Operating Instructions

IO-Link Master with IoT Interface
DataLine
4 Ports
IP 65 / IP 66 / IP 67

AL1350

HW Revision: AA
Firmware: 2.3.x
LR DEVICE: 1.5.0.x

English

80284128 / 02   09 / 2019

Contents

1 Preliminary note 5

1.1 Legal and copyright information 5
1.2 Purpose of the document 5
1.3 Explanation of Symbols 5
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 Internet of Things (IoT) 11
4.1.3 Security mode 11
4.1.4 Parameter setting 11
4.1.5 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 IoT ports 15
6.3 IO-Link ports 16
6.3.1 Connect IO-Link devices for Class A operation 16
6.3.2 Connect IO-Link devices for Class B operation 17
6.4 Connect the device 18

7 Operating and display elements 19

7.1 Overview 19
7.2 LED indicators 20
7.2.1 Status LEDs 20
7.2.2 IoT interface 20
7.2.3 IO-Link ports (Class A) 21
7.2.4 Voltage supply 21

8 Set-up 22
8.1 Read device and diagnostic information 23

9 Configuration 24
9.1 LR DEVICE 25
9.1.1 Remarks 26
9.1.2 IoT: Configure IP settings 27
9.1.3 IoT: Configure security mode 28
9.1.4 IoT: Configure the interface to LR AGENT or LR SMARTOBSERVER 29
9.1.5 IO-Link ports: Activate data transfer to LR AGENT or LR SMARTOBSERVER 29
9.1.6 IO-Link ports: Configure operating mode 30
9.1.7 IO-Link ports: Set the device validation and data storage 31
9.1.8 IO-Link ports: Configuration of fail-safe values 32
9.1.9 Info: Show device information 32
9.1.10 Firmware: Reset device to factory settings 33
9.1.11 Firmware: Reboot the device 33
9.1.12 Configure IO-Link devices 34
9.2 ifm IoT Core 35
9.2.1 First steps 35
9.2.2 General functions 36
9.2.3 Configure the IoT interface 36
9.2.4 IoT interface: Configure security mode 37
9.2.5 Configure IO-Link ports 40
9.2.6 Configure IO-Link devices 42
9.2.7 Set application identification 44
9.2.8 Read / write cyclic process data 45
9.2.9 Control IO-Link master 46
9.2.10 Read diagnostic data of the AL1350 48
9.2.11 Read device information of the IO-Link master 49
9.2.12 Read information about IO-Link devices 49
9.2.13 Subscribe to events 50
9.2.14 MQTT support 52
9.2.15 Programmers' notes 53

10 Maintenance, repair and disposal 58
10.1 Cleaning process 58
10.2 Update firmware 59
10.3 Replace IO-Link device 60

11 Factory settings 61

12 Accessories 62

13 Appendix 63
13.1 Technical data 64
13.1.1 Application 64
13.1.2 Electrical data 64
13.1.3 Inputs / outputs 64
13.1.4 Inputs 65
13.1.5 Outputs 65
13.1.6 Interfaces 65
13.1.7 Operating conditions 66
13.1.8 Approvals / tests 66
13.1.9 Mechanical data 66
13.1.10 Electrical connection 67

13.2 ifm IoT Core 68
13.2.1 Overview: IoT profile 69
13.2.2 Overview: IoT types....................74
13.2.3 Overview: IoT services....................75

14 Index 84

1 Preliminary note

Content

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

33203

1.1 Legal and copyright information

33117

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 - IoT core gateway (DataLine) 4 port IP 65 / IP 66 / IP 67" (art. no.: AL1350).

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
ℹ	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: HW revision	05 / 2019
02	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

34209

The IO-Link master serves as a gateway between intelligent IO-Link devices and the IoT core network. The device is designed for use without a control cabinet in the plant construction.

3.2 Prohibited use

34228

The device may not be used beyond the limits of the technical data (→ **Technical data** (→ S. 64))!

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
Internet of Things (IoT) 11
Security mode 11
Parameter setting 11
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 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

4.1.3 Security mode

54697

The IoT interface offers the following optional sercurity functions:

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

4.1.4 Parameter setting

34210

The device provides the following configuration options:

- Parameter setting of the IO-Link master of the AL1350 with LR DEVICE parameter setting software, IoT core projection software or ifm IoT-Core services.
- Parameter setting of the connected IO-Link devices (sensors, actuators) with LR DEVICE parameter setting software, IoT core projection software or ifm IoT-Core services
- Storage of parameter sets of the connected IO-Link devices for automatic recovery (data storage)

4.1.5 Visual indication

34192

The device has the following visual indicators:

- Status and error indication of the gateway, of the IoT core 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
IoT ports....15
IO-Link ports....16
Connect the device....18

33805

6.1 Notes

51957

❗ A qualified electrician must connect the unit.

- ► The national and international for 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).

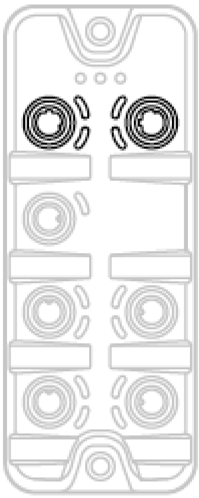
Wiring: → **Technical data** (→ S. 64)

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 30 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 30 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 IoT ports

52272



- ► Connect the unit via the M12 socket X21 and/or X22 with the IoT core network (e.g. laptop/PC with LR DEVICE parameter setting software, laptop/PC with LR SMARTOBSERVER monitoring software, PC/laptop with http request compatible software)
  - ▪ Tightening torque: 0.6...0.8 Nm
- ► To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 or higher (→ **Accessories** (→ S. 62)).
- ► Cover unused sockets with M12 protective caps (art. no.: E73004).
  - ▪ Tightening torque 0.6...0.8 Nm

6.3 IO-Link ports

51958

The IO-Link ports of the AL1350 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.: E73004).

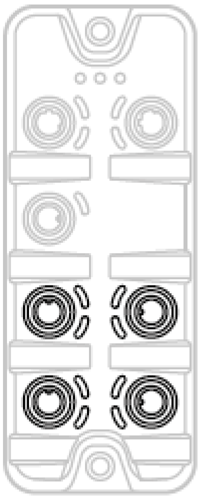
- ▪ Tightening torque 0.6...0.8 Nm

6.3.1 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 or higher (→ **Accessories** (→ S. 62)).

6.3.2 Connect IO-Link devices for Class B operation

51960

Notes on wiring:

- For Class B operation, the IO-Link device must be supplied with an additional auxiliary voltage UA using a Y connection cable.

WARNING!

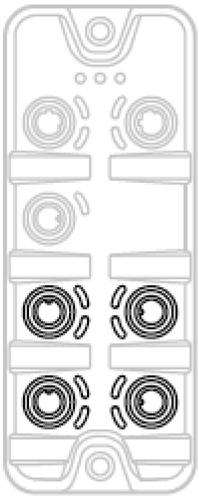
Non-compliance with the electrical separation of the circuits

> Risk of fire!

▶ Ensure that the external supply UA is galvanically separated from the circuit of the IO-Link Master by assuring basic insulation (according to IEC 61010-1, secondary circuit with 30 V DC maximum, supplied from mains circuit up to 300 V of overvoltage category II).

▶ Ensure that the IO-Link devices and the connection technology support the galvanic separation.

In case of operation as port class B, the additional digital input of the IO-Link port (pin 2) is not available!



▶ Connect the connectors of the IO-Link devices via a Y connection cable with the M12 sockets of the IO-Link ports X01...X04.

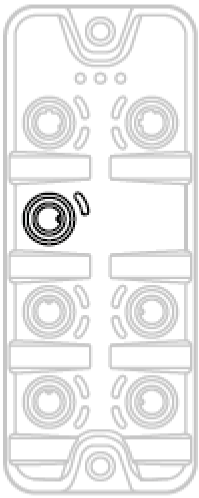
▶ Connect the Y cable to 24 V DC (20...30 V SELV/PELV)

- Tightening torque: 0.6...0.8 Nm

▶ To connect the devices, use M12 connectors with protection rating IP 65 / IP 66 / IP 67 or higher (→ **Accessories** (→ S. 62))!

6.4 Connect the device

33882



- ► Disconnect power.
- ► Connect the IO-Link Master via M12 socket X31 to 24 V DC (20...30 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 or higher (→ **Accessories** (→ S. 62)).

! 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

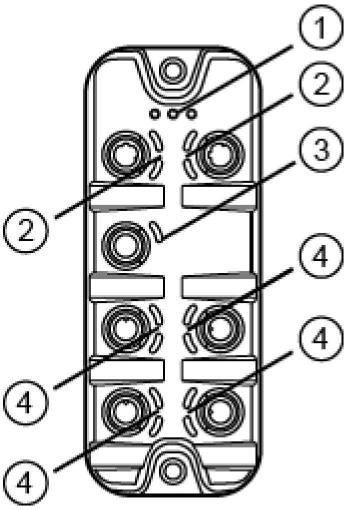
Content

Overview..........19
LED indicators..........20

34063

7.1 Overview

52274



1. Status LEDs RDY and IoT
→ **Status LEDs** (→ S. 20)

2. L/A and 10/100 MBits/s status LEDs of the IoT interface (X21/X22)
→ **IoT interface** (→ S. 20)

3. IOL and DI status LEDs of the IO-Link port (X01...X04)
→ **IO-Link ports (Class A)** (→ S. 21)

4. PWR status LED of the voltage supply (X31)
→ **Voltage supply** (→ S. 21)

7.2 LED indicators

34047

The device only has the following LED indicators:

7.2.1 Status LEDs

52237

The RDY LED indicates the status of the gateway.
The IoT LED indicates the status of the connection to the LR SMARTOBSERVER.

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
IoT	green	off	No connection to the LR SMARTOBSERVER
		on	Connection to the LR SMARTOBSERVER activated

7.2.2 IoT interface

52238

Each IoT port (X21, X22) has the 2 L/A and 10/100 MBits/s LEDs. The LEDs indicate the status of the Ethernet connection.

Status LED			Description
L/A	green	on	Ethernet connection established
		flashes	Data is transmitted via the Ethernet interface.
		off	No Ethernet connection
10/100 Mbits/s	yellow	on	100 MBit/s
		off	10 MBit/s

7.2.3 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

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.

8 Set-up

Content

Read device and diagnostic information 23

52239

When the supply voltage is switched on, the AL1350 starts with the factory settings. The display elements signal the current operating mode (→ **Operating and display elements** (→ S. 19)).

To enable parameter setting of the AL1350 via the IoT core network, the IoT interface must be configured according to the network environment.

► Connect AL1350 via the ports X21/X22 to the IoT core network.

► Configure the IoT interface (→ **IoT: Configure IP settings** (→ S. 27)).

> IoT interface has valid IP settings.

> User can set the parameters of the AL1350.

Further steps:

- Set the parameters of the AL1350 (→ **Configuration** (→ S. 24)).

8.1 Read device and diagnostic information

52240

The AL1350 has an integrated web server. The component provides device and diagnostic information of the IO-Link master and the connected IO-Link devices.

► Connect laptop/PC and AL1350 via the Ethernet interface.

► Start web browser.

► Enter the IP address of the AL1350 into the address field of the browser and confirm with [ENTER].

> 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 port
[Mode]	Operating mode of the IO-Link port
[Comm. Mode]	Baud rate of the IO-Link port
[MasterCycleTime]	Cycle time
[Vendor ID]	IO-Link ID of the manufacturer of the IO-Link device
[Device ID]	IO-Link ID of the IO-Link device
[Name]	Article number of the IO-Link device ▪ For ifm articles: This article number is stored along with a link to the product page on the ifm website.
[Serial]	Serial number of the IO-Link device

- 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 overvoltages
[Undervoltage]	Number of undervoltages
[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 bootloader
[Fieldbus Firmware]	Version of the IoT core firmware

9 Configuration

Content

LR DEVICE.....25
ifm IoT Core.....35

33858

9.1 LR DEVICE

Content

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

33692

On delivery, the AL1350 is configured with the factory settings (→ **Factory settings** (→ S. 61)).

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

9.1.1 Remarks

Content

Offline parameter setting 26
VPN connection 26

34180

Offline parameter setting

34060

The AL1350 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 AL1350 (OFFLINE mode). The configuration created in this way can be stored as a file (*.lrp) and loaded to the AL1350 and activated at a later date.

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

VPN connection

34382

! An active VPN connection blocks the access of the parameter setting software LR DEVICE to the IoT core interface of the AL1350.

► Deactivate the VPN connection in order to be able to access the AL1350 with the LR DEVICE.

9.1.2 IoT: Configure IP settings

34049

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. 61)).

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

54680

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 the interface to LR AGENT or LR SMARTOBSERVER

34048

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: AL1350	

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.

9.1.5 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.

9.1.6 IO-Link ports: Configure operating mode

33694

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 ... 132800	1 microsecond ... 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.

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

33697

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.

9.1.8 IO-Link ports: Configuration of fail-safe values

52242

The user can set the fail-safe values of the outputs of the IO-Link ports X01...X04. The fail-safe values will be activated in case of an interruption of the IoT core connection.

To configure the fail-safe values:

► 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 value of the output for operating mode "digital output (DO)"	[Reset]	OFF
		[Old]	Old value
		[Set]	ON
[Fail-safe IO-Link]*	Fail-safe value of the output for operating mode "IO-Link"	[Off]	No fail-safe
		[Reset]	Fail-safe: OFF
		[Old]	Fail-safe: old value
		[Pattern]	Fail-safe: byte sequence

* ... parameter can only be changed if the IoT core controller is disconnected

► Save changed values on the device.

9.1.9 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	AL1350
[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	

9.1.10 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.

9.1.11 Firmware: Reboot the device

33832

When rebooting the device, all settings are kept.

To restart the AL1350:

► 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.

9.1.12 Configure IO-Link devices

52351

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 AL1350.

> Operating mode of the IO-Link port is "IO-Link" (→ **IO-Link ports: Configure operating mode** (→ S. 30)).

1 Select IO-Link master

▶ Start LR DEVICE.

▶ Update IODD file library
OR:
Import IODD file of the IO-Link device manually.

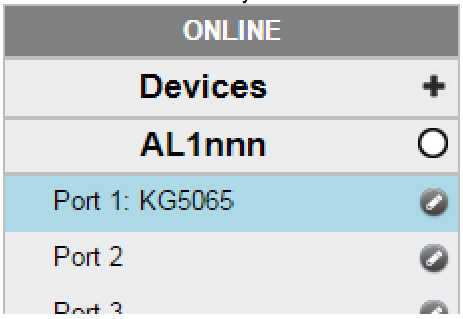
▶ Scan network for devices.

> 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 35
General functions 36
Configure the IoT interface 36
IoT interface: Configure security mode 37
Configure IO-Link ports 40
Configure IO-Link devices 42
Set application identification 44
Read / write cyclic process data 45
Control IO-Link master 46
Read diagnostic data of the AL1350 48
Read device information of the IO-Link master 49
Read information about IO-Link devices 49
Subscribe to events 50
MQTT support 52
Programmers' notes 53

52244

General notes on the ifm IoT Core: → **Programmers' notes** (→ S. 53)

9.2.1 First steps

52245

To read the device description of the AL1350:

► Send the following POST request to the AL1350:
`{"code":"request","cid":-1,"adr":"gettree"}`

> AL1350 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 AL1350 is of type `device` (→ **Overview: IoT types** (→ S. 74)).

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 AL1350
../setelementinfo	Write the uid of the AL1350

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 the IoT interface

52247

The AL1350 is integrated into the IT network via the IoT interface.

Substructure: `iotsetup`

Available data points:

Name	Description	Access
../smobip	IP address of the LR SMARTOBSERVER	rw
../smobport	Port number of the LR SMARTOBSERVER	rw
../smobinterval	Cycle time for the data transfer to the 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 blockwise

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
../security/securitymode	active security mode	rw
../security/password	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. 78)).

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. 39)).

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

54701

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 sonsists 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":"SW9UNGlmbQ=="}
}
```

- Response:

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

9.2.5 Configure IO-Link ports

52248

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 IoT core plc is interupted

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

52344

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

► Calcuate the number of reading steps n: n = first integer value to which the following applies: g < n*h
Example: n= 3, because 550 < 3*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 resquest: 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": "aWZtfgIAAABBTDF4NXhfY25faXRfdDIuMi43Nw..."}

9.2.6 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
../iolreadacyclic	Read a parameter of an IO-Link device (acyclic)
../iolwriteacyclic	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 iolwriteacyclicdata 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/iolwriteacyclic",
"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
}
```

9.2.7 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 AL1350 for the representation in the LR SMARTOBSERVER.

Solution: Change the parameter [Application Tag] with the setdata service to the value [AL1350].

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":"AL1350"}
}
```

- Response:

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

9.2.8 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
../pin2in	Value of the digital input on pin 2 of the IO-Link port	r
../iolinkdevice/pdin	Value of the IO-Link input on pin 4 of the IO-Link port	r
../iolinkdevice/pdout	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 = 0b**11110010**01

Temperature value: 0b11110010 = 242

Therefore: The current temperature value is 24.2 °C.

9.2.9 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

52252

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": "aWZtfgIAAABBTDF4NXhfY25faXRfdDIuMi43Nw..."}

► 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": {}}

9.2.10 Read diagnostic data of the AL1350

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
}
```

9.2.11 Read device information of the IO-Link master

52254

To read the device information of the AL1350:

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 AL1350 can be read with the `getidentity` service (→ **Service: getidentity** (→ S. 78)).

9.2.12 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

9.2.13 Subscribe to events

52255

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: ▪ 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/unsubsribe	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
}
```

9.2.14 MQTT support

54699

The AL1350 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: xxx

- Request:

```
{
"code":"request",
"cid":-1,
"adr":"iolinkmaster/port[1]/portevent/datachanged/subscribe",
"data":{
"callback":"mqtt://172.18.87.50:1883/abc",
"datatosend":["processdatamaster/temperature"}
}
```

- Response:

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

9.2.15 Programmers' notes

Content

IoT Core: General information 53
Access the ifm IoT Core 54
IoT Core: Diagnostic codes 57

34229

IoT Core: General information

52256

The DataLine device family has an IoT Core. The IoT Core allows the user to address the AL1350 from IT networks via a REST API and to integrate it into Internet-of-Things applications.

A device description is stored on the AL1350. 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
```

Description	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. 57))

Example: GET request

54033

Request (via browser):

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

Response:

```
{
"cid":-1,
"data":{"value":"AL1350"},
"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 vergebbare Kennung	
adr	data_point	Data point of the element tree which is to be accessed	
	service	Service to be performed (→ **Overview: IoT services** (→ S. 75))	
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. 57))

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

Example: POST request

54035

Request:

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

Response:

```
{
"cid":4711,
"data":{"value":"AL1350"},
"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

10 Maintenance, repair and disposal

Content

Cleaning process 58
Update firmware 59
Replace IO-Link device 60

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

52258

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

- IoT Core: → **Example: Update firmware** (→ S. 47)
- 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 IoT core 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 AL1350

▶ 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 AL1350.

> IO-Link master copies parameter values from the data memory to the new IO-Link device.

11 Factory settings

33849

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

Parameter	Factory setting
[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
[Host name]	blank
Data storage	empty

12 Accessories

33870

List of accessories of AL1350: → www.ifm.com > Product page > Accessories

13 Appendix

Content

Technical data 64
ifm IoT Core 68

33879

13.1 Technical data

Content

Application 64
Electrical data 64
Inputs / outputs 64
Inputs 65
Outputs 65
Interfaces 65
Operating conditions 66
Approvals / tests 66
Mechanical data 66
Electrical connection 67

34188

13.1.1 Application

33878

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

13.1.2 Electrical data

33808

Electrical data	
Operating voltage [V]	20...30 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...30
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

52260

Interfaces	
Communication interface	Ethernet; IO-Link
Communication interface	IO-Link; TCP/IP; TCP/IP JSON
Ethernet	
Transmission standard	10Base-T; 100Base-TX
Transmission rate [MBit/s]	10; 100
Protocol	DCP, DHCP, Auto IP
Factory settings	▪ IP address: 169.254.X.X ▪ Subnet mask: 255.255.0.0 ▪ 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

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
Pollution Degree	2

13.1.8 Approvals / tests

33877

Approval / tests	
EMC	▪ EN 61000-6-2 ▪ EN 61000-6-4
MTTF [Years]	90

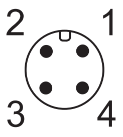
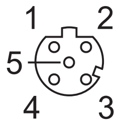
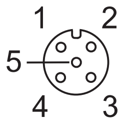
13.1.9 Mechanical data

34050

Mechanical data	
Weight [g]	295
Materials	Housing: PA; socket: brass nickel-plated

13.1.10 Electrical connection

52261

Voltage supply IN X31	
Plug and socket connection	M12
Wiring	 1: + 24 V DC (US) 2: - 3: GND (US) 4: -
IoT port X21, X22	
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 ifm IoT Core

Content

Overview: IoT profile 69
Overview: IoT types 74
Overview: IoT services 75

33803

13.2.1 Overview: IoT profile

Content

Profile: blob 69
Profile: deviceinfo 70
Profile: devicetag 70
Profile: iolinkdevice_full 71
Profile: iolinkmaster 71
Profile: network 72
Profile: parameter 72
Profile: processdata 72
Profile: service 72
Profile: software 73
Profile: software/uploadedablesoftware 73
Profile: timer 73

34054

Profile: blob

52264

Element (identifier)	Characteristics	Mandatory	Comments
blobname	▪ 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	▪ type = structure ▪ profile = deviceinfo		characterises the element as device information
../devicename	type = data	optional	
../devicefamiliy	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	▪ 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	▪ 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	▪ type = structure ▪ profile = iolinkmaster		Executable service
../mode	▪ type = data ▪ profile = parameter	mandatory	
../comspeed	▪ type = data ▪ profile = parameter	mandatory	
../mastercycletime_actual	▪ type = data ▪ profile = parameter	mandatory	
../mastercycletime_preset	▪ type = data ▪ profile = parameter	mandatory	
../validation_datastorage_mode	▪ type = data ▪ profile = parameter	mandatory	
../validation_vendorid	▪ type = data ▪ profile = parameter	mandatory	
../validation_deviceid	▪ type = data ▪ profile = parameter	mandatory	
../additionalpins_in	▪ type = data ▪ profile = processdata	optional	
../additionalpins_out	▪ type = data ▪ profile = processdata	optional	
../portevent	▪ type = data	mandatory	
../iolinkdevice	▪ type = structure ▪ profile = iolinkdevice_full	mandatory	

Profile: network

52266

Element (identifier)	Characteristics	Mandatory	Comments
network	▪ type = structure ▪ profiles = deviceinfo		Characterises the element as device information
../macaddress	▪ type = data ▪ profile = parameter	mandatory	
../ipaddress	▪ type = data ▪ profile = parameter	optional	
../ipv6address	▪ type = data ▪ profile = parameter	mandatory	
../subnetmask	▪ type = data ▪ profile = parameter	mandatory	
../ipdefaultgateway	▪ type = data ▪ profile = parameter	mandatory	
../dhcp	▪ type = data ▪ profile = parameter	optional	
../ipversion	▪ type = data ▪ profile = parameter	optional	
../hostname	▪ type = data ▪ profile = parameter	optional	
../autonegotiation	▪ type = data ▪ profile = parameter	optional	
../portspeed	▪ 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	▪ type = service ▪ profile = service		Executable service

Profile: software

34223

Element (identifier)	Properties	mandatory	Comments
software	▪ 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/uploadedablesoftware

52267

Element (identifier)	Characteristics	Mandatory	Comments
software	▪ type = structure ▪ profiles = software/uploadablesoftware		Software that can be loaded to the device via the IoT Core
../lastinstall	type = data	optional	
../installhistory	type = data	optional	
../container	▪ 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	▪ type = structure ▪ profile = timer		Executable service
../counter	▪ type = data ▪ profile = parameter	mandatory	
../interval	▪ type = data ▪ profile = parameter	optional	
../start	type = service	optional	
../stop	type = service	optional	

13.2.2 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

13.2.3 Overview: IoT services

Content

Service: factoryreset 75
Service: gettree 76
Service: getdata 76
Service: getdatamulti 77
Service: getidentity 78
Service: getsubscriptioninfo 79
Service: iolreadacyclic 80
Service: iolwriteacyclic 80
Service: reboot 80
Service: setblock 81
Service: setdata 81
Service: setelementinfo 82
Service: signal 82
Service: subscribe 83
Service: unsubscribe 83

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 AL1350 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 AL1350 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		AL1350
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","ipdefaultgatewa
y":"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 AL1350.

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. unsubcribe 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 54
Accessories 62
Appendix........63
Application 64
Approvals / tests 66

C

Cleaning process........58
Communication, parameter setting, evaluation........11
Configuration 24
Configure IO-Link devices 34, 42
Configure IO-Link ports 40
Configure the IoT interface........36
Connect IO-Link devices for Class A operation 16
Connect IO-Link devices for Class B operation 17
Connect the device........18
Control IO-Link master 46

D

Digital inputs 12

E

Electrical connection........14, 67
Electrical data 64
Example
  Activate security mode........38
  Change name of the IO-Link master 44
  Change the parameter value of an IO-Link device........43
  Clone the Data Storage of an IO-Link port........41
  GET request........54
  POST request........56
  Publish the temperature to an MQTT broker........52
  Read process data of an IO-Link device 45
  Read several parameter values of the IO-Link master simultaneously 48
  Read the parameter value of an IO-Link device........42
  Request with authentication........39
  reset password........39
  Subscribe to event........51
  Update firmware........47
Explanation of Symbols........5

F

Factory settings 61
Firmware
  Reboot the device 33
  Reset device to factory settings........33
First steps 35
Function 10

G

General 7
General functions........36
GET request 54

I

ifm IoT Core 35, 68
Info
  Show device information........32
Inputs 65
Inputs / outputs 64
Intended use 9
Interfaces........65
Internet of Things (IoT) 11
IO-Link 11
IO-Link ports 16
  Activate data transfer to LR AGENT or LR SMARTOBSERVER 29
  Configuration of fail-safe values........32
  Configure operating mode........30
  Set the device validation and data storage........31
IO-Link ports (Class A) 21
IO-Link supply........12
IoT
  Configure IP settings........27
  Configure security mode 28
  Configure the interface to LR AGENT or LR SMARTOBSERVER........29
IoT Core
  Diagnostic codes........57
  General information........53
IoT interface........20
  Configure security mode 37
IoT ports........15
IT safety 8

L

LED indicators 20
Legal and copyright information 5
LR DEVICE........25

M

Maintenance, repair and disposal 58
Mechanical data........66
Modification history........6
Mount the device 13
Mounting 13
MQTT support........52

N

Note
  Security mode........37
Notes........14

O

Offline parameter setting 26
Operating and display elements........19
Operating conditions........66
Outputs 65
Overview........19
  IoT profile........69
  IoT services 75
  IoT types........74

P

Parameter setting 11
Permitted use........9
POST request........55
Preliminary note........5
Profile
  blob 69
  deviceinfo........70

devicetag 70
iolinkdevice_full 71
iolinkmaster 71
network 72
parameter 72
processdata 72
service 72
software 73
software/uploadedablesoftware 73
timer 73
Programmers' notes 53
Prohibited use 9
Purpose of the document 5

R

Read / write cyclic process data 45
Read device and diagnostic information 23
Read device information of the IO-Link master 49
Read diagnostic data of the AL1350 48
Read information about IO-Link devices 49
Remarks 26
Replace IO-Link device 60
Required background knowledge 7

S

Safety instructions 7
Safety symbols on the device 7
Security mode 11
Service
factoryreset 75
getdata 76
getdatamulti 77
getidentity 78
getsubscriptioninfo 79
gettree 76
iolreadacyclic 80
iolwriteacyclic 80
reboot 80
setblock 81
setdata 81
setelementinfo 82
signal 82
subscribe 83
unsubscribe 83
Set application identification 44
Set-up 22
Status LEDs 20
Subscribe to events 50

T

Tampering with the unit 8
Technical data 64

U

Update firmware 59

V

Visual indication 12
Voltage supply 21
VPN connection 26