IO-Link Wireless System Extensions

Specification
+ Corrigendum

V1.1.3
June 2023

Order No: 10.112

File name: **IOLW-Spec_V1.1.3CorrV02status2023_07_25.docx**

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
c/o PROFIBUS Nutzerorganisation
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 721 986197-0
Fax: +49 721 986197-11
E-mail: info@io-link.com
E-mail: info@io-link.com
Web site: www.io-link.com

CONTENTS

0 Introduction

0.1 General

The base technology of IO-Link™[1]) is subject matter of the international standard IEC 61131-9 (see [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 Master and its associated W-Devices (sensors, actuators, I/O terminals, etc.). The Master can be part of a fieldbus communication system or any stand-alone

[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".

processing unit. The technology also enables the acyclic transfer of parameters to Devices and the propagation of diagnosis information from the Devices to the upper-level automation system (controller, host) via the Master/gateway.
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.

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.

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 W-Tracks in parallel, each W-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 radios for the 2,4 GHz ISM band, divided to frequency channels with a spacing of 1 MHz.
- Frequency Hopping changes the frequency channels for each transmission as a measure against interference, yielding a RFP of 10^{-9} which is similar to a wired connection.
- W-Coexistence with other wireless systems (e.g., WLAN) is achieved with a blocklisting 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 W-Track. In case of more than one W-Track, 10 m can be achieved. These figures are dependent on the machine environment.
- Each transmission W-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 International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent. IEC takes no position concerning the evidence, validity, and scope of this patent right:

EP3229412B2	[MS]	INDUSTRIAL WIRELESS COMMUNICATIONS SYSTEM

The holder of this patent right has assured IEC that s/he is willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from the patent database available at http://patents.iec.ch.

[MS]	**SMC Corporation,** Akihabara UDX15F, 4-14-1, Sotokanda, Chiyoda-ku, Tokyo 101-0021, JAPAN https://www.smcworld.com

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than this identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain 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.

The “PL” (Air interface) for IO-Link Wireless includes the Physical Layer as well as the wireless mechanisms (e.g., pairing, blocklisting, …) 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 Specification for the operation of wireless communication.

2 Normative references

Figure 1 shows its relationships to international fieldbus, wireless communications, EMC, and power source standards.

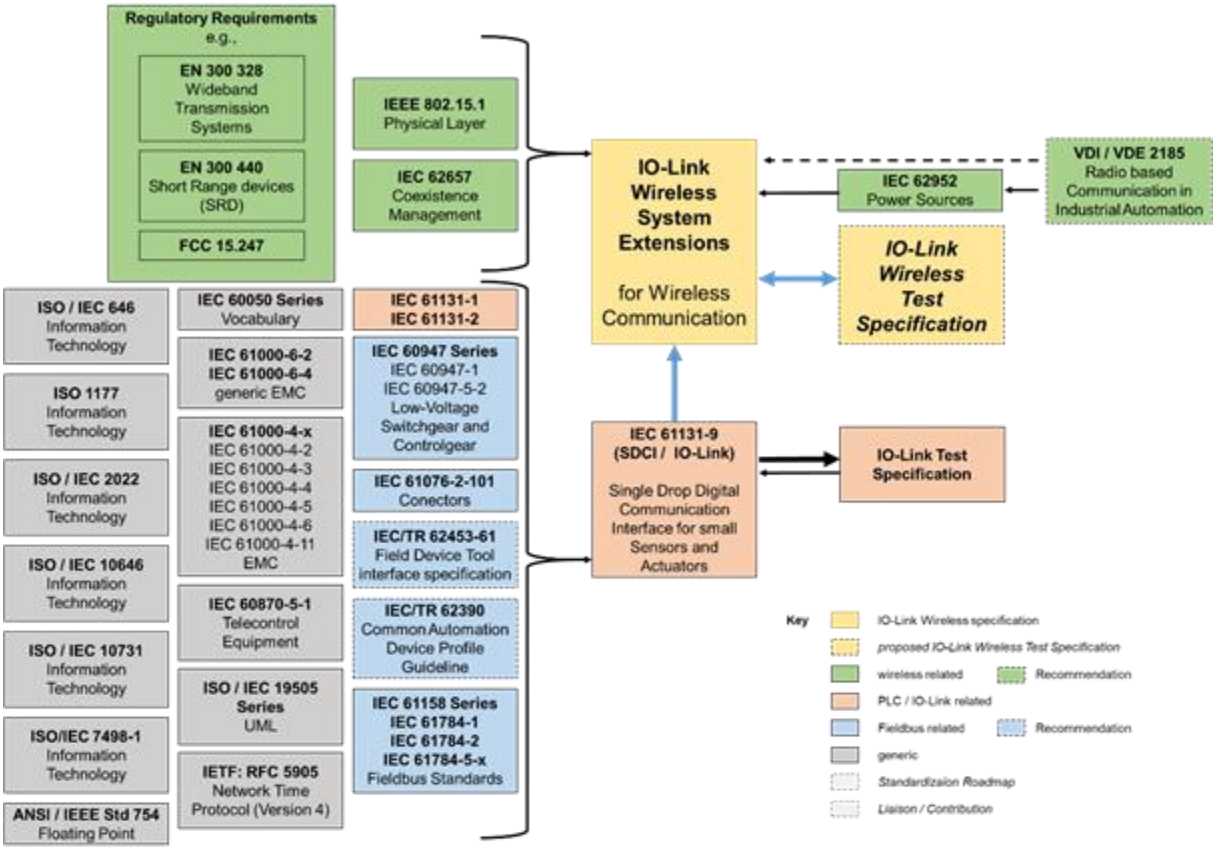


Figure 1 Relationships of this document to standards

In addition to the normative references of the IO-Link Interface and System Specification clause 2 [1], the following documents are relevant.
The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-1, Programmable controllers – Part 1: General information

IEC 61131-2, Industrial-process measurement and control - Programmable controllers - Part 2: Equipment requirements and tests

IEC 61131-9, Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

IEC 61139 (all parts) Industrial Networks – Digital Communication Interface For Small Sensors And Actuators

IEC 61158-1, Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series

IEC 62657-2, Industrial communication networks - Wireless communication networks - Part 2: Coexistence management

IEC/TR 62390, Common automation device – Profile guideline

IEC/TR 62453-61, Field device tool interface specification – Device Type Manager (DTM) Style guide for common object model

ISO/IEC 19505, Information technology – Object Management Group Unified Modeling Language (OMG UML)

ETSI EN 300 328 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"

ETSI EN 300 440 "Short Range Devices (SRD); ETSI EN 300 440 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"

ETSI EG 203 367 Short Range Devices (SRD) ETSI EG 203 367 "Guide to the application of harmonized standards covering articles 3.1b and 3.2 of the Directive 2014/53/EU"

Federal Communications Commission FCC §15.247 Radio frequency devices; Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz

IO-Link Community, IO-Link Interface and System Specification, V1.1.3, June 2019, Order No: 10.002 (available at http://www.io-link.com)

IO-Link Community, IO Device Description (IODD), V1.1.3, January 2021, Order No. 10.012 (available at http://www.io-link.com)

IO-Link Community, IO-Link Smart Sensor Profile 2^{nd} Edition V1.1, September 2021, Order No. 10.042 (available at http://www.io-link.com)

IO-Link Community, IO-Link Common Profile, V1.1, December 2021, Order No. 10.072 (available at http://www.io-link.com)

IO-Link Community, IO-Link Test Specification V1.1.3, January 2021, Order No: 10.032 (available at http://www.io-link.com)

Bluetooth SIG – Core Specification 4.2,
(available at <https://www.bluetooth.com/de/specifications/archived-specifications/ >)

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

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.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1 acknowledge

response information indicating the acceptance of a message

3.1.2 Adaptive Hopping Table

mechanism that enables a change of the hopping table of a W-Track while the communication is already running (improvement of connection)

NOTE: see clause H.4

3.1.3 air interface

radio-based communication between the W-Master and the W-Devices

3.1.4 Alert Back Alive

indication from the W-Device, which informs the W-Master, a communication error has occurred.

3.1.5 Application Layer

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

3.1.6 blocklist

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

3.1.7 broadcast

mechanism to send a MasterCommand to all W-Devices in a W-Track

NOTE: A broadcast is restricted to specific Master commands according to Table 178.

3.1.8 cell

logical grouping of 1 or up to 3 W-Masters with a dedicated coverage area

NOTE: This is often associated to a machine.

3.1.9 checksum

data integrity measures for each Pre-Downlink octets, downlink packet or uplink packet in the Physical Layer

3.1.10 ConfigSyncword

syncword which shall be used for the configuration channels

3.1.11 configuration frequency channels

frequency channels are reserved for configuration purposes

NOTE: see clause 5.5.4.

3.1.12 ConnectionParameter

set of parameters containing the data which are necessary to establish wireless communication

NOTE: The parameters are transmitted via the pairing mechanism (see Table 154). 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

header, indicating the structure and purpose of a W-Message

NOTE: see clause A.3. and clause A.4

3.1.15 communication channel

logical connection between W-Master and W-Device

NOTE: Four communication channels are defined: MasterCommand 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: Not occupied retries by process data are used for acyclic exchange of on-request Data. The configuration channels are not used.

3.1.18 data channel

frequency channel used for (process) data exchange

3.1.19 DataSyncword

syncword used for the data channels

Note: More detailed explanation is given in 5.3.4

3.1.20 disconnected

loss of communication between a W-Device and its W-Master

3.1.21 Double Slot

type of time slot required for the transmission of a Double Slot packet

NOTE: More detailed explanation is given in 4.5.4

3.1.22 downlink

communication link from a W-Master to its associated W-Devices

3.1.23 event

instance of a change of conditions in a W-Device

NOTE: Uppercase "Event" is used for IOLW Event mechanism, while lowercase "event" is used in a generic manner.

3.1.24 frequency band

range in the frequency spectrum that is assigned by regulatory organizations for use for specific applications or a group of applications

NOTE: The ITU as international regulatory organization assigns only radio communication services to a specific range in the frequency spectrum.

{SOURCE: IEC 62657-2)

3.1.25 frequency channel

span of the frequency spectrum which is characterized by lower cut-off frequency and upper cut-off frequency or by center frequency and frequency bandwidth

{SOURCE IEC 62657-2, 2017}

NOTE: see 5.5.2, 5.5.4 and 5.5.5.

3.1.26 frequency division multiple access

access method where users are allocated to individual frequency channels

3.1.27 FullDownLink packet

data structure of the regular downlink packet

NOTE: Definitions see 5.3.8

3.1.28 gaussian frequency shift keying

binary frequency shift modulation with gaussian filter limiting its spectral width

3.1.29 guard interval

time interval between successive uplink packets to avoid collisions on air

3.1.30 header

first part of a message packet

NOTE: The header consists of preamble, Syncword, MasterID, W-Track number, and Acknowledge.

3.1.31 IMA message

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

3.1.32 IMA time

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

3.1.33 InspectionLevel

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

3.1.34 Indexed Service Data Unit

data unit is used for acyclic transmission of parameters which can be segmented in multiple W-Frames

3.1.35 industrial, scientific and medical application

operation of equipment or appliances designed to generate and use locally radio frequency energy for industrial, scientific, medical, domestic or similar purposes, excluding applications in the field of telecommunications
{SOURCE: Radio Regulations – Art.1 §1.15}

3.1.36 ISM frequency band

radio frequencies reserved for industrial, scientific, and medical use

3.1.37 link quality indication

service for evaluation of the functionality and reliability of the W-Master and W-Device connections

3.1.38 MasterID

identification number of a particular W-Master

3.1.39 multicast

transmission of the same message to a group of receivers, identified by their group address
NOTE: the term "multicast" is used even if the group includes all the receivers
SOURCE: IEC 61375-1:

3.1.40 negative acknowledge

indicator for a missing response message
NOTE: EXAMPLE A wrong CRC or no received DLink or ULink

3.1.41 negotiation

step within the pairing procedure for configuration of the ConnectionParameter

3.1.42 packet error probability

probability that a received packet contains at least one bit error (inverted bit with respect to the corresponding bit sent
NOTE: For the calculation see Annex K.

3.1.43 packet error ratio

number of lost or incorrectly received packets divided by the total number of sent packets

3.1.44 pairing

procedure to log in an IO-Link Wireless Device to an IO-Link Wireless Master
NOTE: Pairing is the equivalent procedure to plugging in a cable connection in a wired system between a master and a device.

3.1.45 payload

data of a packet transmitted during communication that does not contain any control or protocol information in respect of the relevant layer
NOTE: see W-Message Figure 120

3.1.46 Port and Device Configuration Tool

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

3.1.47 PortCycle

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

3.1.48 preamble

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

3.1.49 PreDownLink mechanism

mechanism used to minimize the radio-on time to save power of low energy W-Devices

NOTE: The 2 octet payload is used for low energy W-Devices only. Definitions see clause 5.3.8

3.1.50 residual failure probability

probability that the data cannot be transmitted within the set MaxRetry

3.1.51 received signal strength indication

estimated measure of power level that a W-Master (RSSI_M) or W-Device (RSSI_D) is receiving from its counterpart.

3.1.52 roaming

feature that allows mobility to a predefined W-Device between multiple predefined W-Masters by handover disconnect and handover connect procedures

3.1.53 segmented data

data to be splitted over multiple packets

3.1.54 SerialNumber

unique vendor specific code for each individual W-Device

3.1.55 Service Mode

operational mode in which a W-Master W-Track also utilizes the configuration channels

NOTE: This mode is required for discovery, pairing and roaming procedures. A W-Master shall only operate with one of its W-Tracks in Service Mode at the same time.

3.1.56 Single Slot

type of time slot required for the transmission of a Single Slot packet

NOTE: see 4.5.4.

3.1.57 synchronization

matching timing and the frequency tables between a W-Master and corresponding W-Devices

3.1.58 UniqueID

unique identifier for each W-Device

NOTE: 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 C.4.4.1.

3.1.59 unpairing

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

3.1.60 uplink

communication link from a W-Device to its associated W-Master

3.1.61 W-Bridge

W-Device that connects an IO-Link Device via wireless communication interface to a W-Master

3.1.62 W-Coexistence

state in which all wireless communication solutions of a plant using shared medium fulfil all their application communication requirements

3.1.63 W-Cycle

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

3.1.64 W-Device

single peer to a W-Master such as an IO-Link Wireless sensor or actuator

3.1.65 W-Frame

virtual structure to arrange message transmissions comprising a W-Master message (in downlink direction) and all subsequent W-Device messages (in uplink direction)

NOTE: For details see Figure 18.

3.1.66 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

3.1.67 W-Message

content of payload comprising control octet and (segmented) data exchanged between W-Master and W-Device

NOTE: see Figure 18

3.1.68 W-Parameter

generic term that describes all the parameters located in the IO-Link Wireless specific index range

NOTE: see clause C.4.

3.1.69 W-Port

logical IO-Link Wireless port number to address a paired W-Device

3.1.70 W-Sub-cycle

time duration to transmit one W-Frame

NOTE: see Figure 18.

3.1.71 W-Track

physically wireless transmission track comprising a (small band) transceiver with its own antenna and dedicated frequency channels for transmitting the messages between W-Master and W-Device by W-Frames

3.1.72 W-Track and W-Device-Mapper

assignation of a W-Port to a specific W-Track number and Slot number

3.1.73 IO-Link Device

Device according to IEC 61131-9

3.1.74 WLAN channels

frequency blocks intended to be used by WLAN

3.2 Abbreviated terms

ABA	Alert Back Alive
AL	Application Layer
BC	Broadcast
CO	Control octet
CRC	Cyclic redundancy check
DLink	Downlink
DSlot	Double Slot
FDMA	Frequency division multiple access
GFSK	Gaussian frequency shift keying
IL	InspectionLevel
IMA	I am alive
IOL	IO-Link
IOLW	IO-Link Wireless

ISDU	Indexed Service Data Unit
ISM	Industrial, scientific and medical application
LE	Low energy
LQI	Link quality indication
LQI_D	Link quality indication of the W-Device
LQI_M	Link quality indication of the W-Master
NACK	Negative acknowledge
PDCT	Port and Device Configuration Tool
PEP	Packet error probability
PER	Packet error ratio
PLC	Programmable logic controller
RFP	Residual Failure Probability
RSSI	Received Signal Strength Indication
RSSI_D	Received signal strength indication, received by a W-Device
RSSI_M	Received signal strength indication, received by a W-Master
SDCI_W	Wireless digital communication interface
Slot_N	Slot number
SSlot	Single Slot
TDMapper	W-Track and W-Device-Mapper
Track_N	W-Track number
ULink	Uplink
WLAN	Wireless local area network
HMI	Human machine interface
OD	On-request data

3.3 Conventions

3.3.1 General

The service model, service primitives, and the diagrams shown in this document 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 2 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.

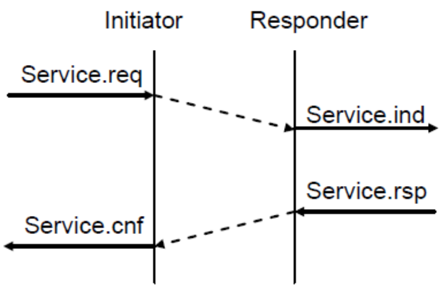


Figure 2 Generalized example of a confirmed service

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 document 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 can 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 3 shows the example of initiator / receiver of services (W-Device)

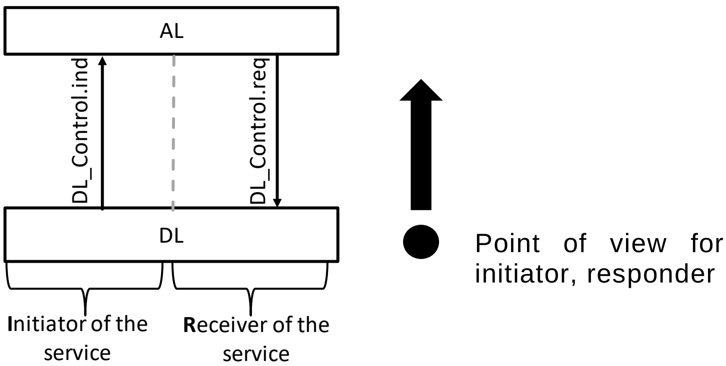


Figure 3 Example of initiator / receiver of services (W-Device)

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

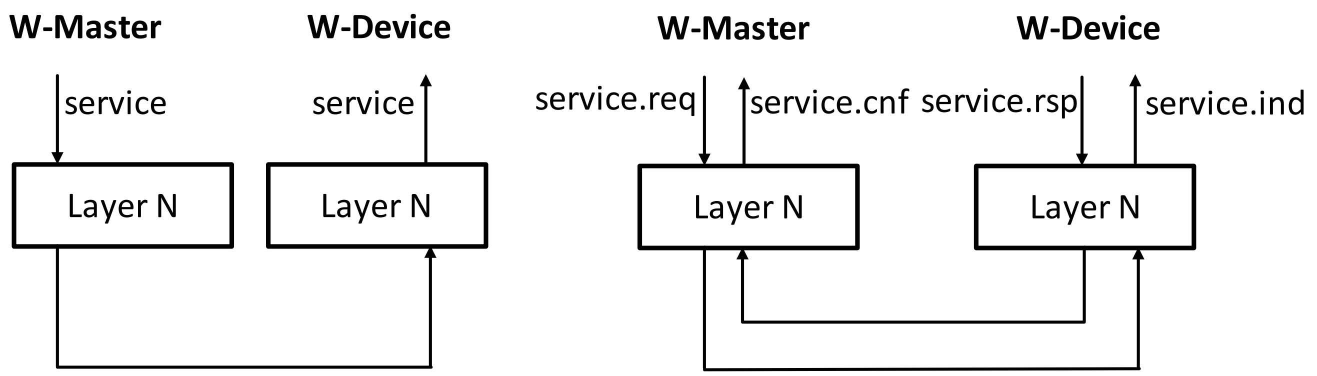


Figure 4 Example of service between W-Master and W-Device in generalized and detailed view

Figure 5 shows the example of confirmed / non-confirmed services (W-Device)

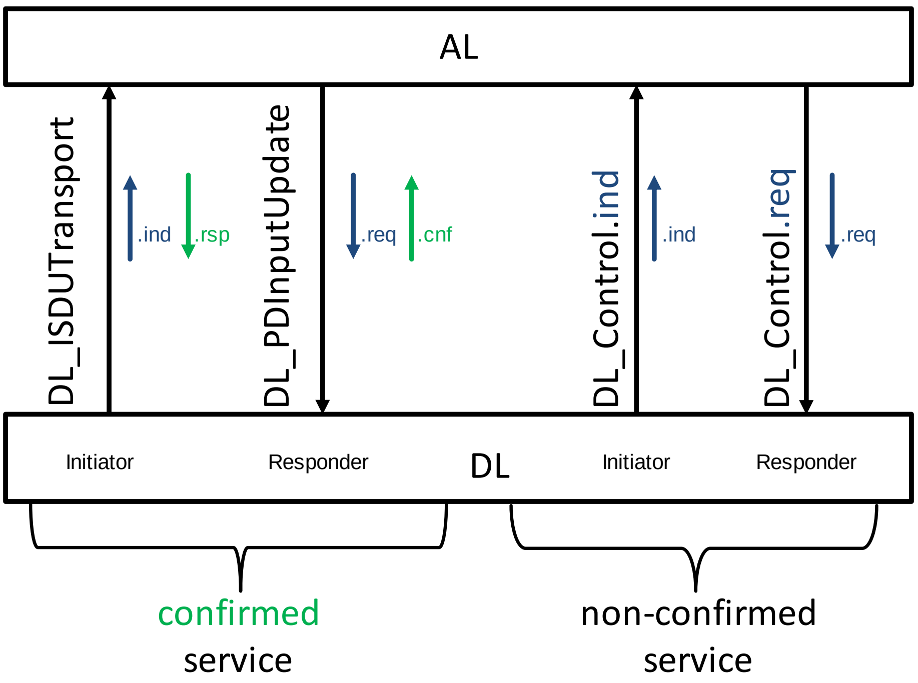


Figure 5 Example of confirmed / non-confirmed services (W-Device)

3.3.7 Transmission octet order

See clause 5.3.1

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 industrial, scientific and medical application (ISM) band and using frequency hopping to reduce the impact of interference. IO-Link Wireless specifies the communication between the IO-Link Wireless Master (W-Master) and one or more IO-Link Wireless Devices (W-Devices).

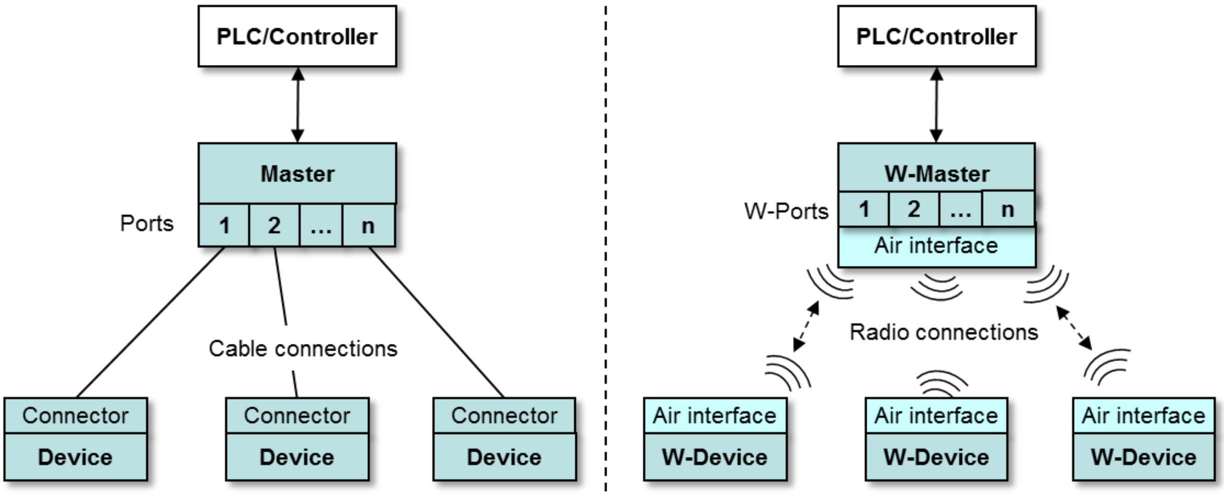


Figure 6 IO-Link and IO-Link Wireless topology

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

4.2.1 General

Figure 7 shows the architecture of an automation topology with an IO-Link Wireless system comparable to an IO-Link system.

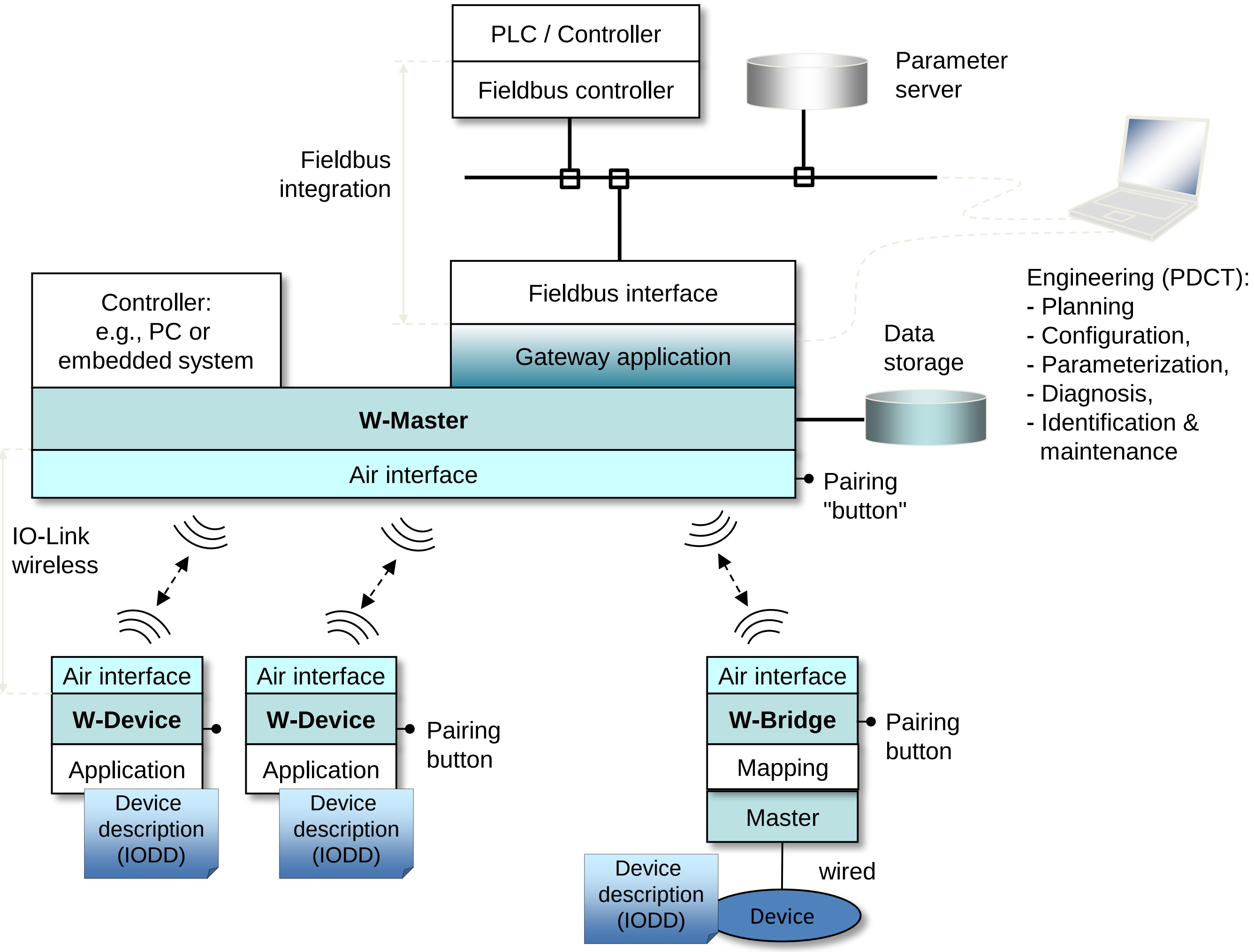


Figure 7 IO-Link Wireless system

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 ([1]) clause 13.5 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 [1] to support faulty W-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.

4.2.2 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 8.

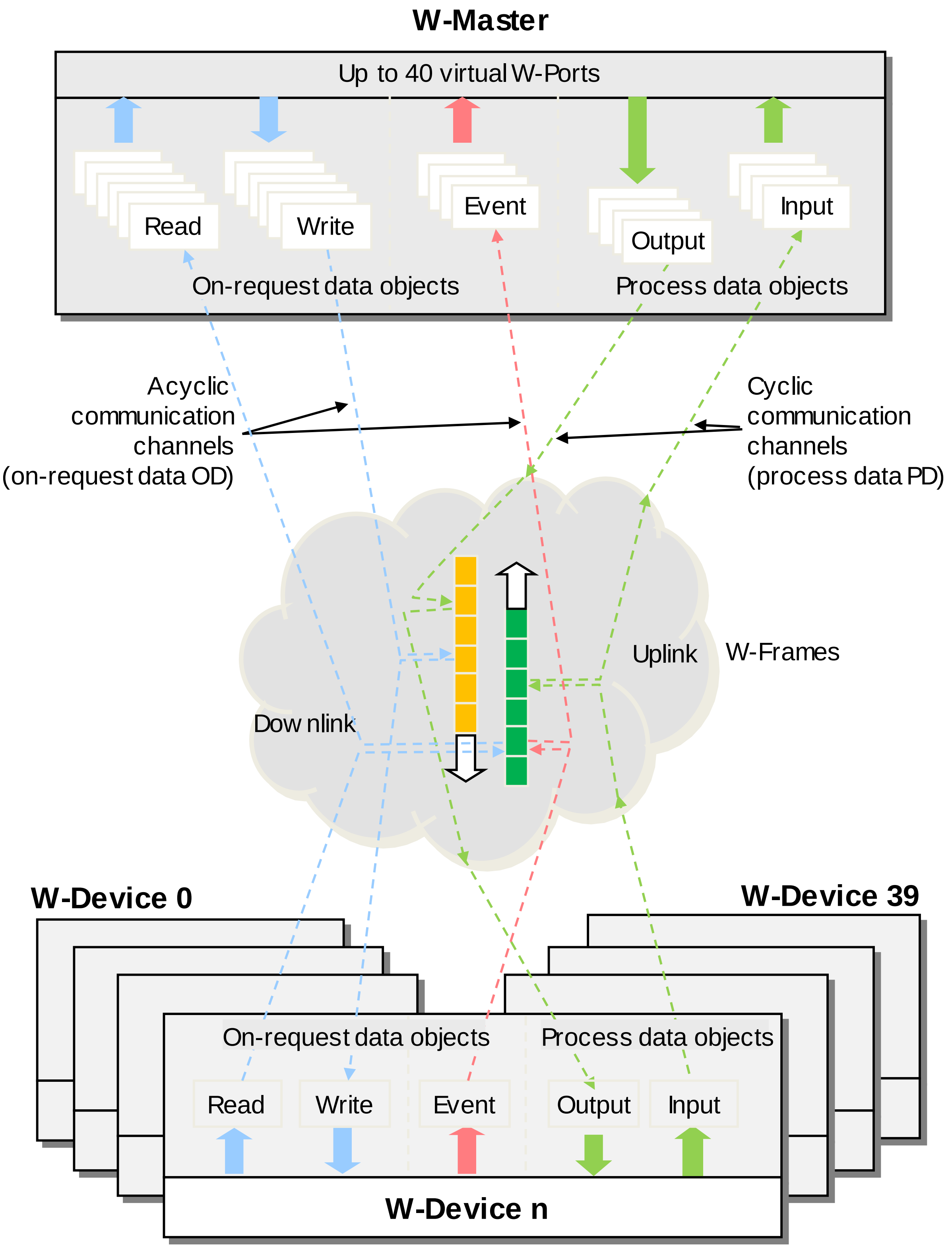


Figure 8 Object transfer at the Application Layer level (AL)

4.2.3 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 W-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 W-Tracks. Each W-Track can serve up to 8 W-Devices and send and receive alternately. All W-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 W-Devices.
During commissioning or roaming Service Mode is used by the W-Master to establish communication with W-Devices (pairing), including 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 11.4) 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).

4.2.4 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.

4.2.5 Role of a W-Bridge

A W-Bridge is a W-Device to connect a single IO-Link Device. The application part of the W-Bridge basically contains an IO-Link Master.
For compatibility reasons towards the IO-Link Device in the System Configuration Tooling, a straightforward parameter mapping of the IO-Link Device via the W-Bridge is desired.

4.2.6 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 11.8.2). The parameters provide all the necessary properties to establish communication and the desired function of a sensor or actuator.

4.2.7 Mapping to fieldbuses

See clause 4.7 in [1] and clause 11.9.1.

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 W-Track as shown in Figure 9. This value is derating to ≤ 10 m if more than one W-Track is active.

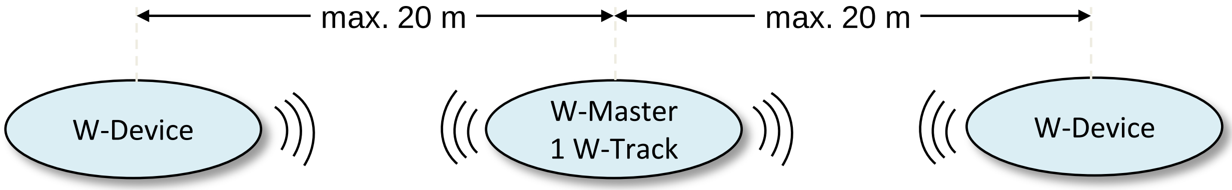


Figure 9 Radius of a cell with a 1 W-Track W-Master

A single W-Master can consist of one up to five W-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 consecutive. To prevent frequency access conflicts between the W-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 blocklist.

One W-Master and a group of associated W-Devices form a W-Master cell as shown in Figure 10. 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.

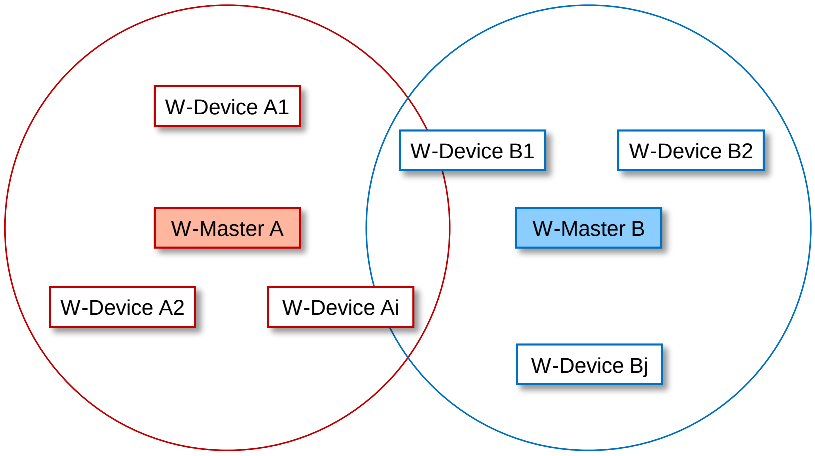


Figure 10 Cell consisting of 2 W-Master cells

Figure 11 shows the IO-Link Wireless concept with partly overlapping W-Master RF coverage areas . In one area, there should not exist more than three W-Masters in order to avoid interference. W-Master cells with a distance of more than 40 m can use the same MasterID again. W-Masters with a distance less than 40 m require distinct MasterIDs. The MasterID is used to calculate individual frequency hopping tables.

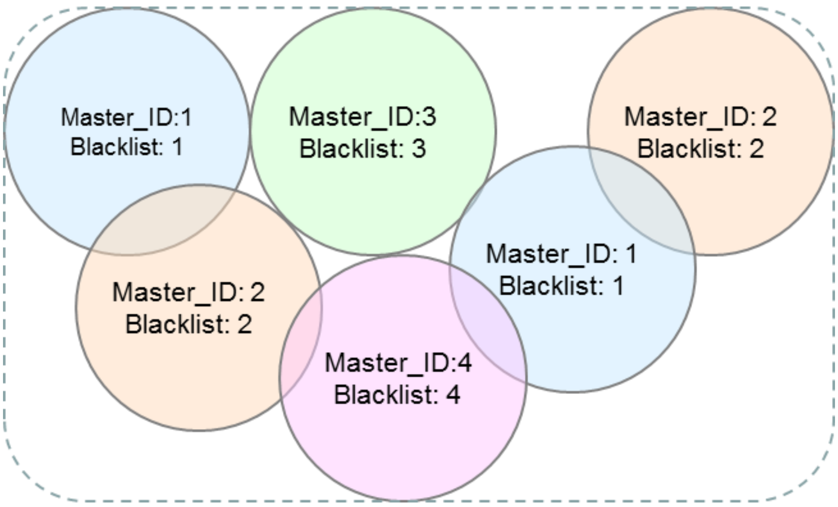


Figure 11 Cell concept

4.4 Wireless Mechanisms

4.4.1 General

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

4.4.2 Scan (W-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 an uplink slot. 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.

4.4.3 Pairing

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

4.4.3.1 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.

4.4.3.2 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 pairing of both partners without using an Engineering Tool or PDCT.

4.4.3.3 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 shall be started previously on the W-Master by manual intervention (i.e., by UniqueID or pressing a button).

4.4.4 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.

4.4.5 Roaming

Roaming is a feature that allows mobility to a predefined W-Device between multiple predefined W-Master cells. A W-Master W-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 can then decide to connect the roaming W-Device by initiating a pairing and configuration sequence.

The support of this function is mandatory for a W-Device and optional for a W-Master.

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 Wireless Quality of 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.

NOTE: 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 shall lead to an autonomous handover disconnect of this W-Device for the associated W-Port within the W-Master.

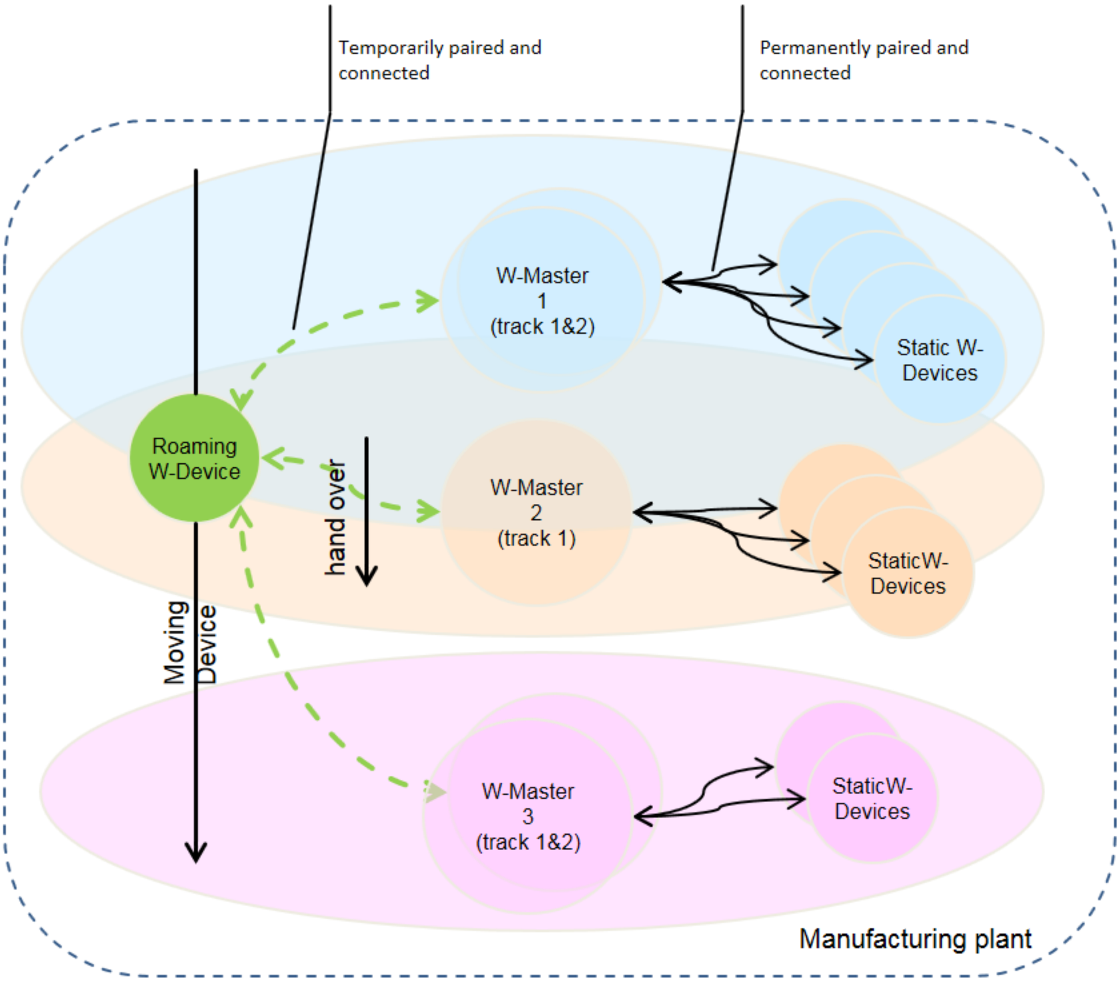


Figure 12 Roaming between W-Master cells

“Roaming” is configurable on the W-Master. On each W-Master, not more than one W-Track shall be configured for Roaming Mode, as indicated in Figure 12 for W-Master 1 and W-Master 3. The W-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 H.2

4.4.6 Automatic Pairing

Automatic pairing (AutoPairing) is a feature that allows the pairing of W-Devices to W-Ports directly after the configuration is set without the need for additional user interaction. The automatic pairing is configured in the "PortMode" element in parameter "WPortConfigList" (see E.1.2). If the “PortMode” element is set to “Cyclic AutoPairing” or “Roaming AutoPairing”, the AutoPairing is activated for this port.
The behavior of the AutoPairing mechanism depends on the active mode of the service track of the W-Master:

- If the service track is in cyclic mode, the pairing is triggered once after the W-Port configuration is set. If the configured W-Device is not available at that time, the pairing procedure will fail. The user will have to start the pairing manually again when the W-Device is available.
- If the service track is in roaming mode, the pairing is triggered once after the W-Port configuration is set. If the W-Device is not available at that time, the pairing procedure will fail. As the W-Master

is permanently scanning for W-Devices in this mode, the pairing can be automatically triggered again, when the configured W-Device is found in the scan results.

The used pairing method for AutoPairing is always “Pairing by UniqueID” as described in clause 4.4.3.1

4.4.7 Transmission Error Handling

Transmitted packets in both uplink and downlink direction are subject to error detection via CRC and shall 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.

4.4.8 “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 shall 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 shall be indicated via system management.

4.4.9 Wireless Quality supervision

Wireless Quality is used for evaluation of the functionality and reliability of the IO-Link Wireless system in its application environment of the wireless connection. The parameter WirelessQuality 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.

4.5 Concept of Air Interface

4.5.1 General

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 [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

4.5.2 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 ("Blocklisting").

4.5.2.1 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 blocklist 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 H.2

4.5.2.2 Blocklist

To suspend frequency ranges, they can be defined in the blocklist. The blocklist is configurable via SetMasterConfig. For more details, see clause H.1.

4.5.2.3 Configuration channels

Configuration of W-Master and W-Devices is required prior to cyclic data exchange. For this purpose, the frequency channels "1" (2401 MHz) and "80" (2480 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 H.3

4.5.2.4 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 H.2).

4.5.3 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 W-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.

4.5.3.1 Downlink

The downlink communication from W-Master to its W-Devices can contain W-Messages for several W-Devices as shown in Figure 13. 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 W-Track.

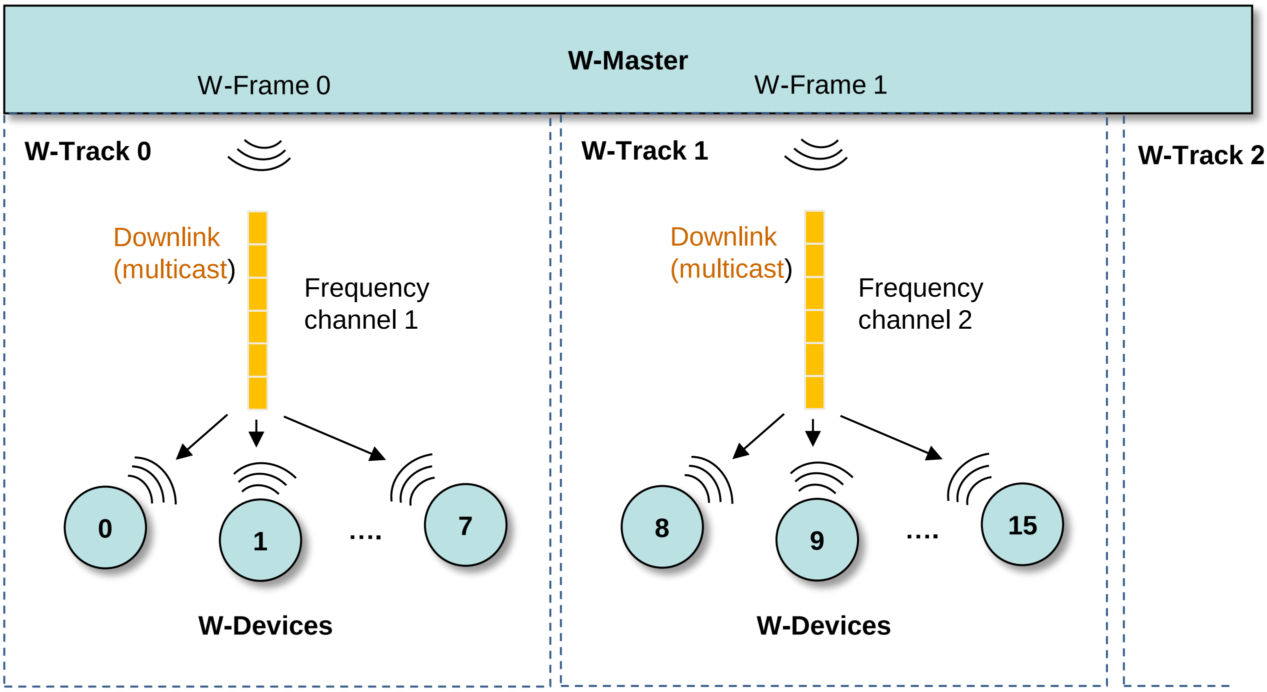


Figure 13 Downlink

4.5.3.2 Uplink

Figure 14 demonstrates the timely staggered delivery of single cast W-Messages of W-Devices to their W-Master.

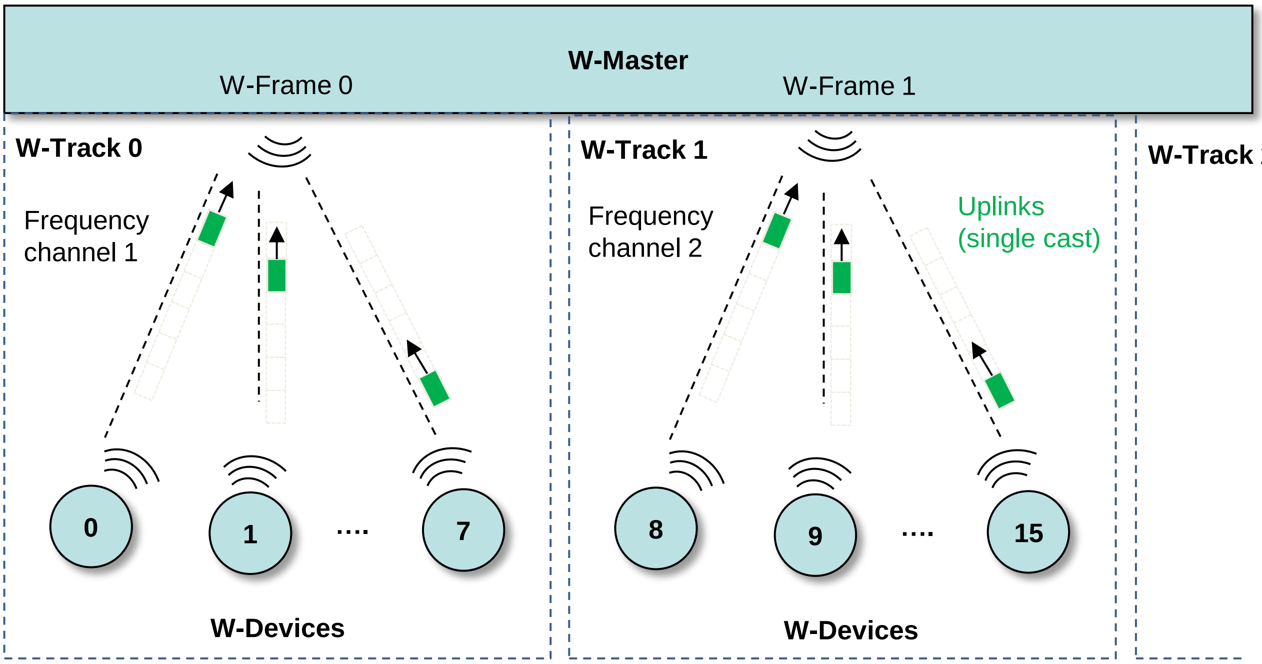


Figure 14 Uplink

4.5.3.3 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 time slots, 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.

4.5.4 SSlots, DSlots, Transmission capacity

The transmission capacity of downlink and uplink is shown in Figure 15. The Downlink can carry 52 octets. An uplink can carry 12 octets or 25 octets, depending on the slot type "SSlot" or "DSlot". DSlots combine two SSlots to operate sensors or actuators with larger process data, but this reduces the number of possible W-Devices per W-Track.

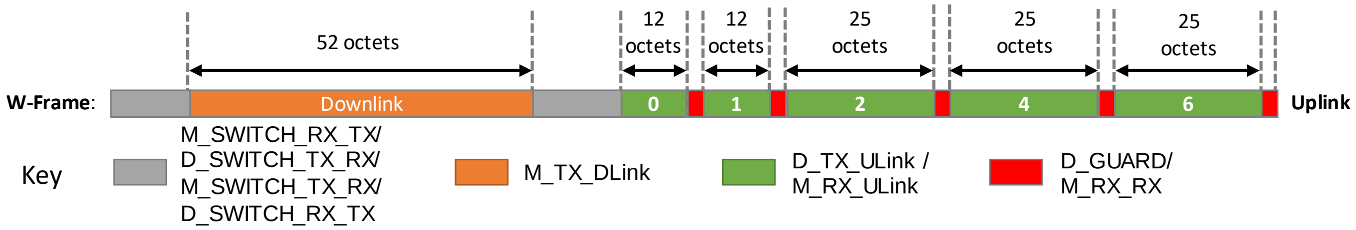


Figure 15 Transmission capacity with SSlots and DSlots

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 D.

4.5.5 Assignment of W-Devices to W-Tracks and slots

A W-Master contains up to 5 W-Tracks, which are numbered from 0 to 4. Each W-Track has up to 8 slots, numbered from 0 to 7. This allows a maximum number of 40 W-Devices per W-Master.
Figure 16 shows the assignment of W-Device numbers to slots and W-Tracks.
The allocation of W-Devices to W-Track and Slot number is performed during commissioning and pairing.

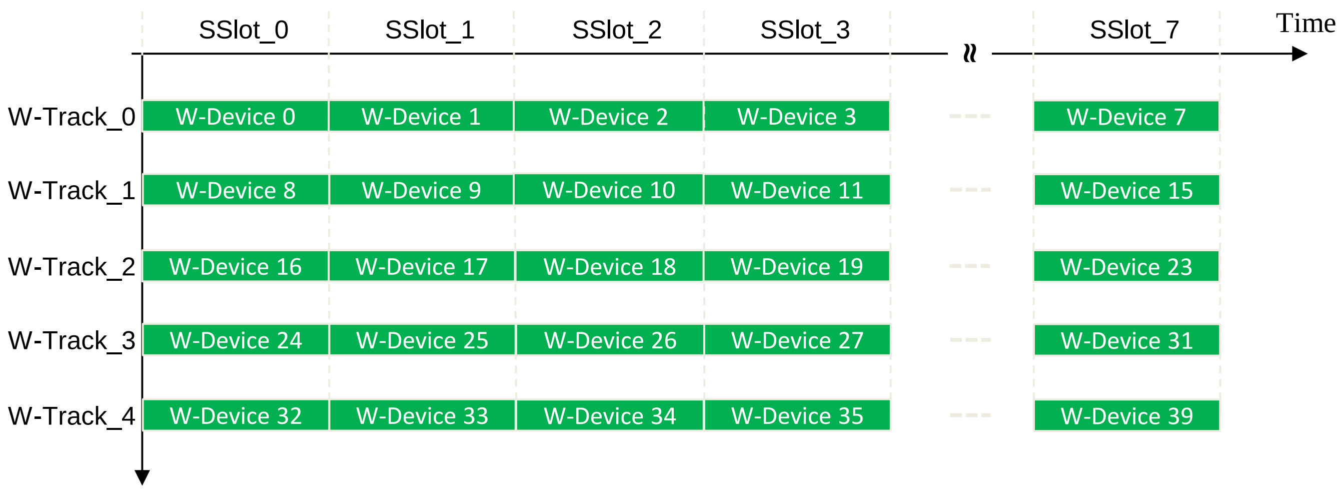


Figure 16 Uplink assignments

Numbering gaps in the W-Device count can occur because of DSlot usage (DSlots shall always be placed on even slots, see Figure 33) or non-used slots in a W-Track.

4.5.6 Assignment of W-Ports to W-Devices

A W-Master provides a limited number of virtual W-Ports, depending on the number of available W-Tracks and the slot configuration, since W-Devices with occupation of a DSlot reduce the number of available W-Ports. The W-Master shall 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 W-Track and W-Device-Mapper (TD-Mapper) uses this information to map a W-Port to the corresponding W-Track and slot (see 6.2.2).

4.5.7 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 17, 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 W-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 W-Track. Each W-Track can serve up to 8 W-Devices and send and receive alternately. All W-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.

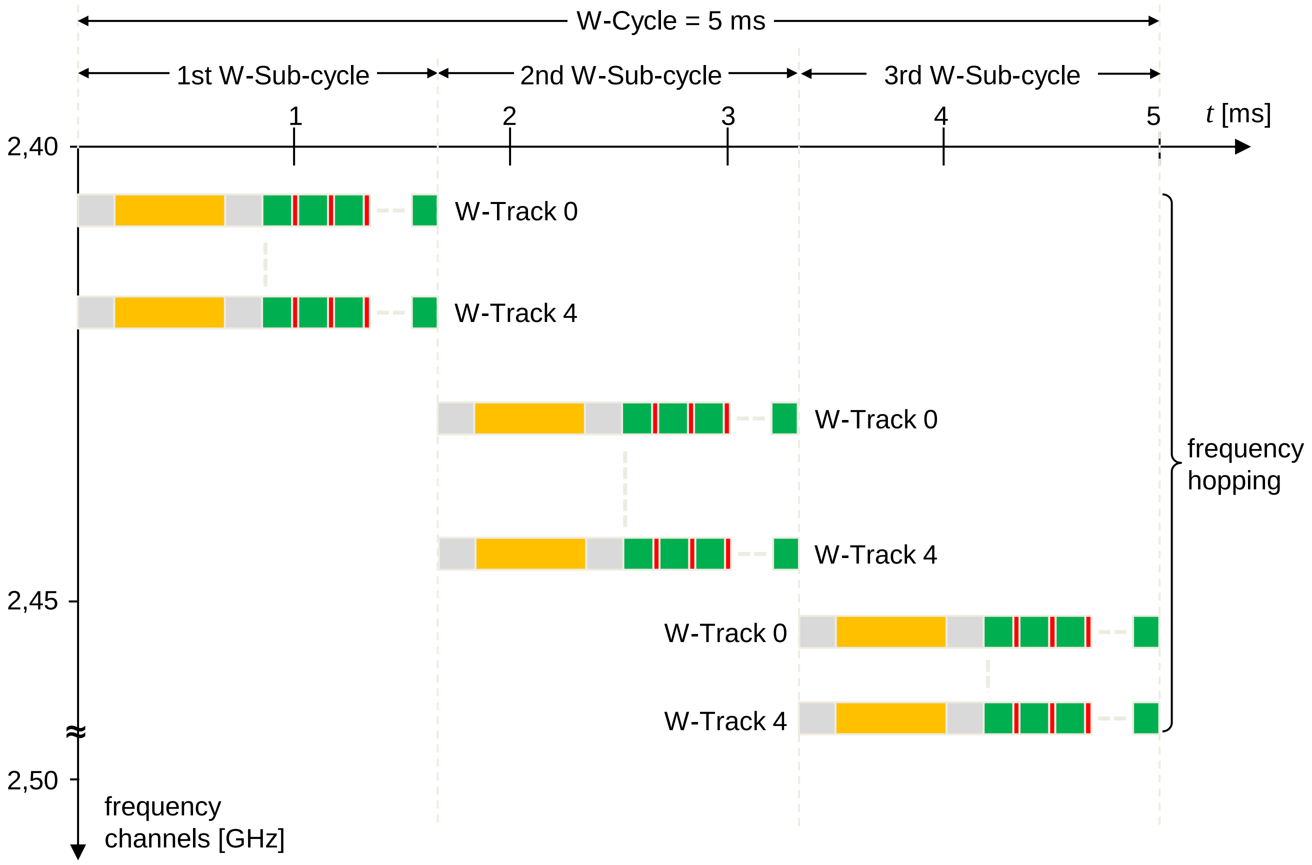


Figure 17 TDMA and FDMA in the W-Cycle

4.5.8 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 18). 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 the W-Devices via multicast. The Uplinks is transmitted subsequently W-Device by W-Device in the respective time slot. The W-Frame is transmitted in a W-Sub-cycle of 1,664 ms.

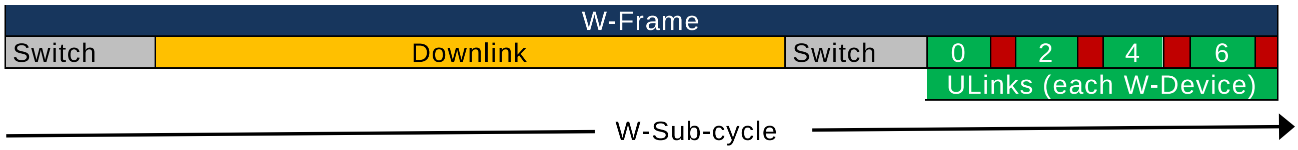


Figure 18 W-Frame and W-Sub-cycle

4.6 Characteristics of the IO-Link Wireless technology

The main characteristics of the IO-Link Wireless technology see clause 0

4.7 Layer model

Different aspects of communication are realized by different communication layers based on the layer model according to IEC 61158-1.
Figure 19 shows the logical structure of W-Master and W-Device.

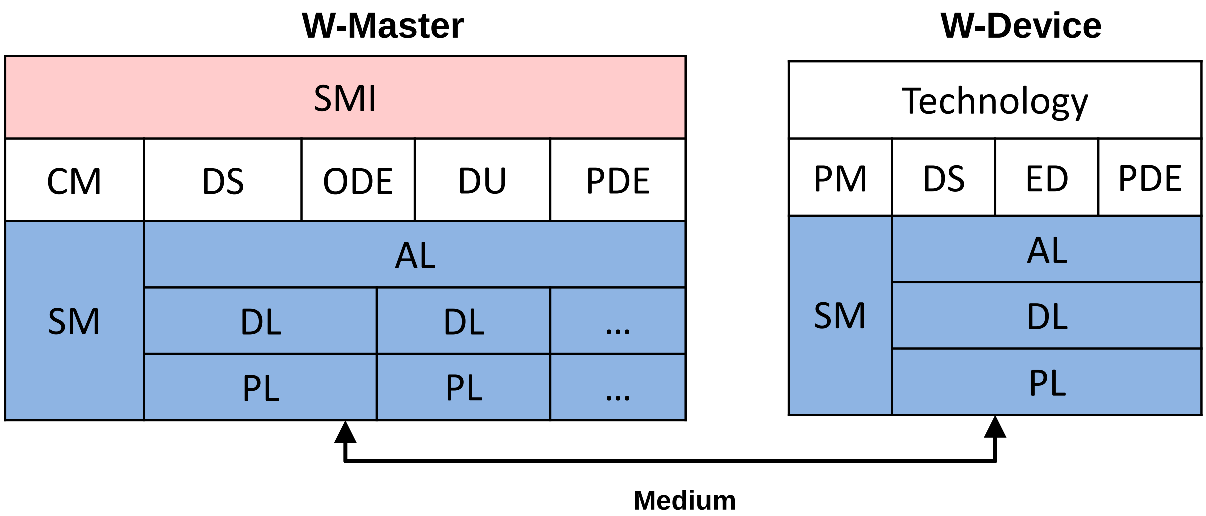


Figure 19 Logical structure of W-Master and W-Device

The “PL” (Air interface) for IO-Link Wireless includes the Physical Layer as well as the wireless mechanisms (e.g., pairing, blocklisting, …) 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 Specification for the operation of IO-Link Wireless communication.

4.8 Conformity assessment

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

5 Physical Layer (PL)

5.1 General

This clause describes the relevant definitions for transceivers and media access on both, W-Master and W-Devices, which shall comply to the requirements described below.

5.2 Base technology, Physical Layer (PL)

5.2.1 General

IO-Link Wireless uses frequencies from 2401 to 2480 MHz of the license-free 2,4 GHz ISM band (industrial, scientific, and medical).
The 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, shall be taken into account.

5.2.2 Transmission rate

The on-air bit duration T_{bit} is 1 µs as shown in Figure 20. Hence, the gross transmission rate is 1 Mbit/s.

5.2.3 Carrier frequency accuracy

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

5.2.4 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.

5.2.5 W-Master Carrier frequency calibration

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

5.2.6 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. (a modulation index value between 0.45 and 0.55 is tolerable).

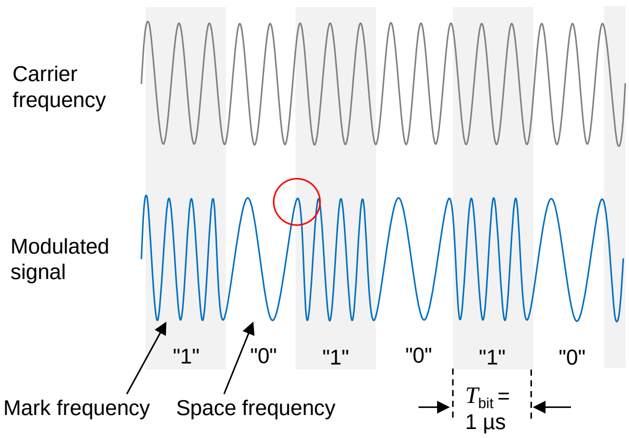


Figure 20 Base technology and modulation

Figure 20 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 20) and cause interfering harmonics. A Gaussian filter reduces this impact. The entire modulation mechanism is named Gaussian Frequency Shift Keying (GFSK).

5.2.7 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 W-Track is used, all W-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.
The transmission power shall be sufficiently high in order to ensure the range and reliability requirements.
The one-track W-Master shall be capable to transmit with a power of at least +6 dBm.
multi-track W-Masters shall be capable to transmit with a power of at least 0 dBm per W-Track.
The W-Device shall be capable to transmit with a power of at least +4 dBm.

5.2.8 Antenna

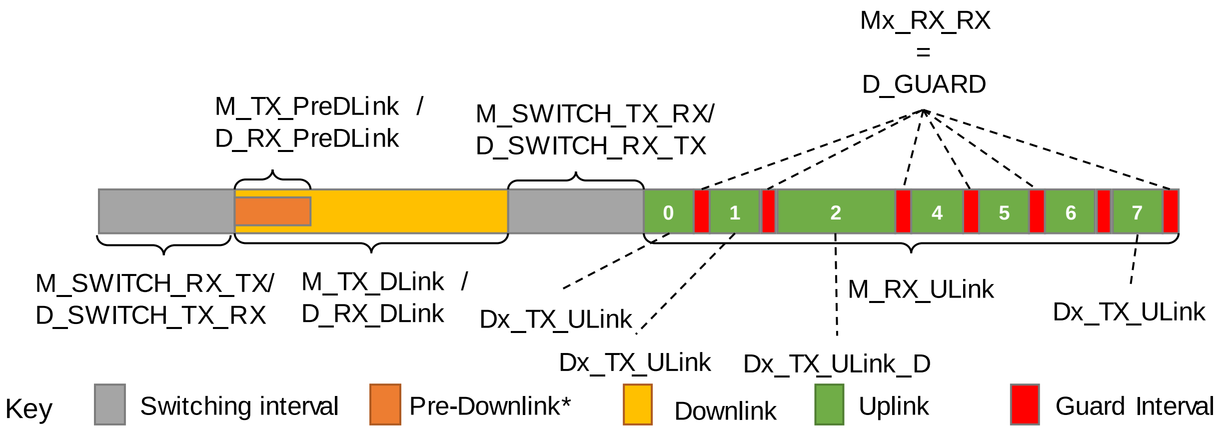
If radio regulations (see 5.2.7) are met, a W-Master or W-Device can use internal or external antennas. If an antenna with direction characteristic is used, also the maximum transmission power of ≤ 10 dBm EIRP shall be observed for any direction.
In order to ensure the range and reliability requirements, a high antenna efficiency is preferable.

5.2.9 IO-Link Wireless receiver sensitivity

The minimal radio sensitivity on the W-Master and W-Device side shall be at least -94 dBm. For details see Annex K.

5.2.10 Transceiver timings

To meet the necessary timings for the W-Sub-cycle as shown in Figure 21 (see clause 5.4) the requirements in Table 1 and Table 2 shall be fulfilled.



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

Figure 21 Transceiver timings

944 **Table 1 Transceiver timings within W-Sub-cycle of W-Master**

W-Master					
Name (see Figure 21)	**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 W-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 packet 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} + D_GUARD) only DSlot: 4 * (200 T_{BIT} + D_GUARD) or mix of SSlot and DSlot 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 than given time of 8T_{BIT}

945
946

Table 2 Transceiver timings within W-Sub-cycle of W-Device

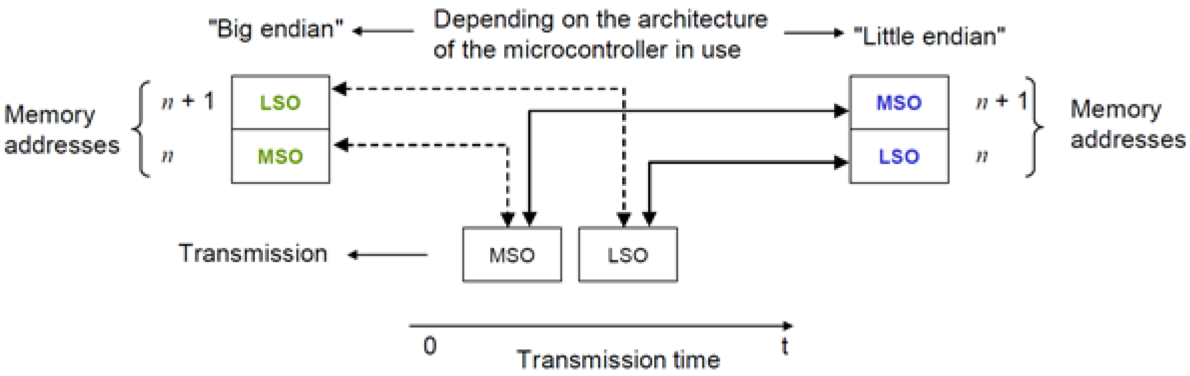
W-Device					
Name (see Figure **21**)	**Minimum**	**Typical**	**Maximum**	**Unit**	**Remark**
Radio frequency deviation	-250	0	250	kHz	The maximum carrier frequency error, which can be tolerated by radio
Frequency correction step	n/a	25	n/a	kHz	Frequency step used by correction of the carrier frequency 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-Device shall receive a complete Downlink packet with 416 bits.
D_RX_PPE-DLink	n/a	88	n/a	T_{BIT}	The low energy W-Devices shall receive a Pre-Downlink part of the complete Downlink packet with 88 bits.
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

5.3 Downlink and Uplink

5.3.1 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 22. The following rule shall apply:

- The Most Significant octet (MSO) transmitted first.



Key

MSO = Most Significant octet

LSO = Least Significant octet

Figure 22 Memory Storage and transmission order for values for WORD based data types

5.3.2 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 23.

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 23 Bit ordering within an octet

The radio transmits payload octets as an octet array over the air as shown in Figure 24. The order of the octets is not altered during data transmission/reception.

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

octets on air

Figure 24 Octet array transmission over the air

5.3.3 Preamble

Each Downlink or Uplink packet 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 Table 3.

Table 3 Octet ordering of Preamble values

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

5.3.4 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 Table 4.

Table 4 Octet ordering of ConfigSyncword

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

DataSyncword

Different syncwords can be used to improve the coexistence behavior in an environment with multiple W-Masters. This syncword is used for the data channels. The octets of the DataSyncword shall have the values and octet order shown in Table 5. This DataSyncword shall be delivered during Pairing Request from the W-Master to the W-Device (see Figure 128). According to this document, only a DataSyncword from Table 5 shall be used. The default DataSyncword is 3E 94 59.

Table 5 List of DataSyncwords

DataSyncword octet 0	DataSyncword octet 1	DataSyncword octet 2
3E	94	59
B3	A3	14
1A	2D	EE
94	A3	DB
9C	92	E8
38	D2	EE
D8	C6	A9
91	B5	E3
2D	A3	13
69	D1	39
D8	72	1D
91	DB	E2
E9	5C	13
C7	A6	14
44	6B	63
A7	72	D2
31	4D	ED

DataSyncword octet 0	DataSyncword octet 1	DataSyncword octet 2
96	39	15
68	5D	72
C6	74	2B
38	B7	15
C9	29	17
8D	1C	4B
DC	9A	5C
49	DC	5C
92	5D	73
6C	DC	2D
A7	9C	2D
68	A3	73
A4	73	17

5.3.5 Downlink and Uplink CRC

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

5.3.6 CRC Transmission

The result of the CRC16 and CRC32 shall be stored in a *Big Endian* format in the transmit buffer. The CRC is transmitted with most significant bit first, see Figure 25 and Figure 26.

Octet	0								1							
Octet-Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
CRC-Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 25 Octet ordering of CRC16 result values

Octet	0								1								2								3							
Octet-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
CRC-Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 26 Octet ordering of CRC32 result values

5.3.7 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 27 shows the realization of the whitening pseudo random number generator using a shift register with a feedback:

Data Whitening LFSR

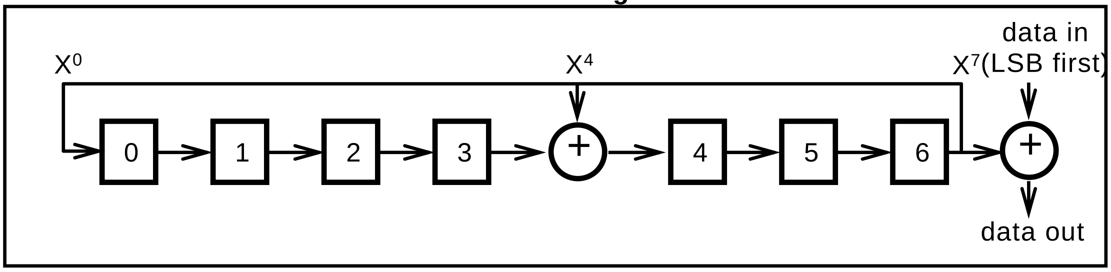


Figure 27 Data Whitening LFSR

For each packet transmission the shift register shall be initialized with a seed value which depends on the current frequency channel number. Table 6 illustrates the relationship between frequency channel number and shift register seed value.

Table 6 LFSR Seed Values

Frequency Channel Number	LFSR Seed Value							
	Bit Position							Value in HEX
	0	1	2	3	4	5	6	[Bit 6 = MSB]
1, 41	1	0	0	0	0	0	1	0x41
2, 42	1	0	0	0	0	1	0	0x21
3, 43	1	0	0	0	0	1	1	0x61
4, 44	1	0	0	0	1	0	0	0x11
5, 45	1	0	0	0	1	0	1	0x51
6, 46	1	0	0	0	1	1	0	0x31
7, 47	1	0	0	0	1	1	1	0x71
8, 48	1	0	0	1	0	0	0	0x09
9, 49	1	0	0	1	0	0	1	0x49
10, 50	1	0	0	1	0	1	0	0x29
11, 51	1	0	0	1	0	1	1	0x69
12, 52	1	0	0	1	1	0	0	0x19
13, 53	1	0	0	1	1	0	1	0x59
14, 54	1	0	0	1	1	1	0	0x39
15, 55	1	0	0	1	1	1	1	0x79
16, 56	1	0	1	0	0	0	0	0x05
17, 57	1	0	1	0	0	0	1	0x45
18, 58	1	0	1	0	0	1	0	0x25
19, 59	1	0	1	0	0	1	1	0x65
20, 60	1	0	1	0	1	0	0	0x15
21, 61	1	0	1	0	1	0	1	0x55
22, 62	1	0	1	0	1	1	0	0x35
23, 63	1	0	1	0	1	1	1	0x75
24, 64	1	0	1	1	0	0	0	0x0D
25, 65	1	0	1	1	0	0	1	0x4D
26, 66	1	0	1	1	0	1	0	0x2D
27, 67	1	0	1	1	0	1	1	0x6D
28, 68	1	0	1	1	1	0	0	0x1D

Frequency Channel Number	LFSR Seed Value							
	Bit Position							Value in HEX
	0	1	2	3	4	5	6	[Bit 6 = MSB]
29, 69	1	0	1	1	1	0	1	0x5D
30, 70	1	0	1	1	1	1	0	0x3D
31, 71	1	0	1	1	1	1	1	0x7D
32, 72	1	1	0	0	0	0	0	0x03
33, 73	1	1	0	0	0	0	1	0x43
34, 74	1	1	0	0	0	1	0	0x23
35, 75	1	1	0	0	0	1	1	0x63
36, 76	1	1	0	0	1	0	0	0x13
37, 77	1	1	0	0	1	0	1	0x53
38, 78	1	1	0	0	1	1	0	0x33
39, 79	1	1	0	0	1	1	1	0x73
40, 80	1	1	0	1	0	0	0	0x0B

5.3.8 Regular Downlink packet

The data structure of the Regular Downlink packet is shown in Figure 28. The distribution of payload of Pre-Downlink and Full-Downlink is dynamically assembled by Message handler (see Annex D). The data structure of the Downlink packet is described in clause B.2 in detail.

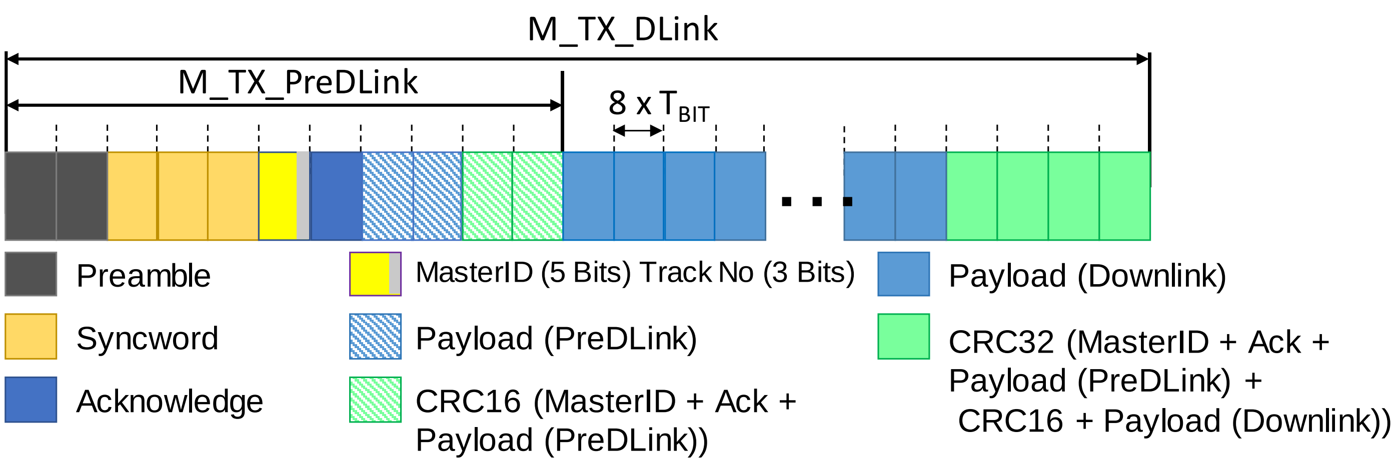


Figure 28 Regular Downlink packet

5.3.9 Configuration Downlink packet

The data structure of the Configuration Downlink packet is shown in Figure 29. The content of the payload is compiled by the Physical Layer (see Annex D). Data structure of the configuration Downlink is described in B.3. in detail.

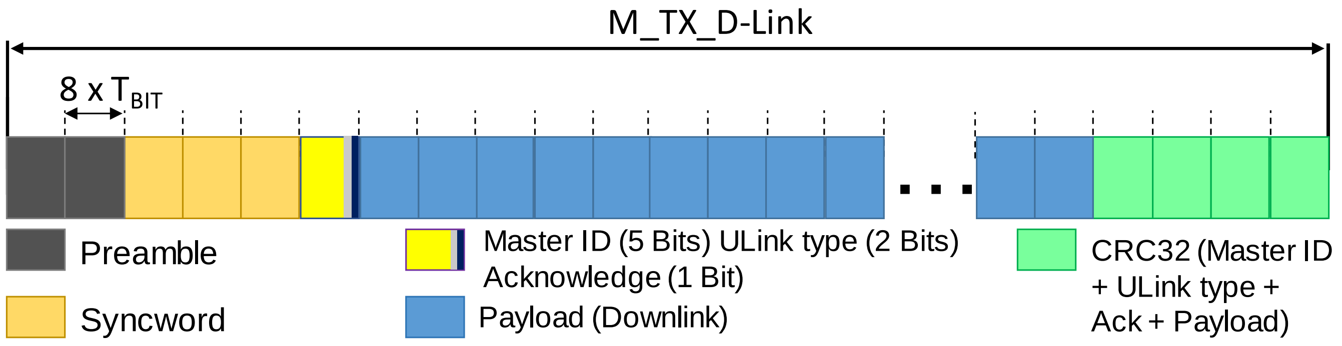


Figure 29 Configuration Downlink packet

5.3.10 Uplink Single Slot (SSlot)

Structure of the SSlot is shown in Figure 30. The DL-A handler compiles the Uplink payload. There are maximal 8 SSlots possible per W-Track of a W-Frame.

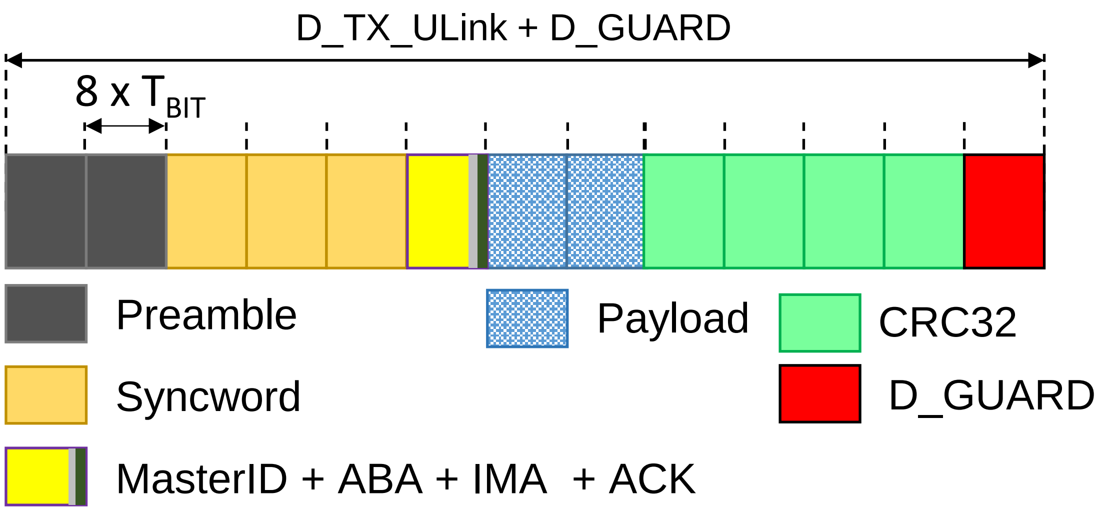


Figure 30 Uplink packet + Guard (SSlot)

5.3.11 Uplink Double Slot (DSlot)

Structure of the DSlot is shown in Figure 31. The DL-A handler compiles the Uplink payload. There are maximal 4 DSlots possible per W-Track of a W-Frame.

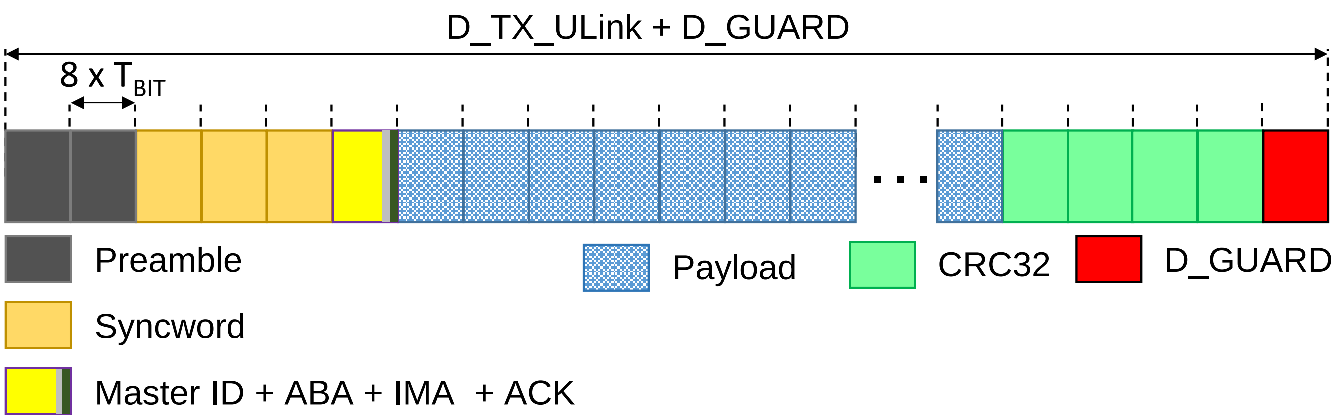


Figure 31 Uplink packed + Guard (DSlot)

5.4 W-Sub-cycle

5.4.1 General

The general concept of the W-Cycle and the W-Sub-cycles is specified in Figure 17. The following subclauses explain definitions for packets within a W-Frame.

5.4.2 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 32). The detailed encoding of W-Messages within W-Sub-cycles are described in Annex D.

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 32.

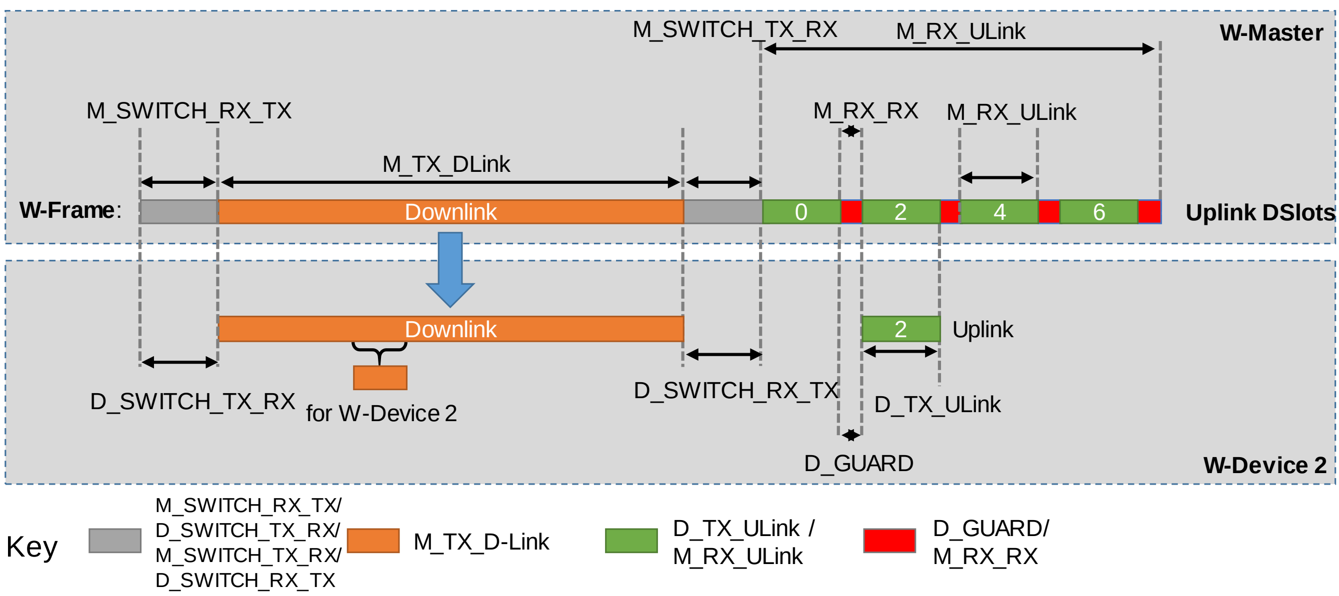


Figure 32 Format of a W-Sub-cycle with DSlots

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 transmissions 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 32.
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 30) or Double Slots (DSlot) with a length 200 µs (see Figure 31). Only by using SSlots, the maximum number of 8 W-Devices per W-Track can be achieved. DSlots shall always start at an even slot number. If in a W-Track an odd number of SSlots is used one SSlot cannot be used e.g., SSlot 5 in Figure 32.

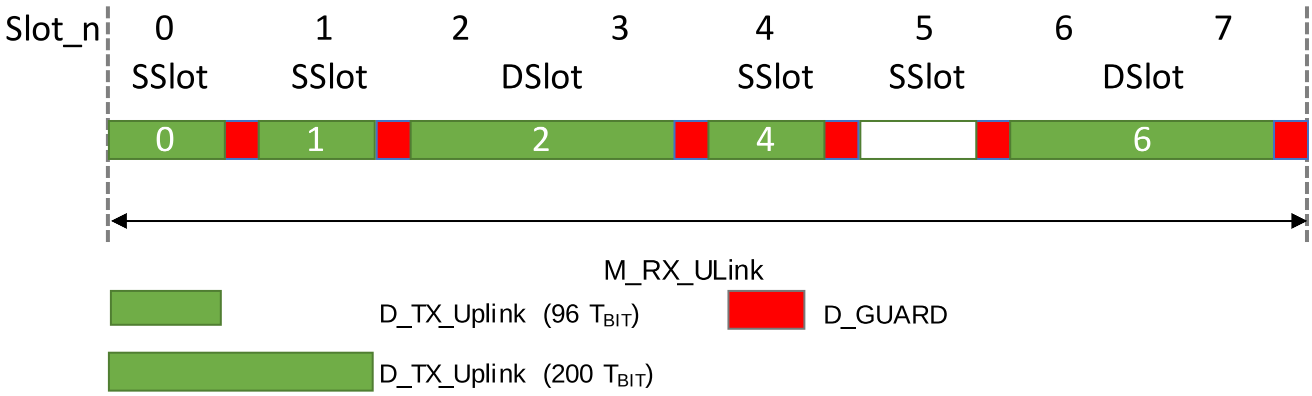


Figure 33 SSlots and DSlots

5.4.3 Regular W-Frame

Figure 34 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 packet addressed to dedicated W-Devices.
The first part of the Downlink packet, the so-called Pre-Downlink, is integral part of the full Downlink packet for regular W-Frames. It contains the acknowledgments and two octets payload. Pre-Downlink part has its own 16 bit CRC signature. Low energy W-Devices may reduce their receiver activity time by only receiving the Pre-Downlink part instead of the full Downlink packet.
All other W-Devices of the W-Track shall receive the entire Downlink packet and the 32 bit CRC signature at the end.

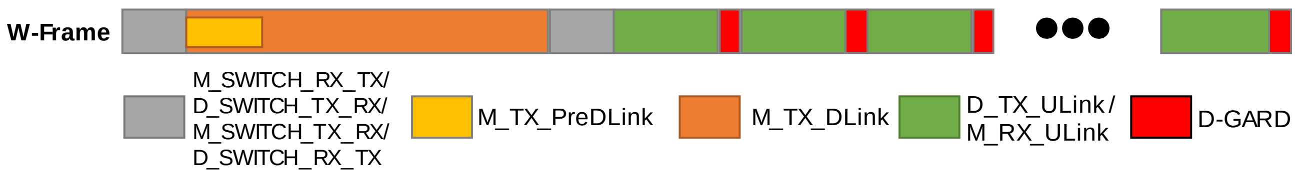


Figure 34 W-Frame structure

5.4.4 Configuration W-Frame

This allows using the entire Downlink space for the transfer of ConnectionParameters. There is no Pre-Downlink part encoded in this frame type. Consequently, only the addressed W-Device returns a message within the Uplink section (see Figure 35).

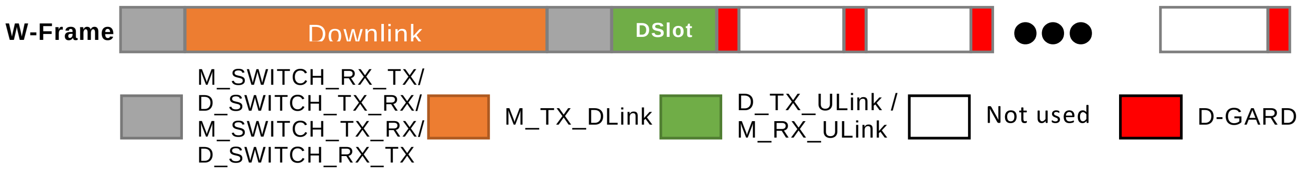


Figure 35 W-Frame type for pairing and configuration

5.5 Medium Access Control (MAC)

5.5.1 General

This clause describes the relevant definitions for media access on both, W-Master and W-Devices, which shall 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 lose 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 Blocklisting. During the pairing and configuration processes, the W-Master downloads these hopping sequences into the unpaired W-Devices.

5.5.2 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 for cyclic data exchange (see 4.5.2.4) and n = 1 and 80 for configuration (see 4.5.2.3).

The minimum spectral distance between the W-Tracks of a W-Master is 3 MHz. Default Hopping Table is according to HT01.
IO-Link Wireless defines the frequency channel hopping table HT01. HT01 omits the frequency channels $f_{1\text{-}2}$ and $f_{79\text{-}80}$. The frequency channels f_1 and f_{80} are reserved for configuration (see 5.5.4). The frequency channels f2 and f79 are always blocklisted (see Table 209). 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 W-Track of a W-Master and its W-Devices is listed. HT01 additional allows blocklisting of each 1 MHz frequency channel (see 5.5.5).

The frequency hopping sequence of all W-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 W-Track of an IO-Link W-Master shall keep the hopping tables of all other W-Tracks of this W-Master, which enables a (e.g., roaming) W-Track to pair a W-Device to another target W-Track of this W-Master. Therefore, the target W-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 7.

Table 7 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 H.2
MasterID	MasterID: The ID the W-Master is assigned to	
Blocklist	List of unused frequency channels	An 80-bit word each bit representing a frequency channel. See clause 5.5.5.
Number_Of_Tracks	Number of W-Tracks of a W-Master	Ensures that all W-Tracks of the W-Master have non-overlapping frequency tables
Frequency Spacing	Minimal frequency distance between different W-Tracks of a W-Master	Ensures the interference between the W-Tracks within a W-Master are minimal. For this purpose, the minimal frequency spacing between the W-Tracks of a W-Master shall be greater or equal to 3 MHz.

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

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

See clause H.2 for calculation rules and examples.

5.5.3 Alternative Hopping algorithms

Hopping algorithms other than for calculation of HT01 are not allowed. Other hopping algorithms are reserved for future use.

5.5.4 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 blocklisted. Clause J.3 describes their utilization in detail.

5.5.5 Blocklisting

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 blocklist (see Table 210). It should be taken into account, that the reduction of available frequency channels can compromise latency in a non-deterministic manner. See clause H.2 for calculation rules and examples.

5.5.6 Wireless Quality

5.5.6.1 Link Quality Indication (LQI_M, LQI_D)

Link Quality Indication is a service for evaluation of the functionality and reliability of the IO-Link Wireless System in certain application environments. The LQI shall be calculated on the W-Master (LQI_M) as well as on the W-Device (LQI_D) Physical Layer. 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 Packet Error Ratio (PER) per sub cycle (PERSubCycle) and MaxRetry are used to calculate the Link Quality Indication with
Equation 3

$$LQI = Min\left(100\ , a_{MaxRetry} * \log_{10} PERSubCycle\right)$$

Equation 3 Calculation of Link Quality Indication

Where the LQI is interpreted in percent and the slope is calculated using MaxRetry to adjust the result to a value of 70% for a system RFP of 10^{-9} with Equation 4

$$a_{MaxRetry} = -\frac{70}{9} * (MaxRetry + 1)$$

Equation 4 Calculation of slope using MaxRetry

Where Packet Error Ratio is calculated every 2^{12} w-sub-cycles with Equation 5

$$PERSubCycle = \frac{SubCycleErrorCounter}{SubCycleCounter}$$

Equation 5 Calculation of PER Sub cycle

NOTE: If SubCycleCounter is less than 2^12 then the LQI returns 0xFF (INVALID).

The LQI vs PERSubCycle behavior for different MaxRetry is shown in Figure 36.

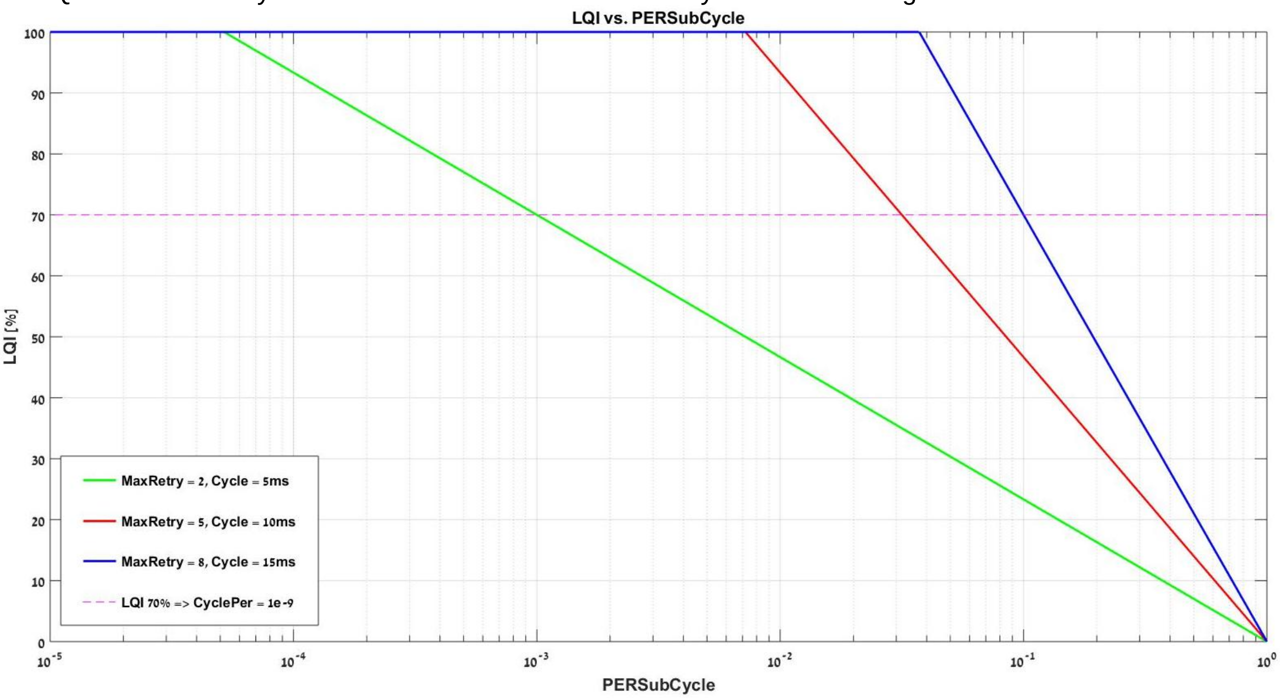


Figure 36 LQI vs PERSubCycle for different MaxRetry

NOTE1: In general, the Link Quality Indication is only an estimate of the expected reliability, rather than a precise measurement.
In certain situations, there may be a discrepancy between the reliability estimated by the Link Quality Indication and the reliability actually achieved.
For example, for a frequency-selective but almost static radio channel between the W-Master and the W-Device (such as multipath propagation due to many reflections and almost no moving objects in the environment), the Link Quality Indication may lead to a worse estimate of reliability than can actually be achieved.

5.5.6.2 Received Signal Strength Indication (RSSI_M, RSSI_D)

The RSSI is an estimated measure of power level that a W-Master (RSSI_M) or W-Device (RSSI_D) is receiving from its counterpart. It is interpreted as an indication of arriving signal power strength at the antenna. The Values of the RSSI can be used by application as a user information.
NOTE that not all radio transceivers supports the RSSI. See clause C.4.8 for definition.

5.6 Physical Layer PL services

5.6.1 Overview

An overview of the Physical Layer and its service primitives is given in Table 8 and Figure 95. They are the interface to the higher protocol layers.

5.6.2 PL Services for W-Master

5.6.2.1 General

Subclause 5.6.2 specifies the PL Services for a W-Master that the PH provides to the System Management and to the Data Link Layer (see Figure 99 for a complete overview of all the services). Table 8 lists the assignments of W-Master to their roles as initiator or receiver for the individual PL services.

Table 8 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 a service	

5.6.2.2 PL_SetTrackConfig (W-Master)

The PL_SetTrackConfig service is used to setup the initial parameter for each W-Track on a W-Master. The parameters of the service primitives are listed in Table 9.

Table 9 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 W-Track.

Parameter Type: Record

Record Elements:

MasterID: This parameter contains the MasterID of the W-Master (see Table 154)

Permitted values: 1-29

BlockList: This parameter contains the frequency channels which shall not be used by the W-Master. For details see H.1.

Track_N: This parameter set up the internal number of a W-Track for calculation of track-dependent hopping sequence.

Each W-Track shall be numbered consecutively within a W-Master. Permitted values: 0-4

SyncTrack: This parameter defines, whether the W-Track is generating or receiving a W-Frame hardware synchronization signal. The synchronization signal is transferred via a hardware pin to Tracks configured as SyncTrack (NO).

Permitted values:

YES (The W-Track is generating a synchronization signal).

NO (To start its W-Sub-cycle the track is waiting for the hardware synchronization signal, which is generated by the track configured as SyncTrack(YES))

DataSyncword

DataSyncword which shall be used for data channels (see clause 5.3.4)

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)

5.6.2.3 PL_SetMode (W-Master)

The PL_SetMode service is used to setup the mode of a W-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 10.

Table 10 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 11)
Permitted values: STOP, CYCLIC, SCAN, ROAMING

TxPower:
This parameter indicates the transmission power level of the W-Track.
Permitted values: 1 to 255 (See 10.10 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 11 specifies the coding of the different parameters.

Table 11 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)

5.6.2.4 PL_Scan (W-Master)

The PL_Scan service is used to report a new unpaired W-Device within the W-Track's proximity via indication. This is only initiated by PL if the W-Track is in ROAMING or SCAN mode. The parameters of the service primitives are listed in Table 12.

Table 12 PL_Scan

Parameter Name	.ind
Argument	M
ParameterList	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: Default type of the W-Device in Uplink
Permitted values: SSLOT, DSLOT (see Table 175)
UniqueID: This parameter indicates the UniqueID of the W-Device. (see Figure 140)
RevisionID: This parameter indicates the protocol version of the found W-Device. (see Figure B.4 in [1])

5.6.2.5 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 13.

Table 13 PL_ScanEnd

Parameter Name	.ind
<none>	

5.6.2.6 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 14.

Table 14 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 140)
SlotType: Default type of the W-Device in Uplink given through W-Device application.
Permitted values: SSLOT, DSLOT (see Table 175)
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 C.4.4.2) for the corresponding slot/W-Device (see 10.3 to detect COMLOST).
MaxRetry: Permitted values: Table 184

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)

5.6.2.7 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 15.

Table 15 PL_Pairing

Parameter Name	.req	.cnf	.ind
Argument	M		M
ParameterList	M		
Info			M
PairedUniqueID			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 C.4.4.1)

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

SlotType: This parameter contains the slot type for the corresponding W-Device.
Permitted values: SSLOT, DSLOT (see Table 175)

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 128 to pair a W-Device via button method)
PAIRING_UNIQUE (PL shall use the W-Frame Figure 128 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 213 (definition of PAIRING_BUTTON_TIMEOUT, PAIRING_UNIQUE_TIMEOUT)

PairedUniqueID
This parameter contains the currently paired UniqueID (see C.4.7)

Info
Permitted values:
PAIRING_SUCCESS (W-Device has been paired)
PAIRING_TIMEOUT (W-Device was not paired within the time given in Timeout)

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)

5.6.2.8 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 16.

Table 16 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 (W-Device has no or lost connection to its W-Master)
1: SYNCED (W-Device is synchronized with its W-Master)

5.6.2.9 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 B.6). The parameters of the service primitives are listed in Table 17

Table 17 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 part
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 W-Master to W-Device)
PL_Transfer.ind: 2 octets (data from W-Device to W-Master, SSlot-Format)
PL_Transfer.ind: 15 octets (data from W-Device to W-Master, DSlot-Format)

ULinkType:
This parameter contains the type of ULink. Permitted values:

DATA (regular ULink received, see B.4. Regular Uplink Frame Annex B).
NOUPLINK (No ULink received)
IMA (IMA ULink received, see Figure 133 and Figure 134 and IMA-Uplink packet Annex B).

Slot_N:
This parameter contains the Slot_N to assign the received ULink to the corresponding W-Port (see 6.2.2 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)

5.6.2.10 PL_QualityService (W-Master)

The PL_QualityService is used to request the current quality of the wireless connection between the W-Master W-Track and the corresponding W-Device (see clause 5.5.6.1). The parameters of the service are listed in Table 18.

Table 18 PL_QualityService

Parameter Name	.req	.cnf
Argument	M	
Slot_N	M	
Result (+)		S
LQI_M		M
RSSI_M		M
LQI_D		M
RSSI_D		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.

LQI_M
Permitted Values: 0 to 100%, INVALID see C.4.8.

RSSI_M
Permitted Values: -128 to 20, INVALID see C.4.8.

LQI_D
This parameter contains the link quality of the W-Device, delivered via the IMA-Uplink packet.
Permitted Values: 0 to 100%, INVALID see C.4.8.

RSSI_D
This parameter contains the RSSI of the W-Device, delivered via the IMA-Uplink packet.

Permitted Values: -128 to 20, INVALID see C.4.8.

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)

5.6.2.11 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 9.3.3.2.4. The parameters of the service primitives are listed in Table 19.

Table 19 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 W-Track number of which to read the hopping table values. Used to read all the hopping table values from any W-Track, see Annex D.

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 B.3.4).

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)

5.6.2.12 PL_SetHopTable (W-Master)

The PL_SetHopTable service is used to set the new hopping table in W-Master PL W-Track. The parameters of the service primitives are listed in Table 20.

Table 20 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 W-Track number of which to set the hopping table values. Used to set the hopping table values for any W-Track, see clause 5.5.2.

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)

5.6.2.13 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 21.

Table 21 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)

5.6.2.14 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 22.

Table 22 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 octets).
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)

5.6.2.15 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 23.

Table 23 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)

5.6.2.16 PL_CmdTrig (W-Master)

The PL_CmdTrig service is used to communicate real time actions to the PL from the command handler, see H.4. The parameters of the service primitives are listed in Table 24.

Table 24 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, 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 corresponding 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)

5.6.3 PL Services for W-Device

5.6.3.1 Overview (W-Device)

Subclause 5.6.3 specifies the PL Services for a W-Device that the PL to System Management and to the Data Link Layer (see Figure 95 for a complete overview of all the services). Table 25 lists the assignments of W-Device to their roles as initiator or receiver for the individual PL services.

Table 25 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 (see 3.3.5) I Initiator of service R Receiver (Responder) of a service	

5.6.3.2 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 26.

Table 26 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 27)

Permitted values: STOP, STOP_KEEP, START

UniqueID: This parameter contains the UniqueID of the W-Device (see Figure 140).

SlotType: Default type of the W-Device in Uplink given through W-Device application.

Permitted values: SSLOT, DSLOT (see Table 175)

DownlinkType: Type of the W-Device in Downlink given through W-Device application.

Permitted values: PRE_DOWNLINK, FULL_DOWNLINK (see Table 27)

TxPower: Permitted values: 1 to 31 (see Table 186)

MaxRetry: Permitted values: Table 185.

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 27 specifies the coding of the different Parameters

Table 27 PL_SetMode coding of Parameters

Parameters	Definition
STOP	Communication disabled; radio turned off
STOP_KEEP	Communication disabled; radio turned off. Keep Connection Parameter.
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 part (reduced receive-on time for low energy W-Devices) only when connected
FULL_DOWNLINK	W-Device is listening for a Full-Downlink packet when connected

5.6.3.3 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 28.

Table 28 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 213.
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)

5.6.3.4 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 29.

Table 29 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

5.6.3.5 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 B.6). The parameters of the service primitives are listed in Table 30.

Table 30 PL_Transfer (W-Device)

Parameter Name	.req	.ind
Argument	C	M
Data	M	M
DataLength	M	M
Acknowledge		M
SlotType		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.

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 packet.

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.

SlotType
The current slot type, used to compute the uplink packet size

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)

5.6.3.6 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 (calculation see 5.5.6) and the RSSI of the W-Device. The parameters of the service are listed in Table 31

Table 31 PL_QualityService (W-Device)

Parameter Name	.req	.cnf
Argument	C	
<none>		
Result (+)		S
LQI_D		M
RSSI_D		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.
LQI_D
Permitted Values: 0 to 100%, INVALID see C.4.8.
RSSI_D
Permitted Values: -128 to 20, INVALID see C.4.8.
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)

5.6.3.7 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 32.

Table 32 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)

5.6.3.8 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 33.

Table 33 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)

5.6.3.9 PL_CmdTrig (W-Device)

The PL_CmdTrig service is used to communicate real time actions in PL triggered by Command handler. PL_CmdTrig delivers the WakeUpTime value. The parameters of the service primitives are listed in Table 34.

Table 34 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, 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)

5.7 Physical Layer PL protocol

5.7.1 Usage of the Configuration Channel

5.7.1.1 General

The Configuration channel is available only when one W-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.

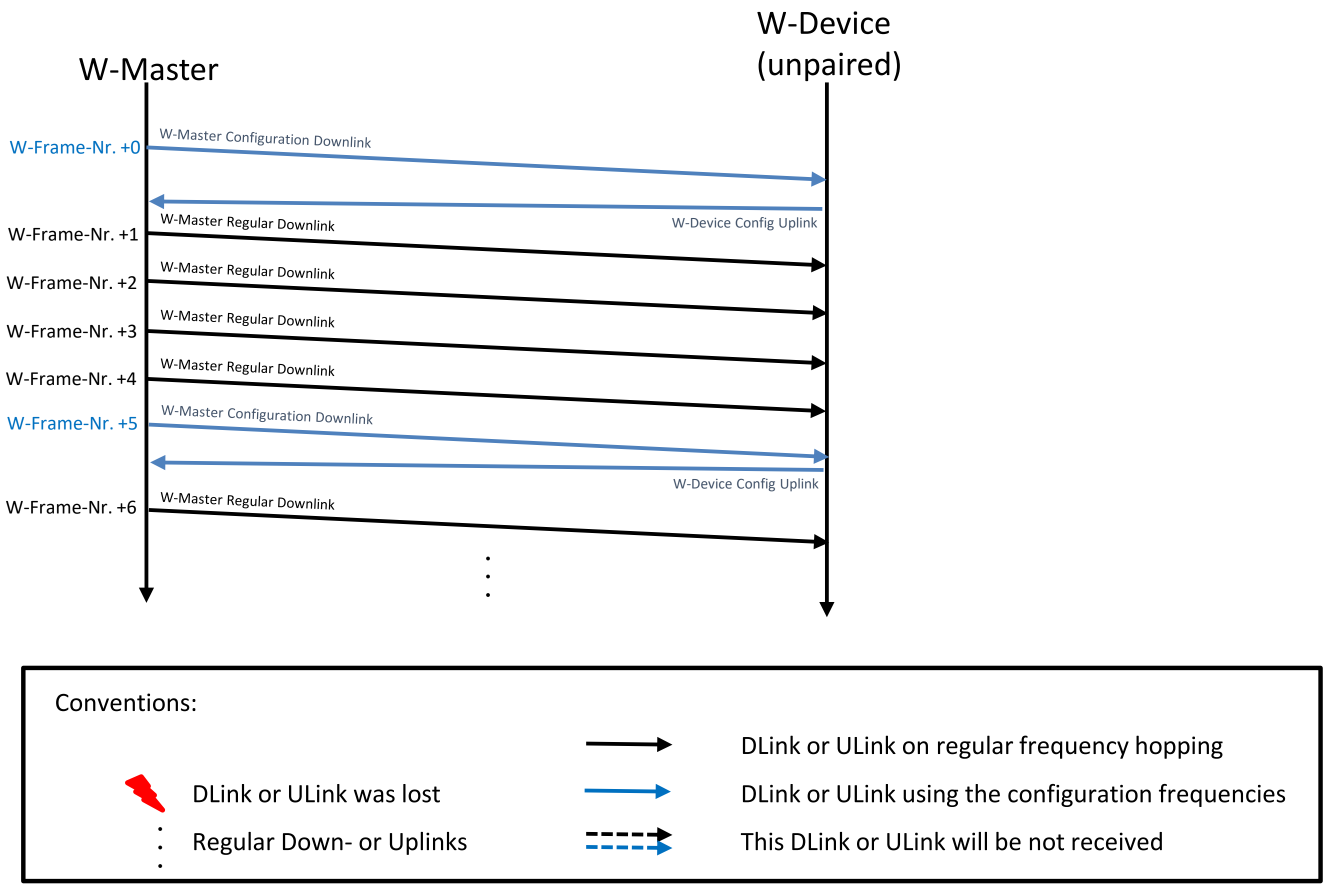


Figure 37 Usage of the Configuration Channels

5.7.1.2 Configuration sequence and retry handling for Scan.

Figure 38 describes the sequence for a discovery procedure. The W-Master sends a Scan Request Downlink (see Figure 121) on each configuration W-Sub-cycle with a continuously incremented request number (Request_N).
If an unpaired W-Device receives the Scan Request, it shall respond with a Scan Response Uplink packet (see Figure 135) on W-Sub-cycle X. X is calculated with Equation 6.

$$X = \mathrm{Request\_N} + \mathrm{Frame\_N}$$

Equation 6 Calculation of Scan Response W-Sub-cycles

Frame_N is the number of configuration W-Sub-cycles to wait between the first received Scan request and the transmission of the Scan Response Uplink packet. Frame_N is calculated with Equation 7.

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

Equation 7 Frame_N 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 8.

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

Equation 8 Slot_N number calculation using the UniqueID.

The W-Master acknowledges the reception of Scan Response by setting the corresponding ack-bit in the next Scan Request. In order to avoid ambiguity, the next Scan Request includes also the received UniqueID of the W-Device (see Figure 128). If a W-Device does not receive an ACK for its scan response it shall not use the next configuration W-Sub-cycle for retry transmission because this could cause further packet collisions. The retry shall be generated after a random number (in the range of 2…31) of configuration W-Sub-cycles instead.

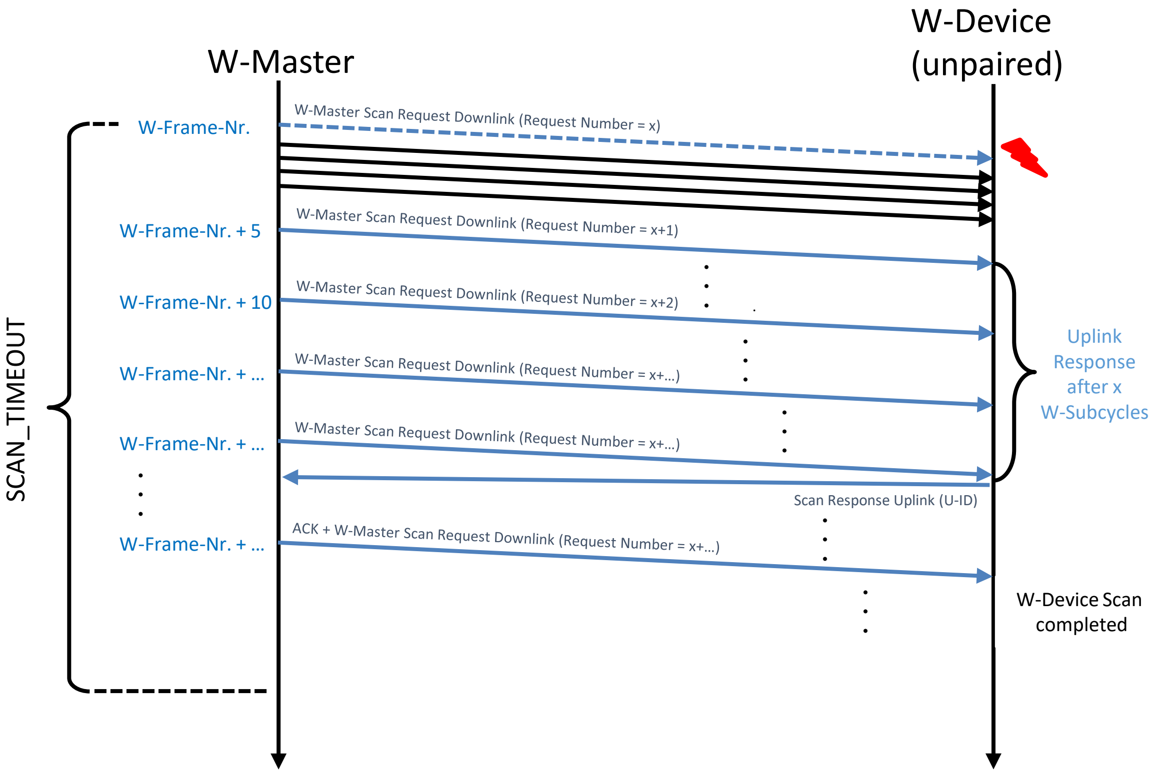


Figure 38 Configuration sequence for Scan

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

5.7.1.3 Configuration sequence and retry handling for pairing by UniqueID.

Figure 39 describes the sequence for pairing by UniqueID. W-Master sends ConnectionParameter via

- Pairing Request Downlink (Roaming Flag = 0), see Figure 128
- Negotiation 1 Request Downlink, see Figure 129.
- Negotiation 2 Request Downlink, see Figure 130.

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 packet, see Figure 136
- Negotiation Response Uplink packets 1 and 2, see Pairing Negotiation Uplink packet Figure 137

If a pairing DLink or a ULink was lost, the Data shall be retransmitted in the next configuration W-Sub-cycle. The maximal number of all retransmissions within a service request is given by the timeout of the corresponding service (see Table 213).

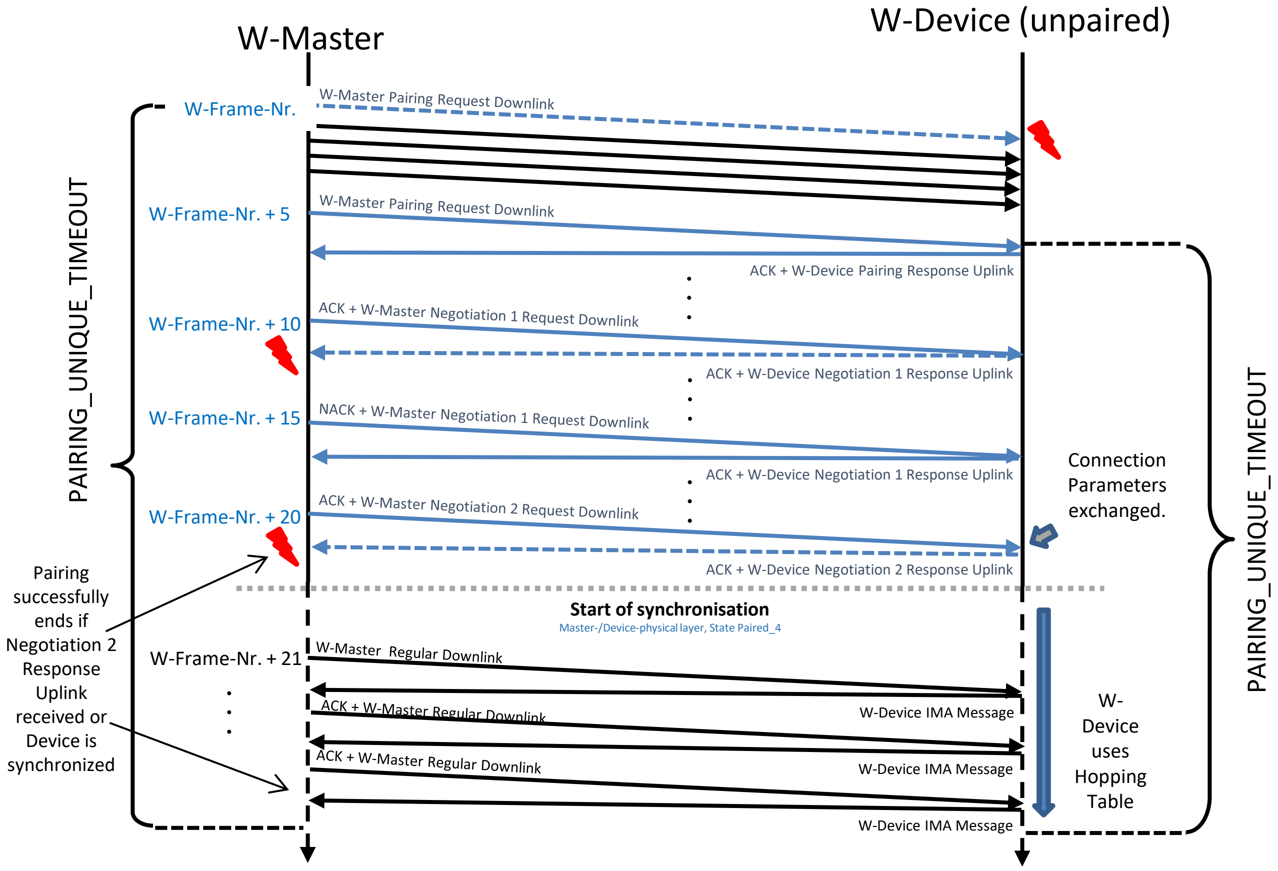


Figure 39 Configuration sequence for pairing by UniqueID

5.7.1.4 Configuration sequence for pairing by Button.

Figure 40 describes the sequence for pairing by Button. If pairing by Button is active the W-Master sends ConnectionParameter via:

- Pairing Request Downlink, see Figure 128.
- Negotiation 1 Request Downlink, see Figure 129.
- Negotiation 2 Request Downlink, see Figure 130.

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 packet, see Figure 136,
- Negotiation Response Uplink packets 1 and 2, see Pairing Negotiation Uplink packet Figure 137
- If a pairing DLink or a ULink was lost, the Data shall be retransmitted in the next configuration W-Sub-cycle. The maximal number of all retransmissions within a service request is given by the timeout of the corresponding service (see Table 213).

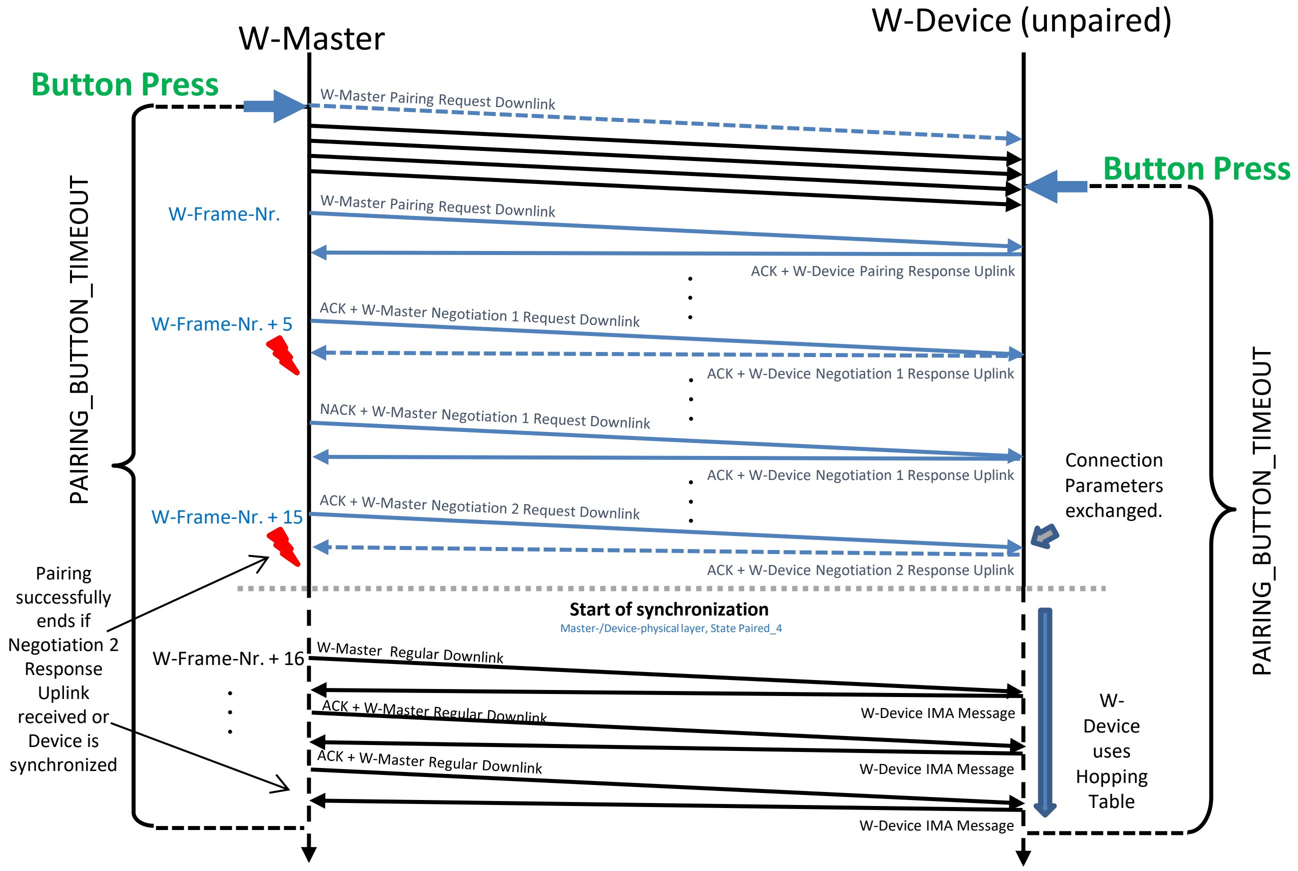


Figure 40 Configuration sequence for pairing by Button

5.7.1.5 Message Sequence Chart for Roaming

Figure 41 describes the “Handover Connect” sequence for a temporary connection in Roaming Mode. A W-Master W-Track in Roaming Mode shall regularly scan for unpaired W-Devices (see 5.7.1.2 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 5.7.1.3., Configuration sequence for pairing by UniqueID) with Roaming Flag = 1 (Pairing Request Downlink, see Figure 128).

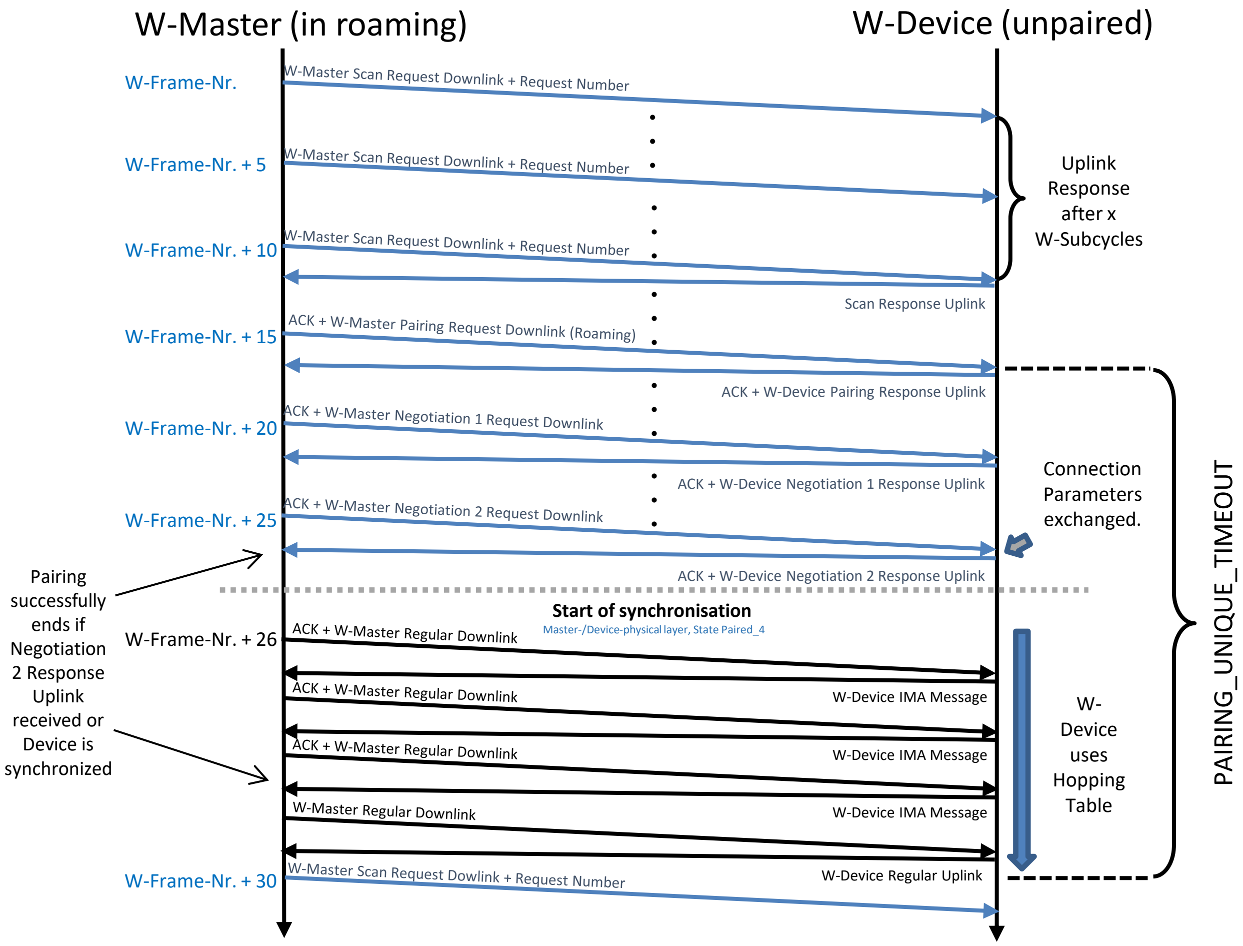


Figure 41 Message Sequence Chart for Roaming

5.7.2 PL W-Master state machine

5.7.2.1 General

Figure 42 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 43) 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.

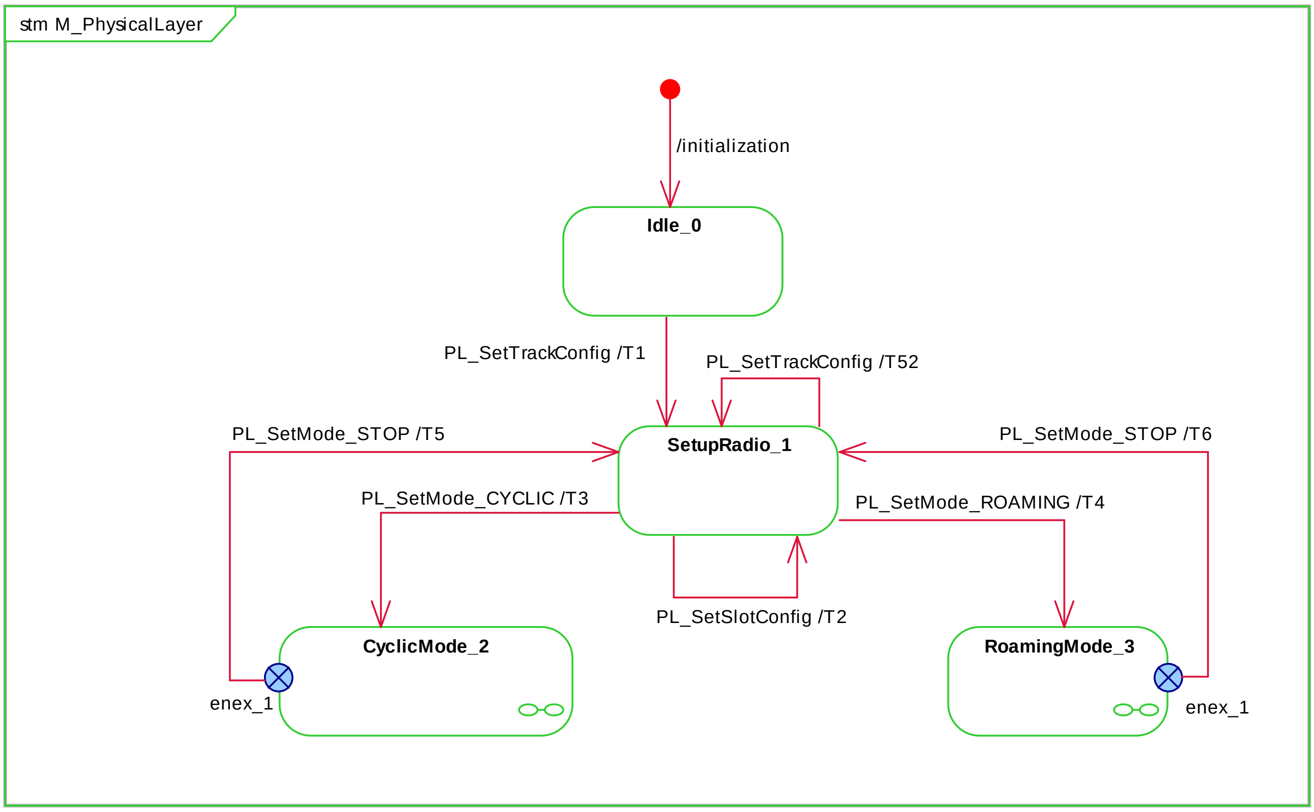
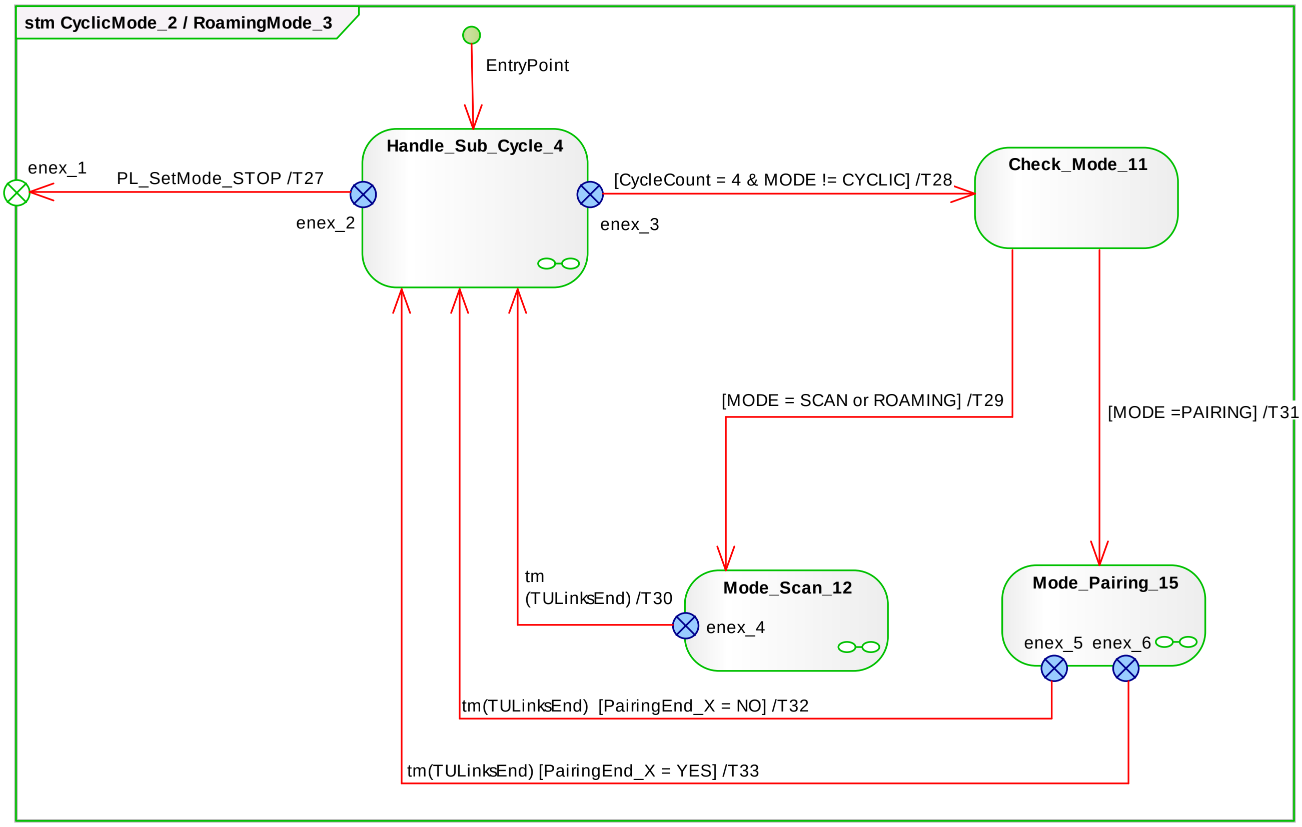


Figure 42 PL W-Master state machine

1954



1955 **Figure 43 Submachine of CyclicMode_2 or RoamingMode_3 of W-Master Physical Layer**

1956

5.7.2.2 Submachine of Handle_Sub_Cycle_4 of W-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 uplink packets and handling of the IMA (I'm Alive) timeouts for all configured W-Devices. This state machine triggers the Scan if the activation is requested by the PL_SetMode.req service or in case the W-Master W-Track is set in to roaming mode. It triggers also pairing if the Pairing.req service invoked.

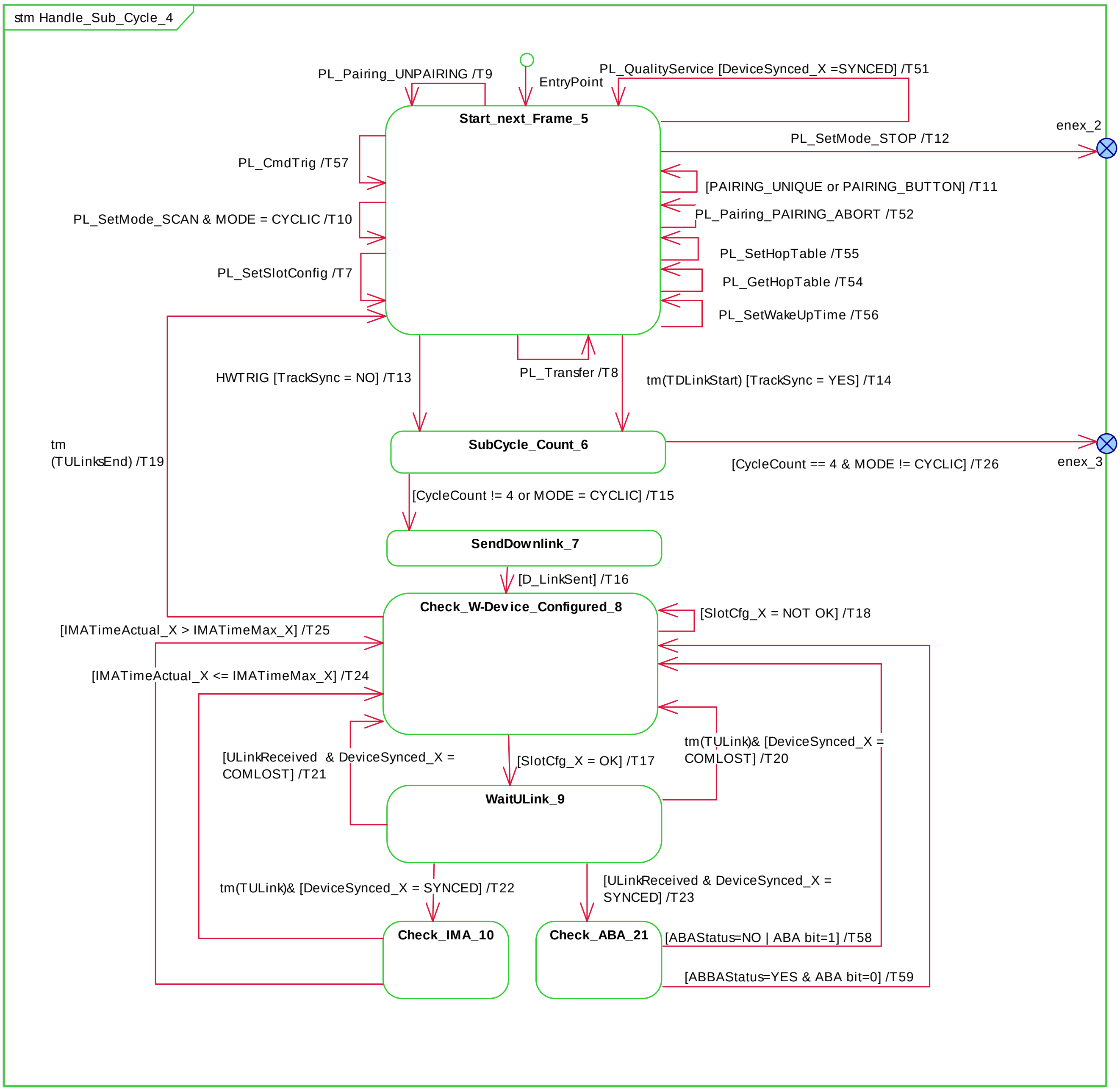


Figure 44 Submachine of Handle_Sub_Cycle_4 of W-Master Physical Layer

5.7.2.3 Submachine for Mode_Scan_12

The Mode_Scan_12 (Figure 45) sub state machine handles transmission of the scan request telegrams to all not configured W-Devices and collects the scan responses.

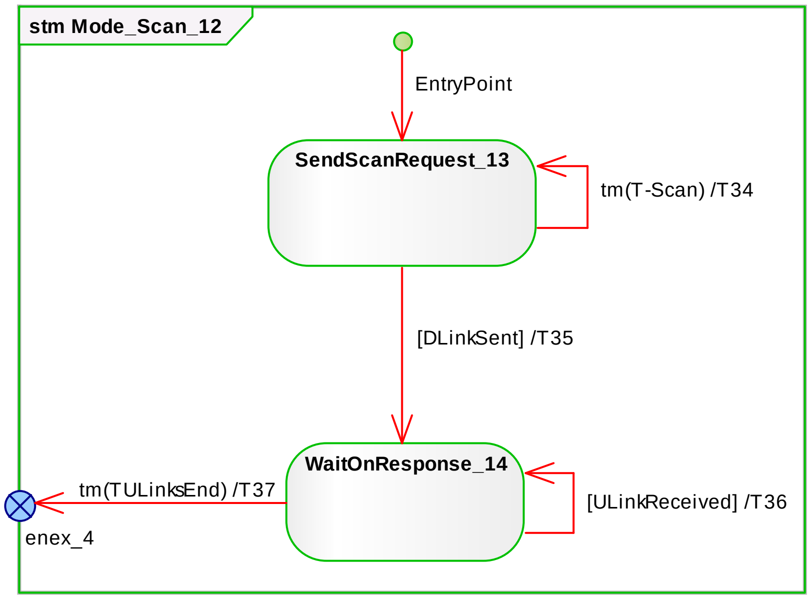


Figure 45 Submachine for Mode_Scan_12

5.7.2.4 Submachine for Mode_Pairing_15

The Mode_Pairing_15 (Figure 46) 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.

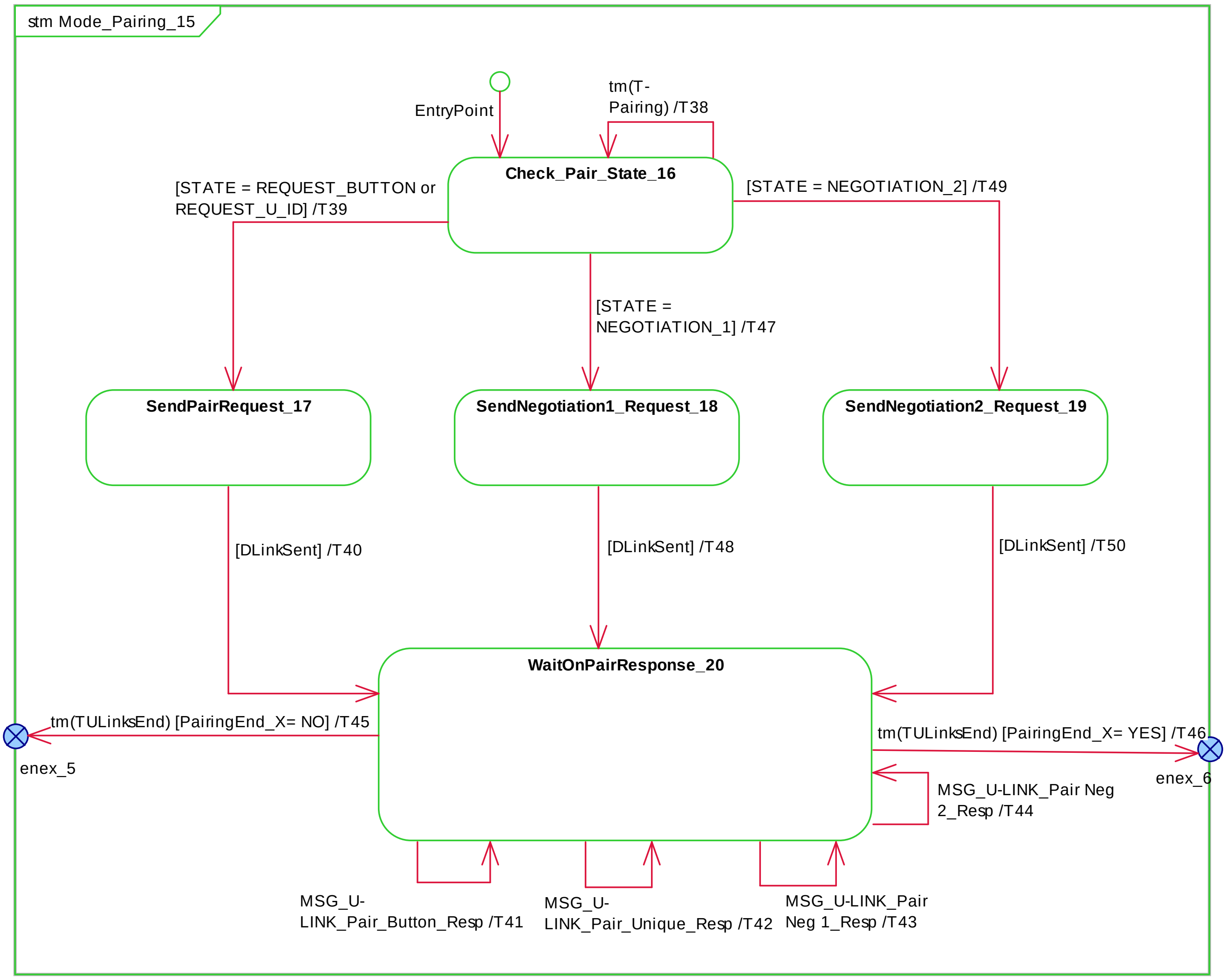


Figure 46 Submachine for Mode_Pairing_15

Table 35 shows the state transition tables of the W-Master Physical Layer.

Table 35 State transition tables of the W-Master Physical Layer

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting for activation by SM_SetTrackConfig via PL_SetTrackConfig service.
SetupRadio_1	Initialization and setup of the radio transceiver (W-Track) for radio operation as specified in 5.2to 5.5 and H.2 Annex F (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 32 and Figure 33). SlotType: indicates the length of ULink (see Figure 131 and Figure 132). IMATime: contains the I am alive time (see clause C.4.4.2) 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 W-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 126). 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 transmission until the Uplink packet has been received or the Timer T_{TUlink} exceeds (See Figure 131 and Figure 132). NOTE: The CRC 32 calculation of a cyclic ULink considers the DeviceDistinguishingID of this slot (_X) (see B.7 Pre-set value for cyclic ULink).
SM: Check_IMA_10	This state handles the IMA supervision for the actual Slot, since the W-Device is synchronized but Uplink packet 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 B.3.2 Scan Downlink)
SM: WaitOnResponse_14	This state handles the reception of the Scan response Uplinks (up to four Scan responses are possible, see B.5.2. Scan Response Uplink packet)
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 B.3.3. Pairing Request Downlink). If STATE = REQUEST_BUTTON, use „Pairing by Button“ DLink, see Figure 128. Pairing by Button If STATE = REQUEST_U-ID, use „Pairing by UniqueID“ DLink, see Figure 128.
SM: SendNegotiation1_Request_18	This state handles the transmission of the Pairing Negotiation 1 Downlink (see Figure 129. Negotiation 1 Downlink)
SM: SendNegotiation2_Request_19	This state handles the transmission of the Pairing Negotiation 2 Downlink (See Figure 130. Negotiation 2 Downlink)
SM: WaitOnPairResponse_20	This state handles the reception of the Pairing response Uplink packet.
SM: Check_ABA_21	This state handles the ABA supervision for the actual Slot, checks the ABAStatus and ABA bit.

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, BlockList and Track_N (see H.2.: Creation of frequency hopping table HT01 with care to blocklisting). Setting the internal variable TrackSync = YES or NO (see 5.6.2.2 PL_SetTrackConfig)
T2	1	1	*Activation by System Management through PL_SetSlotConfig(ParameterList).* PL_SetSlotConfig prepares the corresponding Slot "_X" given in slot number (Slot_N) for a proper connection in the following way: Slot_N: Points to the receive time within the TDMA slot assignment (see Figure 32 and Figure 33) UniqueID: The last 4 octets of the UniqueID (Device Distinguishing ID) is used as final XOR of the CRC32 checksum (see B.7.: Final XOR of a regular ULink) SlotType: Defines the length of the ULink (see Figure 30 and Figure 31) to setup the transceiver receive length. IMATime: Defines the number of W-Sub-cycles to observe the presence of the W-Device (see clause C.4.4.2 for encoding). Set SlotCfg_X = OK. *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).
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.ind(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.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
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 W-Track is configured as SyncTrack = NO.* Start next W-Sub-cycle on rising edge of external trigger HWTRIG from the W-Master W-Track. If WakeUpCountdown higher than 0, decrement by 1.
T14	5	6	*The W-Track is configured as SyncTrack = YES.* 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 126 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 B.7 Final XOR of a regular ULink. *For additional information about timing see Figure 32: 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 is 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 = YES set the hardware trigger HWTRIG (output) to LOW.
T20	9	8	Increment ULink Slot (_X)
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 W-Track mode)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
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	21	*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 B.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 an IMA-Frame (see Figure 133 and Figure 134. IMA-Uplink packet 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. An IMA Timeout error occurred.* Set DeviceSynced_X = COMLOST. Set ABAStatus = NO Report all W-Device states through an invoke of PL_State.ind(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 B.3.2. 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 127 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 = YES set the hardware trigger HWTRIG (output) to LOW. Set Mode = CYCLIC or ROAMING, depending on initial W-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
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 packet has been received.* Set Device_ACK_Service_X =1. Invoke PL_Scan.ind(SlotType, UniqueID, Protocol VersionRevisionID). See 5.6.2.4. PL_Scan (W-Master).
T37	14	4	See T19.
T38	16	16	*Timer TPairing expired.* Invoke PL_Pairing.ind(PAIRING_TIMEOUT). Set PairingEnd_X = YES;

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
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 128. If STATE = REQUEST_U_ID, use pairing request downlink “Pairing Request by UniqueID” see Figure 128. Update the ACK field in the radio output buffer with Device_ACK_Service (see Figure 128 Pairing Request by Button or Pairing Request by UniqueID).
T40	17	20	See T35.
T41	20	20	*Radio received MSG_UPLINK_Pair_Button_Resp (see* Table 174 *Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_1.
T42	20	20	*Radio received MSG_UPLINK_Pair_Unique_Resp (see* Table 174*. Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_1.
T43	20	20	*Radio received MSG_UPLINK_Pair Neg 1_Resp (see* Table 174 *Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Set STATE = NEGOTIATION_2.
T44	20	20	*Radio received MSG_UPLINK_Pair Neg 2_Resp (see* Table 174 *Uplink-MSG-Types).* Set Device_ACK_Service_X =1. Invoke PL_Pairing.ind(PAIRING_SUCCESS). Set PairingEnd_X = YES;
T45	20	4	See T19.
T46	20	4	See T33.
T47	16	18	Load the Negotiation_1 Downlink in the radio output buffer and send the Downlink (see B.3.4 Pairing Negotiation Downlink). Update the ACK field in radio output buffer with Device_ACK_Service_X (see B.3.4 Pairing Negotiation Downlink)
T48	18	20	See T35.
T49	16	19	Load the Negotiation_2 Downlink in the radio output buffer and send Downlink (see B.3.4. Pairing Negotiation Downlink). Update the ACK field in radio output buffer with Device_ACK_Service_X (see B.3.4 Pairing Negotiation Downlink)
T50	19	20	See T35.
T51	5	5	-
T52	1	1	Calculate the frequency hopping table dependent on the parameters MasterID, BlockList and Track_N (see H.2.: Creation of frequency hopping table HT01 with care to blocklisting). Setting the internal variable TrackSync = YES or NO (see 5.6.2.2 PL_SetTrackConfig)
T53	5	5	Stop Timer TPairing. Set PairingEnd_X = YES;
T54	5	5	-
T55	5	5	*Activation by System Management through PL_SetHopTable.* Load the new hopping table.
T56	5	5	*Activation by System Management through PL_SetWakeUpTime(WakeUpTime).* Set internal variable WakeUpCountdown = WakeUpTime.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T57	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, 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)
T58	21	8	If the W-Device ABA bit is on, Set ABAStatus_X = YES
T59	21	8	*Received ABA indication. W-Device had* COMLOST. Set DeviceSynced_X = COMLOST. Set ABAStatus = NO. Report all W-Device states through an invoke of PL_State.ind(DeviceSynced).

1983

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 W-Track is generating or receiving a W-Frame hardware synchronization signal. Permitted values: YES or NO (see 5.6.2.2 PL_SetTrackConfig).
ULinkReceived	Bool	Flag which shall be set by the radio hardware if an Uplink packet 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 5.6.2.8 PL_State.

INTERNAL ITEMS	TYPE	DEFINITION
		NOTE: A change of DeviceSynced_X leads to PL_State.ind(DeviceSynced) where the corresponding flag X is set to SYNCED or COMLOST.
DeviceSynced	Variable	Variable which contains SYNCED or COMLOST of each Slot, coded via flag DeviceSynced_X.
ABAStatus_X	Bool	Flag which indicates the ABA bit status for the corresponding slot Permitted values: YES, NO.
IMATimeActual_X	Variable	Variable to count the number of W-Sub-cycles, if a W-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 C.4.4.2 for encoding), delivered via the service PL_SetSlotConfig, see 5.6.2.6
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 46. 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 5.6.2.7. PL_Pairing-Service and 5.7.1.3 Retry handling during Pairing Mode.
PAIRING_UNIQUE_ TIMEOUT	Constant	See Table 213, see 5.7.1.3 Retry handling during Pairing Mode.
SCAN_TIMEOUT	Constant	See Table 213, see 5.7.1.3 Retry 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

1984

1985 NOTE: X marks the variables which are individual in each Slot_N. The range of _X is slot number 0 to 7.

5.7.3 PL W-Device state machine

5.7.3.1 General

Figure 47 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.

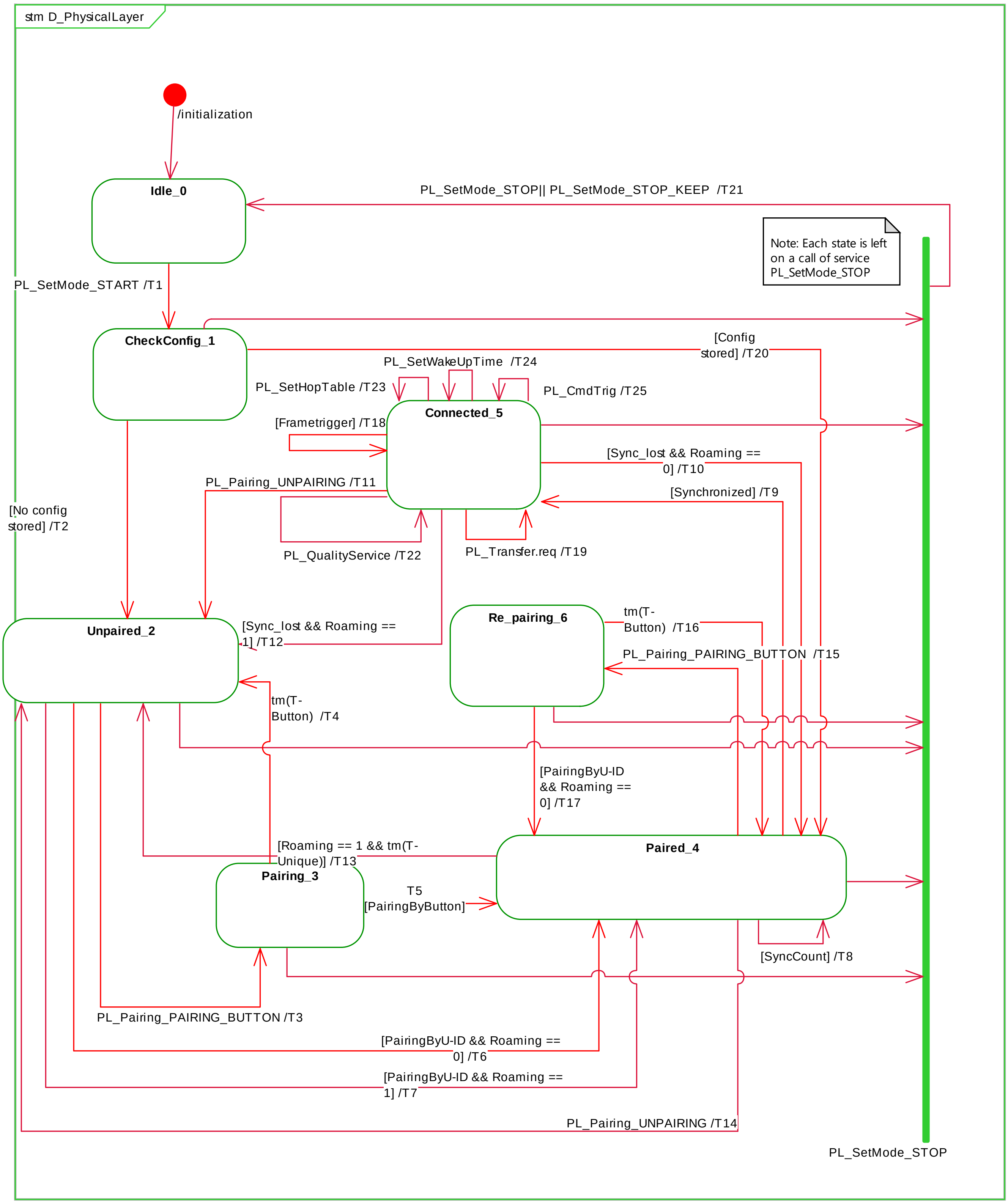


Figure 47 PL W-Device state machine

Table 36 shows the state transition tables of the W-Device Physical Layer.

Table 36 State transition tables of the W-Device Physical Layer (normal W-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 154 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 5.5.4) 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 128) 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 synchronization cycle. See also G.1.6 W-Master not reachable.
Connected_5	The W-Device is connected to its paired W-Master via regular W-communication cycles (see Figure 126)
Re_pairing_6	Waiting for configuration-channels for Scan Request or Pairing Request by UniqueID (via W-Master).

Figure 40Figure 40	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	*Activation by System Management through PL_SetMode.req(Start). (see Table 125 Transition T1).*
T2	1	2	The Device Radio has no valid ConnectionParameter settings stored (see Table 154) 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 213) if timer ($T_{T\text{-}Button}$) expired.
5.3.45.3.4	3	4	*Pairing by Button sequence was successfully executed.* *(See Figure 40 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 5.3.4). Stop timer ($T_{T\text{-}Button}$).
T6	2	4	*Pairing by UniqueID sequence was successfully executed.* *(See Figure 39 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 175).
T7	2	4	*Pairing by UniqueID sequence was successfully executed.* *(See Figure 41 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 175).
T8	4	4	Send IMA ULink to the W-Master on each received DLink (e.g., see Figure 39 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. Set ABA bit to 0. Invoke PL_State.ind(SYNCED) service indication to report that the connection is established. Stop timer ($T_{T\text{-}Unique}$). Stop timer ($T_{T\text{-}Button}$).

Figure 40Figure 40	SOURCE STATE	TARGET STATE	ACTION /Remarks
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. Set ABA bit to 1.
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 (see Table 154). 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 154) Set SyncCounter to 0. Set Roaming to 0.
T13	4	2	Delete non-volatile ConnectionParameter settings (see Table 154). 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 5.7.3.2.* Set SyncLostCounter to 0. Case 2: *DLink received without data for this W-Device (DataLength=0, see clause 5.7.3.2).* 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.

Figure 40Figure 40	SOURCE STATE	TARGET STATE	ACTION /Remarks
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 5.7.3.2).*
T20	1	4	*The Radio has stored a valid ConnectionParameter settings (see* Table 154*).* Invoke PL_State.ind(PAIRED) Set RoamingKeepCfg to 0
T21	Any	0	*Any state shall be left through a call of PL_SetMode(STOP) or PL_SetMode(STOP_KEEP)* *Service via System Management.* If Roaming == 1 and PL_SetMode(STOP_KEEP) is called then set RoamingKeepCfg to 1 and Set tm(T-Unique)
T22	5	5	Invoke PL_QualityService.cnf
T23	5	5	*Activation by System Management through PL_SetHopTable.* Load the new hopping table.
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, starting with HOP-1 frequency

2002

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()
RoamingKeepCfg	Variable	Informs the current restart is called by application. In case of roaming the Connection Parameters shall be kept.
$T_{T\text{-}Unique}$	Time	See Table 213, definition of PAIRING_UNIQUE_TIMEOUT
$T_{T\text{-}Button}$	Time	See Table 213, definition of PAIRING_BUTTON_TIMEOUT

2003

5.7.3.2 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 W-Device received, the PL shall generate an acknowledgement, which shall be automatically transmitted in the following uplink packet. If Transfer.req was not invoked, then the payload shall be set to zero (dummy Uplink packet).

6 Data Link Layer (DL-A)

6.1 General

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 W-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.

6.2 General (W-Master)

6.2.1 General

Figure 99 shows an overview of the structure and the services of the W-Master's Data Link Layer.

6.2.2 W-Track and W-Device-Mapper (TDMapper)

The W-Track and W-Device-Mapper is used to assign a W-Port to a specific W-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 W-Tracks.

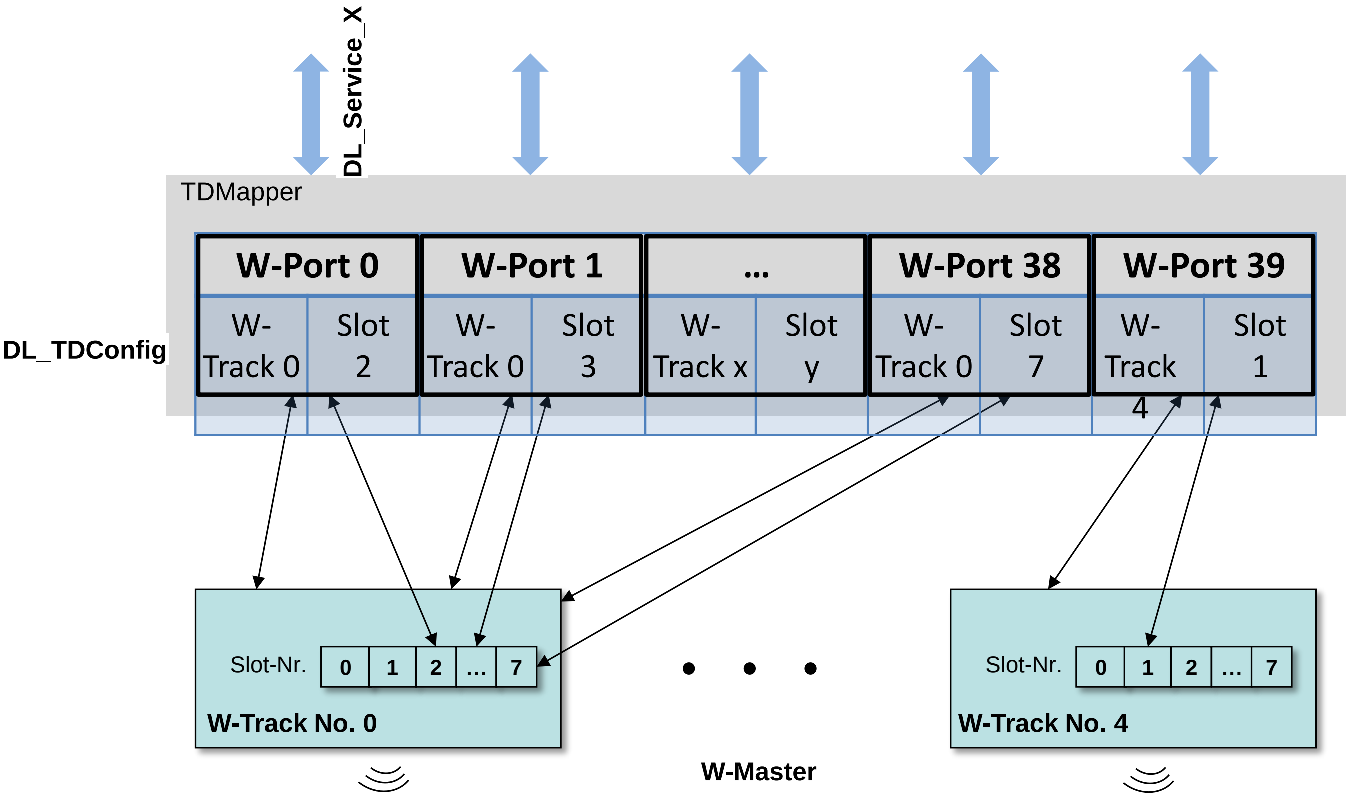


Figure 48 W-Track and W-Device-Mapper (TDMapper)

6.3 General (W-Device)

Figure 95 shows an overview of the structure and the services of the W-Device's Data Link Layer.

6.4 DL-A services

6.4.1 Overview

Table 37 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 37 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*		

6.4.2 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 38.

Table 38 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 [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 W-Track, use is restricted according to Table 178. Permitted values:
YES
NO

MasterCommand
This parameter contains the MasterCommand, see C.2.2 and Table 178.

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).

6.4.3 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 39

Table 39 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 79).

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 transmission.
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 79).
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)

6.4.4 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 sent in DLink, or if the last data have to be retransmitted. The parameters of the service are listed in Table 40.

Table 40 DownLinkAck

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

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 octets
PreDLSet
This parameter is only used for the CMDHANDLER to support low energy W-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 transmission has been confirmed or not.
PD handler, Cmd handler, Event handler and ISDU handler shall decide if a retransmit is needed or not.
SubCycleCounterOut
This parameter is only used for PDHANDLER to indicate the current W-Sub-Cycle of the WMasterCycleTimeOut
SubCycleCounterIn
This parameter is only used for PDHANDLER to indicate the current W-Sub-Cycle of the WMasterCycleTimeIn

6.4.5 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 sent, or the last data have to be retransmitted. The parameters of the service are listed in Table 41

Table 41 UpLinkAck

Parameter Name	.ind
Argument	M
ComChannel	M
Length	M
Acknowledge	M
SubCycleCounterOut	C
SubCycleCounterIn	C

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 packet.
Range: 0 to 15 octets
Acknowledge
This parameter indicates whether the last uplink transmission has been confirmed or not.
PD handler, Event handler and ISDU handler shall decide if a retransmit is needed or not.
SubCycleCounterOut
This parameter is only used for PDHANDLER to indicate the current W-Sub-Cycle of the WMasterCycleTimeOut
SubCycleCounterIn
This parameter is only used for PDHANDLER to indicate the current W-Sub-Cycle of the WMasterCycleTimeIn

6.4.6 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 42.

Table 42 EventMsg

Parameter Name	.req	.ind	.rsp	.cnf
Argument	M	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 packets.
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 [1]
EventData see Table 196

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 79). In case of EOS (end of service), no data are delivered.

6.4.7 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 43.

Table 43 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 79).

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)

6.4.8 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 44.

Table 44 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 packet.
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 transmitted input Process Data. Permitted values: 0 to 32

FlowCtrl
This parameter contains the flow control (see Table 79).

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)

6.5 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.

6.6 Message handler

6.6.1 General

The layer DL-A comprises the Message handler as shown in Figure 49.

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

Figure 49 shows the state machine of the W-Master Message handler. The state machines describe the order how the different W-Messages are placed in the downlink payload. First step is placement of MasterCommands and process data for all slots (0 to 7), next step is placement of Events for all slots (0 to 7) and the final step is placement of ISDU for all slots (0 to 7).

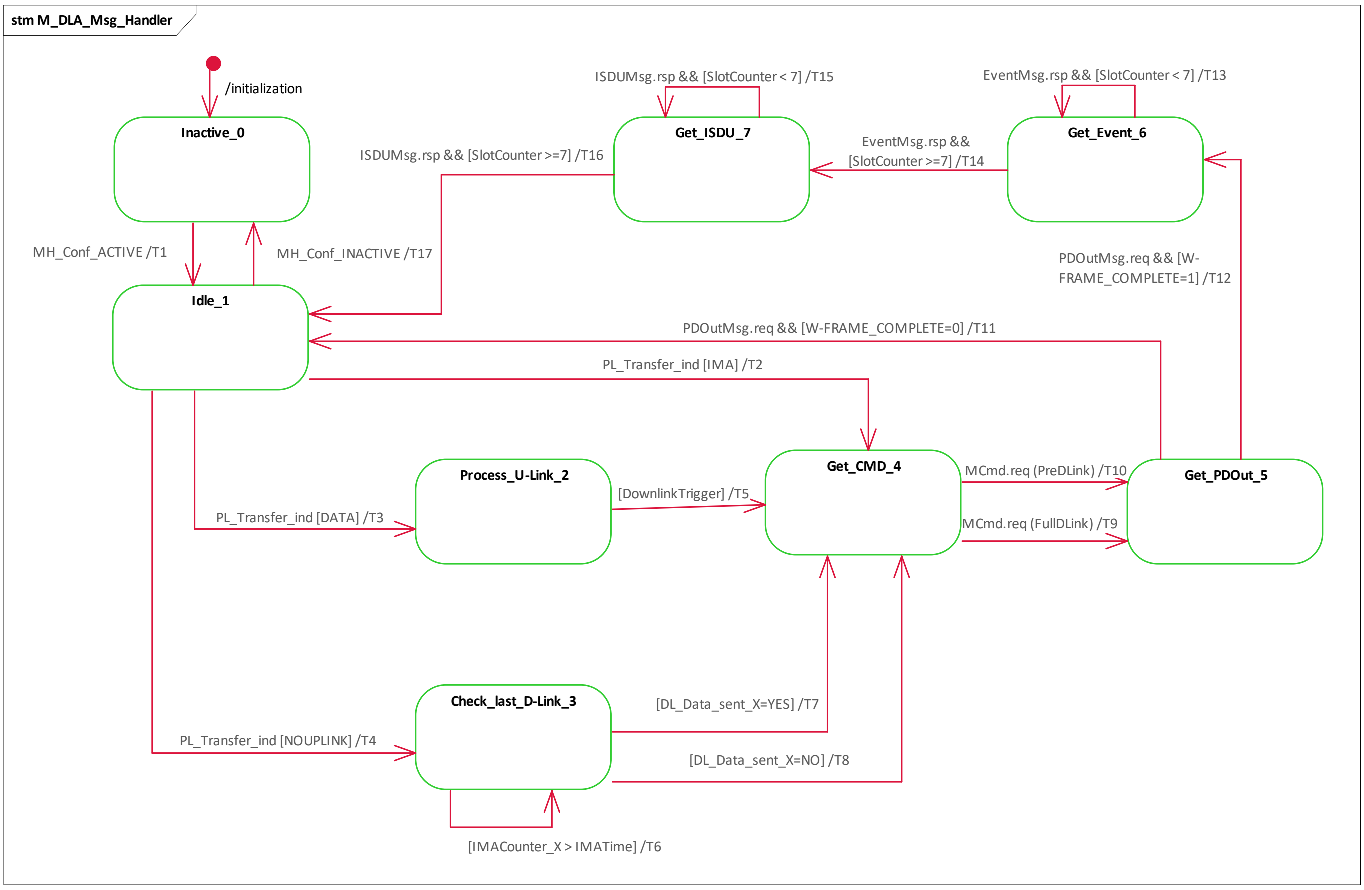


Figure 49 State machine of the W-Master Message handler

2330 **Table 45 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 52). Set RemainingLength to DLink-Payload (37 Octet).
Idle_1	Waiting for trigger PL_Transfer.ind service indication. The PL_Transfer service delivers the slot number (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 123, ULink Control Octet
Check_last_DLink_3	Check if data have been sent for this W-Device / Slot_N in last downlink
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.
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.
GetEvent_6	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.
GetISDU_7	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.

2331

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T1	0	1	*The DL-mode handler activates the Message handler via MH_Conf_ACTIVE.*
T2	1	4	*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 131 and Figure 132) 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.
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	4	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.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION */Remarks*
T7	3	4	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. If RetryCounter_X > MaxRetry A local MaxRetry event shall be reported via invocation of service EventMsg.ind(M_IOLW_Retry_ErrorIOLWM) to the W-Master application. Invoke service DownLinkAck (Slotnumber, CMDHANDLER, RemainingLength, PreDLSet, Acknowledge) to acquire MCMD from Command handler for SlotNumber_X delivered via PL_Transfer.ind.
T8	3	4	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 MasterCommand = DeviceOperate: Set ResetCounters_X = YES. If MCmd.req(SendWMessage=NO): No compilation of downlink CO necessary. Acquire PDOut for SlotNumber_X through invocation of the DownLinkAck(Slotnumber, PDHANDLER, RemainingLength, Acknowledge, SubCycleCounterOut_X, SubCycleCounterIn_X) service.
T10	4	5	If MCmd.req(SendWMessage=YES): set DL_Data_sent_X = 1 and set PreDLSet = YES. If MasterCommand = DeviceOperate: Set ResetCounters_X = YES. Acquire PDOut for SlotNumber_X through invocation of the DownLinkAck(Slotnumber, PDHANDLER, RemainingLength, Acknowledge, SubCycleCounterOut_X, SubCycleCounterIn_X) service.
T11	5	1	W-Frame is not complete. Wait for next ULink / next Slotnumber via PL_Transfer.ind in state Idle_1. 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	5	6	W-Frame is complete, all ULinks have been received. 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. Set SlotCounter to 0. Acquire Event through invocation of the DownLinkAck(SlotCounter, EVHANDLER, RemainingLength, ACK_Buf_X) service. Check all Slots : If ResetCounter_X = YES: set ResetCounter_X = No, set SubCycleCounterOut_X = 0 and set SubCycleCounterIn_X = 0. Increment SubCycleCounterOut_X and SubCycleCounterIn_X by 1. If SubCycleCounterOut_X = WMasterCycleTimeOut_X: set SubCycleCounterOut_X = 0. If SubCycleCounterIn_X = WMasterCycleTimeIn_X: set SubCycleCounterIn_X = 0. *For conversion of WMasterCycleTime to W-Sub-Cycle see C.4.12*

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /*Remarks*
T13	6	6	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). Increment SlotCounter and Invoke DownLinkAck(SlotCounter, EVHANDLER, RemainingLength, ACK_Buf_X) service to acquire data for next Slot.
T14	6	7	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). Set SlotCounter to 0. Acquire ISDU through invocation of the DownLinkAck(SlotCounter, ISDUHANDLER, RemainingLength, ACK_Buf_X) service.
T15	7	7	If ISDUMsg.req(SendWMessage=YES): Set DL_Data_sent_X = 1. Place W-Message to DLink payload and decrease RemainingLength by 2 (for Control Octet). increment SlotCounter and Invoke DownLinkAck(SlotCounter, ISDUHANDLER, RemainingLength, ACK_Buf_X) service to acquire data for next Slot
T16	7	1	If ISDUMsg.rsp(SendWMessage=YES): Set DL_Data_sent_X = 1. Place W-Message to DLink payload and decrease RemainingLength with the delivered length from ISDUMsg.rsp + 2 (for Control Octet). Set SlotCounter to 0. All acyclic data for all Slots / W-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.
T17	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 /W-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.
ACK_Buf_X	Variable	Store Acknowledge for slot number X
PreDLSet	Bool	Marks if the PreDownlink will be used or not.
ResetCounter_X	Variable	Marks if a DeviceOperate MasterCommand has been sent and cycle counters shall be reset.
SubCycleCounterOut_X	Variable	Counter for W-Sub-Cycles of the WMasterCycleTimeOut
SubCycleCounterIn_X	Variable	Counter for W-Sub-Cycles of the WMasterCycleTimeIn

NOTE : _X marks the variables which individual for every Slotnumber. The range of _X is slot number 0 to 7

If a W-Message shall be sent (SendWMessage = YES), the Message handler shall compile the control octet of the corresponding DL-B handler as defined in Table 46 Compilation of Downlink Control Octet.

6.6.3 Compilation of DLink Control Octet

The Master Message handler (see Figure 49) shall compile the control octet for a DLink delivered with the data via the corresponding DL-B handler as defined in Table 46.
See Figure 121 for definition of DLink Control Octet.

Table 46 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)	FlowCtrl	0	**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.

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

Figure 50 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.

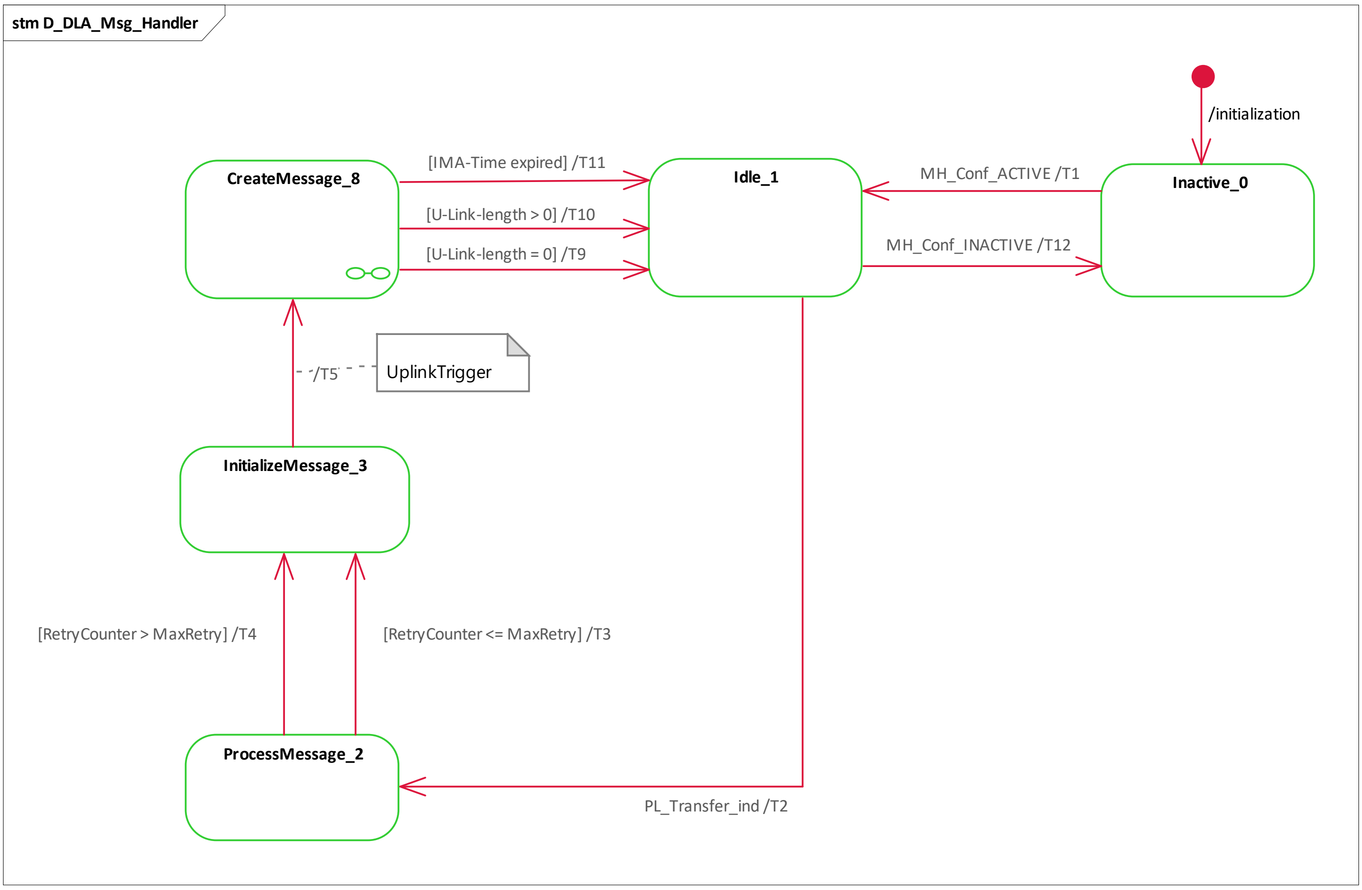
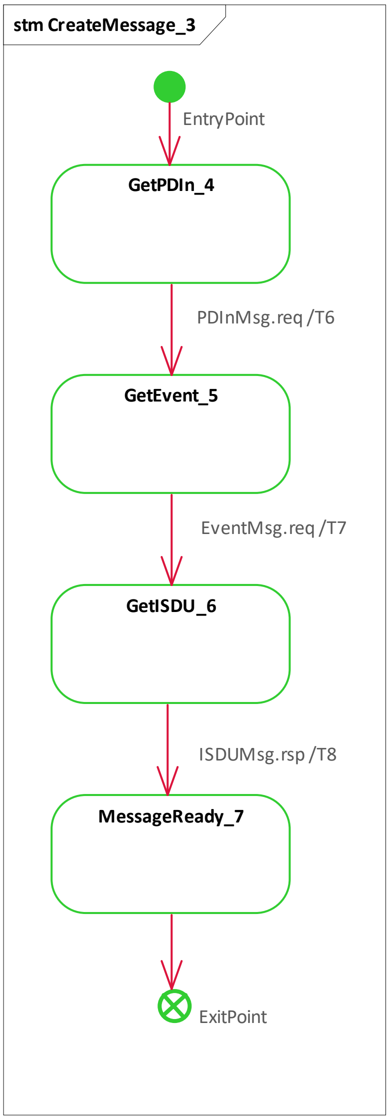


Figure 50 State machine of the W-Device Message handler



2364
2365 **Figure 51 W-Device Message handler sub state machine “CreateMessage_8” (DL-A)**
2366

Table 47 shows the state transition tables of the W-Device Message handler.

Table 47 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 47, 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 121
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 123.

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. which leads to D_IOLW_Retry_Error generated by application see 10.9.3. *NOTE:* *The parameter MaxRetry is delivered via service DL_SetParam.*
T5	3	4	Invoke MCmd.ind, ISDUMsg.ind, EventMsg.cnf and PDOutMsg.ind service indications to distribute received W-Messages. Acquire PDIn through invocation of the service UpLinkAck(PDHANDLER, RemainingLength, Acknowledge). SubCycleCounterOut, SubCycleCounterIn). If MasterCommand = DeviceOperate: set CycleCounterOut = 0 and set CycleCounterIn = 0. Increment SubCycleCounterIn and SubCycleCounterOut by 1. If SubCycleCounterOut = WMasterCycleTimeOut: set SubCycleCounterOut = 0. If SubCycleCounterIn = WMasterCycleTimeIn: set SubCycleCounterIn = 0. *For conversion of WMasterCycleTime to W-Sub-Cycle see C.4.12.*
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.*

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
SubCycleCounterOut	Variable	Counter for W-Sub-Cycle of the WMasterCycleTimeOut
SubCycleCounterIn	Variable	Counter for W-Sub-Cycle of the WMasterCycleTimeIn

6.6.5 Compilation of ULink Control Octet

The Device Message handler (see Figure 50) shall compile the control octet for an ULink delivered via the corresponding DL-B handler as defined in Table 48. See Figure 123 for definition of ULink Control Octet.

Table 48 Compilation of Uplink Control Octet

<table>
<tr><th>Compiled Control Octet →
handler: ↓</th><th>Channel Code (ChC)
(created by Message handler)</th><th>Flow Control (FC)
(delivered by handler)
See NOTE 1</th><th>Data follows</th></tr>
<tr><td rowspan="3">PDInMsg.req</td><td rowspan="2">1
(Process data in)</td><td>FlowCtrl</td><td>Yes</td></tr>
<tr><td>FlowCtrl
= ABORT</td><td>No</td></tr>
<tr><td>2
(Process data in invalid)</td><td>-</td><td>No</td></tr>
<tr><td rowspan="2">EventMsg.req</td><td rowspan="2">4
(EVENT)</td><td>FlowCtrl</td><td>Yes</td></tr>
<tr><td>FlowCtrl
= EOS or ABORT</td><td>No</td></tr>
<tr><td rowspan="2">ISDUMsg.rsp</td><td rowspan="2">3
(ISDU)</td><td>FlowCtrl</td><td>Yes</td></tr>
<tr><td>FlowCtrl
= EOS or ABORT</td><td>No</td></tr>
<tr><td>IMA Uplink</td><td colspan="3">No Control Octet needed.
See Table 30</td></tr>
</table>

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 49 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 49 Service assignments within W-Master and W-Device

Service name	W-Master	W-Device
DL_PDCycle	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_PDCycle (W-Master and W-Device)

The Data Link Layer uses the DL_PDCycle service to indicate the end of a WMasterCycleTime (See C.4.12) period after start of Process Data transmission to the Application Layer. This service has no parameters. The service primitives are listed in Table 50.

Table 50 DL_PDCycle

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 51

Table 51 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 W-Device application and to get control information via the PD handler (see clause A.9 PDVALID PDINVALID). The parameters of the service primitives are listed in Table 52.

Table 52 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 53.

Table 53 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 54.

Table 54 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 55.

Table 55 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 64, Sequence chart for Event). The parameters of the service primitives are listed in Table 56.

Table 56 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 150, see Table A.17 in [1])

Type
This parameter indicates the Event category.
Permitted values: ERROR, WARNING, NOTIFICATION (see Table 152, see Table A.19 in [1])

Mode
This parameter indicates the Event mode.
Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table 153, see Table A.20 in [1])

EventCode
This parameter contains a code identifying a certain Event (see Annex D, see Table D.1 in [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 57.

Table 57 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
In case of a low energy W-Device the indexes from Table 194 are not allowed to read or write by a W-Master and shall return a STATE_CONFLICT

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 7.4.4 ,A.11.1, see Table 74

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 69, see Table 102 in [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 58.

Table 58 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 W-Track and Slot via W-Port Configuration Manager / System Management. The service primitives are listed in Table 59.

Table 59 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 W-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 IO-Link Wireless specific index range via ISDU. Therefore, DL_Read uses the DL_ISDUTransport service. The parameters of the service primitives are listed in Table 60.

Table 60 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 IO-Link Wireless specific index range (see Table 179 and Table 177).
Permitted values: see Table 188 and Table 179
In case of a low energy W-Device the indexes from Table 194 are not allowed to read or write by a W-Master and shall return STATE_CONFLICT.

Subindex
This parameter contains the Subindex of the W-Device parameter in Page 1 (see Table 177) or in the IO-Link Wireless specific index range.
Permitted values: For Page 1 values 1 to 15, for IO-Link Wireless specific indexes, see clause C.4

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)
ISDU_TIMEOUT (ISDU acknowledgement time elapsed, see Figure 69, see Table 102 in [1]).

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 IO-Link Wireless specific index range 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 in Table 61.

Table 61 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 IO-Link Wireless specific index range (see Table 179 and Table 177).
Permitted values: see Table 188 and Table 179
In case of a low energy W-Device the indexes from Table 194 are not allowed to read or write by a W-Master and shall return STATE_CONFLICT .

Subindex
This parameter contains the Subindex of the W-Device parameter in Page 1 (see Table 177) or in the IO-Link Wireless specific index range.
Permitted values: For Page 1 values 1 to 15, for IO-Link Wireless specific indexes, see clause C.4

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 62.

Table 62 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 in Table 63.

Table 63 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 64.

Table 64 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 65.

Table 65 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 C.4.4.3). 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 C.4.4.2). 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.

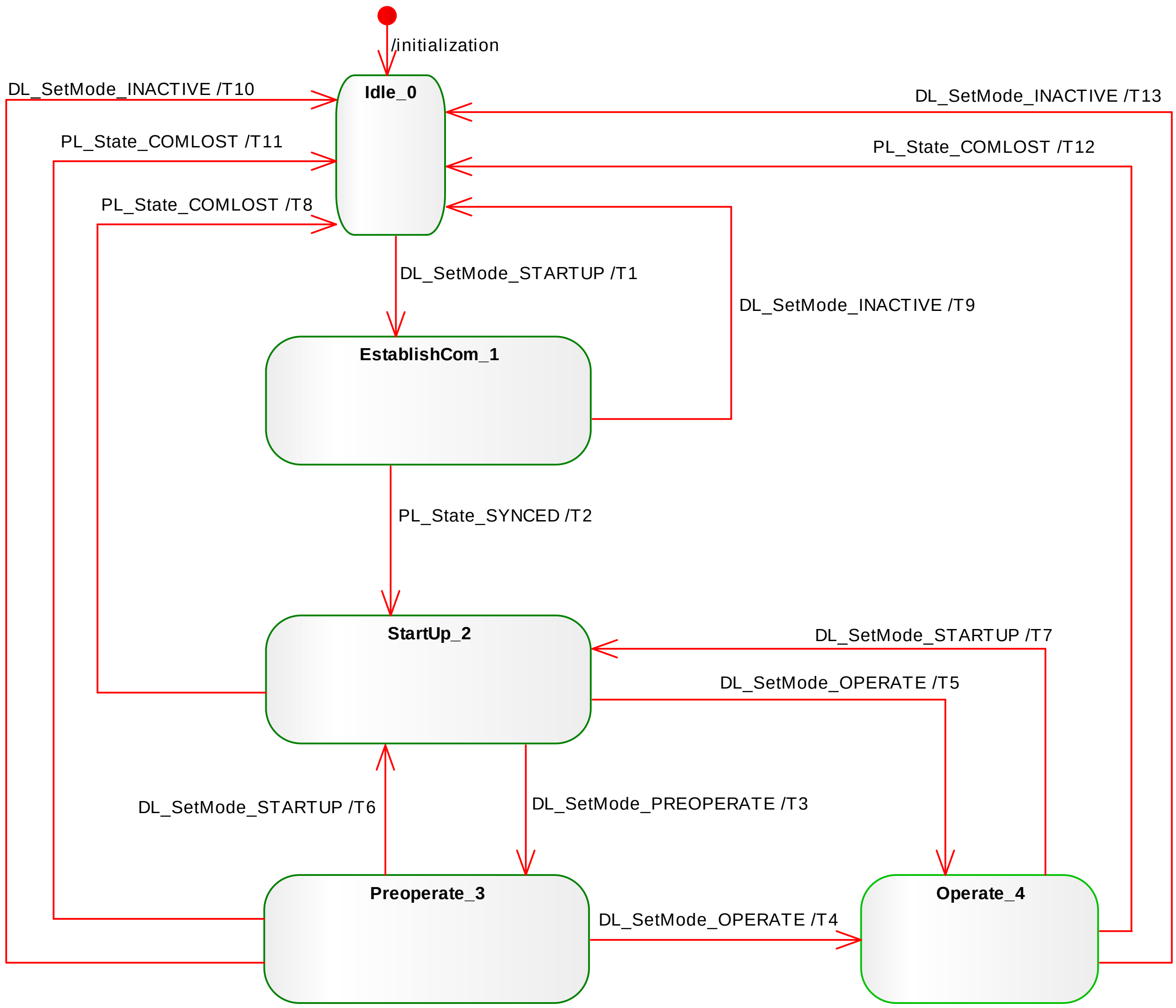


Figure 52 State Machine of the W-Master DL-mode handler

Table 66 shows the state transition tables of the W-Master DL-mode handler.

Table 66 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 81).
Preoperate_3	Commands, Events and ISDU without Process Data
Operate_4	Process Data, Commands, Events and ISDU

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 66) and ISDU handler (call IH_Conf_ACTIVE see Figure 59) 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 59). 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 54) and Event handler (call EH_Conf_ACTIVE see Figure 59). 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 81).
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.

2757

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), or EH (Event handler)
xx_Conf_INACTIVE	Call	This call deactivates the respective handler. xx is substitute for MH (Message handler), IH (ISDU handler), CH (Command handler), PD (PD handler) or EH (Event handler)

2758

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

Figure 53 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.

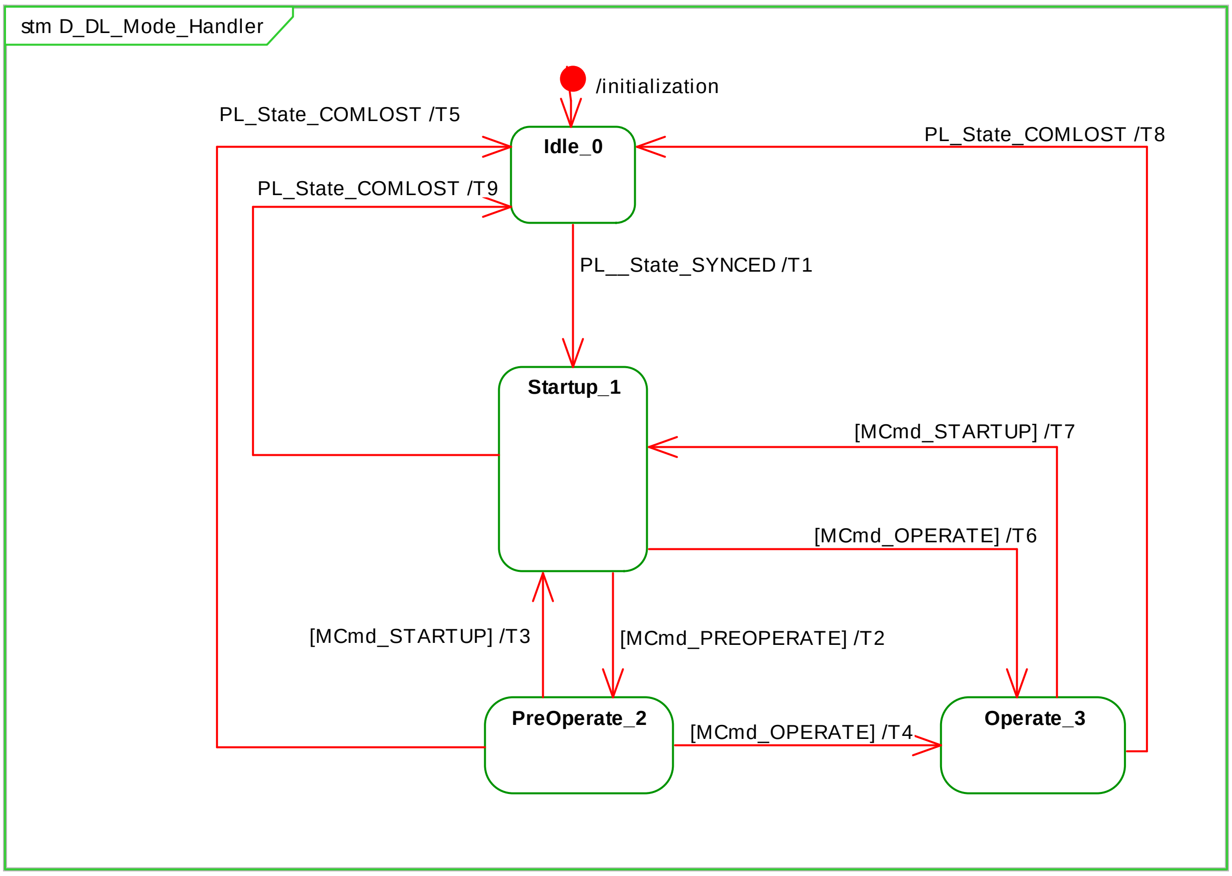


Figure 53 State machine of the W-Device DL-mode handler

Table 67 shows the state transition tables of the W-Device DL-mode handler.

Table 67 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 82)
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 50) , ISDU handler (call IH_Conf_ACTIVE in Figure 60) and Command handler (call CH_Conf_ACTIVE in Figure 62). 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 65). 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 65). 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 57). 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 90 and Table 125)
T6	1	3	*W-Device command handler received MasterCommand (MCmd_OPERATE).* Activate Process Data handler (call PD_Conf_ACTIVE in Figure 57) and Event handler (call EH_Conf_ACTIVE in Figure 65). 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 57) and Event handler (call EH_Conf_INACTIVE in Figure 65). 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 90 and Table 125)
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 90 and Table 125)

7.3 Process Data handler

7.3.1 General

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.

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

7.3.2.1 General

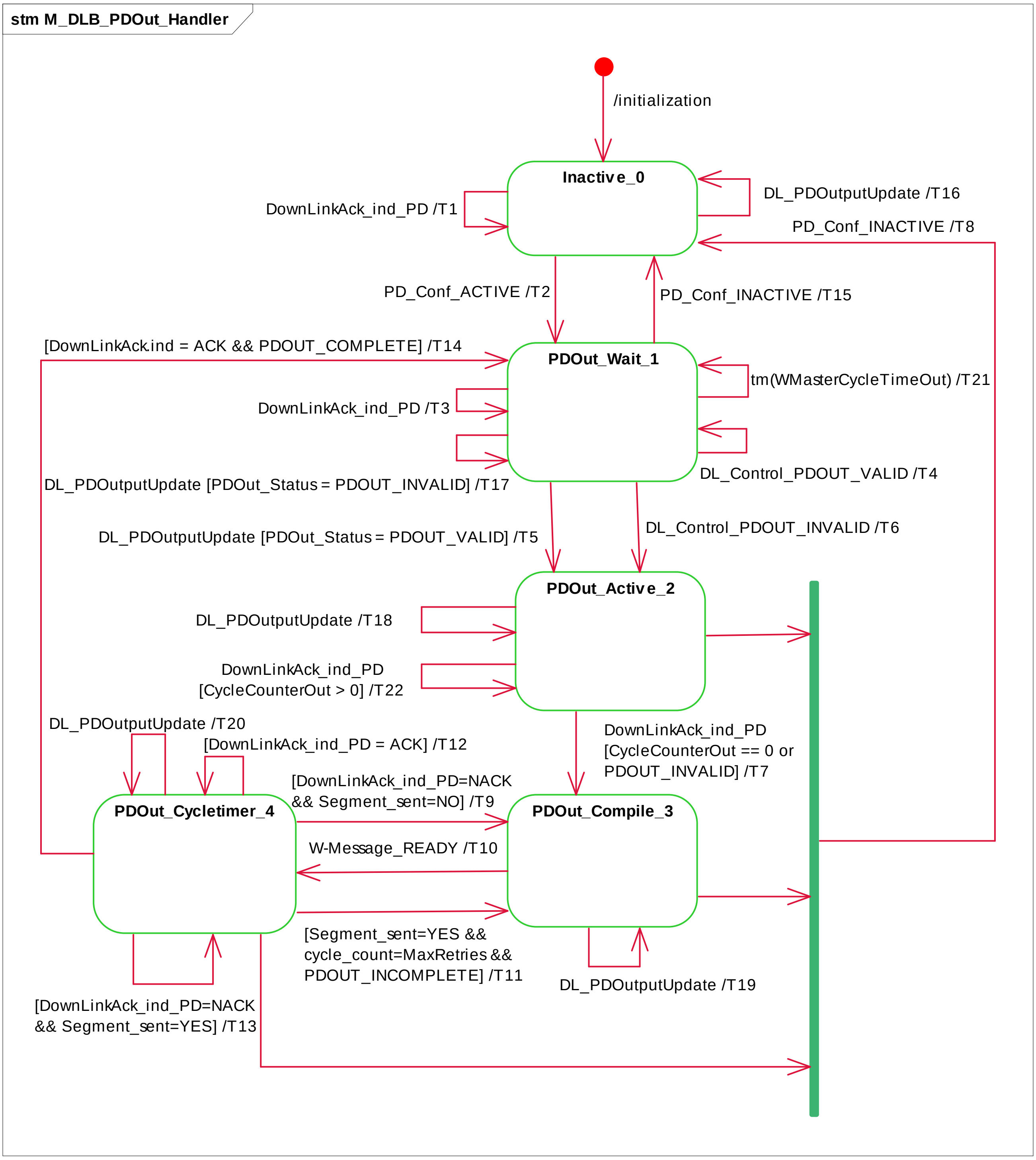


Figure 54 State machine for W-Master PDOut handler

Table 68 shows the state transition tables of the W-Master PDOut handler.

Table 68 State Transition 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 67, 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 121, DLink Control Octet) Maximum segment length shall be limited by parameter MaxPDSegLength (via DL_SetParam) to distribute PDOut data (see Figure 105 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.7.7.2 “Transmission of segmented data (PD, EV, ISDU)”. For Retry-Handling see 7.7.3 “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_Out_Status to PDOUT_INVALID
T3	1	1	No Process Data (PDOut) to send, invoke PDOutMsg.req(SendWMessage = NO).
T4	1	1	Set PD_Out_Status 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. Start Timer “WMasterCycleTimeOut”.
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_Out_Status 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_Out_Status = 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_Out_Status = 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)
T21	1	1	Invoke DL_PDCycle.ind. See NOTE 1.
T22	2	2	- *Waiting for new WMasterCycleTimeOut*

INTERNAL ITEMS	TYPE	DEFINITION
PD_Out_Status	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

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

7.3.2.2 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)

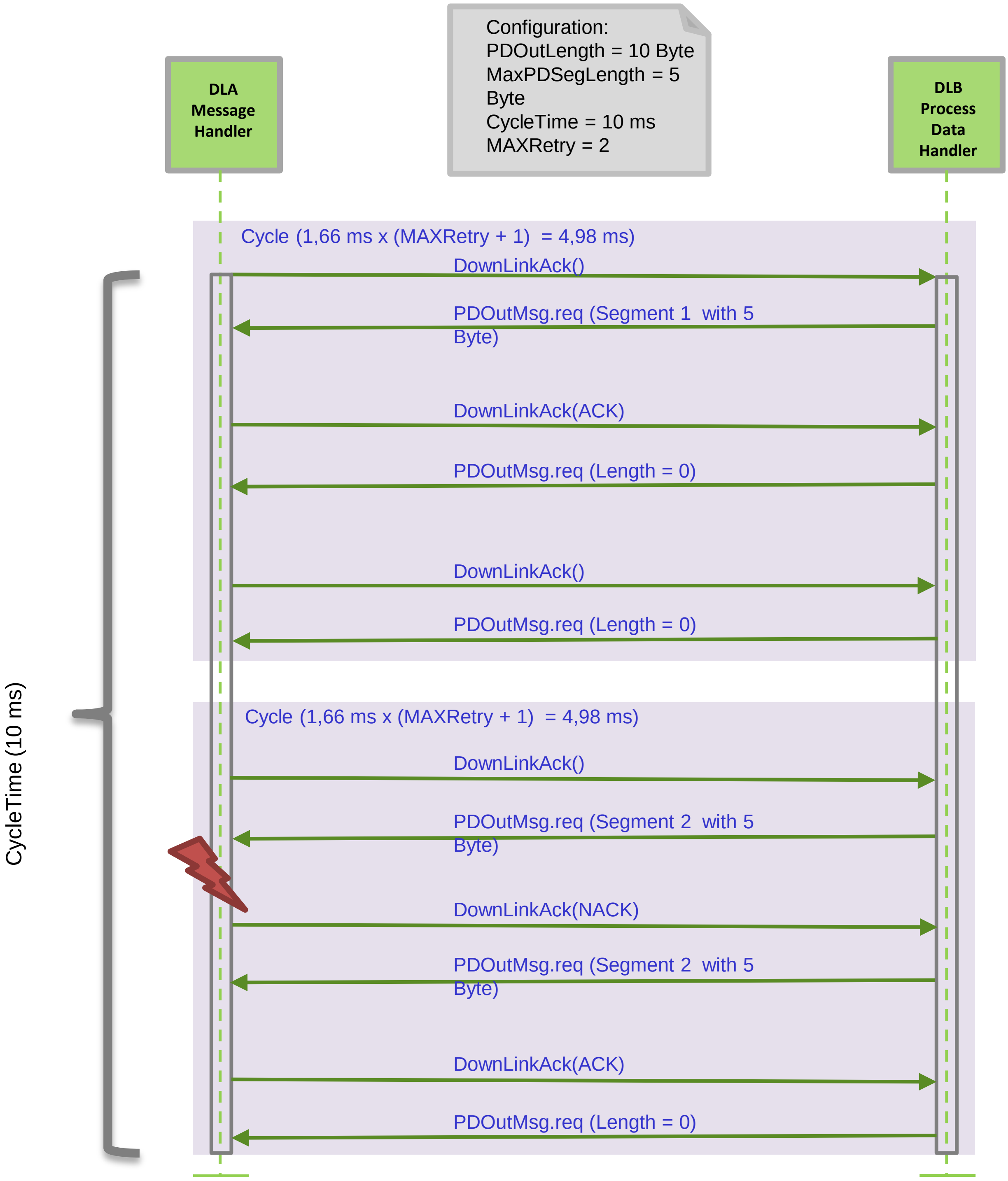


Figure 55 Sequence diagram for PDOut distribution

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

Figure 56 shows the state machine of the W-Master Process Data In handler.

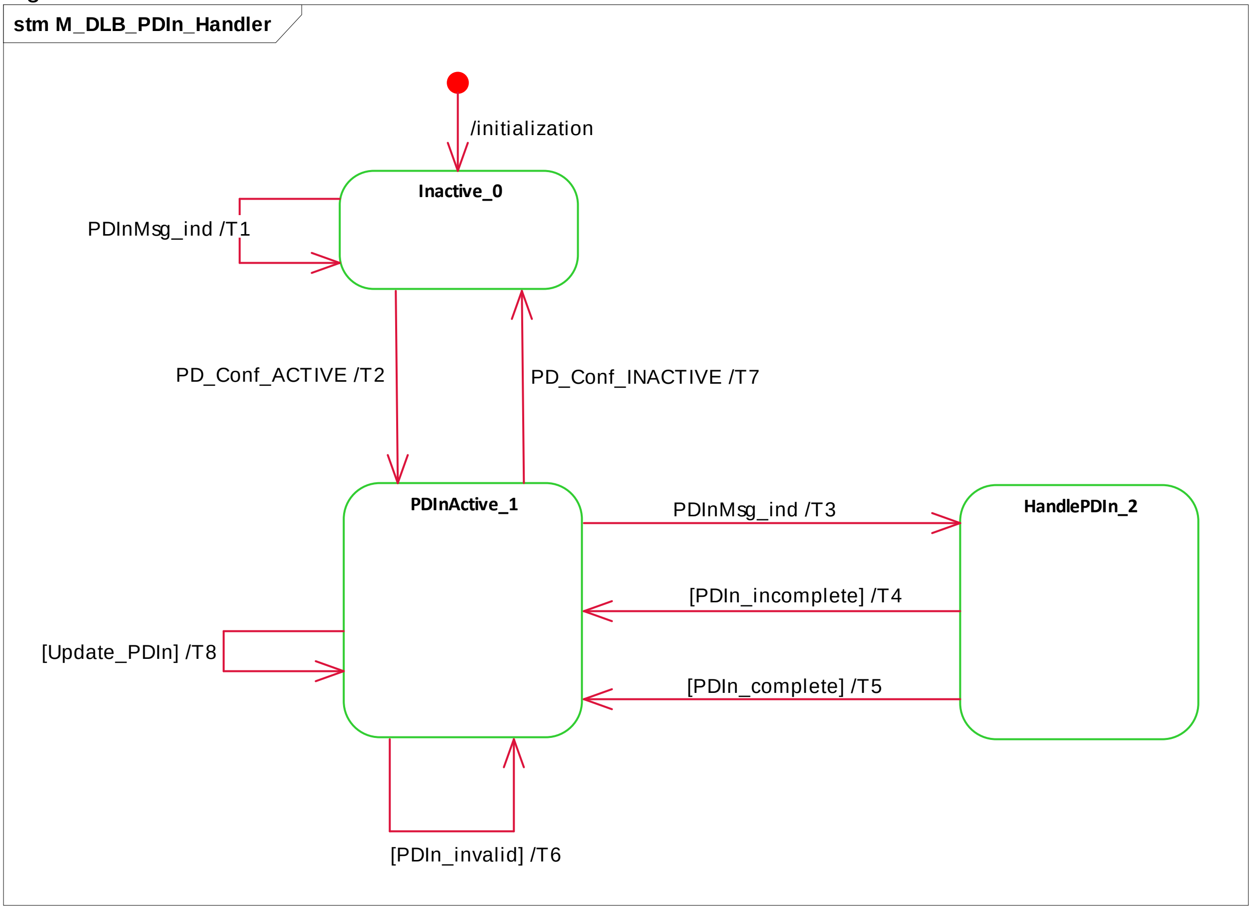


Figure 56 State machine for W-Master PDIn handler

Table 69 shows the state transition tables of the W-Master PDIn handler.

Table 69 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 67, 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. 7.7.2. For Retry-Handling see 7.7.3.

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.*
T4	2	1	-
T5	2	1	-
T6	1	1	DLink Control Octet contained “Process Data In Invalid”. If PDIn_Status = PDIN_VALID, set PDIn_Status = PDIN_INVALID Invoke DL_Control.ind (PDIN_INVALID).
T7	1	0	*W-Master DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T8	1	1	Invoke DL_PDInputTransport.ind (see 7.1.3). If PDIn_Status = PDIN_INVALID, set PDIn_Status = PDIN_VALID Invoke DL_Control.ind (PDIN_VALID).

INTERNAL ITEMS	TYPE	DEFINITION
PDIn_Status	Variable	Indicate if PDIn is valid or invalid 0 = PDIN_INVALID 1 = PDIN_VALID
PDIn_complete	Guard	All segments have been received, PDIn data is complete and received data length of all segments are equal to the W-Device ProcessDataIn (see Table 177)
PDIn_incomplete	Guard	PDIn data is incomplete, wait for next data segment or received data length of all segments are not equal to the W-Device ProcessDataIn (see Table 177)
Update_PDIn	Guard	*To reduce Jitter the PDIn is updated at the end of a WMasterCycleTimeIn*. Update_PDIn is triggered if [PDIn_complete] && DownLinkAck_Ind_PD [SubCycleCounterIn = WMasterCycleTimeIn – 1]
PDIn_invalid	Service	PDInMsg.ind delivered PDIN_INVALID (via ControlOctet)

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

Figure 57 shows the state machine of the W-Device Process Data Out handler.

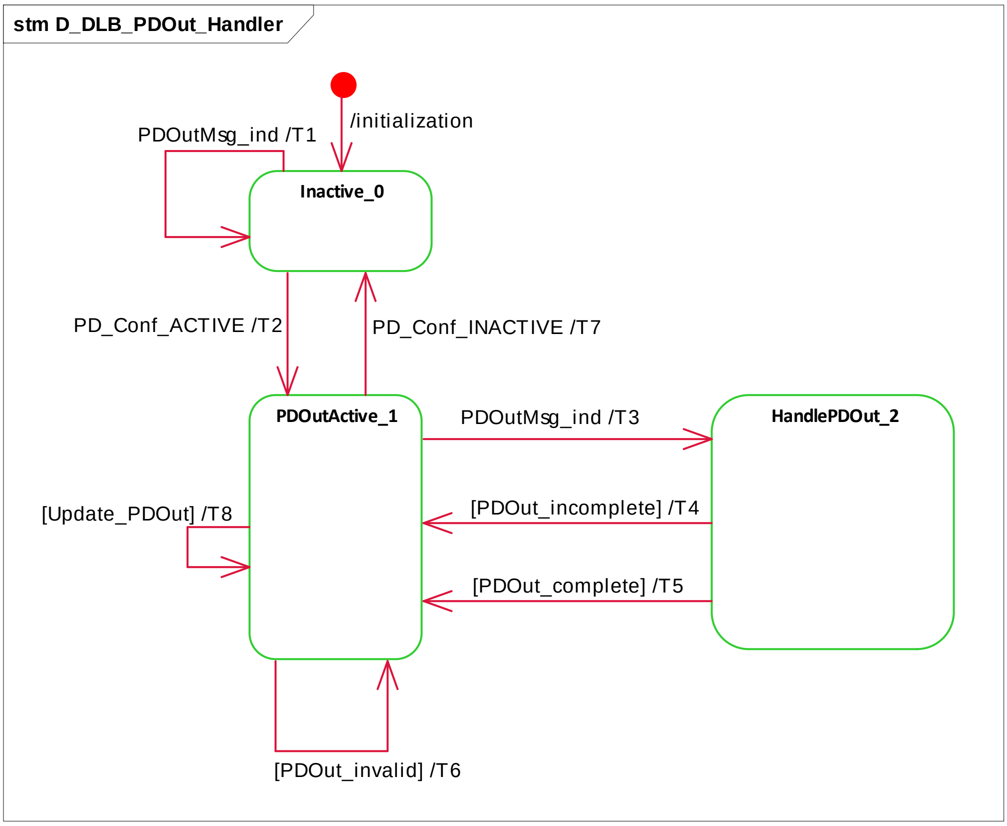


Figure 57 State machine of the W-Device Process Data Out handler

Table 70 shows the state transition tables of the W-Device PDOut handler.

Table 70 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 67, 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. 7.7.2. For Retry-Handling see clause 7.7.3.

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	-
T6	1	1	DLink Control Octet contained "Process Data Out Invalid". If PDOut_Status = PDOUT_VALID, set PDOut_Status = PDOUT_INVALID Invoke DL_Control.ind (PDOUTINVALID).
T7	1	0	*DL-mode handler disables Process Data handler via PD_Conf_INACTIVE.*
T8	1	1	Invoke DL_PDOutputTransport.ind (see 7.1.6). If PDOut_Status = PDOUT_INVALID, set PDOut_Status = PDOUT_VALID Invoke DL_Control.ind (PDOUTVALID).

2827

INTERNAL ITEMS	TYPE	DEFINITION
PDOut_Status	Variable	Indicate if PDOut is valid or invalid 0 = PDOUT_INVALID 1 = PDOUT_VALID
PDOut_complete	Guard	All segments have been received, PDOut data is complete and received data length of all segments are equal to the W-Device ProcessDataOut (see Table 177)
PDOut_incomplete	Guard	PDOut data is incomplete, wait for next data segment or received data length of all segments are not equal to the W-Device ProcessDataOut (see Table 177)
Update_PDOut	Guard	*To reduce Jitter the PDOut is updated at the end of a WMasterCycleTimeOut.* Update_PDOut is triggered if [PDOut_complete] && UpLinkAck_Ind_PD [SubCycleCounterOut = WMasterCycleTimeOut – 1]
PDOut_invalid	Guard	PDOutMsg.ind delivered PDOUT_INVALID (via ControlOctet)

2828

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

Figure 58 shows the state machine of the W-Device Process Data In handler.

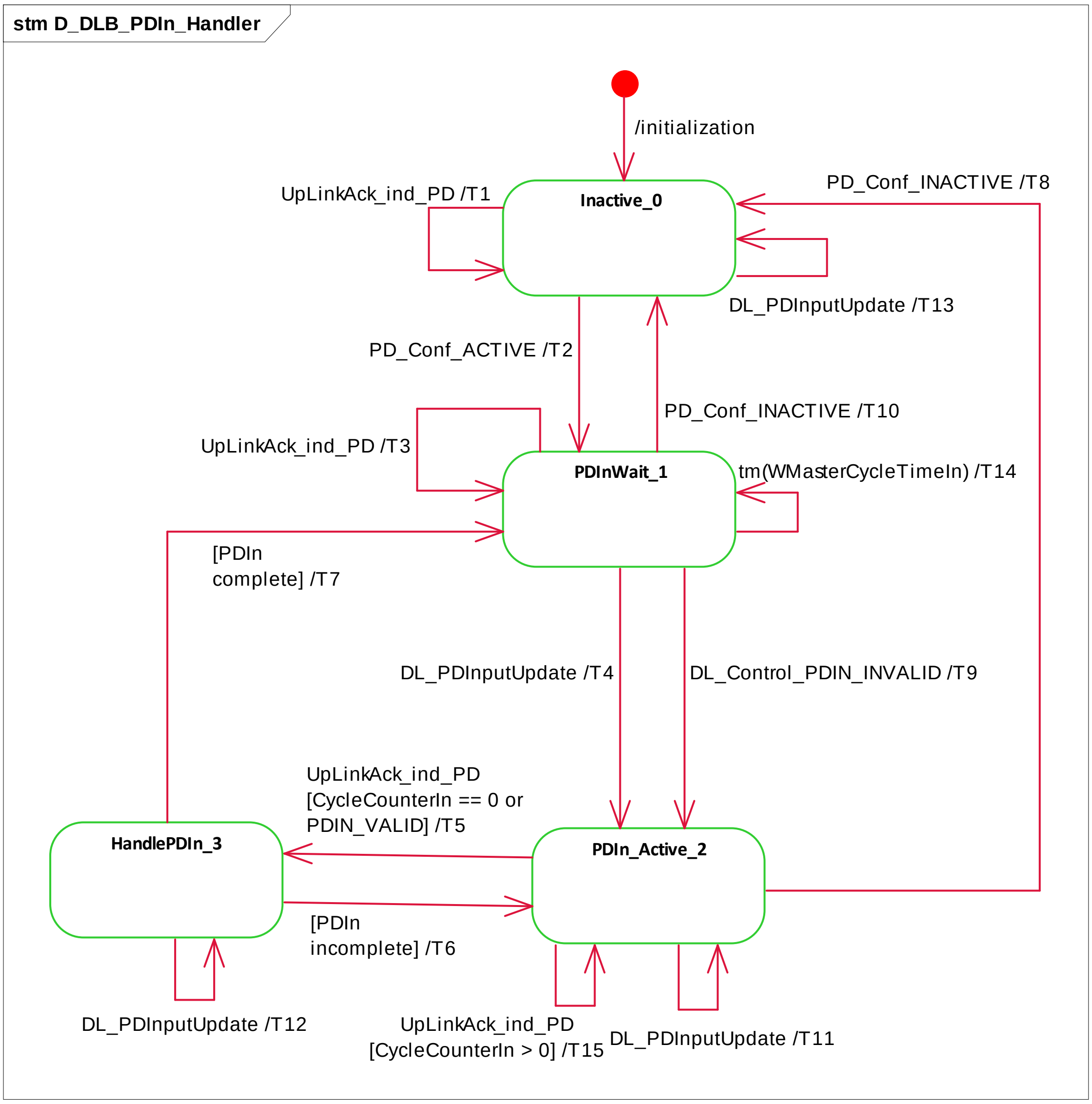


Figure 58 State machine of the W-Device Process Data In handler

Table 71 shows the state transition tables of the W-Device PDIn handler.

Table 71 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 67, 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. 7.7.2 Transmission of segmented data (PD, EV, ISDU). For Retry-Handling see 7.7.3 “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 PDInMsg.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 PDInMsg.req(SendWMessage = NO).
T4	1	2	Prepare input Process Data for PDInMsg.req for next Message handler demand. Start Timer “WMasterCycleTimeIn”.
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)
T14	1	1	Invoke DL_PDCycle.ind.
T15	2	2	*Waiting for new WMasterCycleTimeIn*

7.4 Indexed Service Data Unit (ISDU) handler

7.4.1 General

The general structure of an ISDU is demonstrated in Figure 48 and specified in detail in Clause A.5.in [1] The ISDU allows accessing data objects (parameters and system commands) to be transmitted. The data objects shall be addressed by the “Index” element.

7.4.2 State machine of the W-Master ISDU handler

Figure 59 shows the state machine of the W-Master ISDU handler

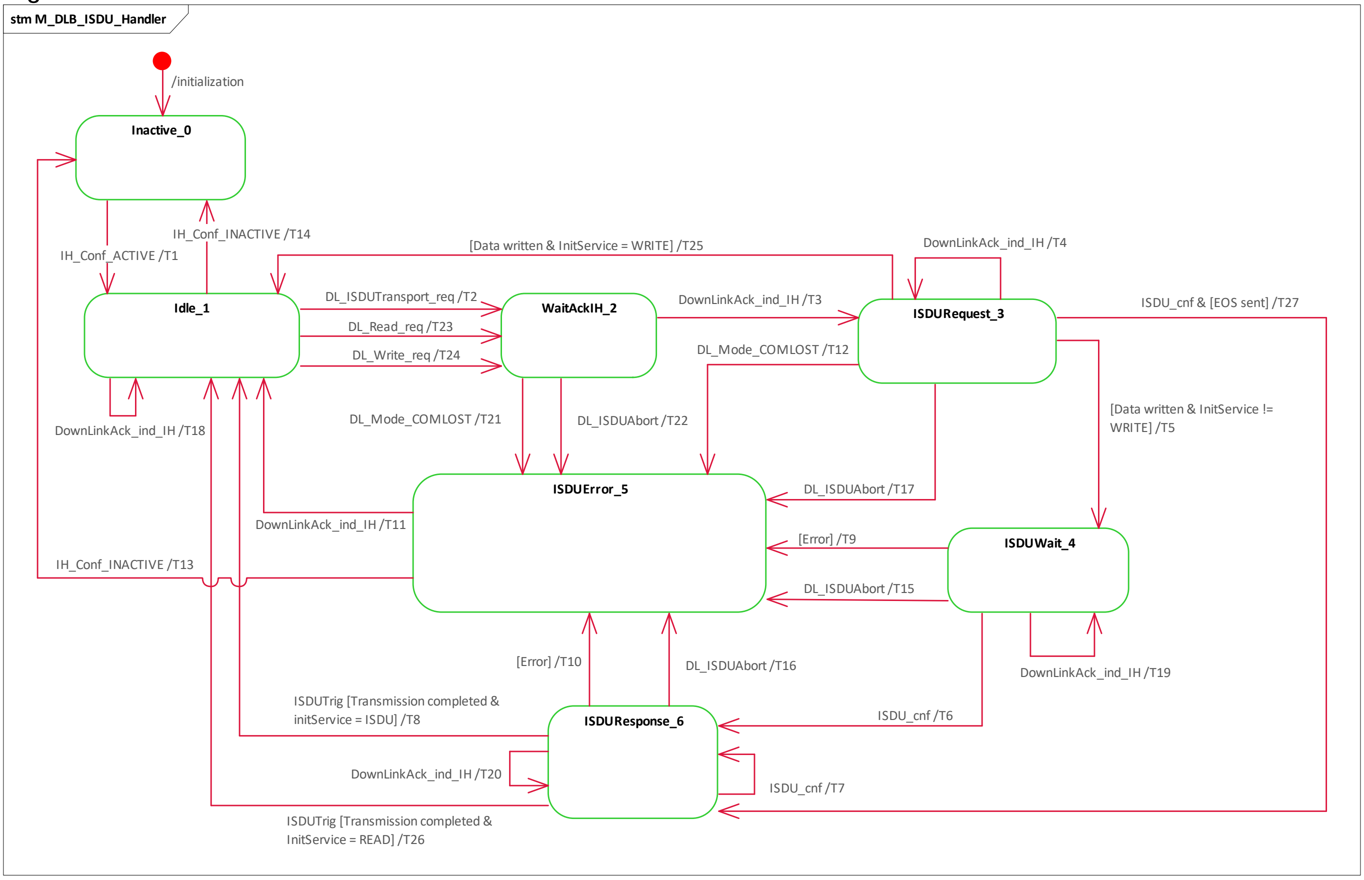


Figure 59 State machine of the W-Master ISDU handler

Table 72 shows the state transition tables of the W-Master ISDU handler.

Table 72 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 7.7.2. For Retry-Handling see 7.7.3.
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 7.7.2. In case of FC = START or 0x0B...0x18 any previously incomplete received data shall be ignored. For Retry-Handling see 7.7.3.

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 79 (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 7.7.2 Transmission of segmented data)
T5	2	3	Start timer (ISDUTime)
T6	3	5	Stop timer (ISDUTime)
T7	6	6	Receive ISDU response data via ISDUMsg.cnf
T8	6	1	Invoke positive DL_ ISDUTransport confirmation when last segment (EOS) has been received (see 7.7.2 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). If InitService = ISDU Invoke negative DL_ ISDUTransport confirmation If InitService = READ Invoke negative DL_ Read confirmation 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	4	4	*No ISDU data to send.* Invoke ISDUMsg.req(SendWMessage = NO).
T20	6	6	*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

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T25	3	1	Return DL_Write.cnf
T26	6	1	Return positive DL_Read confirmation when last segment (EOS) has been received (see clause 7.7.2) In case of low energy W-Device: Invoke MCmd.req(PreDLink) to switch low energy W-Device to PreDownLink
T27	3	6	-

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 102 in [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
EOS sent	Guard	At least one EOS is sent but no acknowledge received

7.4.3 State machine of the W-Device ISDU handler

Figure 60 shows the state machine of the W-Device ISDU handler.

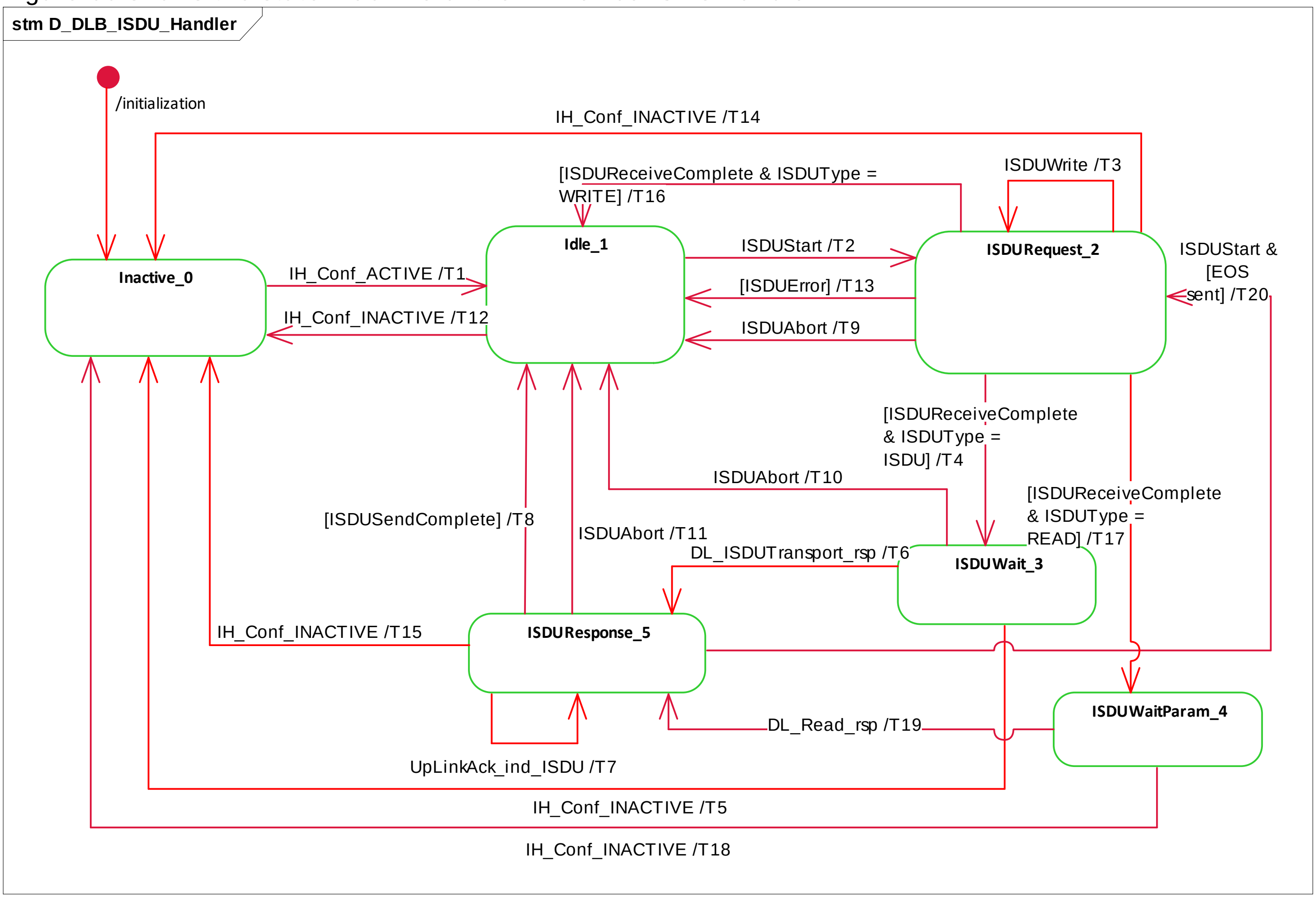


Figure 60 State machine of the W-Device ISDU handler

Table 73 shows the state transition tables of the W-Device ISDU handler.

Table 73 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 67, 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 7.7.3 “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 7.7.3 “Retry-Handling of segmented Data (PD, EV, ISDU)”.

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.*
T20	5	2	*See T2*

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 188) READ (ISDU read with index from Table 188) ISDU (ISDU read or write and no index from Table 188)
EOS sent	Guard	At least one EOS is sent but no acknowledge received

7.4.4 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 [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 74.

Table 74 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

7.5.1 General

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

7.5.2 State machine of the W-Master command handler

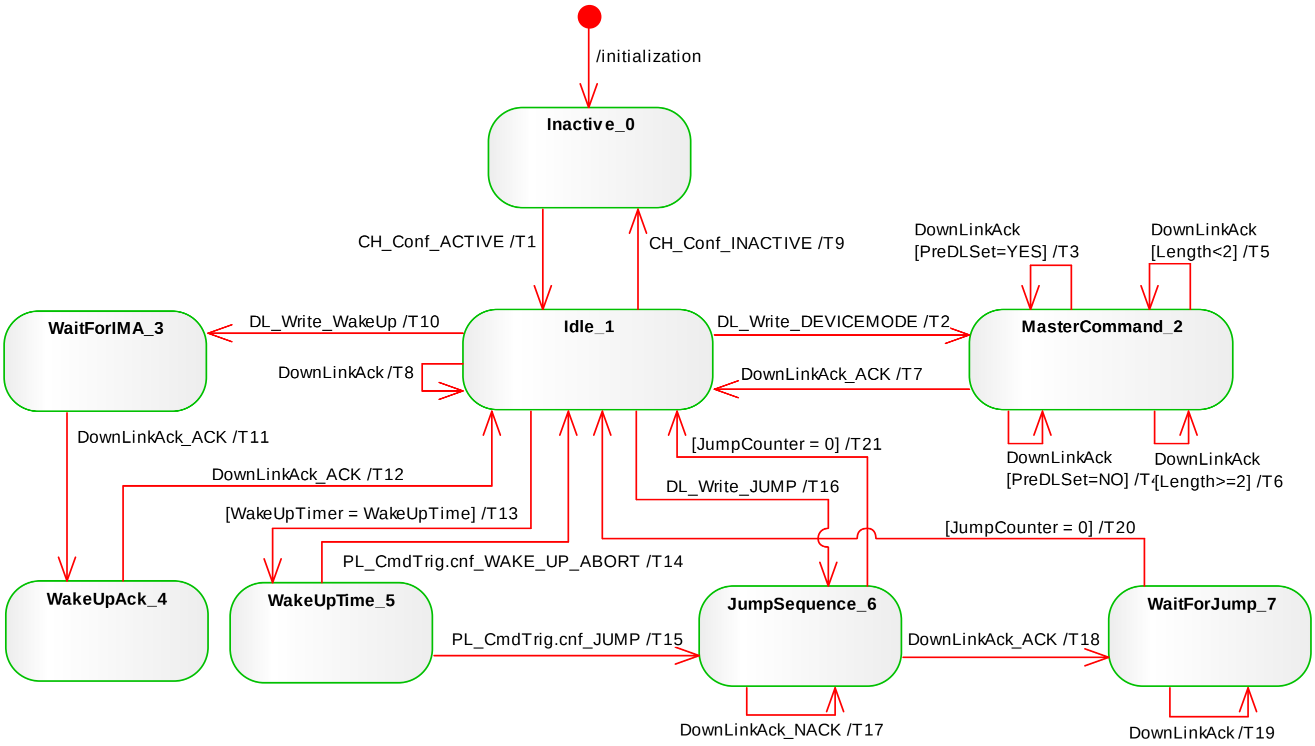


Figure 61 State machine of the W-Master command handler

Table 75 shows the state transition tables of the W-Master command handler.

Table 75 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 66 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.* Invoke MCmd.req(SendWMessage=YES, Broadcast=NO, STARTUP, FullDLink).
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 178.
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 (UnPairing) 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 IO-Link Wireless, additional MasterCommand definitions are available (see Table 178).
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

7.5.3 State machine of the W-Device command handler

Figure 62 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.

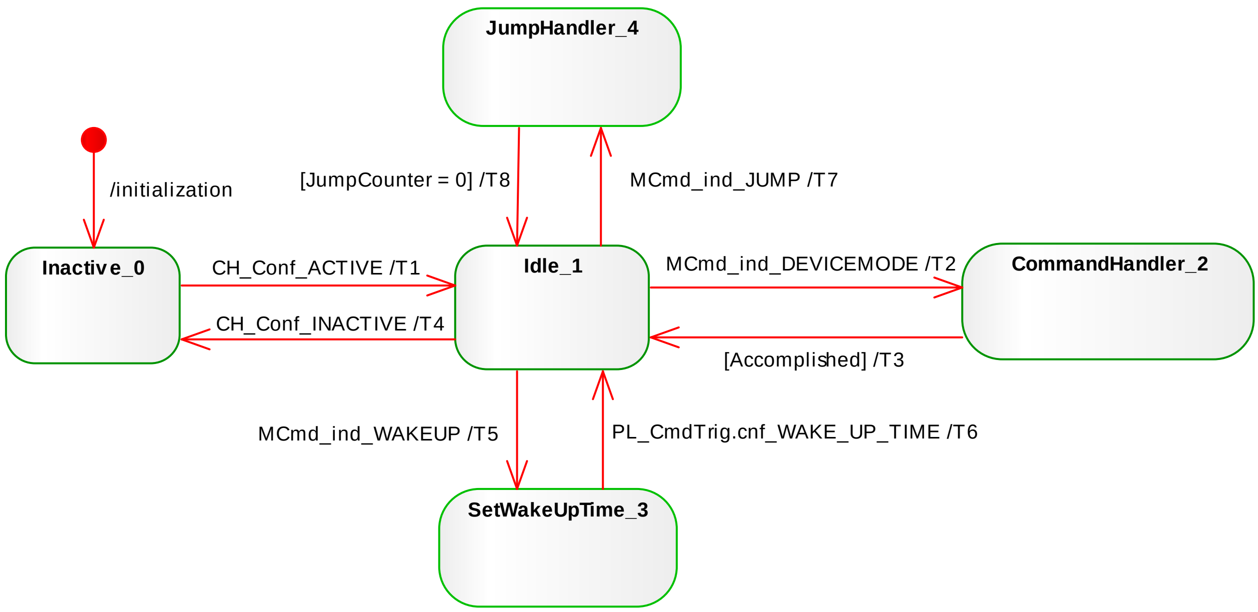


Figure 62 State machine of the W-Device command handler

Table 76 shows the state transition tables of the W-Device Cmd handler.

Table 76 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 178.
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 IO-Link Wireless, additional MasterCommand definitions are available (see Table 178.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

7.6.1 General

An Event transfers a status or an error information.
The Event request is sent from W-Device to W-Master. It is then processed by the W-Master and an Event Confirmation is sent back to the W-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 [1]
EventCodes are specified in Annex D

7.6.2 State machine of the W-Master Event handler

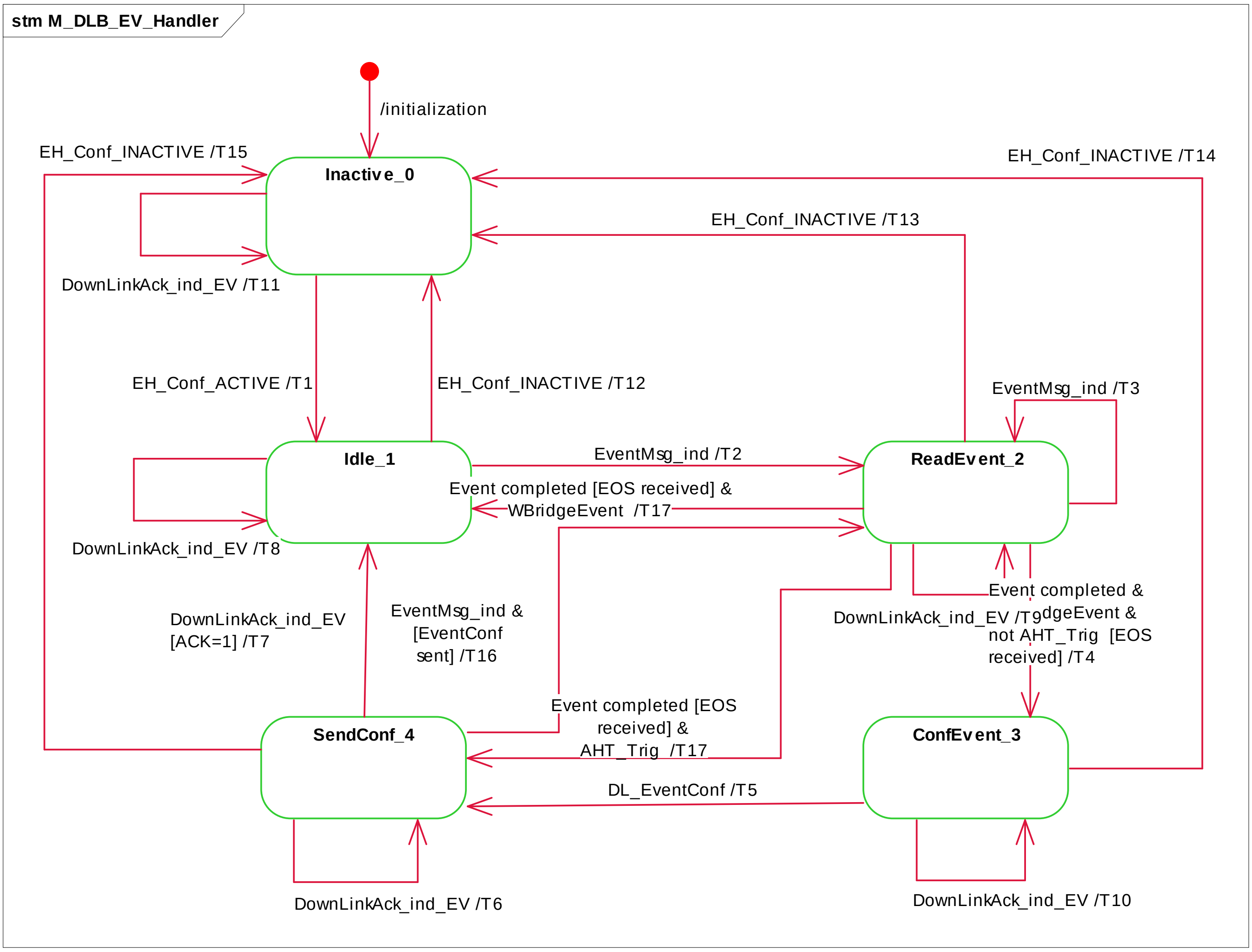


Figure 63 State machine of the W-Master Event handler

Table 77 shows the state transition tables of the W-Master Event handler DL.

Table 77 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 7.7.2). Also, in case of FC = 0x0D (3 octet complete event data) any previously incomplete received data shall be ignored. For Retry-Handling see 7.7.3
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 7.7.3

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 7.7.2 Transmission of segmented data) invoke DL_Event indication to Master AL

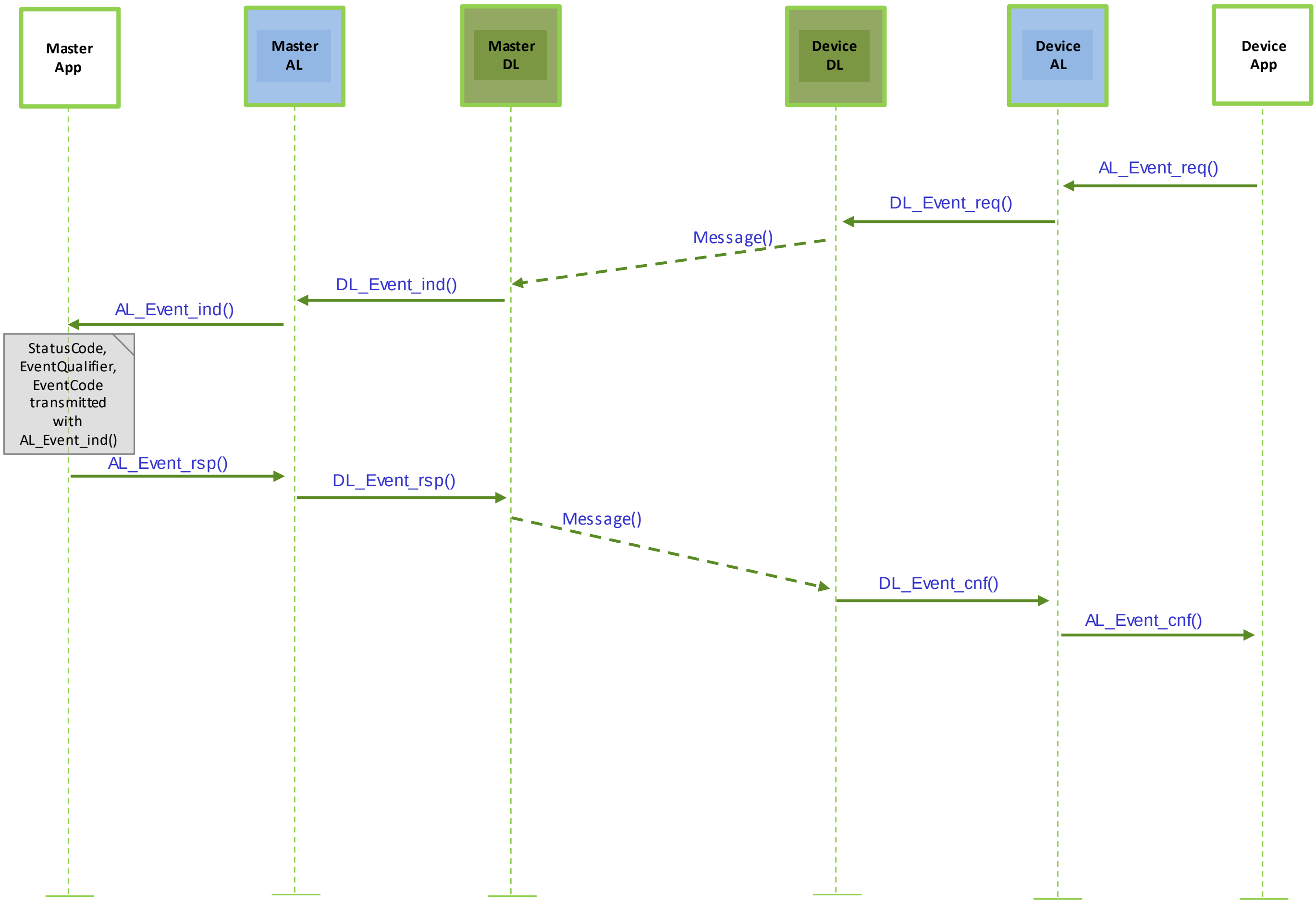
TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T5	3	4	-
T6	4	4	If enough space left in the downlink, invoke EventMsg.rsp(SendWMessage = YES, Slot_N) to deliver Event confirmation to Message handler
T7	4	1	-
T8	1	1	No Event confirmation to send, invoke EventMsg.rsp(SendWMessage = NO).
T9	2	2	No 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	-
T16	4	2	See T2
T17	2	1	After last segment (no data and EOS) has been received (see clause 7.7.1 Transmission of segmented data) invoke DL_Mode.ind(COMLOST)
T18	2	4	Trigger AHT via UpdateDecision for test purpose, see Table 117.

2915

INTERNAL ITEMS	TYPE	DEFINITION
EventConf sent	Guard	At least one Event Confirmation is sent but no acknowledge received
WBridgeEvent	Guard	On receiving Event 0x1800 or 0xFF21 or 0xFF22

2916

2917



2918 **Figure 64 Sequence diagram: Single event scheduling**
2919

7.6.3 State machine of the W-Device Event handler

Figure 65 shows the state machine of the W-Device Event handler.

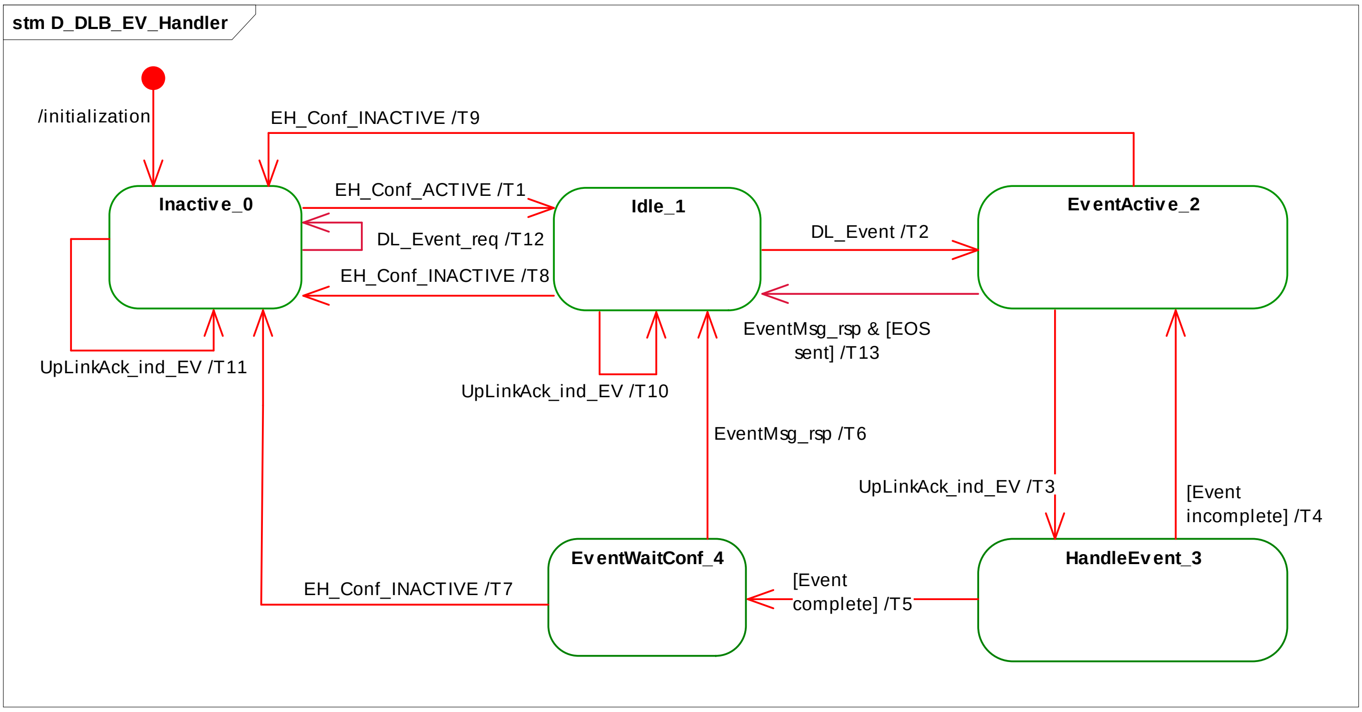


Figure 65 State machine of the W-Device Event handler

Table 78 shows the state transition tables of the W-Device Event handler.

Table 78 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 67, 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 7.7.2.
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)
T13	2	1	See T6

INTERNAL ITEMS	TYPE	DEFINITION
Event complete	Bool	EOS without data sent and acknowledged
Event incomplete	Bool	Data not completely sent
EOS sent	Guard	At least one EOS is sent but no acknowledge received

7.7 Transmission of segmented data and retry handling

7.7.1 General

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) shall generate its own Flow Control considering the acknowledge of the last sent W-Message.

7.7.2 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 55, 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 79).

- 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 7.7.3.
- 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 79 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 1:
A MasterCommand as well as an Event acknowledge does not need segmentation, since this W-Messages are transmitted without data (see Table 46).

Table 79 Flow Control for segmented data

FlowControl (FC)			
FlowControl (FC)	Definition		
0x00 to 0x07	COUNT Counter within a data segment. Increments beginning with 1 after a START. Jumps back from 7 to 0 in case of an overflow.		
0x08	START Start of a segment, i.e., start of an ISDU- request or a response. For the start of a request, any previously incomplete service may be rejected.		
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 2	Definition for ULink Control Octet See NOTE 3	
		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 2:
The DLink Control Octet (see Figure 121) contains a separate field to transmit the length of data. Therefore, these values are unused.

NOTE 3:
The ULink Control Octet (see Figure 123) is coded by only one octet (reduced overhead). Therefore, the DataLength is coded within the Flow Control.

Additionally see A.6. Example for DLink data transmission and A.7 Examples for ULink data transmission for data transmission examples.

7.7.3 Retry-Handling

7.7.3.1 General

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 5.6.2.9.

7.7.3.2 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 6.4.4 and 6.4.5.).

7.7.3.3 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 an 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 99 shows an overview of the structure and services of the W-Master Application Layer(AL).
Figure 95 shows an overview of the structure and services of the W-Device Application Layer (AL).

8.2 Application Layer services

8.2.1 General

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 80 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 80 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_PDCycle	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)

8.2.2 AL_Read (W-Master and W-Device)

The AL_Read service is used to read ISDU Data from an IO-Link Wireless W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 81.

Table 81 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 IO-Link Wireless specific indexes the entire set of W-Parameter (see W-Parameter for IO-Link Wireless in Table 179).
Permitted values: 0 to 65535 (Figure 139 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 B.8, see Annex C in [1]
NOTE: The AL maps DL ErrorInfos into its own AL ErrorInfos using Annex C in [1]

8.2.3 AL_Write (W-Master and W-Device)

The AL_Write service is used to write ISDU Data to an IO-Link Wireless W-Device connected to a specific W-Port. The parameters of the service primitives are listed in Table 82.

Table 82 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 188 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 always returns 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 B.8, see IO-Link Interface and System Specification Annex C in [1]

8.2.4 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 83.

Table 83 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

8.2.5 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 84.

Table 84 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

8.2.6 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 85

Table 85 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)

8.2.7 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 86.

Table 86 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)

8.2.8 AL_PDCycle (W-Master and W-Device)

The AL_PDCycle service indicates the end of a WMasterCycleTime (See C.4.12) period after each start of Process Data transmission . 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 87.

Table 87 AL_PDCycle

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) PDCycle can be used to get an equidistant time between reception of first data packet and activation of PDCycle.

8.2.9 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 88.

Table 88 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)

8.2.10 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 89.

Table 89 AL_NewOutput

Parameter name	.ind
<None>	

8.2.11 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 90.

Table 90 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)

8.2.12 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 91.

Table 91 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 150, see Table A.17 in [1])

Mode

This parameter indicates the Event mode. Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table 153, see Table A.20 in [1])

Type

This parameter indicates the Event category. Permitted values: ERROR, WARNING, NOTIFICATION (see Table 196, see Table A.19 in [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 Annex D and see Annex D in [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)

8.2.13 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 92.

Table 92 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 149).
PDOUT_INVALID (Output Process Data invalid, see Table 149).

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 66 shows the state machine for the handling of ISDU Data within the Application Layer. "AL_Service" represents any AL service in Table 80 related to ISDU. "W-Portx" indicates a particular W-Port number

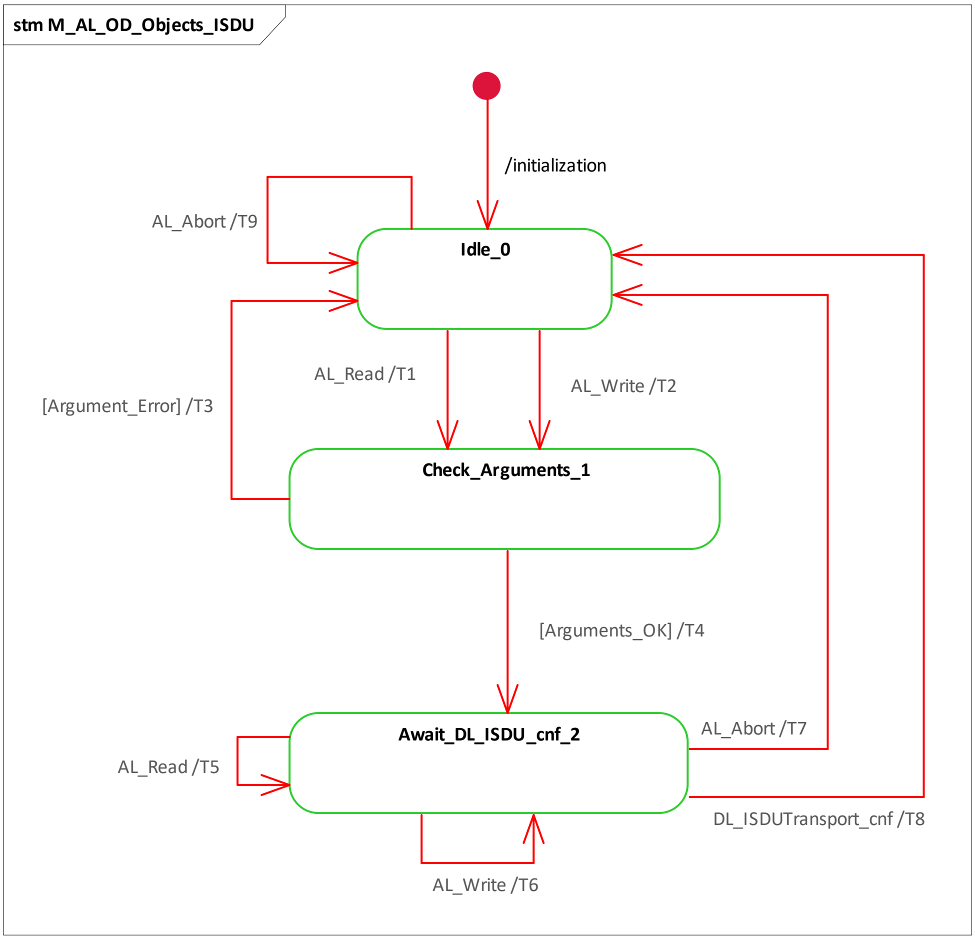


Figure 66 ISDU state machine of the W-Master AL

Table 93 shows the state transition tables of the ISDU W-Master AL.

Table 93 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 99) 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 67 shows the state machine for the handling of ISDU Data within the Application Layer of a W-Device.

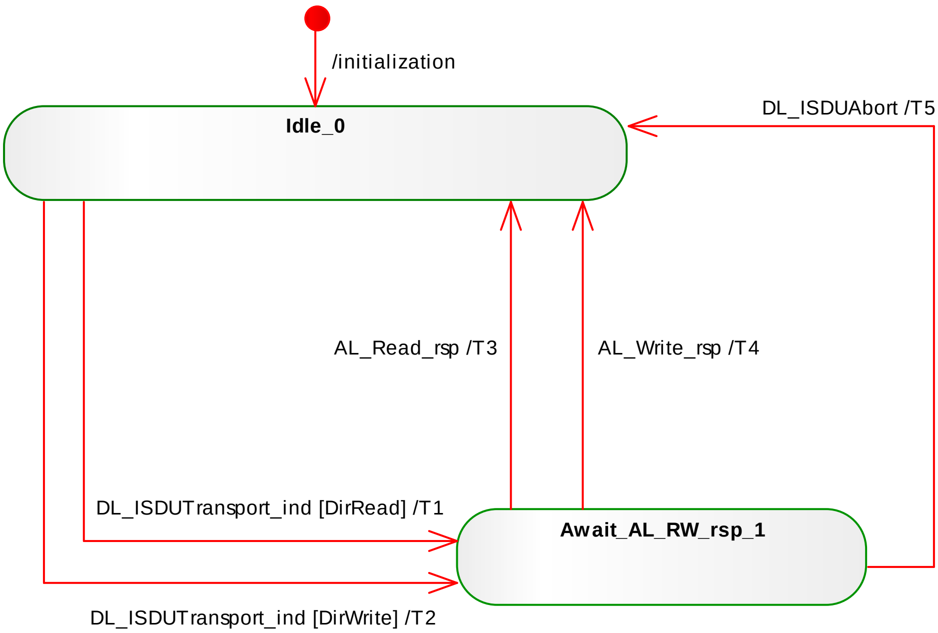


Figure 67 ISDU state machine of the W-Device AL

Table 94 shows the state transition tables of the ISDU W-Device AL.

Table 94 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 68 through Figure 71 demonstrate complete interactions between W-Master and W-Device for several ISDU Data exchange use cases.

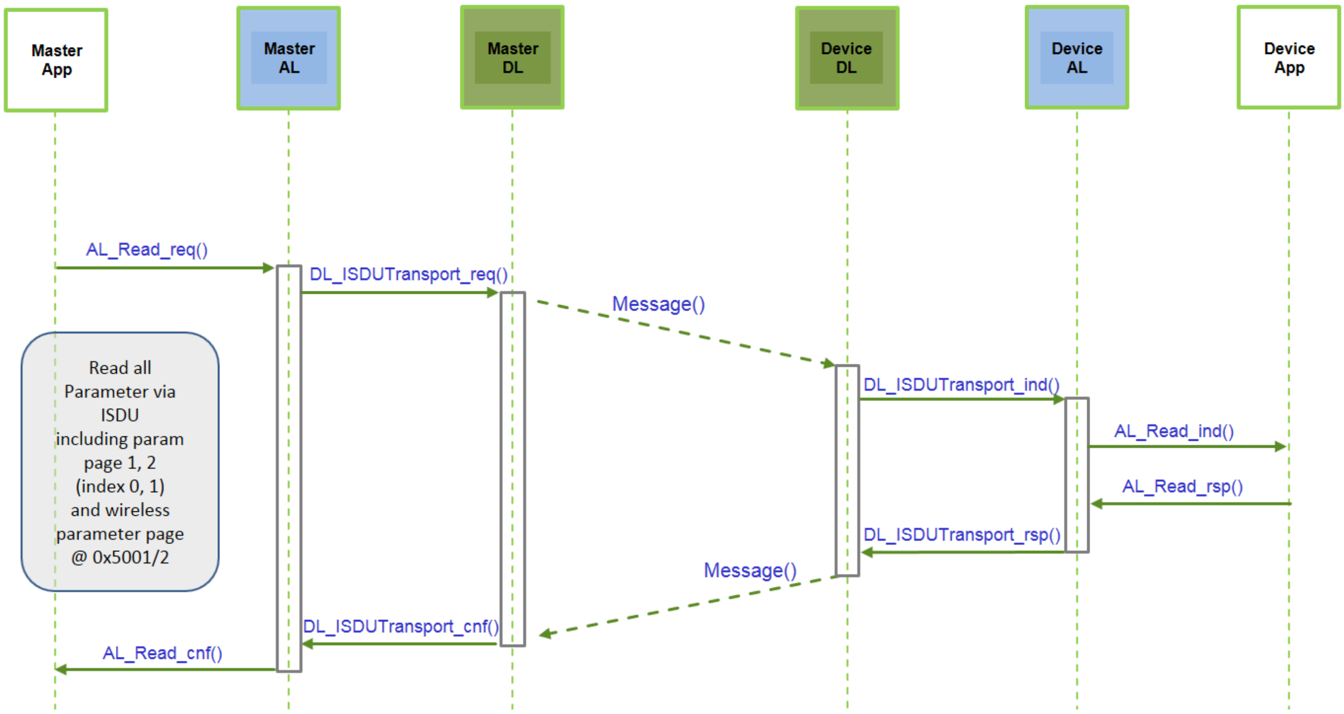


Figure 68 Sequence diagram: ISDU Read Data

Figure 69 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 10.8.5 in [1]).

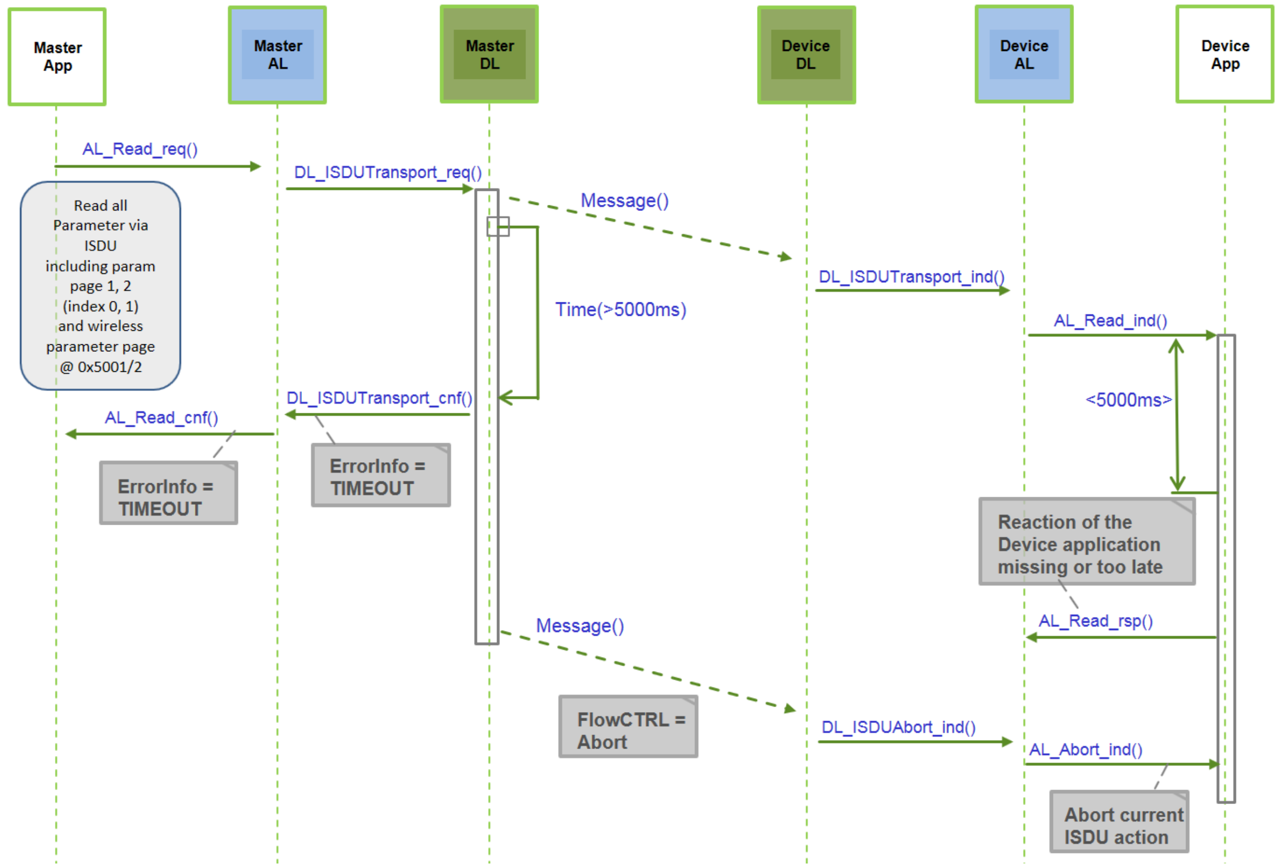


Figure 69 Sequence diagram: ISDU read Data in case of timeout

Figure 70 demonstrates the behavior of ISDU Data exchange in case of an error such as requested Index not available (see Table C.1 in [1]).

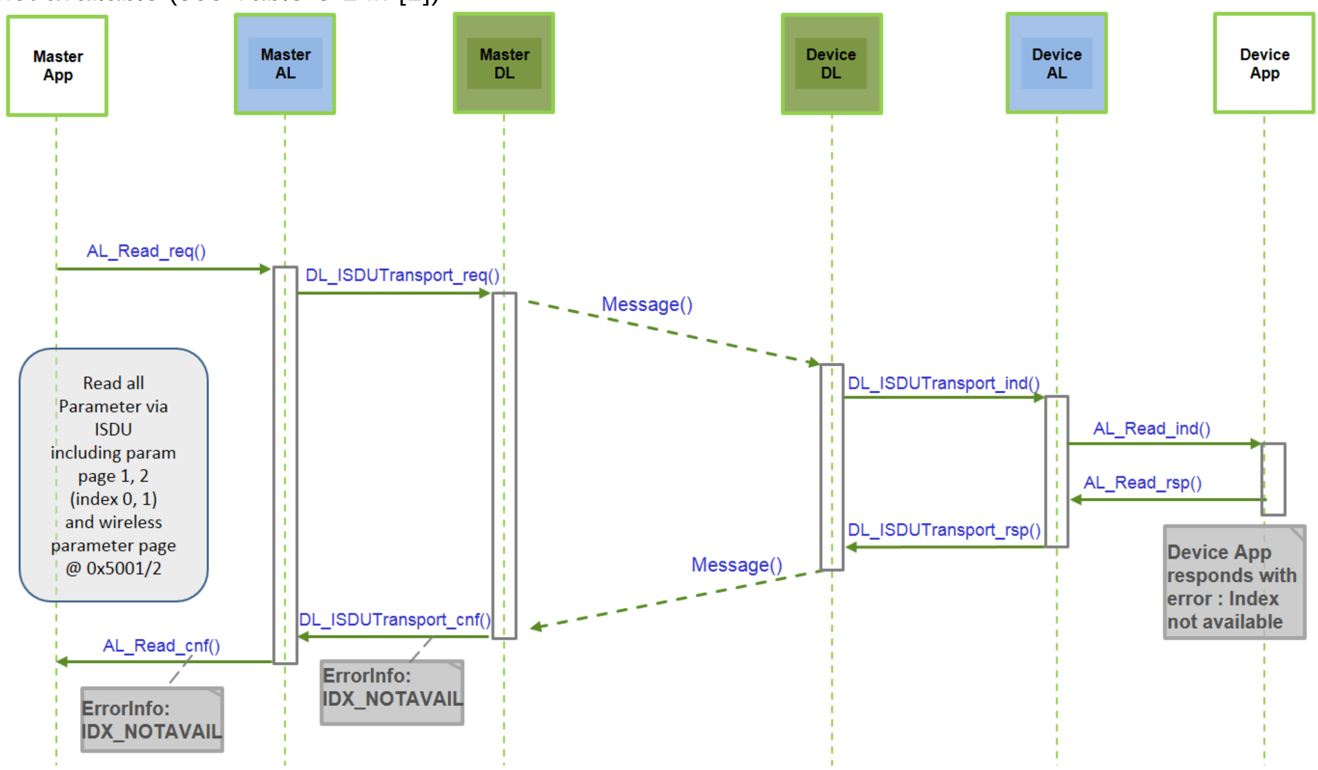


Figure 70 Sequence diagram: ISDU read Data in case of error

Figure 71 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 W-Device with LowPowerDevice attribute activated, the W-Master shall send a MasterCommand to switch the low energy W-Device to listen to the Full-Downlink packet 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 packet part.

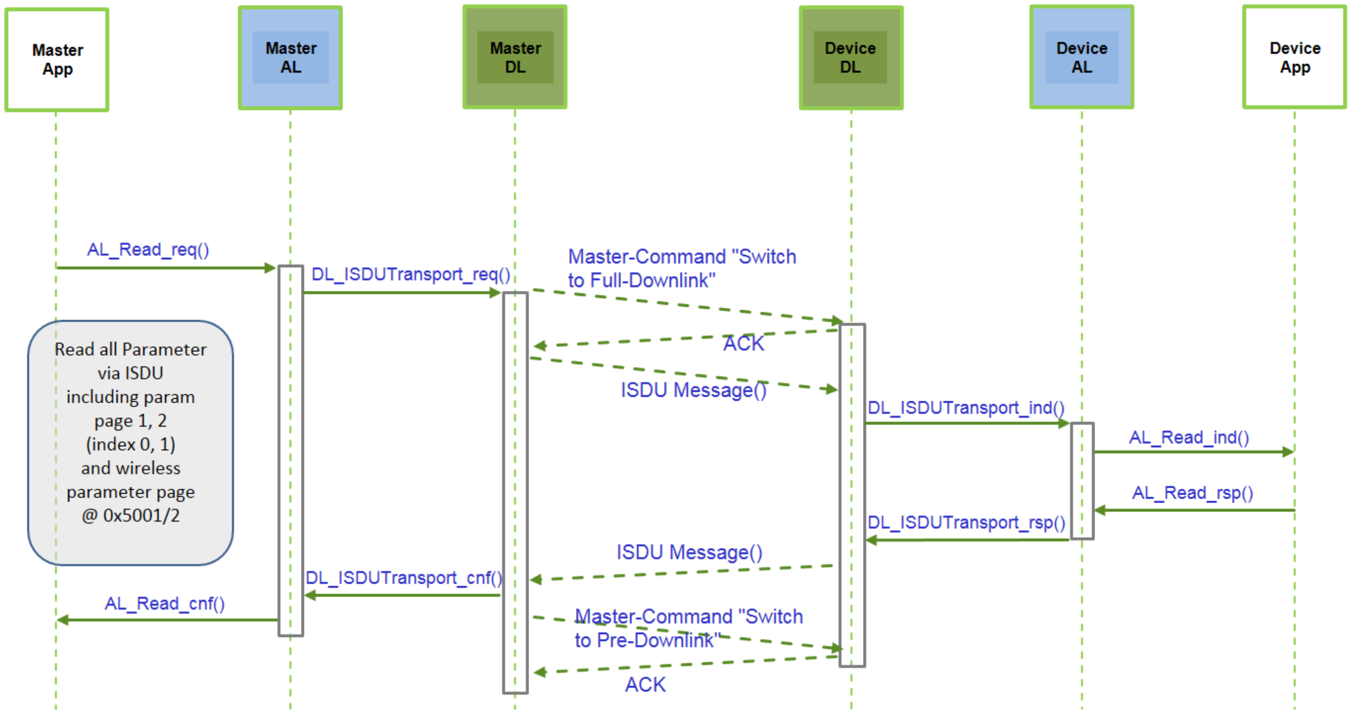


Figure 71 Sequence diagram for low energy W-Devices: ISDU Data

8.3.3 Event processing

8.3.3.1 Event state machine of the W-Master AL

Figure 72 shows the Event state machine of the W-Master Application Layer.

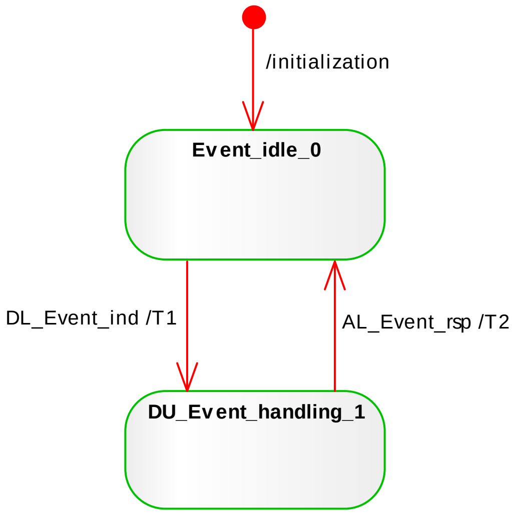


Figure 72 Event state machine of the W-Master AL

Table 95 shows the state transition tables of the Event W-Master AL.

Table 95 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 11.6) 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 73 shows the Event state machine of the W-Device Application Layer

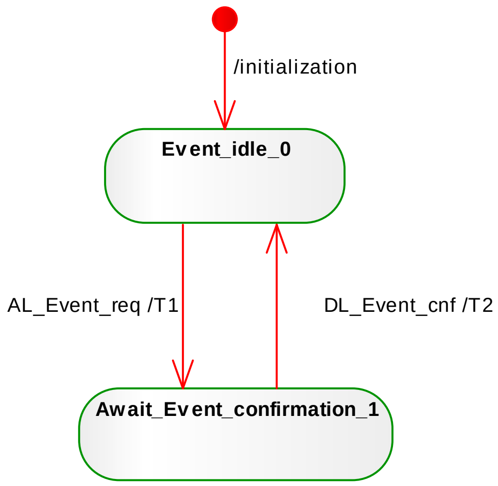


Figure 73 Event state machine of the W-Device AL

Table 96 shows the state transition tables of the Event W-Device AL.

Table 96 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 74 shows how a single Event from a W-Device is processed, in accordance with the relevant state machines.

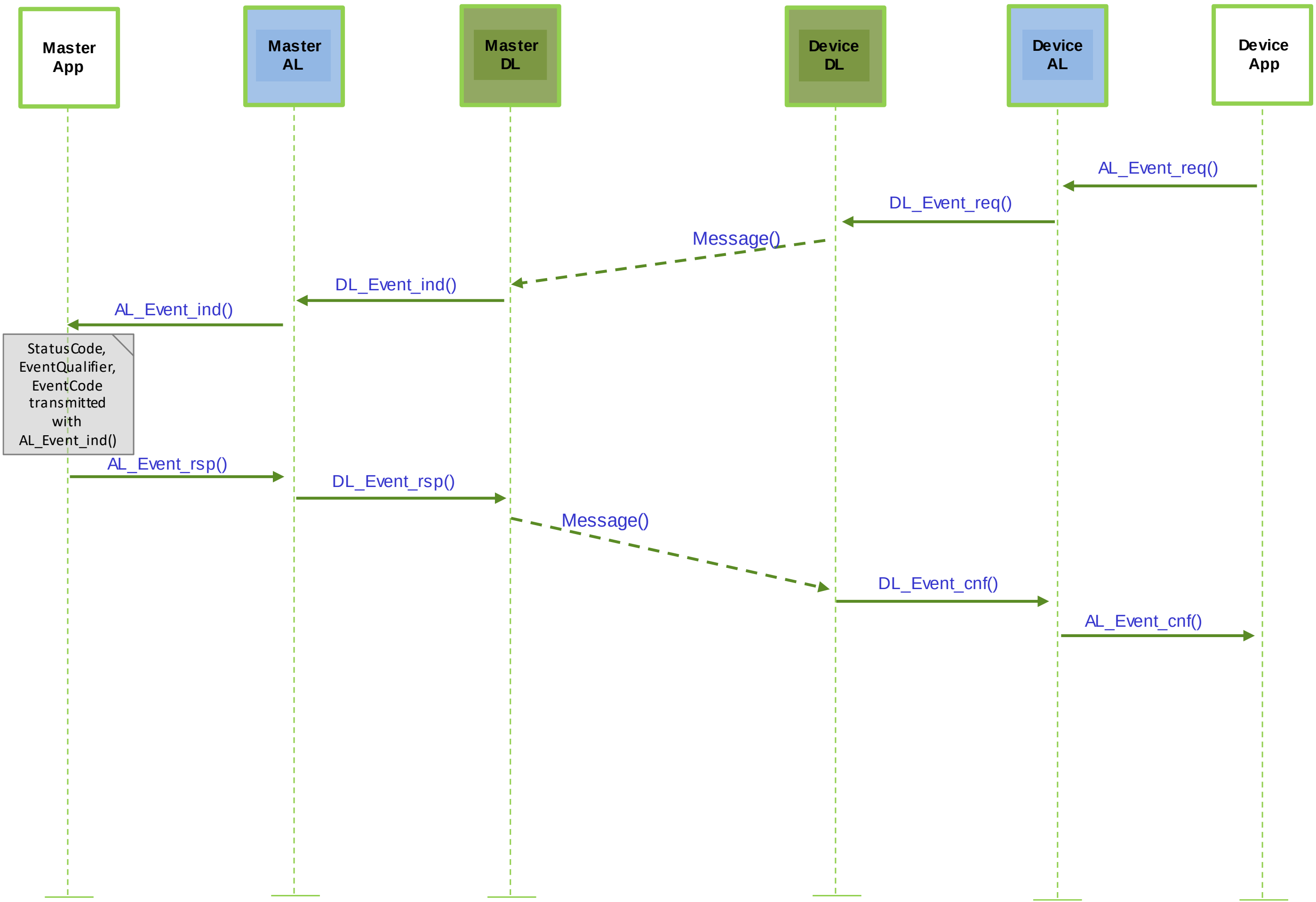


Figure 74 Sequence diagram: Single Event scheduling

8.3.4 Process Data transfer

8.3.4.1 Process Data (PD) state machine of the W-Master-AL

Figure 75 shows the Process Data state machine of the W-Master Application Layer

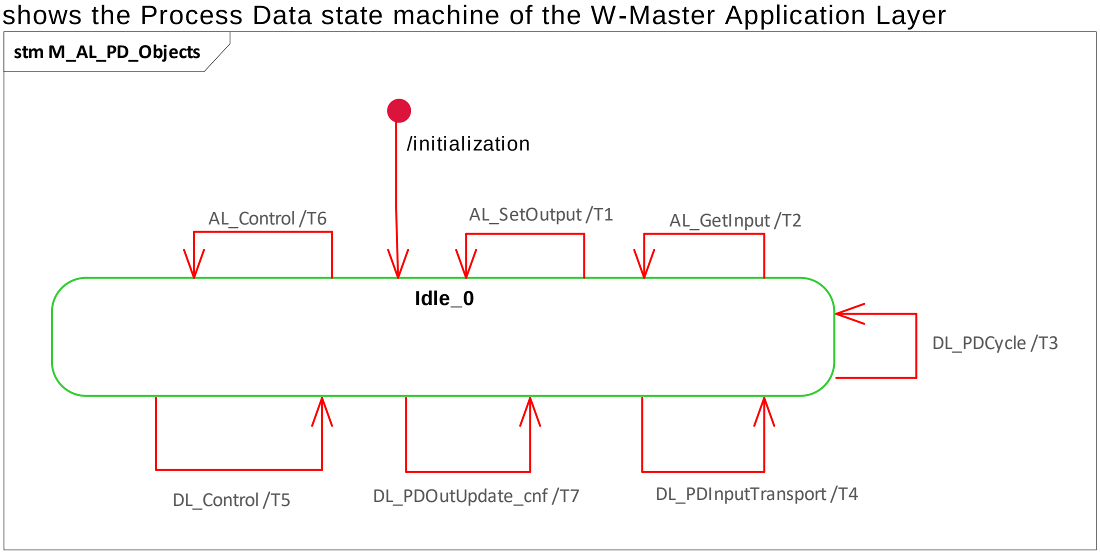


Figure 75 PD state machine of the W-Master AL

Table 97 shows the state transition tables of the PD W-Master AL.

Table 97 State transition tables of the PD W-Master AL

STATE NAME	STATE DESCRIPTION
Idle_0	The W-Master AL is waiting on subordinated AL and DL service calls.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	0	Invoke DL_PDOutputUpdate with Process Data In from AL.
T2	0	0	Read Process Data In.
T3	0	0	Invoke AL_PDCycle.
T4	0	0	DL_PDInputTransport delivers Process Data In from DL. Invoke AL_NewInput.
T5	0	0	Invoke AL_Control with Process Data In qualifier status from DL.
T6	0	0	Invoke DL_Control with Process Data Out qualifier status from AL.
T7	0	0	Invoke AL_SetOutput.cnf

8.3.4.2 Process Data (PD) state machine of the W-Device-AL

Figure 76 shows the Process Data state machine of the W-Device Application Layer

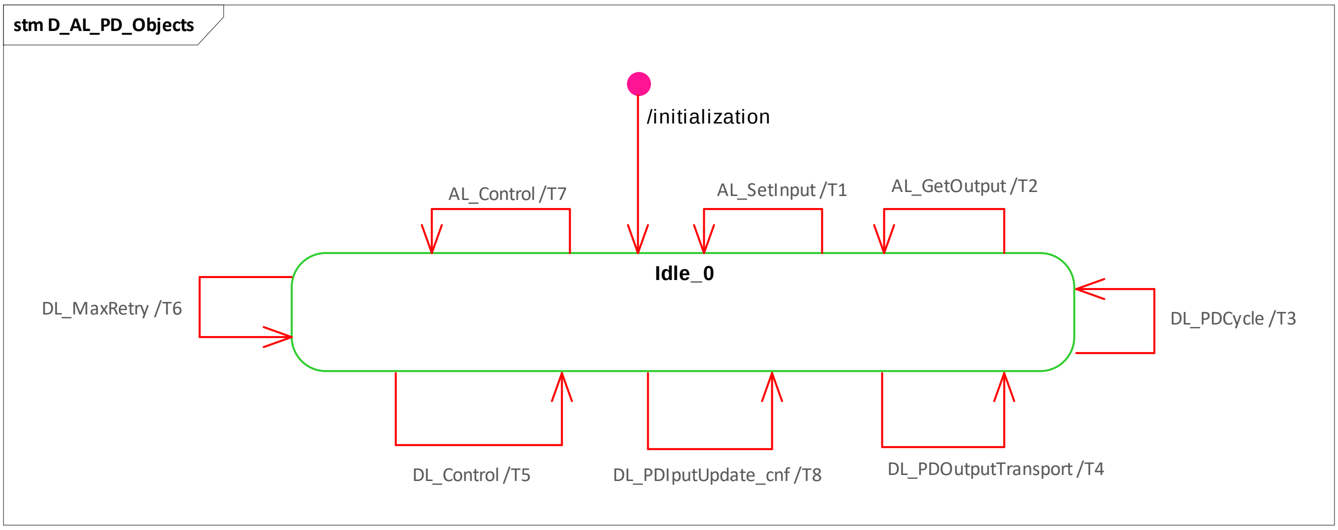


Figure 76 PD state machine of the W-Device-AL

Table 98 shows the state transition tables of the PD W-Device AL.

Table 98 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_PDCycle.
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

8.3.4.3 Process Data cycles

Figure 77 and Figure 78 demonstrate complete interactions between W-Master and W-Device for output and input Process Data use cases.
Figure 55 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.

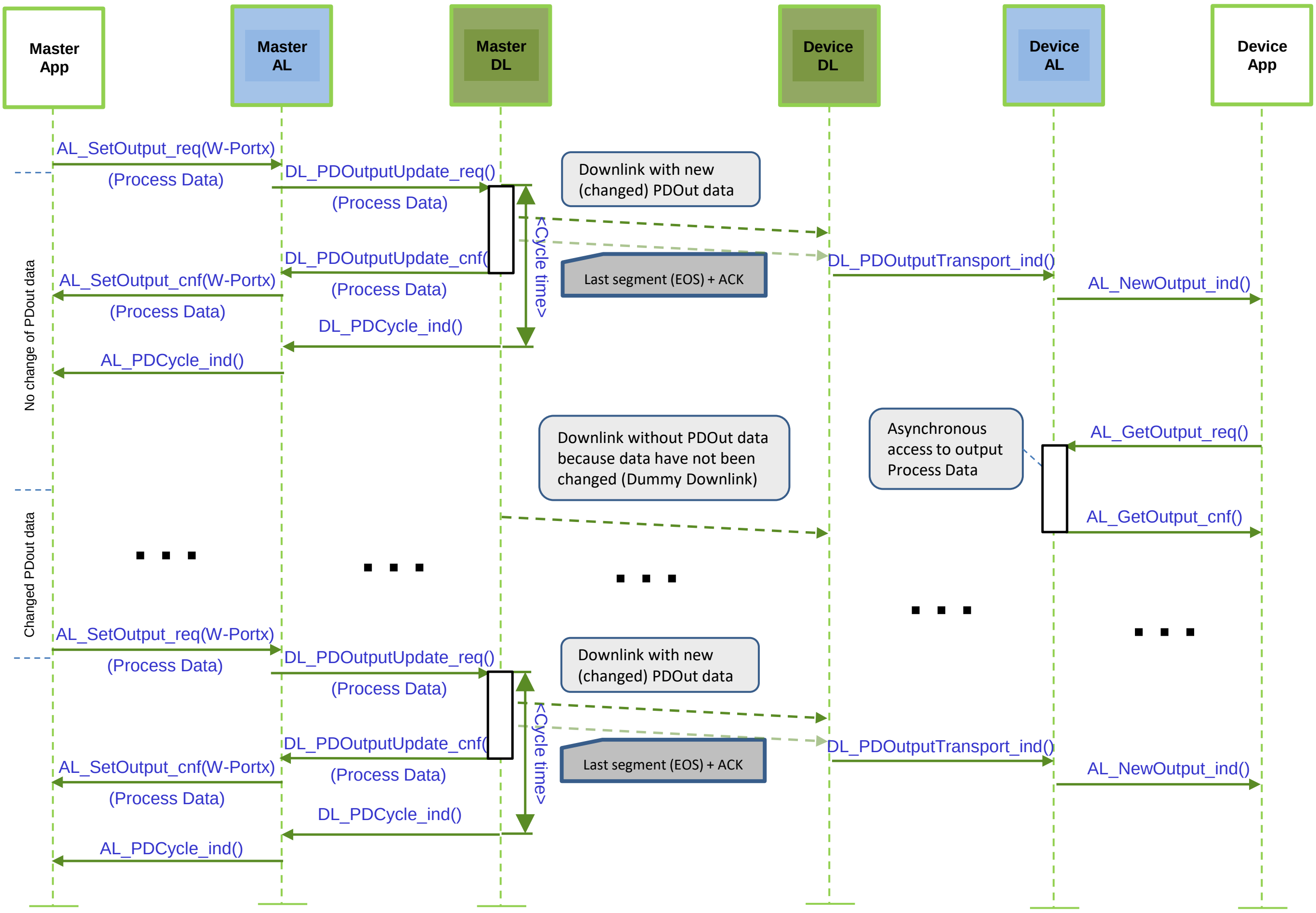


Figure 77 Sequence diagram for output Process Data

Figure 78 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.

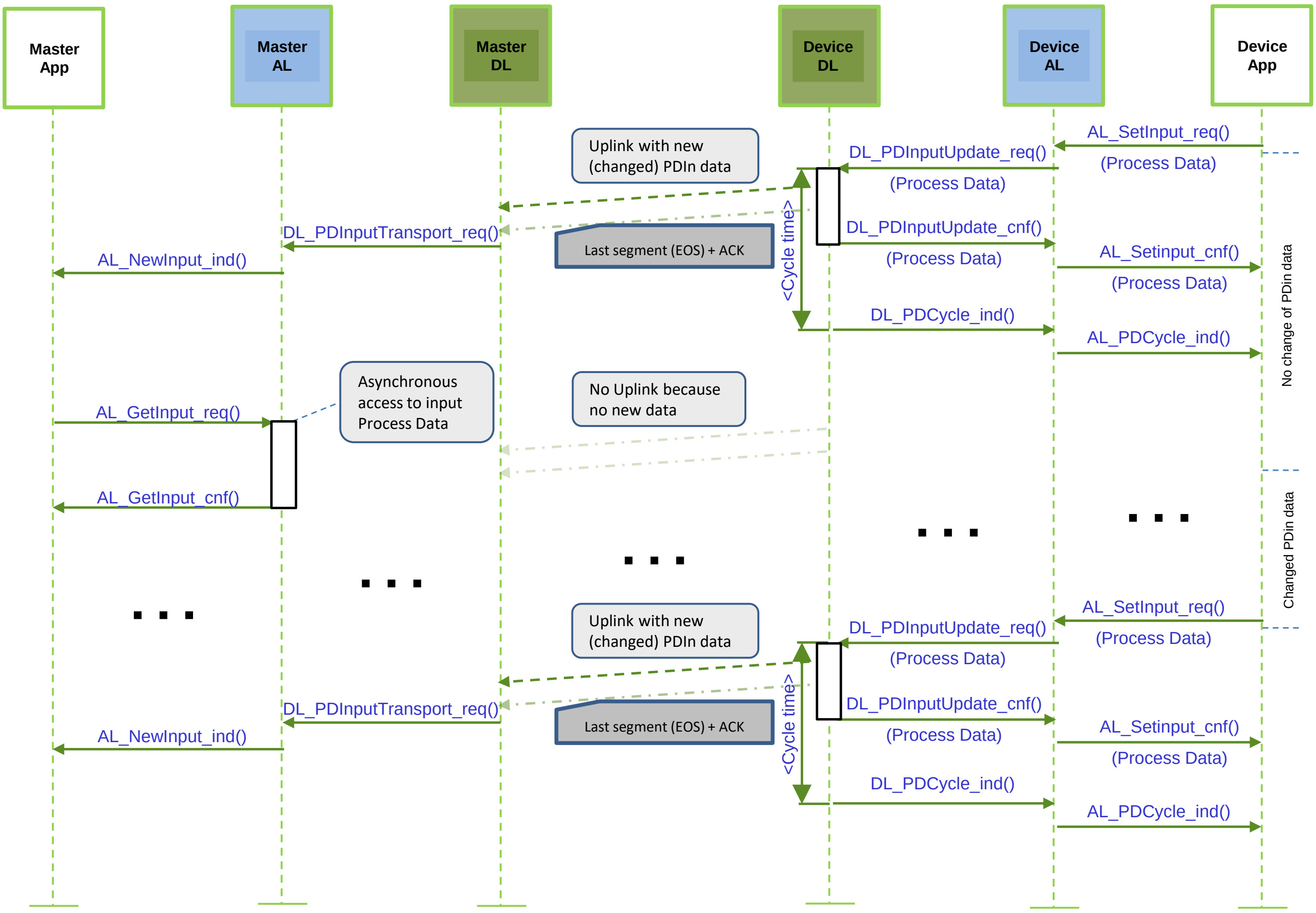


Figure 78 Sequence diagram for input Process Data

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.

NOTE: Some of the functionality described in this clause is implemented within the underlying PL, such as the handling of the W-Messages.

The following subclauses describe the possible operational modes and associated procedures.

9.2 Modes

9.2.1 Service Mode

When a W-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.

9.2.2 Cyclic Mode

9.2.2.1 General

In Cyclic Mode, the W-Master W-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 W-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 W-Track in this mode.

9.2.2.2 IMATime monitoring

The IMATime is continuously supervised within the PL. The IMATime is transferred within the W-Parameter 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 D.2) 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.
The allowed range of the IMATime shall be described in the IODD of the W-Device.

9.2.2.3 WCycleTime calculation

The minimum values for WMasterCycleTimeIn and WMasterCycleTimeOut shall be calculated according to following equations:

$$WMasterCycleTimeIn \geq \left\lceil \frac{PDInLength}{SlotType\_MaxPDLength} \right\rceil * \left\lceil \frac{MaxRetry + 1}{3} \right\rceil * 5ms$$

Equation 9 Calculation of WMasterCycleTimeIn

$$WMasterCycleTimeOut \geq \left\lceil \frac{PDOutLength}{MaxPDSegLength} \right\rceil * \left\lceil \frac{MaxRetry + 1}{3} \right\rceil * 5ms$$

Equation 10 Calculation of WMasterCycleTimeOut

Where SlotType_MaxPDLength is defined by the SlotType:
SSlot: SlotType_MaxPDLength = 1
DSlot: SlotType_MaxPDLength = 14

9.3 System management of the W-Master

9.3.1 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 99.

- SM_SetMasterConfig sets the common configuration of the W-Master for all W-Tracks.
- SM_SetTrackMode sets the mode of a W-Track.
- SM_GetTrackMode gets the mode of a W-Track.
- SM_TrackScanEnd indicates the end of the scan mode.
- SM_TrackScanResult reports a new unpaired W-Device within the W-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.

The Configuration Manager in a W-Master consists of Master Configuration Manager, W-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, BlockList). 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 W-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 79 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 79 represent response services to an initial service.

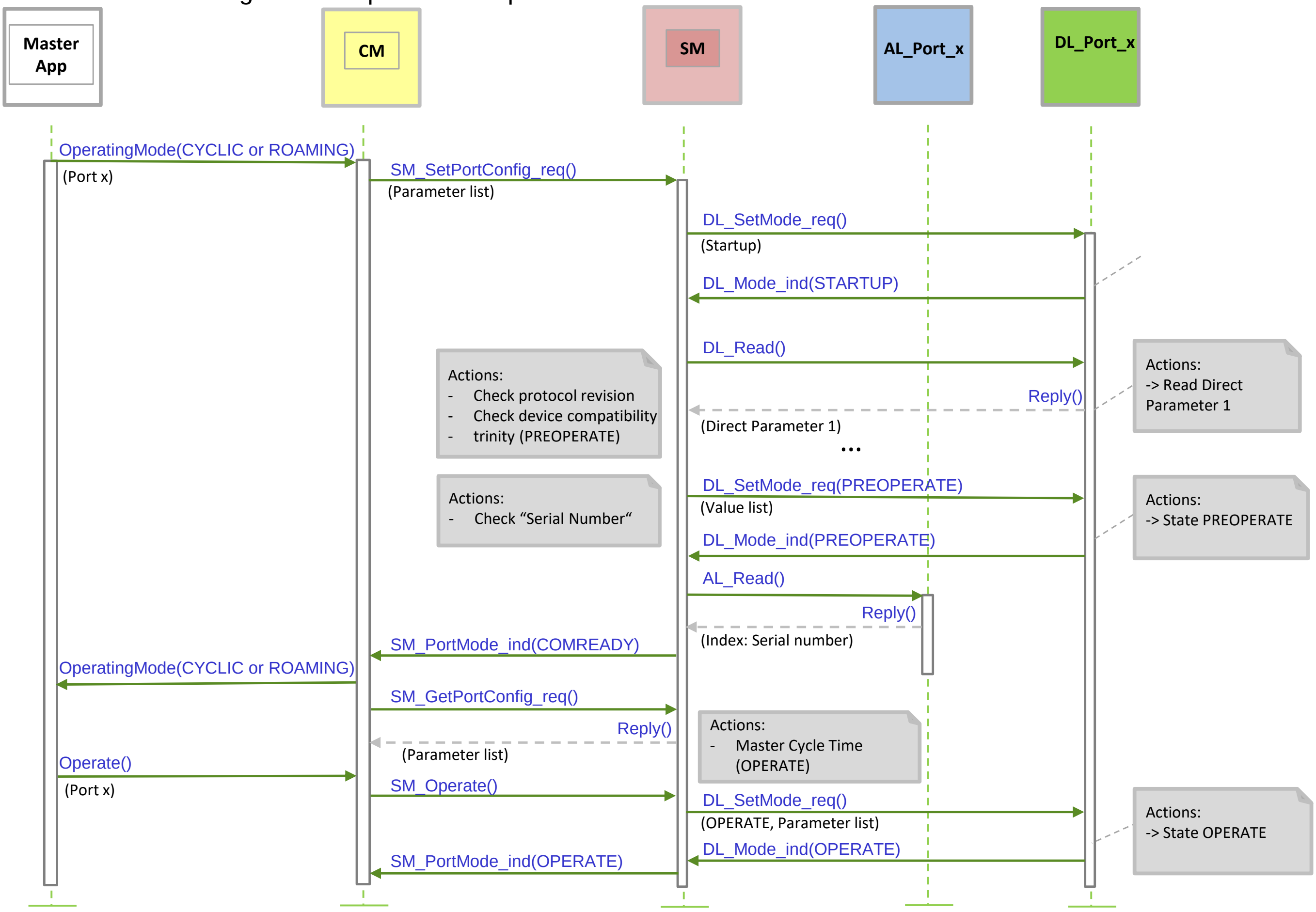


Figure 79 Sequence chart of the use case “port x setup”

9.3.2 System management W-Master services

9.3.2.1 Overview

System management provides the SM W-Master services to the user via its upper interface. Table 99 lists the assignment of the W-Master to its role as initiator or receiver for the individual SM services.

Table 99 SM services within the W-Master

Service Name		**W-Master**
SM_SetMasterConfig	Set common configuration of the W-Master for all W-Tracks	R
SM_SetTrackMode	Set mode of a W-Track	R
SM_GetTrackMode	Get mode of a W-Track	R
SM_TrackScanResult	Report a new unpaired W-Device within the W-Track's proximity to the application	I
SM_TrackScanEnd	Indicates the end of the scan mode	I
SM_SetPortConfig	Set configuration of a virtual W-Port	R
SM_GetPortConfig	Get configuration of a virtual W-Port	R
SM_PortPairing	Pair W-Device to W-Master	R
SM_PortMode	Reports the mode of a W-Port	I
SM_GetPortQuality	Acquire quality of a W-Device connection	R
SM_Operate	Activate a W-Port	R
Key (see 3.3.5) I: Initiator of service R: Receiver (Responder) of service		

9.3.2.2 SM_SetMasterConfig

The SM_SetMasterConfig service is used to set up the W-Master configuration. This configuration is used for all W-Tracks. The parameters of the service primitives are listed in Table 100

Table 100 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 parameters of a W-Master.
Parameter type: Record
Record Elements:

MasterID
This parameter contains the MasterID of the W-Master (see Table 178)
Permitted values: 1 to 29

BlockList
This parameter contains the frequency channels which shall not be used by the W-Master. For details see H.1.

SyncMaster:
This parameter specifies the W-Track number which shall run as W-Frame synchronization master (see 5.6.2.2. Parameter "SyncTrack" in service PL_SetTrackConfig)
Permitted values:
0 (SyncMaster is W-Track 0)
...
4 (SyncMaster is W-Track 4)
5 (all W-Tracks using an external synchronization signal)

AHTEnable:
This parameter contains 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)

9.3.2.3 SM_SetTrackMode

The SM_SetTrackMode service is used to set up one W-Track with the requested W-Track configuration. The parameters of the service primitives are listed in Table 101.

Table 101 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 W-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 W-Track

Permitted values: 1 to 255 (see Table 186).

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)

9.3.2.4 SM_GetTrackMode

The SM_GetTrackMode service is used to read the W-Track configuration from the system management. The parameters of the service primitives are listed in Table 102.

Table 102 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 W-Track parameters of a W-Master W-Track.

Parameter type: Record

Record Elements:

TrackMode

This parameter indicates the actual operational mode of the W-Track

Permitted values: STOP, CYCLIC, SCAN, PAIRING, ROAMING

TxPower

This parameter contains the actual transmit power level of the W-Track

Permitted values: 1 to 255 (see Table 186)

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)

9.3.2.5 SM_TrackScanResult

The SM_TrackScanResult service is used to report a new unpaired W-Device within the W-Track's proximity to the application. This is only done if the W-Track is in ROAMING or SCAN mode. The parameters of the service primitives are listed in Table 103.

Table 103 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:
Default type of the W-Device in Uplink packet given through W-Device application.
Permitted values: SSLOT, DSLOT (see Table 175).

UniqueID:
This parameter indicates the UniqueID of the W-Device. (see Figure 140)

RevisionID:
This parameter indicates the protocol version of the found W-Device (see clause B.1.5 in [1]).

9.3.2.6 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 104.

Table 104 SM_TrackScanEnd

Parameter Name	.ind
<none>	

9.3.2.7 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 105.

Table 105 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, 6.2.2.).
Slot_N
This parameter contains the Slot number within the corresponding W-Track number (see TDMapper, 6.2.2)
Track_N
This parameter selects the W-Track number with which the W-Port is assigned to (see TDMapper, 6.2.2.)
SlotType
This parameter contains the SlotType for corresponding W-Device
Permitted values: SSLOT, DSLOT (see Table 175)
TargetMode
This parameter indicates the requested operational mode of the W-Port
Permitted values: INACTIVE, CYCLIC, ROAMING
UniqueID
Data length: 9 octets
ConfiguredCycleTimeOut
This parameter contains the requested cycle time for PDOut for the OPERATE mode
Permitted values: 1 octet, time encoded according to clause C.4.12
ConfiguredCycleTimeIn
This parameter contains the requested cycle time for PDIn for the OPERATE mode
Permitted values: 1 octet, time encoded according to clause C.4.12C.4.1
IMATime
This parameter contains the requested IMA time for the OPERATE mode
Permitted values: 2 octets, time encoded according to clause C.4.4.2.
MaxRetry
This parameter contains the maximum number of retries for a transmission in OPERATE mode
Permitted values: see Table 185.
ConfiguredRevisionID (CRID)
Data length: 1 octet for the RevisionID (see Table 176)
InspectionLevel:
Permitted values: NO_CHECK, TYPE_COMP, IDENTICAL (see Table 106)
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
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 10.10 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 106 specifies the coding of the different InspectionLevel

Table 106 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 107 specifies the coding of the different Target Modes.

Table 107 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)

9.3.2.8 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 108

Table 108 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, 6.2.2.).

Slot_N
This parameter contains the Slot number within the corresponding W-Track number (see TDMapper, 6.2.2.)

Track_N
This parameter selects the W-Track number which the W-Port is assigned to (see TDMapper, 6.2.2.)

SlotType
This parameter contains the SlotType for corresponding W-Device
Permitted values: SSLOT, DSLOT (see Table 175)

TargetMode
This parameter indicates the requested operational mode of the W-Port
Permitted values: INACTIVE, CYCLIC, ROAMING

RealUniqueID
Data length: 9 octets
This parameter contains the UniqueID of the paired W-Device.

RealCycleTimeOut
This parameter contains the real (actual) cycle time for PDOut in OPERATE mode
Permitted values: 1 octet, time encoded according to clause C.4.12

RealCycleTimeIn
This parameter contains the real (actual) cycle time for PDIn in OPERATE mode
Permitted values: 1 octet, time encoded according to clause C.4.12

IMATime
This parameter contains the requested IMA time for the OPERATE mode
Permitted values: 2 octets, time encoded according to clause C.4.4.2.

MaxRetry
This parameter contains the maximum number of retries for a transmission in OPERATE mode
Permitted values: see Table 185.

RealRevision (RRID)
Data length: 1 octet for the RevisionID (see B.1.5 in [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

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 10.10 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)

9.3.2.9 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 109

Table 109 SM_PortPairing

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

W-Port
This parameter contains the W-Port number

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 213 (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

9.3.2.10 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 110.

Table 110 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)
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)
CYCTIME_FAULT (Device does not support the configured cycle time)

9.3.2.11 SM_GetPortQuality

The SM_GetPortQuality service is used to acquire the quality of a W-Master and W-Device connection, see 5.5.6. The parameters of the service primitives are listed in Table 111.

Table 111 SM_GetPortQuality

Parameter Name	.req	.cnf
Argument	M	
W-Port	M	
Result (+)		S
W-Port		M
LQI_M		M
RSSI_M		M
LQI_D		M
RSSI_D		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
LQI_M
This parameter contains the link quality LQI of the W-Master-Port, delivered via service PL_QualityService.
Permitted Values: 0 to 100%, INVALID see Table 206.
RSSI_M
This parameter contains the RSSI of the W-Master-Port, delivered via service PL_QualityService
Permitted Values: -128 to 20, INVALID see Table 206.
LQI_D

This parameter contains the LQI of the W-Device. The value shall be requested via DL_Read (ISDU). In case of low Energy W-Devices the value from PL_QualityService shall be used.
Permitted Values: 0 to 100%, INVALID see Table 206.

RSSI_D
This parameter contains the RSSI of the W-Device. The value shall be requested via DL_Read (ISDU). In case of low Energy W-Devices the value from PL_QualityService shall be used.
Permitted Values: -128 to 20, INVALID see Table 206.

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)

9.3.2.12 SM_Operate.

The SM_Operate service prompts system management to calculate the WMasterCycleTime (See C.4.12) 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 112.

Table 112 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)

9.3.3 SM W-Master protocol

9.3.3.1 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 81.
The corresponding decision logic is demonstrated via activity diagrams (see Figure 83, Figure 84, Figure 85, and Figure 86).

9.3.3.2 SM W-Master State machines

9.3.3.2.1 State Machine of the W-Master W-Track handler

Figure 80 shows the main state machine of the W-Master W-Track handler. The W-Tracks will be configured (MasterID, Blocklist, …) 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.

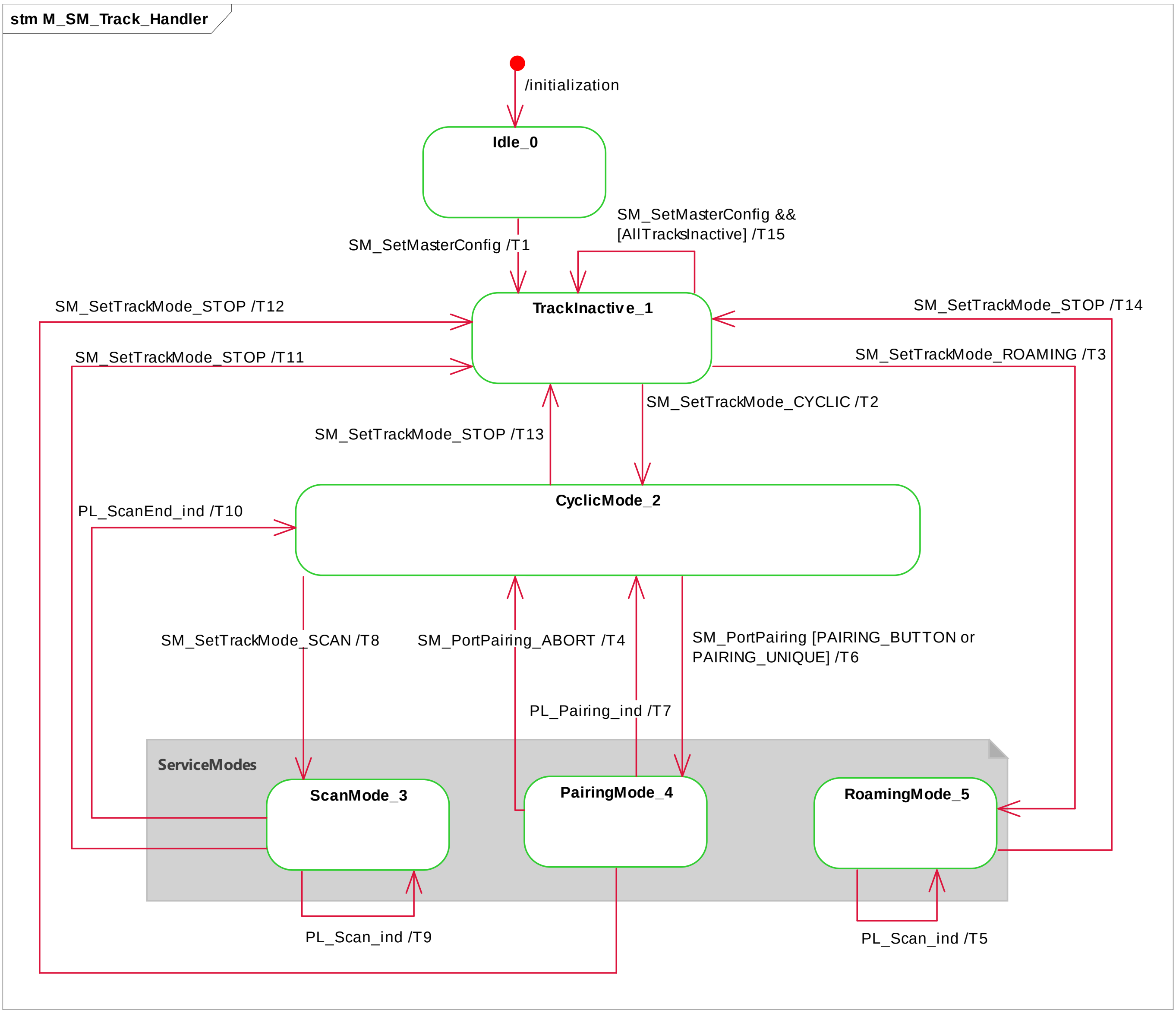


Figure 80 State Machine of the W-Master W-Track handler

Table 113 shows the state transition tables of the W-Master W-Track handler.

Table 113 State transition tables of the W-Master W-Track handler

STATE NAME	STATE DESCRIPTION
Idle_0	-
TrackInactive_1	State is entered after W-Track configuration done via SM_SetMasterConfig. Waiting for activation of operating mode (CYCLIC or ROAMING).
CyclicMode_2	W-Track is active (CYCLIC mode). The gateway application is exchanging Process Data and ready to send or receive On-request Data.
ScanMode_3	W-Track is active (SCAN mode) and scanning for unpaired W-Devices via the configuration channels. Found W-Devices are reported to the application via SM_TrackScanResult.
PairingMode_4	W-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	W-Track is active (ROAMING mode). Additionally, the configuration channels are active. Found W-Devices are reported to the application via SM_TrackScanResult.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_SetTrackConfig() to configure all available W-Master W-Tracks with identical MasterID and Blocklist: PL_SetTrackConfig.req (MasterID, Blocklist, Track_N++, SyncTrack) The W-Tracks shall be configured in the following way: Each W-Track shall get a unique W-Track number from 0 up to 4, (e.g., consecutively numbering → Track_N++) The number given in SM_SetMasterConfig(SyncMaster) selects the W-Track which shall become SyncTrack = YES (except SyncMaster = 5).
T2	1	2	Invoke PL_SetMode (CYCLIC)
T3	1	5	Invoke PL_SetMode (ROAMING)
T4	4	2	Invoke DL_Write.req(WirelessSystemCfg)
T5	5	5	Invoke SM_TrackScanResult to report unpaired W-Devices within the W-Track’s proximity
T6	2	4	-
T7	4	2	-
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

INTERNAL ITEMS	TYPE	DEFINITION
AllTracksInactive	Bool	This value is set if all W-Tracks are in the state TrackInactive_0

9.3.3.2.2 State Machine of the W-Port-handler

Figure 81 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 C.4.7 SerialNumber, since the pairing mechanism covers the identity check of the W-Device.

A W-Bridge shall route the SerialNumber of its connected IO-Link Device to support the SerialNumber check for InspectionLevel in the same way as for a pure IO-Link Device.

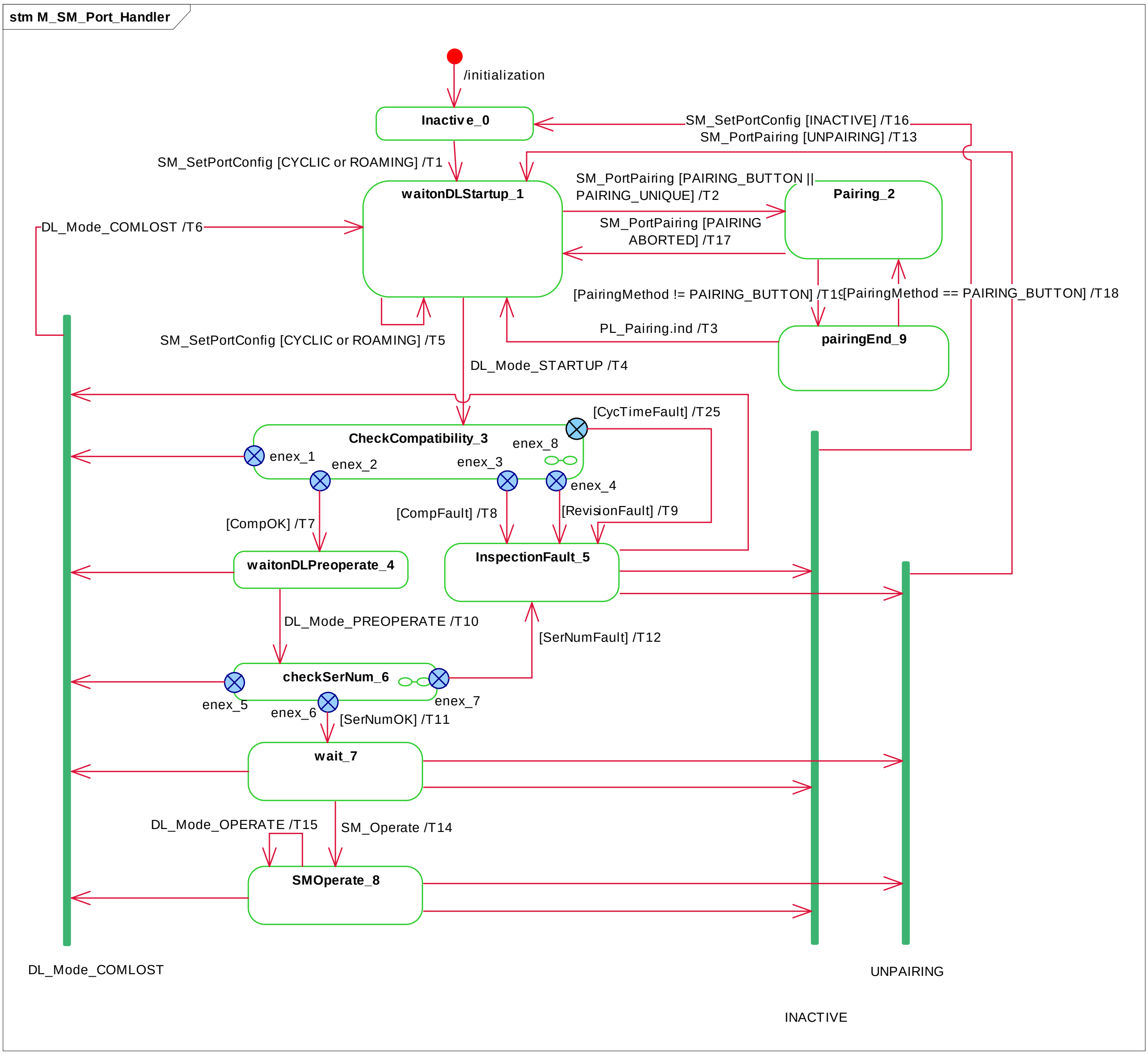


Figure 81 State Machine of the W-Port-handler

Table 114 shows the state transition tables of the Master W-Port-handler.

Table 114 State transition tables 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 82.
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 86.
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.
pairingEnd_9	Wait until pairing completes. Used to check the last pairing process was pairing by button.

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	9	1	Invoke SM_PortMode.ind (PAIRING_SUCCESS or PAIRING_TIMEOUT) If UniqueID after pairing by button is changed from configured UniqueID: invoke EventMsg.ind(IOLW_UniqueID_Changed, LOCAL) to the W-Master application to invoke readback of port configuration.
T4	1	3	VerRetry = 0, CompRetry = 0
T5	1	1	See T1
T6	3,4,5,6,7,8	1	Invoke DL_SetMode.req (STARTUP) 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)
T17	2	1	*Pairing procedure is aborted by the W-Master application*. Invoke PL_Pairing(ABORT)
T18	2	9	*Update Unique ID for W-Track the W-Device shall be paired to.* Invoke PL_SetSlotConfig().
T19	2	9	-
T25	3	5	SM_PortMode.ind(CYCTIME_FAULT), DL_SetMode.req(PREOPERATE, ValueList)

3937

INTERNAL ITEMS	TYPE	DEFINITION
CycTimeFault	Bool	See Figure 84; error variable CYCTIME_FAULT
CompOK	Bool	See Figure 84
CompFault	Bool	See Figure 84; error variable COMP_FAULT
RevisionFault	Bool	See Figure 83; error variable REVISION_FAULT
SerNumFault	Bool	See Figure 87; error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 87
INACTIVE	Variable	A target mode in service SM_SetPortConfig
CYCLIC, ROAMING	Variables	Target Modes in service SM_SetPortConfig
MasterCmd	Service	DL_Write(0x00, 0x01, …)

9.3.3.2.3 SM W-Master submachines

Figure 82 shows the Master W-Port-handler submachine checkCompatibility_3.

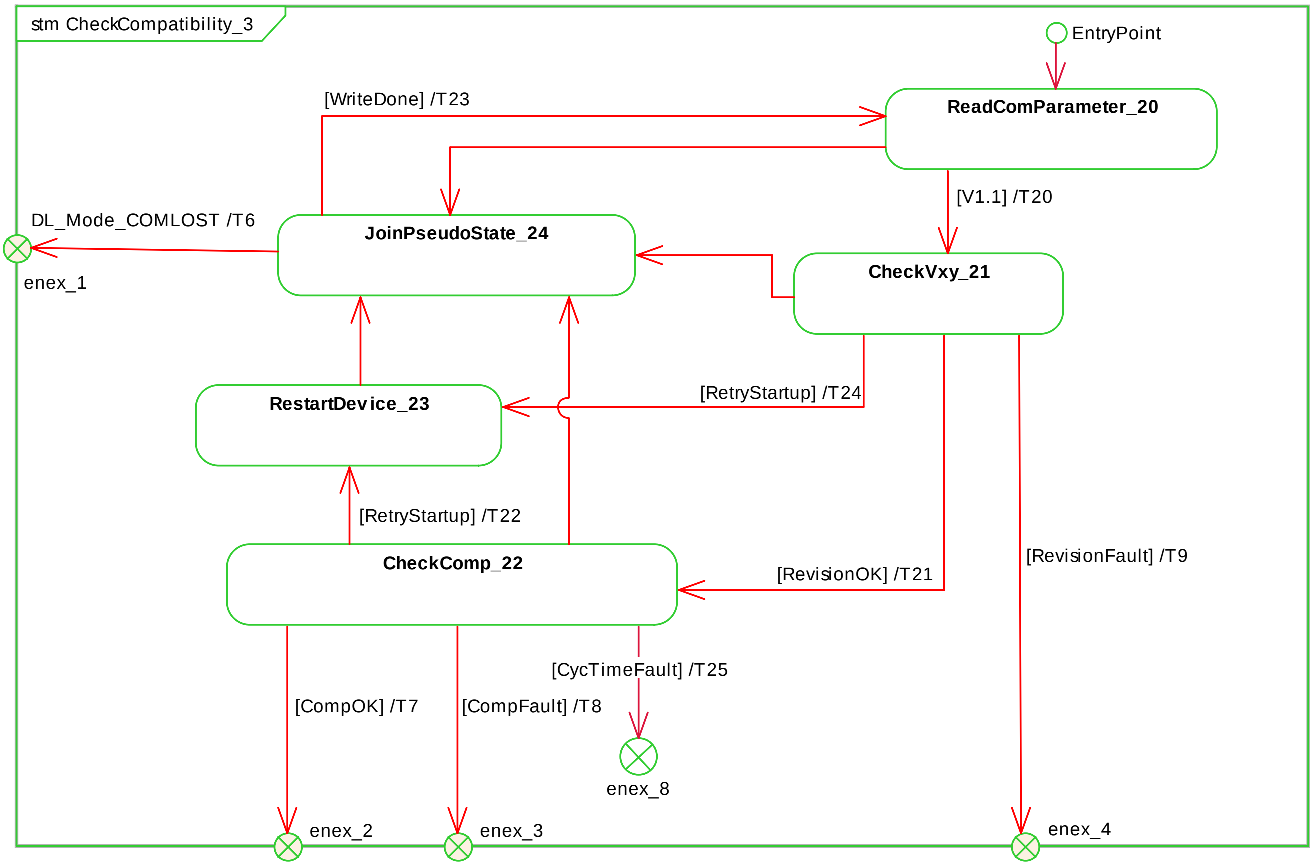


Figure 82 Submachine CheckCompatibility_3 of the W-Port-handler

Table 115 shows the state transition tables of the Submachine Check Compatibility 3 W-Port-handler.

Table 115 State transition tables Submachine Check Compatibility 3 W-Port-handler

STATE NAME	STATE DESCRIPTION
ReadComParameter_20	Write W-Parameter to WirelessSystemCfg by DL_Write(IMATime, MaxRetry, TxPower). Acquires communication parameters from Direct Parameter Page 1 (0x03 to 0x07, see Table 177) and WCycleTime (See C.4.12) via service DL_Read.
CheckVxy_21	A check is performed whether the configured revision (CRID) matches the real (actual) revision (RRID) according to Figure 83
CheckComp_22	Acquires identification parameters from Direct Parameter Page 1 (0x08 to 0x0E) via service DL_Read (see Table 177). The configured InspectionLevel (IL) defines the decision logic of the subsequent compatibility check "CheckComp" according to Figure 84
RestartDevice_23	Writes the compatibility parameters configured protocol revision (CRID) and configured DeviceID (CDID) into the W-Device according to Figure 85
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
CycTimeFault	Bool	See Figure 84; error variable CYCTIME_FAULT
CompOK	Bool	See Figure 84
CompFault	Bool	See Figure 84; error variable COMP_FAULT
RevisionFault	Bool	See Figure 83; error variable REVISION_FAULT
RevisionOK	Bool	See Figure 83
SerNumFault	Bool	See Figure 87, error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 87
V1.0	Bool	Real protocol revision of connected W-Device is in accordance with this standard
RetryStartup	Bool	See Figure 83 and Figure 84
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 83 to Figure 86 are demonstrating the context.

Figure 83 shows the decision logic for the protocol revision check in state "CheckVxy_21". In case of configured W-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 W-Device. If the Device does not accept; the W-Master returns an indication via the SM_Mode service with REV_FAULT.

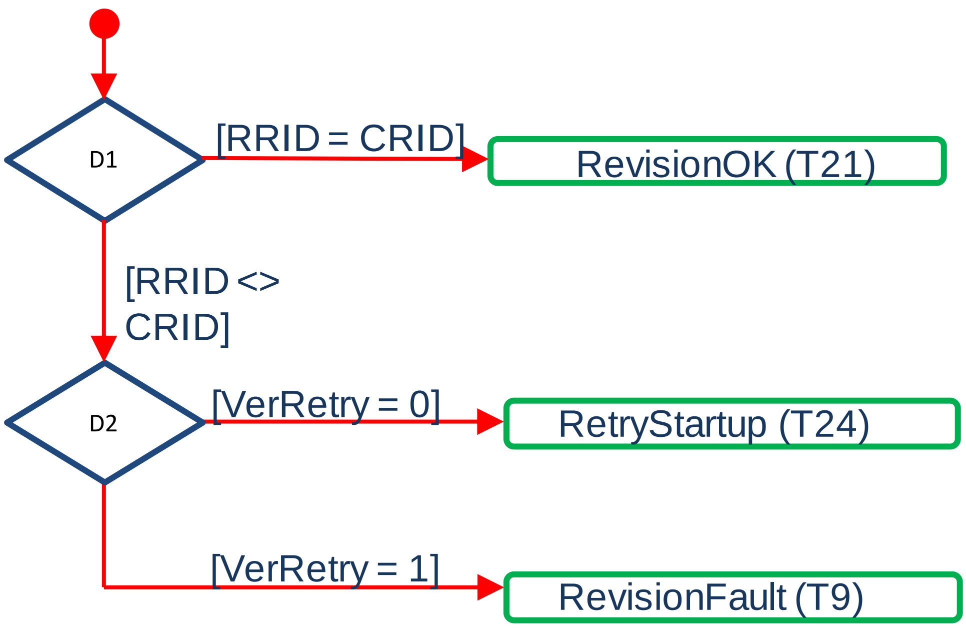
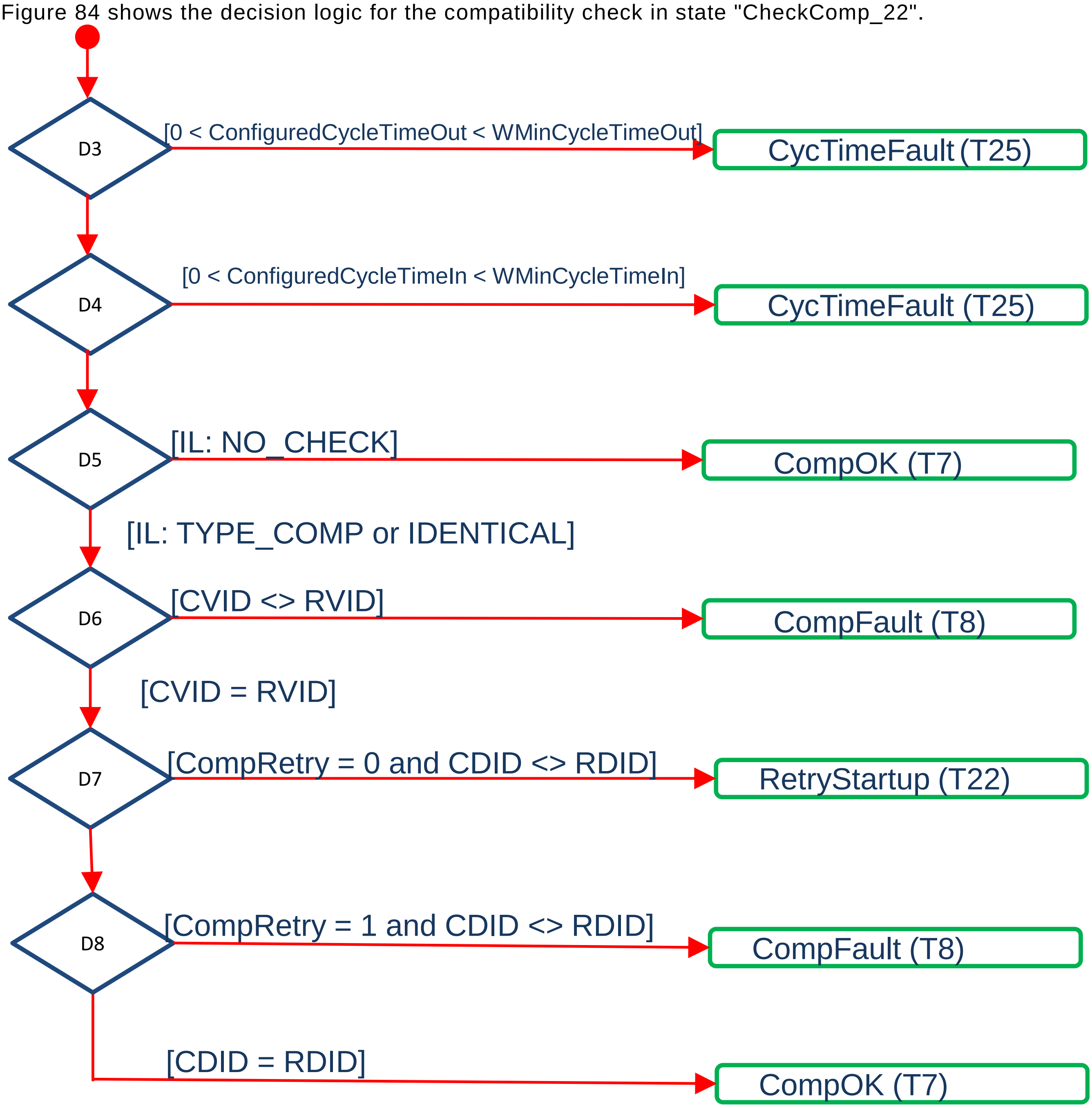


Figure 83 Activities for state „CheckVxy_21“

3963 Figure 84 shows the decision logic for the compatibility check in state "CheckComp_22".



3964
3965 **Figure 84 Activities for state „CheckComp_22“**
3966
3967
3968

Figure 85 shows the activity (write parameter) in state "RestartDevice_23".

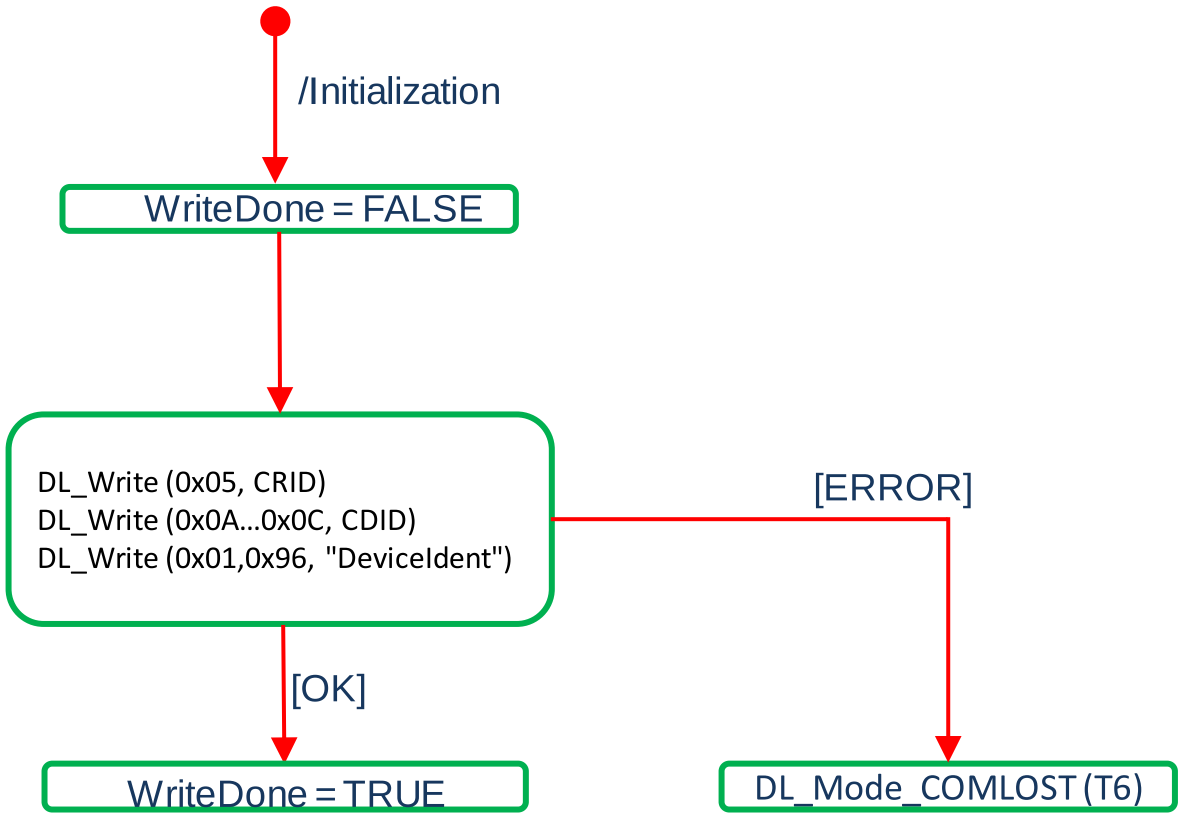


Figure 85 Activities (write parameter) in state "RestartDevice_23"

Figure 86 shows the SM Master submachine "checkSerNum_6". ". State CheckSernum_31 can be skipped (option).

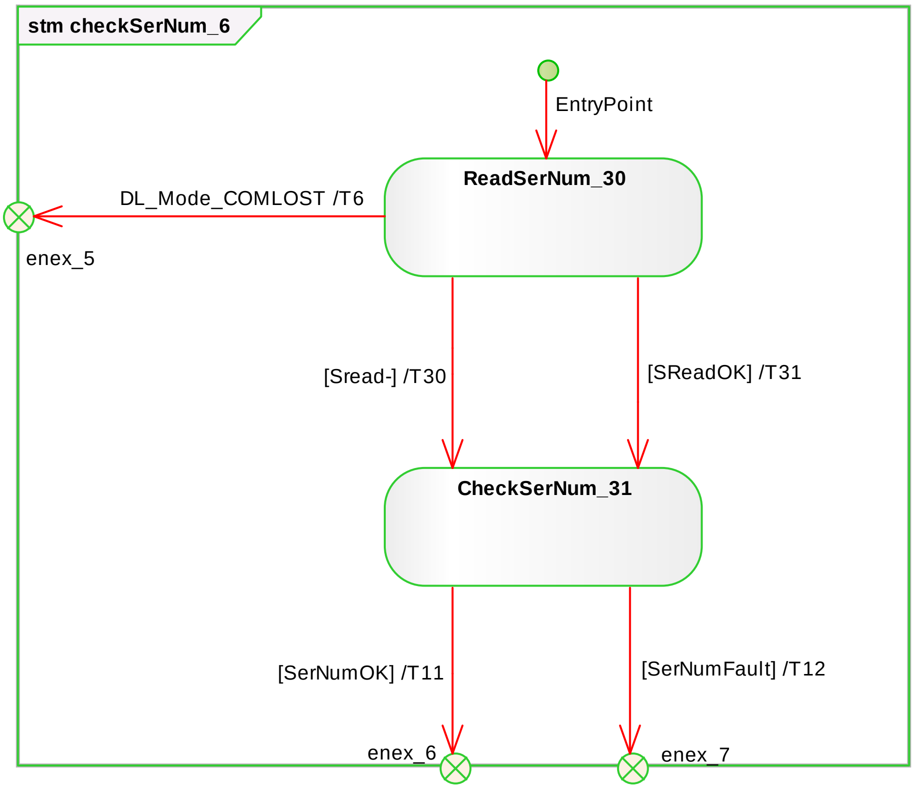


Figure 86 Submachine CheckSerNum_6 of the W-Port-handler

Table 116 shows the state transition tables of the Submachine CheckSerNum_6 of the W-Port-handler.

Table 116 State transition tables 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_31	Optional: SerialNumber checking skipped or checked correctly.

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 87
SERNumFault	Bool	See Figure 87

Figure 87 shows the decision logic (activity) for the state CheckSerNum_6.

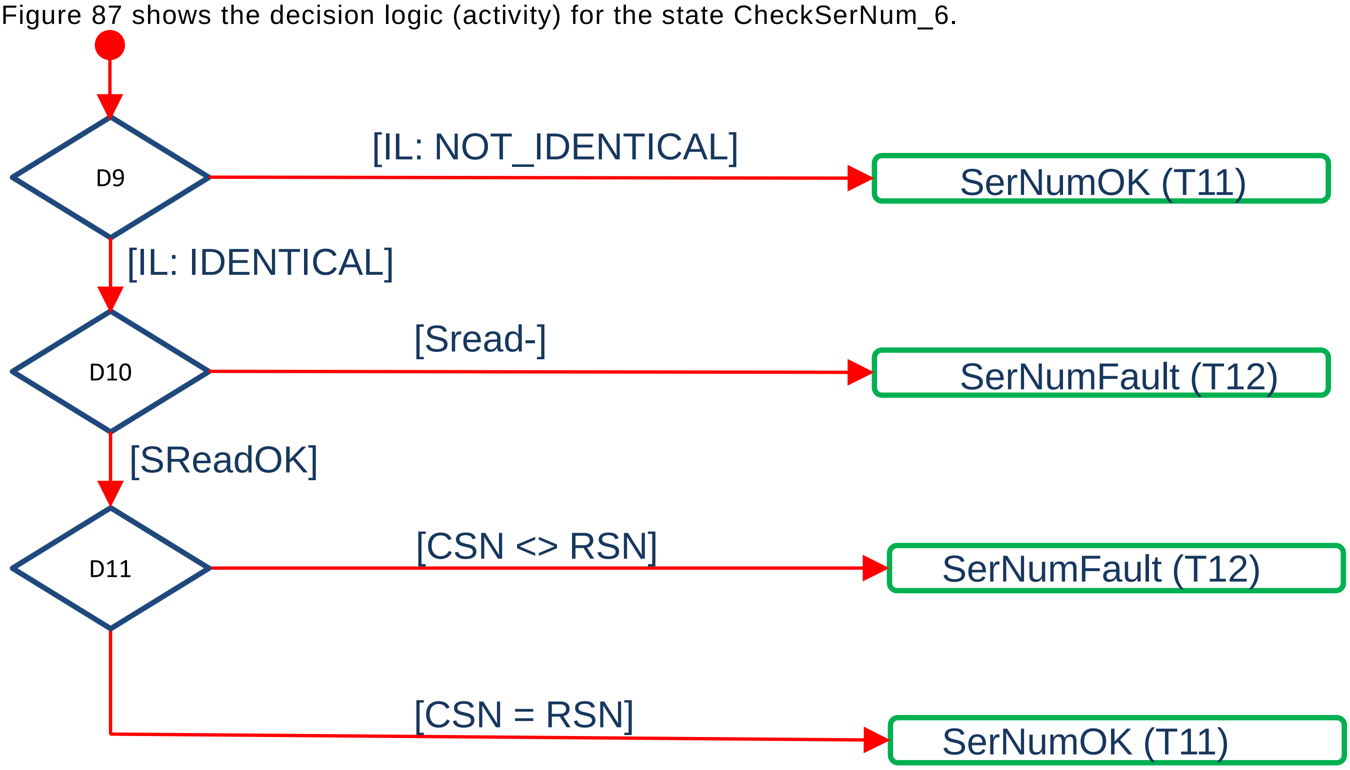


Figure 87 Activities (check SerialNumber) for state CheckSerNum_6

9.3.3.2.4 State Machine of W-Master AHT handler

Figure 88 shows the main state machine of the W-Master AHT-handler. The hopping table will be configured and updated to all W-Tracks and connected W-Devices, see clause H.4.

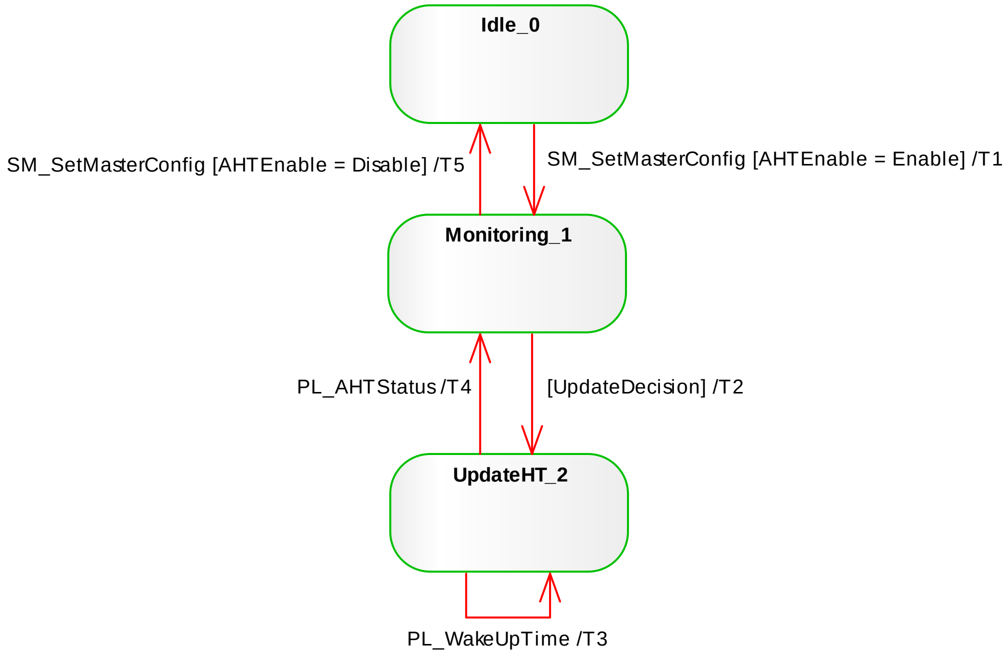


Figure 88 State Machine of the W-Master AHT-handler

Table 117 shows the state transition tables of the W-Master AHT-handler.

Table 117 State transition tables 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.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	1	Invoke PL_GetHopTable(Track_N) for each W-Track number to acquire all the hopping tables of the W-Master. Set AHTEnable = Enable.
T2	1	2	Invoke PL_SetWakeUpTime.req(WakeUpTime) Invoke PL_SetHopTable to dedicated W-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 W-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	0	Set AHTEnable = Disable

INTERNAL ITEMS	TYPE	DEFINITION
AHTEnable	variable	Indicates the AHT operation mode, set by SM_SetMasterConfig
UpdateDecision	guard	Indicates a decision to update the hopping table

9.4 System management of the W-Device

9.4.1 Overview

Figure 95 provides an overview of the 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 W-Device SM interacts with the PL to establish the necessary radio adjustments (see Figure 47 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 W-Device identity and compatibility (identification parameters).
The transitions between the W-Device PL states (see Figure 47) are initiated by the W-Master W-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 W-Device identification parameters through the W-Device applications interface.
The sequence chart in Figure 89 demonstrates the two possibilities of pairing of a typical W-Device sequence. It shows only the actions until the ComEstablish state. The remaining actions until the OPERATE state can be taken from Figure 92.

Figure 89 Sequence chart of a W-Device pairing

9.4.2 System management W-Device services

9.4.2.1 Overview

Subclause 9.3.2 describes the services the W-Device system management provides to its applications as shown in Figure 100

Table 118 lists the assignment of the W-Device to its role as initiator or receiver for the individual system management service.

Table 118 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		

9.4.2.2 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 119.

Table 119 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 initial maximum number of allowed retries in count of W-Sub-Cycles (see clause C.4.4.3).
This value is only used for communication establishment and will be changed by the W-Master at startup.

IMATime
This parameter contains the initial I am alive time. Permitted values: 2 octets, time encoded according to clause C.4.4.2.
This value is only used for communication establishment and will be changed by the W-Master at startup.

TxPower
This parameter contains the initial transmission power for the W-Device (see Table 186).
This value is only used for communication establishment and will be changed by the W-Master at startup.

DLinkType
This parameter contains the downlink type (see Table 27) for the radio to listen (full downlinks or Pre-Downlinks).

SlotType
This parameter contains the uplink type (see Table 175) for the uplink capability of the radio (Single Slot or Double Slot).
UniqueID
This parameter contains the UniqueID from the W-Device (see Figure 140).
WMinCycleTimeOut
This parameter contains the minimum cycle time supported for PDOut by the W-Device (see C.4.12)
WMinCycleTimeIn
This parameter contains the minimum cycle time supported for PDIn by the W-Device (see C.4.12)
RevisionID (RID)
This parameter contains the protocol revision (see clause C.2.3) 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)

9.4.2.3 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 120.

Table 120 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 C.4.4.3).
IMATime
This parameter contains the current I am alive time. Permitted values: 2 octets, time encoded according to clause C.4.4.2.
TxPower
This parameter contains the current transmission power for the W-Device (see Table 186).
DLinkType
This parameter contains the current downlink type (see Table 27) for the radio to listen (full Downlink packets or Pre-Downlink packet parts).
SlotType

This parameter contains the uplink type (see Table 175) for the uplink capability of the radio (Single Slot or Double Slot).

WMasterCycleTimeOut

This parameter contains the WMasterCycleTimeOut to be set by the W-Master system management (see clause C.4.12). This parameter is only valid in the state SM_Operate.

WMasterCycleTimeIn

This parameter contains the WMasterCycleTimeIn to be set by the W-Master system management (see clause C.4.12). This parameter is only valid in the state SM_Operate.

RevisionID (RID)

This parameter contains the current protocol revision (see clause C.2.3) 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 C.2.4).

ProcessDataOut

This parameter contains the current length of PD to be sent by the W-Master (see clause C.2.5).

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)

9.4.2.4 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 121.

Table 121 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 in [1])

Data length: 2 octets

DeviceID (DID)

This parameter contains one of the assigned DeviceIDs (see B.1.9 in [1])

Data length: 3 octets

FunctionID (FID)

This parameter contains one of the assigned FunctionIDs (see B.1.10 in [1]).

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)

9.4.2.5 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 122.

Table 122 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).

9.4.2.6 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 123

Table 123 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)

9.4.2.7 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 124.

Table 124 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)

9.4.3 SM W-Device protocol

9.4.3.1 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.

9.4.3.2 State machine of W-Device System Management

Figure 90 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.

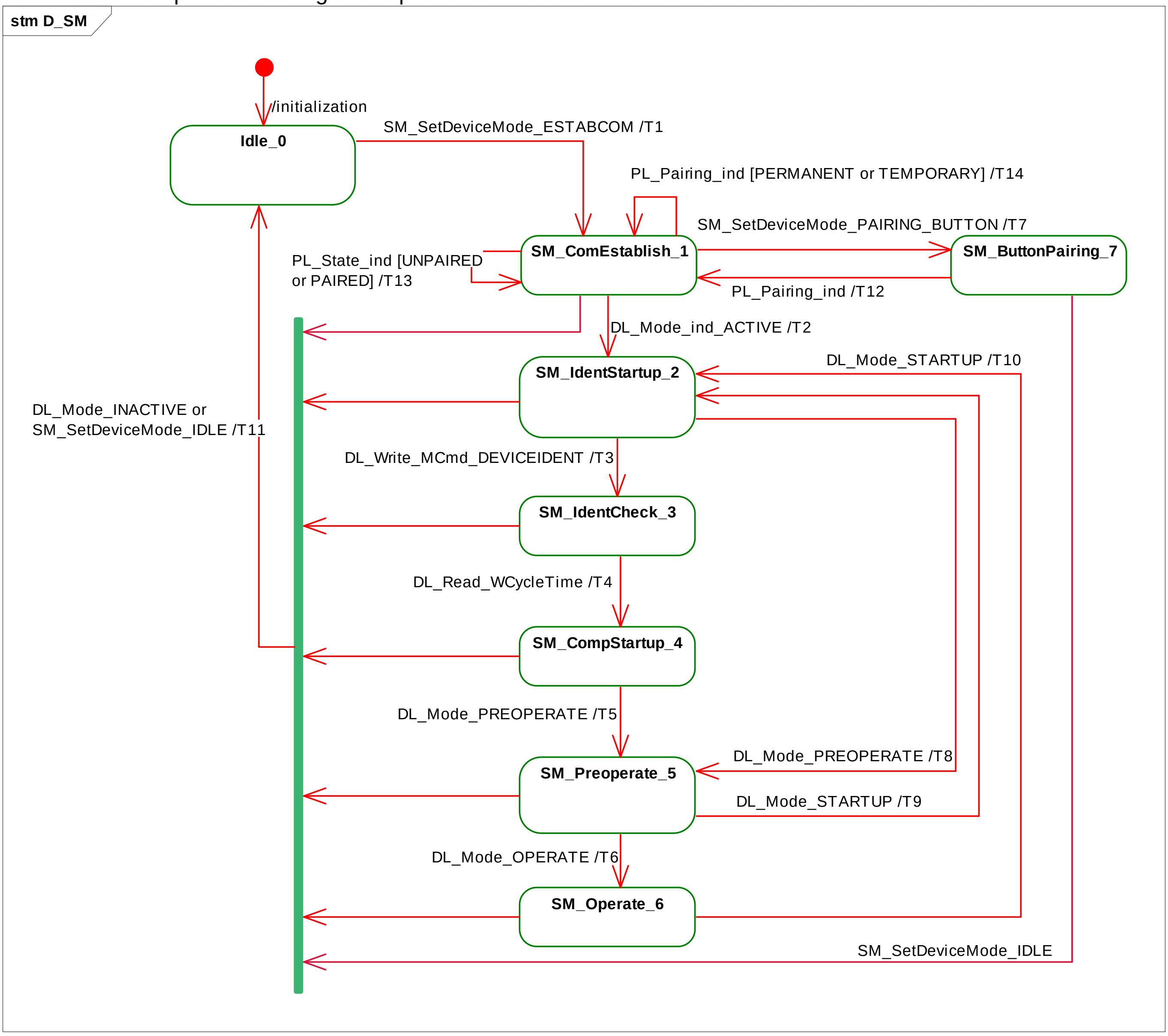


Figure 90 State machine for W-Device System Management

Table 125 shows the state transition tables of the W-Device System Management.

Table 125 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.ind(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 W-Parameters are written by DL_Write(WirelessSystemCfg) and the communication parameter (Direct Parameter page 1, addresses 0x03 to 0x07, and WCycleTime, see C.4.12) 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(W-Parameter, Index 0x5007 = "WCycleTime") 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.ind(ESTABCOM). Set WDeviceMode to INACTIVE (see C.4.4) 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). Set WDeviceMode to IDENTSTARTUP (see C.4.4)
T3	2	3	The W-Device identity check phase is entered by receiving the trigger DL_Write.ind(MCmd_DEVICEIDENT). Invoke SM_DeviceMode(IDENTCHANGE) Set WDeviceMode to IDENTCHANGE (see C.4.4)
T4	3	4	The W-Device compatibility startup phase is entered by receiving the trigger DL_Read.ind(W-Parameter, Index 0x5007 = "WCycleTime").
T5	4	5	The W-Device's preoperate phase is entered by receiving the trigger DL_Mode.ind(PREOPERATE). Invoke SM_DeviceMode(PREOPERATE) Set WDeviceMode to PREOPERATE (see C.4.4)
T6	5	6	The W-Device's operate phase is entered by receiving the trigger DL_Mode.ind(OPERATE). Invoke SM_DeviceMode(OPERATE) Set WDeviceMode to OPERATE (see C.4.4)
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) Set WDeviceMode to PREOPERATE (see C.4.4)
T9	5	2	The W-Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP) Set WDeviceMode to STARTUP (see C.4.4)
T10	6	2	The W-Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP) Set WDeviceMode to STARTUP (see C.4.4)
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). Set WDeviceMode to INACTIVE (see C.4.4) Invoke PL_SetMode(STOP_KEEP) 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

9.4.3.3 State Machine of W-Device AHT handler

Figure 91 shows the main state machine of the W-Device AHT-handler. The handler manages the adaptive hopping table mechanism, see clause H.4

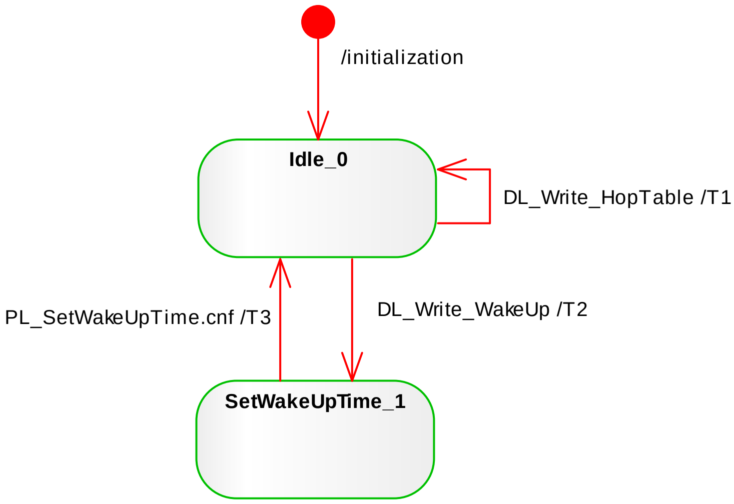


Figure 91 State Machine of the W-Device AHT-handler

Table 126 shows the state transition tables of the W-Device AHT-handler.

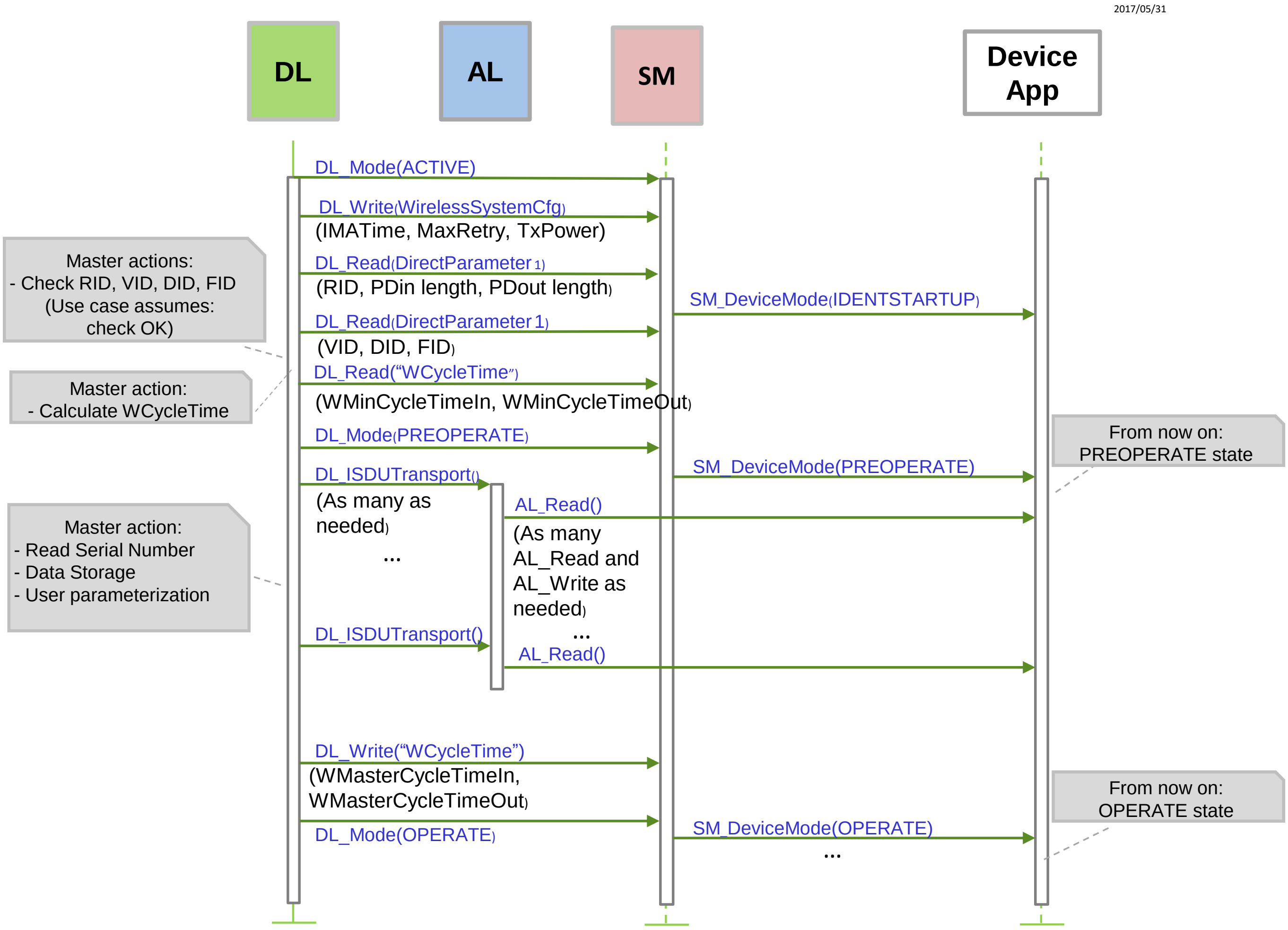
Table 126 State transition tables of the W-Device AHT-handler

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	0	Invoke PL_SetHopTable(HopTable)
T2	0	1	Invoke PL_SetWakeUpTime(WakeUpTime)
T3	1	0	See T1

4262

4263 **9.4.3.4 Start-up and Synchronization**



4264

Figure 92 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).

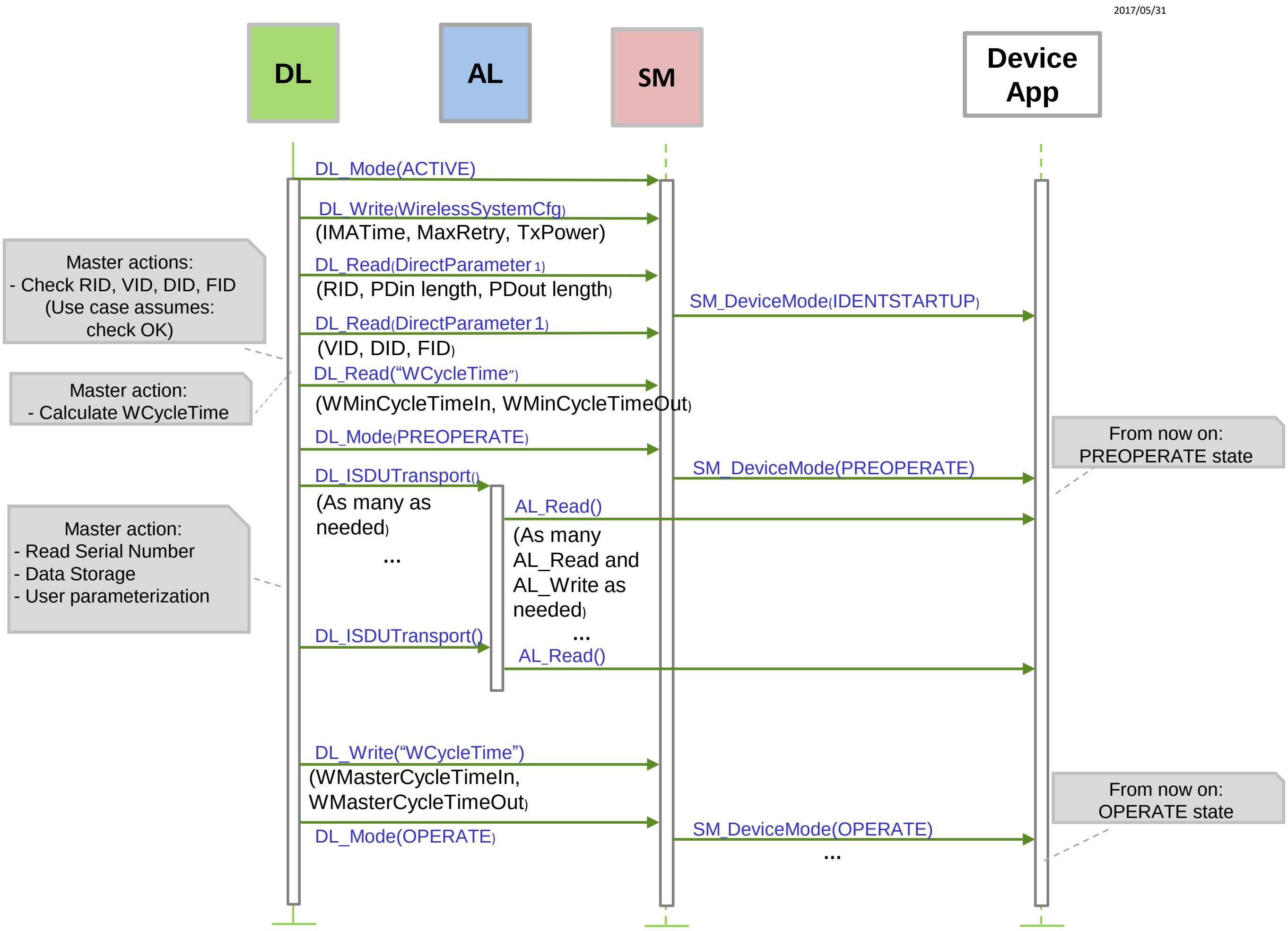


Figure 92 Sequence chart of a regular W-Device startup

Figure 93 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 93 shows only the actions until the PREOPERATE state. The remaining actions until the OPERATE state can be taken from Figure 90.

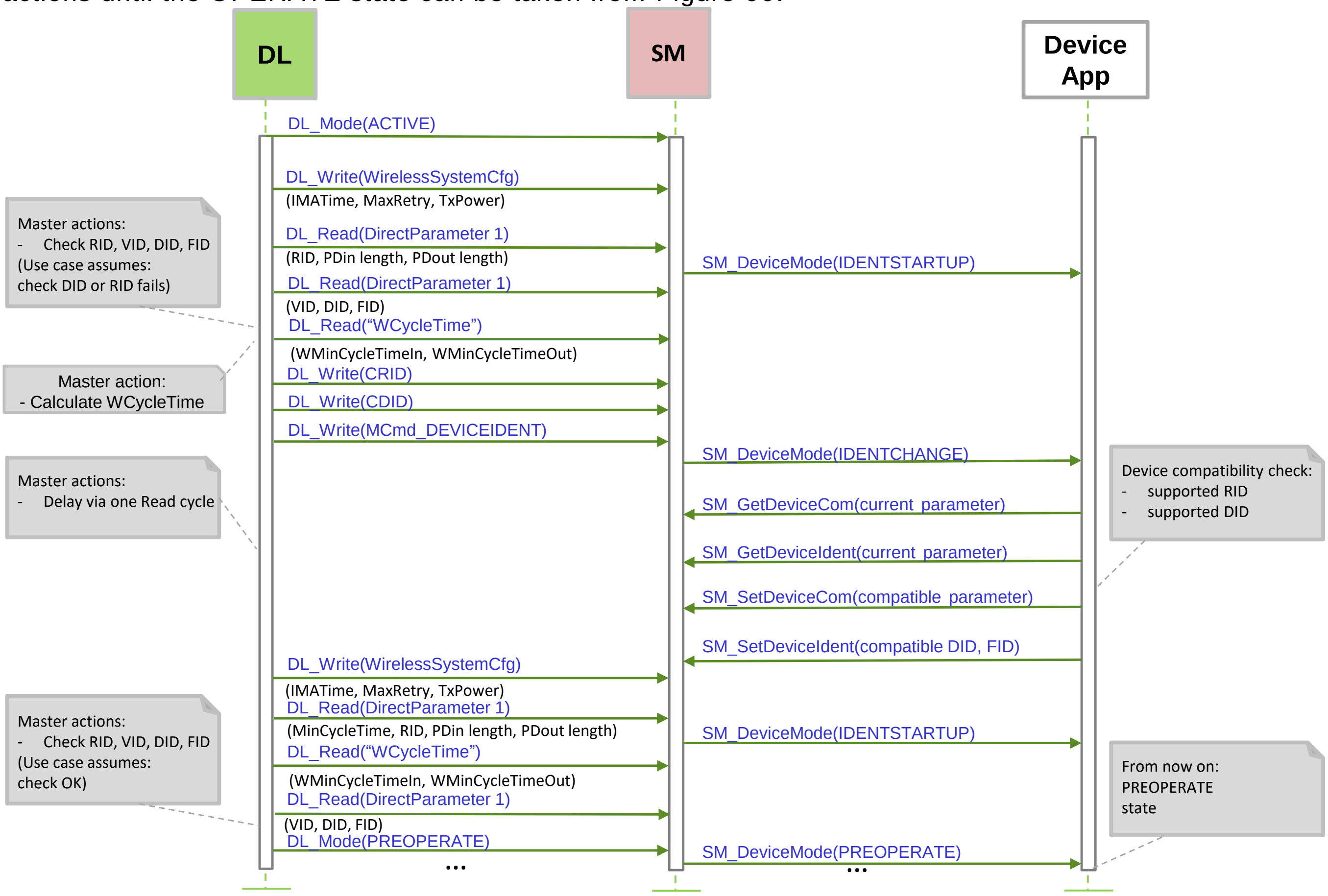


Figure 93 Sequence chart of a W-Device startup in compatibility mode

Figure 94 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.

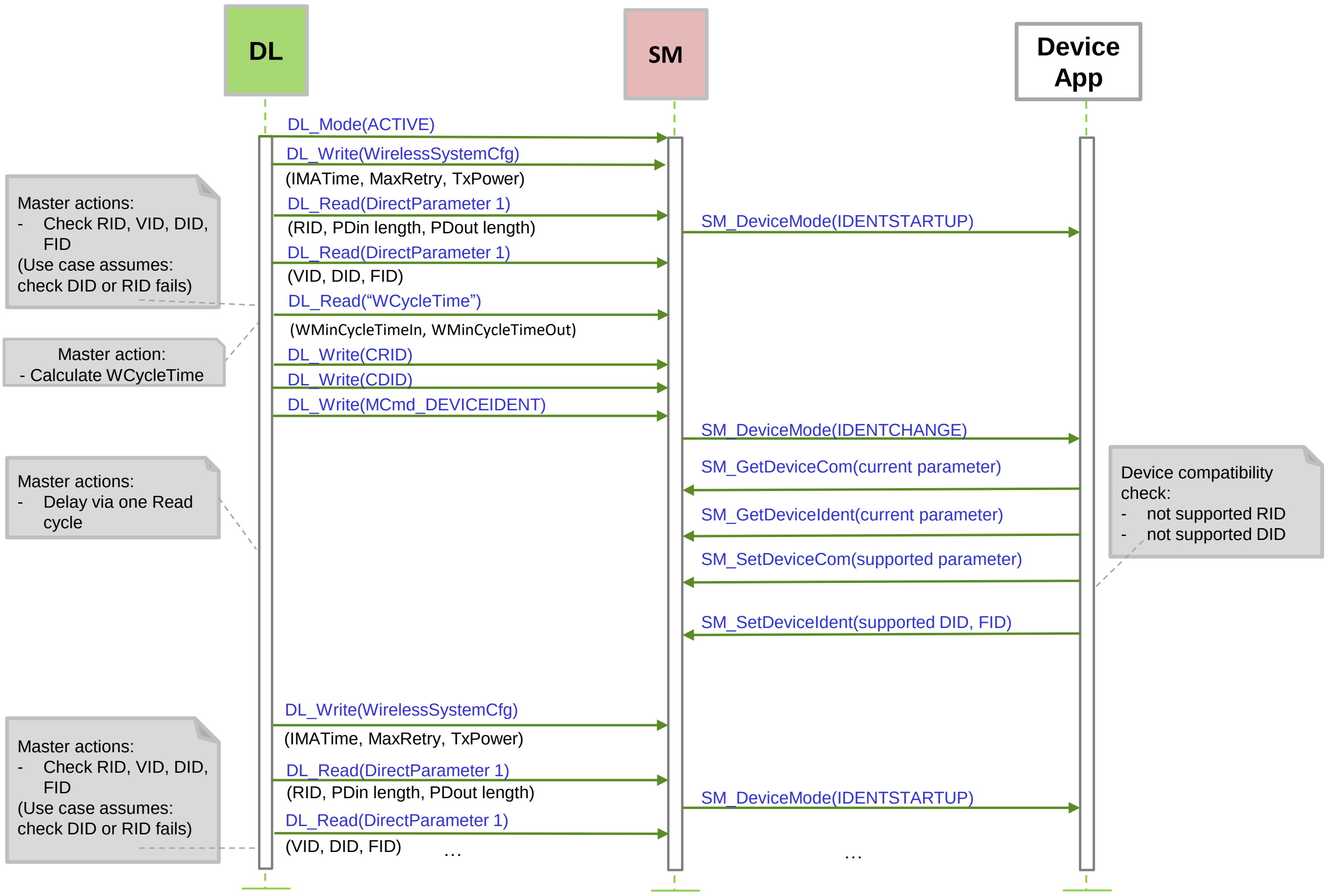


Figure 94 Sequence chart of a W-Device startup when compatibility fails

10 W-Device

10.1 Overview

Figure 95 provides an overview of the complete structure and services of a W-Device.

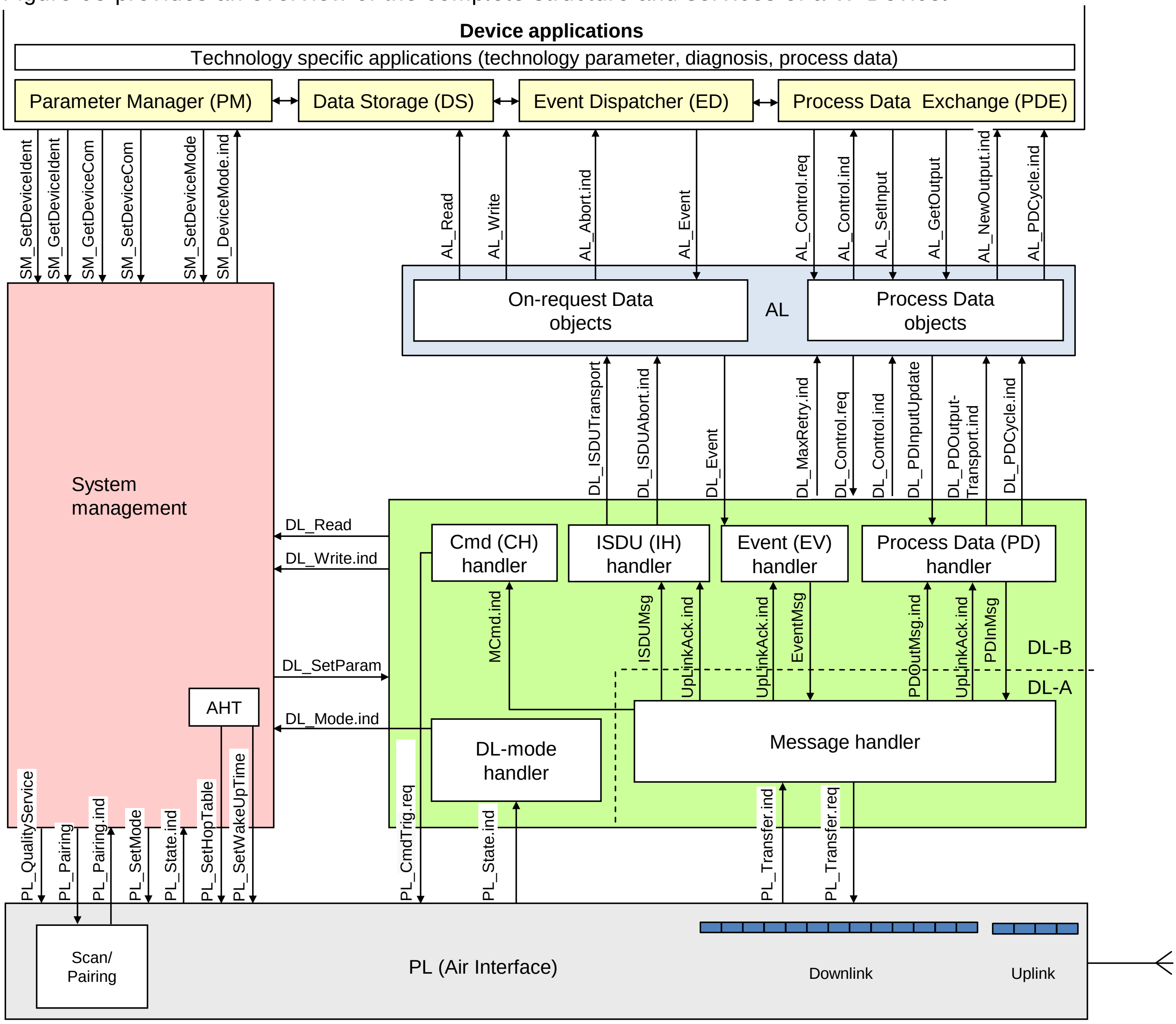


Figure 95 Structure and services of a W-Device

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 6.6.2 and 6.6.4)
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 7.4.2) 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 A.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 6.6.5 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 50). If IMA time is exceeded (monitored by the W-Master, see Figure 49), 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 8 and Annex D).

Mandatory for all W-Devices are the Direct Parameters in page 1. Page 1 contains common communication and identification parameters (see Table 177).

For IO-Link Wireless additional mandatory parameters have been defined, which are listed in Table 179 (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 [1].

10.4.3 Dynamic parameter

See IO-Link specification 10.3.3 in [1].

10.4.4 Single parameter

See IO-Link specification 10.3.4 in [1].

10.4.5 Block parameter

See IO-Link specification 10.3.5 in [1].

10.4.6 Concurrent parameterization access

See IO-Link specification 10.3.6 in [1].

10.4.7 Command handling

See IO-Link specification 10.3.7 in [1].

10.5 Data Storage (DS)

See IO-Link specification 10.4 in [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 A.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 196 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 197 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 11.6 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 A.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 5.7.1.5.
This mandatory functionality is supported by the PL of the W-Device (see 5.7).

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 4.4.3.1 and 5.7.1.3).
This mandatory functionality is supported by the PL of the W-Device (see 5.7).

10.7.4 Pairing by Button / Re-Pairing

This feature enables a W-Device to use the "pairing by Button" mechanism (see Figure 40). 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 9.3.2.7 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 4.4.3.2 and 4.4.3.3.
Therefore, the PL of a W-Device shall store the ConnectionParameter (see Table 154) in non-volatile memory (e.g., flash memory).
This mandatory functionality is supported by the PL of the W-Device (see 5.7).

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 4.4.5 and 5.7.1.5).
Therefore, the PL of a W-Device shall store the ConnectionParameter (see Table 154) 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 5.7).

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 154).
This mandatory functionality is supported by the PL of the W-Device (see 5.7).

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 C.4.4.1), 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 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 10.11.3.1. The indication is optional but highly recommended for a W-Device.
The mandatory "Pairing-Button" supports pairing, re-pairing and further optional functions, see 10.11.3.2.

10.7.10 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 IO-Link interface (see B.2.4 in [1]). The locking is triggered by the reception of a system parameter "Device Access Locks" (see Table 179). The support for these functions is optional for a W-Device.

10.7.11 Data Storage locking

Setting this lock will cause the "State_Property" (Table B.10 in [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.

10.7.12 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 [1]). The support of this function is optional for a W-Device.

10.7.13 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 [1]). The support for this function is optional for a W-Device.

10.7.14 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 11.4). The support of this function is optional for a W-Device.

10.7.15 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 reset options

See IO-Link specification 10.7 in [1].
Additionally, the SystemCommand "Back to box" shall delete the ConnectionParameter.

10.9 W-Device design rules and constraints

10.9.1 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 W-Devices. An overview of the major protocol variables scattered all over the standard is concentrated in Table 127 with associated references. For additional design rules of low energy W-Devices see Annex D.

10.9.2 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 F.3.3 in [1] and in [3].

See 10.2, A.3.1 and 6.6.5 for details on the indication of valid or invalid Process Data via the transmission of Process Data (PDx_Valid) within the data exchange.

10.9.3 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 47 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 W-Device Application shall send the Event (D_IOLW_Retry_Error) vie event channel to the W-Master.

NOTE: This is especially important for actuators such as valves or motor management.

10.9.4 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 47 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.

10.9.5 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 7.7.2 and 7.7.3). 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.

10.9.6 ISDU communication channel

The ISDU communication channel provides a powerful means for the transmission of parameters and commands (see C.4).
The following rules shall be considered when using this channel (see Figure 8).

- 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).
- Indexes from Table 179 are not writeable by a PLC application program.
- After reception of an ISDU request from the W-Master the W-Device shall respond within 5 000 ms (see Table 127). Any violation causes the W-Master to abandon the current task.

10.9.7 DeviceID rules related to W-Device variants

W-Devices with a certain DeviceID and VendorID shall not deviate in communication and functional behavior. This applies for sensors and actuators. Those W-Devices may vary for example in

- housing materials,
- mounting mechanisms,
- other features, and environmental conditions.

10.9.8 Protocol constants

See IO-Link specification 10.8.7 in [1].
The system variable “Wake-up procedure” defined in IO-Link is not used any more. In difference to IO-Link “MaxRetry” is a configurable value for IO-Link Wireless, see C.4.4.3.

Table 127 gives an overview of the protocol constants for W-Devices.

Table 127 Overview of the protocol constants for W-Device

System variable	References	Values	Definition
Detection for COMLOST	Annex D	5 * MaxRetry	ComLost is reported via PL-service PL_State, see Table 35, T10, T12
Detection for “wireless connection synchronized”	Annex D	3 W-Sub-cycles	SYNCED is reported via service PL_State, see Table 35, T9 (3 subsequent DLinks received by W-Device)

10.10 IO Device description (IODD)

10.10.1 General

An IODD (I/O Device Description) is a file that provides all the necessary properties to establish communication and the necessary parameters and their boundaries to establish the desired function of a sensor or actuator.

An IODD (I/O Device Description) is a file that formally describes a Device..

An IODD file shall be provided for each Device and shall include all information necessary to support this document.

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.

The ISDU indexes listed in Table 188 shall not be defined as writable in an IODD .
Index 0x5000 WDeviceMode and 0x5006 AdaptiveHopTable shall not be accessible via IODD.

NOTE: Details of the IODD language to describe a Device can be found in [3].

Figure 96 shows a schematic representation of the use of a W-Device and a W-Bridge to connect an IO-Link Device.

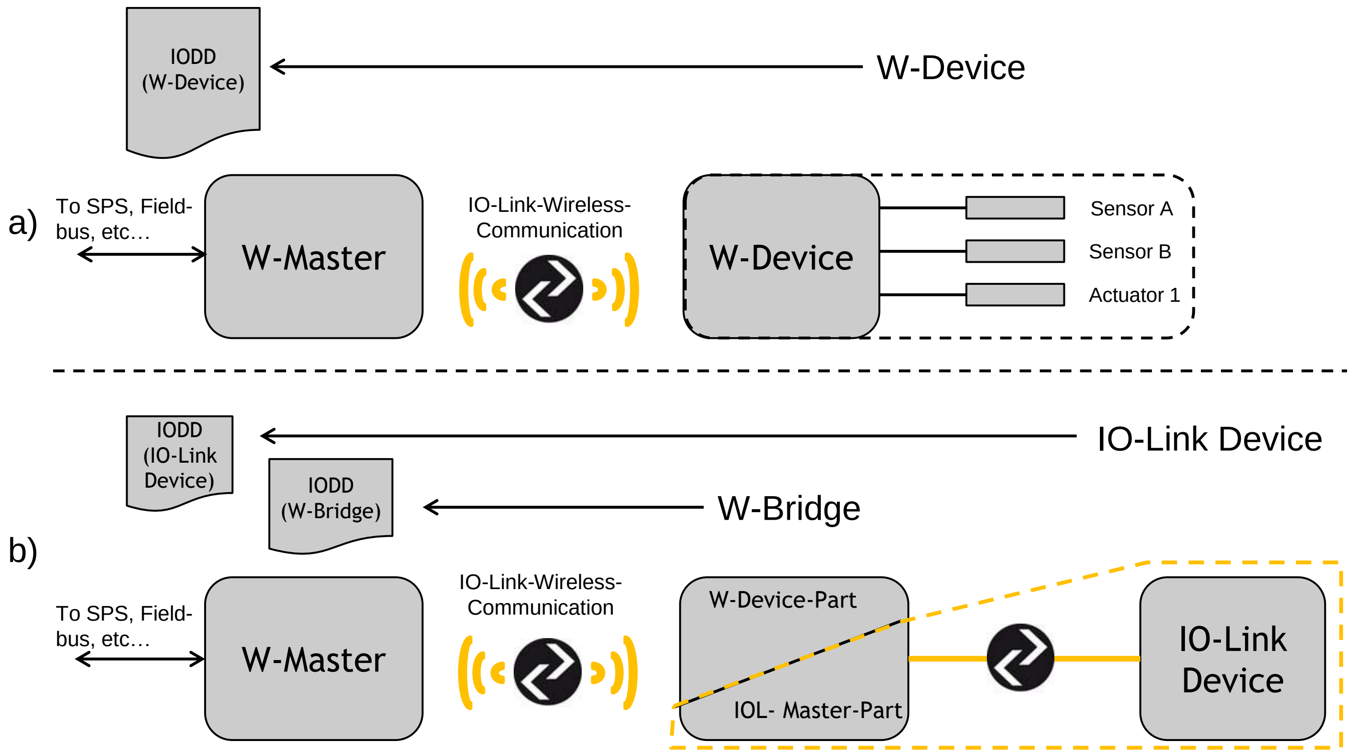


Figure 96 Schematic representation of the use of (a) a W-Device and (b) a W-Bridge to connect an IO-Link Device.

10.10.2 Profile Characteristics

The ProfileID for the IO-Link Wireless is 16386 or 0x4002. A W-Bridge has the extending FunctionClass 32816 (0x8030).

10.10.3 CommNetwork Profile Instance for the IODD

10.10.3.1 General

This section gives an example for the content of the IODD file using “IOLinkWirelessCommNetworkProfileT”. The 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 port device configuration tool uses the IODD of the IO-Link Device and the IODD of the W-Bridge. The W-Parameters are used from the IODD of the W-Bridge and device specific parameters are taken from the IODD of the IO-Link Device.

The W-Parameter 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 IODD of the IO-Link Device shall be replaced by the “IOLinkWirelessCommNetworkProfileT” of the IODD of the W-Bridge. The IODD description from index 0x5000 to 0x50FF shall be added to the IODD, to get an IODD for a W-Device

In the case where the device has an IO-Link and an IO-Link Wireless interface, 2 IODD files, one for the wired and another one for the wireless connection, shall be used to describe it.

10.10.3.2 Example of an IO-Link Wireless extension

10.10.3.2.1 Battery powered W-Device

```
<CommNetworkProfile xsi:type="IOLinkWirelessCommNetworkProfileT" iolinkWirelessRevision="V1.1">
```

```
  <TransportLayers>
    <PhysicalLayer  WMinCycleTimeOut="315000"  WMinCycleTimeIn="315000"  defaultSlotType="SSLOT" 
isABridge="false" maxTxPower="10" isLowPowerDevice="true">
      <Connection xsi:type="OtherConnectionT">
        <ProductRef productId="BatteryDeviceVariant"/>
        <Description textId="TD_BatteryDevice"/>
      </Connection>
    </PhysicalLayer>
  </TransportLayers>
</CommNetworkProfile>
```

10.10.3.2.2 Cable powered W-Device

```
<CommNetworkProfile xsi:type="IOLinkWirelessCommNetworkProfileT" iolinkWirelessRevision="V1.1">
  <TransportLayers>
    <PhysicalLayer    WMinCycleTimeOut="5000"    WMinCycleTimeIn="5000"    defaultSlotType="SSLOT" 
maxTxPower="10" isLowPowerDevice="false">
      <Connection xsi:type="CableConnectionT">
        <ProductRef productId="CableDeviceVariant"/>
        <Wire1 function="L+" color="BN">
          <Name textId="TN_Wire1"/>
        </Wire1>
        <Wire2 function="NC" color="WH">
          <Name textId="TN_Wire2"/>
        </Wire2>
        <Wire3 function="L-" color="BU">
          <Name textId="TN_Wire3"/>
        </Wire3>
        <Wire4 function="Other" color="BK">
          <Name textId="TN_Wire4"/>
        </Wire4>
      </Connection>
    </PhysicalLayer>
  </TransportLayers>
</CommNetworkProfile>
```

The physical layer section contains the following attributes:

WMinCycleTimeOut (m, unsignedInt)
This parameter indicates the minimum cycle time of W-Device in microseconds.
Permitted values: 5000 to 315000, 0 for not used.

WMinCycleTimeIn (m, unsignedInt)
This parameter indicates the minimum cycle time of W-Device in microseconds.
Permitted values: 5000 to 315000, 0 for not used.

defaultSlotType (m, string)
This parameter indicates if that the W-Device is a Single Slot or Double Slot.
Permitted values: SSlot or DSlot (see Table 175)

maxTxPower(m, integer)
This parameter indicates the maximum transmission power of the W-Device in dBm
Permitted values: -20 to 10dBm. For mapping of values see C.4.4.4.

isLowPowerDevice (o, boolean)
This parameter indicates if the W-Device is a low power device and supports the corresponding protocol features.
The default value is “false”.

isABridge (o, boolean)
This parameter indicates if the W-Device is a W-Bridge.
The default value is “false”.

The connection and test sections are used in the same way as for an IO-Link Device.

10.11 W-Device diagnosis

10.11.1 Concepts

See IO-Link specification 10.10.1 in [1].

10.11.2 Events

MODE values shall be assigned as follows (see A.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. Once communication resumed, 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 8) from being flooded, the following requirements apply:

- The same diagnosis information shall not be reported at less than 1 s intervals. That means the Event Dispatcher shall not invoke the AL_Event service with the same EventCode more often than after 1 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 invoke the AL_Event service with an EventCount equal one.
- Errors are prioritized over Warnings.

10.11.3 W-Device HMI

10.11.3.1 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 128.
The IO-Link Wireless indication shall use a green indicator.

Table 128 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 B.2.2 in [1]. 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 97.

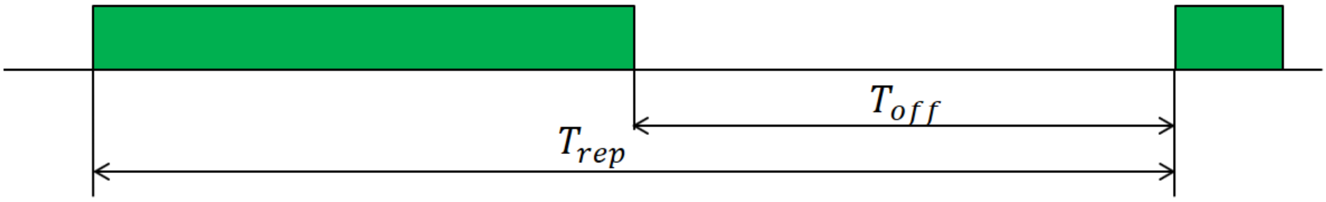


Figure 97 W-Device LED blink timing

The indication of the double flashing LED follows the timing shown in Figure 98.

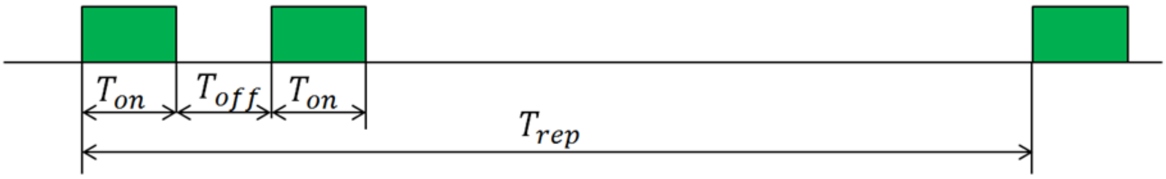


Figure 98 W-Device LED double flash timing.

The additional function of visual indicators for low energy W-Devices are defined in G.1.9.

10.11.3.2 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 4.4.3.2 and 4.4.3.3. Further, each button press shall trigger a “HMI button pressed” Event, see Annex D EventCodes.
The Pairing-Button supports further functions, depending on the duration of the button pressed, see Table 129.

Table 129 Pairing Button functions

Button press timing	Button press function	Remarks
[0.1…0.5] 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.
[>0.5…1] s	No action	
[>1…5] s	Pairing by Button / Re-pairing by Button	Mandatory for all W-Devices
[>5…10] s	No action	
[>10…20] s	Device Reset (see 10.7 in [1])	Highly recommended for low energy W-Devices with an internal power source.
[>20…30] s	No action	
[>30…40] s	Unpairing (see 10.7.6)	Highly recommended for all W-Devices

The button press timings are typical values. A tolerance of 10% shall not be exceeded.

If the W-Device has a more complex user interface (for example a graphical user interface) the Button press functions can be implemented separately with no timing constrains.

10.12 W-Device connectivity

See 4.4.3 (Pairing), 4.4.4 (Unpairing) and 4.4.5 (Roaming) for the different possibilities of pairing W-Devices to W-Master ports and the corresponding mechanisms.

NOTE: For compatibility reasons, this document 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 clause 4.2. 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 99 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.

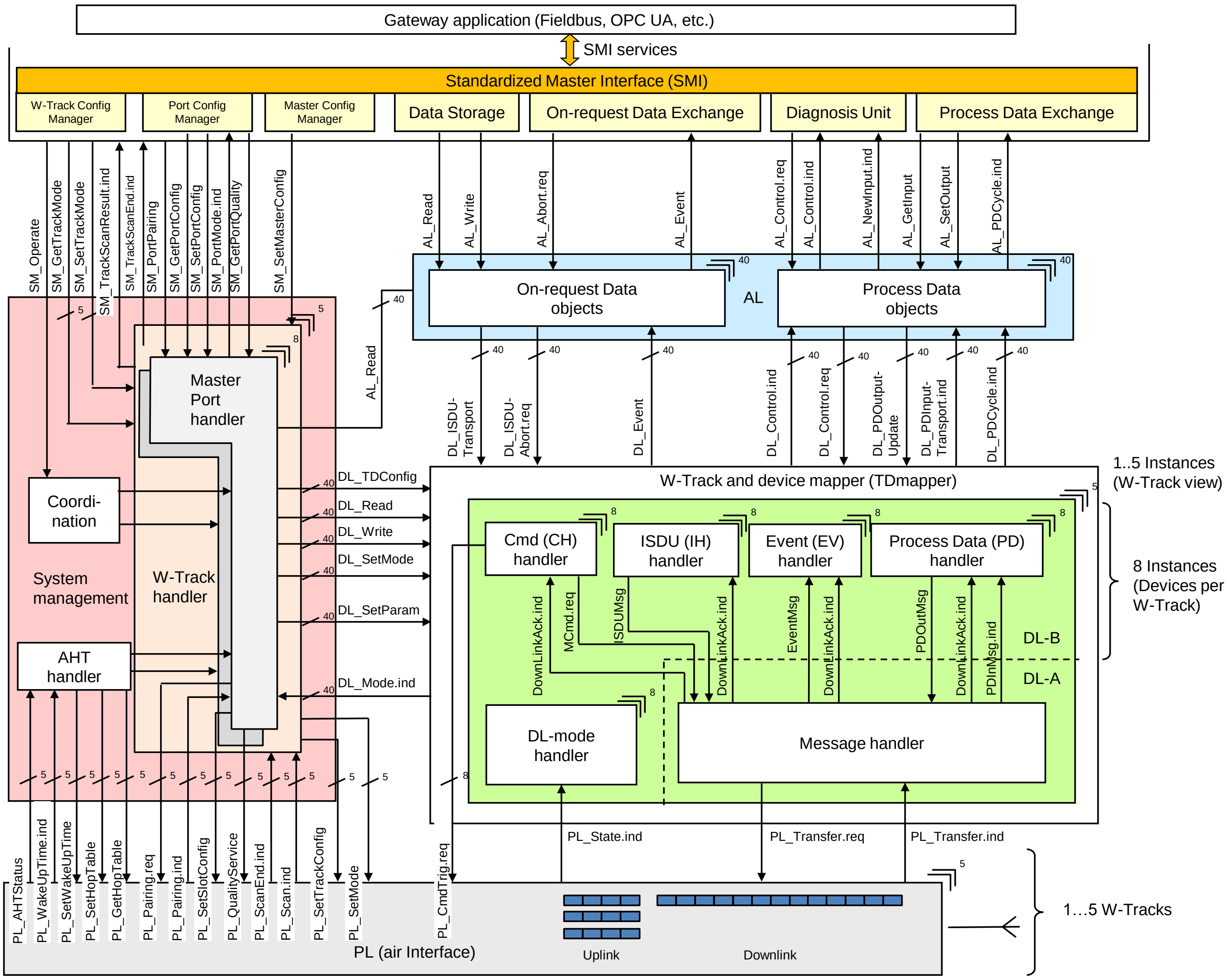


Figure 99 Structure and services of a W-Master

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-, W-Track- and W-Port-Configuration Manager (CM),** transforms the user configuration assignments into W-Port and W-Track set-ups (see 11.2 in [1]);
- **Data Storage (DS)** mechanism, which can be used to save and restore the W-Device parameters (see 11.3 in [1]);

- **Diagnosis Unit (DU)**, which routes Events from the AL to the Data Storage unit or the gateway application (see 11.4 in [1]);
- **On-request Data Exchange (ODE)**, which provides for example acyclic parameter access (see 11.5 in [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 [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 100.

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 document to define any of these gateway applications.

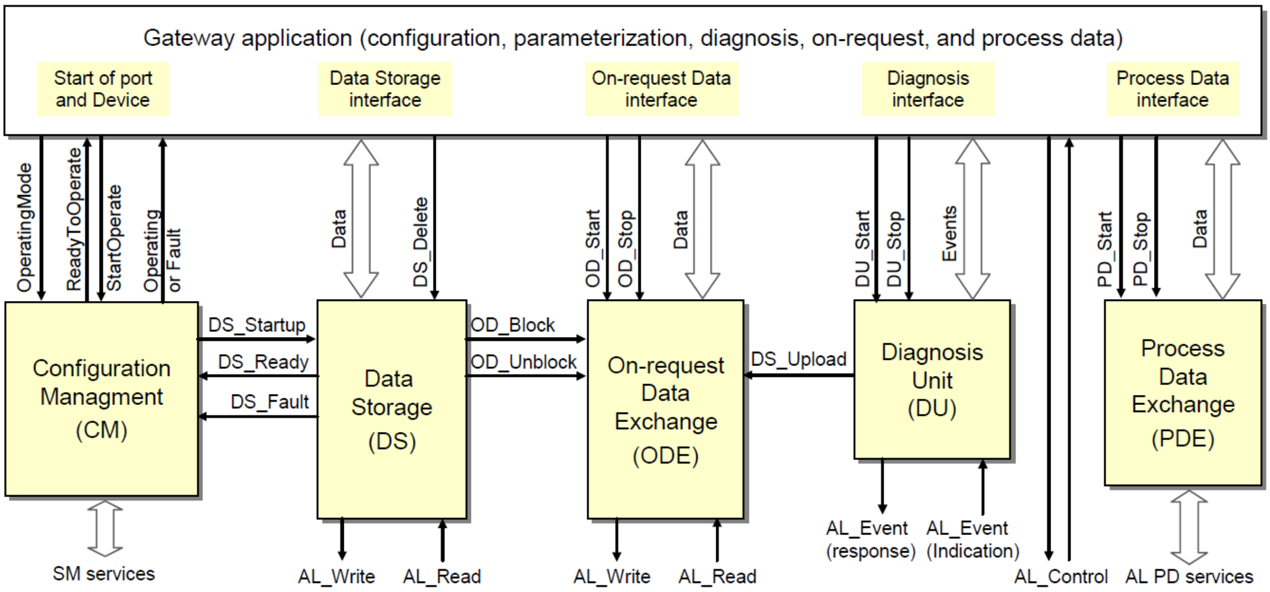


Figure 100 Relationship of the common W-Master applications

The internal variables between the common W-Master applications are specified in Table 130. The main responsibility is assigned to the Configuration Manager (CM) as shown in Figure 100 and explained in 11.3

4793 **Table 130 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 70 and Table 140 State transition tables of the W-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.

4794

11.2 Services of the Standardized Master Interface (SMI)

11.2.1 Overview

Table 131 lists the SMI services available to gateway applications or other clients.
New or changed services are marked.

Table 131 SMI services

Service name	Master	M/O/C	Purpose
SMI_MasterIdentification	R	M	Universal service to identify any Master
SMI_PortConfiguration	R	M	Setting up port configuration
SMI_ReadbackPortConfiguration	R	M	Retrieve current port configuration
SMI_PortStatus	R	M	Retrieve port status
SMI_DSToParServ	R	M	Transfer Data Storage to parameter server
SMI_ParServToDS	R	M	Transfer Parameter server to Data Storage
SMI_DeviceWrite	R	M	ISDU transport to Device
SMI_DeviceRead	R	M	ISDU transport from Device
SMI_ParamWriteBatch	R	O	Batch ISDU transport of parameters (write)
SMI_ParamReadBatch	R	O	Batch ISDU transport of parameters (read)
SMI_DeviceEvent	I	M	Universal "Push" service for Device Events
SMI_PortEvent	I	M	Universal "Push" service for port Events
SMI_PDIn	R	M	Retrieve PD from InBuffer
SMI_PDOut	R	M	Set PD in OutBuffer
SMI_PDInOut	R	M	Retrieve In- and OutBuffer
SMI_WMasterConfiguration	R	M	Setting up W-Master configuration
SMI_ReadbackWMasterConfiguration	R	M	Retrieve current W-Master configuration
SMI_WTrackStatus	R	M	Retrieve W-Track status
SMI_WPortPairing	R	M	Pair or Unpair a configured device
SMI_WScan	R	M	Set a scan request
SMI_WScanStatus	R/I	M	Retrieve scan status results
SMI_WQualityStatus	R	M	Retrieve current wireless quality

SMI services instance start count from 1, following conversion rules with lower layers
SMI PortNumber = W-Port + 1
SMI TrackNumber = Track_N + 1

11.2.2 SMI W-MasterConfiguration

This service allows to set the general configuration of the W-Master.
Table 132 shows the structure of the service.

Table 132 SMI_WMasterConfiguration

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (0x00)	M	
ExpArgBlockID (VoidBlock: 0xFFF0)	M	
ArgBlockLength	M	
ArgBlock (0x0200)	M	
Result (+)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0x0200)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0x0200)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID "VoidBlock" (0xFFF0, see Annex E.1)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock to be "pushed"

ArgBlock

This parameter contains an ArgBlock of the WMasterConfigList family, e.g., 0x0200 (see Table 198)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x8200)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (0xFFF0)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x0200)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, See [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)

ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

ARGBLOCK_INCONSISTENT (incorrect ArgBlock content type)

11.2.3 SMI_ReadbackWMasterConfiguration

This service allows to readback the W-Master Configuration which was written by SMI_WMasterConfiguration service.

Table 133 shows the structure of the service.

Table 133 SMI_ReadbackWMasterConfiguration

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (0x00)	M	
ExpArgBlockID (e.g., 0x0200)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFF0)	M	
Result (+)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID of the WMasterConfigList family, e.g., 0x0200 (see Table 198)

ArgBlockLength

This parameter contains the length of the "VoidBlock" ArgBlock.

ArgBlock

This parameter contains an ArgBlock "VoidBlock" (0xFFF0, see [1] Annex E.17)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Annex E.1)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock.

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, See [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)

ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

11.2.4 SMI_WTrackStatus

This service allows for retrieval of the effective status of the W-Master**.**
Table 134 shows the structure of the service.

Table 134 SMI_WTrackStatus

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (“0” → All Tracks / 1.. Single W-Track)	M	
ExpArgBlockID (e.g., 0x9202)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFF0)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

If PortNumber is 0, the status of all Tracks is requested. For values greater then 0, a specific W-Track is requested.

ExpArgBlockID

This parameter contains the ArgBlockID of the WTrackStatusList family, e.g., 0x9202 (see Table 198)

ArgBlockLength

This parameter contains the length of the "VoidBlock" ArgBlock.

ArgBlock

This parameter contains an ArgBlock "VoidBlock" (0xFFF0, see [1] Annex E.17)

Result (+):

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

ClientID

TrackNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Annex E.1)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

TrackNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock.

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, see [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

11.2.5 SMI_WScan

This service starts a new scan request.

Table 135 shows the structure of the service.

Table 135 SMI_WScan

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (0x00)	M	
ExpArgBlockID (VoidBlock: 0xFFF0)	M	
ArgBlockLength	M	
ArgBlock (0x8201)	M	
Result (+)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0x8201)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0x8201)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

This parameter contains a virtual Port addressing the entire Master unit (0x00)

ExpArgBlockID

This parameter contains the ArgBlockID "VoidBlock" (0xFFF0, see Annex E.1)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock to be "pushed".

ArgBlock

This parameter contains an ArgBlock of the WScanList family, e.g., 0x8201 (see Table 198)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x8201)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (0xFFF0)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x8201)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock.

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, see [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)
ARGBLOCK_INCONSISTENT (incorrect ArgBlock content type)

11.2.6 SMI_WPortPairing

This service starts the pairing process of the requested port.
Table 136 shows the structure of the service.

Table 136 SMI_WPortPairing

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber	M	
ExpArgBlockID (VoidBlock: 0xFFF0)	M	
ArgBlockLength	M	
ArgBlock (0x7200)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0x7200)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0x7200)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID "VoidBlock" (0xFFF0, see Annex E.1)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock to be "pushed”.

ArgBlock

This parameter contains an ArgBlock of type WPortPairing, e.g., 0x7200 (see Table 198)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x7200)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (0xFFF0)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0x7200)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock.

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, see [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)
ARGBLOCK_INCONSISTENT (incorrect ArgBlock content type)

11.2.7 SMI_WScanStatus

This service allows for retrieval of the scan status. It will include all results of the last Scan. Table 137 shows the structure of the service.

Table 137 SMI_WScanStatus

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (0x00)	M	
ExpArgBlockID (e.g., 0x9201)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFF0)	M	
Result (+)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber (0x00)		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID of the WScanStatusList family, e.g., 0x9201 (see Table 198)

ArgBlockLength

This parameter contains the length of the "VoidBlock" ArgBlock.

ArgBlock

This parameter contains an ArgBlock "VoidBlock" (0xFFF0, see [1] Annex E.17)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Annex E.1)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock.

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, see [1] Annex E.18)

Permitted values:

ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

11.2.8 SMI_WQualityStatus

This service allows for retrieval of the wireless quality status.

Table 138 shows the structure of the service.

Table 138 SMI_WQualityStatus

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber (“0” → All Ports / 1.. Single Port)	M	
ExpArgBlockID (e.g., 0x9203)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFF0)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID of the WQualityStatusList family, e.g., 0x9203 (see Table 198)

ArgBlockLength

This parameter contains the length of the "VoidBlock" ArgBlock.

ArgBlock

This parameter contains an ArgBlock "VoidBlock" (0xFFF0, see [1] Annex E.17)

Result (+):

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

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock.

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Annex E.1)

Result (-):

This selection parameter indicates that the service request failed.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF, see [1] Annex E.18)

Permitted values:
ARGBLOCK_NOT SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

11.3 Configuration Manager (CM)

11.3.1 General

See IO-Link specification 11.3.1 in [1].

11.3.2 Coordination of Master applications

Due to IO-Link Wireless specific technology some SMI services are not supported. Table 139 shows not supported SMI services.

Table 139 Not supported SMI services

Service name	Master	M/O/C	Purpose
SMI_PortPowerOffOn	R	O	PortPowerOffOn
SMI_PDInIQ	R	C	Process data in at I/Q (Pin 2 on M12)
SMI_PDOutIQ	R	C	Process data out at I/Q (Pin 2 on M12)
SMI_PDReadbackOutIQ	R	C	Retrieve process data out at I/Q (Pin 2 on M12)

Figure 101 shows the sequence diagram of configuration manager actions.

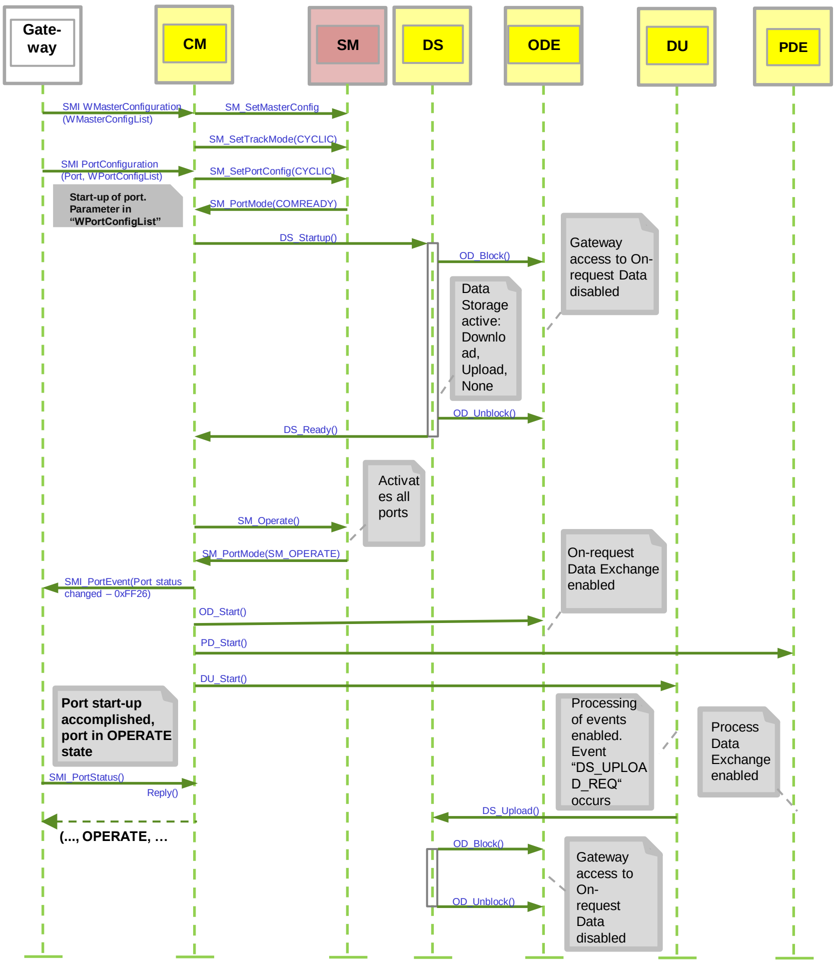


Figure 101 Sequence diagram of configuration manager actions

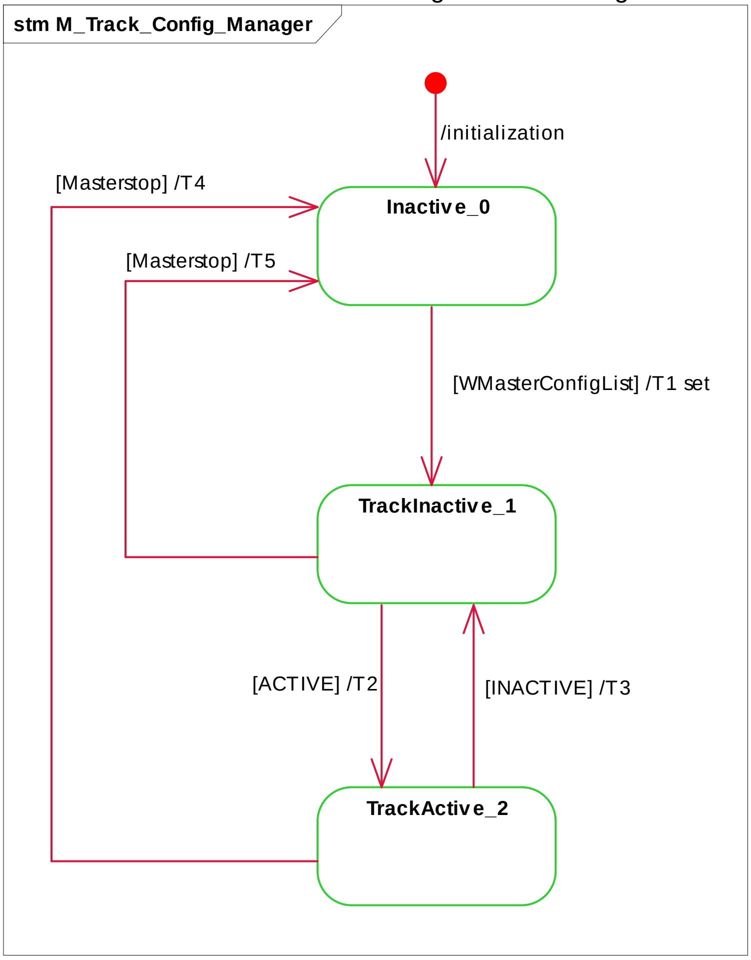
11.3.3 State machines of the Configuration Manager

11.3.3.1 General

The Configuration Manager of a W-Master incorporates two state machines, one for the W-Track configuration and one for the W-Port configuration.

5089 #### 11.3.3.2 State machine of the W-Track Configuration Manager

5090 Figure 102 shows the state machine of the W-Track configuration manager.



5091
5092 **Figure 102 State machine of the W-Track Configuration Manager**
5093

Table 140 shows the state transition tables of the W-Track Configuration Manager.

Table 140 State transition tables of the W-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 W-Track in operating mode (CYCLIC or ROAMING).
TrackACTIVE_2	W-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 W-Track, dependent on the W-Tracks on a W-Master (W-Track number 0 up to 4). Each W-Track shall use the same MasterID and Blocklist.
T2	1	2	Invoke SM_SetTrackMode(CYCLIC or ROAMING) depending on W-Track configuration.
T3	1	2	Invoke SM_SetTrackMode(STOP).
T4	2	0	Invoke SM_SetTrackMode(STOP) for all W-Tracks.
T5	1	0	See T4.

INTERNAL ITEMS	TYPE	DEFINITION
WMasterConfigList set	Guard	Values of “WMasterConfigList” have been set
ACTIVE	Guard	W-Track activated in operating mode (CYCLIC or ROAMING)
INACTIVE	Guard	W-Track deactivated
Masterstop	Guard	Master stopped

11.3.3.3 State machine of the W-Port Configuration Manager

Figure 103 shows the state machine of the W-Port Configuration manager. In general, states and transitions correspond to those of the message handler: STARTUP, PREOPERATE (fault or Data Storage), and at the end OPERATE. Dedicated "SM_PortMode" services are driving the transitions (see 9.3.2.10). Configuration Manager can receive the information COMLOST from Port x Handler through "SM_PortMode" at any time. It also can receive a service "SMI_PortConfiguration" from the gateway application with changed values in "WPortConfigList" at any time (see 11.2). Via service “SMI_ParServToDS, it can also receive a Data Storage object with a changed parameter set from the gateway application triggering action in the Configuration Manager if Data Storage is activated. Port x is started/restarted in all cases. Figure 104 together with Table 142 also shows transitions leading to corresponding changes in “PortStatusInfo” of ArgBlock “WPortStatusList” (see Table 199). Based on these transitions, Events are triggered via SMI_PortEvent. For details see Annex D.3 in [1].

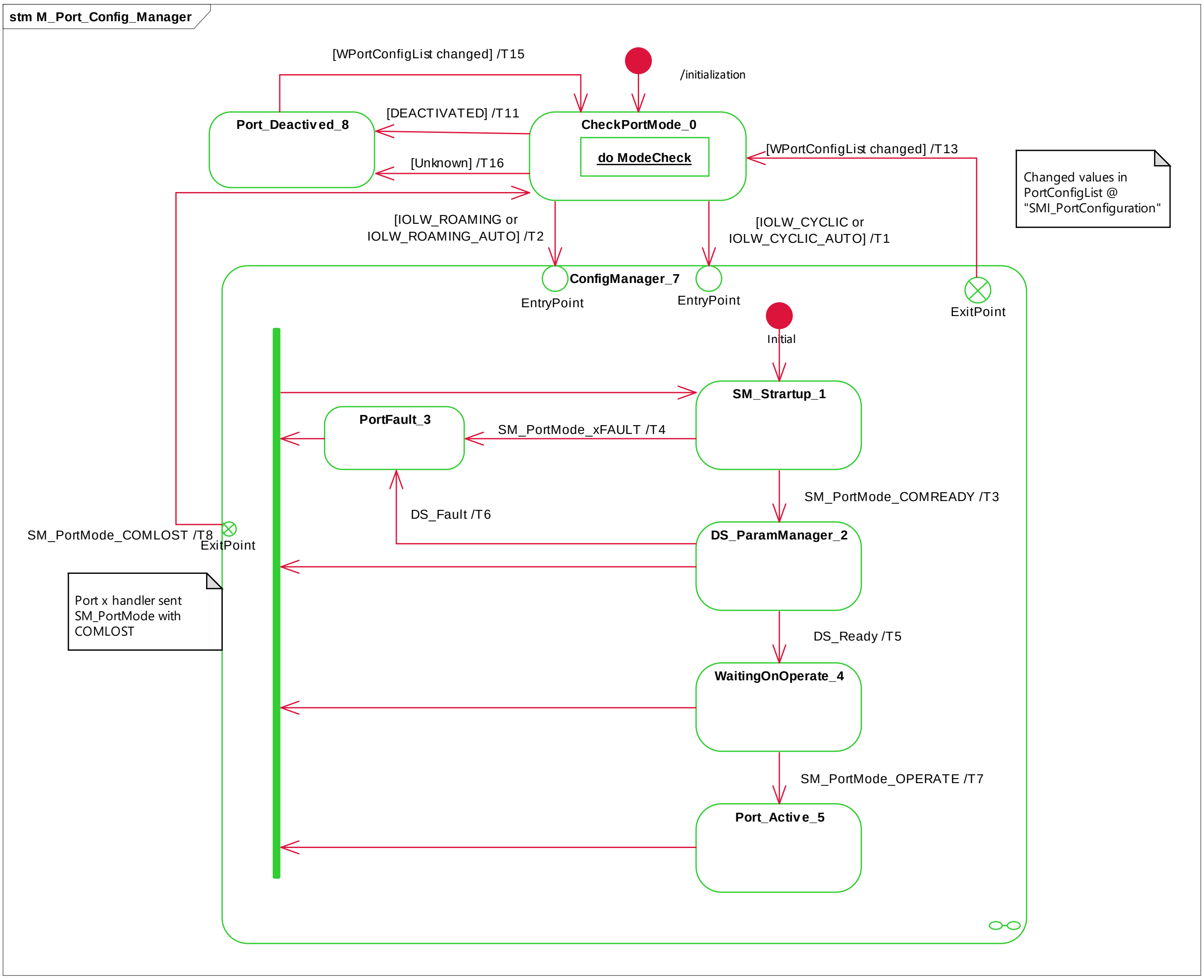


Figure 103 State machine of the W-Port Configuration Manager

Table 141 shows the state transition tables of the W-Port Configuration Manager.

Table 141 State transition tables of the W-Port Configuration Manager

STATE NAME	STATE DESCRIPTION
CheckPortMode_0	Check "Port Mode" element in parameter "WPortConfigList" (see 11.2 and Annex E.1.2)
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 114).
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 faults REVISION_FAULT, COMP_FAULT, SERNUM_FAULT, or DS_FAULT or PORT_DIAG occurred.
WaitingOnOperate_4	Waiting for SM to switch to OPERATE.
Port_Active_5	W-Port is in OPERATE mode. The gateway application is exchanging Process Data and ready to send or receive On-request Data.
ConfigManager_7	This superstate handles Port communication operations and allows all states inside to react on COMLOST via SM_PortMode service. A Port restart is managed inside the superstate triggered by the DS_Change signal (see Table 125 in [1]).
Port_Deactivated_8	Port is in DEACTIVATED mode.
Key: xFAULT: REVISION_FAULT or COMP_FAULT or SERNUM_FAULT	

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	0	7	Invoke DS-Delete if identification (VendorID, DeviceID) within DS is different to configured port identification. Invoke SM_SetPortConfig_CYCLIC
T2	0	7	Invoke DS-Delete if identification (VendorID, DeviceID) within DS is different to configured port identification. Invoke SM_SetPortConfig_ROAMING
T3	1	2	DS_Startup: The DS state machine is triggered. Update parameter elements of "WPortStatusList": - PortStatusInfo = NOT_AVAILABLE - RevisionID = (real) RRID - VendorID = (real) RVID - DeviceID = (real) RDID - WMasterCycleTimeOut = value see C.4.12 - WMasterCycleTimeIn = value see C.4.12 - Port QualityInfo = invalid
T4	1	3	Update parameter elements of "WPortStatusList": - PortStatusInfo = PORT_DIAG - RevisionID = (real) RRID - VendorID = (real) RVID - DeviceID = (real) RDID - Port QualityInfo = invalid
T5	2	4	SM_Operate
T6	2	3	Data Storage failed. Rollback to previous parameter set. Update parameter elements of "WPortStatusList": - PortStatusInfo = PORT_DIAG - RevisionID = (real) RRID - VendorID = (real) RVID - DeviceID = (real) RDID - Port QualityInfo = invalid
T7	4	5	Update parameter elements of "WPortStatusList": - PortStatusInfo = OPERATE - RevisionID = (real) RRID - VendorID = (real) RVID - DeviceID = (real) RDID - Port QualityInfo = x
T8	1,2,3,4,5	0	Update parameter elements of "WPortStatusList": - PortStatusInfo = NO_DEVICE - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)
T11	0	8	Invoke DS-Delete. SM_SetPortConfig_INACTIVE. Update parameter elements of "WPortStatusList": - PortStatusInfo = DEACTIVATED - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)

T13	1,2,3,4,5	0	Update parameter elements of "WPortStatusList": - PortStatusInfo = NOT_AVAILABLE - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)
T14	1,2,3,4,5	1	SM_SetPortConfig_CYCLIC Update parameter elements of "WPortStatusList": - PortStatusInfo = NOT_AVAILABLE - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)
T15	8	0	Update parameter elements of "WPortStatusList": - PortStatusInfo = NOT_AVAILABLE - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)
T16	0	8	Invoke DS-Delete. SM_SetPortConfig_INACTIVE. Update parameter elements of "WPortStatusList": - PortStatusInfo = DEACTIVATED - RevisionID = 0 - VendorID = 0 - DeviceID = 0 - Port QualityInfo = invalid Delete DiagEntries (SOURCE = Device) in WPortStatusList (see Table 200)

5122

INTERNAL ITEMS	TYPE	DEFINITION
WPortConfigList changed	Guard	Values of "WPortConfigList" have changed
DS_Ready	Signal	Data Storage sequence (upload, download) accomplished; see Table 125 in [1]
DS_Fault	Signal	See Table 125 in [1]
DEACTIVATED	Guard	See Table 199
IOLW_CYCLIC	Guard	See Table 199
IOLW_CYCLIC_AUTO	Guard	See Table 199
IOLW_ROAMING	Guard	See Table 199
IOLW_ROAMING_AUTO	Guard	See Table 199
DS_Change	Signal	See Table 125 in [1]]
DS_Active	Guard	Port configured to "Backup + Restore" (3) or "Restore" (4); see Table 197

5123

5124 State "CheckPortMode_0" contains an activity with complex logic for checking the Port mode within a
5125 received Port configuration (see Table 197). Figure 104 shows this activity within the context of the state
5126 machine in Figure 103.

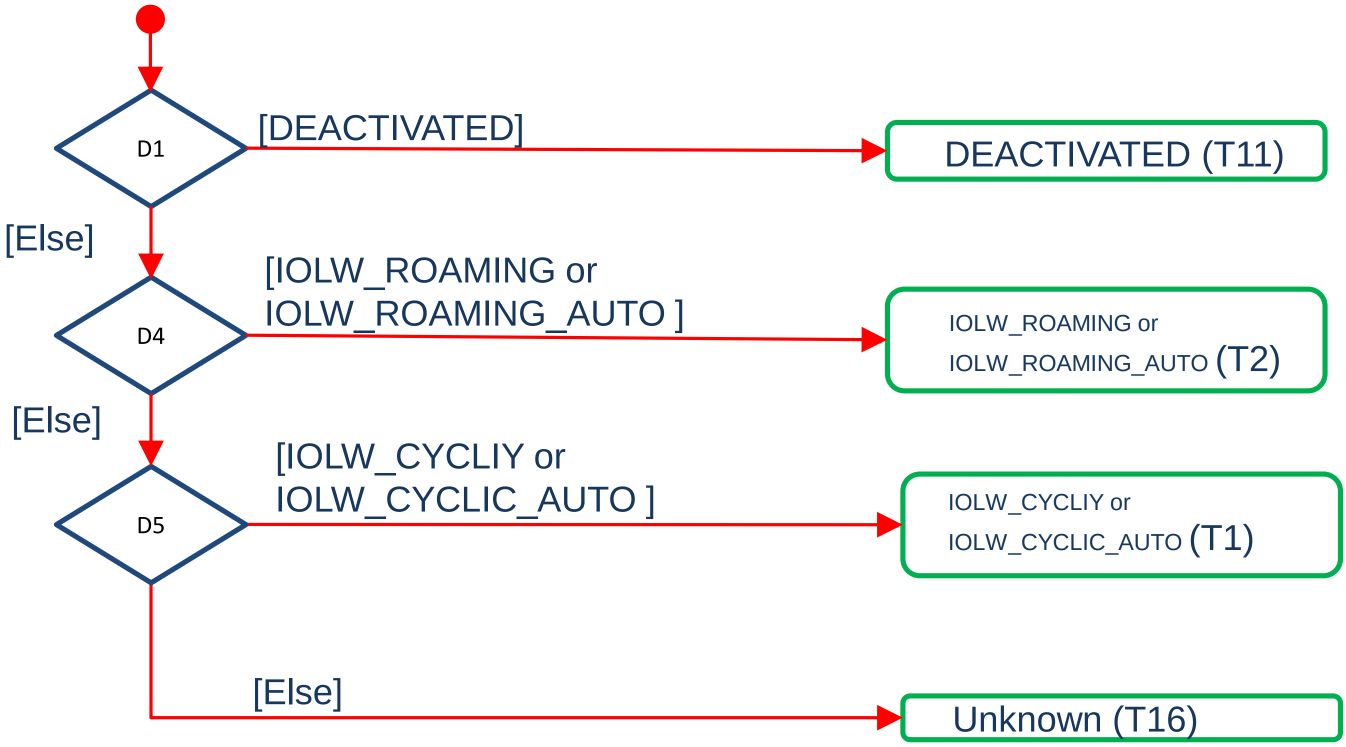


Figure 104 Activity for state "CheckPortMode_0"

11.4 Data Storage (DS)

See IO-Link specification 11.4 in [1].

Due to IO-Link Wireless specific technology some trigger signal in state machines differ from IO-Link specification.

Table 142 shows different triggers in state machines.

Table 142 Different triggers in state machines

State machine	Transition	Trigger IO-Link	Trigger IO-Link Wireless
Main state machine of Data Storage mechanism	T6	COMx	ACTIVE
	T8	NoCOMx	COMLOST
Data Storage submachine “Upload_7”	T33	COMx_ERROR	COMLOST
	T34	COMx_ERROR	COMLOST
	T36	COMx_ERROR	COMLOST
Data Storage submachine “Download_10”	T40	COMx_ERROR	COMLOST
	T42	COMx_ERROR	COMLOST

11.5 On-request Data Exchange (ODE)

See IO-Link specification 11.5 in [1].

11.6 Diagnosis Unit (DU)

See IO-Link specification 11.6 in [1].

11.7 Process Data Exchange (PDE)

11.7.1 General

See IO-Link specification 11.7.1 in [1].

11.7.2 Process Data input mapping

See IO-Link specification 11.7.2 in [1].

The descriptions under Port Modes "IOL_MANUAL" or "IOL_AUTOSTART" applies to all Wireless Port Modes.

11.7.3 Process Data output mapping

See IO-Link specification 11.7.3 in [1].

The descriptions under Port Modes "IOL_MANUAL" or "IOL_AUTOSTART" applies to all Wireless Port Modes.

Port Mode "DO_C/Q" is not supported in IO-Link Wireless.

11.7.4 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 105

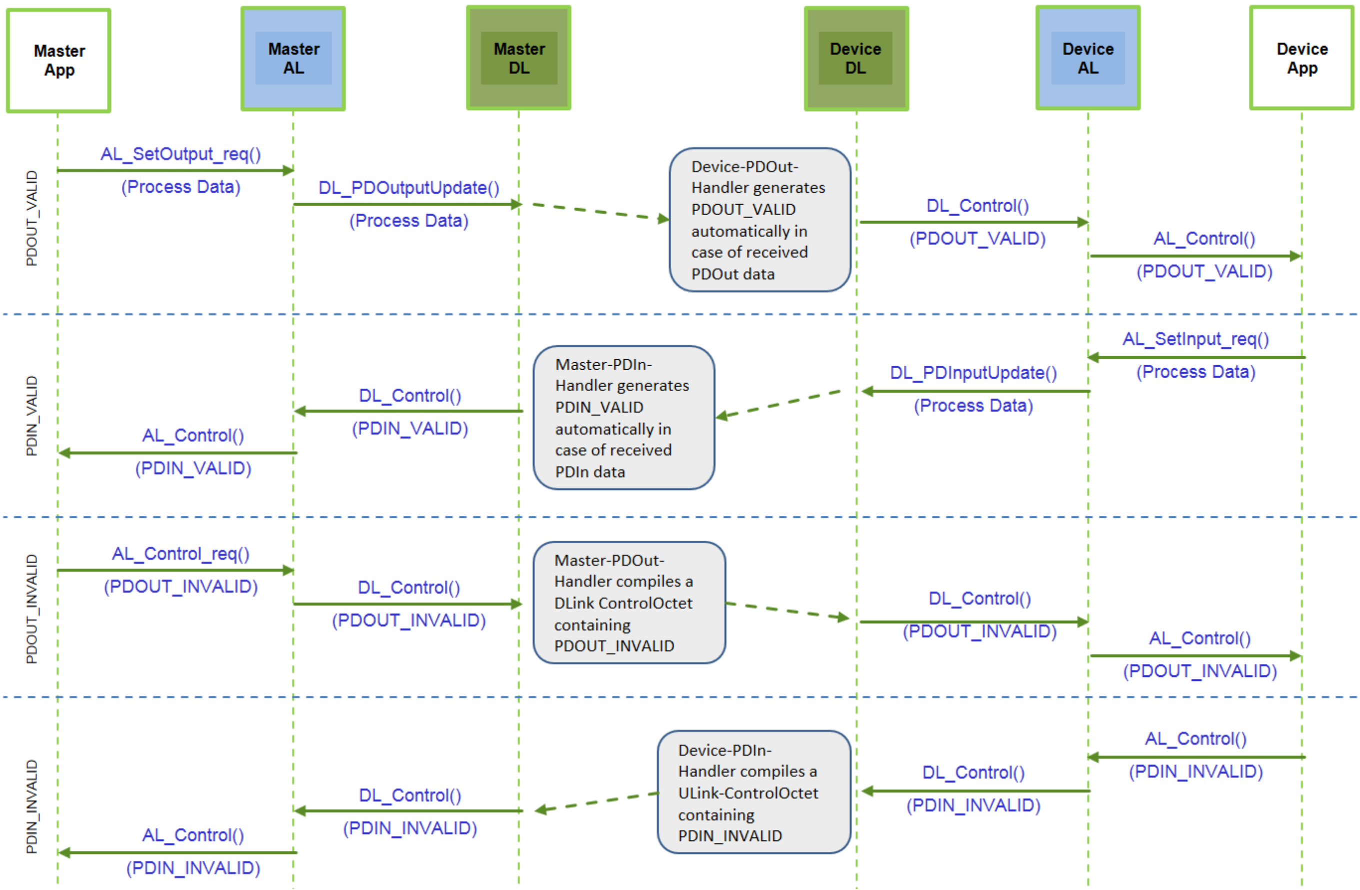


Figure 105 Propagation of PD qualifier status between W-Master and W-Device

The W-Master informs the W-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 A.9 and sequence chart Figure 105.

11.8 Port and Device Configuration Tool (PDCT)

11.8.1 General

Figure 95 in [1] and Figure 112 in [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.

11.8.2 Basic layout examples

Figure 106 shows one example of a PDCT display layout.

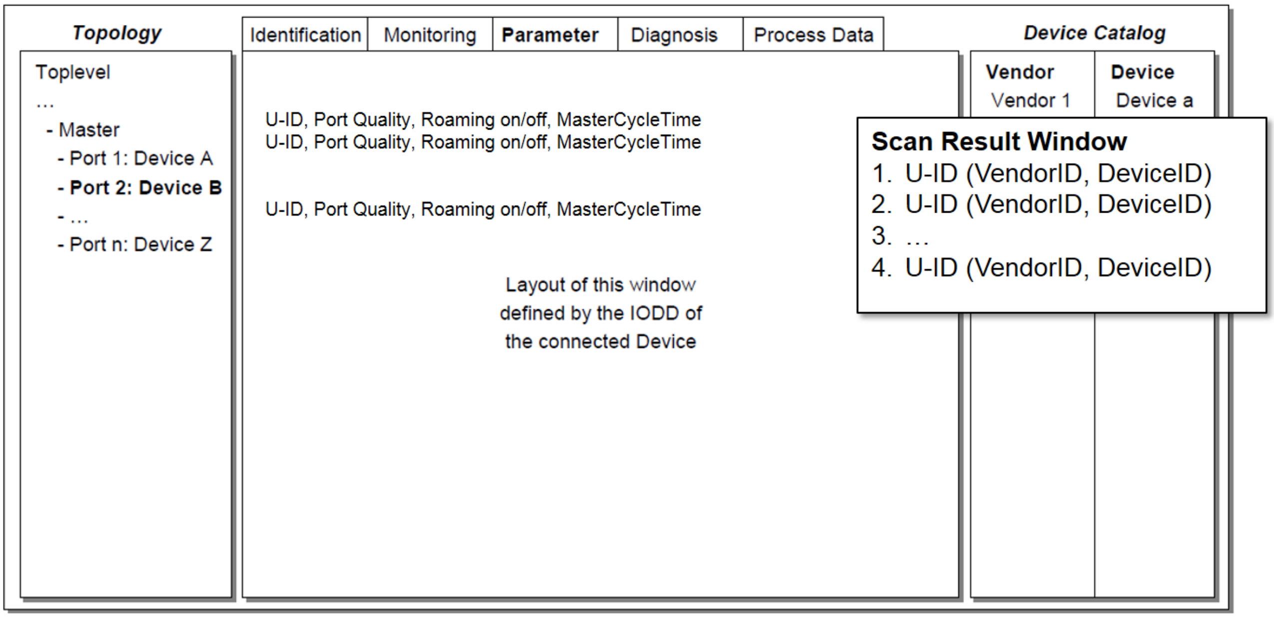


Figure 106 Example 1 of a PDCT display layout

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 107 shows another example of a PDCT display layout.

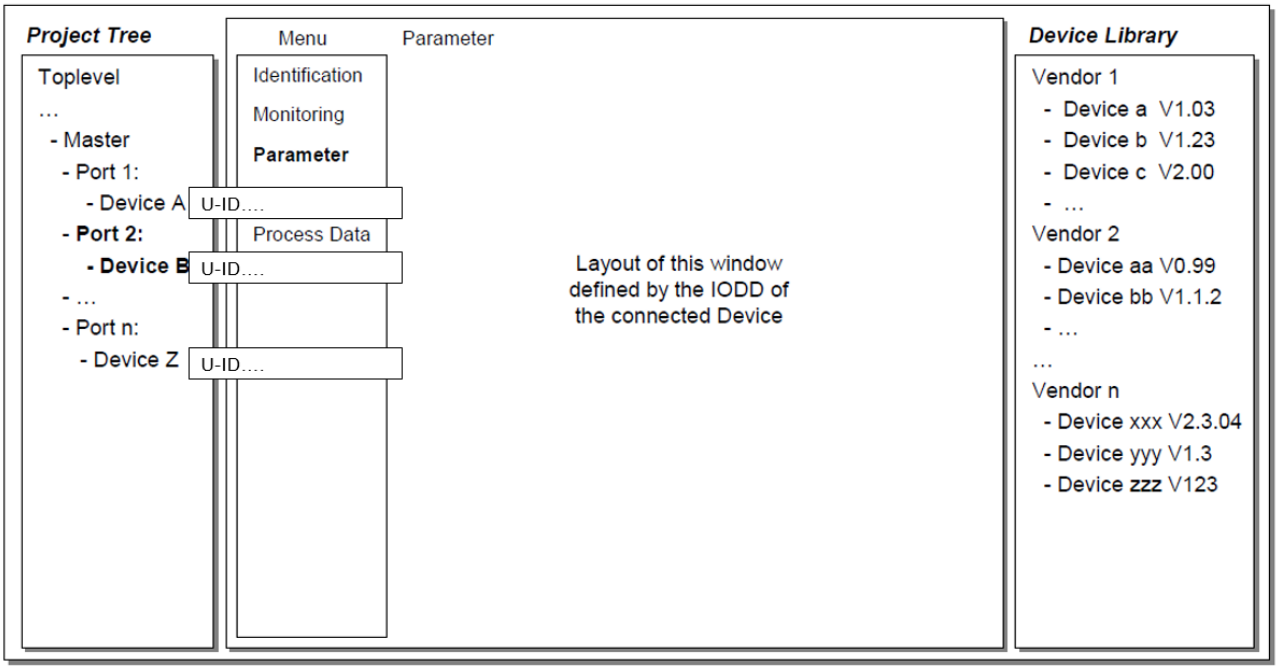


Figure 107 Example 2 of a PDCT display layout

NOTE: Further information can be retrieved from IEC/TR 62453-61.

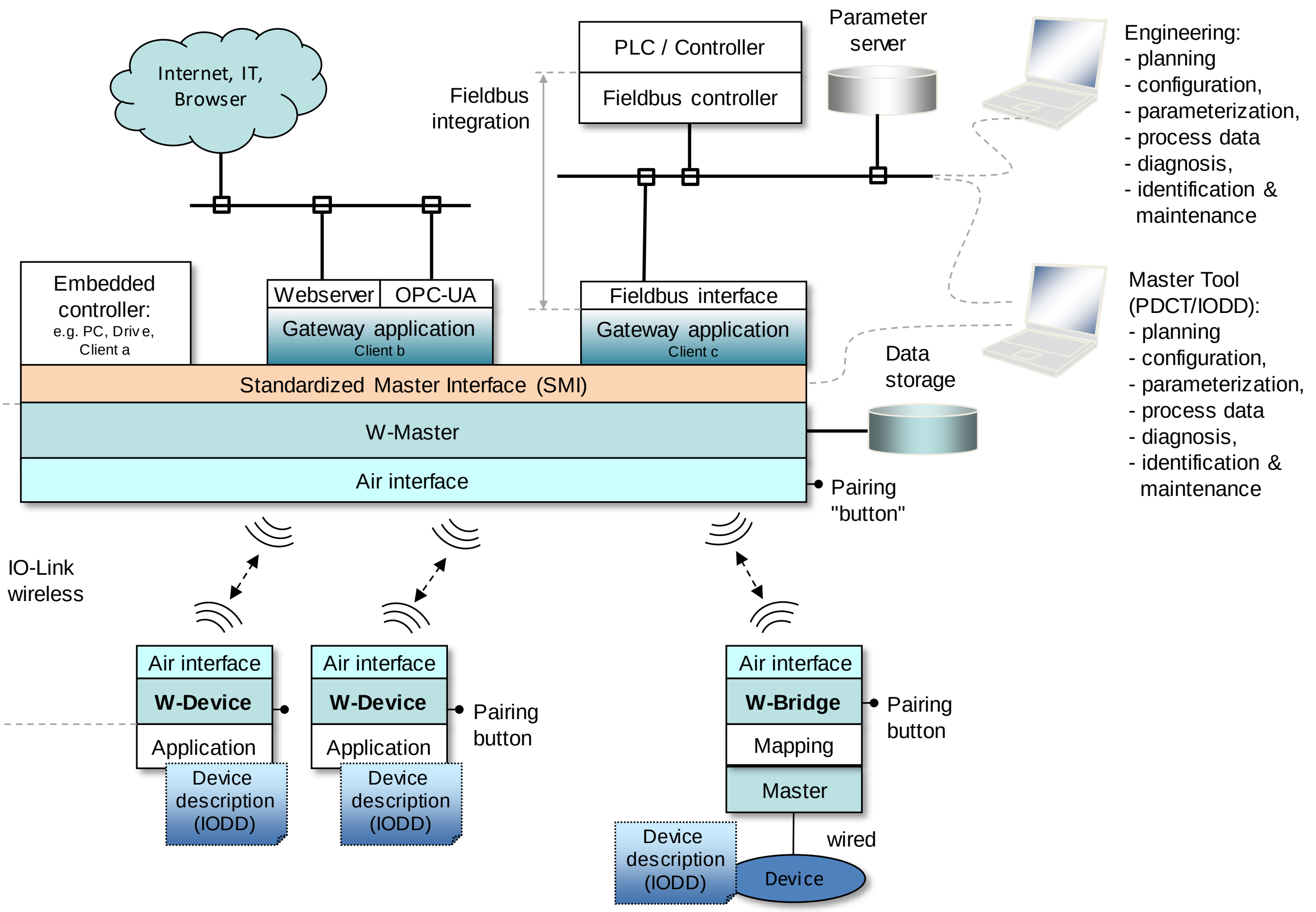
11.9 Gateway application

11.9.1 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 did not 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 [1].

Figure 108 Generic relationship of W-Master and automation technology



11.9.2 Changing W-Device configuration including Data Storage

After each change of W-Device configuration/parameterization (CVID and/or CDID, see 9.2.2.2 in [1]), the associated previously stored data set within the W-Master shall be cleared or marked invalid via the variable DS_Delete.

11.9.3 Parameter server and recipe control

The W-Master may combine the entire parameter sets of the connected W-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 document.

11.9.4 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 document to implement EAP in the IOLW-Master gateway application layer.

11.10 Human machine Interface (HMI), Faulty W-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 4.4.3.2). 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 109 shows the faulty W-Device replacement.

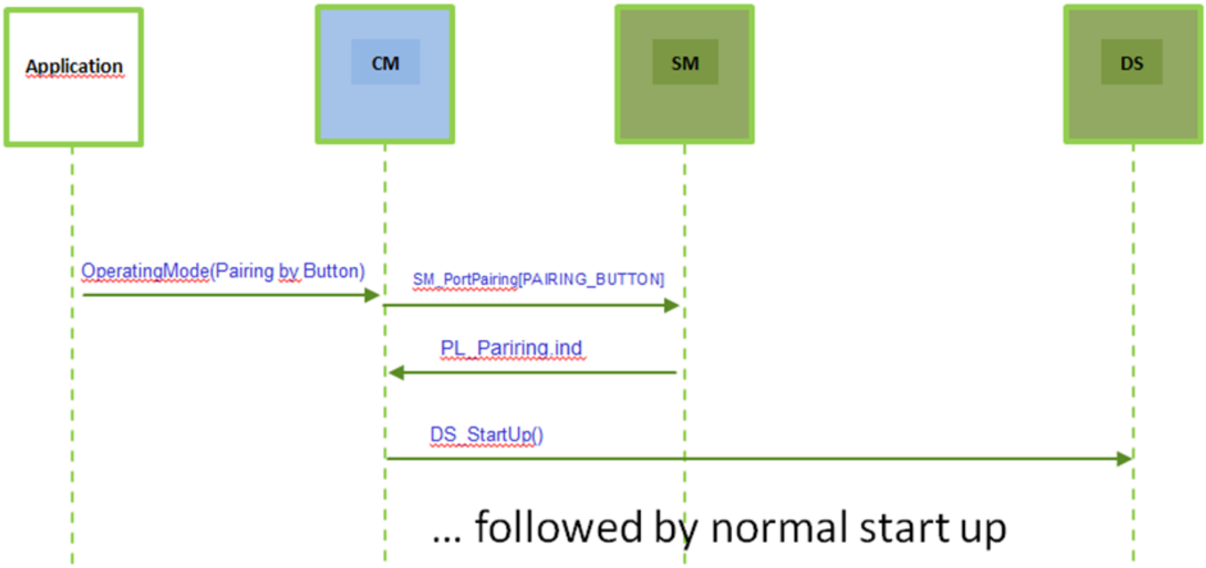


Figure 109 Faulty W-Device replacement

12 W-Bridge

12.1 Overview

A W-Bridge is an adapter which provides IO-Link Wireless functionality to an IO-Link Device. This clause will specify the W-Bridge Application.

The W-Bridge is comprised of a W-Device-Part (see Figure 19) for communication with a W-Master, an IOL-Master-Part (see Figure 9 in [1]) for communication with the IO-Link Device, and a W-Bridge Application to handle forwarding and conversions between the parts. The modifications of the parts required for the W-Bridge are depicted in this clause. The W-Bridge structure is depicted in Figure 110.

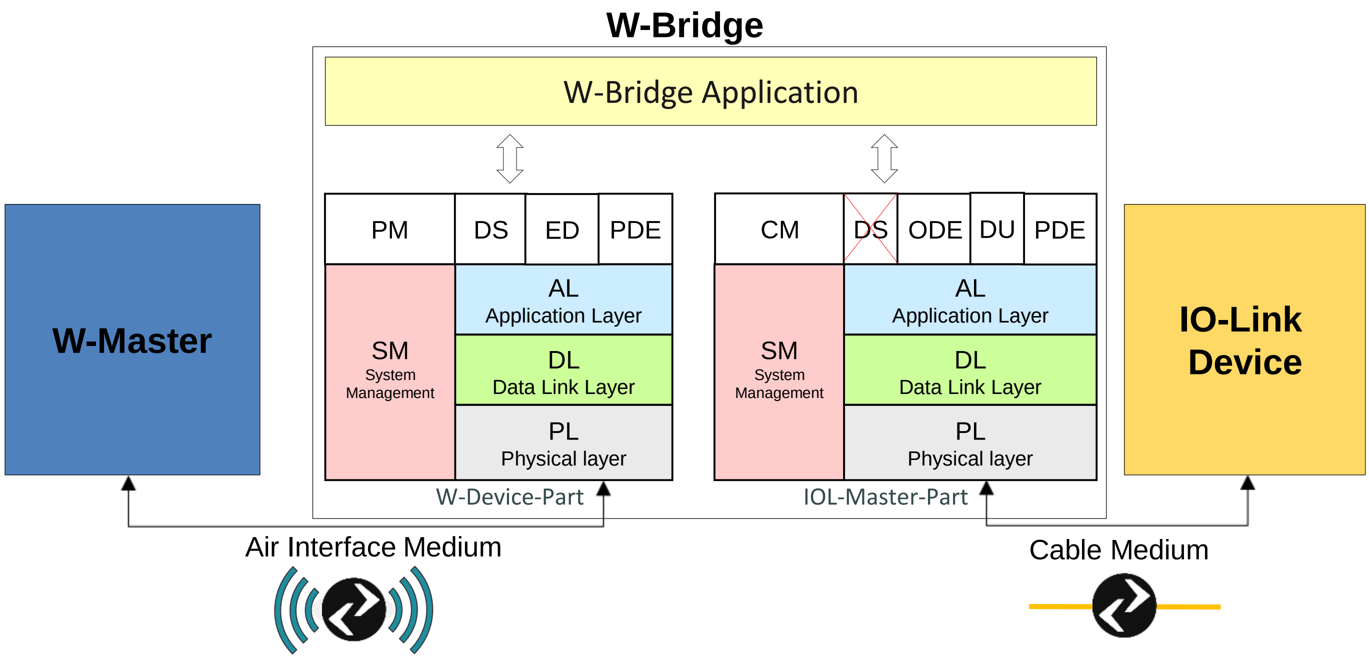


Figure 110 Structure of a W-Bridge

12.2 Process Data Exchange (PDE)

12.2.1 General

The Process Data Exchange in IO-Link Wireless and IO-Link is identical from upper layer view. The Process Data of a W-Bridge is defined by the connected IO-Link Device and forwarded to the W-Master by the W-Bridge application. SIO of a connected IO-Link Device is not supported. A W-Bridge standalone does not support Process Data.

12.2.2 Process Data mapping

Input and output Process Data are mapped directly between AL services of W-Device-Part and IOL-Master-Part in the W-Bridge. For handling of Process Data invalid/valid qualifier status see clause 11.7.3 in [1] and clause 11.7.4 in [1], for AL service see clause 8.2.13 and clause 8.2.12 in [1].

12.3 On-request Data Exchange (ODE)

12.3.1 General

The On-request Data Exchange in IO-Link Wireless and IO-Link is identical from upper layer view. ISDU requests to the connected IO-Link Device shall be forwarded properly.

12.3.2 Wireless Parameters

A W-Bridge shall always allow the configuration of its W-Parameters (see Figure 139).

12.3.3 Parameters of connected IO-Link Device

All Read or write requests shall be forwarded unchanged from W-Bridge to the IO-Link Device, except of the W-Parameters and the Direct Parameter Page 1, see 12.6.

12.4 Data Storage

If an IO-Link Device is connected to the W-Bridge Data Storage shall be handled transparently via ISDU. Therefore, the W-Device-Part and IOL-Master-Part shall disable the Data Storage components.
A W-Bridge in standalone mode shall support Data Storage identical to a W-Device.
The W-Master stores either the IO-Link Device parameters or the W-Bridge standalone parameters according to the W-Master port configuration.

12.5 Diagnosis Unit (DU)

12.5.1 General

Events of a connected IO-Link Device shall be handled transparent. This includes Port-Events, generated by the IOL-Master-Part of the W-Bridge as well as the Events generated by the application of the IO-Link Device.
The Diagnosis Unit (DU) handles Device or Port specific Events from the IOL-Master-Part in the W-Bridge.
The W-Bridge application shall support the IO-Link Wireless Events, see Table 196, Table 197, Annex D.1. and Table D.1 in [1].

12.5.2 Device Events (Wireless)

The IO-Link Wireless Device Events are generated by the W-Device-Part of a W-Bridge according to Table 196 and Annex D.1

12.5.3 Device Events

IO-Link Wireless Device Events of a connected IO-Link Device (see D.2 in [1]) shall be forwarded via the W-Bridge to the W-Master.

12.5.4 Port Events

For proper behavior of the W-Master in combination with the Port-Events generated by the IOL-Master-Part the Port-Events (see Table D.2 in [1]) shall be transmitted as regular event with SOURCE=Master/Port via W-Bridge to the W-Master.
The Port Events in Table 143 shall be blocked by the W-Bridge application and not forwarded to the W-Master.

Table 143 Blocked Port Event Codes

Port Event Codes of IO-Link Master-Part	Description	Remarks
0x1800	No Device (communication)	Device communication handled by W-Bridge application
0x1801	Startup parametrization error	Startup is handled by W-Master
0x1802 to 0x1803	Inspection Level mismatch	Startup is handled by W-Master
0x1809 to 0x180D, 0xFF23 to 0xFF27	Data Storage	Data Storage shall be disabled (see 12.4)
0xFF21	Device plugged in	Device communication handled by W-Bridge application
0xFF22	Device communication lost	Device communication handled by W-Bridge application

12.6 Design Rules

12.6.1 General

The following design rules are defined to achieve a common behavior of the W-Bridge application.

12.6.2 Timing constraints

12.6.2.1 Cycle times

As the Definition of MasterCycleTime and MinCycleTime differ between IO-Link Wireless and IO-Link (see Annex C.2.2 and Annex B.1.3 in [1]) a timing conversion takes place in the W-Bridge.

- The IOL-Master-Part in the W-Bridge shall support COM1, COM2 and COM3.
- The IOL-Master-Part in the W-Bridge shall use MinCycleTime of the connected IO-Link Device as MasterCycleTime. There may be an additional handling time to consider.

- The W-Device-Part in the W-Bridge shall use the fastest possible WMinCycleTimeOut and WMinCycleTimeIn (5ms).
- If PD Data In not fit in one uplink packet the PD Data must be segmented see clause 7.7.2. For calculation of WCycleTime see 9.3.2.3

12.6.2.2 Additional delay by the W-Bridge

The delay added by the W-Bridge to the ISDU response time of the device shall be as low as possible and shall not exceed 100 ms.

NOTE: Adding a delay could not be avoided but keeping it as low as possible will minimize the risk of interoperability problems.

12.6.3 Static behavior

12.6.3.1 W-Bridge with connected IO-Link Device

A W-Bridge with a connected IO-Link Device shall be transparent, except for:

- Wireless specific indexes (see Annex C.4)
- IO-Link Wireless Device events (see Figure 111)
- Direct Parameter Page 1 (see Figure 111)

Figure 111 shows the data object mapping for W-Bridge with connected IO-Link Device.

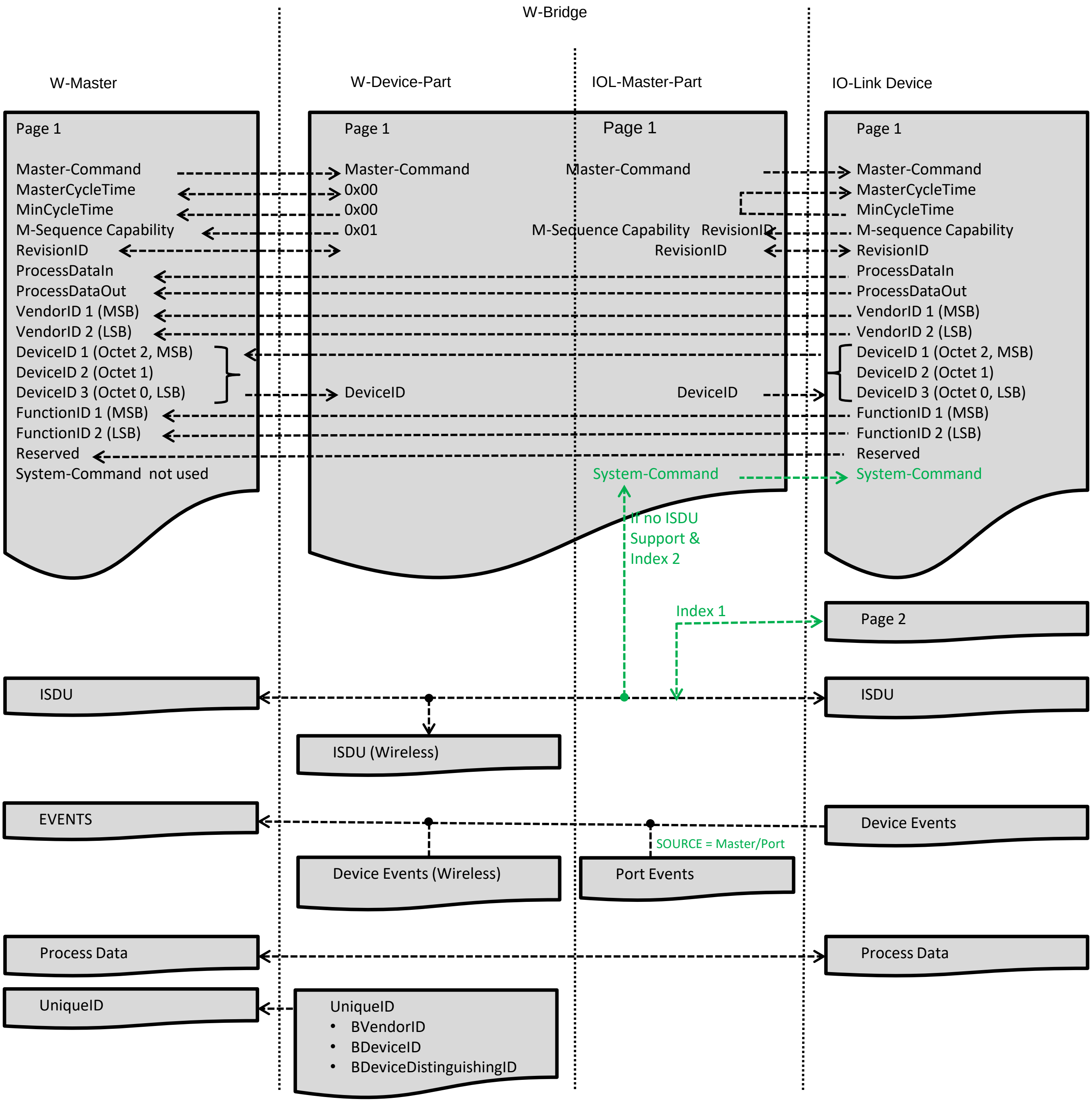


Figure 111 Data object mapping for W-Bridge with connected IO-Link Device

12.6.3.2 W-Bridge standalone

A W-Bridge without a connected IO-Link Device (standalone) shall allow configuration of the W-Bridge and read diagnosis data, process data shall not be available. The ISDU mechanism shall behave identical to a W-Device (see Annex C.1), enabling the W-Bridge to have device specific ISDU parameters (e.g., identification parameters). Operation in standalone mode requires a W-Bridge-specific IODD (see clause 12.7).

Figure 112 shows the data object mapping for W-Bridge stand alone.

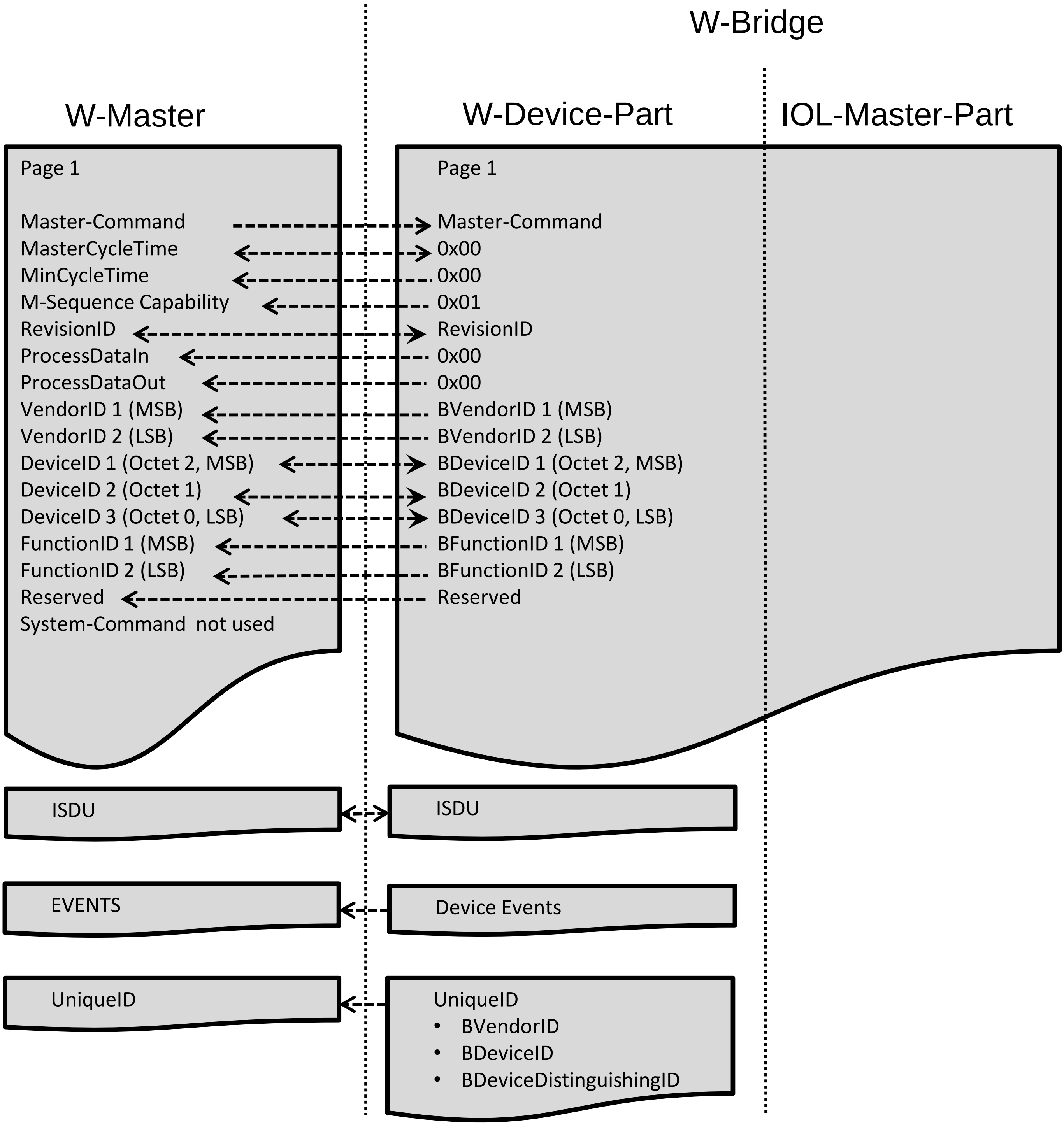


Figure 112 Data object mapping for W-Bridge stand alone

12.6.4 Dynamic behavior

This clause specifies the dynamic behavior of a W-Bridge. It covers the startup and interaction of a W-Bridge. Especially, the connection and disconnection of an IO-Link Device and/or W-Master will be covered.

In General, after power-up, the W-Bridge tries to connect to the IO-Link Device, in any case afterwards the connection to the W-Master can be established. If wireless connection with W-Master is lost the IO-Link Device remains in operation with invalid PDOut data until the wireless connection is reestablished. If wired connection with the IO-Link Device is lost the wireless communication is stopped and reestablished with the W-Bridge identification.

12.6.4.1 State machine of the W-Bridge Application

Figure 113 shows the state machine of the W-Bridge Application

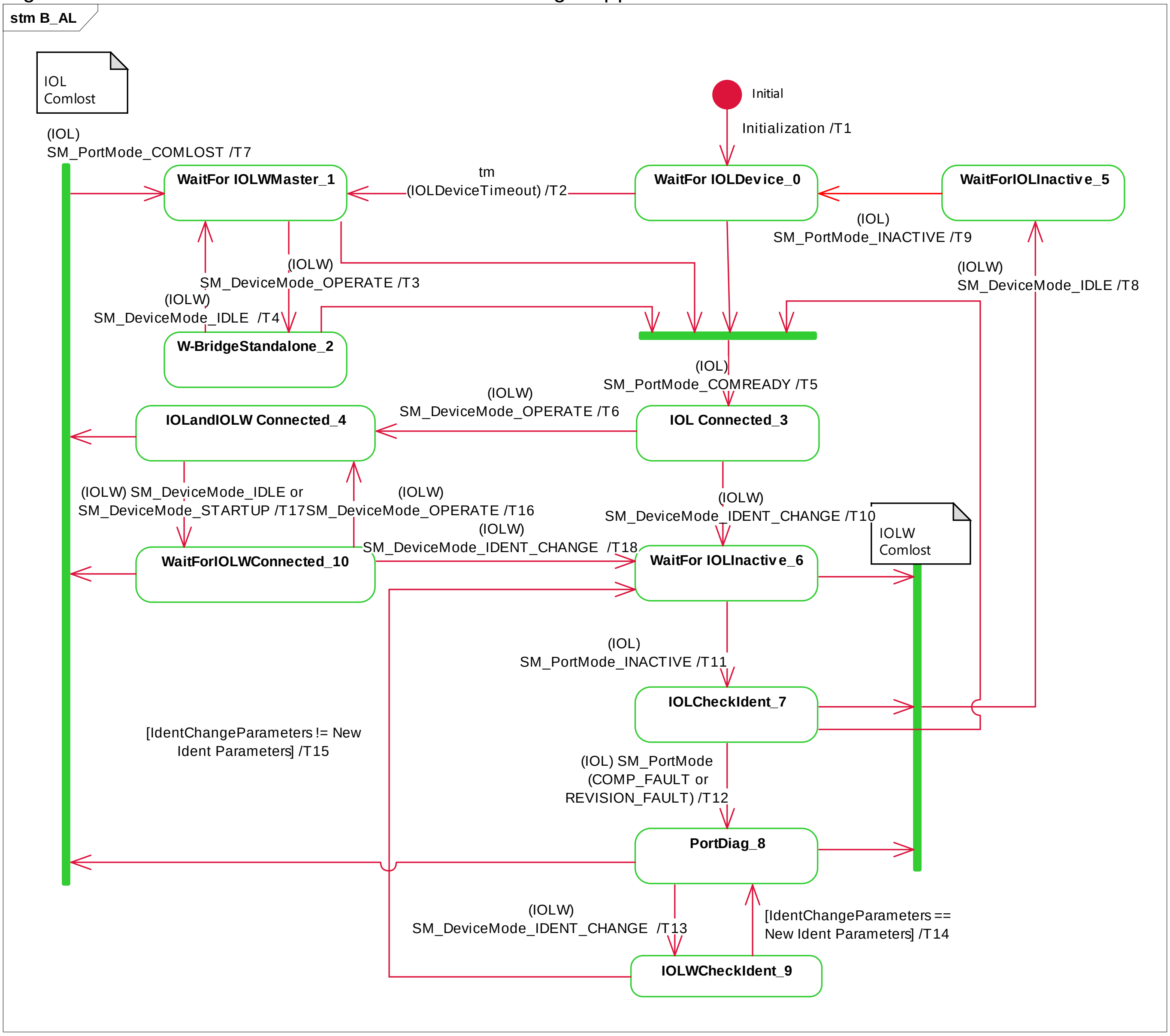


Figure 113 State machine of the W-Bridge Application

Table 144 shows the state transition tables of the W-Bridge Application.

Table 144 State transition tables of the W-Bridge Application

STATE NAME	STATE DESCRIPTION
WaitForIOLDevice_0	Waiting for IOL-Master-Part to establish communication to IO-Link Device
WaitForIOLWMaster_1	Waiting for W-Device-Part to be connected to W-Master
IOLWConnected_2	W-Bridge (standalone) in OPERATE
IOLConnected_3	IOL-Master-Part in PREOPERATE
IOLandIOLWConnected_4	W-Device-Part and IOL-Master-Part in OPERATE
WaitForIOLInactive_5	Waiting for IOL-Master-Part becomes inactive
WaitForIOLInactive_6	Waiting for IOL-Master-Part becomes inactive
IOLCheckIdent_7	Waiting for compatibility check of IOL-Master-Part
PortDiag_8	IOL-Master-Part and W-Device-Part stay in Port Diag until com lost
IOLWCheckIdent_9	Waiting for compatibility check of W-Device-Part
WaitForIOLWConnected_10	Waiting for W-Device-Part to be in OPERATE

TRANSITION	SOURCE STATE	TARGET STATE	ACTION /Remarks
T1	initial	0	Invoke SM_SetPortConfig_AUTOCOM (IOL) Start Timer "IOLDeviceTimeout"
T2	0	1	Invoke SM_SetDeviceIdent (VID, DID, FID) and SM_SetDeviceCom (ProcessDataIn, ProcessDataOut, RID) (W-Bridge) Invoke SM_SetDeviceMode_ESTABCOM (IOLW) ConnectionStatus = 0x00: No IOL-Device connected
T3	1	2	*IOLW Operate in standalone mode*
T4	2	1	-
T5	0, 1, 2, 7	3	Invoke SM_SetDeviceMode_IDLE (IOLW) Invoke SM_SetDeviceIdent (VID, DID, FID) and SM_SetDeviceCom (ProcessDataIn, ProcessDataOut, RID) (IOL-Device) Invoke SM_SetDeviceMode_ESTABCOM (IOLW) ConnectionStatus = 0x10: IOL-Device connected
T6	3	4	Invoke SM_Operate (IOL)
T7	4, 8, 10	1	*IOL ComLost* Invoke SM_SetDeviceMode_IDLE (IOLW) Invoke SM_SetDeviceIdent (VID, DID, FID) and SM_SetDeviceCom (ProcessDataIn, ProcessDataOut, RID) (W-Bridge) Invoke SM_SetDeviceMode_ESTABCOM (IOLW) Invoke SM_SetPortConfig_AUTOCOM (IOL) ConnectionStatus = 0x00: No IOL-Device connected
T8	6, 7, 8	5	*IOLW ComLost* Invoke SM_DeviceMode_IDLE (IOL)
T9	5	0	Invoke SM_SetPortConfig_AUTOCOM (IOL) Invoke SM_SetDeviceMode_ESTABCOM (IOLW) Start Timer "IOLDeviceTimeout"
T10	3	6	Invoke SM_GetDeviceIdent and SM_GetDeviceCom (IOLW) and store results to IdentChangeParameters Invoke SM_SetPortConfig_INACTIVE (IOL)
T11	6	7	Invoke SM_SetPortConfig_CFGCOM (IOL)
T12	7	8	Equivalent to T5
T13	8	9	Invoke SM_GetDeviceCom and SM_GetDeviceIdent
T14	9	8	*Incompatible IO-Link Device, W-Bridge remains in Port Diag* -
T15	9	6	*Store new IdentChangeParameters* Invoke SM_SetPortConfig_INACTIVE (IOL)
T16	10	4	-
T17	4	10	*IOLW ComLost* Invoke AL_CONTROL-PDOUTINVALID (IOL)
T18	10	6	Equivalent to T10

INTERNAL ITEMS	TYPE	DEFINITION
$T_{IOLDeviceTimeout}$	Time	T_{DSIO} Maximum, See table 42 in [1].
IdentChangeParameters	RecordT	RID, VID, DID, FID See Table 177

12.6.4.2 Start-up, Synchronization and Reconnection

Figure 114shows a sequence chart for startup of a W-Bridge with connected IO-Link Device V1.1, matching the W-Parameter of the W-Master port (regular startup).

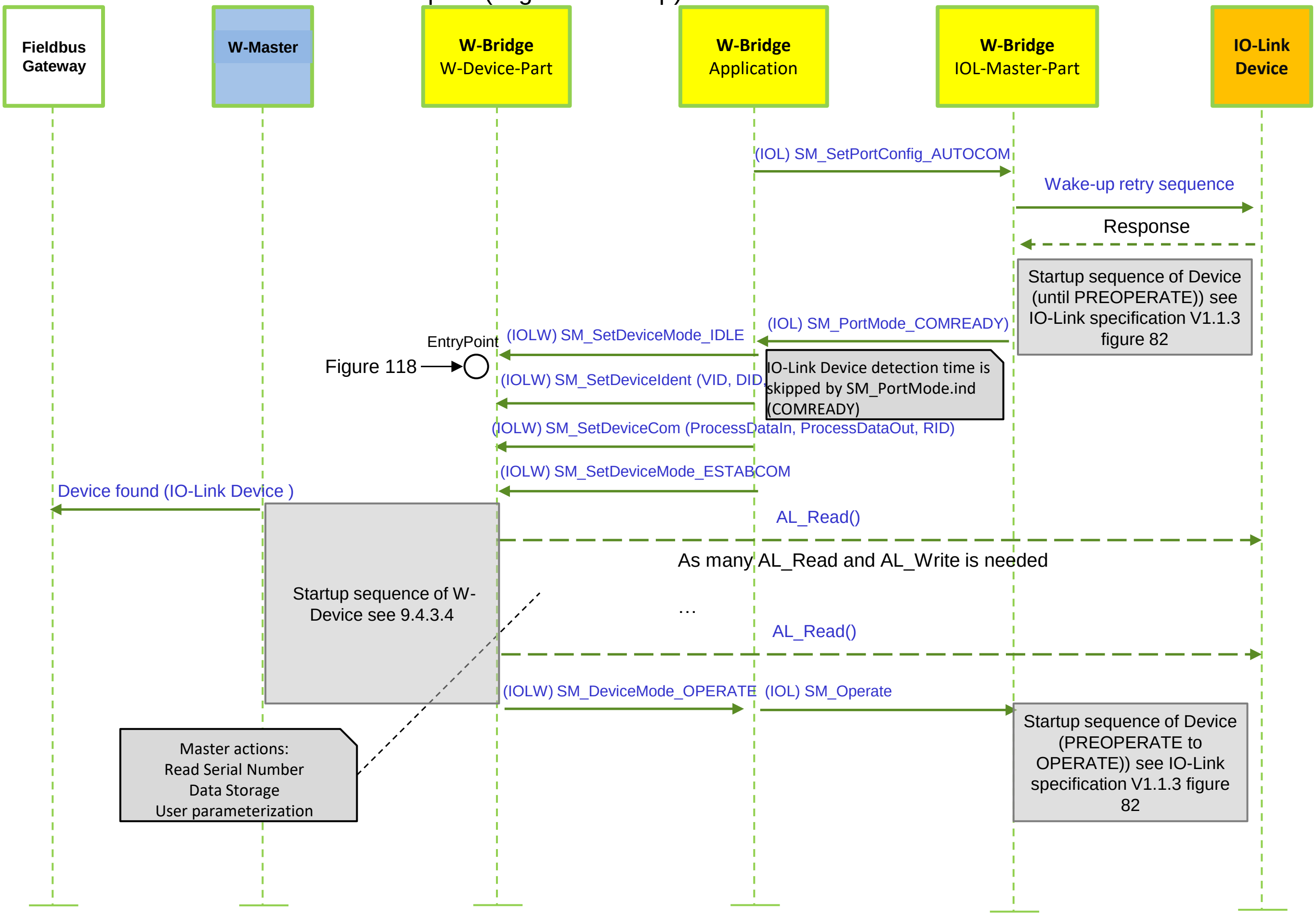


Figure 114 Sequence chart of a W-Bridge startup with connected IO-Link Device V1.1

Figure 115 shows a sequence chart disconnect and reconnect an IO-Link Device while the W-Bridge is in operation.

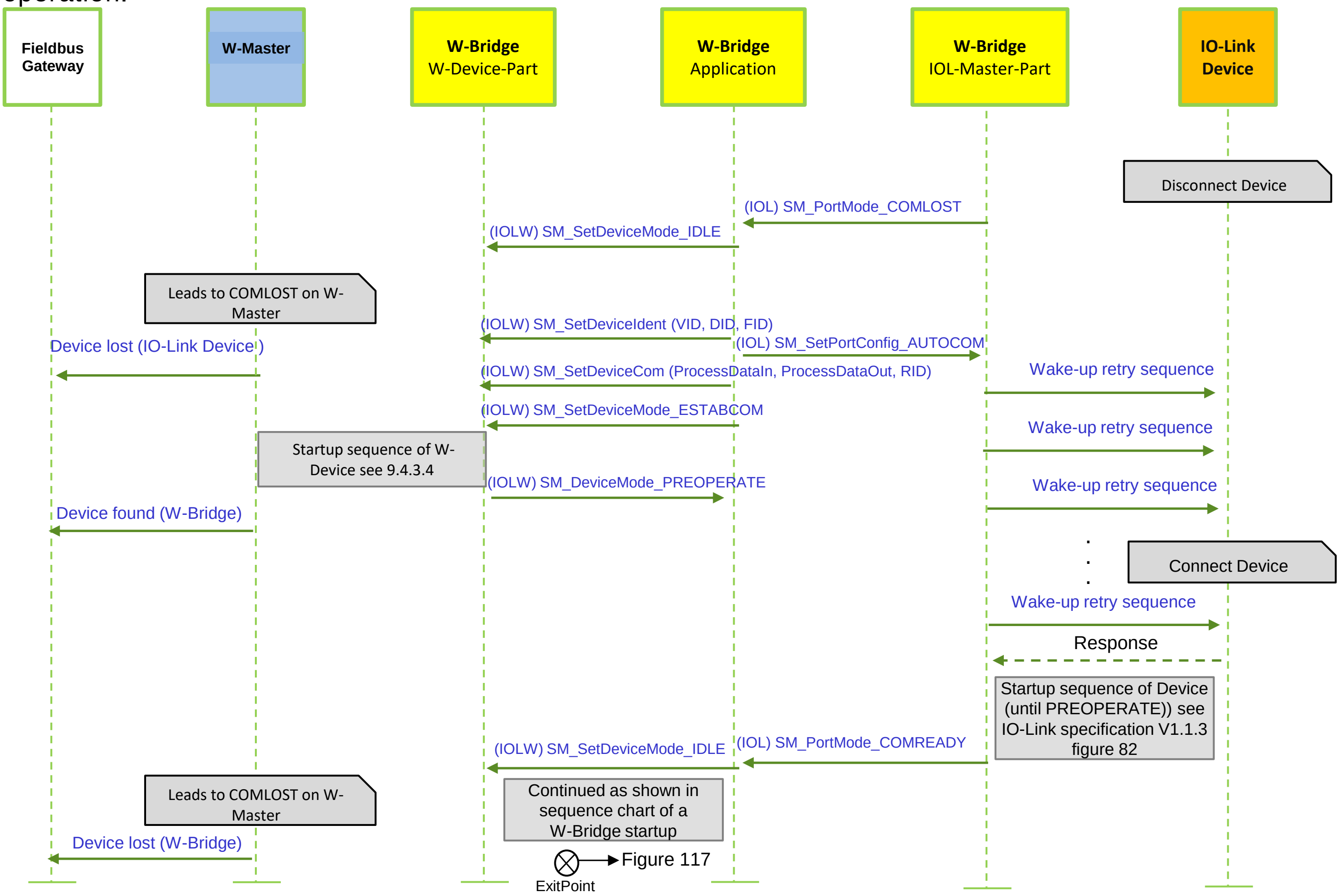


Figure 115 Sequence chart reconnect an IO-Link Device while W-Bridge is in operation

Figure 116 shows a sequence chart of wireless disconnection and reconnection of W-Bridge with connected IO-Link Device

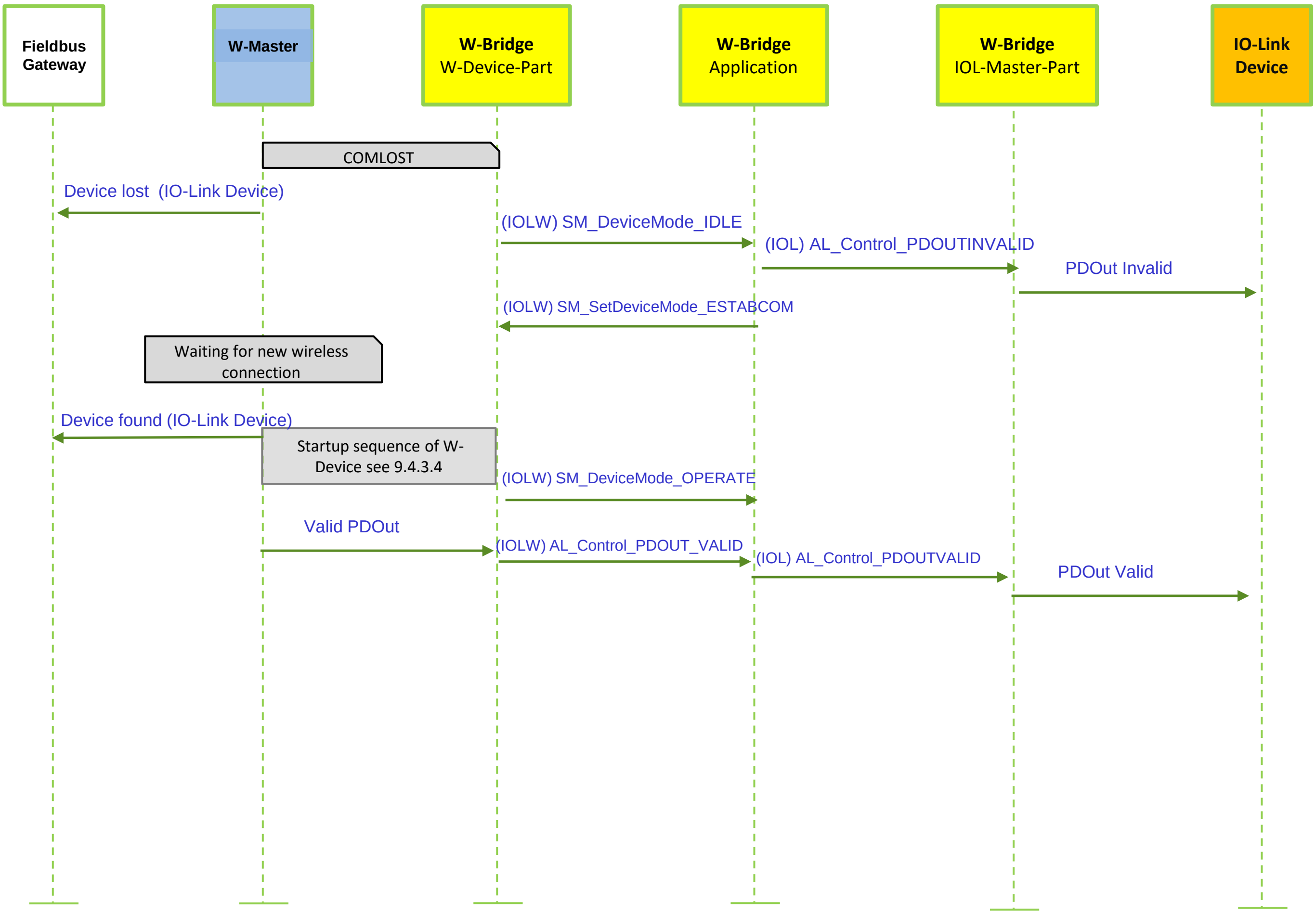


Figure 116 Sequence chart wireless reconnection of W-Bridge with connected IO-Link Device

Figure 117 shows a sequence chart of Device Ident with compatibility with connected IO-Link Device V1.1, but Device Ident shall be modified.

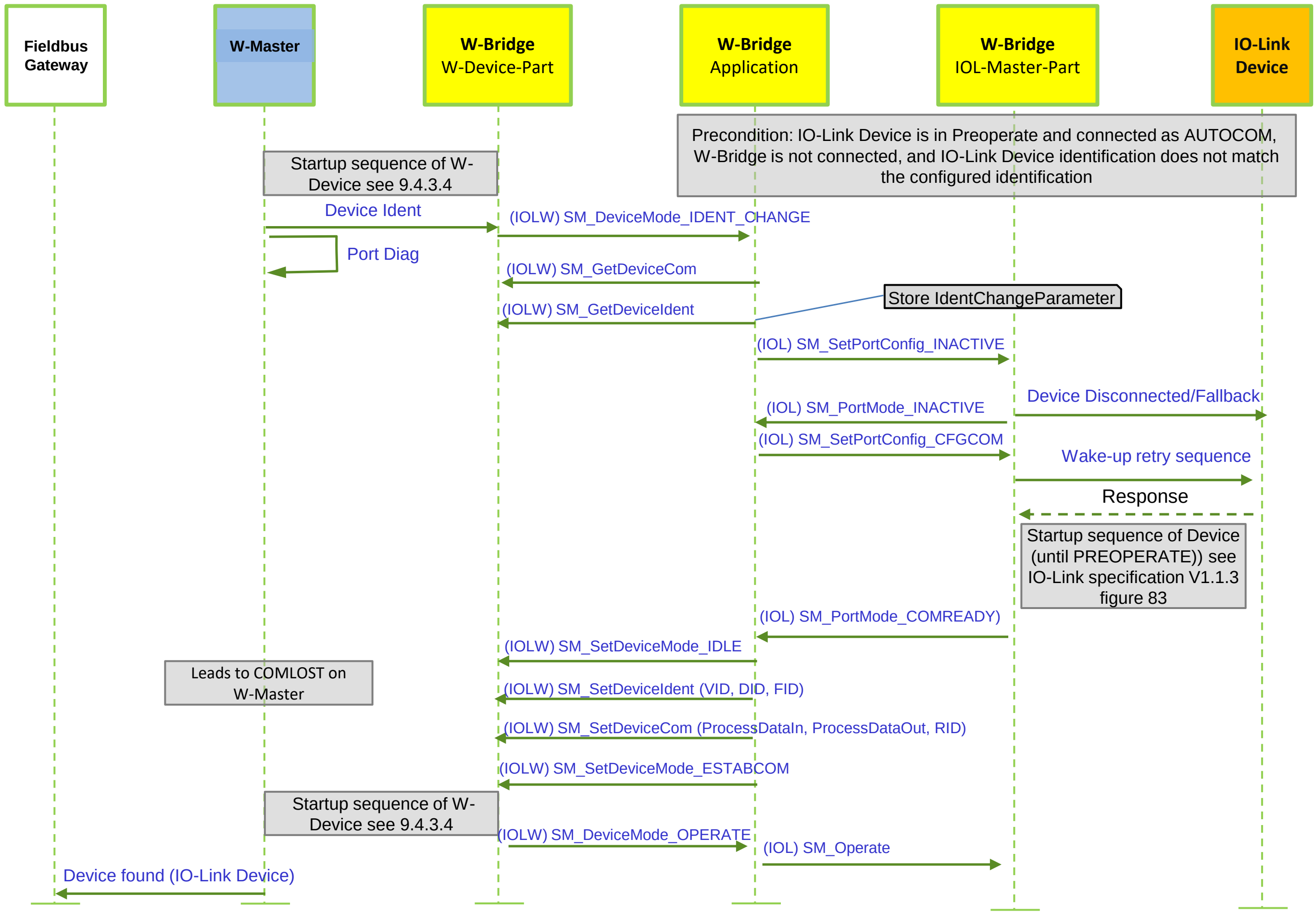


Figure 117 Sequence chart of Device Ident with compatibility

Figure 118 shows a sequence chart of Device Ident with no compatibility with connected IO-Link Device V1.1, when compatibility fails

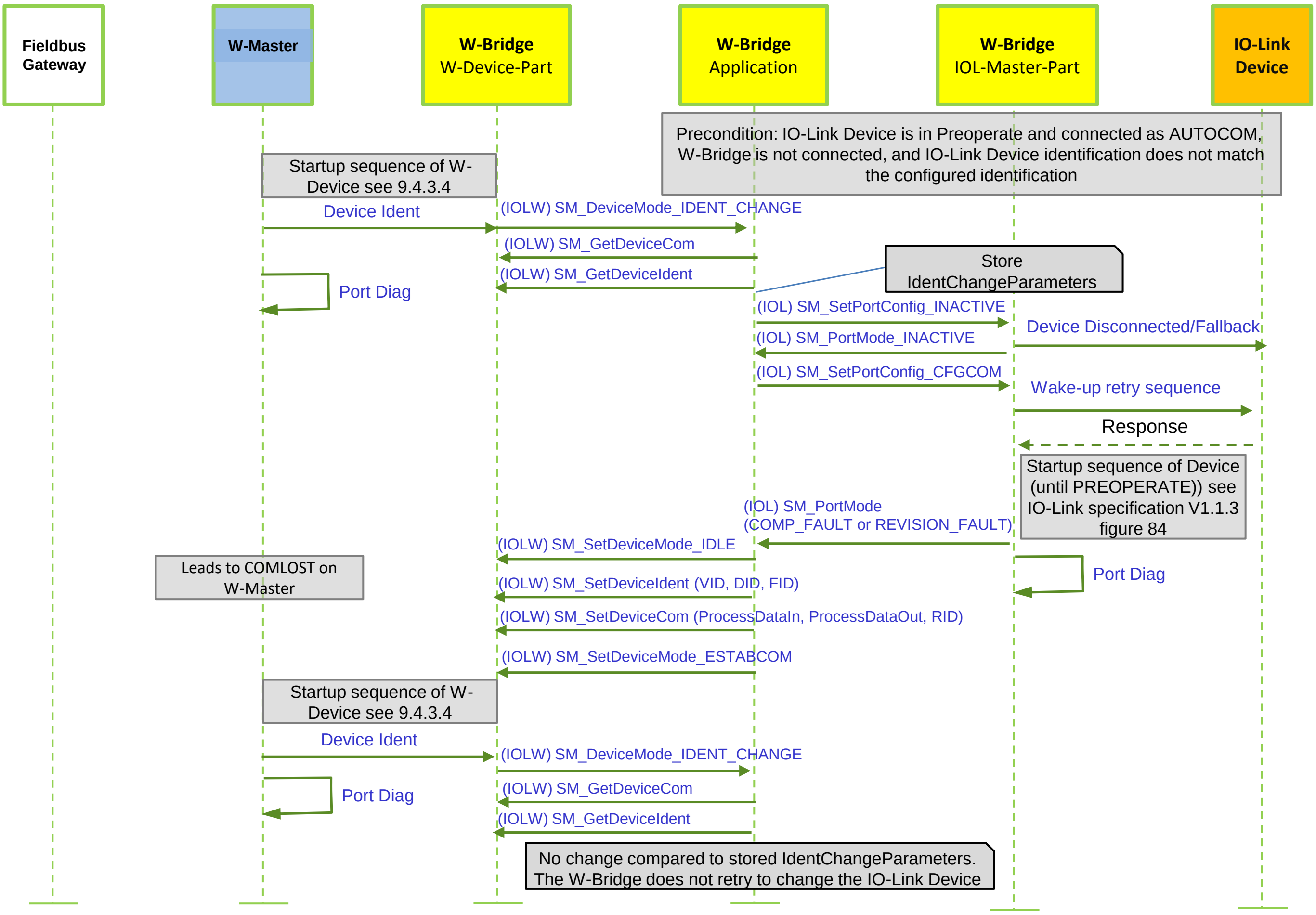


Figure 118 Sequence chart of Device Ident with no compatibility

12.6.5 UniqueID

During Scan and Pairing, a W-Bridge uses the BVendorID, BDeviceID and BDeviceDistinguishingID as UniqueID. The VendorID and DeviceID of a connected IO-Link Device is available after W-Bridge is connected to the W-Master. For details see C.4.4.1 and C.4.9.

12.7 IO Device description (IODD)

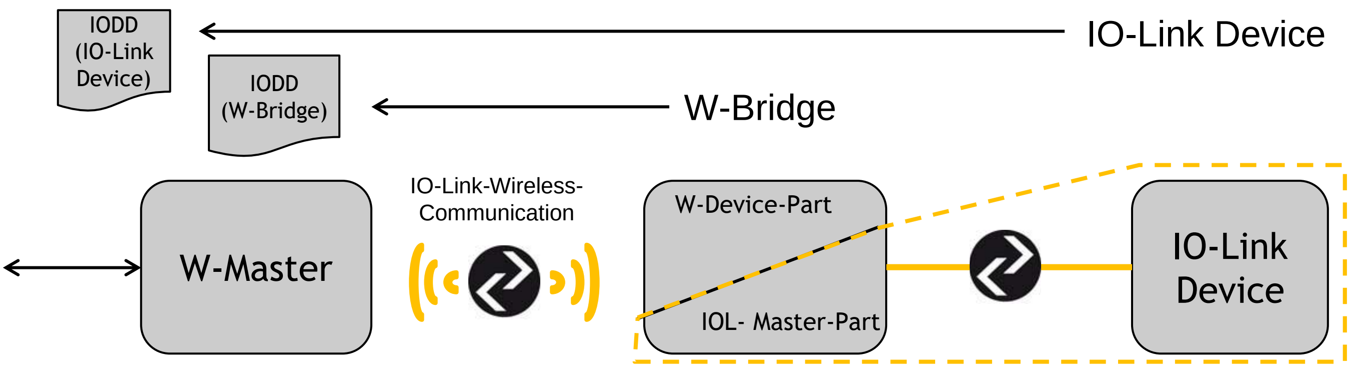


Figure 119 Schematic representation of the use of a W-Bridge to connect an IO-Link Device

The IODD of a W-Bridge describes the W-Bridge without a connected IO-Link Device.

In case of a connected IO-Link Device, the W-Parameters are used from the IODD of the W-Bridge and device specific parameters are taken from the IODD of the IO-Link Device. The parameter selection must be managed by the port device configuration tool.

Annex A

(normative)

W-Messages Codings

A.1 Overview

The W-Master indicates the manner the user data (see A.8) shall be transmitted within a W-Frame.

A.2 Definition of a W-Message

Within the payload of a W-Frame, W-Messages are transmitted in DLink and ULinks (see Figure 120). 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 121 for definition of DLink Control Octet and Figure 123. for definition of ULink Control Octet.

For the generation of the Control Octets, see 6.6.3 and 6.6.5.

For examples of the transmission of W-Messages see A.6 and A.7 .

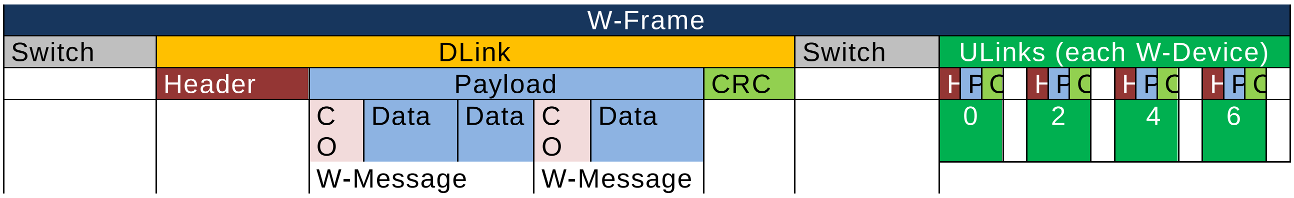


Figure 120 W-Message and Control Octets

A.3 Downlink W-Messages: Control Octets

A.3.1 DLink Control Octet

A.3.1.1 General

The DLink Control Octet is used to send a W-Message to a dedicated W-Device within a DLink.
Figure 121 shows the definition of the Control Octet (2 octet) for a DLink-W-Message

Octet 0								Octet 1							
Slot number (Slot_N) (“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 121 Definition of DLink Control Octet

A.3.1.2 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 46), the DataLength shall be ignored.
DLen is coded in the following way, as shown in Table 145.

Table 145 Definition of DataLength (DLen)

DataLength (DLen)	
DLen	Data length in octet followed by the Control Octet
0	1
1	2
…	…
31	32

A.3.1.3 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 79 Flow Control for segmented data. Examples for the usage of FlowControl see A.6 and A.7.

A.3.1.4 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 146.

Table 146 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 178**.**
6	Reserved	Reserved for future
7	Reserved	Reserved for future

A.3.1.5 Bit 13 to 15: slot number (Slot_N)

These bits contain the “address” (slot number 0 to 7) to which W-Device the W-Message shall be sent.

A.3.2 DLink Control Octets contains MasterCommand.

A.3.2.1 General

Figure 122 shows the DLink-W-Message to transmit a MasterCommand to a W-Device:

Octet 0								Octet 1							
Slot number (Slot_N) (“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 122 DLink Control Octets contains MasterCommand

If the ChC = 5, the second octet (octet 1) shall be used as MasterCommand. For definition of MasterCommand see Table 178

A.3.2.2 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 W-Track. Broadcast supports only limited MasterCommands, see Table 178.

A.3.2.3 Bit 9: Reserved

Reserved for future use.

A.4 Uplink W-Messages

A.4.1 General

The ULink Control Octet is used to send a W-Message from the W-Device to the W-Master within an ULink. Figure 123 shows the definition of the ULink Control Octet (1 octet):

Octet 0							
ChannelCode (ChC)			FlowControl (FC)				
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 123 Definition of ULink Control Octet

A.4.2 ULink Control Octet

A.4.2.1 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 79. Flow Control definition for segmented data. Examples for the usage of FlowControl see A.6 and A.7.

A.4.2.2 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 147.

Table 147 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

A.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 49. For definition of the Control Octet see Figure 121.

The following example shows the placement of different W-Messages in a downlink:

Slot_number 3 (W-Device Address = 3):
8 Octet process data Out
10 Octet acyclic ISDU-write.

Slot_number 5 (W-Device Address = 5):
1 Octet MasterCommand in PreDownLink

Slot_number 1 (W-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 124.

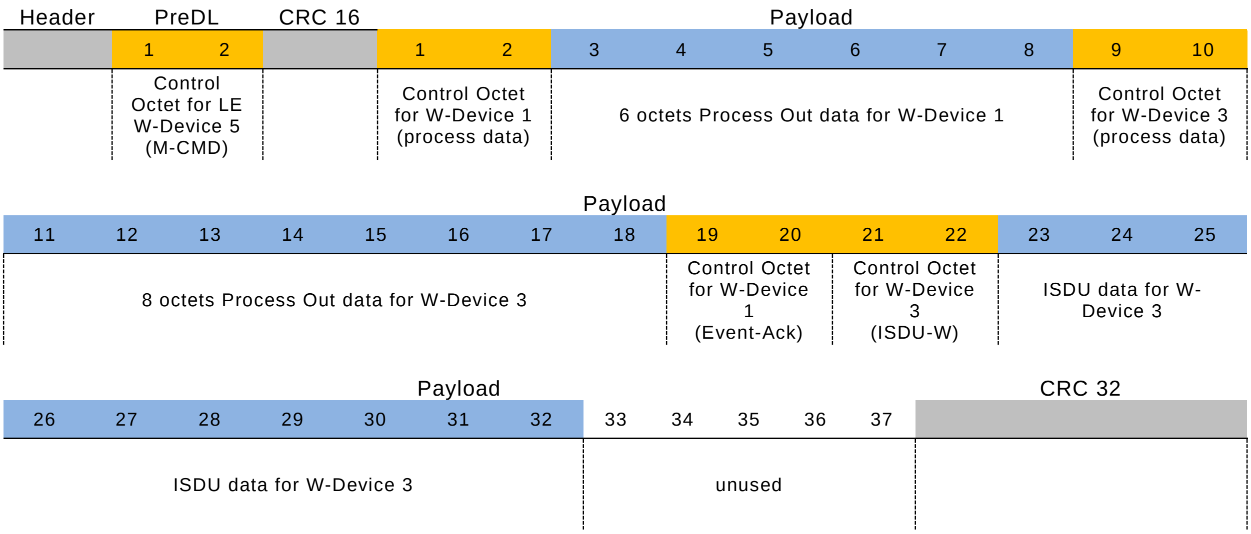


Figure 124 Placement of different W-Messages in a downlink

A.6 Example for DLink data transmission within cyclic process data and segmentation

NOTE:
Maximum downlink payload (37 octet) see Figure 28
For the definition of DLink Control Octet (2 octet) see Figure 121

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: Slot_N = 2; ChC = 1; FC = EOS; DLen = 15
16 Octet Data: 16 octets Process Out data following the Control Octet.
2 Octet Control Octet: Slot_N = 3; ChC = 1; FC = EOS; DLen = 7
8 Octet Data: 8 octets Process Out data following the Control Octet.
2 Octet Control Octet: Slot_N = 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: Slot_N = 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: Slot_N = 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: Slot_N = 2; ChC = 1; FC = EOS; DLen = 15
16 Octet Data: 16 octets Process Out data following the Control Octet.
Control Octet: Slot_N = 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: Slot_N = 2; ChC = 1; FC = EOS; DLen = 15
16 Octet Data: 16 octets Process Out data following the Control Octet.
2 Octet Control Octet: Slot_N = 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: Slot_N = 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: Slot_N = 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: Slot_N = 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.

A.7 Examples for uplink data transmissions

A.7.1 General

Maximum uplink payload of SSlot (2 octet) see Figure 30.
Maximum uplink payload of DSlot (15 octet) see Figure 31.
Size of ULink Control Octet (1 octet) see Figure 123.

A.7.2 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

A.7.3 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.

A.7.4 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.

A.7.5 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.

A.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 148. These data types can be arranged as records (different types) or arrays (same types).

Table 148 Data types for user data

Data type	Reference
BooleanT	See F.2.2 in [1]
UIntegerT	See F.2.3 in [1]
IntegerT	See F.2.4 in [1]
StringT	See F.2.6 in [1]
OctetStringT	See F.2.7 in [1]
Float32T	See F.2.5 in [1]
TimeT	See F.2.8 in [1]
TimeSpanT	See F.2.9 in [1]

A.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 149 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 69 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 W-Devices PDIN data handler generates PDOUT_VALID automatically (see Table 70. 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.

A.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 7.4.4.

A.11 General structure and encoding of Events.

A.11.1 EventQualifier

A.11.1.1 General

The structure of the EventQualifier is shown in Figure 125

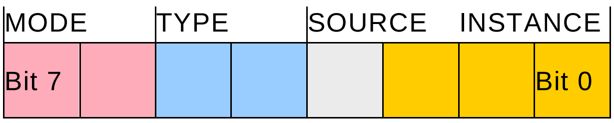


Figure 125 Structure of the EventQualifier

A.11.1.2 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 150

Table 150 Values of INSTANCE

Value	Definition
0	Unknown
1 to 3	Reserved
4	Application
5 to 7	Reserved

A.11.1.3 Bit 3: SOURCE

This bit indicates the source of the Event. Permissible values for SOURCE are listed in Table 151

Table 151 Values of SOURCE

Value	Definition
0	W-Device (remote)
1	W-Master (local)

A.11.1.4 Bits 4 to 5: TYPE

These bits indicate the Event category. Permissible values for TYPE are listed in Table 152.

Table 152 Values of TYPE

Value	Definition
0	Reserved
1	Notification
2	Warning
3	Error

A.11.1.5 Bits 6 to 7: MODE

These bits indicate the Event mode. Permissible values for MODE are listed in Table 153.

Table 153 Values of MODE

Value	Definition
0	reserved
1	Event single shot
2	Event disappears
3	Event appears

A.11.2 EventCode

The EventCode entry contains the identifier of an actual Event. Permissible values for EventCode are listed in Annex D.

Annex B

(normative)

W-Frame Codings, CRC calculation and errors

B.1 Description of ConnectionParameter

The ConnectionParameter in Table 154 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 154.

Table 154 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 Table 5)
SlotType	2 bit (0-1) (see Table 175)

B.2 Downlink packet encodings for Normal Operation

The Figure 126 shows the general structure of the Downlink packet of the W-Frame within a W-Sub-cycle from W-Master to W-Device. The Downlink packet includes the Pre-Downlink packet 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 shall be filled with zeros.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	Track_N (bits 0-2), MasterID (bits 3-7)	Normal DLink ACK	Payload
8	Payload	CRC16[octet 0]	CRC16[octet 1]	Payload
12	Payload	Payload	Payload	Payload
16	Payload	Payload	Payload	Payload
..	Payload	Payload	Payload	Payload
44	Payload	Payload	Payload	Payload
48	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 126 Downlink packet encodings

Table 155 MasterID

Value	Meaning
0, 30, 31	Invalid
1…29	Valid MasterID

Table 156 Track_N

Value	Meaning
0…4	Valid Track_N
5…7	Invalid

Table 157 Normal DLink ACK (8 bit field)

Bit	Meaning
0	Valid ACK bit for W-Device in slot 0
1	Valid ACK bit for W-Device in slot 1
2	Valid ACK bit for W-Device in slot 2
3	Valid ACK bit for W-Device in slot 3
4	Valid ACK bit for W-Device in slot 4
5	Valid ACK bit for W-Device in slot 5
6	Valid ACK bit for W-Device in slot 6
7	Valid ACK bit for W-Device in slot 7

B.3 Downlink packet encodings for Configuration Operation

B.3.1 General

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 158 shall be implemented and used during configuration.

Table 158 Downlink-MSG-Type coding and content (Config Mode only)

Value	Meaning	Payload Content
0x80	MSG_DLink_Pair_Button	MasterID + ULink type + ACK + Track_N, Slot_N + IMATime + Retry Count
0x90	MSG_DLink_Pair_Unique	MasterID + ULink type + ACK + Track_N, Slot_N + IMATime + Retry Count+ UniqueID
0x40	MSG_DLink_Scan_Req	MasterID + ACK + Request_N
0xA0	MSG_DLink_Pair Neg 1	MasterID + ACK + Track_N, Slot_N + Hopping Table (Part 1)
0xB0	MSG_DLink_Pair Neg 2	MasterID + ACK + Track_N, Slot_N + Hopping Table (Part 2) + Col-N

Table 159 Uplink Type

Value	Meaning
00	Single Slot Uplink
01	Double Slot Uplink

Table 160 Config Downlink ACK (1 bit)

Value	Meaning
0	No packet received
1	Last packet received

Table 161 Scan DLink ACK (4 bit field, only used for Scan Request Downlink)

Bit	Meaning
0	Valid ACK bit for W-Device in DSlot 0
1	Valid ACK bit for W-Device in DSlot 2
2	Valid ACK bit for W-Device in DSlot 4
3	Valid ACK bit for W-Device in DSlot 6

Table 162 Roaming Flag

Value	Meaning
00	Roaming not requested
01	Roaming requested

Table 163 Track_N

Value	Meaning
0-4	Valid Track_N
5-7	Invalid

Table 164 Slot_N

Value	Meaning
0-7	Valid Slot_N

B.3.2 Scan Request Downlink packet

In Scan Mode and Roaming Mode, the W-Master is able to discover unpaired W-Devices. This is achieved by transmitting Scan Request packet. A Scan Request packet is shown in Figure 127.
s

After receiving a Scan Request packet, W-Devices shall respond with the Scan Response Uplink packet after a random number of configuration W-Sub-cycles, as described in clause 5.3.6

The W-Master should transmit its MasterID, an Acknowledge for last received Uplink packet, the Scan Request identifier and the consecutive number of Scan Request as Request_N in each configuration Downlink packet during ServiceMode. For the Acknowledgement of last received Uplink packet the W-Master transmits also the received UniqueIDs in the next configuration Downlink packet.

Octet	0	1	2	3
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[octet 0]	ConfigSyncword[octet 1]
4	ConfigSyncword[octet 2]	Unused / MasterID	Scan DLink ACK / MSG_DLINK_type	[MSO] Request_N
8	RequestN [LSO]	[MSO]		
12	UniqueID of last Scan Response [DSlot0] (if ACK Bit is set, otherwise unused)			
16		[LSO]	[MSO]	
20	UniqueID of last Scan Response [DSlot2] (if ACK Bit is set, otherwise unused)			
24			[LSO]	[MSO]
28	UniqueID of last Scan Response [DSlot4] (if ACK Bit is set, otherwise unused)			
32				[LSO]
36	[MSO]			
40	UniqueID of last Scan Response [DSlot6] (if ACK Bit is set, otherwise unused)			
44	[LSO]	unused		
48	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 127 Scan Request packet

B.3.3 Pairing Request Downlink packet

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 Request packets. A Pairing Request packet is shown in Figure 128. The addressed W-Device shall answer with a Pairing Response Uplink packet within the same W-Sub-cycle.

Each Pairing Request packet shall contain the ID of the W-Master, requested Uplink type (SSlot Uplink or DSlot Uplink), the Acknowledge for the last received Uplink packet, 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 packet, 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 Pairing 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.

Octet	0	1	2	3
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[octet 0]	ConfigSyncword[octet 1]
4	ConfigSyncword[octet 2]	ACK, UL_Type, MasterID	Reserved, MSG_DLINK_type	Slot_N, Track_N, Roaming
8	[MSO] UniqueID			
12	UniqueID			
16	UniqueID [LSO]	DataSyncword[octet 0]	DataSyncword[octet 1]	DataSyncword[octet 2]
..	unused			
44	unused			
48	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 128 Pairing Request packet

Table 165 Pairing Request: Unique ID

Value	UniqueID
0x0000000000000000	Pair by Button
0x0000000000000001- 0xFFFFFFFFFFFFFFFF	Pair by Unique ID

B.3.4 Pairing Negotiation Downlink packet

In ServiceMode within the Pairing Procedure, the Negotiation Downlink packets are used by W-Master for configuration of the W-Device. There are two mandatory consecutive Negotiation Downlink packets 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 Downlink packets are containing the MasterID, Uplink Slot Type of the W-Device being configured, the Acknowledge of the last received Uplink packet, Downlink-MSG-type (MSG_DLink_Pair Neg 1 or MSG_DLink_Pair Neg 2), the Slot_N and Track_N, the actual frequency hopping table length, Next Col_N of Cyclic Mode and the frequency hopping table of Cyclic Mode.

Octet	0	1	2	3
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[octet 0]	ConfigSyncword[octet 1]
4	ConfigSyncword[octet 2]	ACK | UL_Type | MasterID	Reserved | MSG_DLINK_type	Slot_N | Track_N | Reserved
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	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 129 Pairing Negotiation packet type 1 => DLink-Message-Type = MSG_DLink_Pair_Neg_1

Table 166 Values for Frequency Table length

Value	Meaning
0-14, 79-255	Invalid
15-78	Valid table length

Table 167 Permitted Values for HOP_N

Value	Meaning
3 – 78	Valid frequency for cyclic data channel
0	End of Frequency Table Delimiter

Table 168 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	0	1	2	3
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[octet 0]	ConfigSyncword[octet 1]
4	ConfigSyncword[octet 2]	ACK, UL_Type, MasterID	Reserved, MSG_DLINK_type	Slot_N, Track_N, Reserved
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	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 130 Pairing Negotiation packet type 2 => DLink-Message-Type = MSG_DLink_Pair_Neg_2

B.4 Uplink packet encodings for Normal Operations

B.4.1 Regular Single Slot Uplink packet

In Cyclic Mode, the Regular Uplink packet shown in Figure 131 is used to transmit process and event data from W-Device to the W-Master. A SSlot Uplink packet can handle 16 bit data payload, which can contain cyclic process data, diagnosis data or event notifications.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK, IMA, ABA, MasterID	Payload	Payload
8	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 131 Regular SSlot Uplink Packet

B.4.2 Regular Double Slot Uplink packet

In Cyclic Mode, the Regular DSlot Uplink packet is used to transmit process and event data from W-Device to W-Master.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK, IMA, ABA, MasterID	Payload	
8	Payload			
12	Payload			
16	Payload			
20	Payload	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]
24	CRC32[octet 3]			

Figure 132 Regular DSlot Uplink packet

Table 169 Uplink IMA

Value	Meaning
0	Normal Uplink
1	IMA Uplink

Table 170 Uplink Alert Back Alive (ABA)

Value	Meaning
0	Normal Uplink
1	Device is back after a loss of communication

Table 171 Uplink ACK

Value	Meaning
0	No packet received
1	Last packet received

B.4.3 IMA Uplink packet

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 DSlot Uplink packet see Figure 133 or an IMA SSlot Uplink packet see Figure 134. IMA Uplink packets should contain an IMA=1 Flag, the Acknowledge for Previously received packet and diagnosis data.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK IMA ABA MasterID	RSSI_D	LQI_D
8	unused			
12	unused			
16	unused			
20	unused	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]
24	CRC32[octet 3]			

Figure 133 DSlot IMA-Uplink packet

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK IMA ABA MasterID	RSSI_D	LQI_D
8	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]	CRC32[octet 3]

Figure 134 SSlot IMA-Uplink packet

Table 172 Diagnosis encoding octet 7 (RSSI)

Bit	7	6	5	4	3	2	1	0
Meaning	RSSI_D No 7	RSSI_D No 6	RSSI_D No 5	RSSI_D No 4	RSSI_D No 3	RSSI_D No 2	RSSI_D No 1	RSSI_D No 0

Table 173 Diagnosis encoding octet 8 (LQI_D)

Bit	7	6	5	4	3	2	1	0
Meaning	LQI_D No 7	LQI_D No 6	LQI_D No 5	LQI_D No 4	LQI_D No 3	LQI_D No 2	LQI_D No 1	LQI_D No 0

B.5 Uplink packet encodings for Configuration Operations

B.5.1 General

In the ServiceMode the System Management, DL-A/B Message handlers are not involved in the Uplink packet 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 (see Table 174), shall be implemented und used during configuration.
Table 174 shows the Uplink-MSG-Type.

Table 174 Uplink-MSG-Type (Config Mode only)

Value	Meaning	Payload Content
0x4	MSG_UPLINK_Scan_Resp	RevisionID + IMATime + UniqueID
0x8	MSG_UPLINK_Pair_Resp Button	RevisionID + IMATime + UniqueID
0x9	MSG_UPLINK_Pair_Resp Unique	RevisionID + IMATime + UniqueID
0xA	MSG_UPLINK_Pair_Neg_1_Resp	Response Only, (no data transfer)
0xB	MSG_UPLINK_Pair_Neg_2_Resp	Response Only, (no data transfer)

Table 175 shows the slot type in config uplink packet.

Table 175 Slot-Type in config Uplink

Value	SlotType
00	Single Slot (SSlot)
01	Double Slot (DSlot)

Table 176 shows the RevisionID encoding.

Table 176 RevisionID

Bits	Value	Meaning
0 to 3	0x0…0xF	MinorRev part of the protocol revision (see B.1.5 in [1])
4 to 7	0x0…0xF	MajorRev part of the protocol revision (see B.1.5 in [1])

B.5.2 Scan Response Uplink packet

In ServiceMode the W-Device answers to a received Scan Request packet with a Scan Response packet. A Scan Response packet as shown in Figure 135 shall contain the MasterID received in the Scan Request packet, the Uplink type, an Uplink Message type, the RevisionID, and its UniqueID.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK / UL type / MasterID	Reserved / MSG_UPLINK_type	RevisionID
8	[MSO] UniqueID	UniqueID	UniqueID	UniqueID
12	UniqueID	UniqueID	UniqueID	UniqueID
16	UniqueID [LSO]	unused	unused	unused
20	unused	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]
24	CRC32[octet 3]			

Figure 135 Scan Response packet

B.5.3 Pairing Response Uplink packet

In ServiceMode, the W-Device shall answer to a Pairing Request Downlink packet with a Pairing Response Uplink packet 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 packet, the RevisionID and the UniqueID of the W-Device as shown in Figure 136.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK (bit 0), UL type (bits 1–2), MasterID (bits 3–7)	Reserved (bits 0–3), MSG_UPLINK_type (bits 4–7)	RevisionID
8	[MSO] UniqueID			
12	UniqueID			
16	UniqueID [LSO]	unused		
20	unused	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]
24	CRC32[octet 3]			

Figure 136 Pairing Response packet

B.5.4 Negotiation Response Uplink packet

In ServiceMode, the W-Device shall respond on each Negotiation Downlink packet it receives. W-Device shall submit the Uplink Type of the W-Device, the acknowledge for the last received Downlink packet, MSG_UPLINK and the W-Device RevisionID as shown in Figure 137.

Octet	0	1	2	3
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[octet 0]	DataSyncword[octet 1]
4	DataSyncword[octet 2]	ACK (bit 0), UL type (bits 1–2), MasterID (bits 3–7)	Reserved (bits 0–3), MSG_UPLINK_type (bits 4–7)	RevisionID
8	[MSO] UniqueID			
12	UniqueID			
16	UniqueID [LSO]	unused		
20	unused	CRC32[octet 0]	CRC32[octet 1]	CRC32[octet 2]
24	CRC32[octet 3]			

Figure 137 Pairing Negotiation Uplink packet

B.6 Acknowledge Generation

The PL in W-Master shall generate an ACK-Bit (see Figure 126) for each W-Device, if the W-Master received a valid Uplink packet.
The PL in W-Device shall generate an ACK-Bit (see Figure 131 and Figure 132) 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 packet is received or the Uplink packet is lost or invalid (e.g., wrong CRC) a NACK (ACK-Bit = 0) is generated.

B.7 CRC16 and CRC32 calculation

The integrity of Uplink and Downlink transmissions shall be protected through CRC-32-AUTODIN-II (CRC32).
The integrity of Pre-Downlink packet part octets shall be protected through CRC16-CCITT (CRC16).

The CRC algorithms are defined as follows:
CRC32:
The CRC32 generator polynomial shall be 0x4C11DB7 ($x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$)
The CRC32 calculation shall be performed across all octets except preamble and syncword.

Each data octet shall enter the CRC32 calculation with least significant bit first (the same bit order as over the air transmission).
The CRC32 shall be transmitted with most significant bit first, i.e., from register bit 31 to register bit 0.

Initial Value (Pre-set)

- 0xFFFFFFFF during communication on configuration channels and for cyclic downlink packets.
- 0xFFFFFFFF xor W-Device distinguish identifier (octet order MSO…LSO) for cyclic uplink packets.

The final xor (residue) during transmission: 0xFFFFFFFF

CRC16:
The CRC16 generator polynomial shall be 0x1021 ($x^{16} + x^{12} + x^5 + 1$)

The CRC16 calculation shall be performed across the two Pre-Downlink packet part octets only and placed at its end.
Each data octet shall enter the CRC16 calculation with least significant bit first (the same bit order as over the air transmission).

The CRC 16 shall be transmitted with most significant bit first, i.e., from register bit 15 to register bit 0.

Initial Value (Pre-set) 0xFFFF.
The final xor during transmission and reception 0x0000

B.8 Errors

B.8.1 General

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.

B.8.2 Checksum errors

Any checksum error in a receiver suppress it is acknowledge to the transmitter.

B.8.3 **IMA Timeout errors**

IMA Timeout errors occurs if the configured IMA time at the W-Master is exceeds.

B.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)

W-Device Parameter and commands

C.1 Overview

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 IO-Link Wireless. 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 138.

The W-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 [1]. For W-Devices, the use of profile(s) is recommended e.g., Smart Sensor Profile and Common Profile, see [4].

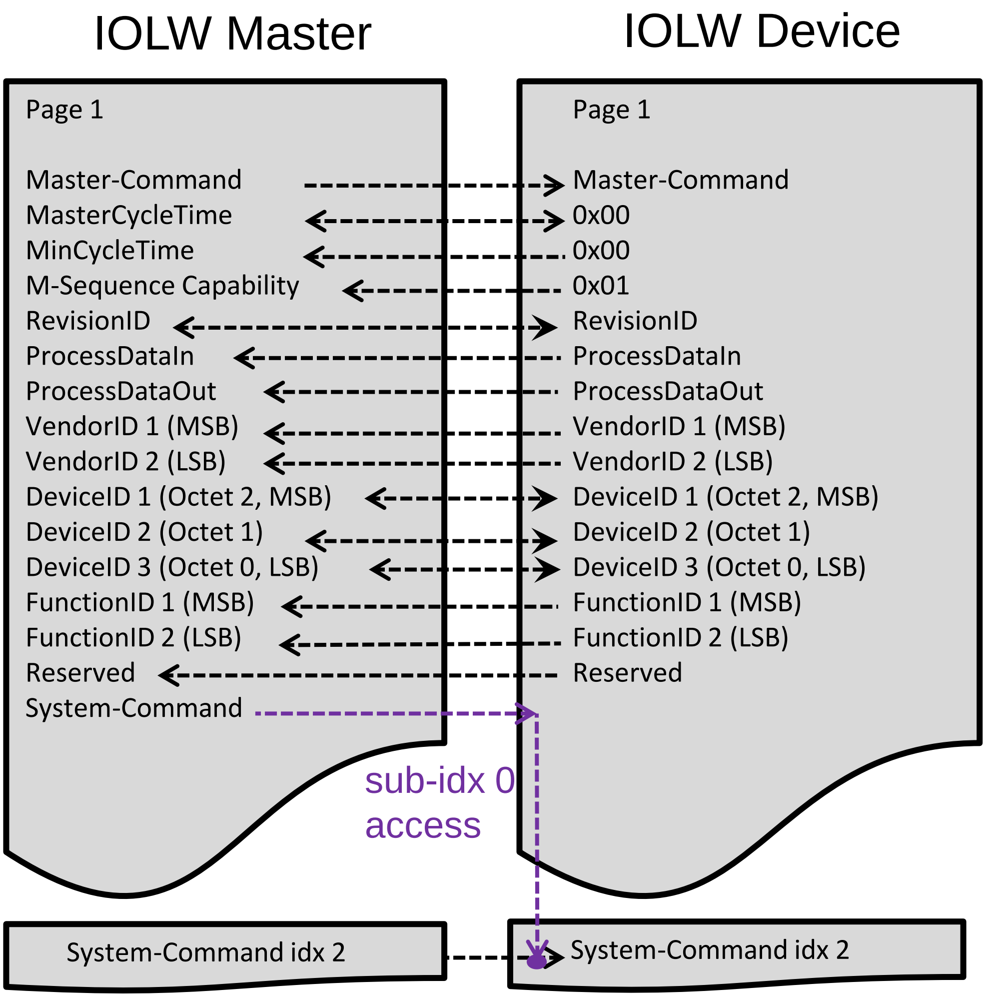


Figure 138 Memory mapping of the direct parameter page 1 of a W-Master with a W-Device.

C.2 Direct Parameter Page 1

C.2.1 General

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 an IO-Link Device in most cases a straight forward mapping of the parameters, see Figure 111.

Despite having the same direct parameter structure, IO-Link and IO-Link 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 0x01.
- 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 177 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 C.2.2
	0x02	R/W	MasterCycleTime	Not used: octet has no effect, read returns 0x00	-
	0x03	R	MinCycleTime	Not used: octet is set to 0x00	-
	0x04	R	M-Sequence Capability	Not used: octet is set to 0x01	-
	0x05	R/W	Revision ID	ID of the used RevisionID for implementation (shall be set to 0x11)	Mandatory/ see C.2.3
	0x06	R	ProcessDataIn	Number and structure of input data (Process Data from W-Device to W-Master)	Mandatory/ see C.2.4
	0x07	R	ProcessDataOut	Number and structure of output data (Process Data from W-Master to W-Device)	Mandatory/ see C.2.5
	0x08	R	VendorID (MSB)	Unique vendor identification	Mandatory/ see C.2.6
	0x09	R	VendorID (LSB)		
	0x0A	R/W	DeviceID 1 (Octet 2, MSB)	Unique Device identification allocated by a vendor	Mandatory/ see C.2.7
	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 C.2.8
	0x0E	R	FunctionID 2 (LSB)		
	0x0F	R	reserved	-	-
	0x10	-	System-Command	Not used (see NOTE 1)	-
For all IO-link Wireless devices SystemCommand on page 1 shall not be used, but index 2 instead. NOTE: A Read operation returns unspecified values					

C.2.2 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 178.

Table 178 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 packet part octets only
0x5E	FullDLink	Switches the W-Device radio to receive full Downlink packets
0x5F	UnPairing	Unpairs the W-Device. The W-Device deletes all its W-Parameters.
0x60 to 0x95	Reserved	
0x96	DeviceIdent	Start check of Direct Parameter page for changed entries
0x97	DeviceStartup	Switches the W-Device from OPERATE or PREOPERATE to STARTUP
0x98	Reserved	
0x99	DeviceOperate	Process output data invalid or not available. Switches the W-Device from STARTUP or PREOPERATE to OPERATE
0x9A	DevicePreoperate	Switches the W-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.
0xFF	WakeUp	Set low energy W-Device WakeUpTimer to 0.

C.2.3 Revision ID

Identical to IO-Link Interface and System Specification: Section B.1.5 in [1]

The RevisionID numbers of the IO-Link and IO-Link Wireless are independent. This revision of the standard specifies RevisionID 1.1 (i.e., RevisionID=0x11).

C.2.4 ProcessDataIn

Identical to IO-Link Interface and System Specification: Section B.1.6 in [1]

Since IO-Link Wireless Devices do not provide a switching signal. SIO mode is not supported. Therefore, bit 6 in ProcessDataIn structure shall be set to 0.

C.2.5 ProcessDataOut

Identical to IO-Link Interface and System Specification: Section B.1.7 in [1]

C.2.6 VendorID

Identical to IO-Link Interface and System Specification: Section B.1.8 in [1]

C.2.7 DeviceID

Identical to IO-Link Interface and System Specification: Section B.1.9 in [1]

C.2.8 FunctionID

Identical to IO-Link Interface and System Specification: Section B.1.10 in [1]

C.2.9 SystemCommand

See IO-Link specification B.2.2 in [1].

C.3 Direct Parameter Page 2

ISDU support is mandatory for W-Devices therefore, the direct parameter page 2 shall not be used by W-Devices.

C.4 IO-Link Wireless specific parameters

C.4.1 Overview

IO-Link Wireless makes use of the same predefined device parameter as IO-Link Devices. Nevertheless, in order to store the W-Parameters new indices have been predefined.

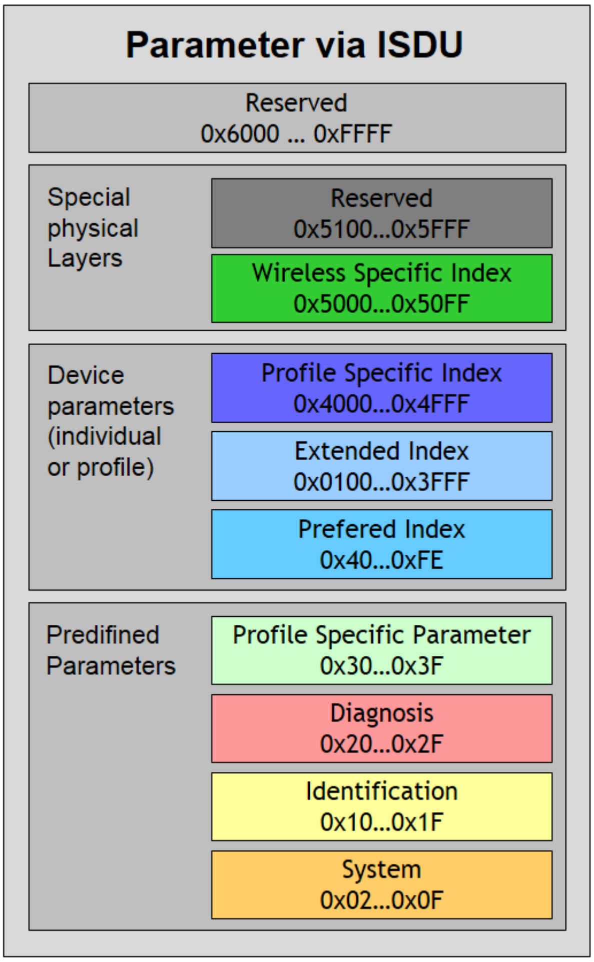


Figure 139 Index space for ISDU data objects

Table 179 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 177
0x0001 (1)	Page 2	R/W		RecordT	O	ISDU support is mandatory for W-Devices therefore, the direct parameter page 2 shall not be used by W-Devices.
0x0002 (2)	SystemCommand	W	1 octet	UIntegerT	M	See [1]
0x0003… 0x0014 (3 to 20)	Similar to IOL	-	-	-	-	See [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 [1]
0x5000 (20480)	WDeviceMode	R	1 octet	UIntegerT	M	See Clause: C.4.3
0x5001 (20481)	WirelessSystemMgmt	R	9 octets	RecordT	M	See Clause: C.4.4
0x5002 (20482)	WirelessSystemCfg	R/W	4 octets	RecordT	M	See Clause: C.4.4
0x5003 (20483)	WirelessQuality	R	2 octet	RecordT	C	See Clause: C.4.8
0x5004 (20484)	WBridgeInfo	R	12 octets	RecordT	C	See Clause: C.4.9
0x5005 (20485)	WRadioInfo	R	12 octets	RecordT	M	See Clause: C.4.5
0x5006 (20486)	AdaptiveHopTable	W	82 octets	RecordT	M	See Clause: C.4.10
0x5007 (20487)	WCycleTime	R/W	4 octet	RecordT	M	See Clause: C.4.12
0x5008-0x503F (20488 to 20543)	Reserved					
0x5040-0x50FF (20544 to 20735)	Wireless Vendor Specific Index	-	-	-	-	Specific to Radio Vendor
0x5100… 0xFFFF (20736 to 65535)	Similar to IO-Link Interface and System Specification	-	-	-	-	See [1]
Key M=Mandatory; O=optional; C=conditional						

C.4.2 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.

Implementation of the SystemCommand feature is mandatory.
The coding of SystemCommand is specified in B.2.2 in [1].

C.4.3 WDeviceMode

This index range stores the mode of a W-Device, see Table 180

Table 180 WDeviceMode

Value	Mode
0	INACTIVE
1	STARTUP
2	IDENT_STARTUP
3	IDENT_CHANGE
4	PREOPERATE
5	OPERATE

C.4.4 Wireless System

This index range stores all the WirelessSystemMgmt and WirelessSystemCfg parameters of a W-Device.

Table 181 Wireless system index assignments

Index	Subindex	Access	Parameter name	Coding	Data type
0x5001	0x00	Gives access to the whole index			
	0x01	R	UniqueID	See Clause: C.4.4.1	OctetStringT9
0x5002	0x00	Gives access to the whole index			
	0x01	R/W	IMATime(Time Base)	See Clause: C.4.4.2	UIntegerT8
	0x02	R/W	IMATime (Multiplier)	See Clause: C.4.4.2	UIntegerT8
	0x03	R/W	MaxRetry	See Clause: C.4.4.3	UIntegerT8
	0x04	R/W	TxPower	See Clause: C.4.4.4	UIntegerT8

C.4.4.1 UniqueID

This mandatory parameter consists of the 2 octet manufacturer distinguishing VendorID (MSO) followed by the 3 octet DeviceID and a 4 octet device distinguishing identifier (LSO). The Device Distinguishing ID shall be a unique value for every sample of all W-Devices produced by that vendor. It is in the responsibility of the vendor to maintain that number space or its computation algorithm.

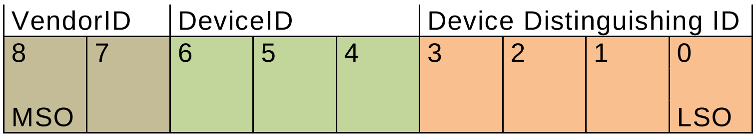


Figure 140 UniqueID octet mapping

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.

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.

C.4.4.2 IMATime

The IMA ("I am alive") time is a mandatory W-Parameter. IMATime is system and W-Device specific. W-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 shall 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 182:

Table 182 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 183:

Table 183 IMA time encoding

Octet 0 (Subindex 0x01)	**Octet 1** (Subindex 0x02)
Time Base encoding see Table 184	Multiplier Permitted values: 1 to 255

A value of 0x01 means that the W-Device stays always on.

Table 184 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 182
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”.

C.4.4.3 MaxRetry

This index stores the maximal number of retries. The minimum and default value is 0x02, thus one primary transmission and 2 retries.

Table 185 Value for the maximal number of retries

Value	**Remark**
0x00, 0x01	Reserved
0x02	2 retries
0x03	3 retries
0x04	4 retries
…	…
0x1F	31 retries
0x20 …0xFF	Reserved

C.4.4.4 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 186 TxPower parameter

TxPower	Predefined Power Level	Values [dBm]
0x00	-	Reserved
0x01	Level 1	-20
0x02	Level 2	-19
…	…	…
0x15	Level 21	0
…	…	…
0x0F	Level 30	9
0x1F	Level 31	10
0x20 – 0xFF	-	Reserved

C.4.5 WRadioInfo

This index range stores all the information related to the radio hardware and software installed on the W-Device.

Table 187 Radio manufacturer identification 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	OctetStringT2
	0x03	R	RadioHWRevision	vendor specific	OctetStringT4
	0x04	R	RadioSWRevision	vendor specific	OctetStringT4

NOTE: RadioVendorIDs are assigned by IO-Link community.

C.4.6 ISDU Mapping for system management

The Table 188 shows which ISDU indexes shall be forwarded by the ISDU handler to the system management. To get WirelessQuality values the W-Device SM forwards the read request to the W-Device service PL_QualityService. These indexes are system-internals and are not used in DataStorage.

Table 188 ISDU indexes for system management

Index	Object name
0x0000	Page 1
0x5001	WirelessSystemMgmt
0x5002	WirelessSystemCfg
0x5003	WirelessQuality
0x5004	WBridgeInfo
0x5005	WRadioInfo
0x5006	AdaptiveHopTable
0x5007	WCycleTime

These indexes are used for private exchanges between the W-Master and the W-Device; the W-Master shall block any write access request from a gateway application to this Index.

C.4.7 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.
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.

C.4.8 Wireless Quality

C.4.8.1 General

This index range stores all the information related to the quality of the wireless connection of the W-Device. This parameter is not used by low energy W-Devices, see clause C.4.11

Table 189 WirelessQuality index assignments

Index	Subindex	Access	Parameter name	Coding	Data type
0x5003	0x00	Gives access to the whole index			
	0x01	R	LQI_D	See Clause: C.4.8.2	UIntegerT8
	0x02	R	RSSI_D	See Clause: C.4.8.3	IntegerT8

C.4.8.2 Link Quality indication (LQI)

This index parameter stores statistical data about the reliability of the radio transmission for this W-Device. The method used for the calculation of the LinkQuality LQI is described in clause 5.5.6.1.

Table 190 LQI parameter

LQI	Values
0x00	0 %
0x01	1 %
0x02	2 %
…	…
0x64	100 %
0x65 – 0xFE	Reserved
0xFF	INVALID

C.4.8.3 Received Signal Strength Indication (RSSI)

This parameter stores the RSSI.

Table 191 RSSI parameter

RSSI	Description
-128 to 20	1 Octet (IntegerT) Range: -128 ≤ N ≤ +20 Units: dBm
+127	INVALID This value (0x81) shall be transmitted, if the RSSI is not available or not supported by a W-Master or W-Device
21 to 126	Reserved

C.4.9 W-Bridge Information

This index range WBridgeInfo stores the parameters used in a W-Bridge configuration. This index is mandatory for W-Bridge. BDeviceID, BVendorID and BFunctionID are similar to DeviceID, VendorID and FunctionID and refer to the W-Bridge, not the connected IO-Link device.

Table 192 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	UIntegerT8

C.4.10 AdaptiveHopTable

This index range AdaptiveHopTable stores the values for the updated hopping sequence, see clause H.4

Table 193 update hopping table index assignments

Index	Subindex	Access	Parameter name	Data type
0x5006	0x00	Gives access to the whole index		
	0x01	W	WakeUpTime	3 Octet
	0x02	W	UpdateType	Octet
	0x03	W	Index	Octets
	0x04	W	Frequency value	OctetString

C.4.11 Blocked ISDU indexes for low energy W-Device

The Table 194 shows which ISDU indexes shall be blocked by the W-Master for read or write when a low energy W-Devices is connected to this W-Port.

Table 194 Blocked ISDU indexes for low energy W-Device

Index	Object name
0x5003	WirelessQuality

C.4.12 WCycleTime

This index range stores all the information related to the WMasterCycleTime and WMinCycleTime of the W-Device.
Table 195 shows the WCycleTime information.

Table 195 WCycleTime information

Index	Subindex	Access	Parameter name	Coding	Data type
0x5007	0x00	Gives access to the whole index			
	0x01	R	WMinCycleTimeOut	See Figure 141	Octet
	0x02	R	WMinCycleTimeIn	See Figure 141	Octet
	0x03	R/W	WMasterCycleTimeOut	See Figure 141	Octet
	0x04	R/W	WMasterCycleTimeIn	See Figure 141	Octet

The WMasterCycleTime is a W-Master parameter and sets up the actual cycle time of a particular W-Port. The WMinCycleTime 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 141.

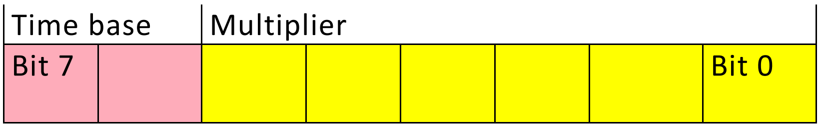


Figure 141 WMasterCycleTime and WMinCycleTime

Bits 0 to 5: Multiplier

These bits contain a 6-bit multiplier for the calculation of WMasterCycleTime or WMinCycleTime. 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 WMasterCycleTime or WMinCycleTime. The permissible combinations for time base and multiplier are listed in Figure 141 WMasterCycleTime and WMinCycleTime along with the resulting values for WMasterCycleTime or WMinCycleTime.

Time base encoding	Time Base value	Calculation	Cycle Time
00	-	-	FreeRunning
01	5 ms (3 W-Sub-Cycles)	Multiplier × Time Base	5 ms to 315 ms (3 to 189 W-Sub-Cycles)
10 to 11	Reserved	Reserved	Reserved
NOTE For W-Devices and W-Bridges the minimum possible transmission time is 5 ms.			

Figure 142 Possible values of WMasterCycleTime and WMinCycleTime

Annex D

(normative)

EventCodes

D.1 General

IO-Link Interface and System Specification [1] 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).

D.2 EventCodes for W-Devices

The Event Codes in Table D.1 – Event Code for Devices in [1] are also valid for W-Devices. D.2lists the additional EventCode identifiers and their definitions.

Table 196 EventCodes for W-Devices

Incident [a]	Origin	Instance	Name	EventCode	Action	Remark
D_IOLW_Retry_Error	REMOTE	APP	D_IOLW_RETRY_ERROR	0xFFB0	-	See Clause 11
HMI button pressed	REMOTE	APP		0xFFB1		
[a] All Events are of StatusCode type 2 (with details), EventQualifier type "Notification", EventQualifierMode "Single-shot"						

D.3 EventCodes for W-Ports

The Event Codes in Table D.3 – Event Code for Ports in [1] are also valid for W-Ports. Table 197 lists the additional EventCode identifiers and their definitions.

Table 197 EventCodes for W-Ports

Incident [1]	Origin	Instance	Name	EventCode	Action	Remark
M_IOLW_Retry_Error	LOCAL	APP	M_IOLW_RETRY_ERROR	0x3000	-	See Clause 11
IOLW_IMATimeout	LOCAL	APP	IOLW_IMATIMEOUT	0x3001	-	See Clause 11
IOLW_UniqueID_Changed	LOCAL	APP	IOLW_UNIQUEID_CHANGED	0x3002	-	
Reserved	LOCAL	APP		0x3003-0x3FFF		
[1] All Events are of StatusCode type 2 (with details), EventQualifier type "Notification", EventQualifierMode "Single-shot"						

Annex E
(normative)

Coding of ArgBlocks

E.1 Overview Arg Blocks

E.1.1 General

ArgBlock types and their ArgBlockIDs are defined in Table 198

Table 198 ArgBlock types and their ArgBlockIDs

ArgBlock type	ArgBlockID	Definition	Used by SMI_ services
MasterIdent	0x0001	[1] Annex E.2	SMI_MasterIdentification (see [1] 11.2.4)
WMasterConfigList	0x0200	See E.1.4	SMI_WMasterConfiguration (see [1] 11.2.5) SMI_ReadBackWMasterConfiguration (see [1] 11.2.6)
PDIn	0x1001	[1] Annex E.10	SMI_PDIn (see [1] 11.2.17)
PDOut	0x1002	[1] Annex E.11	SMI_PDOut (see [1] 11.2.18)
PDInOut	0x1003	[1] Annex E.12	SMI_PDInOut (see [1] 11.2.19)
On-requestData	0x3000	[1] Annex E.5	SMI_DeviceWrite (see [1] 11.2.10)
	0x3001		SMI_DeviceRead (see [1] 11.2.11)
DS_Data	0x7000	[1] Annex E.6	SMI_DSToParServ (see [1] 11.2.8) SMI_ParServToDS (see [1] 11.2.9)
DeviceParBatch	0x7001	[1] Annex E.7	SMI_ParamWriteBatch (see [1] 11.2.12) SMI_ParamReadBatch (see [1] 11.2.13)
IndexList	0x7002	[1] Annex E.8	SMI_ParamReadBatch (see [1] 11.2.13)
WPortPairing	0x7200	See E.1.6	SMI_WPortPairing
WPortConfigList	0x8200	See E.1.2	SMI_PortConfiguration (see [1] 11.2.5) SMI_ReadBackPortConfiguration (see [1] 11.2.6)
WScanConfigList	0x8201	SeeE.1.7	SMI_WScan
WPortStatusList	0x9200	See E.1.3	SMI_PortStatus (see [1] 11.2.7)
WScanStatusList	0x9201	See E.1.8	SMI_WScanStatus
WTrackStatusList	0x9202	See E.1.4	SMI_WTrackStatus
WQualityStatusList	0x9203	See E.1.9	SMI_WQualityStatus
DeviceEvent	0xA000	[1] Annex E.15	SMI_DeviceEvent (see [1] 11.2.14)
PortEvent	0xA001	[1] Annex E.16	SMI_PortEvent (see [1] 11.2.16)
VoidBlock	0xFFF0	[1] Annex E.17	SMI service management
JobError	0xFFFF	[1] Annex E.18	SMI service management

E.1.2 WPortConfigList

Table 199 specifies the WPortConfigList.

Table 199 WPortConfigList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x8200
2	PortMode	This element contains the TargetMode of the W-Device 0: Deactivated 52: Cyclic AutoPairing 53: Cyclic 54: Roaming AutoPairing 55: Roaming	Unsigned8	
3	Validation&Backup	See Table E.3 [1]	Unsigned8	
4	Reserved	(I/Q behavior in IO-Link)	Unsigned8	0
5	Reserved	(PortCycleTime in IO-Link)	Unsigned8	0
6	VendorID	See Table E.3 [1]	Unsigned16	1 to 0xFFFF
8	DeviceID	See Table E.3 [1]	Unsigned32	1 to 0xFFFFFF
12	Slot_N	Slot number of the W-Device	Unsigned8	0 to 7
13	Track_N	W-Track number of the W-Device	Unsigned8	1 to NumOfTracks
14	DeviceTxPower	Transmission power of the W-Device	Unsigned8	1 to 0x1F
15	MaxRetry	Number of Retries for process data	Unsigned8	2 to 0x1F
16	IMATime Base	IMA timeout base	Unsigned8	1 to 0x04
17	IMATime Multi	IMA timeout multiplier	Unsigned8	1 to 0xFF
18	SlotType	0: SSLOT 1: DSLOT	Unsigned8	0 to 0x01
19	LowEnergyDevice	0: Low Energy disabled 1: Low Energy enabled	Unsigned8	0 to 0x01
20	MaxPDSegLength	Maximum process data output segment length	Unsigned8	0 to 0x20
21	WMasterCycleTimeOut	See C.4.12		
22	WMasterCycleTimeIn	See C.4.12		
23	UniqueID	Unique identifier of the W-Device	OctetString[9]	0x000000000000000000 to 0xFFFFFFFFFFFFFFFFFF

E.1.3 WPortStatusList

Table 200 specifies the WPortStatusList ArgBlock.

Table 200 WPortStatusList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Unique ID	UINT16	0x9200
2	PortStatusInfo	See Table 120 in [1] with a new value 10: Pairing Fault	Unsigned8	0 to 0xFF
3	PortQualityInfo	See Table E.4[1]	Unsigned8	-
4	RevisionID	See Table E.4[1]	Unsigned8	0 to 0xFF
5	Reserved	(TransmissionRate in IO-Link)	Unsigned8	-
6	Reserved	(MasterCycleTime in IO-Link)	Unsigned8	-
7	InputDataLength	See Table E.4[1]	Unsigned8	0 to 0x20
8	OutputDataLength	See Table E.4[1]		0 to 0x20
9	VendorID	See Table E.4[1]	Unsigned16	1 to 0xFFFF
11	DeviceID	See Table E.4[1]	Unsigned32	1 to 0xFFFFFF
15	WMasterCycleTimeOut	See C.4.12	Unsigned8	
16	WMasterCycleTimeIn	See C.4.12	Unsigned8	
17	Number of Diags	This element contains the number x of diagnosis entries (DiagEntry 0 to DiagEntry x)	Unsigned8	0 to 0xFF
18	DiagEntry0	This element contains the "EventQualifier" and "EventCode" of a diagnosis (Event). See B.2.21 for coding and how to deal with "Event appears / disappears".	Struct Unsigned8/16	-
21	DiagEntry1	Further entries up to x if applicable…	…	-

E.1.4 WMasterConfigList

Table 201 specifies the WMasterConfig

Table 201 WMasterConfigList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0200
2	WMasterID	Master identifier for connection with W-Devices	Unsigned8	1 to 0x1D
3	Advanced Connectivity	Enable bit per functionality.	Unsigned16	Bit0 – AHT Bit[1:15] – Reserved
5	BlockList	Disabled frequencies, bit per frequency.	OctetString[10]	0x0000 0000 0000 0000 0000 to 0x7FFF FFFF FFFF FFFF FFFE
15	PairingTimeout	Timeout for pairing by BUTTON / UNIQUE in [s]	Unsigned8	5 to 0xFF
16	Reserved	-	Unsigned8	-
17	ServiceTrackN	W-Track number used for service requests (Scan/Pairing/Roaming). Service track must be enabled, if at least another track is enabled.	Unsigned8	1 to 0x05
18	ServiceTrackMode	Mode of service track 0: CYCLIC 1: ROAMING	Unsigned8	0 to 0x01
19	NumberOfTracks	Number of W-Tracks of a W-Master	Unsigned8	Number_Of_Tracks
20	Track1TxPower	Transmission power of the W-Track	Unsigned8	0: Disabled 1 to 0x1F
…				
19+N	TrackNTxPower	Transmission power of the W-Track	Unsigned8	0: Disabled 1 to 0x1F

E.1.5 WTrackStatusList

Table 202 specifies the WTrackStatusList ArgBlock. The actual length of the structure depends on the number of W-Tracks a W-Master has. If only a single track status is requested, Track0 is the requested W-Track.

Table 202 WTrackStatusList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x9202
2	Track1Mode	0 = DEACTIVATED 1 = CYCLIC 2 = ROAMING 3 = SCAN 4 = PAIRING	Unsigned8	0 to 0x04
3	Track1TxPower	W-Track transmission power	Unsigned8	1 to 0x1F
…				
2+2*(N-1)	TrackNMode	0 = DEACTIVATED 1 = CYCLIC 2 = ROAMING 3 = SCAN 4 = PAIRING	Unsigned8	0 to 0x04
3+2*(N-1)	TrackNTxPower	W-Track transmission power	Unsigned8	1 to 0x1F

E.1.6 WPortPairing

Table 203 specifies the WPortPairing ArgBlock.

Table 203 WPortPairing

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Identifier of the Argument Block	Unsigned16	0x7200
2	Pairing Command	0 = Unpairing 1 = PairingUnique 2 = PairingButton	Unsigned8	0 to 0x02

E.1.7 WScanConfigList

Table 204 specifies the WScanConfigList ArgBlock

Table 204 WScanConfigList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Identifier of the Argument Block	Unsigned16	0x8201
2	TxPower	Scan Transmission power	Unsigned8	1 to 0x1F
3	ScanBehavior	0: Scan Status needs to be polled by client with WScanStatus service 1: Each Result is indicated by WScanStatus initiated by the WMaster as broadcast message 2: Final Result is indicated by WScanStatus initiated by the WMaster as broadcast message	Unsigned8	0 to 0x02

E.1.8 WScanStatusList

Table 205 specifies the WScanStatusList ArgBlock. The list is cleared if a new scan is triggered. It wraps around on overflow in roaming mode.

Table 205 WScanStatusList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Identifier of the Argument Block	Unsigned16	0x9201
2	ScanStatus	Scan status 0: No Scan was performed yet 1: Scan in progress 2: Scan ended 3 to 255: Reserved	Unsigned8	0 to 0x02
3	ScanResults	Number of Scan Results	Unsigned8	0 to 0xFF
4	Res1SlotType	Slot type of the first Scan result	Unsigned8	0 to 0x01
5	Res1UniqueID	UniqueID of the first Scan result	OctetString[9]	
14	Res1RevID	RevisionID of the first Scan result	Unsigned8	
…				
4+11*(N-1)	ResNSlotType	Slot type of the last ScanResult	Unsigned8	0 to 0x01
5+11*(N-1)	ResNUniqueID	UniqueID of the last ScanResult	OctetString[9]	
14+11*(N-1)	ResNRevID	RevisionID of the last Scan result	Unsigned8	

E.1.9 WQualityStatusList

Table 206 specifies the WMasterStatusList ArgBlock. If quality status of only one port was requested, then only the first 6 octets are returned, otherwise the quality status of all ports are returned (ArgBlock length: 2 + 4*N).

Table 206 WQualityStatusList

Octet Offset	Element Name	Definition	Data Type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x9203
2	Port 1 LQI_M	LQI calculated by W-Master	Unsigned8	0 to 100
3	Port 1 RSSI_M	RSSI measured by W-Master	Signed8	-128 to 20
4	Port 1 LQI_D	LQI calculated by W-Device	Unsigned8	0 to 100
5	Port 1 RSSI_D	RSSI measured by W-Device	Signed8	-128 to 20
…				
2+4*(N-1)	Port N LQI_M	LQI calculated by W-Master	Unsigned8	0 to 100
3+4*(N-1)	Port N RSSI_M	RSSI measured by W-Master	Signed8	-128 to 20
4+4*(N-1)	Port N LQI_D	LQI calculated by W-Device	Unsigned8	0 to 100
5+4*(N-1)	Port N RSSI_D	RSSI measured by W-Device	Signed8	-128 to 20

Annex F

(normative)

Data Types

This annex refers to IO-Link Interface and System Specification [1], Annex F, 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 G

(normative)

Device design rules for low energy W-Devices

G.1 Low Energy W-Devices

G.1.1 General

For the design of low energy W-Devices, the following support is given by this specification to minimize power consumption:

G.1.2 Low voltage design

To minimize dissipation loss within the W-Device circuitry, the power supply voltage should be chosen as low as possible.

G.1.3 Event triggered activation.

To minimize transmitter activity, an uplink packet is only transmitted when the W-Device has new data to report, or the IMA-timer has expired.

G.1.4 Long IMATime

To minimize both receiver and transmitter activity, the maximum configurable IMATime should be chosen as long as possible.

G.1.5 Pre-Downlink

The Pre-Downlink mechanism is used to minimize the radio-on time to save power. A switch to full Downlink mechanism may be necessary on low energy W-Devices for a higher amount of data e.g., a parameter write.
To minimize receiver activity for synchronization, a W-Device should receive only the Pre-Downlink packet part, provided that no new data is send to the W-Device.

G.1.6 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.

G.1.7 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.

G.1.8 Establish communication.

After waking up a sleeping low energy W-Device (see 10.11.3.2) a paired W-Device starts to synchronize to the W-Master, respectively an unpaired W-Device is waiting for a W-Master Request, see 5.7.1
To minimize the power consumption the W-Device should go back to sleep after the recommended power on time, see Table 207. Therefore, the application has to start or stop the radio via SM_SetDeviceMode.

Table 207 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 213

G.1.9 HMI sleep

A low energy W-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 128 Visual states of W-Device for the remaining W-Device specific timeout.

G.2 Battery lifetime calculation

The following Equation 11 provides support for a rough estimation of battery lifetime for a W-Device.

$$T_{batt} = \frac{Q_{batt} \cdot \frac{1}{24h} \cdot \frac{1}{365d}}{\frac{(12.5\,ms + 0.416\,ms) \cdot I_{receive} + (0.2\,ms) \cdot I_{transmit} + (T_{sleep} + 0.208\,ms + 0.632\,ms) \cdot I_{sleep}}{12.5\,ms + 0.416\,ms + 0.2\,ms + 0.208\,ms + 0.632\,ms + T_{sleep}}}$$

Equation 11 Battery lifetime calculation part 1

= 12.5 ms Synchronization phase and 0.416 ms reception phase
= 0.2 ms Transmitting phase
= 0.208 ms Tx to Rx change phase, 0.632 ms inactive phase and application specific sleeping phase

Thus, the final calculation is shown in Equation 12:

$$T_{batt} = \frac{Q_{batt} \cdot \frac{1}{24\,h} \cdot \frac{1}{365\,d}}{\frac{12.9\,ms \cdot I_{receive} + 0.2\,ms \cdot I_{transmit} + (T_{sleep} + 0.84\,ms) \cdot I_{sleep}}{14.0\,ms + T_{sleep}}}$$

Equation 12 Battery lifetime calculation part 2

The parameters and typical values for battery lifetime calculation are listed in Table 208.

Table 208 Parameters and typical values for battery lifetime calculation

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 is not able to turn on and off immediately before/after its real active time, a tolerance of about 3% to 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 specification regarding timing values see Table 1.

Annex H

(normative)

Frequency Hopping Calculation

H.1 Blocklisting

Blocklisting is a mechanism to avoid on air collision with other wireless systems, such as WLAN. Conventional Bluetooth cannot be blocklisted, because it is an uncoordinated frequency hopper. The blocklist itself uses eighty 1 MHz wide frequency channels.

Table 209 Overview frequency channels for blocklisting

octet	0 (MSO)								1							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
channel number n	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
permitted value	**0**	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x

….

octet	8								9 (LSO)							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
channel number n	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
permitted value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	**0**

x can be 0 or 1:
1 means frequency channel is blocklisted
0 means frequency channel is not blocklisted

NOTE 1: The minimum number of allowed channels (x = 0) in the range of channel 3 to 78 shall be according to the following formula:

minimum number of allowed channels = max(15 , Number_Of_Tracks*Frequency_Spacing)

NOTE 2: For explanation, see 5.5.2.

The blocklisting 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 blocklisted WLAN channel shall be mapped to the blocklist format described in Table 210. The frequency blocks used by IO-Link-Wireless for blocklisting are shown in Table 210. The configuration channels 2401 MHz and 2480 MHz cannot be blocklisted.

Table 210 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

H.2 Creation of frequency hopping table HT01 with blocklisting

H.2.1 General

The creation of the frequency hopping table HT01 is divided into seven steps:

(i) Create an array with all available frequency channels within the 2,4 GHz ISM frequency band.
(ii) Find all blocklisted channels according to the provided blocklist 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 the 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)-1)
 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, for all available W-Tracks (Number_Of_Tracks). Thereby the frequency spacing should be taken into account.

```
Possible_Freq_Matrix = zeros(Number_Of_Tracks, length(array));
for (index = 0:length(array)-1)
  for (index_track = 0 : (Number_Of_Tracks -1))
     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 shall be performed using of Sequence number ***N*** and Prime index ***P*** as following:


```
for (index = 0:(P-1))
    Sequence_index = (N * index)% P;
     for (index_track = 0 : (Number_Of_Tracks -1))
          Frequency_Table(index_track, index) =
          Possible_Freq_Matrix(index_track, Sequence_index)
   end for;
 end for;
```

NOTE: If no permutation is created, a different MasterID shall be selected.

H.2.2 Example with 2 WLAN Channels

MasterID = 10;
Number_Of_Tracks = 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]
BlockList = [0x40 00 00 00 FF FF FE 7F FF FE] (See Table 211, **Figure 143**)
NOTE: The configuration channels shall not be blocklisted.

The influence of the given blocklist on the whole 2,4 GHz ISM Spectrum is demonstrated in Figure 143. If the blocklist is used the occupied frequencies given in are not used.

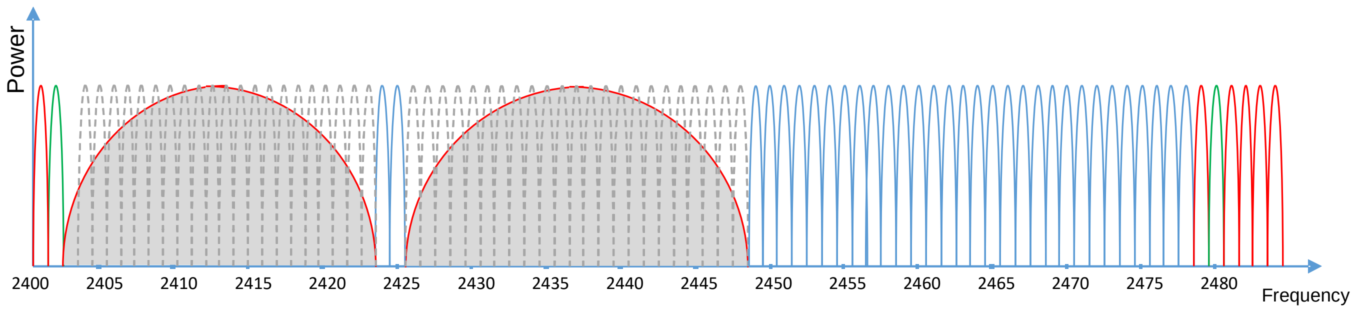


Figure 143 Blocklisting of 2 WLAN channels in 2,4GHz ISM band

Table 211 lists the occupied frequencies in this example.

Table 211 WLAN Channels 1 and 6 Blocklisting example

Blocklisted WLAN Channels	Center Frequency (MHz)	Occupied frequencies (MHz)
1	2412	2401-2423
6	2437	2426-2448

Calculating a frequency table using a given data:

Steps (i) and (ii): Find all not blocklisted channels, create an array:
array = {
2424 2425 2449 2450 2451 2452 2453 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 2449 2450 2451 2452 2453 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 and vii): Generate the Frequency table using Frequency matrix, the Permutation Index and a Sequence number.

	Frequency															
Track 1	2469	2457	2478	2466	2455	2476	2464	2453	2474	2462	2451	2472	2460	2449	2470	2458
Track 2	2472	2460	2449	2469	2458	2424	2467	2456	2477	2465	2454	2475	2463	2452	2473	2461
Track 3	2475	2463	2452	2472	2461	2450	2470	2459	2425	2468	2457	2478	2466	2455	2476	2464
Track 4	2478	2466	2455	2475	2464	2453	2473	2462	2451	2471	2460	2449	2469	2458	2424	2467
Track 5	2449	2469	2458	2478	2467	2456	2476	2465	2454	2474	2463	2452	2472	2461	2450	2470

Track 1	2424	2467	2456	2477	2465	2454	2475	2463	2452	2473	2461	2450	2471	2459	2425
Track 2	2450	2470	2459	2425	2468	2457	2478	2466	2455	2476	2464	2453	2474	2462	2451
Track 3	2453	2473	2462	2451	2471	2460	2449	2469	2458	2424	2467	2456	2477	2465	2454
Track 4	2456	2476	2465	2454	2474	2463	2452	2472	2461	2450	2470	2459	2425	2468	2457
Track 5	2459	2424	2468	2457	2477	2466	2455	2475	2464	2453	2473	2462	2451	2471	2460

Figure 144 Frequency Table for 5 Tracks

H.2.3 Example with one WLAN channel

MasterID = 9;
Number_Of_Tracks = 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]
BlockList = [0x40 00 00 00 FF FF FE 00 00 02] (See Table 7, Figure 145)

The influence of the given blocklist on the whole 2,4GHz ISM Spectrum is demonstrated in Figure 145 Blocklisting of one WLAN channel in 2,4GHz ISM band. If the blocklist is used the occupied frequencies given in Table 212 shall be omitted.

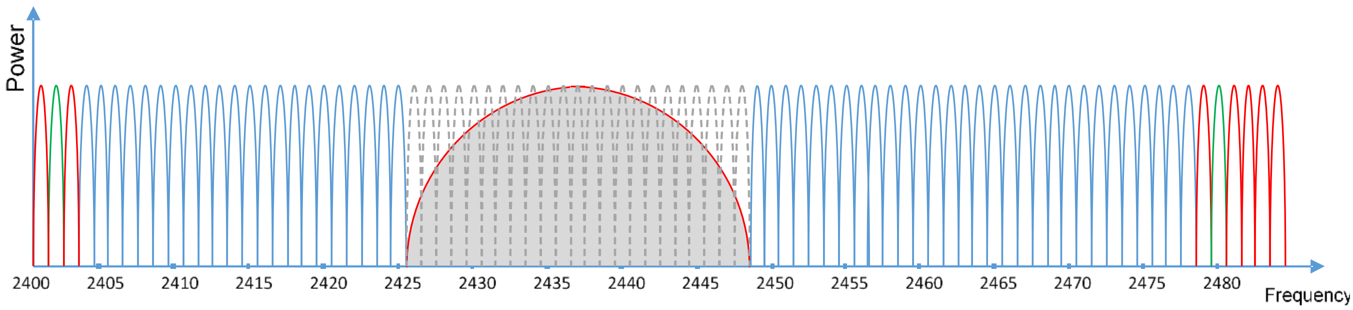


Figure 145 Blocklisting of one WLAN channel in 2,4GHz ISM band

Table 212 WLAN Channel 1 Blocklisting example

Blocklisted WLAN Channels	Center Frequency (MHz)	Occupied frequencies (MHz)
6	2437	2426-2448

Calculating a frequency Table using a given data

Steps (i)+(ii): Find all not blocklisted channels, create an array:
array = {
2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2449 2450 2451 2452 2453 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 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469}

Step (iv): Find Permutation index P:

Length(array) = 53
P = max(Primes <= 53) = 53.

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) = (52/2) – (8/2) = 26 – 4 = 22

Step (vi and vii): Create a Matrix with the possible frequencies respecting the frequency spacing

	Frequency																										
Track 1	2470	2416	2461	2407	2452	2474	2420	2465	2411	2456	2478	2424	2469	2415	2460	2406	2451	2473	2419	2464	2410	2455	2477	2423	2468	2414	2459
Track 1	2405	2450	2472	2418	2463	2409	2454	2476	2422	2467	2413	2458	2404	2449	2471	2417	2462	2408	2453	2475	2421	2466	2412	2457	2403	2425	

Figure 146 Frequencies Table for 1 W-Track

H.3 Modified Sequence for ServiceMode

As stated in 5.4.4 “Configuration Channel”, the ServiceMode shall 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 5^{th} W-Sub-cycle with a configuration frequency. Therefore, one of the configuration frequencies 1 (2401) and 80 (2480) shall be used every 5^{th} 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 10^{th} 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 147).

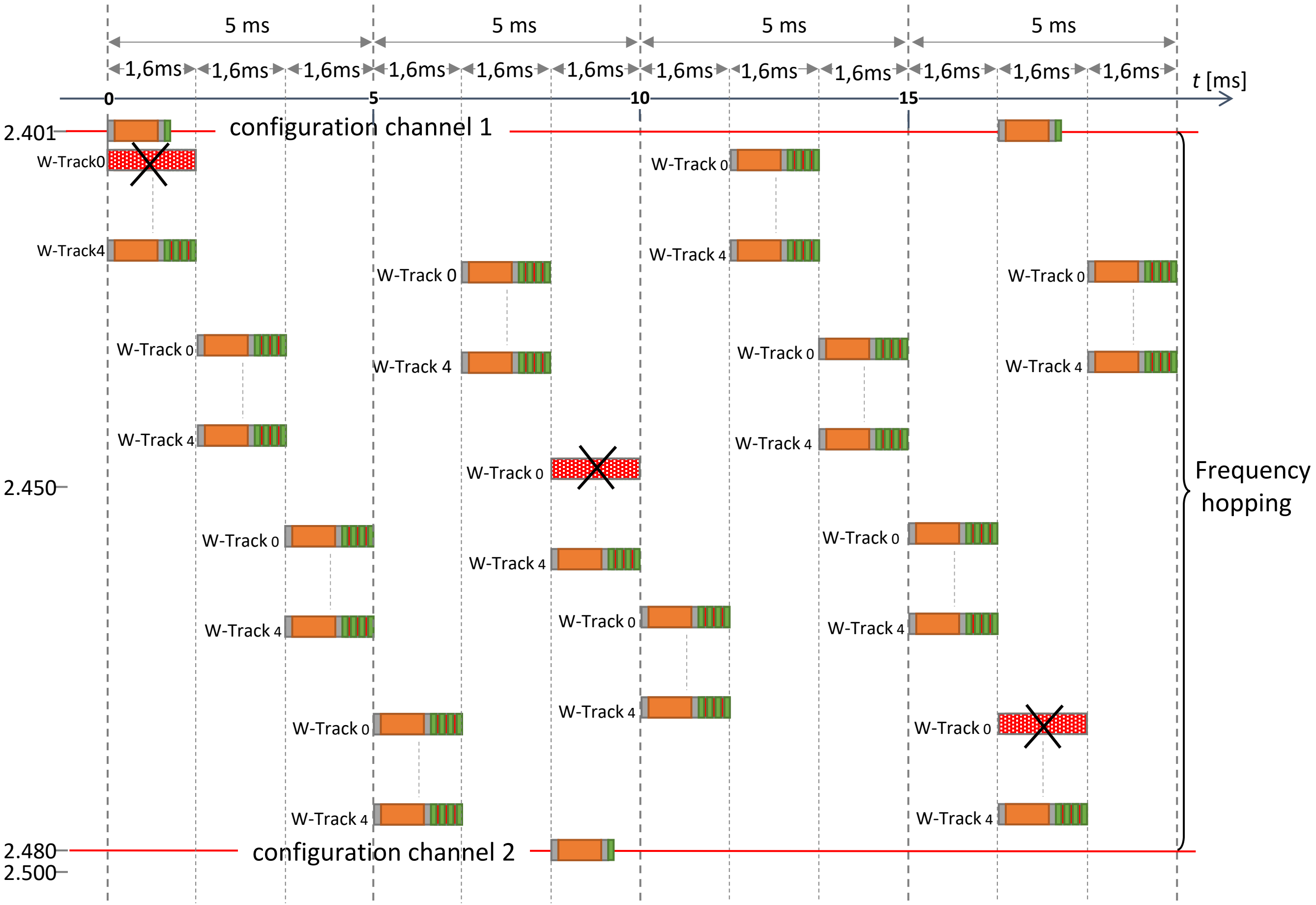


Figure 147 Usage of the configuration frequencies

Figure 147 shows an example of W-Track_0 in ServiceMode. In this W-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 213. The W-Device shall leave the ServiceMode directly after the exchange of the ConnectionParameter.

Table 213 Timeouts for ServiceMode

Timeouts:	W-Master	W-Device
SCAN_TIMEOUT	5 s	-
PAIRING_UNIQUE_TIMEOUT	5 s	200 ms
PAIRING_BUTTON_TIMEOUT	min. 5 s	200 ms

H.4 Adaptive Hopping Table (AHT)

Adaptive hopping table enables to update the hopping table per W-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 W-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 148. W-Master can then trigger a countdown via MasterCommand to all updated W-Devices to synchronize a switch to the new hopping table values, starting with Index 1, see Figure 149. 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
-

Hopping table calculations other than HT01 are reserved for future versions. Therefore, only the update of a FULL_TABLE (see Table 214) is allowed.

Table 214 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 Blocklist.

Table 214 AHT UpdateType

UpdateType Value	UpdateType Name	Description	Set Frequency Index to	Frequency value DataType
0	FULL_TABLE	Replace all frequencies of the hopping table, see Figure 146.	0	OctetString
1	DELETE_CELL	Deletes a frequency from the hopping table according to index. Index x deletes HOP-x. All Indexes from x+1 shall be shifted down by 1	1-78	-
2	ADD_CELL	Adds a frequency to the hopping table. Index x adds HOP-x. All Indexes from x shall be shifted up by 1.	1-78	1 Octet
3	REPLACE_CELL	Replace a frequency in the hopping table, index x replaces HOP-x frequency	1-78	1 Octet

Figure 148 shows an example for using MCmd(WakeUp) in combination with MCmd(Jump) to set a new hopping table. 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 W-Track.

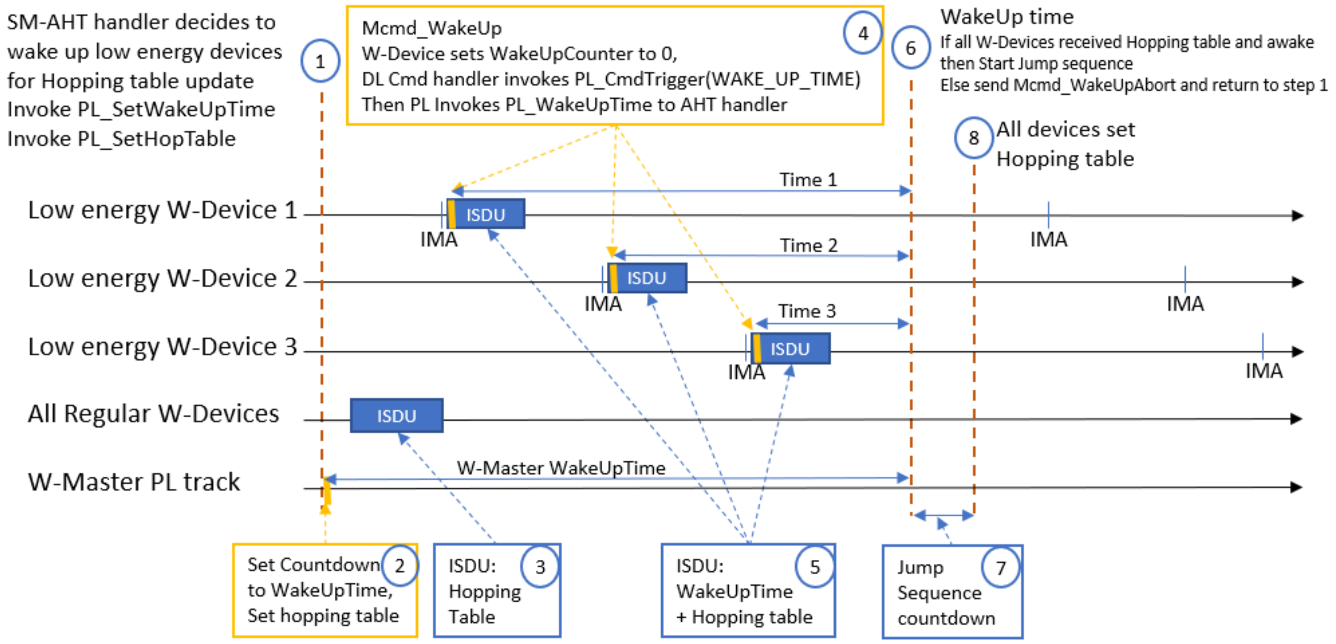


Figure 148 Changing Hopping Table synchronization timeline

Figure 149 shows the sequence of WakeUp of low energy W-Device

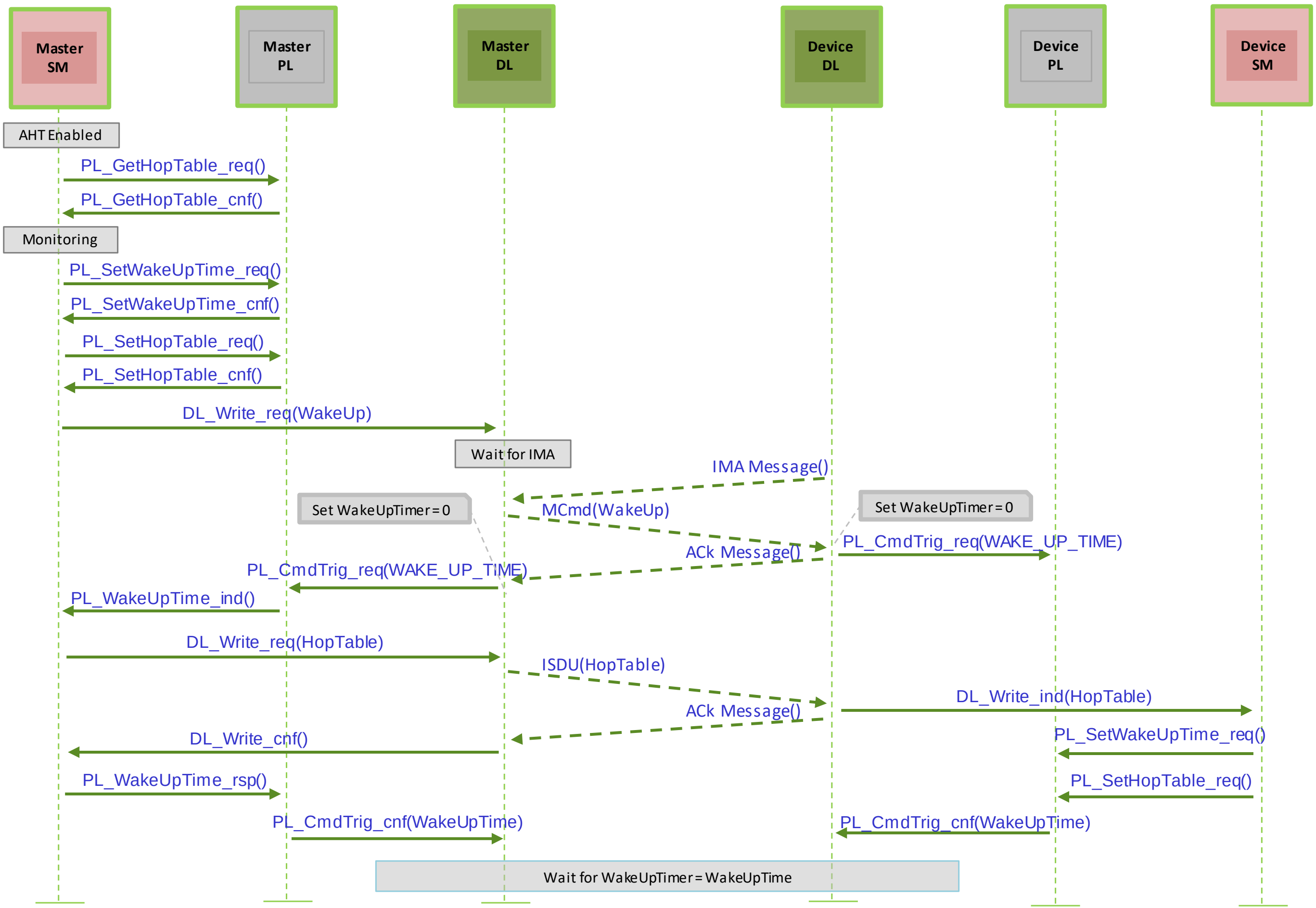


Figure 149 AHT WakeUp sequence

Figure 150 shows the Jump sequences

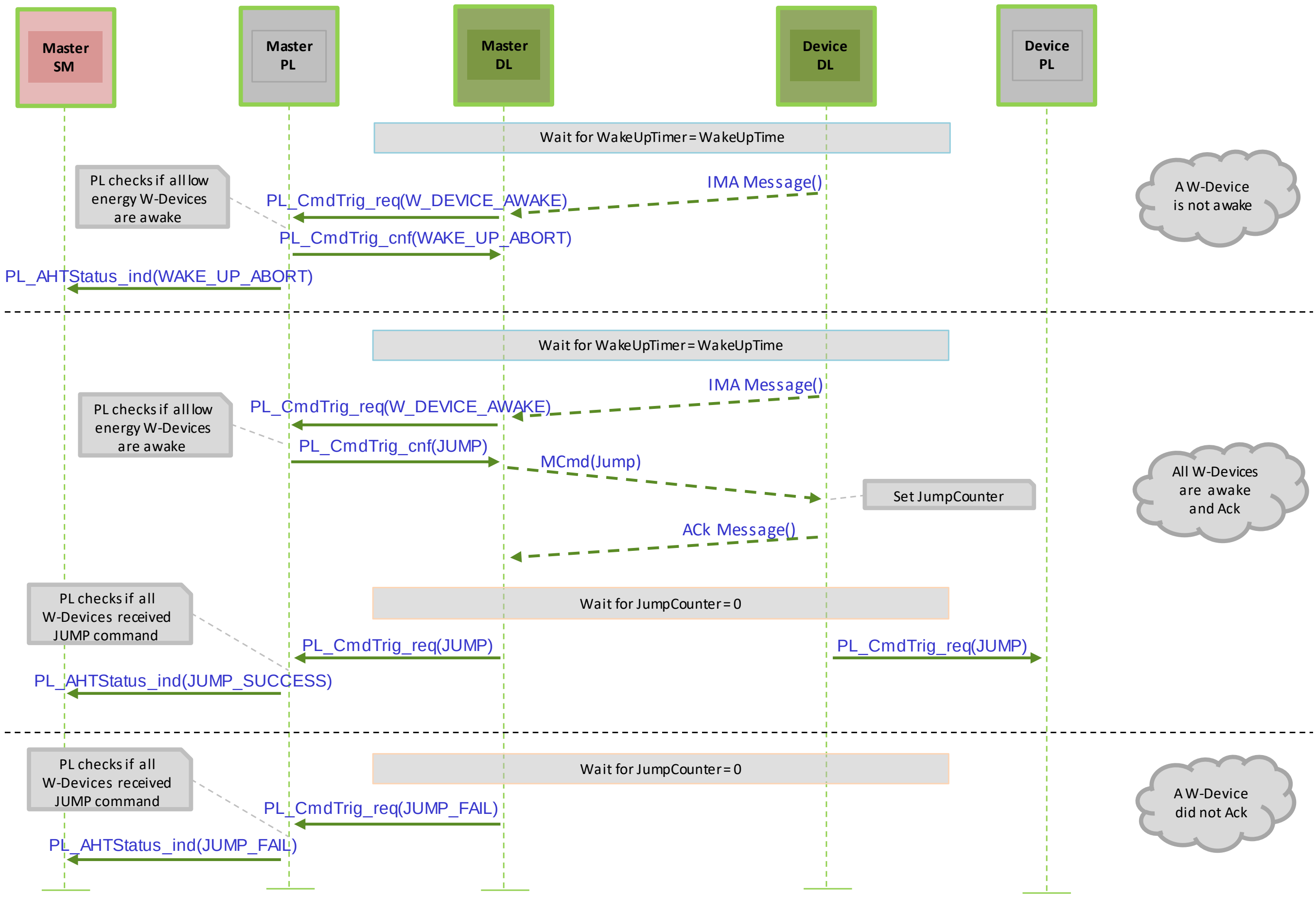


Figure 150 AHT Jump sequence

Annex I

(normative)

How to get a certified product

I.1 General

In order to get a certified IO-Link Wireless product, different testing and certification aspects shown in clause I.2 and I.3 shall be considered:

I.2 Radio Certification

To satisfy the legal jurisdiction under which the wireless equipment shall be used, the locally valid regulatory compliance rules for wireless equipment shall be fulfilled. Currently relevant regulations are outlined in more detail in Annex I “Regulatory Compliance”.

I.3 IO-Link Certification

Compliance to the IO-Link Wireless protocol defined in this standard shall 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 [11].

Annex J

(normative)

Regulatory Compliance

J.1 General

This Annex I provide 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 J.2). Different requirements apply in Europe, which can be met by complying with FCC 15.247 in combination with harmonized standards EN 300 328 (see clause J.3) and EN 300 440 (see clause J.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 [5] ("Bluetooth Low Energy Regulatory Aspects").

Additionally, ETSI Guide EG 203 367 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.

J.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 [7].

J.3 Compliance with ETSI EN 300 328

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 W-Track (antenna) is used in a device, all W-Tracks are sharing the 10 dBm. For example, 3 dBm per W-Track are permitted in case of five W-Tracks.
The required test suites shall be carried out and compliance declared for the relevant technical requirements see [8].

J.4 Compliance with ETSI EN 300 440

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 [9] and [10].

Annex K
(informative)

Radio field planning

K.1 General

The following section explains in a highly simplified and abbreviated form how to estimate the range and reliability of an IOLW system. For simplification, some assumptions (e.g., no interferers, etc.) are made.

K.2 Range and reliability examination

K.2.1 General

Typically, the sensitivity is specified as the minimum receive power required to achieve a certain bit error probability (BEP), i.e., 10^{-3}. In the following two figures Figure K.1 and Figure K.2, the BEP as a function of received power is exemplarily shown for two example chips, based on theoretical assumptions.

Sensitivity EXAMPLE CHIP 1
BEP = 1*10-3 => sensitivity level -91 dBm

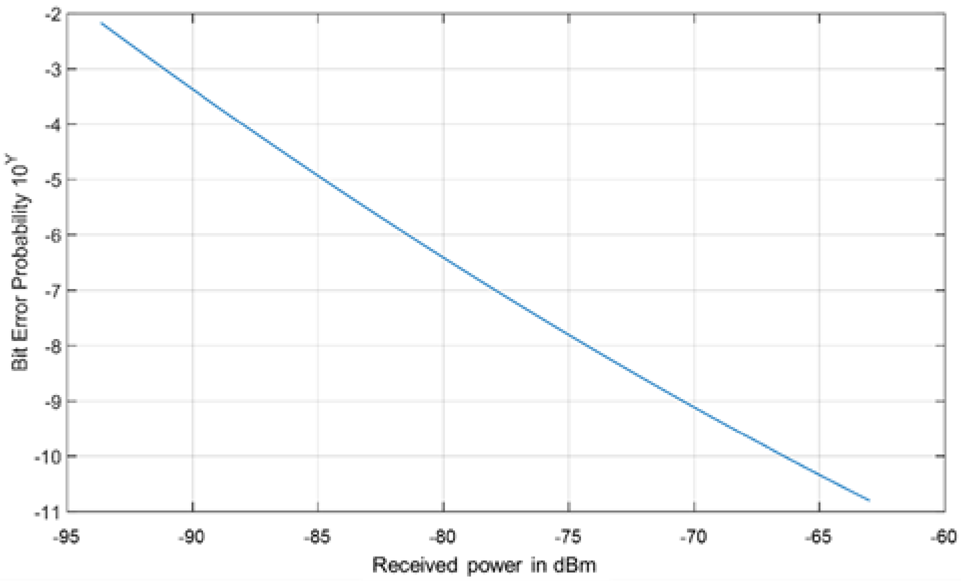


Figure K.1 BEP over received power EXAMPLE CHIP 1 (based on measurements, interpolated).

Sensitivity EXAMPLE CHIP 2
BEP = 1*10-3 => sensitivity level -94 dBm

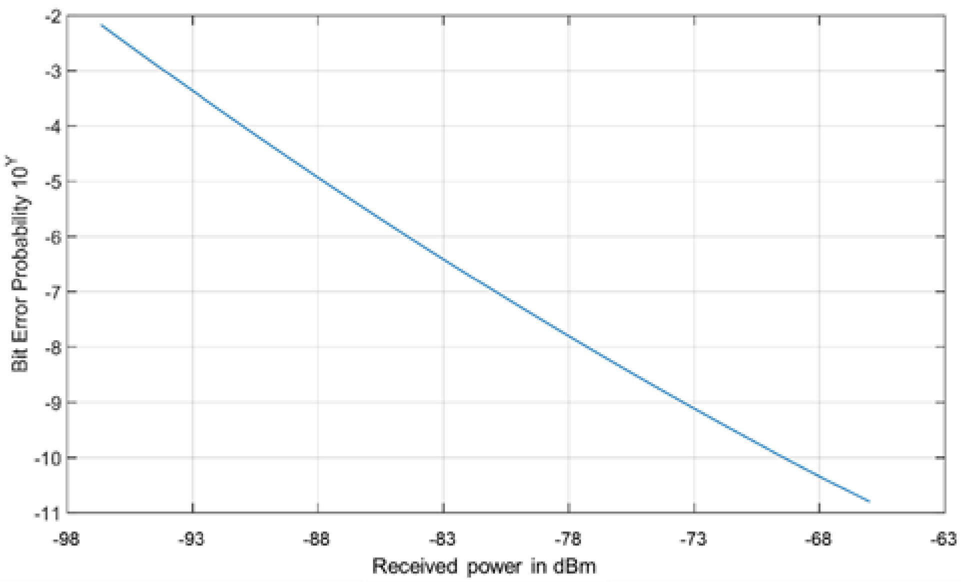


Figure K.2 BEP over received power EXAMPLE CHIP 2 (theoretically derived from previous measurements).

For packet-based communication such as IO-Link Wireless, the BEP is usually not directly measured, but it can be related to the Packet Error Probability (PEP), subject to certain conditions.
In a highly simplified form and under the condition, that at least one-bit error within a packet results in a packet error, the PEP can be calculated from the BEP as follows

$$PEP = 1 - (1 - BEP)^{n\_Bits} ,$$

whereby n_Bits is the number of bits within a packet. For example, for an IO-Link Wireless downlink packet and BEP = 1*10-3 = 0.001, this results in approx. PEP≈0.3 .
Thus, the sensitivity can be derived by a packet error measurement (PEP). It can be performed either, if possible (i.e., accessible antenna connector), as a conducted measurement (i.e., in dBm) or as a radiated (over-the-air) measurement (i.e., dBm EIRP). However, for a final IO-Link Wireless product, the losses caused by the antenna (antenna efficiency) and the matching circuit (including e.g., a “balun”) have to be taken into account.
In general, IOLW does not prescribe a specific orientation of W-Devices and W-Masters in an application environment. Thus, the later positioning or orientation of e.g., W-Devices is generally unknown for a manufacturer, so that all conceivable scenarios with regard to spatial directions of incidence and polarizations must be taken into account in the test. This can be done in a relatively simple way using a reverberation chamber (RC).
Based on the well-proven Friis transmission formula e.g.,[12],[13], the path-loss in free space results in approximately the following attenuation for the given ranges and the 2,4 GHz band:

Friis -> Free space path loss:

2 m	-46 dB
4 m	-52 dB
8 m	-58 dB
10 m	-60 dB
20 m	-66 dB

A negative sign means attenuation here.
As a worst-case estimation, a radio channel without direct line of sight (NLOS = Non-Line of Sight) is assumed which is modelled or described as Rayleigh radio channel. It was shown in e.g. [14], [15] that a obstructed or direct line of sight (OLOS or LOS) or path between a transmitter and a receiver can be described e.g., as a Rician radio channel (Rice channel) Furthermore it was shown in e.g. [14], [15] that deep fading, which can lead to interruptions or outages of a radio connection, occur much less frequently with LOS connections. Therefore, a NLOS or Rayleigh radio channel is assumed to be a worst-case scenario.

K.2.2 Rayleigh -> Non-Line of Sight:

Figure K.3 shows the empirical Probability Density Function (PDF) of an exemplary Rayleigh channel, measured in a reverberation chamber e.g., [16], [17].

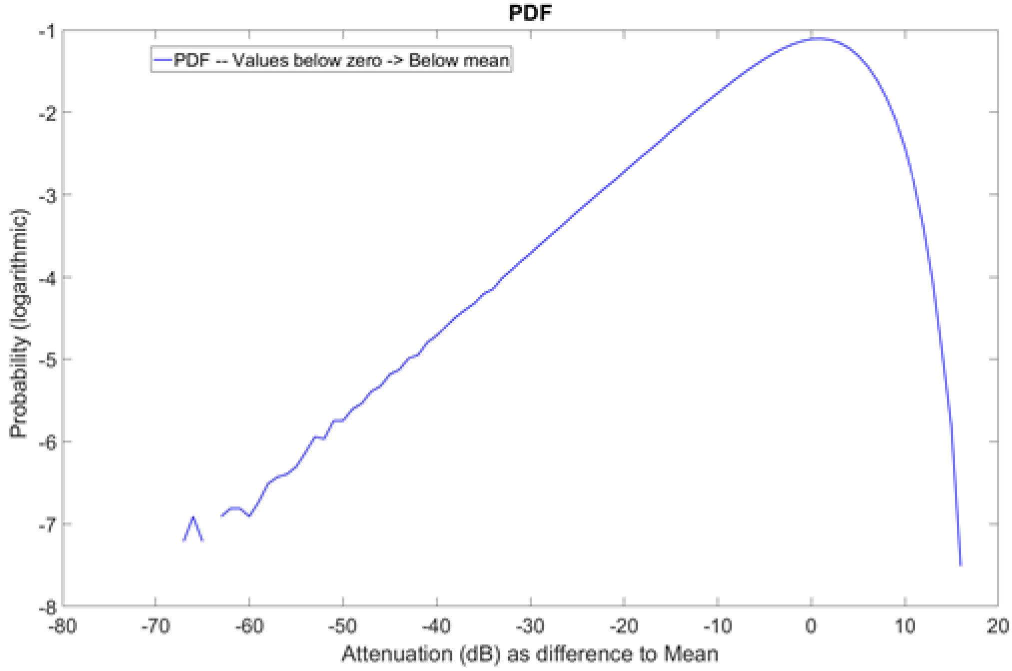


Figure K.3 Exemplary Rayleigh Channel, measured in a reverberation chamber

Probability of attenuation increasing by X dB against mean receive power.
Assuming a radiated transmission power of +2 dBm (EIRP) and the free space path-loss for 20 m results in a (mean) receive power of approximately -64 dBm. Figure K.4 shows the cumulative distribution function (CDF) of the so-called “IO-Link Wireless channel” (i.e., including free space path-loss and Rayleigh fading), normalized to a mean receive power of -64 dBm.

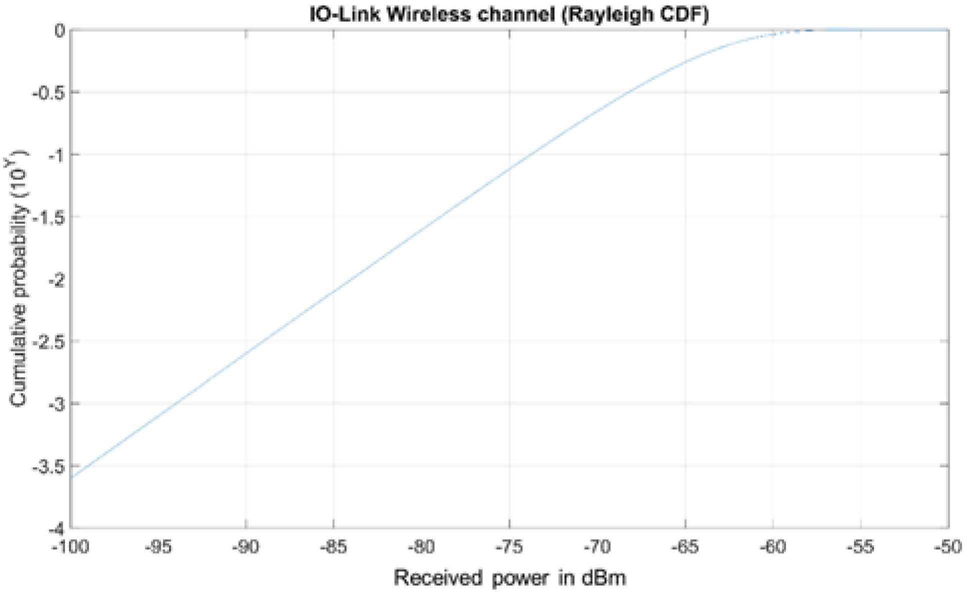


Figure K.4 – Expected IO-Link Wireless channel with Rayleigh CDF (mean power -64 dBm).

As in typical industrial environments the coherence time is orders of magnitudes larger than the packet duration and the coherence bandwidth is also much larger than the signal bandwidth, can be proceeded as follows: Using the resulting CDF in Figure K.4 and the Sensitivity of EXAMPLE CHIP 2 we achieve following example probabilities P for certain fading:

P (less than 10 dB below mean …)≈88.9% ⇒ ≫-74 dBm⇒BEP≪1·10^(-9)→PEP≪4·10^(-7)
P(10 dB below mean receive power)≈0.1 ⇒-74 dBm⇒BEP≈1·10^(-9)→PEP≈4·10^(-7)
P (15 dB below mean receive power)≈0.03 ⇒-79 dBm⇒BEP≈3·10^(-8)→PEP≈1·10^(-5)
P (20 dB below mean receive power)≈0.01 ⇒-84 dBm⇒BEP≈7·10^(-7)→PEP≈3·10^(-4)
P (25 dB below mean receive power)≈0.003 ⇒-89 dBm⇒BEP≈2·10^(-5)→PEP≈8·10^(-3)
P (30 dB below mean receive power)≈0.001 ⇒-94 dBm⇒BEP≈1·10^(-3)→PEP≈3·10^(-1)
P (35 dB or more below mean receive power)≈0.0003 ⇒-99 dBm⇒BEP≈0.1→PEP≈1

To obtain the (mean) error probability over all possible receive power levels of the (Rayleigh-) radio channel, the integral of the product of packet error probability as a function of the reception power and its probability density is calculated e.g. [18], [19], [20]. However, this requires a contiguous knowledge of the PEP as a function of the receive power. If the PEP as a function of receive power is known in steps (e.g., by stepwise measurements), the integration can be approximated by stepwise summation. With the given values follows:

$$\overline{PEP}_{Rayleigh} \approx 0.7 \cdot 10^{-3}$$

Using this intermediate result, the total error probability for a one-way transmission and for 3 attempts (i.e., 2 retries) can be calculated as follows:

The error probability for 3 statistically independent attempts using the numbers before results in the residual failure probability (RFP) which is here (for 2 retries):

$$RFP = \overline{PEP}_{3\ attempts,Rayl.} \approx (0.7 \cdot 10^{-3})^3 \approx \mathbf{3.4 \cdot 10^{-10}}$$

K.2.3 Conclusion (EXAMPLE CHIP 2):

- The (weighted) packet error probability for a distance of about 20 meter (mean receive power = -64 dBm, Rayleigh channel, +2 dBm transmission power, -94 dBm sensitivity @BEP =10^-3),is about (PEP)¯_Rayleigh≈0.7·10^{-3}.
- Error probability by 3 attempts (mean receive power = - 64 dBm, Rayleigh channel, +2 dBm transmission power, -94 dBm sensitivity @BEP =10-3) is RFP≈3.4·10^{-10}
- Thus, a one-trip error probability of less than 10-9 is achievable.
- Additional losses (e.g., caused by antenna cable, efficiency, etc.) have to be compensated for by increased transmission power or better sensitivity in order to achieve the same range and reliability.
- The numerical values mentioned above open up a small margin for shifts: With higher transmission power, a higher reliability and/or range can be achieved or, within certain limits, a correspondingly poorer sensitivity can be tolerated as long as all communication participants meet the corresponding minimum requirements.

Annex L

(informative)

L.1 Coexistence management

This annex describes recommendations on how IO-Link Wireless as a system as well as IO-Link Wireless products can be classified in the context of coexistence management according to IEC 62657.

Table 215 characterizes the "Wireless System Type" properties of IO-Link Wireless according to IEC 62657-2.

L.2 Wireless System Type

Table 215 Wireless System Type

Parameter name	Value	Unit	Usage	Remark
Wireless technology or standard	IO-Link Wireless / IEC 61139-3 SDCI-W	N/A	mandatory	
Regional radio regulations	ETSI EN 300 328, ETSI EN 300 440, FCC 15.247	N/A	selection	
Network topology	star	N/A	mandatory	
Wireless device density		devices per m²	selection	Not applicable
Infrastructure device	no	N/A	mandatory	
Frequency band	2400000000 to 2483500000	Hz	mandatory	without blocklist
Lower cut-off frequency	2400450000	Hz	mandatory	without blocklist
Upper cut-off frequency	2480550000	Hz	mandatory	without blocklist
Frequency hopping procedure	see Specification IEC 61139-3, Annex H	N/A	mandatory	
Modulation	GFSK	N/A	mandatory	
Communication reliability	at least 5.000.000	s	mandatory	at least
Transfer interval	0.005	s	selection	or larger
Transmission gap	0.000208	s	mandatory	minimum
Transmitter sequence	0.001664	s	mandatory	maximum
Dwell time	0.001664	s	mandatory	
Medium access control mechanism	F-/TDMA	N/A	mandatory	
Mechanism for adaptivity	Channel Blocklisting	N/A	mandatory	
Security level	SL3	N/A	mandatory	

Table 216 shows an example to describe a "Wireless Device Type" according to IEC62657-2. According to that document, a "Device" can be an IO-Link W-Master, W-Device or W-Bridge. Thus, the "Wireless Device Type" description according to IEC 62657-2 is product specific. It is recommended that such a table is provided for each IO-Link Wireless product.

Table 216 Wireless Device Type

Parameter name	Value	Unit	Usage	Remark
Transmitter				
Antenna type	e.g., omni-directional antenna; directional antenna; PCB antenna	N/A	selection	
Antenna gain	e.g., +3	dBi	optional	maximum gain
Antenna radiation pattern		N/A	optional	Figure
equivalent isotropic power	e.g., 0,005	W	optional	EIRP
Power spectral density		dBm/ Hz	optional	Figure
Frequency channel	1-80, according to Specification	N/A	mandatory	without blocklisting
Communication reliability	at least 5.000.000	s	mandatory	at least
Transfer interval	0.005	s	selection	at least
Transmission gap	0.001228 or 0.001444 or 0.001548	s	mandatory	minimum
Transmitter sequence	0.000416 or 0.000204 or 0.000096	s	mandatory	maximum
Duty cycle	W-Master: 25%, W-Device 12% or 6%	%	mandatory	maximum
Dwell time	0.001664	s	mandatory	

Bibliography

[1] IO-Link Community, IO-Link Interface and System Specification, V1.1.3, June 2019, Order No. 10.002(available at http://www.io-link.com)

[2] IEC 61131-9, Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

[3] IO-Link Community, IO Device Description (IODD), V1.1.3, January 2021, Order No. 10.012 (available at http://www.io-link.com)

[4] IO-Link Community, IO-Link Smart Sensor Profile 2nd Edition, V1.1, September 2021, Order No. 10.042 (available at http://www.io-link.com)

[5] Bluetooth SIG - Regulatory Committee, "Bluetooth Low Energy Regulatory Aspects", V10r00, 26 April 2011

[6] IO-Link Community, IO-Link Common Profile, V1.1, December 2021, Order No. 10.072 (available at http://www.io-link.com)

[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"

[8] ETSI EN 300 328 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"

[9] ETSI EN 300 440 "Short Range Devices (SRD); 6690 ETSI EN 300 440 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"

[10] ETSI EG 203 367 Short Range Devices (SRD) 6695 ETSI EG 203 367 "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"

[11] IO-Link Wireless System Extensions Test Specification, V1.1.3, January 2023, Order No. 10.242 (available at IOLW-TestSpec V1.1.3 January 2023 (io-link.com)), (under preparation)

[12] A Note on a Simple Transmission Formula, H.T. Friis, Proceedings of the IRE, Volume 34, Issue 5, May 1946, https://doi.org/10.1109/JRPROC.1946.234568

[13] Base Station Antenna Placement of Wireless Sensor/Actuator Networks in Manufacturing Cells, C. Cammin, D. Krush, H. Wattar, R. Heynicke, G. Scholl, 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 10.-13. Sept. 2019, Zaragoza, Spain, https://doi.org/ 10.1109/ETFA.2019.8869331

[14] Wireless Communications: Principles and Practice, T.S. Rappaport, Prentice Hall, 1996.

[15] Test method for narrowband F/TDMA-based wireless sensor/actuator networks including radio channel emulation in severe multipath environments, C. Cammin, D. Krush, R. Heynicke, G. Scholl, J. Sens. Sens. Syst., 7, 183-192, https://doi.org/10.5194/jsss-7-183-2018, 2018.

[16] Charakterisierung einer Modenverwirbelungskammer als Testumgebung für drahtlose Sensor/Aktor-Module, C. Cammin, D. Krush, R. Heynicke, G. Scholl, tm – Technisches Messen, 2018, ISSN (Online) 2196-7113, ISSN (Print) 0171-8096, DOI: https://doi.org/10.1515/teme-2018-0003

[17] Deep Fading in a Reverberation Chamber for Wireless Device Testing, C. Cammin, D. Krush, R. Heynicke, G. Scholl, IOP Conference Series: Materials Science and Engineering, Volume 766,

IEEE Radio and Antenna Days of the Indian Ocean 2019 23.–26. Sept. 2019, Saint-Gilles, Réunion, France DOI: https://doi.org/10.1088/1757-899x/766/1/012004

[18] Simulative performance analysis of IO-link wireless, D. Wolberg, M. Rentschler, P. Gaggero, 14th IEEE International Workshop on Factory Communication Systems (WFCS), 13.-15. June 2018, https://doi.org/10.1109/WFCS.2018.8402352

[19] Grundlagen der Kommunikationstechnik, J.G. Proakis, M. Salehi, 2. Auflage, Pearson Studium, 2004. ISBN: 3-8273-1064-7

[20] Communication Systems Engineering, J.G. Proakis, M. Salehi, 2nd edition, Pearson Education Prentice Hall, 2002.

© Copyright by:

IO-Link Community
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 986 197 0
Fax: +49 (0) 721 / 986 197 11
e-mail: info@io-link.com
http://www.io-link.com/