IO-Link Wireless System Extensions

Specification

Version 1.1
March 2018

Order No: 10.112

File name: **IO-Link_Wireless_System_10112_V1.1_Mar18**

This specification has been prepared by the IO-Link Wireless technology subgroup of the IO-Link community for review.

Any comments, proposals, requests on this document are appreciated through the IO-Link CR database www.io-link-projects.com. Please provide name and email address.
Login: IO-Link.Wireless
Password: Comments

Important notes:

NOTE 1 The IO-Link Consortium Rules shall be observed prior to the development and marketing of IO-Link products. The document can be downloaded from the www.io-link.com portal.

NOTE 2 Any IO-Link device shall provide an associated IODD file. Easy access to the file and potential updates shall be possible. It is the responsibility of the IO-Link device manufacturer to test the IODD file with the help of the IODD-Checker tool available per download from www.io-link.com.

NOTE 3 Any IO-Link devices shall provide an associated manufacturer declaration on the conformity of the device with this specification, its related IODD, and test documents, available per download from www.io-link.com.

Conventions:

In this specification, the following key words (in **bold** text) will be used:

may: indicates flexibility of choice with no implied preference.
should: indicates flexibility of choice with a strongly preferred implementation.
shall: indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformity with this specification.

Publisher:
IO-Link Community
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 721 / 96 58 590
Fax: +49 721 / 96 58 589
E-mail: info@io-link.com
Web site: www.io-link.com

CONTENTS

Introduction

General

The base technology of IO-Link™[1]) is subject matter of the international standard IEC 61131-9 (see REF 2). It specifies a single-drop digital communication interface technology for small sensors and actuators – named SDCI, which extends the traditional switching input and output interfaces as defined in IEC 61131-2 towards a point-to-point communication link using coded switching. This technology enables the cyclic exchange of digital input and output process data between a W-Master and its associated W-Devices (sensors, actuators, I/O terminals, etc.). The W-Master can be part of a fieldbus communication system or any stand-alone processing unit. The technology enables also the acyclic transfer of parameters to W-Devices and the propagation of diagnosis information from the W-Devices to the upper-level automation system (controller, host) via the W-Master/gateway.
Therefore, the market demand for the extension of this technology towards wireless transmission was raised. This document provides the necessary changes and extensions to the basic IO-Link interface and system standard for wireless communication including the radio characteristic, air interface, frequencies, message/frame types, and pairing mechanism as well as the necessary configuration management and the changes of state machines compared to IO-Link Interface and System Specification. Figure 1 shows its relationships to international fieldbus, wireless communications, EMC, and power source standards.

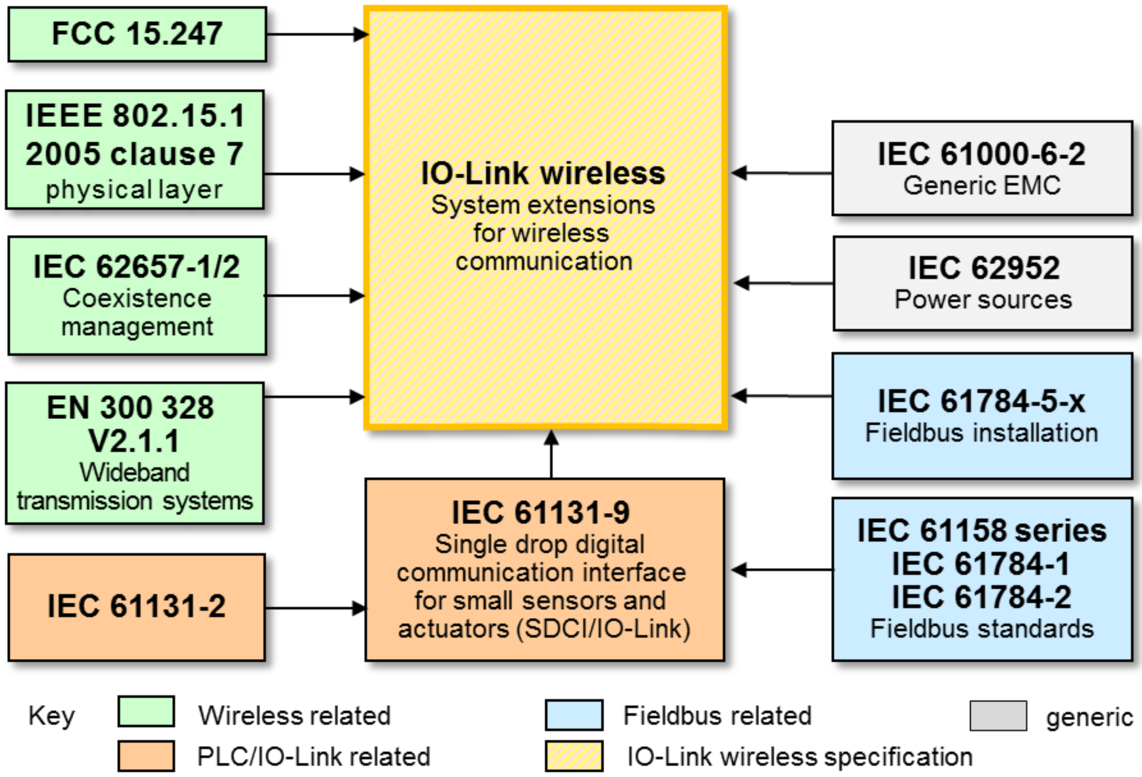


Figure 1 Relationships of this document to standards

IEC 61131-9 is part of a series of standards on programmable controllers and the associated peripherals and should be read in conjunction with the other parts of the series.

Terms of general use are defined in IEC 61131-1 or in the IEC 60050 series. More specific terms are defined in each part.

Conformity with this document cannot be claimed unless the requirements of Annex I are met.

1 IO-Link™ and IO-Link wireless™ are trade names of the "IO-Link Community". This information is given for the convenience of users of this specification and does not constitute an endorsement by the IO-Link Community of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Community".

The main characteristics of the IO-Link Wireless technology are:

- The application interface for cyclic (Process Data) and acyclic data (On-request Data) is compatible to IO-Link; from the user perspective, it is a transparent view on W-Devices.
- A W-Master can handle up to 5 transmission tracks in parallel, each track can handle a maximum of 8 W-Devices, thus supporting up to 40 W-Devices per W-Master.
- Up to 3 W-Master can be placed in a cell, yielding a maximum of 120 W-Devices per W-Master cell.
- A scan service is available for discovery of yet unpaired W-Devices.
- A pairing service is provided to assign W-Devices to a W-Master, corresponding to a logical cable connection.
- There are no limitations for typical relative movement speeds of W-Devices within a single W-Master cell.
- Controlled roaming between multiple W-Master cells is supported by a dedicated handover mechanism.
- A minimum transmission cycle time of 5 ms can support high-speed wireless applications with a payload of up to 32 octets.
- IO-Link Wireless also supports mechanisms for low energy W-Devices.
- IO-Link wireless utilizes in this version radios for the 2.4 GHz ISM band, divided to frequency channels with a distance of 1 MHz.
- Frequency Hopping changes the frequency channels for each transmission as a measure against interference, yielding a PER of 10^{-9} which is similar to a wired connection.
- W-Coexistence with other wireless systems (e.g. WLAN) is achieved with a blacklisting mechanism.
- To comply with regulatory standards, transmission power is limited to ≤ 10 dBm (10 mW) EIRP, still yielding a range of up to 20 m in case of a W-Master cell with one track. In case of more than one track, 10 m can be achieved. These figures are dependent on the machine environment.
- Each transmission track in a W-Master can use its own narrow-band transceiver and dedicated antenna or all of them can use a single shared transceiver and/or antenna.

0.2 Patent declaration

The IO-Link Community draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning the point-to-point wireless communication interface for small sensors and actuators as follows, where the [xx] notation indicates the holder of the patent right

DE 19947344 A2	[ABB]	[1]	SENSOR MIT DRAHTLOSER DATENÜBERTRAGUNG MIT GERINGER LEISTUNGSAUFNAHME
DE 10153462 A1	[ABB]	[2]	Verfahren zum Betrieb eines Systems mit mehreren Knoten und einer Basisstation gemäß TDMA
DE 10334873 A1	[ABB]	[3]	METHOD FOR OPERATING A SYSTEM COMPRISING A PLURALITY OF NODES AND A BASE STATION ACCORDING TO TDMA, AND ASSOCIATED SYSTEM
DE 102006032354 A1	[Festo]	[4]	Funk-Steuerungssystem

IO-Link Community takes no position concerning the evidence, validity and scope of these patent rights. The holders of these patents rights have assured the IO-Link Community that they are willing to negotiate licenses either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with the IO-Link Community.
Information may be obtained from:

[1 - 3x]	ABB Automation GmbH Heidelberg
4	Festo & Co KG, Esslingen

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. For example, they may be subject of patents listed in [1-3] or [4]. The IO-Link Community shall not be held responsible for identifying any or all such patent rights.

The IO-Link Community maintains on-line data bases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

1 Scope

This document specifies IO-Link Wireless communication for factory automation. Different aspects of communication are realized by different communication layers based on the following layer model.

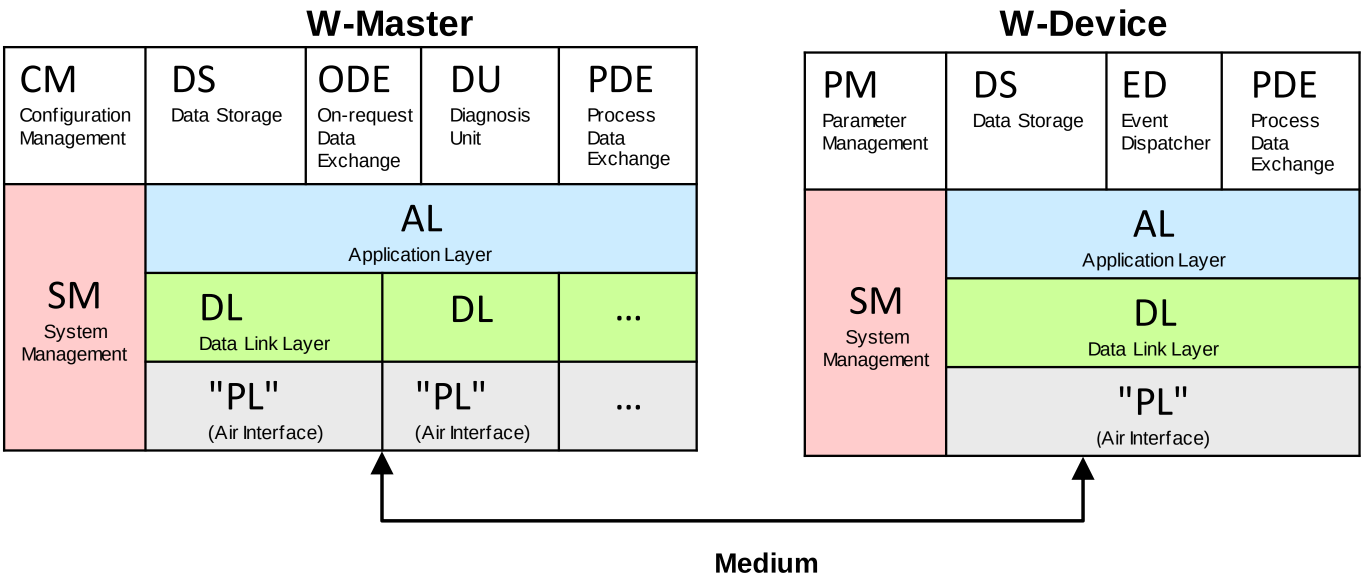


Figure 2 Logical structure of W-Master and W-Device

The "PL" (Air interface) for IO-Link wireless includes the Physical Layer as well as the wireless mechanisms (e.g. pairing, blacklisting, …) with all aspects related to the radio:

- Antenna aspects
- Radio transceivers
- Radio frequencies
- Bidirectional data transmission via downlink and uplink (W-Sub-cycle)
- Media access and frequency hopping patterns
- W-Sub-cycle structures

Following elements specify the Data Link Layer (DL):

- Data scheduling (DL-A)
- Data handling (DL-B)

Following elements specify the Application Layer (AL):

- Data exchange

System Management (SM) realizes:

- Operating states
- Pairing functionality for W-Master and its W-Devices during commissioning and replacement
- Parameterization (download of W-Parameters)

In addition, this document provides the necessary changes and extensions to the IO-Link Interface and System Specificationfor the operation of wireless communication.

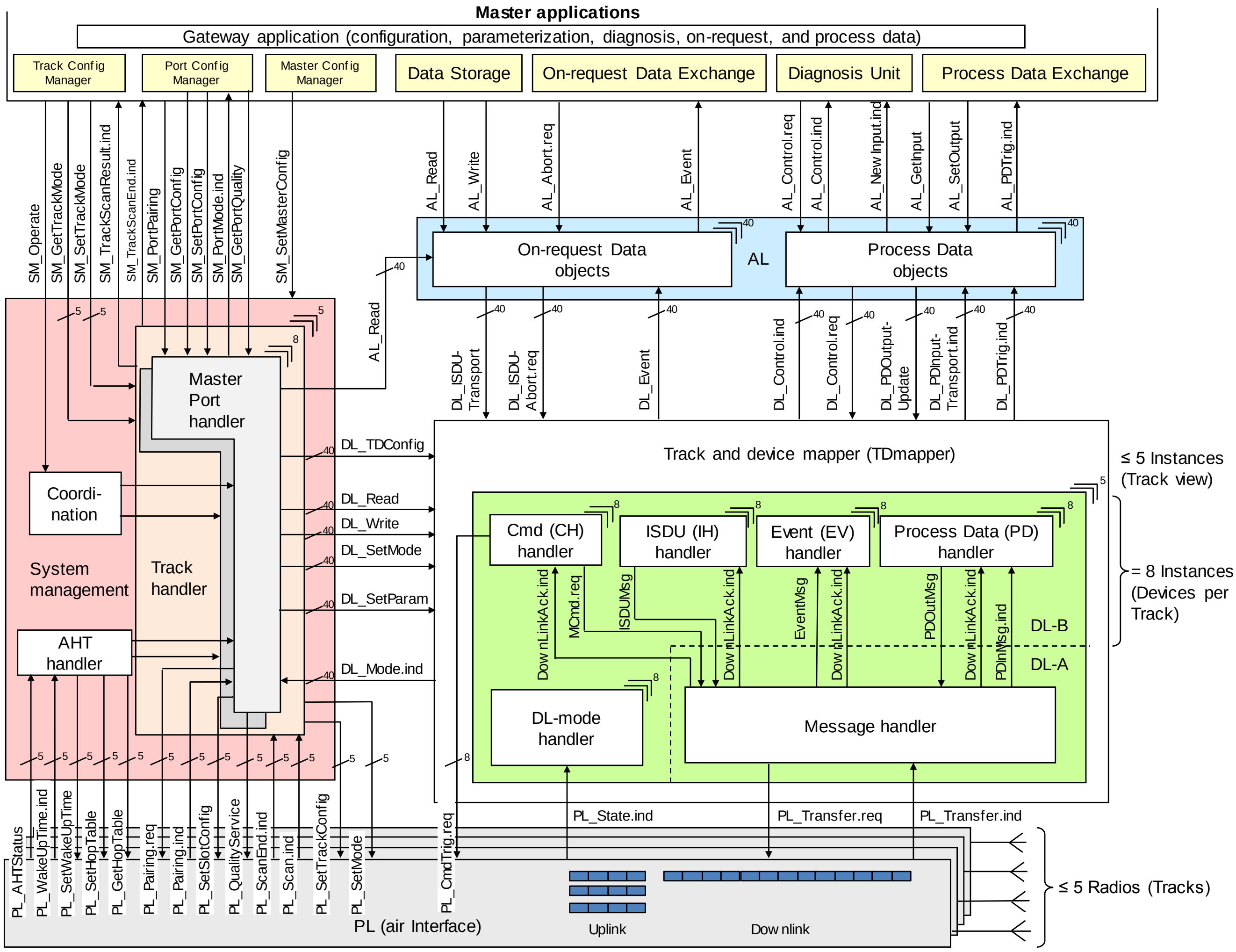


Figure 3 Detailed overview of the W-Master

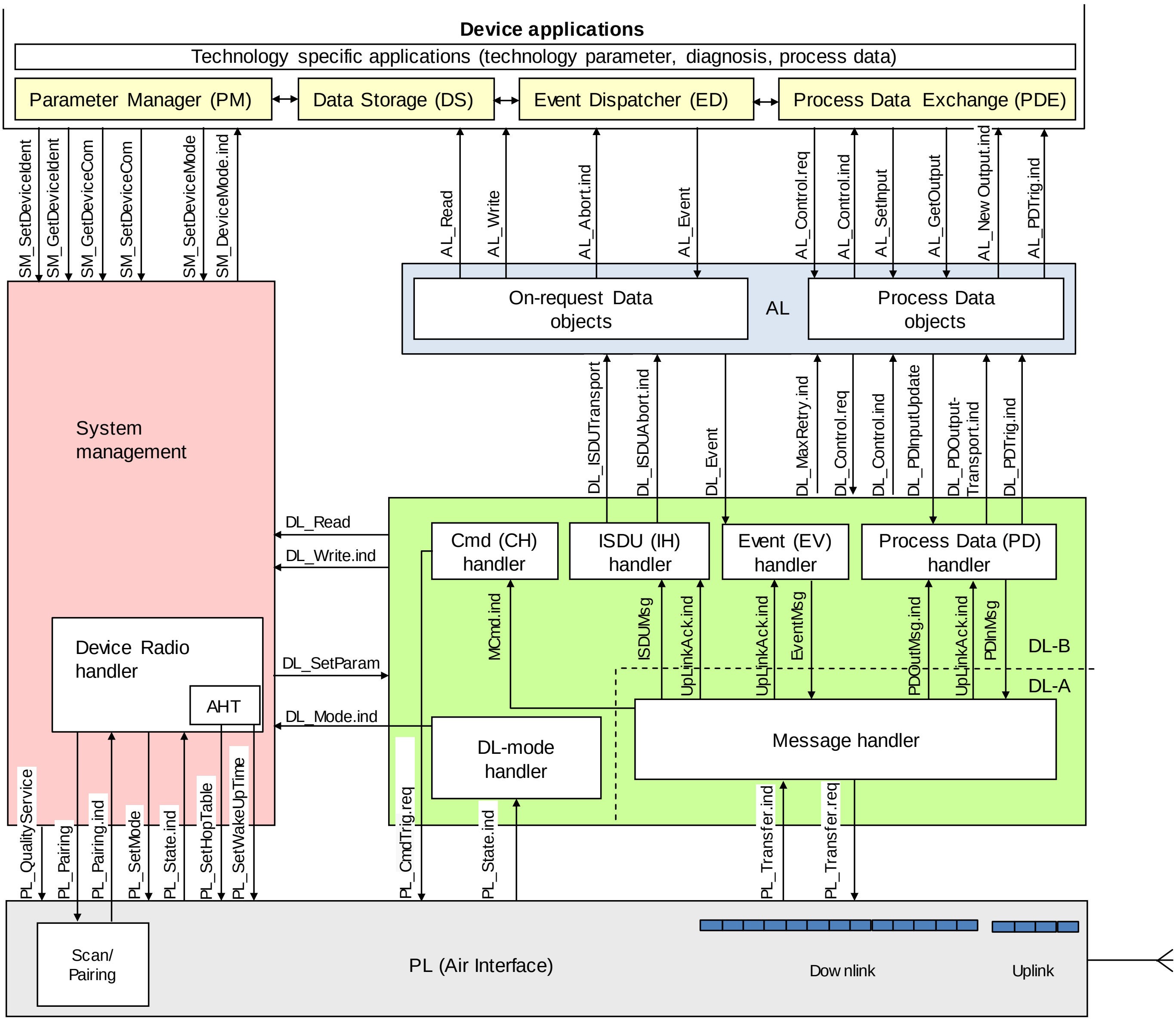


Figure 4 Detailed overview of the W-Device

1.1 Structure of the document

The document is organized in an almost identical structure than the IO-Link Standard. If possible, the IO-Link Standard is referenced instead of repeating passages.
Every time a system element is introduced that is later referenced to, it is desired to name the element uniquely to make it easier for the reader to identify the dependencies that exist throughout the document.
Each clause contains a short entry description about the purpose of the clause.
Clause 2 lists normative references.
Clause 3 defines Terms and abbreviations for the context of this document.
Clause 4 presents a top-level overview of the basic concepts of IO-Link wireless
Clause 5 specifies the air interface (“PL”) of IO-Link wireless,
Clause 6 specifies Data Link Layer (DL-A) services and the DLA Message handler.
Clause 7 specifies Data Link Layer (DL-B) services, and the DL-B layer handlers.
Clause 8 specifies the services and the protocol of the Application Layer (AL).
Clause 9 describes the System Management responsibilities (SM).
Clause 10 specifies features and implementation details for W-Devices. These include Process Data Exchange (PDE), Parameter Management (PM), Data Storage (DS), and Event Dispatcher (ED).

Technology specific applications are not part of this standard. They may be specified in profiles for particular W-Device families.

Clause 11 specifies W-Master applications and features. These include Process Data Exchange (PDE), On-request Data Exchange (ODE), Configuration Management (CM), Data Storage (DS) and Diagnosis Unit (DU).

Several normative and informative annexes are included:

Annex A defines W-Message Codings.

Annex B defines W-Frame Codings, CRC calculation and Errors.

Annex C describes the W-Parameters and commands.

Annex D lists the system Event Codes (diagnosis information of W-Devices).

Annex E is linked to IO-Link Interface and System Specification REF 1 (description of the basic and composite data types).

Annex F contains design rules and constraints concerning low energy W-Devices.

Annex G describes the calculation of the frequency hopping tables.

Annex H informs about certification.

Annex I informs about regulatory compliance.

Annex J defines rules merging IODD and W-IODD file for W-Bridges.

2 Normative references

Identical to IO-Link Interface and System Specification clause 2 REF 1

3 Terms, definitions, symbols, abbreviated terms and conventions

For the purpose of this document, the terms and definitions given in IEC 61131-1 and IEC 61131-2, as well as the following ones apply.

3.1 Common terms and definitions

3.1.1 Acknowledge (ACK)

Response information indicating the acceptance of a message

3.1.2 Adaptive Hopping Table (AHT)

This mechanism enables a change of the hopping table of a track while the communication is already running (improvement of connection), see clause 18.4

3.1.3 Air interface

Radio-based communication links between the W-Master and the W-Devices

3.1.4 Application Layer (AL)

Part of the protocol responsible for the transmission of Process Data objects and On-request Data objects

3.1.5 Blacklist

List of frequency channels not to be used for IO-Link wireless communication within a W-Master

3.1.6 Broadcast (BC)

A mechanism to send a Master command to all W-Devices in a track. A Broadcast is restricted to specific Master commands according to Table 166.

3.1.7 Cell

Logical grouping of 1 or up to 3 W-Masters with a dedicated coverage area, often associated to a "machine".

3.1.8 Checksum

Data integrity measures for each pre-downlink, downlink or uplink in the physical layer

3.1.9 ConfigSyncword

Syncword which shall be used for the configuration channels

3.1.10 Configuration frequency channels

Two Frequency Channels are reserved for configuration purposes, see clause 5.4.4.

3.1.11 Configuration W-Frame

Downlink message to one particular W-Device with configuration data, followed by the corresponding uplink message of that W-Device

3.1.12 ConnectionParameter

A set of parameters containing the data which are necessary to establish wireless communication. The parameters are transmitted via the pairing mechanism (see Table 140). These parameters are only changeable via a new pairing or re-pairing.

3.1.13 Control interval

Time required to change the radio to receive mode, to transmit mode or to change frequencies

3.1.14 Control octet (CO)

Header, indicating the structure and purpose of a W-Message (2 octets in downlink 1 octet in uplink), see clause 12.3. and clause 12.4

3.1.15 Communication channel

Logical connection between W-Master and W-Device.

Note to entry: Four communication channels are defined: master command channel, process data channel, ISDU channel (for parameters), and diagnosis channel (for events).

3.1.16 Communication error

Unexpected disturbance of the transmission

3.1.17 Cyclic mode

Operation mode to achieve the needed short latency time.

Note to entry: Not occupied retries by process data are used for acyclic exchange of on-request Data. The configuration channels are not used.

3.1.18 DataSyncword

Syncword which shall be used for the data channels.

3.1.19 Disconnected

Disconnected describes the loss of communication between a W-Device and its W-Master.

3.1.20 Double Slot (DSlot)

Uplink type of a W-Device which combines two SSlots (15 octet payload) as uplink, see 4.5.3.

3.1.21 Downlink (DLink)

Multicast transmission from a W-Master to its associated W-Devices.
During configuration, a point to point transmission is used between W-Master and a particular W-Device

3.1.22 Event

Instance of a change of conditions in a W-Device
Uppercase "Event" is used for IOLW Event mechanism, while lowercase "event" is used in a generic manner.

3.1.23 Frequency channel

One out of 80 channels in the 2.4 GHz ISM band with frequency spaces of 1 MHz. For details see 5.4.1 and 5.4.4

3.1.24 Frequency division multiple access FDMA

Access method where users are allocated to individual frequency channels (frequency bands)

3.1.25 FullDownLink

Data structure of the regular downlink. Definitions see 5.2.8

3.1.26 Gaussian frequency shift keying (GFSK)

Binary frequency shift modulation with gaussian filter limiting its spectral width

3.1.27 Guard interval

Time interval between successive uplinks to avoid collisions on air

3.1.28 Header

Message part relevant for wireless communication only, consisting of Preamble, Syncword, MasterID, Track Number, and Acknowledge.

3.1.29 MasterID

Identification number of a particular W-Master

3.1.30 IMA message "I am alive"

Message from the W-Device, which informs the W-Master, that it is still functional.

3.1.31 IMA time

User configured watchdog time in which the W-Device shall send a IMA message, if no other messages had been sent.

3.1.32 InspectionLevel (IL)

Degree of verification for the W-Device identity during start-up

3.1.33 ISDU

Indexed service data unit used for acyclic transmission of parameters which may be segmented in multiple W-Frames

3.1.34 ISM band

Radio frequencies reserved for industrial, scientific, and medical use

3.1.35 Negative Acknowledge (NACK)

Indicating a missing response message, e.g. wrong CRC or no received DLink or ULink

3.1.36 Negotiation

Step within the pairing procedure for configuration of the ConnectionParameter

3.1.37 Packet Error Probability (PEP)

The Packet Error Probability is the mean error probability within the last 3000 transmissions. Errors are not acknowledged downlink- or uplink transmissions.

3.1.38 Pairing

Pairing is the equivalent procedure to plugging in a cable connection in a wired system between a master and a device.

3.1.39 Payload

Message parts carrying cyclic Process Data and acyclic On-request Data such as commands, Events and ISDUs (see W-Message Figure 133)

3.1.40 Port and Device Configuration Tool (PDCT)

Engineering support for a W-Master and W-Devices is usually provided by a “Port and Device Configuration Tool”.

3.1.41 PortCycle

The IO update is performed in a cyclic manner, which is determined by the W-Port related cycle time, within which the IO data of the W-Device are read or written

3.1.42 Preamble

Fixed bit pattern used for bit synchronization and calibration of automatic gain control of a radio receiver

3.1.43 PreDownLink

Part of a downlink including preamble up to CRC16. The 2 octet payload is used for low energy W-Devices only. Definitions see clause 5.2.8

3.1.44 Ressidual Failure Probability (RFP)

The Ressidual Failure Probability is the probability that the maximum latency is violated. By the calculation of the RFP the MaxRetry will be taken in account.

3.1.45 Roaming

Feature that allows mobility to a predefined W-Device between multiple predefined W-Masters by "Handover disconnect" and "Handover connect" procedures.

3.1.46 Security

All organizational measures and technical mechanisms to achieve authentication, confidentiality, integrity and availability.
Note: In the context of IOLW communication, encryption is not in the scope of the security goals.

3.1.47 SerialNumber

Unique vendor specific code for each individual W-Device.

3.1.48 Service Mode

Operational mode in which a W-Master track also utilizes the configuration channels. This mode is required for discovery, pairing and roaming procedures. A W-Master shall only operate with one of its tracks in Service Mode at the same time.

3.1.49 Single Slot (SSlot)

Uplink type of a W-Device which uses one SSlot (2 octet payload) as uplink, see 4.5.3.

3.1.50 Synchronization

Is a process of matching timing and the frequency tables between a W-Master and corresponding W-Devices.

3.1.51 Track and W-Device-Mapper (TDMapper)

The TDMapper is located in DL and assigns a W-Port to a specific Track number (Track_N) and Slot number (Slot_N).

3.1.52 UniqueID

Unique 9 octets identifier for each W-Device, consisting of the 16 bit manufacturer distinguishing VendorID, 24 bit DeviceID and a 32 bit W-Device distinguishing identifier, which is related to the SerialNumber. See clause 14.3.9

3.1.53 Unpairing

Unpairing is the user action via PDCT, to delete the permanently stored ConnectionParameter on the W-Device side and subsequently disconnects the W-Device. On the W-Master side the Unpairing command clears the current W-Port configuration which disables the communication with the unpaired W-Device

3.1.54 Uplink (ULink)

Single cast W-message from a W-Device to its W-Master consisting of Header, Payload, and Cyclic Redundancy Check (CRC).

3.1.55 W-Bridge

W-Device that connects a wired IO-Link device via IO-Link Wireless to a W-Master.

3.1.56 W-Coexistence

Wireless communication solutions within an industrial area can fulfill their communication requirements application, using the shared ISM frequency band.

3.1.57 W-Cycle

Combined utilization of TDMA and FDMA with several W-Sub-cycles to achieve a reliable wireless transmission.

3.1.58 W-Device

Single peer to a W-Master such as a IO-Link wireless sensor or actuator

3.1.59 W-Frame

Sequence of messages comprising a W-Master message (DLink) and all subsequent W-Device messages (ULinks). The sequence is transmitted in a W-Sub-cycle consisting of control intervals, downlink, and uplink slots (see Figure 21).

3.1.60 W-Master

Peer connected through W-Ports via radio to one up to n W-Devices and which provides an interface to the gateway to the upper level communication systems or PLCs

3.1.61 W-Message

Content of payload comprising control octet and (segmented) data exchanged between W-Master and W-Device (see Figure 21)

3.1.62 W-Parameter

This is the generic term that describes all the parameters located in the "wireless specific index" range, see clause 14.3.

3.1.63 W-Port

The logical wireless Port number to address a paired W-Device.

3.1.64 W-Sub-cycle

Time duration to transmit one W-Frame (see Figure 21).

3.1.65 Wired IO-Link Device

Device according to IO-Link Interface and System Specification

3.1.66 WLAN channels

Occupied frequency blocks used by WLAN

3.2 Abbreviated terms

IOL = wired IO-Link
IOLW = IO-Link wireless

3.3 Conventions

3.3.1 General

The service model, service primitives, and the diagrams shown in this standard are entirely abstract descriptions. The implementation of the services may reflect individual issues and can be different.

3.3.2 Service primitives

Service primitives are used to represent service provider/consumer interactions (ISO/IEC 10731). Each service consists of up to four service primitives:

- request primitive (.req),
- indication primitive (.ind),
- response primitive (.rsp), and
- confirmation primitive (.cnf).

An indication can occur with or without the corresponding response, a request can occur with or without the corresponding confirmation.

Figure 5 shows a generalized example of a confirmed service. In this case “Initiator” and “Responder” correspond to either Master or Device and not to the layer within one of each.

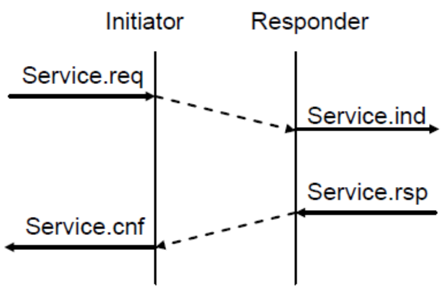


Figure 5 Generalized example of a confirmed service

3.3.3 Service parameters

Service primitives convey parameters which indicate the information available in the provider/ consumer interaction. In any particular interface, not each and every parameter needs to be explicitly stated.
The service specification in this standard uses a tabular format to describe the component parameters of the service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns:

1) parameter name;
2) request primitive (.req);
3) indication primitive (.ind);
4) response primitive (.rsp); and
5) confirmation primitive (.cnf).

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column.

- M Parameter is mandatory for the primitive.
- U Parameter is a user option and can or cannot be provided depending on dynamic usage of the service user. When not provided a default value for the parameter is assumed.
- C Parameter is conditional upon other parameters or upon the environment of the service user.
- – Parameter is never present.
- S Parameter is a selected item.

Some entries are further qualified by items in brackets. These may be:

a) a parameter-specific constraint "(=)" indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table;
b) an indication that some note applies to the entry "(n)" indicates that the following note "n" contains additional information related to the parameter and its use.

3.3.4 Service procedures

The procedures are defined in terms of:

- the interactions between application entities through the exchange of protocol data units; and
- the interactions between a communication layer service provider and a communication layer service consumer in the same system through the invocation of service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the communication layers.

3.3.5 Service attributes

The nature of the different (W-Master and W-Device) services is characterized by attributes. All services are defined from the view of the affected layer towards the layer above.

- I Initiator of a service (.ind) (towards the layer above)
- R Receiver (responder) of a service (.req) (from the layer above)

Figure 6 Example of initiator / receiver of services (W-Device)

Figure 6 shows the example of initiator / receiver of services (W-Device)

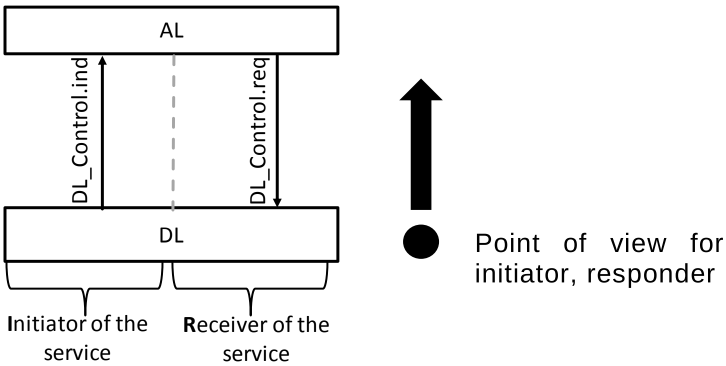


Figure 6 Example of initiator / receiver of services (W-Device)

3.3.6 Figures

For figures that show the structure and services of protocol layers, the following conventions are used:

- an arrow with just a service name represents
  - a request and the corresponding confirmation (the request being in the direction of the arrow);
  - an indication and the corresponding response (the indication being in the direction of the arrow)

- a request without confirmation as well as an indication without response are labelled as such (i.e. service.req, service.ind).

Figure 7 shows the example of service between W-Master and W-Device in generalized and detailed view

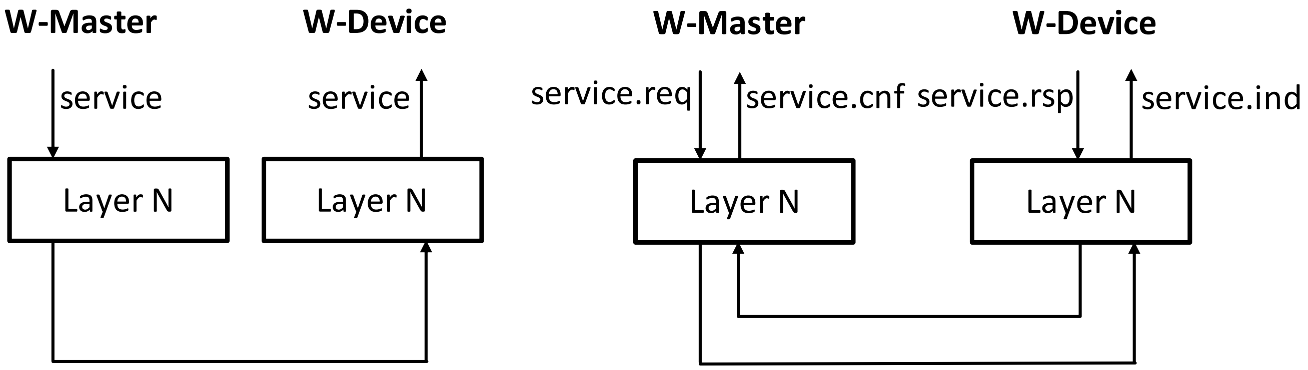


Figure 7 Example of service between W-Master and W-Device in generalized and detailed view

Figure 8 shows the example of confirmed / non-confirmed services (W-Device)

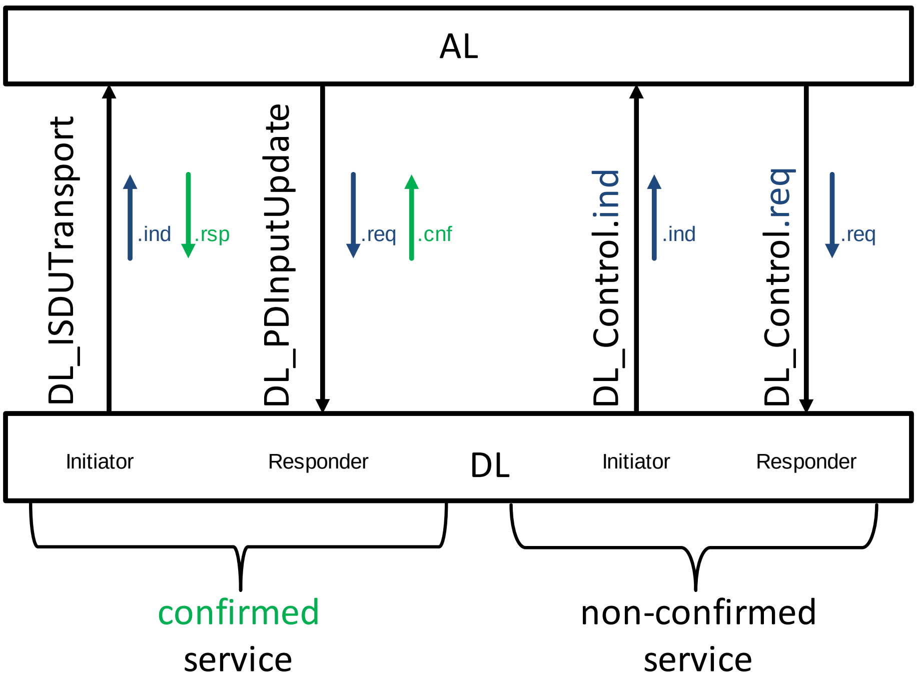


Figure 8 Example of confirmed / non-confirmed services (W-Device)

3.3.7 Transmission octet order

See clause 5.2.1

3.3.8 Behavioral descriptions

For the behavioral descriptions, the notations of UML 2 (ISO/IEC 19505) are used (e.g. state, sequence, activity, timing diagrams, guard conditions). The layout of the associated state-transition tables is following IEC/TR 62390.

Due to design tool restrictions the following exceptions apply. For state diagrams, a service parameter (in capital letters) is attached to the service name via an underscore character, such as for example in DL_SetMode_INACTIVE. For sequence diagrams, the service primitive is attached via an underscore character instead of a dot, and the service parameter is added in parenthesis, such as for example in DL_Event_ind (OPERATE). Timing constraints are labelled "tm (time in ms)".

Asynchronously received service calls are not modelled in detail within state diagrams.

To find the balance between clearness and degree of detailing not all negative responses or confirmations are modelled explicitly in the state machines.

4 Overview of IO-Link wireless

4.1 Purpose and topology

IO-Link wireless is a communication technology intended to replace the cable(s) for remote sensor/actuator control in production automation. The key features of IO-Link wireless technology are real-time capabilities, very low latency and robustness. Applications within factory automation comprise moving parts such as rotating bottle filling, robot arms and linear moving machinery. These applications are difficult to realize with wired sensor/actuator equipment or suffer from frequently broken wires. These kinds of applications are targets of IO-Link wireless.

IO-Link wireless equipment operates in the unlicensed 2.4 GHz ISM band and using frequency hopping to reduce the impact of interference. IO-Link wireless realizes a communication between the air interface of a wireless Master (W-Master) and the air interface of one or more wireless Devices (W-Devices).

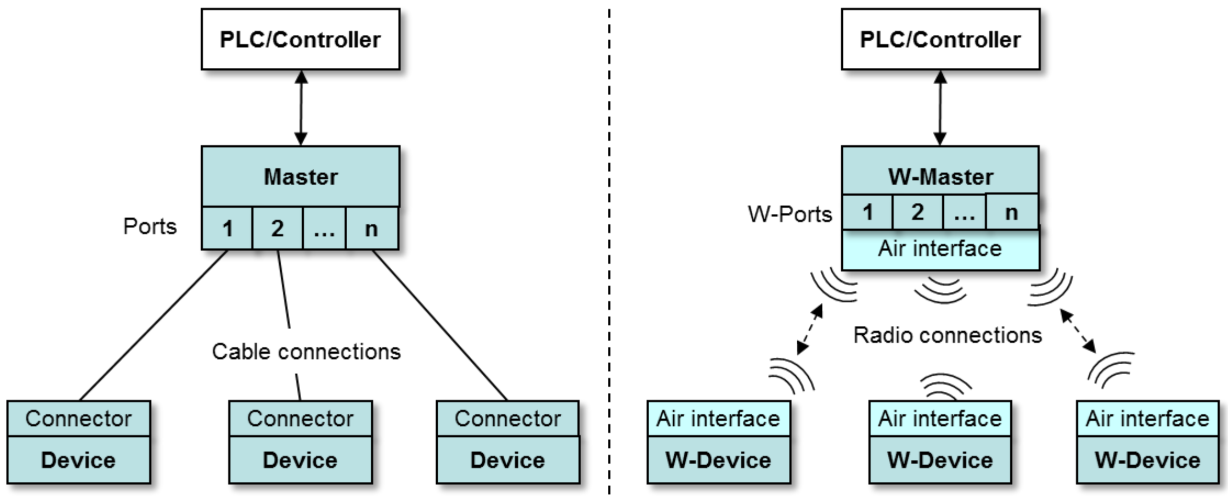


Figure 9 IO-Link and IO-Link wireless topology

From a PLC or Controller users point of view, Master and W-Master provide the same functionality in respect to Process Data (PD) and On-request Data (OD). The main differences between the two topologies is during commissioning, the discovery of available W-Devices ("scan"), the connection configuration ("pairing") and the parameterization of the air interface ("W-Parameter").

4.2 Positioning in the automation hierarchy

Figure 10 shows the architecture of an automation topology with an IO-Link wireless system comparable to the wired version.

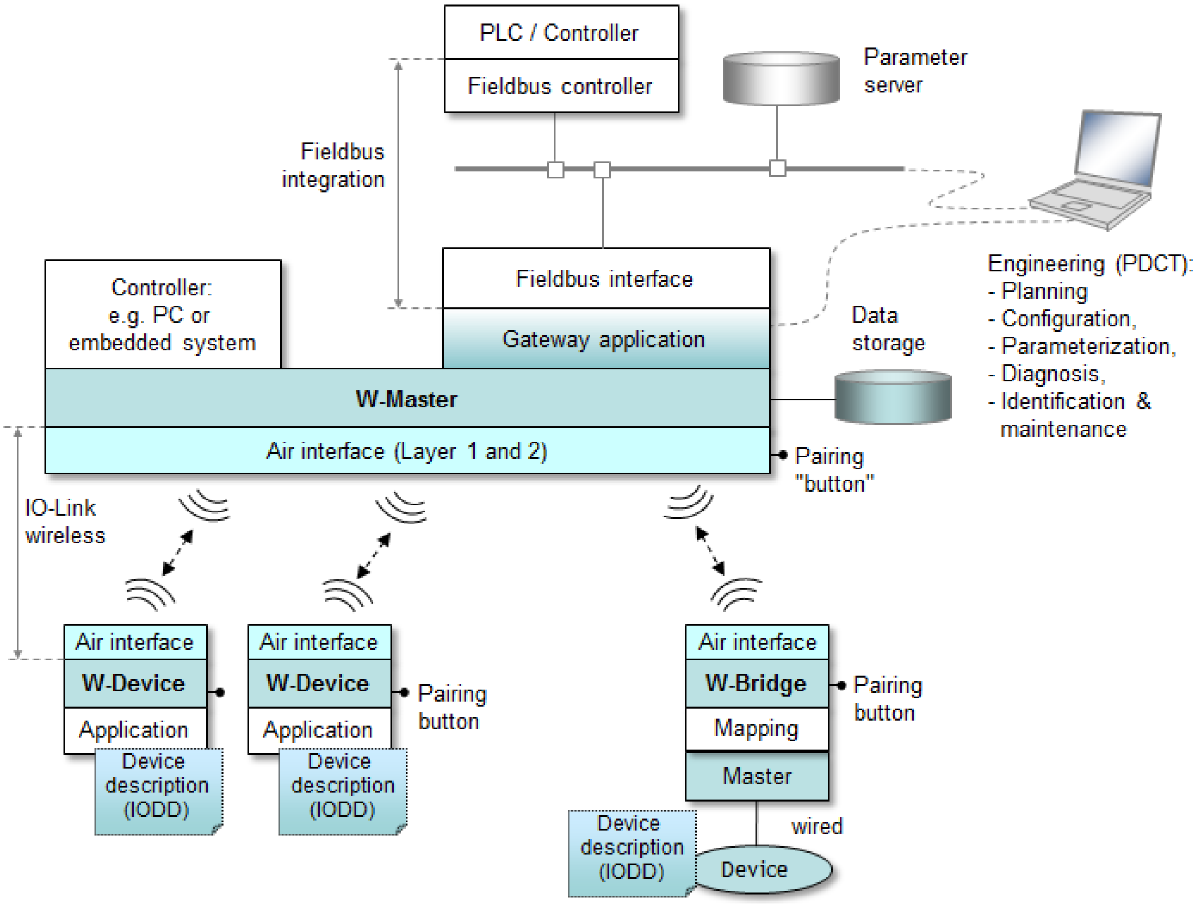


Figure 10 IO-Link wireless system

In all cases a PLC, a PC-based controller, or an embedded system can exchange Process Data (PD) and/or On-request Data (OD) with wire or radio connected devices via Master or W-Master, respectively. Additional to the IO-Link Interface and System Specification (REF 1) clause 11.7.7 the "Port and Device Configuration Tool" (PDCT) for IO-Link can be extended by features like:

- Device discovery and pairing support for unpaired devices.
- Optimizing connection quality of W-Masters and W-Devices.
- W-Coexistence management for a conflict-free layout of the radio transmissions such as overlapping frequencies of non-IO-Link wireless systems.
- Configuration of the W-Parameters as described in the provided IODDs.

IO-Link wireless uses the Data Storage mechanism of IO-Link Interface and System Specification REF 1 to support faulty device replacement. To re-establish a wireless connection after a W-Device replacement, pairing buttons or software tools can be used. With pairing buttons activated on both W-Device and W-Master a W-Device can be exchanged without the need of any software tools.
After replacement, the parameters are downloaded automatically from the Data Storage, if enabled.

4.2.1 Relationship to IO-Link

In relationship to IO-Link, the transfer of the IO-Link objects via the Downlink and Uplink mechanism is outlined in Figure 11.

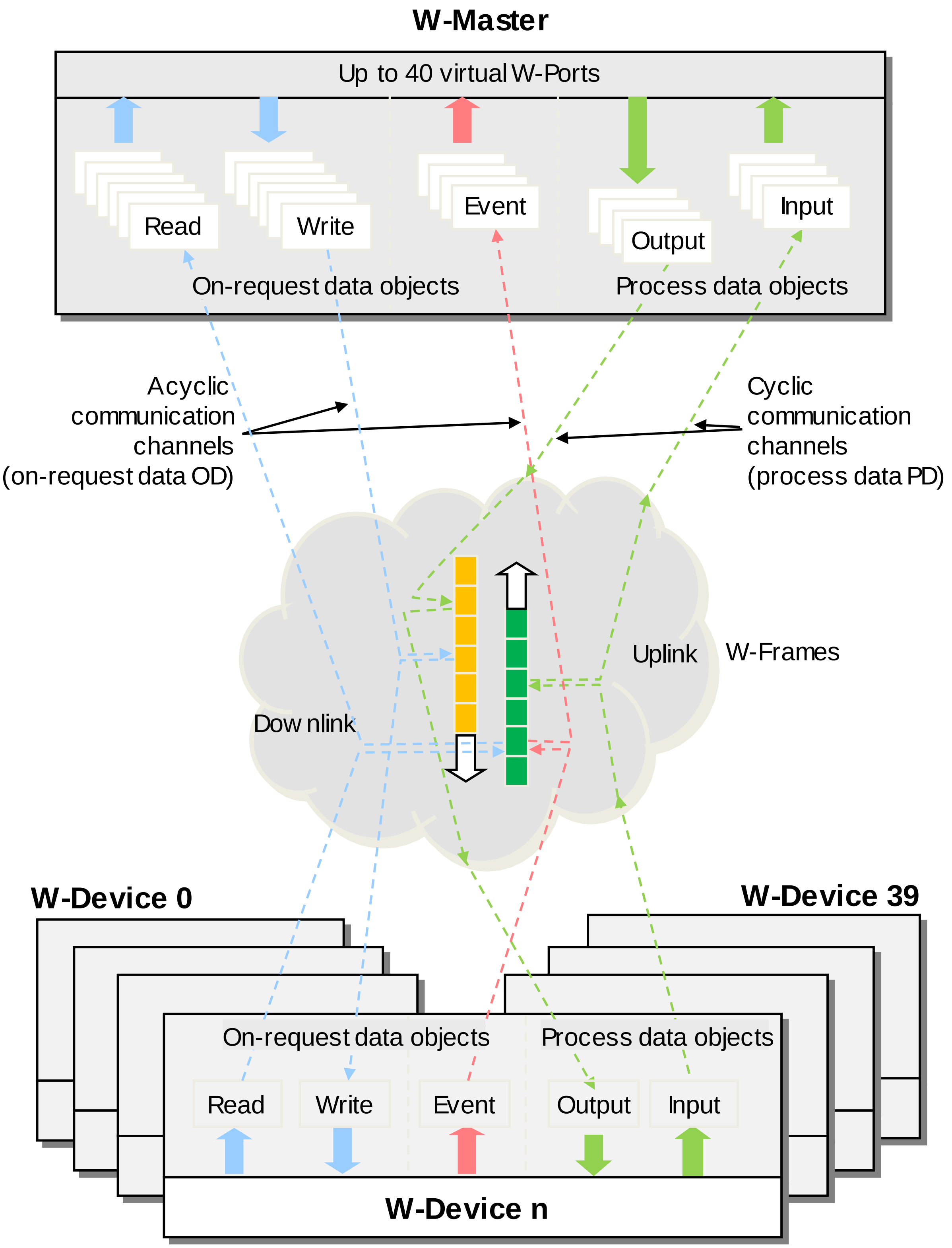


Figure 11 Object transfer at the application layer level (AL)

4.2.2 Role of a W-Master

A W-Master manages up to 40 W-Port instances. The possible max. number of W-Ports depends on the available tracks and slots and how they are utilized.

A W-Master can comprise up to five (small band) transceivers with their own antenna and dedicated frequency channels, called tracks. Each track can serve up to 8 W-Devices and send and receive alternately. All tracks of a W-Master send at the same time on different frequencies according to the computed frequency hopping tables, providing an optimal medium utilization.

The user may manually operate the W-Master for discovery and pairing of devices.

During commissioning or roaming Service Mode is used by the W-Master to establish communication with W-Devices (pairing), includes checking of the "identity" of the W-Device, i.e. its VendorID, DeviceID, and communication properties. If there is a mismatch between W-Device parameters and the stored parameter set within the W-Master, the parameters in the W-Device are overwritten (see 11.3) or the stored parameters within the W-Master are updated depending on the configuration.

After power on with paired W-Devices, the W-Master establishes communication, including all checks described above.

The W-Master is responsible for the assembly and disassembly of all data from or to the W-Devices (see Clause 11).

4.2.3 Role of a W-Device

A W-Device consists of a single transceiver, the IOLW device stack and the technology specific application, i.e. the transducer with its technology parameters. The common W-Device applications are the same as in IO-Link and comprise of configuration parameters, diagnosis information and process data.

4.2.4 Role of a W-Bridge

A W-Bridge is a W-Device to connect a single wired IO-Link Device. The application part of the W-Bridge basically contains a wired IO-Link Master.

For compatibility reasons towards the wired IO-Link Device in the System Configuration Tooling, a straight forward parameter mapping of the wired IO-Link device via the W-Bridge is desired. To achieve this, the IODD of the wired IO-Link Device and the required extension for the W-Bridge are merged together to constitute the W-IODD of the novel entity formed by the W-Bridge and the wired IO-Link Device.

4.2.5 System Configuration Tool

Engineering support for a W-Master is usually provided by a Port and Device Configuration Tool (PDCT). The PDCT configures both W-Port and W-Device properties. It combines both an interpreter of the IO Device Description (IODD) and a configurator (see 11.7.2). The parameters provide all the necessary properties to establish communication and the desired function of a sensor or actuator.

4.2.6 Mapping to fieldbuses

See clause 4.7 in REF 1 and clause 11.8.1.

4.3 Cell concept

Due to the limited transmission power (see Air Interface 4.5), the possible range of a W-Master is limited to max. 20 m in case of only one track as shown in Figure 12. This value is derating to ≤ 10 m if more than one track is active.

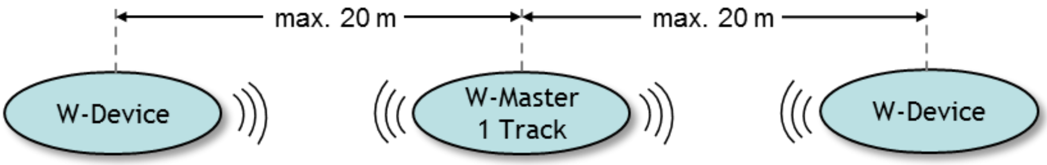


Figure 12 Radius of a cell with a 1 track W-Master

A single W-Master can consist of one up to five tracks. Up to 3 W-Masters are allowed within one cell to a certain extent. If there are more than one W-Master installed in a cell, the MasterID's shall be subsequently. To prevent frequency access conflicts between the tracks, IO-Link wireless provides mechanisms to create

disjoint frequency tables by W-Masters. Every W-Master has its MasterID, a frequency hopping table and a blacklist.
One W-Master and a group of associated W-Devices form a W-Master cell is shown in Figure 13. The W-Master A is connected to W-Devices A1 to Ai. The W-Master B is connected to W-Devices B1 to Bj, whereas both systems are in an overlapping RF coverage area.

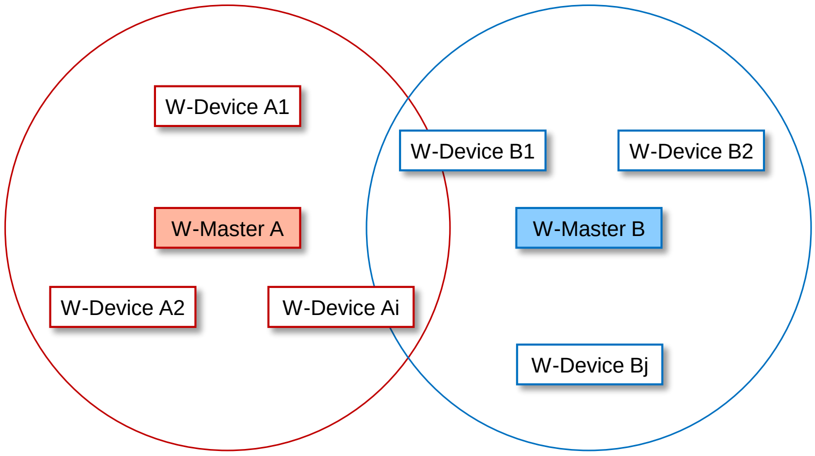


Figure 13 W-Master cell consisting of 2 W-Master

Figure 14 shows the IO-Link wireless concept with partly overlapping W-Master cells. In one area, there should not coexist more than three W-Master in order to avoid interference. W-Master cells with a distance of more than 40 m can use the same MasterID again. W-Master cells with a distance less than 40 m require distinct MasterIDs. The MasterID is used to calculate individual frequency hopping tables.

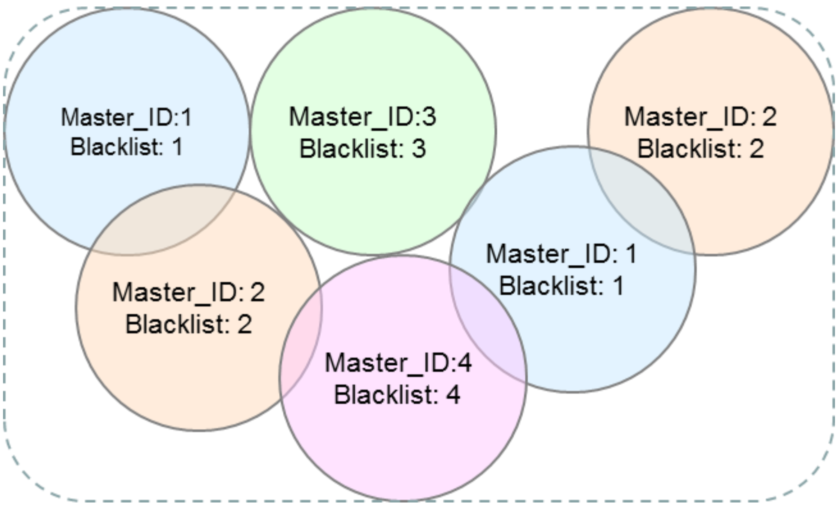


Figure 14 Cell concept

4.4 Wireless Mechanisms

The following mechanisms are used to setup and operate the wireless connections.

4.4.1 Scan (Device Discovery)

After power-on, every unpaired W-Device is waiting for connection establishment from a W-Master on the configuration frequency channels. Upon user request for W-Device discovery, the W-Master sends scan request messages on the configuration frequency channels. Any unpaired W-Device receiving such a scan request message is responding with a scan response message, where the W-Device returns its unique identifier (UniqueID) for authentication purposes before pairing. With the help of this mechanism all unpaired W-Devices in the proximity of the W-Master can be discovered. Subsequently, the application can decide to pair the W-Devices.

Several W-Devices may simultaneously respond within a single uplink. In order to minimize collisions, they are using randomly determined time slot positions within that uplink frequency. In this manner, the W-Master collects all non-paired W-Devices over time within several W-Sub-cycles.

4.4.2 Pairing

Pairing is the equivalent procedure to plug in the cable connection in a wired system.

4.4.2.1 Pairing by UniqueID

This mechanism is provided for pairing of a W-Device with a pre-configured W-Master and reflects the normal commissioning mechanism. The UniqueID of the W-Device is used for automatic identification of the W-Device within the pairing process. An Engineering Tool or HMI such as an PDCT is required for the pre-configuration of the W-Master. See clause 10.7.3.

4.4.2.2 Pairing by Button

This mechanism is for manual pairing without detailed knowledge about the W-Device. No Engineering Tool is required for this kind of pairing. The pairing must be acknowledged on both entities by manual intervention (i.e. pressing a button or equivalent mechanism). In case of a faulty W-Device, which must be replaced with a new, but identical W-Device, IO-Link wireless provides this simplified procedure for the paring of both partners without using an Engineering Tool or PDCT.

4.4.2.3 Re-Pairing by Button

A W-Device previously paired to former W-Master can be re-paired to a new W-Master. If such a W-Device still has the ConnectionParameters of its “old” W-Master, it stays in state Configured. With a button press, the W-Device can be switched by the operator temporarily to state Re_Pairing and listens on the configuration channels for a pairing request message from the new W-Master. The pairing must be started previousely on the W-Master by manual intervention (i.e. by UniqueID or pressing a button).

4.4.3 Unpairing

A W-Device can be removed from a communication relationship with a W-Master. When the operator wants to unpair one of the W-Devices, the W-Master is triggered by the operator (i.e. via HMI) and starts an unpairing procedure on the dedicated W-Device. This sends an unpairing request to the W-Device, which sends an acknowledgement back to the W-Master. The ConnectionParameters on the W-Device and the related W-Port configuration within the W-Master are deleted.

4.4.4 Roaming

Roaming is a feature that allows mobility to a predefined W-Device between multiple predefined W-Master cells. A W-Master track configured to Roaming Mode is sending scan request messages on the configuration channels to detect roaming W-Devices in their range. Disconnected roaming W-Devices listening for a W-Master shall respond with a scan response message to indicate their presence to this W-Master. The application on the W-Masters may then decide to connect the roaming W-Device by initiating a pairing and configuration sequence.

Handover disconnect is initiated by the W-Master when the application (e.g. the PLC) wants to release the W-Device, for example when the application has finished processing with the roaming W-Device in its current state (e.g. in a tool changer or conveyor belt application). Another reason for a disconnect procedure could be that the parameter LinkQuality at the W-Port has degraded to an absolute minimum, indicating that the W-Device leaves the range of the W-Master. Reconnection in the case of link quality degradation to the same W-Master should only be done if the link quality has improved substantially.

For any handover procedure with another W-Master, the scan message followed by a pairing and a W-Device startup sequence is utilized. It must be noted that the handover procedure requests a certain amount of time where no process data can be exchanged.

An unexpected IMA-Failure detected by a W-Master from a roaming W-Device must lead to an autonomous handover disconnect of this W-Device for the associated W-Port within the W-Master.

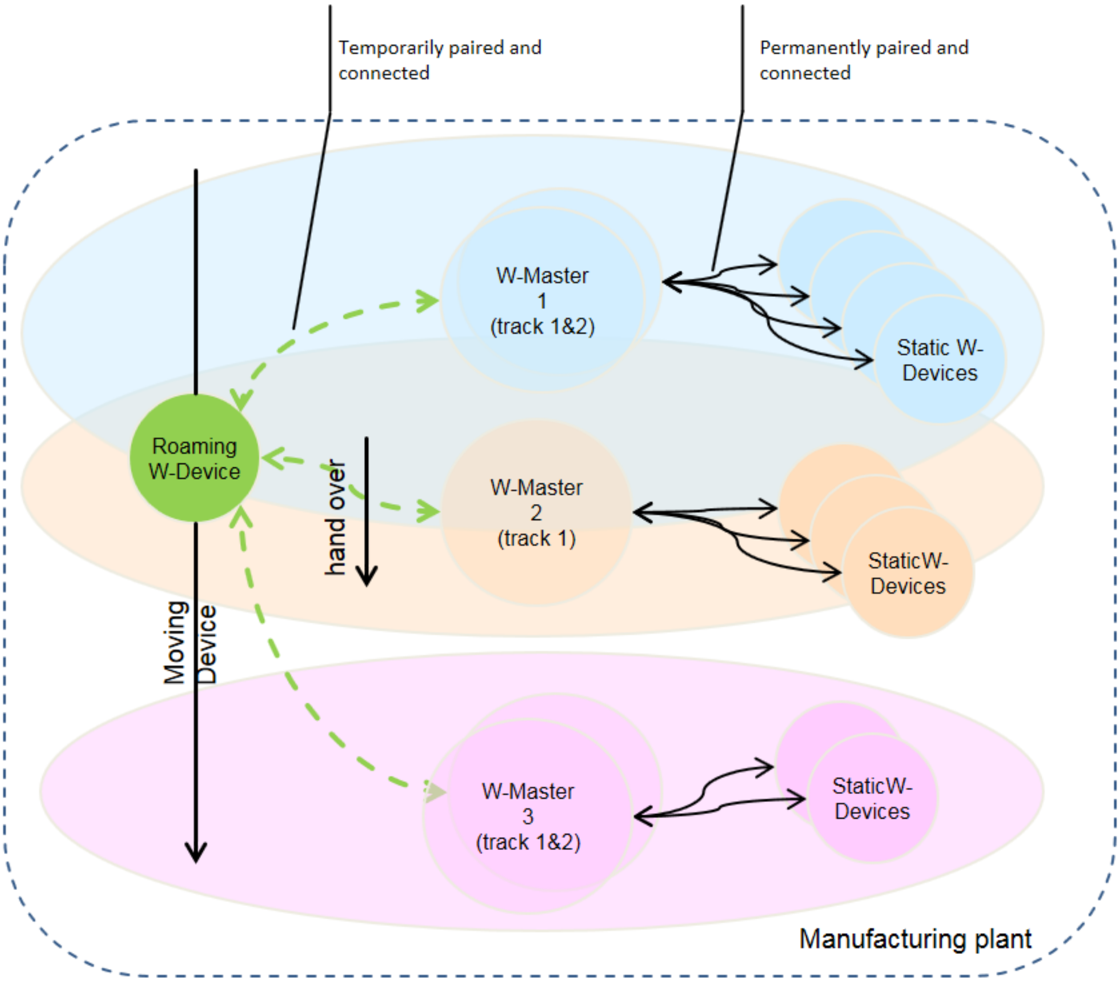


Figure 15 Roaming between W-Master cells

“Roaming” is configurable on the W-Master. On each W-Master, not more than one track shall be configured for Roaming Mode, as indicated in Figure 15 for W-Master 1 and W-Master 3. The tracks in Roaming Mode utilize a dedicated frequency hopping table which includes the configuration channels. For the “handover disconnect” procedure, the entire fault indications (e.g. IMA timeout) to the system/user are suppressed, since it is related to an intended action. Accordingly, all pending diagnosis messages of the related W-Port and W-Device are deleted once the "handover disconnect" procedure is completed. A roaming W-Device does not permanently store its pairing information and discards it when disconnected.
The computation of the frequency hopping tables for roaming is described in clause 18.2.

4.4.5 Transmission Error Handling

Transmitted packets in both uplink and downlink direction are subject to error detection via CRC and must be acknowledged by the receiving side. W-Devices acknowledge correct reception of their Downlink packets within the respective subsequent Uplink packets. Within the next downlink, the W-Master acknowledges correct reception of the Uplink packet to each W-Device. In case of missing acknowledgments, the W-Master uses this information to initiate a retransmission within the same W-Cycle. When all retransmissions fail within a W-Cycle, a communication error is indicated towards system management.

4.4.6 “I am alive” supervision (IMA)

The activity of a W-Device is monitored via an “I am alive” (IMA) timer in the W-Master. When a W-Device has no data to transmit for a time period longer than its configured IMATime, an IMA message must be sent by the W-Device before the IMA-timer expires. When the supervision IMA-timer of the W-Master expires, i.e. because the W-Device is down, a communication error must be indicated via system management.

4.4.7 Link Quality supervision

Link Quality Indication is a service for evaluation of the functionality and reliability of the IO-Link Wireless system in its application environment of the wireless connection. The parameter LinkQuality of a communication link between W-Master and a W-Device is continuously monitored and can be accessed on W-Master by the gateway application via service GetPortQuality. The computation of the LinkQuality Indicator is described in clause 5.4.6.

4.5 Concept of Air Interface

IO-Link wireless uses the license-free 2.4 GHz ISM band (industrial, scientific, and medical) from 2.4 to 2.4835 GHz compliant to REF 5 Bluetooth SIG - Regulatory Committee, "Bluetooth low energy Regulatory Aspects", V10r00, 26 April 2011, which also forms the basis for the well-known Bluetooth®-technology. It is therefore possible to cost-effectively use existing integrated radio circuits available on the market to build IO-Link wireless systems. But it must be noted that there is no system compatibility between Bluetooth© and IO-Link wireless. For more details see clause 5.4

4.5.1 Frequency Division Multiple Access (FDMA)

Using different carrier frequencies in IO-Link wireless follows in principle the Frequency Division Multiple Access (FDMA) technology. IO-Link wireless periodically changes the transmission frequencies ("frequency hopping") to improve robustness against burst interferences. W-Coexistence with other wireless systems and other IO-Link wireless cells is possible through omitting those frequencies within the table of unusable transmission frequencies ("Blacklisting").

4.5.1.1 Frequency Hopping Tables

To compute the frequency hopping tables for a W-Master and its W-Devices, IO-Link wireless defines dedicated channel hopping sequence algorithms that depend on the individual MasterID to achieve W-Coexistence within neighboring IO-Link wireless systems. For W-Coexistence with other wireless systems, a blacklist can be utilized to avoid certain frequency channels in the computed hopping table. The hopping sequence is transferred to the W-Device during pairing. For more details, see clause 18.2.

4.5.1.2 Blacklist

To suspend frequency ranges, they can be defined in the blacklist. The blacklist is configurable via SetMasterConfig. For more details, see clause 18.1.

4.5.1.3 Configuration channels

Configuration of W-Master and W-Devices is required prior to cyclic data exchange. For this purpose, the frequency channels "1" (2 401 MHz) and "80" (2 480 MHz) are exclusively used in an alternating manner for ConnectionParameter exchange and initial scan and pairing of W-Master with its W-Devices. For a detailed description of usage of the configuration channels see 18.3

4.5.1.4 Data channels

The frequency channels 3 (2403 MHz) to 78 (2478 MHz) can be used for cyclic data exchange. This number of frequency channels allows the configuration of W-Master sets disjoint from their cell neighbors for W-Coexistence. The set of frequency channels a W-Master uses is configured in the frequency hopping table (see 18.2).

4.5.2 Time Division Multiple Access (TDMA)

IO-Link wireless uses Time Division Multiple Access (TDMA) principles. A communication exchange between a W-Master and its W-Devices is splitted into a "downlink" phase that is immediately followed by an "uplink" phase for a dedicated track and frequency channel. The transmitters on the W-Master and W-Devices are operating in half-duplex mode, switching between TX and RX mode according to their time slots.

TDMA requires precise timings on both, sender and receiver. At the beginning of a TDMA cycle, the frequency channel is selected from the hopping table. The W-Devices respond at their subsequent respective time slots using the same frequency channel.

4.5.2.1 Downlink

The downlink communication from W-Master to its W-Devices can contain W-Messages for several W-Devices as shown in Figure 16. It is therefore a multicast communication. Immediately after sending the Downlink, the W-Master switches its radios from TX to RX mode, awaiting the subsequent uplink transmissions from the W-Devices of that track.

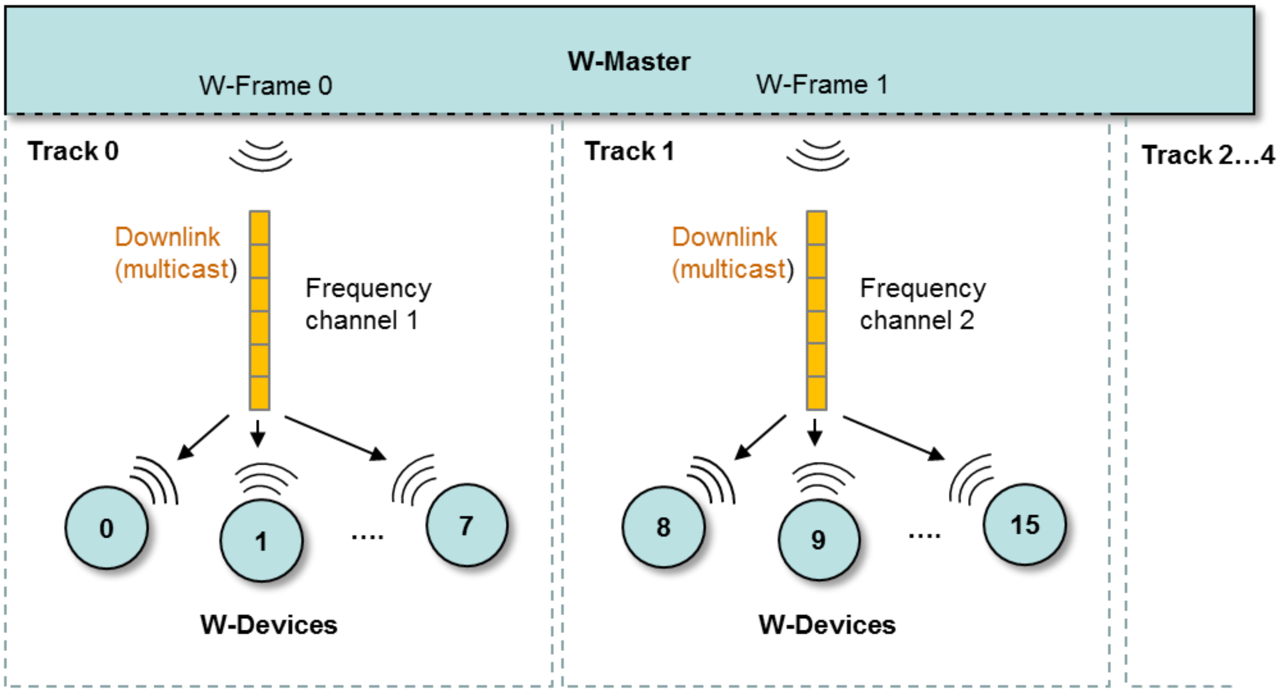


Figure 16 Downlink

4.5.2.2 Uplink

Figure 17 demonstrates the timely staggered delivery of single cast W-Messages of W-Devices to their W-Master.

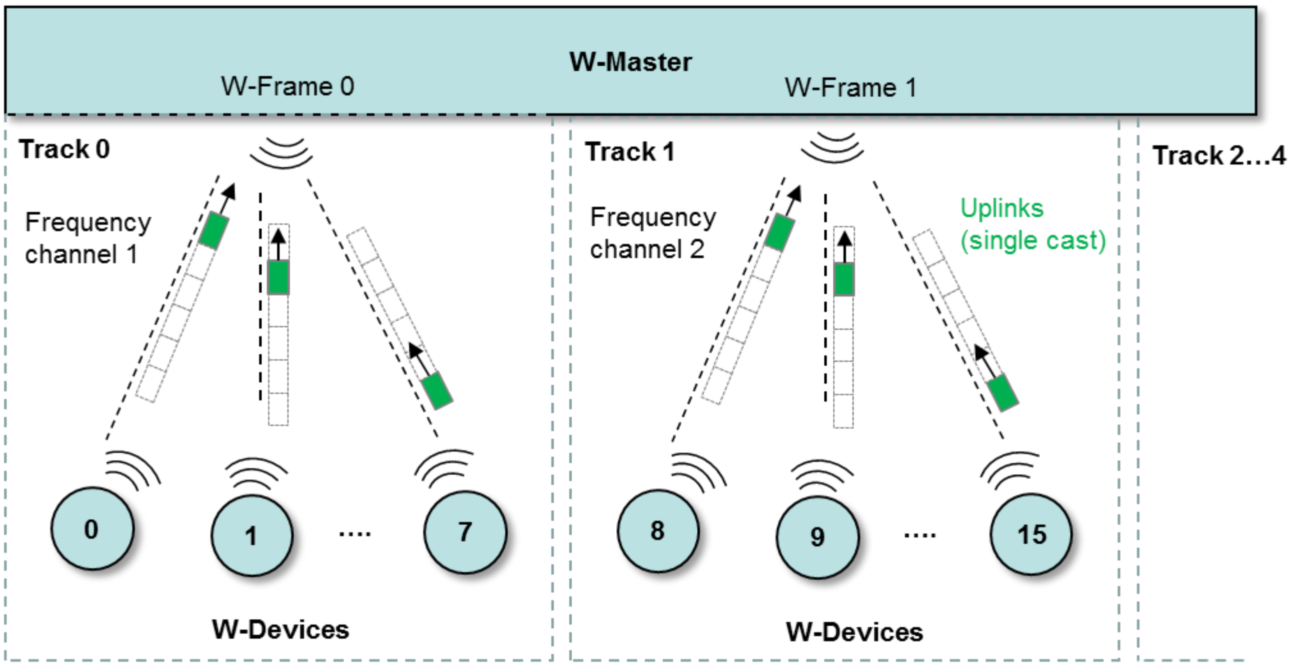


Figure 17 Uplink

4.5.2.3 Synchronization

The W-Master provides the system's master clock which is a downlink sent each W-Sub-cycle. To precisely switch the radio mode and send the Uplinks in the respective timeslots, the clocks of the W-Devices need to be continuously synchronized with the master clock. Synchronization of a W-Device clock takes always place when the W-Device receives a downlink from the W-Master. When the clocks after a longer communication pause between W-Master and W-Device have deviated (this particularly happens using low energy W-Devices), the W-Device may listen for a longer period of time until it detects its W-Masters downlink again.

A paired W-Device that has lost clock synchronization still knows the frequency channels of its W-Master via the frequency table. It just listens at one particular frequency channel until it receives an appropriate downlink of its W-Master and is then able to synchronize to the hopping sequence and uplink time slots.

4.5.3 SSlots, DSlots, Transmission capacity

The transmission capacity of downlink and uplink is shown in Figure 18. The Downlink can carry 52 octets. An uplink message can carry 12 octets or 25 octets, depending on the slot type "SSlot" or "DSlot". DSlots combine the payload of two SSlots to operate sensors or actuators with larger process data, but this reduces the number of possible W-Devices per track.

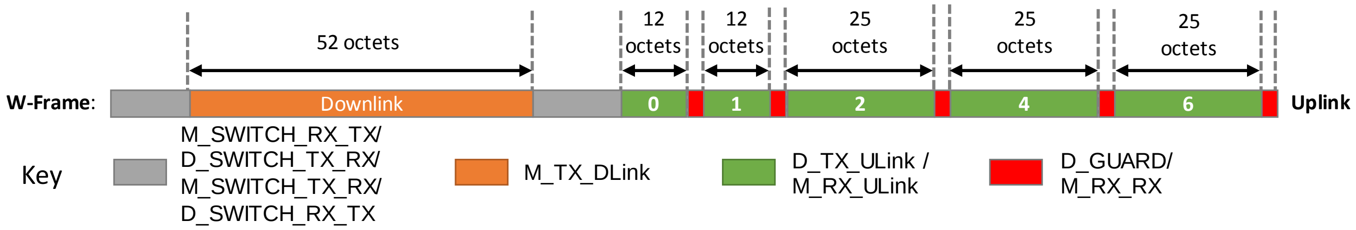


Figure 18 Transmission capacity with SSlots and DSlots

Some octets are required for protocol control data and integrity checksums, finally reducing the usable message payload. The message payload encodings are listed in Annex A clause 12.

4.5.4 Assignment of W-Devices to tracks and slots

A W-Master contains up to 5 tracks, which are numbered from 0 to 4. Each track has up to 8 slots, numbered from 0 to 7. This allows a maximum number of 40 W-Devices per W-Master.

Figure 19 shows the assignment of W-Device numbers to slots and tracks.

The allocation of W-Devices to track and Slot number is performed during commissioning and pairing.

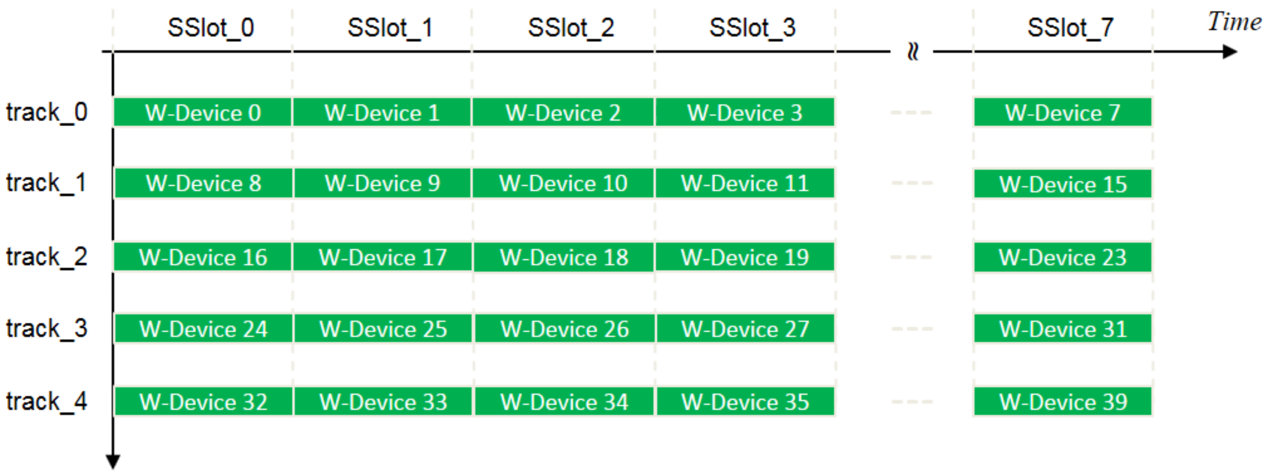


Figure 19 Uplink assignments

Numbering gaps in the W-Device count can occur because of DSlot usage (DSlots shall always be placed on even slots, see Figure 40) or non-used slots in a track.

4.5.5 Assignment of W-Ports to W-Devices

A W-Master provides a limited number of virtual W-Ports, depending on the number of available tracks and the slot configuration, since W-Devices with occupation of a DSlot reduce the number of available W-Ports. The W-Master must therefore administratively map its W-Device slots to these virtual W-Ports, which is performed on application level during commissioning. The Application shall maintain a monotonically increasing numbered list of W-Ports counting from 0 in the sequence of the commissioning operation and assign the W-Device slots autonomously. The mapping between W-Port and W-Device slot numbering is given via the W-Port handler of the System Management SM. The Track and W-Device Mapper (TD-Mapper) uses this information to map a W-Port to the corresponding track and slot (see 6.1.1).

4.5.6 W-Cycle

A W-Cycle utilizes TDMA and FDMA in combination with a retransmission mechanism to achieve a very dependable wireless transmission. The standard duration of a complete W-Cycle is almost 5 ms as shown in Figure 20, consisting of three W-Sub-cycles. The W-Cycle is configurable via SM_SetPortConfig with a granularity of the time duration of a W-Sub-cycle.
The W-Master uses the remaining W-Sub-cycles for retries in case of transmission errors caused by channel interferences. The different frequency channels for these sub-cycles and for each track are the countermeasure against these channel interferences. When no retransmissions are required within a W-Cycle, the otherwise unused bandwidth can be used to transfer acyclic data, such as On-request data (OD) or Events.
A W-Master can comprise up to five (small band) transceivers with their own antenna and dedicated frequency channels, called a track. Each track can serve up to 8 W-Devices and send and receive alternately. All tracks of a W-Master send at the same time on different frequencies according to the computed frequency hopping tables, providing an optimal medium utilization

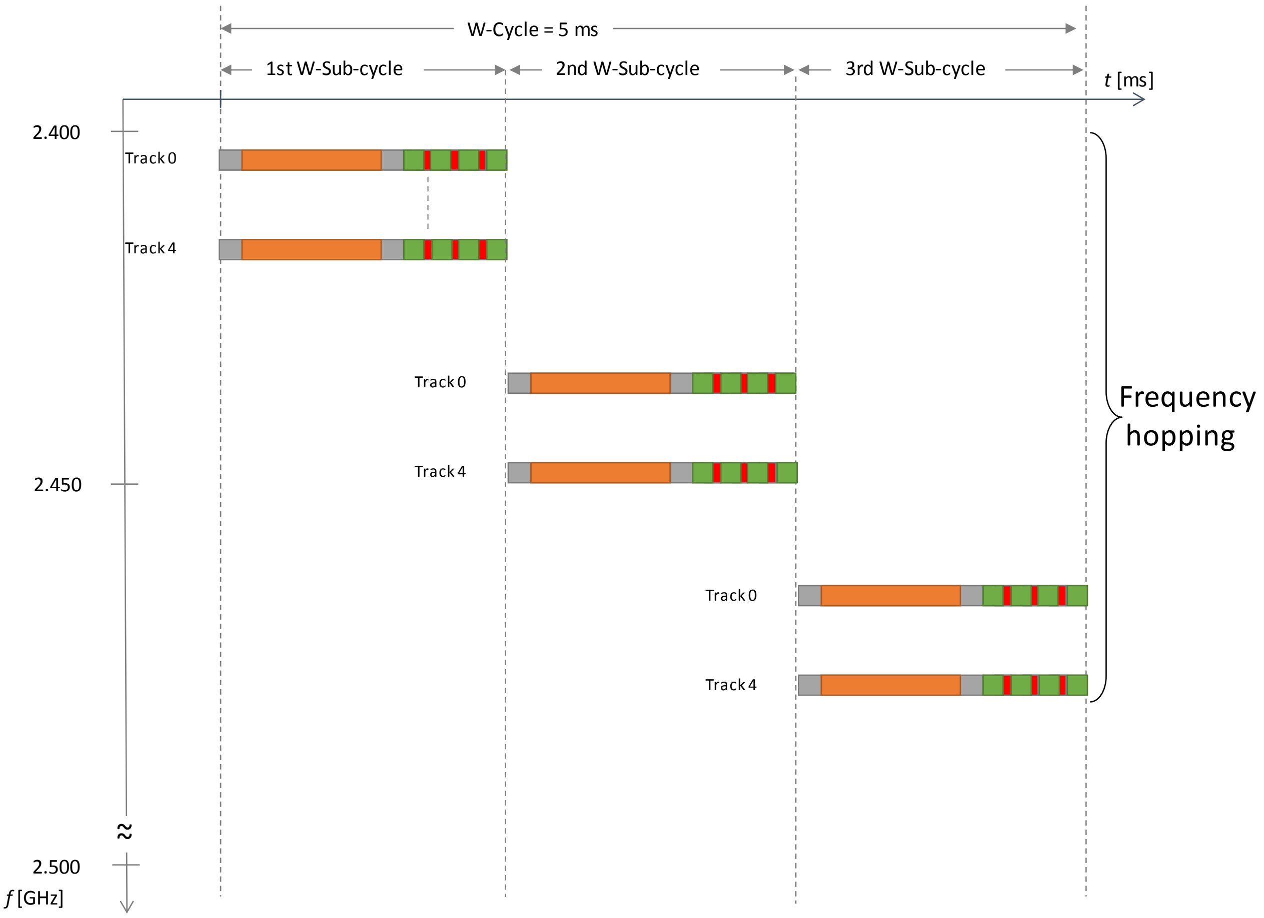


Figure 20 TDMA and FDMA in the W-Cycle

4.5.7 W-Frame

A W-frame is the data structure in which a communication exchange between a W-Master and its W-Devices is organized (see Figure 21). It is structured in Control intervals, Downlink and Uplinks. In Control interval, the radio switches between transmission and reception and in the first Control interval also frequency hopping takes place.
The Downlink addresses all devices via broad cast. The Uplinks is transmitted subsequently W-Device by W-Device in the respective timeslot. The W-Frame is transmitted in a W-Sub-cycle of 1.664ms.

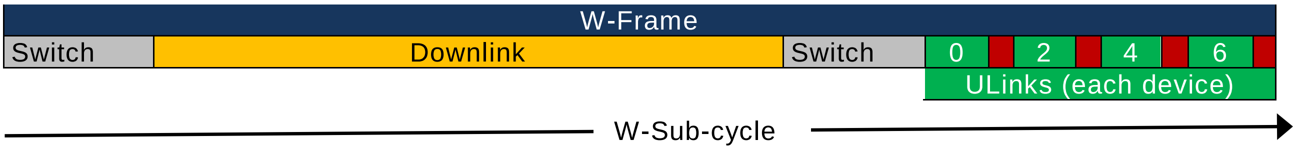


Figure 21 W-Frame and W-Sub-cycle

5 Air interface / Physical Layer (PL)

This clause describes the relevant definitions for transceivers and media access on both, W-Master and W-Devices, which must comply to the requirements described below.
For an easier relationship to the IO-Link Interface and System specification REF **1** the air interface, which contains the wireless mechanisms and the physical layer, is called physical layer (PL) furthermore.

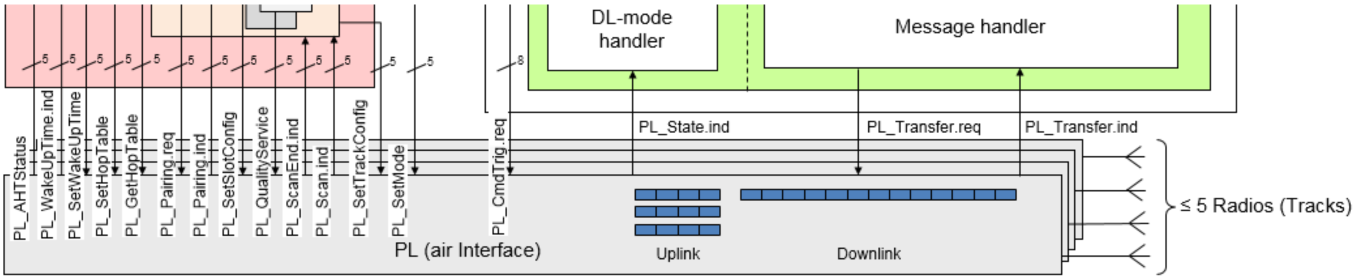


Figure 22 Physical layer (W-Master)

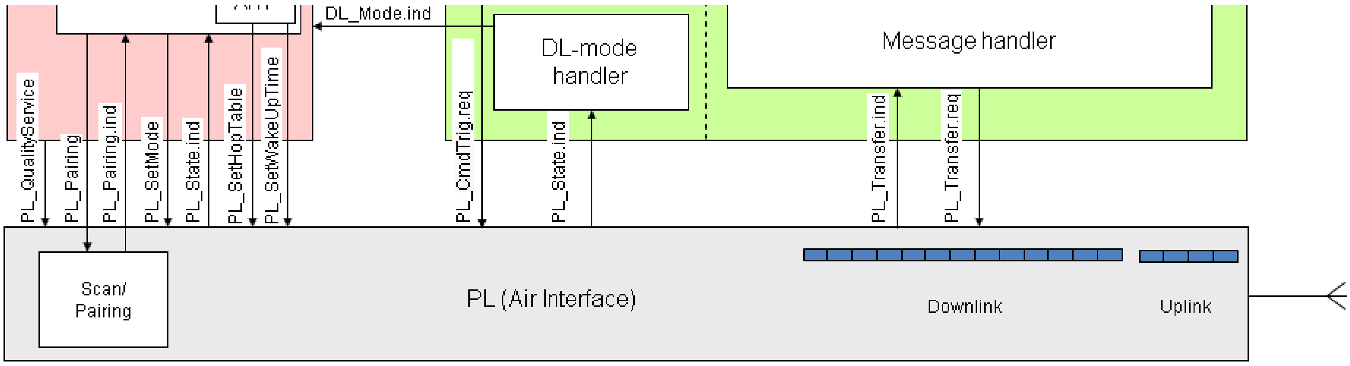


Figure 23 Physical layer (W-Device)

5.1 Base technology, Physical Layer (PL)

IO-Link wireless uses frequencies from 2401 to 2480 GHz of the license-free 2.4 GHz ISM band (industrial, scientific, and medical).
Physical Layer of IO-Link wireless is based on the proven technology used in Bluetooth© version 4.2 (Bluetooth Low Energy). It is therefore possible to use radios available on the market with the restriction that the requirements, described in the following clauses, being taken in account.

5.1.1 Transmission rate

The on-air bit duration T_{bit} is 1 µs as shown in Figure 24. Hence, the gross transmission rate is 1 Mbit/s.

5.1.2 Carrier frequency accuracy

The carrier frequencies f_c of a W-Master or a W-Device shall not deviate more than +/- 20 ppm.

5.1.3 W-Device Carrier frequency calibration

W-Device adjusts their carrier frequency to those of its W-Master. To adjust carrier frequency and compensate aging and thermal drifts, the W-Devices shall measure the frequency deviation during reception of each Downlink. This deviation is used by the W-Device for recalibration of its carrier frequency before each transmission.
If a W-Device is waiting on pairing request from a W-Master longer than two minutes on the configuration channel, it shall start to sweep its carrier frequency in frequency steps of +/- 25 kHz. Each frequency step is to be used four times before the next step. The maximum deviation of the sweep is +/- 250 kHz.

5.1.4 W-Master Carrier frequency calibration

The carrier frequencies of a W-Master should be calibrated to the defined carrier frequency accuracy during manufacturing.

5.1.5 Modulation

IO-Link wireless uses binary Gaussian frequency shift keying (GFSK) modulation scheme with a Gaussian filter bandwidth bit period product BT=0.5. The modulation index shall be 0.5.

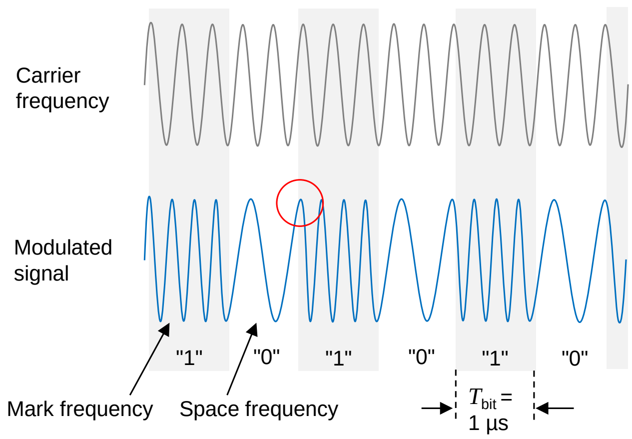


Figure 24 Base technology and modulation

Figure 24 shows an unmodulated carrier frequency and the binary frequency modulated signal. A binary one shall be represented by a positive frequency deviation, and a binary zero shall be represented by a negative frequency deviation.
The frequency transitions are non-linear (red circle in Figure 24) and cause interfering harmonics. A Gaussian filter reduces this impact. The entire modulation mechanism is named Gaussian Frequency Shift Keying (GFSK).

5.1.6 Transmission power

The transmission power shall meet FCC 15.247 and EN 300 328 for the use of the 2.4 GHz ISM frequency band. For this reason, the maximum transmission power of a W-Master or W-Device should not exceed a total of 10 mW. If at a W-Master more than one track is used, all tracks are sharing the 10 mW. Thereby the antenna gain shall be taken in account.
The output power shall be controlled by setting the attribute TransmitPower.

5.1.7 Antenna

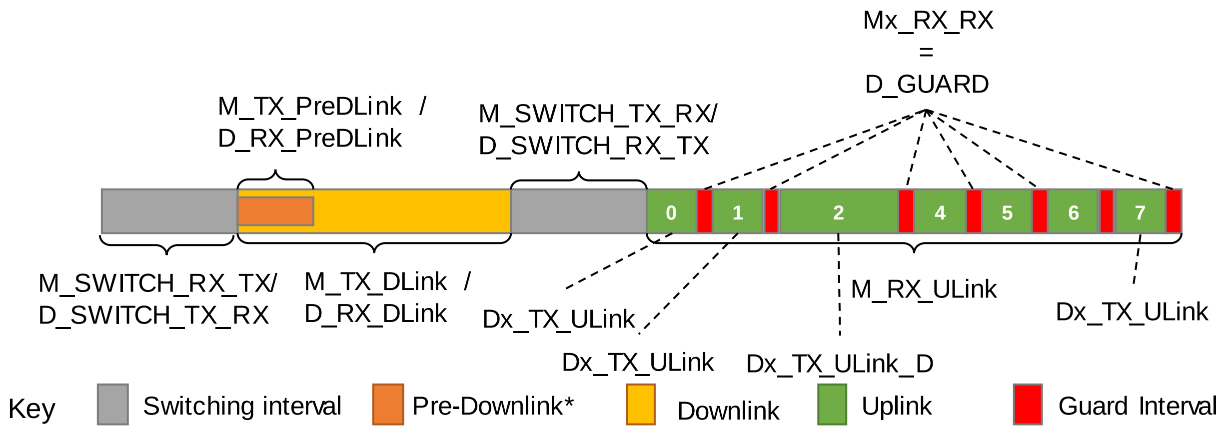
If radio regulations (see 5.1.6) are met, a W-Master or W-Device can use internal or external antennas. If an antenna with direction characteristic is used, also the maximum transmission power of ≤ 10 dBm EIRP shall be observed for any direction.

5.1.8 Receiver sensitivity

The minimal radio sensitivity on the W-Master and W-Device side shall be at least -94 dBm. With a transmission power of 2 mW, a guaranteed range of 10 meters and a message latency less than 5 ms with a remaining failure probability of less than 10^{-9} can be achieved like this.

5.1.9 Transceiver timings

To meet the necessary timings for the W-sub-cycle (See in clause 5.3) as shown in **Figure 25**Figure 25, switching between different transceiver states shall fulfill the requirements listed in Table 1.



*NOTE: Pre-Downlink for Low-Energy-Devices is not listed, since a Low-Energy-Device must be able to receive a complete Downlink.

Figure 25 Transceiver timings

Table 1 Transceiver timings within W-Sub-cycle

W-Master					
Name (see Figure **25**)	**Minimum**	**Typical**	**Maximum**	**Unit**	**Remark**
Oscillator accuracy	-20	0	20	ppm	The maximal oscillator deviation allowed
T_{BIT}	n/a	1	n/a	µs	Bit time at 1 Mbit/s transmission rate
M_SWITCH_RX_TX	-1	208	+1	µs	Time between the end of last Uplink and begin of next Downlink. Within this time, the W-Master transceiver shall change frequency channel and switch from receive (Rx) to transmit (Tx). The transmission of the Downlink shall start immediately after this time interval.
M_TX_DLink	n/a	416	n/a	T_{BIT}	The W-Master transceiver shall transmit a complete Downlink with 416 bits to all devices.
M_TX_PRE-DLink	n/a	88	n/a	T_{BIT}	The W-Master transceiver shall transmit a Pre-Downlink part of the complete Downlink with 88 bits to all W-Devices.
M_SWITCH_TX_RX	-1	208	+1	µs	The time between the end of Downlink and begin of Uplinks. Within this time the transceiver shall switch from transmit (Tx) to receive (Rx). The reception of the Uplinks shall start immediately after this time interval. NOTE: No change of frequency
M_RX_ULink	n/a	832	n/a	T_{BIT}	Receive of all separate W-Device Uplinks within a W-Sub-cycle on frequency of Downlink: only SSlot: 8 * (96 T_{BIT} + M_GUARD) only DSlot: 4 * (200 T_{BIT} + M_GUARD) or mix of SSlot and D-Slot NOTE: See Mx_RX_RX
Mx_RX_RX	n/a	8	n/a	T_{BIT}	Receive- to Receive-Time between two Uplinks except the last Uplink. e.g.: The W-Master transceiver receives an Uplink x. After this Uplink, the transceiver has this time to recover to Rx to receive next Uplink x+1. The recovery time shall be less them given time of 8T_{BIT}

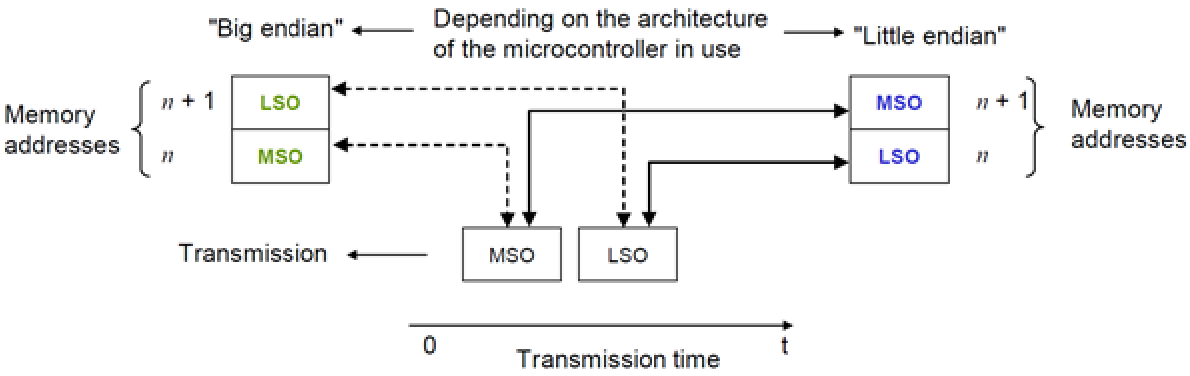
W-Device					
Name (see Figure **25**)	**Minimum**	**Typical**	**Maximum**	**Unit**	**Remark**
Radio frequency deviation	-250	0	250	kHz	The maximum carrier frequency error, which can be tolerated by radio
Frequency correction step	n/a	25	n/a	kHz	Frequency step used by correction of the carrier frequency error
T_{BIT}	n/a	1	n/a	µs	Bit time at 1 Mbit/s transmission rate
D_SWITCH_TX_RX	-1	208	+1	µs	Time between the end of Uplink of slot 7 and begin of next Downlink. Within this time, the W-Device transceiver shall change frequency channel and switch from transmit (Tx) to receive (Rx). The reception of the Downlink for each slot shall start immediately after this time interval.
D_RX_DLink	n/a	416	n/a	T_{BIT}	The W-Master transceiver shall transmit a complete Downlink with 416 bits to all devices.
D_RX_PPE-DLink	n/a	88	n/a	T_{BIT}	The W-Master transceiver shall transmit a Pre-Downlink with 88 bits to low energy devices.
D_SWITCH_RX_TX	-1	208	+1	µs	Time between the end of Downlink and begin of Uplink of slot 0. Within this time the W-Device transceiver shall switch from receive (Rx) to transmit (Tx). The time difference between the end of the time interval D_SWITCH_RX_TX and the start of the transmission for each Uplink can be calculated as following: Slot_N x [D_TX_ULink + D_GUARD] NOTE: No change of frequency
Dx_TX_ULink	n/a	96	n/a	T_{BIT}	Time a single slot W-Device sends its Uplink.
Dx_TX_ULink_D	n/a	200	n/a	T_{BIT}	Time a double slot W-Device sends its Uplink.
D_GUARD	n/a	8	n/a	µs	Guard time between two Uplinks. This prevents from "overlapping on air" of the W-Device Uplink before or after.
D_GUARD/2	n/a	4	n/a	µs	Uncertainty time by reception of the Downlink on W-Device side

5.2 Downlink and Uplink

5.2.1 Transmission octet order for WORD based data types

The values within the payload, independent of the architecture, transmitted in *Big Endian* format as shown in Figure 26. The following rule shall apply:

- The Most Significant octet (MSO) transmitted first.



Key
MSO = Most Significant octet
LSO = Least Significant octet

Figure 26 Memory Storage and transmission order for values for WORD based data types

5.2.2 Downlink and Uplink transmission

The bit ordering within each octet on the air follows the *Little-Endian* format. The Least Significant Bit (LSB) is the first bit, which shall be sent over the air for each octet. For instance, an 8-bit value 0x26(hex) (binary 0010 0110) is transmitted as shown in Figure 27.

b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
0	1	1	0	0	1	0	0

t_0

Time on air

Figure 27 Bit ordering within an octet

The radio transmits payload octets as an octet array over the air as shown in Figure 28.

octets on air

Figure 28The order of the octets is not altered during data transmission/reception.

octet 0	octet 1	....	octet N-1	octet N

octets on air

Figure 28 Octet array transmission over the air

5.2.3 Preamble

Each Downlink or Uplink always starts with the so-called "Preamble", a unique bit pattern. The two octets of the Preamble can contain either the value "0xAA" or "0x55". If the first bit of the syncword on air is "0" the preamble shall be set to "0xAA" otherwise the Preamble shall be set to "0x55". For example used syncword "0x59943E" the preamble shall be set to "0xAA". It shall be stored in the transmit buffer as shown in Figure 29.

First bit of syncword on air	Preamble octet 0	Preamble octet 1
0	AA	AA
1	55	55

Figure 29 Octet ordering of Preamble values

5.2.4 Syncword

The syncword immediately follows the Preamble. The syncword is required for octet synchronization and identification of the packet as an IO-Link wireless packet. The three octets long syncword shall be stored in to the transmit buffer directly after the preamble.

ConfigSyncword

This syncword is used for the configuration channels. The octets of the ConfigSyncword shall have the values and octet order shown in Figure 30.

ConfigSyncword octet 0	ConfigSyncword octet 1	ConfigSyncword octet 2
3E	94	59

Figure 30 Octet ordering of ConfigSyncword

DataSyncword

This syncword is used for the data channels. In this version of the system extension, the octets of the DataSyncword shall have the values and octet order shown in Figure 31 . This DataSyncword shall be transmitted on Pairing Request from the W-Master to the W-Device (see Figure 141). By this mechanism other DataSyncwords are possible for future versions. Currently only the DataSyncword from Figure 31 shall be used.

DataSyncword octet 0	DataSyncword octet 1	DataSyncword octet 2
3E	94	59

Figure 31 Octet ordering of DataSyncword

5.2.5 Downlink and Uplink CRC

CRC are necessary to avoid reception of a wrong message as a right one. Each Pre-Downlink, Downlink and Uplink has a CRC at the end to check its consistence after wireless transmission. The Pre-Downlink CRC has a length of 16 Bit. The Full-Downlink and all Uplinks have a CRC length of 32 Bit. To get the same probability of a correct message for Uplinks and the Full-Downlink they need a longer CRC due to of their data length.

5.2.6 CRC Transmission

The result of the CRC16 and CRC32 shall be stored in a *Big Endian* format in the transmit buffer. See Figure 32 and Figure 33.

octet 0	octet 1
CRC (15:8)	CRC (7:0)

Figure 32 Octet ordering of CRC16 result values

octet 0	octet 1	octet 2	octet 3
CRC (31:24)	CRC (23:16)	CRC (15:8)	CRC (7:0)

Figure 33 Octet ordering of CRC32 result values

5.2.7 Data Whitening

Before transmission and after receiving, the W-Frame is scrambled/descrambled with a data whitening polynomial in order to randomize the data from highly redundant pattern and to minimize DC bias in the W-

Frame. IO-Link wireless shall use the same whitener as the Bluetooth 4.2 with the polynomial shown in Equation 1.

P = X^7+ X^4 + 1

Equation 1 Whitening Polynomial

The Figure 34 shows the realization of the whitening pseudo random number generator using a shift register with a feedback:

Data Whitening LFSR

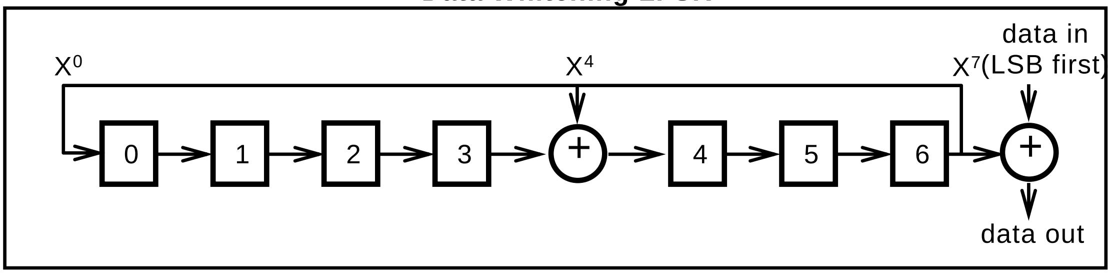


Figure 34 Data Whitening LFSR

5.2.8 Regular Downlink

The data structure of the Regular Downlink is shown in Figure 35. The distribution of payload of Pre-Downlink and Full-Downlink is dynamically assembled by Message handler (see clause 12). The data structure of the Downlink is described in clause 13.2 in detail.

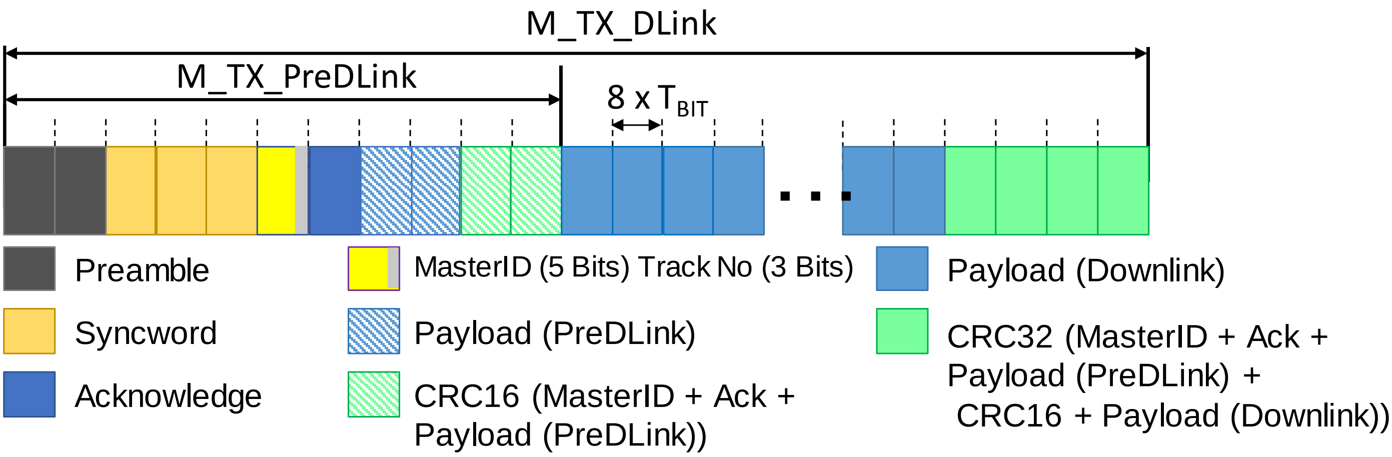


Figure 35 Regular Downlink

5.2.9 Configuration Downlink

The data structure of the Configuration Downlink is shown in Figure 36. The content of the payload is compiled by the Physical Layer (see clause 12). Data structure of the configuration Downlink is described in 13.3. in detail.

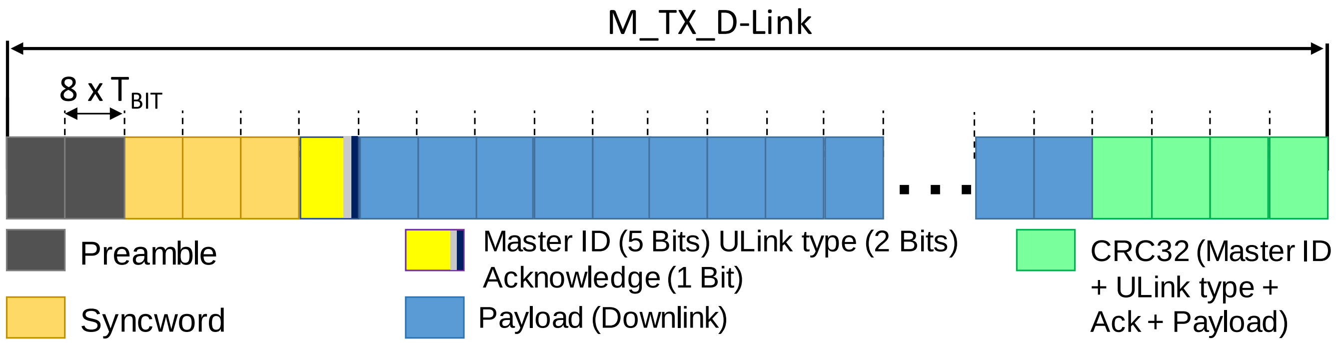


Figure 36 Configuration Downlink

5.2.10 Uplink Single Slot (SSlot)

The data structure of the SSlot Uplink is shown in Figure 37. The DL-A handler compiles the Uplink payload. There are maximal 8 Slots possible per track of a W-Frame.

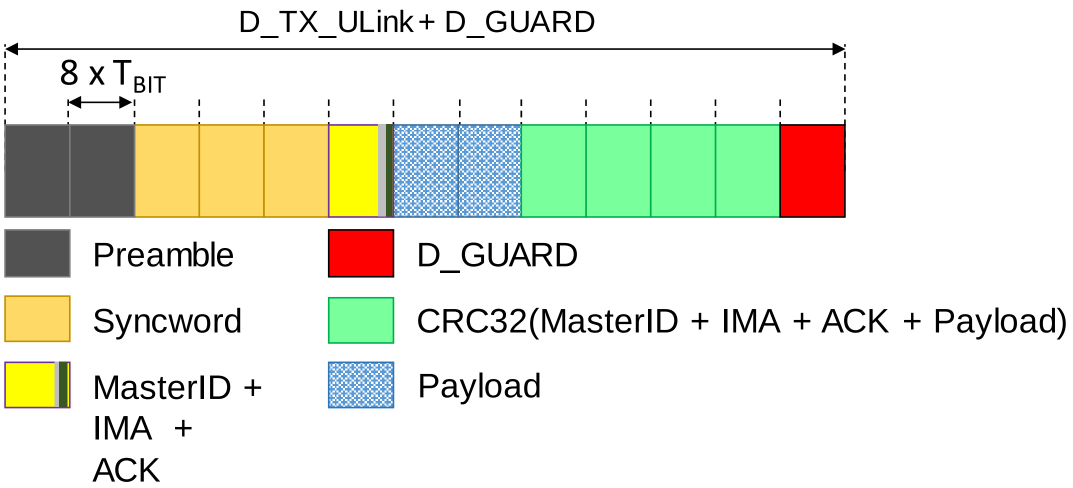


Figure 37 Uplink - SSlot

5.2.11 Uplink Double Slot (DSlot)

The data structure of the DSlot Uplink is shown in Figure 38. The DL-A handler compiles the Uplink payload. There are maximal 4 D-Slot possible per track of a W-Frame.

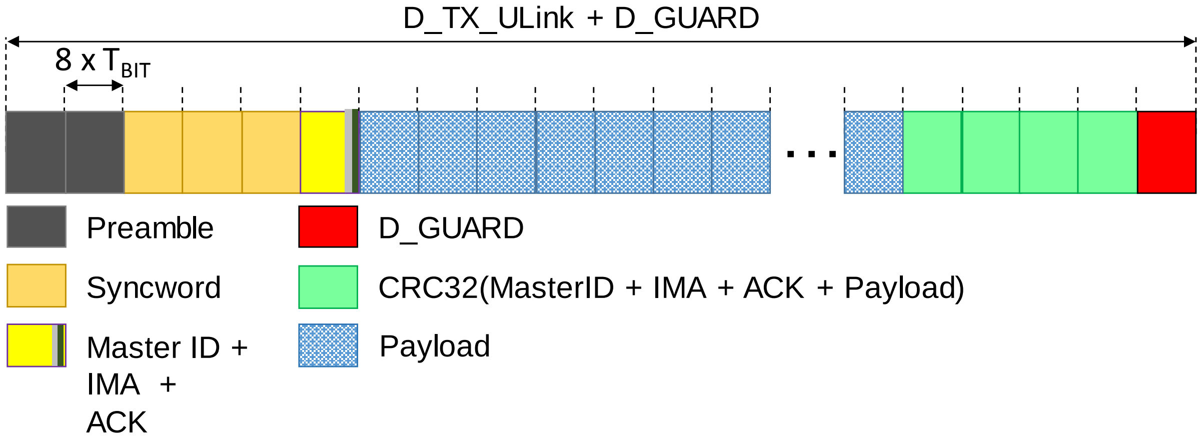


Figure 38 Uplink - DSlot

5.3 W-Sub-cycle

The general concept of the W-Cycle and the W-Sub-cycles is specified in Figure 20. The following subclauses explain definitions for packets within a W-Frame.

5.3.1 W-Sub-cycle structure

A W-Sub-cycle describes a time frame with a duration of 1.664 ms, because the minimum cycle time shall be shorter than 5 ms. For this purpose the W-Sub-cycle has a length of 1.664 ms => 3 x cycle < 5 ms. In a W-Sub-cycle a complete communication exchange between a W-Master and its W-Devices is organized (see Figure 39 Format of a W-Sub-cycle with DSlots). The detailed encoding of W-Messages within W-Sub-cycles are described in clause 12.

The first part of a W-Sub-cycle is a control interval of 208 µs. In this interval, the carrier frequency and transceiver mode are adjusted. After that the so-called "Downlink" starts. The Downlink has a duration of 416 µs and can contain individual W-Message(s) for each W-Device, e.g. W-Device in Slot_N 2 in Figure 39.

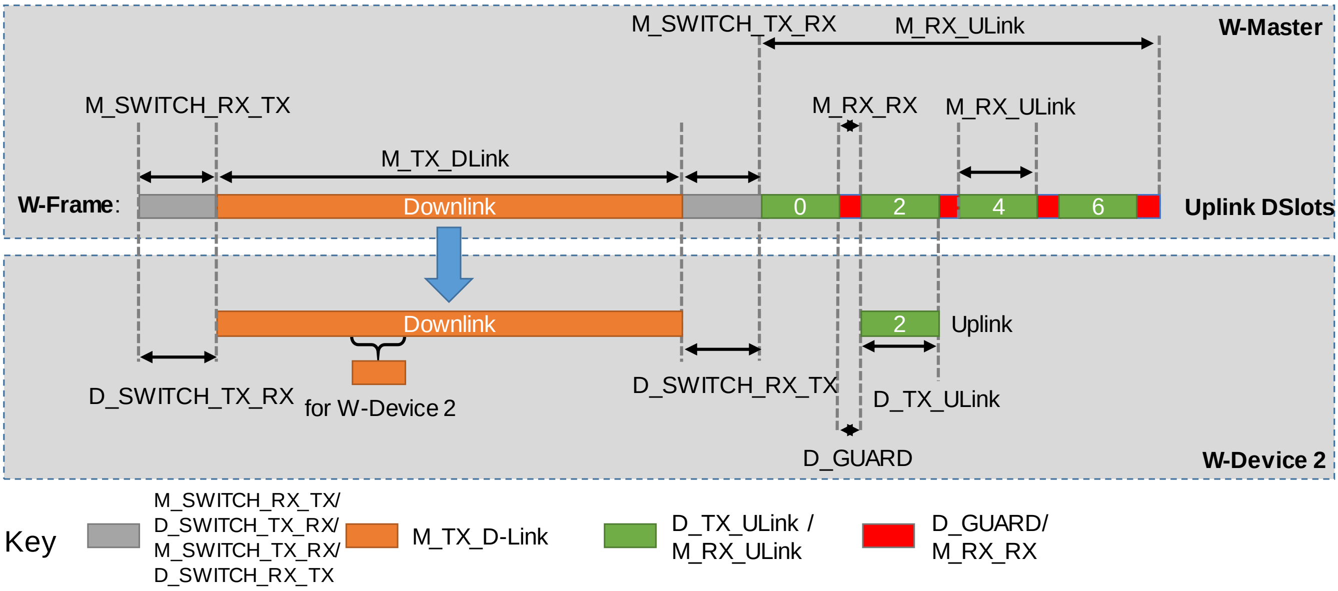


Figure 39 Format of a W-Sub-cycle with DSlots

After the control interval during which the transceivers of the W-Master switches from "transmit" (TX) to "receive" (RX) and of the W-Device vice versa, the Uplink with a total duration of 832 µs starts. In the "Uplink" each W-Device has its own time slot to response, e.g. Slot_N 2 for W-Device 2 in Figure 39. Between sequentially Uplink slots, a guard interval with a duration of 8 µs is placed. At the beginning of the guard interval the previous W-Device stops sending, while the following W-Device starts sending at the end of the guard interval. The guard interval is required for the W-Master to recover. A W-Device can use two kinds of Slots in an Uplink with different duration, Single Slots (SSlot) with 96 µs (see Figure 37) or Double Slots (DSlot) with a length 200 µs (see Figure 38). Only by using SSlots, the maximum number of 8 W-Devices per track can be achieved. DSlots shall always start at an even slot number. If in a track an odd number of SSlots is used one SSlot cannot be used e.g. SSlot 5 in Figure 39.

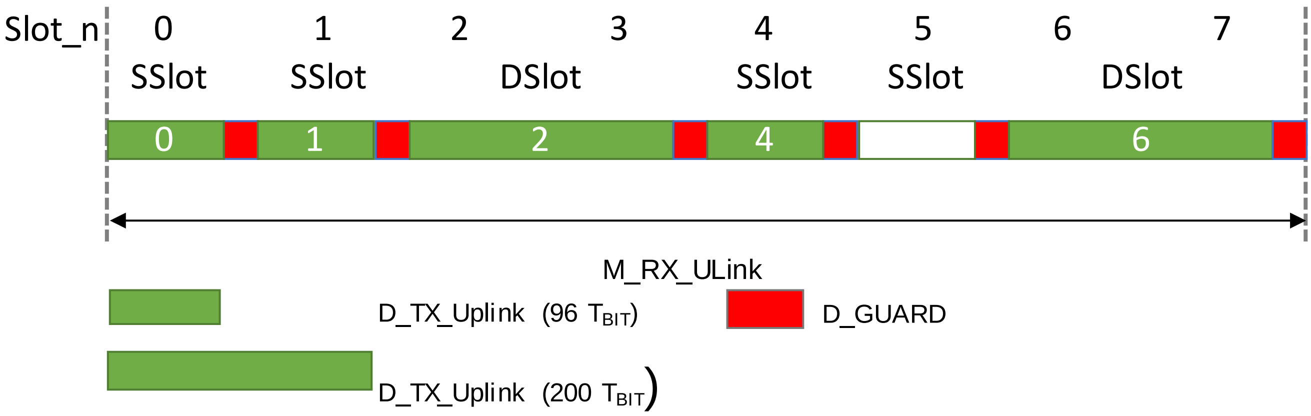


Figure 40 SSlots and DSlots

5.3.2 Regular W-Frame

Figure 41 shows the structure of a regular W-Frame, which is used for cyclic transmission of IO-Link Process Data (PD) and acyclic transmission of On-request Data (OD). This W-Frame can contain multiple W-Messages in its Downlink section addressed to dedicated W-Devices. The first part of the Downlink, the so-called Pre-Downlink, is integral part of the full Downlink for regular W-Frames. It contains the acknowledgments and two octets payload. Pre-Downlink has its own 16 bit CRC signature. low energy W-Devices may reduce their receiver activity time by only receiving the Pre-Downlink instead of the full Downlink.

All other W-Devices of the track shall receive the entire Downlink section and the 32 bit CRC signature at the end.

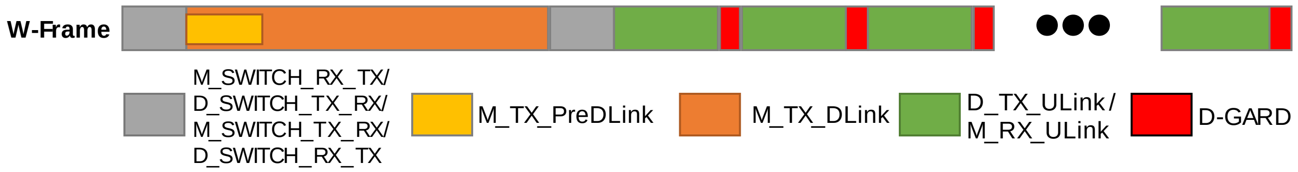


Figure 41 W-Frame structure

5.3.3 Configuration W-Frame

During pairing and configuration via the configuration channel, the W-Master communicates with only one W-Device. This allows using the entire Downlink space for the transfer of ConnectionParameters. There is no Pre-Downlink encoded in this frame type. Consequently, only the addressed W-Device returns a message within the Uplink section (see Figure 42).

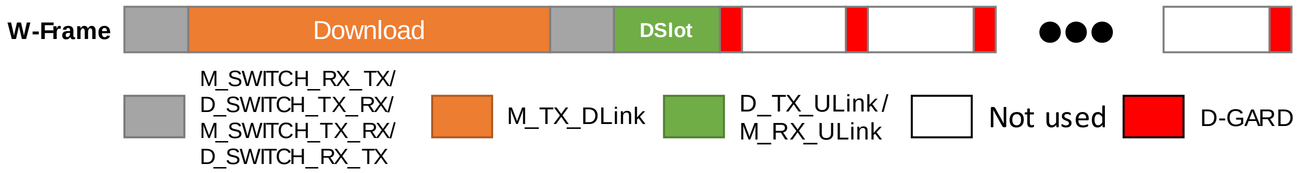


Figure 42 W-Frame type for pairing and configuration

5.4 Medium Access Control (MAC)

This clause describes the relevant definitions for media access on both, W-Master and W-Devices, which must comply with the requirements described below.
W-Devices and W-Master shall operate in synchronous manner in frequency and time domain. Therefore, the synchronization of the W-Devices on a W-Master is necessary. Initial synchronization of the W-Device on its W-Master takes place during the pairing process. A paired W-Device resynchronizes its timing on each successful reception of the Downlink. The W-Devices calculates the next Downlink transmission time and adds a window of uncertainty of 4 µs (DGUARD/2) to control its receiver activity.
W-Devices after a longer inactivity period might loose clock synchronization with their W-Master. In this case it is required to increase the uncertainty window of the W-Device, thus increasing the receiver on-time.
The use of orthogonal frequency channel hopping sequences by the W-Masters and their associated W-Devices allows operational coexistence of overlapping IO-Link wireless systems. The W-Master creates the hopping sequences. For increasing capability of W-Coexistence, the frequency channel hopping sequences can be adapted to environment using Blacklisting. During the pairing and configuration processes, the W-Master downloads these hopping sequences into the unpaired W-Devices.

5.4.1 Frequency channels

The carrier frequencies f_n are defined by the frequency channel number n using Equation 2

$$fn = f_0 + n \times 1\,MHz$$

Equation 2 Carrier frequencies

where

f_0 = 2400 MHz

n = 3 to 78.

The minimum spectral distance between the tracks of a W-Master is 3 MHz. The bandwidth of a modulated carrier frequency is less than 1.1 MHz.

5.4.2 Default Hopping Table HT01

IO-Link wireless defines the frequency channel hopping table HT01. HT01 omits the frequency channels f_{1-2} and f_{79-80}. The frequency channels f_1 and f_{80} are reserved for configuration (see 5.4.4). Hopping table HT01 is organized in rows and columns. In a column, all frequency channels used by a W-Master and its W-Devices within a W-Sub-cycle are listed. In a row, the sequence of frequency channels used by a track of a W-Master and its W-Devices is listed. HT01 additional allows blacklisting of each 1 MHz frequency channel (see 5.4.5).

The frequency hopping sequence of all tracks of overlapping W-Masters shall be orthogonal to avoid transmission collisions within a W-Master. Therefore, in a cell with three W-Masters, the probability of collisions by chance is sufficiently low.
Each track of a IO-Link W-Master shall keep the hopping tables of all other tracks of this W-Master, which enables a (e.g. roaming) track to pair a W-Device to another target track of this W-Master.Therefore the target track can stay in cyclic mode without loss of performance during pairing.

The sequence of frequency channels in HT01 is determined by the HT01 parameters listed in Table 2.

Table 2 HT01 parameter

HT01 parameter	Definition	Remark
Col_N	Column number within the frequency hopping table HT01	The frequency channels of the sequence listed column by column. See 18.2
MasterID	MasterID: The ID the W-Master is assigned to	
Blacklist	List of unused frequency channels	An 80-bit word each bit representing a frequency channel. See Section 5.4.5
Track Number	Ensures that all tracks of the W-Master have non-overlapping frequency tables	
Frequency Spacing	Ensures the interference between the tracks within a W-Master are minimal	For this purpose, the minimal space between the channels within a track frequency group must be greater or equal to 3 MHz but not greater than 5 MHz.

The Hopping Sequence is calculated in the IO-Link Wireless Master according to the following algorithm:

- Determine possible frequencies for the tracks
- Build non-overlapping groups of frequencies
- Build the hopping sequence depending on the MasterID

See clause 18.2 for calculation rules and examples.

5.4.3 Alternative Hopping algorithms

Hopping algorithms other than for calculation of HT01 may be specified in further versions of this specification.

5.4.4 Configuration Frequencies

The frequencies f_1 and f_{80} (i.e. 2401 MHz and 2480 MHz) are exclusively reserved for configuration channels. They shall be used in an alternating manner to reduce frequency related interferences. The configuration frequencies cannot be blacklisted. Clause 18.3 describes their utilization in detail.

5.4.5 Blacklisting

In order to mitigate interference from or to other devices in the 2.4 GHz ISM band affected frequency channels can be omitted by appending them to the blacklist (see Table 183). It should be taken in account, that the reduction of available frequency channels may compromise latency in a non-deterministic manner. See clause 18.2 for calculation rules and examples.

5.4.6 Link Quality Indication

Link Quality Indication is a service for evaluation of the functionality and reliability of the IO-Link Wireless System in certain application environments. This service should be used during commissioning or significant changes during the running period of the IO-Link Wireless System. Optional it can be used during operating mode of the IO-Link Wireless system for monitoring the wireless environment regarding reliability. To analyze connection quality independent from RSSI the Link Quality Indication shall be evaluated on each W-Port. Therefore, the first order Remaining Failure Probability (RFP(1)) is used to calculate Packet Error Probability (PEP) of the last 3000 Packets with Equation 3

Equation 3 Remaining Failure Probability

$$RFP = PEP_{(1)}^{(1+MaxRetry)}$$

The RFP can be described in percent using values from Table 3.

Table 3 Link Quality Indication

RFP	Link Quality
10^{-9}	100 %
$5x10^{-9}$	80 %
10^{-8}	60 %
$5x10^{-8}$	40 %
$8x10^{-8}$	20 %

5.5 Physical Layer PL services

5.5.1 Overview

An overview of the Physical Layer and its service primitives is given in Table 4 and Figure 44. They are the interface to the higher protocol layers.

5.5.2 PL Services for W-Master

Subsequently, the services are specified which are provided by the PL to System Management and to the Data Link Layer (see Figure 22 for a complete overview of all the services). Table 4 lists the assignments of W-Master to their roles as initiator or receiver for the individual PL services.

Table 4 PL Service assignments of W-Master

Service name	Master
PL_SetTrackConfig	R
PL_SetMode	R
PL_Scan	I
PL_ScanEnd	I
PL_SetSlotConfig	R
PL_Pairing	I / R
PL_State	I
PL_Transfer	I / R
PL_QualityService	R
PL_GetHopTable	R
PL_SetHopTable	R
PL_SetWakeUpTime	R
PL_WakeUpTime	I
PL_AHTStatus	I
PL_CmdTrig	R
Key (see 3.3.5) I Initiator of service R Receiver (Responder) of service	

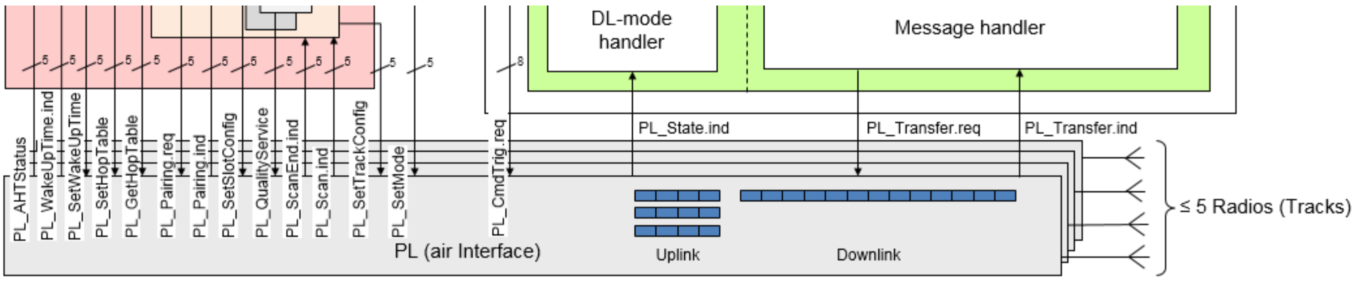


Figure 43 Physical Layer services of the W-Master

5.5.2.1 PL_SetTrackConfig (W-Master)

The PL_SetTrackConfig service is used to setup the initial parameter for each track on a W-Master. The parameters of the service primitives are listed in Table 5.

Table 5 PL_SetTrackConfig

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

ParameterList

This parameter contains the configured parameters of a Track.

Parameter Type: Record

Record Elements:

MasterID: This parameter contains the MasterID of the W-Master (see Table 140)
Permitted values: 1-29

BlackList: This parameter contains the frequency channels which shall not be used by the W-Master.
Permitted values: 0x0000 0000 0000 0000 0000 to 0x7FFF FFFF FFFF FFFF FFFE
(bitwise coded 1MHz channels 2-79 (LSB first))

Track_N: This parameter set up the internal number of a track for calculation of track-dependent hopping sequence.
Each Track shall be numbered consecutively within a W-Master. Permitted values: 0-4

SyncTrack: This parameter defines, whether the Track is running as W-Frame SyncMaster or SyncSlave. The synchronization signal is transferred via a hardware pin to tracks configured as SyncTrack (SLAVE).
Permitted values:
MASTER (The track generates the hardware synchronization-signal on each start of W-Sub-cycle)
SLAVE (To start its W-Sub-cycle the track is waiting for the hardware synchronization signal, which is generated by the track configured as SyncTrack(MASTER))

DataSyncword

DataSyncword which shall be used for data channels (see clause 5.2.4)

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.2 PL_SetMode (W-Master)

The PL_SetMode service is used to setup the mode of a track and configuration for transmission power of the physical layer. This service can also be called during runtime to change the parameters (State ≠ Idle_0). The parameters of the service primitives are listed in Table 6.

Table 6 PL_SetMode

Parameter Name	.req	.cnf
Argument	M	
TrackMode	M	
TxPower	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

TrackMode:

This parameter indicates the requested operational mode of the radio (see Table 7)
Permitted values: STOP, CYCLIC, SCAN, ROAMING

TxPower:

This parameter indicates the transmission power level of the track.
Permitted values: 1 to 255 (See 10.9 for definition)

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information. Permitted values:

STATE_CONFLICT (service unavailable within current state)

PARAMETER_CONFLICT (consistency of parameter set violated)

Table 7 specifies the coding of the different parameters.

Table 7 Definition of parameters for Service PL_SetMode

TargetMode	Definition
STOP	Communication disabled, radio turned off
SCAN	W-Master is working in Scan mode. (Limited performance)
ROAMING	W-Master is working in Roaming mode. (Limited performance)
CYCLIC	W-Master is working in Cyclic mode. (Full performance)

5.5.2.3 PL_Scan (W-Master)

The PL_Scan service is used to report a new unpaired W-Device within the track's proximity via indication. This is only initiated by PL if the track is in ROAMING or SCAN mode. The parameters of the service primitives are listed in Table 8.

Table 8 PL_Scan

Parameter Name	.ind
Argument ParameterList	M S

Argument:

The service-specific parameters are transmitted in the argument.

ParameterList

This parameter contains the information of the found W-Device.

Parameter Type: Record

Record Elements:

SlotType: Type of the W-Device in Uplink

Permitted values: SSLOT, DSLOT (see Table 21)

UniqueID: This parameter indicates the UniqueID of the Device. (see Figure 156)

RevisionID: This parameter indicates the protocol version of the found W-Device. (see Figure B.4 in REF 1)

5.5.2.4 PL_ScanEnd (W-Master)

The PL_ScanEnd service is used to indicate the end of the scan mode. The parameters of the service primitive are listed in Table 9.

Table 9 PL_ScanEnd

Parameter Name	.ind
<none>	

5.5.2.5 PL_SetSlotConfig (W-Master)

The PL-SetSlotConfig service is used to setup the slot configuration for a W-Device. If the connection to W-Device is established, only IMATime shall be changed. If the connection to W-Device is not established, all parameters can be changed.

The parameters of the service primitives are listed in Table 10.

Table 10 PL_SetSlotConfig

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

ParameterList

Parameter Type: Record

UniqueID: This parameter contains the UniqueID of the W-Device (see Figure 156)

SlotType: Type of the W-Device in Uplink given through W-Device application.
Permitted values: SSLOT, DSLOT (see Table 161)

Slot_N: This parameter contains the Slot number for the corresponding W-Device
Permitted values: 0-7. Each DSLOT (only on even Slots allowed) occupies 2 SSLOT's.

IMATime

This parameter contains the I am alive time (see clause 14.3.5) for the corresponding Slot/W-Device (see 10.3 to detect COMLOST).

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.6 PL_Pairing (W-Master)

This service is used to pair or unpair a W-Device from the W-Master via system management. The parameters of the service primitives are listed in Table 11.

Table 11 PL_Pairing

Parameter Name	.req	.cnf	.ind
Argument	M		M
ParameterList	M		
Info			M
Result (+)		S	
Result (-)		S	
ErrorInfo		M	

Argument:

The service-specific parameters are transmitted in the argument.

ParameterList

Parameter Type: Record

UniqueID: This parameter contains the UniqueID of the W-Device (see 14.3.9)

Track_N: This parameter selects the track number where the W-Device should be assigned to.
Used to pair a W-Device to another track for example when one track is in roaming mode.

SlotType: Type of the W-Device in Uplink given through W-Device application.
Permitted values: SSLOT, DSLOT (see Table 21)

Slot_N: This parameter contains the Slot number for the corresponding W-Device

Permitted values: 0-7 SSLOT's. Each DSLOT (only on even Slots allowed) occupies 2 SSLOT's.

Method: This parameter requests the pairing mode which shall be used.

Permitted values:

PAIRING_BUTTON (PL shall use the W-Frame Figure 141 to pair a W-Device via button method)

PAIRING_UNIQUE (PL shall use the W-Frame Figure 141 to pair a W-Device via U-ID)

PAIRING_ABORTED (pairing is stopped by the W-Master application)

UNPAIRING (PL issues the MasterCommand "Unpairing" and clears the configuration of the slot given in Slot_N. No further ULinks can be received)

TargetMode: This parameter requests the mode of the W-Device to be paired

Permitted values: CYCLIC, ROAMING

Timeout: This parameter contains the timeout for a pairing attempt in seconds. See Table 186 (definition of PAIRING_BUTTON_TIMEOUT, PAIRING_UNIQUE_TIMEOUT)

Info

Permitted values:

PAIRING_SUCCESS (Device has been paired)

PAIRING_TIMEOUT (Device was not paired within the time given in Timeout)

PAIRING_WRONG_SLOTTYPE (The Device cannot support the requested SlotType)

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information.

Permitted values:

STATE_CONFLICT (service unavailable within current state)

PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.7 PL_State (W-Master)

The PL_State service is used to signal the state of a running or lost connection for the W-Device on the corresponding SSLOT or DSLOT. The parameters of the service primitives are listed in Table 12.

Table 12 PL_State

Parameter Name	.ind
Argument	M
PLInfo	M

Argument

The service-specific parameters are transmitted in the argument.

PLInfo:

This parameter contains the bit coded status of the connection for each Slot.

Bit 0 represents Slot_N 0. Bit 7 represents Slot_N 7

Bitvalues: 0: COMLOST (Device has no or lost connection to its Master)

1: SYNCED (Device is synchronized with its Master)

5.5.2.8 PL_Transfer (W-Master)

The PL-Transfer service is used to exchange the data between Data Link Layer and Physical Layer. The generation of the ACK-Bits for each W-Device is handled in PL (see 13.6). The parameters of the service primitives are listed in Table 13

Table 13 PL_Transfer

Parameter Name	.req	.ind
Argument	C	M
PreDlData	M	
Data	M	C
DataLength	M	C
ULinkType		C
Slot_N		C
Acknowledge		C
WFrameComplete		C
Result (+)		
Result (-)		
ErrorInfo		

Argument
The service-specific parameters are transmitted in the argument.

PreDlData
This parameter contains the data of the Pre-Downlink
DataLength 2 octet

Data
This parameter contains the data which are transferred from / to the PL (radio interface). Data contains one or more W-Message(s) (Control Octet + corresponding data).

DataLength
This parameter contains the length of transmitted data, dependent on the direction (DLink or ULink) and the uplink type.
Ranges: PL_Transfer.req: up to 37 octets in FULLDOWNLINK (data from master to W-Device)
PL_Transfer.ind: 2 octets (data from W-Device to master, SSlot-Format)
PL_Transfer.ind: 15 octets (data from W-Device to master, DSlot-Format)

ULinkType:
This parameter contains the type of ULink. Permitted values:
DATA (regular ULink received, see 13.4. Regular Uplink Frame Annex B).
NOUPLINK (No ULink received)
IMA (IMA ULink received, see Figure 146 and Figure 147 and IMA-Uplink Frame Annex B).

Slot_N:
This parameter contains the Slot_N to assign the received ULink to the corresponding W-Port (see 6.1.1 TD-Mapper)

Acknowledge
This parameter indicates, whether the last DLink has been confirmed by W-Device or not. PD handler, Event handler and OD handler needs the Acknowledge from PL to decide, if a retransmit of data for the corresponding W-Device is needed or not.

WFrameComplete:
This parameter indicates that the W-Frame has been completed (all ULinks have been processed). The Message handler needs this information to start the assembly of the next Downlink.

Result (+):
This selection parameter indicates that the service request has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

StatusErrorInfo

This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

5.5.2.9 PL_QualityService (W-Master)

The PL_QualityService is used to request the actual link-quality of the wireless connection between the W-Master-track and the corresponding W-Device. The Service responds with the link quality in percent for each W-Device (see clause 5.4.6). The parameters of the service are listed in Table 14.

Table 14 PL_QualityService

Parameter Name	.req	.cnf
Argument	M	
Slot_N	M	
Result (+)		S
Quality		M
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
Slot_N: This parameter indicates the selected Slot_N with its corresponding W-Device.
Permitted values: 0 to 7.
Result (+):
This selection parameter indicates that the service has been executed successfully.
Quality
Parameter type: Octet
Permitted Values: 0 to 100%.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

5.5.2.10 PL_GetHopTable (W-Master)

The PL_GetHopTable service is used to get the hopping table from the PL to the AHT handler in SM for channel monitoring, see clause 9.2.3.2.4. The parameters of the service primitives are listed in Table 15.

Table 15 PL_ GetHopTable

Parameter Name	.req	.cnf
Argument	M	
Track_N	M	
Result (+)		S
Data		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
Track_N: This parameter selects the track number of which to read the hopping table values. Used to read all the hopping table values from any track, see clause 5.4.2.
Permitted values: 0-4
Result (+):

This selection parameter indicates that the service has been executed successfully.

Data

Parameter type: 78 Octets

This parameter contains the used hopping table sequence in PL (see clause 13.3.3).

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information. Permitted values:

STATE_CONFLICT (service unavailable within current state)

PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.11 PL_SetHopTable (W-Master)

The PL_SetHopTable service is used to set the new hopping table in W-Master PL track. The parameters of the service primitives are listed in Table 16.

Table 16 PL_ SetHopTable

Parameter Name	.req	.cnf
Argument	M	
Track_N	M	
UpdateType	M	
Index	M	
Data		
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

Parameter Type: Record

Track_N: This parameter contains the track number of which to set the hopping table values. Used to set the hopping table values for any track, see clause 5.4.2.

Permitted values: 0-4

UpdateType: This parameter contains the type of update, for the usage of UpdateType, Index and data see 18.4. Permitted values:

FULL_TABLE

DELETE_CELL

ADD_CELL

REPLACE_CELL

Index: This parameter contains the index of the changed cell in the hopping table. Permitted values: 0, 1-78

Data: This parameter contains the value/s to replace/add.

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information. Permitted values:

STATE_CONFLICT (service unavailable within current state)

PARAMETER_CONFLICT (consistency of parameter set violated)

PL_SetWakeUpTime (W-Master)

The PL_SetWakeUpTime service is used to set a countdown in the PL with the value WakeUpTime. The parameters of the service primitives are listed in Table 17.

Table 17 PL_SetWakeUpTime

Parameter Name	.req	.cnf
Argument	M	
WakeUpTime	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
WakeUpTime: contains the WakeUpTime in W-Sub-cycles to set in PL.
Permitted values: 0 to 16777215 (3 Octets).
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.12 PL_WakeUpTime (W-Master)

The PL_WakeUpTime service indicates the current WakeUpTime from the PL. The AHT handler sends the WakeUpTime to the corresponding W-Device via ISDU. The parameters of the service primitives are listed in Table 18.

Table 18 PL_WakeUpTime

Parameter Name	.ind	.rsp
Argument	M	
WakeUpTime	M	
Slot_N	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
WakeUpTime: contains the WakeUpTime in W-Sub-cycles to set to the W-Device.
Permitted values: 0 to 16777215 (3 Bytes).
Slot_N: This parameter indicates the selected Slot_N with its corresponding W-Device.
Permitted values: 0 to 7.
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.13 PL_AHTStatus (W-Master)

The PL_AHTStatus service is used to indicate to the AHT handler the status of hopping table update, see 9.2.3.2.4. The parameters of the service primitives are listed in Table 19.

Table 19 PL_ AHTStatus

Parameter Name	.ind	.rsp
Argument	M	
Status	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
Status: contains the update status. Permitted values:
JUMP_SUCCESS (Update completed successfully)
WAKE_UP_ABORT (low energy W-Device did not wake up)
JUMP_FAIL (W-Device did not acknowledge JUMP command)
STOP (PL track has stopped)
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.2.14 PL_CmdTrig (W-Master)

The PL_CmdTrig service is used to communicate real time actions to the PL from the command handler, see 18.4. The parameters of the service primitives are listed in Table 20.

Table 20 PL_CmdTrig

Parameter Name	.req	.cnf
Argument	M	
Command	M	
Result (+)		S
WakeUpTime		S
JumpAction		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
Command: contains the action to perform in PL. Permitted values:
WAKE_UP_TIME (triggers the PL to deliver the current WakeUpCountdown value)
W_DEVICE_AWAKE (indicates low energy W-Device sent IMA at WakeUpTime)
W_DEVICE_NOT_AWAKE (indicates low energy W-Device did not send IMA)
JUMP (switch to new hopping table HT02, starting with Hop-1 frequency)
JUMP_FAIL (W-Device did not acknowledge JUMP command)
Result (+):
This selection parameter indicates that the service has been executed successfully.

WakeUpTime: WakeUpTime value of the corrosponding low energy W-Device in W-Sub-cycles, received for WAKE_UP_TIME command.
Permitted values: 0 to 16777215 (3 Octets).
JumpAction: informs the command handler if low power W-Devices are awake and which action to invoke, received for W_DEVICE_AWAKE/NOT_AWAKE command.
Permitted values:
JUMP (all low energy W-Devices are awake)
WAKE_UP_ABORT (a low energy W-Device did not wake up)

Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.3 PL Services for W-Device

Subsequently, the services are specified which are provided by the PL to System Management and to the Data Link Layer (see Figure 23Figure 59 for a complete overview of all the services). Table 21 lists the assignments of W-Device to their roles as initiator or receiver for the individual PL services.

Table 21 PL Service assignments of W-Device

Service name	W-Device
PL_SetMode	R
PL_Pairing	I / R
PL_State	I
PL_Transfer	I / R
PL_QualityService	R
PL_SetHopTable	R
PL_SetWakeUpTime	R
PL_CmdTrig	R
Key (see3.3.5) I Initiator of service R Receiver (Responder) of service	

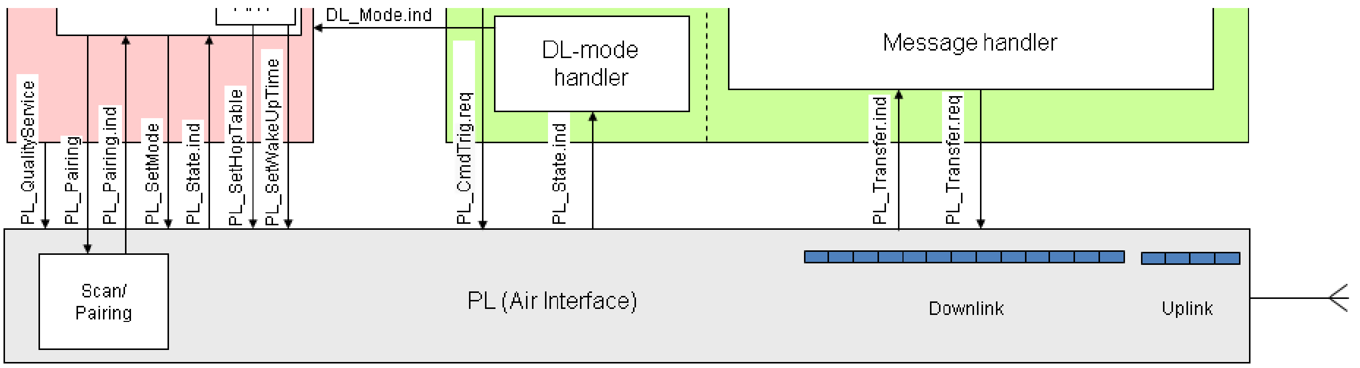


Figure 44 Physical Layer services of the W-Device

5.5.3.1 PL_SetMode (W-Device)

The PL-SetMode service is used to setup the radio characteristics and configurations for startup of the Physical Layer.
This service can also be called during runtime (State ≠ Idle_0) to change the following parameters only: DownlinkType, TxPower and MaxRetry. The parameter “TargetMode: STOP” can also be called during runtime to deactivate radio. All other parameters shall be ignored during runtime. The parameters of the service primitives are listed in Table 22.

Table 22 PL_SetMode (W-Device)

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
This parameter contains the configured identification parameter for the W-Device’s PHY and MAC Layer.
ParameterList
Parameter Type: Record
Record Elements:
TargetMode: This parameter indicates the requested operational mode of the radio (see Table 23)
Permitted values: STOP, START
UniqueID: This parameter contains the UniqueID of the W-Device (see Figure 156)
SlotType: Type of the W-Device in Uplink given through W-Device application. Permitted values: SSLOT, DSLOT (see Table 161)
DownlinkType: Type of the W-Device in Downlink given through W-Device application. Permitted values: PRE_DOWNLINK, FULL_DOWNLINK (see Table 23)
TxPower: Permitted values: 1 to 31 (See Table 176)
MaxRetry: Permitted values: Table 175.
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

Table 23 specifies the coding of the different Parameters

Table 23 PL_SetMode coding of Parameters

Parameters	Definition
STOP	Communication disabled, radio turned off
START	Start radio in Cyclic Mode. W-Device is or can be paired to a W-Master permanently or temporarily (Method shall be selected by W-Master)
PRE_DOWNLINK	W-Device is listening for a pre-downlink (reduced receive-on time for low energy devices) only when connected
FULL_DOWNLINK	W-Device is listening for a full-downlink when connected
SSLOT	W-Device send its uplink in SSLOT format (see Figure 144 Regular SSlot Uplink Packet)
DSLOT	W-Device send its uplink in DSLOT format (see Figure 145 Regular DSlot Uplink Packet

5.5.3.2 PL_Pairing (W-Device)

This service is used to pair / unpair a W-Device from its W-Master via system management or by MasterCommand. The parameters of the service primitives are listed in Table 24.

Table 24 PL_Pairing (W-Device)

Parameter Name	.req	.ind	.cnf
Argument	M	M	
Method	M		
Info		M	
Result (+)			S
Result (-)			S
ErrorInfo			M

Argument
The service-specific parameters are transmitted in the argument.
Method
This parameter indicates the selected pairing mode.
Permitted values: PAIRING_BUTTON, UNPAIRING.
Info
Permitted values:
TIMEOUT (W-Device got no pairing request by W-Master within the time given in Timeout) see Table 186.
PERMANENT (W-Device has been paired permanently)
TEMPORARY (W-Device has been paired temporary (roaming))
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

5.5.3.3 PL_State (W-Device)

The PL_State service is used to indicate the pairing states of the physical layer after its startup or signals the state of a running or lost connection. The parameters of the service primitives are listed in Table 25.

Table 25 PL_State (W-Device)

Parameter Name	.ind
Argument	M
PLInfo	M

Argument

The service-specific parameters are transmitted in the argument.

PLInfo:

This parameter contains the status Information of the Physical Layer

Permitted values:

UNPAIRED W-Device is unpaired
PAIRED W-Device is paired
SYNCED W-Device is synchronized with its W-Master
COMLOST W-Device has no or lost connection to its W-Master

5.5.3.4 PL_Transfer (W-Device)

The PL-Transfer service is used to exchange the data between Data Link Layer and Physical Layer. The generation of the ACK-Bits for the W-Device is handled in PL (see clause 13.6). The parameters of the service primitives are listed in Table 26.

Table 26 PL_Transfer (W-Device)

Parameter Name	.req	.ind
Argument	C	M
Data	M	M
DataLength	M	M
Acknowledge		M
Result (+)		
Result (-)		
ErrorInfo		

Argument

The service-specific parameters of the service request are transmitted in the argument.

Data

This parameter contains the data which is transferred from / to the PL (air interface).

DataLength

This parameter contains the length of transmitted data, dependent on the direction and uplink type.

Ranges: PL_Transfer.ind: 0 to 37 octets (data from W-Master to W-Device)
PL_Transfer.req: 0 to 2 octets (data from W-Device to W-Master, SSlot-Format)
PL_Transfer.req: 0 to 15 octets (data from W-Device to W-Master, DSlot-Format)
PL_Transfer.req with DataLength = 0 causes the PL to send an IMA-Uplink.

Acknowledge

This parameter indicates, whether the last ULink has been confirmed by W-Master or not. PD handler, Event handler and OD handler needs the Acknowledge from PL to decide if a retransmit of data is needed or not.

Result (+):
This selection parameter indicates that the service request has been executed successfully.

Result (-):
This parameter contains supplementary information on the transfer status.

ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

5.5.3.5 PL_QualityService (W-Device)

The PL_QualityService is used to request the actual quality of the wireless connection from PL. The Service response with the link quality in percent of the W-Device (calculation see 5.4.6).
The parameters of the service are listed in Table 27

Table 27 PL_QualityService (W-Device)

Parameter Name	.req	.cnf
Argument	C	
<none>		
Result (+)		S
Data		M
Result (-)		S
ErrorInfo		M

Argument
This service has no parameter for PL.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Data
Parameter type: Octet
Permitted Value: 0 to 100%.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

5.5.3.6 PL_SetHopTable (W-Device)

The PL_SetHopTable service is used to set the new hopping table in W-Device PL.
The parameters of the service primitives are listed in Table 28.

Table 28 PL_ SetHopTable

Parameter Name	.req	.cnf
Argument	M	
UpdateType	M	
Index	M	
Data	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
Parameter Type: Record
UpdateType: This parameter contains the type of update, for the usage of UpdateType, Index and data See 18.4. Permitted values:
FULL_TABLE
DELETE_CELL
ADD_CELL
REPLACE_CELL
Index: This parameter contains the index of the changed cell in the hopping table.
Permitted values: 0, 1-78
Data: This parameter contains the value/s to replace/add.
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.3.7 PL_SetWakeUpTime (W-Device)

The PL_SetWakeUpTime service is used to deliver the WakeUpTime of low energy W-Device from the AHT handler to the PL. The parameters of the service primitives are listed in Table 29.

Table 29 PL_SetWakeUpTime

Parameter Name	.req	.cnf
Argument	M	
WakeUpTime	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
WakeUpTime: contains the WakeUpTime in W-Sub-cycles to set in PL.
Permitted values: 0 to 16777215 (3 Octets).
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)

PARAMETER_CONFLICT (consistency of parameter set violated)

5.5.3.8 PL_CmdTrig (W-Device)

The PL_CmdTrig service is used to communicate real time actions in PL triggered by Command handler. PL_CmdTrig deliveres the WakeUpTime value. The parameters of the service primitives are listed in Table 30.

Table 30 PL_CmdTrig

Parameter Name	.req	.cnf
Argument	M	
Command	M	
Result (+)		S
WakeUpTime		S
Result (-)		S
ErrorInfo		M

Argument:
The service-specific parameters are transmitted in the argument.
Command: contains the action to perform in PL. Permitted values:
WAKE_UP_TIME (triggers the PL to deliver the WakeUpTime to Cmd handler)
JUMP (switch to new hopping table HT02, starting with Hop-1 frequency)
Result (+):
This selection parameter indicates that the service has been executed successfully.
WakeUpTime: WakeUpTime value received from the W-Master in W-Sub-cycles.
Permitted values: 0 to 16777215 (3 Octets).
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information. Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

5.6 Physical Layer PL protocol

5.6.1 Usage of the Configuration Channel

The Configuration channel is available only when one track is configured to ServiceMode. Only in this mode, scan, pairing and roaming activities are possible. The following figures are based on the method where every 5th W-Sub-cycle is substituted with a configuration message on the configuration frequencies. All other W-Frames are transmitted on the regular frequency channels from the frequency hopping table.

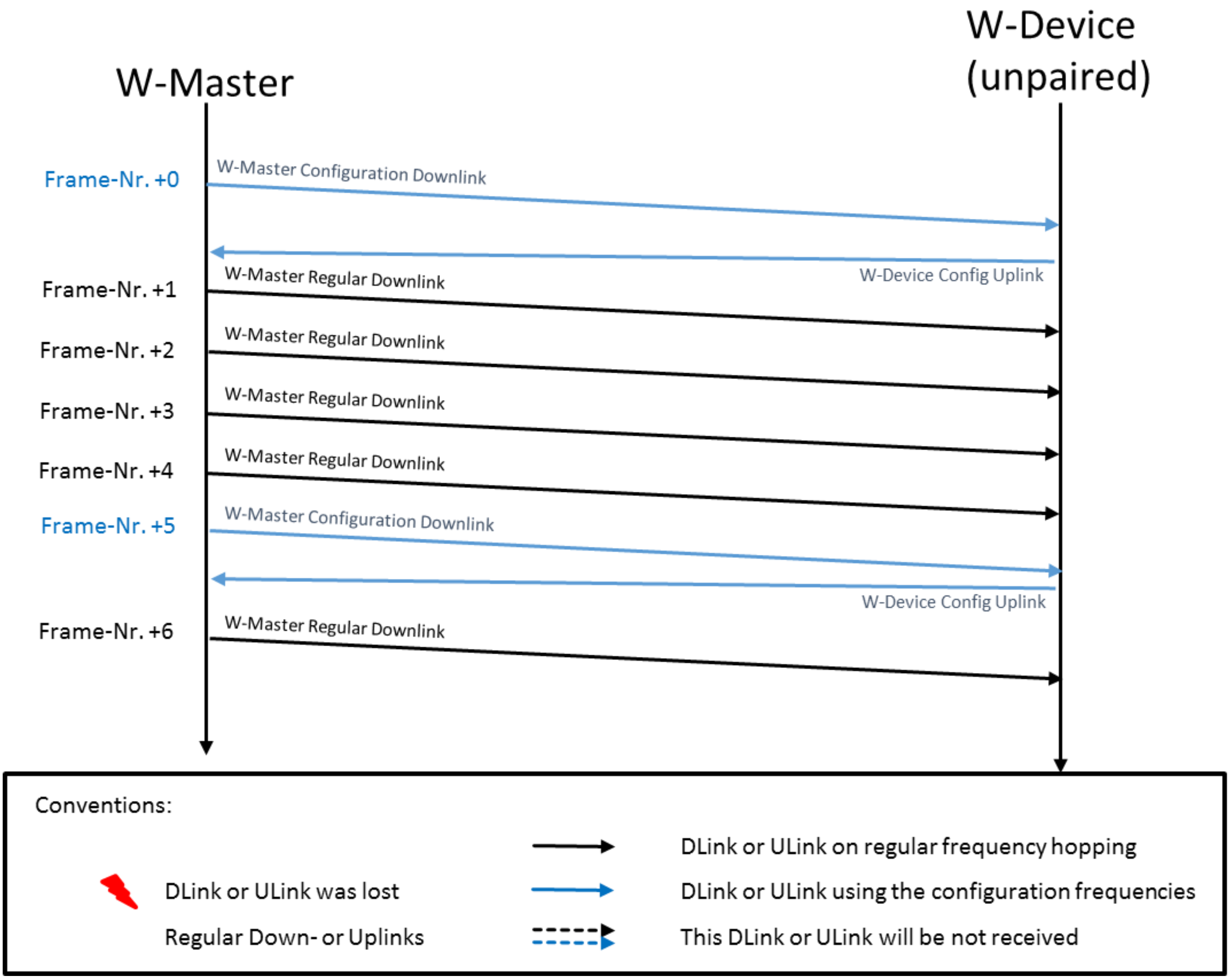


Figure 45 Usage of the Configuration Channels

5.6.1.1 Retry handling during ServiceMode (Scan, Pairing, Roaming)

Figure 46 describes the retry handling for all ServiceModes between W-Master and W-Device. If a DLink or a ULink was lost, the Data shall be retransmitted. The maximal number of all retransmissions within a service request is given by the timeout of the corresponding service (see Table 186).

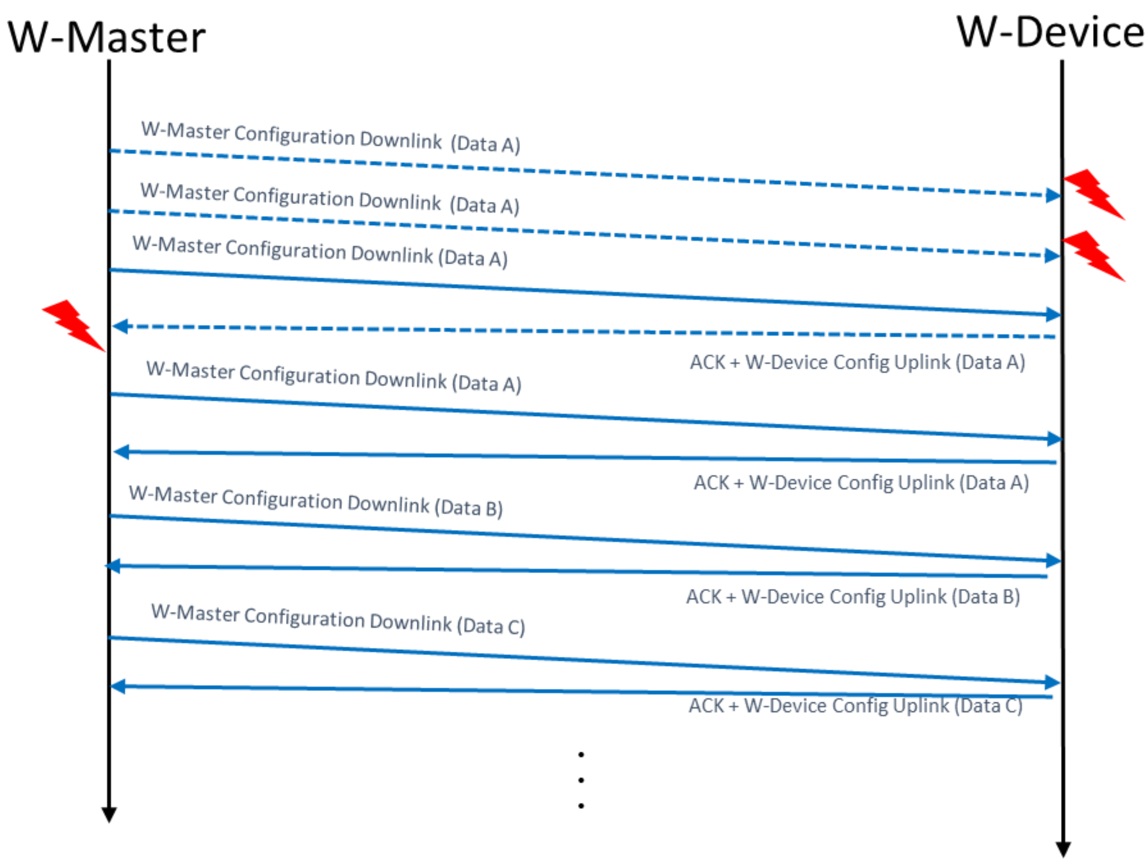


Figure 46 Retry handling during ServiceMode

5.6.1.2 Configuration sequence for Scan

Figure 47 describes the sequence for a discovery procedure. The W-Master sends a Scan Request Downlink (see Figure 134) on each configuration W-Sub-cycle with a continuously incremented request number (RequestN).
If an unpaired W-Device receives the Scan Request, it shall respond with a Scan Response Uplink (see Figure 148) after *X* W-Sub-cycles. The number of W-Sub-cycles to wait shall be calculated as described in Equation 4

$$X = \mathrm{RequestN} + \mathrm{FrameN}$$

Equation 4 Calculation of the number of W-Sub-cycles

where FrameN is the number of W-Sub-cycles between the first received Scan request and the following Scan Response. The Frame number is calculated with Equation 5

$$Frame_N = \left(\sum_{i=1}^{9} UniqueID(i)\right) mod(30)$$

Equation 5 Frame number calculation using a UniqueID of the W-Device

A W-Device shall, irrespective to its Slot Type, respond always as a DSlot in an even Slot. The slot number the W-Device shall use, has to be calculated according to Equation 6.

$$Slot_N = 2 \cdot \left(\sum_{i=1}^{9} UniqueID(i) \right) mod(4)$$

Equation 6 Slot number calculation using the UniqueID.

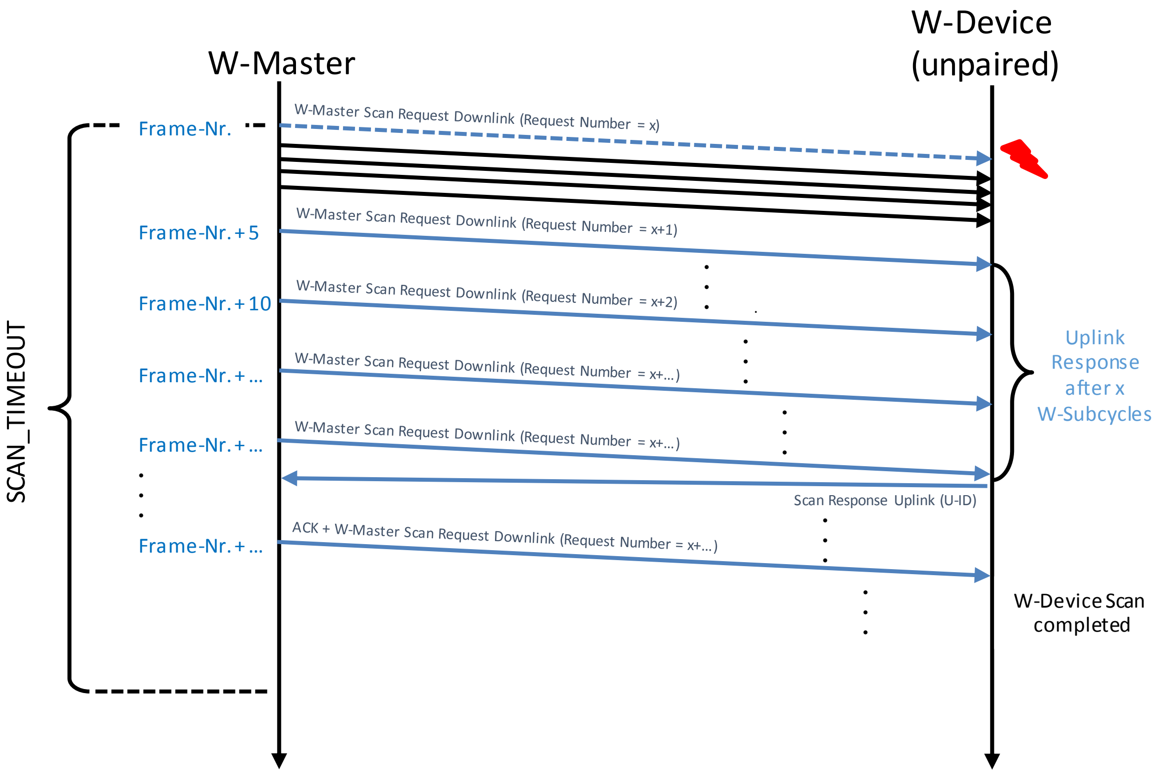


Figure 47 Configuration sequence for Scan

Note:
A W-Device shall not reply twice on Scan Requests of the same W-Master within the same SCAN_TIMEOUT interval.

5.6.1.3 Configuration sequence for pairing by UniqueID

Figure 48 describes the sequence for pairing by UniqueID. W-Master sends ConnectionParameter via

- Pairing Request Downlink (Roaming Flag = 0), see Figure 141
- Negotiation 1 Request Downlink, see Figure 142.
- Negotiation 2 Request Downlink, see Figure 143.

Unpaired W-Device receives the pairing request and if the requested UniqueID is identical to the W-Device UniqueID, shall reply with Response ULinks according to this sequence:

- Pairing Response Uplink, see Figure 149
- Negotiation Response Uplink 1 and 2, see Pairing Negotiation Uplink Packet Figure 150

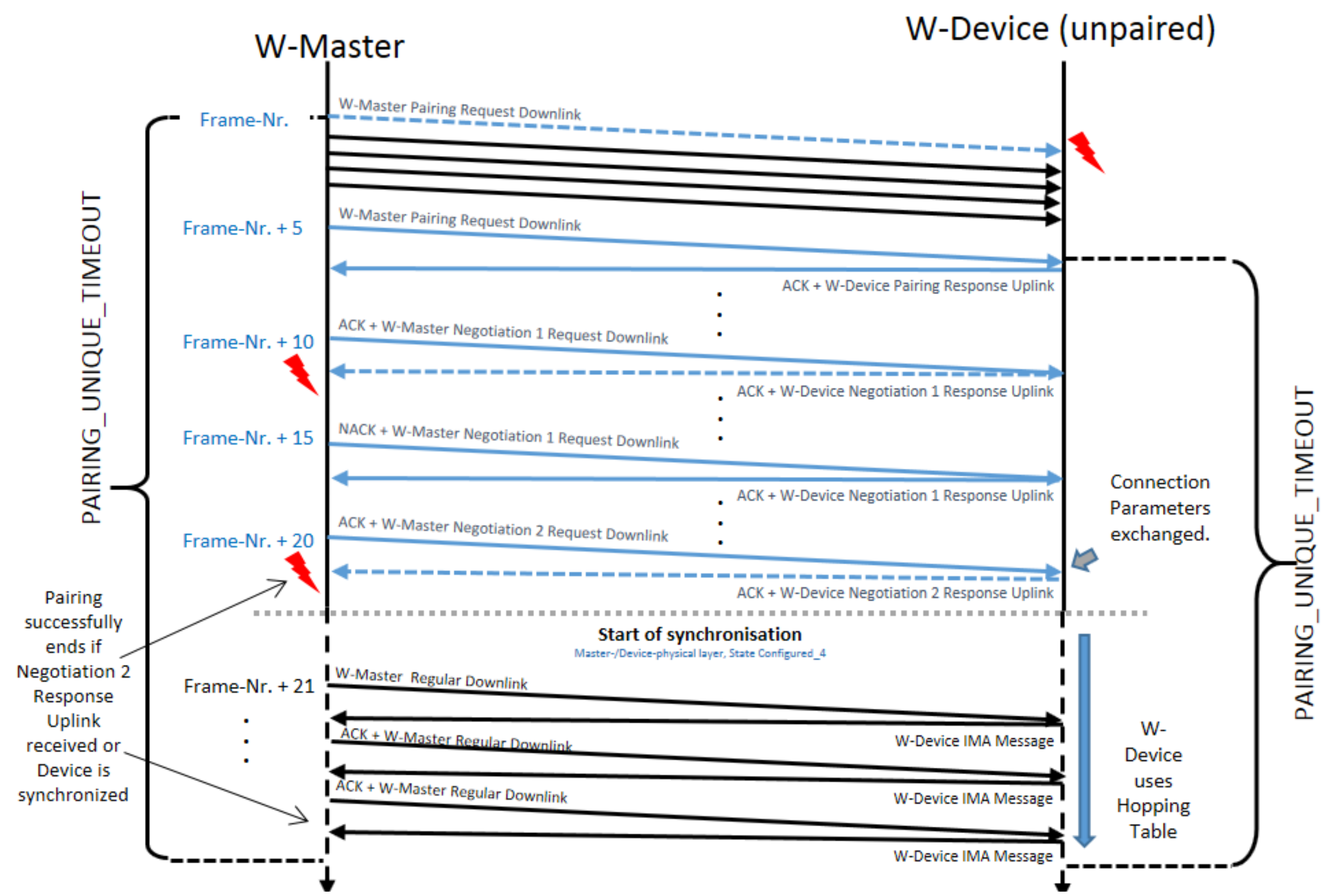


Figure 48 Configuration sequence for pairing by UniqueID

5.6.1.4 Configuration sequence for pairing by Button

Figure 49 describes the sequence for pairing by Button. If pairing by Button is active the W-Master sends ConnectionParameter via:

- Pairing Request Downlink, see Figure 141.
- Negotiation 1 Request Downlink, see Figure 142.
- Negotiation 2 Request Downlink, see Figure 143.

If the unpaired W-Device has been activated by the pairing button and it receives a pairing request, then the W-Device responds with Uplinks according to this sequence:

- Pairing Response Uplink, see Figure 149,
- Negotiation Response Uplink 1 and 2, see Pairing Negotiation Uplink Packet Figure 150

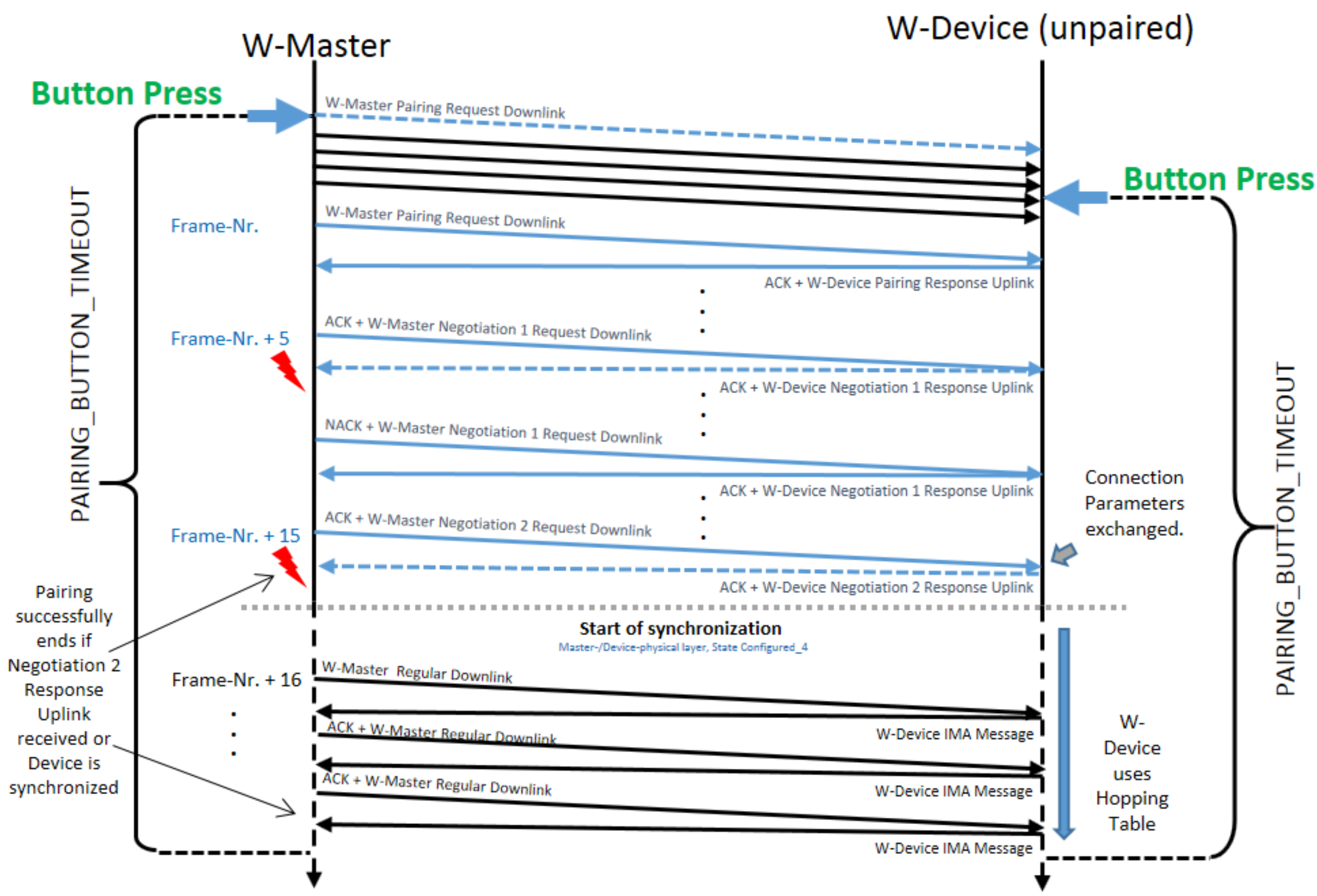


Figure 49 Configuration sequence for pairing by Button

5.6.1.5 Message Sequence Chart for Roaming

Figure 50 describes the “Handover Connect” sequence for a temporary connection in Roaming Mode. A W-Master track in Roaming Mode shall regularly scan for unpaired W-Devices (see 5.6.1.2 Configuration sequence for Scan)

If an unpaired W-Device shall be temporarily paired in Roaming Mode, the W-Master executes a Pairing by UniqueID (see 5.6.1.3., Configuration sequence for pairing by UniqueID) with Roaming Flag = 1 (Pairing Request Downlink, see Figure 141).

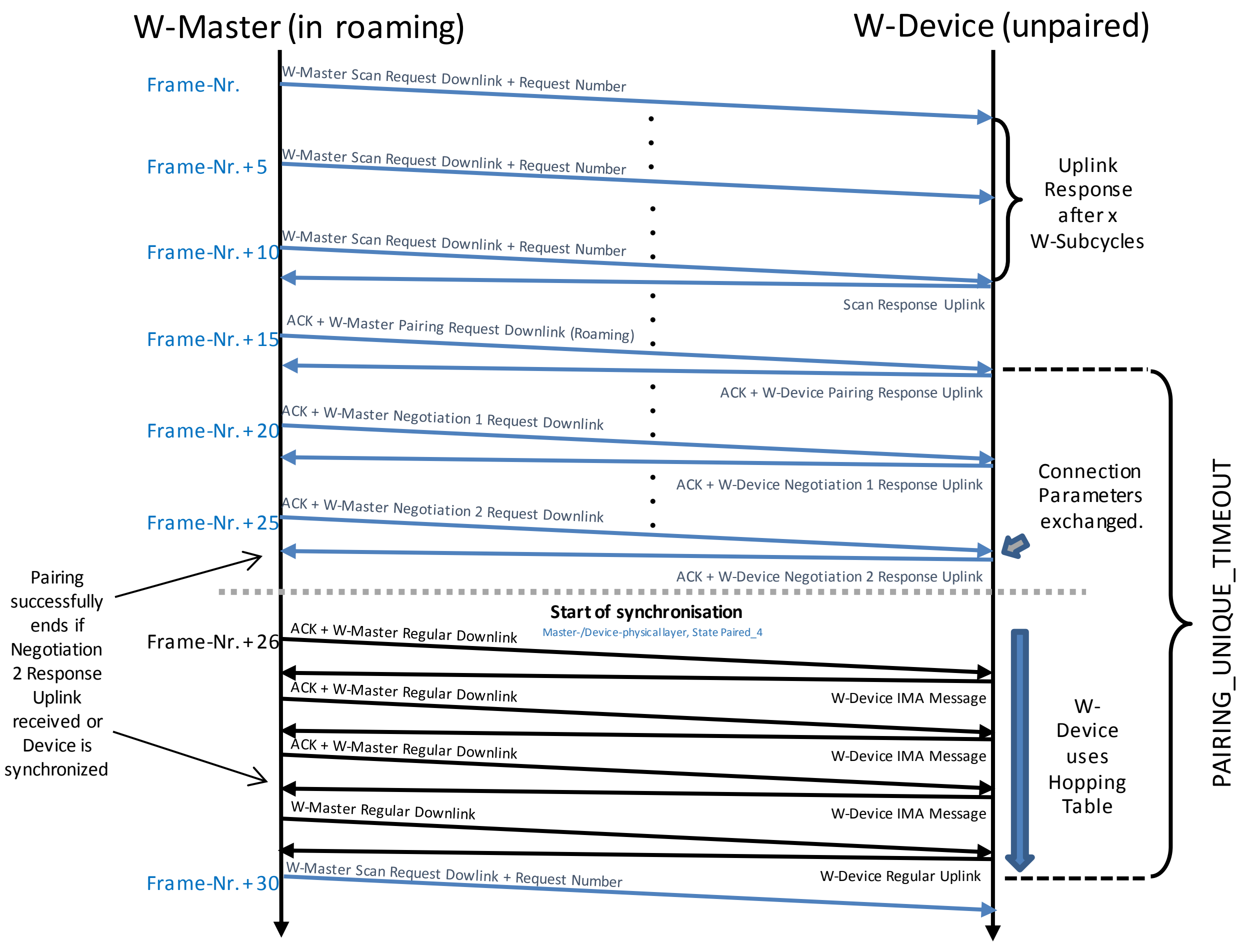


Figure 50 Message Sequence Chart for Roaming

5.6.2 PL W-Master state machine

Figure 51 shows the main state machine of the W-Master Physical Layer. This state machine describes initialization and starting of the Physical Layer. The sub state machines CyclicMode_2 and RoamingMode_3 (Figure 52) handles operation of the Physical Layer in Cyclic and Roaming modes. The only operational difference between the sub state machines is the usage of the scanning mode (Sub State machine Mode_Scan_12) in permanent (Roaming) and "on request" (Cyclic) manner.

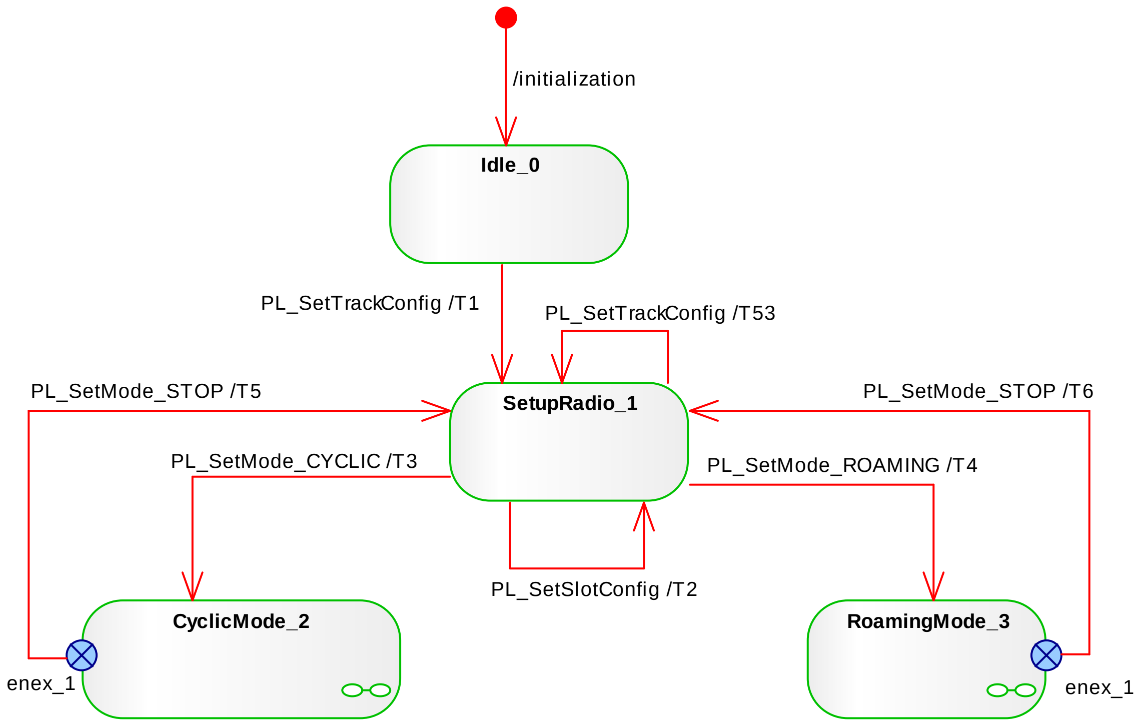
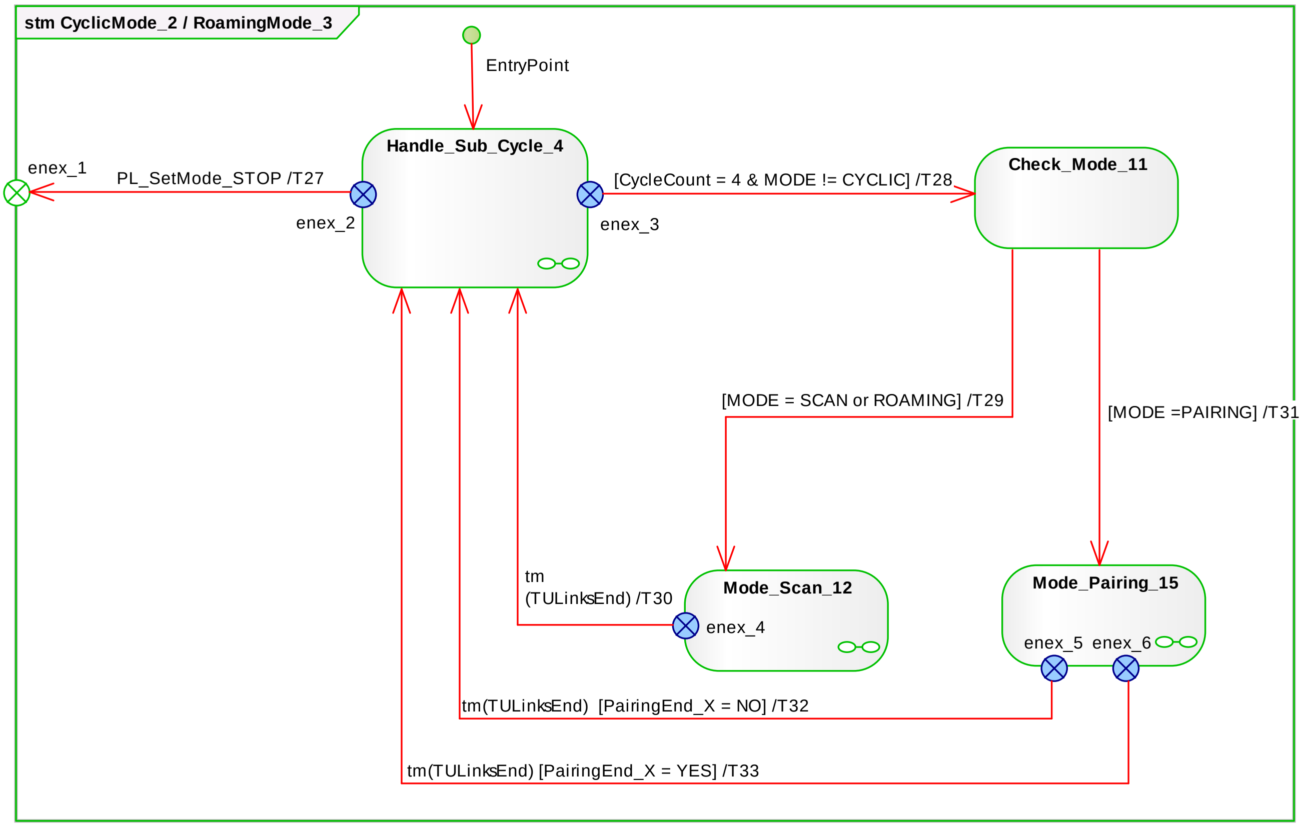


Figure 51 PL W-Master state machine

2250



2251 **Figure 52 Submachine of CyclicMode_2 or RoamingMode_3 of W-Master physical layer**

2252

5.6.2.1 Submachine of Handle_Sub_Cycle_4 of Master physical layer

The Handle_Sub_Cycle_4 sub state machine handles timing control within each sub cycle, transmission of the downlink, reception of the uplinks and handling of the IMA (I'm Alive) timeouts for all configured devices. This state machine triggers the Scan if the activation is requested by the PL_SetMode.req service or in case the W-Master Track is set in to roaming mode. It triggers also pairing if the Pairing.req service invoked.

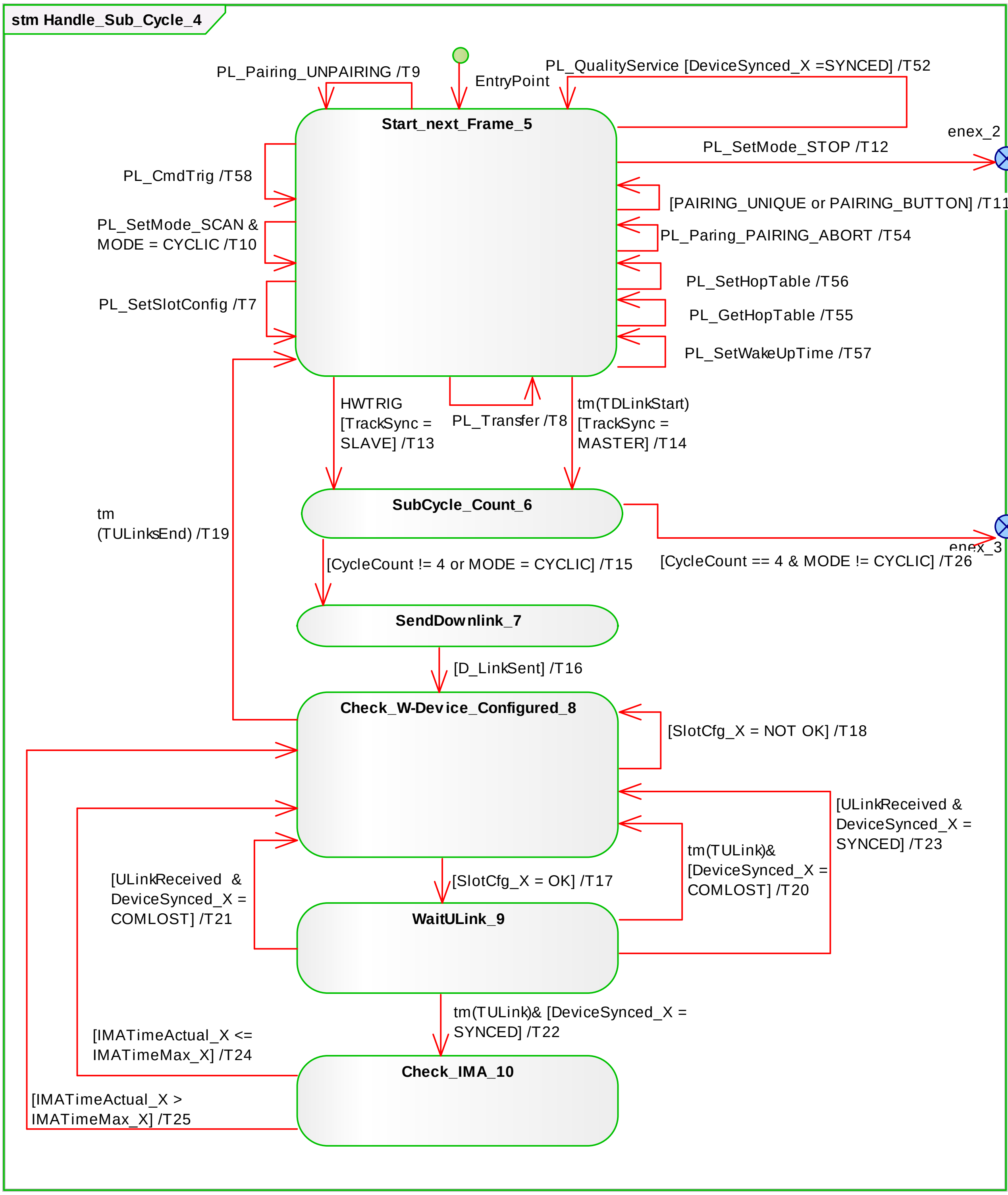


Figure 53 Submachine of Handle_Sub_Cycle_4 of W-Master physical layer

5.6.2.2 Submachine for Mode_Scan_12

The Mode_Scan_12 (Figure 54) sub state machine handles transmission of the scan request telegrams to all not configured devices and collects the scan responses.

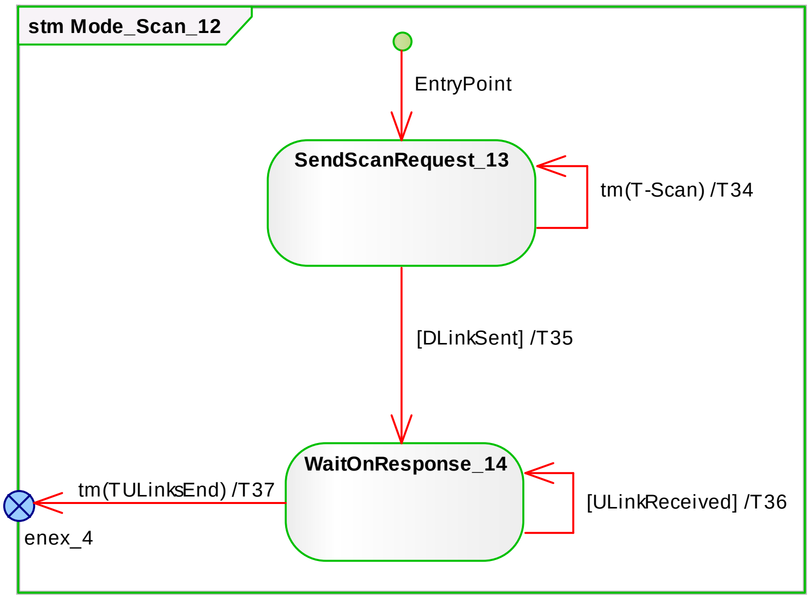


Figure 54 Submachine for Mode_Scan_12

5.6.2.3 Submachine for Mode_Pairing_15

The Mode_Pairing_15 (Figure 55) sub state machine handles the pairing procedure of the unpaired W-Devices. The pairing itself is divided into three steps, the pairing request, and two pairing negotiation steps.

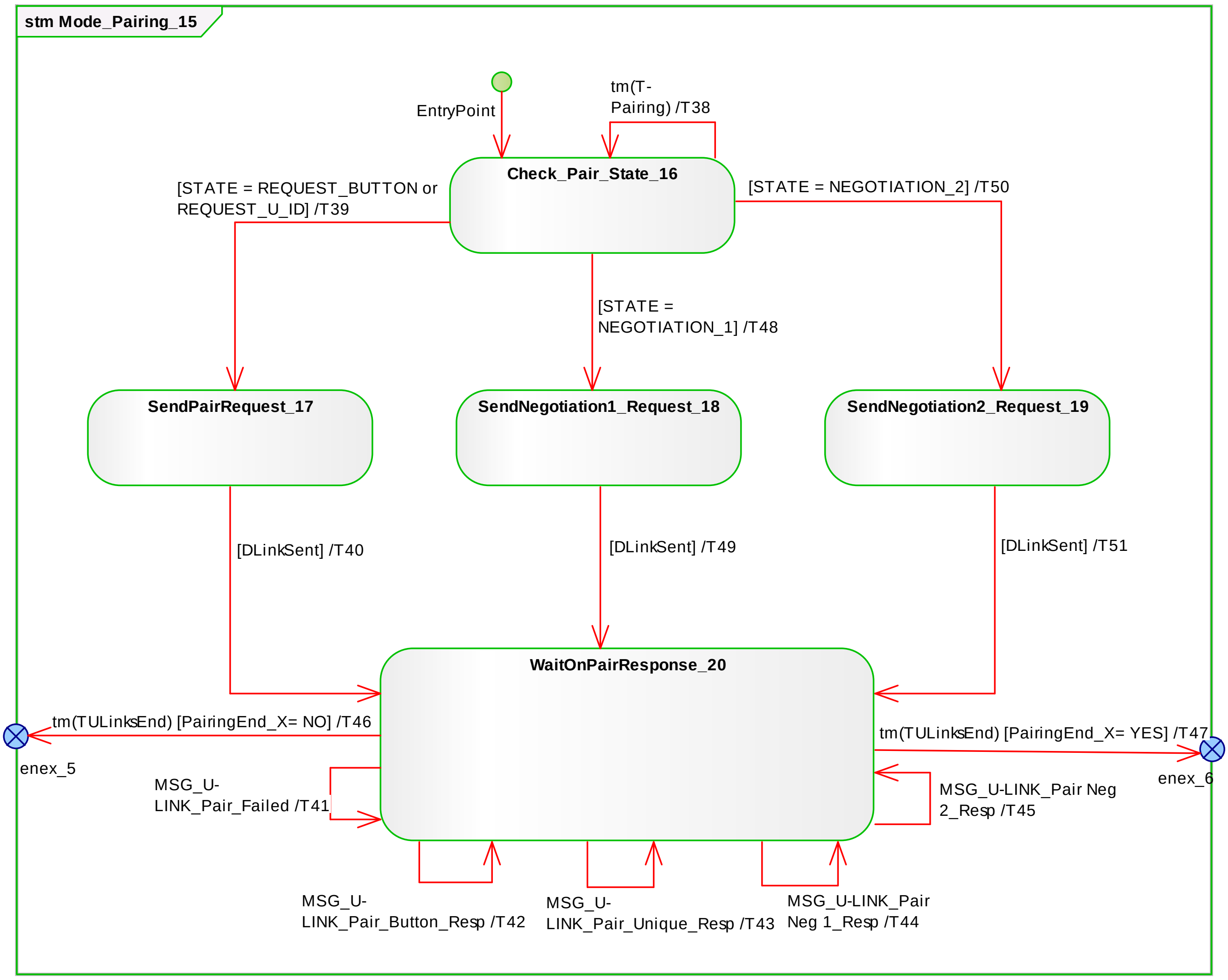


Figure 55 Submachine for Mode_Pairing_15

Table 31 State transition table of the W-Master physical layer

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting for activation by SM_SetTrackConfig via PL_SetTrackConfig service.
SetupRadio_1	Initialisation and setup of the radio transceiver (Track) for radio operation as specified in 5.1to 5.4 and 18.2 Annex G (HoppingTable) Set up the slot configuration for the slot given in Slot_N via service PL_SetSlotConfig: **UniqueID:** the UniqueID of the W-Device which will be connected to this SSlot or DSlot. **Slot_N:** points to the timing slot within the TDMA slot assignment (See Figure 39 Format of a W-Sub-cycle with DSlots and Figure 40 Slots and DSlots). **SlotType:** indicates the length of ULink (See Figure 144 “Regular SSlot Uplink Packet” and Figure 145 “Regular D-Slot Uplink packet”). **IMATime:** contains the I am alive time (see clause 14.3.5) to detect COMLOST.

STATE NAME	STATE DESCRIPTION
CyclicMode_2	Cyclic W-Frame exchange between W-Master and W-Devices in CyclicMode: The state performs the creation of the W-Frames, starting with the transmission of DLink and the handling of ULinks. After each W-Frame (all ULinks processed) this state changes the radio frequency to the next frequency specified by the frequency hopping table.
RoamingMode_3	Cyclic W-Frame exchange between W-Master and W-Devices in RoamingMode: Perform the creation of each W-Frame, starting with the transmission of DLink and the reception of ULinks. After each W-Frame (all ULinks processed) this state changes the radio frequency to the next frequency given by the frequency hopping table. At every 5th W-Sub-cycle, the frequency given by the hopping table is overwritten by one of the configuration frequencies in an alternating sequence. Note: Due to the Scan Request Downlinks every 5th W-Sub-cycle, the cyclic data channel availability at a cycle time of 5 ms might be affected. Therefore, it is recommended to use W-Devices with a W-Cycle of minimum 10 ms within a roaming track.
SM: Handle_SubCycle_4	This Submachine cyclically transmits the W-Frames (DLink payload and processing of all ULinks). It is used by State „CyclicMode_2“ and “RoamingMode_3“, dependent on PL_SetMode(Cyclic or Roaming), see T3 and T4. Furthermore, this state generates a trigger to handle every 5th Frame for the Modes SCAN, PAIRING and ROAMING, selected by service PL_SetMode in state „Start_next_Frame_5“.
SM: Start_next_Frame_5	This state loads the data from MH (reported via service PL_Transfer) to the payload data of the DLink (see Figure 139 Regular DLink). If MH has no data to send (PL_Transfer hasn‘t be called), set the payload data to 0 (DLink without data). Get the next frequency which shall be used for the following DLink from the frequency table.
SM: SubCycle_Count_6	This state is used to trigger every 5th W-Frame for the Modes SCAN, PAIRING and ROAMING.
SM: SendDownlink_7	Sending of the Regular Downlink over the air on the frequency selected in state “Start_next_Frame_5”.
SM:Check_W-Device_Configured_8	This state checks, if the actual Slot (W-Device) is configured.
SM: WaitULink_9	Waiting for the reception of the actual Uplink until the Uplink has been received or the Timer T_{TUlink} exceeds (See Figure 144 and Figure 145 Regular ULink). Note: The CRC 32 of a regular ULink is XOR-ed by W-Device with the last 4 octets of the W-Device’s U-ID (see 13.7 Final XOR of a regular ULink).
SM: Check_IMA_10	This state handles the IMA supervision for the actual Slot, since the W-Device is synchronized but Uplink has not been received.
SM: Check_Mode_11	This state is called every 5th W-Sub-cycle for the Modes SCAN, PAIRING and ROAMING to select the DLink which shall be sent on a configuration frequency.
SM: Mode_Scan_12	This submachine handles the DLink and ULinks for SCAN mode.
SM: SendScanRequest_13	This state handles the generation of the Scan Request Downlink (See 13.3.1 Scan Downlink)
SM: WaitOnResponse_14	This state handles the reception of the Scan response Uplinks (up to four Scan response uplinks are possible, see 13.5.2. Scan Response Uplink)
SM: Mode_Pairing_15	This substate machine handles the Pairing mode.
SM: Check_Pair_State_16	This state handles the generation of the next Pairing downlink depending on the Pairing „STATE“

STATE NAME	STATE DESCRIPTION
SM: SendPairRequest_17	This state handles the transmission of the Pairing Request Downlink (see 13.3.2. Pairing Request Downlink). If STATE = REQUEST_BUTTON, use „Pairing by Button“ DLink, see Figure 141. Pairing by Button If STATE = REQUEST_U-ID, use „Pairing by UniqueID“ DLink, see Figure 141.
SM: SendNegotiation1_Request_18	This state handles the transmission of the Pairing Negotiation 1 Downlink (see Figure 142. Negotiation 1 Downlink)
SM: SendNegotiation2_Request_19	This state handles the transmission of the Pairing Negotiation 2 Downlink (See Figure 143. Negotiation 2 Downlink)
SM: WaitOnPairResponse_20	This state handles the reception of the Pairing response Uplink.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T1	0	1	*Activation of PL by System Management via PL_SetTrackConfig*. Calculate the frequency hopping table dependent on the parameters MasterID, BlackList and Track_N (see 18.2.: Creation of frequency hopping table HT01 with care to blacklisting). Setting the internal variable TrackSync = MASTER or SLAVE (see 5.5.2.1 PL_SetTrackConfig) *Note:* *MASTER: Generates the synchronization hardware signal (HWTRIG) (output) for slave tracks for synchronization.* *SLAVE: The Track shall use the (input) synchronization hardware signal to send the DLinks, see T13 and T14.*
T2	1	1	*Activation by System Management through PL_SetSlotConfig(ParameterList).* PL_SetSlotConfig prepares the corresponding Slot "_X" given in Slotnumber (Slot_N) for a proper connection in the following way: **Slot_N:** Points to the receive time within the TDMA slot assignment (See Figure 39.: Format of a W-Sub-cycle with DSlots and Figure 40 Slots and DSlots) **UniqueID:** The last 4 octets of the UniqueID (Device Distinguishing ID) is used as final XOR of the CRC32 checksum (see 13.7.: Final XOR of a regular ULink) **SlotType:** Defines the length of the ULink (see Figure 37 Uplink- SSlot and Figure 38 Uplink- DSlot) to setup the transceiver receive length. **IMATime:** Defines the number of W-Sub-cycles to observe the presence of the W-Device (see clause 14.3.5 for encoding). Set SlotCfg_X = OK. *Note: PL_SetMode shall return PARAMETER_CONFLICT if the SlotType is DSlot and Slot_N not even.*
T3	1	2	*Activation by System Management through PL_SetMode(CYCLIC).* Set internal variable Mode = CYCLIC. Set CycleCount = 0. Set radio Tx power for the transceiver. Start Timer $T_{DLinkStart}$ with the value of M_SWITCH_TX_RX (208 µs), see Table 1
T4	1	3	*Activation by System Management through PL_SetMode(ROAMING).* Set internal variable Mode = ROAMING. Set CycleCount = 0. Set radio Tx power for the transceiver. Start Timer $T_{DLinkStart}$ with the value of M_SWITCH_TX_RX (208 µs), see Table 1.
T5	2	1	Stop the transmission of DLinks and reset the W-Track transceiver. Radio operation is deactivated after this command. Invoke PL_AHTStatus(STOP)
T6	3	1	See T5.
T7	5	5	See T2.
T8	5	5	Update the radio transmit buffer with payload for next DLink, delivered from MH via PL_Transfer.req. Note: If the PL_Transfer.req is not called from MH, set the payload to zero (dummy_DLink).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T9	5	5	*Unpairing is triggered by W-Port handler via Service PL_Pairing.req(UNPAIRING, Slot_N).* Set Bit in SlotCfg_X = NOT OK. This marks the Slot as unused. Set Bit in DeviceSynced_X = COMLOST. Invoke PL_State(DeviceSynced) to report the W-Device's states DL-mode handler.
T10	5	5	Set Mode = SCAN. This activates the handling of the DLink and ULinks every 5th Frame (see T26). Start timer $T_{T\text{-}Scan}$ with the value SCAN_TIMEOUT.
T11	5	5	Set Mode = PAIRING. This activates the handling of DLink and ULinks every 5th Frame (see T26). Set PairingEnd_X = NO. Set STATE = REQUEST_BUTTON or REQUEST_U-ID, dependent on the parameter PL_Pairing(Method). Start timer $T_{T\text{-}Pairing}$ with the value PAIRING_UNIQUE_TIMEOUT or PAIRING_BUTTON_TIMEOUT, dependent on the parameter given via PL_Pairing(Method, Timeout).
T12	5	0	See T5.
T13	5	6	*The HW-Track is configured as SyncSlave.* Start next W-Sub-cycle on rising edge of external trigger HWTRIG from the master Track. If WakeUpCountdown higher than 0, decrement by 1.
T14	5	6	*The HW-Track is configured as SyncMaster.* Start next W-Sub-cycle if Timer $T_{DLinkStart}$ exceeded. Set the hardware trigger HWTRIG (output) to HIGH. If WakeUpCountdown higher than 0, decrement by 1.
T15	6	7	Increment CycleCount. Update the ACK field in the radio output buffer with Device_ACK_Cyclic (See Figure 139 W-Frame encodings)
T16	7	8	*Transceiver has sent DLink.* Start Timer $T_{ULinksEnd}$ with the value of M_RX_Uplink. If the Timer exceeds, all ULinks have been processed and the W-Frame ends. Set Device_ACK_Cyclic= 0.
T17	8	9	*This Slot (_X) is configured, if the Unique-ID is <> 0. Set up the Radio to receive the Slot and detect a possible Slot timeout:* Switch the transceiver to RX to receive this configured ULink. For the slot timeout detection start timer T_{ULink} on dependence of the SlotType (see Table 1 Transceiver timings): SLOT: DxTX_ULink for Slot + D_GUARD (96 T_{BIT} + 8 T_{BIT}) DSLOT: DxTX_ULink_D for DSlot + D_GUARD (200 T_{BIT} + 8 T_{BIT}) Update CRC32 final XOR with Device Distinguishing ID for this Slot (_X), see 13.7 Final XOR of a regular ULink. *For additional information about timing see Figure 39.: SSlots and DSlots.*
T18	8	8	*This Slot_X is not configured*. Increment _X to check / setup next ULink. Note: A Slot is not configured, if it's unique ID = 0
T19	8	5	*WFrameComplete since timer $T_{ULinksEnd}$ exceeded.* Start Timer $T_{DLinkStart}$ with the value of M_SWITCH_TX_RX (208 µs), see Table 1 Transceiver timings. Invoke PL_Transfer.ind(WFrameComplete = YES). If TrackSync = MASTER set the hardware trigger HWTRIG (output) to LOW.
T20	9	8	Increment ULink Slot (_X)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T21	9	8	*First ULink of Slot_X (W-Device_X) received. Set W-Device as synchronized:* Set DeviceSynced_X = SYNCED. Set Device_ACK_Cyclic_X =1 Invoke PL_State.ind(DeviceSynced). Invoke PL_Transfer.ind(ULinkType = IMA, Slot_N = _X, Ack/Nack, WFrameComplete = NO). Set IMATimeActual_X = 0. Increment ULink Slot (_X) *To complete a pairing request in case of retransmits during pairing:* If PairingEnd_X = NO, set PairingEnd_X = YES and set Mode = CYCLIC or ROAMING (dependent on previous track mode)
T22	9	10	*No ULink has been received in the given time of timer T_{ULink}.* Invoke PL_Transfer.ind(ULinkType = NOUPLINK, Slot_N = _X, NACK, WFrameComplete = NO). Increment ULink Slot (_X)
T23	9	8	*ULink has been received.* Set Device_ACK_Cyclic_X =1, Set IMATimeActual_X = 0. Increment ULink Slot (_X). If the W-Device has sent data (see 13.4. Regular ULink Frame Annex B): Invoke PL_Transfer.ind(Data, DataLength, ULinkType = DATA, Slot_N = _X, Ack/Nack, WFrameComplete = NO) If the W-Device has sent a IMA-Frame (see Figure 146 and Figure 147. IMA-Uplink Frame Annex B): Invoke PL_Transfer.ind(ULinkType = IMA, Slot_N = _X, Ack/Nack, WFrameComplete = NO).
T24	10	8	Increment IMATimeActual_X for I am alive time observation.
T25	10	8	*IMATimeMax reached. A Latency error occurred.* Set DeviceSynced_X = COMLOST. Report all W-Device states through an invoke of PL_State(DeviceSynced).
T26	6	11	*5th W-Sub-cycle reached. Handle every 5th Frame for the modes Pairing, Scan and Roaming.* Set CycleCount = 0.
T27	4	0	See T5.
T28	4	11	See T26.
T29	11	12	Load the “Scan Request” (see 13.3.1. Scan Request) downlink into radio output buffer and start the radio transmission. Update the ACK field in radio the output buffer with the Device_ACK_Service (See Figure 140 Scan Request).
T30	12	4	See T19.
T31	11	15	-
T32	15	4	See T19.
T33	15	4	*WFrameComplete since timer $T_{ULinksEnd}$ exceeded.* Start Timer $T_{DLinkStart}$ with the value of M_SWITCH_TX_RX (208µs), see Table 1 Transceiver timings. Invoke PL_Transfer.ind(WFrameComplete = YES). If TrackSync = MASTER set the hardware trigger HWTRIG (output) to LOW. Set Mode = CYCLIC or ROAMING, depending on initial track mode, see T3 / T4.
T34	13	13	*Timer $T_{T\text{-}Scan}$ expired, leave scan mode after this W-Sub-cycle.* Set Mode = CYCLIC Invoke PL_ScanEnd.ind

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T35	13	14	*Transceiver has sent the DLink.* Start Timer $T_{ULinksEnd}$ with the value of M_RX_Uplink. *If the Timer exceeds, all ULinks have been processed and the W-Frame ends.* Set Device_ACK_Service_X= 0.
T36	14	14	*A Scan Request response uplink has been received.* Set Device_ACK_Service_X =1. Invoke PL_Scan.ind(SlotType, UniqueID, Protocol VersionRevisionID). See 5.5.2.3. PL_Scan (master).
T37	14	4	See T19.
T38	16	16	*Timer TPairing expired.* Invoke PL_Pairing.ind(PAIRING_TIMEOUT). Set PairingEnd_X = YES;
T39	16	17	Load the pairing request downlink in the radio output buffer, dependent on pairing mode and send downlink: If STATE = REQUEST_BUTTON, use pairing request downlink "Pairing Request by Button", see Figure 141. If STATE = REQUEST_U_ID, use pairing request downlink "Pairing Request by UniqueID" see Figure 141. Update the ACK field in the radio output buffer with Device_ACK_Service (See Figure 141 Pairing Request by Button or Pairing Request by UniqueID).
T40	17	20	See T35.
T41	20	20	*Radio received MSG_UPLINK_Pair_Failed (see Table 162 Uplink-MSG-Types):* Set Device_ACK_Service_X =1. Invoke PL_Pairing ind(PAIRING_WRONG_SLOTTYPE). Set PairingEnd_X = YES;
T42	20	20	*Radio received MSG_UPLINK_Pair_Button_Resp (see Table 162 Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_1.
T43	20	20	*Radio received MSG_UPLINK_Pair_Unique_Resp (see Table 162. Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_1.
T44	20	20	*Radio received MSG_UPLINK_Pair Neg 1_Resp (see Table 162 Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_2.
T45	20	20	*Radio received MSG_UPLINK_Pair Neg 2_Resp (see Table 162 Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Invoke PL_Pairing.ind(PAIRING_SUCCESS). Set PairingEnd_X = YES;
T46	20	4	See T19.
T47	20	4	See T33.
T48	16	18	Load the Negotiation_1 Downlink in the radio output buffer and send the Downlink (see 13.3.3 Pairing Negotiation Downlink). Update the ACK field in radio output buffer with Device_ACK_Service_X (See 13.3.3 Pairing Negotiation Downlink)
T49	18	20	See T35.
T50	16	19	Load the Negotiation_2 Downlink in the radio output buffer and send Downlink (see 13.3.3. Pairing Negotiation Downlink). Update the ACK field in radio output buffer with Device_ACK_Service_X (See 13.3.3 Pairing Negotiation Downlink)
T51	19	20	See T35.
T52	5	5	-

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T53	1	1	Calculate the frequency hopping table dependent on the parameters MasterID, BlackList and Track_N (see 18.2.: Creation of frequency hopping table HT01 with care to blacklisting). Setting the internal variable TrackSync = MASTER or SLAVE (see 5.5.2.1 PL_SetTrackConfig) Note: MASTER: Generates the synchronization hardware signal (HWTRIG) (output) for slave tracks for synchronization. SLAVE: The Track shall use the (input) synchronization hardware signal to send the DLinks, see T13 and T14.
T54	5	5	Stop Timer TPairing. Set PairingEnd_X = YES;
T55	5	5	-
T56	5	5	*Activation by System Management through PL_SetHopTable.* Load the new hopping table to HT02
T57	5	5	*Activation by System Management through PL_SetWakeUpTime(WakeUpTime).* Set internal variable WakeUpCountdown = WakeUpTime.
T58	5	5	*Activation by Data Link Command Handler through PL_CmdTrig.* **Command:** Case WAKE_UP_TIME: Invoke PL_WakeUpTime(WakeUpCountdown, Slot_N) Case W_DEVICE_AWAKE: Indicates low energy W-Device sent IMA at WakeUpTime, if all low energy W-Devices sent an IMA then return PL_CmdTrig.cnf(JUMP) Case W_DEVICE_NOT_AWAKE: Indicates low energy W-Device did not send IMA at WakeUpTime, return PL_CmdTrig.cnf(WAKE_UP_ABORT) and PL_AHTStatus(WAKE_UP_ABORT) Case JUMP: Switch to new hopping table HT02 starting with HOP-1 frequency, invoke PL_AHTStatus(JUMP_SUCCESS) Case JUMP_FAIL: Indicates a W-Device did not acknowledge JUMP command. Invoke PL_AHTStatus(JUMP_FAIL)

INTERNAL ITEMS	TYPE	DEFINITION
$T_{DLinkStart}$	Const Time	See Table 1, M_SWITCH_TX_RX
$T_{ULinksEnd}$	Const Time	See Table 1, M_RX_ULink
T_{ULink}	Time	Timer to switch radio to RX and to check if an ULink has been received within the given time. The timer shall be loaded dependent on the Slot-Type: SSLOT: DxTX_ULink for SSlot + D_GUARD (96 T_{BIT} + 8 T_{BIT}) DSLOT: DxTX_ULink_D for DSlot + D_GUARD (200 T_{BIT} + 8 T_{BIT})
$T_{T\text{-}Pairing}$	Time	Timer is used with the values PAIRING_BUTTON_TIMEOUT or PAIRING_UNIQUE_TIMEOUT, see T10.
$T_{T\text{-}Scan}$	Const Time	Timer is used with the value SCAN_TIMEOUT, see T10.
Mode	Variable	This variable is used to select the different DLinks. Permitted values: CYCLIC, ROAMING, SCAN or PAIRING.
CycleCount	Variable	W-Sub-cycle Counter.
TrackSync	Variable	Defines, whether the Track is running as W-Frame synchronization master or slave

INTERNAL ITEMS	TYPE	DEFINITION
		Permitted values: MASTER or SLAVE (see 5.5.2.1 PL_SetTrackConfig).
ULinkReceived	Bool	Flag which shall be set by the radio hardware if an Uplink was received.
DLinkSent	Bool	Flag which shall be set by the radio hardware if the downlink has been sent.
PairingEnd_X	Bool	Flag which indicates if the pairing is completed. Permitted values: YES, NO.
SlotCfg_X	Bool	Flag which indicates if the corresponding slot is configured. Permitted values: YES, NO.
DeviceSynced_X	Bool	Flag which indicates if the W-Device for the corresponding slot is available / synchronized. Permitted values: SYNCED, COMLOST. See 5.5.2.7 PL_State.
IMATimeActual_X	Variable	Variable to count the number of W-Sub-cycles, if a Device is synchronized but no ULink has been received, see T24.
IMATimeMax_X	Variable	This Variable keeps the value IMATime in W-Sub-cycles (see clause 14.3.5 for encoding), delivered via the service PL_SetSlotConfig, see 5.5.2.5
Device_ACK_Cyclic_X	Variable	This Variable keeps the bit coded acknowledgement for received ULink in Cyclic Mode
Device_ACK_Service_X	Variable	This Variable keeps the bit coded acknowledgement for received ULink in ServiceMode
STATE	Variable	Variable to keep the states during pairing procedure, see Figure 55. Submachine for Mode_Pairing_15 Permitted Values: REQUEST_BUTTON, REQUEST_U_ID, NEGOTIATION1, NEGOTIATION2.
PAIRING_BUTTON_ TIMEOUT	Parameter	This parameter is delivered via service PL_Pairing. See 5.5.2.6. PL_Pairing-Service and 5.6.1.1. Retry handling during Pairing Mode.
PAIRING_UNIQUE_ TIMEOUT	Constant	Fixed to 3 s, see 5.6.1.1Retry handling during Pairing Mode.
SCAN_TIMEOUT	Constant	Fixed to 5 s, see 5.6.1.1Retry handling during Scan Mode.
WakeUpCountdown	Variable	This variable counts down to 0, starting value delivered via the service PL_SetWakeUpTime. Current value sent to AHT via PL_WakeUpTime

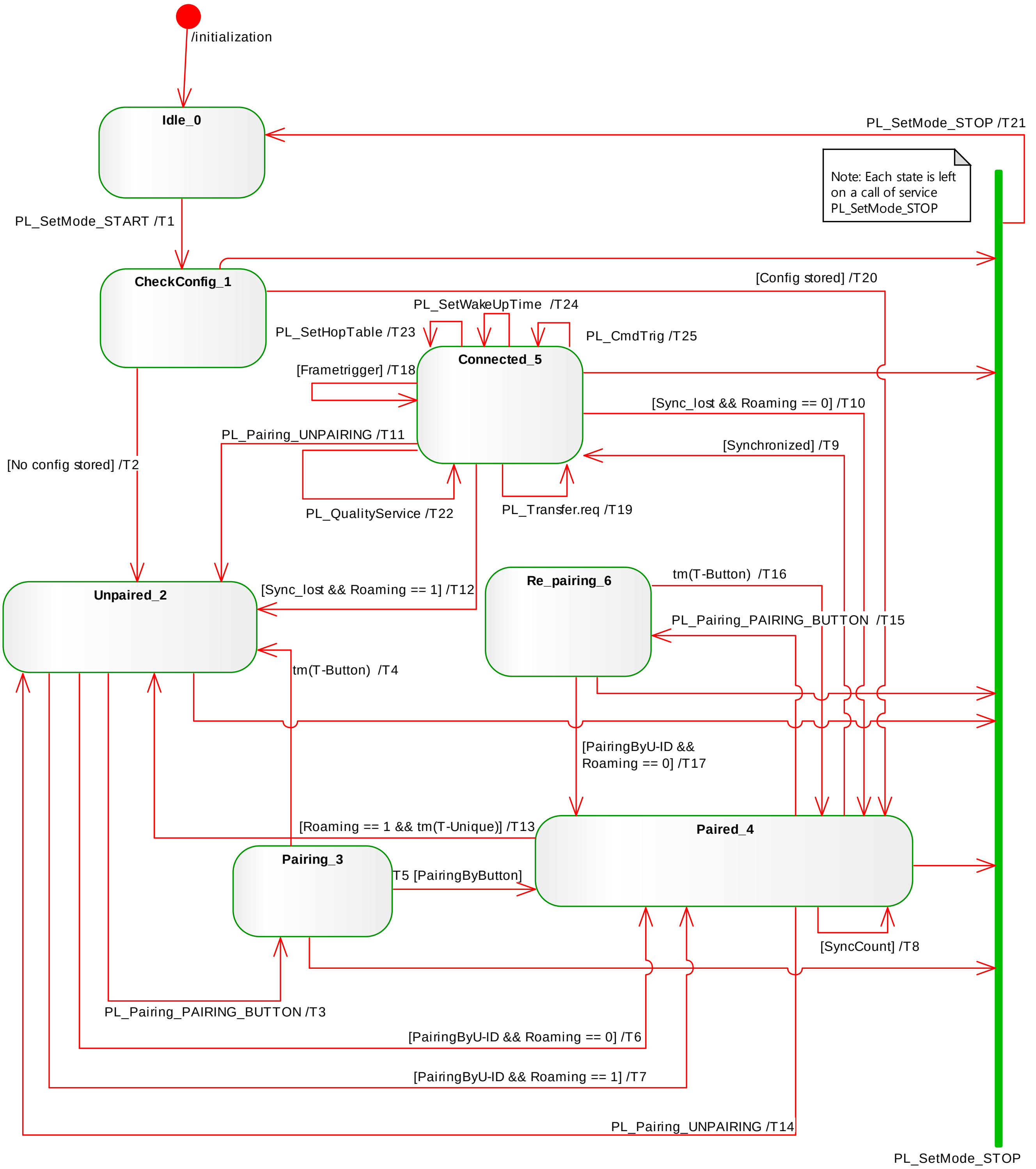
2277

2278 Note: X marks the variables which are individual in each Slot_N. The range of _X is 0 to 7 SlotNumbers

5.6.3 PL W-Device state machine

Figure 56 shows the main state machine of the W-Device Physical Layer. The state machine handles the initial initialization of the W-Device, Pairing, Unpairing and Cyclic operation. Depending on the pairing request, the W-Device is either configured as a Roaming or as a Cyclic W-Device. The main difference between two modes is the reaction on a communication Sync_lost event. In case the W-Device is configured for roaming, it goes in to the Unpaired_2 state immediately. Otherwise, it goes to the Paired_4 state and waits until W-Master comes back in range or pairing by button is activated by application layer.

2288



2289 **Figure 56 PL W-Device state machine**

2290

Table 32 State transition tables of the W-Device physical layer (normal device)

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting for activation via System Management through Service PL_SetMode.
CheckConfig_1	Check for availability of ConnectionParameter in non-volatile memory (see Table 140 ConnectionParameter).
Unpaired_2	Waiting for a Scan Request or a Pairing Request by UniqueID (via W-Master) or a button-press on the W-Device. The W-Device shall listen on configuration-channels (see 5.4.4) for receiving configuration downlinks via UniqueID (call by U-ID). If a W-Master Pairing Request Downlink(MSG_DLink_PAIR_UNIQUE) is received, start the timer ($T_{T\text{-}Unique}$).
Pairing_3	Waiting for Pairing Request by button from W-Master (MSG_DLink_PAIR_BUTTON). W-Device shall listen on the configuration channels (see Figure 141) to receive a configuration downlinks (call by button)
Paired_4	The W-Device has a valid ConnectionParameter setting. It shall wait on the frequency at Col_N for synchronization. When resynchronization at Col_N is not successful within an appropriate time, Col_N shall subsequently be incremented for the next synchronisation cycle. See also 17.1.5 W-Master not reachable.
Connected_5	The W-Device is connected to its paired W-Master via regular W-communication cycles (see Figure 139)
Re_pairing_6	Waiting for configuration-channels for Scan Request or Pairing Request by UniqueID (via W-Master).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by System Management through PL_SetMode.req(Start). (see Table 120 Transition T1).*
T2	1	2	The Device Radio has no valid ConnectionParameter settings stored (see Table 140) Invoke PL_State.ind(UNPAIRED)
T3	2	3	*The W-Device's pairing (by button) state is entered via service PL_Pairing.req(PAIRING_BUTTON) through SM_SetDeviceMode(PAIRING_BUTTON).* Start timer ($T_{T\text{-}Button}$).
T4	3	2	Invoke PL_Pairing.ind(TIMEOUT) (see Table 186) if timer ($T_{T\text{-}Button}$) expired.
T5	3	4	*Pairing by Button sequence was successfully executed.* *(See Figure 49 Configuration sequence for pairing by Button)* *Valid ConnectionParameter were successfully received.* *Store ConnectionParameter in non-volatile memory.* Invoke PL_Pairing.ind(PERMANENT) to report a successful pairing. Set SyncCounter to 0. Set received DataSyncword (see clause 5.2.4). Stop timer ($T_{T\text{-}Button}$).
T6	2	4	*Pairing by UniqueID sequence was successfully executed.* *(See Figure 48 Configuration sequence for pairing by UniqueID* *Valid ConnectionParameter were successfully received.* Store ConnectionParameter in non-volatile memory. Invoke PL_Pairing.ind(PERMANENT) to report a successful pairing. Set SyncCounter to 0. Set received DataSyncword (see Table 161).
T7	2	4	*Pairing by UniqueID sequence was successfully executed.* *(See Figure 50 Message Sequence Chart for Roaming / temporary connection)* *Valid ConnectionParameter were successfully received.* Store ConnectionParameter in volatile memory only. Invoke PL_Pairing.ind(TEMPORARY) to report a successful pairing. Set SyncCounter to 0 Set received DataSyncword (see Table 161).
T8	4	4	Send IMA ULink to the W-Master on each received DLink (e.g. see Figure 48 Configuration sequence for pairing by UniqueID / Start of synchronization) If the DLink has been received successfully, increment SyncCounter (SyncCounter = SyncCounter+1) Otherwise set the SyncCounter to 0.
T9	4	5	*The connection is synchronized, if SyncCounter >= Sync.* Set SyncLostCounter to 0. Invoke PL_State.ind(SYNCED) service indication to report that the connection is established. Stop timer ($T_{T\text{-}Unique}$). Stop timer ($T_{T\text{-}Button}$).
T10	5	4	The synchronization is lost, if SyncLostCounter > 5 * MaxRetry. Invoke PL_State.ind(COMLOST) service indication to report that the connection has been lost. Set SyncCounter to 0.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T11	5	2	*Unpairing was triggered by MasterCommand via Service PL_Pairing.req(UNPAIRING).* Invoke PL_State.ind(COMLOST) service indication to report that the W-Device is not connected. Delete non-volatile ConnectionParameter settings (seeTable 140). Set SyncCounter to 0. Set Roaming to 0.
T12	5	2	The synchronization is lost, if SyncLostCounter > 5 * MaxRetry Invoke PL_State.ind(COMLOST) service indication to report that the W-Device is not connected. Delete volatile ConnectionParameter settings (see Table 140) Set SyncCounter to 0. Set Roaming to 0.
T13	4	2	Delete non-volatile ConnectionParameter settings (see Table 140). Set SyncCounter to 0. Set Roaming to 0.
T14	4	2	See T11.
T15	4	6	*The W-Device's re_pairing state is entered via Service PL_Pairing.req(PAIRING_BUTTON) by SM_SetDeviceMode (PAIRING).* Start timer ($T_{T\text{-}Button}$).
T16	6	4	See T4.
T17	6	4	See T6, Stop timer ($T_{T\text{-}Button}$).
T18	5	5	Invoke PL_Transfer_ind to report the Frametrigger (W-Frame-Sub-cycle) to message-handler in following cases: Case 1: *DLink received, data for this W-Device are available (DataLength>0); see clause 5.6.3.1).* Set SyncLostCounter to 0. Case 2: *DLink received without data for this W-Device (DataLength=0, see clause 5.6.3.1).* Set SyncLostCounter to 0. Case 3: *No DLink received while W-Device is still synchronized / connected. Acknowledge=0. DataLength=0.* *(The Frametrigger shall be generated by timer with a time of W-Frame-Sub-cycle).* Increment SyncLostCounter.
T19	5	5	Update the radio transmit buffer with payload for the next ULink, delivered from MH via PL_Transfer.req. *Note: If the PL_Transfer.req is not called from MH, set the payload to zero (dummy_ULink, see clause 5.6.3.1).*
T20	1	4	*The Radio has stored a valid ConnectionParameter settings (see* Table 140*).* Invoke PL_State.ind(PAIRED)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T21	Any	0	*Any state shall be left through a call of PL_SetMode(Stop) Service via System Management.*
T22	5	5	Invoke PL_QualityService.cnf
T23	5	5	*Activation by System Management through PL_SetHopTable.* Load the new hopping table to HT02
T24	5	5	*Activation by System Management through PL_SetWakeUpTime.* Return PL_CmdTrig.cnf(WakeUpTime)
T25	5	5	*Activation by Data Link Command Handler through PL_CmdTrig.* **Command:** Case WAKE_UP_TIME: No action *Cmd handler is waiting for WakeUpTime via PL_CmdTrig.cnf(WakeUpTime).* Case JUMP: Switch to new hopping table HT02 starting with HOP-1 frequency

INTERNAL ITEMS	TYPE	DEFINITION
SyncCounter	Variable	Counter for received downlink frames (see T8).
Sync	Constant	Sync = 3.
SyncLostCounter	Variable	Counter for lost downlink frames (see T9 and T16).
Roaming	Variable	This volatile Flag indicates, whether the W-Device is paired permanently or temporary (see T6, T7, T10, T12). Variable shall be initialized to 0 during initialization and is transmitted during pairing procedure.
MaxRetry	Variable	Value to generate Sync_Lost. This Variable is transmitted via service DL_SetParam.
WakeUpTime	variable	Value for low energy W-Device, sent to Cmd handler via PL_CmdTrig.cnf()
$T_{T\text{-}Unique}$	Time	See Table 186, definition of PAIRING_UNIQUE_TIMEOUT
$T_{T\text{-}Button}$	Time	See Table 186, definition of PAIRING_BUTTON_TIMEOUT

5.6.3.1 DLink processing in the W-Device via PL

On the reception of each DLink, the PL shall parse the DLink payload to collect all W-Messages directed to this addressed W-Device or any broadcast Master-Command. The collected W-Messages (ControlOctets included) shall be sent to the DL-A message handler via Transfer.ind. If there are one or more W-Messages for the current Device received, the PL shall generate an acknowledgement, which shall be automatically transmitted in the following uplink. If Transfer.req was not invoked, then the payload shall be set to zero (dummy Uplink).

6 Data Link Layer (DL-A)

The data link layers are concerned with the delivery of messages between a W-Master and a W-Device. A set of DL-services is available to the application layer (AL) for the exchange of Process Data (PD) and Event or ISDU data. Another set of DL-services is available to system management (SM) for the retrieval of Device identification parameters and the setting of state machines within the DL. The DL uses PL-Services for controlling the physical layer (PL). The DL takes care of the error detection of messages (whether internal or reported from the PL) and the appropriate remedial measures (e.g. retry).

The data link layers are structured due to the nature of the data categories into Process Data handlers and Event / ISDU handlers which are in turn using a Message handler to deal with the requested transmission of messages. Each handler comprises its own state machine.

The data link layer is subdivided in a DL-A section with its own internal services and a DL-B section with the external services.

The DL uses additional internal administrative calls between the handlers which are defined in the "internal items" section of the associated state-transition tables.

6.1 General (W-Master)

Figure 57 shows an overview of the structure and the services of the W-Master's data link layer.

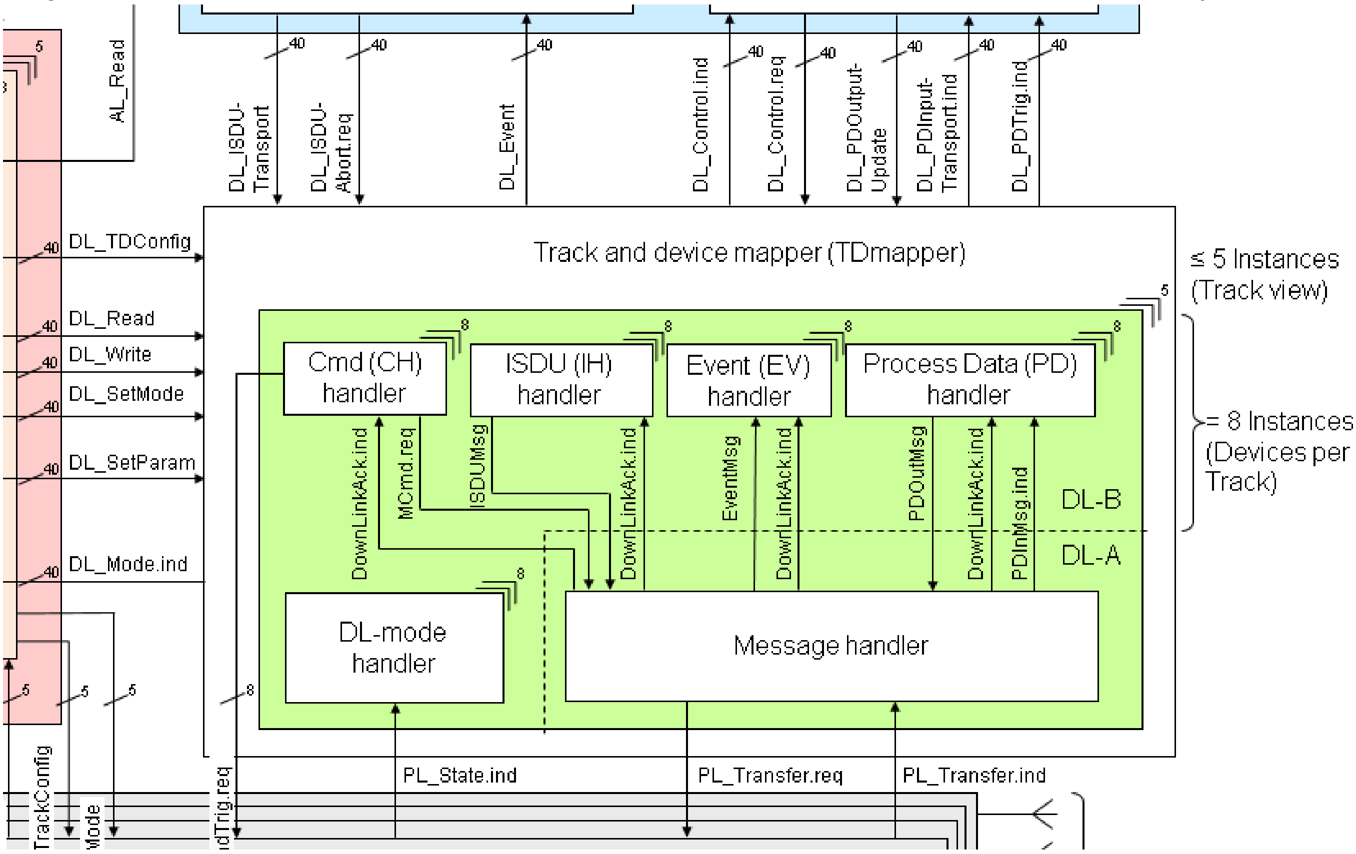


Figure 57 Structure and services of the data link layer (W-Master)

6.1.1 Track and W-Device-Mapper (TDMapper)

The track and W-Device mapper is used to assign a W-Port to a specific Track number (Track_N) and Slot number (Slot_N). Each Slot_N represents a W-Device, whereupon the W-Device communicates via this Slot to the W-Master. The Slot_N is transmitted during pairing to the selected W-Device. The configuration of the TDMapper is done by SM_SetPortConfig service (via DL_TDConfig). This mapping table enables a flexible assignment of W-Devices without changing of the W-Port, e.g. distribution of W-Devices within the tracks.

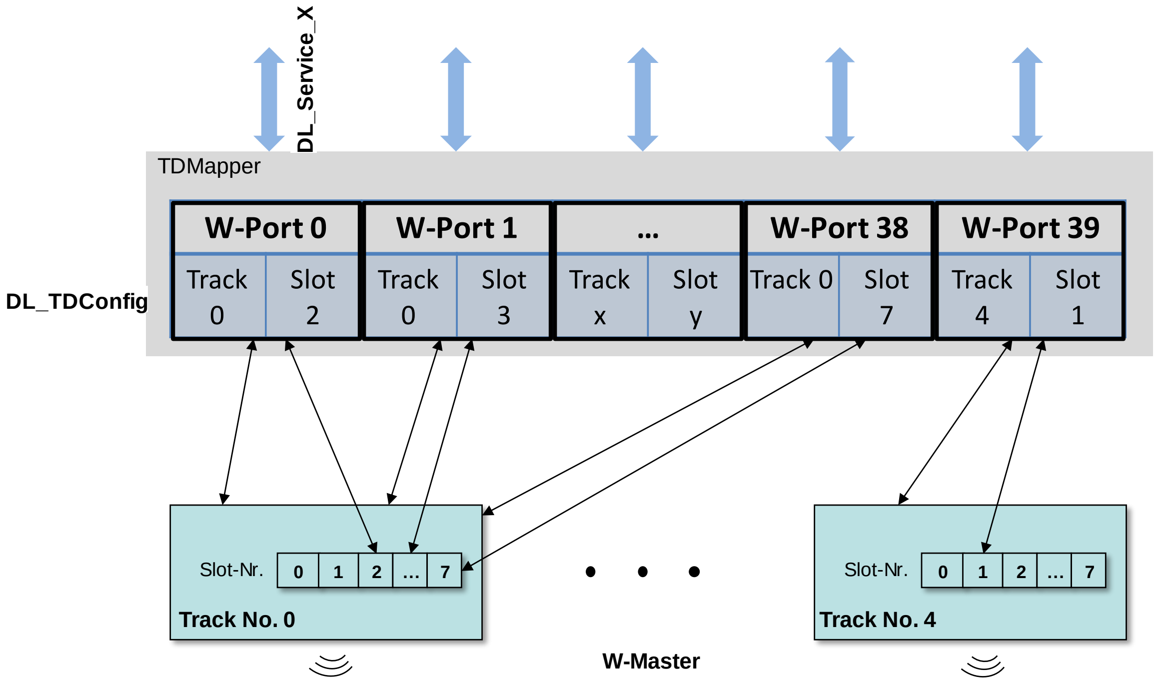


Figure 58 Track and W-Device-Mapper (TDMapper)

6.2 General (W-Device)

Figure 59 shows an overview of the structure and the services of the W-Device's data link layer.

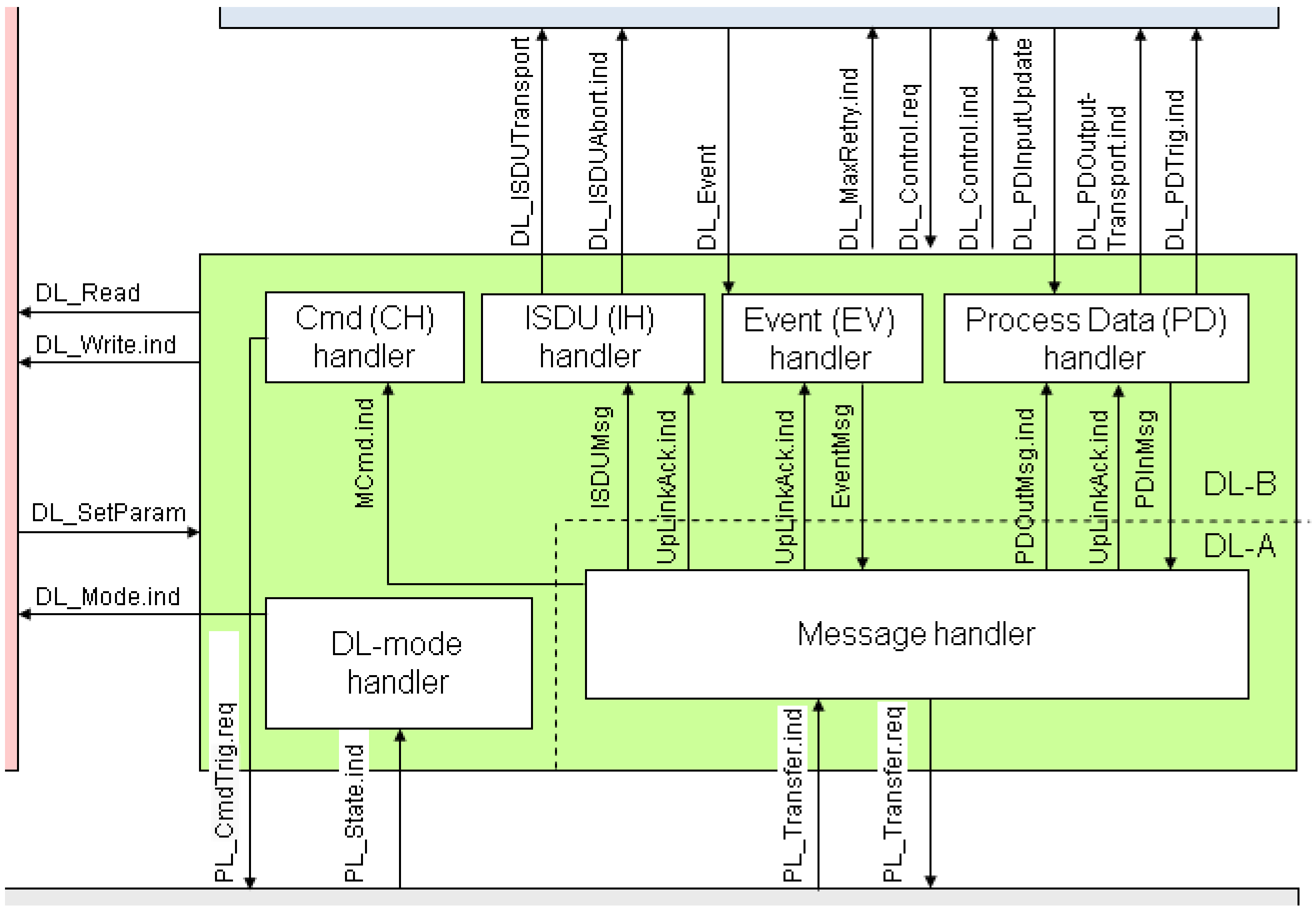


Figure 59 Structure and services of the data link layer (W-Device)

6.3 DL-A services

6.3.1 Overview

Table 33 lists the assignment of W-Master and W-Device to their roles as initiator (I) or receiver (R) in the context of the execution of their individual DL-A services.

Table 33 DL-A services within W-Master and W-Device

Service name	W-Master	W-Device
MCmd	R	I
ISDUMsg	R	I
DownLinkAck	I	
UplinkAck		I
EventMsg	I	R
PDOutMsg	R	I
PDInMsg	I	R
Key (see 3.3.5) *I Initiator of service* *R Receiver (responder) of service*		

6.3.2 MCmd (W-Master and W-Device)

The MCmd service provides the MasterCommand to change e.g. the W-Device STARTUP, PREOPERATE and OPERATE states. The parameters of the service primitives are listed in Table 34.

Table 34 MCmd

Parameter Name	.req	.ind
Argument	M	M
SendWMessage	M	
Slot_N	M	
Broadcast	M	
MasterCommand	C	M
Length	C	
DLType	C	

For further abbreviations and definitions of service parameters see clause 3.3.2 in REF 1.

Argument
The service-specific parameters are transmitted in the argument.

SendWMessage
This parameter signals, if a W-Message (and possible data) shall be added to the Downlink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)
NO (No W-Message needs to be sent)

Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

Broadcast
This parameter signals, if a MasterCommand shall be received by all connected W-Devices in the Track, use is restricted according to Table 165. Permitted values:

YES
NO
MasterCommand
This parameter contains the MasterCommand, see 14.1.1 and Table 165.
Length
This parameter contains the length of data to transmit. If no MasterCommand shall be sent, set Length to 0. Permitted values: 0 or 1.
DLType
This parameter informs the Message handler whether the MasterCommand is transmitted in PreDownLink (for low energy W-Devices) or FullDownLink.
Permitted values:
PreDLink (MasterCommand shall be transmitted in the PreDownLink).
FullDLink (MasterCommand shall be transmitted in the FullDownLink).

6.3.3 ISDUMsg (W-Master and W-Device)

The ISDUMsg service is used to set up the ISDU-request Data for the next message to be sent. In turn, the confirmation of the service contains the data from the receiver. The parameters of the service primitives are listed in Table 35

Table 35 ISDUMsg

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
SendWMessage	M			
Slot_N	M			
Data	C	C		
Length	C	M		
FlowCtrl	C	M		
Result (+)			S	S
SendWMessage			M	
Slot_N				M
Data			C	C(=)
Length			C	M
FlowCtrl			C	M
Result (-)			S	S
Slot_N				M
ErrorInfo			M	M

Argument
The service-specific parameters are transmitted in the argument.
SendWMessage
This parameter signals, if a W-Message (possibly containing data) shall be added to the Downlink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)
NO (No W-Message needs to be sent)
Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7
Data
This parameter contains the data to transmit. Data type: Octet string

Length
This parameter contains the length of data to transmit. Permitted values: 0 to 32

FlowCtrl
This parameter contains the flow control value (see Table 75).

Result (+):
This selection parameter indicates that the service has been executed successfully.

SendWMessage
This parameter signals, if a W-Message (and possible data) shall be added to the Uplink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)
NO (No W-Message needs to be sent)

Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

Data
This parameter contains the read data values.

Length
This parameter contains the length of the received data package. Permitted values: 0 to 32

FlowCtrl
This parameter contains the flow control value (see Table 75).

Result (-):
This selection parameter indicates that the service failed.

Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available)
STATE_CONFLICT (service unavailable within current state)

6.3.4 DownLinkAck (W-Master)

The service DownLinkAck is only available on the W-Master. The service triggers the appropriate handler (PD handler, Cmd handler, EV handler, or ISDU handler) to provide their data for the next DLink. Also, this service delivers the acknowledgement from the last ULink. With this acknowledgement, each handler has to decide, if new data may be send in DLink, or if the last data have to be retransmitted. The parameters of the service are listed in Table 36.

Table 36 DownLinkAck

Parameter Name	.ind
Argument	M
Slot_N	M
ComChannel	M
Length	M
PreDLSet	C
Acknowledge	M

Argument
The service-specific parameters are transmitted in the argument.

Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

ComChannel
This parameter indicates the selected handler.

Permitted values: PDOUTHANDLER, CMDHANDLER, EVHANDLER, ISDUHANDLER.

Length

This parameter contains the remaining space for the next DLink.

Range: 0 to 37 Bytes

PreDLSet

This parameter is only used for the CMDHANDLER to support LP-Devices which indicates, if the PreDownLink is already in use.

Permitted values:

NO (PreDownLink is empty and can be used)

YES (PreDownLink is already in use)

Acknowledge

This parameter indicates, whether the last uplink has been confirmed or not.

PD handler, Cmd handler, Event handler and ISDU handler shall decide if a retransmit is needed or not.

6.3.5 UpLinkAck (W-Device)

The service UpLinkAck is only available on the W-Device. The service triggers the appropriate handler (PD handler, EV handler, or ISDU handler) to provide data for the next Uplink message. With the Acknowledge from the last downlink each handler has to decide, if new data have to be send, or the last data have to be retransmitted. The parameters of the service are listed in Table 37

Table 37 UpLinkAck

Parameter Name	.ind
Argument	M
ComChannel	M
Length	M
Acknowledge	M

Argument

The service-specific parameters are transmitted in the argument.

ComChannel

This parameter indicates the selected handler.

Permitted values: PDHANDLER, EVHANDLER, ISDUHANDLER.

Length

This parameter contains the remaining space for the next Uplink.

Range: 0 to 15 octets

Acknowledge

This parameter indicates, whether the last uplink has been confirmed or not.

PD handler, Event handler and ISDU handler shall decide if a retransmit is needed or not.

6.3.6 EventMsg (W-Master and W-Device)

The EventMsg service is used to provide events through the diagnosis communication channel. The parameters of the service primitives are listed in Table 38.

Table 38 EventMsg

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M	M	
SendWMessage	M		M	
Slot_N		M	M	
Data	C	C		M
Length	C	M	C	M
FlowCtrl	C	M	C	M

Argument
The service-specific parameters are transmitted in the argument.

SendWMessage
This parameter signals, if a W-Message (possibly containing data) shall be added to the Downlink or Uplink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)
NO (No W-Message needs to be sent)

Slot_N
This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

Data
This parameter contains the whole or segmented Event Data which contains EventQualifier and EventData.
Data type: Octet string (3 Octet)
Note: EventQualifier see A.6.4 in REF 1
EventData see Table 180

Length
This parameter contains the length of data to transmit. If no event shell be sent, set Length to 0.
Permitted values: 0 (W-Master acknowledge) or 3 (W-Device event).

FlowCtrl
This parameter contains the flow control value (see Table 75). In case of EOS (end of service), no data are delivered.

6.3.7 PDOutMsg (W-Master and W-Device)

The PDOutMsg service is used to provide the Process Data through the process communication channel from W-Master to a W-Device. This service delivers the Control Octet (CO) with PDOut data to or from the Message handler. The parameters of the service primitives are listed in Table 39.

Table 39 PDOutMsg

Parameter Name	.req	.ind	.cnf
Argument	M	M	
SendWMessage	M		
Slot_N	M		
Data	C	C	
Length	C	M	
FlowCtrl	C	M	
PDOutInvalid	C	M	
Result (+)			S
Slot_N			M
Result (-)			S
Slot_N			M
ErrorInfo			M

Argument
The service-specific parameters are transmitted in the argument.

SendWMessage
This parameter signals, if a W-Message (and possible data) shall be added to the Downlink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)

NO (No W-Message needs to be sent)

Slot_N

This parameter contains the Slot number (W-Device Address) for the corresponding W-Device. Permitted values: 0 to 7

Data

This parameter contains the whole or segmented Process Data to be transferred from W-Device to W-Master.
Data type: Octet string

Length

This parameter contains the length of the received output Process Data. Permitted values: 0 to 32

FlowCtrl

This parameter contains the flow control value (see Table 75).

PDOutInvalid

This parameter is used to inform the Message handler to generate the “Process Data Out Invalid” via Function Code in the DLink Control Octet

Result (+):

This selection parameter indicates that the service has been executed successfully.

Slot_N

This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

Result (-):

This selection parameter indicates that the service failed.

Slot_N

This parameter contains the Slot number for the corresponding W-Device.
Permitted values: 0 to 7

ErrorInfo

This parameter contains the error information.
Permitted values:
NO_COMM (no communication available)
STATE_CONFLICT (service unavailable within current state)

6.3.8 PDInMsg (W-Master and W-Device)

The PDInMsg service is used to provide the Process Data to be sent through the process communication channel from a W-Device to its W-Master.
This service delivers the Control Octet (CO) with PDIn data to or from the Message handler. The parameters of the service primitives are listed in Table 40.

Table 40 PDInMsg

Parameter Name	.req	.ind	.cnf
Argument	M	M	
SendWMessage	M		
Slot_N		M	
Data	C	C	
Length	C	M	
FlowCtrl	C	M	
PDInInvalid	C	M	
Result (+)			S
Result (-)			S
ErrorInfo			M

Argument

The service-specific parameters are transmitted in the argument.

SendWMessage
This parameter signals, if a W-Message (and possibly also data) shall be added to the Uplink.
Permitted values:
YES (Message handler shall compile the Control Octet and add possible data to transmit)
NO (No W-Message needs to be sent)
Slot_N
This parameter contains the Slot number (Device Address) for the corresponding W-Device.
Permitted values: 0 to 7
Data
This parameter contains the whole or segmented Process Data to be transferred from W-Device to W-Master.
Data type: Octet string
Length
This parameter contains the length of the transmitted input Process Data. Permitted values: 0 to 32
FlowCtrl
This parameter contains the flow control (see Table 75).
PDInInvalid
This parameter is used to inform the Message handler to generate the "Process Data In Invalid"-Function Code in ULink Control Octet
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available)
STATE_CONFLICT (service unavailable within current state)

6.4 Acknowledgments (DownLinkAck and UpLinkAck)

W-Devices acknowledge correct reception of the downlink message within their uplink messages. Within the next downlink, the W-Master acknowledges correct reception of the last uplink messages to each W-Device. In case of negative acknowledgments, both the W-Master and W-Devices use this information to initiate transmission retries.

6.5 Message handler

6.5.1 General

The layer DL-A comprises the Message handler as shown in Figure 60, Figure 61 and Figure 62.

6.5.2 State machine of the W-Master Message handler (DL-A)

Figure 60 shows the state machine of the W-Master Message handler. The two sub-state machines describe the order how the different W-Messages are placed in the downlink payload. The submachine Compile_PDOut (see Figure 61) handles the placement of MasterCommands and process data for all slots (0 to 7) in a first step. In a second step, the submachine Compile_Acyclic (see Figure 62) handles the placement of acyclic data for Event and ISDU.

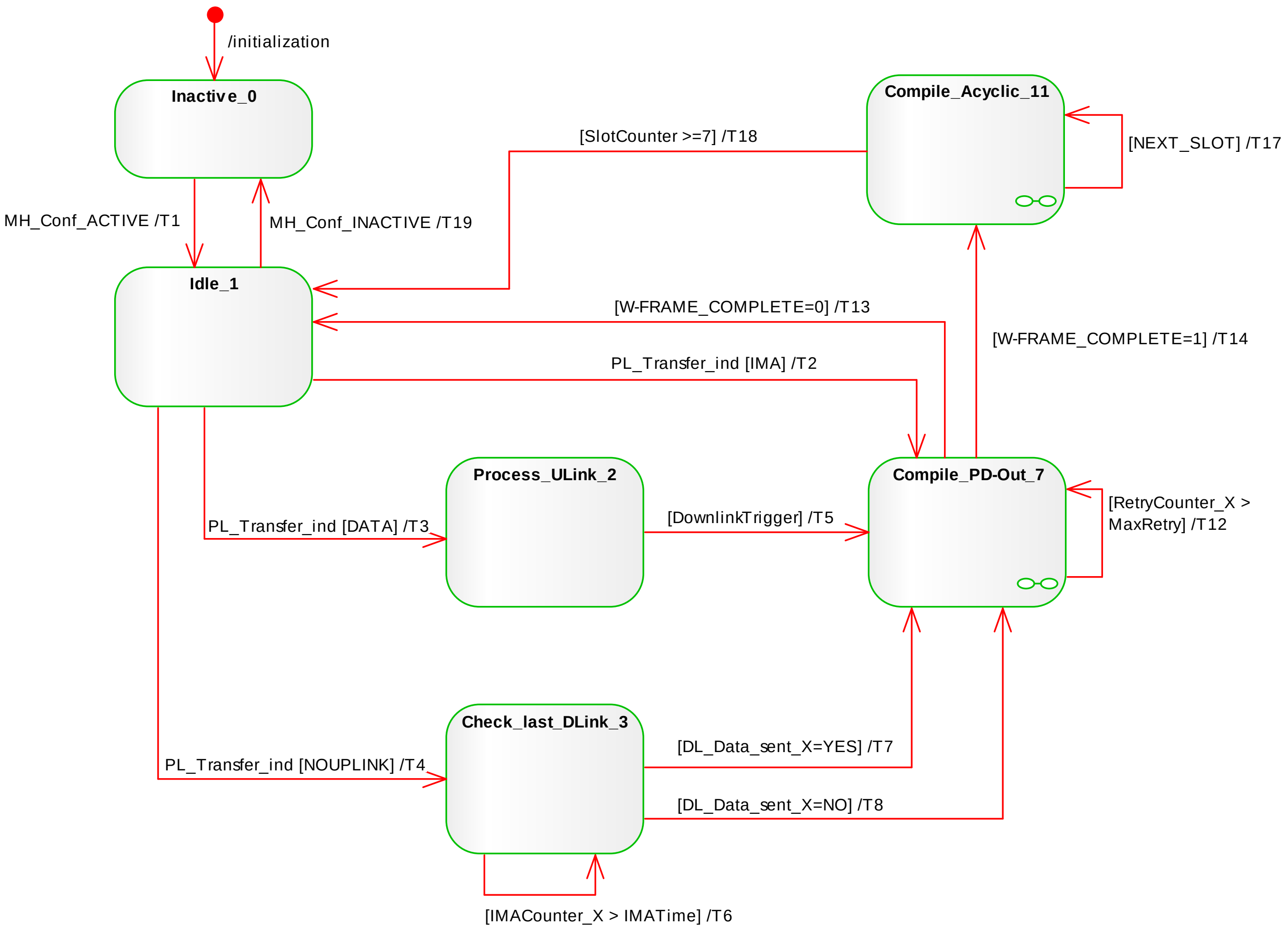
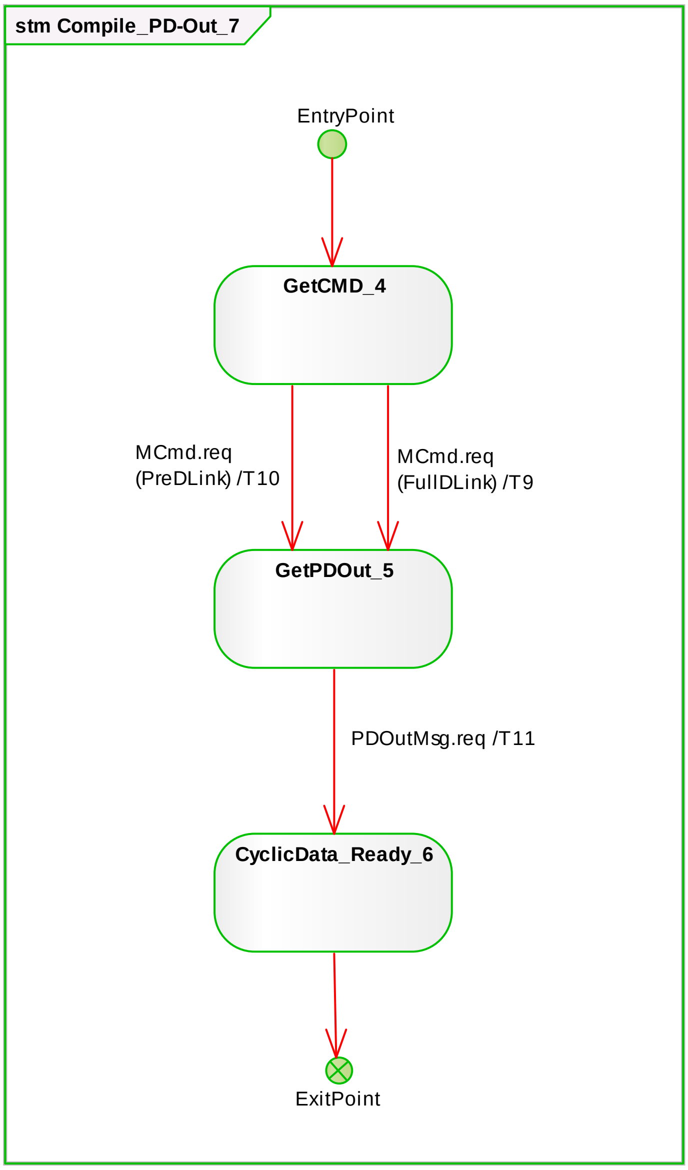


Figure 60 State machine of the W-Master Message handler

2617



2618 **Figure 61 Sub-State machine Compile_PD-Out_7 of the message handler**
2619

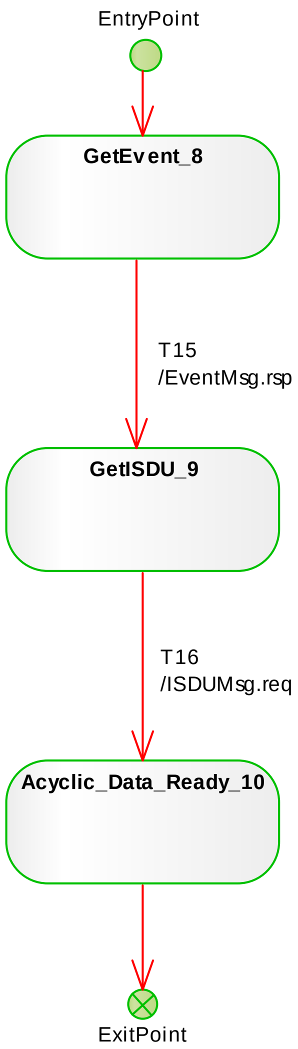


Figure 62 Sub-State machine Compile Acyclic 11 of the message handler

2624 **Table 41 State transition tables of the W-Master Message handler**

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by W-Master DL-mode handler through MH_Conf_ACTIVE (see Figure 65). Set RemainingLength to DLink-Payload (37 Octet).
Idle_1	Waiting for trigger PL_Transfer.ind service indication. The PL_Transfer service delivers the Slotnumber (0 up to 7) and further parameters within a W-Sub-cycle, which represents a W-Device at this Slot.
Process_ULink_2	Check message for valid ULink Control Octets. For message encoding of the ULink Control Octet see Figure 136, ULink Control Octet
Check_last_DLink_3	Check if data have been sent for this W-Device / Slot_N in last downlink
SM: Get_CMD_4	The Message handler starts to compile the message for the next DLink using the DownLinkAck service to acquire a MasterCommand from the Command handler. The Message handler waits on the MCmd.req service and then changes to state GetPDOut_5.
SM: GetPDOut_5	The Message handler uses the DownLinkAck service to acquire PDOut data from the PDOut handler. The Message handler waits for the PDOutMsg.req service to complement an already acquired MCmd.
SM: CyclicData_Ready_6	MasterCommand and / or PDOut data are ready for this Slot_N_X.
Compile_PDOut_7	Compile MCmd and PDOut W-Messages for actual Slot / W-Device as part of the next DLink from the Service MCmd.req and PDOutMsg.req. Each handler shall deliver the DLink Control Octet with its corresponding data. With the internal Variable W-FRAME_COMPLETE all MasterCommands and PDOut data has been compiled for all 0 up to 7 Slot_Ns / Devices. See Sub-State machine MH_XX.
SM: GetEvent_8	The Message handler uses the DownLinkAck service to acquire a possible Event response from the Event handler. The Message handler waits on the EventMsg.rsp service to complement the already acquired PDOut / MCmd.
SM: GetISDU_9	The Message handler uses the DownLinkAck service to acquire ISDU from the ISDU handler. The Message handler waits on the ISDUMsg.req service to complement the already acquired PD / MCmd / Event data.
SM: AcyclicData_Ready_10	Acyclic data (Event, ISDU) are ready for this Slot_N_X.
Compile_Acyclic_11	After the compilation of MasterCommand / PDOut data for each Slot / W-Device, compile acyclic data (Event, ISDU) for all Slots/Devices until the DLink payload is filled up (RemainingLength = 0). Each handler shall deliver the DLink Control Octet with its corresponding data. Remaining acyclic data can be sent in the following DLink, after possible PDOut data are compiled. See Sub-State machine MH_YY.

2625

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /*Remarks*
T1	0	1	*The DL-mode handler activates the Message handler via MH_Conf_ACTIVE.*
T2	1	7	*PL_Transfer.ind reported an IMA ULink.* If PL_Transfer.ind delivers WFrameComplete, set W-FRAME_COMPLETE to 1, otherwise to 0. Set IMACounter_X = 0. Invoke service DownLinkAck (Slotnumber, CMDHANDLER, RemainingLength, PreDLSet, Acknowledge) to acquire MasterCommand from Command handler. Store Acknowledge in ACK_Buf_X for this Slotnumber.
T3	1	2	PL_Transfer.ind reported a received ULink (see Figure 144 and Figure 145) with data for SlotNumber_X. If PL_Transfer.ind delivers WFrameComplete, set W-FRAME_COMPLETE = 1, otherwise to 0. Set IMACounter_X = 0. Set RetryCounter_X = 0. Store Acknowledge in ACK_Buf_X for this Slotnumber.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /*Remarks*
T4	1	3	No ULink has been received for SlotNumber_X. Increment IMACounter_X. If PL_Transfer.ind delivers WFrameComplete, set W-FRAME_COMPLETE = 1, otherwise to 0. Clear Acknowledge in ACK_Buf_X for this Slotnumber.
T5	2	7	Process the received data from ULink with SlotNumber_X to the appropriate handler. Invoke PDInMsg.ind, EventMsg.ind and ISDUMsg.cnf service indications. Invoke service DownLinkAck (Slotnumber, CMDHANDLER, RemainingLength, PreDLSet, Acknowledge) to acquire MasterCommand from Command handler for SlotNumber_X delivered via PL_Transfer.ind.
T6	3	3	A local IMA timeout event shall be reported via invocation of service EventMsg.ind(IOLW_IMATimeout, LOCAL) to the W-Master application.
T7	3	7	For the current SlotNumber_X, data have been sent in last DLink (stored via DL_Data_sent_X) which was not confirmed via the ACK-Bit in ULink (since no ULink received). Increment RetryCounter_X. Set DL_Data_sent_X = 0. Invoke service DownLinkAck (Slotnumber, CMDHANDLER, RemainingLength, PreDLSet, Acknowledge) to acquire MCMD from Command handler for SlotNumber_X delivered via PL_Transfer.ind.
T8	3	7	Invoke service DownLinkAck (Slotnumber, CMDHANDLER, RemainingLength, PreDLSet, Acknowledge) to acquire MCMD from Command handler for the SlotNumber_X delivered via PL_Transfer.ind.
T9	4	5	If MCmd.req(SendWMessage=YES): Compile downlink Control Octet, place in FullDownLink payload and set DL_Data_sent_X = 1. Decrease RemainingLength with the delivered length from (MCmd.req + 1 octet for downlink CO). If MCmd.req(SendWMessage=NO): No compilation of downlink CO necessary. Acquire PDOut for SlotNumber_X through invocation of the DownLinkAck(Slotnumber, PDOUTHANDLER, RemainingLength, Acknowledge) service.
T10	4	5	If MCmd.req(SendWMessage=YES): set DL_Data_sent_X = 1 and set PreDLSet = YES. Acquire PDOut for Slot_X / W-Device_X through invocation of the DownLinkAck(PDOUTHANDLER) service. DownLinkAck service delivers the remaining Payload-Bytes (RemainingLength) for the next DLink and ACK of last DLink-Frame (Retry-Handling) to the PDOut handler.
T11	5	6	If PDOutMsg.req(SendWMessage=YES): Place W-Message to DLink payload and decrease RemainingLength with the delivered length from PDOutMsg.req – 2 (for Control Octet). Set DL_Data_sent_X = 1.
T12	7	7	A local MaxRetry event shall be reported via invocation of service EventMsg.ind(IOLWM_Retry_Error LOCAL) to the W-Master application.
T13	7	1	W-Frame is not complete. Wait for next ULink / next Slotnumber via PL_Transfer.ind in state Idle_1.
T14	7	11	W-Frame is complete, all ULinks have been received. Compile data for Event and ISDU for all Slots subsequently via T17: Set SlotCounter to 0. Acquire Event through invocation of the DownLinkAck(SlotCounter, EVHANDLER, RemainingLength, ACK_Buf_X) service.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T15	8	9	If EventMsg.rsp(SendWMessage=YES): Set DL_Data_sent_X = 1. Place W-Message to DLink payload and decrease RemainingLength with the delivered length from EventMsg.rsp – 2 (for Control Octet). Acquire ISDU through invocation of the DownLinkAck(SlotCounter, ISDUHANDLER, RemainingLength, ACK_Buf_X) service.
T16	9	10	If ISDUMsg.req(SendWMessage=YES): Set DL_Data_sent_X = 1, place W-Message to DLink payload and increment SlotCounter to aquire data for next Slot.
T17	11	11	Invoke DownLinkAck(SlotCounter, EVHANDLER, RemainingLength, ACK_Buf_X) to acquire acyclic data (Event, ISDU) for the next Slot counted in SlotCounter. See Sub-State machine Compile_Acyclic_11.
T18	11	1	All acyclic data for all Slots / Devices have been acquired. Downlink is ready to send. Invoke PL_Transfer.req to send DLink within the next W-Sub-Cycle. Set RemainingLength to DLink-Payload (37 Octet) for composition of the following DLink. Set PreDLSet = NO to indicate a free PreDownLink for the next W-Sub-cycle.
T19	1	0	*W-Device Message handler changes state to Inactive_0.*

INTERNAL ITEMS	TYPE	DEFINITION
RemainingLength	Variable	Remaining length in DLink payload.
W-FRAME_COMPLETE	Variable	Marks the W-Sub-cycle as completed, if all ULinks have been processed.
SlotCounter	Variable	Counter to compile the acyclic data for all Slots / Devices
RetryCounter_X	Variable	Counter for not acknowledged DLinks.
IMACounter_X	Variable	Counter to observe ULink-IMA-frames which shall be sent by W-Device_X.
DL_Data_sent_X	Variable	Variable to store the information, that data have been sent in last DLink for the corresponding Slot / W-Device.
PreDLSet	Bool	Marks if the PreDownlink will be used or not.

Note 1: _X marks the variables which individual for every Slotnumber. The range of _X is 0 to 7 SlotNumbers

Note 2: If a W-Message shall be sent (SendWMessage = YES), the Message handler must compile the control octet of the corresponding DL-B handler as defined in Table 42 Compilation of Downlink Control Octet.

6.5.3 Compilation of DLink Control Octet

The Master Message handler (see Figure 60) shall compile the control octet for a DLink delivered with the data via the corresponding DL-B handler as defined in Table 42.
See Figure 134 for definition of DLink Control Octet.

Table 42 Compilation of Downlink Control Octet

Compiled Control Octet handler: → ↓	**Slot-number** (delivered by handler)	**Channel Code (ChC)** (created by Message handler)	**Flow Control (FC)** (delivered by handler)	**Data Length (DLen)** (delivered by handler)	**Data follows**
MCmd.req	Slot_N	**5** (MasterCommand)	MasterCommand (delivered by handler)		**No**
PDOutMsg.req	Slot_N	**1** (Process data out)	FlowCtrl	0 to 31 See Note 1	**Yes**
			FlowCtrl (ABORT)	0	**No**
		2 (Process data out invalid)	-	-	**No**
EventMsg.rsp (Event Ack)	Slot_N	**4** (EVENT)	-	-	**No**
ISDUMsg.req	Slot_N	**3** (ISDU)	FlowCtrl	0 to 31 See Note 1	**Yes**
			FlowCtrl = EOS or ABORT	0	**No**
Empty Downlink See Note 2	-	**0** (INVALID)	-	-	**No**

Note 1:
Data Length is coded from 0 to 31 which means, that the transmitted data are 1 to 32 Octet.

Note 2:
An empty downlink (all payload data zero) is automatically created by PL, if the W-Master has no data to send to any W-Device.

6.5.4 State machine of the W-Device Message handler (DL-A)

Figure 63 shows the state machine of the W-Device Message handler. The Message handler is triggered via PL for each W-Sub-cycle to distribute received W-Messages and / or as trigger to send W-Messages within an ULink. The sub state machine CreateMessage_8 handles the compilation of ULink W-Messages in a predefined order.

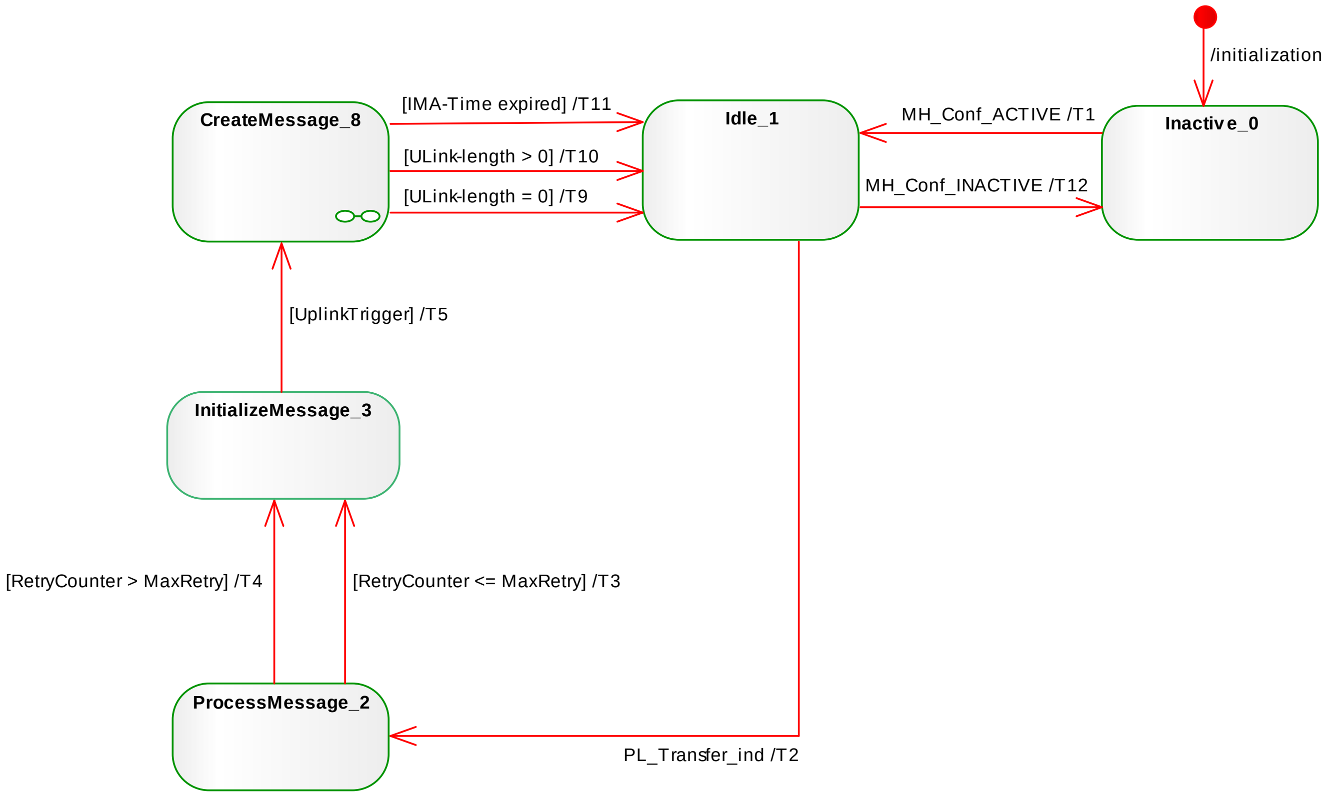


Figure 63 State machine of the W-Device Message handler

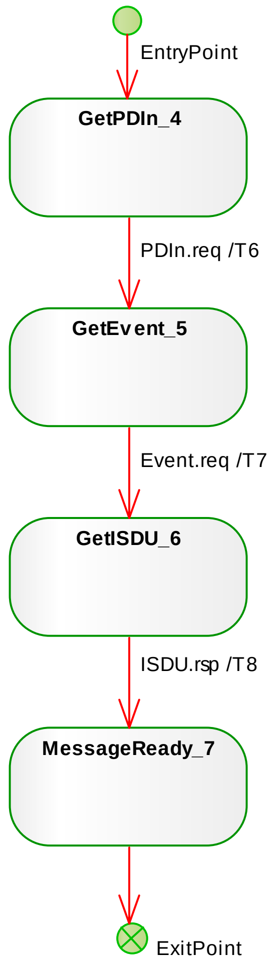


Figure 64 W-Device Message handler sub state machine “CreateMessage_8” (DL-A)

2660

Table 43 State transition tables of the W-Device Message handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through MH_Conf_ACTIVE (see Table 43, Transition T1).
Idle_1	Waiting for Trigger (each W-Sub-cycle) through PL_Transfer.ind service indication (T2).
ProcessMessage_2	Check message for valid DLink Control Octet. For message encoding of the DLink Control Octet see Figure 134
InitializeMessage_3	Set RemainingLength of ULink payload to 2 (SSlot-W-Device) or 15 (DSlot-W-Device).
SM: GetPDIn_4	The Message handler starts to compile the message for the next ULink using the UpLinkAck service to acquire PDIn from the Process Data handler. The Message handler waits on the PDInMsg.req service and then changes to state GetEvent_5.
SM: GetEvent_5	The Message handler uses the UpLinkAck service to acquire an Event from the Event handler. The Message handler waits on the EventMsg.req service to complement the already acquired PDIn.
SM: GetISDU_6	The Message handler uses the UpLinkAck service to acquire ISDUMsg.rsp from the ISDU handler. The Message handler waits on the ISDUMsg service to complement the already acquired PD / Event.
SM: Message_Ready_7	ULink data ready
CreateMessage_8	Compile Message for next ULink from PDInMsg.req, EventMsg.req and ISDUMsg.rsp services (see submachine). For the Message encoding of the ULink Control Octet see Figure 136.

2661

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*DL-mode handler activates Message handler via MH_Conf_ACTIVE.*
T2	1	2	*Service PL_Transfer_ind indicates a received (or lost) DLink.* If PL_Transfer_ind delivers pos. Acknowledge, set IMACounter = 0, Set RetryCounter = 0. If PL_Transfer_ind delivers neg. Acknowledge, increment RetryCounter.
T3	2	3	-
T4	2	3	A real-time fault shall be reported via invocation of service DL_MaxRetry. *Note:* *The parameter MaxRetry is delivered via service DL_SetParam.*
T5	3	4	Invoke MCmd.ind, ISDUMsg.ind and PDOutMsg.ind service indications to distribute received W-Messages. Acquire PDIn through invocation of the service UpLinkAck(PDHANDLER, RemainingLength, Acknowledge).
T6	4	5	If PDInMsg.req(SendWMessage=YES): Place W-Message to ULink payload and decrease RemainingLength with the delivered length from PDInMsg.req – 1 (for Control Octet). Acquire Event through invocation of the service UpLinkAck(EVENTHANDLER, RemainingLength, Acknowledge).
T7	5	6	If EventMsg.req(SendWMessage=YES): Place W-Message to ULink payload and decrease RemainingLength with the delivered length from PDInMsg.req – 1 (for Control Octet). Acquire ISDU through invocation of the service UpLinkAck(ISDUHANDLER, RemainingLength, Acknowledge).
T8	6	7	If ISDUMsg.rsp(SendWMessage=YES): Place W-Message to ULink payload and set RemainingLength to 2 (SSlot-W-Device) or 15 (DSlot-W-Device).
T9	8	1	*No ULink-Data have to be sent.* Increment IMACounter.
T10	8	1	Invoke service PL_Transfer.req(Data, DataLength) with ULink-Data for transmission to W-Master.
T11	8	1	To indicate its presence to W-Master, the W-Device shall send an IMA-Frame, if IMACounter >= SendIMA through invocation of service PL_Transfer.req(DataLength=0).
T12	1	0	*The W-Device Message handler changes state to Inactive_0.*

2662

INTERNAL ITEMS	TYPE	DEFINITION
RemainingLength	Variable	Remaining length in ULink payload.
RetryCounter	Variable	Counter for not acknowledged ULinks.
IMACounter	Variable	Counter to send ULink-IMA-frames.
SendIMA	Variable	Limit for IMACounter (see T11) to send an IMA-ULink to W-Master. This value is calculated by the following formula: SendIMA = (IMATime in W-Sub-cycle)-MaxRetry-10

2663

6.5.5 Compilation of ULink Control Octet

The Device Message handler (see Figure 63) shall compile the control octet for an ULink delivered via the corresponding DL-B handler as defined in Table 44. See Figure 136 for definition of ULink Control Octet.

Table 44 Compilation of Uplink Control Octet

Compiled Control Octet → **handler:** ↓	**Channel Code (ChC)** (created by Message handler)	**Flow Control (FC)** (delivered by handler) See Note 1	**Data follows**
PDInMsg.req	**1** (Process data in)	FlowCtrl	**Yes**
		FlowCtrl = ABORT	**No**
	2 (Process data in invalid)	**-**	**No**
EventMsg.req	**4** (EVENT)	FlowCtrl	**Yes**
		FlowCtrl = EOS or ABORT	**No**
ISDUMsg.rsp	**3** (ISDU)	FlowCtrl	**Yes**
		FlowCtrl = EOS or ABORT	**No**
IMA Uplink	**No Control Octet needed.** See Table 26		

Note: For uplink W-Messages the length of data is coded in the Flow Control.

7 Data Link Layer (DL-B)

7.1 DL-B services

7.1.1 Overview of services within W-Master and W-Device

This clause defines the services of the data link layer to be provided to the application layer and system management via its external interfaces. Table 45 lists the assignments of W-Master and W-Device to their roles as initiator or receiver for the individual DL services. Empty fields indicate no availability of this service on W-Master or W-Device.

Table 45 Service assignments within W-Master and W-Device

Service name	W-Master	W-Device
DL_PDTrig	I	I
DL_PDInputTransport	I	
DL_Control	I, R	I, R
DL_PDOutputUpdate	R	
DL_PDOutputTransport		I
DL_PDInputUpdate		R
DL_Event	I	R
DL_ISDUTransport	R	I
DL_ISDUAbort	R	I
DL_TDConfig	R	
DL_Read	R	I
DL_Write	R	I
DL_SetMode	R	
DL_Mode	I	I
DL_MaxRetry		I
DL_SetParam	R	R
Key (see 3.3.5) All services are defined from the view of the affected layer towards the layer above. - I Initiator of a service (towards the layer above) - R Receiver (responder) of a service (from the layer above)		

7.1.2 DL_PDTrig (W-Master and W-Device)

The data link layer uses the DL_PDTrig service to indicate the end of a W-MasterCycleTime period after start of Process Data reception to the application layer. This service has no parameters. The service primitives are listed in Table 46

Table 46 DL_PDTrig

Parameter Name	.ind
<none>	

7.1.3 DL_PDInputTransport (W-Master)

The data link layer on the W-Master uses the DL_PDInputTransport service to transfer the content of input data (Process Data from W-Device to W-Master) to the application layer. The parameters of the service primitives are listed in Table 47

Table 47 DL_PDInputTransport

Parameter Name	.ind
Argument	M
InputData	M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data to be transmitted to the application layer.
Parameter type: Octet string

7.1.4 DL_Control (W-Master and W-Device)

The W-Master uses the DL_Control service to convey control information via the process data channel to the corresponding technology specific device application and to get control information via the PD handler (see clause 12.9 PDVALID PDINVALID). The parameters of the service primitives are listed in Table 48.

Table 48 DL_Control

Parameter Name	.req	.ind
Argument	M	M
ControlCode	M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

ControlCode
This parameter indicates the status of the Process Data (PD)
Permitted values:
PDIN_VALID (Input Process Data valid)
PDIN_INVALID (Input Process Data invalid)
PDOUT_VALID (Output Process Data valid)
PDOUT_INVALID (Output Process Data invalid or missing)

7.1.5 DL_PDOutputUpdate (W-Master)

The W-Master's application layer uses the DL_PDOutputUpdate service to update the output data (Process Data from W-Master to W-Device) on the data link layer. The parameters of the service primitives are listed in Table 49.

Table 49 DL_PDOutputUpdate

Parameter Name	.req	.cnf
Argument	M	
OutputData	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

OutputData
This parameter contains the Process Data provided by the application layer.
Parameter type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.1.6 DL_PDOutputTransport (W-Device)

The data link layer on the W-Device uses the DL_PDOutputTransport service to transfer the content of output Process Data to the application layer (from W-Master to W-Device). The parameters of the service primitives are listed in Table 50.

Table 50 DL_PDOutputTransport

Parameter Name	.ind
Argument	M
OutputData	M

Argument
The service-specific parameters are transmitted in the argument.

OutputData
This parameter contains the Process Data to be transmitted to the application layer.
Parameter type: Octet string

7.1.7 DL_PDInputUpdate (W-Device)

The W-Device's application layer uses the DL_PDInputUpdate service to update the input data (Process Data from W-Device to W-Master) on the data link layer. The parameters of the service primitives are listed in Table 51.

Table 51 DL_PDInputUpdate

Parameter Name	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data provided by the application layer.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state

7.1.8 DL_Event (W-Master and W-Device)

The service DL_Event transfers a status or error information. The W-Device application triggers the Event transfer. Additional DL_Event requests are ignored until the previous one has been confirmed (see Figure 77, Sequence chart for Event). The parameters of the service primitives are listed in Table 52.

Table 52 DL_Event

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
Instance	M	M		
Type	M	M		
Mode	M	M		
EventCode	M	M		

Argument
The service-specific parameters are transmitted in the argument.

Instance
This parameter indicates the Event source.
Permitted values: Application (see Table 136, see Table A.17 in REF 1)

Type
This parameter indicates the Event category.
Permitted values: ERROR, WARNING, NOTIFICATION (see Table 138, see Table A.19 in REF 1)

Mode
This parameter indicates the Event mode.
Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table 139, see Table A.20 in REF 1)

EventCode
This parameter contains a code identifying a certain Event (see clause 15, see Table D.1 in REF 1).
Parameter type: 16 bit unsigned integer

7.1.9 DL_ISDUTransport (W-Master and W-Device)

The DL_ISDUTransport service is used to transport an ISDU. This service is used by the W-Master to send a service request from the W-Master application layer to the W-Device. It is used by the W-Device to send a service response to the W-Master from the W-Device application layer. The parameters of the service primitives are listed in Table 53.

Table 53 DL_ISDUTransport

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
ValueList	M	M		
Result (+)			S	S
Data			C	C
Qualifier			M	M
Result (-)			S	S
ISDUTransportErrorInfo			M	M

Argument
The service-specific parameters are transmitted in the argument.

ValueList
This parameter contains the relevant operating parameters
Parameter type: Record

Index
Permitted values: 0 to 65535

Subindex
Permitted values: 0 to 255

Data
Parameter type: Octet string

Direction
Permitted values:
READ (Read operation),
WRITE (Write operation)

Result (+):
This selection parameter indicates that the service has been executed successfully.

Data
Parameter type: Octet string

Qualifier
Permitted values: an I-Service W-Device response according to clause 7.4.312.11.1, see Table 70

Result (-):
This selection parameter indicates that the service failed.

ISDUTransportErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state),
ISDU_TIMEOUT (ISDU acknowledgement time elapsed, see Figure 84, see Table 97 in REF 1),
VALUE_OUT_OF_RANGE (Service parameter value violates range definitions)

7.1.10 DL_ISDUAbort (W-Master and W-Device)

The DL_ISDUAbort service aborts the current ISDU transmission. The service primitives are listed in Table 54.

Table 54 DL_ISDUAbort

Parameter Name	.req	.ind
<none>		

7.1.11 DL_TDConfig (W-Master)

The DL_TDConfig service is used to configure the mapping of a W-Port to the corresponding Track and Slot via W-Port Configuration Manager / System Management. The service primitives are listed in Table 55.

Table 55 DL_TDConfig (W-Master)

Parameter Name	.req	.cnf
Argument	M	
ValueList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ValueList
This parameter contains the parameters for the TDmapper. Parameter type: Record

Track_N
This parameter contains the track number.
Permitted values: 0 to 4

Slot_N
This parameter contains the Slot number for the corresponding W-Device
Permitted values: 0 to 7

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)
STATE_CONFLICT (service unavailable within current state)

7.1.12 DL_Read (W-Master and W-Device)

The DL_Read service is used by system management to read a W-Device parameter value in direct parameter page 1 or in the extended wireless parameter pages via ISDU. Therefore DL_Read uses the DL_ISDUTransport service. The parameters of the service primitives are listed in Table 56.

Table 56 DL_Read

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
Index	M	M		
Subindex	M	M		
Result (+)			M	S
Value			M	M(=)
Result (-)				S
ErrorInfo				M

Argument
The service-specific parameters are transmitted in the argument.
Index
This parameter contains the Index of the W-Device parameters in Page 1 or in the extended wireless parameter page (see Table 168Table 164).
Permitted values: see Table 171Table 168
Subindex
This parameter contains the Subindex of the W-Device parameter in Page 1 (see Table 164) or in the extended wireless parameter page.
Permitted values: For Page 1 values 1 to 15, for extended wireless parameters, see clause 14.3
Value
This parameter contains the W-Device parameter value to be written.
Result (+):
This selection parameter indicates that the service has been executed successfully.
Value
This parameter contains read W-Device parameter values.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.1.13 DL_Write (W-Master and W-Device)

The DL_Write service is used by system management to write a W-Device parameter value to direct parameter page 1 or to the extended wireless parameter pages via ISDU. Therefore DL_Write uses the ISDUMsg service for ISDU and the MCmd service in case of a Master command. The parameters of the service primitives are listed inTable 57.

Table 57 DL_Write

Parameter Name	.req	.ind	.cnf
Argument	M	M	
Index	M	M	
Subindex	M	M	
Value	M	M	
Result (+)			S
Result (-)			S
ErrorInfo			M

Argument
The service-specific parameters are transmitted in the argument.

Index
This parameter contains the Index of the W-Device parameters in Page 1 or in the extended wireless parameter page (see Table 168Table 164).
Permitted values: see Table 171Table 168

Subindex
This parameter contains the Subindex of the W-Device parameter in Page 1 (see Table 164) or in the extended wireless parameter page.
Permitted values: For Page 1 values 1 to 15, for extended wireless parameters, see clause 14.3

Value
This parameter contains the W-Device parameter value to be written.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.1.14 DL_SetMode (W-Master)

The DL_SetMode service is used by system management to set up the data link layer's state machines and to send the characteristic values required for operation to the data link layer. The parameters of the service primitives are listed in Table 58.

Table 58 DL_SetMode

Parameter Name	.req	.cnf
Argument	M	
Mode	M	
ValueList	U	
Result (+)		S
W-Port		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Mode
This parameter indicates the requested mode of the W-Master's DL on an individual W-Port.
Permitted values:
INACTIVE (handler shall change to the INACTIVE state),
STARTUP (handler shall change to STARTUP state),
PREOPERATE (handler shall change to PREOPERATE state),
OPERATE (handler shall change to OPERATE state)

ValueList
This parameter contains the relevant operating parameters.
Data structure: record
PDInputLength (to be propagated to Message handler and PDIn handler)
PDOutputLength (to be propagated to Message handler)

Result (+):
This selection parameter indicates that the service has been executed successfully.

W-Port
This parameter contains the number of the related W-Port.

Result (-):
This selection parameter indicates that the service failed.

W-Port
This parameter contains the number of the related W-Port.

ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state),
PARAMETER_CONFLICT (consistency of parameter set violated)

7.1.15 DL_Mode (W-Master and W-Device)

The DL uses the DL_Mode service to report to system management that a certain operating status has been reached. The parameters of the service primitives are listed inTable 59.

Table 59 DL_Mode

Parameter Name	.ind
Argument	M
W-Port	C
RealMode	M

Argument
The service-specific parameters are transmitted in the argument.

RealMode
This parameter indicates the status of the DL-mode handler.
Permitted values:
INACTIVE (handler changed to the INACTIVE state)
COMLOST (communication lost)
ACTIVE (handler changed to the ACTIVE state)
STARTUP (handler changed to the STARTUP state)
PREOPERATE (handler changed to the PREOPERATE state)
OPERATE (handler changed to the OPERATE state)

7.1.16 DL_MaxRetry (W-Device)

The service DL_MaxRetry indicates a real-time fault to application for W-Device dependent error handling, when RetryCounter exceeded the configured value MaxRetry.
The parameters of the service are listed in
Table 60.

Table 60 DL_MaxRetry

Parameter Name	.ind
<none>	

7.1.17 DL_SetParam (W-Master and W-Device)

The DL_SetParam service is used to change parameters for retry and IMA handling in the Message handler.
The parameters of the service primitives are listed in
Table 61.

Table 61 DL_SetParam

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured communication parameters for a W-Device.
Parameter type: Record
Record Elements:

MAXRetry
This parameter contains the maximum number of allowed retries in count of W-Sub-Cycles (see clause 14.3.6). This info is delivered to the Message handler and the W-Master-PDOut handler.

IMATime
This parameter contains the I am alive time(see clause 14.3.5). This info is delivered to the Message handler.

MaxPDSegLength (only W-Master)
This parameter contains the maximum segment length of the PDOut data to the Message handler to distribute PDOut data within multiple W-Cycles. This info is delivered to the W-Master-PDOut handler.

LowPowerDevice
This info is delivered to the CommandHandler, ISDU handler and Process Data handler to switch a low energy W-Device to PreDownLink or FullDownLink.

Permitted values: YES, NO.

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information.

Permitted values:

VALUE_OUT_OF_RANGE (service parameter value violates range definitions)

7.2 DL-mode handler

7.2.1 General

The W-Master DL-mode handler is responsible to establish communication using services of the Physical Layer (PL) and internal administrative calls to control and monitor the states of other handlers.

The W-Device DL-mode handler receives MasterCommands to synchronize with the W-Master DL-mode handler states STARTUP, PREOPERATE, OPERATE and manages the activation and deactivation of handlers as appropriate.

7.2.2 State machine of the W-Master DL-mode handler

After reception of the service DL_SetMode(STARTUP) from system management, the W-Master waits for synchronization with the W-Device.
The purpose of state "Startup_2" is to check a W-Device's identity in state "PreOperate_3", the W-Master may assign parameters to the W-Device using ISDUs. Cyclic exchange of Process Data is performed in state "Operate". Within this state additional data such as Commands, Events and ISDUs can be transmitted.

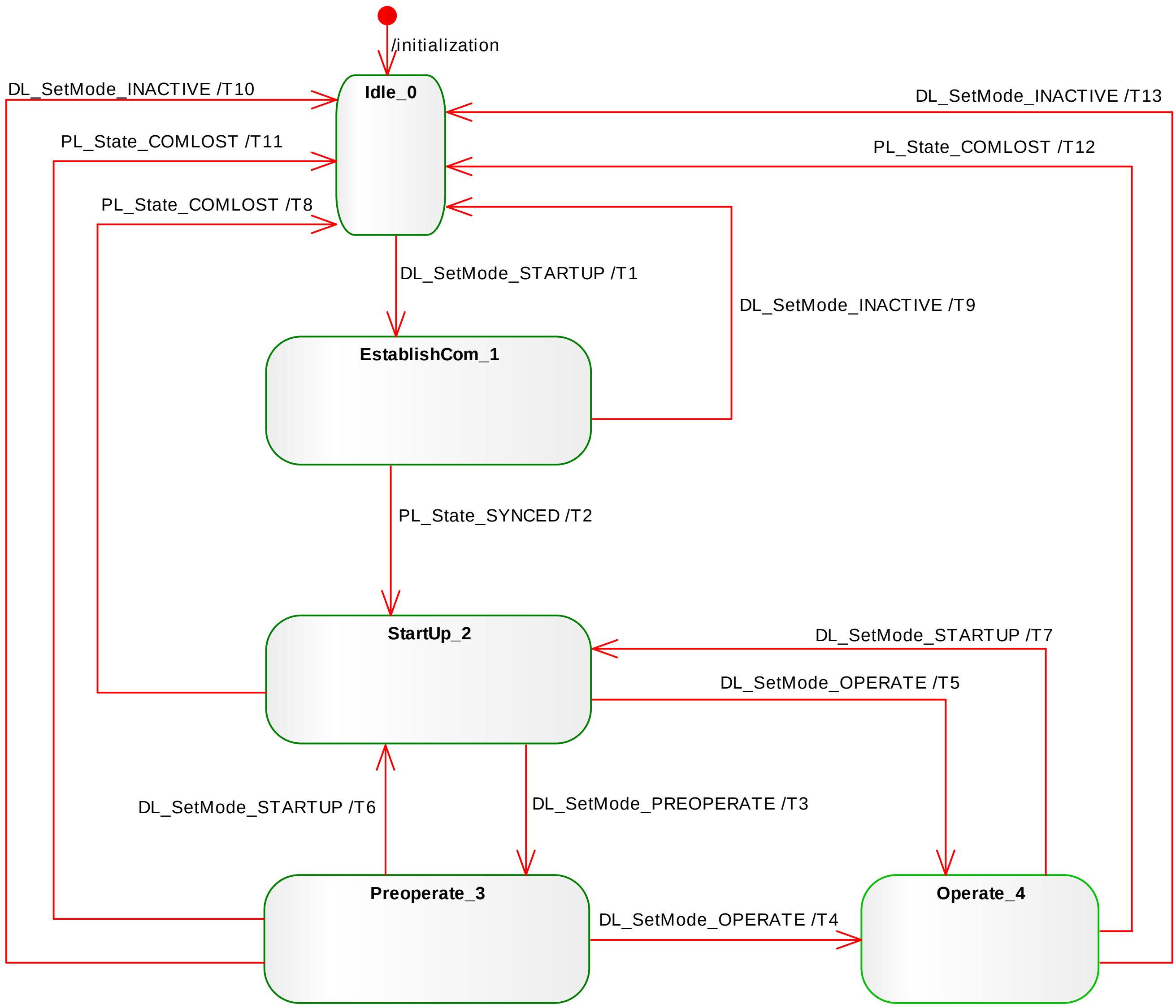


Figure 65 State Machine of the W-Master DL-mode handler

3041 **Table 62 State transition tables of the W-Master DL-mode handler**

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting for communication request from System Management (SM): DL_SetMode (STARTUP)
EstablishCom_1	Waiting for synchronization with W-Device
Startup_2	System Management uses the STARTUP state for W-Device identification, check and communication configuration (see Figure 96).
Preoperate_3	Commands, Events and ISDU without Process Data
Operate_4	Process Data, Commands, Events and ISDU

3042

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke MH_Conf_ACTIVE to activate Message handler.
T2	1	2	Activate Command handler (call CH_Conf_ACTIVE see Table 62 DL-mode handler W-Master) and ISDU handler (call IH_Conf_ACTIVE see Figure 72 ISDU-handler W-Master) Indicate state via service DL_Mode.ind (ACTIVE) and DL_Mode.ind (STARTUP) to SM.
T3	2	3	SM requested the PREOPERATE state. Activate Event handler (call EH_Conf_ACTIVE see Figure 72. EventHandler). Invoke DL_Mode.ind (PREOPERATE) to SM.
T4	3	4	SM requested the OPERATE state. Activate the Process Data handler (PD_Conf_ACTIVE see PDHandler W-Master). Invoke DL_Mode.ind (OPERATE) to SM.
T5	2	4	SM requested the OPERATE state. Activate the Process Data handler (call PD_Conf_ACTIVE see Figure PDHandler W-Master) and Event handler (call EH_Conf_ACTIVE see Figure 72..EVHandler W-Master). Invoke DL_Mode.ind (OPERATE) to SM.
T6	3	2	SM requested the STARTUP state. Deactivate Event handler (call EH_Conf_INACTIVE). Invoke DL_Mode.ind (STARTUP) to SM.
T7	4	2	SM requested the STARTUP state. Deactivate Process Data (call PD_Conf_INACTIVE) and Event handler (call EH_Conf_INACTIVE). Invoke DL_Mode.ind (STARTUP) to SM.
T8	2	0	Physical Layer delivers state through Service PL_State.ind(COMLOST) Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (COMLOST) to SM.
T9	1	0	SM requested the INACTIVE state. Deactivate all handlers (call xx_Conf_INACTIVE). Invoke DL_Mode.ind (INACTIVE) to SM.
T10	3	0	SM requested the INACTIVE state. Deactivate all handlers (call xx_Conf_INACTIVE). Invoke DL_Mode.ind (INACTIVE) to SM.
T11	3	0	Physical Layer delivers state through Service PL_State.ind(COMLOST) Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (COMLOST) to SM (see Figure 96. State machine of the W-Port handler)
T12	4	0	Physical Layer delivers state through Service PL_State.ind(COMLOST) Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (COMLOST) to SM
T13	4	0	SM requested the INACTIVE state. Deactivate all handlers (call xx_Conf_INACTIVE). Invoke DL_Mode.ind (INACTIVE) to SM.

INTERNAL ITEMS	TYPE	DEFINITION
xx_Conf_ACTIVE	Call	This call activates the respective handler. xx is substitute for MH (Message handler), IH (ISDU handler), CH (Command handler), and/or EH (Event handler)
xx_Conf_INACTIVE	Call	This call deactivates the Message handler. xx is substitute for MH (Message handler), IH (ISDU handler), CH (Command handler), and/or EH (Event handler)

7.2.3 State machine of the W-Device DL-mode handler

Figure 66 shows the state machine of the W-Device DL-mode handler. In state PreOperate_2 and Operate_3 different sets of handlers within the W-Device are activated.

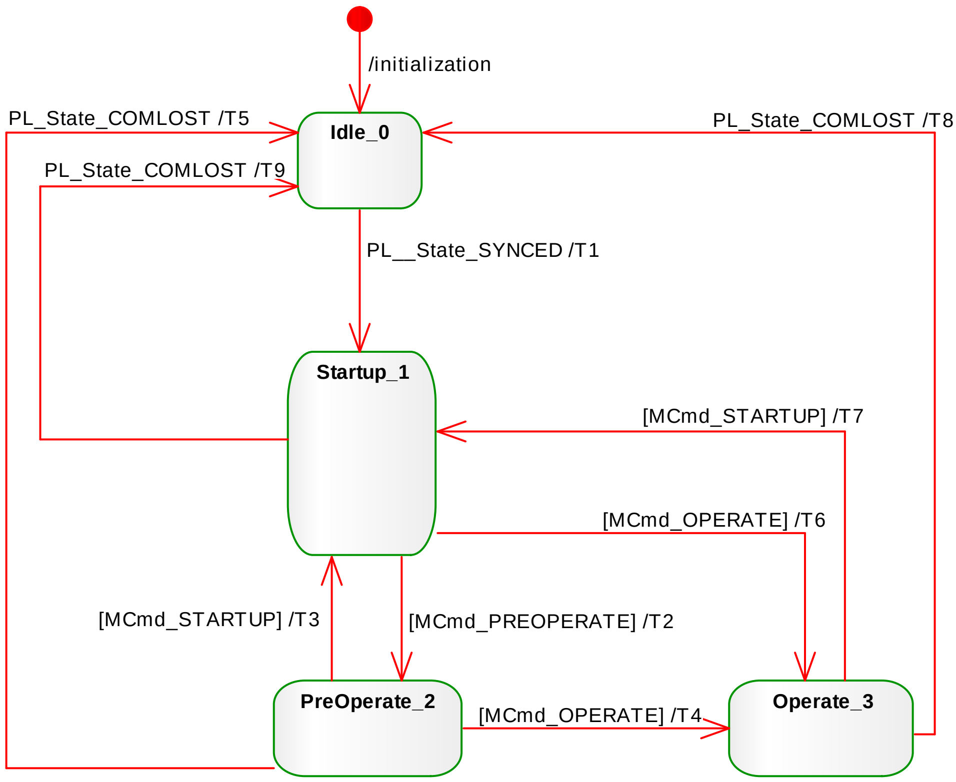


Figure 66 State machine of the W-Device DL-mode handler

Table 63 State transition tables of the W-Device DL-mode handler

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting for established connection
Startup_1	Compatibility check (see Figure 97)
PreOperate_2	On-request Data exchange (parameter, commands, Events) without Process Data
Operate_3	Process Data (PD) and On-request Data exchange (parameter, commands, Events)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Physical Layer delivers state through service PL_State.ind(SYNCED).* Activate Message handler (call MH_Conf_ACTIVE in Figure 63) , ISDU handler (call IH_Conf_ACTIVE in Figure 73) and Command handler (call CH_Conf_ACTIVE in Figure 75). Indicate state via service DL_Mode.ind (ACTIVE) to SM.
T2	1	2	*W-Device command handler received MasterCommand (MCmd_PREOPERATE).* Activate Event handler (call EH_Conf_ACTIVE in Figure 78). Indicate state via service DL_Mode.ind (PREOPERATE) to SM.
T3	2	1	*W-Device command handler received MasterCommand (MCmd_STARTUP).* Deactivate Event handler (call EH_Conf_INACTIVE in Figure 78). Indicate state via service DL_Mode.ind (STARTUP) to SM.
T4	2	3	*W-Device command handler received MasterCommand (MCmd_OPERATE).* Activate Process Data handler (call PD_Conf_ACTIVE in Figure 70). Indicate state via service DL_Mode.ind (OPERATE) to SM.
T5	2	0	*Physical Layer delivers state through Service PL_State.ind(COMLOST).* Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM (see Figure **106** and Table 120)
T6	1	3	*W-Device command handler received MasterCommand (MCmd_OPERATE).* Activate Process Data handler (call PD_Conf_ACTIVE in Figure 70) and Event handler (call EH_Conf_ACTIVE in Figure 78). Indicate state via service DL_Mode.ind (OPERATE) to SM.
T7	3	1	*W-Device command handler received MasterCommand (MCmd_STARTUP).* Deactivate Process Data handler (call PD_Conf_INACTIVE in Figure 70) and Event handler (call EH_Conf_INACTIVE in Figure 78). Indicate state via service DL_Mode.ind (STARTUP) to SM.
T8	3	0	*Physical Layer delivers state through Service PL_State.ind(COMLOST).* Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM (see Figure 106 and Table 120)
T9	1	0	*Physical Layer delivers state through Service PL_State.ind(COMLOST).* Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM (see Figure **106** and Table 120)

7.3 Process Data handler

The transport of output Process Data is performed using the services DL_PDOutputUpdate and DL_PDOutputTransport.
The transport of input Process Data is performed using the services DL_PDInputUpdate and DL_PDInputTransport.
Via service DL_Control Process Data can be set to valid or invalid.

7.3.1 State machine of the W-Master Process Data Out handler

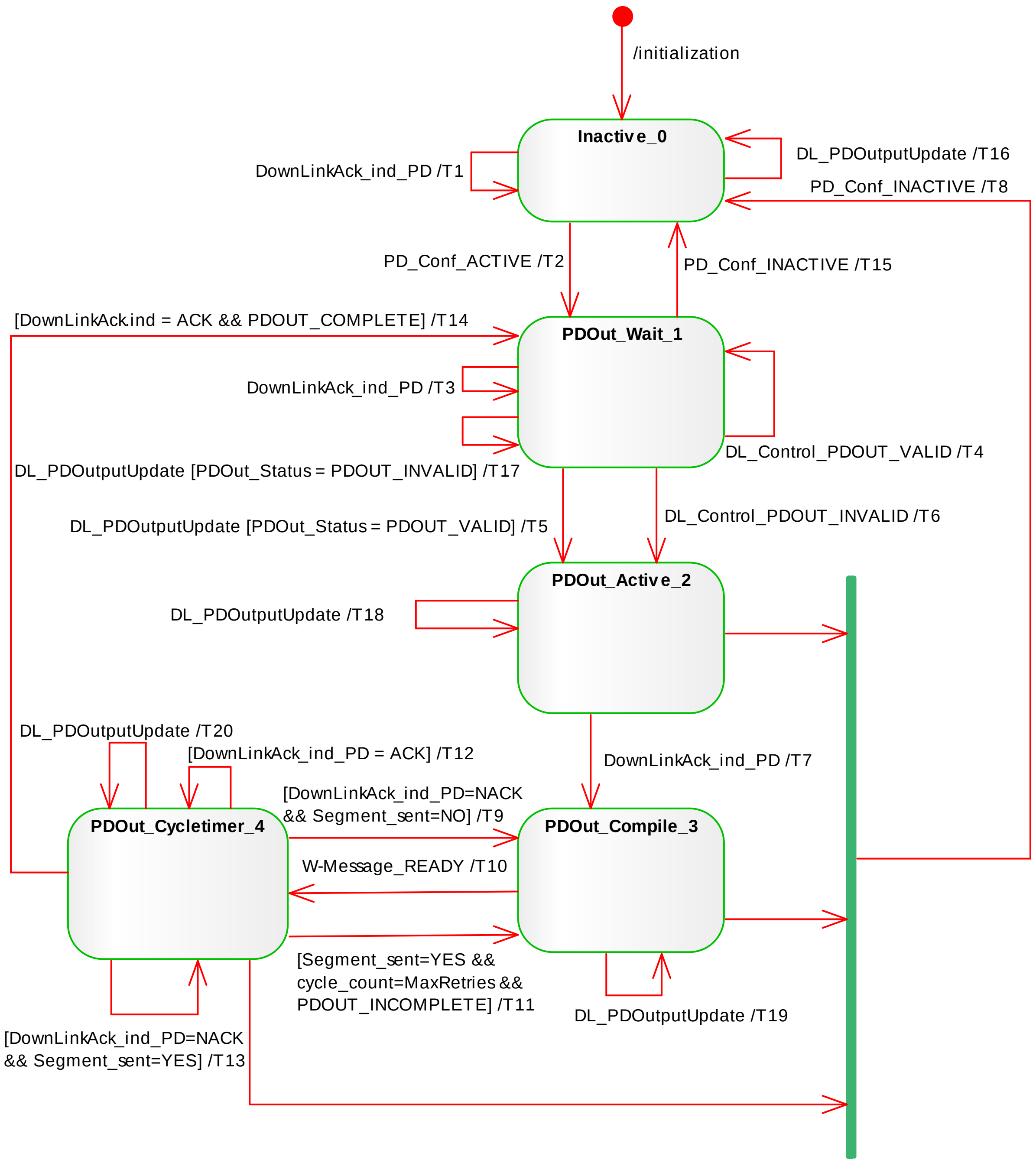


Figure 67 State machine for W-Master PDOut handler

Table 64 StateTransition tables of the W-Master PDOut handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through PD_Conf_ACTIVE (see Table 63, Transition T1).
PDOut_Wait_1	Waiting for DL_PDOutputUpdate from application.
PDOut_Active_2	handler active and waiting on DownLinkAck_ind_PD.
PDOut_Compile_3	Compile W-Message under conditions of DLink Control Octet (see Figure 134, DLink Control Octet) Maximum segment length shall be limited by parameter MaxPDSegLength (via DL_SetParam) to distribute PDOut data (see Figure 128 PDOut distribution sequence chart) Set Variable PDOut_Completion to PDOUT_COMPLETE if all PDOut data Octets are transmitted otherwise set to PDOUT_INCOMPLETE. PDOut-Data transmission uses the mechanism of segmented data transfer, see.7.7.1 “Transmission of Segmented Data (PD, EV, ISDU)”. For Retry-Handling see 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU)”.
PDOut_Cycletimer_4	Handle timing / distribution for PDOut segmented data within multiple W-Sub-cycles. If a data segment was not acknowledged, send retry immediately with next W-Sub-cycle. In case of an acknowledged data segment wait for „x“ W-Sub-cycles and send the next data segment (distribution). Note: „x“ = MaxRetry

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	No Process Data (PDOut) to send, invoke PDOutMsg.req(SendWMessage = NO).
T2	0	1	*W-Master DL-mode handler enables Process Data handler via PD_Conf_ACTIVE.* Set PD_OutStatus to PDOUT_INVALID
T3	1	1	No Process Data (PDOut) to send, invoke PDOutMsg.req(SendWMessage = NO).
T4	1	1	Set PD_OutStatus to PDOUT_VALID.
T5	1	2	In case of low energy W-Device: Invoke MCmd.req(FullDLink) and wait for pos. Acknowledge from Cmd handler to switch low energy W-Device to FullDownLink.
T6	1	2	In case of low energy W-Device: Invoke MCmd.req(FullDLink) and wait for pos. Acknowledge from Cmd handler to switch a low energy W-Device to FullDownLink. Set PD_OutStatus to PDOUT_INVALID.
T7	2	3	Set cycle_count = 0
T8	2, 3, 4	0	*W-Master DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T9	4	3	Increase cycle_count, if cycle_count equals MaxRetries set cycle_count back to 0*. Resend data in next W-Message (retry.)*
T10	3	4	If „PD_OutStatus = PDOUT_INVALID“ invoke PDOutMsg.req(PDOutInvalid), otherwise invoke PDOutMsg.req to output Process Data with max. Length of MaxPDSegLength Octets to Message handler PDOutMsg.req(SendWMessage = YES, Slot_N, Data, Length, FlowCtrl).
T11	4	3	Set cycle_count = 0, Set Segment_sent = NO (send new segment in next W-Message).
T12	4	4	Set Segment_sent = YES, no Process Data (PDOut) to send, invoke PDOutMsg.req(SendWMessage = NO). Increase cycle_count.
T13	4	4	No Process Data (PDOut) to send, invoke PDOutMsg.req(SendWMessage = NO). Increase cycle_count.
T14	4	1	*Last PDOut transmission (last segment) is complete and acknowledged.* In case of low energy W-Device: Invoke MCmd.req(PreDLink) to switch low energy W-Device back to PreDownLink. If „PD_OutStatus = PDOUT_VALID“ invoke DL_PDOutputUpdate.cnf
T15	1	0	*W-Master DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T16	0	0	Invoke DL_PDOutputUpdate.cnf(NO_COMM)
T17	1	1	Invoke DL_PDOutputUpdate.cnf(STATE_CONFLICT)
T18	2	2	Invoke DL_PDOutputUpdate.cnf(STATE_CONFLICT)
T19	3	3	Invoke DL_PDOutputUpdate.cnf(STATE_CONFLICT)
T20	4	4	Invoke DL_PDOutputUpdate.cnf(STATE_CONFLICT)

INTERNAL ITEMS	TYPE	DEFINITION
PD_OutStatus	Variable	Indicate if PDOut is valid or invalid 0 = PDOUT_INVALID 1 = PDOUT_VALID
cycle_count	Variable	Counting variable for W-Sub-cycles
Segment_sent	Variable	Indicate if segment is sent and acknowledged. NO = Segment not sent and acknowledged YES = Segment sent and acknowledged
PDOut_Completion	Variable	Indicate if PDOut transmission is complete. 0 = PDOUT_INCOMPLETE 1 = PDOUT_COMPLETE

7.3.1.1 Sequence diagram for PDOut distribution

This sequence chart shows an example communication between W-Master Message handler and W-Master PDOut handler, dependent on the following parameters, configured via SM_SetPortConfig / DL_SetParam. The parameters are used to distribute PDOut data in one or more W-Cycles, if e.g. a W-Cycle of 5 ms is not needed.

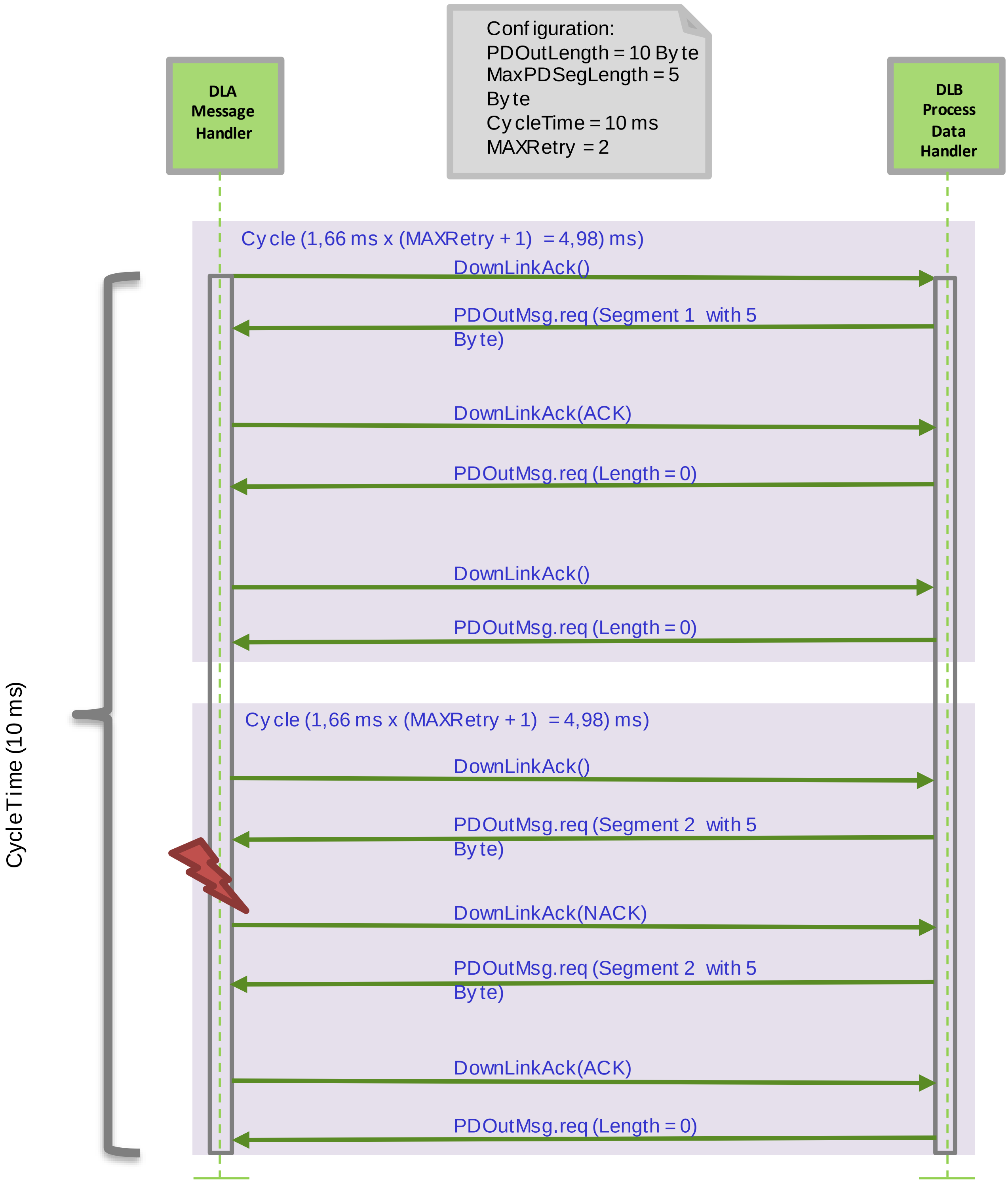
MaxPDSegLength:

Limits the PDOut data which shall be delivered to the Message handler.
E.g. by this the PDOutData will be splitted in 2 W-Cycles.

MaxRetry:

Contains the maximum number of allowed retries for the last sent data(segment)

3084



3085 **Figure 68 Sequence diagram for PDOut distribution**

3086

7.3.2 State machine of the W-Master Process Data In handler

Figure 69 shows the state machine of the W-Master Process Data In handler.

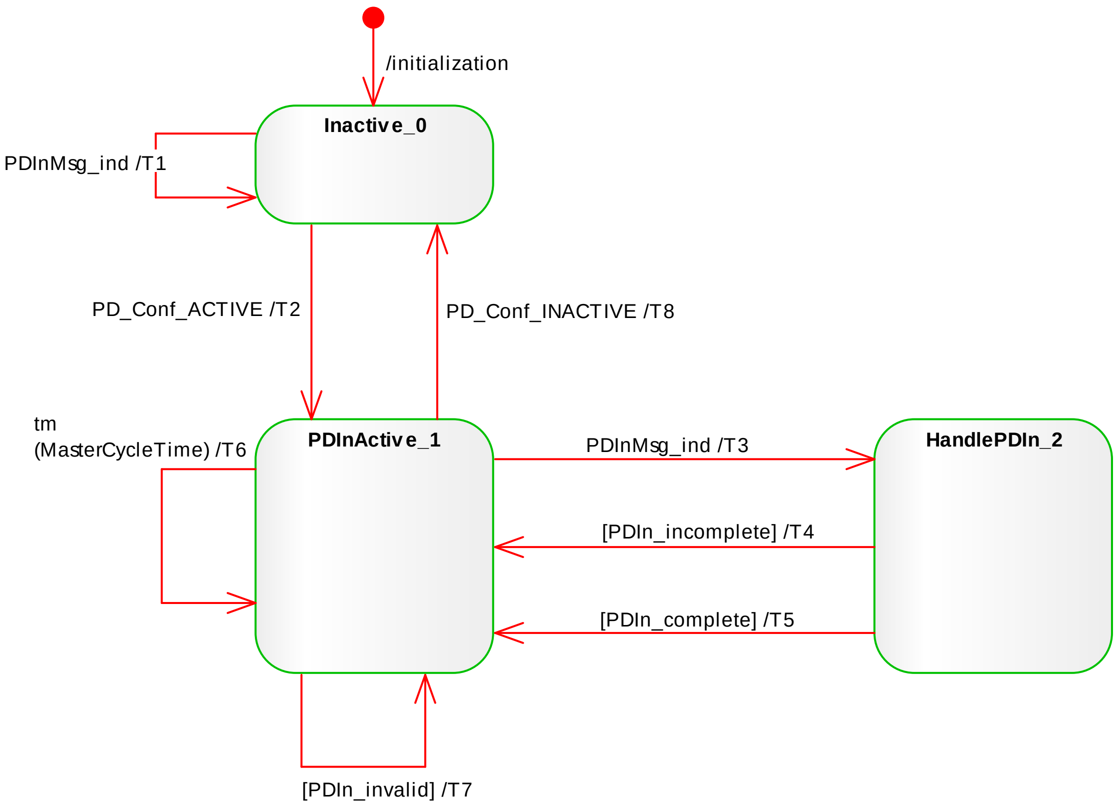


Figure 69 State machine for W-Master PDIn handler

Table 65 State transition tables of the W-Master PDIn handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through PD_Conf_ACTIVE (see Table 63, Device DL-Mode-handler Transition T1).
PDInActive_1	handler active and waiting for next Message handler demand via PDInMsg.ind service.
Handle_PDIn_2	Handle PDIn-Data. PDIn-Data transmission uses the mechanism of segmented data transfer, see. 7.7.1 “Transmission of Segmented Data (PD, EV, ISDU)”. For Retry-Handling see 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU)”.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	*Ignore Process Data (PDIn).*
T2	0	1	*W-Master DL-mode handler enables Process Data handler via PD_Conf_ACTIVE.*
T3	1	2	*Message handler delivers input Process Data or segment of input Process Data.* Start Timer "MasterCycleTime"(one-shot, not retriggerable) at each start of Process Data reception. See Note 1.
T4	2	1	-
T5	2	1	Invoke DL_PDInputTransport.ind (see 7.1.3) Invoke DL_Control.ind (PDIN_VALID).
T6	2	1	Invoke DL_PDTrig.ind. See Note 1.
T7	1	1	DLink Control Octet contained „Process Data In Invalid". Invoke DL_Control.ind (PDIN_INVALID).
T8	1	0	*W-Master DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*

INTERNAL ITEMS	TYPE	DEFINITION
PDIn_complete	Guard	All segements have been received, PDin data are complete and received data length of all segments are equal to the W-Device ProcessDataIn (see Table 164)
PDIn_incomplete	Guard	PDin data are incomplete, wait for next data segment or received data length of all segments are not equal to the W-Device ProcessDataIn (see Table 164)
PDIn_invalid	Service	PDInMsg.ind delivered PDIN_INVALID (via ControlOctet)

Note 1: To minimize Jitter caused by different transmission qualities, especially with segmented data (variations on the numbers of retries) PDTrig can be used to get an equidistant time between reception of first data packet and activation of PDTrig.

7.3.3 State machine of the W-Device Process Data Out handler

Figure 70 shows the state machine of the W-Device Process Data Out handler.

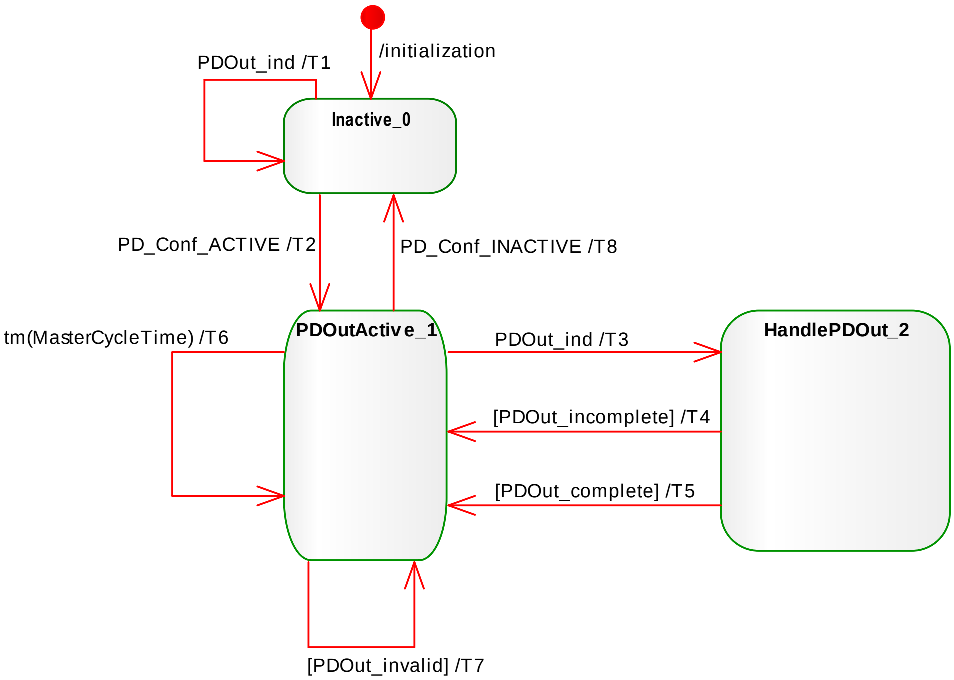


Figure 70 State machine of the W-Device Process Data Out handler

Table 66 State transition tables of the W-Device PDOut handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through PD_Conf_ACTIVE (see Table 63, Transition T1).
PDOutActive_1	handler active and waiting on next Message handler demand via PDOutMsg.ind service.
Handle_PDOut_2	Handle PDOut-Data. PDOut-Data transmission uses the mechanism of segmented data transfer, see. 7.7.1 Transmission of Segmented Data (PD, EV, ISDU). For Retry-Handling see clause 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU)”.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	*Ignore Process Data (PDOut).*
T2	0	1	*W-Device DL-mode handler enables Process Data handler via PD_Conf_ACTIVE.*
T3	1	2	*Message handler delivers output Process Data or segment of output Process Data.* Start Timer “MasterCycleTime“ (one-shot, not retriggerable) at each start of Process Data reception.
T4	2	1	-
T5	2	1	Invoke DL_PDOutputTransport.ind (see 7.1.6) Invoke DL_Control.ind (PDOUTVALID).
T6	2	1	Invoke DL_PDTrig.ind (see 7.1.2).
T7	1	1	DLink Control Octet contained „Process Data Out Invalid”. Invoke DL_Control.ind (PDOUTINVALID).
T8	1	0	*DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*

7.3.4 State machine of the W-Device Process Data In handler

Figure 71 shows the state machine of the W-Device Process Data In handler.

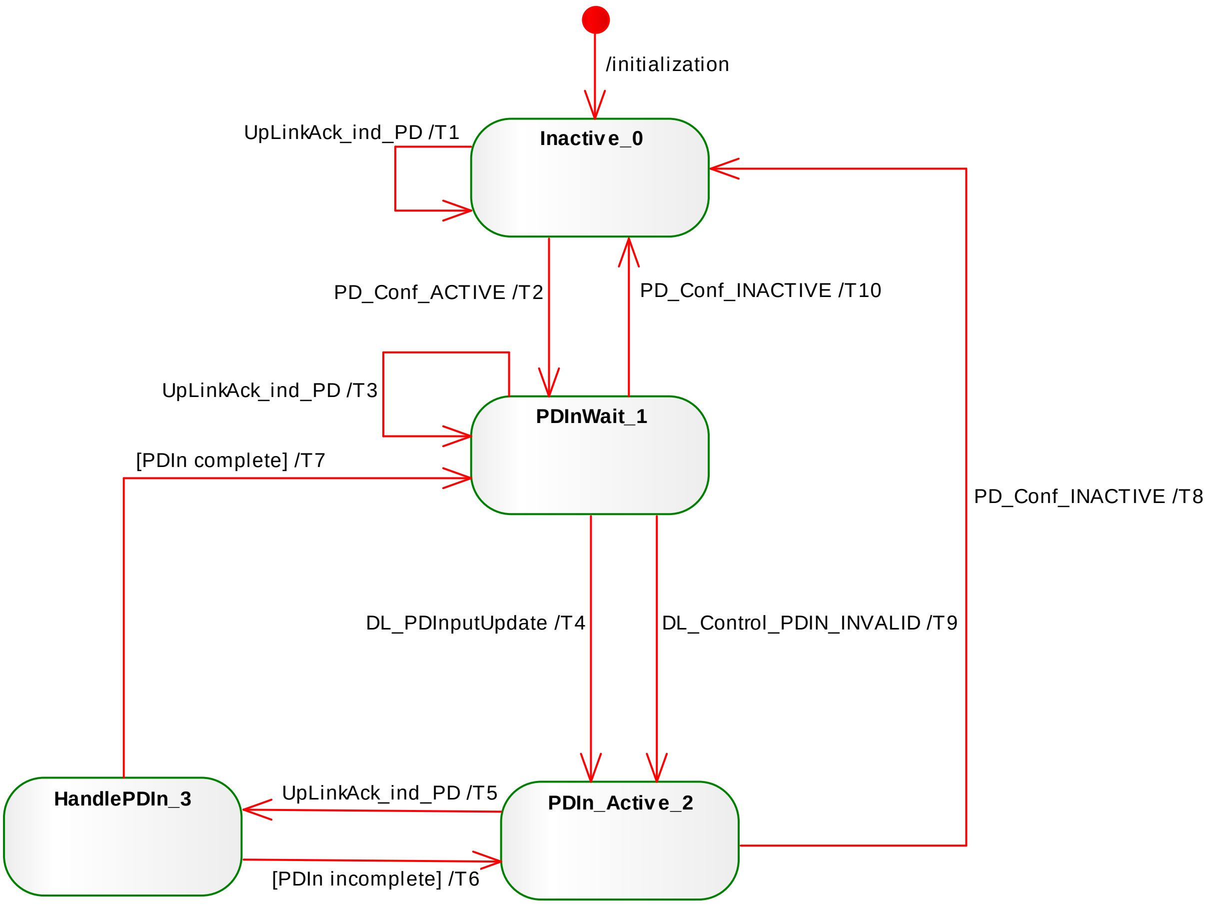


Figure 71 State machine of the W-Device Process Data In handler

Table 67 State transition tables of the W-Device PDIn handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through PD_Conf_ACTIVE (see Table 63, Transition T1).
PDInWait_1	Waiting for DL_PDInputUpdate from application.
PDInActive_2	handler active and waiting on UpLinkAck_ind_PD.
Handle_PDIn_3	Handle PDIn-Data. PDIn-Data transmission uses the mechanism of segmented data transfer, see. 7.7.1 Transmission of Segmented Data (PD, EV, ISDU). For Retry-Handling see 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU)”.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	No Process Data (PDin) to send, invoke PDin.req(SendWMessage = NO).
T2	0	1	*W-Device DL-mode handler enables Process Data handler via PD_Conf_ACTIVE.*
T3	1	1	No Process Data (PDin) to send, invoke PDin.req(SendWMessage = NO).
T4	1	2	Prepare input Process Data for PDInMsg.req for next Message handler demand
T5	2	3	*Message handler requests PDIn-Data.* Invoke PDInMsg.req to deliver input Process Data to Message handler PDInMsg.req(SendWMessage, Data, Length, FlowCtrl).
T6	3	2	-
T7	3	1	*Last PDIn transmission (last segment) is complete and acknowledged.* Invoke DL_PDInputUpdate.cnf
T8	2	0	*DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T9	1	2	Invoke PDInMsg.req(PDIN_INVALID) to generate „Process Data In Invalid“ in ULink Control Octet.
T10	1	0	*DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T11	2	2	Invoke DL_PDInputUpdate.cnf(STATE_CONFLICT)
T12	3	3	Invoke DL_PDInputUpdate.cnf(STATE_CONFLICT)
T13	0	0	Invoke DL_PDInputUpdate.cnf(NO_COMM)

7.4 Indexed Service Data Unit (ISDU) handler

The general structure of an ISDU is demonstrated in Figure 48 and specified in detail in Clause A.5.in REF 1

The ISDU allows accessing data objects (parameters and system commands) to be transmitted. The data objects shall be addressed by the "Index" element.

7.4.1 State machine of the W-Master ISDU handler

Figure 72 shows the state machine of the W-Master ISDU handler

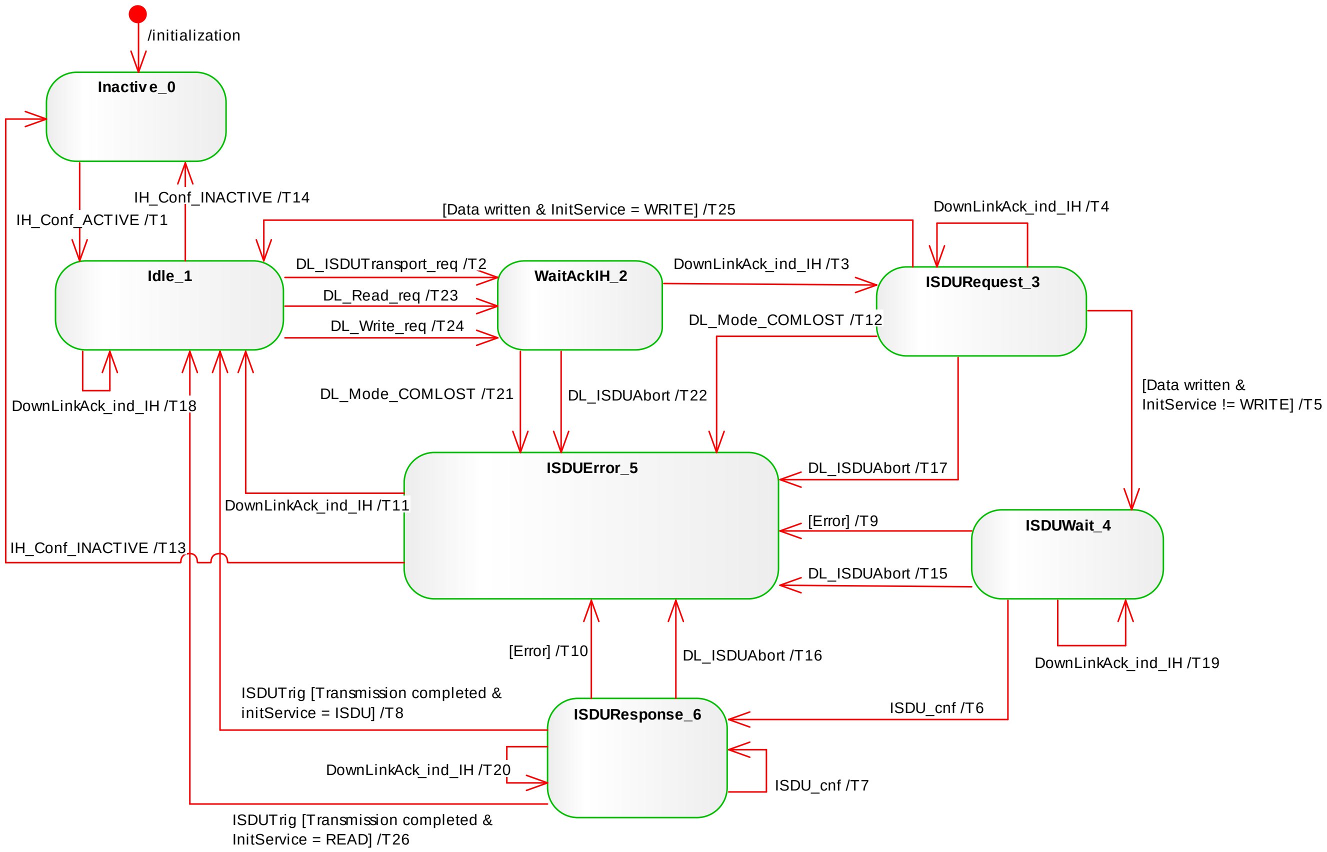


Figure 72 State machine of the W-Master ISDU handler

Table 68 State transition tables of the W-Master ISDU handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
Idle_1	Waiting for transmission of next ISDU Data. Services DL_Read and DL_Write are mapped to DL_ISDU_Transport.
WaitAckIH_2	Waiting for DownlinkAck_IH
ISDURequest_3	Transmission of ISDU request data. ISDU data transmission uses the mechanism of segmented data transfer "see 7.7.1 Transmission of Segmented Data (PD, EV, ISDU)"). For Retry-Handling see 7.7.2 "Retry-Handling of segmented Data (PD, EV, ISDU)".
ISDUWait_4	Waiting for response from W-Device. Observe ISDUTime
ISDUError_5	Error handling after detected errors: Invoke negative DL_ISDU_Transport response with ISDUTransportErrorInfo
ISDUResponse_6	Get response data from W-Device. ISDU data transmission uses the mechanism of segmented data transfer "see 7.7.1 Transmission of Segmented Data (PD, EV, ISDU)"). For Retry-Handling see 7.7.2 "Retry-Handling of segmented Data (PD, EV, ISDU)".

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	-
T2	1	2	Set InitService to ISDU In case of low energy W-Device: Invoke MCmd.req(FullDLink) and wait for pos. Acknowledge from Cmd handler to switch low energy W-Device to FullDownLink
T3	2	3	If enough space left in the downlink, invoke ISDUMsg.req (SendWMessage = YES, Slot_N, Length, data, flowCtrl = START).
T4	2	2	If enough space left in the downlink, invoke ISDUMsg.req with FlowCTRL under conditions of Table 75 (FlowCTRL / DLink CO) ISDUMsg.req(SendWMessage = YES, Slot_N, Length, data, flowCtrl). After all data were sent, invoke ISDUMsg.req with EOS (without data) ISDUMsg.req(SendWMessage = YES, Slot_N, EOS) (see 7.7.1 Transmission of segmented data)
T5	2	3	Start timer (ISDUTime)
T6	3	5	Stop timer (ISDUTime)
T7	5	5	Receive ISDU response data via ISDUMsg.cnf
T8	5	1	Invoke positive DL_ ISDUTransport confirmation when last segment (EOS) has been received (see 7.7.1 Transmission of segmented data) In case of low energy W-Device : Invoke MCmd.req(PreDLink) to switch low energy W-Device to PreDownLink
T9	3	4	-
T10	5	4	-
T11	4	1	*On receiving DownLinkAck_ind_IH invoke ISDUMsg.req with ISDU abortion:* ISDUMsg.req (flowCtrl = ABORT). Invoke negative DL_ ISDUTransport confirmation In case of low energy W-Device : Invoke MCmd.req(PreDLink) to switch low energy W-Device to PreDownLink
T12	2	4	-
T13	4	1	*In case of lost communication, the Message handler informs the DL-mode handler which in turn uses the administrative call IH_Conf_INACTIVE. No actions during this transition required.*
T14	1	0	-
T15	3	4	-
T16	5	4	-
T17	2	4	-
T18	1	1	*No ISDU data to send.* Invoke ISDUMsg.req(SendWMessage = NO).
T19	3	3	*No ISDU data to send.* Invoke ISDUMsg.req(SendWMessage = NO).
T20	5	5	*No ISDU data to send.* Invoke ISDUMsg.req(SendWMessage = NO).
T21	2	5	-
T22	2	5	-
T23	1	2	Set InitService to READ In case of low energy W-Device: Invoke MCmd.req(FullDLink) and wait for pos. Acknowledge from Cmd handler to switch low energy W-Device to FullDownLink
T24	1	2	Set InitService to WRITE In case of low energy W-Device: Invoke MCmd.req(FullDLink) and wait for pos. Acknowledge from Cmd handler to switch low energy W-Device to FullDownLink
T25	3	1	Return DL_Write.cnf

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T26	6	1	Return positive DL_Read confirmation when last segment (EOS) has been received (see clause 7.7.1 Transmission of segmented data) In case of low energy W-Device : Invoke MCmd.req(PreDLink) to switch low energy W-Device to PreDownLink

INTERNAL ITEMS	TYPE	DEFINITION
Data written	Service	Last segment (EOS) has been received and acknowledged
ISDUTime	Time	Measurement of W-Device response time (ISDU acknowledgement time, see Table 97 in REF 1)
Error	Variable	Any detectable error within the ISDU transmission or DL_ISDUAbort requests, or any violation of the ISDU acknowledgement time
InitService	Guard	Variable to store the service request DL_ISDUTransport.req = ISDU DL_Read.req = READ DL_Write.req = WRITE

7.4.2 State machine of the W-Device ISDU handler

Figure 73 shows the state machine of the W-Device ISDU handler.

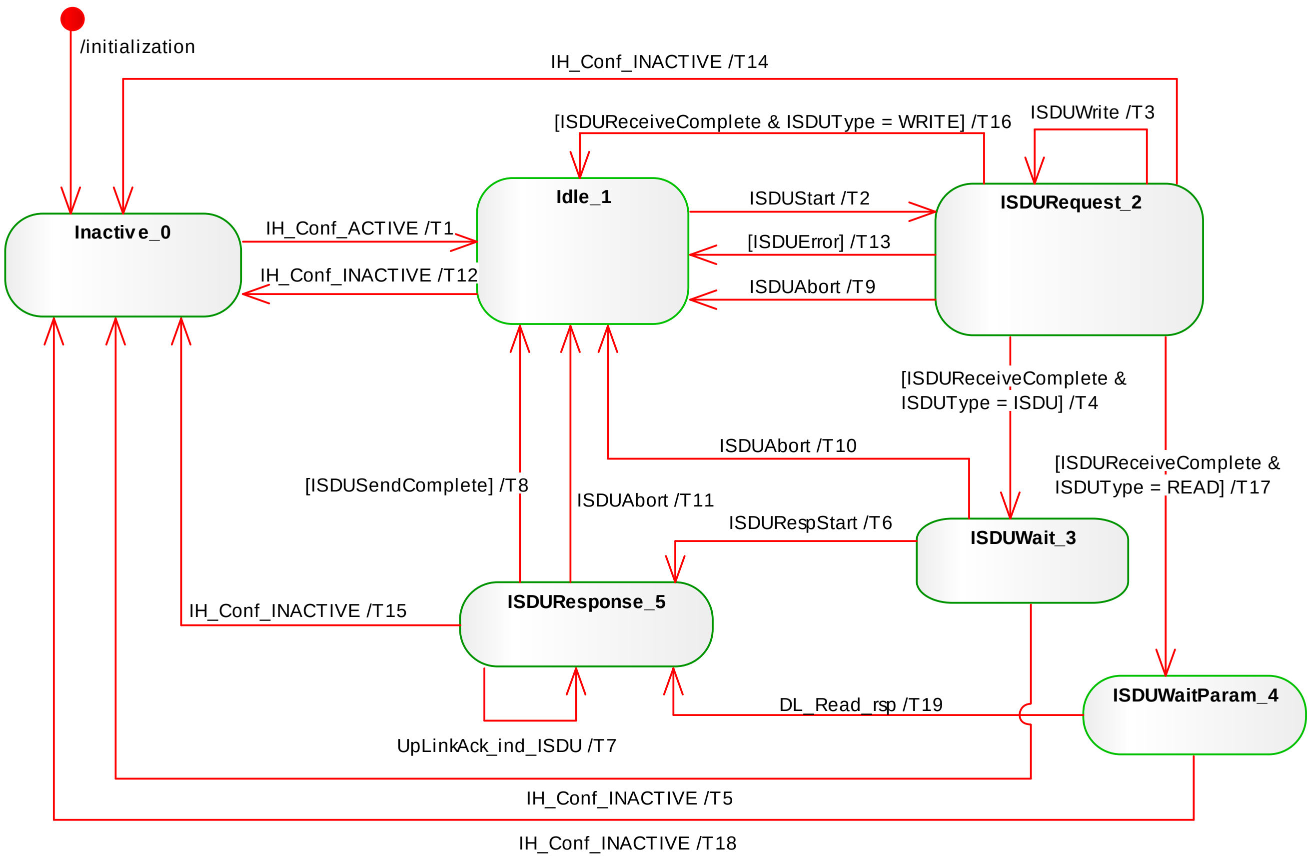


Figure 73 State machine of the W-Device ISDU handler

3139 **Table 69 State transition tables of the W-Device ISDU handler**

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through IH_Conf_ACTIVE (see Table 63, Transition T2).
Idle_1	Waiting for next ISDU transmission
ISDURequest_2	Reception of ISDU request. ISDU data transmission uses the mechanism of segmented data transfer “Transmission of Segmented Data (PD, EV, ISDU)”**)**. For Retry-Handling see 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU).
ISDUWait_3	Waiting for data from application layer to transmit (see DL_ISDUTransport)
ISDUWaitParam_4	Waiting for data from system management to transmit (see DL_Read)
ISDUResponse_5	Transmission of ISDU response data via Message handler. ISDU data transmission uses the mechanism of segmented data transfer “Transmission of Segmented Data (PD, EV, ISDU)”**)**. For Retry-Handling see 7.7.2 “Retry-Handling of segmented Data (PD, EV, ISDU).

3140

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by the W-Device DL-mode handler.*
T2	1	2	*Start receiving of ISDU request data.*
T3	2	2	*Receive ISDU request data.*
T4	2	3	Invoke DL_ISDUTransport.ind to AL if last segment (EOS without data, see clause 7.1.9) has been received
T5	3	0	*Deactivation by the W-Device DL-mode handler.*
T6	3	5	*Response from AL.*
T7	5	5	*Message handler requests ISDU response.* Invoke ISDUMsg.rsp(SendWMessage = YES, Data, Length, FlowCtrl) to deliver ISDU response data to Message handler.
T8	5	1	-
T9	2	1	Invoke DL_ISDUAbort
T10	3	1	Invoke DL_ISDUAbort
T11	5	1	Invoke DL_ISDUAbort
T12	1	0	Deactivation by the W-Device DL-mode handler
T13	2	1	Signal ISDU error
T14	2	0	Deactivation by the W-Device DL-mode handler
T15	5	0	Deactivation by the W-Device DL-mode handler
T16	2	1	Invoke DL_Write.ind to SM if last segment (EOS without data, see clause 7.1.9) has been received, see clause 7.1.13.
T17	2	4	Invoke DL_Read.ind to SM if last segment (EOS without data, see clause 7.1.9) has been received, see 7.1.12.
T18	4	0	*Deactivation by the W-Device DL-mode handler.*
T19	4	5	*Response from SM.*

3141

INTERNAL ITEMS	TYPE	DEFINITION
ISDUStart	Service	ISDUMsg.ind(Data, Length, Start)
ISDUWrite	Service	ISDUMsg.ind(Data, Length, FlowCtrl)
ISDUReceiveComplete	Guard	If ISDUMsg.ind(EOS) received
ISDURespStart	Service	DL_ ISDUTransport.rsp(ValueList)
ISDUSendComplete	Guard	If ISDUMsg.rsp(EOS) sent and acknowledged
ISDUAbort	Service	ISDUMsg.ind(Abort)
ISDUError	Guard	If ISDU structure is incorrect
ISDUType	Guard	This variable shall be set to the following values: WRITE (ISDU write with index from Table 171) READ (ISDU read with index from Table 171) ISDU (ISDU read or write and no index from Table 171)

7.4.3 General structure and encoding of ISDUs

The encoding of ISDU data (I-Service-octet and data) delivered by the ISDU handler shall be implemented equal to IO-Link (see A.5 in REF 1.), with the exception of the definition of the nibble "I-Service". This specification shall only support the I-Service Read Request or Write Request with 16-bit Index and Subindex, as defined in Table 70.

Table 70 Definition of the nibble "I-Service"

I-Service (binary)	Definition		Index format
	W-Master	W-Device	
0000	Reserved	Reserved	
0001	Reserved	Reserved	
0010	Reserved	Reserved	
0011	Write Request	Reserved	16-bit Index and Subindex
0100	Reserved	Write Response (-)	none
0101	Reserved	Write Response (+)	none
0110	Reserved	Reserved	
0111	Reserved	Reserved	
1000	Reserved	Reserved	
1001	Reserved	Reserved	
1010	Reserved	Reserved	
1011	Read Request	Reserved	16-bit Index and Subindex
1100	Reserved	Read Response (-)	none
1101	Reserved	Read Response (+)	none
1110	Reserved	Reserved	
1111	Reserved	Reserved	

7.5 Command handler

The Command handler translates change requests for W-Device mode from W-Master´s system management into corresponding MasterCommands.

7.5.1 State machine of the W-Master command handler

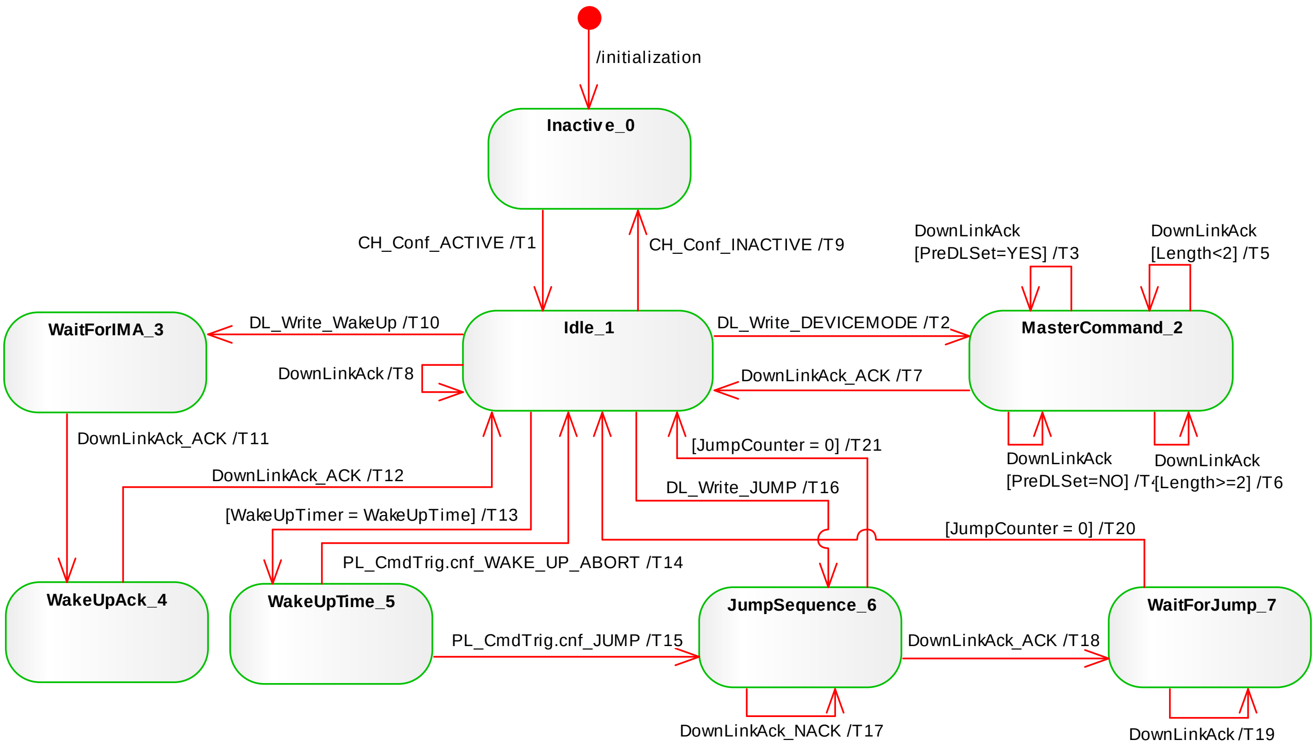


Figure 74 State machine of the W-Master command handler

Table 71 State transition tables of the W-Master command handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by W-Master DL-mode handler through CH_Conf_ACTIVE (see Table 62 DL-mode handler).
Idle_1	Waiting for new command from SM: DL_SetMode (change W-Device mode, for example to OPERATE), or waiting on DownLinkAck service primitive.
MasterCommand_2	Prepare data for MCmd.req service primitive. Waiting for demand from DownLinkAck service
WaitForIMA_3	Waiting for low energy W-Device to wake up and transmit an IMA message
WakeUpAck_4	Waiting for Ack on WakeUp MasterCommand
WakeUpTime_5	Check if low energy W-Device is awake
JumpSequence_6	Jump sequence started, waiting for an acknowledgement
WaitForJump_7	Countdown until end of Jump sequence

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by DL-mode handler.*
T2	1	1	The service DL_Write(DEVICEMODE) translates into: INACTIVE: MCmd.req (MasterCommand = 0x5C) STARTUP: MCmd.req (MasterCommand = 0x97) PREOPERATE: MCmd.req (MasterCommand = 0x9A) OPERATE: MCmd.req (MasterCommand = 0x99) For further MasterCommand definitions see Table 165.
T3	2	2	PreDownLink already in use, invoke MCmd.req(SendWMessage=NO).
T4	2	2	Invoke MCmd.req(SendWMessage=YES, Slot_N, MasterCommand, PreDLink) to send MasterCommand in PreDownLink.
T5	2	2	Not enough space left in the FullDownLink, invoke MCmd.req(SendWMessage=NO).
T6	2	2	Invoke MCmd.req(SendWMessage=YES, Slot_N, MasterCommand, FullDLink) to send MasterCommand in FullDownLink.
T7	2	1	Invoke MCmd.req(SendWMessage=NO) Invoke a positive DL_Write confirmation in case of: MasterCommand = 0x5C (Inactive) MasterCommand = 0x5F (UnParing) MasterCommand = 0x96 (DeviceIdent) MasterCommand = 0x97 (DeviceStartup) MasterCommand = 0x99 (DeviceOperate) MasterCommand = 0x9A (DevicePreoperate)
T8	1	1	No MasterCommand to send, invoke MCmd.req(SendWMessage=NO).
T9	1	0	*Deactivation by DL-mode handler.*
T10	1	3	-
T11	3	4	Invoke MCmd.req(SendWMessage=YES, Broadcast=NO, WakeUp, PreDLink)).
T12	4	1	Set WakeUpTimer = 0 Invoke PL_CmdTrig.req(WAKE_UP_TIME)
T13	1	5	if low energy W-Device is awake Invoke PL_CmdTrig.req(W_DEVICE_AWAKE) else invoke PL_CmdTrig.req(W_DEVICE_NOT_AWAKE)
T14	5	1	A low energy W-Device did not wake up, update is aborted.
T15	5	6	JumpCounter = MaxJump Invoke MCmd.req(SendWMessage=YES, Broadcast=YES, JUMP, PreDLink)
T16	1	6	JumpCounter = MaxJump Invoke MCmd.req(SendWMessage=YES, Broadcast=YES, JUMP, PreDLink)
T17	6	6	W-Device did not Acknowledge, decrease JumpCounter by 1. Invoke MCmd.req(SendWMessage=YES, Broadcast=YES, JUMP, PreDLink)
T18	6	7	Decrease JumpCounter by 1
T19	7	7	Decrease JumpCounter by 1
T20	7	1	Invoke PL_CmdTrig.req(JUMP)
T21	6	1	Invoke PL_CmdTrig.req(JUMP_FAIL)

INTERNAL ITEMS	TYPE	DEFINITION
DEVICEMODE	Label	Any of the MasterCommand definitions: INACTIVE, STARTUP, PREOPERATE or OPERATE For wireless, additional MasterCommand definitions are available (see Table 165.Mastercommand)
WakeUpTimer	Variable	This variable is a counter to WakeUpTime
WakeUpTime	Variable	This variable is the WakeUpTime of the low energy W-Device
JumpCounter	Variable	This variable is a countdown for switching to new hopping table
MaxJump	constant	Max number of jump retries, MaxJump = 0xE

7.5.2 State machine of the W-Device command handler

Figure 75 shows the W-Device state machine of the Command handler. It is driven by MasterCommands from the Master's Command handler to control the W-Device modes.

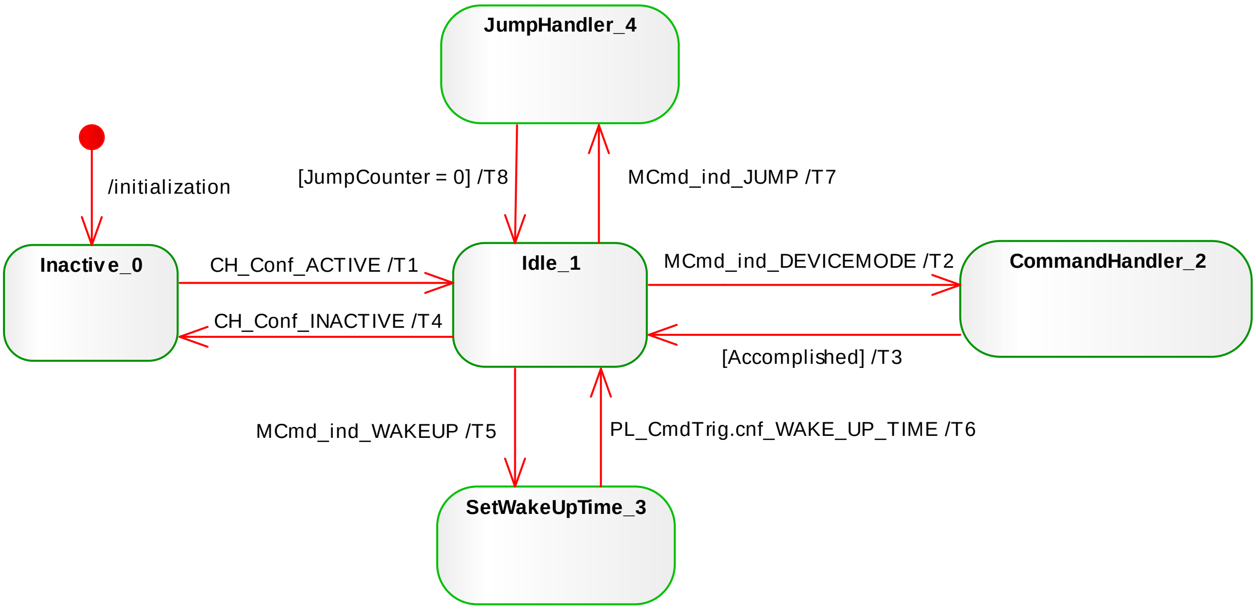


Figure 75 State machine of the W-Device command handler

Table 72 State transition tables of the W-Device Cmd handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
Idle_1	Waiting for next MasterCommand
Command_Handler_2	Decompose MasterCommand and invoke specific actions: If MasterCommand = 0x5C then change W-Device state to INACTIVE. If MasterCommand = 0x97 then change W-Device state to STARTUP. If MasterCommand = 0x9A then change W-Device state to PREOPERATE. If MasterCommand = 0x99 then change W-Device state to OPERATE. For the complete MasterCommand list see Table 165
SetWakeUpTime_3	Wait for WakeUpTime to be set in SM
JumpHandler_4	Countdown until end of Jump sequence

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by the W-Device DL-mode handler through CH_Conf_ACTIVE.*
T2	1	2	MasterCommand received Invoke DL_Write.ind(0x0000, 0x01, MCmd)
T3	2	1	Changing of W-Device State is accomplished
T4	1	0	*Deactivation by the W-Device DL-mode handler through CH_Conf_INACTIVE.*
T5	1	3	Set WakeUpTimer = 0, Increment by 1 every W-Sub-Cycle Invoke PL_CmdTrig.req(WAKE_UP_TIME)
T6	3	1	Set WakeUpTime value given by PL
T7	1	4	Set JumpCounter = last 4 bits of JUMP MCmd
T8	4	1	Invoke PL_CmdTrig.req(JUMP)

INTERNAL ITEMS	TYPE	DEFINITION
DEVICEMODE	Label	Any of the MasterCommand definitions: INACTIVE, STARTUP, PREOPERATE or OPERATE For wireless, additional MasterCommand definitions are available (see Table 165.Mastercommand)
WakeUpTimer	Variable	This variable is a counter to WakeUpTime
WakeUpTime	Variable	This variable is the WakeUpTime of the low energy W-Device
JumpCounter	Variable	This variable is a countdown for switching to new hopping table

7.6 Event handler

An Event transfers a status or an error information.
The Event request is sent from W-Device to Master. It is then processed by the Master and an Event Confirmation is sent back to the Device. Events are serviced one by one, so further Event requests are ignored until the current Event has been serviced and confirmed.
The general structure and coding of Events is specified in Annex A.6. in REF 1
EventCodes are specified in Table 180.

7.6.1 State machine of the W-Master Event handler

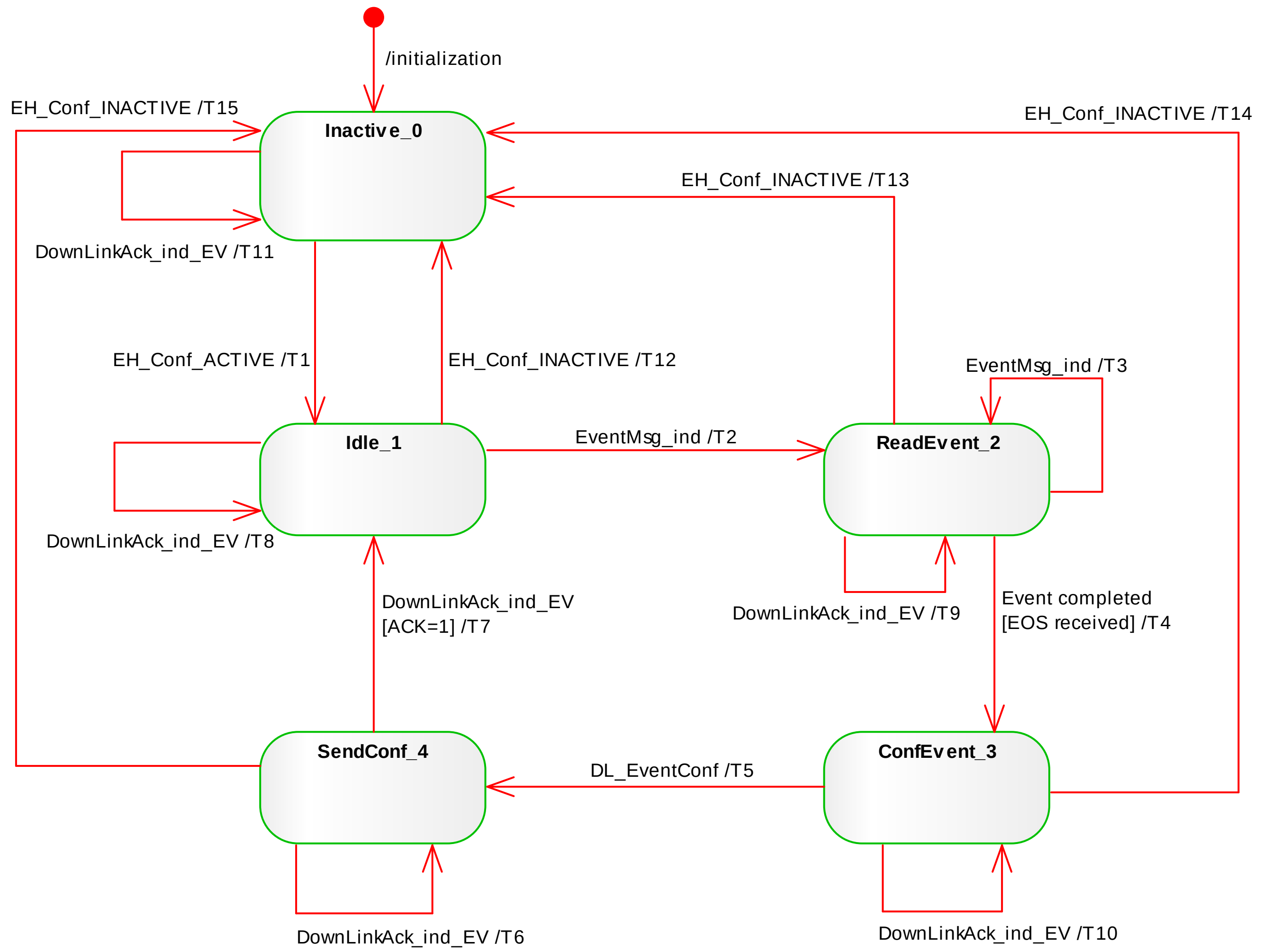


Figure 76 State machine of the W-Master Event handler

Table 73 State transition tables of the W-Master Event handler DL

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
Idle_1	Waiting for next Event indication
ReadEvent_2	Get Event data from W-Device through service Event indication. EV data transmission uses the mechanism of segmented data transfer (see 7.7.1 “Transmission of Segmented Data (PD, EV, ISDU)”). For Retry-Handling see 7.7.2
ConfEvent_3	Waiting for Event confirmation through service DL_Event.rsp from W-Master AL.
SendConf_4	Wait for DownLinkAck_ind_EV For Retry-Handling see 7.7.2

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	-
T2	1	2	Get Event StatusCode octet from service EventMsg.ind
T3	2	2	Get segmented data from EventMsg.ind
T4	2	3	After last segment (no data and EOS) has been received (see clause 7.7.1 Transmission of segmented data) invoke DL_Event indication to Master AL

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T5	3	4	-
T6	4	4	If enough space left in the downlink, invoke EventMsg.rsp(SendWMessage = YES, Slot_N) to deliver Event confirmation to Message handler
T7	4	1	-
T8	1	1	No Event confirmation to send, invoke EventMsg.rsp(SendWMessage = NO).
T9	2	2	No Event confirmation to send, invoke EventMsg.rsp(SendWMessage = NO).
T10	3	3	No Event confirmation to send, invoke EventMsg.rsp(SendWMessage = NO).
T11	0	0	No Event confirmation to send, invoke EventMsg.rsp(SendWMessage = NO).
T12	1	0	-
T13	2	0	-
T14	3	0	-
T15	4	0	-

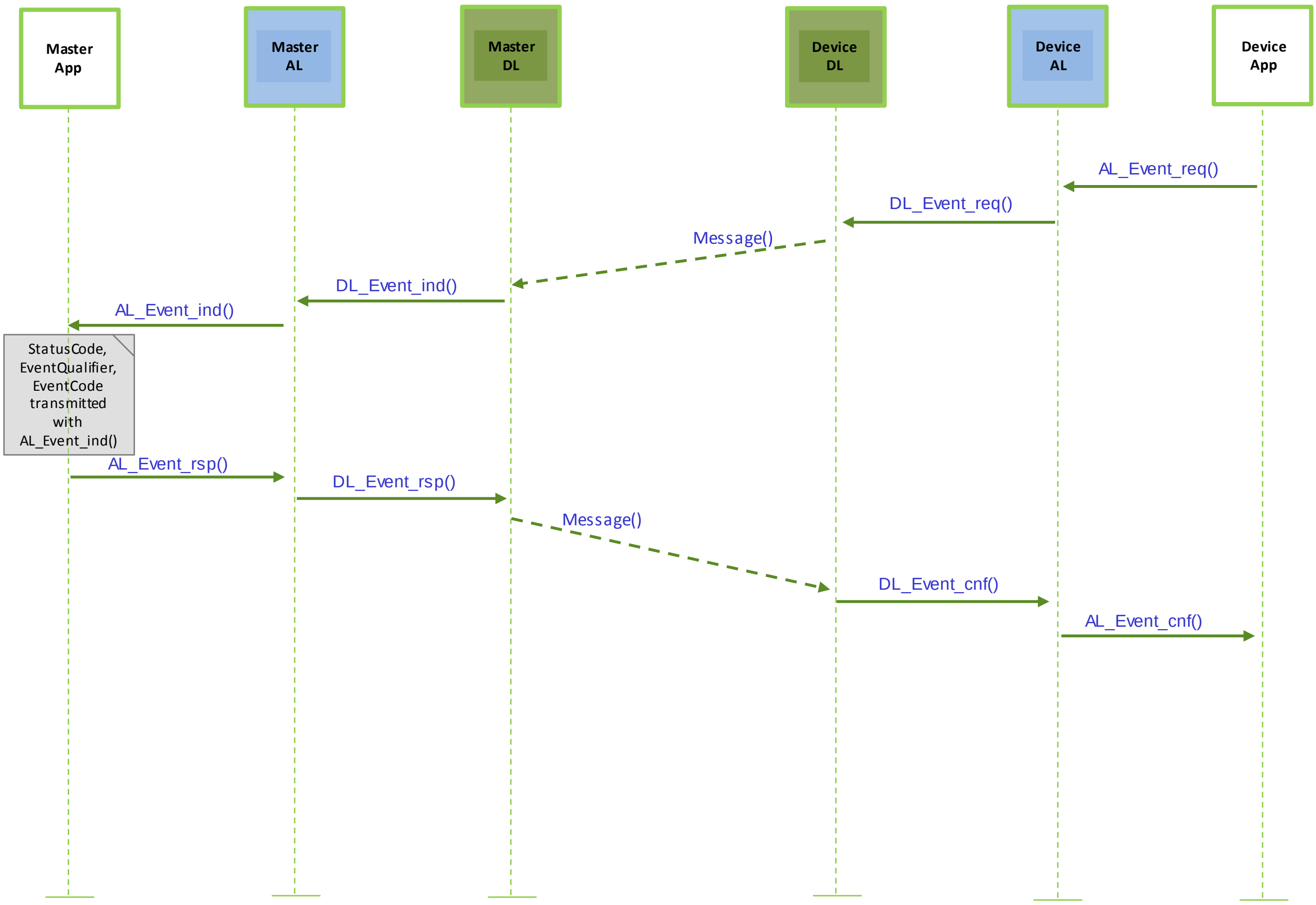


Figure 77 Sequence diagram: Single event scheduling

7.6.2 State machine of the W-Device Event handler

Figure 78 shows the state machine of the W-Device Event handler.

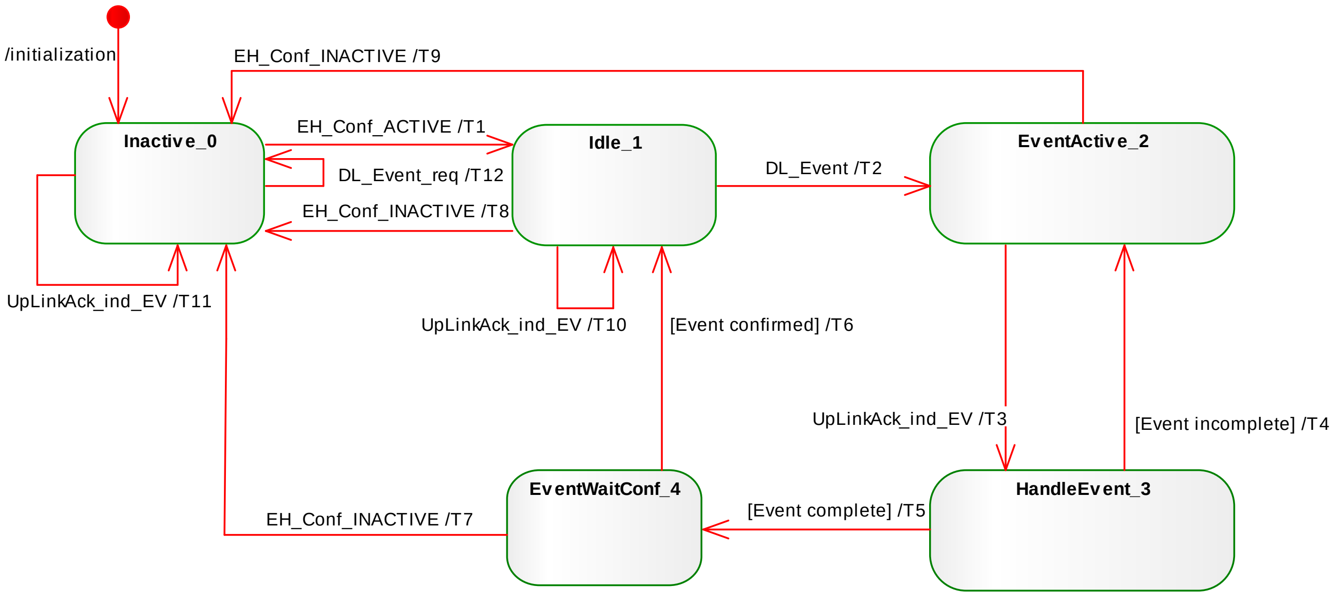


Figure 78 State machine of the W-Device Event handler

Table 74 State transition tables of the W-Device Event handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the W-Device DL-mode handler through EH_Conf_ACTIVE (see Table 63, Transition T1).
Idle_1	Waiting for Event indicated by DL_Event from application.
EventActive_2	Handler active and waiting for UpLinkAck_ind_EV.
HandleEvent_3	Handle EV data. EV data transmission uses the mechanism of segmented data transfer and retry handling, see clause 7.7.
EventWaitConf_4	Waiting for Event confirmation received from W-Master.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by the W-Device DL-mode handler through EH_Conf_ACTIVE.*
T2	1	2	Service DL_Event from AL indicates the occurrence of an Event.
T3	2	3	Message handler requests EV-Data through UpLinkAck_ind_EV. Invoke EventMsg.req(SendWMessage = YES, Data, Length, FlowCtrl) to deliver Event Data to Message handler.
T4	3	2	-
T5	3	4	Last EV transmission is complete (EOS without data) and acknowledged by W-Master see 7.7.1.
T6	4	1	*Event confirmation received from W-Master.* Invoke DL_Event.cnf
T7	4	0	*Deactivation by the W-Device DL-mode handler through EH_Conf_INACTIVE.* Invoke DL_Event.cnf(ErrorInfo = NO_COMM)
T8	1	0	*Deactivation by the W-Device DL-mode handler through EH_Conf_INACTIVE.*
T9	2	0	*Deactivation by the W-Device DL-mode handler through EH_Conf_INACTIVE.* Invoke DL_Event.cnf(ErrorInfo = NO_COMM)
T10	1	1	No Event to send, invoke EventMsg.req(SendWMessage = NO)
T11	0	0	No Event to send, invoke EventMsg.req(SendWMessage = NO)
T12	0	0	Invoke DL_Event.cnf(ErrorInfo = STATE_CONFLICT)

INTERNAL ITEMS	TYPE	DEFINITION
Event complete	Service	EOS without data received and acknowledged

7.7 Transmission of Segmented Data and retry handling

Data which can‘t be sent in one message have to be transmitted within a number of segments. To achieve a proper mechanism particularly in combination with possible retransmits, each DL-B handler (Process Data handler, ISDU handler and Event handler) must generate its own Flow Control considering the acknowledge of the last sent W-Message.

7.7.1 Transmission of segmented Data

The transmission of segmented data is possible for Process Data (e.g. for distribution of process data within a W-Cycle, see Figure 68, Events and ISDU Data.
The ULink and DLink Control Octets accommodates a counter (=FlowControl). FlowControl is controlling the segmented data flow by counting the sequences necessary to transmit segmented data (see Table 75).

- A segment begins with FlowControl = START.
- All following segments use FlowControl = COUNT to number each data segment. In case of a retry during COUNT, take account to 7.7.2.
- The transmission of the last segment differs between Process Data and Event- or ISDU-data:
  a. Last segment for Process Data Out (transmitted via DLink):
     To indicate a complete data transmission to W-Device set FlowControl = EOS immediately.
  b. Last segment for Process Data In (transmitted via ULink):
     To indicate a complete data transmission to W-Master set FlowControl = PDataLength (see Table 75 column Process Data In)
  c. Last segment for acyclic Event- and ISDU-data (transmitted via DLink or ULink):
     To indicate a complete data transmission, the Event handler or ISDU-handler shall send a

separate W-Message with FlowControl = EOS and without data to achieve data consistency due to retransmits.

Note:
A MasterCommand as well as an Event acknowledge doesn't need segmentation, since this W-Messages are transmitted without data (see Table 42).

Table 75 Flow Control for segmented data

FlowControl (FC)			
FlowControl (FC)	Definition		
0x00 to 0x07	COUNT Counter within a data segment. Increments beginning with 1 after an START. Jumps back from 7 to 0 in case of an overflow.		
0x08	START Start of a segment, i.e., start of a ISDU- request or a response. For the start of a request, any previously incomplete service may be rejected. For a start request associated with a response, a W-Device shall send "No Service" until its application returns response data		
0x09	EOS End of Segment indicates a completed transmission. (Event- or ISDU: separate EOS within next DLink)		
0x0A	ABORT Abort entire service. The W-Master responds by rejecting received response data. The W-Device responds by rejecting received request data and may generate an abort.		
	Definition for DLink Control Octet See Note 1	Definition for ULink Control Octet See Note 2	
		Process Data In (data transmission complete)	Event- or ISDU data (separate EOS within next ULink)
0x0B	Unused	PDataLength = 1 octet	DataLength = 1 octet
0x0C…0x17	Unused	PDataLength = 2…13 octet	DataLength = 2…13 octet
0x18	Unused	PDataLength = 14 octet	DataLength = 14 octet
0x19 to 0x1F	Reserved	Reserved	Reserved

Note 1:
The DLink Control Octet (see Figure 134) contains a separate field to transmit the length of data. Therefore, these values are unused.

Note 2:
The ULink Control Octet (see Figure 136) is coded by only one octet (reduced overhead). Therefore, the DataLength is coded within the Flow Control.

Additionally see 12.6. Example for DLink data transmission and 12.7 Examples for ULink data transmission for data transmission examples.

7.7.2 Retry-Handling

For an appropriate data transmission, the "Sender" shall retransmit its last W-Message, if the service DownLinkAck or UpLinkAck delivered a negative Acknowledge (NACK or not received message) to the corresponding handler (ProcessData-, Event-, Command-, or ISDU-handler) see 5.5.2.8.

7.7.2.1 Retry handling in case of not Segmented data

The corresponding handler shall retransmit its last W-Message, depending on the remaining payload in the DLink or ULink (see service 6.3.4 DownLinkAck and 6.3.5 UpLinkAck)

7.7.2.2 Retry handling in case of Segmented data / Flow Control

If the “Sender” does not receive an ACK for its last sent W-Message, it has to forward NACK to the layer above and it has to resend the last data and the value of the last FlowCtrl.
If the “Receiver” thereupon gets new FlowCtrl = last FlowCtrl, it has to reject the last received data segment and use the new received data segment instead. This behavior is essentially, since a W-Message with a ACK could be corrupted (e.g. on air), which leads in a NACK on the receiver side.

Note: “Sender” or “Receiver” can be W-Master or W-Device

8. Application Layer (AL)

8.1 General

Figure 79 shows an overview of the structure and services of the W-Master application layer(AL)

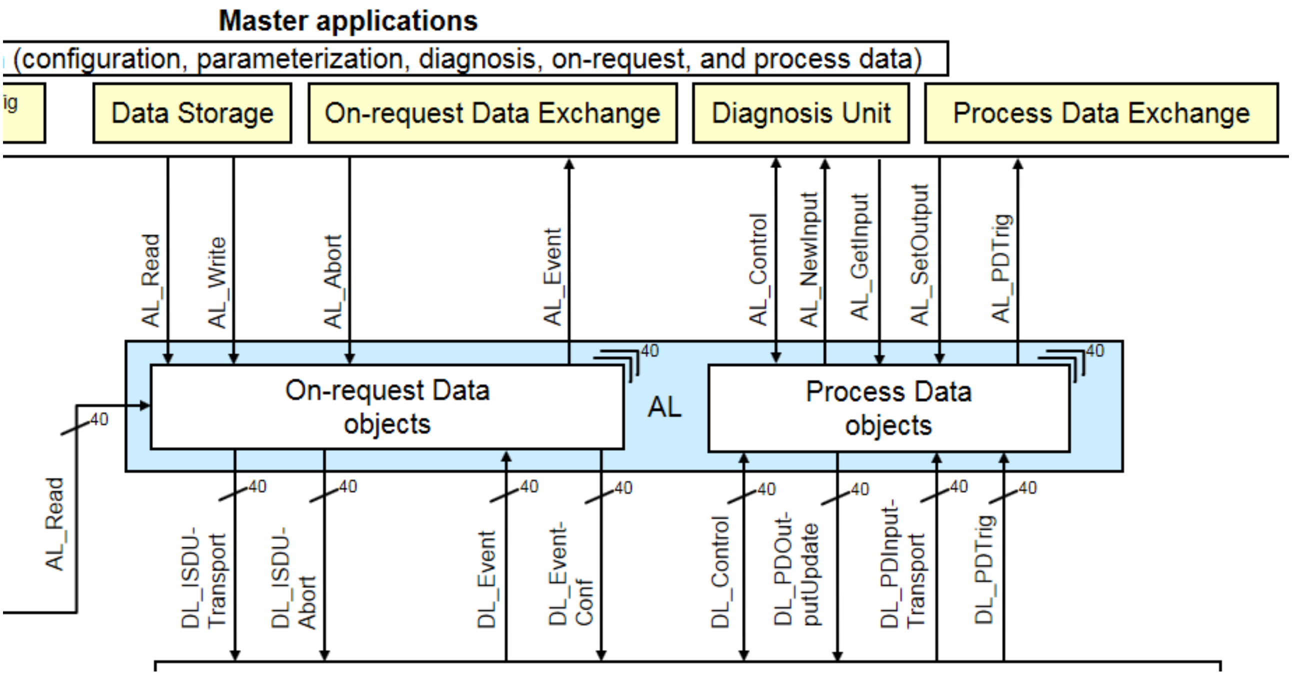


Figure 79 Structure and services of the application layer (W-Master)

Figure 80 shows an overview of the structure and services of the W-Device application layer (AL).

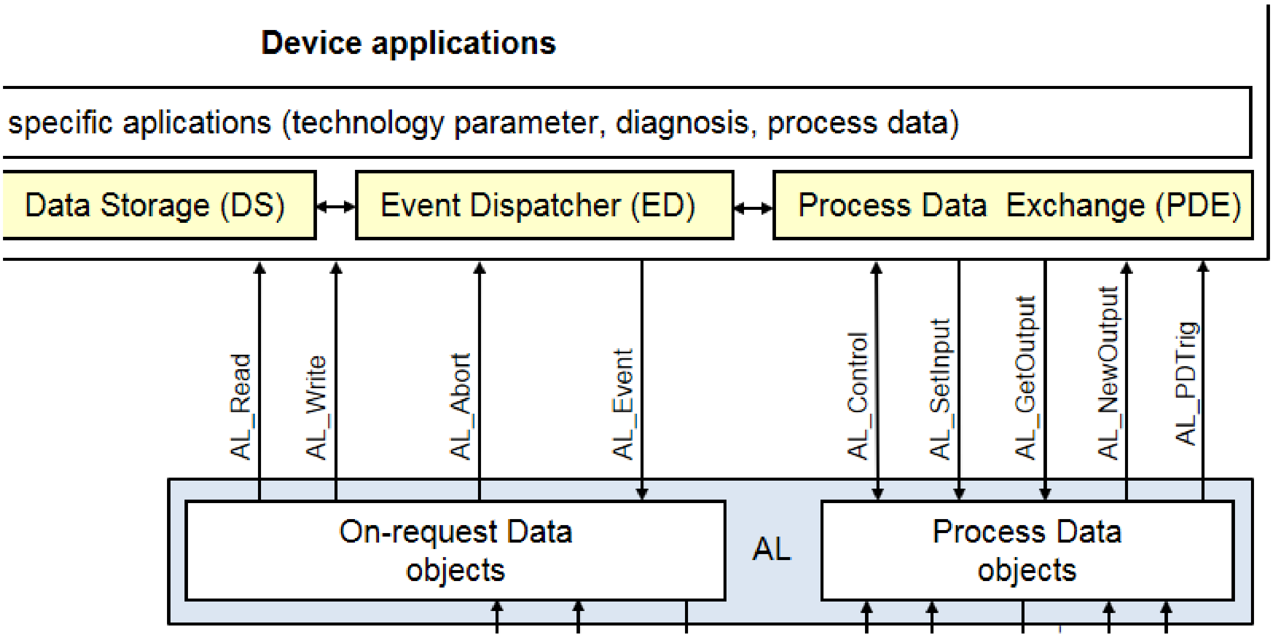


Figure 80 Structure and services of the application layer (W-Device)

8.2 Application Layer services

This clause defines the services of the application layer (AL) to be provided to the W-Master and W-Device applications and system management via its external interfaces. Table 76 lists the assignments of W-Master and W-Device to their roles as initiator or receiver for the individual AL services. Empty fields indicate no availability of this service on W-Master or W-Device.

Table 76 AL services within W-Master and W-Device

Service name	W-Master	Device
AL_Read	R	I
AL_Write	R	I
AL_Abort	R	I
AL_NewInput	I	
AL_GetInput	R	
AL_SetInput		R
AL_PDTrig	I	I
AL_GetOutput		R
AL_NewOutput		I
AL_SetOutput	R	
AL_Event	I	R
AL_Control	I, R	I, R
Key (see 3.3.5) All services are defined from the view of the affected layer towards the layer above. - I Initiator of a service (towards the layer above) - R Receiver (responder) of a service (from the layer above)		

8.2.1 AL_Read (W-Master and W-Device)

The AL_Read service is used to read ISDU Data from a IO-Link Wireless W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 77.

Table 77 AL_Read

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
W-Port	M			
Index	M	M		
Subindex	M	M		
Result (+)			S	S(=)
W-Port				M
Data			M	M(=)
Result (-)			S	S(=)
W-Port				M
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the W-Port number for the ISDU Data to be read.

Index
This parameter indicates the address of the ISDU Data objects to be read from the W-Device. Index 0 in conjunction with Subindex 0 addresses the entire set of Direct Parameters in Page 1 or in conjunction with Subindex 1 to 16 the individual parameters.

Index 1 in conjunction with Subindex 0 addresses the entire set of Direct Parameters in Page 2 or in conjunction with Subindex 1 to 16 the individual parameters from address 16 to 31 (W-Bridge). Subindex 0 in conjunction with the wireless parameter page addresses the entire set of Wireless Parameters (see Extended Parameter Pages for IO-Link Wireless in Table 168).
Permitted values: 0 to 65535 (Figure 155 parameter via ISDU)

Subindex
This parameter indicates the element number within a structured ISDU Data object. A value of 0 indicates the entire set of elements.
Permitted values: 0 to 255

Result (+):
This selection parameter indicates that the service has been executed successfully.

W-Port
This parameter contains the W-Port number of the requested ISDU Data.

Data
This parameter contains the read values of the ISDU Data.
Parameter type: Octet string

Result (-):
This selection parameter indicates that the service failed.

W-Port
This parameter contains the W-Port number of the requested ISDU Data.

ErrorInfo
This parameter contains the error information.
Permitted values: see Clause 13.8, see Annex C in REF 1
NOTE: The AL maps DL ErrorInfos into its own AL ErrorInfos using Annex C in REF 1

8.2.2 AL_Write (W-Master and W-Device)

The AL_Write service is used to write ISDU Data to a IO-Link Wireless W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 78.

Table 78 AL_Write

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M		
W-Port	M			
Index	M	M		
Subindex	M	M		
Data	M	M(=)		
Result (+)			S	S(=)
W-Port				M
Result (-)			S	S(=)
W-Port				M
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the W-Port number for the ISDU Data to be written.

Index
This parameter indicates the address of the ISDU Data objects to be written to the W-Device. Indexes from Table 171 always return a negative result.
Index 1 in conjunction with Subindex 0 addresses the entire set of Direct Parameters in Page 2 or in conjunction with Subindex 1 to 16 the individual parameters from address 16 to 31 (W-Bridge), it returns always a positive result.

Permitted values: 0 to 65535

Subindex

This parameter indicates the element number within a structured ISDU Data object. A value of 0 indicates the entire set of elements (only possible if all subindices have write access rights!).

Permitted values: 0 to 255

Data

This parameter contains the values of the ISDU Data.

Parameter type: Octet string

Result (+):

This selection parameter indicates that the service has been executed successfully.

W-Port

This parameter contains the W-Port number of the ISDU Data.

Result (-):

This selection parameter indicates that the service failed.

W-Port

This parameter contains the W-Port number of the ISDU Data.

ErrorInfo

This parameter contains the error information.

Permitted values: see clause 13.8, see IO-Link Interface and System Specification Annex C in REF 1

8.2.3 AL_Abort (W-Master and W-Device)

The AL_Abort service is used to abort a current AL_Read or AL_Write service on a specific W-Port. Invocation of this service abandons the response to an AL_Read or AL_Write service in progress on the W-Master. The parameters of the service primitives are listed in Table 79

Table 79 AL_Abort

Parameter Name	.req	.ind
Argument	M	M
W-Port	M	

Argument

The service-specific parameters are transmitted in the argument.

W-Port

This parameter contains the W-Port number of the service to be abandoned

8.2.4 AL_NewInput (W-Master)

The AL_NewInput local service indicates the receipt of updated input data within the Process Data of a W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 80.

Table 80 AL_NewInput

Parameter Name	.ind
Argument	M
W-Port	M

Argument

The service-specific parameters are transmitted in the argument.

W-Port

This parameter specifies the W-Port number of the received Process Data

8.2.5 AL_GetInput (W-Master)

The AL_GetInput service reads the input data within the Process Data provided by the data link layer of a W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 81

Table 81 AL_GetInput

Parameter Name	.req	.cnf
Argument	M	
W-Port	M	
Result (+)		S
W-Port		M
InputData		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter specifies the W-Port number of the received Process Data.

Result (+):
This selection parameter indicates that the service has been executed successfully.

W-Port
This parameter specifies the W-Port number of the received Process Data.

InputData:
This parameter contains the values of the requested process input data of the specified W-Port.
Parameter type: Octet string

Result (-):
This selection parameter indicates that the service failed.

W-Port
This parameter contains the W-Port number for the Process Data.

ErrorInfo
This parameter contains the error information. Permitted values: NO_DATA (DL did not provide Process Data)

8.2.6 AL_SetInput (W-Device)

The AL_SetInput local service updates the input data within the Process Data of a W-Device. The parameters of the service primitives are listed in Table 82.

Table 82 AL_SetInput

Parameter Name	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data values of the input data to be transmitted.

Parameter type: Octet string

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information.

Permitted values:

NO_COMM (no communication available),

STATE_CONFLICT (service unavailable within current state

8.2.7 AL_PDTrig (W-Master and W-Device)

The AL_PDTrig service indicates the end of a W-MasterCycleTime period after each start of Process Data reception. The W-Device application can use this service to achieve equidistant Process Data periods (see Note 1) by eliminating jitter due to retry handling.

The parameters of the service primitives are listed in Table 83.

Table 83 AL_PDTrig

Parameter Name	.ind
Argument	M
W-Port	C

Argument

The service-specific parameters are transmitted in the argument.

W-Port

This parameter contains the W-Port number of the received new Process Data (W-Master only).

Note 1: To minimize Jitter caused by different transmission qualities, especially with segmented data (variations on the numbers of retries) PDTrig can be used to get an equidistant time between reception of first data packet and activation of PDTrig.

8.2.8 AL_GetOutput (W-Device)

The AL_GetOutput service reads the output data within the Process Data provided by the data link layer of the W-Device. The parameters of the service primitives are listed in Table 84.

Table 84 AL_GetOutput

Parameter Name	.req	.cnf
Argument	M	
Result (+)		S
OutputData		M
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

Result (+):

This selection parameter indicates that the service has been executed successfully.

OutputData

This parameter contains the Process Data values of the requested output data.

Parameter type: Octet string

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information.

Permitted values:
NO_DATA (DL did not provide Process Data)

8.2.9 AL_NewOutput (W-Device)

The AL_NewOutput local service indicates the receipt of updated output data within the Process Data of a W-Device. This service has no parameters. The service primitives are shown in Table 85.

Table 85 AL_NewOutput

Parameter name	.ind
<None>	

8.2.10 AL_SetOutput (W-Master)

The AL_SetOutput local service updates the output data within the Process Data of a W-Master. The parameters of the service primitives are listed in Table 86.

Table 86 AL_SetOutput

Parameter Name	.req	.cnf
Argument	M	
W-Port	M	
OutputData	M	
Result (+)		S
W-Port		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
W-Port
This parameter contains the W-Port number of the Process Data to be written.
OutputData
This parameter contains the output data to be written at the specified W-Port.
Parameter type: Octet string
Result (+):
This selection parameter indicates that the service has been executed successfully.
W-Port
This parameter contains the W-Port number for the Process Data.
Result (-):
This selection parameter indicates that the service failed.
W-Port
This parameter contains the W-Port number for the Process Data.
ErrorInfo
This parameter contains the error information.
Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (Service unavailable within current state)

8.2.11 AL_Event (W-Master and W-Device)

The AL_Event service indicates one pending status or error message. The source of one Event can be local (W-Master) or remote (W-Device). The Event can be triggered by a communication layer or by an application. The parameters of the service primitives are listed in Table 87.

Table 87 AL_Event

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	M	M	
W-Port		M	M	
Instance	M	M		
Mode	M	M		
Type	M	M		
Origin		M		
EventCode	M	M		
Result (+)				S
Result (-)				S
ErrorInfo				M

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the W-Port number of the Event data. **Instance**
This parameter indicates the Event source. Permitted values: Unknown, Application (see Table 136, see Table A.17 in REF 1)

Mode
This parameter indicates the Event mode. Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table 139, see Table A.20 in REF 1)

Type
This parameter indicates the Event category. Permitted values: ERROR, WARNING, NOTIFICATION (see Table 180, see Table A.19 in REF 1)

Origin
This parameter indicates whether the Event was generated in the local communication section or remotely (in the W-Device). Permitted values: LOCAL, REMOTE

EventCode
This parameter contains a code identifying a certain Event. Permitted values: see Table 180, see Annex D in REF 1)

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains the error information.
Permitted values: STATE_CONFLICT (Service unavailable within current state), NO_COMM (no communication available)

8.2.12 AL_Control (W-Master and W-Device)

The AL_Control service contains the Process Data qualifier status information transmitted to and from the W-Device application. The parameters of the service primitives are listed in Table 88.

Table 88 AL_Control

Parameter Name	.req	.ind
Argument	M	M
W-Port	C	C
ControlCode	M	C
MaxRetry		C

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the number of the related W-Port.

ControlCode
This parameter contains the qualifier status of the Process Data (PD).
Permitted values:
PDIN_VALID (Input Process Data valid)
PDIN_INVALID (Input Process Data invalid)
PDOUT_VALID (Output Process Data valid, see Table 135).
PDOUT_INVALID (Output Process Data invalid, see Table 135).

MaxRetry (W-Device only)
This parameter contains information of a real-time fault.
Permitted Values:
YES (MaxRetry occurred)
NO (MaxRetry not occurred)

8.3 Application layer protocol

8.3.1 Overview

The application layer manages the data transfer with all its assigned W-Ports. That means, AL service calls need to identify the particular W-Port they are related to.

8.3.2 ISDU processing

8.3.2.1 ISDU state machine of the W-Master AL

Figure 81 shows the state machine for the handling of ISDU Data within the application layer. "AL_Service" represents any AL service in Table 76 related to ISDU. "W-Portx" indicates a particular wireless W-Port number

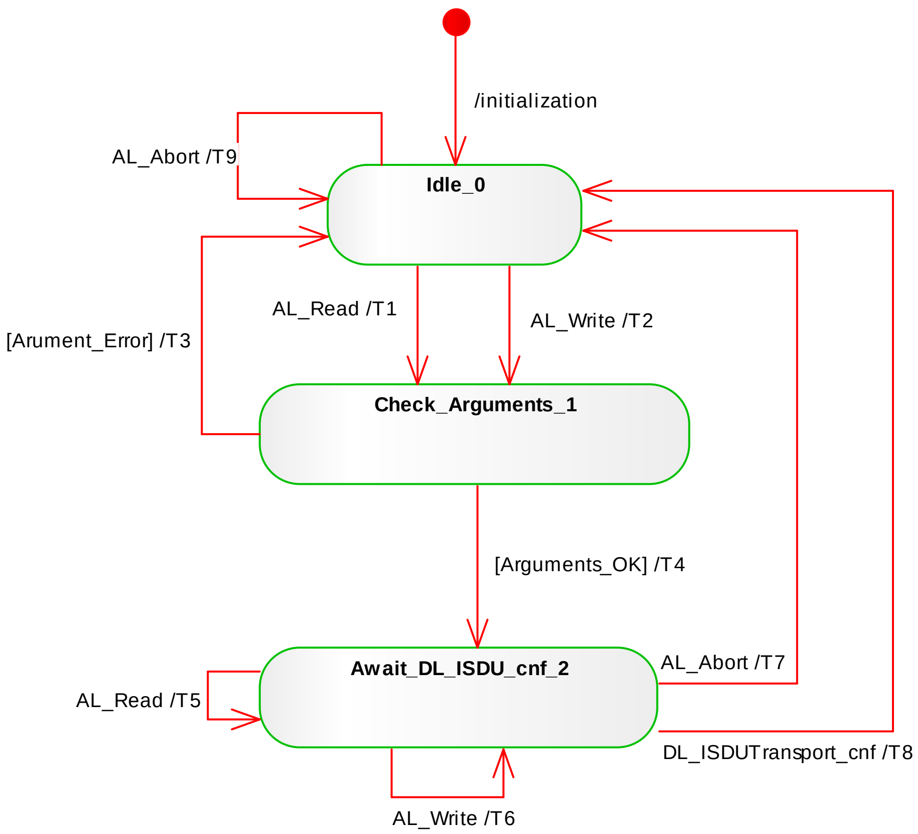


Figure 81 ISDU state machine of the W-Master AL

Table 89 State transition tables of the ISDU W-Master AL

STATE NAME	STATE DESCRIPTION
Idle_0	AL_Read or AL_Write invocations from the W-Master applications or from the W-Master Port handler (see Figure 79) can be accepted within this state.
Check_Arguments_1	Within this state the arguments of the AL_Read/AL_Write service call are checked (see 8.2.1, 8.2.2 for permitted values), and the internal items Argument_Error and Arguments_OK are set accordingly.
Await_DL_ISDU_cnf_2	The W-Master AL remains in this state until a DL_ISDUTransport.cnf is received.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	-
T2	0	1	-
T3	1	0	Invoke AL_Read.cnf/AL_Write.cnf with negative Result.
T4	1	2	Invoke DL_ISDUTransport.req.
T5	2	2	Invoke AL_Read.cnf with negative Result.
T6	2	2	Invoke AL_Write.cnf with negative Result.
T7	2	0	Invoke DL_ISDUAbort.req.
T8	2	0	Invoke AL_Read.cnf/AL_Write.cnf
T9	0	0	-

INTERNAL ITEMS	TYPE	DEFINITION
Argument_Error	Bool	Illegal values within the service body, for example "W-Port number or Index out of range"
Arguments_OK	Bool	No errors in the arguments.

8.3.2.2 ISDU state machine of the W-Device AL

Figure 82 shows the state machine for the handling of ISDU Data within the application layer of a W-Device.

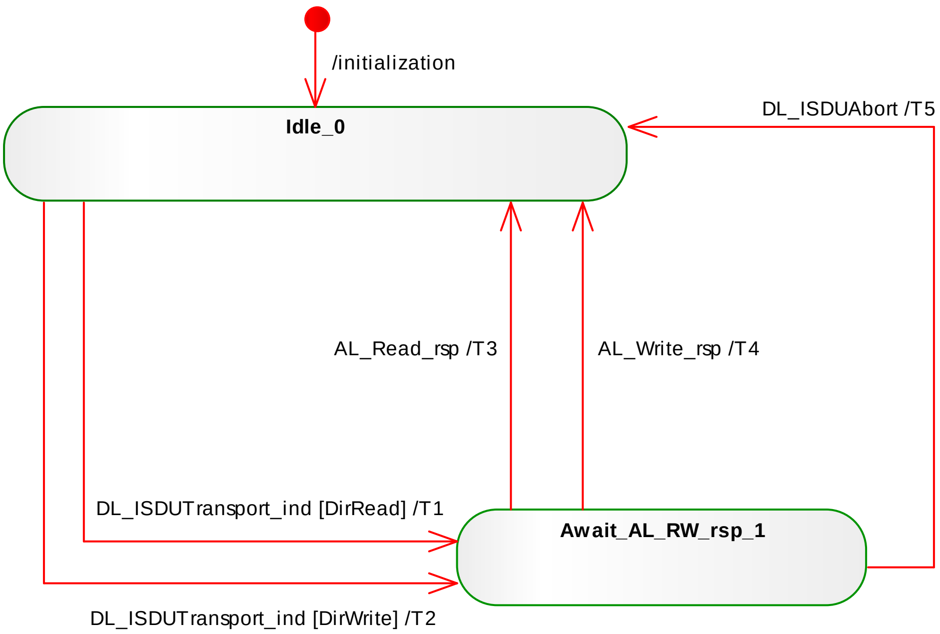


Figure 82 ISDU state machine of the W-Device AL

Table 90 State transition tables of the ISDU W-Device AL

STATE NAME	STATE DESCRIPTION
Idle_0	The W-Device AL is waiting on subordinated DL service calls triggered by W-Master messages.
Await_AL_RW_rsp_1	The W-Device AL is waiting on a response from the technology specific application (read or write access via ISDU).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke AL_Read.
T2	0	1	Invoke AL_Write.
T3	1	0	Invoke DL_ISDUTransport(read)
T4	1	0	Invoke DL_ISDUTransport(write)
T5	1	0	Current waiting on AL_Read or AL_Write abandoned.

INTERNAL ITEMS	TYPE	DEFINITION
DirRead	Bool	Access direction: DL_ISDUTransport(read) causes an AL_Read
DirWrite	Bool	Access direction: DL_ISDUTransport(write) causes an AL_Write

8.3.2.3 Sequence diagrams for ISDU Data

Figure 83 through Figure 86 demonstrate complete interactions between W-Master and W-Device for several ISDU Data exchange use cases.

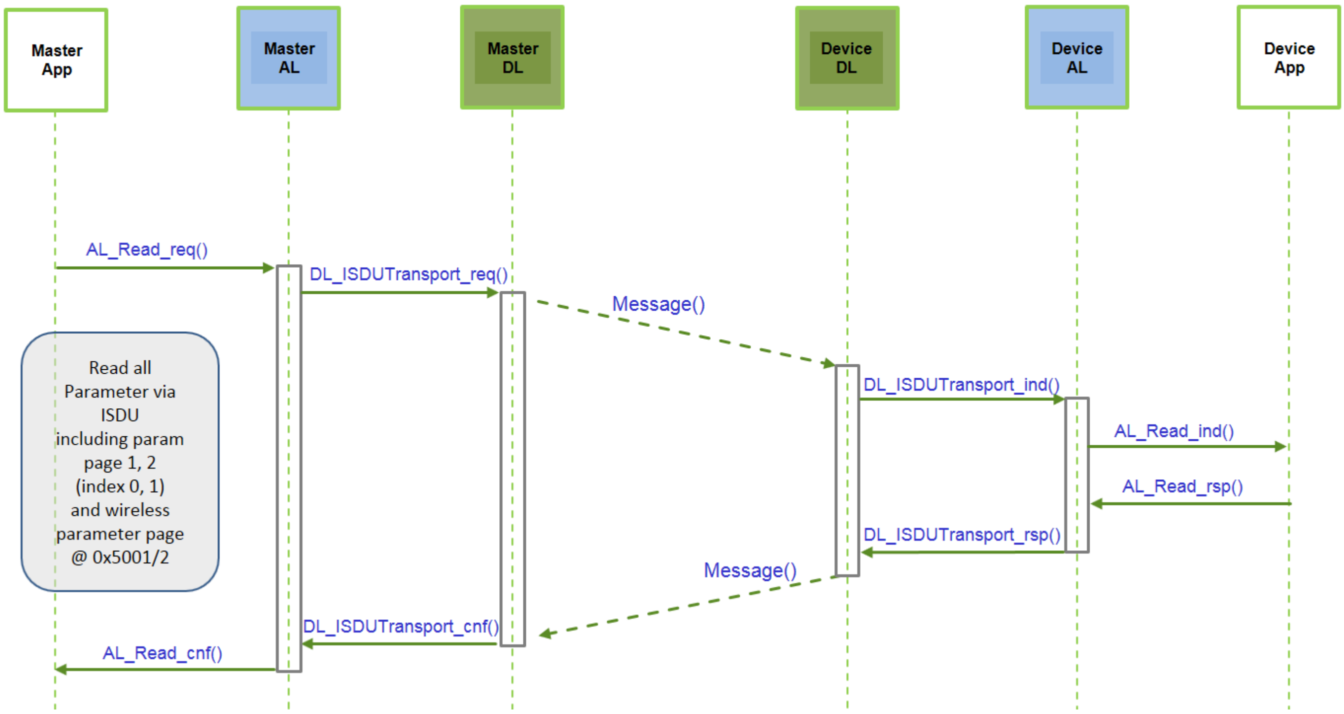


Figure 83 Sequence diagram: ISDU Read Data

Figure 84 demonstrates the behavior of ISDU Data exchange in case of a timeout (5 s). A W-Device shall respond within less than the "ISDU acknowledgement time" (see clause 10.7.5 in REF 1).

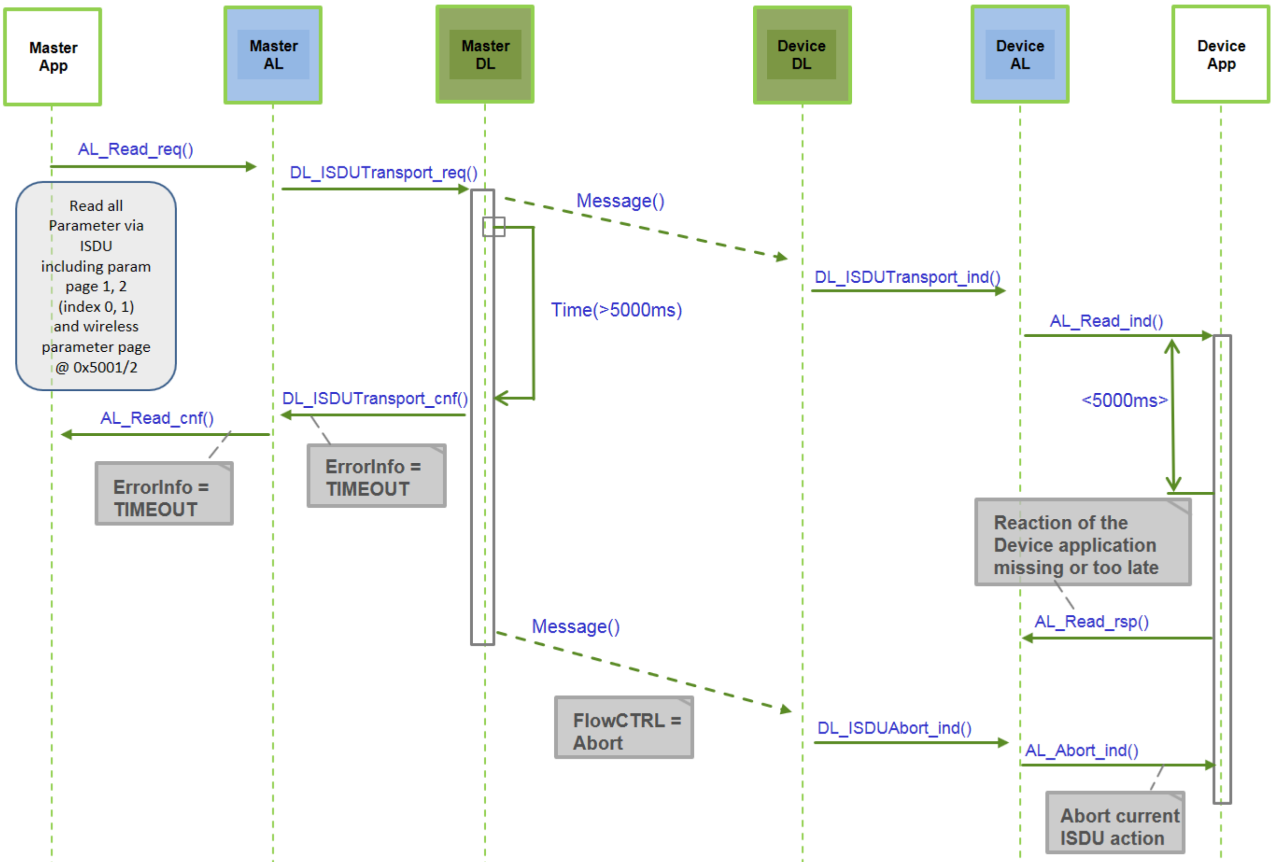


Figure 84 Sequence diagram: ISDU read Data in case of timeout

Figure 85 demonstrates the behavior of ISDU Data exchange in case of an error such as requested Index not available (see Table C.1 in REF 1).

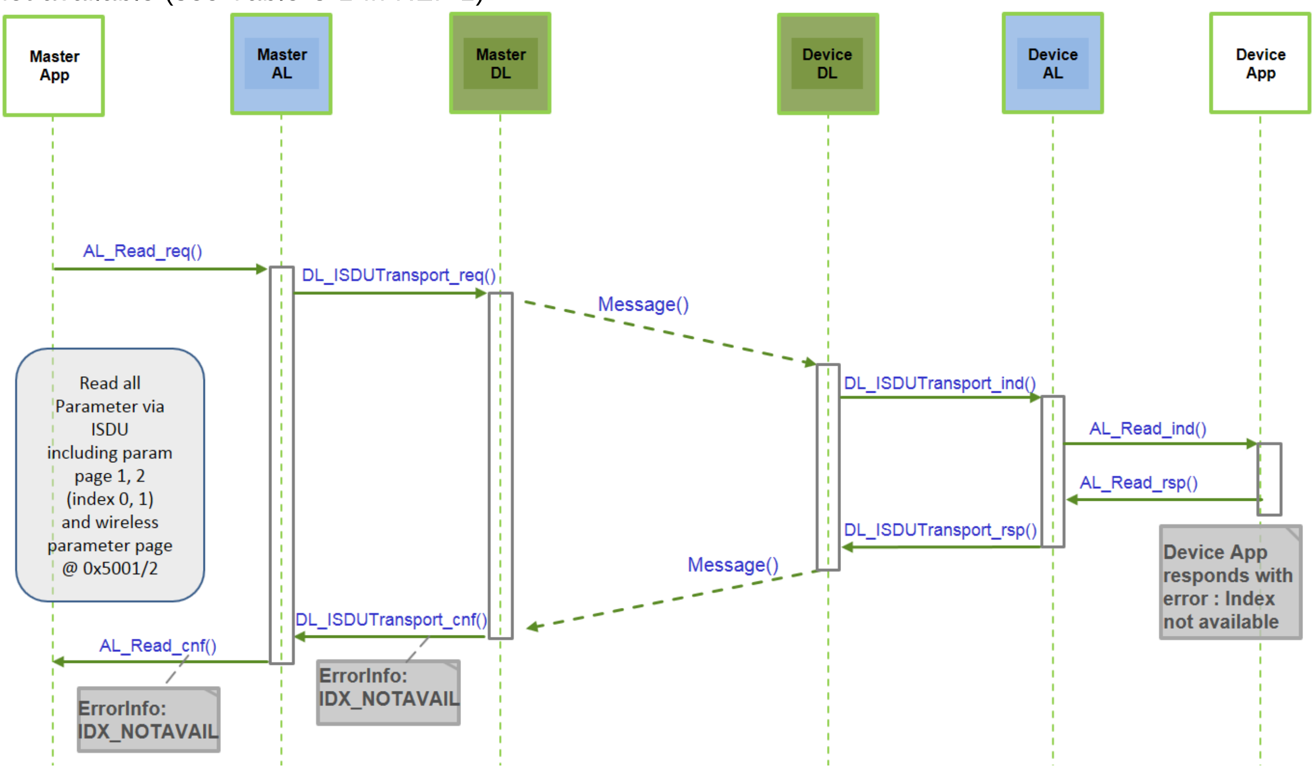


Figure 85 Sequence diagram: ISDU read Data in case of error

Figure 86 demonstrates the behavior of ISDU Data exchange in case of interaction with a low energy W-Device.

If a W-Port is paired with a low energy (LP) W-Device with LowPowerDevice attribute activated, the W-Master shall send a MasterCommand to switch the LP W-Device to listen to the Full-Downlink prior to the ISDU data transmission.
After ISDU data transmission, the W-Master shall send a MasterCommand to switch the LE W-Device back to Pre-Downlink.

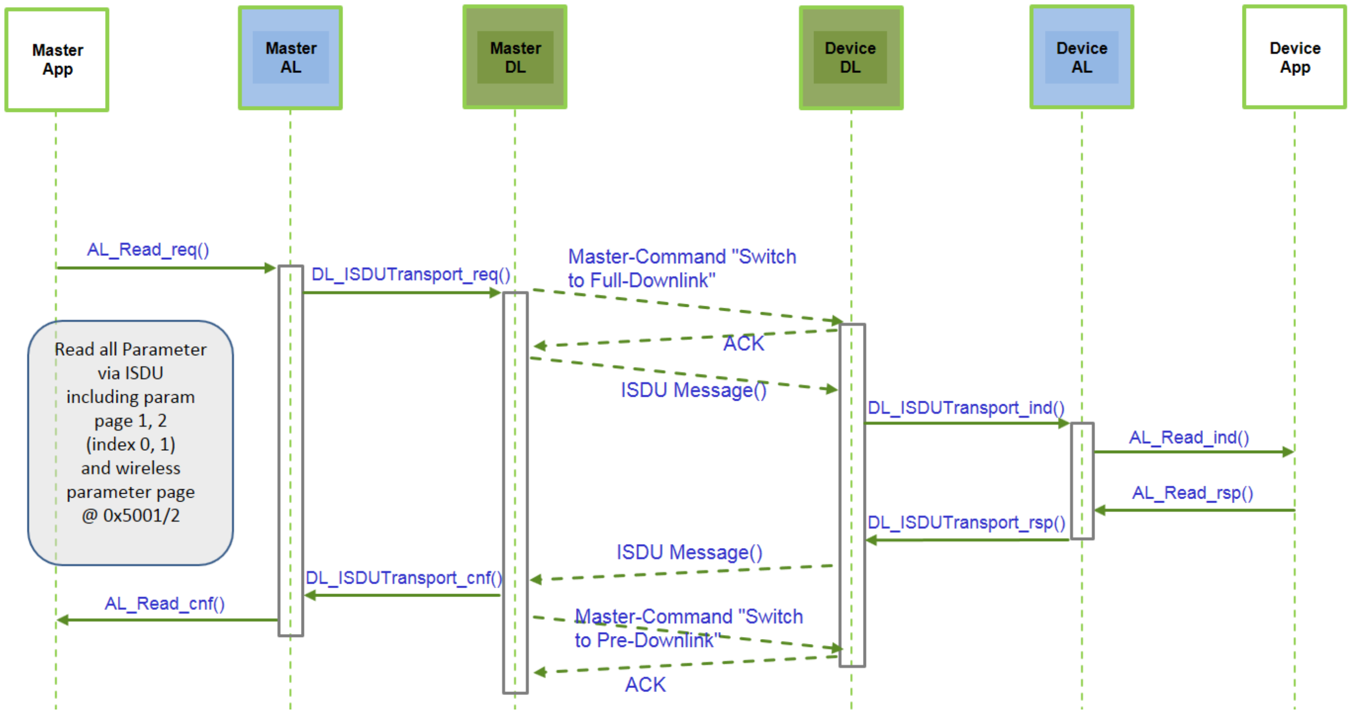


Figure 86 Sequence diagram for low energy W-Devices: ISDU Data

8.3.3 Event processing

8.3.3.1 Event state machine of the W-Master AL

Figure 87 shows the Event state machine of the W-Master application layer.

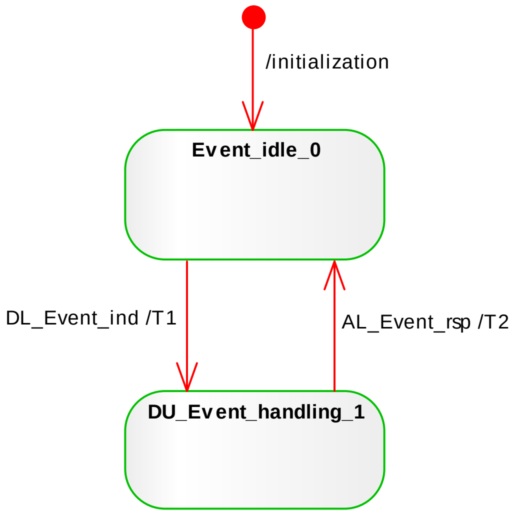


Figure 87 Event state machine of the W-Master AL

Table 91 State transition tables of the Event W-Master AL

STATE NAME	STATE DESCRIPTION
Event_idle_0	The W-Master AL is ready to accept DL_Events (diagnosis information) from the DL.
DU_Event_handling_1	Analyze Event data and invoke AL_Event.ind to Diagnosis Unit. The W-Master AL remains in this state as long as the Diagnosis Unit (see 11.5) did not acknowledge the AL_Event.ind.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	1	2	Invoke AL_Event.ind
T2	2	1	Invoke DL_Event.rsp

INTERNAL ITEMS	TYPE	DEFINITION
-		

8.3.3.2 Event state machine of the W-Device AL

Figure 88 shows the Event state machine of the W-Device application layer

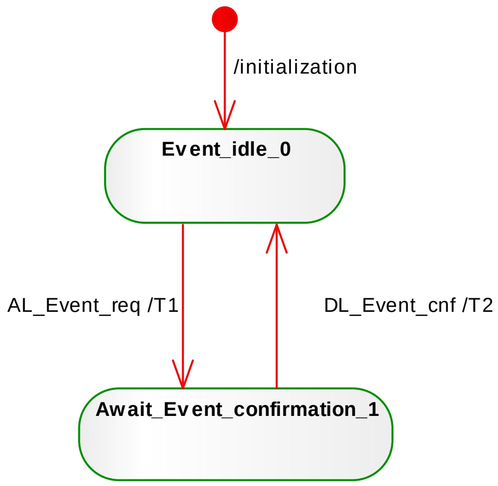


Figure 88 Event state machine of the W-Device AL

Table 92 State transition tables of the Event W-Device AL

STATE NAME	STATE DESCRIPTION
Event_Idle_0	The W-Device AL is ready to accept one AL_Event (diagnosis information) from the technology specific W-Device applications for the transfer to the DL. The W-Device applications can create one new Event during this time.
Await_Event_response_1	The W-Device AL propagated an AL_Event with diagnosis information and waits on a DL_Event confirmation of the DL. The W-Device AL shall not accept any new AL_Event during this time.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke DL_Event.req
T2	1	0	Invoke AL_Event.cnf

8.3.3.3 Single Event scheduling

Figure 93 shows how a single Event from a W-Device is processed, in accordance with the relevant state machines.

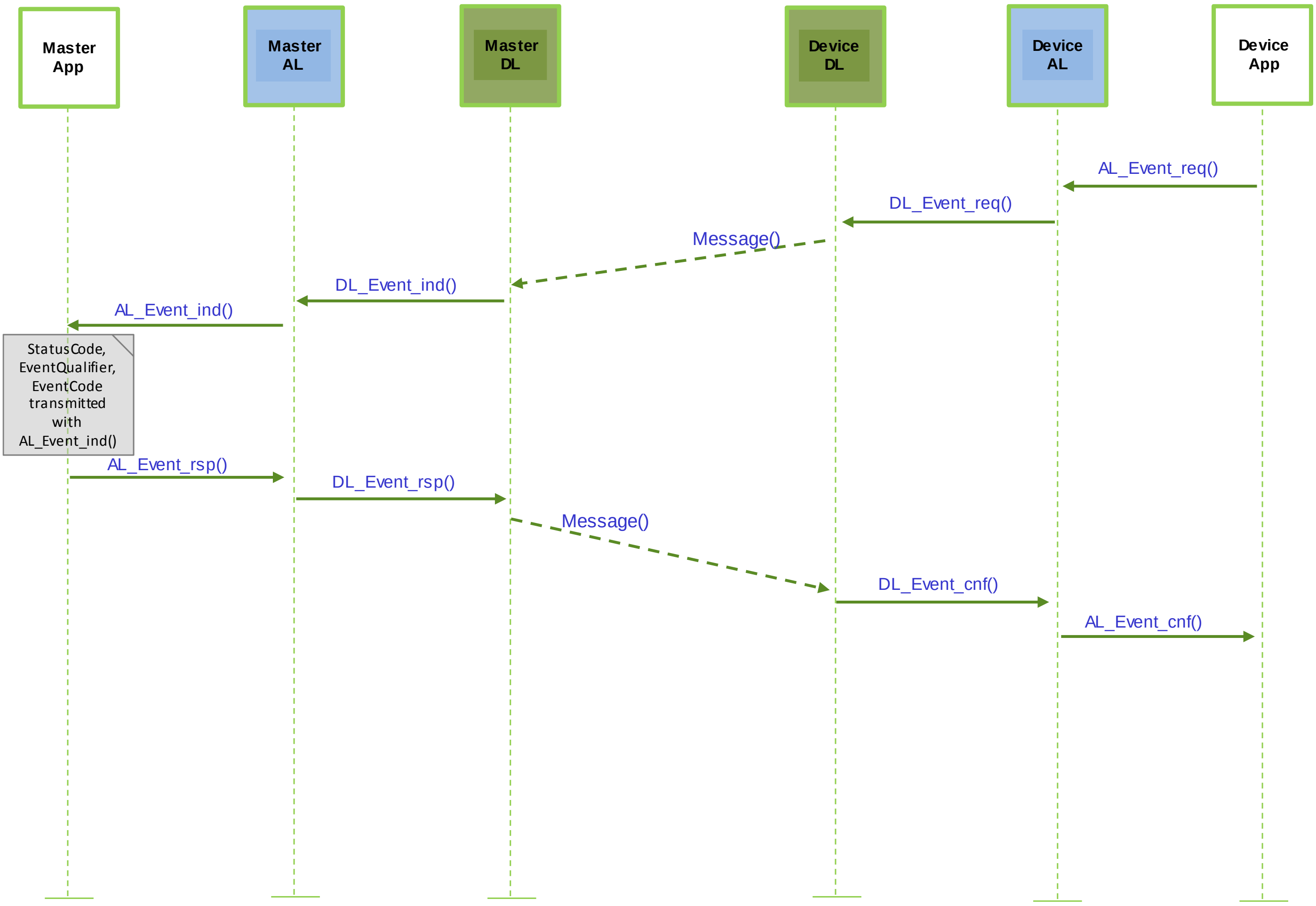


Figure 89 Sequence diagram: Single Event scheduling

8.3.4 Process Data transfer

8.3.4.1 Process Data (PD) state machine of the W-Device-AL

Figure 90 shows the Process Data state machine of the W-Device application layer

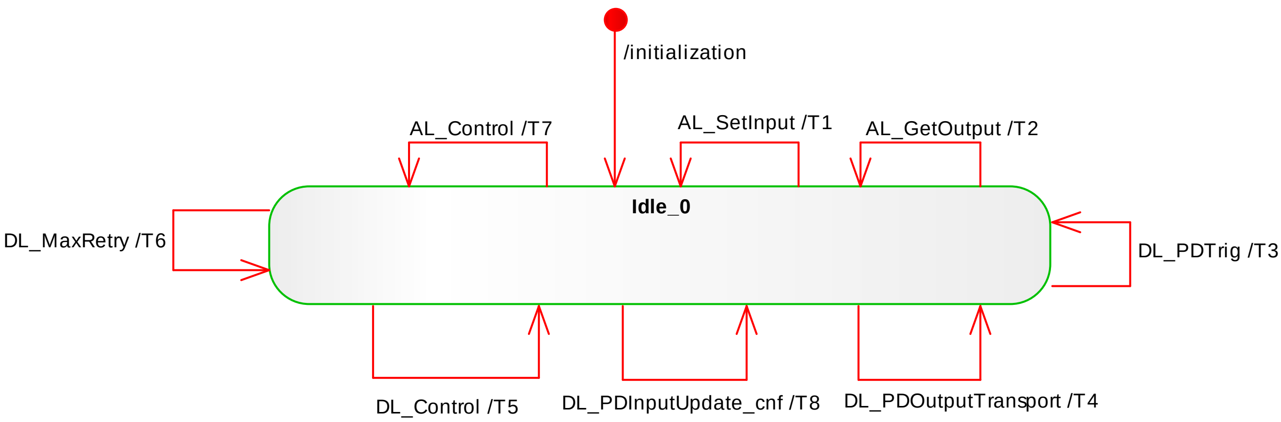


Figure 90 PD state machine of the W-Device-AL

Table 93 State transition tables of the PD W-Device AL

STATE NAME	STATE DESCRIPTION
Idle_0	The W-Device AL is waiting on subordinated AL and DL service calls.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	Invoke DL_PDInputUpdate with Process Data In from AL.
T2	0	0	Read Process Data Out.
T3	0	0	Invoke AL_PDTrig.
T4	0	0	DL_PDOutputTransport delivers Process Data Out from DL. Invoke AL_NewOutput.
T5	0	0	Invoke AL_Control with Process Data Out qualifier status from DL.
T6	0	0	Invoke AL_Control with real-time fault.
T7	0	0	Invoke DL_Control with Process Data In qualifier status from AL.
T8	0	0	Invoke AL_SetInput.cnf

8.3.4.2 Process Data cycles

Figure 91 and Figure 92 demonstrate complete interactions between W-Master and W-Device for output and input Process Data use cases.
Figure 74 demonstrates how the AL and DL services of W-Master and W-Device are involved in the cyclic exchange of output Process Data. The W-Device application is able to acquire the current values of output PD via the AL_GetOutput service.

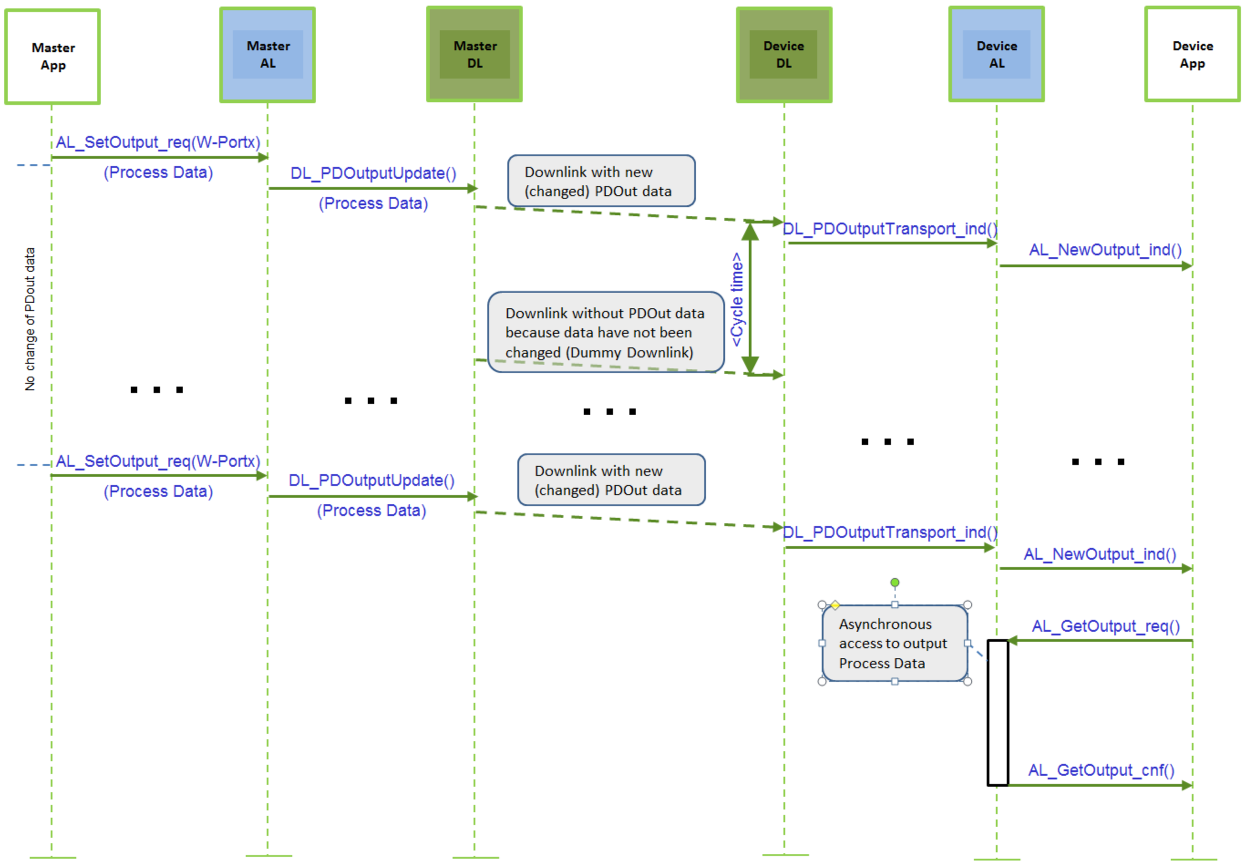


Figure 91 Sequence diagram for output Process Data

Figure 92 demonstrates how the AL and DL services of W-Master and W-Device are involved in the cyclic exchange of input Process Data. The W-Master application is able to acquire the current values of input PD via the AL_GetInput service.

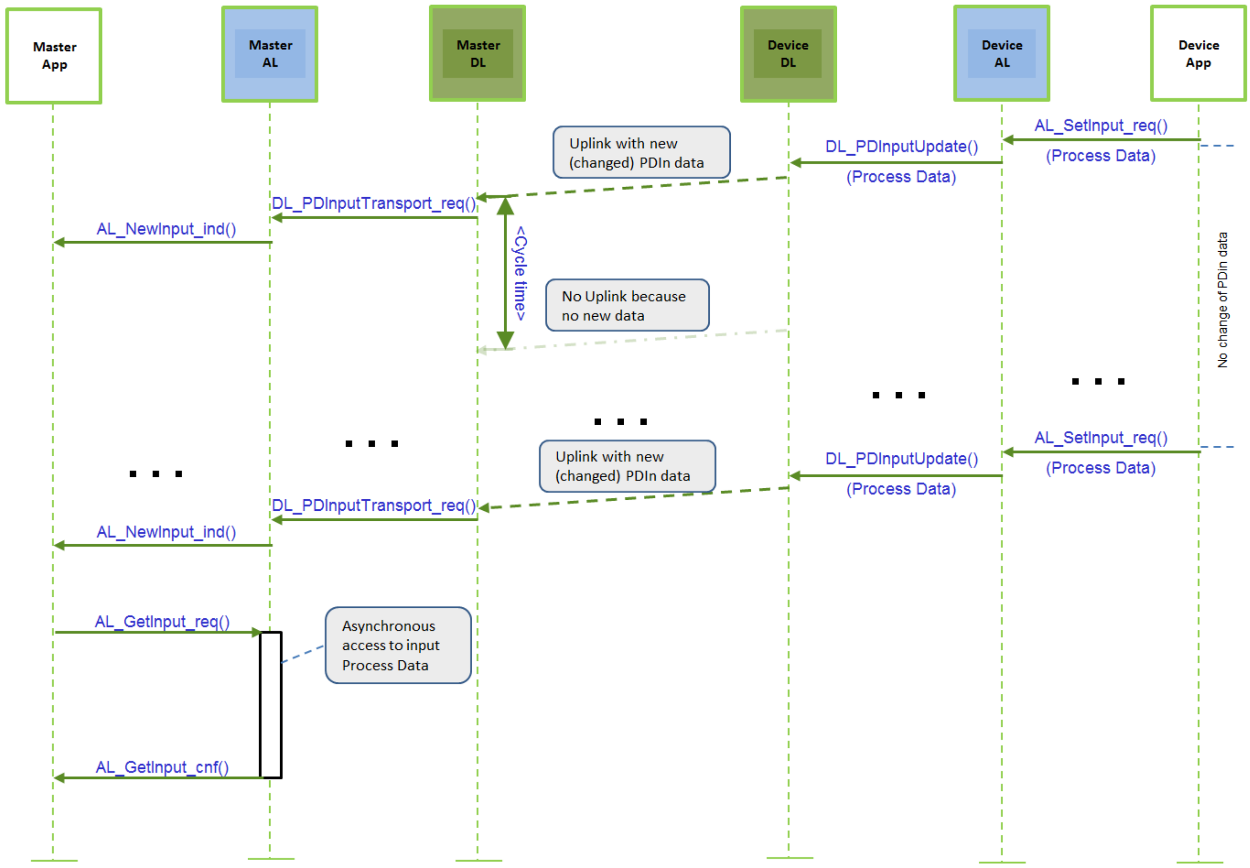


Figure 92 Sequence diagram for input Process Data

9. System management (SM)

9.1 General

The system management (SM) services are used for the coordinated startup and configuration of the possible operational modes within the W-Master and the corresponding W-Devices. Since the difference between the SM of the W-Master and the W-Device is significant, the structure of this clause separates the services and protocols of W-Master and W-Device.

It must be noted that some of the functionality described in this clause is implemented within the underlying PL, such as the handling of the wireless messages.

The following subclauses describe the possible operational modes and associated procedures.

9.1.1 Service Mode

When a track is configured to operate in ServiceMode, the frequency hopping table also utilizes the configuration frequency channels. The ServiceMode can be configured as Scan Mode, Pairing Mode or Roaming Mode. Scan and Pairing Mode are terminating automatically after the intended procedure is completed. Roaming Mode stays permanently active and a discovery procedure is regularly carried out by issuing “Scan Request” messages on the configuration channels. This is required for the “Handover Connect” procedure.

9.1.2 Cyclic Mode

In Cyclic Mode, the W-Master track communicates with the W-Device via the assigned data channel by utilizing the frequency hopping table without configuration frequencies. This mode is utilized with fixed W-Devices.
After successful pairing of all W-Devices for a track, the W-Master can switch via SM_SetTrackMode the mode from ServiceMode to Cyclic Mode. On the W-Device, the Cyclic Mode is immediately entered after successful sending of the final "Pairing Negotiation Response". Scan, Pairing and Roaming is no longer possible on this track in this mode.

9.1.2.1 IMATime monitoring

The IMATime is continuously supervised within the PL. The IMATime is transferred within the extended wireless parameter set to the W-Device during the STARTUP procedure via SM_SetPortConfig.
The monitoring is started after the W-Device is synchronized. In case of an IMATimeout a COMLOST and an IOLW_IMATimeout event (see clause 15.1) will be generated towards the application.
When an offered IMATime (e.g. from PDCT) is rejected by the W-Device, this is indicated via an ISDU ErrorType (e.g. PAR_VALOUTOFRNG) towards the application. In this case, the W-Master executes a DL_Read(IMATime) and starts monitoring using the value from the W-Device until the application changes this setting.
Note: The allowed range of the IMATime must be described in the IODD of the W-Device.

9.2 System management of the W-Master

9.2.1 Overview

The W-Master SM
- Establishes the required communication protocol revision.
- Checks the W-Device compatibility (actual W-Device identifications match expected values).
- Adjusts adequate cycle times.
- Computes the frequency hopping tables.
- Assigns W-Port numbers to the wireless communication relations.

For this it uses the following services shown in Figure 93.
- SM_SetMasterConfig sets the common configuration of the W-Master for all tracks.
- SM_SetTrackMode sets the mode of a wireless track.
- SM_GetTrackMode gets the mode of a wireless track.
- SM_TrackScanEnd indicates the end of the scan mode.
- SM_TrackScanResult reports a new unpaired W-Device within the track’s proximity to the application.
- SM_SetPortConfig transfers the necessary parameters (configuration data) from Configuration Management (CM) to System Management (SM). The communication is then started implicitly.

- SM_PortMode reports the result of the setup back to CM, in case of negative result via corresponding "errors", such as mismatching revisions and incompatible W-Devices.
- SM_GetPortConfig reads the actual and effective parameters.
- SM_Operate switches the ports into the "OPERATE" mode.
- SM_GetPortQuality delivers the quality of the port connection.
- SM_PortPairing handles the pairing process.

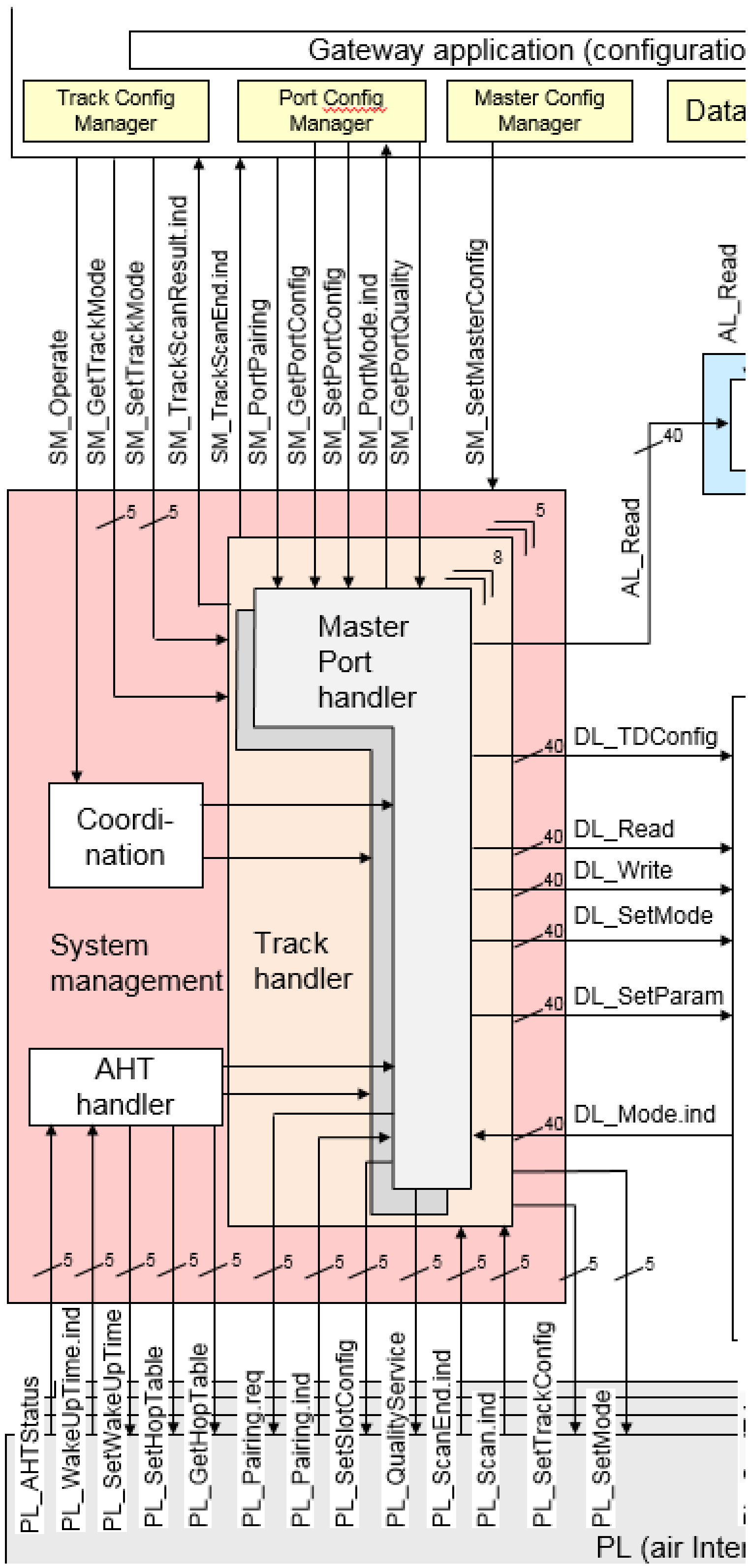


Figure 93 Structure and services of the W-Master system management

The Configuration Manager in a W-Master consists of Master Configuration Manager, Track Configuration Manager and Port Configuration Manager. During initialization, the W-Master's Configuration management (CM) first reads the configuration for the W-Master (MasterID, Blacklist). In the next step, the W-Master

Configuration will be applied, and the radios will be switched on by SM_SetMasterConfig and the W-Master starts sending Downlinks on the track specific frequency channels until it gets an answer from a W-Device in the corresponding uplink slot. The W-Devices are then synchronized, and the W-Master application may call the DL service DL_SetMode (STARTUP) to create the required instances of the Master DL-mode handler.

Figure 94 demonstrates the actions between the layers W-Master application (W-Master App), Configuration Management (CM), System Management (SM), Data Link (DL) and Application Layer (AL) for the startup use case of a particular port.

This particular use case is characterized by the following statements:

- The W-Device for the available configuration is connected and inspection is successful
- The W-Device uses the correct RevisionID according to this specification
- The configured InspectionLevel is "type compatible" (SerialNumber is read out of the W-Device, but not checked).

Dotted arrows in Figure 94 represent response services to an initial service.

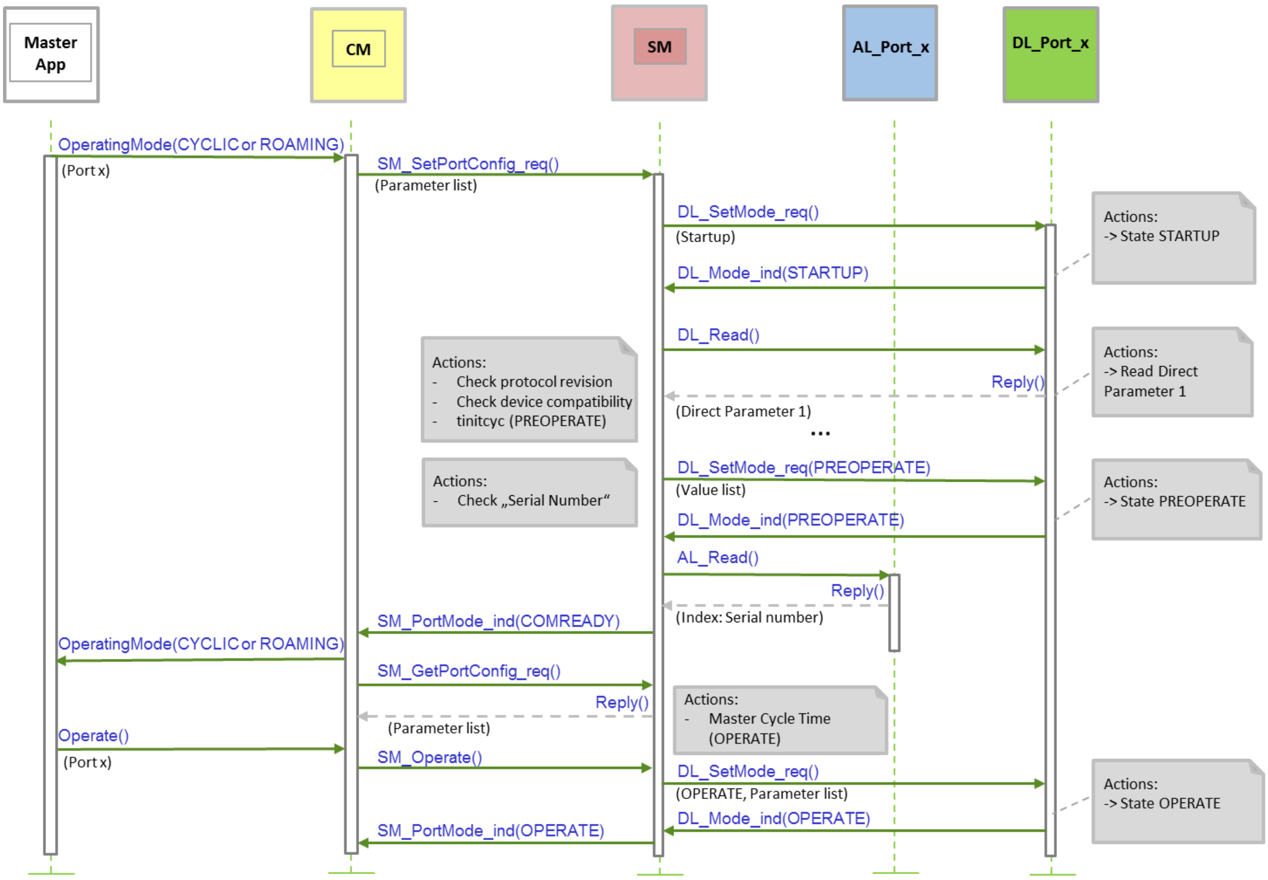


Figure 94 Sequence chart of the use case “port x setup”

9.2.2 System management W-Master services

9.2.2.1 Overview

System management provides the SM W-Master services to the user via its upper interface. Table 94 lists the assignment of the W-Master to its role as initiator or receiver for the individual SM services.

Table 94 SM services within the W-Master

Service Name		**W-Master**
SM_SetMasterConfig	Set common configuration of the W-Master for all tracks	R
SM_SetTrackMode	Set mode of a wireless track	R
SM_GetTrackMode	Get mode of a wireless track	R
SM_TrackScanResult	Report a new unpaired W-Device within the track's proximity to the application	I
SM_TrackScanEnd	Indicates the end of the scan mode	I
SM_SetPortConfig	Set configuration of a virtual wireless port	R
SM_GetPortConfig	Get configuration of a virtual wireless port	R
SM_PortPairing	Pair W-Device to W-Master	R
SM_PortMode	Reports the mode of a wireless port	I
SM_GetPortQuality	Acquire quality of a W-Device connection	R
SM_Operate	Activate a wireless port	R
Key (see 3.3.5) I: Initiator of service R: Receiver (Responder) of service		

9.2.2.2 SM_SetMasterConfig

The SM_SetMasterConfig service is used to set up the W-Master configuration. This configuration is used for all tracks. The parameters of the service primitives are listed in Table 95

Table 95 SM_SetMasterConfig

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured master parameters of a W-Master.
Parameter type: Record
Record Elements:

MasterID
This parameter contains the MasterID of the W-Master (see Table 165)
Permitted values: 1 to 29

BlackList
This parameter contains the frequency channels which shall not be used by the W-Master.
Permitted values: 0x0000 0000 0000 0000 0000 to 0x7FFF FFFF FFFF FFFF FFFE
(bitwise coded 1MHz channels 2-79 (LSB first))

SyncMaster:
This parameter specifies the track number which shall run as W-Frame synchronization master (see 5.5.2.1. Parameter “TrackSynchronization” in service PL_SetTrackConfig)
Permitted values:
0 (SyncMaster is track 0)
...
4 (SyncMaster is track 4)
5 (all tracks using an external synchronization signal)

AHTEnable:
This parameter conatins the AHT operation mode. Permitted values:
ENABLE
DISABLE

Result (+):
This selection parameter indicates that the service has been executed successfully

Result (-):
This selection parameter indicates that the service failed

ErrorInfo
This parameter contains the error information
Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

9.2.2.3 SM_SetTrackMode

The SM_SetTrackMode service is used to set up one track with the requested track configuration. The parameters of the service primitives are listed in Table 96.

Table 96 SM_SetTrackMode

Parameter Name	.req	.cnf
Argument	M	
TrackMode	M	
TxPower	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
TrackMode
This parameter indicates the requested operational modes of the track
Permitted values:
STOP (Communication disabled, radio turned off),
CYCLIC (W-Master is working in Cyclic mode (Full performance),
SCAN (W-Master is working in Scan mode (Limited performance)),
ROAMING (W-Master is working in Roaming mode. (Limited performance).
TXPower
This parameter contains the requested transmit power level of the track
Permitted values: 1 to 255 (see Table 176).
Result (+):
This selection parameter indicates that the service has been executed successfully
Result (-):
This selection parameter indicates that the service failed
ErrorInfo
This parameter contains the error information
Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)
STATE_CONFLICT (service unavailable within current state)

9.2.2.4 SM_GetTrackMode

The SM_GetTrackMode service is used to read the track configuration from the system management. The parameters of the service primitives are listed in Table 97.

Table 97 SM_GetTrackMode

Parameter Name	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
Result (+):
This selection parameter indicates that the service has been executed successfully
ParameterList
This parameter contains the configured track parameters of a W-Master track.
Parameter type: Record
Record Elements:
TrackMode
This parameter indicates the actual operational mode of the track
Permitted values: STOP, CYCLIC, SCAN, PAIRING, ROAMING
TXPower
This parameter contains the actual transmit power level of the track
Permitted values: 1 to 255 (see Table 176)
Result (-):
This selection parameter indicates that the service failed
ErrorInfo
This parameter contains the error information
Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)
STATE_CONFLICT (service unavailable within current state)

9.2.2.5 SM_TrackScanResult

The SM_TrackScanResult service is used to report a new unpaired W-Device within the track's proximity to the application. This is only done if the track is in ROAMING or SCAN mode. The parameters of the service primitives are listed in Table 98

Table 98 SM_TrackScanResult

Parameter Name	.ind
Argument	M
ParameterList	M

Argument:
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the information of the found W-Device.
Parameter Type: Record
Record Elements:

SlotType:
Type of the W-Device in Uplink given through W-Device application.
Permitted values: SSLOT, DSLOT (see Table 161).

UniqueID:
This parameter indicates the UniqueID of the W-Device. (see Figure 156)

RevisionID:
This parameter indicates the protocol version of the found W-Device (see clause B.1.5 in REF 1).

9.2.2.6 SM_TrackScanEnd (Master)

The SM_TrackScanEnd service is used to indicate the end of the scan mode. The parameters of the service primitive are listed in Table 99.

Table 99 SM_TrackScanEnd

Parameter Name	.ind
<none>	

9.2.2.7 SM_SetPortConfig

The SM_SetPortConfig service is used to set up the requested W-Device configuration. The parameters of the service primitives are listed in Table 100.

Table 100 SM_SetPortConfig

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
W-Port		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured W-Port and W-Device parameters of a W-Master W-Port.

Parameter type: Record
Record Elements:

W-Port
This parameter contains the W-Port number (see TDMapper, 6.1.1.).

Slot_N
This parameter contains the Slot number within the corresponding track number (see TDMapper, 6.1.1)

Track_N
This parameter selects the track number with which the W-Port is assigned to (see TDMapper, 6.1.1.)

SlotType
This parameter indicates the expected SlotType for corresponding W-Device
Permitted values: SSLOT, DSLOT (see Table 161)

TargetMode
This parameter indicates the requested operational mode of the W-Port
Permitted values: INACTIVE, CYCLIC, ROAMING

UniqueID
Data length: 9 octets

ConfiguredCycleTime
This parameter contains the requested cycle time for the OPERATE mode
Permitted values:
0 (FreeRunning)
Time (see clause 14.1.2)

IMATime
This parameter contains the requested IMA time for the OPERATE mode
Permitted values: 2 octets, time encoded according to clause14.3.5.

MaxRetry
This parameter contains the maximum number of retries for a transmission in OPERATE mode
Permitted values: see Table 175.

ConfiguredRevisionID (CRID)
Data length: 1 octet for the RevisionID (see Table 163)

InspectionLevel:
Permitted values: NO_CHECK, TYPE_COMP, IDENTICAL (see Table 101)

ConfiguredVendorID (CVID)
Data length: 2 octets
NOTE VendorIDs are assigned by the IO-Link community

ConfiguredDeviceID (CDID)
Data length: 3 octets

ConfiguredFunctionID (CFID)
Data length: 2 octets

ConfiguredSerialNumber (CSN)
Data length: up to 16 octets

PDInLength
Data length of process data in
Permitted values: 0 to 32

PDOutLength
Data length of process data out
Permitted values: 0 to 32

MaxPDSegLength (only W-Master)
This parameter contains the maximum segment length of the PDOut data to the Message handler to distribute PDOut Data within multiple W-Cycles.

DeviceTXPower
This parameter contains the transmit power level of the W-Device
Permitted values: 1 to 255, see 10.9 IODD for definition

LowPowerDevice
Permitted values: YES, NO

Result (+):
This selection parameter indicates that the service has been executed successfully

W-Port

This parameter contains the W-Port number

Result (-):

This selection parameter indicates that the service failed

W-Port

This parameter contains the W-Port number

ErrorInfo

This parameter contains the error information

Permitted values:

PARAMETER_CONFLICT (consistency of parameter set violated)

STATE_CONFLICT (service unavailable within current state)

Table 101 specifies the coding of the different InspectionLevel

Table 101 Definition of the InspectionLevel (IL)

Parameter	InspectionLevel (IL)		
	NO_CHECK	TYPE_COMP	IDENTICAL
DeviceID (DID) (compatible)	-	Yes (RDID=CDID)	Yes (RDID=CDID)
VendorID (VID)	-	Yes (RVID=CVID)	Yes (RVID=CVID)
SerialNumber (SN)	-	-	Yes (RSN = CSN)
NOTE: For W-Devices with missing SerialNumber, the IL shall not be set to IDENTICAL.			

Table 102 specifies the coding of the different Target Modes.

Table 102 Definitions of the Target Modes

Target Mode	Definition
INACTIVE	Communication disabled
CYCLIC	W-Master is working in Cyclic mode. (Full performance)
ROAMING	W-Master is working in Roaming mode. (Limited performance)

9.2.2.8 SM_GetPortConfig

The SM_GetPortConfig service is used to acquire the real (actual) W-Device configuration. The parameters of the service primitives are listed in Table 103

Table 103 SM_GetPortConfig

Parameter Name	.req	.cnf
Argument	M	
W-Port	M	
Result (+)		S
ParameterList		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

W-Port

This parameter contains the W-Port number

Result (+):
This selection parameter indicates that the service has been executed successfully

ParameterList
This parameter contains the configured W-Port and W-Device parameters of a W-Master W-Port.
Parameter type: Record
Record Elements:

W-Port
This parameter contains the W-Port number (see TDMapper, 6.1.1.).

Slot_N
This parameter contains the Slot number within the corresponding track number (see TDMapper, 6.1.1.)

Track_N
This parameter selects the track number which the W-Port is assigned to (see TDMapper, 6.1.1.)

SlotType
This parameter indicates the expected SlotType for corresponding W-Device
Permitted values: SSLOT, DSLOT (see Table 161)

TargetMode
This parameter indicates the requested operational mode of the W-Port
Permitted values: INACTIVE, CYCLIC, ROAMING

UniqueID
Data length: 9 octets

RealCycleTime
This parameter contains the real (actual) cycle time for the OPERATE mode
Permitted values:
0 (FreeRunning)
Time (see clause 14.1.2)

IMATime
This parameter contains the requested IMA time for the OPERATE mode
Permitted values: 2 octets, time encoded according to clause 14.3.5.

MaxRetry
This parameter contains the maximum number of retries for a transmission in OPERATE mode
Permitted values: see Table 175.

RealRevision (RRID)
Data length: 1 octet for the RevisionID (see B.1.5 in REF 1)

RealVendorID (RVID)
Data length: 2 octets
NOTE VendorIDs are assigned by the IO-Link community

RealDeviceID (RDID)
Data length: 3 octets

RealFunctionID (RFID)
Data length: 2 octets

RealSerialNumber (RSN)
Data length: up to 16 octets

PDInLength
Data length of process data in
Permitted values: 0 to 32

PDOutLength
Data length of process data out
Permitted values: 0 to 32

MaxPDSegLength (only W-Master)
This parameter contains the maximum segment length of the PDOut data to the Message handler to distribute PDOut
Data within multiple W-Cycles.

DeviceTXPower
This parameter contains the transmit power level of the W-Device
Permitted values: 1 to 255, see 10.9 IODD for definition

LowPowerDevice
Permitted values: YES, NO

Result (-):
This selection parameter indicates that the service failed

W-Port
This parameter contains the W-Port number

ErrorInfo
This parameter contains the error information
Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)

9.2.2.9 SM_PortPairing

The SM_PortPairing service is used to pair a W-Device to the W-Master. The parameters of the service primitives are listed in Table 104

Table 104 SM_PortPairing

Parameter Name	.req	.cnf
Argument	M	
PairingMethod	M	
Timeout	M	
Result (+)		S
W-Port		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

PairingMethod
This parameter indicates the pairing mode which shall be used.
Permitted values: PAIRING_BUTTON, PAIRING_UNIQUE, UNPAIRING, PAIRING_ABORTED.

Timeout
This parameter contains the timeout for a pairing attempt in seconds. See Table 186 (definition of PAIRING_BUTTON_TIMEOUT, PAIRING_UNIQUE_TIMEOUT)
Permitted values: PAIRING_BUTTON_TIMEOUT, PAIRING_UNIQUE_TIMEOUT

Result (+):
This selection parameter indicates that the service has been executed successfully

W-Port
This parameter contains the W-Port number

Result (-):
This selection parameter indicates that the service failed

W-Port
This parameter contains the W-Port number

ErrorInfo
This parameter contains the error information
Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)
STATE_CONFLICT (service unavailable within current state

9.2.2.10 SM_PortMode

The SM_PortMode service is used to indicate changes or faults of the local communication mode. These shall be reported to the W-Master application. The parameters of the service primitives are listed in Table 105.

Table 105 SM_PortMode

Parameter Name	.ind
Argument	M
W-Port	M
Mode	M

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the W-Port number

Mode
Permitted values:
PAIRING_SUCCESS (W-Device has been paired)
PAIRING_TIMEOUT (W-Device hasn‘t been paired within the given timeout)
PAIRING_WRONG_SLOTTYPE (W-Device has different SlotType as requested)
INACTIVE (Communication disabled)
PORTREADY (W-Port configuration successful)
COMREADY (Communication established and inspection successful)
OPERATE (W-Port is ready to exchange Process Data)
COMLOST (Communication failed, new synchronization procedure required)
REVISION_FAULT (Incompatible protocol revision)
COMP_FAULT (Incompatible W-Device or Legacy-Device according to the InspectionLevel)
SERNUM_FAULT (Mismatching SerialNumber according to the InspectionLevel)

9.2.2.11 SM_GetPortQuality

The SM_GetPortQuality service is used to acquire the quality of a W-Device connection. The parameters of the service primitives are listed in Table 106.

Table 106 SM_GetPortQuality

Parameter Name	.req	.cnf
Argument	M	
W-Port	M	
Result (+)		S
W-Port		M
Quality		M
Result (-)		S
W-Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

W-Port
This parameter contains the W-Port number

Result (+):
This selection parameter indicates that the service has been executed successfully

W-Port
This parameter contains the W-Port number

Quality
This parameter contains the quality of a W-Device connection.
Permitted Values: 0 to 100%.

Result (-):
This selection parameter indicates that the service failed

W-Port
This parameter contains the W-Port number
ErrorInfo
This parameter contains the error information
Permitted values:
STATE_CONFLICT (service unavailable within current state)

9.2.2.12 SM_Operate

The SM_Operate service prompts system management to calculate the MasterCycleTime of the ports when they are acknowledged positively with Result (+). This service is effective on all the ports. The parameters of the service primitives are listed in Table 107.

Table 107 SM_Operate

Parameter Name	.req	.cnf
Result (+)		S
Result (-)		S
ErrorInfo		M

Result (+):
This selection parameter indicates that the service has been executed successfully
Result (-):
This selection parameter indicates that the service failed
ErrorInfo
This parameter contains the error information
Permitted values:
TIMING_CONFLICT (the requested combination of cycle times for the activated ports is not possible)

9.2.3 SM W-Master protocol

9.2.3.1 Overview

Due to the comprehensive configuration, parameterization, and operational features of IOLW the description of the behavior with the help of state diagrams becomes rather complex. Similar to the DL state machines 9.2.3 uses the possibility of submachines within the main state machines.
Comprehensive compatibility check methods are performed within the submachine states. These methods are indicated by "do method" fields within the state graphs, for example in Figure 96.
The corresponding decision logic is demonstrated via activity diagrams (see Figure 98, Figure 99, Figure 100, and Figure 101).

9.2.3.2 SM W-Master State machines

9.2.3.2.1 State Machine of the W-Master Track handler

Figure 95 shows the main state machine of the W-Master Track handler. The tracks will be configured (MasterID, Blacklist, ...) and after setting active, the different operating modes (CYCLIC, ROAMING, ...) can be set. The service PL_Scan delivers every single W-Device that has been found within a scan.

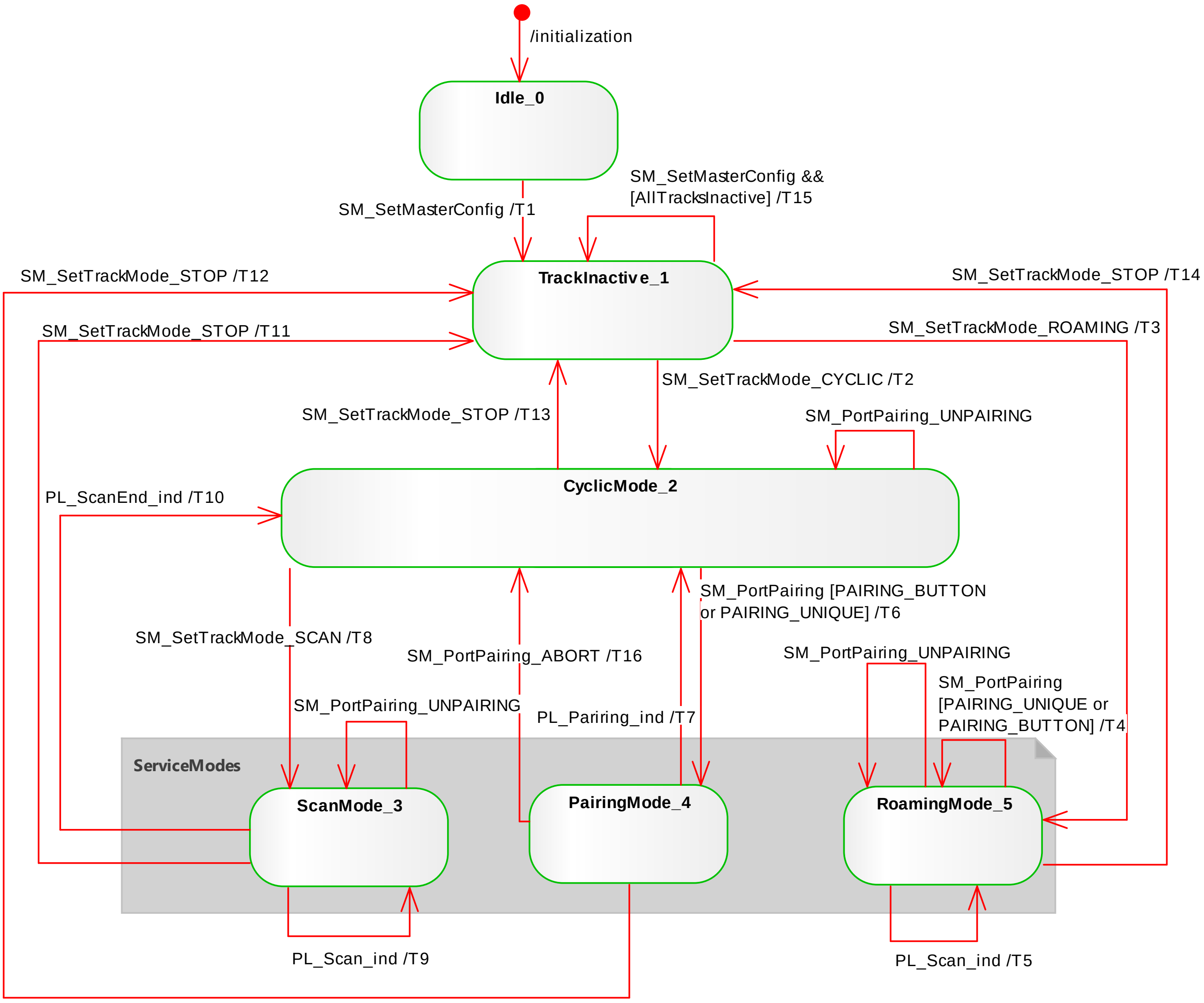


Figure 95 State Machine of the W-Master Track handler

4142 **Table 108 State transition table of the W-Master Track handler**

STATE NAME	STATE DESCRIPTION
Idle_0	-
TrackInactive_1	State is entered after track configuration done via SM_SetMasterConfig. Waiting for activation of operating mode (CYCLIC or ROAMING).
CyclicMode_2	Track is active (CYCLIC mode). The gateway application is exchanging Process Data and ready to send or receive On-request Data.
ScanMode_3	Track is active (SCAN mode) and scanning for unpaired devices via the configuration channels. Found devices are reported to the application via SM_TrackScanResult.
PairingMode_4	Track is active (PAIRING mode). Additionally, the configuration channels are active. This state is left by PL_Pariring.ind automatically, if pairing is done.
RoamingMode_5	Track is active (ROAMING mode). Additionally, the configuration channels are active. Found devices are reported to the application via SM_TrackScanResult.

4143

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_SetTrackConfig() to configure all available W-Master tracks with identical MasterID and Blacklist: PL_SetTrackConfig.req (MasterID, Blacklist, Track_N++, MASTER / SLAVE) The tracks shall be configured in the following way: Each track shall get a unique track number from 0 up to 4, (e.g. consecutively numbering → Track_N++) The number given in SM_SetMasterConfig(SyncMaster) selects the track which shall become TrackSynchronization = MASTER (except SyncMaster = 5).
T2	1	2	Invoke PL_SetMode (CYCLIC)
T3	1	5	Invoke PL_SetMode (ROAMING)
T4	5	5	Invoke PL_Pairing
T5	5	5	Invoke SM_TrackScanResult to report unpaired W-Devices within the track's proximity
T6	2	4	Invoke PL_Pairing (PAIRING_UNIQUE or PAIRING_BUTTON)
T7	4	2	Invoke SM_PortMode.ind (PAIRING_SUCCESS or PAIRING_TIMEOUT) after pairing is done.
T8	2	3	Invoke PL_SetMode (SCAN).
T9	3	3	See T5.
T10	3	2	Scan procedure is finished and reported by PL via service PL_ScanEnd.ind Invoke SM_TrackScanEnd.ind
T11	3	1	Invoke PL_SetMode (STOP)
T12	4	1	See T11.
T13	2	1	See T11.
T14	5	1	See T11.
T15	1	1	See T1
T16	4	2	*Paring procedure is aborted by the W-Master application.* Invoke PL_Pairing(ABORT)

4144

INTERNAL ITEMS	TYPE	DEFINITION
AllTracksInactive	Bool	This value is set if all tracks are in the state TrackInactive_0

4145

9.2.3.2.2 State Machine of the W-Port-handler

Figure 96 shows the main state machine of the Master W-Port-handler. Two submachines for the compatibility and SerialNumber check are specified in subsequent sections. In case of communication disruption, the system management is informed via the service DL_Mode (COMLOST). Only the SM_SetPortConfig service allows reconfiguration of a port. The service SM_PortPairing allows pairing and unpairing of a W-Device. The service SM_Operate (effective on all ports) causes no effect in any state except in state "wait_7".

The SerialNumber of a pure W-Device shall follow the rules in clause 14.3.8 SerialNumber, since the pairing mechanism covers the identity check of the W-Device.

A W-Bridge shall route the SerialNumber of its connected wired IO-Link Device to support the SerialNumber check for InspectionLevel in the same way as for a pure wired device.

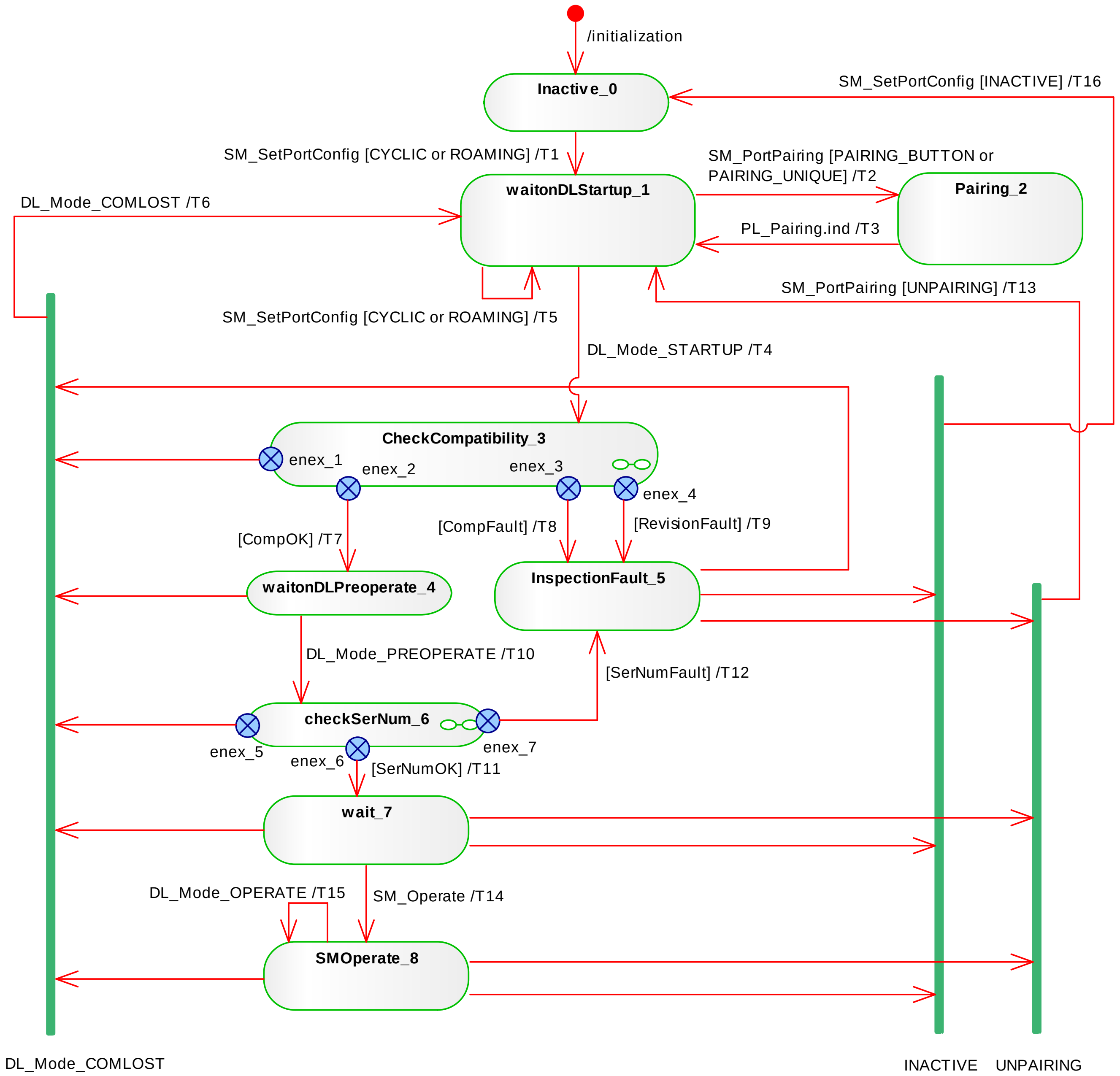


Figure 96 State Machine of the W-Port-handler

Table 109 shows the state transition tables of the Master W-Port-handler.

Table 109 State transition table of the W-Port-handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for configuration of W-Port from W-Port Config Manager
waitonDLStartup_1	Waiting for W-Device to be synced
Pairing_2	Waiting for pairing response from PL
CheckCompatibility_3	W-Port is started, and revision and W-Device compatibility is checked. See Figure 97.
waitonDLPreoperate_4	Wait until the PREOPERATE state is established and all the On-Request handlers are started. W-Port is ready to communicate.
InspectionFault_5	W-Port is ready to communicate. However, cyclic Process Data exchange cannot be performed due to incompatibilities.
CheckSerNum_6	SerialNumber is checked depending on the InspectionLevel (IL). See Figure 101.
wait_7	W-Port is ready to communicate and waits on service SM_Operate from CM.
SM Operate_8	W-Port is in state OPERATE and performs cyclic Process Data exchange.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_SetSlotConfig with configuration of W-Port from W-Port Config Manager Invoke DL_SetMode.req (STARTUP) Invoke DL_SetParam(ValueList) Invoke DL_TDConfig(ValueList)
T2	1	2	Invoke PL_Pairing.req (PAIRING_BUTTON or PAIRING_UNIQUE) depending on PairingMethod in SM_PortPairing.
T3	2	1	Invoke SM_PortMode.ind to signal pairing state to application
T4	1	3	VerRetry = 0, CompRetry = 0
T5	1	1	Invoke PL_SetSlotConfig with updated configuration of W-Port Invoke DL_SetParam(ValueList) Invoke DL_TDConfig(ValueList)
T6	3,4,5,6,7,8	1	Invoke DL_SetMode.req (INACTIVE) and SM_PortMode.ind (COMLOST) due to communication fault
T7	3	4	Write MasterCmd (DevicePreoperate) , Invoke DL_SetMode.req (PREOPERATE, ValueList)
T8	3	5	Invoke SM_PortMode.ind (COMP_FAULT), Write MasterCmd (DevicePreoperate) , DL_SetMode.req (PREOPERATE, ValueList)
T9	3	5	Invoke SM_PortMode.ind (REVISION_FAULT), Write MasterCmd (DevicePreoperate) , DL_SetMode.req (PREOPERATE, ValueList)
T10	4	6	-
T11	6	7	Invoke SM_PortMode.ind (COMREADY)
T12	6	5	Invoke SM_PortMode.ind (SERNUM_FAULT)
T13	5,7,8	1	Write MasterCmd (UnPairing) Invoke PL_Pairing.req (UNPAIRING)
T14	7	8	Write MasterCmd (DeviceOperate) Invoke DL_SetMode.req (OPERATE, ValueList)
T15	8	8	Invoke SM_PortMode.ind (OPERATE)
T16	5,7,8	0	Write MasterCmd (Inactive) SM_PortMode.ind (INACTIVE), DL_SetMode.req (INACTIVE)

INTERNAL ITEMS	TYPE	DEFINITION
CompOK	Bool	See Figure 99
CompFault	Bool	See Figure 99; error variable COMP_FAULT
RevisionFault	Bool	See Figure 98; error variable REVISION_FAULT
SerNumFault	Bool	See Figure 102; error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 102
INACTIVE	Variable	A target mode in service SM_SetPortConfig
CYCLIC, ROAMING	Variables	Target Modes in service SM_SetPortConfig
MasterCmd	Service	DL_Write(0x00, 0x01, ..)

9.2.3.2.3 SM W-Master submachines

Figure 97 shows the Master W-Port-handler submachine checkCompatibility_3.

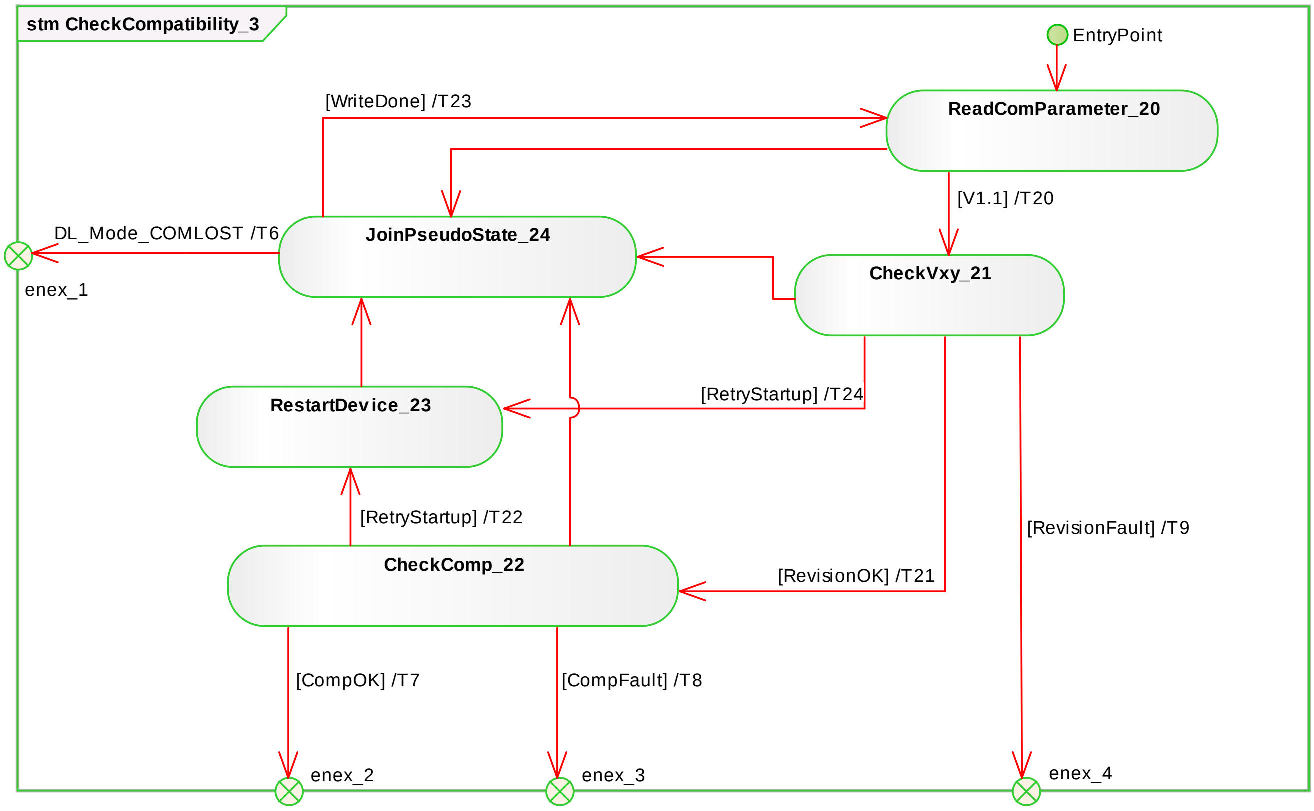


Figure 97 Submachine CheckCompatibility_3 of the W-Port-handler

Table 110 State transition table Submachine Check Compatibility 3 W-Port-handler

STATE NAME	STATE DESCRIPTION
ReadComParameter_20	Acquires communication parameters from Direct Parameter Page 1 (0x02 to 0x06) via service DL_Read (see Table 164.).
CheckVxy_21	A check is performed whether the configured revision (CRID) matches the real (actual) revision (RRID) according to Figure 98
CheckComp_22	Acquires identification parameters from Direct Parameter Page 1 (0x07 to 0x0D) via service DL_Read (see Table 164). The configured InspectionLevel (IL) defines the decision logic of the subsequent compatibility check "CheckComp" according to Figure 99
RestartDevice_23	Writes the compatibility parameters configured protocol revision (CRID) and configured DeviceID (CDID) into the W-Device according to Figure 100
JoinPseudoState_24	This pseudo state is used instead of a UML join bar. No guards involved.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T20	20	21	-
T21	21	22	-
T22	22	23	CompRetry = CompRetry +1
T23	24	20	-
T24	21	23	VerRetry = VerRetry +1

INTERNAL ITEMS	TYPE	DEFINITION
CompOK	Bool	See Figure 99
CompFault	Bool	See Figure 99; error variable COMP_FAULT
RevisionFault	Bool	See Figure 98; error variable REVISION_FAULT
RevisionOK	Bool	See Figure 98
SerNumFault	Bool	See Figure 102 error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 102
V1.0	Bool	Real protocol revision of connected W-Device is in accordance with this standard
RetryStartup	Bool	See Figure 98 and Figure 99
VerRetry	Variable	Internal counter
CompRetry	Variable	Internal counter
WriteDone	Bool	Finalization of the restart service sequence

Some states contain complex logic to deal with the compatibility and validity checks. Figure 98 to Figure 101 are demonstrating the context.

Figure 98 shows the decision logic for the protocol revision check in state "CheckVxy_21". In case of configured Devices, the following rule applies: if the configured revision (CRID) and the real revision (RRID) do not match, the CRID will be transmitted to the Device. If the Device does not accept, the Master returns an indication via the SM_Mode service with REV_FAULT.

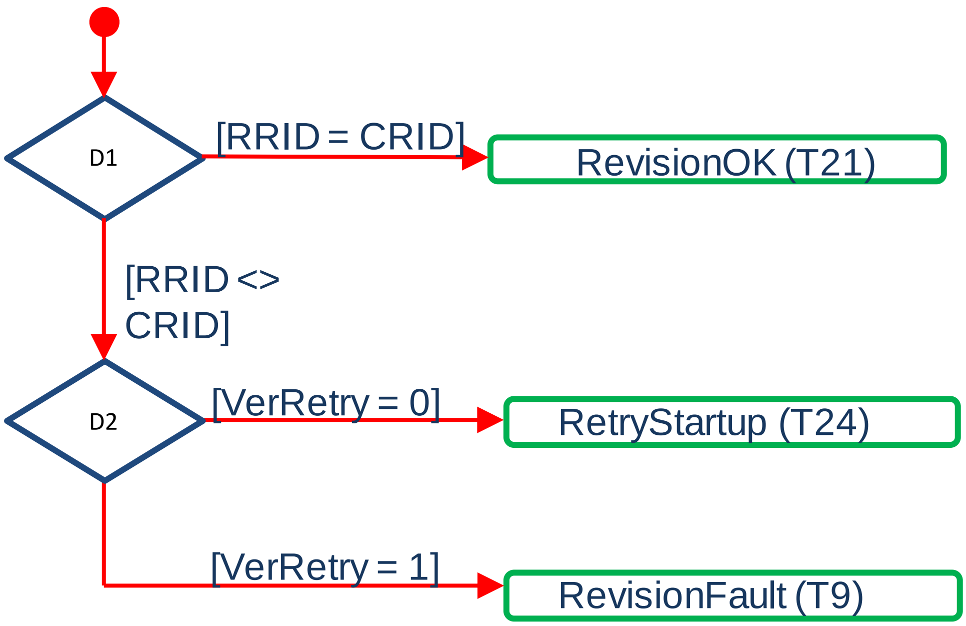


Figure 98 Activities for state „CheckVxy_21“

Figure 99 shows the decision logic for the compatibility check in state "CheckComp_22".

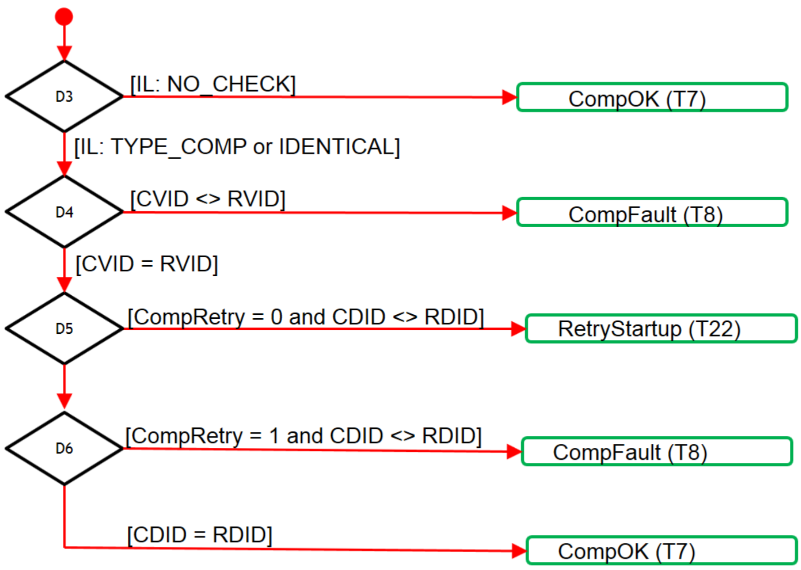


Figure 99 Activities for state „CheckComp_22“

Figure 100 shows the activity (write parameter) in state "RestartDevice_23".

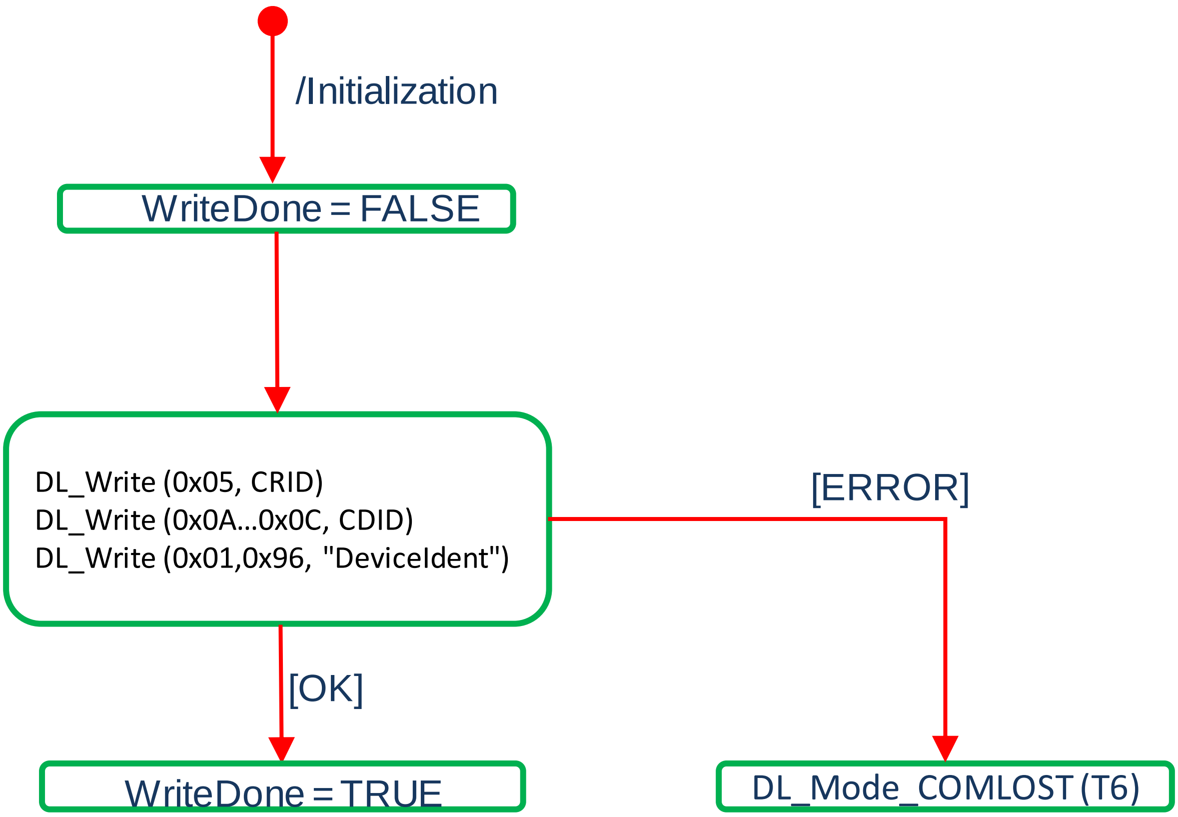


Figure 100 Activities (write parameter) in state "RestartDevice_23"

Figure 101 shows the SM Master submachine "checkSerNum_6". This check is mandatory.

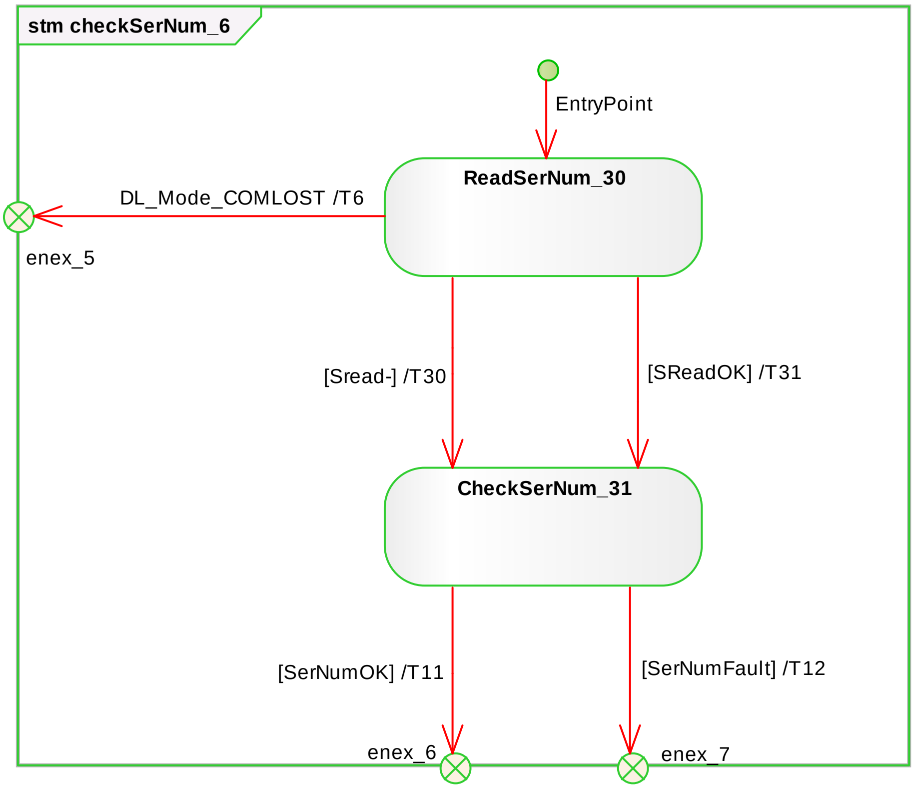


Figure 101 Submachine CheckSerNum_6 of the W-Port-handler

Table 111 State transition table Submachine CheckSerNum_6 of the W-Port-handler

STATE NAME	STATE DESCRIPTION
ReadSerNum_30	Acquires the SerialNumber from the W-Device via AL_Read.req (Index: 0x0015). A positive response (AL_Read(+)) leads to SReadOK = true. A negative response (AL_Read(-)) leads to SRead- = true.
CheckSerNum_61	The configured (CSN) stored in W-Master and the real (RSN) SerialNumber from the W-Device are checked against each other depending on the InspectionLevel (IL) according to Figure 102.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T30	30	31	
T31	30	31	

INTERNAL ITEMS	TYPE	DEFINITION
SRead-	Bool	Negative response of service AL_Read (Index 0x0015)
SReadOK	Bool	SerialNumber read correctly
SERNumOK	Bool	See Figure 102
SERNumFault	Bool	See Figure 102

Figure 102 shows the decision logic (activity) for the state CheckSerNum_6.

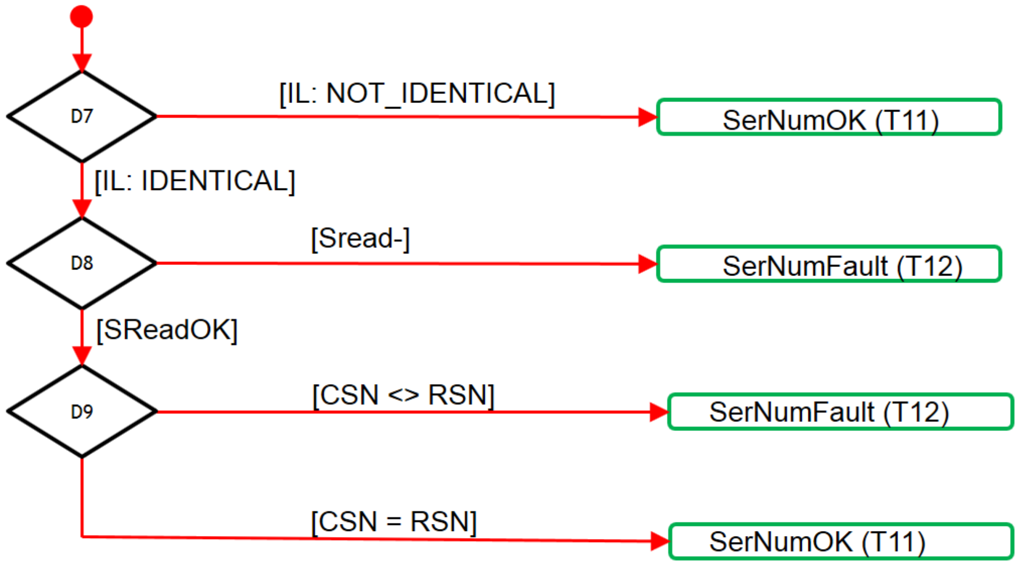


Figure 102 Activities (check SerialNumber) for state CheckSerNum_6

9.2.3.2.4 State Machine of W-Master AHT handler

Figure 103 shows the main state machine of the W-Master AHT-handler. The hopping table will be configured and updated to all tracks and connected W-Devices, see clause 18.4.

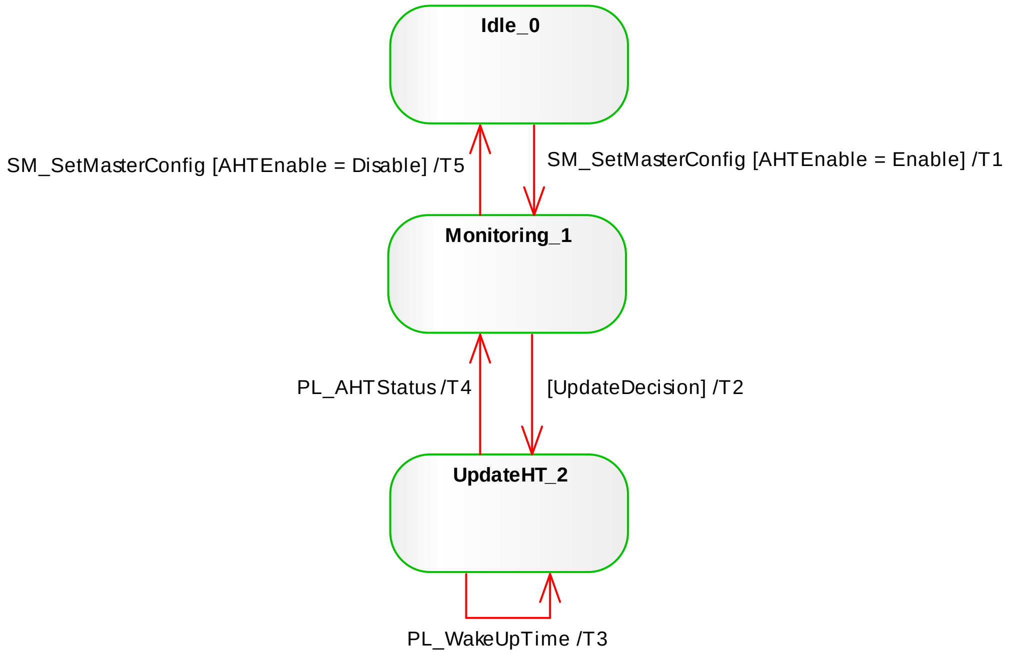


Figure 103 State Machine of the W-Master AHT-handler

4223 **Table 112 State transition table of the W-Master AHT-handler**

STATE NAME	STATE DESCRIPTION
Idle_0	-
Monitoring_1	State monitors the hopping frequencies and decides if to perform an update. Data will be collected by radio manufacturer services
UpdateHT_2	Update sequence in progress.

4224

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_GetHopTable(Track_N) for each Track number to accuire all the hopping tables of the W-Master. Set AHTEnable = Enable.
T2	1	2	Invoke PL_SetWakeUpTime.req(WakeUpTime) Invoke PL_SetHopTable to dedicated track Low energy W-Devices: Invoke DL_Write.req(WakeUp) All regular W-Devices: Invoke DL_Write.req(HopTable)
T3	2	2	Invoke DL_Write.req (HopTable + WakeUpTime)
T4	2	1	Status: JUMP_SUCCESS - Invoke PL_SetHopTable to all other tracks WAKE_UP_ABORT – restart sequence, invoke T2 JUMP_FAIL – return to previous HopTable, invoke T2 STOP - PL track has stopped, update aborted
T5	1	3	Set AHTEnable = Disable

4225

INTERNAL ITEMS	TYPE	DEFINITION
AHTEnable	variable	Indicates the AHT operaion mode, set by SM_SetMasterConfig
UpdateDecision	guard	Indicates a decision to update the hopping table

4226

9.3 System management of the W-Device

9.3.1 Overview

Figure 104 provides an overview of the structure and services of the W-Device system management.

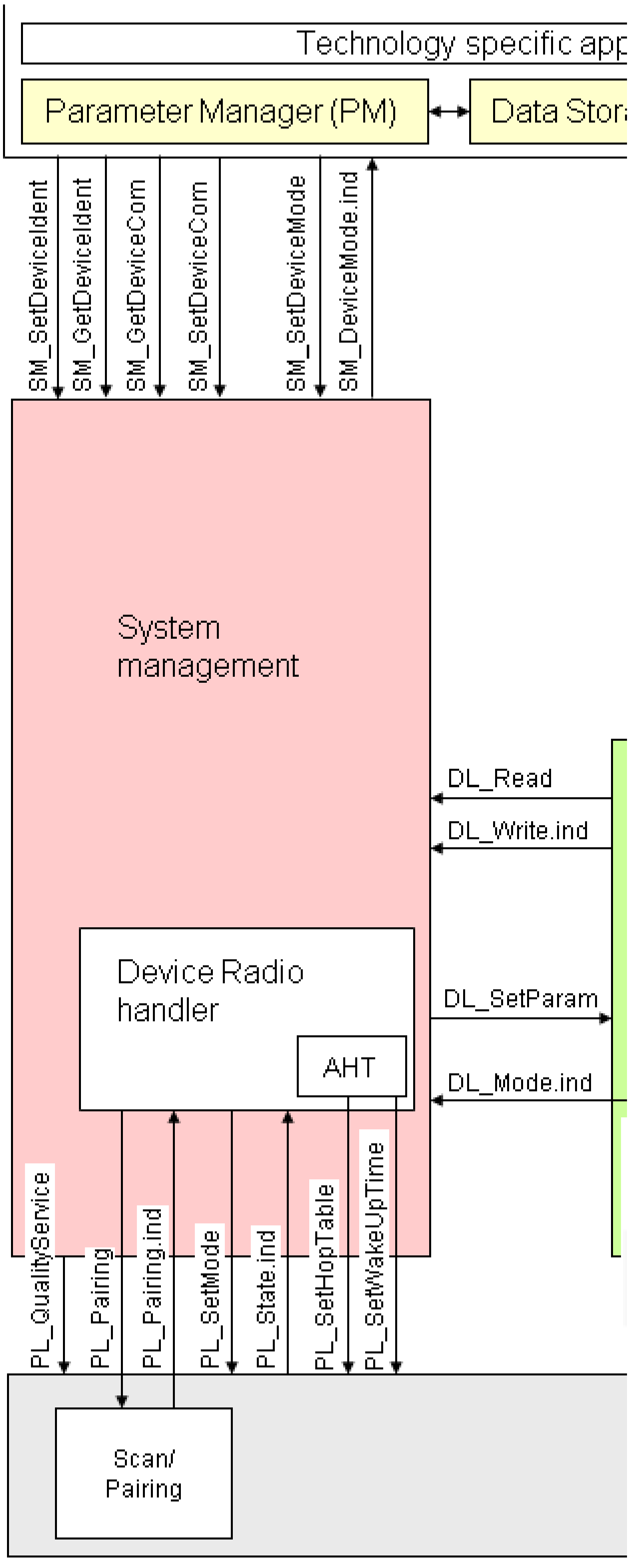


Figure 104 Structure and services of the W-Device system management

The System Management (SM) of the W-Device provides the central controlling instance via the PL through all the phases of initialization, communication startup and communication.

The Device SM interacts with the PL to establish the necessary radio adjustments (see Figure 56 PL W-Device state machine), with the DL to get the necessary information from the W-Master and with the W-Device applications to ensure the Device identity and compatibility (identification parameters).
The transitions between the W-Device PL states (see Figure 56) are initiated by the W-Master track activities (scan, pairing, synchronization, …) and triggered through the Device Data Link Layer via the DL_Mode indications and DL_Write requests (commands).
The SM provides the Device identification parameters through the Device applications interface.
The sequence chart in Figure 105 demonstrates the two possibilities of pairing of a typical Device sequence. It shows only the actions until the ComEstablish state. The remaining actions until the OPERATE state can be taken from
Figure 108.

Figure 105 Sequence chart of a Device pairing

9.3.2 System management W-Device services

9.3.2.1 Overview

Subclause 9.3.2 describes the services the W-Device system management provides to its applications as shown in Figure 104

Table 113 lists the assignment of the W-Device to its role as initiator or receiver for the individual system management service.

Table 113 System management services within the W-Device

Service Name		W-Device
SM_SetDeviceCom	Configure communication properties supported by W-Device	R
SM_GetDeviceCom	Read the current communication properties	R
SM_SetDeviceIdent	Configure W-Device identification data	R
SM_GetDeviceIdent	Read W-Device identification parameter	R
SM_SetDeviceMode	Set W-Device into a defined operational state during initialization	R
SM_Device Mode	Indicate changes of communication states to the W-Device application	I
Key (see 3.3.5) I: Initiator of service R: Receiver (Responder) of service		

9.3.2.2 SM_SetDeviceCom

The SM_SetDeviceCom service is used to configure the communication properties supported by the W-Device in the system management. The parameters of the service primitives are listed in Table 114.

Table 114 SM_SetDeviceCom

Parameter Name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured communication parameters for a W-Device.
Parameter type: Record
Record Elements:

MAXRetry
This parameter contains the maximum number of allowed retries in count of W-Sub-Cycles (see clause 14.3.6).

IMATime
This parameter contains the I am alive time. Permitted values: 2 octets, time encoded according to clause 14.3.5.

TXPower
This parameter contains the transmission power for the W-Device (see Table 176).

DLinkType
This parameter contains the downlink type (see Table 23) for the radio to listen (full downlinks or pre-downlinks).

SlotType
This parameter contains the uplink type (see Table 161) for the uplink capability of the radio (single slot or double slot).

UniqueID
This parameter contains the UniqueID from the W-Device (see Figure 156).

MinCycleTime

This parameter contains the minimum cycle time supported by the W-Device (see 14.1.2)

RevisionID (RID)

This parameter contains the protocol revision (see clause 14.1.3) supported by the W-Device.

ProcessDataIn

This parameter contains the length of PD to be sent to the W-Master.

ProcessDataOut

This parameter contains the length of PD to be sent by the W-Master.

Result (+):

This selection parameter indicates that the service has been executed successfully.

Result (-):

This selection parameter indicates that the service failed.

ErrorInfo

This parameter contains the error information.

Permitted values:

PARAMETER_CONFLICT (consistency of parameter set violated)

9.3.2.3 SM_GetDeviceCom

The SM_GetDeviceCom service is used to read the current communication properties from the system management. The parameters of the service primitives are listed in Table 115.

Table 115 SM_GetDeviceCom

Parameter Name	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument

The service-specific parameters are transmitted in the argument.

Result (+):

This selection parameter indicates that the service has been executed successfully.

ParameterList

This parameter contains the configured communication parameters for a W-Device.

Parameter type: Record

Record Elements:

MAXRetry

This parameter contains the current number of allowed retries in count of W-Sub-cycles (see clause 14.3.6).

IMATime

This parameter contains the current I am alive time. Permitted values: 2 octets, time encoded according to clause 14.3.5.

TXPower

This parameter contains the current transmission power for the W-Device (see Table 176).

DLinkType

This parameter contains the current downlink type (see Table 23) for the radio to listen (full downlinks or pre-downlinks).

SlotType

This parameter contains the uplink type (see Table 161) for the uplink capability of the radio (single slot or double slot).

MasterCycleTime

This parameter contains the MasterCycleTime to be set by the W-Master system management (see clause 14.1.2). This parameter is only valid in the state SM_Operate.

RevisionID (RID)

This parameter contains the current protocol revision (see clause 14.1.3) within the system management of the W-Device.

ProcessDataIn
This parameter contains the current length of PD to be sent to the W-Master (see clause 14.1.4).
ProcessDataOut
This parameter contains the current length of PD to be sent by the W-Master (see clause 14.1.5).
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state, should only be returned if no communication parameters are set)

9.3.2.4 SM_SetDeviceIdent

The SM_SetDeviceIdent service is used to configure the W-Device identification data in the system management. The parameters of the service primitives are listed in Table 116.

Table 116 SM_SetDeviceIdent

Parameter name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
ParameterList
This parameter contains the configured identification parameter for a W-Device.
Parameter type: Record
Record Elements:
VendorID (VID)
This parameter contains the VendorID assigned to a W-Device (see B.1.8)
Data length: 2 octets
DeviceID (DID)
This parameter contains one of the assigned DeviceIDs (see B.1.9)
Data length: 3 octets
FunctionID (FID)
This parameter contains one of the assigned FunctionIDs (see B.1.10).
Data length: 2 octets
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

9.3.2.5 SM_GetDeviceIdent

The SM_GetDeviceIdent service is used to read the W-Device identification parameter from the system management. The parameters of the service primitives are listed in Table 117.

Table 117 SM_GetDeviceIdent

Parameter name	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
Result (+):
This selection parameter indicates that the service has been executed successfully.
ParameterList
This parameter contains the configured communication parameters of the W-Device.
Parameter type: Record
Record Elements:
VendorID (VID)
This parameter contains the actual VendorID of the W-Device (see B.1.8)
Data length: 2 octets
DeviceID (DID)
This parameter contains the actual DeviceID of the W-Device (see B.1.9)
Data length: 3 octets
FunctionID (FID)
This parameter contains the actual FunctionID of the W-Device (see B.1.10).
Data length: 2 octets
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state should only be returned if no identification data are set).

9.3.2.6 SM_SetDeviceMode

The SM_SetDeviceMode service is used to set the W-Device into a defined operational state during initialization. The parameters of the service primitives are listed in Table 118

Table 118 Service SM_SetDeviceMode

Parameter Name	.req	.cnf
Argument	M	
Mode	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.
Mode
Permitted values:
IDLE (W-Device changes to waiting for configuration via application)
ESTABCOM (W-Device changes to waiting for synchronization or pairing by UniqueID)
PAIRING_BUTTON (W-Device changes to waiting for pairing by button)
Result (+):
This selection parameter indicates that the service has been executed successfully.
Result (-):
This selection parameter indicates that the service failed.
ErrorInfo
This parameter contains the error information.
Permitted values:
STATE_CONFLICT (service unavailable within current state)

9.3.2.7 SM_DeviceMode

The SM_DeviceMode service is used to indicate changes of communication states to the W-Device application. The parameters of the service primitives are listed in Table 119.

Table 119 Service SM_DeviceMode

Parameter Name	.ind
Argument	M
Mode	M

Argument
The service-specific parameters are transmitted in the argument.

Mode
Permitted values:

IDLE	(W-Device changed to waiting for configuration)
ESTABCOM	(W-Device changed to the SM mode "SM_ComEstablish")
UNPAIRED	(W-Device is unpaired at startup)
PAIRED	(W-Device is paired at startup)
TIMEOUT	(timeout occurred)
PERMANENT	(W-Device has been paired permanently)
TEMPORARY	(W-Device has been paired as roaming W-Device)
PAIRING_BUTTON	(W-Device changed to waiting for pairing by button)
STARTUP	(W-Device changed to the STARTUP mode)
IDENT_STARTUP	(W-Device changed to the SM mode "SM_IdentStartup")
IDENT_CHANGE	(W-Device changed to the SM mode "SM_IdentCheck")
PREOPERATE	(W-Device changed to the PREOPERATE mode)
OPERATE	(W-Device changed to the OPERATE mode)

9.3.3 SM W-Device protocol

9.3.3.1 Overview

The behavior of the W-Device is mainly driven by W-Master messages. Compared to IO-Link (cyclic Process Data exchange) the transmission of Process Data between a W-Master and a W-Device is only necessary if they change. Therefore, a W-Device can send Process Input Data without an explicit request of the W-Master. A W-Device can also send events without a W-Master request.

9.3.3.2 State machine of W-Device System Management

Figure 106 shows the state machine for W-Device System Management, it evaluates the different communication phases during startup and controls communication status of the W-Device.

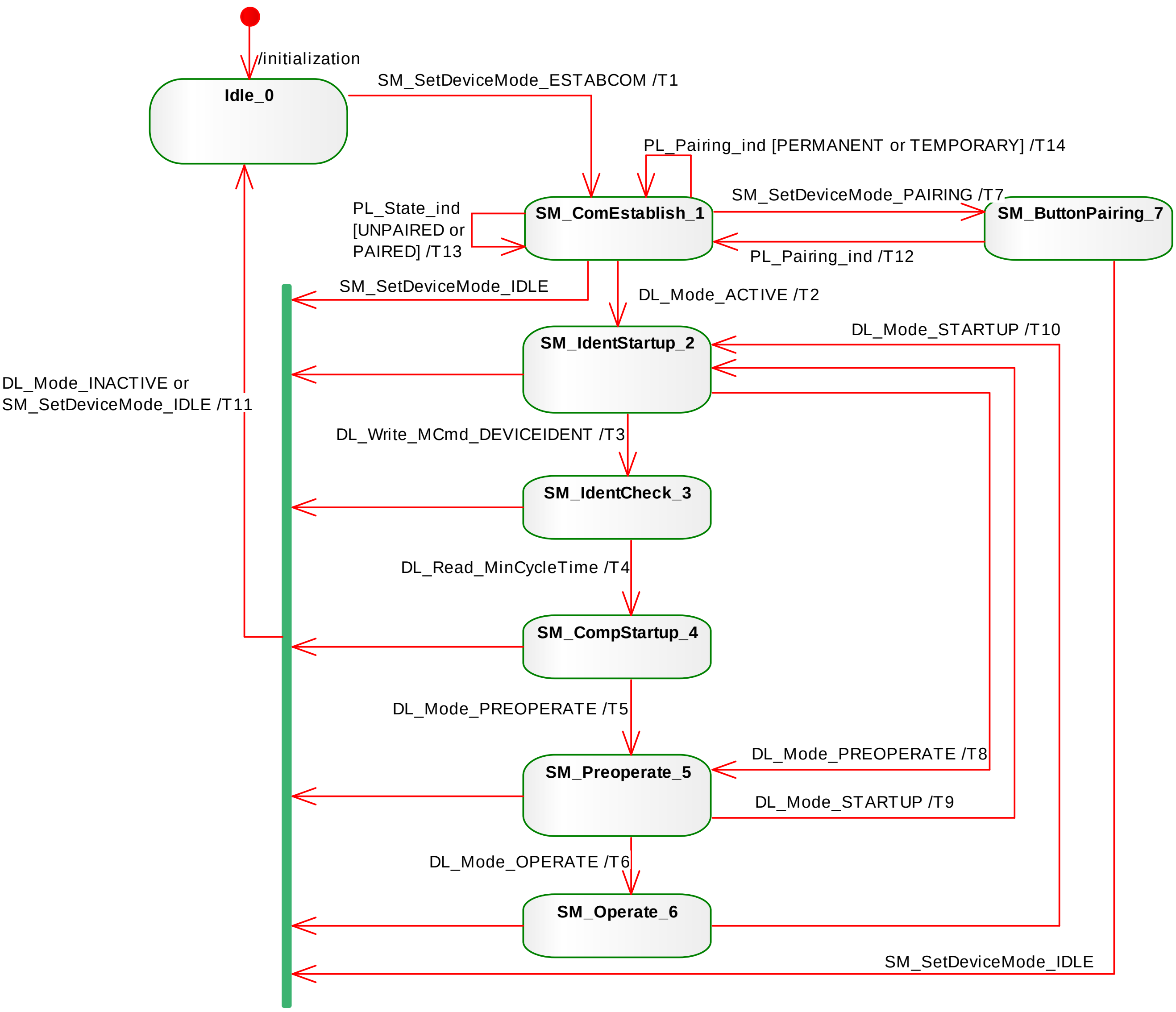


Figure 106 State machine for W-Device System Management

Table 120 State transition tables of the W-Device System Management

STATE NAME	STATE DESCRIPTION
Idle_0	In SM_Idle the SM is waiting for configuration by the W-Device application. The state is left on receiving a SM_SetDeviceMode(ESTABCOM) request from the W-Device application. The following sequence of services shall be executed between W-Device application and SM: Invoke SM_SetDeviceCom(initial parameter list) Invoke SM_SetDeviceIdent(VID, initial DID, FID)
Com_Establish_1	In SM_ComEstablish the SM is waiting for the communication to be established. The state is left on a DL_Mode (ACTIVE) from DL-mode handler, if the W-Device is connected to W-Master. In case of no connection (the W-Device is out of range or not paired) this state is kept. In this state, it is possible to pair the W-Device only via UniqueID triggered by W-Master
IdentStartup_2	In this state the communication parameter (Direct Parameter page 1, addresses 0x02 to 0x06) are read by the W-Master SM via DL_Read requests. In SM_IdentStartup the identification data (VID, DID, FID) are read and verified by the W-Master. In case of incompatibilities the W-Master SM writes the supported Revision (RID) and configured DeviceID (DID) to the W-Device. The state is left upon reception of a DL_Mode(INACTIVE), a DL_Mode(PREOPERATE) indication (compatibility check passed), or a DL_Write(MCmd_DEVICEIDENT) request (new compatibility requested).
IdentCheck_3	In SM_IdentCheck the SM waits for new initialization of identification parameters by application. The state is left on receiving a DL_Mode(INACTIVE) indication or a DL_Read(Direct Parameter page 1, addresses 0x02 = "MinCycleTime") request. Within this state the W-Device application shall check the RID and DID parameters from the SM and set these data to the supported values. Therefore, the following sequence of services shall be executed between W-Device application and SM. Invoke SM_GetDeviceCom(configured RID, parameter list) Invoke SM_GetDeviceIdent(configured DID, parameter list) Invoke W-Device application checks and provides compatibility function and parameters Invoke SM_SetDeviceCom(new supported RID, new parameter list) Invoke SM_SetDeviceIdent(new supported DID, parameter list)
CompStartup_4	In SM_CompStartup the communication and identification data are reread and verified by the W-Master SM. The state is left on receiving a DL_Mode(INACTIVE) or a DL_Mode(PREOPERATE) indication.
Preoperate_5	During SM_Preoperate the SerialNumber can be read and verified by the W-Master SM, as well as Data Storage and W-Device parameterization may be executed. The state is left on receiving a DL_Mode(INACTIVE), a DL_Mode(STARTUP) or a DL_Mode(OPERATE) indication.
Operate_6	During SM_Operate the cyclic Process Data exchange and acyclic On-request Data transfer are active. The state is left on receiving a DL_Mode(INACTIVE) or a DL_Mode(STARTUP) indication.
ButtonPairing_7	In SM_ButtonPairing the SM is waiting for Pairing by Button.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*The W-Device is switched to the communication mode by receiving the trigger SM_SetDeviceMode(ESTABCOM) by application.* Invoke SM_DeviceMode(ESTABCOM). Invoke DL_SetParam(initial parameter list) Invoke PL_SetMode(START).
T2	1	2	The W-Device application receives an indication that the communication has been established by receiving the trigger DL_Mode.ind(ACTIVE). Invoke SM_DeviceMode(IDENTSTARTUP)
T3	2	3	The W-Device identity check phase is entered by receiving the trigger DL_Write.ind(MCmd_DEVICEIDENT). Invoke SM_DeviceMode(IDENTCHANGE)
T4	3	4	The W-Device compatibility startup phase is entered by receiving the trigger DL_Read.ind(Direct Parameter page 1, address 0x02 = "MinCycleTime").
T5	4	5	The W-Device's preoperate phase is entered by receiving the trigger DL_Mode.ind(PREOPERATE). Invoke SM_DeviceMode(PREOPERATE)
T6	5	6	The W-Device's operate phase is entered by receiving the trigger DL_Mode.ind(OPERATE). Invoke SM_DeviceMode(OPERATE)
T7	1	7	The W-Device is switched to the pairing by button mode by receiving the trigger SM_SetDeviceMode(PAIRING_BUTTON) from W-Device application. Invoke PL_Pairing(PAIRING_BUTTON)
T8	2	5	The W-Device's preoperate phase is entered by receiving the trigger DL_Mode.ind(PREOPERATE). Invoke SM_DeviceMode(PREOPERATE)
T9	5	2	The W-Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP)
T10	6	2	The W-Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP)
T11	2, 3, 4, 5, 6	0	The W-Device is switched to SM_Idle mode by receiving the trigger DL_Mode.ind(INACTIVE) or SM_SetDeviceMode(IDLE). Invoke PL_SetMode(STOP) Invoke SM_DeviceMode(IDLE)
T12	7	1	The transition is triggered by PL_Pairing.ind(TIMEOUT) or PL_Pairing.ind(PERMANENT) Invoke SM_DeviceMode(ESTABCOM). Invoke SM_DeviceMode(TIMEOUT or PERMANENT).
T13	1	1	Invoke SM_DeviceMode.ind(PAIRED or UNPAIRED) to indicate PL-State after startup
T14	1	1	Invoke SM_DeviceMode.ind(PERMANENT or TEMPORARY) to indicate PL-State after pairing

9.3.3.3 State Machine of W-Device AHT handler

Figure 107 shows the main state machine of the W-Device AHT-handler. The handler manages the adaptive hopping table mechanism, see clause 18.4.

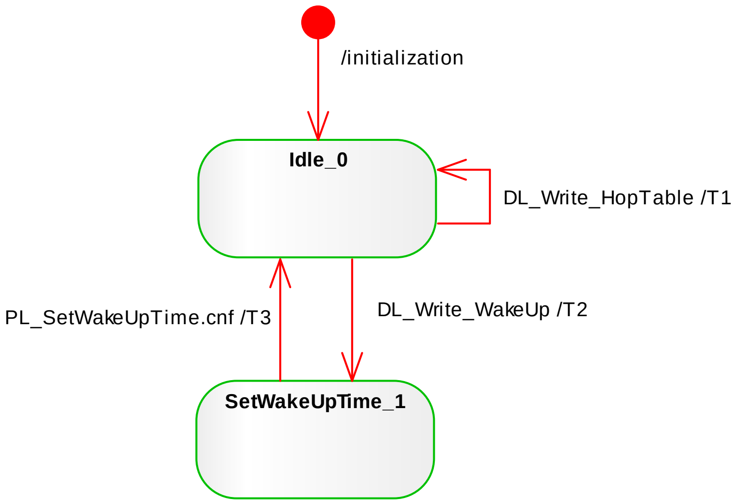


Figure 107 State Machine of the W-Device AHT-handler

Table 121 State transition table of the W-Device AHT-handler

STATE NAME	STATE DESCRIPTION
Idle_0	-
SetWakeUpTime_1	Set the WakeUpTime of low energy W-Device.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_SetHopTable(HopTable)
T2	1	2	Invoke PL_SetWakeUpTime(WakeUpTime)
T3	2	2	See T1

9.3.3.4 Start-up and Synchronization

Figure 108 shows a typical sequence chart for the SM communication startup of a W-Device matching the W-Parameter of the W-Master port (regular startup).

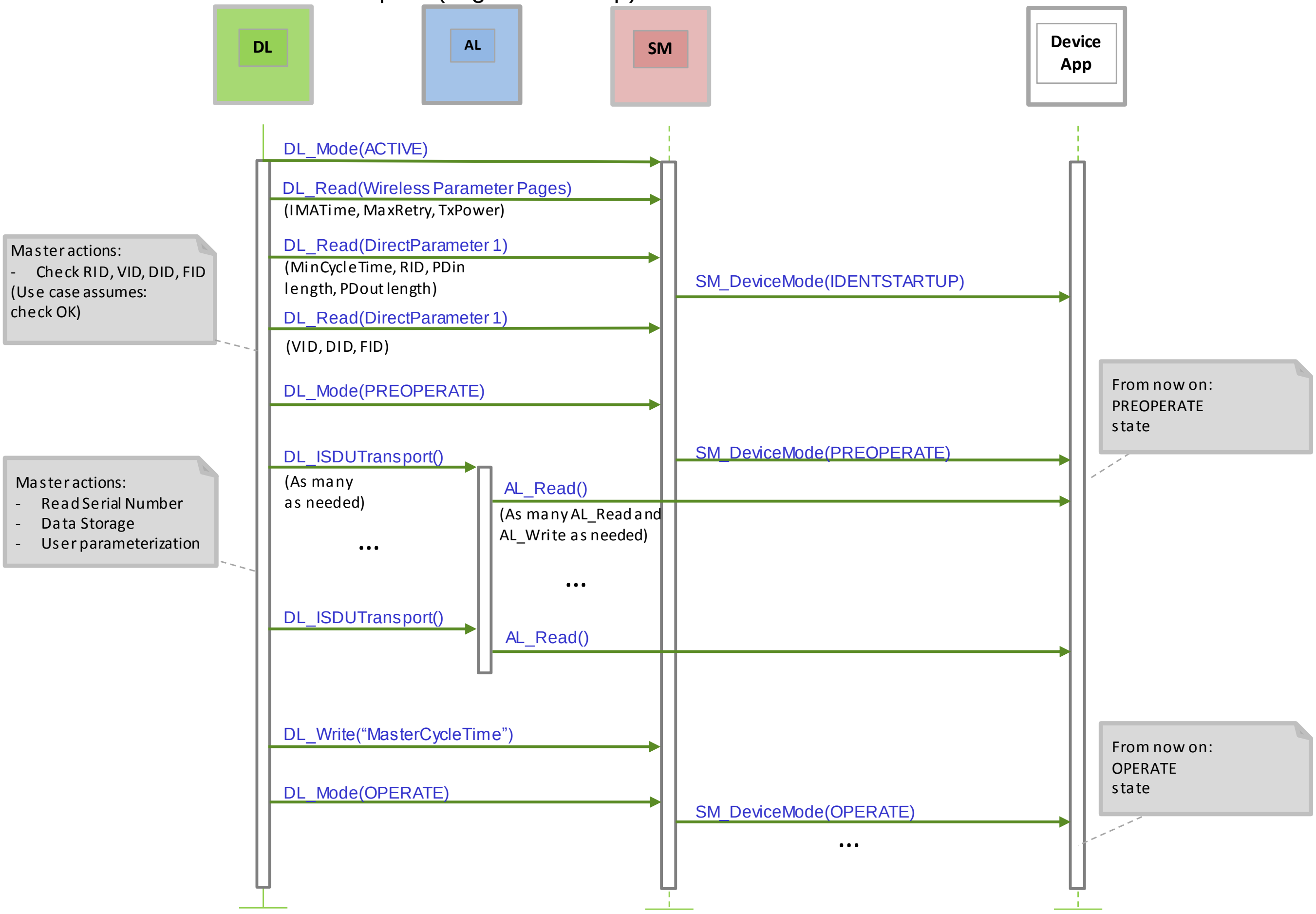


Figure 108 Sequence chart of a regular W-Device startup

Figure 109 shows a typical sequence chart for the SM communication startup of a W-Device not matching the W-Parameter of the W-Master port (compatibility mode). In this mode, the W-Master tries to overwrite the W-Device's identification parameters to achieve a compatible and a workable mode.
The sequence chart in Figure 109 shows only the actions until the PREOPERATE state. The remaining actions until the OPERATE state can be taken from Figure 106.

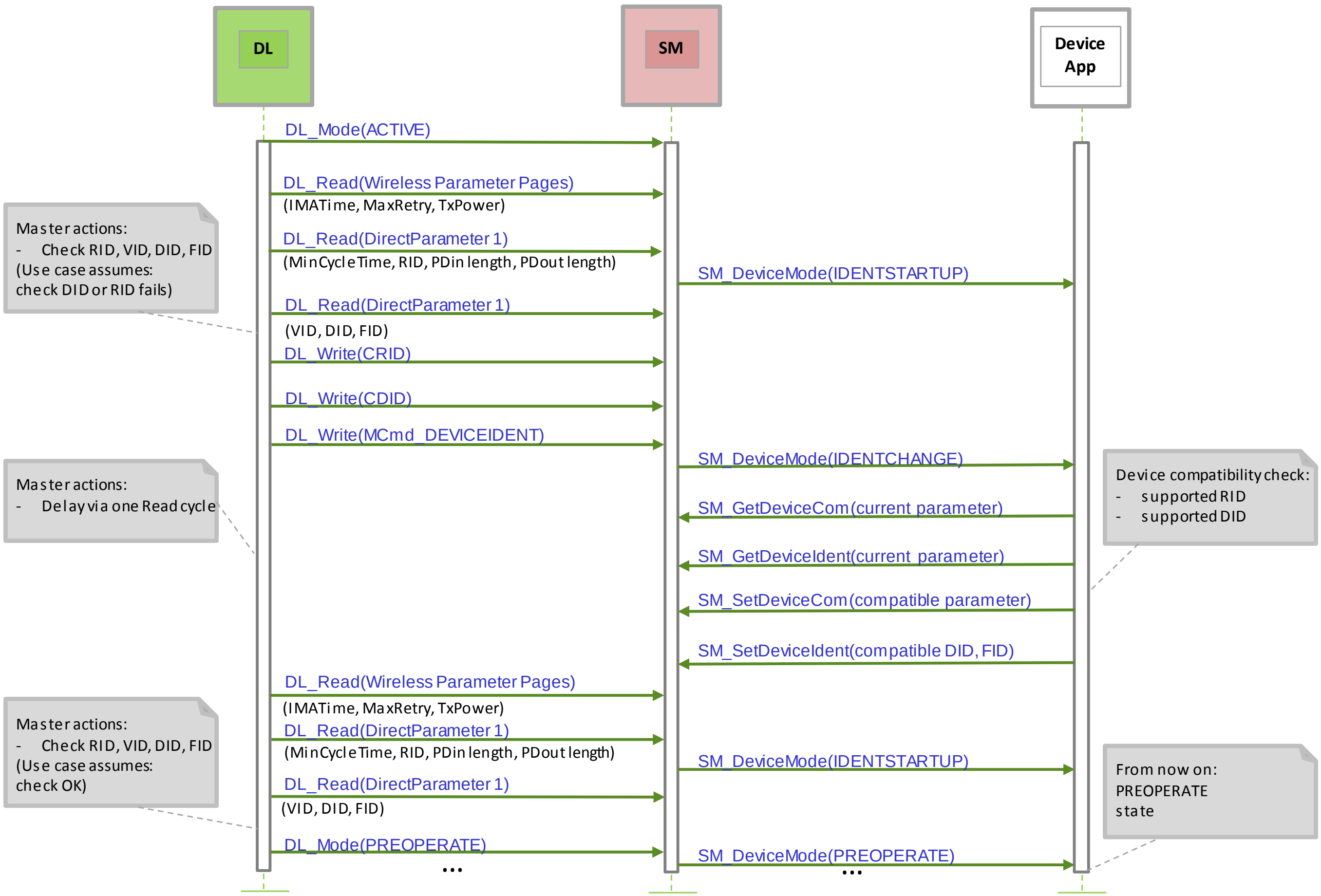


Figure 109 Sequence chart of a Device startup in compatibility mode

Figure 110 shows a typical sequence chart for the SM communication startup of a W-Device not matching the W-Master port. The system management of the W-Master tries to reconfigure the W-Device with alternative W-Device identification parameters (compatibility mode). In this use case, the alternative parameters are assumed to be incompatible.

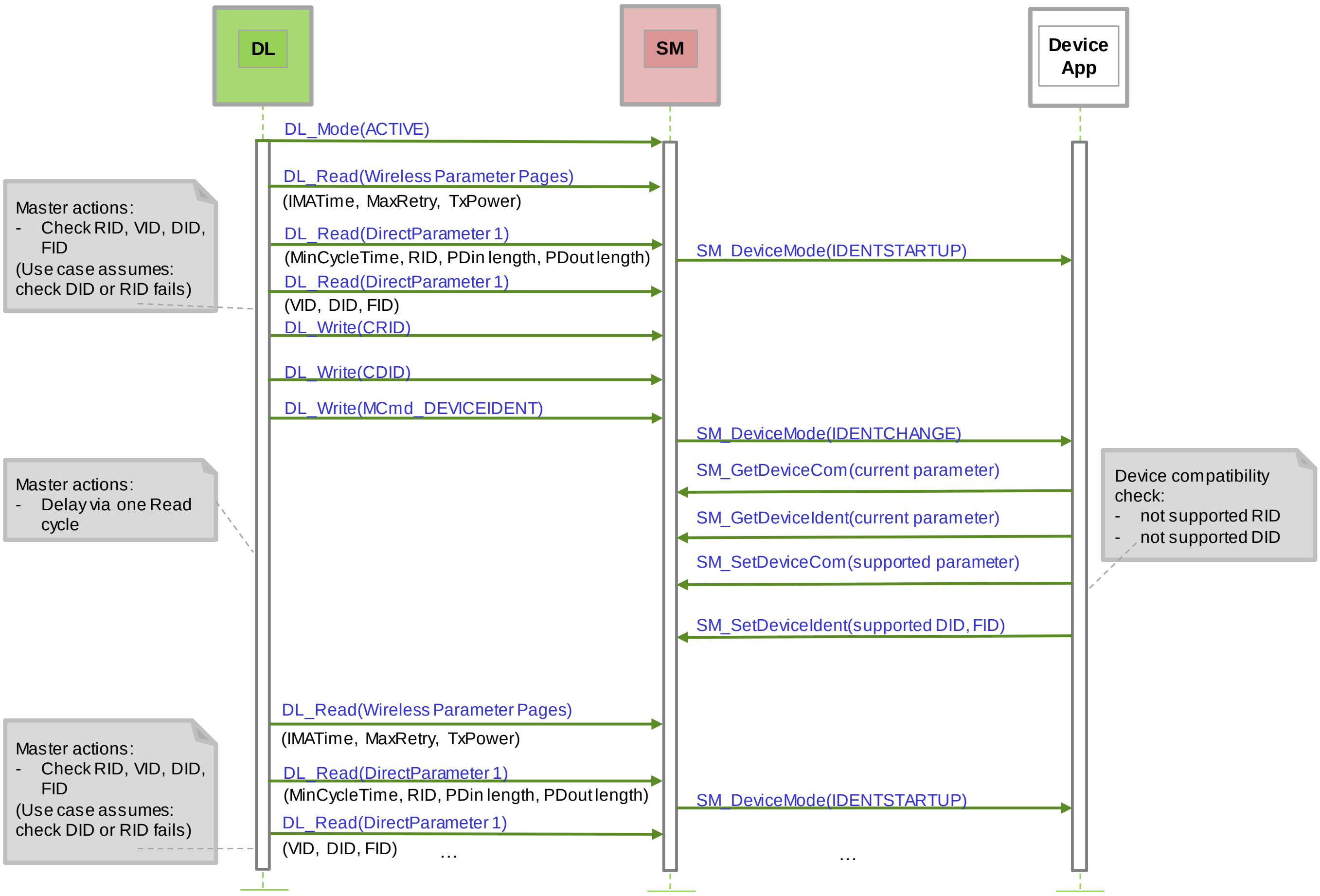


Figure 110 Sequence chart of a Device startup when compatibility fails

10 W-Device

10.1 Overview

Figure 111 provides an overview of the complete structure and services of a W-Device.

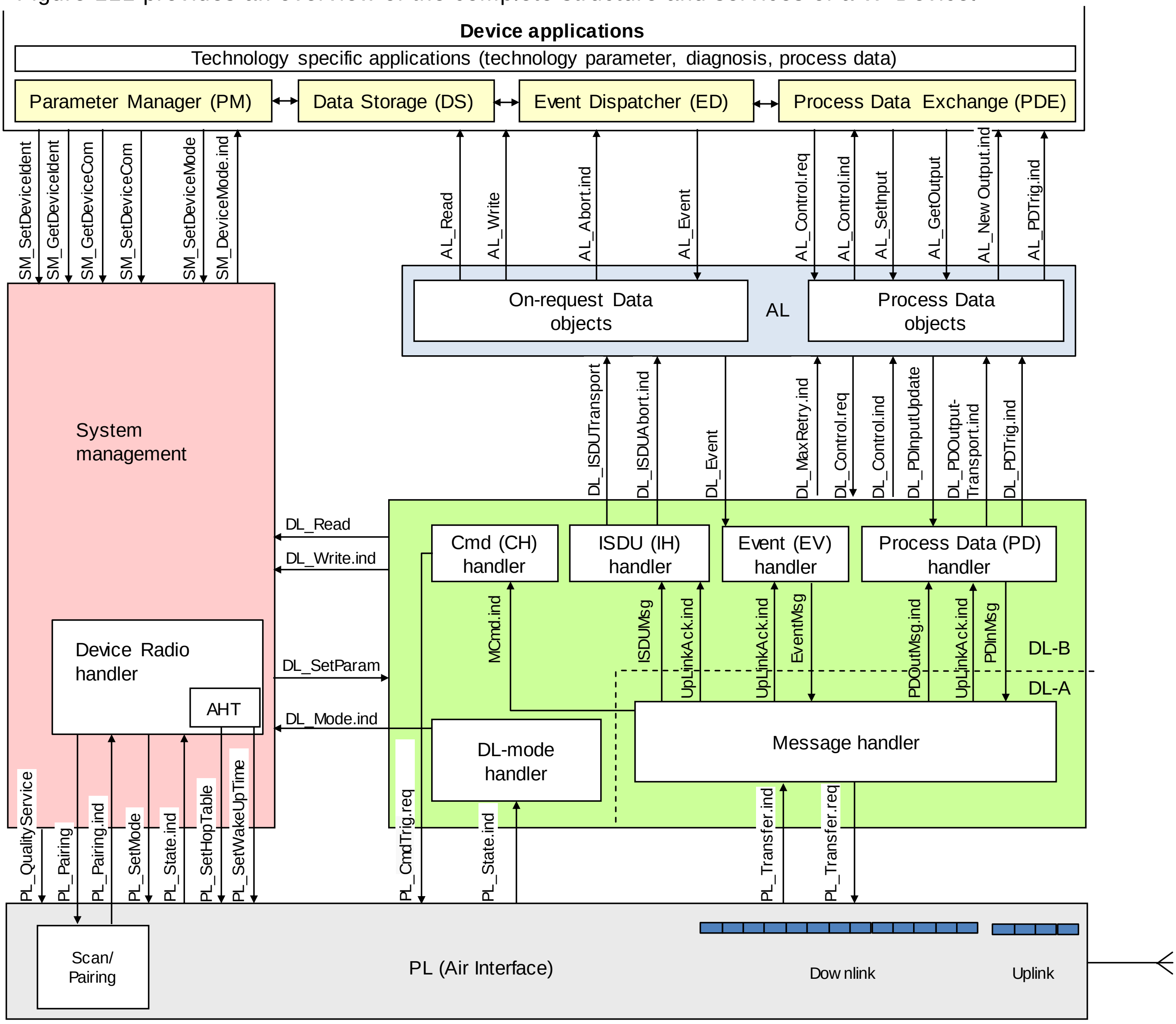


Figure 111 Structure and services of a W-Device

The W-Device applications comprise first the technology specific application consisting of the radio physical and medium access layer (PL) with its technology parameters, its diagnosis information, and its Process Data. The common W-Device applications comprise:

- Parameter Manager (PM), dealing with compatibility and correctness checking of complete sets of technology (vendor) specific and common system parameters (see 10.3);
- Data Storage (DS) mechanism, which optionally uploads or downloads parameters to the W-Master (see 10.4);
- Event Dispatcher (ED), supervising states and conveying diagnosis information such as notifications, warnings, errors, and W-Device requests as peripheral initiatives (see 10.5);
- Process Data Exchange (PDE) unit, conditioning the data structures for transmission in case of a sensor or preparing the received data structures for signal generation. It also controls the operational states to ensure the validity of Process Data (see 10.2).

10.2 Process Data Exchange (PDE)

The Process Data Exchange unit transmits and receives Process Data without interference from the On-request Data (parameters, commands, and Events), given by the priority in the W-Master and W-Device Message handler (see 6.5.2 and 6.5.4)
Due to the continuous transmission of DLinks (W-Device synchronization with or without data) and “I’m alive” ULinks (see 10.3) from W-Device to a W-Master, a transmission of Process Data is only necessary if they change.

An actuator (output Process Data) shall observe the transmission and enter a default appropriate state, for example keep last value, stop, or de-energize, whenever the data transmission is interrupted (COMLOST, see 7.2.3 and 10.7.3). The Process Data of an actuator automatically become valid, if the W-Master’s sends Process Data (see 7.4.1) prior to regular operation after restart in case of an interruption.

NOTE: A transmission of output Process Data is only possible, if the W-Master’s Process data handler is enabled via PDOUT_VALID.

Within DLinks, an actuator (output Process Data) receives a W-Message "Process Data Out Invalid" (see 12.3.1 DLink Control Octet), whenever the output Process Data are invalid and receives a W-Message with new Process Data, whenever they become valid again.

There is no need for a sensor W-Device (input Process Data) to monitor the data exchange. However, if the W-Device is not able to guarantee valid Process Data, the PD status "Process Data In invalid" shall be signaled to the W-Master application via the W-Message “Process Data In Invalid” (see 12.4.1 ULink Control Octet).

Each W-Cycle shall be used to transmit process data, while retransmits shall be used for acyclic On-request Data, if retry / retries for process data are not necessary. It is also possible to transmit On-request Data in a W-Cycle if no Process Data have to be sent.

10.3 IMA handling

If there are no W-Messages to transmit (e.g. no process data change for long time), the W-Device shall send an IMA message before the IMA time will be reached (see Figure 63). If IMA time is exceeded (monitored by the W-Master, see Figure 60), a communication error shall be reported via W-Master’s system management. Also, the PL_State service reports a COMLOST for this W-Port to the W-Master’s system management.

10.4 Parameter Manager (PM)

10.4.1 General

A W-Device can be parameterized by using the Direct Parameters or the Index memory space. In IO-Link wireless both are accessible by the help of ISDUs (see Figure 11 and clause 14).

Mandatory for all W-Devices are the Direct Parameters in page 1. Page 1 contains common communication and identification parameters (see Table 164).

For IO-Link Wireless additional mandatory parameters have been defined, which are listed in Table 168 (Index 0x5000 to 0x50FF). These parameters contains the necessary information for the wireless connection and represents an extension of the Parameter Page 1. Access to these parameters is performed via AL_Read and AL_Write.

Direct Parameter page 2 optionally offers space for a maximum of 16 octets of technology (vendor) specific parameters. Access to the Direct Parameter page 2 is performed via AL_Read and AL_Write.

The transmission of parameters to and from the spacious Index memory can be performed in two ways: single parameter by single parameter or as a block of parameters. Single parameter transmission as specified in 10.4.4 is secured via several checks and confirmation of the transmitted parameter. A negative acknowledgement contains an appropriate error description and the parameter is not activated. Block parameter transmission as specified in 10.4.5 defers parameter consistency checking and activation until after the complete transmission. The W-Device performs the checks upon reception of a special command and returns a confirmation or a negative acknowledgement with an appropriate error description. In this case the transmitted parameters shall be rejected and a roll back to the previous parameter set shall be performed to ensure proper functionality of the W-Device.

10.4.2 Parameter manager state machine

See IO-Link specification 10.3.2 in REF 1.

10.4.3 Dynamic parameter

See IO-Link specification 10.3.3 in REF 1.

10.4.4 Single parameter

See IO-Link specification 10.3.4 in REF 1.

10.4.5 Block parameter

See IO-Link specification 10.3.5 in REF 1.

10.4.6 Concurrent parameterization access

See IO-Link specification 10.3.6 in REF 1.

10.4.7 Command handling

See IO-Link specification 10.3.7 in REF 1.

10.5 Data Storage (DS)

10.5.1 General

See IO-Link specification 10.4.1 in REF 1.

10.5.2 Data Storage state machine

See IO-Link specification 10.4.2 in REF 1.
Use Table 181 for "DS_UPLOAD_REQ" Event instead Table D.2 from IO-Link.

10.5.3 DS configuration

See IO-Link specification 10.4.3 in REF 1.

10.5.4 DS memory space

See IO-Link specification 10.4.4 in REF 1.

10.5.5 DS Index_List

See IO-Link specification 10.4.5 in REF 1.

10.5.6 DS parameter availability

See IO-Link specification 10.4.6 in REF 1.

10.5.7 DS without ISDU

The ISDU-Mechanism is mandatory for W-Device. To support IO-Link Devices without ISDU via a W-Bridge, see IO-Link specification 10.4.7 in REF 1 anyway.

10.5.8 DS parameter change indication

See IO-Link specification 10.4.8 in REF 1.

10.6 Event Dispatcher (ED)

Any of the W-Device applications can generate predefined system status information when SDCI operations fail or technology specific information (diagnosis) as a result from technology specific diagnostic methods occur. The Event Dispatcher turns this information into an Event according to the definitions in 12.11. The Event consists of an EventQualifier indicating the properties of an incident and an EventCode ID representing a description of this incident together with possible remedial measures. Table 180 comprises a list of predefined IDs and descriptions for application oriented incidents. Ranges of IDs are reserved for profile specific and vendor specific incidents. Table 181 comprises a list of predefined IDs for SDCI specific incidents.

Events are classified in "Errors", "Warnings", and "Notifications". See 10.2 for these classifications and see 11.5 for how the W-Master is controlling and processing these Events.

The Event Dispatcher handles each Event one by one and each Event is acknowledged with a single command (DLink Control Octet, see 12.3.1) from W-Master to W-Device.

10.7 W-Device features

10.7.1 General

The following W-Device features are defined to a certain degree in order to achieve a common behavior. They are accessible via standardized or W-Device specific methods or parameters. The availability of these features is defined in the IODD of a W-Device, except Pairing by Button.

10.7.2 Scan

This feature enables the detection of unpaired W-Device's within a W-Master's proximity during commissioning or for Roaming, see 5.6.1.5.
This mandatory functionality is supported by the PL of the W-Device (see 5.6).

10.7.3 Pairing by UniqueID

This feature enables the pairing of an unpaired W-Device to a W-Master Port by a pairing request via the W-Device's UniqueID (see 4.4.2.1 and 5.6.1.3).
This mandatory functionality is supported by the PL of the W-Device (see 5.6).

10.7.4 Pairing by Button / Re-Pairing

This feature enables a W-Device to use the "pairing by Button" mechanism (see Figure 49). The mechanism is predominantly used to change a damaged W-Device without the need of a Port and Device Configuration Tool" (PDCT).
It is also possible to pair a W-Device to an unused, preconfigured W-Port during commissioning phase. Therefore, a W-Port configuration is needed by the W-Master Application, see 9.2.2.7 SM_SetPortConfig.
The "Pairing-Button" or a similar trigger is mandatory for a W-Device. An overview for pairing by Button or Re-pairing by Button is given in 4.4.2.2 and 4.4.2.3.
Therefore, the PL of a W-Device shall store the ConnectionParameter (see Table 140) in non-volatile memory (e.g. flash memory).
This mandatory functionality is supported by the PL of the W-Device (see 5.6).

10.7.5 Roaming

This feature is used to pair a W-Device temporary to a W-Master, to allow predefined W-Device mobility between multiple predefined W-Masters (see 4.4.4 and 5.6.1.5).
Therefore, the PL of a W-Device shall store the ConnectionParameter (see Table 140) in volatile memory (e.g. RAM memory). In case of a terminated or lost connection, the W-Device is available for other W-Master's.

This mandatory functionality is supported by the PL of the W-Device (see 5.6).

10.7.6 Unpairing

This feature removes a paired or connected roaming W-Device from a W-Master port. The PL of the W-Device shall clear its ConnectionParameter (see Table 140).

This mandatory functionality is supported by the PL of the W-Device (see 5.6).

10.7.7 W-Device backward compatibility

This feature enables a W-Device to play the role of a previous W-Device revision. In the start-up phase the W-Master system management overwrites the W-Device's inherent DeviceID (DID) with the requested former DeviceID. The W-Device's technology application shall switch to the former functional sets or subsets assigned to this DeviceID. W-Device backward compatibility support is optional for a W-Device.

As a W-Device can provide backward compatibility to previous DeviceIDs (DID), these compatible Devices shall support all parameters and communication capabilities of the previous W-Device ID. Thus, the W-Device is permitted to change any communication or identification parameter in this case.

Since the UniqueID of a W-Device contains the DeviceID (see 0), an overwrite of the DeviceID shall NOT lead in an update of the UniqueID.

10.7.8 Protocol revision compatibility

This feature enables a W-Device to adjust its protocol layers to a previous IOLW protocol version. In the start-up phase the W-Master system management can overwrite the W-Device's inherent protocol RevisionID (RID) in case of discrepancy with the RevisionID supported by the W-Master. Revision compatibility support is optional for a W-Device.

10.7.9 Factory settings

This feature enables a W-Device to restore parameters to the original delivery status. The Data Storage flag and other dynamic parameters such as "Error Count" (see B.2.17 in REF 1), "Device Status" (see B.2.18 in REF 1), and "Detailed Device Status" (see B.2.19 in REF 1) shall be reset when this feature is applied. This does not include vendor specific parameters such as for example counters of operating hours.

NOTE In this case an existing stored parameter set within the W-Master will be automatically downloaded into the W-Device after its start-up.

It is the vendor's responsibility to guarantee the correct function under any circumstances. The reset is triggered by the reception of the SystemCommand "Restore factory settings" (see Table 166). Reset to factory settings is optional for a W-Device.

10.7.10 Application reset

This feature enables a W-Device to reset the technology specific application. It is especially useful whenever a technology specific application has to be set to a predefined operational state without communication interruption and a shut-down cycle. The reset is triggered by the reception of a SystemCommand "Application reset" (see Table 166). Reset of the technology specific application is optional for a W-Device.

10.7.11 W-Device reset

This feature enables a W-Device to perform a "warm start". It is especially useful whenever a W-Device has to be reset to an initial state such as power-on. In this case communication will be interrupted. The warm start is triggered by the reception of a SystemCommand "W-Device reset" (see Table 166). Warm start is optional for a W-Device.

10.7.12 Device human machine interface (HMI)

This feature indicates the operational state of the W-Device's communication interface or the W-Device state itself. The indication of the modes is specified in 10.10.3.1. The indication is optional but highly recommended for a W-Device.

The mandatory “Pairing-Button” supports pairing, re-pairing and further optional functions, see 10.10.3.2.

10.7.13 Parameter access locking

This feature enables a W-Device to globally lock or unlock write access to all writeable W-Device parameters accessible via the wireless interface (see B.2.4 in REF 1). The locking is triggered by the reception of a system parameter "Device Access Locks" (see Table 168). The support for these functions is optional for a W-Device.

10.7.14 Data Storage locking

Setting this lock will cause the "State_Property" (Table B.10 in REF 1) to switch to "Data Storage locked" and the W-Device not to send a DS_UPLOAD_REQ Event. The support for this function is mandatory for a W-Device if the Data Storage mechanism is implemented.

10.7.15 W-Device parameter locking

Setting this lock will disable overwriting W-Device parameters via on-board control or adjustment elements such as teach-in buttons (see B.2.4 in REF 1). The support of this function is optional for a W-Device.

10.7.16 W-Device user interface locking

Setting this lock will disable the operation of on-board human machine interface displays and adjustment elements such as teach-in or pairing button(s) on a W-Device (see B.2.4 in REF 1). The support for this function is optional for a W-Device.

10.7.17 Data Storage concept

The Data Storage mechanism in a W-Device allows to automatically save parameters in the Data Storage server of the W-Master and to restore them upon Event notification. Data consistency is checked in either direction within the W-Master and W-Device. Data Storage mainly focuses on configuration parameters of a W-Device set up during commissioning (see 10.5 and 11.3). The support of this function is optional for a W-Device.

10.7.18 Block Parameter

The Block Parameter transmission feature in a W-Device allows transfer of parameter sets from a PLC program without checking the consistency single data object by single data object. The validity and consistency check is performed at the end of the Block Parameter transmission for the entire parameter set. This function mainly focuses on exchange of parameters of a W-Device to be set up at runtime (see 10.4). The support of this function is optional for a W-Device.

10.8 W-Device design rules and constraints

10.8.1 General

In addition to the protocol definitions in form of state, sequence, activity, and timing diagrams some more rules and constraints are required to define the behavior of the Devices. An overview of the major protocol variables scattered all over the standard is concentrated in Table 122 with associated references.
For additional design rules of low energy W-Devices see clause 17.

10.8.2 Process Data

The process communication channel transmits the Process Data without any interference of the On-request Data communication channels. Process Data exchange starts automatically whenever the W-Device is switched into the OPERATE state via message from the W-Master.
The format of the transmitted data is W-Device specific and varies from no data octets up to 32 octets in each communication direction.
Recommendations:
- Data structures should be suitable for use by PLC applications.
- It is highly recommended to comply with the rules in E.3.3 in REF 1 and in REF 3.

See 10.2, 12.3.1 and 12.4.1 for details on the indication of valid or invalid Process Data via the transmission of Process Data (PDx_Valid) within the data exchange.

10.8.3 MaxRetry error detection

It is the responsibility of the W-Device designer to define the appropriate behavior of the W-Device in case communication with the W-Master exceeds the configured maximum Retries for a data transmission (transition T4 in Figure 56 handles detection of the MaxRetry error, reported via AL_Control (MaxRetry) to the W-Device Application). This Error indicates that the configured cycle time has not been kept, e.g. a W-Device is at the edge of the RF coverage area.
If the AL_Control reports a MaxRetry error, the Device Application shall send the Event (IOLW_Retry_Error) vie event channel to the W-Master.
NOTE This is especially important for actuators such as valves or motor management.

10.8.4 Communication loss

It is the responsibility of the W-Device designer to define the appropriate behavior of the W-Device in case communication with the W-Master is lost (transition T10 in Figure 56 handles detection of the communication loss (reported via PL_State service), while 10.2 define resulting W-Device actions).

NOTE This is especially important for actuators such as valves or motor management.

10.8.5 Direct Parameter

Compared to IO-Link (using the page communication channel) a Direct Parameter access for IO-Link wireless is redirected to the ISDU communication channel, except the MasterCommand (see 7.7.1 and 7.7.2). The access to the Direct Parameter pages provides no handshake mechanism (similar to IO-Link), to ensure proper reception or validity of the transmitted parameters. The Direct Parameter page can only be accessed single octet by single octet (Subindex) or as a whole (16 octets). Therefore, the consistency of parameters larger than 1 octet cannot be guaranteed in case of single octet access.

The parameters from the Direct Parameter page cannot be saved and restored via the Data Storage mechanism.

10.8.6 ISDU communication channel

The ISDU communication channel provides a powerful means for the transmission of parameters and commands (see 14.3).

The following rules shall be considered when using this channel (see Figure 11).

- Index 0 Subindex 1 (MasterCommand) is not accessible via the ISDU communication channel.
- All other Subindices of Index 0 (Direct Parameter page 1) included Index 1 (Direct Parameter page 2) are redirected by the W-Master to the Direct Parameter page 1 / 2 using the ISDU communication channel.
- Index 3 cannot be accessed by a PLC application program. The access is limited to the W-Master application only (Data Storage).
- After reception of an ISDU request from the W-Master the W-Device shall respond within 5 000 ms (see Table 122). Any violation causes the W-Master to abandon the current task.

10.8.7 DeviceID rules related to W-Device variants

Devices with a certain DeviceID and VendorID shall not deviate in communication and functional behavior. This applies for sensors and actuators. Those Devices may vary for example in

- housing materials,
- mounting mechanisms,
- other features, and environmental conditions.

10.8.8 Protocol constants

Table 122 gives an overview of the major protocol constants for Devices.

Table 122 Overview of the protocol constants for W-Device

System variable	References	Values	Definition
ISDU acknowledgement time, for example after a SystemCommand	14.1.9	5 000 ms	Time from reception of an ISDU for example SystemCommand and the beginning of the response message of the W-Device (see Figure 75)
Maximum number of entries in Index List	B.2.3, REF 1	70	Each entry comprises an Index and a Subindex. 70 entries result in a total of 210 octets.
Preset values for unused or reserved parameters, for example FunctionID	Annex C	0 (if numbers) 0x00 (if characters)	Engineering shall set all unused parameters to the preset values.
Detection for COMLOST	0	5 * MaxRetry	ComLost is reported via PL-service PL_State, see Figure 46, T10, T12
Detection for “wireless connection synchronized”	0	3 W-Sub-cycles	SYNCED is reported via service PL_State, see Figure 46, T9 (3 sub-sequent DLinks received by W-Device)
MinCycleTime	14.1.2	N * 5 ms	W-Device or W-Bridge defines its minimum cycle time to acquire input or process output data.
Usable Index range	14.3	See Table 168	This version of the standard reserves some areas within the total range of 65535 Indices.
Errors and warnings	13.8	50 ms	An Event with MODE "Event appears" shall stay at least for the duration of this time.
EventCount	6.3.6	1	Constraint for AL_Event.req.

10.9 I/O W-Device description (W-IODD)

An IODD (I/O Device Description) is a file that formally describes an IO-Link Device. It provides all the necessary properties to establish communication and describes the Device parameters and their boundaries to establish the desired function of a sensor or actuator.

An IODD file shall be provided for each W-Device, and shall include all information necessary to support this standard.

The IODD can be used by engineering tools for PLCs and/or W-Masters for the purpose of identification, configuration, definition of data structures for Process Data exchange, parameterization, and diagnosis decoding of a particular Device. Additionally, IODDs are also used for automatic IO-Link Wireless conformance testing.

NOTE Details of the IODD language to describe a Device can be found in REF 3 and clause 21.

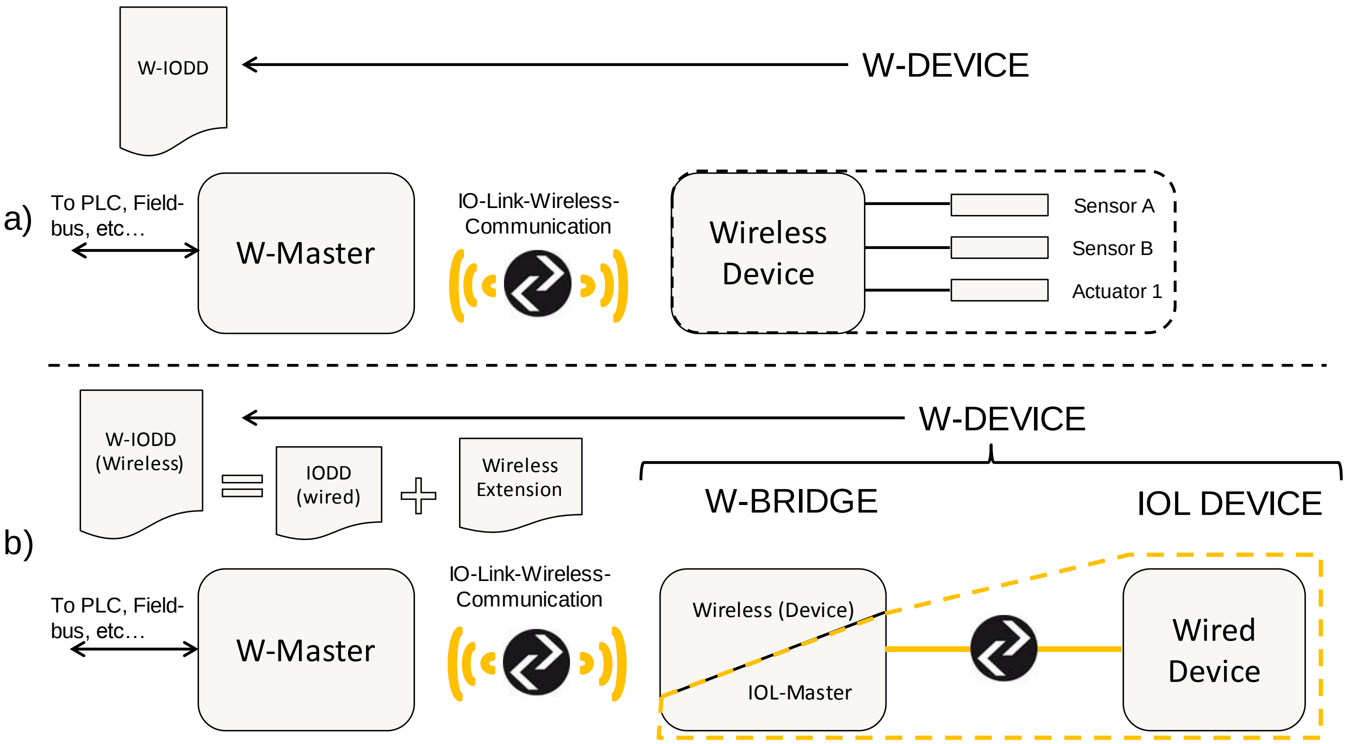


Figure 112 Schematic representation of the use of (a) a W-Device and (b) a W-Bridge to connect a wired IO-link Device.

10.9.1 CommNetwork Profile Instance for the IODD file for IO-Link wireless

This section gives an example for the content of the IODD file using “IOLink**Wireless**CommNetworkProfileT”. The wired connection part is optional and should be used to describe a W-Device equipped with wired power connection.

In the case of a W-Bridge configuration, the IODD of the wired IO-Link Device and the required extension for wireless can be merged together to constitute the W-IODD of the novel entity formed by the W-Bridge and the wired IO-Link Device. For more details see clause 21.

The wireless specific parameters which shall be used as an extension to the Direct Parameter page 1 are located from 0x5000 to 0x50FF.

In most cases the “CommNetworkProfile” of the wired IODD must be replaced by the “IOLinkWirelessCommNetworkProfileT” of the wireless IODD. The IODD description from index 0x5000 to 0x50FF must be added to the wired IODD. to get a IODD for a W-Device

In the case where the device has a wired and a wireless interface, 2 IODD files, one for the wired and another one for the wireless connection, must be used to describe it.

10.9.1.1 Example of a wireless extension:

```
<CommNetworkProfile xsi:type="IOLinkWirelessCommNetworkProfileT" iolinkRevision="V1.1">
<TransportLayer>
<PhysicalLayer minCycleTime="50" doubleSlot=”yes” isABridge=”yes”>

<Connection xsi:type="M5ConnectionT">
        <Wire1/>
        <Wire2 function="NC"/>
        <Wire3/>
        <Wire4/>
```

```
    </Connection>

</PhysicalLayer>
</TransportLayer>
</CommNetworkProfile>
```

PhysicalLayer(mandatory)

- minCycleTime (required) value is expressed in micro-second
- doubleSlot (required) value = [“yes” | “no”]
- isABridge (required) value = [“yes” | “no”]

Connection (line powered, optional)

This parameter describes the sensor’s wiring. In case of a W-Device which gets its energy line powered, it describes the type of connector and the feature of each pin.

10.10 W-Device diagnosis

10.10.1 Concepts

This standard provides only most common EventCodes in 15.1. It is the purpose of these common diagnosis information to enable an operator or maintenance person to take fast remedial measures without deep knowledge of the W-Device's technology. Thus, the text associated with a particular EventCode shall always contain a corrective instruction together with the diagnosis information.

Fieldbus-W-Master-Gateways tend to only map few EventCodes to the upper system level. Usually, vendor specific EventCodes defined via the IODD can only be decoded into readable instructions via a Port and W-Device Configuration Tool (PDCT) or specific vendor tool using the IODD.

Condensed information of the W-Device's "state of health" can be retrieved from the parameter "Device Status" (see B.2.18 in REF 1). Table 136 provides an overview of the various possibilities for Devices and shows examples of consumers for this information.

If implemented, it is also possible to read the number of faults since power-on or reset via the parameter "Error Count" (see B.2.17in REF 1) and more information in case of profile Devices via the parameter "Detailed Device Status" (see B.2.19 in REF 1).

NOTE Profile specific values for the "Detailed Device Status" are given in REF 4.

If required, it is highly recommended to provide additional "deep" technology specific diagnosis information in the form of W-Device specific parameters (see Table 168) that can be retrieved via port and W-Device configuration tools for Masters or via vendor specific tools. Usually, only experts or service personnel of the vendor are able to draw conclusions from this information.

Table 123 Classification of W-Device diagnosis incidents

Diagnosis incident	Appear/ disappear	Single shot	Parameter	Destination	Consumer
Error (fast remedy; standard EventCodes)	yes	-	-	PLC or HMI (fieldbus mapping)	Maintenance and repair personnel
Error (IODD: vendor specific EventCodes; see Table D.1)	yes	-	-	PDCT or vendor tool	Vendor service personnel
Error (via W-Device specific parameters)	-	-	See Table B.8 REF 1	PDCT or vendor tool	Vendor service personnel
Warning (fast remedy; standard EventCodes)	yes	-	-	PLC or HMI	Maintenance and repair personnel
Warning (IODD: vendor specific EventCodes; see Table D.1)	yes	-		PDCT or vendor tool	Vendor service personnel
Warning (via W-Device specific parameters)	-	-	See Table B.8 REF 1		
Notification (Standard EventCodes)	-	yes		PDCT	Commissioning personnel
Detailed W-Device status	-	-		PDCT or vendor tool	Commissioning personnel and vendor service personnel
Number of faults via parameter "Error Count"	-	-	See B.2.1 REF 1		
W-Device "health" via parameter "W-Device Status"	-	-	See B.2.18, Table B.13 REF 1	HMI, Tools such as "Asset Management"	Operator

10.10.2 Events

MODE values shall be assigned as follows (see 12.11.1):

- Events of TYPE "Error" shall use the MODEs "Event appears / disappears"
- Events of TYPE "Warning" shall use the MODEs "Event appears / disappears"
- Events of TYPE "Notification" shall use the MODE "Event single shot"

The following requirements apply:

- The Event which is already placed in the Event queue are discarded by the Event Dispatcher when communication is interrupted or cancelled.

NOTE After communication resumes, the technology specific application is responsible for proper reporting of the current Event causes.

- It is the responsibility of the Event Dispatcher to control the "Event appears" and "Event disappears" flow. Once the Event Dispatcher has sent an Event with MODE "Event appears" for a given EventCode, it shall not send it again for the same EventCode before it has sent an Event with MODE "Event disappears" for this same EventCode.
- Each Event shall use static mode, type, and instance attributes.
- Each vendor specific EventCode shall be uniquely assigned to one of the TYPEs (Error, Warning, or Notification).

In order to prevent the diagnosis communication channel (see Figure 11) from being flooded, the following requirements apply:

- The same diagnosis information shall not be reported at less than 60 s intervals, that is the Event Dispatcher shall not invoke the AL_Event service with the same EventCode more often than 60 s.
- The Event Dispatcher shall not issue an "Event disappears" less than 50 ms after the corresponding "Event appears".
- Subsequent incidents of errors or warnings with the same root cause shall be disregarded, that means one root cause shall lead to a single error or warning.
- The Event Dispatcher shall not invoke the AL_Event service with an EventCount greater than one.
- Errors are prioritized over Warnings.

10.10.3 W-Device HMI

10.10.3.1 Visual indicators

The indication of IO-Link Wireless communication on the W-Device is optional (but highly recommended). The different states shall then be implemented, see Table 124.
The IO-Link Wireless indication shall use a green indicator.

Table 124 Visual states of W-Device

W-Device state	**LED indication**	**LED-Timing** (see NOTE 1)	**Initiator for Application**
Unpaired	permanent on	LED on	SM_DeviceMode.ind(UNPAIRED)
Paired	blink	Trep=700 ms; Toff=350 ms	SM_DeviceMode.ind(PAIRED)
Connected	inverted flash	Trep=1000 ms; Toff=100 ms	SM_DeviceMode.ind(PREOPERATE) SM_DeviceMode.ind(OPERATE)
Wink	double flash	Trep=1000 ms; Ton=100 ms; Toff=100 ms	See Table 166, Reserved for profiles.

NOTE 1: The LED timings are typical values. A tolerance of 10% shall not be exceeded.

The indication of the blinking LED follows the timing shown in Figure 113.

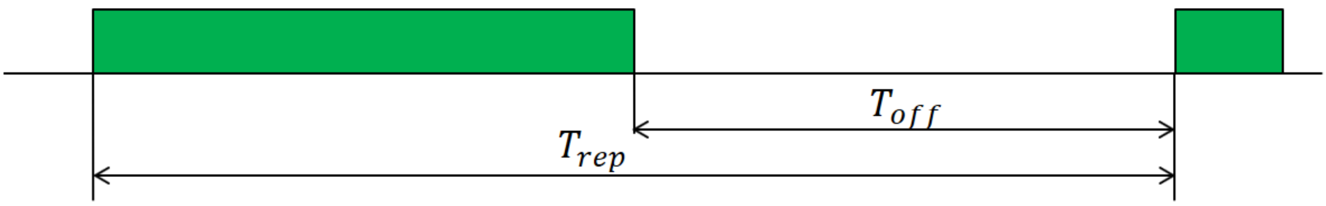


Figure 113 Device LED blink timing

The indication of the double flashing LED follows the timing shown in Figure 114.

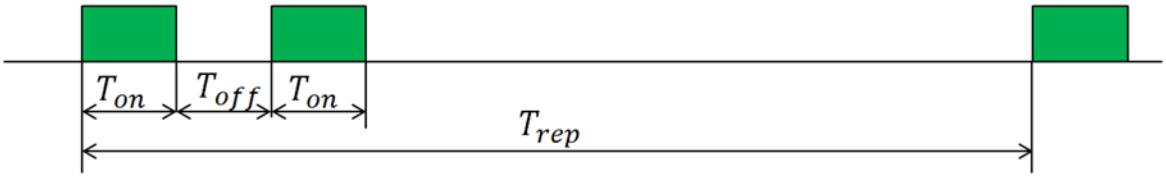


Figure 114 Device LED double flash timing.

The additional function of visual indicators for low energy W-Devices are defined in 17.1.8.

10.10.3.2 Pairing Button

The "Pairing-Button" or a similar trigger is mandatory for a W-Device. An overview for pairing by Button or Re-pairing by Button is given in 4.4.2.2 and 4.4.2.3. Further, each button press shall trigger a "HMI button pressed" Event, see Table 180 EventCodes.
The Pairing-Button supports further functions, depending on the duration of the button pressed, see Table 125.

Table 125 Pairing Button functions

Button press timing	Button press function	Remarks
[0.1…1] s	Wake up a sleeping W-Device and / or activation of the visual indicators	Highly recommended for low energy W-Devices with an internal power source.
[>1…3] s	No action	
[>3…10] s	Pairing by Button / Re-pairing by Button	mandatory for all W-Devices
[>10…30] s	No action	
>30s	Device Reset (see 10.7.11)	Highly recommended for low energy W-Devices with an internal power source.

NOTE: The button press timings are typical values. A tolerance of 10% shall not be exceeded.

10.11 W-Device connectivity

See 4.4.2 (Pairing), 4.4.3 (Unpairing) and 4.4.4 (Roaming) for the different possibilities of pairing W-Devices to W-Master ports and the corresponding mechanisms.

NOTE For compatibility reasons, this standard does not prevent W-Devices from providing additional functions.

11 W-Master

11.1 Overview

The W-Master handles the communication between the application and its associated W-Devices. The recommended relationship between the IO-Link wireless technology and a fieldbus technology was already presented in clause4.2. Even though this may be the major use case in practice, it does not automatically imply that the IO-Link wireless technology depends on the integration into fieldbus systems. It can also be directly integrated into PLC systems, industrial PC, or other control systems without fieldbus communication in between.

Figure 115 provides an overview of the complete structure and services of a W-Master. The purpose of the different layers and their service interfaces are described in the previous clauses.

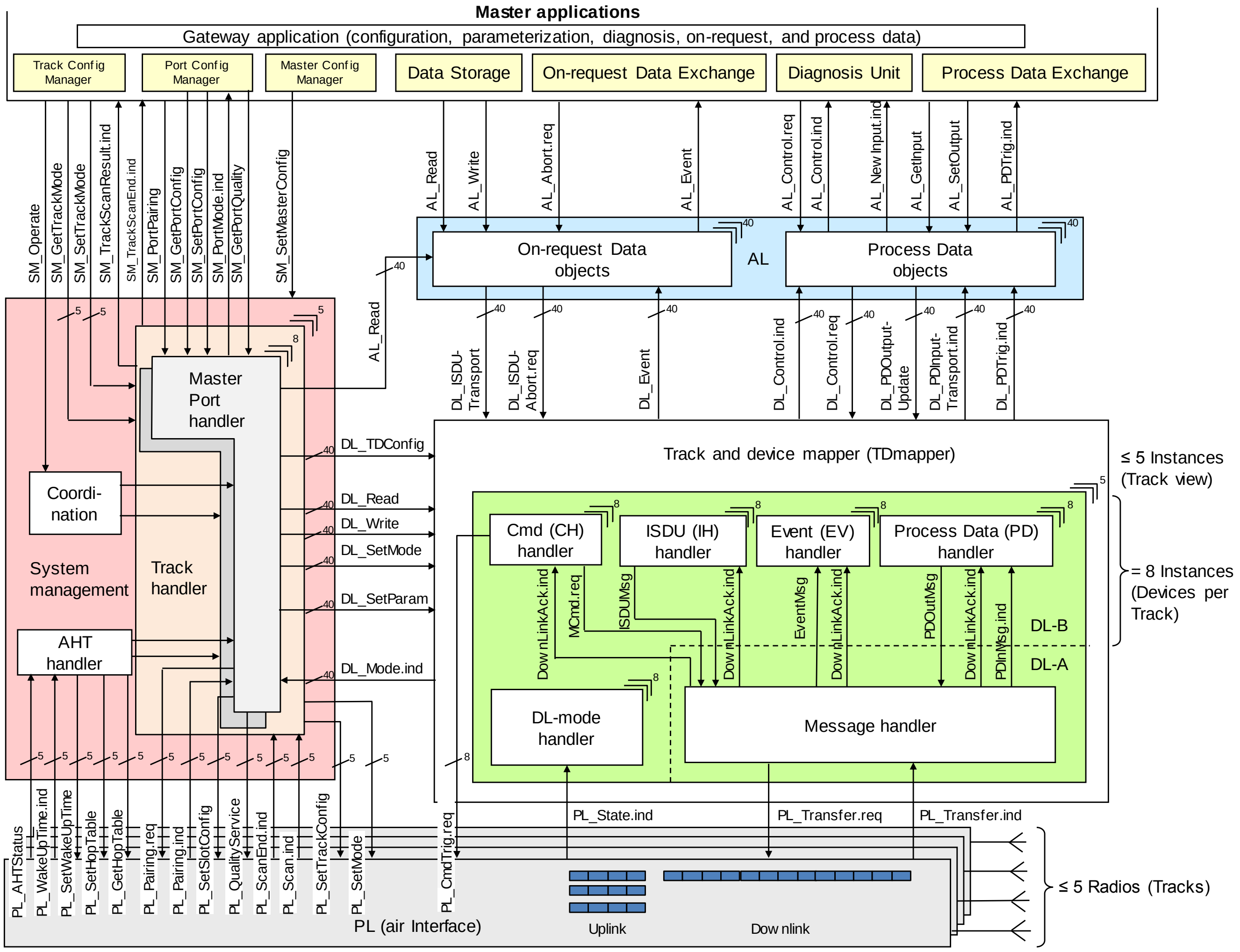


Figure 115 Structure and services of a W-Master

The W-Master applications comprise first a fieldbus specific gateway or direct connection to a PLC (host) for the purpose of start-up configuration and parameterization as well as Process Data exchange, user-program-controlled parameter change at runtime, and diagnosis propagation. For the purpose of configuration, parameterization, and diagnosis during commissioning a so-called "Port and Device Configuration Tool" (PDCT) is connected either directly to the W-Master or via fieldbus communication. These instruments are using the following common W-Master applications.

- **W-Master-, Track- and W-Port-Configuration Manager (CM),** transforms the user configuration assignments into W-Port and track set-ups (see 11.2 in REF 1);
- **Data Storage (DS)** mechanism, which can be used to save and restore the W-Device parameters (see 11.3 in REF 1);

- **Diagnosis Unit (DU)**, which routes Events from the AL to the Data Storage unit or the gateway application (see 11.4 in REF 1);
- **On-request Data Exchange (ODE)**, which provides for example acyclic parameter access (see 11.5 in REF 1);
- **Process Data Exchange (PDE)** builds the bridge to upper level automation instruments. It also controls the operational states to ensure the validity of Process Data (see 11.6 in REF 1).

These W-Master applications provide standard methods/functions to the available Services, specified in the following subclauses.

The Configuration Manager (CM) and the Data Storage mechanism (DS) need special coordination in respect to On-request Data, see Figure 116 and **Fehler! Verweisquelle konnte nicht gefunden werden.**.

The gateway application maps these functions into the features of a particular fieldbus/PLC or directly into a host system. It is not within the scope of this standard to define any of these gateway applications.

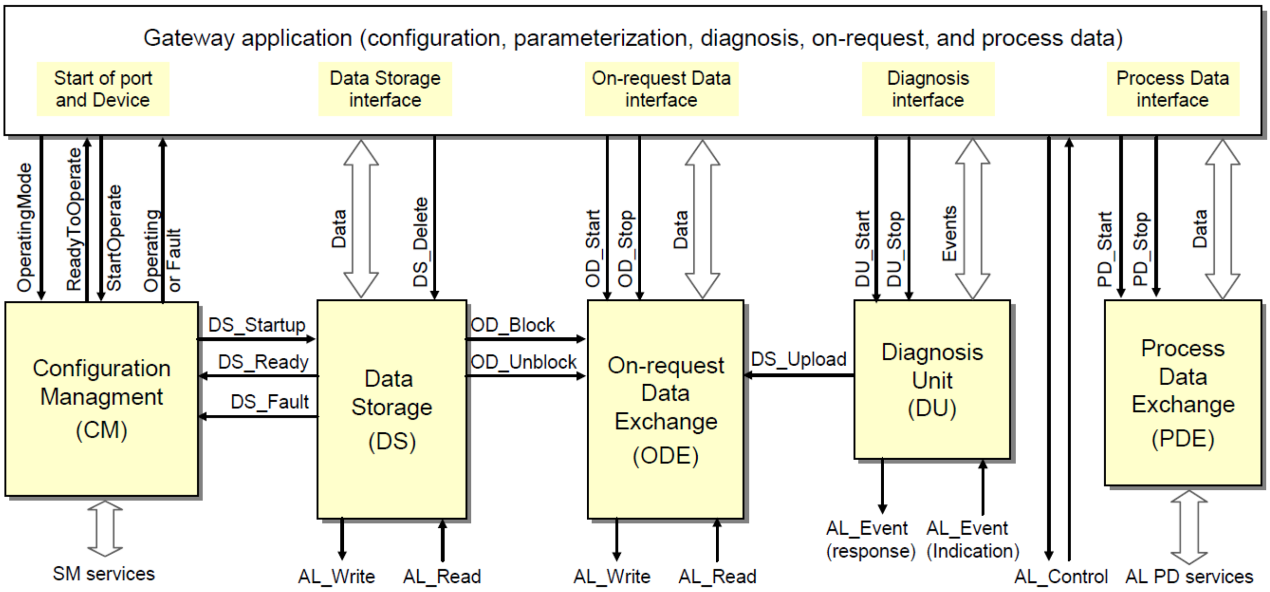


Figure 116 Relationship of the common W-Master applications

The internal variables between the common W-Master applications are specified in Table 126.. The main responsibility is assigned to the Configuration Manager (CM) as shown in Figure 116 and explained in 11.2

Table 126 Internal variables and Events to control the common W-Master applications

Internal Variable	Definition
OperatingMode	This variable activates the W-Port and provides the configuration parameters.
ReadyToOperate	This variable indicates correct configuration of the W-Port.
StartOperate	This variable allows for explicit change of all ports to the OPERATE mode.
Operating	This variable indicates all ports are in cyclic Process Data exchange mode
Fault	This variable indicates abandoned communication at any W-Port (see Figure 85 and Table 127 State transition tables of the Track Configuration Manager).
DS_Startup	This variable triggers the Data Storage (DS) state machine causing an Upload or Download of W-Device parameters if required (see 11.3).
DS_Ready	This variable indicates the Data Storage has been accomplished successfully; operating mode is CYCLIC or ROAMING (see 9.2.2.2)
DS_Fault	This variable indicates the Data Storage has been aborted due to a fault.
DS_Delete	Any verified change of W-Device configuration leads to a deletion of the stored data set in the Data Storage.
DS_Upload	This variable triggers the Data Storage state machine in the W-Master due to the special Event "DS_UPLOAD_REQ" from the W-Device.
OD_Start	This variable enables On-request Data access via AL_Read and AL_Write.
OD_Stop	This variable indicates that On-request Data access via AL_Read and AL_Write is acknowledged with a negative response to the gateway application.
OD_Block	Data Storage upload and download actions disable the On-request Data access through AL_Read or AL_Write. Access by the gateway application is denied.
OD_Unblock	This variable enables On-request Data access via AL_Read or AL_Write.
DU_Start	This variable enables the Diagnosis Unit to propagate remote (W-Device) or local (W-Master) Events to the gateway application.
DU_Stop	This variable indicates that the W-Device Events are not propagated to the gateway application and not acknowledged. Available Events are blocked until the DU is enabled again.
PD_Start	This variable enables the Process Data exchange with the gateway application.
PD_Stop	This variable disables the Process Data exchange with the gateway application.

11.2 Configuration Manager (CM)

11.2.1 General

The coordinating role of the configuration manager amongst all the common W-Master applications is described in the IO-link spec, clause 11.2.1 in REF 1. After setting up a W-Port to the assigned modes (see 11.2.2.1 through 11.2.2.3 in REF 1) CM starts the Data Storage mechanism (DS) and returns the variable "Operating" or "Fault" to the gateway application.

In case of the variable "Operating" of a particular W-Port, the gateway application activates the state machines of the associated Diagnosis Unit (DU), the On-request Data Exchange (ODE), and the Process Data Exchange (PDE).

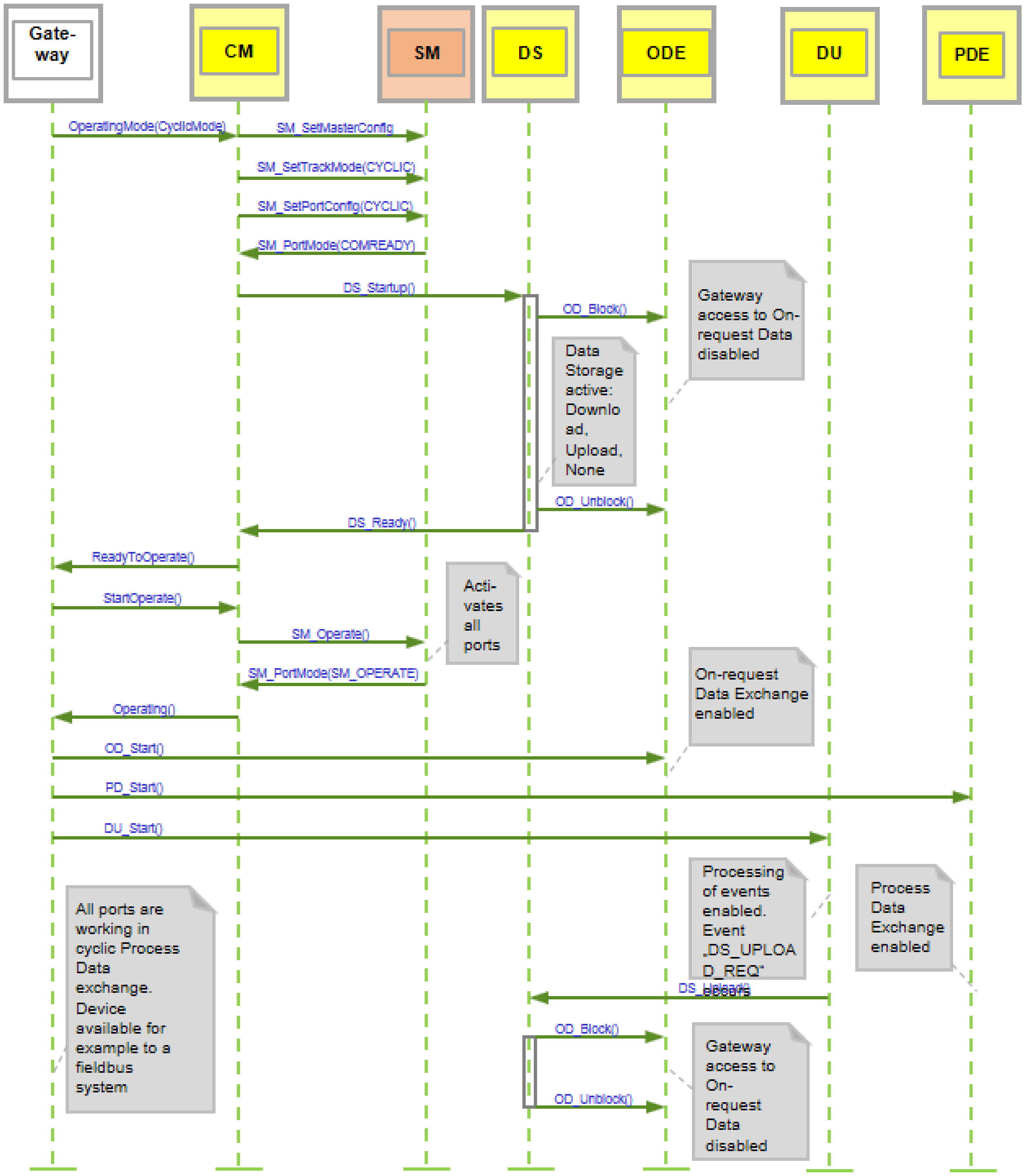


Figure 117 Sequence diagram of configuration manager actions

11.2.2 Configuration parameter

11.2.2.1 Track OperatingMode

One of the following operating modes can be selected. All modes are mandatory. The OperatingMode is significant to all W-Ports of a track see SM_SetPortConfig and SM_GetPortConfig.

Inactive

All W-Ports of the selected track are deactivated. The corresponding Process Data for input and output is zero.

CyclicMode

The track is configured for continuous cyclic communication. Process and On-request Data will be transmitted. The connection and the W-Device specific cycle time will be monitored. It is not possible to scan for unpaired W-Devices or pair W-Devices. Roaming is not supported in this mode.

ServiceMode

In addition to the cyclic communication, the configuration channels are activated on this track to support scan, pairing and roaming activities. To avoid collisions on the configuration channels, only one track of a multi-track W-Master shall be operated in the ServiceMode at the same time.

11.2.2.2 PortOperatingMode

PortInactive

The W-Master port is deactivated. The corresponding Process Data for input and output is zero.

PortCyclicMode

For operate a port in Cyclic Mode it is necessary to configure the corresponding Track in Mode or ServiceMode.

PortRoamingMode

For operate a port in PortRoamingMode it is necessary to configure one track permanently in ServiceMode

11.2.2.3 PortCycle

The process data update is performed in a cyclic manner and is determined by the W-Port related cycle time, within which the process data of the W-Device are read or written.

It is highly recommended to perform the process data update right after the read of the input data (AL_GetInput) within the same port cycle.

It is also highly recommended to write the process output data to the W-Devices within one port cycle (AL_SetOutput).

One of the following W-Port cycle modes can be selected. None of the modes is mandatory but it is highly recommended to support all modes.

FreeRunning

The W-Port cycle timing is not restricted.

FixedValue

The port cycle timing is fixed to a specific value which shall be set via service SM_SetPortConfig for all ports. If the Device is not able to achieve this timing, for example if the timing is lower than the MinCycleTime of the Device, an error shall be generated. The fixed value can be written in the ConfiguredCycleTime parameter as specified in 11.2.2.4.

11.2.2.4 Configured/Real CycleTime

This parameter contains the requested ConfiguredCycleTime or actual RealCycleTime for the specific ports. It shall be passed as a value via SM_SetPortConfig and SM_GetPortConfig. The RealCycleTime must always be equal or greater than the MinCycleTime.

11.2.2.5 PDConfig

This set of parameters contains the rules for the Process Data mapping between the W-Device Process Data stream and the gateway Process Data stream (see example in Figure 127 for the definitions).

LenIn

This parameter contains the requested length of the W-Device input ProcessDataIn Bits

PosIn

This parameter contains the offset within the gateway input Process Data stream in Bit.

SrcOffsetIn

This parameter contains the offset within the W-Device Input Process Data stream in Bit.

LenOut

This parameter contains the requested length of the W-Device output ProcessDataOut Bits.

PosOut

This parameter contains the offset within the gateway output Process Data stream in Bit.

SrcOffsetOut

This parameter contains the offset within the W-Device Output Process Data stream in Bit.

11.2.2.6 DeviceIdentification

This set of parameters contains the actual configured W-Device identification.

VendorID

This parameter contains the requested or read vendor specific ID as specified in B.1.8. in REF 1

DeviceID

This parameter contains the requested or read W-Device specific ID as specified in B.1.9. in REF 1

SerialNumber

This parameter contains the requested or read SerialNumber as specified in B.2.13. in REF 1

InspectionLevel

This parameter contains the requested InspectionLevel as specified in Table 78. in REF 1

11.2.2.7 DataStorageConfig

This set of parameter items contains the settings of the Data Storage (DS) mechanism.

ActivationState

This parameter contains the requested state of the DS mechanism for this W-Port. The following modes are supported:

DS_Enabled

The DS mechanism is active and provides the full functionality as specified in clause 11.3.2

DS_Disabled

The DS mechanism is inactive, and the complete parameter set of this W-Port remains stored.

DS_Cleared

The DS mechanism is disabled, and the stored parameter set of this W-Port is cleared.

DownloadEnable

The DS mechanism is permitted to write data to the connected W-Device.

UploadEnable

The DS mechanism is permitted to read data from the connected W-Device.

11.2.3 State machine of the Track Configuration Manager

Figure 118 shows the state machine of the Track configuration manager.

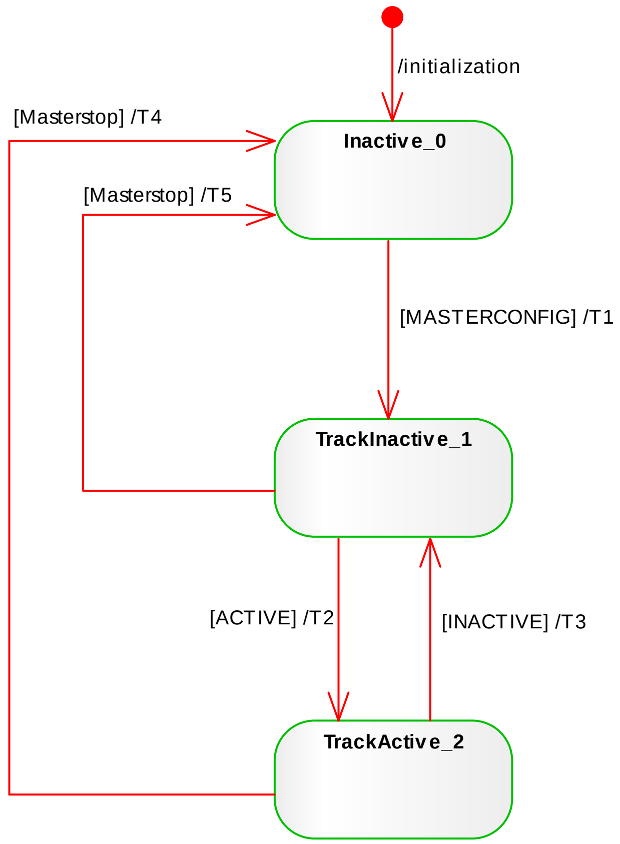


Figure 118 State machine of the Track Configuration Manager

Table 127 State transition tables of the Track Configuration Manager

STATE NAME	STATE DESCRIPTION
INACTIVE_0	Waiting for activation by W-Master application. W-Master configuration is not set.
TrackINACTIVE_1	W-Master configuration loaded. Waiting for activation of track in operating mode (CYCLIC or ROAMING).
TrackACTIVE_2	Track is active (CYCLIC, SCAN, PAIRING or ROAMING mode). Depending on the W-Port configurations the gateway application is exchanging Process Data and ready to send or receive On-request Data. For SCAN, PAIRING or ROAMING mode additionally the configuration channels are active.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke SM_SetMasterConfig to configure each track, dependent on the HW-Tracks on a W-Master (track number 0 up to 4). Each Track shall use the same MasterID and Blacklist.
T2	1	2	Invoke SM_SetTrackMode(CYCLIC or ROAMING) depending on track configuration.
T3	1	2	Invoke SM_SetTrackMode(STOP).
T4	2	0	Invoke SM_SetTrackMode(STOP) for all tracks.
T5	1	0	See T4.

11.2.4 State machine of the W-Port Configuration Manager

Figure 119 shows the state machine of the W-Port configuration manager.

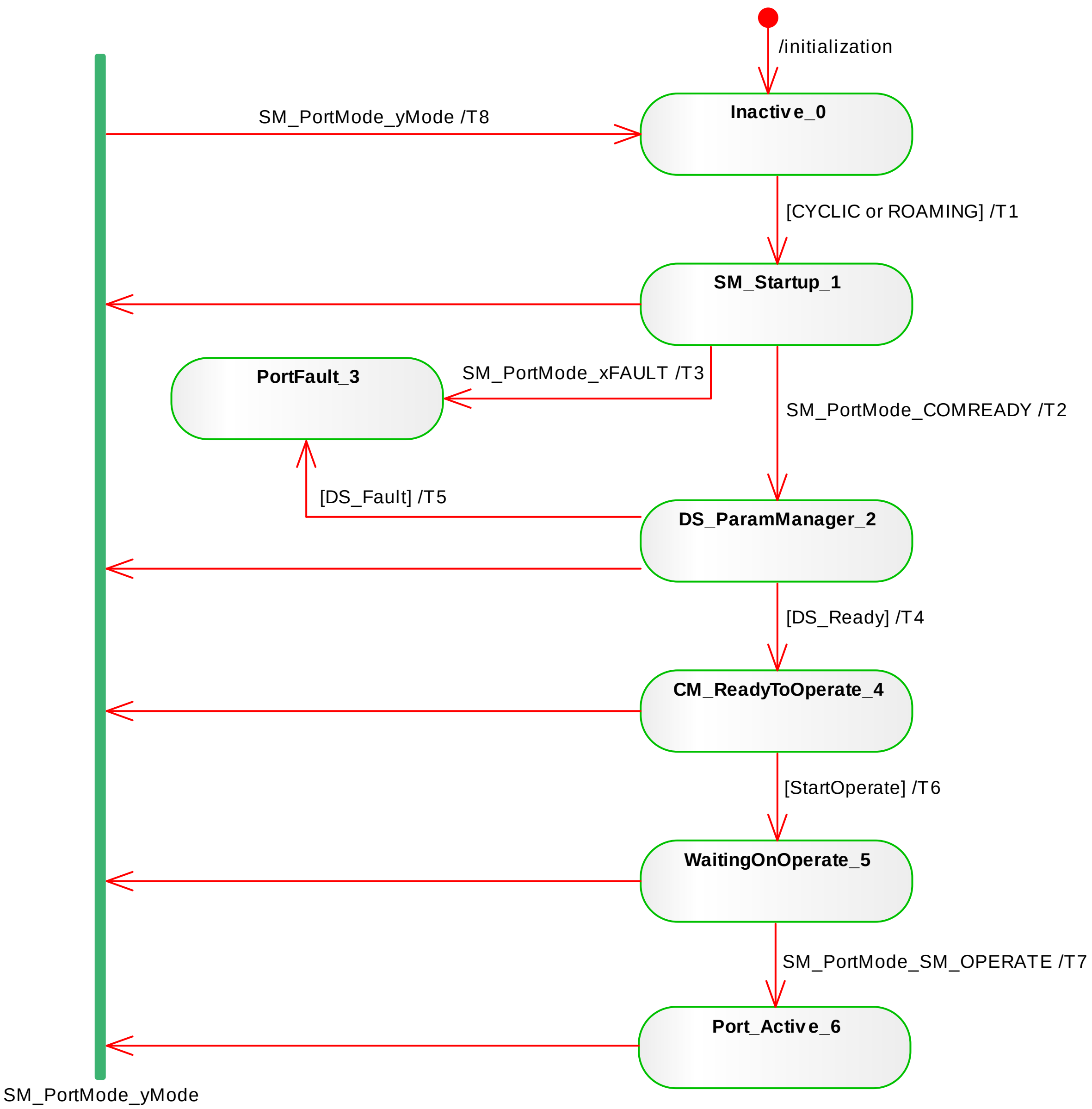


Figure 119 State machine of the W-Port Configuration Manager

Key:
xFAULT: REV_FAULT or COMP_FAULT or SERNUM_FAULT
yMODE: INACTIVE or COMLOST

Table 128 State transition tables of the W-Port Configuration Manager

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for any of the OperatingMode variables from the gateway application: CYCLIC, ROAMING.
SM_Startup_1	Waiting for an established communication or loss of communication or any of the faults REVISION_FAULT, COMP_FAULT, or SERNUM_FAULT (see Table 109).
DS_ParamManager_2	Waiting for accomplished Data Storage startup. Parameter are downloaded into the W-Device or uploaded from the W-Device.
PortFault_3	W-Device in state PREOPERATE (communicating). However, one of the three faults REVISION_FAULT, COMP_FAULT, SERNUM_FAULT, or DS_Fault occurred.
CM_ReadytoOperate_4	W-Port is waiting until the gateway application indicates "StartOperate".
WaitingOnOperate_5	Waiting for SM to switch to OPERATE.
PortActive_6	W-Port is in OPERATE mode. The gateway application is exchanging Process Data and ready to send or receive On-request Data.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke SM_SetPortConfig(CYCLIC) or SM_SetPortConfig(ROAMING)
T2	1	2	DS_Startup: The DS state machine is triggered.
T3	1	3	"Fault" indication to gateway application (REVISION_FAULT, COMP_FAULT, or SERNUM_FAULT), see Figure 116.
T4	2	4	Indication to gateway application: ReadyToOperate
T5	2	3	Data Storage failed. Rollback to previous parameter set.
T6	4	5	SM_Operate.
T7	5	6	Indication to gateway application: "Operating" (see Figure 117).
T8	1,2,3,4,5, 6	0	SM_SetPortConfig_INACTIVE. "Fault" indication to gateway application: COMLOST or INACTIVE

INTERNAL ITEMS	TYPE	DEFINITION
DS_Ready	Bool	Data Storage sequence (upload, download) accomplished. W-Port operating mode is CYCLIC or ROAMING.
DS_Fault	Bool	See Table 126
StartOperate	Bool	Gateway application causes the W-Port to switch to OPERATE.
CYCLIC	Bool	One of the Operating Modes (see 11.2.2.1)
ROAMING	Bool	One of the Operating Modes (see 11.2.2.1)

11.3 Data Storage (DS)

11.3.1 Overview

Data Storage between W-Master and W-Device is specified within this standard, whereas the adjacent upper Data Storage mechanisms depend on the individual fieldbus or application system. The W-Device holds a standardized set of objects providing parameters for Data Storage, memory size requirements, control and state information of the Data Storage mechanism. Changes of Data Storage parameter sets are detectable via the "Parameter Checksum" (see 10.4.8 in REF **1**).

11.3.2 DS data object

The structure of a Data Storage data object is specified see Table F.1 in REF **1**.

The W-Master shall always hold the header information (Parameter Checksum, VendorID, and DeviceID) for the purpose of checking and control. The object information (objects 1…*n*) will be stored within the non-volatile memory part of the W-Master (see Annex F in REF **1**.). Prior to a download of the Data Storage data object (parameter block), the W-Master will check the consistency of the header information with the particular W-Device.

The maximum permitted size of the Data Storage data object is 2 x 2^10 octets. It is mandatory for W-Masters to provide at least this memory space per W-Port if the Data Storage mechanism is implemented.

11.3.3 DS state machine

The Data Storage mechanism is called right after establishing the communication, before entering the OPERATE mode. During this time, any other communication with the W-Device shall be rejected by the gateway.
Figure 120 shows the state machine of the Data Storage mechanism.

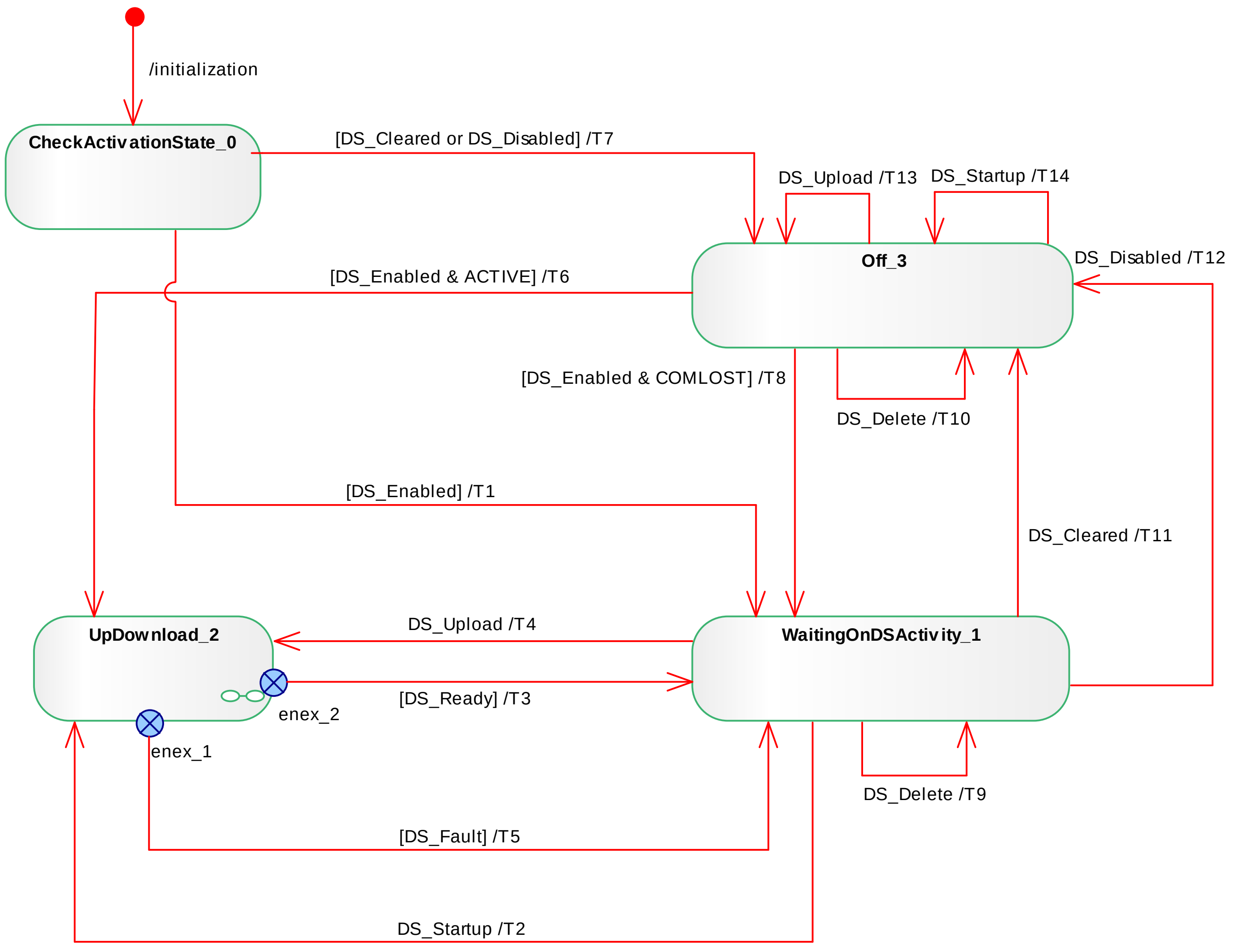


Figure 120 Main state machine of the Data Storage mechanism

Figure 121 shows the submachine of the state "UpDownload_2". This submachine can be invoked by the Data Storage mechanism or during runtime triggered by a "DS_UPLOAD_REQ" Event.

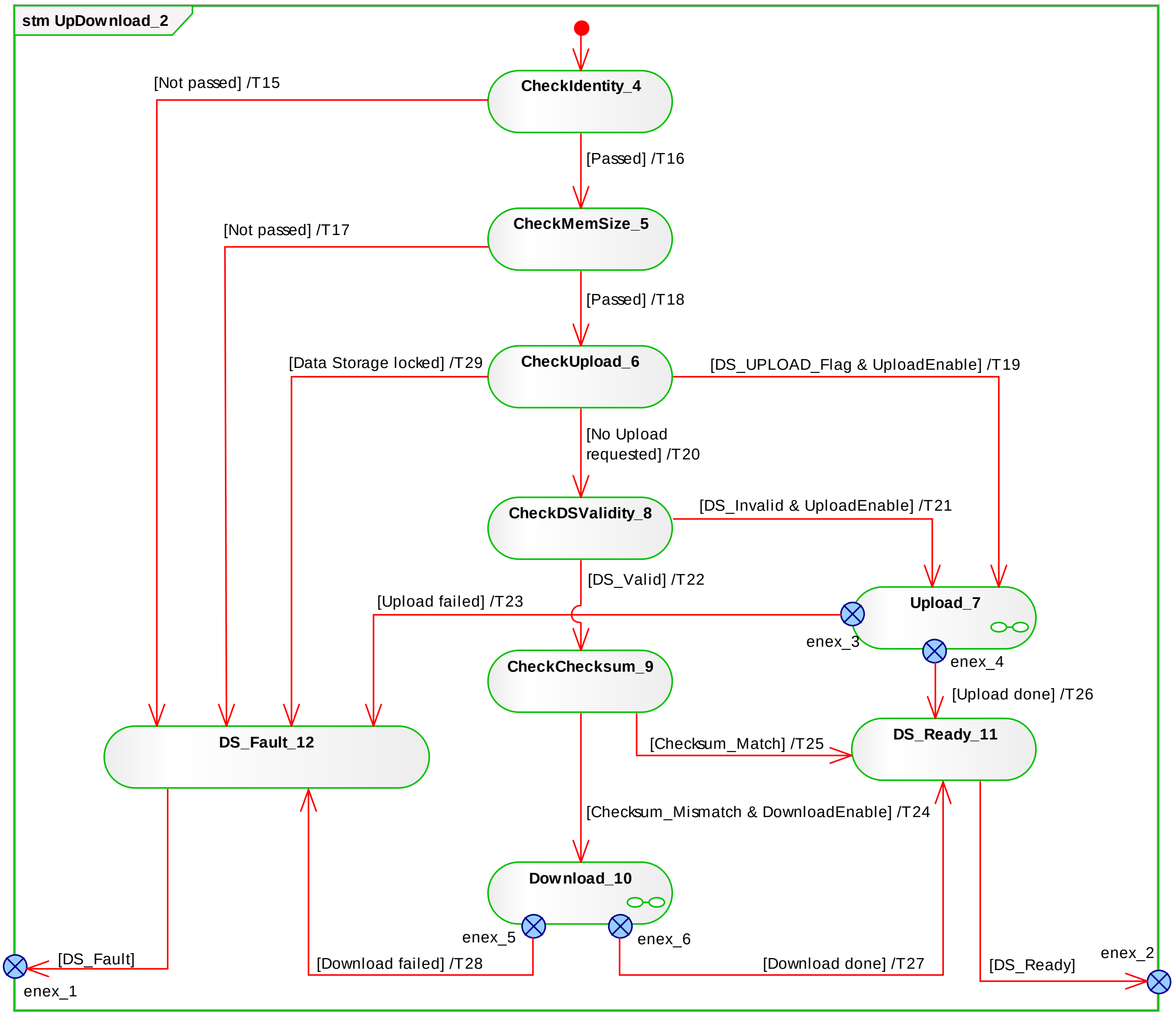


Figure 121 Submachine "UpDownload_2" of the Data Storage mechanism

Figure 122 shows the submachine of the state "Upload_7".
This state machine can be invoked by the Data Storage mechanism or during runtime triggered by a DS_UPLOAD_REQ Event.

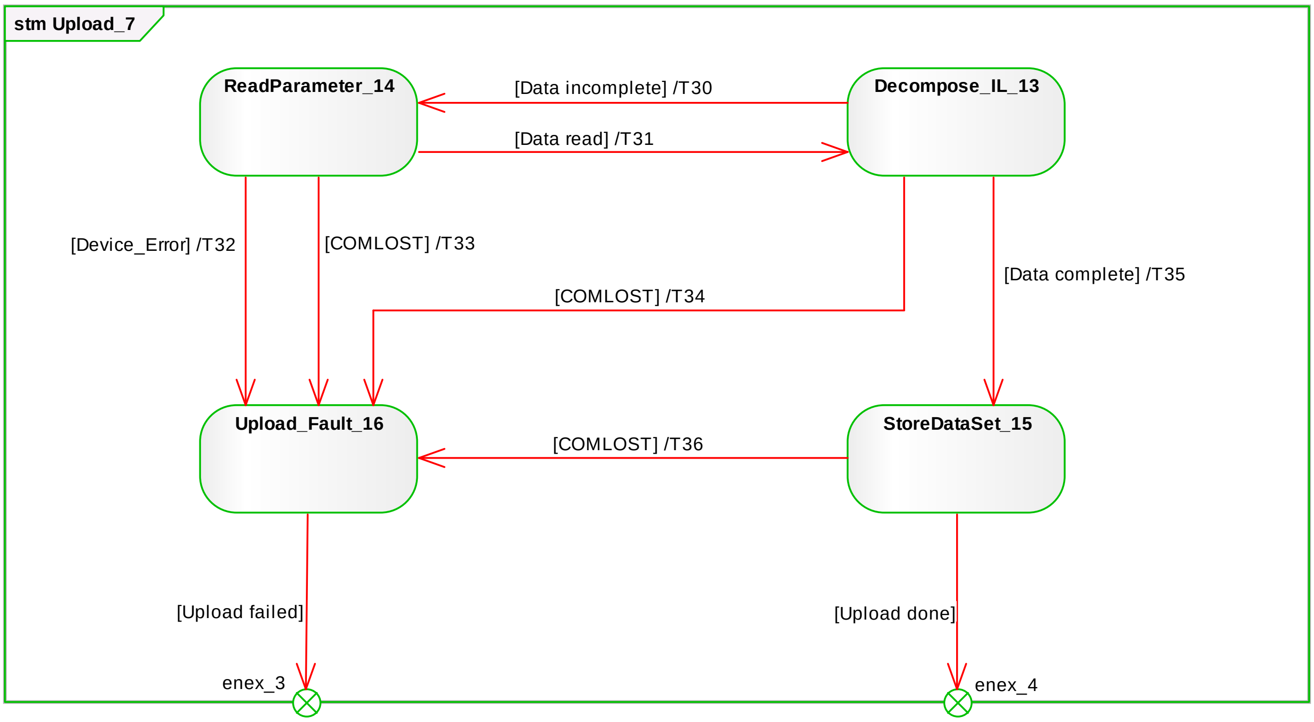


Figure 122 Data Storage submachine "Upload_7"

Figure 123 demonstrates the Data Storage upload sequence using the Data Storage Index (DSI) specified in B.2.3 and Table B.10 in REF 1. The structure of Index_List is specified in Table B.11 in REF 1. The DS_UPLOAD_FLAG shall be reset at the end of each sequence (see Table B.10 in REF 1).

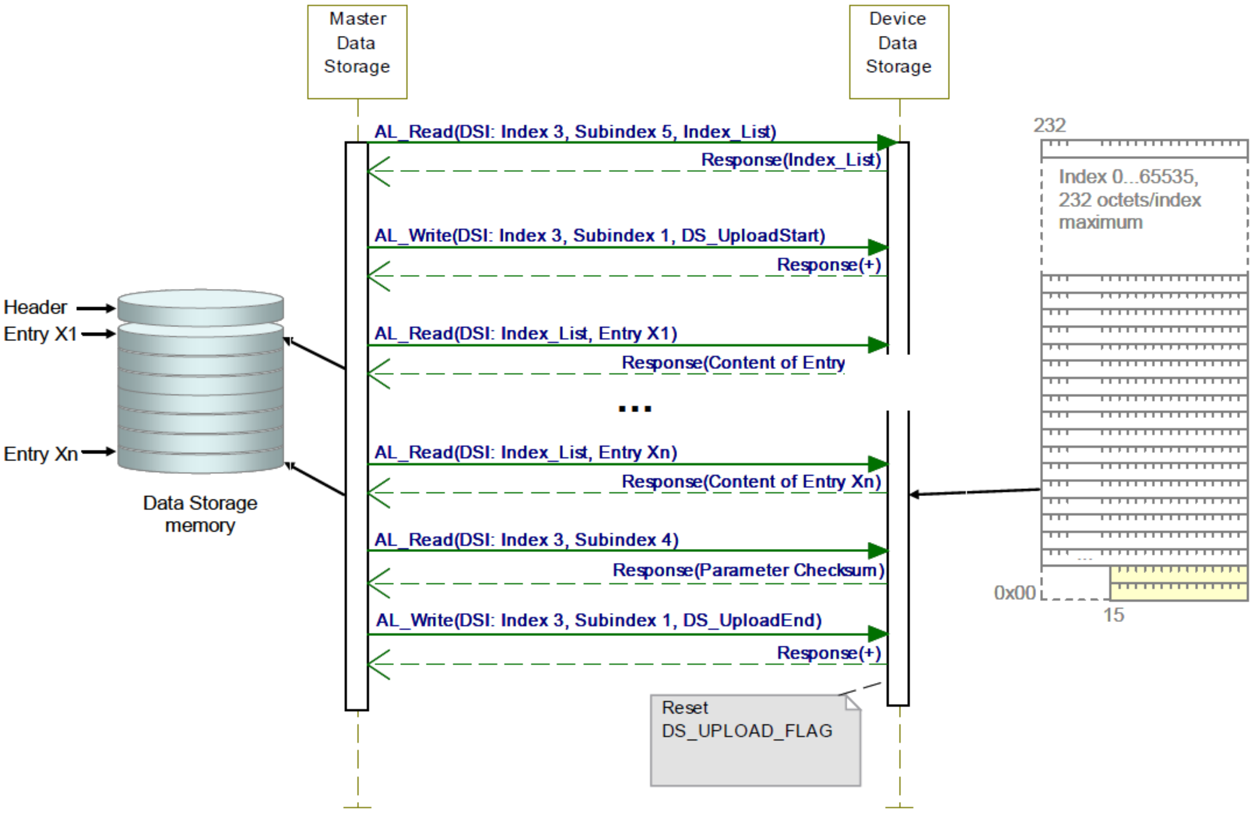


Figure 123 Data Storage upload sequence diagram

Figure 124 shows the submachine of the state "Download_10". This state machine can be invoked by the Data Storage mechanism.

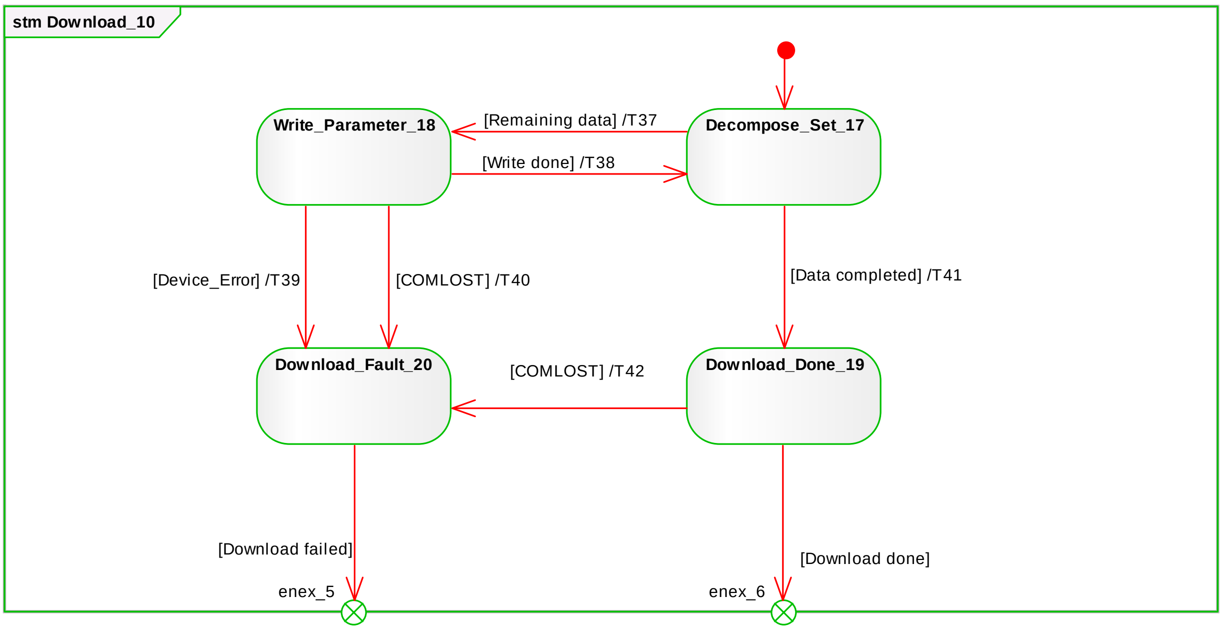


Figure 124 Data Storage submachine "Download_10"

Figure 125 demonstrates the Data Storage download sequence using the Data Storage Index (DSI) specified in B.2.3 and Table B.10 in REF 1. The structure of Index_List is specified in Table B.11 in REF 1. The DS_UPLOAD_FLAG shall be reset at the end of each sequence (see Table B.10 in REF 1).

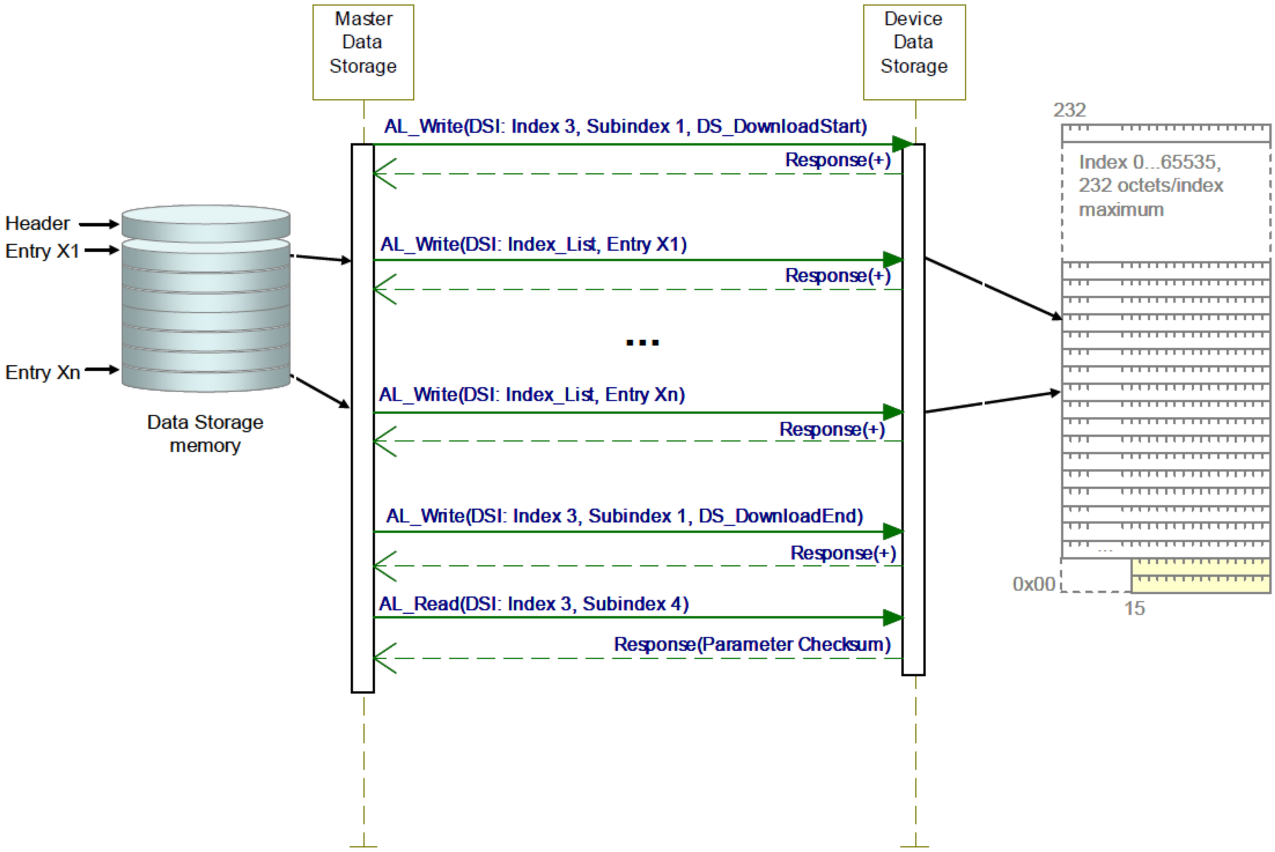


Figure 125 Data Storage download sequence diagram

Table 129 shows the states and transitions of the Data Storage state machines.

Table 129 States and transitions of the Data Storage state machines

STATE NAME	STATE DESCRIPTION
CheckActivationState_0	Check current state of the DS configuration: Independently from communication status, DS_Startup from configuration management or an Event DS_UPLOAD_REQ is expected.
WaitingOnDSActivity_1	Waiting for upload request, W-Device startup, all changes of activation state independent of the W-Device communication state.
UpDownload_2	Submachine for up/download actions and checks
Off_3	Data Storage handling switched off or deactivated
SM: CheckIdentity_4	Check W-Device identification (DeviceID, VendorID) against parameter set within the Data Storage (see Table F.2 in REF 1). Empty content does not lead to a fault.
SM: CheckMemSize_5	Check data set size (Index 3, Subindex 3) against available W-Master storage size.
SM: CheckUpload_6	Check for DS_UPLOAD_FLAG within the Data Storage Index (see Table B.10 in REF 1).
SM: Upload_7	Submachine for the upload actions
SM: CheckDSValidity_8	Check whether stored data within the W-Master is valid or invalid. A W-Master could be replaced between upload and download activities. It is the responsibility of a W-Master designer to implement a validity mechanism according to the chosen use cases.
SM: CheckChecksum_9	Check for differences between the data set content and the W-Device parameter via the "Parameter Checksum" within the Data Storage Index (see Table B.10 in REF 1).
SM: Download_10	Submachine for the download actions
SM: DS_Ready_11	Prepare DS_Ready indication to the Configuration Management (CM)
SM: DS_Fault_12	Prepare DS_Fault indication from "Identification_Fault", "SizeCheck_Fault", "Upload_Fault", and "Download_Fault" to the Configuration Management (CM)
SM: Decompose_IL_13	Read Index List within the Data Storage Index (see Table B.10 in REF 1). Read content entry by entry of the Index List from the W-Device (see Table B.11 in REF 1).
SM: ReadParameter_14	Wait until read content of one entry of the Index List from the W-Device is accomplished.
SM: StoreDataSet_15	Task of the gateway application: store entire data set according to Table F.1 and Table F. in REF 1)
SM: Upload_Fault_16	Prepare Upload_Fault indication from "W-Device_Error" and "COM_ERROR" as input for the higher level indication DS_Fault.
SM: Decompose_Set_17	Write parameter by parameter of the data set into the W-Device according to Table F.1 in REF 1
SM: Write_Parameter_18	Wait until write of one parameter of the data set into the W-Device is accomplished.
SM: Download_Done_19	Download completed. Read back "Parameter Checksum" from the Data Storage Index according to Table B.10 in REF 1. Save this value in the stored data set according to Table F.2 in REF 1.
SM: Download_Fault_20	Prepare Download_Fault indication from "W-Device_Error" and "COM_ERROR" as input for the higher level indication DS_Fault.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	-
T2	1	2	-
T3	2	1	OD_Unblock; Indicate DS_Ready to CM
T4	1	2	Confirm Event "DS_UPLOAD_REQ"
T5	2	1	DS_Break (AL_Write, Index 3, Subindex 1); clear intermediate data (garbage collection); rollback to previous parameter state; DS_Fault (see Figure 116 OD_Unblock.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T6	3	2	-
T7	0	3	-
T8	3	1	-
T9	1	1	Clear saved parameter set (see Table F.1 and Table F.2 in REF 1)
T10	3	3	Clear saved parameter set (see Table F.1 and Table F.2 in REF 1)
T11	1	3	Clear saved parameter set (see Table F.1 and Table F.2 in REF 1)
T12	1	3	-
T13	3	3	Confirm Event "DS_UPLOAD_REQ"; no further action
T14	3	3	DS_Ready to CM
T15	4	12	Indicate DS_Fault(Identification_Fault) to the gateway application
T16	4	5	Read "Data Storage Size" according to Table B.10 in REF 1, OD_Block
T17	5	12	Indicate DS_Fault(SizeCheck_Fault) to the gateway application
T18	5	6	Read "DS_UPLOAD_FLAG" according to Table B.10 in REF 1.
T19	6	7	Data Storage Index 3, Subindex 1: "DS_UploadStart" (see Table B.10 in REF 1)
T20	6	8	-
T21	8	7	Data Storage Index 3, Subindex 1: "DS_UploadStart" (see Table B.10 in REF 1)
T22	8	9	-
T23	7	12	Data Storage Index 3, Subindex 1: "DS_Break" (see Table B.10 in REF 1). Indicate "DS_Fault(Upload)" to the gateway application
T24	9	10	Data Storage Index 3, Subindex 1: "DS_DownloadStart" (see Table B.10 in REF 1)
T25	9	11	-
T26	7	11	Data Storage Index 3, Subindex 1: "DS_UploadEnd"; read Parameter Checksum (see Table B.10 in REF 1)
T27	10	11	-
T28	10	12	Data Storage Index 3, Subindex 1: "DS_Break" (see Table B.10 in REF 1) Indicate "DS_Fault(Download)" to the gateway application.
T29	6	12	Indicate DS_Fault(Data Storage locked) to the gateway application
T30	13	14	AL_Read (Index List)
T31	14	13	-
T32	14	16	-
T33	14	16	-
T34	13	16	-
T35	13	15	Read "Parameter Checksum" (see Table B.10 in REF 1)
T36	15	16	-
T37	17	18	Write parameter via AL_Write
T38	18	17	-
T39	18	20	-
T40	18	20	-
T41	17	19	Data Storage Index 3, Subindex 1: "DS_DownloadEnd" (see Table B.10 in REF 1) Read "Parameter Checksum" (see Table B.10 in REF 1).
T42	19	20	-

INTERNAL ITEMS	TYPE	DEFINITION
DS_Cleared	Bool	Data Storage handling switched off, see 11.2.2.6 in REF 1.
DS_Disabled	Bool	Data Storage handling deactivated, see 11.2.2.6 in REF 1.
DS_Enabled	Bool	Data Storage handling activated, see 11.2.2.6 in REF 1.
COMLOST	Bool	Error in communication detected
W-Device_Error	Bool	Access to Index denied, AL_Read or AL_Write.cnf(-) with ErrorCode 0x80
DS_Startup	Variable	Trigger from CM state machine, see Figure 116
COMLOST	Bool	No communication
ACTIVE	Bool	communication working properly
DS_UPLOAD_REQ	Event	See Table D.2
UploadEnable	Bool	Data Storage handling configuration, see 11.2.2.6 in REF 1.
DownloadEnable	Bool	Data Storage handling configuration, see 11.2.2.6 in REF 1.
DS_Valid	Bool	Valid parameter set available within the W-Master. See state description "SM: CheckDSValidity_8"
DS_Invalid	Bool	No valid parameter set available within the W-Master. See state description "SM: CheckDSValidity_8"
Checksum_Mismatch	Bool	Acquired "Parameter Checksum" from W-Device does not match the checksum within Data Storage (binary comparison)
Checksum_Match	Bool	Acquired "Parameter Checksum" from Device matches the checksum within Data Storage (binary comparison)

11.3.4 Parameter selection for Data Storage

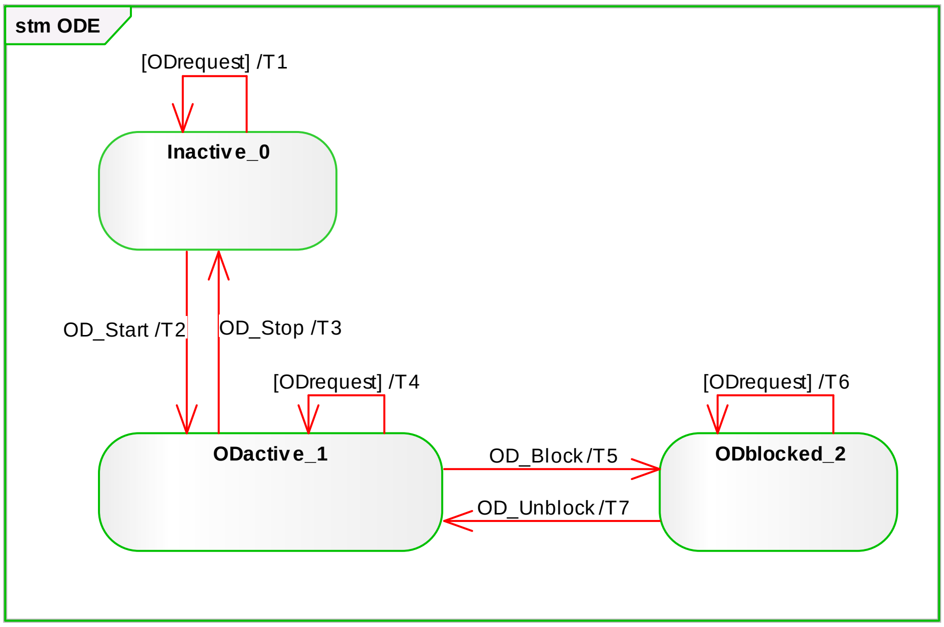
The W-Device designer defines the parameters that are part of the Data Storage mechanism.

The IODD marks all parameters not included in Data Storage with the attribute "excludedFromDataStorage". However, the Data Storage mechanism shall not consider the information from the IODD but rather the Parameter List read out from the W-Device.

11.4 On-request Data Exchange (ODE)

Fehler! Verweisquelle konnte nicht gefunden werden. shows the state machine of the W-Master's On-request Data Exchange. This behavior is mandatory for a W-Master.

During an active data transmission of the Data Storage mechanism, all On-request Data requests are blocked.



On-request Data Exchange

Table 130 shows the state transition table of the On-request Data Exchange state machine.

Table 130 State transition table of the ODE state machine

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
ODactive_1	On-request Data communication active using AL_Read or AL_Write
ODblocked_2	On-request Data communication blocked

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	Access blocked (inactive): indicates "Service not available" to the gateway application
T2	0	1	-
T3	1	0	-
T4	1	1	AL_Read or AL_Write
T5	1	2	-
T6	2	2	Access blocked temporarily: indicates "Service not available" to the gateway application
T7	2	1	-

INTERNAL ITEMS	TYPE	DEFINITION
ODrequest	Variable	On-request Data read or write requested via AL_Read or AL_Write

11.5 Diagnosis Unit (DU)

The Diagnosis Unit (DU) routes Events from the AL to the Data Storage unit or the gateway application. These Events primarily contain diagnosis information.

Main goal for diagnosis information is to alert an operator in an efficient manner. That means:

- no diagnosis information flooding
- report of the root cause of an incident within a W-Device or within the W-Master and no subsequent correlated faults
- diagnosis information shall provide information on how to maintain or repair the affected component for fast recovery of the automation system.

Within IO-Link Wireless, diagnosis information of Devices is conveyed to the W-Master via Events consisting of EventQualifiers and EventCodes (see clause 15. The associated human readable text is available for standardized EventCodes within this standard (see Table 180) and for vendor specific EventCodes within the associated IODD file of a W-Device. The standardized EventCodes can be mapped to semantically identical or closest fieldbus channel diagnosis definitions within the gateway application. Vendor specific IODD coding can be mapped to specific channel diagnosis definitions (individual code and associated human readable information) within the fieldbus device description file.

Fieldbus engineering tools and process monitoring systems (human machine interfaces) can use the fieldbus device description to decode the received fieldbus diagnosis code into human readable diagnosis text.

Diagnosis information flooding is avoided by flow control, which allows for only one Event per W-Device to be propagated to the W-Master/gateway application at a time.

The gateway application is able to start or stop the Diagnosis Unit (see Figure 116). When stopped, the DU is deferring any received AL_Event.ind call until the DU is started again.

The special DS_UPLOAD_REQ Event (see clause 15. and Table 181) of a W-Device shall be redirected to the common W-Master application Data Storage. Those Events are acknowledged by the DU itself and not propagated to the gateway.

Figure 126 shows an example of the diagnosis information flow through a complete SDCI/fieldbus system.

NOTE The flow can end at the W-Master/PDCT or be more integrated depending on the fieldbus capabilities.

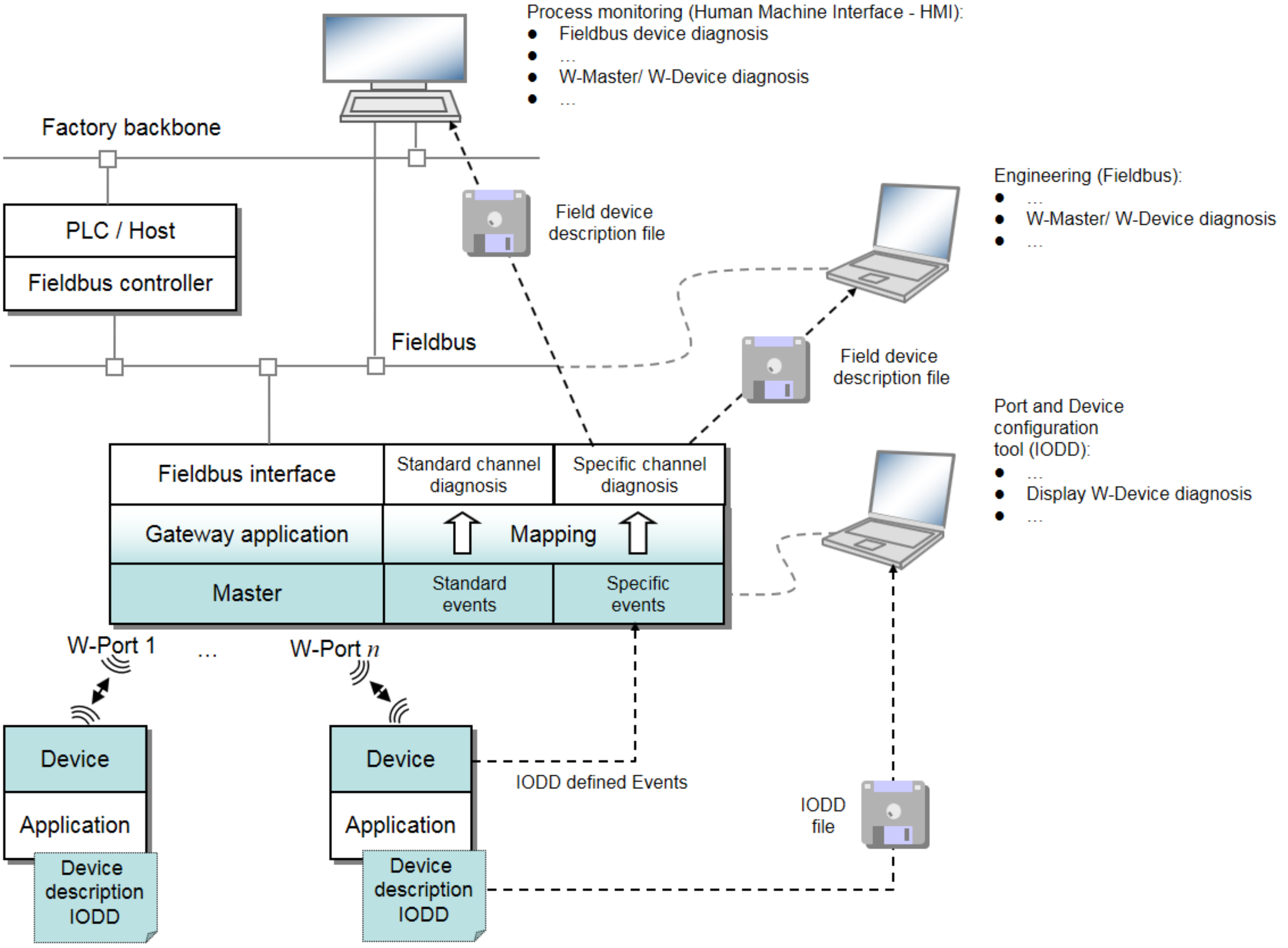


Figure 126 System overview of IO-Link diagnosis information propagation via Events

11.6 Process Data Exchange (PDE)

11.6.1 General

The Process Data Exchange provides the transmission of Process Data between the gateway application and the connected W-Device.

After an established communication and Data Storage, the W-Port is ready for any On-request Data (OD) transfers. The Process Data communication is enabled whenever the specific W-Port or all ports are switched to the OPERATE mode.

11.6.2 Process Data mapping

According to 11.2.2.5 the input and output Process Data are mapped to a specific part of the gateway Process Data stream.

Figure 127 shows a sample mapping of the Process Data from 3 W-Master ports to the Gateway Process Data stream.

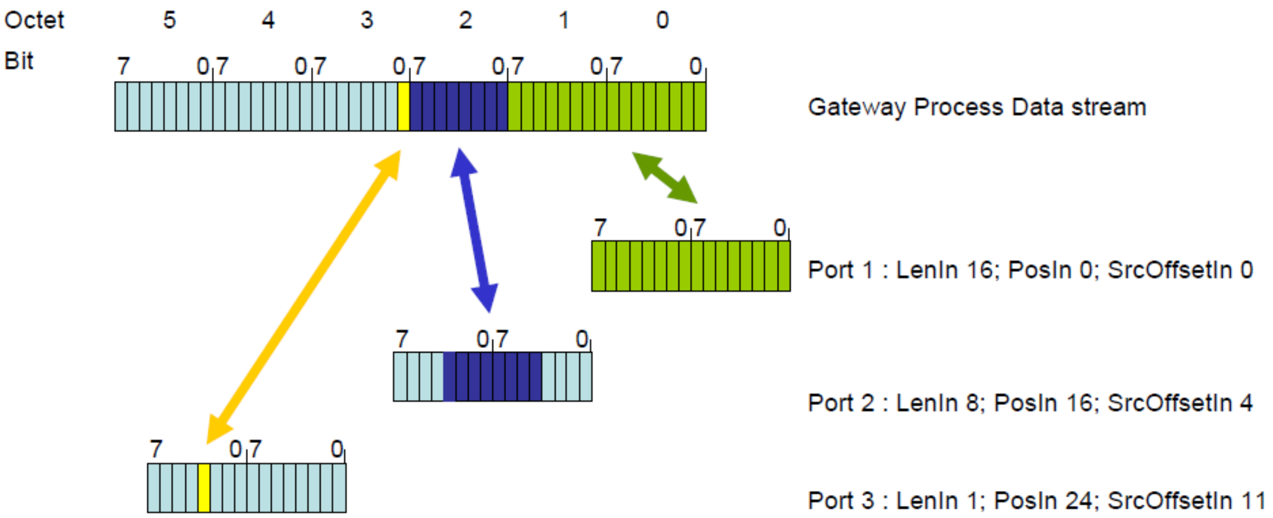


Figure 127 Process Data mapping from ports to the gateway data stream

11.6.3 Process Data invalid/valid qualifier status

A sample transmission of an output PD qualifier status "invalid" from W-Master AL to W-Device AL is shown in the upper section of Figure 128

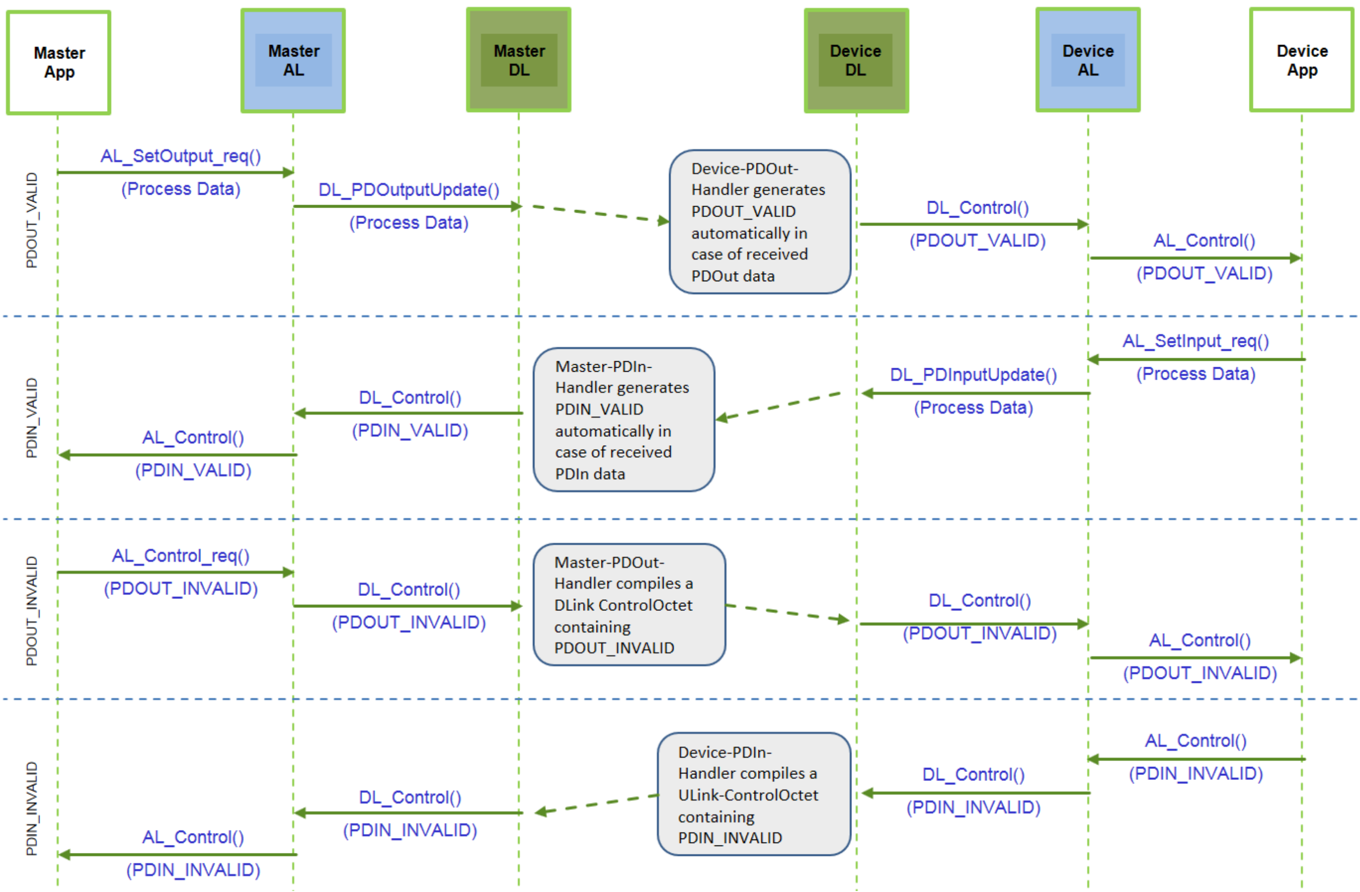


Figure 128 Propagation of PD qualifier status between W-Master and W-Device

The Master informs the Device about the output Process Data qualifier status dependent on the PDOut state.

PDOUT_VALID:
The Device PDOut-handler generates the PDOUT_VALID automatically by receiving PDOut Process data.

PDOUT_INVALID:
The Master PDOut-handler sends the PDOUT_INVALID via the DLink Control Octet.

For input Process Data, the W-Device sends its Process Data qualifier status for PDIN_VALID / PDIN_INVALID in the same manner as the PDOut state from W-Master.

For detailed information see 12.9 and sequence chart Figure 128.

11.7 Port and Device configuration tool (PDCT)

11.7.1 General

Figure 93 in REF 1 and Figure 106 in REF 1 demonstrate the necessity of a tool to configure ports, parameterize the W-Device, display diagnosis information, and provide identification and maintenance information. Depending on the degree of integration into a fieldbus system, the PDCT functions can be reduced, for example if the W-Port configuration can be achieved via the field device description file of the particular fieldbus.

The PDCT functionality can be integrated partially (navigation, parameter transfer, etc.) or completely into the engineering tool of the particular fieldbus.

11.7.2 Basic layout examples

Figure 129 shows one example of a PDCT display layout.

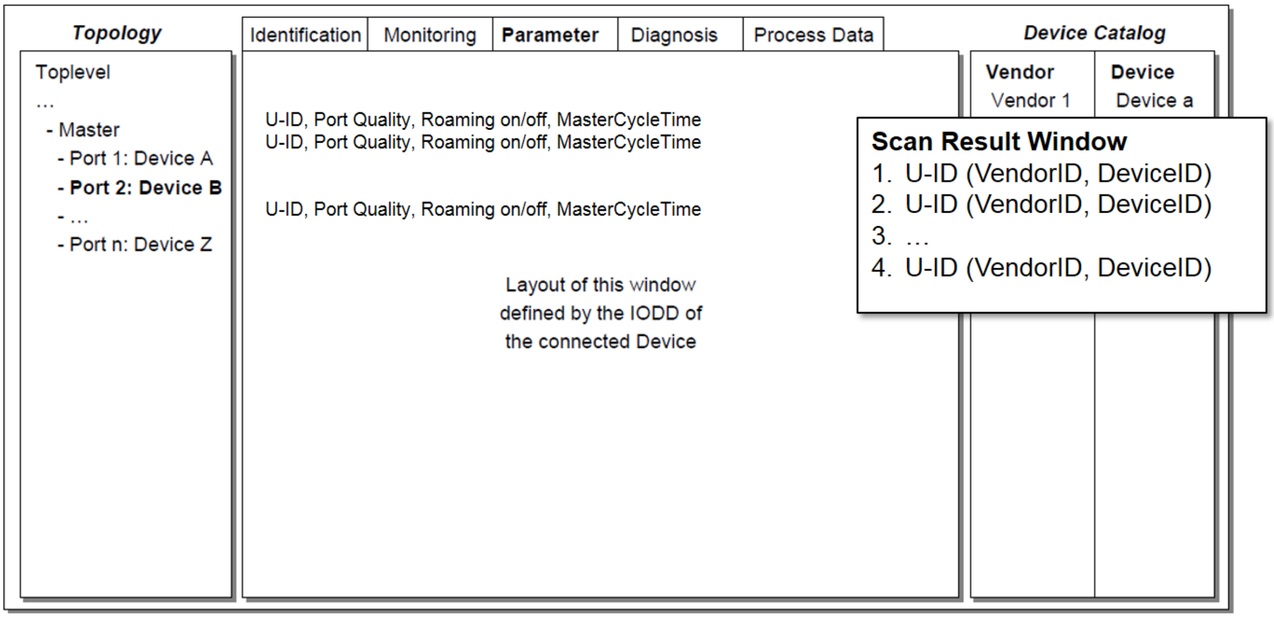


Figure 129 Example 1 of a PDCT display layout

The PDCT display should always provide a navigation window for a project or a network topology, a window for the particular view on a chosen W-Device that is defined by its IODD, and a window for the available Devices based on the installed IODD files.

Figure 130 shows another example of a PDCT display layout.

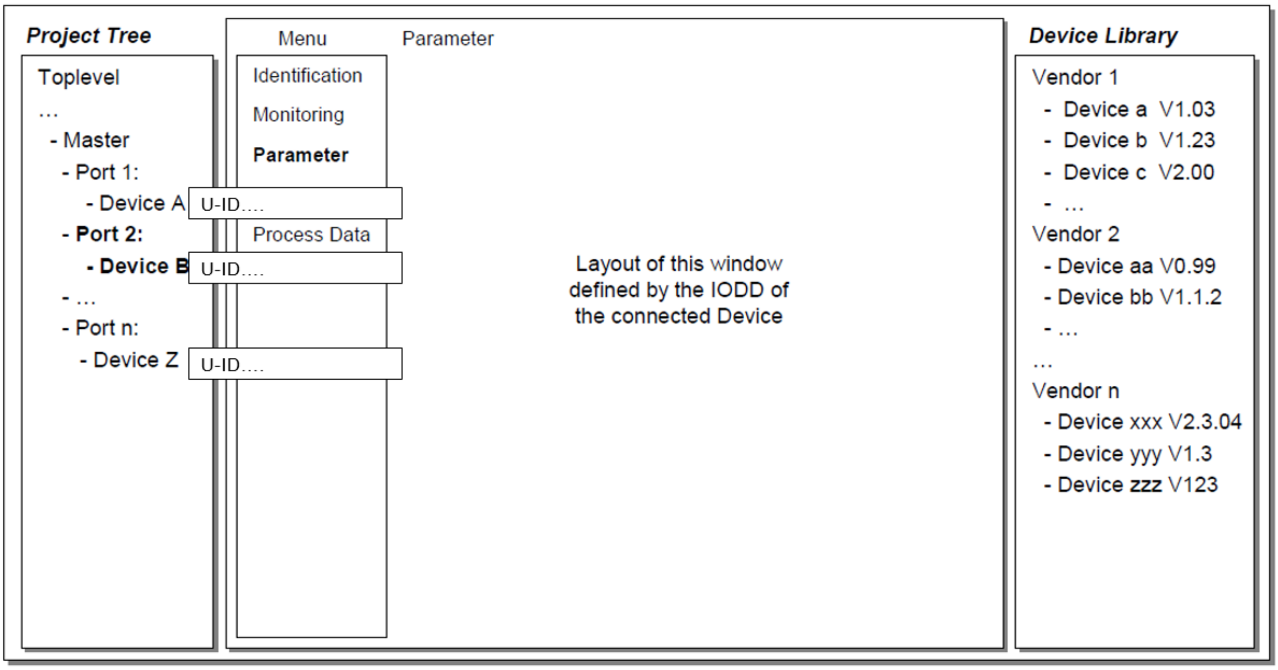


Figure 130 Example 2 of a PDCT display layout

NOTE Further information can be retrieved from IEC/TR 62453-61.

11.8 Gateway application

11.8.1 General

The Gateway application depends on the individual host system (fieldbus, PLC, etc.) the W-Master applications are embedded in. It is the responsibility of the individual system to specify the mapping of the W-Master services and variables.

However, the designers of IO-Link didn't have a chance to specify a detailed Master interface into existing fieldbuses. Therefore, the IO-Link community decided to define a Standardized Master Interface (SMI) which should be followed within the IO-Link wireless - System Extension. For detailed definition of the Standardized Master Interface see REF 12 IO-Link Community, Addendum 2017, V2.0, December 2017, Order No. 10.152

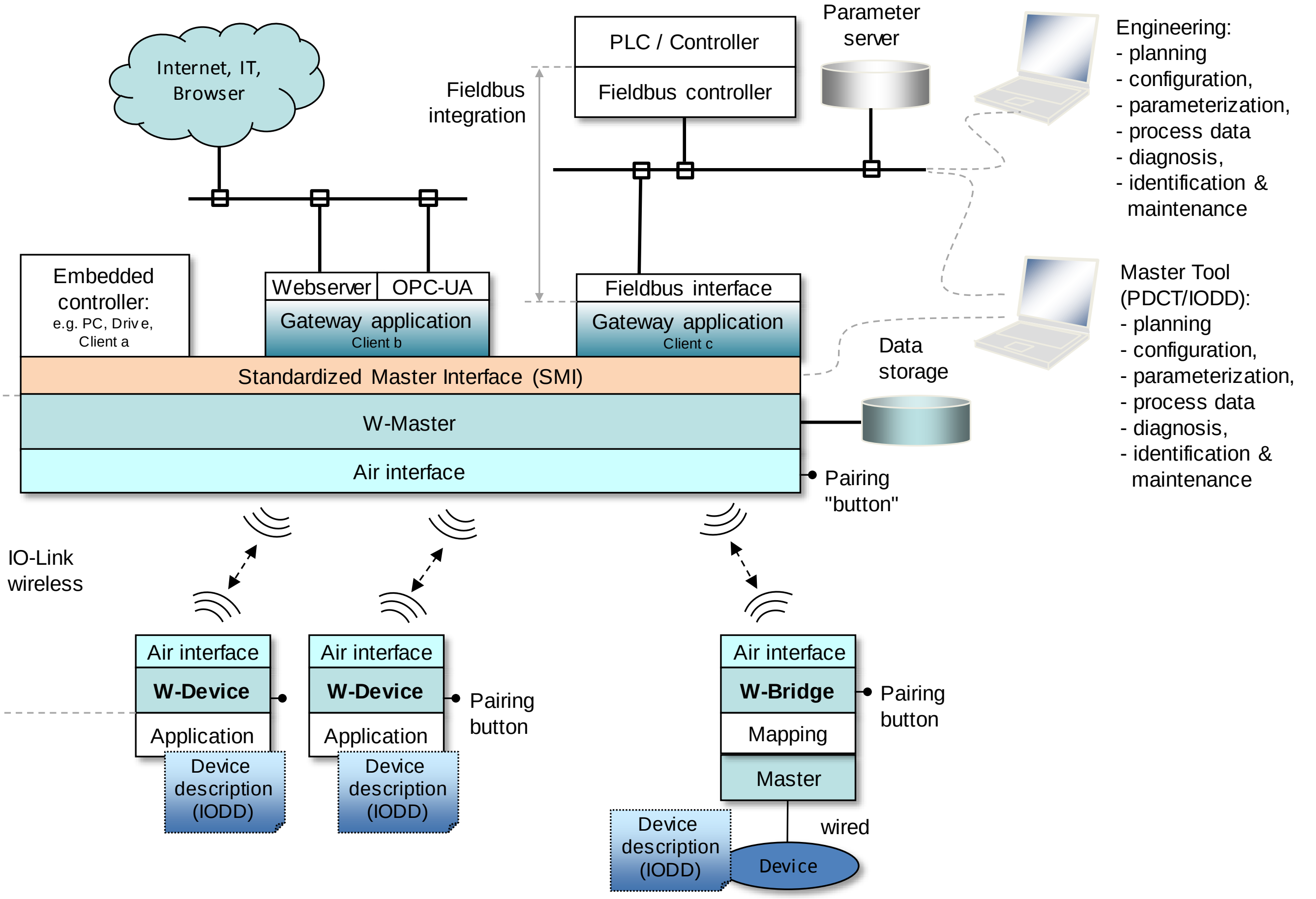


Figure 131 Generic relationship of W-Master and automation technology

11.8.2 Changing W-Device configuration including Data Storage

After each change of W-Device configuration/parameterization (CVID and/or CDID, see 9.2.2.2 REF **1**), the associated previously stored data set within the W-Master shall be cleared or marked invalid via the variable DS_Delete.

11.8.3 Parameter server and recipe control

The W-Master may combine the entire parameter sets of the connected Devices together with all other relevant data for its own operation, and make this data available for higher level applications. For example, this data may be saved within a parameter server which may be accessed by a PLC program to change recipe parameters, thus supporting flexible manufacturing.

NOTE The structure of the data exchanged between the W-Master and the parameter server is outside the scope of this standard.

11.8.4 Interoperability to 5G Systems

In the requirements for 5G systems, the EAP framework for subscriber network access authentication according to RFC 5247 is proposed as primary authentication mechanism. When a vendor intends to implement a W-Master with interoperability to 5G management frameworks, it is therefore recommended by this standard to implement EAP in the IOLW-Master gateway application layer.

11.9 Human machine Interface (HMI)

11.9.1 Faulty device replacement

It is possible to replace a faulty W-Device without using a configuration tool (PDCT). Therefore, a minimum HMI functionality is mandatory.
The W-Master displays the W-Port of a faulty W-Device. By pressing a button or a similar interface of the W-Master the pairing by button process will be started and the W-Master is waiting for a W-Device, which activates the pairing by button mode (see 4.4.2.2). Depending on the inspection level check the W-Device will be paired.
After a successful pairing, the W-Master will change back in cyclic or roaming mode.
In case of multiple faulty W-Devices, the replacement will be done one by one or by using an optional extended menu. The pairing button process has to be locked in case of non-device fault.

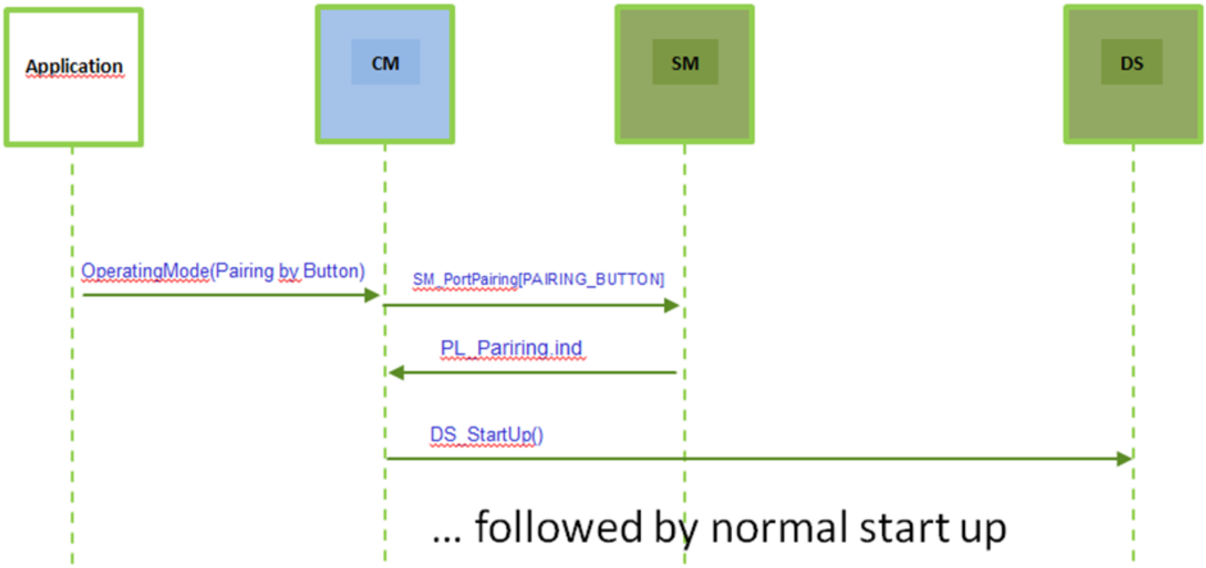


Figure 132 Faulty device replacement

Annex A

(normative)

12 W-Messages Codings

12.1 Overview

The Master indicates the manner the user data (see 12.8) shall be transmitted within a W-Frame.

12.2 Definition of a W-Message

Within the payload of a W-Frame, W-Messages are transmitted in DLink and ULinks (see Figure 133). W-Messages are used to serve the IO-Link Wireless mechanisms such as Process data, MasterCommand and EVENT- or ISDU-data.

A W-Message in a DLink or an ULink consists of Control Octets (CO), followed by data, or without data (e.g. MasterCommand).
See Figure 134 for definition of DLink Control Octet and Figure 136. for definition of ULink Control Octet.

For the generation of the Control Octets, see 6.5.3 Compilation of DLink W-Message and 6.5.5 Compilation of ULink W-Message.

For examples of the transmission of W-Messages see 12.6 Example for downlink data transmission and 12.7 Examples for uplink data transmissions.

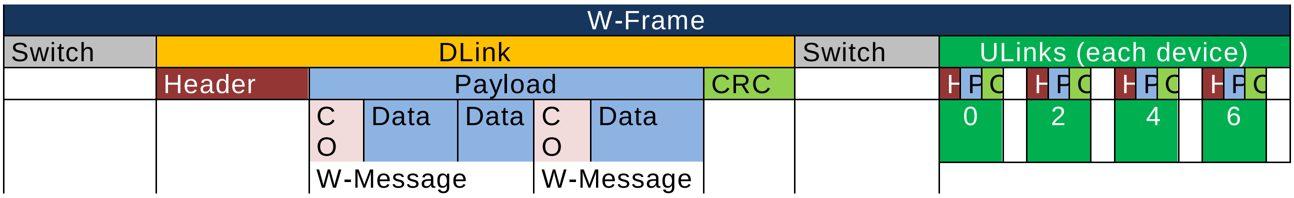


Figure 133 W-Message and Control Octets

12.3 Downlink W-Messages: Control Octets

12.3.1 DLink Control Octet

The DLink Control Octet is used to send a W-Message to a dedicated W-Device within a DLink.
Figure 134 shows the definition of the Control Octet (2 octet) for a DLink-W-Message

Octet 0								Octet 1							
Slotnumber (SN) ("DeviceAddress")			ChannelCode (ChC)			FlowControl (FC)					DataLength (DLen)				
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 134 Definition of DLink Control Octet

12.3.1.1 Bit 0 to 4: DataLength (DLen)

These bits contain a 5 bit value from 0 to 31 to transmit the data length of the data which are following after the W-Message. If the W-Message contains no data (see Table 42), the DataLength shall be ignored.
DLen is coded in the following way:

Table 131 Definition of DataLength (DLen)

DataLength (DLen)	
DLen	Data length in octet followed by the Control Octet
0	1
1	2
...	...
31	32

12.3.1.2 Bit 5 to 9: FlowControl (FC)

The FlowControl is controlling the segmented data flow for Process data, Event- or ISDU data. The defined values for the FlowControl are listed in Table 75 Flow Control for segmented data. Examples for the usage of FlowControl see 12.6 Example for DLink data transmission and 12.7. Examples for ULink data transmission.

12.3.1.3 ChannelCode (ChC)

These bits indicate the communication channel code for the access to the user data. The defined values for the communication channel parameter are listed in Table 132.

Table 132 Definition of ChannelCode (ChC) for DLink

ChannelCode (ChC)		
Value	Definition	Remarks
0	INVALID	W-Message is invalid and shall be ignored by W-Device
1	Process data	W-Master sends Process data out to W-Device
2	Process data INVALID	W-Master sends PDOUT_INVALID to W-Device
3	ISDU	W-Master sends ISDU data
4	EVENT	W-Master sends event acknowledge to W-Device
5	MasterCommand	W-Master sends a MasterCommand to W-Device, see Table 165**.**
6	Reserved	Reserved for future
7	Reserved	Reserved for future

12.3.1.4 Bit 13 to 15: Slotnumber (SN)

These bits contain the "address" (Slotnumber 0 to 7) to which W-Device the W-Message shall be sent.

12.3.2 DLink Control Octets contains MasterCommand

Figure 135 shows the DLink-W-Message to transmit a MasterCommand to a W-Device:

Octet 0								Octet 1							
Slotnumber (SN) ("DeviceAddress")			ChannelCode (ChC) = 5			Reserved	BC	MasterCommand							
Bit 15	Bit 14	Bit 13	1	0	1	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 135 DLink Control Octets contains MasterCommand

If the ChC = 5, the 2nd octet (octet 1) shall be used as MasterCommand. For definition of the MasterCommand see Table 165

12.3.2.1 Bit 8: Broadcast (BC)

Bit 8 marks a Broadcast message. If this bit is equal to 1 then MCmd is for all W-Devices on Track. Broadcast supports only limited MasterCommands, see Table 165.

12.3.2.2 Bit 9: Reserved

Reserved for future use.

12.4 Uplink W-Messages

12.4.1 ULink Control Octet

The ULink Control Octet is used to send a W-Message from the W-Device to the W-Master within an ULink. Figure 136 shows the definition of the ULink Control Octet (1 octet):

<table>
<tr><th colspan="8">Octet 0</th></tr>
<tr><td colspan="3">ChannelCode (ChC)</td><td colspan="5">FlowControl (FC)</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
</table>

Figure 136 Definition of ULink Control Octet

12.4.1.1 Bit 0 to 4: FlowControl (FC)

The FlowControl is controlling the segmented data flow for Process data, Event- or ISDU data. The defined values for the FlowControl are listed in Table 75. Flow Control definition for segmented data. Examples for the usage of FlowControl see 12.6 Example for DLink data transmission and 12.7 Examples for ULink data transmission.

12.4.1.2 ChannelCode (ChC)

These bits indicate the communication channel code for the access to the user data. The defined values for the communication channel parameter are listed in Table 133.

Table 133 Definition of ChannelCode (ChC) for ULink

ChannelCode (ChC)		
Value	Definition	Remarks
0	INVALID	W-Message is invalid and shall be ignored by W-Master
1	Process data	W-Device sends Process data in to W-Master
2	Process data INVALID	W-Device sends PDIN_INVALID to W-Master
3	ISDU	W-Device sends ISDU data to W-Master
4	EVENT	W-Device sends EVENT data to W-Master
5	Reserved	Reserved for future
6	Reserved	Reserved for future
7	Reserved	Reserved for future

12.5 Example for combination of several W-Messages within a DLink / PreDLink

The W-Master Message handler collects all data delivered via all DL-B handler for each W-Device and compiles the Control Octet for all W-Messages subsequently. Further in the Message handler place the compiled Control Octet with the delivered handler-data to the payload of a downlink in a predefined order, see Figure 60. For definition of the Control Octet see Figure 134.

The following example shows the placement of different W-Messages in a downlink:

Slotnumber 3 (Device Address = 3):
8 Octet process data Out
10 Octet acyclic ISDU-write

Slotnumber 5 (Device Address = 5):
1 Octet MasterCommand in PreDownLink

Slotnumber 1 (Device Address = 1):
6 Octet process data Out
Event acknowledge

The W-Master Message handler places the W-Messages in the following way into DLink payload, see Figure 137.

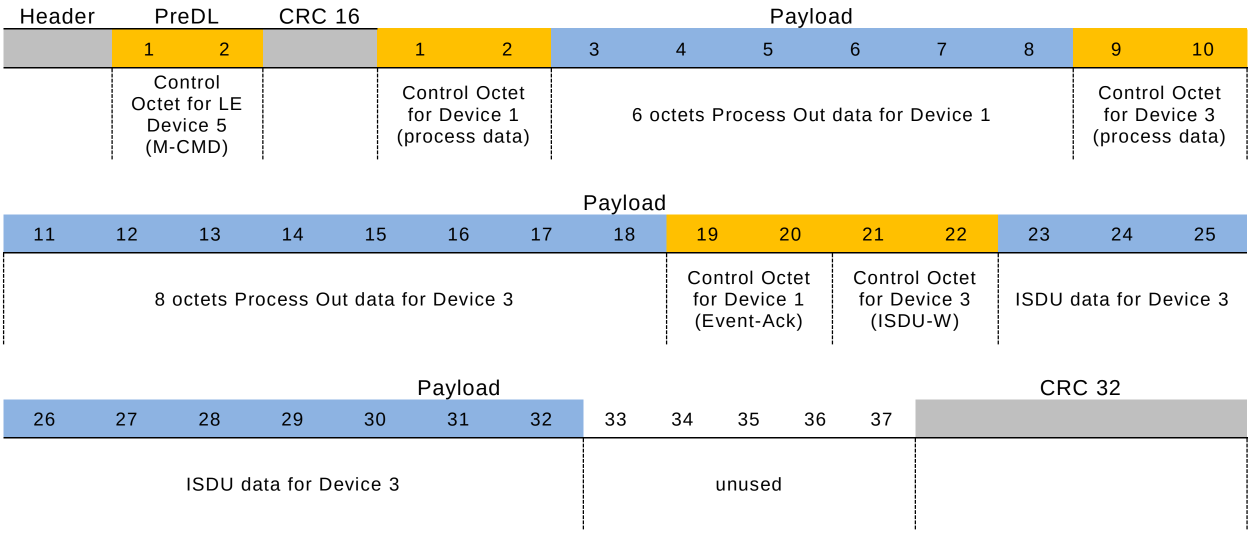


Figure 137 Placement of different W-Messages in a downlink

12.6 Example for DLink data transmission within cyclic process data and segmentation

Note:
Maximum downlink payload (37 octet) see Figure 35
For the definition of DLink Control Octet (2 octet) see Figure 134

This example demonstrates how the W-Master sends:

- 16 octets Process Data Out to the W-Device at Slot 2 (W-Cycle = 5 ms)
- 8 octets Process Data Out to the W-Device at Slot 3 (W-Cycle = 10 ms)
- 50 octets acyclic ISDU Data to the W-Device at Slot 5 (acyclic)
- Acyclic Event acknowledge to the W-Device at Slot 3 (acyclic, see W-Sub-cycle x+8)
- PDOUT_INVALID to the W-Device at Slot 2 (acyclic, see W-Cycle x+12)

DLink for W-Cycle x:
2 Octet Control Octet: SN = 2; ChC = 1; FC = EOS; DLen = 15
16 Octet Data: 16 octets Process Out data following the Control Octet
2 Octet Control Octet: SN = 3; ChC = 1; FC = EOS; DLen = 7
8 Octet Data: 8 octets Process Out data following the Control Octet
2 Octet Control Octet: SN = 5; ChC = 3; FC = START; DLen = 6
7 Octet Data: 7 octet ISDU data following the Control Octet

DLink for W-Sub-cycle x+1:
2 Octet Control Octet: SN = 5; ChC = 3; FC = 1; DLen = 31
32 Octet Data: 32 octet ISDU data following the Control Octet

DLink for W-Sub-cycle x+2:
2 Octet Control Octet: SN = 5; ChC = 3; FC = 2; DLen = 10
11 Octet Data: 11 octet ISDU data following the Control Octet

DLink for W-Cycle x+3:
- 2 Octet Control Octet: SN = 2; ChC = 1; FC = EOS; DLen = 15
- 16 Octet Data: 16 octets Process Out data following the Control Octet
- Control Octet: SN = 5; ChC = 3; FC = EOS; DLen = x
- Data: No data to transmit. Only Control Octet is transmitted to send EOS.

DLink for W-Sub-cycle x+4: nothing to transmit

DLink for W-Sub-cycle x+5: nothing to transmit

DLink for W-Cycle x+6:
- 2 Octet Control Octet: SN = 2; ChC = 1; FC = EOS; DLen = 15
- 16 Octet Data: 16 octets Process Out data following the Control Octet
- 2 Octet Control Octet: SN = 3; ChC = 1; FC = EOS; DLen = 7
- 8 Octet Data: 8 octets Process Out data following the Control Octet

DLink for W-Sub-cycle x+7: nothing to transmit

DLink for W-Sub-cycle x+8:
- Control Octet: SN = 3; ChC = 4; FC = x; DLen = x
- Data: No data to transmit. Only Control Octet is transmitted to Event-Ack.

DLink for W-Cycle x+9:
- 2 Octet Control Octet: SN = 2; ChC = 1; FC = EOS; DLen = 15
- 16 Octet Data: 16 octets Process Out data following the Control Octet

DLink for W-Sub-cycle x+10: nothing to transmit

DLink for W-Sub-cycle x+11: nothing to transmit

DLink for W-Cycle x+12:
- 2 Octet Control Octet: SN = 2; ChC = 2; FC = x; DLen = x
- Data: No data to transmit. Only Control Octet is transmitted for PDOUT_INVALID.

DLink for W-Sub-cycle x+…: nothing to transmit

12.7 Examples for uplink data transmissions

Note:
Maximum uplink payload of SSlot (2 octet) see Figure 37.
Maximum uplink payload of DSlot (15 octet) see Figure 38.
Size of ULink Control Octet (1 octet) see Figure 136.

12.7.1 DSlot W-Device sends 8 octets not segmented Process Data In to W-Master

W-Cycle x:
- Control Octet: ChC = 1; FC = 18 (data length = 8)
- Data: 8 octets Process In data following the Control Octet

12.7.2 DSlot W-Device sends 32 octets segmented Process Data In to W-Master

W-Cycle x:
- Control Octet: ChC = 1; FC = 8 (Segment Start)
- Data: 14 octets Process In data (ULink payload filled completely with Control Octet and data)

W-Cycle x+1:
- Control Octet: ChC = 1; FC = 1 (Segment Counter)
- Data: 14 octets Process In data (ULink payload filled completely with Control Octet and data)

W-Cycle x+2:

Control Octet: ChC = 1; FC = 14 (data length = 4)
Data = 4 octet Process In data. Segmented data transmission is complete.

12.7.3 SSlot W-Device responds with 3 octets segmented ISDU Data to W-Master

W-Cycle/ W-Sub-cycle x*:
Control Octet: ChC = 3; FC = 8 (Segment Start)
Data: 1 octet ISDU data following the Control Octet
W-Sub-cycle x+1:
Control Octet: ChC = 3; FC = 1 (Segment Counter)
Data: 1 octet ISDU data following the Control Octet
W-Sub-cycle x+2:
Control Octet: ChC = 3; FC = 2 (Segment Counter)
Data: 1 octet ISDU data following the Control Octet
W-Sub-cycle x+3:
Control Octet: ChC = 3; FC = 9 (EOS)
Data: No data to transmit, W-Message contains the separate EOS for ISDU.

* W-Cycle/ W-Sub-cycle x*: A W-Device can send ISDU-data also in a W-Cycle, if no process data are available to send.

12.7.4 DSlot W-Device sends 4 octets Process Data In every 5 ms and responds with 25 octets segmented ISDU Data to W-Master

W-Cycle x:
Control Octet: ChC = 1; FC = 14 (data length=4)
Data: 4 octets Process In data following the Control Octet
Control Octet: ChC = 3; FC = 8 (Segment Start)
Data: 9 (15-6) octet ISDU-data (ULink payload filled up with ISDU-data)
W-Sub-cycle x+1:
Control Octet: ChC = 3; FC = 1 (Segment Counter)
Data: 14 octet ISDU data (ULink payload filled completely with Control Octet and ISDU-data)
W-Sub-cycle x+2:
Control Octet: ChC = 3; FC = 12 (data length = 2)
Data: 2 octet ISDU data
W-Cycle x+3:
Control Octet: ChC = 1; FC = 14 (data length=4)
Data: 4 octets Process In data following the Control Octet
Control Octet: ChC = 3; FC = 9 (EOS)
Data: No data to transmit, W-Message contains the separate EOS for ISDU.
W-Sub-cycle x+4: unused – no ULink to send
W-Sub-cycle x+5: unused – no ULink to send
W-Cycle x+6:
Control Octet: ChC = 1; FC = 14 (data length = 4)
Data: 4 octet process data following the Control Octet

*If the W-Device send process data, the W-Cycle is used to transmit them. Additionally, acyclic ISDU- or Event- data can be added to fill up the ULink payload. Further, ISDU- or Event- data are transmitted in the following W-Sub-cycles, if they are not needed to retransmit process data.

12.8 User data (PD or OD)

User data is a general term for both, Process Data and On-request Data. The length of user data can vary from 0 to 35 octets depending on the transmission direction (downlink or uplink) and the W-Device's SlotType (DSlot or SSlot). An overview of the available data types is shown in Table 134. These data types can be arranged as records (different types) or arrays (same types).

Table 134 Data types for user data

Data type	Reference
BooleanT	See E.2 in REF 1
UIntegerT	See E.2.3 in REF 1
IntegerT	See E.2.4 in REF 1
StringT	See E.2.6 in REF 1
OctetStringT	See E.2.7 in REF 1
Float32T	See E.2.5 in REF 1
TimeT	See E.2.8 in REF 1
TimeSpanT	See E.2.9 in REF 1

12.9 PDVALID PDINVALID

To support low energy W-Devices the minimization of data transmission is necessary. Due to this it is possible to exchange process data only on a change of them. If process data becomes invalid it shall not be send any more. PDx_INVALID is transmitted via AL_Control / DL_Control and the ULink control octets instead.
The generation of PDVALID or PDINVALID is specified in the following way:

Table 135 PDVALID PDINVALID

PDIN_VALID:	With each reception of process input data from a W-Device, the W-Masters PDIN data handler generates PDIN_VALID automatically (see Table 65 MASTER-PDIn handler, T5).
PDIN_INVALID:	W-Device application sends PDIN_INVALID via AL/DL_Control and the ULink control octet to the W-Master.
PDOUT_VALID:	With each reception of process output data from W-Master, the devices PDIN data handler generates PDOUT_VALID automatically (see Table 66. DEVICE-PDOut handler, T5)
PDOUT_INVALID:	W-Master application sends PDOUT_INVALID via AL/DL_Control and the DLink control octet to the W-Device.

12.10 General structure and encoding of ISDUs

The encoding of ISDU data delivered by the ISDU handler shall be implemented equal to IO-Link, see 7.4.3.

12.11 General structure and encoding of Events

12.11.1 EventQualifier

The structure of the EventQualifier is shown in Figure 138

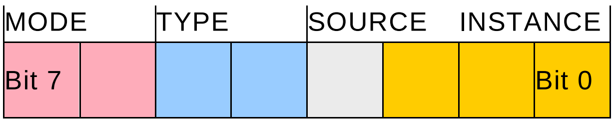


Figure 138 Structure of the EventQualifier

12.11.1.1 Bits 0 to 2: INSTANCE

These bits indicate the particular source (instance) of an Event thus refining its evaluation on the receiver side. Permissible values for INSTANCE are listed in Table 136

Table 136 Values of INSTANCE

Value	Definition
0	Unknown
1 to 3	Reserved
4	Application
5 to 7	Reserved

12.11.1.2 Bit 3: SOURCE

This bit indicates the source of the Event. Permissible values for SOURCE are listed in Table 137

Table 137 Values of SOURCE

Value	Definition
0	W-Device (remote)
1	W-Master (local)

12.11.1.3 Bits 4 to 5: TYPE

These bits indicate the Event category. Permissible values for TYPE are listed in Table 138.

Table 138 Values of TYPE

Value	Definition
0	Reserved
1	Notification
2	Warning
3	Error

12.11.1.4 Bits 6 to 7: MODE

These bits indicate the Event mode. Permissible values for MODE are listed in Table 139.

Table 139 Values of MODE

Value	Definition
0	reserved
1	Event single shot
2	Event disappears
3	Event appears

12.11.2 EventCode

The EventCode entry contains the identifier of an actual Event. Permissible values for EventCode are listed in clause 15

Annex B

(normative)

13 W-Frame Codings, CRC calculation and errors

13.1 Description of ConnectionParameter

The ConnectionParameter in Table 140 describe a subset of parameters which are necessary for a communication in Cyclic Mode. These parameters are transmitted to the W-Device during pairing and are managed by Medium Access Layer (MAC Layer). These parameters are not accessible by application. These parameters shall be stored in non-volatile memory if the W-Device is used as Normal-Device. These parameters shall be stored in volatile memory only if the W-Device is used as Roaming-Device

The parameters are listed in Table 140.

Table 140 Description of ConnectionParameter

ConnectionParameter	TYPE
MasterID	5 Bit (1-29)
Slot_N	3 Bit (0-7)
Track_N	3 Bit (0-4)
HoppingTable	Octet String
DataSyncword	3 Octet (see Figure 31)

13.2 W-Frame Downlink encodings for Normal Operation

The Figure 139 shows the general structure of the Downlink part of the W-Frame within a W-Sub-cycle from W-Master to W-Device. The Downlink includes the Pre-Downlink part ending with the CRC16. The remaining octets to the CRC32 reflects the payload space, which carry cyclic and acyclic data in Cyclic Mode. Unused fields must be filled with zeros.

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble		DataSyncword	
4	DataSyncword	MasterID (bits 0-4), Track_N (bits 5-7)	ACK	Payload
8	Payload	CRC16		Payload
12	Payload			
16	Payload			
..	Payload			
44	Payload			
48	CRC 32			

Figure 139 W-Frame encodings

Table 141 MasterID

Value	Meaning
0	invalid
1..29	Valid MasterID

Table 142 Track_N

Value	Meaning
0..4	Valid Track_N

Table 143 Normal Downlink ACK

Value	Meaning
0..7	Valid ACK for devices 0...7 as bit-fields

13.3 W-Frame Downlink encodings for Configuration Operation

In ServiceMode, the configuration channels are utilized to transmit configuration requests in downlink direction towards the W-Device. The ServiceMode covers Scan, Pairing and Negotiation procedures. The downlink message types listed in Table 144 shall be implemented and used during configuration.

Table 144 Downlink-MSG-Type content (Config Mode only)

Value	Meaning	Payload Content
0x80	MSG_DLink_Pair_Button	MasterID + ULink type + ACK + Device_N + IMATime + Retry Count
0x90	MSG_DLink_Pair_Unique	MasterID + ULink type + ACK + Device_N + IMATime + Retry Count+ UniqueID
0x40	MSG_DLink_Scan_Req	MasterID + ACK + RequestN
0xA0	MSG_DLink_Pair Neg 1	MasterID + ACK + Device_N + Hopping Table (Part 1)
0xB0	MSG_DLink_Pair Neg 2	MasterID + ACK + Device_N + Hopping Table (Part 2) + Col-N

Table 145 Uplink Type

Value	Meaning
00	Single Slot Uplink
01	Double Slot Uplink

Table 146 Config Downlink ACK

Value	Meaning
0	no packet received
1	last packet received

Table 147 Downlink-MSG-Type coding

Types	Hex Code	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSG_DLink_Pair_Button	0x80	1	0	0	0	X	X	X	X
MSG_DLink_Pair_Unique	0x90	1	0	0	1	X	X	X	X
MSG_DLink_Scan_Req	0x40	0	1	0	0	X	X	X	X
MSG_DLink_Pair Neg 1	0xA0	1	0	1	0	X	X	X	X
MSG_DLink_Pair Neg 2	0xB0	1	0	1	1	X	X	X	X

Table 148 Device_N

Bit	5	4	3	2	1	0
Meaning	Track_N (2)	Track_N (1)	Track_N (0)	Slot_N (2)	Slot_N (1)	Slot_N (0)

Table 149 Roaming Flag

Value	Meaning
00	Roaming not requested
01	Roaming requested

Table 150 Track_N

Value	Meaning
0-4	Valid
5-7	Invalid

Table 151 Slot_N

Value	Meaning
0-7	Valid

13.3.1 Scan Request Downlink

In Scan Mode and Roaming Mode, the W-Master is able to discover unpaired W-Devices. This is achieved by transmitting Scan Request messages shown in Figure 140 in configuration downlinks.

After receiving a Scan Request, W-Devices shall respond with the Scan Response Uplink after a random number of W-Sub-cycles, as described in clause 12.6.1.

The W-Master should transmit its MasterID, an Acknowledge for last received Uplinks, the Scan Request identifier and the consecutive number of Scan Request as Request_N in each configuration Downlink during ServiceMode.

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble		ConfigSyncword	
4	ConfigSyncword	MasterID, unused	MSG_DLink_type, ACK	Request_N
8	Request_N	unused		
12	unused			
..	unused			
44	unused			
48	CRC 32			

Figure 140 Scan Request

13.3.2 Pairing Request Downlink

In ServiceMode (Pairing State), the W-Master has to address a specific unpaired W-Device. Therefore, the W-Master starts the configuration process with sending Pairing Requests shown in Figure 141 in configuration Downlinks. The addressed W-Device shall answer with a Pairing Response Uplink within the same W-Sub-cycle.

Each Pairing Request shall contain the ID of the W-Master, requested Uplink type (SSlot Uplink or DSlot Uplink), the Acknowledge for the last received Uplink, the Pairing Request command, the roaming flag, the W-Device number, UniqueID, and DataSyncword.

If a W-Device receives an active Roaming Flag in a Pairing Request, it changes its mode to Roaming mode. In this mode, the Pairing by Button and Re-pairing features are deactivated on the W-Device

ServiceMode supports two pairing mechanisms:
- Pairing Request by Button.
- Pairing Request by UniqueID

During Paring Request by Button, the UniqueID shall be set to zero. In this case, the W-Master does not address the W-Device. Only the W-Device which was already set into the Pairing by Button mode shall respond on the W-Master request.

Pairing Request by UniqueID transfers the UniqueID of the W-Device the W-Master tries to pair. Pairing by UniqueID is used for two cases: pairing of the W-Device during system configuration or temporarily pairing of W-Device in Roaming mode.

<table>
<tr><td>Octet</td><td>1</td><td colspan="3">2</td><td>3</td><td colspan="2">4</td></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>0 1 2 3 4</td><td>5 6</td><td>7</td><td>0 1 2 3 4 5 6 7</td><td>0 1</td><td>2 3 4 5 6 7</td></tr>
<tr><td>0</td><td colspan="4">Preamble</td><td colspan="3">ConfigSyncword</td></tr>
<tr><td>4</td><td>ConfigSyncword</td><td>MasterID</td><td>UL type</td><td>ACK</td><td>MSG_DLink_type</td><td>Roaming Flag</td><td>Device_N</td></tr>
<tr><td>8</td><td colspan="7">UniqueID</td></tr>
<tr><td>12</td><td colspan="7">UniqueID</td></tr>
<tr><td>16</td><td>UniqueID</td><td colspan="6">DataSyncword</td></tr>
<tr><td>..</td><td colspan="7">unused</td></tr>
<tr><td>44</td><td colspan="7">unused</td></tr>
<tr><td>48</td><td colspan="7">CRC 32</td></tr>
</table>

Figure 141 Pairing Request

Table 152 Pairing Request: Unique ID

Value	UniqueID
0x000000000000	Pair by Button
0x000000000001- 0xFFFFFFFFFF	Pair by Unique ID

13.3.3 Pairing Negotiation Downlink

In ServiceMode within the Paring Procedure, the Negotiation Downlinks are used by W-Master for configuration of the W-Device. There are two mandatory consecutive Negotiation Downlinks necessary to be able to transmit frequency tables. The unused fields at the end are filled with zero. The hopping sequence itself is encoded in the given sequence of the channels, each octet reflecting a 1 MHz channel in the 2.4 GHz-ISM-Band.

Negotiation Downlinks are containing the MasterID, Uplink Slot Type of the W-Device being configured, the Acknowledge of the last received Uplink, Downlink-MSG-type (MSG_DLink_Pair Neg 1 or MSG_DLink_Pair Neg 2), the Device_N (combination of Slot_N, Track_N), the actual frequency hopping table length, current Col_N of Cyclic Mode and the frequency hopping table of Cyclic Mode.

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 | 5 | 6 | 7	0 1 2 3 4 5 6 7	0 | 1 | 2 3 4 5 6 7
0	Preamble		ConfigSyncword	
4	ConfigSyncword	MasterID (bits 0-4), UL type (bits 5-6), ACK (bit 7)	MSG_DLink_type	Reserved (bits 0-1), Device_N (bits 2-7)
8	Table length	HOP-1	HOP-2	HOP-3
12	HOP-4	HOP-5	HOP-6	HOP-7
16	HOP-8	HOP-9	HOP-10	HOP-11
20	HOP-12	HOP-13	HOP-14	HOP-15
24	HOP-16	HOP-17	HOP-18	HOP-19
28	HOP-20	HOP-21	HOP-22	HOP-23
32	HOP-24	HOP-25	HOP-26	HOP-27
36	HOP-28	HOP-29	HOP-30	HOP-31
40	HOP-32	HOP-33	HOP-34	HOP-35
44	HOP-36	HOP-37	HOP-38	HOP-39
48	CRC 32			

Figure 142 Pairing Negotiation type 1 => DLink-Message-Type = MSG_DLink_Pair_Neg_1

Table 153 Values for Frequency Table length

Value	Meaning
0-14, 79-255	invalid
15-78	Valid table length

Table 154 Permitted Values for HOP_N

Value	Meaning
0,2,3–79,81-83	Valid frequency for cyclic data channel
0	End of Frequency Table Delimiter

Table 155 HOP_N Bit coding

Bit	7	6	5	4	3	2	1	0
Meaning	0	HOP_N (6)	HOP_N (5)	HOP_N (4)	HOP_N (3)	HOP_N (2)	HOP_N (1)	HOP_N (0)

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble		ConfigSyncword	
4	ConfigSyncword	MasterID, UL type, ACK	MSG_DLink_type	Reserved, Device_N
8	Col_N	HOP-40	HOP-41	HOP-42
12	HOP-43	HOP-44	HOP-45	HOP-46
16	HOP-47	HOP-48	HOP-49	HOP-50
20	HOP-51	HOP-52	HOP-53	HOP-54
24	HOP-55	HOP-56	HOP-57	HOP-58
28	HOP-59	HOP-60	HOP-61	HOP-62
32	HOP-63	HOP-64	HOP-65	HOP-66
36	HOP-67	HOP-68	HOP-69	HOP-70
40	HOP-71	HOP-72	HOP-73	HOP-74
44	HOP-75	HOP-76	HOP-77	HOP-78
48	CRC 32			

Figure 143 Pairing Negotiation type 2 => DLink-Message-Type = MSG_DLink_Pair_Neg_2

13.4 Uplink encodings for Normal Operations

13.4.1 Regular Single Slot Uplink (SSlot)

In Cyclic Mode, the Regular Uplink packet shown in Figure 144 is used to transmit process and event data from W-Device to the W-Master. A message in SSlot Uplink telegram can handle 16 bit data payload, which can contain cyclic process data, diagnosis data or event notifications.

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble		DataSyncword	
4	DataSyncword	MasterID, IMA=0, ACK	Payload	
8	CRC 32			

Figure 144 Regular SSlot Uplink Packet

13.4.2 Regular Double Slot Uplink (DSlot)

In Cyclic Mode, the Regular DSlot Uplink packet is used to transmit process and event data from W-Device to W-Master.

Octet	1	2	3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble	Preamble	DataSyncword	DataSyncword
4	DataSyncword	MasterID, IMA= 0, ACK	Payload	Payload
8	Payload	Payload	Payload	Payload
12	Payload	Payload	Payload	Payload
16	Payload	Payload	Payload	Payload
20	Payload	CRC 32	CRC 32	CRC 32
24	CRC 32			

Figure 145 Regular DSlot Uplink Packet

Table 156 Uplink IMA

Value	Meaning
0	Normal Uplink
1	IMA Uplink

Table 157 Uplink ACK

Value	Meaning
0	no packet received
1	last packet received

13.4.3 IMA Uplink

W-Master as well as W-Device controls the time between two successive Uplink packets. If this time in W-Master is greater than the defined IMA time, an Event should be initiated by the W-Master application. If this time in W-Device is greater than defined IMA time the W-Device Message handler causes an IMA packet with diagnosis data to avoid an IMA alert at the W-Master.
Depending on Uplink Type, the W-Device uses an IMA D-Slot Uplink see Figure 146 or an IMA Slot Uplink see Figure 147. IMA Uplinks should send an IMA=1 Flag, the Acknowledge for Previously received packet and diagnosis data.

octet	1	2	3	4
	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble	Preamble	DataSyncword	DataSyncword
4	DataSyncword	MasterID, IMA =1, ACK	RSSI	LinkQuality
8	unused	unused	unused	unused
12	unused	unused	unused	unused
16	unused	unused	unused	unused
20	unused	CRC 32	CRC 32	CRC 32
24	CRC 32			

Figure 146 DSlot IMA-Uplink Packet

octet	1	2			3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7			0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble				DataSyncword	
4	DataSyncword	MasterID	IMA =1	ACK	RSSI	LinkQuality
8	CRC 32					

Figure 147 SSlot IMA-Uplink-Packet

Table 158 Diagnosis encoding octet 3 (RSSI)

Bit	7	6	5	4	3	2	1	0
Meaning	RSSI No 7	RSSI No 6	RSSI No 5	RSSI No 4	RSSI No 3	RSSI No 2	RSSI No 1	RSSI No 0

Table 159 Diagnosis encoding octet 4 (Link Quality)

Bit	7	6	5	4	3	2	1	0
Meaning	Link Quality No 7	Link Quality No 6	Link Quality No 5	Link Quality No 4	Link Quality No 3	Link Quality No 2	Link Quality No 1	Link Quality No 0

13.5 Uplink encodings for Configuration Operations

In the ServiceMode the System Management, DL-A/B Message handlers are not involved in the Uplink assembly, therefore the data flow control shall be implemented in the MAC layers of the IO-Link wireless stack.
The ServiceMode itself covers Scan, Pairing and Negotiation procedures. Therefore, five message types, presented in the Uplink-MSG-Type tables, shall be implemented und used during configuration.

13.5.1 Definition of Uplink encodings

Table 160 Uplink-MSG-Type (Config Mode only)

Value	Meaning	Payload Content
0x40	MSG_UPLINK_Scan_Resp	RevisionID + IMATime + UniqueID
0x80/0x90	MSG_UPLINK_Pair_Resp	RevisionID + IMATime + UniqueID
0xA0	MSG_UPLINK_Pair_Neg_1_Resp	Response Only, (no data transfer)
0xB0	MSG_UPLINK_Pair_Neg_2_Resp	Response Only, (no data transfer)
0xF0	MSG_UPLINK_Pair_Failed	Response Only, (no data transfer)

Table 161 Slot-Type in config Uplink

Value	SlotType
00	Single Slot
01	Double Slot

Table 162 Uplink-MSG-Type Coding

Header Type	Hex Code	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSG_UPLINK_Scan_ Resp	0x40	0	1	0	0	X	X	X	X
MSG_UPLINK_Pair_Button_Resp	0x80	1	0	0	0	X	X	X	X
MSG_UPLINK_Pair_Unique_Resp	0x90	1	0	0	1	X	X	X	X
MSG_UPLINK_Pair Neg 1_Resp	0xA0	1	0	1	0	X	X	X	X
MSG_UPLINK_Pair Neg 2_Resp	0xB0	1	0	1	1	X	X	X	X
MSG_UPLINK_Pair_Failed	0xF0	1	1	1	1	X	X	X	X

Table 163 RevisionID

Bits	Value	Meaning
0 to 3	0x0…0xF	MinorRev part of the protocol revision (see page 217 in REF 1)
4 to 7	0x0…0xF	MajorRev part of the protocol revision (see page 217 in REF 1)

13.5.2 Scan Response Uplink

In ServiceMode the W-Device answers to a received Scan Request with a Scan Response. A Scan Response shown in Figure 148 must contain the MasterID received in the Scan Request, the Uplink type, an Uplink Message type, the RevisionID, and its UniqueID.

Octet	1								2								3								4							
Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	Preamble																ConfigSyncword															
4	ConfigSyncword								MasterID					UL type		ACK	MSG_UPLINK_type								RevisionID							
8	UniqueID																															
12	UniqueID																															
16	UniqueID								unused																							
20	unused								CRC 32																							
24	CRC 32																															

Figure 148 Scan Response Packet

13.5.3 Pairing Response Uplink

In ServiceMode, the W-Device shall answer to a Pairing Request Downlink with a Pairing Response Uplink within the same W-Sub-cycle. The W-Device shall submit the received MasterID, the Uplink Type of the W-Device, the acknowledge for the last received Downlink, the RevisionID and the UniqueID of the W-Device as shown in Figure 149.

Octet	1								2								3								4							
Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	Preamble																ConfigSyncword															
4	ConfigSyncword								MasterID					UL type		ACK	MSG_UPLINK_type								RevisionID							
8	UniqueID																															
12	UniqueID																															
16	UniqueID								unused																							
20	unused								CRC 32																							
24	CRC 32																															

Figure 149 Pairing Response Packet

13.5.4 Negotiation Response Uplink

In ServiceMode, the W-Device shall respond on each Negotiation Downlink it receives. W-Device shall submit the Uplink Type of the W-Device, the acknowledge for the last received Downlink, MSG_UPLINK and the W-Device RevisionID as shown in Figure 150.

Octet	1	2			3	4
Bit	0 1 2 3 4 5 6 7	0 1 2 3 4	5 6	7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0	Preamble	Preamble			ConfigSyncword	ConfigSyncword
4	ConfigSyncword	MasterID	UL type	ACK	MSG_UPLINK_type	RevisionID
8	UniqueID	UniqueID			UniqueID	UniqueID
12	UniqueID	UniqueID			UniqueID	UniqueID
16	UniqueID	unused			unused	unused
20	unused	CRC 32			CRC 32	CRC 32
24	CRC 32					

Figure 150 Pairing Negotiation Uplink Packet

13.6 Acknowledge Generation

The PL in W-Master shall generate an ACK-Bit (see Figure 139) for each W-Device, if the W-Master received a valid Uplink.
The PL in W-Device shall generate an ACK-Bit (see Figure 144 and Figure 145) if the W-Device received a valid Downlink with data for this specific W-Device from W-Master.
In both cases: If no acknowledge within an Uplink is received or the Uplink is lost or invalid (e.g. wrong CRC) a NACK (ACK-Bit = 0) is generated.

13.7 CRC16 and CRC32 calculation

The integrity of Uplink and Downlink transmissions is protected through 32 bit CRC defined in IEEE 802.3 (CRC32).
The integrity of Pre-Downlink shall be protected through CRC16-CCITT (CRC16).

The CRC algorithms are defined as follows:

The CRC32 Generator polynomial is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
Initial Value (Pre-set) 0xFFFFFFFF.
The final xor (residue) during transmission: 0xFFFFFFFF
The final xor (residue) during reception: 0xC704DD7B

Note: In Cyclic Mode, the final XOR value shall be updated during the Uplink exchange by W-Master and W-Device with the “W-Device distinguishing identifier” as following:
The final xor during reception shall be set to [0xC704DD7B xor W-Device distinguish identifier]

CRC16:
generator polynomial is $x^{16} + x^{12} + x^{5} + 1$
Initial Value (Pre-set) 0xFFFF.
The final xor during transmission and reception 0x0000

The CRC16 calculated over Pre-Downlink only and placed at its end.

13.8 Errors

The Acknowledgement bit/bits and the checksum are two independent mechanisms to secure the data transfer.
Remedy: The W-Master or W-Device can repeat the packet for maximum 2 times (see clause 4). DL-A/DL-B handler in W-Master or W-Devices assumes content of the payload within the next W-Sub-cycle.

13.8.1 Checksum errors

Any checksum error in a receiver suppress it’s acknowledge to the transmitter.

13.8.2 Latency errors

The latency error occurs if an expected cyclic message is not received within the W-Cycle.

13.8.3 IMA Timeout errors

IMA Timeout errors occurs if the configured IMA time at the W-Master is exceeds.

13.8.4 False positive Error

False Positive errors occurs when interference falsifies a transmitted packet in a way the CRC and other integrity checks at the receiver cannot detect.

Annex C

(normative)

14 W-Device Parameter and commands

This section describes and defines the parameters and commands within a W-Device. Compared to the IO-Link Interface and System Specification, the page communication channel is not implemented in wireless IO-Link. Thus index 0 and 1 remain solely accessible using the ISDU channel. For compatibility reasons towards IO-Link Interface and System Specification, the memory structure of page 1 and 2 is kept. A detailed memory mapping for W-Devices can be found in Figure 151 and for W-Bridges in Figure 152.

The wireless parameters are addressed via index 0x5000 to 0x50FF.

All other mechanisms described in the IO-Link Interface and System Specification are fully supported, for a more complete description please refer to the IO-Link Interface and System Specification REF 1. For W-Devices, the use of profile(s) is recommended e.g. smart sensor profile and common profile, see REF 4.

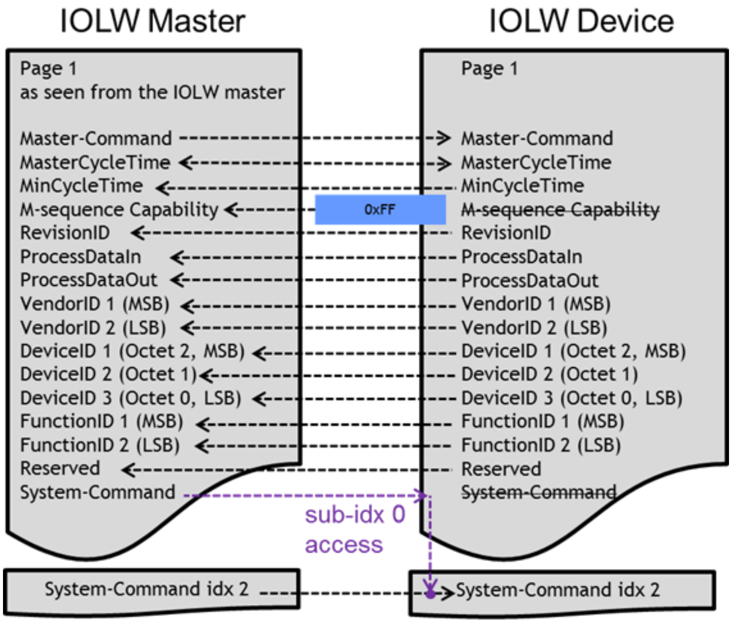


Figure 151 Memory mapping of the direct parameter page 1 of a W-Master with a W-Device.

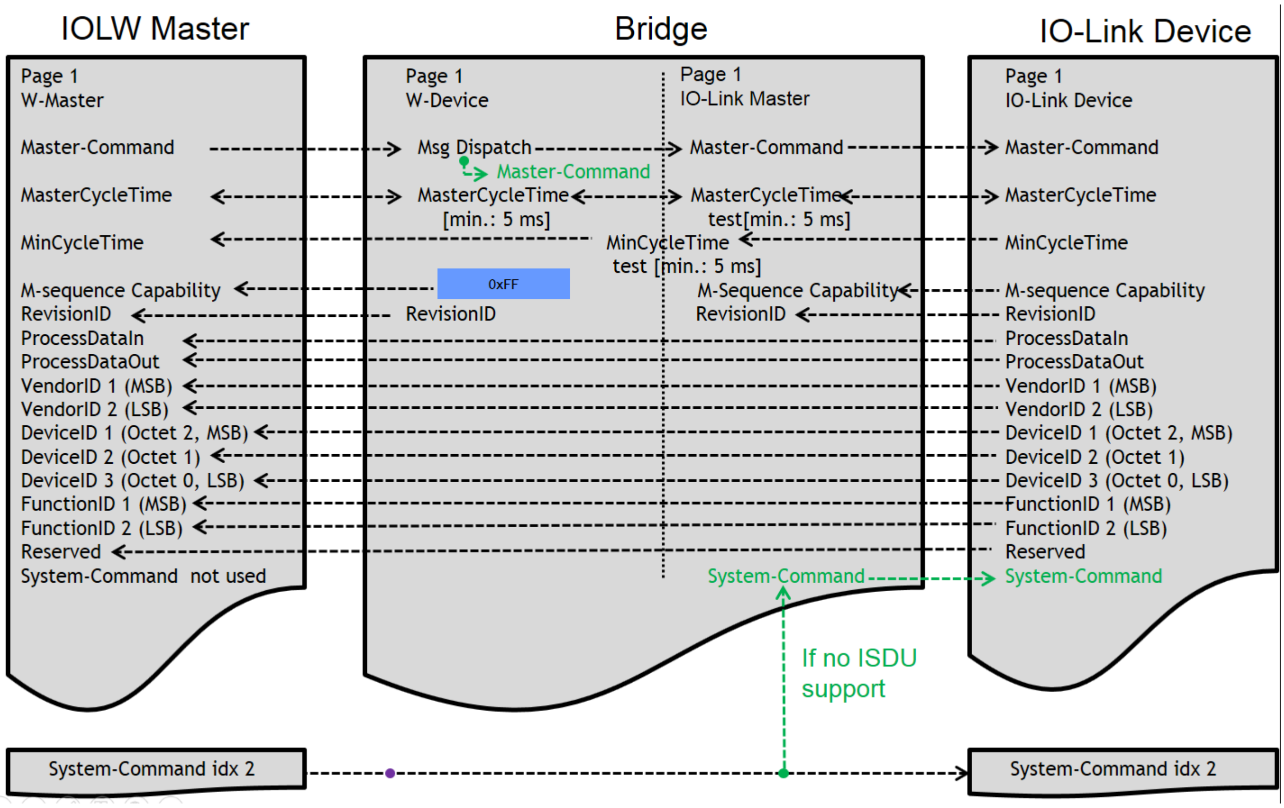


Figure 152 Memory mapping of the direct parameter page 1 of a W-Master with a W-Bridge connected to an IO-Link device.

14.1 Direct Parameter Page 1

For compatibility reasons towards IO-Link Interface and System Specification, the direct parameter page 1 is kept identical in its structure. This allows in the case of a W-Bridge application with a wired IO-Link Device in most cases a straight forward mapping of the parameters, see Figure 152.

Despite having the same direct parameter structure, wired and wireless devices differ in the following way:

- A read request on idx 0 sub-idx 0 returns the whole page 1
- A read request on idx 0 sub-idx 4 (i.e. M-SequenceCapability) returns 0xFF.
- A write request on idx 0 sub-idx 0 is ignored by the subindices which are “read-only”
- A write request on idx 0 sub-idx 10 is redirected within the device towards idx 2.

Table 164 Direct Parameter Page 1

Index	Subindex	Access	Parameter name	Description	Implementation / reference
	0x00	To read the whole Direct Parameter Page 1 in one go			
0x0000	0x01	W	MasterCommand	Master command to switch to operating states (see Note 2)	Mandatory/ see 14.1.1
	0x02	R/W	MasterCycleTime	Identical to IO-Link Interface and System Specification	Mandatory/ see 14.1.2
	0x03	R	MinCycleTime	Identical to IO-Link Interface and System Specification	Mandatory/ see 14.1.2
	0x04	R	M-Sequence Capability	Not used: the byte is set to 0xFF	-
	0x05	R/W	Revision ID	ID of the used RevisionID for implementation (shall be set to 0x11)	Mandatory/ see 14.1.3
	0x06	R	ProcessDataIn	Number and structure of input data (Process Data from Device to W-Master)	Mandatory/ see 14.1.4
	0x07	R	ProcessDataOut	Number and structure of output data (Process Data from W-Master to W-Device)	Mandatory/ see 14.1.5
	0x08	R	VendorID (MSB)	Unique vendor identification	Mandatory/ see 14.1.6
	0x09	R	VendorID (LSB)		
	0x0A	R/W	DeviceID 1 (Octet 2, MSB)	Unique Device identification allocated by a vendor	Mandatory/ see 14.1.7
	0x0B	R/W	DeviceID 2 (Octet 1)		
	0x0C	R/W	DeviceID 3 (Octet 0, LSB)		
	0x0D	R	FunctionID 1 (MSB)	Reserved (Engineering shall set both octets to "0x00")	see 14.1.8
	0x0E	R	FunctionID 2 (LSB)		
	0x0F	-	-	-	-
	0x10	-	System-Command	Not used (see Note 1)	-

NOTE 1 For all IO-link wireless device SystemCommand on page 1 shall not be used, but index 2 instead.
NOTE 2 A Read operation returns unspecified values

14.1.1 MasterCommand

The W-Master application is able to check the status of a W-Device or to control its behavior with the help of MasterCommands. The permissible value definitions for these parameters are specified in Table 165.

Table 165 Types of MasterCommands.

MasterCommand		
Value	**MasterCommand**	**Description**
0x00 to 0x5B	Reserved	
0x5C	Inactive	Switches the W-Device state machines to inactive
0x5D	PreDLink	Switches the W-Device radio to receive Pre-Downlink W-frames
0x5E	FullDLink	Switches the W-Device radio to receive full Downlink W-frames
0x5F	UnParing	Unpairs the W-Device. The W-Device deletes all its wireless communication parameters. Invoke PL_Pairing(UNPAIRING) on W-Device
0x60 to 0x95	Reserved	
0x96	DeviceIdent	Start check of Direct Parameter page for changed entries
0x97	DeviceStartup	Switches the Device from OPERATE or PREOPERATE to STARTUP
0x98	Reserved	
0x99	DeviceOperate	Process output data invalid or not available. Switches the Device from STARTUP or PREOPERATE to OPERATE
0x9A	DevicePreoperate	Switches the Device from STARTUP to state PREOPERATE
0x9B to 0xEF	Reserved	
0xF0 to 0xFE	Jump (Broadcast)	Triggers a countdown starting with the value of last 4 bits (14 to 0) for updating the hopping table to new values. MCmd is sent (decrease 1 each sub cycle) until all W-Devices have acknowledged. When counter reaches 0 then W-Master and W-Device shall use the new hopping table immediately, starting with the first frequency (HOP-1, see Figure 142)
0xFF	WakeUp	Set low energy W-Device WakeUpTimer to 0 which increases every sub cycle. W-Device wakes up when WakeUpTimer reaches WakeUpTime
NOTE For low energy W-Devices, the Pre-Downlink is used to minimize the radio-on time to save power. A switch to full Downlink may be necessary on low energy W-Device s for a higher amount of data e.g. a parameter write.		

14.1.2 MasterCycleTime and MinCycleTime

The MasterCycleTime is a Master parameter and sets up the actual cycle time of a particular W-Port.

The MinCycleTime is a W-Device parameter to inform the W-Master about the shortest cycle time supported by this W-Device.

The structure of these two parameters is shown in Figure 153.

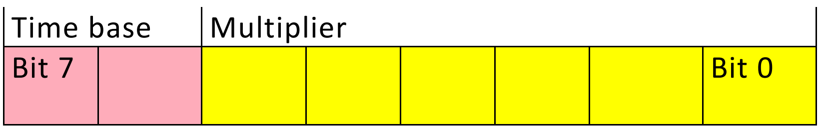


Figure 153 MasterCycleTime and MinCycleTime

Bits 0 to 5: Multiplier

These bits contain a 6-bit multiplier for the calculation of MasterCycleTime or MinCycleTime. Permissible values for the multiplier are 0 to 63.

Bits 6 to 7: Time Base

These bits specify the time base for the calculation of MasterCycleTime or MinCycleTime.
The permissible combinations for time base and multiplier are listed in Figure 154 along with the resulting values for MasterCycleTime or MinCycleTime.

Time base encoding	Time Base value	Calculation	Cycle Time
00	-	-	FreeRunning
01	5 ms	Multiplier × Time Base	5 ms to 315 ms
10 to 11	Reserved	Reserved	Reserved
NOTE For W-Devices and W-Bridges the minimum possible transmission time is 5 ms.			

Figure 154 Possible values of MasterCycleTime and MinCycleTime

14.1.3 Revision ID

Identical to IO-Link Interface and System Specification: Section B.1.5 in REF 1

The RevisionID numbers of the wired and wireless are independent. This revision of the standard specifies RevisionID 1.1 (i.e. RevisionID=0x11).

14.1.4 ProcessDataIn

Identical to IO-Link Interface and System Specification: Section B.1.6 in REF 1

The exact size and content of the PD is described by the profile, e.g. smart profile for sensors. Due to the nature of the wireless protocol the slot sizes stay unchanged in any case.

14.1.5 ProcessDataOut

Identical to IO-Link Interface and System Specification: Section B.1.7 in REF 1

The exact size and content of the PD is described by the profile, e.g. smart profile for sensors. Due to the nature of the wireless protocol the slot sizes stay unchanged in any case

14.1.6 VendorID

Identical to IO-Link Interface and System Specification: Section B.1.8 in REF 1

14.1.7 DeviceID

Identical to IO-Link Interface and System Specification : Section B.1.9 in REF 1

14.1.8 FunctionID

Identical to IO-Link Interface and System Specification: Section B.1.10 in REF 1

14.1.9 SystemCommand

Devices with ISDU support shall use the ISDU Index 0x0002 to receive the SystemCommand. The commands shall be acknowledged. A positive acknowledge indicates the complete and correct finalization of the requested command. A negative acknowledge indicates the command cannot be realized or ended up with an error. A SystemCommand shall be executed within less than 5 s to fulfill the ISDU timing requirements (see Table 97 in REF 1). Implementation of the SystemCommand feature is mandatory for W-Masters and optional for Devices. The coding of SystemCommand is specified in Table 166. For wireless specific commands, the system command’s range from 0x30 to 0x3F is reserved and used.

Table 166 Coding of SystemCommand (ISDU)

Command (hex)	Command (dec)	Command name	M/O	Definition
0x00	0	Reserved	-	-
0x01	1	ParamUploadStart	O	Start parameter upload
0x02	2	ParamUploadEnd	O	Stop parameter upload
0x03	3	ParamDownloadStart	O	Start parameter download
0x04	4	ParamDownloadEnd	O	Stop parameter download
0x05	5	ParamDownloadStore	O	Finalize parameterization and start Data Storage
0x06	6	ParamBreak	O	Cancel all Param commands
0x07 to 0x3F	7 to 63	Reserved	-	-
0x40 to 0x7F	64 to 127	Reserved for profiles	-	-
0x80	128	Device reset	O	-
0x81	129	Application reset	O	-
0x82	130	Restore factory settings	O	-
0x83 to 0x9F	131 to 159	Reserved	-	-
0xA0 to 0xFF	160 to 255	Vendor specific	-	-
NOTE See 10.3 Key M = mandatory; O = optional				

14.2 Direct Parameter Page 2

The direct parameter page 2 shall not be used by W-Devices. Nevertheless, page 2 is kept to ensure backward compatibility in the case of a W-Bridge usage with IO-Link device, which are not ISDU compatible. For a pure W-Device a reading attempt on index 1 shall return a “Index not available” error message (error code: 0x80, Additional code: 0x11).

Table 167 Direct parameter Page 2

Index	Subindex	Access	Parameter name	Coding / description	Data type
0x0001	0x01… 0x10	Optional	Vendor Specific	Device Specific Parameters	-

14.3 Wireless Parameter (W-Parameter)

14.3.1 Overview

IO-Link wireless makes use of the same predefined device parameter as standard IO-Link devices. Nevertheless, in order to store the wireless specific parameters new indices have been predefined.

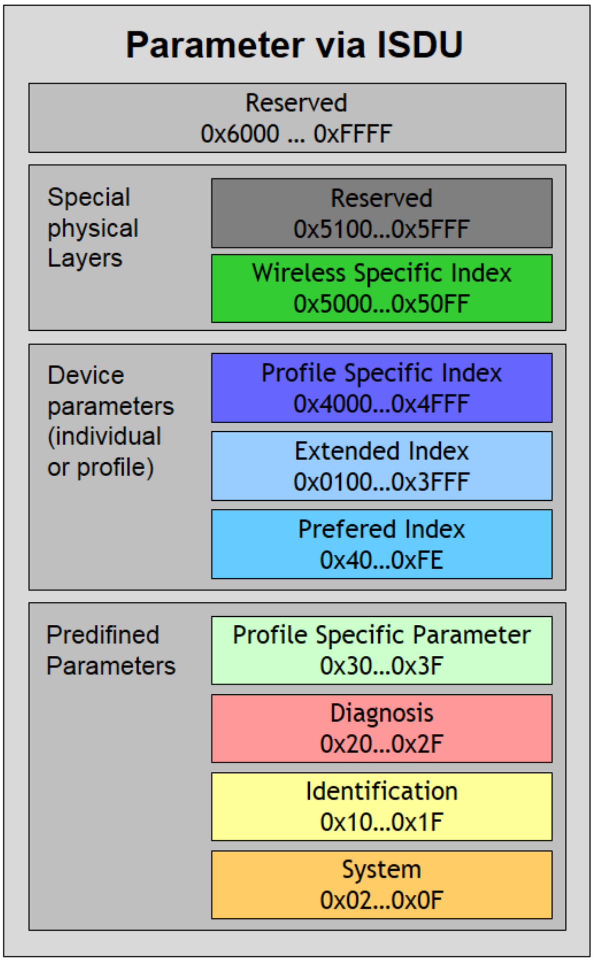


Figure 155 Index space for ISDU data objects

Table 168 Index assignment of data objects (W-Device parameter)

Index (dec)	Object name	Access	Length	Data type	M/O/C	Remark
0x0000 (0)	Page 1	R/W		RecordT	M	See Table 164
0x0001 (1)	Page 2	R/W		RecordT	O	Pure IO-Link wireless device shall not use this index. In the case of a W-Bridge it shall be implemented to ensure backward compatibility with IO-Link device.
0x0002 (2)	SystemCommand	W	1 octet	UIntegerT	M	See REF 1
0x0003… 0x0014 (3 to 20)	Similar to IOL.	-	-	-	-	See REF 1
0x0015 (21)	SerialNumber	R	Max. 16 octets	StringT	M	Vendor specific serial number (similar to IOL)
0x0016… 0x4FFF (22 to 20479)	Similar to IOL					See REF 1
0x5000 (20480)	Reserved	-	-	-	-	-
0x5001 (20481)	WirelessSystemMgmt	R	9 octets	RecordT	M	See Section: 14.3.3
0x5002 (20482)	WirelessSystemCfg	R/W	4 octets	RecordT	M	See Section: 14.3.3
0x5003 (20483)	LinkQuality	R	1 octet	UIntegerT	M	See Section: 14.3.10
0x5004 (20484)	WBridgeInfo	R	12 octets	RecordT	O	See Section: 14.3.11
0x5005 (20485)	WRadioInfo	R	12 octets	RecordT	M	See Section: 14.3.11
0x5006 (20486)	AdaptiveHopTable	W	82 octets	RecordT	M	See Section: 14.3.12
0x5007- 0x50FF (20487 to 20735)	Reserved					
0x5100… 0xFFFF (20736 to 65535)	Similar to IO-Link Interface and System Specification	-	-	-	-	See REF 1
Key M=Mandatory; O=optional; C=conditional						

14.3.2 SystemCommand

The ISDU Index 0x0002 shall be used to receive SystemCommands. Any received commands shall be acknowledged. A positive acknowledge indicates the complete and correct finalization of the requested command. A negative acknowledge indicates the command cannot be executed or terminated with an error. Any SystemCommand shall be executed within less than 5 s to fulfill the ISDU timing requirements. The W-Master may act as a proxy for a temporarily unreachable W-Device.
Implementation of the SystemCommand feature is mandatory for W-Masters and optional for Devices. The coding of SystemCommand is specified in Table 166.

14.3.3 Wireless System

This index range stores all the WirelessSystemMgmt and WirelessSystemCfg parameters of a W-Device.

Table 169 Wireless system index assignments

Index	Subindex	Access	Parameter name	Coding	Data type
0x5001	0x00	Gives access to the whole index			
	0x01	R	UniqueID	See Section: 14.3.9	OctetStringT9
0x5002	0x00	Gives access to the whole index			
	0x01	R/W	IMATime	See Section: 14.3.5	OctetStringT2
	0x02	R/W	MaxRetry	See Section: 14.3.6	UIntegerT8
	0x03	R/W	TxPower	See Section: 14.3.7	UIntegerT8

14.3.3.1 WRadioInfo

This index range stores all the information related to the radio hardware and software installed on the device.

Table 170 Radio manufacturer indentification information

Index	Subindex	Access	Parameter name	Coding	Data type
0x5005	0x00	Gives access to the whole index			
	0x01	R	RadioVendorID	similar to VendorID	OctetStringT2
	0x02	R	RadioModuleID	vendor specific similar to DeviceID	OctetStringT2
	0x03	R	RadioHWRevision	vendor specific	OctetStringT4
	0x04	R	RadioSWRevision	vendor specific	OctetStringT4

NOTE RadioVendorIDs are assigned by IO-Link community.

14.3.4 ISDU Mapping for system management

The Table 171 shows which ISDU indexes shall be forwarded to the system management.

Table 171 ISDU indexes for system management

Index	Object name
0x0000	Page 1
0x5001	WirelessSystemMgmt
0x5002	WirelessSystemCfg
0x5006	AdaptiveHopTable

14.3.5 IMATime

The IMA ("I am alive") time is a mandatory W-Parameter. IMATime is system and W-Device specific. Device manufacturer shall submit the maximal and minimal IMA times for each W-Device. (i.e. as mapped parameter in the W-Device itself). This information can be used by W-Master during configuring of the W-Device for performance optimization.
In Normal mode, W-Master and W-Device control the time between two successive uplink messages of each W-Device. If there are no other messages to transmit, the W-Device shall send an IMA message before IMA time will be reached. If IMA time is exceeded on W-Master, a communication error must be reported via system management and a failsafe may be performed by the application.

The minimum IMA time is dependent of the number of MaxRetry. Therefore the minimum IMA time shall be calculated shown in Table 172:

Table 172 Minimum and maximum IMA time

Minimum IMA time	W-Sub-cycle duration [ms] * (MaxRetry + 1)
Maximum IMA time	Limited to 10 minutes

The IMA time encoding is shown in Table 173:

Table 173 IMA time encoding

Octet 0	Octet 1
Time Base encoding see Table 174	Multiplier Permitted values: 1 to 255

A value of 0x01 means that the device stays always on.

Table 174 Time value encoding table for the IMATime

Time base encoding	Time Base Value	Conversion to W-Sub-cycles	Remark
0x00	-	-	Reserved
0x01	1.664 ms	1	Limits see Table 172
0x02	5 ms	3	
0x03	1 s	600	
0x04	1 minute	36,000	
0x05 …0xFF	-	-	Reserved

The IMATime value is calculated by multiplying the “time base” with the “multiplier”.

14.3.6 MaxRetry

This index stores the maximal number of retries. The minimum and default value is 0x02, thus one primary transmission and 2 retries.

Table 175 Value for the maximal number of retries

Value	Remark
0x00, 0x01	Reserved
0x02	2 retry
0x03	3 retry
0x04	4 retry
…	…
0x1F	31 retry
0x20 …0xFF	Reserved

14.3.7 TxPower

This parameter stores the currently used transmission power. The transmission power is encoded in predefined power levels which values shall be defined in the vendor’s documentation. If those values are not otherwise specified the values in Table 12 are valid. If the requested power value is not support by the radio, the later shall round the Tx Power value to the closest matching one and correct the stored value accordingly. The corrected value replaces then the original value.

Table 176 TxPower parameter

TxPower	Predefined Power Level	Values [dBm]
0x00	-	Reserved
0x01	Level 1	-20
0x02	Level 2	-19
…	…	…
0x14	Level 20	0
…	…	…
0x0F	Level 30	9
0x1F	Level 31	10
0x20 – 0xFF	-	Reserved

14.3.8 SerialNumber

This mandatory parameter shall contain a unique vendor specific code for each individual W-Device. It is a read-only object of data type StringT with a maximum fixedLength of 16. This real SerialNumber (RSN) can be used by the Application for compatibility checks against a configured SerialNumber (CSN) provided by the application, depending on the InspectionLevel (IL).

NOTE: In case the vendor does not maintain a separate number space for the SerialNumber, the UniqueID shall be converted to StringT representation and used as SerialNumber.

14.3.9 UniqueID

This mandatory parameter consists of the 2 octet manufacturer distinguishing VendorID (MSO) followed by the 3 octet W-DeviceID and a 4 octet device distinguishing identifier (LSO). The Device Distinguishing ID must be a unique value for every sample of all devices produced by that vendor. It is in the responsibility of the vendor to maintain that number space or its computation algorithm.

VendorID		DeviceID			Device Distinguishing ID			
8 MSO	7	6	5	4	3	2	1	0 LSO

Figure 156 UniqueID octet mapping

The UniqueID is either stored in non-volatile memory of the W-Device during production of the device sample or generated in the W-Device during startup.

NOTE: The vendor should keep a clear relationship between the SerialNumber and the UniqueID of a W-Device. It is highly recommended that the Device Distinguishing ID is derived from the SerialNumber or vice versa.

14.3.10 Link Quality

This index stores statistical data about the reliability of the radio transmission for this W-Device. The method used for the calculation of the LinkQuality is described in clause 5.4.6).

Table 177 LinkQuality parameter

LinkQuality	Values
0x00	0 %
0x01	1 %
0x02	2 %
…	…
0x64	100 %
0x65 – 0xFF	Reserved

14.3.11 W-Bridge Information

This index range WBridgeInfo stores the parameters used in a W-Bridge configuration. BDeviceID, BVendorID and BFunctionID are similar to DeviceID, VendorID and FunctionID and refer to the W-Bridge, not the connected IO-Link device.

Table 178 W-Bridge information index assignments

Index	Subindex	Access	Parameter name	Coding	Data type
0x5004	0x00	Gives access to the whole index			
	0x01	R	BDeviceID	Octet 1: DeviceID 1 (MSB) Octet 2: DeviceID 2 Octet 3: DeviceID 3(LSB)	OctetStringT3
	0x02	R	BVendorID	Octet 1: VendorID 1 (MSB) Octet 2: VendorID 2(LSB)	OctetStringT2
	0x03	R	BFunctionID	Octet 1: FunctionID 1 (MSB) Octet 2: FunctionID 2(LSB)	OctetStringT2
	0x04	R	BDevice DistinguishingID	Octet 1: DeviceD_ID1 (MSB) Octet 2: DeviceD_ID 2 Octet 3: DeviceD_ID 3 Octet 4: DeviceD_ID4(LSB)	OctetStringT4
	0x05	R	ConnectionStatus	0x00: No device connected 0x10: Device connected 0x11: Device connected but not communication could be established	UIntegerT8

14.3.12 AdaptiveHopTable

This index range AdaptiveHopTable stores the values for the updated hopping sequence, see clause 18.4.

Table 179 update hopping table index assignments

Index	Subindex	Access	Parameter name	Data type
0x5006	0x01	W	WakeUpTime	3 Octet
	0x02	W	UpdateType	Octet
	0x03	W	Index	Octets
	0x04	W	Frequency value	OctetString

Annex D

(normative)

15 EventCodes

IO-Link Interface and System Specification defines the concept of Events in clause 7.3.8.1, general structure and encoding of Events in Clause A.6 and Table D.1 lists the specified EventCode identifiers and their definitions.

An EventCode identifies an actual incident. The EventCodes are created by the technology specific Device application (instance = APP).

The Event Codes for IO-Link wireless are placed in the range 0xFFB0 to 0xFFBF as indicated in Table C1.

15.1 EventCodes for Devices

Table 180 lists the specified EventCode identifiers and their definitions. The EventCodes are created by the technology specific Device application (instance = APP).

Table 180 EventCodes

EventCodes	Definition and recommended maintenance action	Device Status Value (NOTE 1)	TYPE
0x0000	No malfunction	0	Notification
0x1000	General malfunction – unknown error	4	Error
0x1001 to 0x17FF	Reserved		
0x1800 to 0x18FF	Vendor specific		
0x1900 to 0x3FFF	Reserved		
0x4000	Temperature fault – Overload	4	Error
0x4001 to 0x420F	Reserved		
0x4210	Device temperature over-run – Clear source of heat	2	Warning
0x4211 to 0x421F	Reserved		
0x4220	Device temperature under-run – Insulate Device	2	Warning
0x4221 to 0x4FFF	Reserved		
0x5000	Device hardware fault – Device exchange	4	Error
0x5001 to 0x500F	Reserved		
0x5010	Component malfunction – Repair or exchange	4	Error
0x5011	Nonvolatile memory loss – Check batteries	4	Error
0x5012	Batteries low – Exchange batteries	2	Warning
0x5013	HMI button pressed	0	Notification
0x5014 to 0x50FF	Reserved		
0x5100	General power supply fault – Check availability	4	Error
0x5101	Fuse blown/open – Exchange fuse	4	Error
0x5102 to 0x510F	Reserved		
0x5110	Primary supply voltage over-run – Check tolerance	2	Warning
0x5111	Primary supply voltage under-run – Check tolerance	2	Warning
0x5112	Secondary supply voltage fault (W-Port Class B) – Check tolerance	2	Warning
0x5113 to 0x5FFF	Reserved		
0x6000	Device software fault – Check firmware revision	4	Error

EventCodes	Definition and recommended maintenance action	Device Status Value (NOTE 1)	TYPE
0x6001 to 0x631F	Reserved		
0x6320	Parameter error – Check data sheet and values	4	Error
0x6321	Parameter missing – Check data sheet	4	Error
0x6322 to 0x634F	Reserved		
0x6350	Parameter changed – Check configuration	4	Error
0x6351 to 0x76FF	Reserved		
0x7700	Wire break of a subordinate device – Check installation	4	Error
0x7701 to 0x770F	Wire break of subordinate device 1 …device 15 – Check installation	4	Error
0x7710	Short circuit – Check installation	4	Error
0x7711	Ground fault – Check installation	4	Error
0x7712 to 0x8BFF	Reserved		
0x8C00	Technology specific application fault – Reset Device	4	Error
0x8C01	Simulation active – Check operational mode	3	Warning
0x8C02 to 0x8C0F	Reserved		
0x8C10	Process variable range over-run – Process Data uncertain	2	Warning
0x8C11 to 0x8C1F	Reserved		
0x8C20	Measurement range over-run – Check application	4	Error
0x8C21 to 0x8C2F	Reserved		
0x8C30	Process variable range under-run – Process Data uncertain	2	Warning
0x8C31 to 0x8C3F	Reserved		
0x8C40	Maintenance required – Cleaning	1	Notification
0x8C41	Maintenance required – Refill	1	Notification
0x8C42	Maintenance required – Exchange wear and tear parts	1	Notification
0x8C43 to 0x8C9F	Reserved		
0x8CA0 to 0x8DFF	Vendor specific		
0x8E00 to 0xAFFF	Reserved		
0xB000 to 0xBFFF	Reserved for profiles		
0xC000 to 0xFEFF	Reserved		
0xFF00 to 0xFFAF	SDCI specific EventCodes (see Table D.2 in REF 1)		
0xFFB0 to 0xFFBF	IOLW specific EventCodes (see Table 94)		
NOTE 1 See B.2.18 in REF 1			

These W-Port-related events in Table 126 are processed via AL_Event. Table 180 lists basic IOLW Events related to system management, W-Device or W-Master application, and specifies how they are encoded. Other types of Events may be reported but are not specified in this standard. Processing of these Events by the W-Master is vendor specific.

Table 181 EventCodes used for IOLW

Incident [a]	Origin	Instance	Name	EventCode	Action	Remark
System management						
W-Device communication lost	LOCAL	APP	DEV_COM_LOST	0xFF22	PD stop	See Clause 11
Data Storage identification mismatch	LOCAL	APP	DS_IDENT_MISMATCH	0xFF23	-	See Clause 11
Data Storage buffer overflow	LOCAL	APP	DS_BUFFER_OVERFLOW	0xFF24	-	See Clause 11
Data Storage parameter access denied	LOCAL	APP	DS_ACCESS_DENIED	0xFF25	-	See Clause 11
IOLW_Retry_Error	LOCAL	APP	IOLW_RETRY_ERROR	0xFFB1	-	See Clause 11
IOLW_IMATimeout	LOCAL	APP	IOLW_IMATIMEOUT	0xFFB2	-	See Clause 11
Unspecified						
Incorrect Event signaling	LOCAL	DL	EVENT	0xFF31	Event.ind	See Clause 11
Device specific application						
IOLW_Retry_Error	REMOTE	APP	IOLW_RETRY_ERROR	0xFFB9	-	See Clause 11
Data Storage upload request	REMOTE	APP	DS_UPLOAD_REQ	0xFF91	Event.ind	
Reserved	REMOTE	APP		0xFF98	Event.ind	Shall not be used

[a] All Events are of StatusCode type 2 (with details), EventQualifier type "Notification", EventQualifierMode "Single-shot"

Annex E

(normative)

16 Data Types

This annex refers to IO-Link Interface and System Specification REF 1, Annex E, which specifies basic and composite data types. Examples demonstrate the structures and the transmission aspects of data types for singular use or in a packed manner.

Annex F

(normative)

17 Device design rules for low Energy W-Devices

17.1 Low Energy W-Devices

For the design of Low-energy W-Devices, the following support is given by this specification to minimize power consumption:

17.1.1 Low voltage design

To minimize dissipation loss within the W-Device circuitry, the power supply voltage should be chosen as low as possible.

17.1.2 Event triggered activation

To minimize transmitter activity, an uplink is only transmitted when the W-Device has new data to report or the IMA-timer has expired.

17.1.3 Long IMATime

To minimize both receiver and transmitter activity, the maximum configurable IMATime should be chosen as long as possible.

17.1.4 Pre-downlink

To minimize receiver activity for synchronization, a W-Device should receive only the pre-downlink, provided that no new data is received to the W-Device.

17.1.5 W-Master not reachable

A W-Device that has lost connectivity to its W-Master and could not resynchronize should stop listening by issuing SM_SetDeviceMode(IDLE) until e.g. the next IMA time cycle has expired or when an event at the W-Device occurs, e.g. the button on the W-Device has been pressed by the operator.

17.1.6 Quick Synchronization

To minimize receiver activity for synchronization after a longer IMA sleep period, a W-Device listens on its assigned frequency according to the hopping sequence within an uncertainty window. To minimize the worst-case resynchronization time, the usage rate of a certain frequency within the hopping table could be increased. This frequency channel shall then be used by the W-Device for the resynchronization procedure. The W-Device shall analyze its hopping table and use the most used frequency channel for resynchronization purposes.

17.1.7 Establish communication

After waking up a sleeping low energy W-Device (see 10.10.3.2) a paired W-Device starts to synchronize to the W-Master, respectively an unpaired W-Device is waiting for a Master Request, see 5.6.1
To minimize the power consumption the W-Device should go back to sleep after the recommended power on time, see Table 182. Therefore the application has to start or stop the radio via SM_SetDeviceMode.

Table 182 Recommended power on time

	On time	
Synchronization	270 ms	2*80 W-Sub-cycle
Scan	300 ms	50 ms + 30 x 5 W-Sub-cycle
Pairing by Button Pairing by UniqueID	200 ms	See Table 186

17.1.8 HMI sleep

A low energy Device should deactivate the visual indication after a W-Device specific timeout (e.g. 5 min) for power saving reasons.
By pressing the pairing button or by receiving a “WinkOn” SystemCommand, the visual indication shall be activated until the W-Device specific timeout exceeds. After receiving the “WinkOff” SystemCommand the visual indication shall signal the W-Device state defined in Table 124 Visual states of W-Device for the remaining W-Device specific timeout.

17.2 Battery lifetime calculation

The following formula provides support for a rough estimation of battery lifetime for a W-Device.

$$T_{batt} = \frac{Q_{batt} \cdot \frac{1}{24h} \cdot \frac{1}{365d}}{\frac{(12.5\,ms + 0.416\,ms) \cdot I_{receive} + (0.2\,ms) \cdot I_{transmit} + (T_{sleep} + 0.208\,ms + 0.632\,ms) \cdot I_{sleep}}{12.5\,ms + 0.416\,ms + 0.2\,ms + 0.208\,ms + 0.632\,ms + T_{sleep}}}$$

= 12.5 ms Synchronization phase and 0.416ms reception phase
= 0.2 ms Transmitting phase
= 0.208 ms Tx to Rx change phase, 0.632ms inactive phase and application specific sleeping phase

Thus:

$$T_{batt} = \frac{Q_{batt} \cdot \frac{1}{24\,h} \cdot \frac{1}{365\,d}}{\frac{12.9\,ms \cdot I_{receive} + 0.2\,ms \cdot I_{transmit} + (T_{sleep} + 0.84\,ms) \cdot I_{sleep}}{14.0\,ms + T_{sleep}}}$$

Factor	Unit	Description	Typical value
T_{batt}	[years]	Calculated battery lifetime in years	8 years
Q_{batt}	[Ah]	Capacity of the battery	1.2 Ah
T_{sleep}	[seconds]	Average sleeping time between two active phases	10 s
I_{sleep}	[µA]	Current drain when transceiver is inactive	2 µA
$I_{transmit}$	[mA]	Avg. current drain when transceiver is in transmitting mode	9 mA
$I_{receive}$	[mA]	Avg. current drain when transceiver is in receiving mode	6 mA

The above formula is based on the assumptions:

The W-Device is only active, while sending or receiving data. This just includes the yellow and green areas in the figure below. While a physical transceiver chip isn’t able to turn on and off immediately before/after its real active time, a tolerance of about 3…5% should be considered regarding the battery lifetime for this point.

An amount of 46 channels is used for the hopping table.

No retries have been used during the data transfer. In an ambient with no excessive RF-disturbances, this should be near to the real-world scenario.

The synchronization process will take an average of 12.5 ms, before the W-Device is able to communicate with the W-Master again after a long (e.g. some minutes) sleep phase. This estimation is based on an average of 7.5 Sub-Cycles required for the synchronization.

The formula further is based on the IOLW-specs regarding timing values. To clarify the used times please check the following extract of the IOLW-timing diagram below.

Annex G

(normative)

18 Frequency Hopping Calculation

18.1 Blacklisting

Blacklisting is a mechanism to avoid on air collision with other wireless systems, such as WLAN. Conventional Bluetooth cannot be blacklisted, because it is an uncoordinated frequency hopper. The blacklist itself uses eighty 1 MHz wide frequency channels.
The blacklisting examples described here are focused on WLAN according to IEEE 802.11 for the 2,4GHz ISM band, which supports 13 different, overlapping 22 MHz frequency blocks. Each blacklisted WLAN channel shall be mapped to the blacklist format described in 5.4.5. The frequency blocks used by IO-Link-Wireless for blacklisting are shown in Table 183. The configuration channels 2401 MHz and 2480 MHz cannot be blacklisted.

Table 183 Frequency table for WLAN channels

WLAN Channels	Centre Frequency (MHz)	Occupied frequencies (MHz)
1	2412	2401-2423
2	2417	2406-2428
3	2422	2411-2433
4	2427	2416-2438
5	2432	2421-2443
6	2437	2426-2448
7	2442	2431-2453
8	2447	2436-2458
9	2452	2441-2463
10	2457	2446-2468
11	2462	2451-2473
12	2467	2456-2478
13	2472	2461-2483

18.2 Creation of frequency hopping table HT01 with blacklisting

The creation of the frequency hopping table HT01 is divided into seven steps:

(i) Create an array with the all available frequency channels within the 2.4 GHz ISM frequency Band.
(ii) Find all blacklisted channels according to the provided blacklist and remove them from the frequency array.
(iii) Perform a circular shift of the array depending on MasterID in order to randomize the starting frequency.
CircularShift(array, MasterID);
(iv) Discover the permutation index ***P***. The permutation index is a greatest prime number that is smaller or equal to the length of an array created in the previous step:

```
for (index = 0: length(primes_array))
 if (prime_array(index) <= length(array)) then
    P = prime_array(index);
  end if;
end for;
```

(v) Calculate a Sequence number ***N*** in according to the MasterID

```
if ((MasterID % 2 )== 0) then
   N = int16((P-1)/2) + (MasterID/2);
else
   N = int16((P-1)/2) - ((MasterID – 1)/2);
```

```
    end if;
```

(vi) Create a Matrix with the possible frequencies, the frequency spacing should be taken in to account.

```
    Possible_Freq_Matrix = zeros(5,length(array));
    for (index = 0:length(array))
       Possible_Freq_Matrix (0,index) = array(index);
      for (index_track = 1 : Track_N)
         Possible_Freq_Matrix (index_track,index) = array(((index +
         (Spacing*index_track)) % length(array)));
      end for;
    end  for;
```

(vii) Generate frequency hopping table from frequency matrix. Selecting of the appropriate frequency is perform using of Sequence number ***N*** and Prime index ***P*** as following:

```
For (index = 0:P)
    Sequence_index = (N * index)% P;
    for (index_track = 0 : Track_N)
        Frequency_Table(index_track, index) =
        Possible_Freq_Matrix(index_track, Sequence_index)
  end for;
 end for;
```

18.2.1 Example with 2 WLAN Channels

MasterID = 10;
Track number = 5;
Spacing = 3;
Primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79]
BlackList = [0x0000 00000 FFFF F3EF FFF7] (See Table 184, Figure 157)

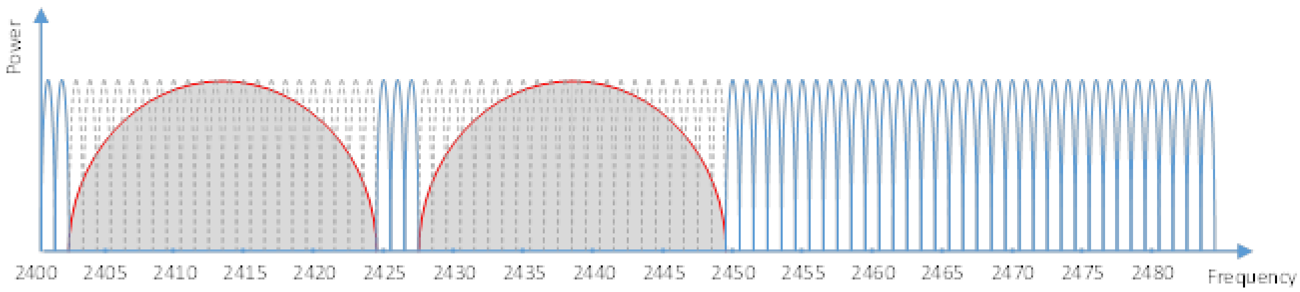


Figure 157 Blacklisting of 2 WLAN channels in 2.4GHz ISM Band

The influence of the given blacklist on the whole 2.4 GHz ISM Spectrum is demonstrated in Figure 157. If the blacklist is used the occupied frequencies given in are not used.

Table 184 WLAN Channels 1 and 6 Blacklisting example

Blacklisted WLAN Channels	Center Frequency (MHz)	Occupied frequencies (MHz)
1	2412	2402-2423
6	2437	2427-2448

Calculating a frequency Table using a given data:

Steps (i) and (ii): Find all not blacklisted channels, create an array:
array = {
2424 2425 2426 2427 2428 2429 2430 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478}

Step (iii): Circular Shift; shift length = 10:
array = {
2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2424 2425 2426 2427 2428 2429 2430 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468}

Step (iv): Find Permutation index **P**:

Length(array) = 32
P = max(Primes < 32) = 31.

Step (v): Calculate a Sequence number N in consideration of the MasterID

mod((MasterID),2) = mod(10,2) = 0 **=>**
N = ((***P***-1) / 2) + (MasterID / 2) = (31-1 / 2) + (10/2) = 20

Step (vi): Create a Matrix with the possible frequencies respecting the frequency spacing:

	Frequency																															
Track 1	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2424	2425	2426	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468
Track 2	2473	2474	2475	2476	2477	2478	2424	2425	2426	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472
Track 3	2476	2477	2478	2424	2425	2426	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475
Track 4	2424	2425	2426	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478
Track 5	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2424	2425	2426

Figure 158 Possible Frequencies for 5 Tracks.

Step (vii): Generate the Frequency table using Frequency matrix, the Permutation Index and a Sequence number.

	Frequency																															
Track 1	2457	2478	2466	2455	2476	2464	2430	2474	2462	2428	2472	2460	2426	2470	2458	2424	2467	2456	2477	2465	2454	2475	2463	2429	2473	2461	2427	2471	2459	2425	2469
Track 2	2461	2427	2470	2459	2425	2468	2457	2478	2466	2455	2476	2464	2430	2474	2462	2428	2471	2460	2426	2469	2458	2424	2467	2456	2477	2465	2454	2475	2463	2429	2473
Track 3	2464	2430	2473	2462	2428	2471	2460	2426	2469	2458	2424	2467	2456	2477	2465	2454	2474	2463	2429	2472	2461	2427	2470	2459	2425	2468	2457	2478	2466	2455	2476
Track 4	2467	2456	2476	2465	2454	2474	2463	2429	2472	2461	2427	2470	2459	2425	2468	2457	2477	2466	2455	2475	2464	2430	2473	2462	2428	2471	2460	2426	2469	2458	2424
Track 5	2470	2459	2424	2468	2457	2477	2466	2455	2475	2464	2430	2473	2462	2428	2471	2460	2425	2469	2458	2478	2467	2456	2476	2465	2454	2474	2463	2429	2472	2461	2427

Figure 159 Frequency Table for 5 Tracks

18.2.2 Example with one WLAN channel

MasterID = 9;
Track number = 1;
Spacing = 3;
Primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79]
BlackList = [0x0000 0000 FFFF F300 0000] (See table 2, Figure 1)

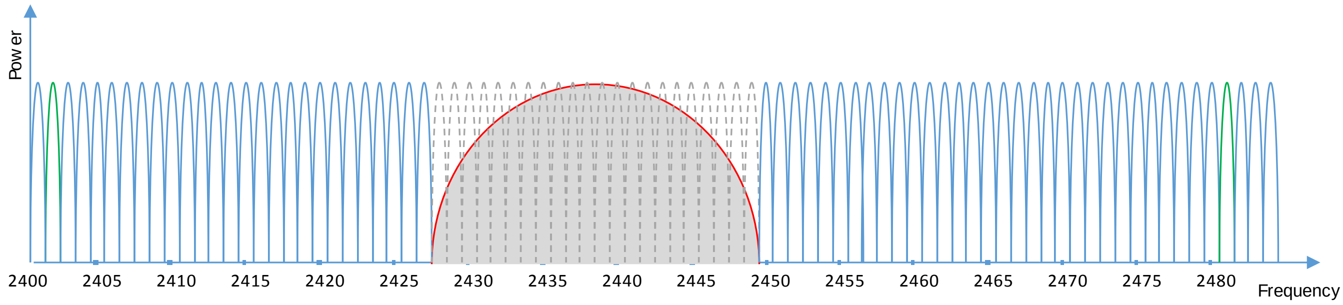


Figure 160 Blacklisting of one WLAN channel in 2.4GHz ISM Band

The influence of the given blacklist on the whole 2.4GHz ISM Spectrum is demonstrated in Figure 160 Blacklisting of one WLAN channel in 2.4GHz ISM Band. If the blacklist is used the occupied frequencies given in Table 185 shall be omitted.

Table 185 WLAN Channel 1 Blacklisting example

Blacklisted WLAN Channels	Center Frequency (MHz)	Occupied frequencies (MHz)
6	2437	2427-2448

Calculating a frequency Table using a given data

Steps (i)+(ii): Find all not blacklisted channels, create an array:
array = {
2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478}

Step (iii): Circular Shift; shift size = 9:
array = {
2470 2471 2472 2473 2474 2475 2476 2477 2478 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469}

Step (iv): Find Permutation index P:
Length(array) = 52
P = max(Primes < 52) = 47.

Step (v): Calculate a Sequence number N in consideration of the MasterID
mod((MasterID),2) = mod(9,2) = 1 =>
N = ((P-1) / 2) + ((MasterID -1) / 2) = (53-1 / 2) + ((9-1)/2) = 19

Step (vi): Create a Matrix with the possible frequencies respecting the frequency spacing

	Frequency																															
Track 1	2470	2471	2472	2473	2474	2475	2476	2477	2478	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425
Track 1	2426	2427	2428	2429	2430	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469										

Figure 161 Possible Frequencies for 1 Track

Step (vii): Generate the Frequency array using Frequency matrix, the Permutation Index and a Sequence number

	Freqeuncy																															
Track 1	2414	2456	2405	2424	2471	2415	2457	2406	2425	2472	2416	2458	2407	2426	2473	2417	2459	2408	2427	2474	2418	2460	2409	2428	2475	2419	2461	2410	2429	2476	2420	2462
Track 1	2411	2430	2477	2421	2463	2412	2454	2478	2422	2464	2413	2455	2404	2423	2470																	

Figure 162 Frequency Table for 1 Track

18.3 Modified Sequence for ServiceMode

As stated in 5.4.4 “Configuration Channel”, the ServiceMode must also utilize the configuration frequencies during the IO-Link Wireless installation phase, for W-Device exchange in exceptionally cases or permanently for roaming. The ServiceMode is called in case of an adding of the W-Device (PL_Pairing.req) or invoking a scanning on the W-Devices in neighborhood (PL_Scan.req).

In W-Master and W-Device the ServiceMode frequency hopping table is implemented by the temporal exchange of a frequency in the normal frequency hopping table every 5th W-Sub-cycle with a configuration frequency. Therefore, one of the configuration frequencies 1 (2401) and 80 (2480) shall be used every 5th W-Sub-cycle in an alternating manner on the W-Master side. In order to ensure the reception of the configuration telegrams, the configuration frequencies shall switch every 10th W-Sub-cycle on the W-Device side. This frequency alternation is implemented using a so called column counter (Col_N) of the hopping frequencies in the frequency hopping table, which will switch to one of the alternating configuration frequencies (see Figure 163).

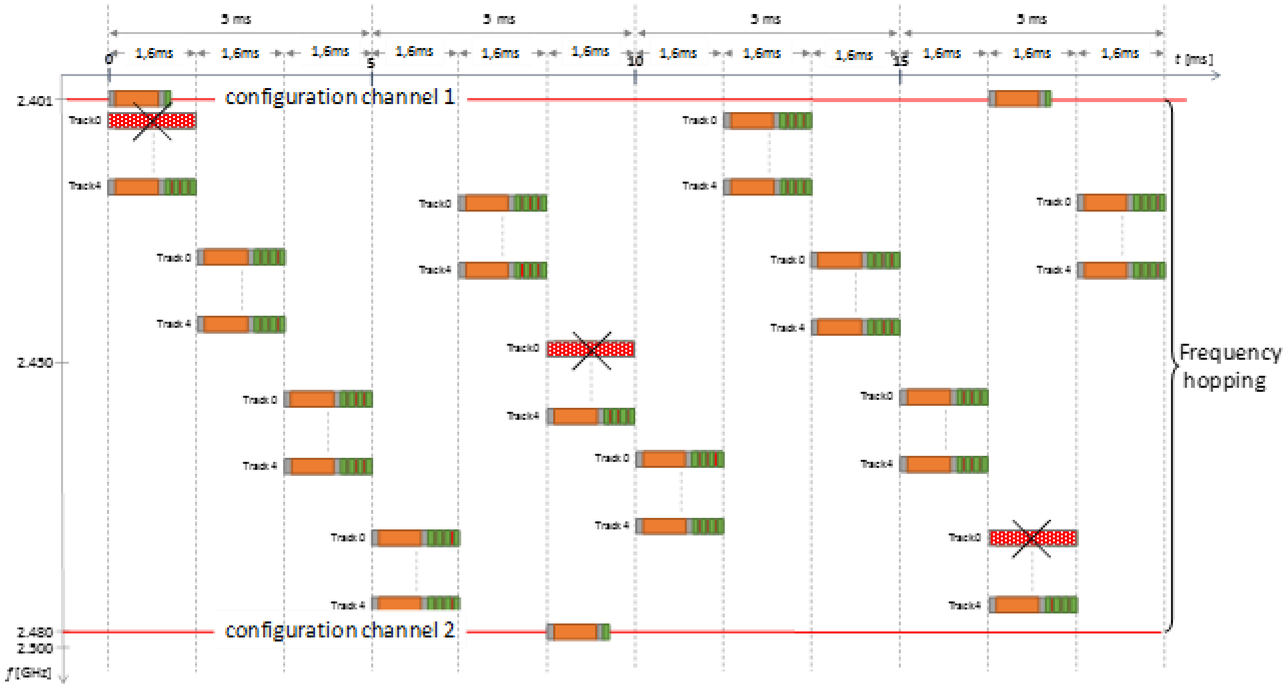


Figure 163 Usage of the configuration frequencies

Figure 163 shows an example of Track_0 in ServiceMode. In this track, every 5th W-Sub-cycles is substituted by a configuration W-Sub-cycle, the four others remain as regular cyclic data W-Sub-cycles. Besides the content, the carrier frequency in every 5th column in the frequency hopping table is alternately substituted with one of the configuration frequencies.
Configuration W-Sub-cycles are replacing time slots of the regular W-Sub-cycles, thus consuming transmission capacity on the expense of randomly selected slots, which might statistically reduce the guaranteed communication availability of these affected slots.

The modification of the frequency hopping table has only a temporary effect for Scan Mode and Pairing Mode. The maximum activation time of the ServiceMode on the W-Master side is given by the timeouts configured for Scan and Pairing procedures. For scan and pairing procedure on the W-Device, the default activation time is given by the values in Table 186. The W-Device shall leave the ServiceMode directly after the exchange of the ConnectionParameter.

Table 186 Timeouts for ServiceMode

Timeouts:	W-Master	W-Device
SCAN_TIMEOUT	5 s	-
PAIRING_UNIQUE_TIMEOUT	min. 5 s	200 ms
PAIRING_BUTTON_TIMEOUT	min. 5 s	200 ms

18.4 Adaptive Hopping Table (AHT)

Adaptive hopping table enables to update the hopping table per track in a W-Master and its corresponding W-Devices. The frequency channels are monitored in the W-Master SM AHT handler and if an update is decided then the update data is sent to W-Devices via ISDU. For a track with low energy W-Devices the AHT will invoke a wake up sequence to synchronize a deterministic time when all W-Devices are listening, see Figure 164. W-Master can then trigger a countdown via MasterCommand to all updated W-Devices to synchronize a switch to the new hopping table values, see Figure 165. The support of this function is mandatory for a W-Device and optional for a W-Master.
This includes:

- PL_SetHopTable
- PL_GetHopTable
- PL_SetWakeUpTime
- PL_WakeUpTime
- PL_AHTStatus
- PL_CmdTrig
- DL Cmd handler - states 3 to 7 and all their related transitions
- SM AHT Handler - all states and transitions

Table 187 shows the update types, index and data types used by PL_SetHopTable service. The Data values contain the new frequencies of the hopping table which are restricted according to the Blacklist.

Table 187 AHT UpdateType

UpdateType	Description	Index	DataType
FULL_TABLE	Replace all frequencies of the hopping table, see Figure 162 Frequency Table for 1 Track	0	OctetString
DELETE_CELL	Deletes a frequency from the hopping table according to index. Index x deletes HOP-x.	1-78	-
ADD_CELL	Adds a frequency to the hopping table. Index x adds HOP-x and incremants all the indexes larger then x by 1.	1-78	1 Octet
REPLACE_CELL	Replace a frequency in the hopping table, index x replaces HOP-x frequency	1-78	1 Octet

Figure 164 shows an example for using MCmd(WakeUp) in combination with MCmd(Jump) to set a new hopping table HT02. This figure describes the timeline view for WakeUp of several W-Devices and synchronizing the switch of the hopping table to all W-Devices in the track.

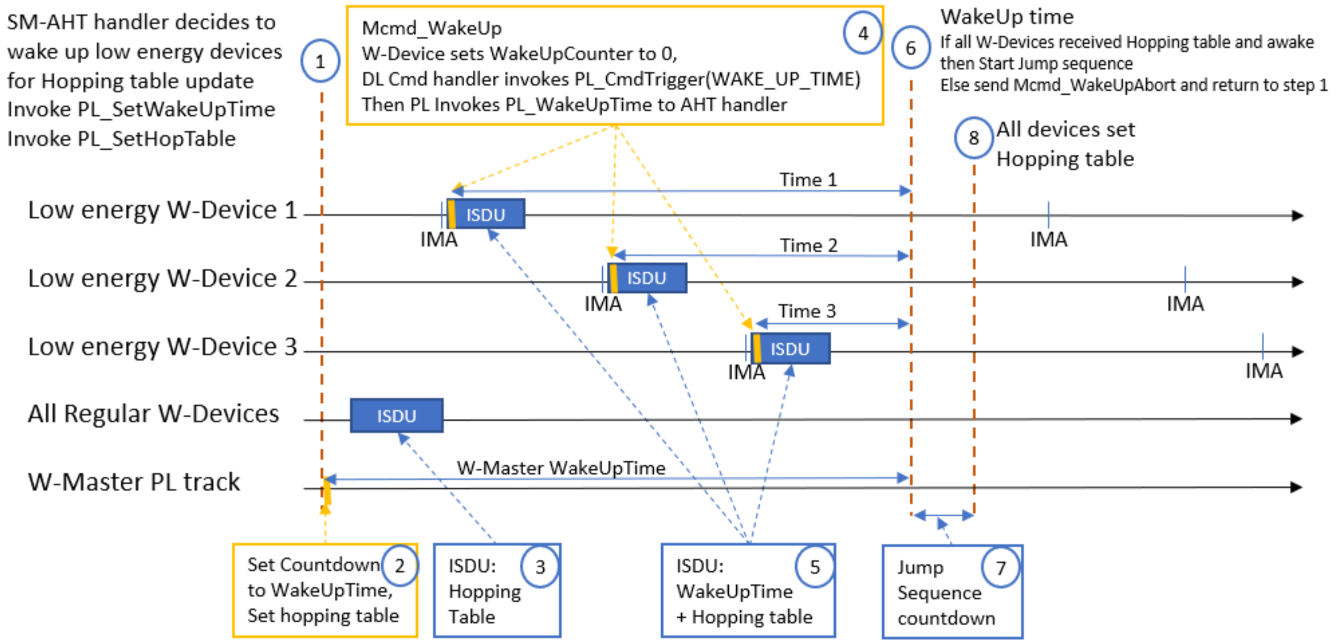


Figure 164 Changing Hopping Table synchronization timeline

Figure 165 shows the sequence of WakeUp of low energy W-Device

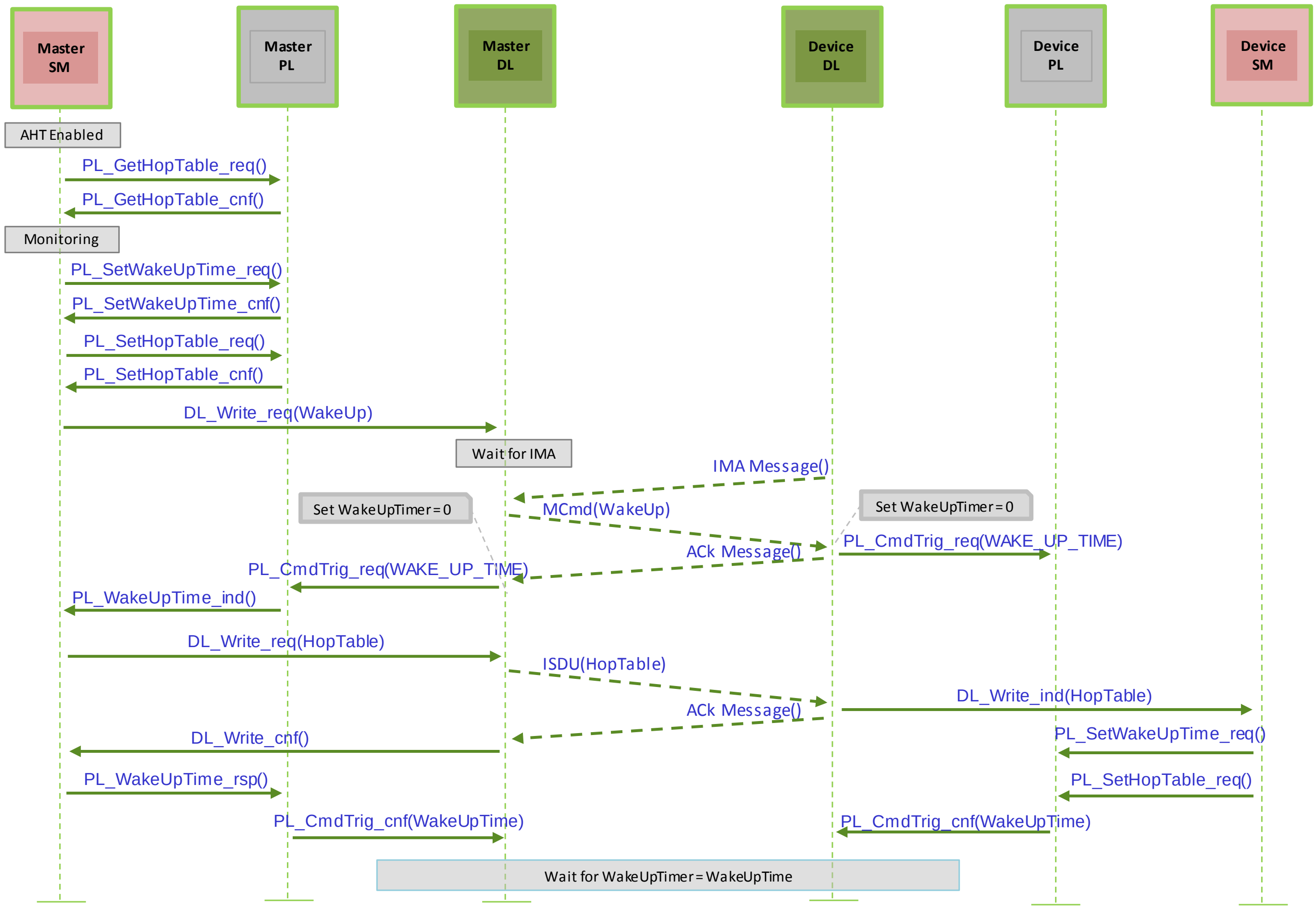
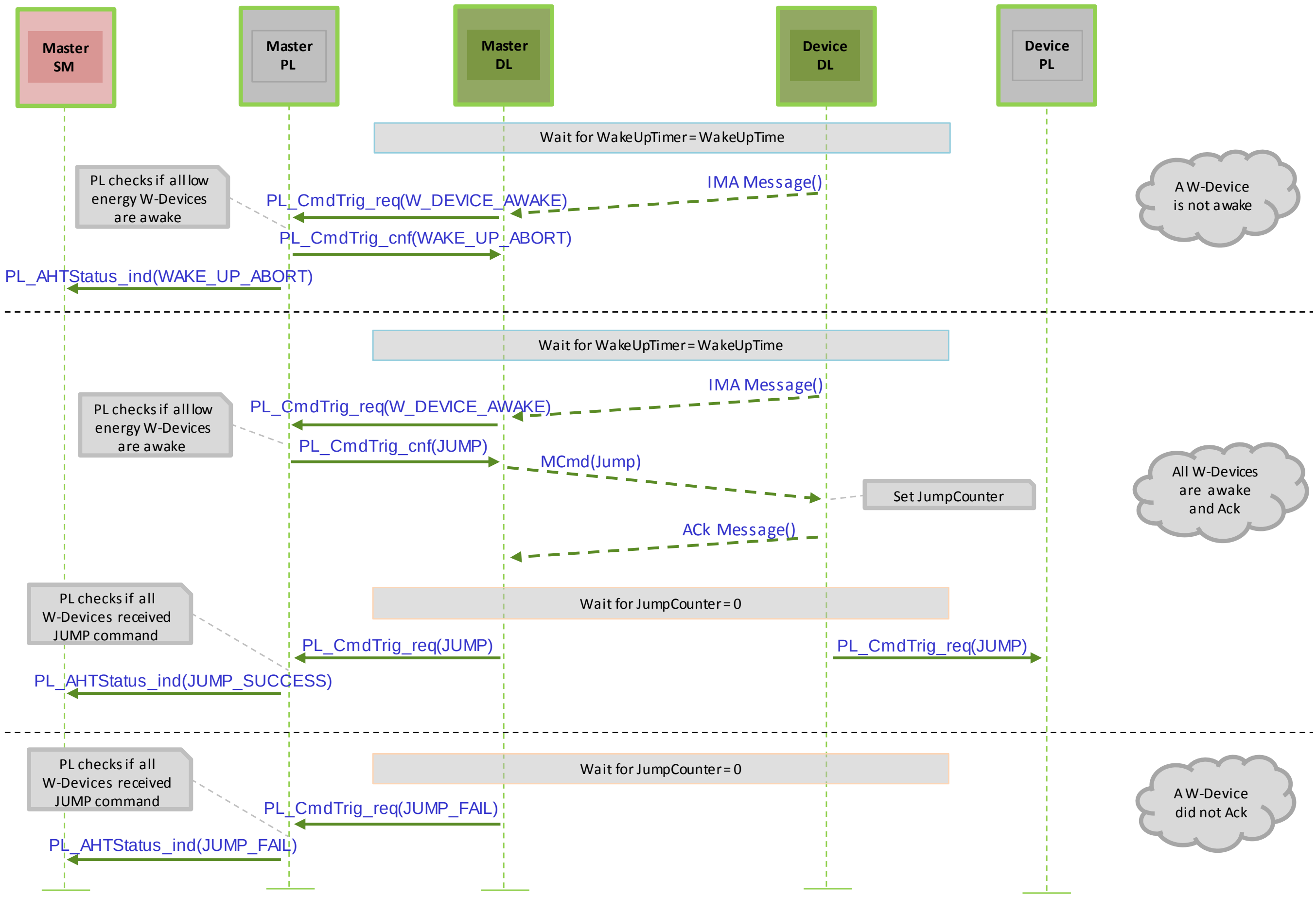


Figure 165 AHT WakeUp sequence

6530 Figure 166 shows the Jump sequences
6531



6532 **Figure 166 AHT Jump sequence**
6533

Annex H

(normative)

19 How to get a certified product

In order to get a certified IO-Link wireless product, different testing and certification aspects must be considered:

19.1 Radio Certification

To satisfy the legal jurisdiction under which the wireless equipment shall be used, the locally valid regulatory compliance rules for wireless equipment must be fulfilled. Currently relevant regulations are outlined in more detail in Annex I “Regulatory Compliance”.

19.2 IO-Link Certification

Compliance to the IO-Link wireless protocol defined in this standard must be documented with a manufacturer self-declaration and associated test reports for the specific product, containing the aspects of both protocol conformity and performance conformity.
The required testing procedures and recommended test lab services towards the testing references will also be described in the separate IO-Link wireless test specification see REF 11

Annex I

(normative)

20 Regulatory Compliance

20.1 General

This Annex I provides requirements for compliance of IO-Link wireless devices operating in the 2.4 GHz ISM band with several regulatory standards. For operation in the United States, FCC 15.247 must be met (see clause 20.2). Additional requirements apply in Europe, which can be met by complying with FCC 15.247 in combination with harmonized standards EN 300 328 (see clause 20.3) and EN 300 440 (see clause 20.4).

It is generally recommended to handle the regulatory standards in a similar approach as the Bluetooth low Power Special Interest Group has outlined in (REF 5: "Bluetooth Low Energy Regulatory Aspects")

Additionally, ETSI Guide EG 203 367 (V1.1.0) contains guidance information in assessing conformity against the essential requirements of the Radio Equipment Directive 2014/53/EU (RED) for the combination of radio and non-radio products as well as the integration of several radios into a single equipment.

NOTE: In this Annex I, the terms “devices” and "equipment" are used synonymously and refer to electronics with radios operating according to the appropriate standard.

20.2 Compliance with FCC 15.247

To comply with FCC 15.247, the manufacturer should declare IO-Link wireless equipment according to FCC §15.247-a2 as “**systems using digital modulation**”, where "**the minimum 6 dB bandwidth shall be at least 500 kHz**". This requires static testing at the relevant frequency channels (typically band edges and center channel) while frequency hopping algorithms are not subject of compliance testing procedures, see REF 7.

20.3 Compliance with ETSI EN 300 328 V2.1.1 (2016-11)

EN 300 328 is listed as a harmonized standard under the Radio Equipment Directive 2014/53/EU.

To comply with EN 300 328, the manufacturer should declare its IO-link wireless equipment as utilizing "**other types of Wide Band modulation**" with **"RF Output power is less than 10 dBm e.i.r.p."** and being a **"non-adaptive equipment".**
The limit of 10 dBm shall apply for any combination of power level and intended antenna assembly. If more than one track (antenna) is used in a device, all tracks are sharing the 10 dBm. For example, 3 dBm per track are permitted in case of five tracks.
The required test suites must be carried out and compliance declared for the relevant technical requirements see REF 8.

20.4 Compliance with ETSI EN 300 440 V.2.1.1 (2017-03)

EN 300 440 is listed as harmonized standard under the Radio Equipment Directive 2014/53/EU.

To comply with EN 300 440, the manufacturer should declare its IO-link wireless equipment as "**Non-specific short-range device**" restricted to a "**Maximum radiated peak power (e.i.r.p.)**" of "**10 mW e.i.r.p.**"
The e.i.r.p. is defined as the "**maximum radiated power of the transmitter and its antenna**", thus antenna gains better than 0 dBi require an adequate power adjustment.
The required test suites must be carried out and compliance declared for the relevant technical requirements see REF 9 and REF 10.

Annex J

21 Wireless IODD File handling

21.1 File naming convention

Wireless IODD files must follow the naming convention described below:

IO-Link Device:	<VendorName>-<DeviceName>-YYYYMMDD-IODD1.1.xml
IO-Link W-Device:	<VendorName>-<DeviceName>-**WD**-YYYYMMDD-IODD1.1.xml
IO-Link W-Bridge:	<VendorName>-<DeviceName>-**WB**-YYYYMMDD-IODD1.1.xml
IO-Link W-Bridge with Device:	<VendorName>-<DeviceName>-**WBD**-YYYYMMDD-IODD1.1.xml

21.2 Rules merging IODD and W-IODD file for W-Bridges

This section describes how the W-IODD of a W-Bridge and the IODD of wired IO-link Device shall be merged.

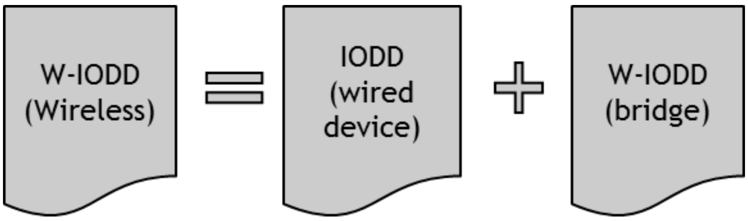


Figure 167 Merging IODD and W-IODD file for W-Bridges

The IODD of the wired IO-Link Device is taken as the basis document and is modified to correspond to the description of a W-Device. The information about the W-Device is being copied from the W-IODD of the bridge.

Step 1) The <CommNetworkProfile … > </CommNetworkProfile>, see clause 10.9.1.1, tag and its content of the wired device IODD shall be replaced by its counterpart of the bridge-IODD.

Step 2) The <StdVariableRef id="V_SystemCommand"> must be completed with the wireless system command value 64 and 65, see Table 166.

```
<VariableCollection>
      <StdVariableRef id="V_DirectParameters_1" />
      <StdVariableRef id="V_DirectParameters_2" />
      <StdVariableRef id="V_SystemCommand">
…
              <StdSingleValueRef value="64" />
<StdSingleValueRef value="65" />
…
       </StdVariableRef>
       …
</VariableCollection>
```

Step 3) All <Variable … index="i" …> tags with index between 0x5000 and 0x50FF must be copied from the W-IODD of the bridge.

Step 4) The text from the W-IODD stored within <ExternalTextCollection>…</ExternalTextCollection> must also be transferred corresponding to the changes described at steps 1 to 4. Merging conflicts, for example due to identical variable name, shall be prompted within the engineering tool for correction by the user.

Step 5) The IODD checker must be executed in order to update the <Stamp crc="xxxxxxxx"> tag and validate the newly created IODD file.

Step 6) For traceability, the merging process and tool are documented in the subelement <BridgeLayer> of CommNetworkProfile:


```
<CommNetworkProfile xsi:type="IOLinkWirelessCommNetworkProfileT" iolinkRevision="V1.1">
   <TransportLayer>
      <PhysicalLayer minCycleTime="…" doubleSlot=”…” isABridge=”yes”>
         …
      </PhysicalLayer>
      <BridgeLayer>
         <DeviceIODD fileName="<VendorName>-<DeviceName>YYYYMMDD-IODD1.1.xml" />
         <BridgeIODD fileName="<VendorName>-<DeviceName>-WB-YYYYMMDD-IODD1.1.xml" />
         <MergerTool vendorName="…" name="…" version="nn.nn.nn" />
      </BridgeLayer>
   </TransportLayers>
…
</CommNetworkProfile>
```

Step 7) The newly created W-IODD file for the IO-Link W-Bridge with Device must be saved following the naming convention stated under clause 21.1

22 Bibliography

REF 1 IO-Link Community, IO-Link Interface and System Specification, V1.1.2, July 2013, Order No. 10.002

REF 2[IEC 61131-9, Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

REF 3 IO-Link Community, IO Device Description (IODD), V1.1, Order No. 10.012 (available at http://www.io-link.com)

REF 4 IO-Link Community, IO-Link Smart Sensor Profile 2nd edition, V1.0, Order No. 10.042 (available at http://www.io-link.com)

REF 5 Bluetooth SIG - Regulatory Committee, "Bluetooth Low Energy Regulatory Aspects", V10r00, 26 April 2011

REF 6 IO-Link Community, IO-Link Common Profile, V0.9.9, Order No. 10.072 (available at http://www.io-link.com)

REF 7 FCC §15.247 "Radio frequency devices; Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz"
FCC §15.247 "Radio frequency devices; Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz"

REF 8 ETSI EN 300 328 V2.1.1 “Wideband transmission systems
ETSI EN 300 328 V2.1.1 “Wideband transmission systems; Data transmission equipment operating in the 2.4 GHz ISM band and using wide band modulation techniques; Harmonized Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU”

REF 9 ETSI EN 300 440 V2.1.1 "Short Range Devices (SRD)
ETSI EN 300 440 V2.1.1 "Short Range Devices (SRD); Radio equipment to be used in the 1 GHz to 40 GHz frequency range; Harmonized Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU"

REF 10 ETSI EN 300 440 V2.1.1 "Short Range Devices (SRD)
ETSI EG 203 367 V.1.1.1 "Guide to the application of harmonized standards covering articles 3.1b and 3.2 of the Directive 2014/53/EU (RED) to multi-radio and combined radio and non-radio equipment"

REF 11 IO-Link wireless test specification

REF 12 IO-Link Community, Addendum 2017, V2.0, December 2017, Order No. 10.152

IO-Link Community
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
e-mail: info@io-link.com
http://www.io-link.com/