rückfallwerte.



Korrespondierender Parameter: [Fail-safe Mode] (→ **Mapping: Port-Konfiguration** (→ S. 104))

Die Anzahl der benötigten Rückfallwerte ergibt sich aus der Größe der Ausgangsdaten (→ **Configuration Area** (→ S. [103](#))).

Kommandoanforderung

11016

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	Port No.	
501	reserviert	
502	reserviert	
503	0x30	User ID
504	Byte Length N	
505	Failsafe Mode	
506	reserviert / Failsafe Data (Byte 1)	reserviert / Failsafe Data (Byte 0)
...
521	reserviert / Failsafe Data (Byte 31)	reserviert / Failsafe data (Byte 30)

Legende:

- | | | | | |
|--------------------------------------|---|--------|---------------|------------------------|
| ▪ [Port No.] | Nummer des IO-Link Ports | 1 Word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0008 | Port X08 |
| ▪ [User ID] | ID zur Identifikation des Kommandos | 1 Byte | 0x00...0xFF | |
| ▪ [Data Length
(Number of Bytes)] | Anzahl der Bytes, die relevante Nutzdaten
enthalten (wird nur ausgewertet für
Command = 0x02) | 1 Word | 0x0002 | 2 Bytes |
| | | | ... | ... |
| | | | 0x0022 | 34 Bytes |
| ▪ [Failsafe Mode] | Rückfall-Modus für die Ausgänge der IO-Link
Ports bei einer Unterbrechung der Modbus
TCP-Verbindung | 1 Word | 0x0000 | No Failsafe |
| | | | 0x0001 | Failsafe: Reset Value |
| | | | 0x0002 | Failsafe: Old Value |
| | | | 0x0003 | Failsafe: with Pattern |
| ▪ [Failsafe Data
(Byte n)] | Rückfallwerte für die Ausgänge (nur bei Failsafe
Mode = 0x0003) | 1 Byte | 0x00 ... 0xFF | |

Kommandoantwort

10990

Register	Inhalt	
	Bits 8-15	Bits 0-7
0	Port No.	
1	reserviert	
2	reserviert	
3	0x30	User ID
4	Result	
5	Data Length (Number of Bytes)	
6	reserviert / Error Code	Failsafe Mode / Additional Code
7 ... 21	reserviert	

Legende:

- | Parameter | Beschreibung | Größe | Werte |
|------------------------------------|---|--------|---|
| [Port No.] | Nummer des IO-Link Ports | 1 Word | 0x0001 Port X01 |
| | | | 0x0002 Port X02 |
| | | | ... |
| | | | 0x0008 Port X08 |
| [User ID] | reflektierte User ID aus Anforderungskanal | 1 Byte | 0x00...0xFF |
| [Result] | Status der Kommandoabarbeitung | 1 Word | 0x0000 OK |
| | | | 0x00FF Fehler |
| [Data Length
(Number of Bytes)] | Anzahl der Bytes, die relevante Nutzdaten
enthalten | 1 Word | 0x0001 1 Byte |
| | | | 0x0002 2 Bytes |
| [Failsafe Mode] | Rückfall-Modus für die Ausgänge der IO-Link
Ports bei einer Unterbrechung der Modbus
TCP-Verbindung | 1 Byte | 0x00 No Failsafe |
| | | | 0x01 Failsafe: Reset Value |
| | | | 0x02 Failsafe: Old Value |
| | | | 0x03 Failsafe: with Pattern |
| [Error Code] | Fehlercode | 1 Byte | → Error Codes (→ S. 118) |
| [Additional Code] | zusätzlicher Fehlercode | 1 Byte | → Additional Codes (→ S. 118) |

Kommando 0x40 – Reboot

7639

Der Befehl startet den AL1940 neu.

Kommandoanforderung

21515

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	reserviert	
501	reserviert	
502	reserviert	
503	0x40	User ID
504	reserviert	
505	0x00AA	
506 ... 521	reserviert	

Legende:

- [User ID] ID zur Identifikation des Kommandos 1 Byte 0x00...0xFF

Kommandoantwort

25156

Register	Inhalt	
	Bits 8-15	Bits 0-7
0	reserviert	
1	reserviert	
2	reserviert	
3	0x40	User ID
4	Result	
5	Data Length (Number of Bytes)	
6	reserviert / Error Code	0xAA / Additional Code
7 ... 21	reserviert	

Legende:

- [User ID] reflektierte User ID aus Anforderungskanal 1 Byte 0x00...0xFF
- [Result] Status der Kommandoabarbeitung 1 Word 0x0000 OK
0x00FF Fehler
- [Data Length (Number of Bytes)] Anzahl der Bytes, die relevante Nutzdaten enthalten 1 Word 0x0001 1 Byte
0x0002 2 Bytes
- [Error Code] Fehlercode 1 Byte → **Error Codes** (→ S. [118](#))
- [Additional Code] zusätzlicher Fehlercode 1 Byte → **Additional Codes** (→ S. [118](#))

Kommando 0x50 – Factory Reset

7254

Der Befehl setzt alle Parameter auf die Werkseinstellungen (→ **Werkseinstellungen** (→ S. [94](#))).

Kommandoanforderung

11060

Register	Inhalt	
	Bits 8-15	Bits 0-7
500	reserviert	
501	reserviert	
502	reserviert	
503	0x50	User ID
504	reserviert	
505	0x0055	
506 ... 520	reserviert	

Legende:

- [User ID] ID zur Identifikation des Kommandos 1 Byte 0x00...0xFF

Kommandoantwort

21514

Register	Inhalt	
	Bits 8-15	Bits 0-7
0	reserviert	
1	reserviert	
2	reserviert	
3	0x50	User ID
4	Result	
5	Data Length (Number of Bytes)	
6	reserviert / Error Code	0x55 / Additional Code
7 ... 21	reserviert	

Legende:

- [User ID] reflektierte User ID aus Anforderungskanal 1 Byte 0x00...0xFF
- [Result] Status der Kommandoabarbeitung 1 Word 0x0000 OK
0x00FF Fehler
- [Data Length (Number of Bytes)] Anzahl der Bytes, die relevante Nutzdaten enthalten 1 Word 0x0001 1 Byte
0x0002 2 Bytes
- [Error Code] Fehlercode 1 Byte → **Error Codes** (→ S. [118](#))
- [Additional Code] zusätzlicher Fehlercode 1 Byte → **Additional Codes** (→ S. [118](#))

14.3 ifm-IoT-Core

Inhalt

Übersicht: IoT-Profile	129
Übersicht: IoT-Typen	136
Übersicht: IoT-Dienste	137

8988

14.3.1 Übersicht: IoT-Profile

Inhalt

Profil: blob	129
Profil: deviceinfo	130
Profil: devicetag	130
Profil: iolinkdevice_full	131
Profil: iolinkmaster	131
Profil: mqttCmdChannel	132
Profil: mqttCmdChannelSetup	132
Profil: mqttConnection	132
Profil: mqttSetup	133
Profil: network	133
Profil: parameter	134
Profil: processdata	134
Profil: runcontrol	134
Profil: service	134
Profil: software	134
Profil: software/uploadedablessoftware	135
Profil: timer	135

17711

Profil: blob

9584

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

Profil: mqttCmdChannel

60217

Element (identifier)	Eigenschaften	Pflicht	Kommentar
mqttCmdChannel	<ul style="list-style-type: none"> type = structure profile = commChannel 		Profil des MQTT-Kommandokanals
../type	<ul style="list-style-type: none"> type = data data type = STRING 	obligatorisch	Protokolltyp der Schnittstelle
../status	<ul style="list-style-type: none"> type = data data type = STRING 	obligatorisch	Zustand des MQTT-Kommandokanals (Mögliche Werte: init, running, stopped, error)
../mqttCmdChannelSetup	type = profile		Unterprofil: Profil: mqttCmdChannelSetup (→ S. 132)

Profil: mqttCmdChannelSetup

60220

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

Profil: mqttConnection

60216

Element (identifier)	Eigenschaften	Pflicht	Kommentar
mqttConnection	<ul style="list-style-type: none"> type = structure profile = commInterface 		MQTT-Verbindung im IoT Core
../type	<ul style="list-style-type: none"> type = data data type = STRING 	obligatorisch	Protokolltyp der Schnittstelle
../status	<ul style="list-style-type: none"> type = data data type = STRING 	obligatorisch	globaler Zustand des MQTT (Mögliche Werte: init, running, stopped, error)
../mqttSetup	type = profile		Unterprofil: Profil: mqttSetup (→ S. 133)
../mqttCmdChannel	type = profile		Unterprofil: Profil: mqttCmdChannel (→ S. 132)

Profil: mqttSetup

60218

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

Profil: network

11179

Element (identifier)	Eigenschaften	Pflicht	Kommentar
network	<ul style="list-style-type: none"> type = structure profiles = deviceinfo 		kennzeichnet Element als Geräteinformation
../macaddress	<ul style="list-style-type: none"> type = data profile = parameter 	obligatorisch	
../ipaddress	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../ipv6address	<ul style="list-style-type: none"> type = data profile = parameter 	obligatorisch	
../subnetmask	<ul style="list-style-type: none"> type = data profile = parameter 	obligatorisch	
../ipdefaultgateway	<ul style="list-style-type: none"> type = data profile = parameter 	obligatorisch	
../dhcp	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../ipversion	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../hostname	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../autonegotiation	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../portspeed	<ul style="list-style-type: none"> type = data profile = parameter 	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"> type = profile profile = runcontrol 		Steuerung des MQTT-Kommandokanals
../start	type = service	obligatorisch	Dienst: start (→ S. 147)
../stop	type = service	obligatorisch	Dienst: stop (→ S. 148)
../reset	type = service	obligatorisch	Dienst: reset (→ S. 145)

Profil: service

16575

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

Profil: software

10999

Element (identifier)	Eigenschaften	Pflicht	Kommentar
software	<ul style="list-style-type: none"> type = structure profile = software 		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"> type = structure profiles = software/uploadablessoftware 		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"> type = data profile = blob 	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"> type = structure profile = timer 		
../counter	<ul style="list-style-type: none"> type = data profile = parameter 	obligatorisch	
../interval	<ul style="list-style-type: none"> type = data profile = parameter 	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.....	137
Dienst: getblobdata.....	138
Dienst: getdata	138
Dienst: getdatamulti.....	139
Dienst: getelementinfo.....	139
Dienst: getidentity	140
Dienst: getsubscriberlist	141
Dienst: getsubscriptioninfo	142
Dienst: gettree	143
Dienst: install	144
Dienst: iolreadacyclic.....	144
Dienst: iolwriteacyclic	144
Dienst: querytree	145
Dienst: reboot	145
Dienst: reset	145
Dienst: setblock	146
Dienst: setdata.....	147
Dienst: signal	147
Dienst: start	147
Dienst: start_stream_set.....	148
Dienst: stop.....	148
Dienst: stream_set.....	148
Dienst: subscribe	149
Dienst: unsubscribe	150
Dienst: validation_useconnecteddevice	150

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 AL1940 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		AL1940
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"> keine Angabe: alle Ebenen werden angezeigt 0: keine Unterelemente anzeigen ("subs") 1: Unterelemente anzeigen 2: Unterelemente bis zur 2. Ebene anzeigen 3: Unterelemente bis zur 3. Ebene anzeigen ... 20: Unterelemente bis zur 20. Ebene anzeigen

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"> lifetime: Wert wird mit IoT Core gespeichert; Wert bleibt gültig auch nach Neuart des Geräts uptime: Wert wird bis zum nächsten Neustart des Geräts gespeichert

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 AL1940 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"> JSON: http://ipaddress:port/path JSON: ws:///path JSON: mqtt://ipadress:port/topic CSV: tcp://ipaddress:port/path
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"> json: JSON-formatiert csv: CSV mit Standard-Separator (,) csv0: CSV-formatiert mit Komma-Separator (,) csv1: CSV-formatiert mit Semikolon-Separator (;)
duration	obligatorisch	STRING	Dauer der Speicherung des Werts <ul style="list-style-type: none"> lifetime: Wert wird mit IoT Core gespeichert; Wert bleibt gültig auch nach Neustart des Geräts uptime: Wert wird bis zum nächsten Neustart des Geräts gespeichert once: nur eine Benachrichtigung schicken, Benutzer muss Abonnement direkt wieder abmelden

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 stucture

Anfragedaten (data): keine

Rückgabedaten (data): keine

15 Index

A

Acyclic Command Channel	116
Additional Codes	118
AL1940 in Modbus-Projekt einbinden	81
Allgemein	7
Allgemeine Funktionen	43
Änderungshistorie	6
Anforderungskanal (Request)	116
Anhang	96
Antwortkanal (Response)	117
Auf den ifm-IoT-Core zugreifen	39
Ausgänge	98
Ausgangsdaten einzelner IO-Link Ports schreiben	87
Ausgangsdaten mehrerer IO-Link Ports schreiben	86
Azyklische Dienste nutzen	91
Azyklische Kommandos	119

B

Bedien- und Anzeigeelemente	19
Beispiel	
Abonnement ändern	66
Abonnements prüfen	67
Anfrage mit Authentifikation	51
Benachrichtigungen abmelden	67
Benachrichtigungen abonnieren	47, 65
Benachrichtigungen im CSV-Format abonnieren	66
Bezeichnung des IO-Link Master ändern	63
Datenspeicher eines IO-Link Ports klonen	53
Digitalen Ausgangswert schreiben (Betriebsart)	56
Digitalen Eingangswert lesen (Betriebsart)	57
DNS-Unterstützung nutzen	46
Eigenschaften eines Elements lesen	44
Firmware aktualisieren	62
Gerätebeschreibung durchsuchen	46
GET-Request	39
IO-Link Master in eine CODESYS-Projekt einbinden	81
IO-Link Prozesswert lesen (Betriebsart)	55
IO-Link Prozesswert schreiben (Betriebsart)	56
Mehrere Parameterwerte des IO-Link Master gleichzeitig lesen	45
MQTT-Kommandokanal konfigurieren	72
Parameterwert eines IO-Link Devices ändern	59
Parameterwert eines IO-Link Devices lesen	58
Passwort rücksetzen	51
POST-Request	41
Sicherheitsmodus aktivieren	50
Teilbaum ausgeben	44
Temperatur an MQTT-Broker veröffentlichen	73
WebSockets nutzen	68
Benachrichtigung löschen	76
Benachrichtigungen abonnieren	64
Benachrichtigungen verwalten	75
Bestimmungsgemäße Verwendung	9
Betrieb	92

C

Configuration Area	103
--------------------------	-----

D

Diagnoseinformationen und Events lesen	87
Diagnostic Data	105

Dienst

factoryreset	137
getblobdata	138
getdata	138
getdatamulti	139
getelementinfo	139
getidentity	140
getsubscriberlist	141
getsubscriptioninfo	142
gettree	143
install	144
iolreadacyclic	144
iolwriteacyclic	144
querytree	145
reboot	145
reset	145
setblock	146
setdata	147
signal	147
start	147
start_stream_set	148
stop	148
stream_set	148
subscribe	149
unsubscribe	150
validation_useconnecteddevice	150
Digitale Eingänge	12
DNS-Unterstützung	46

E

Eingänge	98
Eingänge / Ausgänge	97
Eingangsdaten einzelner IO-Link Ports lesen	85
Eingangsdaten mehrerer IO-Link Ports lesen	84
Einsatzbereich	97
Elektrische Daten	97
Elektrischer Anschluss	14, 100
Elemente im Gerätebaum suchen	77
Error Codes	118
Erste Schritte	43
Ethernet-Ports	20

F

Fieldbus	
IP-Einstellungen konfigurieren	30, 52
Länge der Prozessdaten einstellen	31
Firmware	
Gerät auf Werkseinstellungen rücksetzen	35
Gerät neu starten	35
Firmware aktualisieren	80, 93
Funktion	10

G

Gateway	
Anwendungskennung einstellen	63
Firmware aktualisieren	62
Gerät rücksetzen, neu starten und lokalisieren	60
Geräteinformationen lesen	61
Zustands- und Diagnoseinformationen lesen	61
Gerät anschließen	18
Gerät montieren	13
Geräteinformationen lesen	89
GET-Request	39

H

Hinweis	
Exception Codes	91
Sicherheitsmodus	50
Verbindungszustände	71
Hinweise	14, 25
Hinweise für Programmierer	38

I

ifm-IoT-Core	37, 128
Inbetriebnahme	22
Info	
Geräteinformationen zeigen	34
Input Data	108
Internet of Things (IoT)	11
IO-Link	11
IO-Link Device tauschen	93
IO-Link Devices	
Auf Parameter zugreifen	58
Geräteinformationen lesen und schreiben	60
IO-Link Events anzeigen	60
IO-Link Devices für Class-A-Betrieb anschließen	16
IO-Link Devices für Class-B-Betrieb anschließen	17
IO-Link Devices konfigurieren	35, 89
IO-Link Master konfigurieren	78, 82
IO-Link Master steuern	89
IO-Link Ports	16
Betriebsart Pin 4 (US) einstellen	52
Datenübertragung zu LR AGENT oder LR SMARTOBSERVER konfigurieren	55
Gerätevalidierung und Datenspeicherung konfigurieren	53
Port-Events anzeigen	58
Prozessdaten lesen / schreiben	55
IO-Link Ports (Class A)	21
IO-Link Ports konfigurieren	83
IO-Link-Ports	
Betriebsart konfigurieren	32
Datenübertragung zu LR AGENT oder LR SMARTOBSERVER einstellen	31
Gerätevalidierung und Datenspeicherung einstellen	33
Rückfallwerte einstellen	34
IO-Link-Versorgung	12
IoT	
IP-Einstellungen konfigurieren	26, 48
Schnittstelle zu LR AGENT und LR SMARTOBSERVER konfigurieren	49
Schnittstelle zum LR AGENT oder LR SMARTOBSERVER konfigurieren	29
Sicherheitsmodus konfigurieren	27, 49
Zugriffsrechte konfigurieren	28, 48
IoT Core	
Allgemeine Informationen	38
IoT-Core	
Diagnosecodes	42
IoT-Core-Visualizer nutzen	74
IoT-Port	15, 21
IT-Sicherheit	8

K

Kommando 0x10 – Set Mode	120
Kommando 0x20 – Set Validation ID / Data Storage	122
Kommando 0x30 – Set Fail-safe Data Pattern	124
Kommando 0x40 – Reboot	126
Kommando 0x50 – Factory Reset	127
Kommandoanforderung	120, 122, 124, 126, 127

Kommandoantwort	121, 123, 125, 126, 127
Kommunikation, Parametrierung, Auswertung	11

L

LED-Anzeigen	20
LR DEVICE	24

M

Mapping	
Compact Input Block	111
Compact Output Block	113
Diagnoseinformationen	110
Diagnostics	106
Digitale Ausgangsdaten	113
Digitale Eingangsdaten	109
Events	107
Port-Konfiguration	104
PQI	115
Statusinformationen	115
Statusinformationen IO-Link Ports	111
Mechanische Daten	99
Modbus TCP	11, 81, 101
Hinweise für Programmierer	90
Modbus TCP-Ports verbinden	14
Montage	13
MQTT-Kommandokanal konfigurieren	70
MQTT-Unterstützung	70

N

Neue Benachrichtigung erstellen	75
Notwendige Vorkenntnisse	7

O

Offline-Parametrierung	25
Optische Signalisierung	12
Output Data	112

P

Parametrierung	12, 23
Parametrierung mit LR DEVICE	25
POST-Request	40
Profil	
blob	129
deviceinfo	130
devicetag	130
iolinkdevice_full	131
iolinkmaster	131
mqttCmdChannel	132
mqttCmdChannelSetup	132
mqttConnection	132
mqttSetup	133
network	133
parameter	134
processdata	134
runcontrol	134
service	134
software	134
software/uploadedablessoftware	135
timer	135
Prozessdaten lesen und schreiben	79

R

Rechtliche Hinweise	5
---------------------------	---

Regeln für den Zugriff auf Modbus-Register	90
Register.....	102
Reinigung.....	93

S

Schnittstellen	98
Sicherheitshinweise	7
Sicherheitsmodus	11
Sicherheitssymbole auf dem Gerät	7
Single Port Access.....	114
Spannungsversorgung	21
Speicherdauer einstellen	47
Status-LEDs.....	20

T

Technische Daten.....	97
-----------------------	----

U

Überblick.....	19
Übersicht	
IoT-Dienste	137
IoT-Profil	129
IoT-Typen	136
Umgebungsbedingungen	99
Unterstützte Function Codes	90

V

Vorbemerkung	5
--------------------	---

W

Wartung, Instandsetzung und Entsorgung.....	93
Web Socket nutzen.....	68
Web-based Management nutzen	92
Werkseinstellungen	94

Z

Zeichenerklärung.....	6
Zubehör.....	95
Zulassungen / Prüfungen	99
Zweck des Dokuments	5