



Operating Instructions

IO-Link Master with Modbus TCP Interface

DataLine

4 Ports

IP 65 / IP 66 / IP 67 / IP 69K

AL1341

HW Revision: AB

Firmware: 2.3.x

LR DEVICE: 1.5.0.x

English

Contents

1	Preliminary note	5
1.1	Legal and copyright information	5
1.2	Purpose of the document	5
1.3	Explanation of Symbols	6
1.4	Modification history	6
2	Safety instructions	7
2.1	General	7
2.2	Required background knowledge	7
2.3	Safety symbols on the device	7
2.4	IT safety	8
2.5	Tampering with the unit	8
3	Intended use	9
3.1	Permitted use	9
3.2	Prohibited use	9
4	Function	10
4.1	Communication, parameter setting, evaluation	11
4.1.1	IO-Link	11
4.1.2	Modbus TCP	11
4.1.3	Internet of Things (IoT)	11
4.1.4	Security mode	11
4.1.5	Parameter setting	12
4.1.6	Visual indication	12
4.2	Digital inputs	12
4.3	IO-Link supply	12
5	Mounting	13
5.1	Mount the device	13
6	Electrical connection	14
6.1	Notes	14
6.2	Modbus TCP ports	15
6.3	IoT port	15
6.4	IO-Link ports	16
6.4.1	Connect IO-Link devices for Class A operation	16
6.5	Connect the device	16
7	Operating and display elements	18
7.1	Overview	18
7.2	LED indicators	19
7.2.1	Status LEDs	19
7.2.2	Ethernet interface	19
7.2.3	IoT port	20
7.2.4	Voltage supply	20
7.2.5	IO-Link ports (Class A)	20

8	Set-up	21
8.1	Read device and diagnostic information	22
9	Configuration	23
9.1	LR DEVICE	24
9.1.1	Remarks	25
9.1.2	IoT: Configure IP settings	26
9.1.3	IoT: Configure security mode	27
9.1.4	IoT: Configure access rights	28
9.1.5	IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER	29
9.1.6	Fieldbus: Configure IP settings	30
9.1.7	Fieldbus: set the length of the process data	31
9.1.8	IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER	31
9.1.9	IO-Link ports: Configure operating mode	32
9.1.10	IO-Link ports: Set the device validation and data storage	33
9.1.11	IO-Link ports: set fail-safe values	34
9.1.12	Info: Show device information	34
9.1.13	Firmware: Reset device to factory settings	35
9.1.14	Firmware: Reboot the device	35
9.1.15	Configure IO-Link devices	36
9.2	ifm IoT Core	37
9.2.1	First steps	37
9.2.2	General functions	38
9.2.3	Configure IoT interface	39
9.2.4	IoT interface: Configure security mode	40
9.2.5	Configure the fieldbus interface	43
9.2.6	Configure IO-Link ports	44
9.2.7	Configure IO-Link devices	46
9.2.8	Set application identification	48
9.2.9	Read / write cyclic process data	49
9.2.10	Control IO-Link master	50
9.2.11	Read diagnostic data of the AL1341	52
9.2.12	Read device information of the IO-Link master	53
9.2.13	Read information about IO-Link devices	53
9.2.14	Subscribe to events	54
9.2.15	MQTT support	56
9.2.16	Programmers' notes	57
9.3	Modbus TCP	61
9.3.1	Integrate the AL1341 into the Modbus project	61
9.3.2	Set IO-Link master	63
9.3.3	Set IO-Link ports	64
9.3.4	Read input data of several IO-Link ports	65
9.3.5	Read input data of individual IO-Link ports	66
9.3.6	Write output data of several IO-Link ports	67
9.3.7	Write output data of individual IO-Link ports	68
9.3.8	Read diagnostic information and events	69
9.3.9	Read device information	70
9.3.10	Control IO-Link master	70
9.3.11	Configure IO-Link devices	70
9.3.12	Modbus TCP: Programmers' notes	71

10	Maintenance, repair and disposal	74
10.1	Cleaning process	74
10.2	Update firmware	75
10.3	Replace IO-Link device	76
11	Factory settings	77
12	Accessories	78
13	Appendix	79
13.1	Technical data	80
13.1.1	Application	80
13.1.2	Electrical data	80
13.1.3	Inputs / outputs	80
13.1.4	Inputs	81
13.1.5	Outputs	81
13.1.6	Interfaces	81
13.1.7	Operating conditions	82
13.1.8	Approvals / tests	82
13.1.9	Mechanical data	82
13.1.10	Electrical connection	83
13.2	Modbus TCP	84
13.2.1	Register	85
13.2.2	Acyclic commands	100
13.3	ifm IoT Core	109
13.3.1	Overview: IoT profile	110
13.3.2	Overview: IoT types	115
13.3.3	Overview: IoT services	116
14	Index	125

1 Preliminary note

Content

Legal and copyright information	5
Purpose of the document	5
Explanation of Symbols	6
Modification history	6

33203

1.1 Legal and copyright information

33117

© All rights reserved by ifm electronic gmbh. No part of this manual may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used on our pages are the property of the respective rights owners:

- AS-i is the property of the AS-International Association, (→ www.as-interface.net)
- CAN is the property of the CiA (CAN in Automation e.V.), Germany (→ www.can-cia.org)
- CODESYS™ is the property of the 3S – Smart Software Solutions GmbH, Germany (→ www.codesys.com)
- DeviceNet™ is the property of the ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® is the property of the → ODVA™
- EtherCAT® is a registered trade mark and patented technology, licensed by Beckhoff Automation GmbH, Germany
- IO-Link® is the property of the → PROFIBUS Nutzerorganisation e.V., Germany (→ www.io-link.com)
- ISOBUS is the property of the AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ www.aef-online.org)
- Microsoft® is the property of the Microsoft Corporation, USA (→ www.microsoft.com)
- Modbus® is the property of the Schneider Electric SE, France (→ www.schneider-electric.com)
- PROFIBUS® is the property of the PROFIBUS Nutzerorganisation e.V., Germany (→ www.profibus.com)
- PROFINET® is the property of the → PROFIBUS Nutzerorganisation e.V., Germany
- Windows® is the property of the → Microsoft Corporation, USA

1.2 Purpose of the document

34227

This document is only for device types "IO-Link master - Modbus TCP gateway (DataLine) 4 port IP 65 / IP 66 / IP 67 / IP 69K" (art. no.: AL1341).

It is part of the device and contains information about the correct handling of the product.

- Read this document before using the device.
- Keep this document during the service life of the device.

1.3 Explanation of Symbols

34171



WARNING!

Death or serious irreversible injuries may result.



CAUTION!

Slight reversible injuries may result.



NOTICE!

Property damage is to be expected or may result.



Important note

Non-compliance can result in malfunction or interference



Information

Supplementary note



Request for action



Reaction, result



"see"

abc

Cross-reference

123

Decimal number

0x123

Hexadecimal number

0b010

Binary number

[...]

Designation of pushbuttons, buttons or indications

1.4 Modification history

34492

Version	Topic	Date
00	New creation of document	04 / 2019
01	Corrected: Technical data - Max. current load per output	09 / 2019

2 Safety instructions

Content

General	7
Required background knowledge	7
Safety symbols on the device	7
IT safety	8
Tampering with the unit	8

28333

2.1 General

33834



The plant manufacturer is responsible for the safety of the plant in which the device is installed.

If the device is used in a way that is not intended by the manufacturer, the protection supported by the device may be impaired.

Non-observance of the instructions, operation which is not in accordance with use as prescribed below, wrong installation or incorrect handling can affect the safety of operators and machinery.

- ▶ Observe these operating instructions.
- ▶ Adhere to the warning notes on the product.

2.2 Required background knowledge

34185

This document is intended for specialists. Specialists are people who, based on their relevant training and experience, are capable of identifying risks and avoiding potential hazards that may be caused during operation or maintenance of the product.

The document contains information about the correct handling of the product.

2.3 Safety symbols on the device

34199



General warning

Observe instructions in chapter "Electrical connection" (→ **Electrical connection** (→ S. [14](#)))!

2.4 IT safety

54678

NOTICE!

If the device is operated in an unprotected network environment.

- > Unauthorised read or write access to data is possible.
- > Unauthorised manipulation of the device function is possible.
- ▶ Check and restrict access options to the device:
 - Restrict access to authorised persons.
 - Do not connect the device to open networks or the internet.

If access from the internet is inevitable:

- ▶ choose a safe method to connect with the device (e. g. VPN).
- ▶ Use encrypted data transmission (e. g. https / TLS).

2.5 Tampering with the unit

33190



WARNING!

Tampering with the unit.

- > In case of non-compliance:
 - Possible affects on safety of operators and machinery
 - Expiration of liability and warranty
- ▶ Do not open the devices!
- ▶ Do not insert any objects into the devices!
- ▶ Prevent metal foreign bodies from penetrating!

3 Intended use

Content

Permitted use	9
Prohibited use.....	9

34079

3.1 Permitted use

34211

The IO-Link master serves as a gateway between intelligent IO-Link devices and the Modbus TCP network. The device is designed for use without a control cabinet in the food industry.

3.2 Prohibited use

34228

The device may not be used beyond the limits of the technical data (→ **Technical data** (→ S. [80](#)))!

4 **Function**

Content	
Communication, parameter setting, evaluation	11
Digital inputs	12
IO-Link supply.....	12

33836

4.1 Communication, parameter setting, evaluation

Content

IO-Link	11
Modbus TCP	11
Internet of Things (IoT)	11
Security mode	11
Parameter setting	12
Visual indication	12

33860

4.1.1 IO-Link

34084

The device offers the following IO-Link functions:

- IO-Link master (IO-Link revision 1.0 and 1.1)
- 4 IO-Link ports for connection of IO-Link devices
- Provision of process data of the connected IO-Link devices for LR SMARTOBSERVER monitoring software (→ www.ifm.com)

4.1.2 Modbus TCP

33676

The device offers the following Modbus TCP functions:

- Provision of the functions of a Modbus TCP Slave
- 2 port switch for access to the Modbus TCP interface (X21/X22)
- Gateway for transmission of the process and parameter data between the connected IO-Link devices and the higher-level Modbus TCP controller

4.1.3 Internet of Things (IoT)

54679

The device offers the following IoT functions:

- Gateway for the transmission of process, parameter and monitoring data between IO-Linkmaster / IO-Link devices and the IT network level
- REST-API to access process and parameter data
- Supported protocols: TCP/IP JSON, MQTT

4.1.4 Security mode

54697

The IoT interface offers the following optional security functions:

- Secure data transfer via encrypted connection (Secure Layer Transport - TLS)
- Access protection via authentication

4.1.5 Parameter setting

34210

The device provides the following configuration options:

- Parameter setting of the IO-Link master of the AL1341 with LR DEVICE parameter setting software, Modbus TCP projection software or ifm IoT-Core services.
- Parameter setting of the connected IO-Link devices (sensors, actuators) with LR DEVICE parameter setting software, Modbus TCP projection software or ifm IoT-Core services
- Storage of parameter sets of the connected IO-Link devices for automatic recovery (data storage)

4.1.6 Visual indication

34192

The device has the following visual indicators:

- Status and error indication of the gateway, of the Modbus TCP connection and of the system
- Status display of the voltage supply
- Status and activity display of the Ethernet connection
- Status, error and short circuit/overload indication of the IO-Link ports

4.2 Digital inputs

33817

The device has 4 additional digital inputs (type 2 according to EN 61131-2).

The digital inputs are on pin 2 of the IO-Link ports X01...X04.

All inputs refer to the potential of the device supply (pin 3).

4.3 IO-Link supply

34077

The device has 4 supplies for IO-Link devices.

The IO-Link ports X01...X04 are ports class A.

Every supply provides short circuit monitoring.

The device ensures fire protection for the connected IO-Link devices by providing a power-restricted circuit at the IO-Link ports (according to IEC61010-1 and Class 2 according to UL1310).

5 Mounting

Content

Mount the device	13
------------------------	----

34058

5.1 Mount the device

34059



- ▶ Disconnect the system from power before installation.
 - ▶ For installation choose a flat mounting surface.
 - ▶ Please observe the maximum tightening torque.
-
- ▶ Fix the unit to the mounting surface using 2 M5 mounting screws and washers.
 - Tightening torque: 1.8 Nm
 - ▶ Ground the unit via the two mounting screws of the upper mounting lugs.

6 Electrical connection

Content

Notes	14
Modbus TCP ports.....	15
IoT port	15
IO-Link ports	16
Connect the device	16

33805

6.1 Notes

51957



A qualified electrician must connect the unit.

- The national and international regulations setting up electrical equipment must be complied with.

The unit is only suitable for operation using SELV/PELV voltages.

- Please note the information concerning IO-Link wiring!

This unit contains components that may be damaged or destroyed by electrostatic discharge (ESD).

- Please observe the required precautions against electrostatic discharge!

The IP rating of the overall system depends on the protection ratings of the individual devices, the applied connection elements and the corresponding protective caps.

- Provide cables with a strain relief depending on the mounting conditions to avoid excessive strain on the installation points and the M12 connections.
- Ensure correct fit and proper assembly of the M12 connecting parts. If these instructions are not complied with, the specified protection rating cannot be guaranteed.

For UL applications:

- To connect the IO-Link master and the IO-Link devices, only use UL-certified cables of the CYJV or PVVA category with a minimum temperature of 80 °C (75 °C in case of maximum ambient temperature of 40 °C).

Wiring: → **Technical data** (→ S. [80](#))

By means of basic insulation according to EN61010-1, the circuits are separated from each other and from device surfaces that could be touched (secondary circuit with 28 V DC maximum, supplied from mains circuit up to 300 V overvoltage category II).

By means of basic insulation according to EN61010-1, the communication interfaces are separated from each other and from device surfaces that could be touched (secondary circuit with 28 V DC maximum, supplied from mains circuit up to 300 V overvoltage category II). They are designed for network environment 0 according to IEC TR62102.

6.2 Modbus TCP ports

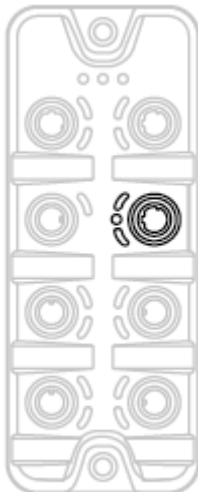
33671



- ▶ Connect the device via the M12 socket X21 and/or X22 to the Modbus TCP network (e.g. Modbus TCP PLC, additional Modbus TCP device)
 - Tightening torque: 0.6...0.8 Nm
- ▶ To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. [78](#))).
- ▶ Cover the unused sockets with M12 protective caps (art no. E12542).
 - Tightening torque 0.6...0.8 Nm

6.3 IoT port

34044



- ▶ Connect the device via the M12 socket X23 to the IT network (e.g. laptop/PC with LR DEVICE parameter setting software, laptop/PC with LR SMARTOBSERVER monitoring software, laptop/PC with software capable of processing http requests).
 - Tightening torque: 0.6...0.8 Nm
- ▶ To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. [78](#))).
- ▶ Cover the unused sockets with M12 protective caps (art no. E12542)
 - Tightening torque 0.6...0.8 Nm

6.4 IO-Link ports

51958

The IO-Link ports of the AL1341 meet the requirements of the IO-Link specifications 1.0 to 1.1.2.

- ▶ Please note the information concerning IO-Link wiring!
- ▶ Cover unused sockets with M12 protective caps (art. no.: E12542).
 - Tightening torque 0.6...0.8 Nm

6.4.1 Connect IO-Link devices for Class A operation

51959

Wiring information:

- The connected IO-Link devices must be supplied exclusively via the IO-Link master.
- The additional digital inputs IO-Link ports X01...X04 (pin 2) have a type 2 behaviour according to the standard EN61131-2. The connected electronics must be electrically suited for this.



- ▶ Connect the connectors of the IO-Link devices with the M12 sockets of the IO-Link ports X01...X04.
 - Tightening torque: 0.6...0.8 Nm
 - Maximum cable length per IO-Link port: 20 m
- ▶ To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. [78](#))).

6.5 Connect the device

33882



- ▶ Disconnect power.
- ▶ Connect the IO-Link Master via M12 socket X31 to 24 V DC (20...28 V SELV/PELV; according to EN61010-1, secondary circuit with maximum 30 V DC supplied by mains circuit up to 300 V of overvoltage category II).
 - Tightening torque: 0.6...0.8 Nm
 - Maximum cable length: 25 m
- ▶ To connect the device, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 / IP 69K or higher (→ **Accessories** (→ S. [78](#))).



When using cable length greater than 25 m keep in mind the voltage drop as well as the required minimum voltage supply of 20 V!

7 Operating and display elements

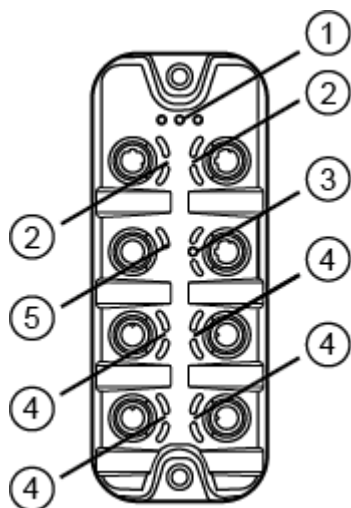
Content

Overview	18
LED indicators	19

34063

7.1 Overview

34356



- ① RDY, RUN and ERR status LEDs
→ **Status LEDs** (→ S. [19](#))
- ② LNK and ACT status LEDs of the Modbus TCP interfaces 1 (X21) and 2 (X22)
→ **Ethernet interface** (→ S. [19](#))
- ③ LNK, ACT status-LEDs and IoT LED of the IoT interface (X23)
→ **IoT port** (→ S. [20](#))
- ④ IOL and DI status-LEDs of the IO-Link port (X01...X04)
→ **IO-Link ports (Class A)** (→ S. [20](#))
- ⑤ PWR status LED of the voltage supply (X31)
→ **Voltage supply** (→ S. [20](#))

7.2 LED indicators

34047

The device only has the following LED indicators:

7.2.1 Status LEDs

34436

The RDY LED indicates the status of the gateway.

The RUN LED indicates the current state of the Modbus TCP state machine.

The ERR LED indicates occurring errors.

Status LED			Description
RDY	green	on	Status: OK
		flashes 5 Hz	Status: Error
		flashes (200 ms on, 800 ms off)	Status: Firmware update is running
		off	Status: Gateway not running or gateway booting
ERR	red	on	communication error
		flashes 10 Hz	boot error
		flashes (200 ms on, 200 ms off, 200 ms on, 1000 ms off)	watchdog error (Modbus TCP or process data)
		flashes (200 ms on, 1000 ms off)	local error
		flashes 2.5 Hz	invalid configuration
		off	no error
RUN	green	on	connection established
		flashes 1 Hz	ready, but not yet configured
		flashes 5 Hz	waiting for connection
		off	not ready

7.2.2 Ethernet interface

34348

Each Ethernet interface (X21, X22) has 2 LEDs (LNK and ACT). The LEDs indicate the status of the Ethernet connection.

Status LED			Description
LNK	green	on	Ethernet connection established
		off	No Ethernet connection
ACT	yellow	flashes	Data is transmitted via the Ethernet interface.
		off	No data transmission

7.2.3 IoT port

34043

The IoT port (X23) has the 3 LNK, ACT and IoT LEDs. The LEDs indicate the status of the Ethernet connection and the device identification.

Status LED			Description
LNK	green	on	Ethernet connection established
		off	No Ethernet connection
ACT	yellow	flashes	Data is transmitted via the Ethernet interface.
		off	No data transmission
IoT	green	flashes	Device identification active

7.2.4 Voltage supply

34191

The interface for voltage supply (X31) has the LED that is marked as US. The LED indicates the status of the voltage supply.

Status LED			Description
US	green	on	The supply voltage Us is applied.
		off	No supply voltage is applied or the applied supply voltage is too low.

7.2.5 IO-Link ports (Class A)

34074

Each IO-Link port Class A has 2 LEDs marked as IOL and DI. The LEDs indicate the status of the IO-Link port.

Status LED			Description
IOL	yellow	off	Port configured as DI / DO: pin 4 (C/Q) = OFF
		on	Port configured as DI / DO: pin 4 (C/Q) = ON
	green	flashes 1 Hz	Port configured as IO-Link: no IO-Link device detected
		flashes 2 Hz	Port configured as IO-Link: PROOPERATE state
		on	Port configured as IO-Link: OPERATE state
	red	flashes 2 Hz	Port configuration error or short circuit or overload (US)
		on	Transmission error
DI	yellow	off	Digital input : pin 2 (DI) = OFF
		on	Digital input: pin 2 (DI) = ON

8 Set-up

Content

Read device and diagnostic information	22
--	----

52357

When the supply voltage is switched on, the AL1341 starts with the factory settings. The display elements signal the current operating mode (→ **Operating and display elements** (→ S. [18](#))).

To enable parameter setting of the AL1341, the IoT interface and / or the fieldbus interface must be configured according to the network environment.

- ▶ Configure IoT interface (LR DEVICE: → **IoT: Configure IP settings** (→ S. [26](#)) or → **Configure IoT interface** (→ S. [39](#))).
- ▶ Configure fieldbus interface (LR DEVICE: → **Fieldbus: Configure IP settings** (→ S. [30](#)) or IoT: → **Configure the fieldbus interface** (→ S. [43](#))).
- > IoT / fieldbus interface has valid IP settings.
- > User can set the parameters of the AL1341.

Further steps:

- Optional: Update firmware of AL1341 (→ **Update firmware** (→ S. [75](#))).
- Set the parameters of the AL1341 (→ **Configuration** (→ S. [23](#))).

8.1 Read device and diagnostic information

In order to read the diagnostic information about the current device status via the web interface:

- ▶ Connect laptop/PC and AL1341 via the Ethernet internet.
- ▶ Start web browser.
- ▶ Enter the IP address of the AL1341 into the address field of the browser and press [ENTER] to confirm.
- > Web browser shows the web interface of the device.
- > The page shows the following data:
 - Table with connected IO-Link devices

Name	Description
[Port]	Number of the IO-Link interface
[Mode]	Operating mode of the IO-Link interface
[Comm. Mode]	Baud rate of the IO-Link interface
[MasterCycleTime]	Cycle time
[Vendor ID]	ID of the manufacturer of the IO-Link device
[Device ID]	ID of the IO-Link device
[Name]	Article number of the IO-Link device <ul style="list-style-type: none"> ▪ For ifm articles: This article number is stored along with a link to the produkt page on the ifm website.
[Serial]	Serial number of the IO-Link device
[LR Mode / Interval]	Cycle time for the communication with the SmartObserver

- Diagnostic information of the device

Name	Description
[SW-Version]	
[Current]	Current (in mA)
[Voltage]	Voltage (in mV)
[Short Circuit]	Number of detected short circuits
[Overload]	Number of detected overloads
[Undervoltage]	Number of detected under voltages
[Temperature]	Device temperature (in °C)

- Version information of the installed firmware components

Name	Description
[Firmware]	Firmware version
[Container]	Version of the firmware container
[Bootloader Version]	Version of the boot loader
[Fieldbus Firmware]	Version of the Modbus TCP firmware

9 Configuration

Content	
LR DEVICE.....	24
ifm IoT Core.....	37
Modbus TCP.....	61

33858

9.1 LR DEVICE

Content	
Remarks	25
IoT: Configure IP settings	26
IoT: Configure security mode	27
IoT: Configure access rights	28
IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER	29
Fieldbus: Configure IP settings	30
Fieldbus: set the length of the process data	31
IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER	31
IO-Link ports: Configure operating mode	32
IO-Link ports: Set the device validation and data storage	33
IO-Link ports: set fail-safe values	34
Info: Show device information	34
Firmware: Reset device to factory settings	35
Firmware: Reboot the device	35
Configure IO-Link devices	36

33692

On delivery, the AL1341 is configured with the factory settings (→ **Factory settings** (→ S. [77](#))).

Required software: LR DEVICE (1.5.0.x or higher) (art.-no.: QA0011/QA0012)

9.1.1 Remarks

Content

Offline parameter setting	25
Parameter setting with LR DEVICE	25

34180

Offline parameter setting

34060

The AL1341 supports the offline parameter setting. In this context, the user creates and stores a configuration for the IO-Link master and the connected IO-Link devices without being connected to the AL1341 (OFFLINE mode). The configuration created in this way can be stored as a file (*.lrp) and loaded to the AL1341 and activated at a later date.



Further information about offline parameter setting: → Operating instructions LR DEVICE

Parameter setting with LR DEVICE

34437

Parameter setting of the AL1341 with the LR DEVICE is only possible via the IoT interface X23.

9.1.2 IoT: Configure IP settings

For access to the IO-Link master via the IT infrastructure the user has to set the IP settings of the IoT port.



To configure the IP settings with DHCP, a DHCP server has to be active in the IT network. If no DHCP server can be reached in the IT network, an IP address is automatically assigned to the IoT port with the Zeroconfig protocol (address range: → **Factory settings** (→ S. [77](#))).

To configure the IP settings of the IoT interface:

- ▶ Select [IoT] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[DHCP]	Activate/deactivate the DHCP client of the device	[Static IP]	IP settings were set by the user
		[DHCP]	IP settings are set by a DHCP server in the network.
[IP address]*	IP address of the IoT port	Factory setting: 169.254.X.X	
[Subnet mask]*	Subnet mask of the Ethernet network	Factory setting: 255.255.0.0	
[Default gateway IP address]*	IP address of the network gateway	Factory setting: 0.0.0.0	
[MAC address]	MAC address of the IoT port	The value is firmly set.	

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

- ▶ Save changed values on the device.

9.1.3 IoT: Configure security mode

The IoT interface of the IO-Link offers a security mode. It enables secure data transmission via transport encryption and restriction of the access to IO-Link masters and IO-Link devices via user authentication.

To configure the security mode:

- ▶ Select [IoT] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Security mode HTTPS]	Set the security mode	[Disabled]	Security mode disabled
		[Enabled]	Security mode enabled
[Security password]	Password Note: The set password is not displayed.		

- ▶ Save changed values on the device.



The security mode only protects the access to the device via the IoT interface.
The user name "administrator" cannot be changed.



The security mode can be enabled without setting the password. During the attempt to write to the device, LR DEVICE requires to enter and confirm the password.

After entering the password, the user has unrestricted access to IO-Link masters and connected IO-Link devices. The password will only be requested again if the current LR DEVICE session is over (e. g. after rebooting the LR DEVICE).

To change the set password:

- ▶ Sign in with a valid password.
- ▶ Enter the new password in the field [Security password].
- ▶ Write changes to the device.
- > The new password is set.
- > LR DEVICE requires entering the new password to be able to access to the IO-Link master.

9.1.4 IoT: Configure access rights

The access rights define which instance may read and / or write the parameter data, process data and event/diagnostic messages.

In order to configure the access rights to the IO-Link master:

- ▶ Select [IoT] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Access Rights]	The access rights to the parameter data, process data and the event/diagnostic messages of the IO-Link master as well as the connected IO-Link devices	[Modbus TCP + IoT]	<ul style="list-style-type: none"> ▪ Modbus TCP and IoT Core have read and write access rights to parameters and process data ▪ Modbus TCP and <IoT Core> have read access rights to events/alarms
		[Modbus TCP + IoT (read-only)]	<ul style="list-style-type: none"> ▪ Modbus TCP has read and write access rights to parameters and process data ▪ Modbus TCP has read access rights to events/alarms ▪ IoT Core only has read access rights to parameters, process data and events/alarms
		[IoT only]	<ul style="list-style-type: none"> ▪ IoT Core has read and write access rights to parameters and process data ▪ IoT has read access rights to events/alarms ▪ Modbus TCP has no access rights

- ▶ Save changed values on the device.



If the parameter [Access rights] is set to [Modbus TCP + IoT] via IoT and Modbus TCP projection, then the parameter values set in the Modbus TCP projection software apply.
If the parameter [Access rights] is set to [IoT only] via IoT, then set the parameter [Access rights] to [Keep settings] in the Modbus TCP projection software.

Changes of the parameter [Access Rights] are only effective after restarting the device
(→ **Firmware: Reboot the device** (→ S. [35](#)))

9.1.5 IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER

To enable transfer of process data from the IO-Link master to LR AGENT or LR SMARTOBSERVER, the interface has to be configured accordingly.

- ▶ Select [IoT] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[IP address LR Agent or SMARTOBSERVER]	IP address of LR AGENT or LR SMARTOBSERVER	Factory setting: 255.255.255.255	
[Port LR Agent or SMARTOBSERVER]	Port number that is used to send process data to LR AGENT or LR SMARTOBSERVER	0 ... 65535	Factory setting: 35100
[Interval LR Agent or SMARTOBSERVER]	Cycle time for the transfer of the process data to LR AGENT or LR SMARTOBSERVER (value in milliseconds)	[Off]	no transfer
		500 ... 2147483647	500 ms ... 2147483647 ms
[Application Tag]	Source identifier of the IO-Link master in the structure of LR AGENT or LR SMARTOBSERVER (String32)	Factory setting: AL1341	



After changing the parameter [Port LR Agent or SMARTOBSERVER] or [Application Tag], it may take 120 seconds before the device establishes a new TCP connection.

To prevent the delay:

- ▶ Reboot the device after changing the the parameter.
- ▶ Save changed values on the device.

9.1.6 Fieldbus: Configure IP settings



The configuration of the IP settings of the fieldbus port is only possible via LR DEVICE and IoT.

To configure the IP settings of the Modbus TCP interface:

- ▶ Select [Fieldbus] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[DHCP]	Activate / deactivate the DHCP client of the device	[Static IP]	IP parameters are set by the user
		[DHCP]	IP parameters are set by a DHCP server in the network.
		[BOOTP]	IP parameters are set via the Bootstrap Protocol (BOOTP)
[IP address]*	IP address of the Modbus TCP interface	Factory setting: 192.168.1.250	
[Subnet mask]*	Subnet mask of the IP network	Factory setting: 255.255.255.0	
[Default gateway IP address]*	IP address of the gateway	Factory setting: 0.0.0.0	
[MAC address]	MAC address of the Modbus TCP interface	The value is firmly set.	
[Fieldbus firmware]	Firmware version of the Modbus TCP stack	e.g. 2.6.0.5	
[connectiontimeout]	Max. value for Connection Timeout (value in milliseconds)	1 ... 300000	

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

- ▶ Save changed values on the device.

9.1.7 Fieldbus: set the length of the process data

54681

To set the length of the process data to be transmitted and the arrangement of the bytes:

- ▶ Select [Fieldbus] menu.
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Process data length]	Length of the process input data and process output data per IO-Link port	2 bytes input 2 bytes output	2 bytes input data, 2 bytes output data
		4 bytes input 4 bytes output	4 bytes input data, 4 bytes output data
		8 bytes input 8 bytes output	8 bytes input data, 8 bytes output data
		16 bytes input 16 bytes output	16 bytes input data, 16 bytes output data
		32 bytes input 32 bytes output	32 bytes input data, 32 bytes output data
[Swap]	Arrangement of the bytes in process data	off	as Array of Bytes
		on	as Integer16 value; during an update of the process data, bytes are exchanged word by word (input data and output data)

- ▶ Save changed values on the device.

9.1.8 IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER

33690

The user can decide separately for each IO-Link port whether the process data of the connected IO-Link devices should be transferred to LR AGENT or LR SMARTOBSERVER.



To transfer process data the interface to the LR AGENT or LR SMARTOBSERVER has to be correctly configured (→ **IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER** (→ S. [29](#))).

To activate / deactivate data transfer:

- ▶ Select [Port x] menu (x = 1...4).
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Transmission to LR Agent or SMARTOBSERVER]	Transfer of process data of the connected IO-Link device to LR AGENT oder LR SMARTOBSERVER	[Disabled]	Transfer process data
		[Enabled]	Don't transfer process data

- ▶ Save changed values on the device.

9.1.9 IO-Link ports: Configure operating mode

The IO-Link ports X01...X04 of the device support the following operating modes:

- Digital input (DI): binary input signal at pin 4 (C/Q) of the IO-Link port
- Digital output (DO): binary output signal at pin 4 (C/Q) of the IO-Link port
- IO-Link: IO-Link data transfer via pin 4 (C/Q) of the IO-Link port

The user can set the operating mode separately for each IO-Link port.

To set the operating mode of an IO-Link port:

- ▶ Select [Port x] menu (x = 1...4).
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Mode]	Operating mode of the IO-Link port	[Disabled]	Port deactivated
		[DI]	Operation as digital input
		[DO]	Operation as digital output
		[IO-Link]	Operation as IO-Link interface
[Cycle time actual]**	Current cycle time of the data transfer between IO-Link master and IO-Link device on the port (value in microseconds)	Parameter can only be read	
[Cycle time preset]*	Cycle time of the data transfer between the IO-Link master and the IO-Link device at the port (value in microseconds)	0	The device automatically sets the fastest possible cycle time.
		1	1 microsecond
		... 132800	... 132800 microseconds
[Bitrate]**	Current transmission rate of the data transfer between the IO-Link master and the IO-Link device on the port	Parameter can only be read	

* ... Parameter only available if [Mode] = [IO-Link]

** ... Parameter only visible if the IO-Link device is connected to the IO-Link port.

- ▶ Save changed values on the device.

9.1.10 IO-Link ports: Set the device validation and data storage

The user can choose how the IO-Link ports are to behave with regard to the device validation and the storage / recovery of parameter data of the connected IO-Link device.

The following options are available:

Option	Validation of the IO-Link device	Storage of the parameter values	Recovery of the parameter values
[No check and clear]	no	no	no
[Type compatible V1.0 device]	yes, test the compatibility with IO-Link standard V1.0	no	no
[Type compatible V1.1 device]	yes, test the compatibility with IO-Link standard V1.1	no	no
[Type compatible V1.1 device with Backup + Restore]	yes, test the compatibility with IO-Link standard V1.1 and identity of design (vendor ID and device ID)	yes, automatic storage of the parameter values; changes of the current parameter values will be stored	yes, recovery of the parameter values when connecting an identical IO-Link device with factory settings
[Type compatible V1.1 device with Restore]	yes, test the compatibility with IO-Link standard V1.1 and identity of design (vendor ID and device ID)	no, there is no automatic storage changes of the current parameter values will not be stored	yes, recovery of the parameter values when connecting an identical IO-Link device with factory settings



The options only apply if the IO-Link port is in the operating mode "IO-Link".

For options [Type compatible V1.1 device with Backup + Restore] and [Type compatible V1.1 device with Restore]: If the vendor ID and device ID are changed in the online mode, the data memory will be deleted and a new backup of the parameter values of the connected IO-Link device will be created in the IO-Link master.

To configure the device validation and the data storage:

- ▶ select [Port x] menu (x = 1...4).
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Validation / Data Storage]	Supported IO-Link standard and behaviour of the IO-Link master when connecting a new IO-Link device at port x (x = 1...4)	[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 of the manufacturer that is to be validated	0...65535	Factory setting: 0# ifm electronic: 310
[Device ID]	ID of the IO-Link device that is to be validated	0...16777215	Factory setting: 0

- ▶ Save changed values on the device.

9.1.11 IO-Link ports: set fail-safe values

34459

In case the Modbus TCP connection is interrupted, fail-safe values can be assigned to the outputs of the IO-Link ports.

To set the fail-safe values of the IO-Link ports:

- ▶ Select [Port x] menu (x = 1...4).
- > The menu page shows the current settings.
- ▶ Set the following parameters as required:

Name	Description	Possible values	
[Fail-safe digital out]	Fail-safe values for output (operating mode "DO")	Reset	Reset value (LOW)
		Old	hold old value
		Set	Set value (HIGH)
[Fail-safe IO-Link]	Fail-safe value for output (operating mode "IO-Link")	Off	no fail-safe value
		Reset	reset value
		Old	hold old value
		Pattern	provide sample

- ▶ Save changed values on the device.

9.1.12 Info: Show device information

34065

To read the general information of the ifm IO-Link master:

- ▶ Select [Info] menu.
- > The menu page shows the current settings.

Name	Description	Possible values
[Product code]	Article number of the IO-Link master	AL1341
[Device family]	Device family of the IO-Link master	IO-Link master
[Vendor]	Vendor	ifm electronic gmbh
[SW-Revision]	Firmware of the IO-Link master	
[HW revision]	Hardware version of the IO-Link master	
[Bootloader revision]	Bootloader version of the IO-Link master	
[Serial number]	Serial number	

9.1.13 Firmware: Reset device to factory settings

33838

When the IO-Link master is reset, all parameters are set to the factory settings:

To reset the device to factory settings:

- ▶ Select [Firmware] menu.
- > The menu page shows the current settings.
- ▶ Click on [Factory Reset] to reset the device.
- > LR DEVICE sets the device to the factory settings.

9.1.14 Firmware: Reboot the device

33832

When rebooting the device, all settings are kept.

To restart the AL1341:

- ▶ Select [Firmware] menu.
- > The menu page shows the current settings.
- ▶ Click on [Reboot] to reboot the device.
- > LR DEVICE reboots the ifm IO-Link master.

9.1.15 Configure IO-Link devices

To configure the IO-Link devices connected to the device with the LR DEVICE parameter setting software:

Requirements:

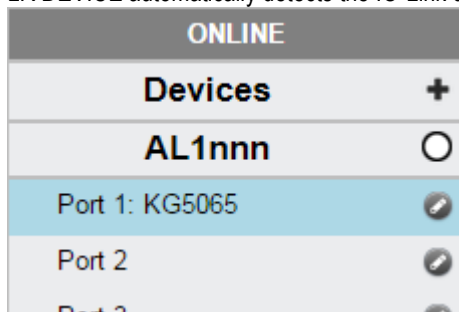
- > IO-Link master is correctly installed and connected to the LR DEVICE parameter setting software.
- > The IO-Link device is correctly connected to the AL1341.
- > Operating mode of the IO-Link port is "IO-Link" (→ **IO-Link ports: Configure operating mode** (→ S. 32)).
- > IoT has write access rights to the IO-Link master (→ **IoT: Configure access rights** (→ S. 28)).

1 Select IO-Link master

- ▶ Start LR DEVICE.
- ▶ Update IODD file library
OR:
Import IODD file of the IO-Link device manually.
- ▶ Scan network for devices.
- > LR DEVICE detects IO-Link master.

2 Add IO-Link device

- ▶ Under [ONLINE]: Click on the required IO-Link master.
- > LR DEVICE automatically detects the IO-Link devices connected to the IO-Link master (e.g. ifm sensor KG5065).



3 Configure IO-Link device

- ▶ Mouse click on the port to which the IO-Link device is connected.
- > LR DEVICE reads and shows the current parameter values of the IO-Link device.
- ▶ Configure IO-Link device.



Information about the available parameters of the IO-Link device: → IO Device Description (IODD) of the IO-Link device

- ▶ Save the changed configuration on the IO-Link device.

9.2 ifm IoT Core

Content

First steps	37
General functions	38
Configure IoT interface	39
IoT interface: Configure security mode	40
Configure the fieldbus interface	43
Configure IO-Link ports	44
Configure IO-Link devices	46
Set application identification	48
Read / write cyclic process data	49
Control IO-Link master	50
Read diagnostic data of the AL1341	52
Read device information of the IO-Link master	53
Read information about IO-Link devices	53
Subscribe to events	54
MQTT support	56
Programmers' notes	57

52244



General notes on the ifm IoT Core: → **Programmers' notes** (→ S. [57](#))

9.2.1 First steps

52245

To read the device description of the AL1341:

- ▶ Send the following POST request to the AL1341:
{"code": "request", "cid": -1, "adr": "gettree"}
- > AL1341 returns the device description as structured JSON object.
- ▶ Identify all substructures and the data points contained therein in the tree structure of the JSON object.
- ▶ Identify the applicable services for the access to substructures and the data points contained therein.

9.2.2 General functions

52246

The AL1341 is of type device (→ **Overview: IoT types** (→ S. [115](#))).

Besides gettree, the following services can be applied to the root element of type device.

Service	Description
../getidentity	Read device information
../getdatamulti	Read several parameter values sequentially
../getelementinfo	Read the uid of the AL1341
../setelementinfo	Write the uid of the AL1341

Depending on the read and write access rights, the following services can be applied to elements of type data:

Service	Description
../getdata	Read the value of the element
../setdata	Write the value of the element

9.2.3 Configure IoT interface

Via the IoT interface the AL1341 will be integrated in the IT network.

Substructure: iotsetup

Available data points:

Name	Description	Access
../accessrights	Access rights to the IO-Link master	rw
../smobip	IP address of the LR SMARTOBSERVER	rw
../smobport	Port number of the LR SMARTOBSERVER	rw
../smobinterval	Cycle time for data transmission to LR SMARTOBSERVER (value in milliseconds)	rw
../network/dhcp	Configuration of the IP settings of the IoT port	rw
../network/ipaddress	IP address of the IoT port	rw
../network/subnetmask	Subnet mask of the network segment	rw
../network/ipdefaultgateway	IP address of the network gateway	rw

rw ... read and write

Applicable services:

Name	Description
../network/setblock	Write all values of the substructure at once



If the parameter [Access rights] is set to [Modbus TCP + IoT] using IoT and Modbus TCP projection, then the parameter values set in the Modbus TCP projection software apply.

If the parameter [Access rights] is set to [IoT only] via IoT, then set the parameter [Access rights] to [Keep settings] in the Modbus TCP projection software.

Changes of the parameter [Access Rights] are only effective after restarting the device
(→ **Firmware: Reboot the device** (→ S. [35](#)))

9.2.4 IoT interface: Configure security mode

54683

The access to the IoT interface of the IO-Link master can be protected with a security mode:

Sub-structure: `iotsetup`

Available data points:

Name	Description	Access
<code>../security/securitymode</code>	active security mode	rw
<code>../security/password</code>	Password for authentication (Base64 coded)	w

rw ... read and write

w ... write only



Valid character set for the Base64 coding / decoding of the password: UTF-8

Online tool for coding / decoding: → www.base64encode.org

Note: Security mode

54684

The security mode enables restricting access to the IO-Link master and the connected IO-Link devices from the IT network. In the activated security mode, the following restrictions apply:

- Access only with authentication (password-protected user account)
- Access only via secure https connection (Transport Layer Security - TLS)



The security mode only protects the access to the device via the IoT interface.

The standard value for users is: administrator

The set password cannot be read with `getdata`.

The current status of the security function can be read with the `getidentity` service (→ **Service: `getidentity`** (→ S. [119](#))).

For the authentication, the user must additionally provide the POST requests with a valid user name and password in the field "auth". The user name and the password will be shown as Base64-coded character strings (→ **Example: Request with authentication** (→ S. [42](#))).

The following requests can be done if the security mode is enabled, also without authentication:

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

Example: Activate security mode

Task: Activate the security mode of the IO-Link interface of the IO-Link master. Set the password "password" (Base64 coded: cGFzc3dvcmQ=)

Solution: The activation consists of 2 steps:

1 Activate security mode

Use service setdata with datapoint `iotsetup/security/securitymode` to activate the security mode.

- Request:

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

- Response:

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

2 Set required password

Use service setdata with data point `iotsetup/security/password` to set the required password.

- Request:

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

- Response:

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

Example: Request with authentication

54685

Task: The temperature of the IO-Link master is to be read. The security function is enabled (current password: password).

Solution: Read the data point processdatamaster/temperature with the getdata service. The request must be sent using https. The user name and the password are transferred as a Base64-coded character string ("administrator" = "YWRtaW5pc3RyYXRvcg==", "password" = "cGFzc3dvcmQ=")

- Request:

```
{
  "code": "request",
  "cid": -1,
  "adr": "processdatamaster/temperature/getdata",
  "auth": { "user": "YWRtaW5pc3RyYXRvcg==", "passwd": "cGFzc3dvcmQ=" }
}
```

- Response:

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

Example: reset password

54686

Task: The existing password is to be reset.

Solution: To reset a password, disable the security mode. To disable it, enter the user name and the password (the fields "user" and "passwd").

- Request:

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

- Response:

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

9.2.5 Configure the fieldbus interface

34476

Via the fieldbus interface (ports X21 / X22) the AL1341 will be integrated in the Modbus TCP network.

Substructure: fieldbussetup

Available data points:

Last name	Description	Access
../fieldbusfirmware	Firmware version of the IO-Link master	r
../network/macaddress	MAC address of the fieldbus port	r
../network/ipaddress	IP address of the fieldbus port	rw*
../network/subnetmask	Subnet mask of the network segment	rw*
../network/ipdefaultgateway	IP address of the network gateway	rw*
../network/dhcp	Activate/deactivate the DHCP client of the device	rw
../connectionstatus	Status of the connection to the Modbus TCP network	r
../configuration/processdataconfiguration	Length of the process input data and process output data	rw*
../configuration/connectiontimeout	max. value for fieldbus connection timeout	rw*
../configuration/swap	Arrangement of the bytes	rw*
../configuration/port[n]/failsafedigitalout	Fail-safe value for the digital output - pin 4 (DO)	rw*
../configuration/port[n]/failsafeiolink	Fail-safe value for output data IO-Link	rw*

n ... 1...4

r ... read only

rw ... read and write

* ... parameter only editable, if connection to Modbus TCP plc is interrupted

Applicable services:

Name	Description
../network/setblock	write all values of substructure at once

9.2.6 Configure IO-Link ports

The user can configure the IO-Link ports X01...X04 separately.

Substructure: iolinkmaster/port[n] (n = 1...4).

Available data points:

Name	Description	Access
../senddatatosmob	Send process data to LR SMARTOBSERVER	rw*
../mastercycletime_preset	Cycle time of the data transfer at the IO-Link port (value in microseconds)	rw
../mastercycletime_actual	Current cycle time of the data transfer at the IO-Link port (value in microseconds)	r
../portevent	Activity display	rw
../mode	Operating mode of the IO-Link port	rw*
../comspeed	Data transfer rate of the IO-Link port	rw
../validation_datastorage_mode	Response of the IO-Link port when a new IO-Link device is connected	rw*
../validation_vendorid	IO-Link ID of the manufacturer that is to be validated	rw*
../validation_deviceid	IO-Link ID of the device that is to be validated	rw*
../datastorage	Data storage area of the port	rw
../datastorage/maxsize	Maximum size of the data storage area (in bytes)	r
../datastorage/chunksize	Size of a data segment (in bytes)	r
../datastorage/size	Size of the data storage area (in bytes)	r

r ... read only

rw ... read and write

* ... parameter only editable, if connection to the Modbus TCP plc is interrupted

Applicable services:

Service	Description
../validation_useconnecteddevice	Validate the IO-Link device connected to the IO-Link port
../datastorage/getblobdata	Read the content of the data storage area
../datastorage/stream_set	Transfer an individual data segment
../datastorage/start_stream_set	Start sequential transmission of several data segments

Example: Clone the Data Storage of an IO-Link port

Task: Save the Data Storage of IO-Link port X02 of IO-Link master 1 and restore the data at IO-Link master 2.

Solution: The cloning process consists of 2 steps. In the first step, the Data Storage of the IO-Link port of IO-Link master 1 is saved. In the second step, the saved data is restored at the Data Storage of port IO-Link port of IO-Link master 2.

Save Data Storage:

1 Preparations

- ▶ Read size of segments of Data Storage (h = number of bytes):
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/chunksize/getdata"}`
 Example: h = 256
- ▶ Read total size of Data Storage area (g = number of bytes):
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[2]/datastorage/size/getdata"}`
 Example: g = 550
- ▶ Calculate the number of reading steps n: n = first integer value to which the following applies: $g < n \cdot h$
 Example: n = 3, because $550 < 3 \cdot 256$

2 Read Data Storage of IO-Link port

- ▶ Read Data Storage segment by segment ("pos" is the byte offset, at which the reading process with length "length" starts).
`{"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}}`
 Example:
 1st read request: pos = 0, length = 256
 2nd read request: pos = 256, length = 256
 3rd read request: pos = 512, length = 256
- > Each segment value will be returned as BASE64 coded string.
- ▶ Join segments.

Restore Data Storage:

1 Preparations

- ▶ Determine the size of the saved Data Storage value (n = number of bytes).
 Example: n = 550
- ▶ Read size of segments (s = number of bytes):
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/chunksize/getdata"}`
 Example: s = 256

2 Transfer Data Storage strings

- ▶ Start transfer of Data Storage string ("size" = size of Data Storage string):
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/start_stream_set", "data": {"size": n}}`
 Example: size = 550
- ▶ Transfer Data Storage string segment by segment ("value" = string value of length s):
`{"code": "request", "cid": -1, "adr": "/iolinkmaster/port[1]/datastorage/stream_set", "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDluMi43Nw..."}}`

9.2.7 Configure IO-Link devices

52249

The ifm IoT Core supports the configuration of the connected IO-Link devices. A parameter is accessed via IO-Link index and subindex (→ IO Device Description (IODD) of the device).

Substructure: `iolinkmaster/port[n]/iolinkdevice` (n = 1...4)

Applicable services:

Service	Description
<code>../iolreadacyclic</code>	Read a parameter of an IO-Link device (acyclic)
<code>../iolwriteacyclic</code>	Write a parameter of an IO-Link device (acyclic)

Example: Read the parameter value of an IO-Link device

33847

Task: Read the serial number of the ifm temperature sensor TN2531 at IO-Link port X02

Solution: Read the serial number with the `iolreadacyclic` service from the IO-Link device (index: 21, subindex: 0)

- **Request:**

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

- **Return:**

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

The returned value is given in hexadecimal format. The conversion of the HEX value in a STRING value is: G0214280710

Example: Change the parameter value of an IO-Link device

33844

Task: Set the output configuration OUT1 of the ifm temperature sensor TN2531 at IO-Link port X02 to the value "Hnc / hysteresis function, normally closed".

Solution: Change the parameter [ou1] of the sensor to the value 4 using the `iolwritecyclicdata` service. The parameter can be accessed via IO-Link index 580, subindex 0 (→ IO-Link description of the sensor).

- Request:

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

The value has to be given in hexadecimal format. The conversion of the STRING value in a HEX value is: 34.

- Response:

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

9.2.8 Set application identification

52337

The user can set the application name of the IO-Link master:

Substructure: devicetag

Available data points:

Name	Description	Access
../applicationtag	Name of the IO-Link master (application tag)	rw

rw ... read and write

Example: Change name of the IO-Link master

a33823

Task: Set the name of the IO-Link master to AL1341 for the representation in the LR SMARTOBSERVER.

Solution: Change the parameter [Application Tag] with the setdata service to the value [AL1341].

The data point of the parameter [Application Tag] in the device description object is /devicetag/applicationtag.

- Request:

```
{
"code": "request",
"cid": 4711,
"adr": "/devicetag/applicationtag/setdata",
"data": {"newvalue": "AL1341"}
}
```

- Response:

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


9.2.9 Read / write cyclic process data

52250

To access the cyclic process data of the IO-Link ports X01...X04:

Substructure: `iolinkmaster/port[n]` ($n = 1...4$)

Available data points:

Name	Description	Access
<code>../pin2in</code>	Value of the digital input on pin 2 of the IO-Link port	r
<code>../iolinkdevice/pdin</code>	Value of the IO-Link input on pin 4 of the IO-Link port	r
<code>../iolinkdevice/pdout</code>	Value of the IO-Link output on pin 4 of the IO-Link port	rw*

r = only read

rw = read and write

* = only changeable, if not connected to fieldbus PLC

Example: Read process data of an IO-Link device

33842

Task: Read the current measured value of the ifm temperature sensor TN2531 at IO-Link port X02

Solution: Read the data point for the process input data with the `getdata` service.

- Request:

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

- Response:

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

The return value is given in hexadecimal format. Besides the temperature value the return value comprises additional information (→ IO Device Description (IODD) of the sensor). The temperature value is shown in bits 2 to 15.

$0x03C9 = 0b1111001001$

Temperature value: $0b11110010 = 242$

Therefore: The current temperature value is 24.2 °C.

9.2.10 Control IO-Link master

52251

Different services and management functions can be carried out on the IO-Link master.

Substructure: firmware

Available data points:

Name	Description	Access
../version	Software version	r
../type	Software type	r
../container	Area for updating the firmware	w
../container/maxsize	Maximum size of the container area (in bytes)	r
../container/chunksize	Size of a data segment (in bytes)	r
../container/size	Size of the container area (in bytes)	r

r = only read

w = write only

Applicable services:

Name	Description
../factoryreset	Reset IO-Link master to factory settings
../reboot	Reboot IO-Link master
../signal	Trigger the flashing of the status LED
../install	Install firmware transferred to the IO-Link master
../container/stream_set	Transfer an individual data segment
../container/start_stream_set	Start sequential transmission of several data segments

Example: Update firmware

Task:

Update the firmware of the device; size of the firmware file: 356676 bytes

Solution:

The firmware is transferred to the device in fragments (chunks). The size of the fragments depends on the size of the flash memory of the IO-Link master. To transfer the firmware, the firmware file must be converted into a character string using BASE64.

1 Preparations

- ▶ Determine the size of the fragments (g = number of bytes):
`{"code": "request", "cid": -1, "adr": "/firmware/container/chunksize/getdata"}`
- ▶ Convert the firmware file into a BASE64 string.

2 Start the transfer of the firmware

- ▶ Start the transfer of the firmware via the service `start_stream_set` (parameter "size": size of the firmware file):
`{"code": "request", "cid": -1, "adr": "/firmware/container/start_stream_set", "data": {"size": 356676}}`

3 Load the firmware into the flash memory of the IO-Link master

- ▶ Send the BASE64 string of the firmware file to the IO-Link master fragment by fragment (value = string value with length g).
`{"code": "request", "cid": -1, "adr": "/firmware/container/stream_set", "cid": -1, "data": {"value": "aWZtfgIAAABBTDF4NXhfY25faXRfdDluMi43Nw..."}}`
- ▶ Repeat step 3 until all fragments of the firmware file have been sent to the IO-Link master.
- > IO-Link master stores the segments received in the container area.

4 Install firmware

- ▶ Start the installation of the transmitted firmware.
`{"code": "request", "cid": -1, "adr": "/firmware/install", "data": {}}`

9.2.11 Read diagnostic data of the AL1341

52253

The user can read diagnostic data of the status of the IO-Link masters.

Substructure: processdatamaster

Available data points:

Name	Description	Access
../temperature	Temperature of the IO-Link master (value in °C)	r
../voltage	Voltage applied (value in V)	r
../current	Current (value in A)	r
../supervisionstatus	Diagnostic information of the device supply	r

r ... read only

Example: Read several parameter values of the IO-Link master simultaneously

33840

Task: The following current values are to be read by the IO-Link master: temperature, serial number

Solution: Read the current parameter values using the getdatamulti service (data point temperature: /processdatamaster/temperature; data point serial number: /deviceinfo/serialnumber)

- Request:

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

- Response:

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

9.2.12 Read device information of the IO-Link master

52254

To read the device information of the AL1341:

Substructure: deviceinfo

Available data points:

Name	Description	Access
../productcode	Article number	r
../vendor	Producer	r
../devicefamily	Device family	r
../hwrevision	Hardware revision	r
../serialnumber	Serial number	r
../swrevision	Firmware version	r
../bootloaderrevision	Bootloader version	r
../extensionrevisions	Firmware and bootloader version	r
../fieldbustype	Fieldbus	r

r ... read only

Additional information about the AL1341 can be read with the getidentity service (→ **Service: getidentity** (→ S. [119](#))).

9.2.13 Read information about IO-Link devices

52339

The user can obtain information about the IO-Link devices connected to the IO-Link ports.

Substructure: iolinkmaster/port[n]/iolinkdevice (n = 1...4)

Available data points:

Name	Description	Access
../status	Status of the connected IO-Link device	r
../vendorid	IO-Link ID of the vendor	r
../deviceid	IO-Link ID of the IO-Link device	r
../productname	Product name of the IO-Link device	r
../serial	Serial number of the IO-Link device	r
../applicationspecifictag	Device-specific identification (application tag)	rw

r ... read only

rw ... read and write

9.2.14 Subscribe to events

If a data point has the subelement datachanged, the user can subscribe to events.

Available data points:

Name	Description	Access
timer[n]/counter	Current value that can be subscribed to	r
timer[n]/interval	Cycle time of the update of the subscribed values	rw
iolinkmaster/port[n]/portevent	Display of the following events on IO-Link port n: <ul style="list-style-type: none"> IO-Link device connected IO-Link device disconnected Operating mode of the IO-Link port changed 	r
iolinkmaster/port[n]/iolinkdevice/iolinkevent	Display of IO-Link events	r

r ... read only

rw ... read and write

Applicable services:

Name	Description
../datachanged/subscribe	Subscribe to an event message
../datachanged/unsubscribe	Unsubscribe from an event message
../datachanged/getsubscriptioninfo	Show information about event messages

Example: Subscribe to event

33853

Task: The current values of the following parameters should be sent regularly to a network server with IP address 192.168.0.4: product name of the IO-Link device at IO-Link port X02, cyclic input data of the IO-Link device at IO-Link port X02 and the operating temperature of the IO-Link master.

Solution: Subscribe to the required data using the subscribe service.

- Request:

```
{
"code":"request",
"cid":4711,
"adr":"/timer[1]/counter/datachanged/subscribe",
"data":{"callback":"http://192.168.0.44:80/temp",
"datatosend":[
"/iolinkmaster/port[2]/iolinkdevice/productname",
"/iolinkmaster/port[2]/iolinkdevice/pdin",
"/processdatamaster/temperature"]
}
}
```

Additionally the interval of the timer[1] has to be set to a value between 500 ms and 2147483647 ms.

- Request:

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

- Response:

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

9.2.15 MQTT support

54699

The AL1341 can operate as a client in a MQTT-based communication environment. By using the subscribe service it is possible to send messages to a MQTT broker (PUBLISH).

Example: Publish the temperature to an MQTT broker

54687

Task: Publish the temperature of an IO-Link master to an MQTT broker (IP address MQTT broker: 192.168.82.100, port: 1883, Topic:abc).

Solution:

- Request:

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

- Response:

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


9.2.16 Programmers' notes

Content	
IoT Core: General information	57
Access the ifm IoT Core	58
IoT Core: Diagnostic codes	60

34229

IoT Core: General information

52256

The DataLine device family has an IoT Core. The IoT Core allows the user to address the AL1341 from IT networks via a REST API and to integrate it into Internet-of-Things applications.

A device description is stored on the AL1341. This device description is a structured, machine-readable data object in JSON format. All current values of parameters, process data, diagnostic data and device information are mapped in this data object. These data values can be read and changed by means of services.

Access the ifm IoT Core

52257

The user can access the ifm IoT Core via HTTP requests. The following request methods are available.

GET request

33804

Using the GET method the user has read access to a data point.

The syntax of the request to the IoT Core is:

`http://ip/datapoint/service`

Parameter	Description
ip	IP address of the IoT interface
data_point	Data point which is to be accessed
service	Service

The syntax of the return of the IoT Core is:

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

Parameter	Description
id	Correlation ID for the assignment of request and return
resp_data	Value of the data point; depending on the data type of the data point
diag_code	Diagnostic code (→ IoT Core: Diagnostic codes (→ S. 60))

Example: GET request

54033

Request (via browser):

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

Response:

```
{
  "cid":-1,
  "data":{"value":"AL1341"},
  "code":200
}
```

POST request

54700

Using a POST request the user has read and write access to a data point.

The syntax of the request to the IoT Core is:

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

Field	Parameter	Description
code	code_id	Service class
		▪ request Request
		▪ transaction Transaction
		▪ event Event
cid	id	Correlation ID for the assignment of request and response; vom Nutzer frei vergebare Kennung
adr	data_point	Data point of the element tree which is to be accessed
	service	Service to be performed (→ Overview: IoT services (→ S. 116))
data*	req_data	Data to be transferred to the IoT Core (e.g. new values); syntax depending on the service
auth**	usr_id	user name (base64 coded); default value: administrator
	password	password (base64 coded)

* = optional; only required for services, that submit data to the IoT core (e. g. setdata)

** = optional; only required, if security mode is activated

The syntax of the return of the IoT Core is:

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

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and response (see request)
data*	resp_data	Value of the data point; syntax depending on the service
code	diag_code	Diagnostic code (→ IoT Core: Diagnostic codes (→ S. 60))

* = optional; only required for services, that receive data from the IoT core (e.g. getdata)

Example: POST request

54035

Request:

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

Response:

```
{
  "cid": 4711,
  "data": {"value": "AL1341"},
  "code": 200
}
```

IoT Core: Diagnostic codes

54688

Code	Text	Description
200	OK	Request successfully processed
230	OK but needs reboot	Request successfully processed; IO-Link master must be restarted
231	OK but block request not finished	Request successfully processed; blockwise request, but not yet finished
232	Data has been accepted, but internally modified	New values have been accepted, but were adjusted by the IO-Link master
233	IP settings of the IoT core changed; application has to reboot the device; Wait for min. 1 second before the device is rebooted	IP settings have been successfully changed, IO-Link master will be restarted; wait for at least 1 second
400	Bad request	Invalid request
401	Unauthorized	Non authorised request
403	Forbidden	Forbidden request
500	Internal Server Error	Internal fault
503	Service Unavailable	The service is not available (e. g. IO-Link port in wrong operating mode; no IO-Link device at IO-Link port)
530	The requested data is invalid	Invalid process data
531	IO-Link error	Error in IO-Link master / device
532	PLC connected Error	Error: The IO-Link master is still connected with the fieldbus PLC

9.3 Modbus TCP

Content	
Integrate the AL1341 into the Modbus project	61
Set IO-Link master.....	63
Set IO-Link ports.....	64
Read input data of several IO-Link ports.....	65
Read input data of individual IO-Link ports	66
Write output data of several IO-Link ports.....	67
Write output data of individual IO-Link ports	68
Read diagnostic information and events	69
Read device information.....	70
Control IO-Link master	70
Configure IO-Link devices	70
Modbus TCP: Programmers' notes	71

34455

On the field bus side, the device can be configured with any Modbus TCP compatible projection software.

9.3.1 Integrate the AL1341 into the Modbus project

34456

The AL1341 provides the functionality of a Modbus-TCP slave. The user can integrate the IO-Link master via the profile of a generic Modbus-TCP slave to a fieldbus project.
The IO-Link master, the IO-Link Ports and the process data are configured via the Modbus register of the AL1341.

Example: Integrate IO-Link master in a CODESYS project



Familiarise yourself with the following CODESYS functions!

- Modbus master:
→ Online help > Fieldbus support > Modbus configurator > Modbus master
- Modbus slave device:
→ Online help > Fieldbus support > Modbus configurator > Modbus slave device

Task: Integrate IO-Link master in a CODESYS project

Hardware:

- AC14 DL as Modbus-TCP master
- AL1341 as Modbus-TCP slave

Solution:

Preparation:

► Create CODESYS project with AC14 DL.

1 Create Modbus-TCP master

- In the device tree: Right-click on [X8] node
- > Context menu appears.
- In the context menu: Select [Add Device...].
- > Dialogue window appears.
- Select the following settings:
 - 1. [Vendor]: Select [ifm electronic].
 - 2. [Device]: Select [Modbus_TCP_Master].
 - 3. [Name]: Enter a unique name.
- Click on [Add Device].
- > Device tree shows Modbus-TCP master as sub-node of the interface X8.

2 Create Modbus-TCP slave (AL1341)

- In the device tree: Right-click on the node of the added Modbus-TCP master
- In the context menu: Select [Add Device...].
- > Dialogue window appears.
- Select the following settings:
 - 1. [Vendor]: Select [ifm electronic].
 - 2. [Device]: Select [Modbus_TCP_Slave].
 - 3. [Name]: Enter a unique name
- Click on [Add Device].
- > Device tree shows AL1341 as sub-node of the Modbus-TCP master.

3 Configure Modbus-TCP slave

- In the following tabs, set the parameters as required:
 - 1. [General]: Set IP address and Unit ID
 - 2. [Modbus Slave Channel]: Add Modbus register
 - 2. [ModbusTCPSlave I/O Mapping]: Map Modbus register values to variables

9.3.2 Set IO-Link master

Register area for the access to the configuration of the IO-Link master: → **Configuration Area**
(→ S. [86](#))

The area contains the following data:

- Access rights to the IO-Link master
- Data length of the IO-Link input and output data of all IO-Link ports
- Alignment of the bytes in a data word



- Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!
- When writing several registers at once, ensure that the transferred parameter data has the correct length!

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

r/w ... read and write

9.3.3 Set IO-Link ports

34460

Register area for the access to the configuration of the IO-Link ports: → **Configuration Area** (→ S. [86](#))

The area contains the following data:

- Operating mode of IO-Link ports
- Device validation and Data storage settings
- Failsafe values of outputs



- ▶ Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!
- ▶ When writing several registers at once, ensure that the transferred parameter data has the correct length!

Register	Contents	Access
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

r/w = read and write

In addition, the user can set the IO-Link ports of the AL1341 via the following acyclic commands:

- "Set Mode": → **Command 0x10 – Set mode** (→ S. [101](#))
- "Set Validation ID / Data Storage": → **Command 0x20 – Set validation ID / data storage** (→ S. [103](#))
- "Set Fail-safe Data Pattern": → **Command 0x30 – Set fail-safe data pattern** (→ S. [105](#))

The commands use the process mechanisms of the acyclic command channel (→ **Use acyclic services** (→ S. [73](#))).

9.3.4 Read input data of several IO-Link ports

Register area for compact access to the input data of the IO-Link ports X01...X04: → **Input Data**
(→ S. [91](#))

The area contains the following data:

- Combined digital inputs - pin 2 / pin 4 (DI)
- Status information of the IO-Link ports
- Status information of the IO-Link devices
- Combined IO-Link input data



Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!

The parameter "Invalid Data" indicates whether the read IO-Link input data is valid.

- Also read and evaluate the corresponding status information in addition to the input data of the ports!

Register	Contents	Access
197	Port X01...X04: Digital Input - pin 2 / pin 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

r = read only

n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ **Configuration Area** (→ S. [86](#)))

9.3.5 Read input data of individual IO-Link ports

Register area for separate access to input data of the individual IO-Link ports: → **Single Port Access** (→ S. [95](#))

The area contains the following data for each IO-Link port X01...X04:

- Digital input data at pin 2 / pin 4 (DI)
- Status information of IO-Link port
- Diagnostic and status information of the connected IO-Link devices
- Input data IO-Link



Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!

The parameter "Invalid Data" indicates whether the read IO-Link input data is valid.

- Also read and evaluate the corresponding diagnostic information!

Register	Contents	Access
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

r ... read only

n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ **Configuration Area** (→ S. [86](#)))

9.3.6 Write output data of several IO-Link ports

Register area for compact access to the output data of the IO-Link ports X01...X04: → **Output Data** (→ S. [94](#))

The area contains the following data:

- Digital output data on pin 4 (DO)
- IO-Link output data of the IO-Link ports



- Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!

Several connected register areas can be written with one write command.

The IO-Link master writes only the outputs in "Compact Output Data" that are completely covered by the transferred output data.

Example: The configured process data length is 4 bytes. If all in all 5 words have been transferred to register 600, the IO-Link master writes the outputs X01 (words 1+2) and X02 (words 3+4). The output X03 is not written.

- When writing the IO-Link outputs, ensure that the output data has the correct length!

The output data is invalid in the following situations:

- no Ethernet cable connected
- PLC has terminated the connection
- Connection to the PLC has a 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

r/w = read and write

n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ **Configuration Area** (→ S. [86](#)))

9.3.7 Write output data of individual IO-Link ports

Register area for separate access to output data of individual IO-Link ports: → **Single Port Access** (→ S. [95](#))

The area contains the following data for each IO-Link port X01...X04:

- Digital output data at pin 4 (DO)
- IO-Link output data



- Observe the general rules for access to the Modbus registers (→ **Rules for accessing the Modbus registers** (→ S. [71](#)))!

If the user tries to write more than 34 bytes of output data to a single port, the IO-Link master cancels the execution of the command and returns the error code "2 ILLIGIAL ADDRESS".

- When writing outputs, ensure that the length of the transferred output data corresponds with the configured process data length.

The output data is invalid in the following situations:

- No Ethernet cable connected
- PLC has terminated the connection
- Connection to the PLC has a timeout

Register	Contents	Access
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

r/w ... read and write

n = [2,4,8,16,32]; is determined by parameter [Process Data Length] (→ **Configuration Area** (→ S. [86](#)))

9.3.8 Read diagnostic information and events

Register area for the access to diagnostic information of the IO-Link ports X01...X04:→**Diagnostic data** (→ S. [88](#))

The area contains the following data:

- Status/error flags for port configuration
- Vendor ID / device ID of the connected IO-Link devices
- Events and corresponding event codes



► Observe the general rules for access to the Modbus registers (→**Rules for accessing the Modbus registers** (→ S. [71](#)))!



A maximum of 3 events per IO-Link port are displayed.

One-time events will be deleted after a minimum of 10 s (Event Single Shot).

Occurring events indicate the time of the error occurrence (event appears). If the error cause disappears, this is indicated by a further event (event disappears). Both event types always occur in pairs.

Register	Contents	Access
30	Port X01: Diagnostic Data	r
40	Port X02: Diagnostic Data	r
50	Port X03: Diagnostic Data	r
60	Port X04: Diagnostic Data	r

r = read only

Additional diagnostic and status details are provided in the following register areas:

- "Input Data" area: →**Read input data of several IO-Link ports** (→ S. [65](#))
- "Single Port Access" area: →**Read input data of several IO-Link ports** (→ S. [65](#))

9.3.9 Read device information

34451

The user can read device information using the FC43.

The AL1341 supports the following data records ("Read Device ID code"):

- Basic Device Identification (0x01): contained data objects: → Modbus TCP specification
- Regular Device Identification (0x02): contained data objects: → Modbus TCP specification
- Specific Device Identification (0x04): contained data objects:

Object ID	Object name / description	Data type	Possible values
0x00	VendorName	ASCII string	ifm electronic
0x01	ProductCode	ASCII string	1341
0x02	MajorMinorRevision	ASCII string	e.g. V1.001
0x03	VendorURL	ASCII string	www.ifm.com
0x04	ProductName	ASCII string	IO-Link Master DL MOD 4P IP69K
0x05	ModelName	ASCII string	1341
0x06	UserApplicationName	ASCII string	MODBUS IO-Link master

9.3.10 Control IO-Link master

34458

The user can control the IO-Link master using the following acyclic commands:

- "Reboot": → **Command 0x40 – Reboot** (→ S. [107](#))
- "Factory Reset": → **Command 0x50 – Factory Reset** (→ S. [108](#))

The commands use the process mechanisms of the acyclic command channel (→ **Use acyclic services** (→ S. [73](#))).

9.3.11 Configure IO-Link devices

33881

The IO-Link master supports the configuration of the connected IO-Link devices from the Modbus TCP projection software. The parameters of an IO-Link device are set via IO-Link index and subindex. The number of the configurable parameters depends on the connected IO-Link device.



Available parameters of the IO-Link devices: → IO Device Description (IODD) of the IO-Link device

The user can read and write IO-Link index and subindex using the following methods:

- Acyclic communication: → **Use acyclic services** (→ S. [73](#))

9.3.12 Modbus TCP: Programmers' notes

Content

Rules for accessing the Modbus registers	71
Supported function codes	71
Note: Exception Codes	72
Use acyclic services	73

33665

Rules for accessing the Modbus registers

54703

The following general rules apply for access to the Modbus registers:

- ▶ Only use the valid function codes to read or write Modbus registers (→ **Supported function codes** (→ S. [71](#))).
- ▶ After every read or write access check the validity of the transmitted data (→ **Mapping: PQI** (→ S. [96](#))).

Supported function codes

34440

The AL1341 supports the following function codes for read and/or write access to the Modbus register:

Function code	Function name / description
03 (0x03)	Read Multiple Registers
04 (0x04)	Read Input Register
06 (0x06)	Write Single Register
16 (0x10)	Write Multiple Registers
23 (0x17)	Read/Write Multiple Registers
43 (0x2B)	Read Device Identification



Detailed information about the function codes: → MODBUS-TCP specification

Note: Exception Codes

54689

If processing the function code request was without error, the response message will have the following content:

- Response Function Code: Request Function Code
- Response Data: Requested data

If an error occurs while accessing the registers, the AL1341 replies with an error code instead of the function code. The response message has the following content:

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

The following exception codes are available:

Exception Code	Name	Description
0x1	ILLEGAL FUNCTION	Invalid function code
0x2	ILLEGAL DATA ADDRESS	Invalid data address
0x3	ILLEGAL DATA VALUE	Invalid data values
0x4	SERVER DEVICE FAILURE	Error in IO-Link master

Use acyclic services

34471

The AL1341 has a command interface to execute acyclic commands. A cyclic command consists of a request and a response.

Register	Contents	Access
500	Command Request Channel (Fieldbus PLC >>> IO-Link Master)	r/w
0	Command Response Channel (IO-Link Master >>> Fieldbus PLC)	r

Structure of the acyclic command channel: → **Acyclic Command Channel** (→ S. [97](#))

General procedure of the acyclic communication:

1 Write Command Request

- ▶ In the request channel: Write required data (except for [User ID]).
- > Write required [User ID].
- > Changed [User ID] signals a new command.
- > In the response channel: registers are reset to 0.
- > Acyclic command channel is blocked.
- > Processing of the command is started.

2 Check status

- ▶ In the response channel: Check [Command Status] register.
- > If [Command Status] <> 0: continue with step 3
- > If [Command Status] == 0: repeat step 2.

3 Read Command Response

- ▶ In the response channel: read returned user data.
- > Acyclic command channel is released.

10 Maintenance, repair and disposal

Content	
Cleaning process	74
Update firmware	75
Replace IO-Link device	76

51990

The operation of the unit is maintenance-free.

- ▶ Dispose of the unit in an environmentally friendly way in accordance with the applicable national regulations when it is no longer used.

10.1 Cleaning process

51991

- ▶ Clean the surface of the unit when necessary.
- ▶ Do not use any caustic cleaning agents for this!

10.2 Update firmware

The firmware of the IO-Link master can be updated with the following options:

- IoT Core: → **Example: Update firmware** (→ S. [51](#))
- Web interface:



If the firmware update is not successful, deactivate all connections to the LR SMARTOBSERVER and LR DEVICE and repeat the process.

- ▶ Deactivate the connection to the Modbus TCP PLC.
- ▶ Set the parameter [IP address LR SMARTOBSERVER] to 255.255.255.255 or 0.0.0.0 (→ **IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER** (→ S. [29](#))).
- ▶ Stop the LRAgent.LRDevice service in the Windows task manager.

After the firmware update, check the settings of the interface to the LR SMARTOBSERVER!

Prerequisites

- > Zip file with new firmware has been downloaded and unpacked.
- > Ethernet connection between laptop/PC and device is established.
- > Security mode is disabled.

1 Call up web interface

- ▶ Start web browser.
- ▶ Enter the following into the address field of the browser and confirm with [ENTER]:
`http://<IP address of the device>/web/update`
- > Web browser shows the [Firmware Update] page.

2 Load new firmware to AL1341

- ▶ Click on [Select file].
- > Dialogue window appears.
- ▶ Select the firmware file (.bin) and click on [Open].
- ▶ Click on [Submit] to start the firmware update.
- > Firmware is being loaded to the device.
- > After successful storage, the success message is displayed.

3 Restart the device

- ▶ Click on [Restart device now] to restart the device.
- > The status LED RDY flashes quickly.
- > Firmware is updating.
- ▶ Follow the instructions in the browser.

10.3 Replace IO-Link device

34182

To replace an IO-Link device:

Requirement:

- > IO-Link device is with factory settings.
- > IO-Link device supports IO-Link standard 1.1 or higher.

1 Set data storage

- ▶ Set the following parameters of the IO-Link port:
Validation and Data Storage = [Type compatible V1.1 device with Restore]
- ▶ Save changes.

2 Replace IO-Link device

- ▶ Disconnect old IO-Link device from IO-Link master.
- ▶ Connect new IO-Link device with the same IO-Link port of the AL1341.
- > IO-Link master copies parameter values from the data memory to the new IO-Link device.

11 Factory settings

34509

In the factory settings, the device has the following parameter settings:

Parameter	Factory setting
[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 interface)	169.254.X.X
[Subnet mask] (IoT interface)	255.255.0.0
[IP gateway address] (IoT interface)	0.0.0.0
[Modbus TCP name]	blank
Data memory (Data Storage)	empty

12 Accessories

33870

List of accessories of AL1341: → www.ifm.com > Product page > Accessories

13 Appendix

Content	
Technical data	80
Modbus TCP.....	84
ifm IoT Core.....	109

33879

13.1 Technical data

Content

Application	80
Electrical data	80
Inputs / outputs	80
Inputs	81
Outputs	81
Interfaces	81
Operating conditions	82
Approvals / tests	82
Mechanical data	82
Electrical connection	83

34188

13.1.1 Application

33878

Application	
Application	Hygienic systems; I/O modules for field applications
Daisy-chain function	Fieldbus interface

13.1.2 Electrical data

33808

Electrical data	
Operating voltage [V]	20...28 DC; (US; to SELV/PELV)
Current Consumption [mA]	300...3900; (US)
Protection class	III
Sensor supply US	
Max. current load total [A]	3.6

13.1.3 Inputs / outputs

34068

Inputs / outputs	
Total number of inputs and outputs	8; (configurable)
Number of Inputs and Outputs	Number of digital inputs: 8; Number of digital outputs: 4

13.1.4 Inputs

34069

Inputs	
Number of digital inputs	8; (IO-Link Port Class A: 4 x 2)
Switching level high [V]	11...28
Switching level low [V]	0...5
Digital inputs protected against short circuits	yes

13.1.5 Outputs

34053

Outputs	
Number of digital outputs	4; (IO-Link Port Class A: 4 x 1)
Max. current load per output [mA]	300
Short-circuit protection	yes

13.1.6 Interfaces

34389

Interfaces	
Communication interface	Ethernet; IO-Link
Communication interface	IO-Link; TCP/IP; Modbus TCP
Ethernet	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [MBit/s]	10; 100
Protocol	TCP/IP; Modbus TCP
Factory settings	<ul style="list-style-type: none"> IP address: 192.168.1.250 Subnet mask: 255.255.255.0 Gateway IP address: 0.0.0.0 MAC address: see type label
IO-Link master	
Type of transmission	COM 1 / COM 2 / COM 3
IO-Link revision	V1.1
Number of ports Class A	4
IoT interface	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [Mbits/s]	10; 100
Protocol	DGP, DCHP, Auto IP
Factory settings	<ul style="list-style-type: none"> IP address: 169.254.X.X Subnet mask: 255.255.0.0 Gateway IP address: 0.0.0.0 MAC address: see type label

13.1.7 Operating conditions

34062

Operating conditions	
Applications	Indoor use
Ambient temperature [°C]	-25...60
Storage temperature [°C]	-25...85
Max. perm. relative air humidity [%]	90
Max. height above sea level [m]	2000
Protection rating	IP 65; IP 66; IP 67; IP 69K; (operation with stainless steel protective caps: IP 69K)
Pollution Degree	2

13.1.8 Approvals / tests

33877

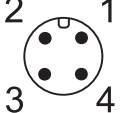
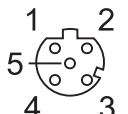
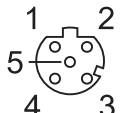
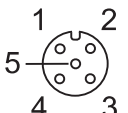
Approval / tests	
EMC	<ul style="list-style-type: none"> ▪ EN 61000-6-2 ▪ EN 61000-6-4
MTTF [Years]	90

13.1.9 Mechanical data

34050

Mechanical data	
Weight [g]	283,5
Materials	Housing: PA grey; socket: 1.4404 (stainless steel / 316L)

13.1.10 Electrical connection

Voltage supply IN X31		
Plug and socket connection	M12	
Wiring		1: + 24 V DC (US) 2: - 3: GND (US) 4: -
Ethernet IN / OUT X21, X22		
Plug and socket connection	M12	
Wiring		1: TX + 2: RX + 3: TX - 4: RX - 5: -
IoT X23		
Plug and socket connection	M12	
Wiring		1: TX + 2: RX + 3: TX - 4: RX - 5: -
Process connection IO-Link ports Class A X01...X04		
Plug and socket connection	M12	
Wiring		1: Sensor supply (US) L+ 2: DI 3: Sensor supply (US) L- 4: C/Q IO-Link 5: -

13.2 **Modbus TCP**

Content	
Register	85
Acyclic commands	100

33674

13.2.1 Register

Content	
Configuration Area.....	86
Diagnostic data.....	88
Input Data	91
Output Data	94
Single Port Access	95
Acyclic Command Channel	97

34442

The AL1341 saves the configuration data, process data and status/diagnostic data in Modbus registers.

Configuration Area

Register	Contents	
	Bits 8-15	Bits 0-7
8998	Access Rights	Process Data Length
8999	reserved	Byte Swap
9000	Port X01: Port Configuration (→ Mapping: port configuration (→ S. 87))	
9006	Port X02: Port Configuration (→ Mapping: port configuration (→ S. 87))	
9012	Port X03: Port Configuration (→ Mapping: port configuration (→ S. 87))	
9018	Port X04: Port Configuration (→ Mapping: port configuration (→ S. 87))	

Legend:

- [Access Rights] The access rights to the parameter data, process data and the events/diagnostic messages of the IO-Link master as well as the connected 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] Length of the process input data and process output data

1 byte	0x00	2 bytes input / 2 bytes output data <ul style="list-style-type: none"> ▪ Input Data: 7 Words ▪ Output Data: 5 Words
	0x01	4 bytes input / 4 bytes output data <ul style="list-style-type: none"> ▪ Input Data: 11 Words ▪ Output Data: 9 Words
	0x02	8 bytes input / 8 bytes output data <ul style="list-style-type: none"> ▪ Input Data: 19 Words ▪ Output Data: 17 Words
	0x03	16 bytes input / 16 bytes output data <ul style="list-style-type: none"> ▪ Input Data: 35 Words ▪ Output Data: 33 Words
	0x04	32 bytes input / 32 bytes output data <ul style="list-style-type: none"> ▪ Input Data: 67 Words ▪ Output Data: 65 Words
- [Byte Swap] Sequence of bytes in the data word

1 byte	0x00	as Array of Bytes
	0x01	as integer16 value; when process data is updated, bytes will be exchanged (input data and output data).
- [Port Configuration] Configuration of the IO-Link port

12 bytes	→ Mapping: port configuration (→ S. 87)
----------	---

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

- | | | | | |
|---------------------------------|---|---------|-----------------------|-----------------------------------|
| ▪ [Port Mode] | Operating mode of the IO-Link port | 1 byte | 0x00 | deactivated |
| | | | 0x01 | Digital input (DI) |
| | | | 0x02 | Digital output (DO) |
| | | | 0x03 | IO-Link |
| ▪ [Master Cycle Time] | Cycle time of the data transmission between the IO-Link master and the IO-Link device | 1 byte | 0x00 | As fast as possible |
| | | | 0x01 | 2 milliseconds |
| | | | 0x02 | 4 milliseconds |
| | | | 0x03 | 8 milliseconds |
| | | | 0x04 | 16 milliseconds |
| | | | 0x05 | 32 milliseconds |
| | | | 0x06 | 64 milliseconds |
| | | | 0x07 | 128 milliseconds |
| ▪ [Validation ID] | Supported IO-Link standard and behaviour of the IO-Link master when connecting new IO-Link devices to the IO-Link port | 1 byte | 0x00 | No validation |
| | | | 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 of the manufacturer of the device on the IO-Link port | 2 bytes | 0x0000...0xFFFF | |
| ▪ [Device ID] | Device ID of the device on the IO-Link port
Device ID = 0x123456
▪ Device ID (MSB) = 0x12
▪ Device ID = 0x34
▪ Device ID (LSB) = 0x56 | 3 bytes | pro Byte: 0x00...0xFF | |
| ▪ [Failsafe Mode -- IO-Link] | Fail-safe mode for output data when the Modbus TCP connection is interrupted | 1 byte | 0x00 | No Failsafe |
| | | | 0x01 | Failsafe Reset Value |
| | | | 0x02 | Failsafe Old Value |
| | | | 0x03 | Failsafe with Pattern |
| ▪ [Failsafe Mode -- Pin 4 (DO)] | Fail-safe value for the operating mode "digital output (DO)" | 1 byte | 0x00 | Failsafe Reset Value |
| | | | 0x01 | Failsafe Old Value |
| | | | 0x02 | Failsafe Set Value |

Diagnostic data

SYS_OBJECTID>

Register	Contents	
	Bits 8-15	Bits 0-7
30	reserved	Port X01: → Mapping: Diagnostics (→ S. 89)
31	Port X01: Vendor ID	
32	reserved	Port X01: Device ID (MSB)
33	Port X01: Device ID	Port X01: Device ID (LSB)
34...39	Port X01: Events (→ Mapping: events (→ S. 90))	
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)	

Legend:

- [Vendor ID] Vendor ID of the manufacturer of the device on the IO-Link port 2 bytes 0x0000...0xFFFF
- [Device ID] Device ID of the device on the IO-Link port 3 bytes per byte: 0x00...0xFF
 - Device ID = 0x123456
 - Device ID (MSB) = 0x12
 - Device ID = 0x34
 - Device ID (LSB) = 0x56

Mapping: Diagnostics

34504

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved	Wrong Length PD OUT	Wrong Length PD IN	Cycle time	Wrong Vendor ID/ Device ID	reserved	reserved	IOL mode

Legend:

▪ [IOL Mode]	Operating type of the IO-Link port	1 bit	0x0	Other
			0x1	IO-Link
▪ [Wrong Vendor ID/ Device ID]	Verification if the current and configured vendor ID and device ID are identical	1 bit	0x0	OK
			0x1	No match
▪ [Wrong Cycle Time]	Verification if the current and configured cycle time are identical	1 bit	0x0	OK
			0x1	No match
▪ [Wrong Length PD IN]	Verification if the size of the received input data are identical with the configured size	1 bit	0x0	OK
			0x1	Configured size too small
▪ [Wrong Length PD OUT]	Verification if the size of the sent output data is identical with the size expected by the IO-Link device	1 bit	0x0	OK
			0x1	Configured size too small

34481

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

Legend:

- | Field | Description | Size | Values | Meaning |
|---------------------|---|---------|--|-----------------------|
| [Event m: Mode] | Mode: mode of the event | 2 bits | 0x0 | reserved |
| | | | 0x1 | One-time event |
| | | | 0x2 | Event has disappeared |
| | | | 0x3 | Event has appeared |
| [Event m: Type] | Type: category of the event | 2 bits | 0x0 | reserved |
| | | | 0x1 | Notification |
| | | | 0x2 | Warning |
| | | | 0x3 | Error |
| [Event m: Src] | Source: source of the event | 1 bit | 0x0 | IO-Link Device |
| | | | 0x1 | IO-Link Master |
| [Event m: Instance] | Type: trigger of the event | 2 bits | 0x0 | unknown |
| | | | 0x1...0x3 | reserved |
| | | | 0x4 | Application |
| | | | 0x5...0x7 | reserved |
| [Event m: Code] | Code: event code; depends on the device | 2 bytes | → IODD description of the IO-Link device | |

Input Data

34447

Register	Contents
197	Port X01...X04: Digital Input - Pin 2 / Pin 4 (DI) (→ Mapping: digital input data (→ S. 92))
198	Port X01...X04: Diagnostic Information (→ Mapping: diagnostic information (→ S. 92))
199	Port X01...X04: Status Information IO-Link Ports (→ Mapping: Status information IO-Link ports (→ S. 93))
200	Port X01...X04: Compact Input Block (4n bytes) (→ Mapping: Compact Input Block (→ S. 93))

Legend:

- [Digital Input - Pin 2 / Pin 4 (DI)] Digital input data Pin 2 / Pin 4 (operating mode DO) of 4 IO-Link ports 2 bytes
- [Diagnostic Information] Diagnostic information 2 bytes
- [Status Information IO-Link Ports] Status information of the IO-Link ports 2 bytes
- [Compact Input Block (4n Bytes)] Input data (operating mode IO-Link) of 4 IO-Link ports 4n bytes per byte: 0x00...0xFF
n = [2,4,8,16,32]; is determined by parameters [Process Data Length]
(→ **Configuration Area** (→ S. [86](#)))

Mapping: Status information IO-Link ports

34485

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

Legend:

- [Data invalid] indicates the status of the process input data on the IO-Link port 1 bit 0x0 data valid
0x1 data invalid
- [Dev Not Conn] Device Connected: indicates the connection to the device on the IO-Link port 1 bit 0x0 available
0x1 not available

Mapping: Compact Input Block

54567

IO-Link port	Register area				
	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

Output Data

34488

Register	Contents	
	Bits 8-15	Bits 0-7
599	Port X01...X04: Digital Output - Pin 4 (DO) (→ Mapping: Digital output data (→ S. 94))	
600	Port X01...X04: Compact Output Block (4n bytes) (→ Mapping: Compact Output Block (→ S. 94))	

Legend:

- [Digital Output - Pin 4] Digital output data - pin 4 (operating mode DO) of 4 IO-Link ports 2 bytes (DO)]
- [Compact Output Block (4n Bytes)] Output data (operating mode IO-Link) of 4 IO-Link ports 4n bytes per byte: 0x00...0xFF
n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ **Configuration Area** (→ S. 86))

Mapping: Digital output data

34493

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: Pin 4	X03: Pin 4	X02: Pin 4	X01: Pin 4

Legend:

- [Pin4] Signal level on pin 4 of the IO-Link port (DO) 1 bit 0x0 LOW
0x1 HIGH

Mapping: Compact Output Block

54568

IO-Link port	Register area				
	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

34444

Register	Contents	
	Bits 8-15	Bits 0-7
1000	Port X01: Digital Input - pin 2	Port X01: Digital Input - pin 4 (DI)
1001	Port X01: → Mapping: Status information (→ S. 96)	Port X01: → Mapping: PQI (→ S. 96)
1002	Port X01: Input Data IO-Link (n bytes)	
1100	reserved	Port X01: Digital Output - pin 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)	

Legend:

- | | | | | |
|-----------------------------------|--|---------|-----------------------|------|
| ▪ [Digital Input - pin 2] | pin 2 signal level (if used) | 1 byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Digital Input - pin 4 (DI)] | pin 4 signal level (operating mode DI) | 1 byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Input Data IO-Link (n Bytes)] | Input data (operating mode IO-Link) (n bytes)
n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ Configuration Area (→ S. 86)) | n bytes | per byte: 0x00...0xFF | |
| ▪ [Digital Output - pin 4(DO)] | pin 4 signal level (operating mode DO) | 1 byte | 0x00 | LOW |
| | | | 0x01 | HIGH |
| ▪ [Output Data IO-Link (n Bytes)] | Output data (operating mode IO-Link) (n bytes)
n = [2,4,8,16,32]; is determined by parameters [Process Data Length] (→ Configuration Area (→ S. 86)) | n bytes | per byte: 0x00...0xFF | |

Acyclic Command Channel

34449

The following Modbus registers are available for acyclic data transmission:

Register	Contents	
	Bits 8-15	Bits 0-7
500	Command Request Channel (→ Request channel (→ S. 97))	
0	Command Response Channel (→ Response channel (→ S. 98))	

Legend:

- [Command Request Channel] Area for transmission of command request (fieldbus PLC >>> IO-Link master) 44 bytes
- [Command Response Channel] Area for transmission of command response (IO-Link master >>> fieldbus PLC) 44 bytes

Request channel

34450

Register	Contents	
	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)

Legend:

- [Port No.] Number of the IO-Link port 1 Word 0x0001 Port X01
0x0002 Port X02
...
0x0004 Port X04
- [Index] Index of the IO-Link object 1 Word 0x0000...0xFFFF
- [Subindex] Subindex of the IO-Link object 1 Word 0x0000...0xFFFF
- [Command] Command number 1 byte 0x01 Read
0x02 Write
- [User ID] ID to identify the command 1 byte 0x00...0xFF
- [Data Length (Number of Bytes)] Number of bytes that contain relevant user data (is only evaluated for Command = 0x02) 1 Word 0x0000 0 bytes
...
0x0022 34 bytes
- [Data (Byte n)] user data n bytes per byte: 0x00 ... 0xFF

34453

Register	Contents	
	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)

Legend:

- | Field | Description | Size | Value(s) | Meaning |
|---------------------------------|---|---------|--|---|
| [Port No.] | Number of the IO-Link port | 1 Word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0004 | Port X04 |
| [Index] | Index of the IO-Link object | 1 Word | 0x0000...0xFFFF | |
| [Subindex] | Subindex of the IO-Link object | 1 Word | 0x0000...0xFFFF | |
| [Command] | Command number | 1 byte | 0x01 | Read |
| | | | 0x02 | Write |
| [User ID] | reflected User ID from request channel | 1 byte | 0x00...0xFF | |
| [Result] | Status of the command processing | 1 Word | 0x0000 | OK |
| | | | 0x000F | OK, but data length too long (only with [Command] = 0x02) |
| | | | 0x00FF | Error |
| [Data Length (Number of Bytes)] | Number of bytes that contain relevant user data | 1 Word | 0x0000 | 0 bytes |
| | | | ... | |
| | | | 0x0020 | 32 bytes |
| [Error Code] | Error Code | 1 byte | → Error codes (→ S. 99) | |
| [Additional Code] | Additional error codes | 1 byte | → Additional Codes (→ S. 99) | |
| [Data (Byte n)] | User data (byte n) | n bytes | per byte: 0x00...0xFF | |

Error codes

34342

Error code	Description
0x71	Service not available (unknown command has been sent to the IO-Link port)
0x72	Port blocked (another cyclic process accesses the IO-Link port)
0x73	Forbidden (access rights don't allow command processing)
0x74	Invalid data (wrong parameter has been sent in the command)
0x76	Wrong port (wrong port number)
0x77	Wrong port function (wrong port function or wrong parameter has been sent to the device)
0x78	Invalid length (set length is > 0x20)
0x80	Error in the device application; observe additional code (→ Additional Codes (→ S. 99))

Additional Codes

54584

Code	Name	Description
0x00	APP_DEV	Device application error - no details
0x11	IDX_NOTAVAIL	Index not available
0x12	SUBIDX_NOTAVAIL	Subindex not available
0x20	SERV_NOTAVAIL	Service temporarily not available
0x21	SERV_NOTAVAIL_LOCTRL	Service temporarily not available - local control
0x22	SERV_NOTAVAIL_DEVCTRL	Service temporarily not available - device control
0x23	IDX_NOT_WRITEABLE	Access denied
0x30	PAR_VALOUTOFRNG	Parameter value out of range
0x31	PAR_VALGTLIM	Parameter value above limit
0x32	PAR_VALLTLIM	Parameter value below limit
0x33	VAL_LENVERRUN	Parameter length overrun
0x34	VAL_LENUNDRUN	Parameter length underrun
0x35	FUNC_NOTAVAIL	Function not available
0x36	FUNC_UNAVAILTEMP	Function temporarily not available
0x40	PAR_SETINVALID	Invalid parameter set
0x41	PAR_SETINCONSIST	Inconsistent parameter set
0x82	APP_DEVNOTRDY	Application not ready



Additional Codes are only available, if Error Code = 0x80 (→ **Error codes** (→ S. 99))

13.2.2 Acyclic commands

Content

Command 0x10 – Set mode	101
Command 0x20 – Set validation ID / data storage.....	103
Command 0x30 – Set fail-safe data pattern.....	105
Command 0x40 – Reboot.....	107
Command 0x50 – Factory Reset.....	108

34331

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

▪ [Port No.]	Number of the IO-Link port	1 word	0x0001	Port X01
			0x0002	Port X02
		
			0x0004	Port X04
▪ [User ID]	reflected User ID from request channel	1 byte	0x00 .. 0xFF	
▪ [Result]	Status of the command processing	1 byte	0x00	OK
			0xFF	Error
▪ [Data Length (Number of Bytes)]	Number of bytes that contain relevant user data	1 word	0x0001	1 byte
			0x0002	2 bytes
▪ [Target Mode]	Operating type of the IO-Link port	1 byte	0x00	deactivated
			0x01	digital input (DI)
			0x02	digital output (DO)
			0x03	IO-Link
▪ [Error Code]	Error ID	1 byte	→ Error codes (→ S. 99)	
▪ [Additional Code]	Additional error codes	1 byte	→ Additional Codes (→ S. 99)	

34497

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

Legend:

- | | | | | |
|---|--|--------|--|-------------------------------|
| ▪ [Port No.] | Number of the IO-Link port | 1 word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0004 | Port X04 |
| ▪ [User ID] | reflected user ID from request channel | 1 byte | 0x00 .. 0xFF | |
| ▪ [Result] | Status of the command processing | 1 byte | 0x00 | OK |
| | | | 0xFF | Error |
| ▪ [Data Length
(Number of
Bytes)] | Number of bytes that contain relevant user data | 1 word | 0x0001 | 1 byte |
| | | | 0x0002 | 2 bytes |
| ▪ [Validation ID] | Supported IO-Link standard and behaviour of the IO-Link master when connecting new IO-Link devices to the IO-Link port | 1 byte | 0x00 | no validation |
| | | | 0x01 | V1.0 device |
| | | | 0x02 | V1.1 device |
| | | | 0x03 | V1.1 device, backup + restore |
| | | | 0x04 | V1.1 device, backup |
| ▪ [Error Code] | Error ID | 1 byte | → Error codes (→ S. 99) | |
| ▪ [Additional Code] | additional error codes | 1 byte | → Additional Codes (→ S. 99) | |

Command 0x30 – Set fail-safe data pattern

34379

The command sets the behaviour of the outputs when the Modbus TCP connection and the corresponding fail-safe values are interrupted.



Corresponding parameter: [Fail-safe Mode] (→ **Mapping: port configuration** (→ S. [87](#)))

The number of the required fail-safe values results from the size of the output data (→ **Configuration Area** (→ S. 86)).

Command request

34498

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

Legend:

- | | | | | |
|--------------------------------------|---|--------|--------------|------------------------|
| ▪ [Port No.] | Number of the IO-Link port | 1 word | 0x0001 | Port X01 |
| | | | 0x0002 | Port X02 |
| | | | ... | ... |
| | | | 0x0004 | Port X04 |
| ▪ [User ID] | ID to identify the command | 1 byte | 0x00 .. 0xFF | |
| ▪ [Data Length
(Number of Bytes)] | Number of bytes that contain relevant user data
(is only evaluated for Command = 0x02) | 1 word | 0x0002 | 2 bytes |
| | | | ... | ... |
| | | | 0x0022 | 34 bytes |
| ▪ [Failsafe Mode] | Fail-safe mode for the outputs of the IO-Link ports in case of an interruption of the Modbus TCP connection | 1 word | 0x0000 | No Failsafe |
| | | | 0x0001 | Failsafe: Reset Value |
| | | | 0x0002 | Failsafe: Old Value |
| | | | 0x0003 | Failsafe: with Pattern |
| ▪ [Failsafe Data
(Byte n)] | Fail-safe values for the outputs (only with fail-safe mode = 0x0003) | 1 byte | 0x00 .. 0xFF | |

Command response

34496

Register	Contents	
	Bits 8-15	Bits 0-7

Command 0x40 – Reboot

34457

The command reboots the AL1341.

Command request

34494

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

Legend:

- [User ID] ID to identify the command 1 byte 0x00 .. 0xFF

Command response

34508

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

Legend:

- [User ID] reflected User ID from request channel 1 byte 0x00 .. 0xFF
- [Result] Status of the command processing 1 word 0x0000 OK
0x00FF Error
- [Data Length (Number of Bytes)] Number of bytes that contain relevant user data 1 word 0x0001 1 byte
0x0002 2 bytes
- [Error Code] Error ID 1 byte → **Error codes** (→ S. [99](#))
- [Additional Code] Additional error codes 1 byte → **Additional Codes** (→ S. [99](#))

Command 0x50 – Factory Reset

34499

The command resets all parameters to the factory settings (→ **Factory settings** (→ S. [77](#))).

Command request

34501

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

Legend:

- [User ID] ID to identify the command 1 byte 0x00 .. 0xFF

Command response

34507

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

Legend:

- [User ID] reflected User ID from request channel 1 byte 0x00 .. 0xFF
- [Result] Status of the command processing 1 word 0x0000 OK
0x00FF Error
- [Data Length (Number of Bytes)] Number of bytes that contain relevant user data 1 word 0x0001 1 byte
0x0002 2 bytes
- [Error Code] Error ID 1 byte → **Error codes** (→ S. [99](#))
- [Additional Code] additional error codes 1 byte → **Additional Codes** (→ S. [99](#))

13.3 ifm IoT Core

Content	
Overview: IoT profile.....	110
Overview: IoT types	115
Overview: IoT services	116

33803

13.3.1 Overview: IoT profile

Content

Profile: blob	110
Profile: deviceinfo	111
Profile: devicetag	111
Profile: iolinkdevice_full	112
Profile: iolinkmaster	112
Profile: network	113
Profile: parameter	113
Profile: processdata	113
Profile: service	113
Profile: software	114
Profile: software/uploadedablessoftware	114
Profile: timer	114

34054

Profile: blob

52264

Element (identifier)	Characteristics	Mandatory	Comments
blobname	<ul style="list-style-type: none"> type = data profiles = blob 		Characterises the element as device information
../size	type = data	mandatory	
../chunksize	type = data	mandatory	
../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	

Profile: deviceinfo

34207

Element (identifier)	Properties	mandatory	Comments
deviceinfo	<ul style="list-style-type: none"> type = structure profile = deviceinfo 		characterises the element as device information
../devicename	type = data	optional	
../devicefamily	type = data	optional	
../devicevariant	type = data	optional	
../devicesymbol	type = data	optional	
../deviceicon	type = data	optional	
../serialnumber	type = data	mandatory	
../productid	type = data	optional	
../productname	type = data	optional	
../productcode	type = data	mandatory	
../producttext	type = data	optional	
../ordernumber	type = data	optional	
../productiondate	type = data	optional	
../productioncode	type = data	optional	
../hwrevision	type = data	mandatory	
../swrevision	type = data	mandatory	
../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	

Profile: devicetag

34206

Element (identifier)	Properties	mandatory	Comments
devicetag	<ul style="list-style-type: none"> type = structure profile = devicetag 		
../applicationtag	type = data	mandatory	
../applicationgroup	type = data	optional	
../machinecode	type = data	optional	
../tenant	type = data	optional	

Profile: iolinkdevice_full

52265

Element (identifier)	Characteristics	Mandatory	Comments
iolinkdevice	<ul style="list-style-type: none"> type = structure profile = iolinkdevice_full 		Structure of an IO-Link device
../vendorid	type = data	mandatory	
../deviceid	type = data	mandatory	
../productname	type = data	mandatory	
../serial	type = data	mandatory	
../applicationspecifictag	type = data	mandatory	
../pdin	type = data	mandatory	
../pdout	type = data	mandatory	
../status	type = data	mandatory	
../iolreadacyclic	type = data	mandatory	
../iolwriteacyclic	type = data	mandatory	
../iolinkevent	type = data	mandatory	

Profile: iolinkmaster

34205

Element (identifier)	Properties	Mandatory	Comments
masterport	<ul style="list-style-type: none"> type = structure profile = iolinkmaster 		Executable service
../mode	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../comspeed	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../mastercycletime_actual	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../mastercycletime_preset	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../validation_datastorage_mode	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../validation_vendorid	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../validation_deviceid	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../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 	mandatory	
../iolinkdevice	<ul style="list-style-type: none"> type = structure profile = iolinkdevice_full 	mandatory	

Profile: network

52266

Element (identifier)	Characteristics	Mandatory	Comments
network	<ul style="list-style-type: none"> type = structure profiles = deviceinfo 		Characterises the element as device information
../macaddress	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../ipaddress	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../ipv6address	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../subnetmask	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../ipdefaultgateway	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../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	

Profile: parameter

34215

The profile is used to mark the elements of type data as parameters (acyclic data). The profile defines no substructure.

Profile: processdata

34225

The profile is used to mark the elements of type data as process data (cyclic data). The profile does not define a substructure.

Profile: service

34224

Element (identifier)	Properties	mandatory	Comments
service	<ul style="list-style-type: none"> type = service profile = service 		Executable service

Profile: software

34223

Element (identifier)	Properties	mandatory	Comments
software	<ul style="list-style-type: none"> type = structure profile = software 		characterises the element as software
../version	type = data	mandatory	
../type	type = data	mandatory	
../status	type = structure	optional	
../diag	type = structure	optional	

Profile: software/uploadablessoftware

52267

Element (identifier)	Characteristics	Mandatory	Comments
software	<ul style="list-style-type: none"> type = structure profiles = software/uploadablessoftware 		Software that can be loaded to the device via the IoT Core
../lastinstall	type = data	optional	
../installhistory	type = data	optional	
../container	<ul style="list-style-type: none"> type = data profile = blob 	mandatory	
../preinstall	type = service	optional	
../install	type = service	mandatory	
../postinstall	type = service	optional	
../abortinstall	type = service	optional	
../installstatus	type = data	optional	

Profile: timer

34226

Element (identifier)	Properties	mandatory	Comments
timer	<ul style="list-style-type: none"> type = structure profile = timer 		Executable service
../counter	<ul style="list-style-type: none"> type = data profile = parameter 	mandatory	
../interval	<ul style="list-style-type: none"> type = data profile = parameter 	optional	
../start	type = service	optional	
../stop	type = service	optional	

13.3.2 Overview: IoT types

34055

The ifm IoT Core uses the following element types:

Name	Description
structure	Element is a structure element (like a folder in a file system)
service	Element is a service that can be addressed from the network
event	Element is an event that can be started by the firmware and sends messages.
data	Element is a data point
device	Root element a device represents

13.3.3 Overview: IoT services

Content	
Service: factoryreset	116
Service: gettree	117
Service: getdata	117
Service: getdatamulti	118
Service: getidentity	119
Service: getsubscriptioninfo.....	120
Service: iolreadacyclic	121
Service: iolwriteacyclic.....	121
Service: reboot	121
Service: setblock	122
Service: setdata	122
Service: setelementinfo	123
Service: signal	123
Service: subscribe	124
Service: unsubscribe	124

34056

Service: factoryreset

34184

Name: factoryreset
Description: The service sets the parameters of the device to the factory settings.
Request data (field "data"): none
Response data (field "data"): none

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

Service: gettree

34175

Name: gettree**Description:** The service reads the complete device description of the AL1341 and provides it as JSON object.**Request data (field "data"):** none**Response data (field "data"):**

Data field	Required field	Data type	Description
Identifier	mandatory	STRING	Identifier of the root element
type	mandatory	STRING	Type of the element
format	optional	JSON object	Format of the data content
uid	optional	STRING	
profiles	optional	JSON array	
subs	mandatory	JSON array	Subelements
hash	optional	STRING	

Example:

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

Service: getdata

34183

Name: getdata**Description:** Service reads the value of a data point and provides it.**Request data (field "data"):** none**Return data (field "data"):**

Data field	Required field	Data type	Description
value	mandatory	STRING	Value of the element/data point

Example:

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

Service: getdatamulti

34174

Name: getdatamulti**Description:** The service sequentially reads the values of several data points and provides them. The value and the diagnostic code are provided for each data point.**Request data (field "data"):**

Data field	Required field	Data type	Description
datatosend	mandatory	ARRAY OF STRINGS	List of data points to be requested; data points must support the service getdata ("datatosend":["url1","url2",...,"urlx"])
consistent	optional	BOOL	

Response data (field "data"): for each requested data point

Data field	Required field	Data type	Description
url	mandatory	STRING	Data point request
code	mandatory	INT	Diagnostic code of the request
data	mandatory	STRING	Value of the data point

Service: getidentity

54690

Name: getidentity**Description:** The service reads the device information of the AL1341 and issues it.**Request data ("data" field):** none**Return data ("data" field):**

Data field	Required field	Data type	Description
iot		Device	Device description as JSON object
iot.name	mandatory	STRING	
iot.uid	optional	STRING	
iot.version	mandatory	STRING	
iot.catalogue	optional	ARRAY OF OBJECTS	
iot.deviceclass	optional	ARRAY OF STRING	
iot.serverlist	optional	ARRAY OF OBJECTS	
device	optional		AL1341
device.serialnumber	optional		Serial number
device.hwrevision	optional		Hardware version
device.swrevision	optional		Software version
device.custom	optional		
Security	optional		Security options
security.securitymode	optional	ENUM	shows if the security mode is activated
security.authscheme	optional	ENUM	shows the active authentication scheme
security.ispasswordset	optional	BOOL	shows whether a password has been set
security.activeconnection	optional	ENUM	shows the currently used communication interface
			▪ tcp_if unencrypted http connection at the IoT interface, port 80
			▪ tls_if encrypted https connection at the IoT interface, port 443
			▪ fb_if unencrypted http connection at the fieldbus interface, port 80

Service: getsubscriptioninfo

34172

Name: getsubscriptioninfo**Description:** The service provides information about an existing subscription (subscribe).**Request data (field "data"):**

Data field	Required field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path

Response data (field "data"):

Data field	Required field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path
datatosend	mandatory	ARRAY OF STRINGS	List of subscribed data points

Example:

- Request:**

```
{
  "code": "request",
  "cid": 4715,
  "adr": "/timer[1]/counter/datachanged/getsubscriptioninfo",
  "data": {
    "callback": "http://192.168.0.44:80/temp"
  }
}
```

- Response:**

```
{
  "cid": 4715,
  "data": {
    "callback": "http://192.168.0.44:80/temp",
    "datatosend": [
      "/iolinkmaster/port[2]/iolinkdevice/productname",
      "/iolinkmaster/port[2]/iolinkdevice/pdin",
      "/processdatamaster/temperature"
    ]
  },
  "code": 200
}
```


Service: iolreadacyclic

34178

Name: iolreadacyclic**Description:** The service acyclically reads the parameter value of an IO-Link device. The parameter is accessed via IO-Link index and subindex.**Request data (field "data"):**

Data field	Required field	Data type	Description
index	mandatory	NUMBER	IO-Link index of the parameter
subindex	mandatory	NUMBER	IO-Link subindex of the parameter

Response data (field "data"):

Data field	Required field	Data type	Description
value	mandatory	STRING	Value of the parameter; Value in hexadecimal format

Service: iolwriteacyclic

34177

Name: iolwriteacyclic**Description:** The service acyclically writes the parameter value of an IO-Link device. The parameter is accessed via IO-Link index and subindex.**Request data (field "data"):**

Data field	Required field	Data type	Description
index	mandatory	NUMBER	IO-Link index of the parameter
subindex	mandatory	NUMBER	IO-Link subindex of the parameter
value	mandatory	STRING	New value of the parameter; Value in hexadecimal format

Response data (field "data"): none**Service: reboot**

34176

Name: reboot**Description:** The service reboots the device.**Request data (field "data"):** none**Return data (field "data"):** none**Example:**

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

Service: setblock

34186

Name: setblock**Description:** The service simultaneously sets the values of several data points of a structure.**Request data (field "data"):**

Data field	Required field	Data type	Description
datatosend	mandatory	ARRAY OF (STRINGS)	List of data points and their new values; data points must support the service setdata
consistent	optional	BOOL	

Response data (field "data"): none

Example:

```
{
  "code": "request",
  "cid": 4711,
  "adr": "/iotsetup/network/setblock",
  "data": {
    "consistent": true,
    "datatosend": [
      "ipaddress": "192.168.0.6",
      "ipdefaultgateway": "192.168.0.250"
    ]
  }
}
```

Service: setdata

34196

Name: setdata**Description:** The service sets the value of the data point.**Request data (field "data"):**

Data field	Required field	Data type	Description
newvalue	mandatory	STRING	New value of the element/data point

Response data (field "data"): none

Example:

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

Service: setelementinfo

34195

Name: setelementinfo**Description:** The service sets the uid of an element.**Request data (field "data"):**

Data field	Required field	Data type	Description
url	mandatory	STRING	URL of the element to be changed
uid	optional	STRING	UID to be set
profiles	optional	JSON array	
format	optional	JSON object	

Response data (field "data"):

Data field	Required field	Data type	Description
identifier	mandatory	STRING	Identifier of the element
type	mandatory	STRING	Type of the element
format	optional	JSON object	Format of the data or the service content
uid	optional	STRING	
profiles	optional	JSON array	
hash	optional	STRING	

Service: signal

33819

Name: signal**Description:** The service starts the flashing of the status LEDs of the AL1341.**Request data (field "data"):** none**Return data (field "data"):** none**Example:**

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

Service: subscribe

34194

Name: subscribe

Description: The service subscribes to the values of data points. The data points to be subscribed are transferred as a list. The IO-Link master sends changes to the data drain defined in callback.

Request data (field "data"):

Data field	Required field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path
datatosend	mandatory	ARRAY OF STRINGS	List from URLs of data elements; elements have to support getdata

Response data (field "data"): none**Service: unsubscribe**

34197

Name: unsubscribe

Description: The service deletes an existing subscription. unsubscribe is successful if cid and the callback address are registered for a subscription (subscribe). If the STRING "DELETE" is provided in callback, the IO-Link master deletes all active subscriptions.

Request data (field "data"):

Data field	Required field	Data type	Description
callback	mandatory	STRING	Address to which IoT Core event notifications are to be sent; complete URL: http://ipaddress:port/path

Response data (field "data"): none

14 Index

A

Access the ifm IoT Core	58
Accessories	78
Acyclic Command Channel	97
Acyclic commands	100
Additional Codes	99
Appendix	79
Application	80
Approvals / tests	82

C

Cleaning process	74
Command 0x10 – Set mode	101
Command 0x20 – Set validation ID / data storage	103
Command 0x30 – Set fail-safe data pattern	105
Command 0x40 – Reboot	107
Command 0x50 – Factory Reset	108
Command request	101, 103, 105, 107, 108
Command response	102, 104, 105, 107, 108
Communication, parameter setting, evaluation	11
Configuration	23
Configuration Area	86
Configure IO-Link devices	36, 46, 70
Configure IO-Link ports	44
Configure IoT interface	39
Configure the fieldbus interface	43
Connect IO-Link devices for Class A operation	16
Connect the device	16
Control IO-Link master	50, 70

D

Diagnostic data	88
Digital inputs	12

E

Electrical connection	14, 83
Electrical data	80
Error codes	99
Ethernet interface	19
Example	
Activate security mode	41
Change name of the IO-Link master	48
Change the parameter value of an IO-Link device	47
Clone the Data Storage of an IO-Link port	45
GET request	58
Integrate IO-Link master in a CODESYS project	62
POST request	60
Publish the temperature to an MQTT broker	56
Read process data of an IO-Link device	49
Read several parameter values of the IO-Link master simultaneously	52
Read the parameter value of an IO-Link device	46
Request with authentication	42
reset password	42
Subscribe to event	55
Update firmware	51
Explanation of Symbols	6

F

Factory settings	77
Fieldbus	
Configure IP settings	30
set the length of the process data	31
Firmware	
Reboot the device	35
Reset device to factory settings	35
First steps	37
Function	10

G

General	7
General functions	38
GET request	58

I

ifm IoT Core	37, 109
Info	
Show device information	34
Input Data	91
Inputs	81
Inputs / outputs	80
Integrate the AL1341 into the Modbus project	61
Intended use	9
Interfaces	81
Internet of Things (IoT)	11
IO-Link	11
IO-Link ports	16
Activate data transfer to LR AGENT or LR SMARTOBSERVER	31
Configure operating mode	32
set fail-safe values	34
Set the device validation and data storage	33
IO-Link ports (Class A)	20
IO-Link supply	12
IoT	
Configure access rights	28
Configure IP settings	26
Configure security mode	27
Configure the interface to LR AGENT or LR SMARTOBSERVER	29
IoT Core	
Diagnostic codes	60
General information	57
IoT interface	
Configure security mode	40
IoT port	15, 20
IT safety	8

L

LED indicators	19
Legal and copyright information	5
LR DEVICE	24

M

Maintenance, repair and disposal	74
Mapping	
Compact Input Block	93
Compact Output Block	94
diagnostic information	92
Diagnostics	89
digital input data	92

Digital output data	94
events	90
port configuration	87
PQI	96
Status information	96
Status information IO-Link ports	93
Mechanical data	82
Modbus TCP	11, 61, 84
Programmers' notes	71
Modbus TCP ports	15
Modification history	6
Mount the device	13
Mounting	13
MQTT support	56

N

Note	
Exception Codes	72
Security mode	40
Notes	14

O

Offline parameter setting	25
Operating and display elements	18
Operating conditions	82
Output Data	94
Outputs	81
Overview	18
IoT profile	110
IoT services	116
IoT types	115

P

Parameter setting	12
Parameter setting with LR DEVICE	25
Permitted use	9
POST request	59
Preliminary note	5
Profile	
blob	110
deviceinfo	111
devicetag	111
iolinkdevice_full	112
iolinkmaster	112
network	113
parameter	113
processdata	113
service	113
software	114
software/uploadedablessoftware	114
timer	114
Programmers' notes	57
Prohibited use	9
Purpose of the document	5

R

Read / write cyclic process data	49
Read device and diagnostic information	22
Read device information	70
Read device information of the IO-Link master	53
Read diagnostic data of the AL1341	52
Read diagnostic information and events	69
Read information about IO-Link devices	53

Read input data of individual IO-Link ports	66
Read input data of several IO-Link ports	65
Register	85
Remarks	25
Replace IO-Link device	76
Request channel	97
Required background knowledge	7
Response channel	98
Rules for accessing the Modbus registers	71

S

Safety instructions	7
Safety symbols on the device	7
Security mode	11
Service	
factoryreset	116
getdata	117
getdatamulti	118
getidentity	119
getsubscriptioninfo	120
gettree	117
iolreadacyclic	121
iolwriteacyclic	121
reboot	121
setblock	122
setdata	122
setelementinfo	123
signal	123
subscribe	124
unsubscribe	124
Set application identification	48
Set IO-Link master	63
Set IO-Link ports	64
Set-up	21
Single Port Access	95
Status LEDs	19
Subscribe to events	54
Supported function codes	71

T

Tampering with the unit	8
Technical data	80

U

Update firmware	75
Use acyclic services	73

V

Visual indication	12
Voltage supply	20

W

Write output data of individual IO-Link ports	68
Write output data of several IO-Link ports	67