CC-Link IE Field Network / PROFINET coupler specification

version 0.92d – Date: September 2016 - PI Order-No.: 2.822

Contents

List of Figures

List of Tables

Revision Log

Sub number	Revision description	Issued
i0.1d	Initial Draft specification	Working Group
i0.2d	Diagnosis Chapter introduced	Working Group
i0.3d	Internal version. Review comments from Ishiyama-san. All chapters upgrade.	Working Group
i0.4d	Internal version as of July 22 All chapters upgrade.	Working Group
i0.5d	Take account of review comment ahead of F2F meeting	Working Group
i0.6d	Take account of F2F meeting.	Working Group
i0.7d	Working document	Working Group
i0.8d	RC0 for WG internal review	Working Group
i0.9d	RC1 for WG internal review	Working Group
i0.91d	RC2 for WG internal review	Working Group
i0.92d	RC3 for WG internal review	Working Group

1 Management Summary - Scope of this Document

The purpose of this document is to specify a coupling device between PROFINET and CC-Link IE Field Network.

By coupling device, it is meant a system capable to interconnect two networks, one supporting PROFINET, the other one supporting CC-Link IE Field as protocol, by making available a configurable set of values to the other system, acting as a slave on each network. Such a system is identified as being a “Coupler” although this document. The figure below gives an overview of coupler’s concept.

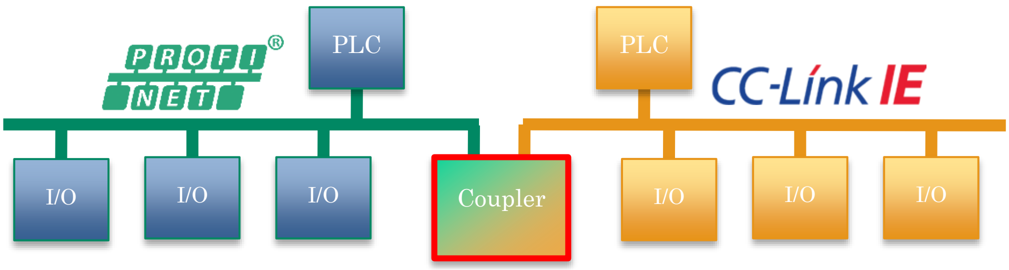

Fig. 4.1-1 coupling PROFINET / CC-Link IE Field

During the writing of this specification, other coupling device concepts have been identified, integrating a master. Such kind of coupling device are referred as a “Link”. Link is out of scope of this specification.

The scope of this document is to provide a normative specification for the PROFINET/CC-Link IE Field coupler. Therefore normative statements like **"shall"**, **"should"** and **"may"** are used.

The configuration software of the coupler is out of scope of this document.

2 List of affected patents

There is no affected patent known by the members of the working group. The list is empty. No patent search, neither external nor internal, has been done by the members of the working group up to now.

CLPA and PROFIBUS&PROFINET International do not guarantee the completeness of this empty list.

3 Related Documents and References

Table 4.1-1 Related Documents

Reference	Description
[1]	BAP-C2005 CC-Link IE Field pecification
[2]	BAP-C2006 SLMP (Seamless Message Protocol) Specification
[3]	BAP-C2008 Control & Communication System Profile Specification
[4]	PROFINET specification V2.3Ed2MU3
[5]	GSDML specification V2.33

[6]	CC-Link IE Field Network intelligent device station conformance test specifications BAP-C0401-037
[7]	PROFINET test specification V1.00 (Testspec-PN_2572_V0100_Jan16.pdf)

4 Definitions and Abbreviations

4.1 Definitions

Table 4.1-1 Definitions

Coupler	A device with two network interfaces, acting as a PROFINET IO-Device on one interface, acting as CC-Link IE Field slave on the other.
Link	A device with two network interfaces, acting as a slave on one interface and acting as a master on the other interface.
network master	A network node setting up connection with other network nodes.
CC-Link IE Field Master	A device used as a master to enable control communication on the CC-Link IE Field Network
CC-Link IE Field Slave	A device used as a salve to exchange data on the CC-Link IE Field Network
PROFINET IO-Controller	A device used as a master to enable control communication on the PROFINET network
PROFINET IO-Device	A device used as a slave to exchange data on the PROFINET network
CC-Link IE Field Intelligent Device	A device used as a slave has functions other than exchange data
Mapping Model	The exact logical data model used in coupler to describe data exchange between different networks
Conversion Model	One part of Mapping Model, used by coupler users to define links between different networks
Subsystem	A system linked by one of the coupler network interface, in which master(s) and slave(s) are connect on the network
Device Description Files	Files to describe the information of network slave.
Coupler Variable	This is the master variable to be transported from one network to the other network.

<table>
<tr><td>Coupling Application (CA)</td><td>Application inside the coupler is responsible to map information from one side of the coupler to the other, and vice versa.</td></tr>
<tr><td>Consistency Key (CK)</td><td>This key is generated by the coupler configuration tool, allowing to prove GSDML and CSP+ files, are describing the same coupler configuration.</td></tr>
<tr><td>Coupler Configuration Tool (CCT)</td><td>Tool computer based allowing user to generate and download configuration of coupling application to the coupler.</td></tr>
<tr><td>LSB/MSB</td><td>Least Significant bit (in Red)/ Most Significant bit (in Blue)
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>MSB</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>LSB</td></tr>
</table></td></tr>
</table>

4.2 Abbreviations

Table 4.2-1 Abbreviation

CSP+	CC-Link Family System Profile
GSDML	Generic Station Description Markup Language
N/W	Network
I/F	Interface
CRXB	Coupler RX Buffer, containing bit data sent to the CC-Link IE Field Master.
CRYB	Coupler RY Buffer, containing bit data received from the CC-Link IE Field Master.
CRWrB	Coupler RWr Buffer, containing data (different than bit) sent to the CC-Link IE Field Master.
CRWwB	Coupler RWw Buffer, containing data (different than bit) received from the CC-Link IE Field Master.
COS	Coupler Output Submodule, containing the data sent by the PROFINET IO-Controller.
CIS	Coupler Input Submodule, containing the data received by the PROFINET IO-Controller.
CMS	Coupler Management Submodule, containing status information.
IO-C	PROFINET IO-Controller
IO-D	PROFINET IO-Device
SLMP	Seamless Message Protocol

5 Technical concepts

The figure bellow shows the general concept of the coupler.

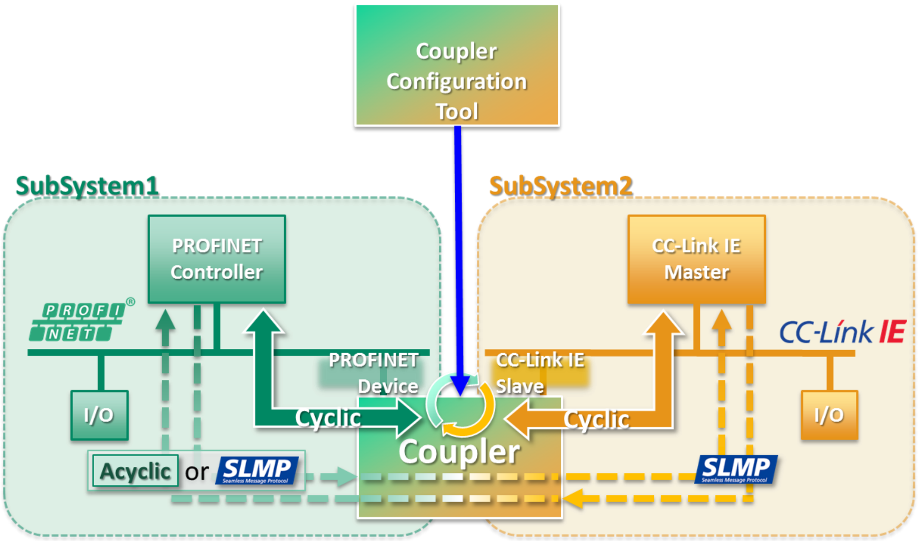

Fig. 4.2-1 Coupler Concept

The role of the Coupler is to allow a PROFINET subsystem to inter operate with CC-Link IE Field subsystem, and allow a CC-Link IE Field subsystem to inter operate with a PROFINET subsystem. In other words, the coupler allow a machine designed on CC-Link IE Field technology to operate with a machine designed with PROFINET technology, and vice versa.

The coupler is connected to the PROFINET subsystem, acting as one of the PROFINET IO-Device of the PROFINET subsystem. On its other interface, the coupler is connected to the CC-Link IE Field subsystem, acting as a CC-Link IE Field Intelligent Device.

The end user might defines the data (name, type, direction) to be exchanged between the two subsystems, and provide this data configuration to the two subsystem Masters, and to the coupler. A Coupler Configuration Tool (CCT) has to be provided by the coupler manufacturer to easier this operation. CCT is going to generate two description files, a GSDML file for usage with the PROFINET subsystem IO-Controller and a CSP+ file for usage with the CC-Link IE Field subsystem Master. In addition, the CCT is going to set some CC-Link IE Field parameters to the Coupler, and download the data configuration to the coupler.

The coupler shall comply to the PROFINET specifications [4] and CC-Link IE Field specifications [1], and PI and CLPA certification test succeeded. Furthermore, it might be understood this paper describes only the specific behaviours of the coupler, all others are ruled by respective protocol specifications (e.g. SLMP Bridge).

As the Coupler is acting as a slave on each subsystem, the coupler is not initiating any communication by itself (e.g. a master shall manage the subsystem and open connections).

The coupler is responsible to transfer data from one side to its other side, both cyclic and acyclic data exchange.

6 Architecture

6.1 Overview

This chapter specifies generic architecture for communication between the CC-Link IE Field Network and PROFINET by a coupler. Mapping and configuration for each of the networks are mainly within the generic architecture in this chapter.

For more mapping and configuration details, please refer to the chapters after chapter 6.

6.2 Basic Structure for Coupler

The structure of the coupler is given in Fig. 6.2-1.

As a basic structure, the coupler has 2 network slave interfaces and a coupling application. The coupler specification for the CC-Link IE Field Network and PROFINET is defined in this document. The coupler shall have the following network slave interfaces.

- CC-Link IE Field Network side: CC-Link IE Field Intelligent Device.
- PROFINET side: PROFINET IO-Device.

The coupler might have additional network interfaces, a slave interface for other networks, or web server interface as a third network slave interface. This is out of scope of this version of the specification.

Network slave interfaces are internally connected by a coupling application. With this function, a Network Master (PROFINET IO-Controller and CC-Link IE Field Master) can exchange data with each other.

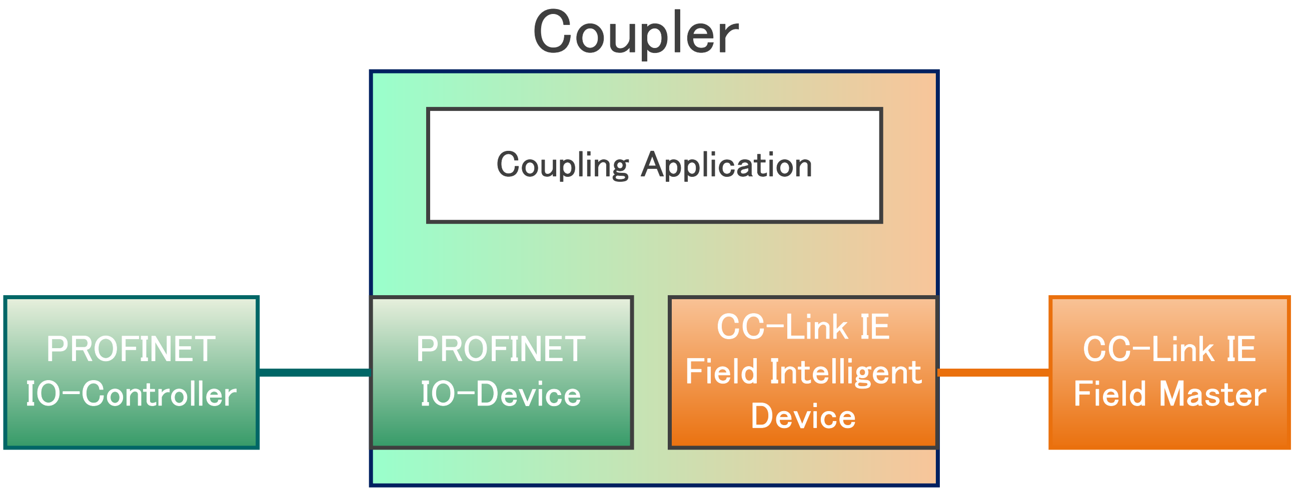

Fig. 6.2-1 Basic Structure for Coupler

6.3 Mapping Model for Coupler

The data mapping model for the coupler is defined for data exchange between both network masters (CC-Link IE Field Network and PROFINET).

As shown in Fig. 6.3-1, the data handled by the coupler is categorized into coupler variables and 5 types of actual data. In principle, each actual data type is internally mapped to the coupler variables and the coupler variable is mapped to cyclic or acyclic memory. It provides accessibility of each data type from both network masters.

Details of actual data in the coupler are as follows. Please refer to figure 6.3.1 for the details of the coupler variables.

- CC-Link IE Field Network Data from/to CC-Link IE Field Master.

  Data on network frames sent or received by the CC-Link IE Field Master like cyclic or acyclic (SLMP) data.
- PROFINET Data from/to PROFINET IO-Controller.

  Data on network frames sent or received by the PROFINET IO-Controller like cyclic or acyclic data.
- Internal CC-Link IE Field Slave Data in the coupler.

  Internal data of the CC-Link IE Field Slave managed by the coupler, like connection status.
- Internal PROFINET IO-Device Data in the coupler.

  Internal data of the PROFINET IO-Device managed by the coupler, like connection status.
- Internal Coupler Data.

  The coupler's own data, like status, parameter or consistency key (CK).

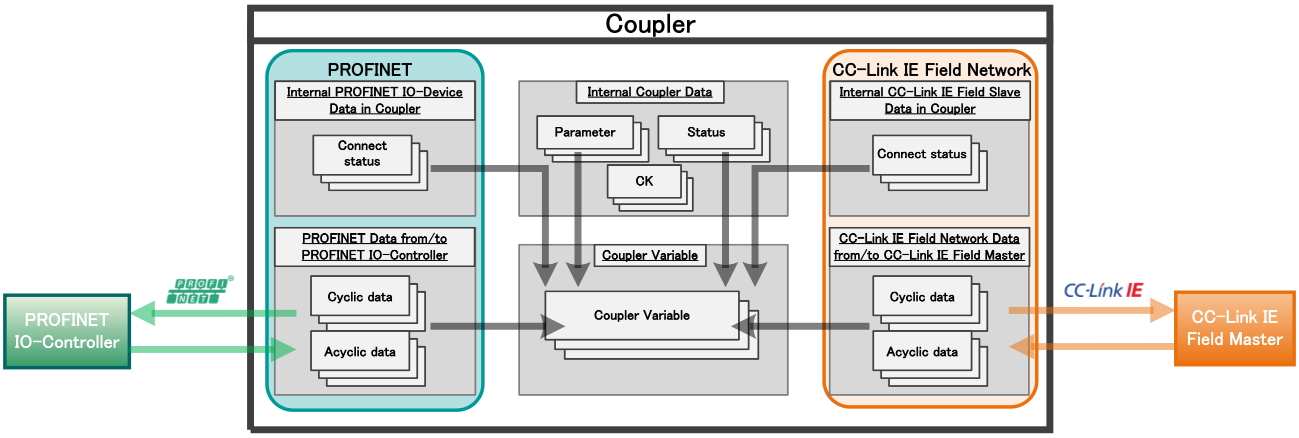

Fig. 6.3-1 Coupler Mapping Model

The definition of coupler variables and the mapping between the coupler variables and the cyclic/acyclic memory shall be configured by CCT.

6.3.1 Details of Coupler Variables

Coupler variables are data which are used by the coupling application. The coupler variable is mapped to actual data (5 types of actual data described as above) in the background. Both network masters can access actual data via coupler variables.

The end user can define coupler variables, the mapping between them and the cyclic/acyclic memory by using the CCT. The coupler vendor can also define coupler variables and the mapping.

Based on the definition of coupler variables and the mapping information, device description files (CSP+ and GSDML files) are generated by the CCT. The device description files provide the interface of the machine/subsystem. It looks like one of network slave devices to the user.

The data format of coupler variables are as shown in Table 6.3-1. Same data format is used for the mapping of cyclic and acyclic.

Table 6.3-1 Data Format of Coupler Variables

Element	Description
Variable name	The variable name shall be unique over the coupler configuration, maximum 255 characters, compliant with IEC61131-3 / programming language.
Variable description	A textual description of the variable should be entered by user.
Variable data type	Data type of coupler variable. Available data types are described in Chapter 7.1.1.
Transport type	Cyclic or Acyclic (Message Interface or User defined Buffer).
Variable direction	Direction for communication of coupler variable. CC-Link IE Field Network to PROFINET or PROFINET to CC-Link IE Field Network.

6.3.1.1 Type of Coupler Variable

According to the definer, a coupler variable is classified into 3 levels as listed in Table 6.3-2.

Table 6.3-2 Type of Coupler Variable

Type	Description
Predefined data	Coupler common data as defined in this specification. The coupler shall use the coupler common data. Example: • Status of coupler and network, like CP_Error on cyclic. • Operation data to achieve coupler function, like CK on cyclic. • Acyclic control registers on cyclic.

Vendor defined data	Vendor specific data that defined by coupler vendor based on their product. Coupler vendors can set up vendor defined data to be transported within the cyclic & acyclic messages. Example: • Vendor specific coupler status. • Information or operation data of coupler.
User defined data	Configurable data used by an end user based on their system. Basically this data is mapped to cyclic or acyclic data. The user can define their coupler variable with the CCT. Hence the coupler vendor shall provide the interface to define this data using the CCT. Example: • Operation status/data of the machine mapped to cyclic or acyclic. • Status of the machine mapped to cyclic or acyclic.

Please refer to chapters 7, 8, & 9 for the details of who is responsible for coupler variable definitions.

6.3.1.2 Mapping to Network Architecture

This section describes the mapping method of each network architecture.

6.3.1.2.1 CC-Link IE Field Network

CC-Link IE Field Network has two types of memory:

- Memory for cyclic communication (RX/RY/RWr/RWw).
- Memory for acyclic communication.

CC-Link IE Field Network has several methods of acyclic communication. SLMP shall be used in this specification. Please refer to chapters 7, 8 & 9 for details.

6.3.1.3 PROFINET

PROFINET IO-Device could have a modular architecture. The architecture shown in Table 6.3-3 is used for this specification.

Please refer to chapters 7, 8 & 9 for details.

Table 6.3-3 Mapping Area of PROFINET

Slot	Subslot	Description
0	-	DAP
X	1	Coupler Output Submodule. Submodule managing PROFINET outputs. (Cyclic data from PROFINET IO-Controller to coupler).

	2	Coupler Input Submodule. Submodule managing PROFINET inputs. (Cyclic data from coupler to PROFINET IO-Controller).
	5	Coupler Management Submodule. Management Submodule, reflecting coupler status to PROFINET.

X: Shall be slot 1 for PROFINET non modular IO-Device.

6.4 Configuration

6.4.1 Configuration Procedure

The CCT is used to configure the coupler. The CCT can define the coupler variables and the mapping between the coupler variables and the cyclic/acyclic memory (Please refer to section 6.3.1 for details).

The CCT uses coupler variables for the purpose of generating CSP+ and GSDML files.Those files can be imported from configuration tools of PROFINET and the CC-Link IE Field Network. Therefore, the CCT can inform configuration information of the coupler to both configuration tools.

Configuration of the coupler is composed of the following steps:

At STEP1, an end user defines coupler variables, a mapping between the coupler variables and the cyclic/acyclic memory to be handled by the CCT.

At STEP2, the end user downloads the coupler variables and the mapping definitions defined at step 1 and a Consistency Key (CK) to the coupler, as well as generated GSDML and CSP+ files.

At STEP3 (*1), the end user performs the commissioning of the CC-Link IE Field Master based on the generated CSP+ file.

At STEP4 (*1), the end user performs the commissioning of the PROFINET IO-Controller based on the GSDML file.

*1: The end user may configure either early.

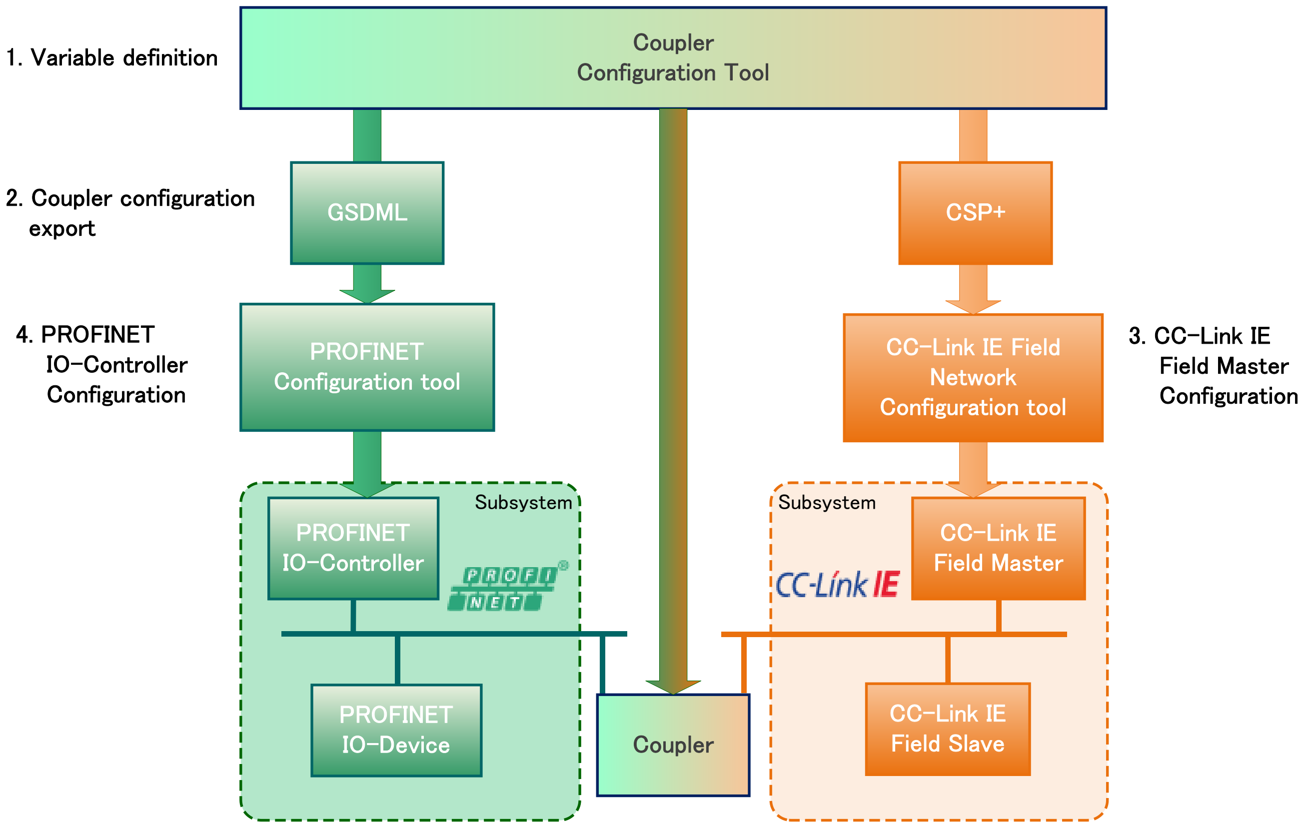

Fig. 6.4-1 Configuration Procedure

6.4.2 Device Description File

The CCT can export two types of device description files, CSP+ and GSDML. The CSP+ file is imported into the CC-Link IE Field Network configuration tool and the GSDML file is imported into the PROFINET configuration tool. The configuration of the coupler can be downloaded to each configuration tool by importing device description files.

The CCT uses coupler variables and the mapping between the coupler variables and the cyclic/acyclic memory for the purpose of generating device description files. Note that CSP+ and GSDML have different formats. However, both files have areas to describe cyclic data mapping, acyclic communication procedures and parameters. The CCT can export both device description files by converting the information in a suitable manner.

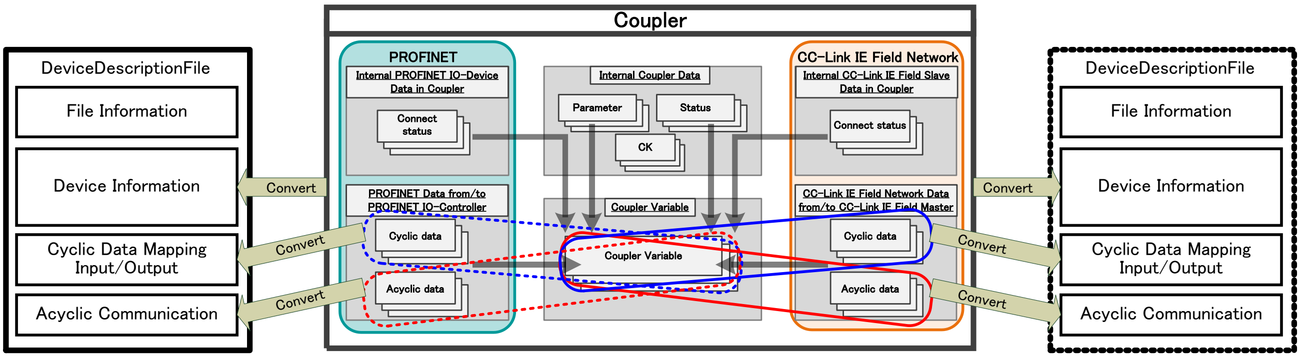

Fig. 6.4-2 Device Description Files

6.5 Cyclic Communication

6.5.1 Use Case

The coupler's cyclic communication function is used by one side's network master to control I/O data exchange with other side's subsystem. (See Fig. 6.5-1).

One side's network master can control the other side's subsystem by periodically sending or receiving the I/O data to/from the coupler.

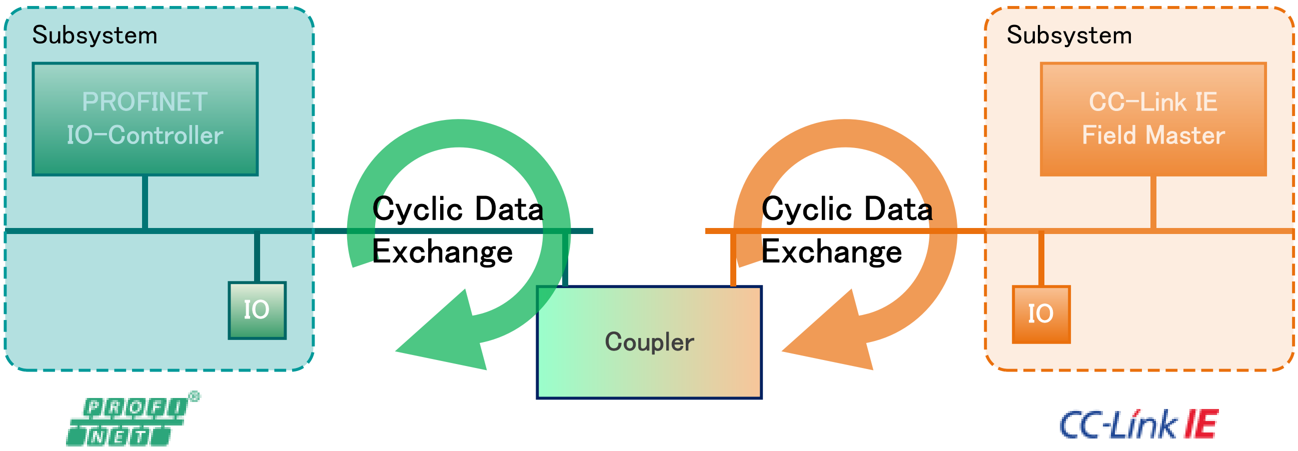

Fig. 6.5-1 Cyclic Use Case

6.6 Acyclic Communication

6.6.1 Use Case

The coupler's acyclic communication function is used for the following use cases.

- Monitor.
- Parameter Setting.

6.6.1.1 Monitor

In this use case, an end user can use the coupler's acyclic communication function to monitor the other side's subsystem information by SCADA or an HMI. (See Fig. 6.6-1)

SCADA or HMI can use the acyclic communication read request to obtain parameter settings, error data, or statistics from other side's subsystem.

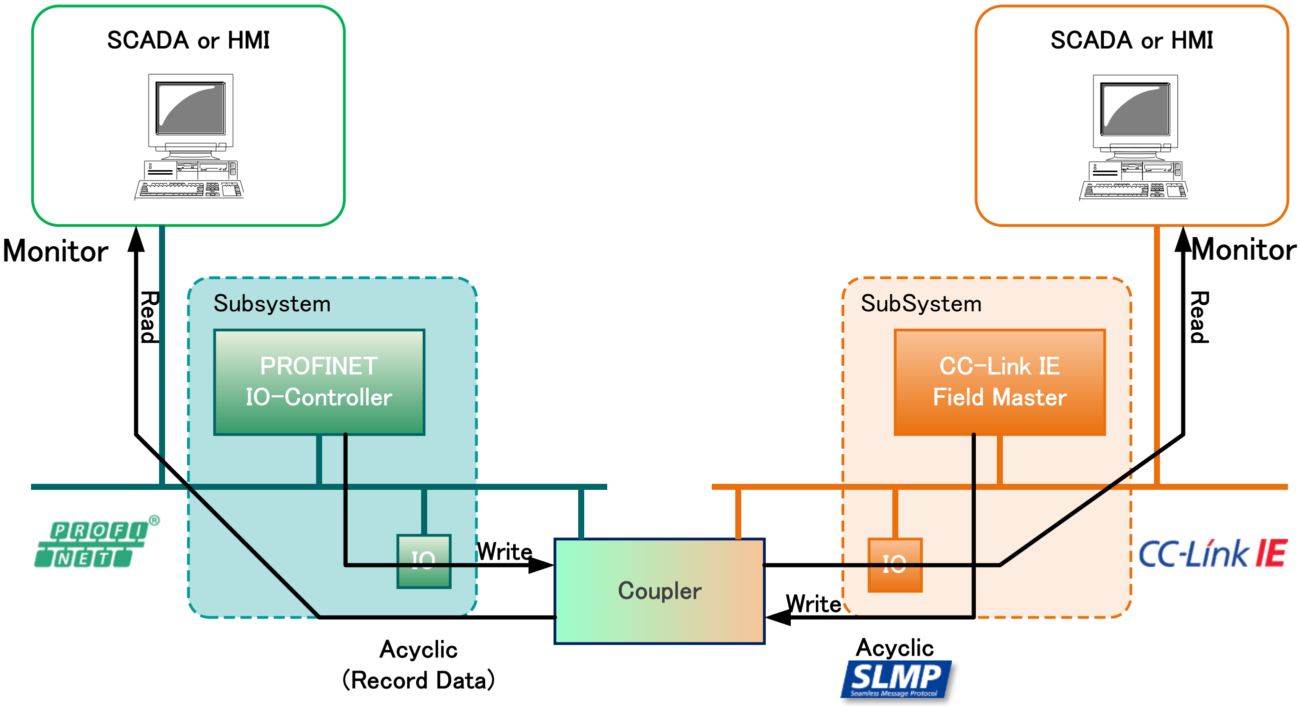

Fig. 6.6-1 Acyclic Use Case (Monitoring)

6.6.1.2 Parameter Setting

In this use case, an end user can use the coupler's acyclic communication function to set up the parameters of the other side's subsystem via the network master, SCADA or an HMI. (See Fig. 6.6-2)

The network master, SCADA or HMI can send acyclic communication data to the coupler, which stores the setting parameters of the other side's subsystem.

The other side's system or subsystem should change its parameters according to the above acyclic communication data.

For example, if the above acyclic communication data includes an operation mode parameter for the other side's subsystem, the other side's subsystem should change its operation mode as the parameter indicates.

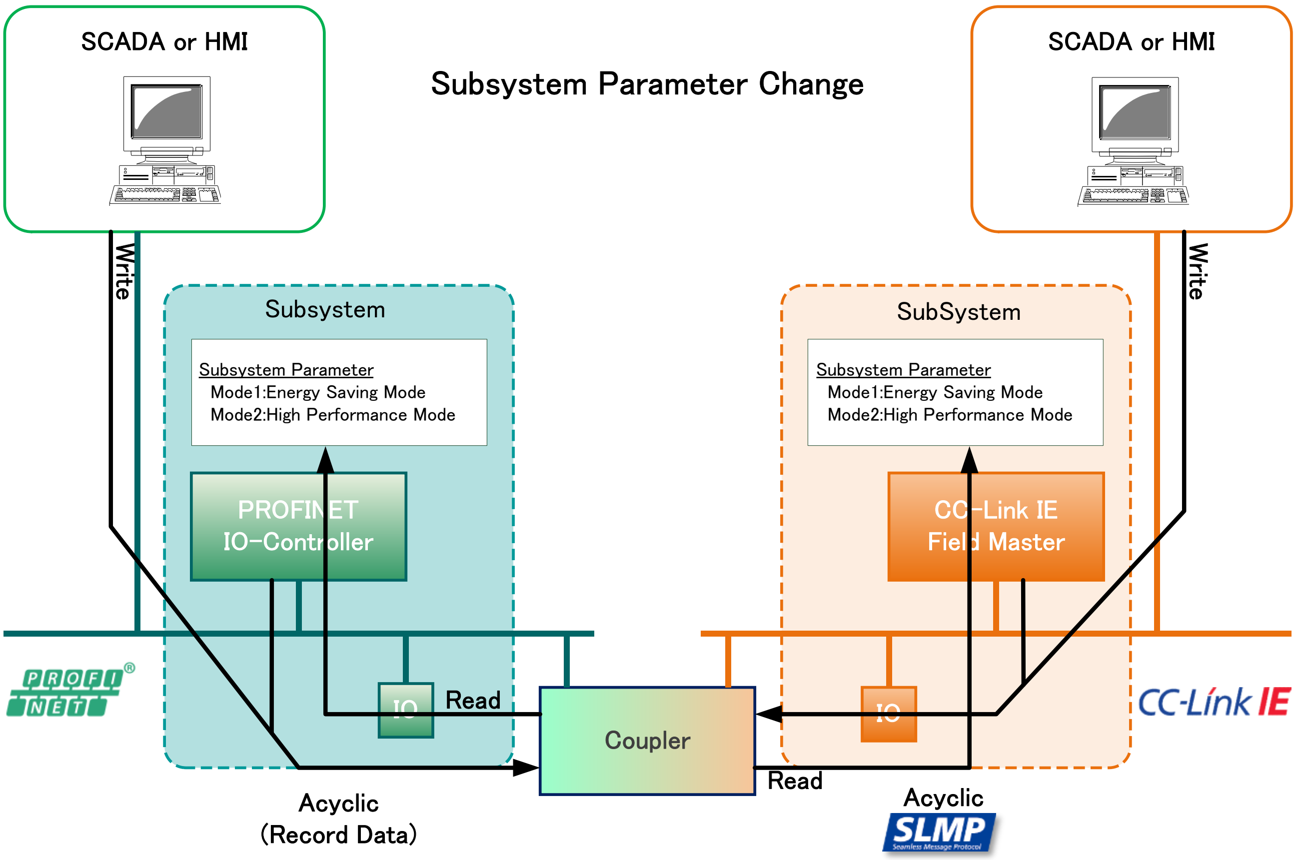

Fig. 6.6-2 Acyclic Use Case (Parameter Setting)

6.7 Diagnostics

6.7.1 Use Case

The diagnostics function of the coupler is used for the following use cases.

- Collecting Diagnostic Data.
- Collecting User Defined Status.

6.7.1.1 Collecting Diagnostic Data

The coupler's diagnostics function is used by one side's network master, SCADA or HMI to get the error data of the coupler, or the status data of the other side's network master.

The coupler error data is mapped to the status information defined by the network protocol, where the status information is stored in a form such as a MyStatus frame, or IOXS. Thus an end user can obtain the coupler error data without a specified configuration.

The patterns of the coupler error data can be as follows:

- A disconnection between the network master and coupler.
- An error occurs when processing the coupler application.

The other side's network master status information is mapped to the cyclic communication data, where one side's master, SCADA or HMI can obtain the status information by cyclic communication.

The status information of a network master can be used for indicating the RUN/STOP status or waiting for acyclic communication status etc.

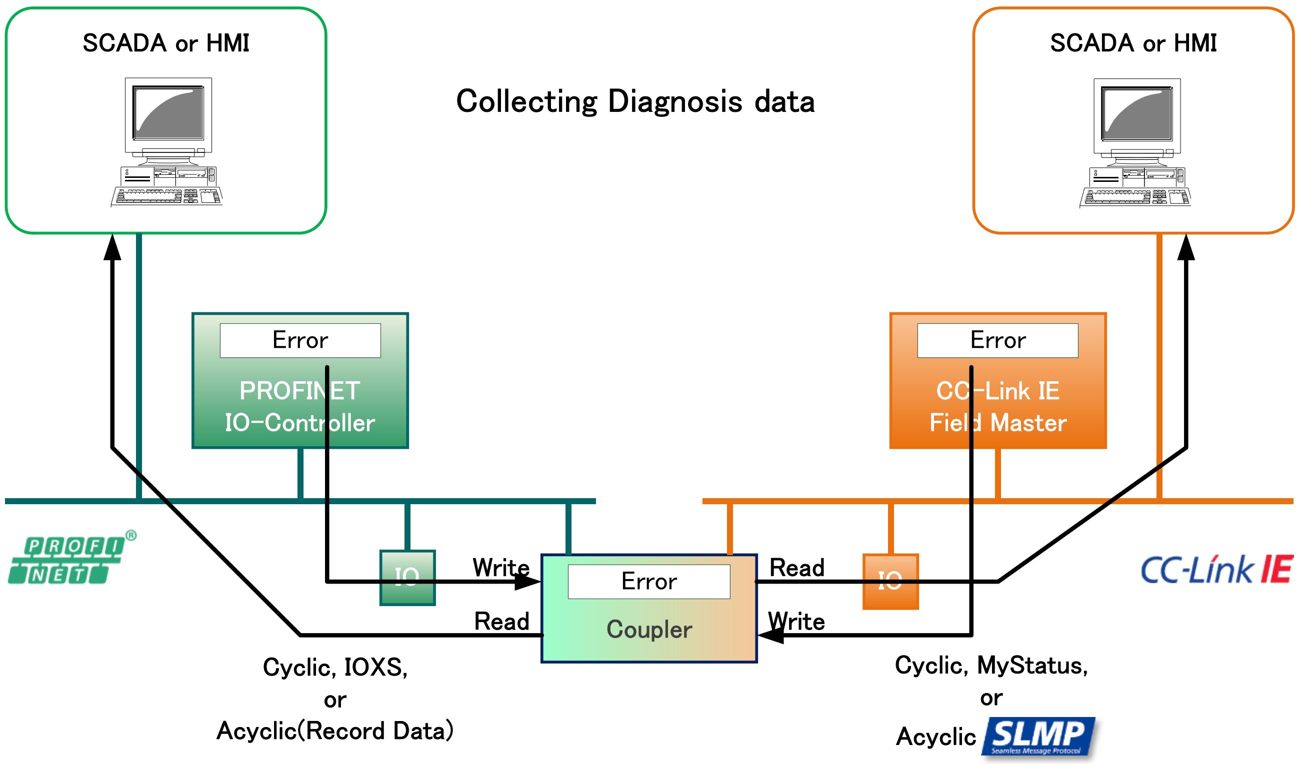

Fig. 6.7-1 Diagnostic Use Case (Collecting Diagnostic Data)

6.7.1.2 Collecting User Defined Status

In addition to the above, the network master can get unique diagnostic data defined by the end user by mapping to cyclic/acyclic data.

For example, the network master can get diagnostic data of a robot subsystem if the end user defines “Robot Status” or “Robot System Status” etc. as diagnostic data, and transmits the diagnostic data to the coupler.

This use case is only to map the user defined diagnostic data to cyclic/acyclic communication.

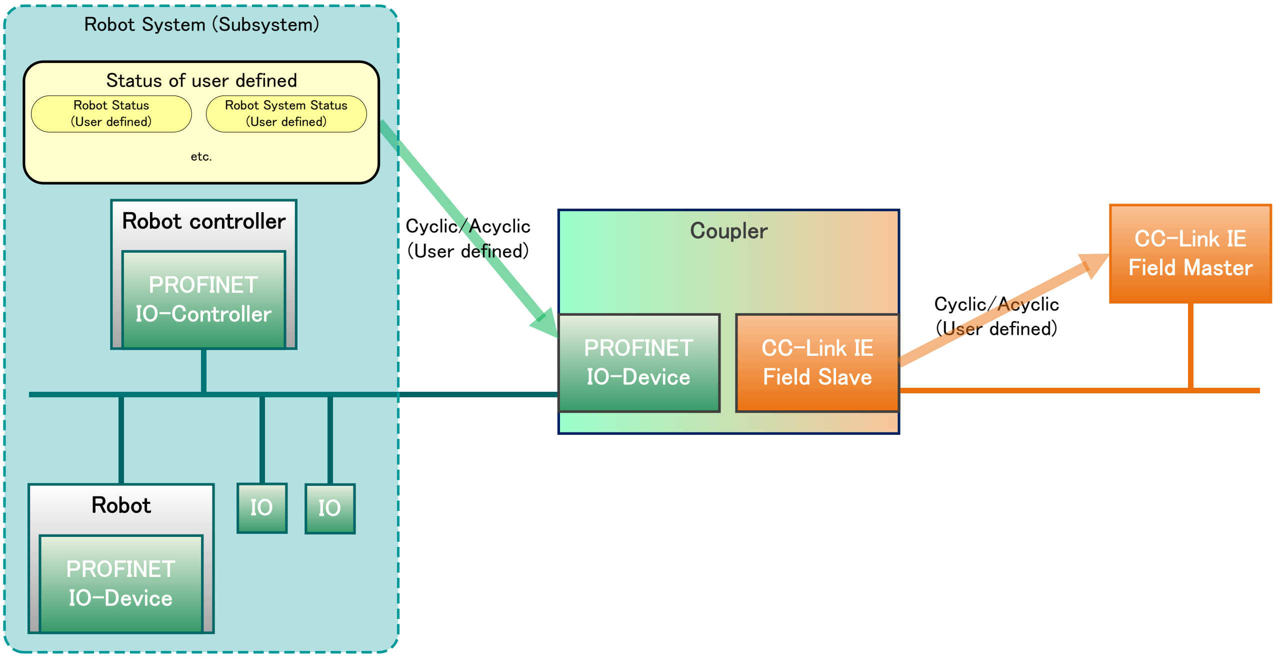

Fig. 6.7-2 Diagnostic Use Case (Collecting User Defined Status: CC-Link IE Field Network Side)

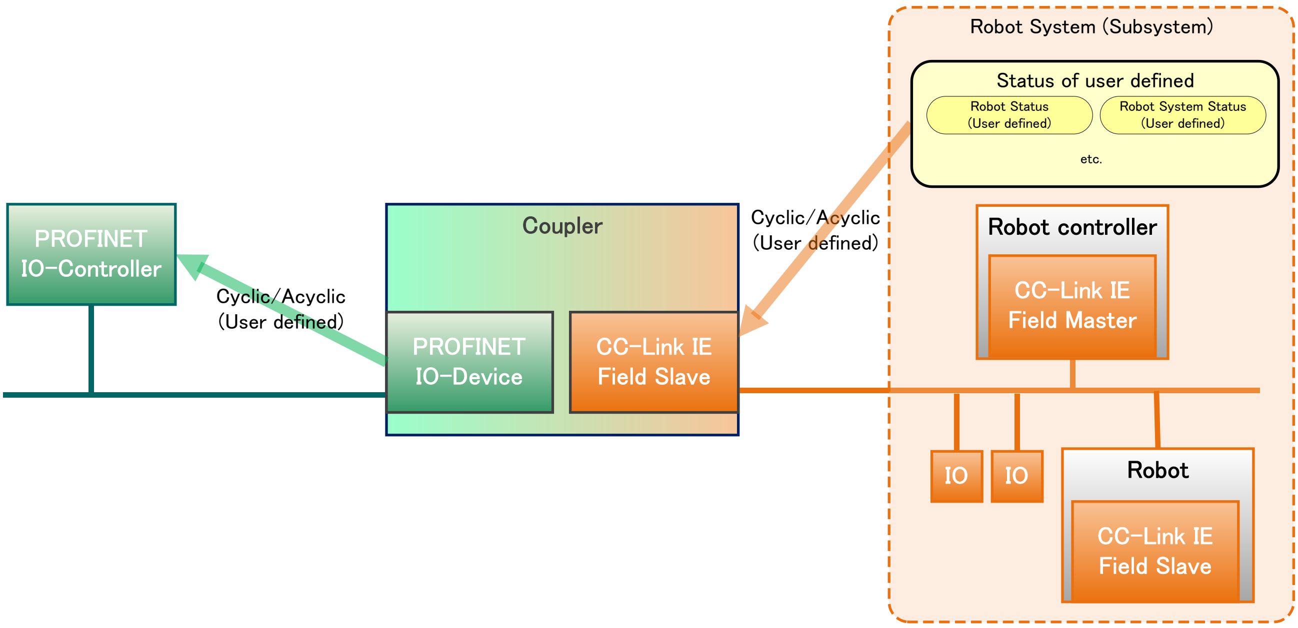

Fig. 6.7-3 Diagnostic Use Case (Collecting User Defined Status: PROFINET Side)

6.8 Implementation Specification for Coupler

Table 6.8-1 describes the coupler’s implementation specification.

Table 6.8-1 Implementation Specification for Coupler

No.	Function	Implementation
1	Cyclic communication	Mandatory
2	Message Interface (Acyclic communication)(*1)	Optional(*2)
3	User Defined Buffer (Acyclic communication) (*1)	Optional(*2)
4	Consistency Key (CK)	Mandatory

*1: Please refer to chapter 9.

*2: Optional, but the coupler shall support at least one of these.

7 Configuration

This chapter describes the different steps to be operated by the end user in order to get the Coupler operating. The software (Coupler Configuration Tool –CCT), provided by the coupler manufacturer, generates two description files, a GSDML and a CSP+ files, and allows download of the configuration to the coupler.

The coupler user is responsible to use and configure the respective Masters using the generated description files. Based on the different levels of support regarding the slave description files, the integration of the coupler in Master configuration could lead to partial usage of description files contained information. Then, manual entry inside the Master configuration tool might be required.

In addition, introduction of description files into the master configuration tools remains under user responsibility, and thus could lead to mixing different file versions. As manual operations take place, a mechanism checking the two network configuration are based on same file versions, is required.

The protocol used to download the coupler configuration to the coupler is not under the scope of this specification.

7.1 End user configuration steps

The table below is listing the steps necessary to be operated by the end user with the CCT in order to configure the coupler. Basic operation regarding the address setting for each fieldbus is not explained within this document, but have to take place before to ensure communication. This is the case for example for PROFIET IO-Device Station name, network number and station number of CC-Link IE Field Network.

Table 7.1-1 End user configuration steps

Action Step	Name	Description
1	Variable Definition	End user defines the variables to be handled by the coupling application. The mapping of variables inside the buffer belong to the CCT, thus being or not configurable by the end user.
2	Coupler configuration export	End user requests export of the coupler configuration, leading to download of the coupler configuration inside the coupler and generation of GSDML and CSP+ files.
3	CC-Link IE Field Master configuration	End user operates the commissioning of the CC-Link IE Field Master. Note: Error could occur as old file could be used by operator mistake.
4	PROFINET IO-Controller Configuration	End user operates the commissioning of the PROFINET IO-Controller based on generated GSDML. Note: Error could in any cases occur as old file could be used by operator mistake.

The end user could check all operations have been successfully operated by running diagnosis.

As integrating GSDML and CSP+ files into the network engineering tools are being a manual operation, errors like mixing several versions of the files have to be considered, and detected. In order to ensure the consistency between the description files, a consistency key (CK) have to be generated by the CCT.

7.1.1 End user configuration: Step 1: Variable definition

End user is requested to define the variables. CCT shall allow user to define variable in the following way:

- Variable name: The variable name shall be unique over the coupler configuration, maximal 255 characters, compliant with IEC61131-3 / programming language.
- Variable description: A textual description of the variable should be entered by user.
- Variable data type:

  The following table defines the different data types being handled by the coupler.

Table 7.1-2 Variable definition

DataTypes	Description
Integer8	A 8 bit signed integer value from -128 to 127
Integer16	A 16 bit signed integer value from -32 768 to 32 767
Integer32	A 32 bit signed integer value from -2 147 483 648 to 2 147 483 647
Unsigned8	A 8 bit unsigned integer value from 0 to 256
Unsigned16	A 16 bit signed integer value from 0 to 65535
Unsigned32	A 32 bit unsigned integer value from 0 to 2^{32}-1
Float32	Single-precision floating point precision format
BOOL	A Boolean value as TRUE = 1 and FALSE = 0
Bit	A bit being 0 or 1.

Note: all other formats are not taken in account for this specification version.

- Transport type:
  - Cyclic.
  - Acyclic (Message Interface or User defined Buffer).
- Variable direction:
  - CC-Link IE Field Network to PROFINET.
  - PROFINET to CC-Link IE Field Network.

7.1.2 End user configuration: Step 2: Coupler configuration export

This step consists of the download to the coupler of the variable configuration, as well as the generation of the CSP+ and GSDML files. At this step, the CCT shall generate the consistency key (CK) contained inside GSDML and CSP+. The CCT shall also permit the end user to view this key (CK). The CK shall be transferred to the coupler by the CCT.

7.1.3 End user configuration: Step 3: CC Link IE Field Master configuration

Please refers to the CC-Link IE Field Master engineering tool about how to configure a CC Link IE Field communication to the coupler by using the CSP+ file, or entering manually the communication parameters.

7.1.4 End user configuration: Step 4: PROFINET IO-Controller configuration

Please refers to the PROFINET IO-Controller engineering tool in order to configure a PROFINET communication to the coupler by using the GSDML file.

7.2 Consistency Key (CK)

The consistency key (CK) shall be generated by CCT while exporting the coupler description files. The CK shall consist of one unsigned 16 bits value randomly generated, each time the description files are modified.

The CK shall be used to generate SubmoduleIdentNumber in the GSDML file. The submoduleIdentNumbers have to be provided by the CCT to the coupler application.

The CK shall be transported from the CC-Link IE Field aster to the coupler within the cyclic data, and thus being part of the CSP+ file.

The coupler application shall check the CK provided by PROFINET IO-Controller, CCT and CC-Link IE Field Master, if a mismatch is detected, the “CK Mismatch” (see section Coupler Error Code) error code shall be reported, and CP_Error bit set.

7.3 Generate GSDML/CSP+

The CCT shall generate description files, based on end user choices. The description files would contains the variable definition, location in transport buffer and the CK.

7.3.1 CSP+

CSP+ file is describing the CC-Link IE Field device, and is used by the CC-Link IE Field Master engineering tool in order to commission the device. Please refer to the CSP+ specification for details [3]. This specification does only specify the CSP+ elements that are specific to a PROFINET/ CC-Link IE Field coupler. All the other element of the CSP+ coupler file might respect to CSP+ specification.

7.3.1.1 CK transport

The consistency key shall be put inside the CSP+ file inside **DeviceConfigurationID** in the **DEVICE_INFO** part in the **DEVICE** section. The Items of **DeviceConfigurationID** are shown in the following table.

Table 7.3-1 Attributes & Items of DeviceConfigurationID

Attributes and Items	Description
LABEL	Shall be set to "DeviceConfigurationID".
NAME	Shall be set to "DeviceConfigurationID".
DATATYPE	Shall be set to "STRING(32)".
DATA	Set 16bits Consistency Key as binary numbers.

RWw0 register shall be used to transport the CK allowing to compare the CK from CC-Link IE Field Master and the CK in the Coupler, given by CCT. The way to describe register assignment in the CSP+ is described in the following section.

7.3.1.2 Cyclic and Acyclic CC-Link IE Data types

The table below lists the different **DataType** to be used in function of the coupler variable type:

Table 7.3-2 CSP+ data types

Coupler Variable Types	CSP+ data type
Integer8	INT8
Integer16	INT16
Integer32	INT32
Unsigned8	UINT8
Unsigned16	UINT16
Unsigned32	UINT32
Float32	REAL
BOOL	BOOL
Bit	Bit

All BOOL and bit coupler variable are being transported by CRXB and CRYB according their direction.

7.3.1.3 Data alignment in buffers

In CRWwB and in CRWrB, in order to warranty the alignment of data on a word boundary, CCT shall add for each variable of type INT8 or UINT8 a padding byte.

7.3.1.4 Length of buffers

Coming from CC-Link IE Field Network specification [1], buffer length shall be a multiple of 4. CRXB, CRYB, CRWwB and CRWrB data length (including padding elements) shall be placed in the **COMM_IF_INFO part** as specified in the table below.

Table 7.3-3 CSP+ buffer data size

Coupler Buffer	CSP+ data length field
CRXB	RXSize
CRYB	RYSize
CRWrB	RWrSize
CRWwB	RWwSize

As on CC-Link IE Field bit data are being transported in different buffer than other data types, and on PROFINET, there is only one container for all data types, CCT shall check the total size of data is compatible with maximum cycle data to be transported by PROFINET.

7.3.1.5 CRXB & CRWrB (CC-Link IE Field Master cyclic input data)

The CC-Link IE Field Master input data, transported by CRXB and CRWrB, have to be described into the CSP+ file into the part **BLOCK_INPUT**.

For each CC-Link IE Field Master input variable a **blockInputMember** element shall be generated, and shall contain the attributes and items defined in the following table:

Table 7.3-4 Attributes and items of blockInputMember

Attributes and Items	Description
LABEL	Used to identify the element. Limited to 32 characters (See CSP+ specification for details [3]). The LABEL is used when the element is referenced from other element with the ref attributes.
NAME	It shall contain the variable name. characters (See CSP+ specification for details [3])
DATATYPE	See 7.3.1.2

Optionally, the field "**COMMENT**" should be used to transport the description of the variable.

For each CC-Link IE Field Master input variable a commIfOutputMember Element shall be generated, and shall contain the attributes and items defined in the following table:

Table 7.3-5 Attributes and Items of CommIfOutputMember

Attributes and Items	Description
LABEL	Used to identify the element. Shall be equal to **LABEL** within **blockInputMember**.
NAME	It shall contain the variable name. Shall be equal to **NAME** within **blockInputMember**.
DATATYPE	Shall be equal to **DATATYPE** within **blockInputMember**.
ACCESS	Shall be set to "RF".
ASSIGN	RX for bit area, address (Hex) to follow. RWw for register area, address (Hex) to follow. Note: For a variable in register area address 10, use RWwA Note: Case is important. See CSP+ specification [3]
REF	Shall be set to "BlockSection.BlockInput."X Where X is the **LABEL** of a blocInputMember

7.3.1.6 CRYB & CRWwB (CC-Link IE Field cyclic output data)

The CC-Link IE Field output data, transported by CRYB and CRWwB, have to be described into the CSP+ file into the part **BLOCK_OUTPUT**.

For each CC-Link IE Field variable a **blockOutputMember** element shall be generated, and shall contain the attributes and items defined in the following table:

Table 7.3-6 attributes and Items of blockOutputMember

Attribute and Items	Description
LABEL	Used to identify the element. Limited to 32 characters (See CSP+ specification for details [3])
NAME	It shall contain the variable name. characters (See CSP+ specification for details [3])
DATATYPE	See 7.3.1.2

Optionally, the field "**COMMENT**" should be used to transport the description of the variable.

For each CC-Link IE Field output variable a **commIfInputMember** element shall be generated, and shall contain the attributes and items defined in the following table:

Table 7.3-7 Attributes & items of commIfInputMember

Attribute and Items	Description
LABEL	Used to identify the element. Shall be equal to **LABEL** within **blockOutputMember**.
NAME	It shall contain the variable name. Shall be equal to **NAME** within **blockOutputMember**.
DATATYPE	Shall be equal to **DATATYPE** within **blockOutputMember**.
ACCESS	Shall be set to "RF".
ASSIGN	RY for bit area, address (Hex) to follow. RWr for register area, address (Hex) to follow. Note: For a variable in register area address 11, use RWrB Note: Case is important. See CSP+ specification [3]
REF	Shall be set to "BlockSection.BlockOutput."X Where X is the **LABEL** of a **blockOutputMember**

7.3.1.7 Acyclic Communication

The important parts for CC-Link IE Acyclic communication are **MESSAGE**, **COMM_IF_PARAMETER**, and **BLOCK_PARAMETER**.

The data which can be accessed with acyclic communication is described in the **blockParameterMember** element in **BLOCK_PARAMETER** part in the **BLOCK** section. The **blockParameterMember** element is referenced from the **commIfParameterMember** element in the **COMM_IF_PARAMETER** part in the **COMM_IF** section. The **commIfParameterMember** element is referenced from the messageElement in the **MESSAGE** part in the **COMM_IF** section.

The attributes and items of those elements are shown in the following table.

Table 7.3-6 Attributes and Items of messageElement

Attribute and Items	Description
LABEL	It is the identifier of elements. The prefix "SLMP" shall be described.
NAME	It is User-defined variable name.
TARGET	The reference to COMM_IF_PARAMETER part. The parameter which is included in referenced part is the target of the SLMP command.
MESSAGE_TYPE	"PARAMETER" is described.
REQUEST_TYPE	Request type is described (e.g. "reReqMT_Binary").
REQUEST_DATA	It is request data. The Following description is sample of value of this element. In the case of reading parameter e.g. <0x0613><0x0000><$(ASSIGN)> In the case of writing parameter e.g. <0x1613><0x0000><$(ASSIGN)><$(VALUE)> The $(ASSIGN) is the value of ASSIGN element in the TARGET part. The $(ASSIGN) is the parameter value to write.
REQUEST_DATATYPE	It is the datatype of the request data. The Following description is sample of value of this element. In the case of reading parameter: e.g. <WORD><WORD><DWORD><WORD> In the case of writing parameter e.g. <WORD><WORD><DWORD><WORD><$(DATATYPE)> The $(DATATYPE) is the value of DATATYPE element in the TARGET part.
RESPONSE_TYPE	It is Response type is described (e.g. "reResMT_Binary").

RESPONSE_DATA	It is the response data. The Following description is sample of value of this element. In the case of reading parameter. e.g. <$(VALUE)> In the case of writing parameter. Definition is not needed.
RESPONSE_DATATYPE	It is the datatype of the response data. The Following description is sample of value of this element. In the case of reading parameter. e.g. <$(DATATYPE)> In the case of writing parameter. Definition is not needed.
ERR_TYPE	It is the definition of the error code. The Following description is sample of value of this element. In the case of reading parameter. e.g. "rdERRMT_Binary" is described. In the case of reading parameter. e.g. "wrERRMT_Binary" is described.

Table 7.3-8 Attributes and Items of commIfParameterMember

Attributes & Items	Description
LABEL	This is the identifier of elements, it shall be equal to the **LABEL** in the referencing part.
NAME	It shall be equal to the **NAME** in the referencing part. The content is coming from the CCT as it is the User-defined variable name.
DATATYPE	The Datatype of variable. Please refer Cyclic and Acyclic CC-Link IE Data types about usable datatype in this part. Shall be equal to the **DATATYPE** in the referencing part.
ASSIGN	The SLMP command and data are described. e.g. "<StartAddr><wl>” The StartAddr is the memory start address of the parameter. In the case of Message interface, consistently same address is described. In the case of User Defined Buffer, the address is different from each other. The wl is word length of the memory area for the parameter.
REF	Shall be set to “BlocSec.BlockParameter.”X, Where X is the LABEL of a blockParameter.
COMMENT	The Optional textual description of User-defined variable.

Table 7.3-9 Attributes and Items of blockParameterMember

Attributes & Items	Description
LABEL	It is the identifier of elements.
NAME	It is User-defined variable name.
DATATYPE	The Datatype of variable. Please refer Cyclic and Acyclic CC-Link IE Data types about usable datatype in this part. Shall be equal to the **DATATYPE** in the referencing part.
ACCESS	It is the access attribute of the element. In case of variables. In the case of Message interface, "R"(Read) or "W"(Write) is described depending on setting of the Variable direction. In the case of User Defined Buffer, "RW"(Read and Write) is described.
COMMENT	The Optional textual description of User-defined variable.

7.3.2 GSDML

GSDML file is describing the IO-Device, used by the PROFINET IO-Controller engineering tools for commissioning IO-Device. Please refer to GSDML specification for details [5]. This specification does only specify the GSDML elements that are specific to a PROFINET/ CC-Link IE Field coupler. All the other element of the coupler GSDML file might respect to GSDML specification.

As submodules are contained in a module, the coupler shall expose a module located in slot 1, for non modular system (see Cyclic communication on PROFINET). ModuleID shall be set to 0x19720710.

Submodule are described in a list of submodule, inside a **SubmoduleItemList** tag.

7.3.2.1 COS submodule

The IO-C output data are transported within the COS. Each IO-C output variable are transported from the PROFINET IO-Controller to the coupler by the COS. IO-C output variable shall be transported in big endian format, packed all together. Moving the COS variable to the CC-Link IE data format shall be operated by the coupler application.

The CCT shall generate a GSDML with by setting the COS's **submoduleIdenNumber** as follow

Table 7.3-10 COS Submodule Ident Number

COS SubmoduleIdentNumber Bits	Name	Description
0 to 14	User specific	CCT could set whatever value there.
15	IO-C output	Set to 1
16 to 31	Consistency Key	CCT shall copy the CK value (big endian coding)

At PROFINET connection, as the coupler application have plugged into the PROFINET IO-Device the COS with the SubmoduleIdentNumber provided by the CCT, if a different submodule (not the latest GSDML file in used on PROFINET IO-Controller engineering tool) is detected by the PROFINET IO-Device, a modulediffbloc information would be generated.

As the COS has been fixed in the subslot 1, the tag **FixedInSubslots**=”1” shall be used.

The COS data shall be described into a GSDML tag <**OUTPUT**>.

7.3.2.2 CIS submodule

The IO-C input data are transported within the CIS. Each IO-C input variables are transported from the coupler to PROFINET IO-Controller by the CIS. IO-C input variables shall be transported in big endian format, packed all together. Moving the CC-Link IE data to the CIS data format shall be operated by the coupler application.

The CCT shall generate a GSDML with by setting the CIS’s **submoduleIdenNumber** as follow:

Table 7.3-11 CIS Submodule Ident Number

CIS SubmoduleIdentNumber Bits	Name	Description
0 to 14	User specific	CCT could set whatever value there.
15	Intbound	Clear (Set to 0)
16 to 31	Consistency Key	CCT shall copy the CT value (big endian coding)

At PROFINET connection, as the coupler application have plugged into the PROFINET IO-Device the CIS with the **SubmoduleIdentNumber** provided by the CCT, if a different submodule (not the latest GSDML file in used on PROFINET IO-Controller engineering tool) is detected by the PROFINET Io-Device, a modulediffbloc information would be generated.

As the CIS has been fixed in the subslot 2, the tag **FixedInSubslots**=”2” shall be used.

The CIS data shall be described into a GSDML tag <**INPUT**>.

7.3.2.3 Cyclic Data in COS and CIS

In a PROFINET submodule, the data content is described by listing the different data items inside the **<INPUT>** or **<OUTPUT>** tag using the DataItem element.

The table below lists the different DataType to be used in function of the coupler variable type:

Table 7.3-12 GSDML data type

Coupler variable Types	GSDML data type
Integer8	“Integer8”
Integer16	“Integer16”
Integer32	“Integer32”
Unsigned8	“Unsigned8”
Unsigned16	“Unsigned16”
Unsigned32	“Unsigned32”
Float32	“Float32”

BOOL	"Unsigned8" with UseAsBits attribute set to true.*
Bit	"Unsigned8" with UseAsBits attribute set to true.*

Each DataItem (excluded bit) shall refer to a TextID. This TextID refers to a Text element in the PrimaryLanguage element where the variable name is set as value.

Bit variables shall UseAsBits set to true, and each defined end user bit shall lead to an element BitDataItem, with its own BitOffset and TextID.

7.3.2.4 Acyclic Data in COS and CIS

In a PROFINET submodule, the data to be transferred acyclicaly (by means of PROFINET Read/Write commands), have to be described within a record. This is described inside the GSDML file by using the tag <**ParameterRecordDataItem**>. Datatype are same as for cyclic Data.

7.3.2.5 CMS

The CMS transports the status information back to the PROFINET IO-Controller. CMS **SubmoduleIdentNumber** shall be set to 0x19710205. As the CMS has been fixed in the subslot 5, the tag **FixedInSubslots**="5" shall be used.

8 Cyclic

This chapter describes how cyclic data are mapped by the coupling application (CA) from/to CC-Link IE Field from/to PROFINET, as well as how the data are transported.

8.1 Cyclic communication on CC-Link IE Field

On CC-Link IE, all cyclic communication take place on four buffer level. The figure below shows the buffer structure of the coupler CC-Link IE Field Intelligent Device.

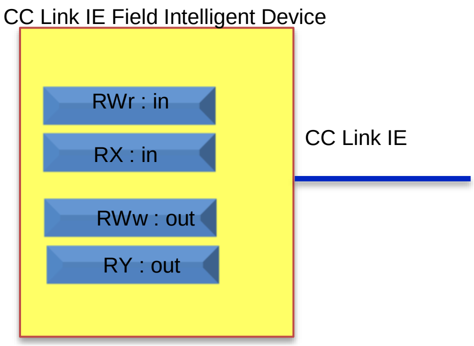

Fig. 8.1-1 Cyclic buffer in CC-Link IE Field

The table below lists them all with their description.

Table 8.1-1 CC-Link IE Field Buffers

Name	Description
Coupler RX buffer	RX buffer manages the BOOL data sent to the CC-Link IE Field Master by the coupler. RX address RX0 to RX7 are reserved for internal coupler status bits.
Coupler RWr buffer	RWr buffer manages the data sent to the CC-Link IE Field Master by the coupler, not being bits.
Coupler RY buffer	RY buffer manages the BOOL data sent by the CC-Link IE Field Master to the coupler.
Coupler RWw buffer	RWw buffer manages the data sent by the CC-Link IE Field Master to the coupler, not being bits.

The CC-Link IE Field input data transported by CRXB and CRWrB are transported on PROFINET side by COS. The maximum amount of data being carried by CRXB plus CRWrB is driven by the maximal capacity of COS.

- Size of CRXB + Size of CRWrB <= size of COS

The CC-Link IE Field output data transported by CRYB and CRWwB are transported on PROFINET side by CIS. The maximum amount of data being carried by CRYB plus CRWwB is driven by the maximal capacity of CIS.

- Size of CRYB + Size of CRWwB <= size of CIS

Data are transported in little endian format, with alignment on word boundaries. The coupler application shall ensure the correct endianness while copying variable from/to PROFINET to/from CC-Link IE Field.

8.1.1 Coupler RX buffer (CRXB)

The CRXB contains the BOOL CC-Link IE Field input data defined by the end user. Data are located from address RX8 of the CRXB.

Starting at address RX0, the following bits shall be available:

Table 8.1-2 RX Buffer status bit

Bit Address	Name	Description
RX0	CP_Error	TRUE: An error has been detected by the coupler application. A detailed error code is provided. FALSE: No error detected.
RX1	Reserved	Shall be clear
RX2	PN_Connected	TRUE: A PROFINET IO-Controller is connected to the coupler. FALSE: No PROFINET IO-Controller connected to the coupler.
RX3	PN_DataStatus	TRUE: Connected PROFINET IO-Controller is in RUN mode. (APDU_Status) FALSE: Connected PROFINET IO-Controller is in STOP mode. (APDU_Status)
RX4	PN_ACYC_MSG	TRUE: An acyclic message is waiting (PROFINET IO-Controller has written an acyclic message in the coupler). FALSE: There is no acyclic message waiting.
RX5	CIE_ACYC_BUF	TRUE: Indicates an acyclic message could be transferred to the coupler by CC-Link IE Field Master (Buffer is available). FALSE: Indicates the coupler could not handle an additional acyclic message from the CC-Link IE Field Master.
RX6-RX7	Reserved	Shall be clear

Those bits provide status information to the CC-Link IE Field Master about the PROFINET IO-Controller.

8.1.2 Coupler RWr buffer (CRWrB)

The CRWrB contains the CC-Link IE Field input data defined by the end user. Coupler variables are located from RWr2 of the CRWrB.

Table 8.1-3 RWr Buffer content

Word Address	Name	Description
RWr0	PN_ACYC_LEN	Length of the message sent by PROFINET IO-C to CC-Link IE Field Master.
RWr1 (lower byte)	Error_Code	Please see chapter Coupler Error Code
RWr1 (high byte)	Unused	Padding byte
RWr2 - RWrx	Coupler variable	End user defined cyclic variable start address

8.1.3 Coupler RY buffer (CRYB)

The CRWwB contains the BOOL CC-Link IE Field output data defined by the user. Data are located from RY0 0 of the CRYB.

8.1.4 Coupler RWw buffer (CRWwB)

The CRWwB contains the CC-Link IE Field output data defined by the end user. Data are located from RWw1 of the CRWwB. CRWwB shall transport at RWw0, the CK.

Table 8.1-4 RWw Buffer content

Word Address	Name	Description
RWw0	CK	Consistency key
RWw1 - RWwx	Coupler variable	End user defined cyclic variable start address

8.2 Cyclic communication on PROFINET

On PROFINET, all cyclic communication take place at the submodule level. The figure below shows the sub module structure of the coupler IO-Device.

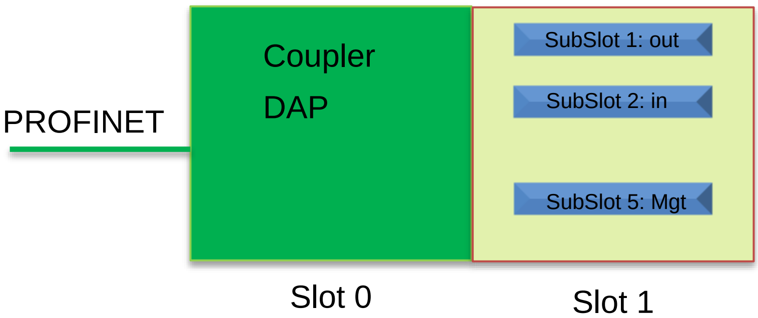

Fig. 8.2-1 Submodule model

The Slot 0 shall contains the DAP as per PROFINET specification [4]. The number of switch port, as they are defined as DAP submodule, implies an impact on the input cyclic data size capacity of the coupler.

Note: PROFINET specification does not require to have a switch built in.

The coupler module shall be located in API 0x4608 (hex). In a PROFINET compact model, the coupler module shall be located in slot 1 (see figure), but in a modular system, the slot location of the coupler module remains a manufacturer design decision.

The submodule located in subslots of slot 1 are being responsible to support PROFINET cyclic data exchange.

The table below lists slot1's subslots all with their description.

Table 8.2-1 Coupler submodule definition

Subslot	Name	Description
1	Coupler Output Submodule	Submodule managing PROFINET outputs. (Cyclic data from PROFINET IO-Controller to Coupler)
2	Coupler Input Submodule	Submodule managing PROFINET inputs. (Cyclic data from coupler to PROFINET IO-Controller)
5	Coupler Management Submodule	Management Submodule, reflecting Coupler statuses to PROFINET

The coupler shall implement those submodules.

Based on the previous IO-Device architecture definition description, here are the elements transported in the PROFINET RT frames. The maximum PROFINET RT user data length is 1440 bytes.

IO-C input

Table 8.2-2 Coupler PROFINET input elements

Element	Length(byte)	Description
DAP IOPS	3	DAP Submodules IOPS
CMS IOPS	1	Coupler Management submodule IOPS
CMS data	1	Coupler Management submodule data
CIS IOPS	1	Coupler Input submodule IOPS
CIS Data	X	Coupler PROFINET input data
COS IOCS	1	Coupler output submodule IOCS

IO-C output

Table 8.2-3 Coupler PROFINET output elements

Element	Length(byte)	Description
DAP IOCS	3	DAP Submodules IOCS
CMS IOCS	1	Coupler Management submodule IOPS
CIS IOCS	1	Coupler Input submodule IOPS
COS IOPS	1	Coupler output submodule IOCS
COS data	X	Coupler PROFINET output data

8.2.1 Coupler Output Submodule (COS)

This submodule is containing the output data defined by the end user. The COS IOPS is driven by the PROFINET IO-Controller application. The size of COS is limited to **1434 bytes**.

As per PROFINET definition:

- The coupler application shall not transfer the output data to CC-Link IE Field side, as far the coupler Output Submodule IOPS is not set to GOOD, and the received APDUStatus.ProviderStatus is RUN.
- The data are transported in big Endian and packed all together.

The coupler application shall, if above conditions are satisfied, transfer the data to the CC-Link IE Field side. The user defined content of the submodule is given to the coupler by the coupler.

COS's IOCS shall be set to GOOD.

It is responsibility of the PROFINET IO-Controller program, based on GSDML to fill COS data with user configured data.

An additional connection to the COS is out of scope of this version of the specification.

8.2.2 Coupler Input Submodule (CIS)

This submodule is containing the input data defined by the end user. The maximal size of the CIS is limited to **1431 bytes**, if the coupler has two port switch for PROFINET.

The coupler application shall set the CIS's IOPS to GOOD whenever a CC-Link IE Field Master is connected to the coupler with MyStatus.nodeStatus set to RUN. If a configuration fault is detected, then the CIS IOPS has to be set to BAD by the coupler application.

The coupler application shall fill the CIS data according the user configuration. The data are transported in big Endian and packed all together.

CIS's IOCS is ignored by the coupler application.

An additional connection to the CIS is out of scope of this version of the specification.

8.2.3 Coupler Management Submodule (CMS)

The CMS shall act as an input submodule (data from the coupler to the PROFINET IO-Controller), containing some status bits, an error code and the length of acyclic message as defined in the below table.

Table 8.2-4 CMS status bit definition & error code

Submodule data (offset in byte)	Data type	Name	Description
Byte0, bit 0 (0)	Bit	CP_Error	TRUE: An error has been detected by the coupler. Detailed error code is provided(see below) FALSE: No error detected.
Byte0, bit 1 (0)	Bit	Reserved	Not Used (Shall be clear)
Byte0, bit 2 (0)	Bit	CIE_Connected	TRUE: A CC-Link IE Field Master is connected to the coupler. FALSE: No CC-Link IE Field Master connected to the coupler.
Byte0, bit 3 (0)	Bit	CIE_DataStatus	TRUE: Connected CC-Link IE Field Master is in RUN mode. (MyStatus.nodeStatus) FALSE: Connected CC-Link IE Field Master is in STOP mode. (MyStatus.nodeStatus)
Byte0, bit 4 (0)	Bit	CIE_ACYC_MSG	TRUE: An acyclic message is waiting (CC-Link IE Field Master has written an acyclic message in the coupler). FALSE: There is no acyclic message waiting.
Byte0, bit 5 (0)	Bit	PN_ACYC_BUF	TRUE: Indicates an acyclic message could be transferred to the coupler by PROFINET-IOC (Buffer is free). FALSE: Indicates the coupler could not handle an additional acyclic message from the PROFINET IO-C.
Byte0, bit 6&7 (0)	Bit	Reserved	Not used (Shall be clear)
Byte1 (1)	Byte	Error_Code (byte)	Detailed Coupler Error code (See section Coupler Error Code)
Word1 (2)	Word	CIE_ACYC_LEN	Length in words of the acyclic message sent by CC-Link IE Field Master.

An additional connection to the CMS is out of scope of this version of the specification. CMS IOPS shall always be GOOD.

8.3 Coupler Application (CA)

For cyclic communications, the Coupler Application (CA) is responsible to forward the data from one coupler side to the other side, and backward.

The CA shall take care of the endianness and alignment constrains of PROFINET and CC-Link IE Field, as shown in the table below.

Table 8.3-1 Coupler Application constrains

Constrains	CC-Link IE Field	PROFINET
Endianness	Little Endian	Big Endian
Alignment	On a word boundary	All packed

8.3.1 Endianess

The CA shall transform the endianness of coupler variable, depending the direction of the data.

8.3.1.1 CC-Link IE Field to PROFINET data

The CA shall convert CC-Link IE Field to PROFINET data, for all data where endianness maters, from little endian to big endian.

8.3.1.2 PROFINET TO CC-Link IE Field data

The CA shall convert PROFINET to CC-Link IE Field data, for all data where endianness maters, from big endian to little endian.

8.3.2 Alignment

The CA shall copy the data at the correct location inside the COS and CIS, as well as in the CRXB, CRYB, CRWrB and CRWwB.

The location of the coupler variable on PROFINET and/or CC-Link IE Field are engineered by the CCT.

8.3.3 Transport of different data types

On PROFINET, transport of variables is made on behalf of COS and CIS. On CC-Link IE Field, the handling buffer depends on the data type. Bits are handled by RY and RX while other data type are carried by RWr and RWw.

On PROFINET, transports of bit is done by splitting a byte in several bits (Used as bit from GSDML). Then, for each packet of 8 bits to be transported, a byte is used.

8.3.4 PROFINET Module differences

As CK is used to generate COS and CIS, in case of mismatch between coupler configuration and used GSDML, the PROFINET interface is going to detect a submodule difference and request the CA to check for differences. In that case, the CP_Error bit shall be set and the Coupler Error Status has to be set to CK_Mismatch (see section Coupler Error Code). In addition, for each submodule difference detected, the CA shall set the COS/CIS submodule status to wrong. Then, only the CMS submodule is being operational.

9 Acyclic

This chapter describes the acyclic communication functionalities of the coupler. The coupler only responds to acyclic communication requests and cannot initiate acyclic communication.

For accessing the coupler from the PROFINET side PROFINET read and write requests are used. From CC-Link IE Field Network side the coupler can be accessed by SLMP messages.

SLMP (Seamless Message Protocol) is a common protocol for seamless communication between network devices. SLMP write message refers to the command MemoryWrite (1613) and SLMP read message refers to the command MemoryRead (0613).

The message interface allows the exchange of messages with a size of up to 1920 Bytes (960 Words) between both network sides.

Further a user defined data buffer allows the exchange of data. The data mapping is configured by the coupler's configuration tool in order to fulfill the specific data transfer requirements of the application.

An overview of the coupler's acyclic communication is shown in the figure below.

Coupler acyclic interface overview

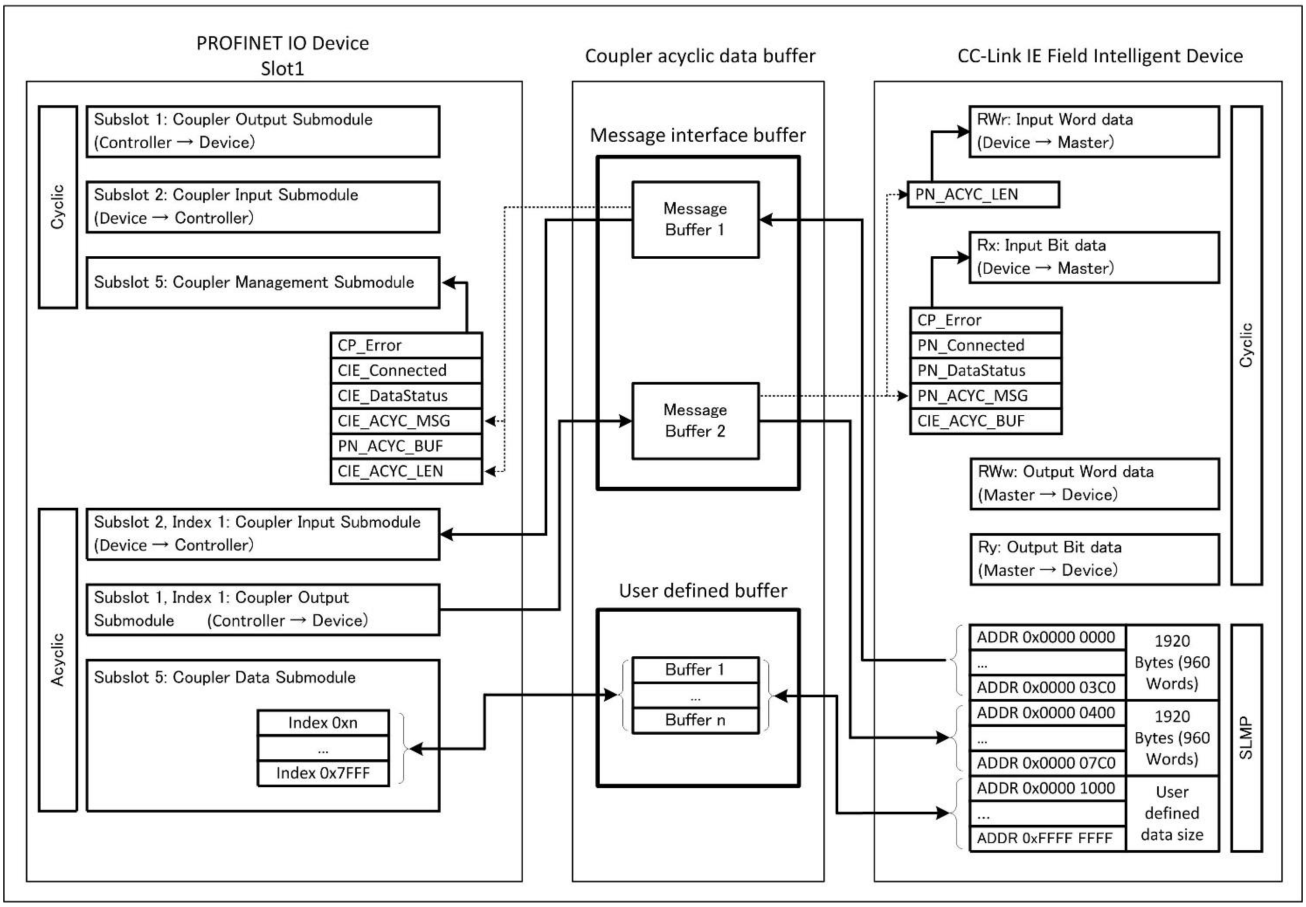

Fig. 8.31-1 Acyclic communication overview

9.1 Message interface

The message interface allows passing an acyclic message to the other network side. The control of the message interface uses cyclically transmitted status bits and words. Two message buffers are available, one for message transfer from PROFINET network to CC-Link IE Field Network and one for message transfer from CC-Link IE Field Network to PROFINET network.

The message interface provides a data transport mechanism, which can be used by the application to implement more sophisticated communication protocols (e.g. request and response data).

The PROFINET IO-Controller can send a write request to the coupler if the message buffer is available for writing, which is indicated by the PN_ACYC_BUF bit. After reception of the message the coupler indicates that a message is available to the CC-Link IE Field Master by raising the PN_ACYC_MSG bit and showing the length of the message in the PN_ACYC_LEN word. Using SLMP the CC-Link IE Field Master can retrieve the message data from the coupler, which will then reset the PN_ACYC_MSG bit and PN_ACYC_LEN word.

The CC-Link IE Field Master can send a write message via SLMP to the coupler if the message buffer is available for writing, which is indicated by the CIE_ACYC_BUF bit. After reception of the message the coupler indicates that a message is available to the PROFINET IO-Controller by raising the CIE_ACYC_MSG bit and showing the length of the message in the CIE_ACYC_LEN bytes. By using a read request the PROFINET IO-Controller can retrieve the message data from the coupler, which will then reset the CIE_ACYC_MSG bit and CIE_ACYC_LEN word.

9.1.1 PROFINET Acyclic Message Interface

The table below describes the cyclic CMS status bits and bytes, which are used on the PROFINET network side to control the acyclic message interface.

Table 9.1-1 CMS Status Bits and Bytes used for Acyclic Message Interface Control

Byte.Bit/ Word	Name	Description
B0.4	CIE_ACYC_MSG	TRUE: An acyclic message is waiting (CC-Link IE Field Master has written an acyclic message to the coupler). FALSE: There is no acyclic message waiting.
B0.5	PN_ACYC_BUF	TRUE: Indicates an acyclic message could be transferred to the coupler by PROFINET-IOC. FALSE: Indicates the coupler could not handle an additional acyclic message from the PROFINET IO-C.
W1	CIE_ACYC_LEN	The length in Words of the waiting acyclic message.

The mapping of the acyclic message data of the coupler's PROFINET network side is described in the following table. The data can be accessed by the PROFINET IO Controller by acyclic read or write request.

Table 9.1-2 PROFINET Acyclic Message Data

Address	Name	Description
Slot 1/ Subslot 1/ Index 1	Output Data	Message data to be sent from PROFINET network to CC-Link IE Field Network
Slot 1/ Subslot 2/ Index 1	Input Data	Message data to be sent from CC-Link IE Field Network to PROFINET network

9.1.2 CC-Link IE Field Acyclic Message Interface

The two tables below describe the Bits and Words, which are used on the CC-Link IE Field Network side to control the acyclic message interface.

Table 9.1-3 RX Buffer for Acyclic Message Interface Control

Bit Address	Name	Description
RX4	PN_ACYC_MSG	TRUE: An acyclic message is waiting (PROFINET IO-Controller has written an acyclic message to the coupler). FALSE: There is no acyclic message waiting.
RX5	CIE_ACYC_BUF	TRUE: Indicates an acyclic message could be transferred to the coupler by CC-Link IE Field Master. FALSE: Indicates the coupler could not handle an additional acyclic message from the CC-Link IE Field Master.

Table 9.1-4 RWr Buffer for Acyclic Message Control

Word Address	Name	Description
RWr0	PN_ACYC_LEN	The length in Words of the waiting acyclic message.

The mapping of the acyclic message data of the coupler's CC-Link IE Field Network side is described in the following table. The data can be accessed by the CC-Link IE Field Master via SLMP read or write message.

Table 9.1-5 CC-Link IE Field Acyclic Message Data

Word Address	Name	Description
0x0000 0000	Output Data	Message data to be sent from CC-Link IE Field Network to PROFINET network
…		
0x0000 03C0		
0x0000 0400	Input Data	Message data to be sent from PROFINET network to CC-Link IE Field Network
…		
0x0000 07C0		

9.1.3 Connection Timeout Error

In case of a connection timeout error the coupler could either hold or clear the latest uncollected data. The behaviour should be settable by the coupler's configuration tool.

9.2 User Defined Buffer

The coupler's user defined buffer can be configured according to the application's requirements by using the coupler's configuration tool. The configuration tool allows setting up buffers with addresses both on PROFINET network and CC-Link IE Field Network side. Further the data sizes and start addresses can be set individually with a maximum data size of 1920 Bytes (960 Words).

The buffers can be accessed from PROFINET network side via read or write request. From CC-Link IE Field Network side the buffers can be accessed via SLMP read or write message. There is no notification to the PROFINET IO Controller or CC-Link IE Field Master that data has been updated.

Table 9.2-1 User defined Buffer for Acyclic Data Exchange

Address		Name	Description
PROFINET network	CC-Link IE Field Network [Word address]		
Slot 1/ Subslot 5/ Index 0x0	0x0000 1000	Data x	User defined data, setting by coupler's configuration tool
	….		
	….		

…	…	….	…
Slot 1/ Subslot 5/ Index 0x7FFF	… … 0x7FFF FFFF	Data y	User defined data, setting by coupler's configuration tool

9.3 Sequence

9.3.1 Message interface

The message interface allows passing acyclic messages from one network side to the other network side.

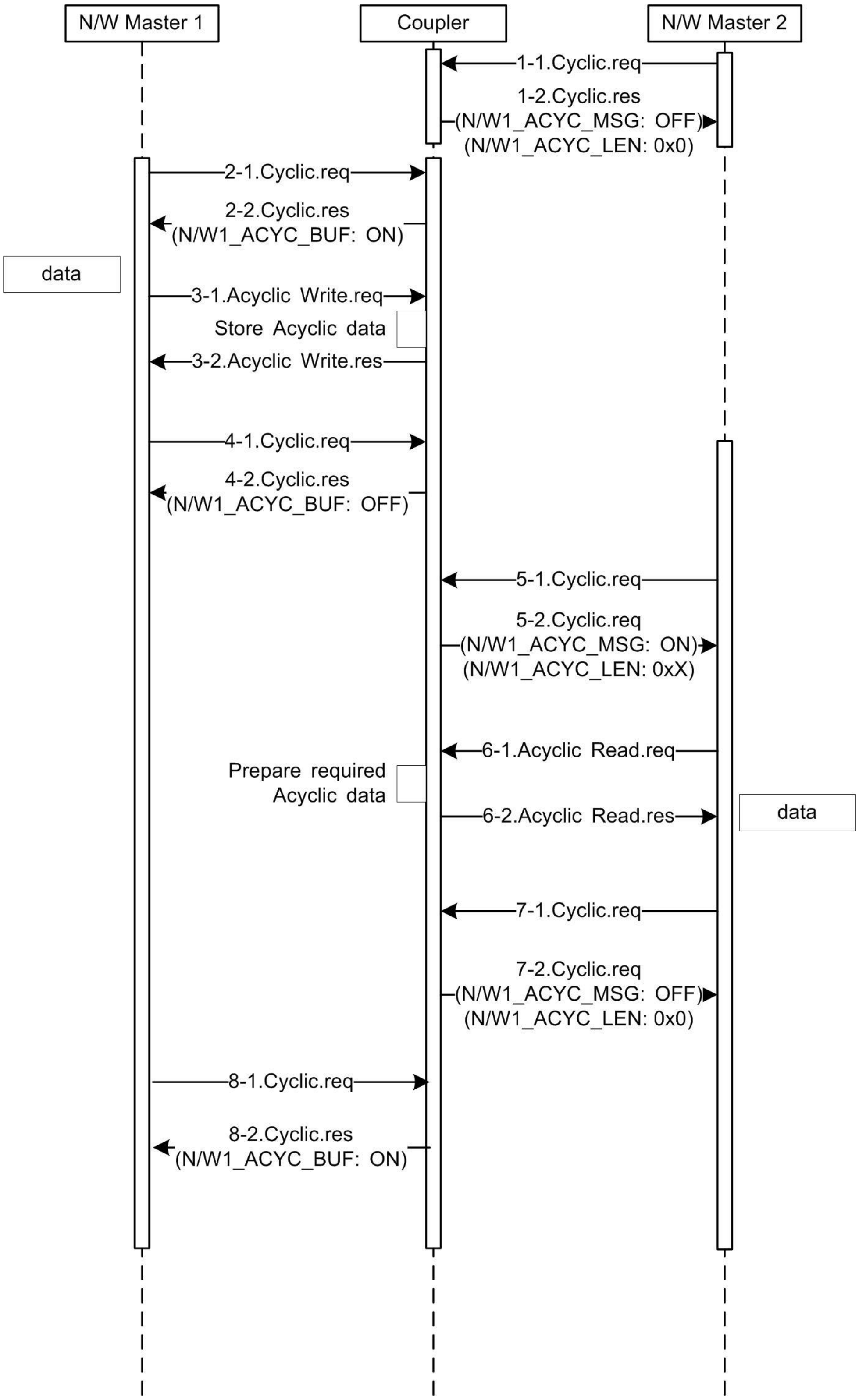

Fig. 9.3-1 Sequence of Message Interface Usage

1-1)
N/W Master 2 sends the cyclic request service.

1-2)
The coupler replies with the cyclic response service. The acyclic message waiting bit (N/W1_ACYC_MSG) is set to OFF and the N/W1_ACYC_LEN register contains 0x0.

2-1)
N/W Master 1 sends the cyclic request service.

2-2)
The coupler replies with the cyclic response service. The message buffer available bit (N/M2_ACYC_BUF) is set to ON, which means that the message buffer is available.

3-1)
N/W Master 1 sends the acyclic write request service.

3-2)
The coupler replies with the acyclic write response service.

4-1)
N/W Master 1 sends the cyclic read request service.

4-2)
The coupler replies with cyclic response service. The message buffer available bit (N/W2_ACYC_BUF) is set to OFF, which means that the message buffer is busy.

5-1)
N/W Master 2 sends the cyclic read request service.

5-2)
The coupler replies with cyclic read response service. The message available bit (N/W1_ACYC_MSG) is set to ON and the N/W1_ACYC_LEN register shows the length of the message.

6-1)
N/W Master2 sends the acyclic read request service.

6-2)
The coupler prepares the acyclic data and replies with the acyclic read response service.

7-1)
N/W Master 2 sends the cyclic request service.

7-2)
The coupler replies with the cyclic response service. The message available bit (N/W1_ACYC_MSG) bit is set to OFF and the N/W1_ACYC_LEN register contains 0x0.

8-1)
N/W Master 1 sends the cyclic request service.

8-2)
The coupler replies with the cyclic response service. The message buffer available bit (N/W2_ACYC_BUF) is set to ON, which indicated that a new message can be written.

9.3.2 Access of User Defined Buffer

The coupler's user defined data buffer can be accessed directly and independently from both network sides. The data consistency is kept by the coupler.

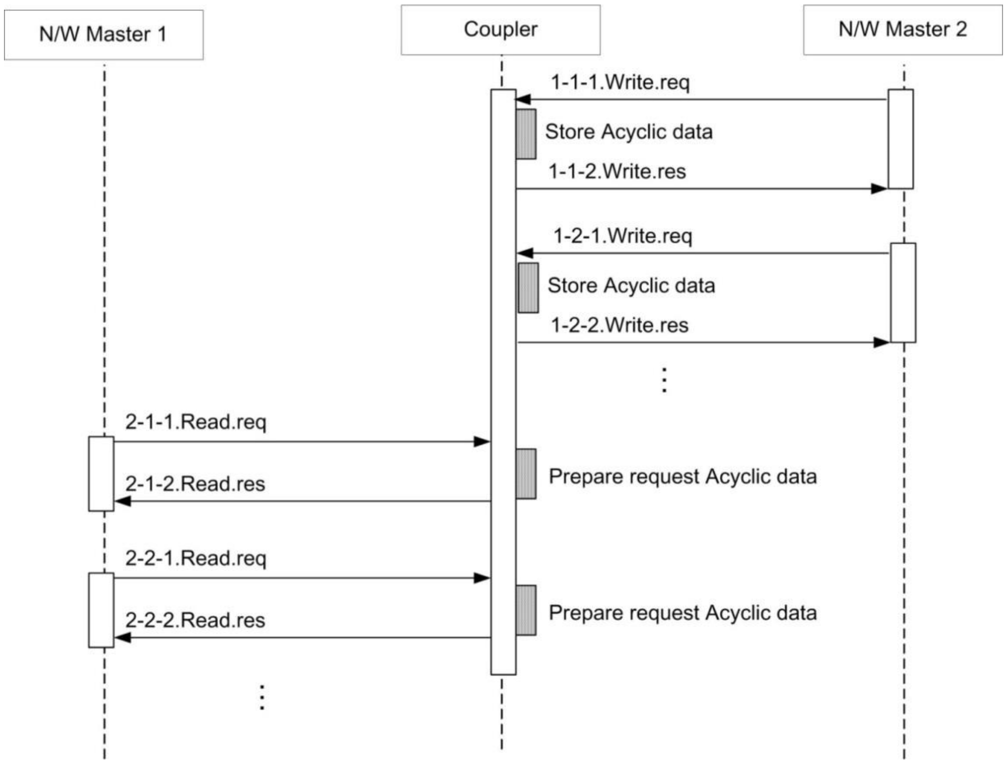

Fig. 9.3-2 Data Access Sequence

1-1-1)
N/W Master 2 sends acyclic write request service including the data.

1-1-2)
Coupler stores the data and replies with the acyclic response service.

1-2-1)
N/W Master 2 sends acyclic write request service including the data.

1-2-2)
Coupler stores the data and replies with the acyclic response service.

2-1-1)
N/W Master 1 sends acyclic the read request service.

2-1-2)
Coupler prepares the requested data and replies with the acyclic response service including the data.

2-2-1)
N/W Master 1 sends acyclic the read request service.

2-2-2)
Coupler prepares the requested data and replies with the acyclic response service including the data.

10 Diagnostics

This chapter describes the diagnostic functionalities of the coupler. The coupler provides the diagnosis information about the common status of the coupler itself. Additionally the diagnosis information from the opposite network is provided. This diagnosis information is mapped by the coupler to corresponding diagnosis mechanism of each network side.

10.1 Coupler Diagnostic Information

In order to cover different use cases the diagnostic information is available via different network specific mechanisms. I.e. it can be indicated using cyclic data, acyclic data, alarm mechanism.

An overview of the coupler's diagnosis mechanisms is shown in figure 10-1.

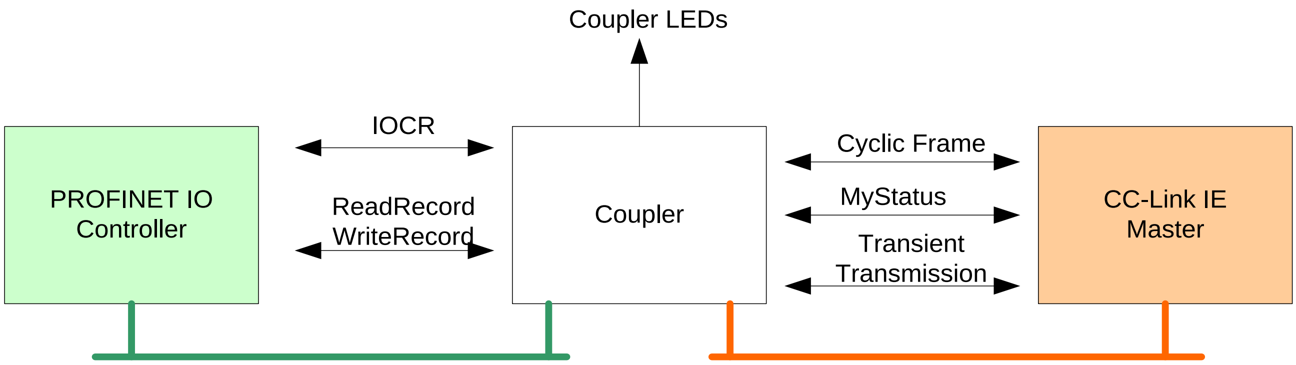

Fig. 10.1-1 Coupler Diagnostics

Coupler diagnosis data consist of several status flags which are explained in this chapter.

10.1.1 Coupler Error Flag

The CP_Error flag shall be provided by the coupler to report a detected error state.

Table 10.1-1 Coupler Error

Name	Description
CP_Error	CP_Error flag is used together with Coupler Error Code (see chapter 10.1.2) to provide more details about detected error state.

10.1.2 Coupler Error Code

Coupler error code shall be used together with CP_error Bit to report the detected error state. The error code byte shows only the last error detected and is overwritten if further error is detected; there is no buffer for diagnosis history available.

Error code is set to 0x00 (no error) if no error detected or error is vanished. The CP_Error flag shall be set to FALSE if Error code equals 0x00 (no error).

Following error codes are defined:

Table 10.1-2 Coupler Error Code

Error code	Description
0x00	No error
0x01	CK Mismatch. Indicates if a mismatch between the CK provided by the CCT, the GSDML file or the CSP+ has been detected.
0x02	Configuration Error. No configuration found (i.e. configuration was not yet provided by CCT).
0x03	Configuration Error. No valid configuration (i.e. configuration provided by CCT can not be loaded as it is not valid).
0x04..0x7F	Reserved for future, not used
0x80..FF	Used for vendor specific coupler error code

10.1.3 CC-Link IE Field Diagnosis

Coupler provides following diagnosis flags of the CC-Link IE Field Network side to PROFINET Controller:

Table 10.1-3 Coupler CC-Link IE Field Diagnosis

Name	Description
CIE_Connected	Indicates whether a CC-Link IE Field Master is connected to the coupler and exchanges cyclic data.
CIE_DataStatus	Indicates if the connected CC-Link IE Field Master is in RUN or STOP mode (MyStatus.nodeStatus).
CIE_ACYC_MSG	Indicates if an acyclic message is available from opposite side (CC-Link IE Field Master has written an acyclic message in the coupler).
CIE_ACYC_LEN	Indicates the length of the message from CC-Link IE Field Master in the acyclic message buffer. Used together with PN_ACYC_MSG
PN_ACYC_BUF	Indicates if a buffer is available in the coupler to receive an acyclic message from PROFINET IO-Controller

10.1.4 PROFINET Diagnosis

Coupler provides following diagnosis flags of the PROFINET side to CC-Link IE Field Master:

Table 10.1-4 PROFINET Diagnosis

Name	Description
PN_Connected	Indicates if a PROFINET IO-Controller is connected to the coupler and exchanges cyclic data.
PN_DataStatus	Indicates if the connected PROFINET IO-Controller is in RUN or STOP mode (APDU Status).
PN_ACYC_MSG	Indicates if an acyclic message from opposite side is available for read Flag is activated if the PROFINET IO-Controller has written an acyclic message in the coupler through WriteRecord.req (see chapter 9)
PN_ACYC_LEN	Indicates the length of the message from PROFINET IO-Controller in the acyclic message buffer. Used together with PN_ACYC_MSG
CIE_ACYC_BUF	Indicates if a buffer is available in the coupler to receive an acyclic message from CC-Link IE Field Master

Coupler does not provide any channel related diagnosis information to PROFINET.

10.2 Coupler Indicators

Coupler shall provide network specific indicators according to requirements of PROFINET IO [4] and CC-Link IE Field Network [1] specifications.

Coupler should provide a coupler status indicator to signalize availability of power and status of coupling application.

If LED indicator is used as coupler status indicator, a green LED shall be used to indicate following coupler states:

Table 10.2-1 Coupler Indicators

LED Indicator state	Description
OFF	Coupler not powered
Blinking slow (0.5Hz)	Coupler is ready to operate, but has detected some error (see chapter 10.1.1) or both networks are not connected.
Solid ON	Coupler is operating, one or both network sides are connected and exchange data.

10.3 Diagnosis Data Mapping

10.3.1 Mapping of Diagnosis Data to PROFINET

On the PROFINET IO-Device side of the coupler the diagnosis information shall be provided using following mechanisms:

- Coupler Error Flag (see chapter 10.1.1) and Error Code (see chapter 10.1.2) as well as CC-Link IE Field Diagnosis (see chapter 10.1.3) shall be mapped to coupler management submodule CMS (s. chapter 8.2.3) and transferred to PROFINET IO-Controller cyclically.
- Mapping of CC-Link IE data to CIS submodule is made and IOPS of CIS submodule is set to GOOD only if CIE_Connected = TRUE and CIE_DataStatus = TRUE. Otherwise the IOPS of CIS shall be set to BAD and not CC-Link IE data shall be mapped to CIS submodule.

10.3.2 Mapping to CC-Link IE Field Diagnosis

On the CC-Link IE Field Slave side of the coupler the diagnosis information is provided using following mechanisms:

- Coupler Error Flag (see chapter 10.1.1) and Error Code (see chapter 10.1.2) as well as PROFINET Diagnosis (see chapter 10.1.4) shall be mapped to Coupler RX buffer (CRXB) (see chapter 8.1.1) and transferred to CC-Link IE Field Master cyclically.
- Mapping of PROFINET IO data from COS submodule to CC-Link IE Field is made only if PN_Connected = TRUE and PN_DataStatus = TRUE and COS.IOPS = GOOD.
- MyStatus.nodeStatus of coupler shall be set to “Minor error” if PN_Connected = FALSE or, COS.IOPS = BAD or PN_DataStatus = FALSE.

10.4 Diagnostic Use Cases

10.4.1 Configuration mismatch

Configuration mismatch is detected comparing the local CK which is loaded by CCT to the coupler and expected CK, which is configured by the network controller (see chapter 7.2)

The mismatch has influence on the connection establishment (see chapter 11). Other network side can get the diagnosis information about mismatch of CK evaluating CP_Error flag and Error Code (see chapter 10.1.1 and 10.1.2).

On PROFINET network side changes to the expected configuration (i.e. changes in the engineering data) shall be dealt with according to the PROFINET IO specification [4]. The user changes the expected configuration in the engineering tool of the PROFINET IO-Controller.

On CC-Link IE Field Network side changes to the expected configuration shall be dealt according to the CC-Link IE Field Network specification[1], i.e. lead to a new start-up sequence.

10.4.2 Configuration Errors

If coupler application has started and no configuration is found or the configuration is not valid, then coupler set CP_Error = TRUE and Error Code to 0x02 or 0x03 (see chapter 10.1.2).

The coupler shall allow the PROFINET IO Controller to connect, but shall pull the CIS and COS submodules because no valid configuration for their submoduleIDs and data length is available. CMS submodule shall be always plugged allow connection establishment and access to the coupler diagnostic information.

This state of coupler can be recognised by PROFINET IO Controller during connection establishment as the coupler generates ModuleDiffBlock. CC-Link IE Field Master can recognise the configuration error checking CP_Error flag and Error Code.

Coupler application shall set the CIE_DataStatus=FALSE and PN_DataStatus = FALSE. The PN_Connected and CIE_Connected shall be set according to the state of connection.

User shall load valid configuration to the coupler to leave this state (see chapter 10.4.4)

10.4.3 Connection loss

If the connection to PROFINET IO-Controller is lost the coupler shall report this to CC-Link IE setting PN_Connected = FALSE and PN_DataStatus = FALSE as well as PN_ACYC_MSG = FALSE (no message available) and updating this in the CRXB. Coupler application shall stop mapping of other data (CRYB, CRWrB, CRWwB). Data exchange with CC-Link IE Field Master shall be kept.

Coupler is waiting until a new connection to PROFINET IO Controller is established. As soon as the APDUStatus of Data becomes RUN the coupler reports this to CC-Link IE setting PN_Connected = TRUE, PN_DataStatus = TRUE. Coupler shall proceed with data mapping between both networks. MyStatus.nodeStatus of coupler shall be set to “Minor error” if PN_Connected = FALSE.

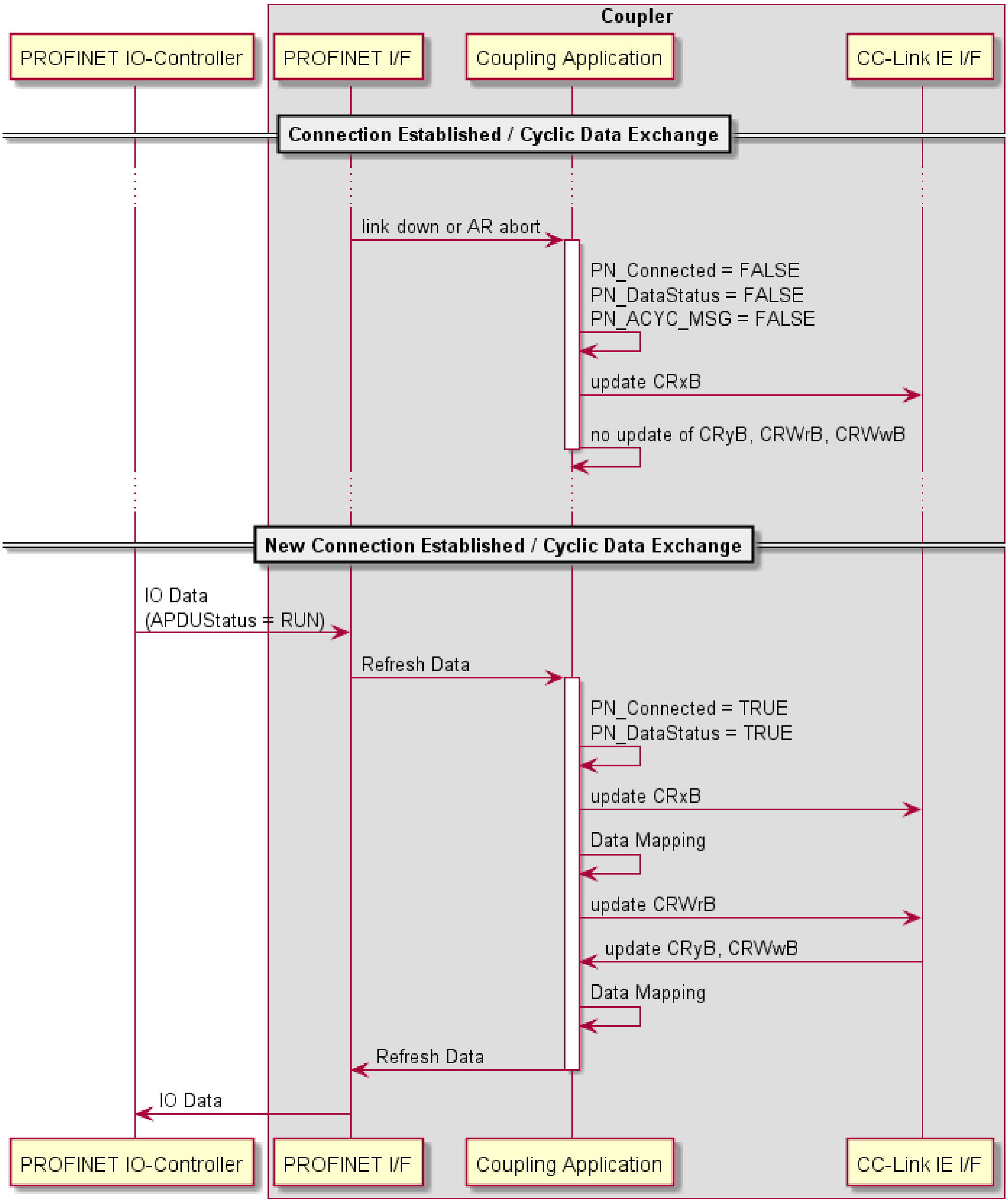

Fig. 10.4-1 Connection loss with PROFINET IO-Controller

If the connection to CC-Link IE Field Master is lost the coupler shall report this to PROFINET setting CIE_Connected = FALSE and CIE_DataStatus = FALSE as well as CIE_ACYC_MSG = FALSE (no message available) and updating this in the CMS submodule. Coupler application shall stop mapping of other data (COS, CIS submodules), set the IOPS of COS submodule and IOCS of CIS submodule to BAD. Data exchange (AR) with PROFINET IO Controller shall be kept allow data exchange with CMS submodule.

Coupler is waiting until a new connection to CC-Link IE Field Master is established. As soon as the MyStatus.nodeStatus becomes RUN the coupler reports this to PROFINET setting CIE_Connected =

TRUE, CIE_DataStatus = TRUE, set the IOPS of CIS submodule and IOCS of COS submodule to GOOD. Coupler shall proceed with data mapping between both networks.

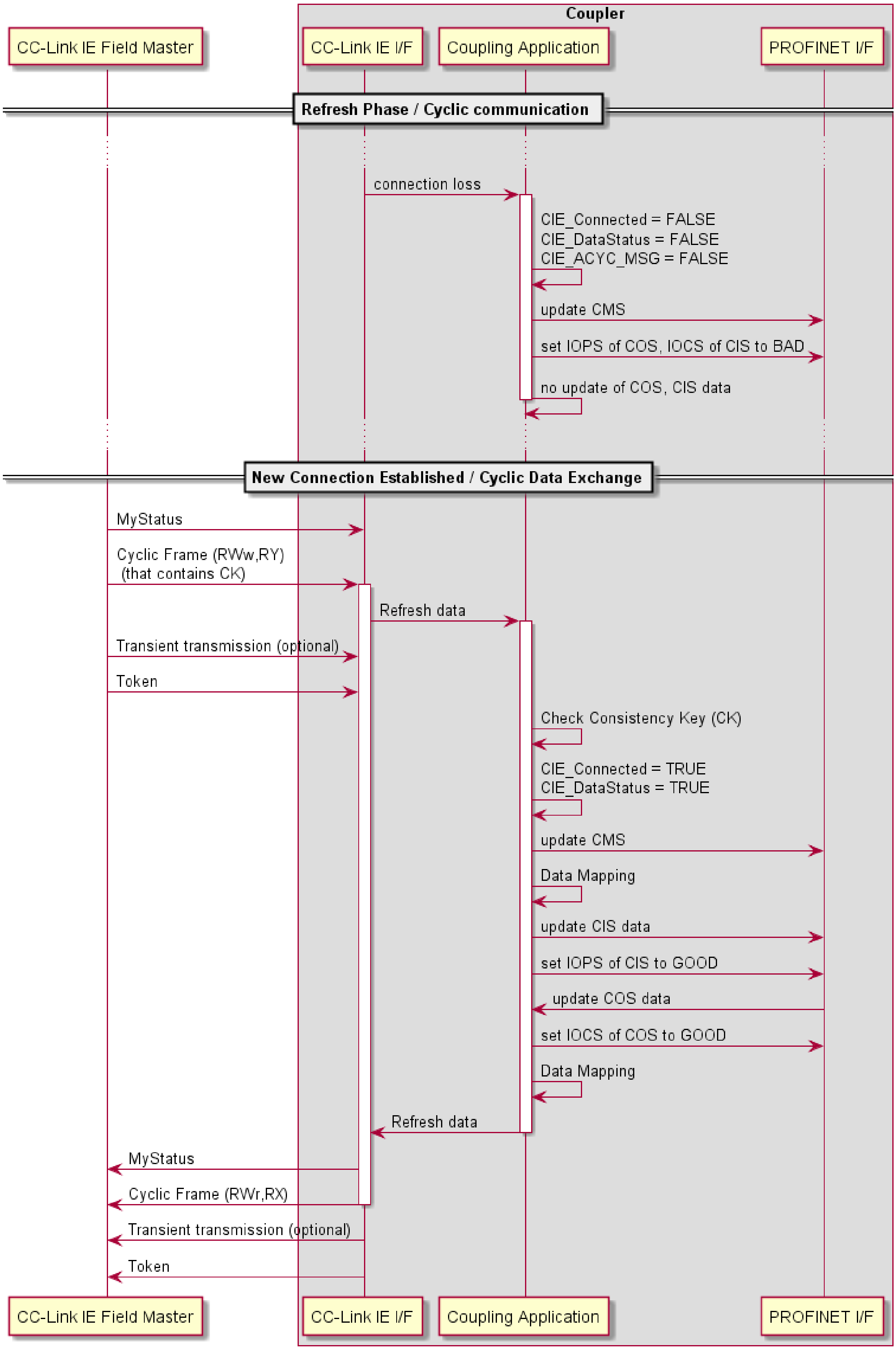

Fig. 10.4-2 Connection loss with CC-Link IE Field Master

If connection is lost to both PROFINET IO-Controller and CC-Link IE Field Master, the coupler shall reset all coupler diagnostic flags and wait for new connection (see chapter 11).

10.4.4 Configuration reload

If CCT is connected to coupler and loads a new configuration coupler shall proceed following sequence:

- Drop existing connections to PROFINET IO Controller and CC-Link IE Field Master
- Reset all coupler diagnostic flags
- Evaluate new coupler configuration and allow the PROFINET IO Controller and CC-Link IE Field Master to establish a new connection (see chapter 11)

11 Start-up Sequence

This chapter gives an overview of the start-up sequence of the coupler.

In general both network sides of the coupler shall be able to start-up independent from the other side. This means that the connection to PROFINET IO-Controller can be established independent if the connection to CC-Link IE Field Master is available. CC-Link IE Field Master connection can be established independently of the PROFINET IO-Controller connection. The actual status of the network connection and status of the application is available as diagnosis information to other network.

11.1 Connection Establishment with PROFINET IO-Controller

A connection will be established between the PROFINET IO-Controller and the coupler as shown on the figure below.

PROFINET IO-Controller sends a connect.req with expected configuration to the coupler. Coupler checks if the expected configuration corresponds to the actual configuration of the coupler. Chapter 10.4.1 and 10.4.2 describe possible use cases and their diagnosis.

The PROFINET IO-Controller will parameterize the device by writing the parameter records of the valid (sub-)modules. This data has to be mapped to the predefined CC-Link IE Field acyclic field by the coupler application. When the PROFINET IO-Controller has finished writing the parameters, it will generate a ParamEnd Control message.

Provider data for PROFINET will be updated after the coupler has finished updating the internal status and mapping memory. Coupler sends an Application Ready Control message to PROFINET IO-Controller. Connection is established as soon as coupler received a confirmation from PROFINET IO-Controller.

Coupler sends the application ready signal to PROFINET IO-Controller independent of the status of the communication on the CC-Link IE Field Network. The status of the CC-Link IE Field Network connection and N/W Master application is mapped to the coupler management submodule CMS and transferred to PROFINET IO-Controller (s. chapter 8.2.3).

The result of the CK configuration and CK checking is provided via CP_Error and Error Code (see chapter 10.1.1 and 10.1.2) to other network. CK Mismatch error on CC-Link IE Field side is indicated with this flag, but shall have no influence on connection establishment to PROFINET Io Controller if CK check for PROFINET configuration has no mismatch error.

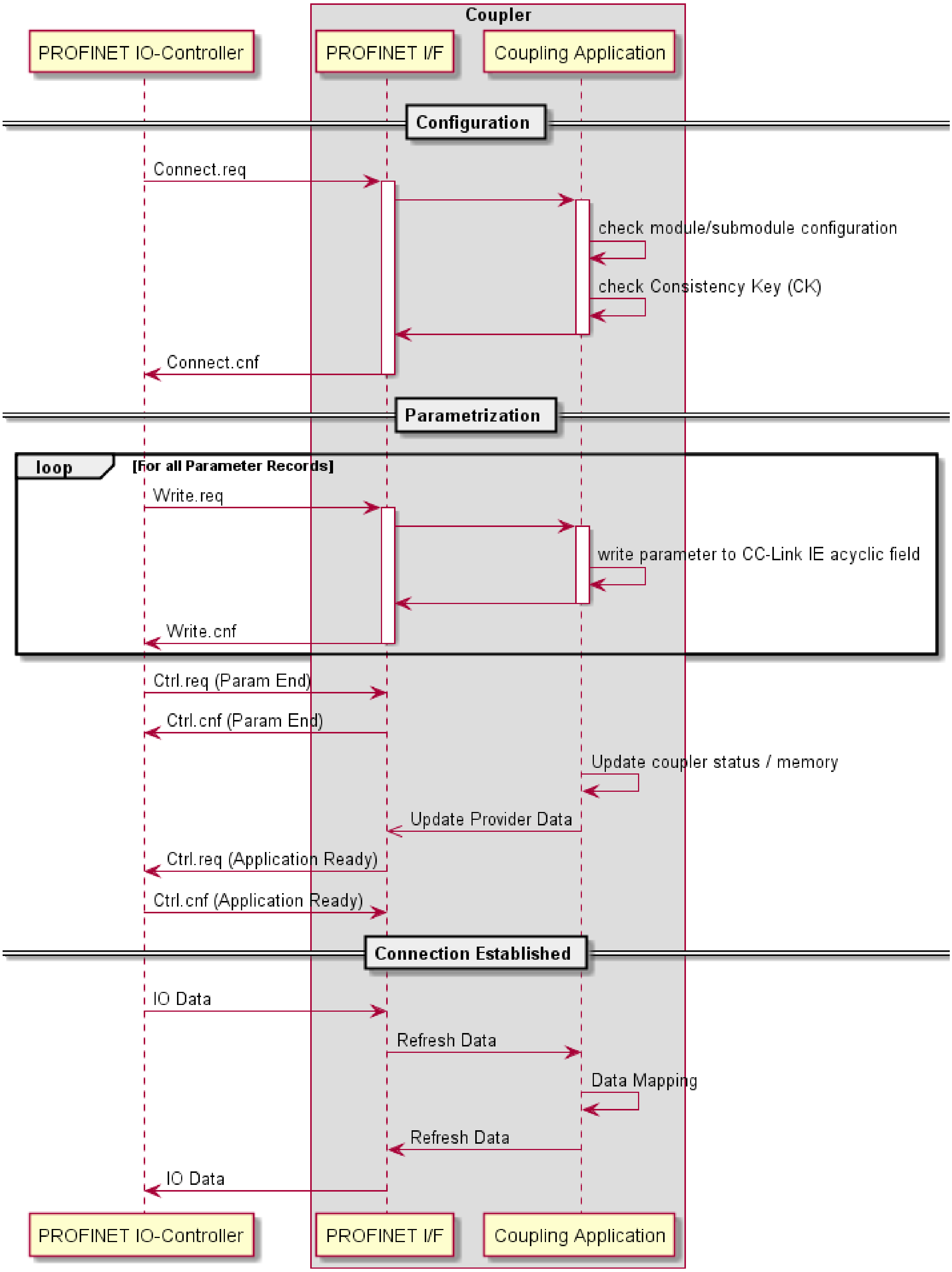

Fig. 11.1-1 Connection establishment with PROFINET IO-Controller

11.2 Connection Establishment with CC-Link IE Field Master

A connection will be established between the CC-Link IE Field Master and the coupler according to CC-Link IE Field Network specification [1]. In the initialization sub phase, after power ON or return after reset, preparatory tasks are executed until refresh begins according to the following procedure:

Identification / Connection:

- The master station sends a TestData frame to check the identification information and connection information of nodes connected to the network. When the slave station receives the TestData frame, the slave station transmit the frame from other ports of CC-Link IE Field in case the slave has more than two CC-Link IE Field ports. It also transmits a TestDataAck frame to the master station to notify node identification information and connection information.
- The master station receives the TestDataAck frames to verify the connected slave stations, collects the connection information and determines the token passing route based on the slave station identification information.
- The master station sends a Setup frame to each slave station to set the determined token passing route. Upon receipt of the Setup frame, the slave station sends a SetupAck frame to the master station to respond to token passing route setup and notify the master station of its node identification information.
- The master station receives the SetupAck frame, confirming that the passing route has been set in the slave station of the transmission source of the SetupAck frame. The master station receives the SetupAck frame from all slave stations, establishing the token passing route.

Parametrization:

- The master station sends a MyStatus frame, Parameter frame, and Token frame. The MyStatus frame is a frame for notifying all nodes of node status information. The Parameter frame is a frame for distributing cyclic transmission parameters to slave stations from the master station, and is individually transmitted to the slave stations. The Token frame indicating the token specifies the node that is to be the next token holding node, and is transmitted to all nodes.
- As coupler becomes the token holding node it transmits a MyStatus frame and Token frame. The MyStatus frame is transmitted to all nodes to notify the nodes of the node information and parameter reception status. The Token frame specifies the node that is to become the next token holding node.
- The master station uses the parameter reception status information included in the MyStatus frame transmitted by the coupler and checks the status of parameter reception. Once the parameter reception statuses of all slave stations have been checked, the master station transmits a ParamCheck frame to each slave station and requests the stations to reflect parameters. Each slave station then sends its parameter reflection status to the master station using the MyStatus frame.
- The master station checks the parameter reflection status of each slave station using the information included in the MyStatus frame of the slave station. Upon confirmation that all stations have reflected the parameters, preparation for cyclic transmission is complete.

In the refresh phase the data exchanged cyclically between CC-Link IE Field Network nodes according to token passing route. The token holding node sends its MyStatus frame, cyclic frame and transient frames.

Consistency Key (CK) is send by CC-Link IE Field Master with Cyclic Frame at the Refresh Phase. The information of CK is set by the PLC program in the Cyclic Frame. Coupler checks if the expected configuration corresponds to the actual configuration of the coupler. Chapter 10.4.1 and 10.4.2 describe possible use cases and their diagnosis.

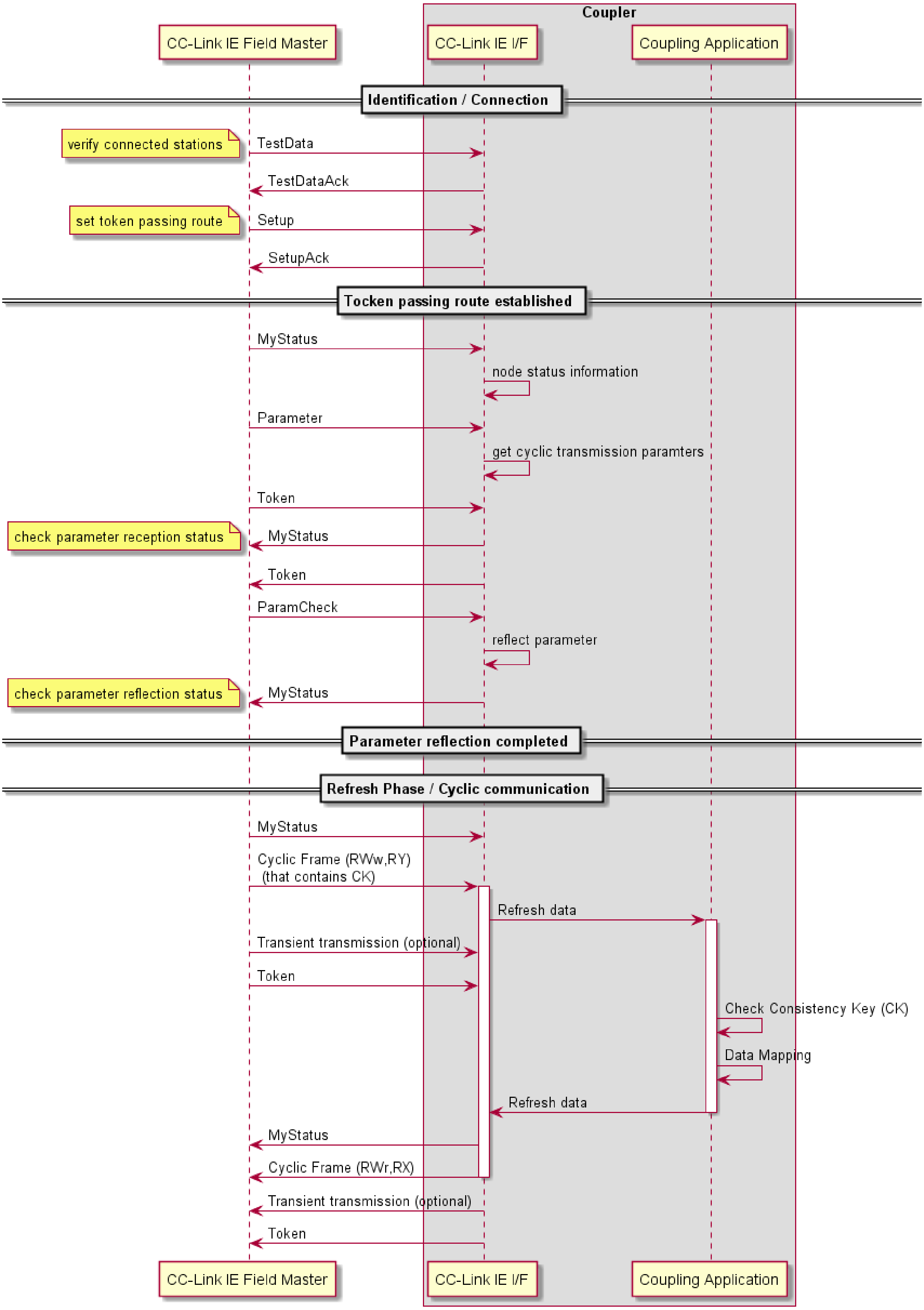

Fig. 11.2-1 Connection establishment with CC-Link IE Field Master

12 Requirement for certification tests

The coupler implementations shall comply with PI PROFINET IO-Device certification specification [7] AND CLPA CC-Link IE Field Intelligent Device certification specification [6].

A coupler shall succeed on both PI and CLPA certification tests, and certificates shall be available.

A specific coupler test specification might be issued.

13 Annex A: Coupler example

13.1.1 Example

In order to illustrate the coupler implementation, we are going to consider the transport of the following variables:

Table 11.2-1 Example: data to be transported

Variable name	Type	Direction
BIT_PROFINET_TO_CCLINKIE	bit	PROFINET to CC-Link IE Field Network
BYTE_PROFINET_TO_CCLINKIE	Integer8	PROFINET to CC-Link IE Field Network
WORD_PROFINET_TO_CCLINKIE	Integer16	PROFINET to CC-Link IE Field Network
DWORD_PROFINET_TO_CCLINKIE	Integer32	PROFINET to CC-Link IE Field Network
BIT_CCLINKIE_TO_PROFINET	bit	CC-Link IE Field Network to PROFINET
BYTE_CCLINKIE_TO_PROFINET	Integer8	CC-Link IE Field Network to PROFINET
WORD_CCLINKIE_TO_PROFINET	Integer16	CC-Link IE Field Network to PROFINET
DWORD_CCLINKIE_TO_PROFINET	Integer32	CC-Link IE Field Network to PROFINET

13.1.1.1 PROFINET Submodules

PROFINET does not define the order of the submodule within the PROFINET cyclic frames, this is belonging to the IO-Controller Engineering tool implementation. Most of the time, they are ordered by growing subslot number.

PROFINET output RTC frame (PROFINET IO-Controller to coupler)

Table 11.2-2 Example: PROFINET output frame

<table>
<tr><th>Element</th><th>Length(byte)</th><th colspan="3">Description</th></tr>
<tr><td>DAP IOCS</td><td>3</td><td colspan="3">DAP Submodules IOCS</td></tr>
<tr><td>CMS IOCS</td><td>1</td><td colspan="3">Coupler Management submodule IOPS</td></tr>
<tr><td>CIS IOCS</td><td>1</td><td colspan="3">Coupler Input submodule IOPS</td></tr>
<tr><td>COS IOPS</td><td>1</td><td colspan="3">Coupler output submodule IOCS</td></tr>
<tr><td rowspan="2">COS data</td><td rowspan="2">8</td><td>Variable</td><td>Byte offset</td><td>Length(Byte)</td></tr>
<tr><td>BIT_PROFINET_TO_CCLINKIE</td><td>0</td><td>1

Bit 0 shows the variable. All other bits unused.</td></tr>
</table>

		BYTE_PROFINET_TO_CCLINKIE	1	1
		WORD_PROFINET_TO_CCLINKIE	2	2
		DWORD_PROFINET_TO_CCLINKIE	4	4

PROFINET Input RTC frame (coupler to PROFINET IO-Controller)

Table 11.2-3 Example: PROFINET input frame

Element	Length(byte)	Description		
DAP IOPS	3	DAP Submodules IOPS		
CMS IOPS	1	Coupler Management submodule IOPS		
CMS data	3	Coupler Management submodule data: Coupler status and Acyclic length.		
CIS IOPS	1	Coupler Input submodule IOPS		
CIS Data	8	**Variable**	**Byte offset**	**Length(Byte)**
		BIT_CCLINKIE_TO_PROFINET	0	1 Bit 0 shows the variable. All other bits unused.
		BYTE_CCLINKIE_TO_PROFINET	1	1
		WORD_CCLINKIE_TO_PROFINET	2	2
		DWORD_CCLINKIE_TO_PROFINET	4	4
COS IOCS	1	Coupler output submodule IOCS		

13.1.1.2 CC-Link IE Buffers

RX Buffer (CRXB)

Table 11.2-4 Example: CC-Link IE Field input bit frame

Address	Length(byte)	Description		
RX0 to RX7	1	Status bits (see 8.1.1)		
RX8	1	**Variable**	**Byte offset**	**Length(Byte)**

		BIT_PROFINET_TO_CCLINKIE	0	1

RWr Buffer (CRWr)

Table 11.2-5 Example: CC-Link IE Field input word frame

Address	Address	Length(byte)	Variable
RWr0	0	2	PN_ACYC_LEN
RWr1	1	1	Error_Code
	1	1	Padding byte
RWr2	0	1	BYTE_ PROFINET_TO_CCLINKIE
	1	1	Padding byte
RWr3	2	2	WORD_PROFINET_TO_CCLINKIE
RWr4 & RWr5	4	4	DWORD_PROFINET_TO_CCLINKIE

RY Buffer (CRYB)

Table 11.2-6 Example: CC-Link IE Field output bit frame

Address	Length(byte)	Description		
RY0	1	**Variable**	**Byte offset**	**Length(Byte)**
		BIT_CCLINKIE_TO_PROFINET	0	1 Bit 0 shows the variable. All other bits unused.

RWw Buffer (CRWw)

Table 11.2-7 Example: CC-Link IE Field output word frame

Address	Offset (Byte)	Length(byte)	Variable
RWw0	0	2	CK
RWw1	2	1	BYTE_CCLINKIE_TO_PROFINET
	3	1	Reserved
RWw2	4	2	WORD_CCLINKIE_TO_PROFINET
RWw3	6	4	DWORD_CCLINKIE_TO_PROFINET