INTERNATIONAL STANDARD

ISO 15745-5

First edition
2007-02-01

Industrial automation systems and integration — Open systems application integration framework —

Part 5: Reference description for HDLC-based control systems

Systèmes d'automatisation industrielle et intégration — Cadres d'intégration d'application pour les systèmes ouverts —

Partie 5: Description de référence pour les systèmes de contrôle fondés sur HDLC

Reference number
ISO 15745-5:2007(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

ISO 15745-5 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*.

ISO 15745 consists of the following parts under the title *Industrial automation systems and integration — Open systems application integration framework*

— *Part 1: Generic reference description*

— *Part 2: Reference description for ISO 11898-based control systems*

— *Part 3: Reference description for IEC 61158-based control systems*

— *Part 4: Reference description for Ethernet-based control systems*

— *Part 5: Reference description for HDLC-based control systems*

Introduction

The application integration framework (AIF) described in ISO 15745 defines elements and rules that facilitate:

— the systematic organization and representation of the application integration requirements using integration models;

— the development of interface specifications in the form of application interoperability profiles (AIPs) that enable both the selection of suitable resources and the documentation of the "as built" application.

ISO 15745-1 defines the generic elements and rules for describing integration models and AIPs, together with their component profiles - process profiles, information exchange profiles, and resource profiles. The context of ISO 15745 and a structural overview of the constituents of an AIP are given in Figure 1 of ISO 15745-1:2003.

This part of ISO 15745 extends the generic AIF described in ISO 15745-1 by defining the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles specific to HDLC[1] -based control systems (CC-Link[2]). CC-Link is based on HDLC technology.

In particular, this part of ISO 15745 describes technology specific profile templates for the device profile and the communication network profile. Within an AIP, a device profile instance or a communication network profile instance is part of the resource profile defined in ISO 15745-1. The device profile and the communication network profile XML instance files are included in a resource profile XML instance using the ProfileHandle_DataType as specified in ISO 15745-1:2003, 7.2.5.

AIFs specified using the elements and rules of ISO 15745-1 can be easily integrated with the component profiles defined using the elements and rules specified in this part of ISO 15745.

[1] HDLC is used in this document as a synonym for ISO/IEC 13239.

[2] CC-Link is the trade name of the CC-Link Partner Association (CLPA). This information is given for the convenience of users of ISO 15745 and does not constitute an endorsement by ISO of the trade name holder or any of its products. Compliance to this part of ISO 15745 does not require use of the trade name CC-Link. Use of the trade name CC-Link requires permission of the CLPA.

Industrial automation systems and integration — Open systems application integration framework — Part 5: Reference description for HDLC-based control systems

1 Scope

This part of ISO 15745 defines the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles specific to HDLC-based control systems.

NOTE Generic elements and rules for describing integration models and application interoperability profiles, together with their component profiles (process profiles, information exchange profiles, and resource profiles) are specified in ISO 15745-1.

This part of ISO 15745 is to be used in conjunction with ISO 15745-1 to describe an application integration framework.

2 Normative references

The following referenced documents are indispensable for the application 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.

ISO 15745-1:2003, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

ISO/IEC 13239, *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*

REC-xml-20040204, *Extensible Markup Language (XML) 1.0 Third Edition – W3C Recommendation 04 February 2004*

REC-xmlschema-1-20010502, *XML Schema Part 1: Structures – W3C Recommendation 02 May 2001*

REC-xmlschema-2-20010502, *XML Schema Part 2: Datatypes – W3C Recommendation 02 May 2001*

UML V1.4, *OMG - Unified Modeling Language Specification (Version 1.4, September 2001)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 15745-1 apply.

NOTE The UML terminology and notation used in this part of ISO 15745 is described in Annex A of ISO 15745-1:2003 and Annex B of this part of ISO 15745.

4 Abbreviated terms

AIF	Application Integration Framework
AIP	Application Interoperability Profile
ASCII	American Standard Code for Information Interchange
CSP	CC-Link System Profile (see BAP-05028)
HDLC	High-level Data Link Control (see ISO/IEC 13239)
IAS	Industrial Automation Systems
I/O	Input and Output
LT	CC-Link/LT
OSI	Open System Interconnection
RWr	Remote register for read
RWw	Remote register for write
RX	Remote input
RY	Remote output
SEMI	Semiconductor Equipment and Materials International (in this context refer to SEMI standard, SEMI E54.12-0701)
UML	Unified Modeling Language (see UML V1.4)
XML	eXtensible Markup Language (see REC-xml-20040204)

5 Technology specific elements and rules

5.1 Integration models and IAS interfaces

The AIP developer shall develop the integration model using the rules described in ISO 15745-1, and shall ensure that the HDLC-based device and communication network profiles (whether representing the interface requirements or those derived from existing devices/communication networks) include the necessary IAS interfaces. The IAS interfaces included in the profile shall be identified in the header section (see ISO 15745-1:2003, 7.2.2).

NOTE IAS interfaces are described in ISO 15745-1:2003, Annex B.

5.2 Profile templates

5.2.1 General

The HDLC-based technology specific profile templates are derived from the generic profile templates specified in ISO 15745-1:2003, clause 7.

5.2.2 Contents and syntax

ISO 15745 specifies profile templates that are XML schemas (REC-xmlschema-1-20010502 and REC-xmlschema-2-20010502) and use a common general structure. The device and communication network profiles based on these templates typically contain:

- information needed to identify the connected device,
- a description of device data that can be accessed via the network,
- a description of the communication capabilities supported by the device,
- additional vendor-specific information.

However, CC-Link technology uses specific legacy ASCII syntax. Hence, for backward compatibility, template definitions (see Annex A) include the following:

- communication network and device profile templates, as defined in ISO 15745-1,
- ISO 15745 template to encapsulate files with legacy ASCII syntax ("wrapper").

5.2.3 Header

The profile template header defined in ISO 15745-1:2003, 7.2.2, is used for HDLC-based technology specific profile templates. Each technology uses one or more names to identify the technology or its particular component(s) (see Table 1). The selected name shall be stored in the ProfileTechnology attribute in the header section.

Table 1 — ProfileTechnology names

ProfileTechnology name	Technology
CC-Link	CC-Link
CSP	CC-Link

5.3 Technology specific profiles

The technology specific communication network profile structure and communication related aspects of device profile structure based on HDLC technology are described in clause 6. The technology included is CC-Link (see 6.2).

The related profile template definitions are specified in Annex A.

6 Device and communication network profiles for HDLC-based control systems

6.1 General

The CC-Link technology specific device and communication network profiles are described in 6.2.

6.2 CC-Link

6.2.1 Device profile

6.2.1.1 General

Figure 1 shows the class structure of the CC-Link device profile.

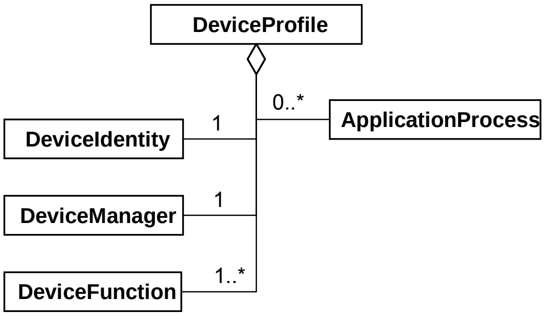

Figure 1 — CC-Link device profile class diagram

The DeviceProfile class of any version of CC-Link except version LT may include the ApplicationProcess class. The DeviceProfile class of CC-Link version LT shall not include the ApplicationProcess class.

The available formats for CC-Link device profiles are described in A.2.

The XML schema representing the CC-Link device profiles template is defined in A.2.1.3. The file name of this XML schema shall be “CC-Link_DeviceProfile.xsd”.

The XML schema representing the encapsulation of a CC-Link CSP into the ISO 15745 device profile template is defined in A.2.2.2. The file name of this XML schema shall be “CSP_DeviceProfile_wrapper.xsd”. The legacy CSP ASCII syntax itself is described in A.4.

6.2.1.2 Device identity

Figure 2 shows the class structure of the DeviceIdentity class.

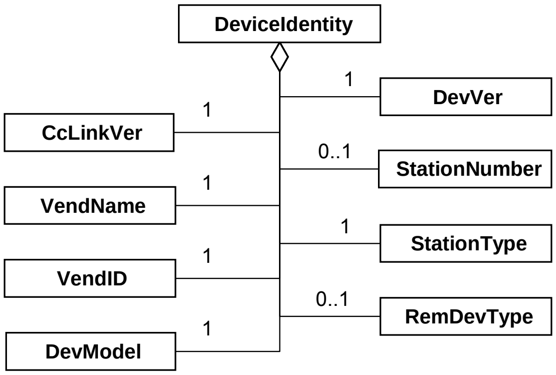

Figure 2 — CC-Link DeviceIdentity class diagram

The DeviceIdentity class shall consist of the child classes shown in Figure 2 and specified in Table 2. Child classes of DeviceIdentity class apply to profile, type, and instance.

Table 2 — Child classes of DeviceIdentity class

Class	Description
CcLinkVer	CC-Link version used by the device
VendName	vendor name
VendID	vendor code
DevModel	model name
DevVer	version number
StationNumber	Identifier of the device
StationType	type code for slave station
RemDevType	type code for remote device station

6.2.1.3 Device manager

6.2.1.3.1 General

Figure 3 shows the structure of the DeviceManager class.

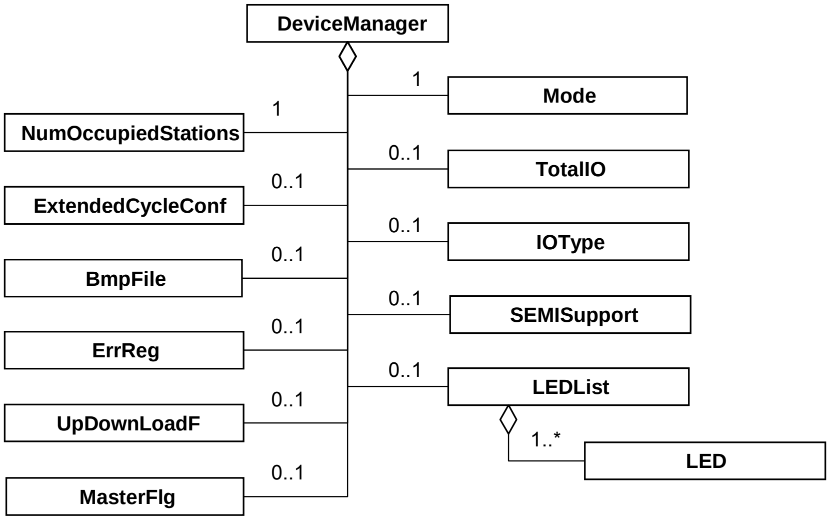

Figure 3 — CC-Link DeviceManager class diagram

The DeviceManager class consists of the child classes shown in Figure 3 and Table 3.

Table 3 — Child classes of DeviceManager class

Class	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
NumOccupiedStations	Mandatory	Mandatory	Mandatory	Mandatory
ExtendedCycleConf	Not applicable	Not applicable	Optional	Not applicable
BmpFile	Mandatory	Mandatory	Mandatory	Not applicable
ErrReg	Optional	Optional	Optional	Not applicable
UpDownLoadF	Mandatory	Mandatory	Mandatory	Not applicable
MasterFlg	Optional	Optional	Optional	Not applicable
Mode	Mandatory	Mandatory	Mandatory	Mandatory
TotalIO	Optional	Optional	Optional	Optional
IOType	Optional	Optional	Optional	Optional
SEMISupport	Not applicable	Optional	Optional	Not applicable
LEDList	Optional	Optional	Optional	Optional
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable for a device profile of a specified CC-Link version.				

6.2.1.3.2 NumOccupiedStations

NumOccupiedStations is the number of occupied stations.

6.2.1.3.3 ExtendedCycleConf

ExtendedCycleConf is the extension level of an amount of I/O points used in the extended cyclic communication.

6.2.1.3.4 BmpFile

BmpFile is the name of the associated bitmap file.

6.2.1.3.5 ErrReg

ErrReg is a register used to store an error code.

6.2.1.3.6 UpDownLoadF

UpDownLoadF indicates whether an upload and download is applicable or not.

6.2.1.3.7 MasterFlg

MasterFlg indicates whether the device can be a standby master or not.

6.2.1.3.8 Mode

Mode indicates supporting operation modes (see Table A.3).

6.2.1.3.9 TotalIO

TotalIO is the total number of I/O points (see Table A.4).

6.2.1.3.10 IOType

IOType is the type of I/O (see Table A.5).

6.2.1.3.11 SEMISupport

SEMISupport indicates whether the SEMI standard function is supported or not.

6.2.1.3.12 LEDList, LED

The LEDList is a listing of available LEDs. The LED describes the name and the possible state of the LED on the device (see Table A.6).

6.2.1.4 Device function

Figure 4 shows the structure of the DeviceFunction class.

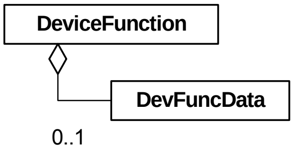

Figure 4 — CC-Link DeviceFunction class diagram

NOTE The definition of the DevFuncData of the DeviceFunction is outside the scope of this part of ISO 15745.

6.2.1.5 Application process

6.2.1.5.1 General

Figure 5 shows the class structure of the ApplicationProcess class.

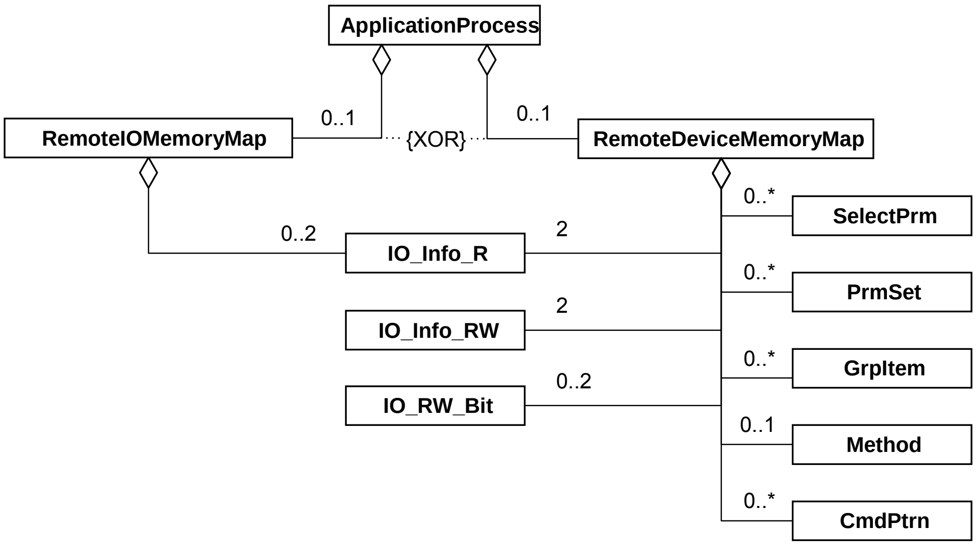

Figure 5 — CC-Link ApplicationProcess class diagram

6.2.1.5.2 RemoteIOMemoryMap

6.2.1.5.2.1 General

RemoteIOMemoryMap is a CSP profile equivalent for a remote I/O device. The CSP profile is described in A.4.

6.2.1.5.2.2 IO_Info_R

Figure 6 shows the class structure of the IO_Info_R class.

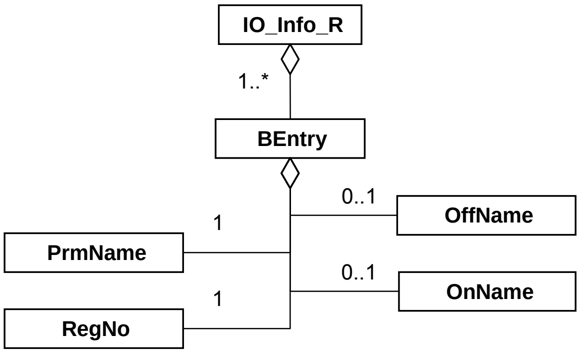

Figure 6 — CC-Link IO_Info_R class diagram

The IO_Info_R represents IO_Info_RX section and IO_Info_RY section of the CSP profile. A BEntry shall contain the child classes shown in Figure 6 and A.2.1.2.4.1.

6.2.1.5.3 RemoteDeviceMemoryMap

6.2.1.5.3.1 General

RemoteDeviceMemoryMap is a CSP profile equivalent for a remote device.

6.2.1.5.3.2 IO_Info_R

The class structure of the IO_Info_R class is described in 6.2.1.5.2.2.

6.2.1.5.3.3 IO_Info_RW

Figure 7 shows the class structure of IO_Info_RW class.

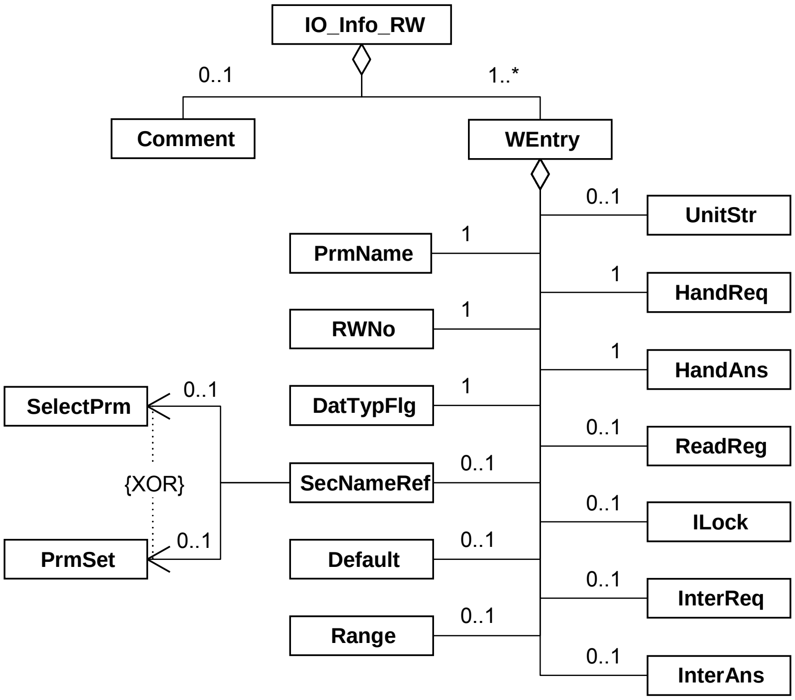

Figure 7 — CC-Link IO_Info_RW class diagram

Comment class is used to annotate. A WEntry contains the child classes shown in Figure 7 and A.2.1.2.4.2. A WEntry represents parameters of a word register.

6.2.1.5.3.4 IO_RW_Bit

Figure 8 shows the class structure of IO_RW_Bit class.

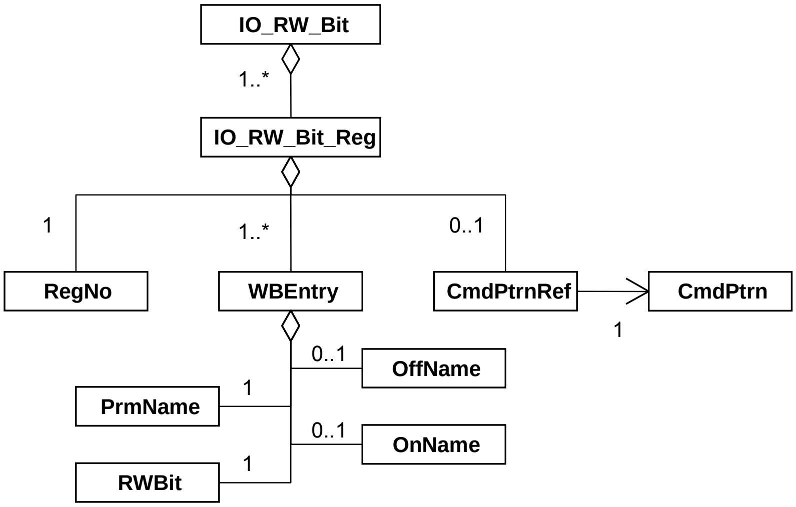

Figure 8 — CC-Link IO_RW_Bit class diagram

IO_RW_Bit_Reg class is a container of the associated subclasses. RegNo class represents register number. WBEntry class contains the child classes shown in Figure 8 and A.2.1.2.4.3. WBEntry class represents parameters of bits used in a word register. CmdPtrnRef is a reference to CmdPtrn.

6.2.1.5.3.5 SelectPrm

Figure 9 shows the class structure of SelectPrm class.

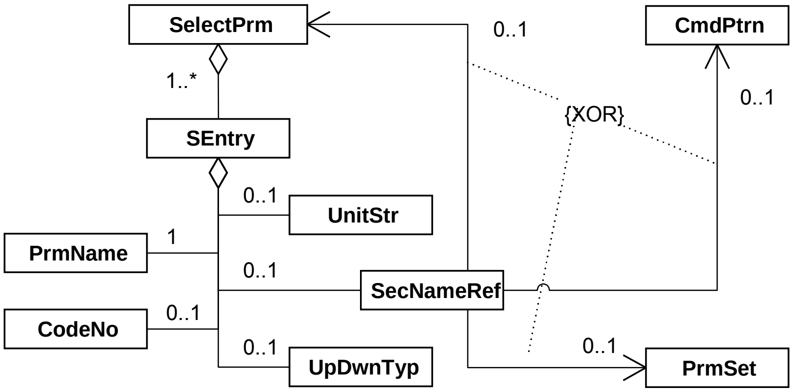

Figure 9 — CC-Link SelectPrm class diagram

SEntry class contains the child classes shown in Figure 9 and A.2.1.2.4.4. A SEntry represents an entry of the "Select parameter section" defined in a CSP profile.

6.2.1.5.3.6 PrmSet

Figure 10 shows the class structure of PrmSet class.

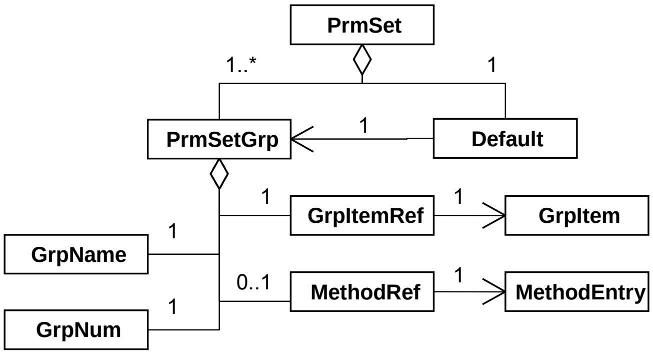

Figure 10 — CC-Link PrmSet class diagram

PrmSetGrp class contains the child classes shown in Figure 10 and A.2.1.2.4.5. PrmSetGrp represents an entry of the parameter set section defined in a CSP profile. The Default class is a reference to the default PrmSetGrp.

6.2.1.5.3.7 GrpItem

Figure 11 shows the class structure of GrpItem class.

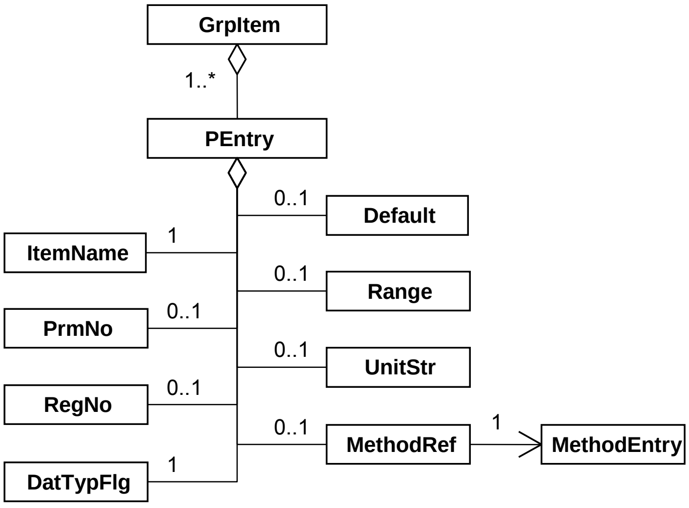

Figure 11 — CC-Link GrpItem class diagram

A PEntry contains the child classes shown in Figure 11 and A.2.1.2.4.6. PEntry represents the items of a parameter entry defined in a CSP profile.

6.2.1.5.3.8 Method

Figure 12 shows the class structure of Method class.

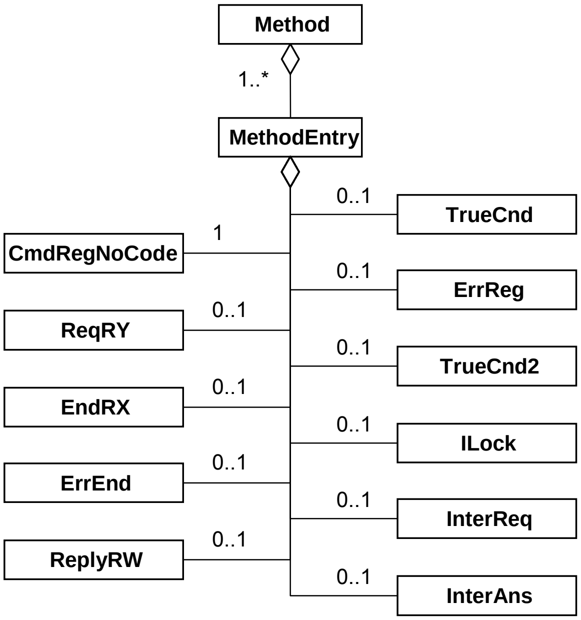

Figure 12 — CC-Link Method class diagram

MethodEntry contains the child classes shown in Figure 12 and A.2.1.2.4.7. A MethodEntry represents a method defined in a CSP profile.

6.2.1.5.3.9 CmdPtrn

Figure 13 shows the class structure of CmdPtrn class.

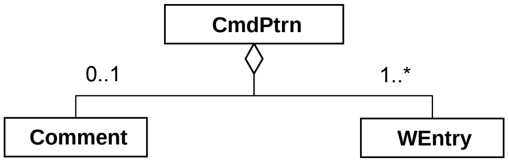

Figure 13 — CC-Link CmdPtrn class diagram

Comment class is used to annotate. A WEntry contains the child classes shown in Figure 7 and A.2.1.2.4.2. A WEntry represents parameters of a word register.

6.2.2 Communication network profile

6.2.2.1 General

Figure 14 shows the class structure of the CC-Link communication network profile.

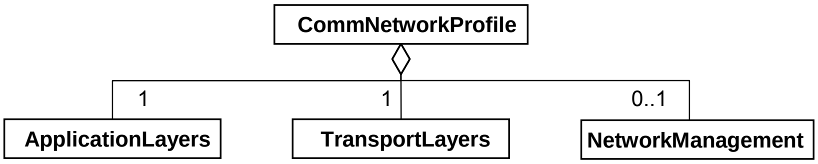

Figure 14 — CC-Link communication network profile class diagram

The available formats for CC-Link communication network profiles are described in A.3.

The XML schema representing the CC-Link communication network profile template is defined in A.3.1.3. The file name of this XML schema shall be “CC-Link_CommNetworkProfile.xsd.”

The XML schema representing the encapsulation of a CC-Link CSP file into the ISO 15745 communication network profile template is defined in A.3.2.2. The file name of this XML schema shall be "CSP_CommNetworkProfile_wrapper.xsd."

6.2.2.2 Application layers

The ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the communication network integration model. It states the supported application service elements and their associate services. ApplicationLayers class consists of the child classes shown in Figure 15 and Table 4.

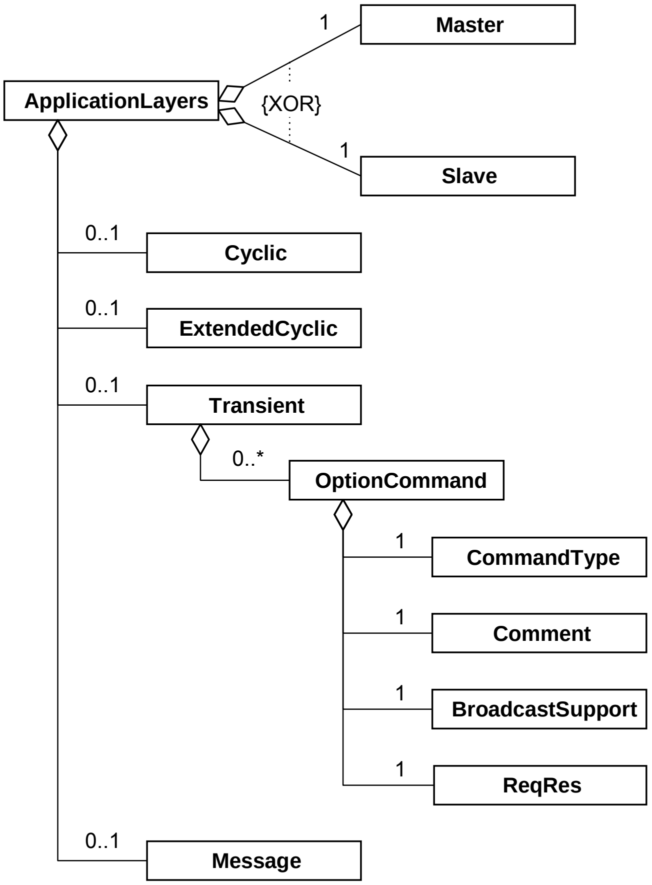

Figure 15 — CC-Link ApplicationLayers class diagram

Table 4 — Constraints for child classes of ApplicationLayers class

Class	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
Master	Mandatory	Mandatory	Mandatory	Mandatory
Slave	Mandatory	Mandatory	Mandatory	Mandatory
Cyclic	Mandatory	Mandatory	Optional	Mandatory
ExtendedCyclic	Not applicable	Not applicable	Mandatory	Not applicable
Transient	Optional	Optional	Optional	Not applicable
Message	Not applicable	Mandatory	Optional	Not applicable
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

The child classes of ApplicationLayers represent the following:

- Master indicates that the device is a master;
- Slave indicates that the device is a slave;

- Cyclic indicates that the device supports cyclic communication;
- ExtendedCyclic indicates that the device supports extended cyclic communication;
- Transient indicates whether or not the device supports transient communication and documents optional commands defined by the vendor;
- Message indicates whether or not the device supports message communication.

6.2.2.3 Transport layers

6.2.2.3.1 General

The TransportLayers class represents the combined profiles for the lower 4 OSI layers of the CC-Link communication network integration model. TransportLayers class consists of the child classes shown in Figure 16.

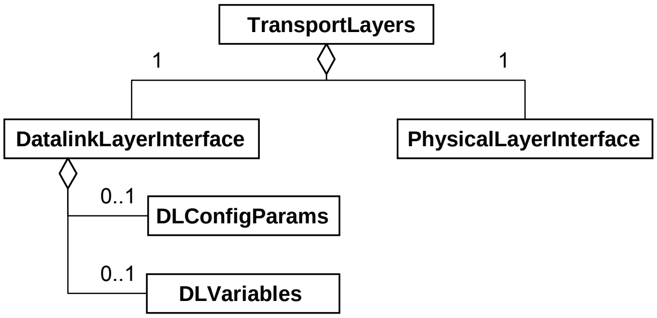

Figure 16 — CC-Link TransportLayers class diagram

6.2.2.3.2 PhysicalLayerInterface

The PhysicalLayerInterface class defines the properties associated with the physical layer of CC-Link.

6.2.2.3.3 DatalinkLayerInterface

The DatalinkLayerInterface class defines the properties associated with the data link layer of CC-Link. The child classes of DatalinkLayerInterface are represented as follows:

- DLConfigParams defines the properties associated with data link layer configuration;
- DLVariables defines the properties associated with data link layer monitoring.

6.2.2.4 Network management

The NetworkManagement class represents the properties relating to the network management of the CC-Link communication network integration model. The NetworkManagement class consists of the child classes shown in Figure 17 and Table 5.

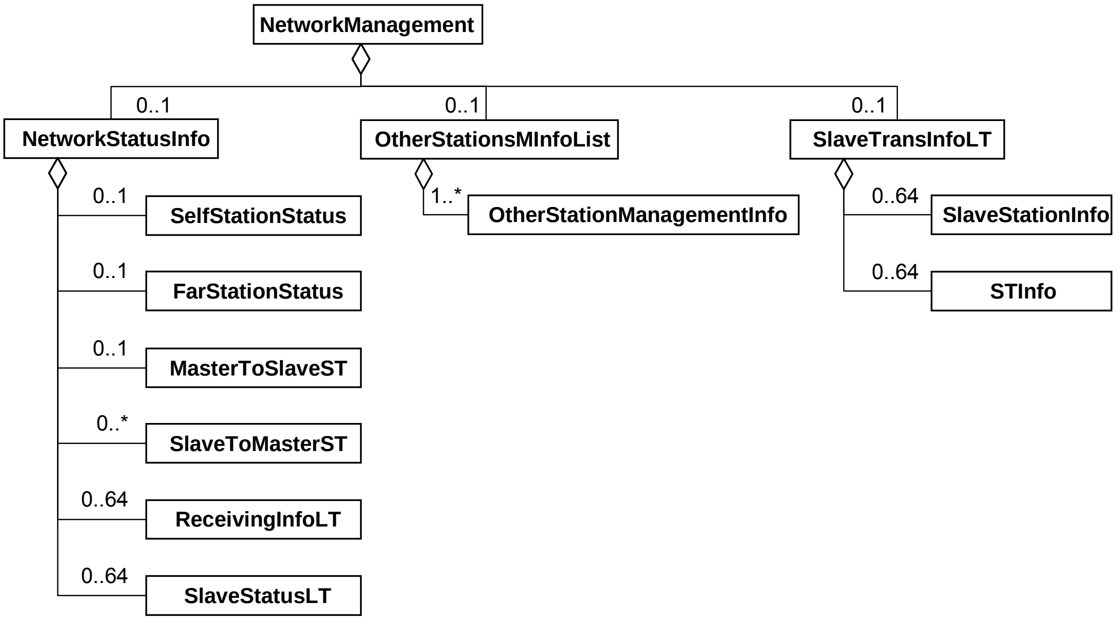

Figure 17 — CC-Link NetworkManagement class diagram

Table 5 — Constraints of the child classes of CC-Link NetworkManagement class

class	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
NetworkStatusInfo	Optional	Optional	Optional	Optional
OtherStationsMInfoList	Optional	Optional	Optional	Not applicable
SlaveTransInfoLT	Not applicable	Not applicable	Not applicable	Optional
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

The child classes of NetworkManagement class are as follows:

— NetworkStatusInfo and the associated child classes define the properties associated with status of CC-Link network;

— OtherStationsMInfoList and OtherStationManagementInfo define the properties associated with the other stations;

— SlaveTransInfoLT and the associated child classes define the properties associated with the transmission information of slave stations.

Annex A
(normative)

CC-Link profile templates

A.1 General

Configuration tools currently available for CC-Link devices use a specially formatted ASCII file, referred to as the CSP (CC-Link System Profile). The CSP provides information about a device, such as:

- information needed to identify the connected device,
- a description of device data that can be accessed via the network (e.g. configurable parameters),
- a description of the hardwired connection capabilities supported by the device (e.g. connections),
- additional vendor-specific information.

The CSP allows a configuration tool to automate the device configuration process. The CSP requirements provide an open, consistent and compatible approach for performing device configuration

The CSP information is very similar to the information required in both communication network and device profiles, hence, A.2, A.3, and A.4 specify respectively the format for:

- communication network and device profile templates, as defined in ISO 15745-1:2003,
- encapsulation of legacy CSP files in the ISO 15745 templates (“wrappers”), and
- the legacy CSP files format, including common semantics information.

A.2 Device profile template description

A.2.1 Device profile template description – XML based

A.2.1.1 General

The device profile XML file shall comply with the device profile XML schema as specified in A.2.1.3.

Contents of this XML schema are derived from the device profile class diagrams shown in 6.2.1.

A.2.1.2 Semantics of XML schema elements

A.2.1.2.1 ProfileBody

This main element is associated with a set of attributes which provide additional information about the profile file. The attributes are shown in Table A.1. The semantics of the attributes are described in Table A.35.

Table A.1 — Attributes of ProfileBody

Attribute	Description	Constraint
FileComment	File comment	Optional
CreateDate	File creation date	Required
CreateTime	File creation time	Required
ModDate	Last modified date	Optional
ModTime	Last modified time	Optional
Version	CSP file version	Required

A.2.1.2.2 DeviceIdentity

A.2.1.2.2.1 CcLinkVer

This element specifies a CC-Link version used by the device.

A.2.1.2.2.2 VendName

This element specifies a vendor name of the device (see Table A.37).

A.2.1.2.2.3 VendID

This element specifies a vendor ID of the device (see Vendor identification in Table A.37).

A.2.1.2.2.4 DevModel

This element specifies a model name of the device (see Device model name in Table A.37).

A.2.1.2.2.5 DevVer

This element specifies a version number of the device (see Version in Table A.37).

A.2.1.2.2.6 StationNumber

This element specifies a unique identifier of the device among the devices which are connected to each other in a CC-Link network.

A.2.1.2.2.7 StationType

This element specifies a type of slave station (see Station type in Table A.37).

A.2.1.2.2.8 RemDevType

This element specifies a type of remote device station (see Remote device type in Table A.37 and Table A.38).

A.2.1.2.3 DeviceManager

A.2.1.2.3.1 NumOccupiedStations

This element specifies the number of network visible stations occupied by a device.

A.2.1.2.3.2 ExtendedCycleConf

This element specifies the multiplicity of an amount of I/O points in the extended cyclic communication. Table A.2 shows the semantics of ExtendedCycleConf.

Table A.2 — ExtendedCycleConf element

Value	Description
0	Single
1	Double
2	Quadruple
3	Octuple

A.2.1.2.3.3 BmpFile

This element specifies an associated bitmap file name.

A.2.1.2.3.4 ErrReg

This element specifies a storage register used to store an error code.

A.2.1.2.3.5 UpDownLoadF

The element is a flag which indicates whether the CSP file supports upload/download.

The value 0 means unsupported and the value 1 means supported.

A.2.1.2.3.6 MasterFlg

The element is a flag to indicate whether the device can be a standby master station or not. An intelligent device type slave device can also have the capability of being a standby master.

The value 0 means that the device is not capable of being a standby master station. The value 1 means that the device is capable of being a standby master station.

A.2.1.2.3.7 Mode

This element specifies an operation mode of the device. Table A.3 shows the semantics of Mode.

Table A.3 — Mode element

Value	Description
“Online”	Datalink is working normally.
“LineTest1”	Checking whether the master device communicates normally with all slave devices.
“LineTest2”	Checking whether the master device communicates normally with a specific slave device.
“ParamCheckTest”	Checking the content of the parameters
“HardwareTest”	Checking whether or not hardware works properly.

A.2.1.2.3.8 TotalIO

This element specifies the number of I/O points. Table A.4 shows the semantics of TotalIO.

Table A.4 — TotalIO element

Value	Description
0	Dependent on the number of occupied stations
1	8 points
2	32 points
3	16 points

A.2.1.2.3.9 IOType

The element specifies the type of I/O. Table A.5 shows the semantics of IOType.

Table A.5 — IOType element

Value	Description
0	The device uses both inputs and outputs and the same numbers are used for both inputs and outputs (starting from RX0 and RY0, respectively)
1	Input
2	Output
3	The device uses both inputs and outputs but the same numbers are not used for input (RX) and output (RY).

A.2.1.2.3.10 SEMISupport

This element whose type is Boolean specifies whether the SEMI standard function is supported or not. The value TRUE represents that the SEMI function is supported and the value FALSE represents the function is not supported.

A.2.1.2.3.11 LEDList, LED

Table A.6 describes the attributes of the LED class.

NOTE The semantics of the values are defined in BAP-05027. The semantics of the values for Version LT are defined in BAP-05031.

Table A.6 — Attributes of LED class

Attribute	Description	Values
Name	Name of LED	“PW” “RUN” “ERR.” “MST” “S MST” “LOCAL” “CPU R/W” “SW” “M/S” “PRM” “TIME” “LINE” “L RUN” “L ERR.” “SD” “RD”
State	State of LED	“ON” “OFF” “BLINK”

A.2.1.2.4 ApplicationProcess

A.2.1.2.4.1 BEntry

BEntry elements are specified in Table A.7.

Table A.7 — BEntry elements

Element	Description
PrmName	Name of the signal
RegNo	Register number
OffName	Label of the parameter of which value is 0
OnName	Label of the parameter of which value is 1

A.2.1.2.4.2 WEntry

WEntry elements are specified in Table A.8.

Table A.8 — WEntry elements

Element	Description
PrmName	Name of the signal
RWNo	Register number
DatTypFlg	Data type
SecNameRef	Reference to SelectPrm, PrmSet or nothing
Default	Default value
Range	Range of the value of the parameter
UnitStr	The unit of the parameter
HandReq	RY Handshake device name and its status
HandAns	RX Handshake device name and its status
ReadReg	Read word register
ILock	Interlock
InterReq	Inter-sequence request register
InterAns	Inter-sequence answer register

A.2.1.2.4.3 WBEntry

WBEntry elements are specified in Table A.9.

Table A.9 — WBEntry elements

Element	Description
PrmName	Name of the signal
RWBit	Bit register number
OffName	Label of the parameter of which value is 0
OnName	Label of the parameter of which value is 1

A.2.1.2.4.4 SEntry

SEntry elements are specified in Table A.10.

Table A.10 — SEntry elements

Element	Description
PrmName	Name of the signal
CodeNo	Code number
UnitStr	The unit of the parameter
SecNameRef	Reference to SelectPrm, PrmSet, CmdPtrn or nothing
UpDwnTyp	Upload/download type

A.2.1.2.4.5 PrmSetGrp

PrmSetGrp elements are specified in Table A.11.

Table A.11 — PrmSetGrp elements

Element	Description
GrpName	Group name
GrpNum	Number of group items
GrpItemRef	Reference to GrpItem
MethodRef	Reference to MethodEntry

A.2.1.2.4.6 PEntry

PEntry elements are specified in Table A.12.

Table A.12 — PEntry elements

Element	Description
ItemName	Item name
PrmNo	Parameter number
RegNo	Register number
DatTypFlg	Data type
Default	Default value
Range	Setting range
UnitStr	The unit of the parameter
MethodRef	Reference to MethodEntry

A.2.1.2.4.7 MethodEntry

MethodEntry elements are specified in Table A.13.

Table A.13 — MethodEntry elements

Element	Description
CmdRegNoCode	Command setting register and command code
ReqRY	Command operation request
EndRX	Command operation complete
ErrEnd	Command operation abnormal complete
ReplyRW	Reply code storage register
TrueCnd	Condition to be used for evaluating a replied value
ErrReg	Error code storage register
TrueCnd2	Condition to be used for further evaluation of the device condition
ILock	Interlock
InterReq	Inter-sequence request register
InterAns	Inter-sequence answer register

A.2.1.3 XML schema : CC-Link_DeviceProfile.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
targetNamespace="http://www.cc-link.org/ISO15745/DeviceProfile"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns="http://www.cc-link.org/ISO15745/DeviceProfile"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>


   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>

   </xsd:element>


   <xsd:annotation>
      <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
   </xsd:annotation>

   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP" />
         <xsd:enumeration value="Process" />
         <xsd:enumeration value="InformationExchange" />
         <xsd:enumeration value="Resource" />
         <xsd:enumeration value="Device" />
         <xsd:enumeration value="CommunicationNetwork" />
         <xsd:enumeration value="Equipment" />
         <xsd:enumeration value="Human" />
         <xsd:enumeration value="Material" />
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
         <xsd:element name="ProfileTechnology" type="xsd:string" />
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI" />
               <xsd:enumeration value="HCI" />
               <xsd:enumeration value="ISI" />
               <xsd:enumeration value="API" />
               <xsd:enumeration value="CMI" />
               <xsd:enumeration value="ESI" />
               <xsd:enumeration value="FSI" />
               <xsd:enumeration value="MTI" />
               <xsd:enumeration value="SEI" />
               <xsd:enumeration value="USI" />
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4" />
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>


   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceIdentity"/>
            <xsd:element ref="DeviceManager"/>
            <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
            <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attribute name="FileComment" type="xsd:string" use="optional"/>
         <xsd:attribute name="CreateDate" type="xsd:date" use="required"/>
         <xsd:attribute name="CreateTime" type="xsd:time" use="required"/>
         <xsd:attribute name="ModDate" type="xsd:date" use="optional"/>
         <xsd:attribute name="ModTime" type="xsd:time" use="optional"/>
         <xsd:attribute name="Version" type="xsd:string" use="required"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:annotation>
      <xsd:documentation>* ISO 15745 CC-LINK DEFINED ELEMENTS *</xsd:documentation>
   </xsd:annotation>

   <!-- DeviceProfile -->

   <xsd:element name="DeviceIdentity" type="DeviceIdentityTYPE"/>
   <xsd:element name="DeviceManager" type="DeviceManagerTYPE"/>
   <xsd:element name="DeviceFunction" type="DeviceFunctionTYPE"/>
```

```
<xsd:element name="ApplicationProcess" type="ApplicationProcessTYPE"/>

<!-- DeviceIdentity -->

<xsd:element name="CcLinkVer">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="8"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="VendName">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="65"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="VendID">
   <xsd:simpleType>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="4"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="DevModel">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="65"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="DevVer">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="8"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="StationNumber" nillable="true">
   <xsd:simpleType>
      <xsd:restriction base="xsd:short">
         <xsd:minInclusive value="0"/>
         <xsd:maxInclusive value="64"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="StationType" type="xsd:short">
   <xsd:annotation>
      <xsd:documentation>
         0: Remote I/O station
         1: Remote device station
         2: Intelligent device station (includes local stations and standby master station)
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>

<xsd:element name="RemDevType">
   <xsd:annotation>
      <xsd:documentation>
         1: PLC
         2: Personal computer
         3: Digital I/O
         4: Analog I/O
         5: Positioning
         6: Thermostat
         7: HMI
         8: ID
         09: Serial converting device
         1D: CC-Link - CC-Link/LT bridge
         1F: Protocol converting device
         20: Inverter
         21: Servo
         22: NC
         23: Robot
         24: Power distribution control device
         30: Sensor
         31: Actuator
         32: Barcode
         33: Indicator (weight)
         34: High-speed counter
         35: Key switch
         36: Protocol analyzer
         37: Aerial transmitter
         38: Conveyor control device
         39: Power supply control device
         3A: Welding machine control device
         3B: Gas detector
         3C: Solenoid valve
         3D: Robot (multi-purpose I/O)
         3E: Printer control device
         3F: Motor control device
         40: Vacuum pump
         41: Multi-axis controller
         42: Multi-purpose VME board
         43: Power supply (for mass flow controller)
         44: Mass flow controller
         45: Power reception and distribution device
         46: Control centre
         47: Welding control device
         48: Indicator (multi-purpose)
         49: PID controller
         4A: Vacuum gauge
         4B: Wireless device
         4C: Digital / Analog I/O
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="2"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>


<!-- DeviceManager -->
<xsd:element name="NumOccupiedStations">
   <xsd:simpleType>
      <xsd:restriction base="xsd:short">
         <xsd:minInclusive value="1"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="ExtendedCycleConf">
   <xsd:annotation>
      <xsd:documentation>
         0: Single
         1: Double
         2: Quadruple
         3: Octuple
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="1"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="BmpFile">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="257"/>
      </xsd:restriction>
   </xsd:simpleType>
```

```
</xsd:element>

<xsd:element name="ErrReg">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="8"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="UpDownLoadF" type="xsd:short">
   <xsd:annotation>
      <xsd:documentation>
         0: not supported
         1: supported
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>

<xsd:element name="MasterFlg" type="xsd:short">
   <xsd:annotation>
      <xsd:documentation>
         0: not capable of being a standby master station
         1: capable of being a standby master station
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>

<xsd:element name="Mode">
   <xsd:simpleType>
      <xsd:list>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="Online"/>
               <xsd:enumeration value="LineTest1"/>
               <xsd:enumeration value="LineTest2"/>
               <xsd:enumeration value="ParamCheckTest"/>
               <xsd:enumeration value="HardwareTest"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:list>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="TotalIO">
   <xsd:annotation>
      <xsd:documentation>
         0: Dependent on the number of occupied stations
         1: 8 points
         2: 32 points
         3: 16 points
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="1"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

<xsd:element name="IOType">
   <xsd:annotation>
      <xsd:documentation>
         0: Both front and back are mixed
         1: Input
         2: Output
         3: Mixed
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="1"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
```

```
<xsd:element name="SEMISupport" type="xsd:boolean"/>

<xsd:element name="LEDList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="LED" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>

<xsd:element name="LED">
   <xsd:complexType>
      <xsd:attribute name="Name" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="PW"/>
               <xsd:enumeration value="RUN"/>
               <xsd:enumeration value="ERR."/>
               <xsd:enumeration value="MST"/>
               <xsd:enumeration value="S MST"/>
               <xsd:enumeration value="LOCAL"/>
               <xsd:enumeration value="CPU R/W"/>
               <xsd:enumeration value="SW"/>
               <xsd:enumeration value="M/S"/>
               <xsd:enumeration value="PRM"/>
               <xsd:enumeration value="TIME"/>
               <xsd:enumeration value="LINE"/>
               <xsd:enumeration value="L RUN"/>
               <xsd:enumeration value="L ERR."/>
               <xsd:enumeration value="SD"/>
               <xsd:enumeration value="RD"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="State" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="ON"/>
               <xsd:enumeration value="OFF"/>
               <xsd:enumeration value="BLINK"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>


<!-- ApplicationProcess -->
<xsd:element name="RemoteIOMemoryMap">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="IO_Info_R" minOccurs="0" maxOccurs="2"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:key name="Key_RemoteIOMemoryMap_IO_Info_R">
      <xsd:selector xpath="IO_Info_R"/>
      <xsd:field xpath="@CSPSecName"/>
   </xsd:key>
</xsd:element>

<xsd:element name="RemoteDeviceMemoryMap">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="IO_Info_R" minOccurs="2" maxOccurs="2"/>
         <xsd:element ref="IO_Info_RW" minOccurs="2" maxOccurs="2"/>
         <xsd:element ref="IO_RW_Bit" minOccurs="0" maxOccurs="2"/>
         <xsd:element ref="SelectPrm" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="PrmSet" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="GrpItem" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="Method" minOccurs="0"/>
         <xsd:element ref="CmdPtrn" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:key name="Key_SelectPrm_PrmSet">
      <xsd:selector xpath="SelectPrm|PrmSet"/>
      <xsd:field xpath="@id"/>
          </xsd:key>
          <xsd:keyref name="Keyref_SelectPrm_PrmSet" refer="Key_SelectPrm_PrmSet">
             <xsd:selector xpath="IO_Info_RW/WEntry/SecNameRef|CmdPtrn/WEntry/SecNameRef"/>
             <xsd:field xpath="@ref"/>
          </xsd:keyref>
          <xsd:key name="Key_CmdPtrn">
             <xsd:selector xpath="CmdPtrn"/>
             <xsd:field xpath="@id"/>
          </xsd:key>
          <xsd:keyref name="Keyref_CmdPtrn" refer="Key_CmdPtrn">
             <xsd:selector xpath="IO_RW_Bit/CmdPtrnRef"/>
             <xsd:field xpath="@ref"/>
          </xsd:keyref>
          <xsd:key name="Key_SelectPrm_PrmSet_CmdPtrn">
             <xsd:selector xpath="SelectPrm|PrmSet|CmdPtrn"/>
             <xsd:field xpath="@id"/>
          </xsd:key>
          <xsd:keyref name="Keyref_SelectPrm_PrmSet_CmdPtrn" refer="Key_SelectPrm_PrmSet_CmdPtrn">
             <xsd:selector xpath="SelectPrm/SecNameRef"/>
             <xsd:field xpath="@ref"/>
          </xsd:keyref>
          <xsd:key name="Key_GrpItem">
             <xsd:selector xpath="GrpItem"/>
             <xsd:field xpath="@id"/>
          </xsd:key>
          <xsd:keyref name="Keyref_GrpItem" refer="Key_GrpItem">
             <xsd:selector xpath="PrmSet/PrmSetGrp/GrpItemRef"/>
             <xsd:field xpath="@ref"/>
          </xsd:keyref>
          <xsd:key name="Key_RemoteDeviceMemoryMap_IO_Info_R">
             <xsd:selector xpath="IO_Info_R"/>
             <xsd:field xpath="@CSPSecName"/>
          </xsd:key>
          <xsd:key name="Key_IO_Info_RW">
             <xsd:selector xpath="IO_Info_RW"/>
             <xsd:field xpath="@CSPSecName"/>
          </xsd:key>
          <xsd:key name="Key_IO_RW_Bit">
             <xsd:selector xpath="IO_RW_Bit"/>
             <xsd:field xpath="@CSPSecName"/>
          </xsd:key>
          <xsd:key name="Key_MethodEntry">
             <xsd:selector xpath="Method/MethodEntry"/>
             <xsd:field xpath="@id"/>
          </xsd:key>
          <xsd:keyref name="Keyref_MethodEntry" refer="Key_MethodEntry">
             <xsd:selector xpath="PrmSet/PrmSetGrp/MethodRef|GrpItem/Pentry/MethodRef"/>
             <xsd:field xpath="@ref"/>
          </xsd:keyref>
       </xsd:element>

       <!-- ApplicationProcess - IO_Info_R -->
       <xsd:element name="IO_Info_R">
          <xsd:complexType>
             <xsd:sequence>
                <xsd:element ref="BEntry" maxOccurs="unbounded"/>
             </xsd:sequence>
             <xsd:attribute name="CSPSecName" use="required">
                <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="IO_Info_RX"/>
                      <xsd:enumeration value="IO_Info_RY"/>
                   </xsd:restriction>
                </xsd:simpleType>
             </xsd:attribute>
             <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
          </xsd:complexType>
          <xsd:key name="Key_BEntry">
             <xsd:selector xpath="BEntry"/>
             <xsd:field xpath="@id"/>
          </xsd:key>
       </xsd:element>

       <xsd:element name="BEntry">
          <xsd:complexType>
             <xsd:sequence>
```

```
            <xsd:element name="PrmName">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="RegNo">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="OffName" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="OnName" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
         </xsd:sequence>
         <xsd:attribute name="id" type="xsd:string" use="required"/>
      </xsd:complexType>
   </xsd:element>

   <!-- ApplicationProcess - IO_Info_RW -->
   <xsd:element name="IO_Info_RW">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Comment" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:sequence>
               <xsd:element ref="WEntry" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:sequence>
         <xsd:attribute name="CSPSecName" use="required">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="IO_Info_RWw"/>
                  <xsd:enumeration value="IO_Info_RWr"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:attribute>
         <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
      </xsd:complexType>
      <xsd:key name="Key_IO_Info_RW_WEntry">
         <xsd:selector xpath="WEntry"/>
         <xsd:field xpath="@id"/>
      </xsd:key>
   </xsd:element>

   <xsd:element name="WEntry">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="PrmName">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="RWNo">
   <xsd:simpleType>

      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="129"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
<xsd:element name="DatTypFlg" type="xsd:unsignedShort">
   <xsd:annotation>
      <xsd:documentation>
         0: Used as 16-bit unsigned hexadecimal data.
         1: Used as 16-bit signed decimal data.
         2: Used as 16-bit unsigned decimal data.
         3: Used as bit data.
         4: Used as 8-bit unsigned hexadecimal data. (Upper 8 bits)
         5: Used as 8-bit unsigned hexadecimal data. (Lower 8 bits)
         6: Used as 32-bit signed decimal data.
         7: Used as 32-bit unsigned decimal data.
         8: Used as 32-bit unsigned hexadecimal data.
         9: Used as a floating point value (32-bit data).
         10: Used as 4-bit unsigned hexadecimal data 1).
         11: Used as 4-bit unsigned hexadecimal data 2).
         12: Used as 4-bit unsigned hexadecimal data 3).
         13: Used as 4-bit unsigned hexadecimal data 4).
         14: Byte array
         15: Word array
         16: Dummy data type
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>
<xsd:element name="SecNameRef" minOccurs="0">
   <xsd:complexType>
      <xsd:attribute name="ref" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="Default" minOccurs="0">
   <xsd:simpleType>
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:integer"/>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:hexBinary"/>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:float"/>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
</xsd:element>
<xsd:element name="Range" minOccurs="0">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="129"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
<xsd:element name="UnitStr" minOccurs="0">
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="17"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
<xsd:element name="HandReq">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:restriction base="StringWithOptOnOffTYPE">
            <xsd:maxLength value="9"/>
         </xsd:restriction>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="HandAns">
   <xsd:complexType>
      <xsd:simpleContent>
```

```
                  <xsd:restriction base="StringWithOptOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ReadReg" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="8"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="ILock" type="ILockTYPE" minOccurs="0"/>
         <xsd:element name="InterReq" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="InterAns" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>

<!-- ApplicationProcess - IO_RW_Bit -->
<xsd:element name="IO_RW_Bit">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="IO_RW_Bit_Reg" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="CSPSecName" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="IO_RWw_Bit"/>
               <xsd:enumeration value="IO_RWr_Bit"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>

<xsd:element name="IO_RW_Bit_Reg">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="CmdPtrnRef" minOccurs="0">
            <xsd:complexType>
               <xsd:attribute name="ref" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="RegNo">
            <xsd:simpleType>
               <xsd:restriction base="xsd:hexBinary">
                  <xsd:maxLength value="1"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:sequence>
            <xsd:element ref="WBEntry" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:sequence>
      <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
   </xsd:complexType>
   <xsd:key name="Key_WBEntry">
```

```
      <xsd:selector xpath="WBEntry"/>
      <xsd:field xpath="@id"/>
   </xsd:key>
</xsd:element>

<xsd:element name="WBEntry">

   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="PrmName">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="RWBit">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="9"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="OffName" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="OnName" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>

<!-- ApplicationProcess - SelectPrm -->
<xsd:element name="SelectPrm">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="SEntry" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
      <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
   </xsd:complexType>
   <xsd:key name="Key_SEntry">
      <xsd:selector xpath="SEntry"/>
      <xsd:field xpath="@id"/>
   </xsd:key>
</xsd:element>

<xsd:element name="SEntry">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="PrmName">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="CodeNo" type="xsd:hexBinary" minOccurs="0"/>
         <xsd:element name="UnitStr" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="17"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="SecNameRef" minOccurs="0">
```

```
            <xsd:complexType>
               <xsd:attribute name="ref" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="UpDwnTyp" type="xsd:short" minOccurs="0">
            <xsd:annotation>
               <xsd:documentation>
                  0: Not supproted
                  1: Upload
                  2: Download
                  3: Both upload and download are supported
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
      </xsd:sequence>

      <xsd:attribute name="id" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>

<!-- ApplicationProcess - PrmSet -->
<xsd:element name="PrmSet">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Default">
            <xsd:complexType>
               <xsd:attribute name="ref" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:sequence>
            <xsd:element ref="PrmSetGrp" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
      <xsd:attribute name="GrpNum" type="xsd:short" use="required"/>
   </xsd:complexType>
   <xsd:key name="Key_PrmSetGrp">
      <xsd:selector xpath="PrmSetGrp"/>
      <xsd:field xpath="@id"/>
   </xsd:key>
   <xsd:keyref name="Keyref_PrmSetGrp" refer="Key_PrmSetGrp">
      <xsd:selector xpath="Default"/>
      <xsd:field xpath="@ref"/>
   </xsd:keyref>
</xsd:element>

<xsd:element name="PrmSetGrp">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="GrpName">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="GrpNum" type="xsd:short"/>
         <xsd:element name="GrpItemRef">
            <xsd:complexType>
               <xsd:attribute name="ref" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="MethodRef" minOccurs="0">
            <xsd:complexType>
               <xsd:attribute name="ref" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>

<!-- ApplicationProcess - GrpItem -->
<xsd:element name="GrpItem">
   <xsd:complexType>
      <xsd:sequence>
```

```
            <xsd:element ref="PEntry" maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attribute name="id" type="xsd:string" use="required"/>
         <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
      </xsd:complexType>
      <xsd:key name="Key_PEntry">
         <xsd:selector xpath="PEntry"/>
         <xsd:field xpath="@id"/>
      </xsd:key>
   </xsd:element>

   <xsd:element name="PEntry">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ItemName">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="65"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="PrmNo" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="17"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="RegNo" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="ReadRegNos" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
                     <xsd:element name="WriteRegNos" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="DatTypFlg" type="xsd:short"/>
            <xsd:element name="Default" minOccurs="0">
               <xsd:simpleType>
                  <xsd:union>
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer"/>
                     </xsd:simpleType>
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:hexBinary"/>
                     </xsd:simpleType>
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:float"/>
                     </xsd:simpleType>
                  </xsd:union>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="Range" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="129"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="UnitStr" minOccurs="0">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="17"/>
                  </xsd:restriction>
               </xsd:simpleType>
            </xsd:element>
            <xsd:element name="MethodRef" minOccurs="0">
               <xsd:complexType>
                  <xsd:attribute name="ref" type="xsd:string" use="required"/>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
         <xsd:attribute name="id" type="xsd:string" use="required"/>
      </xsd:complexType>
```

```
</xsd:element>

<!-- ApplicationProcess - Method -->
<xsd:element name="Method">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="MethodEntry" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
   </xsd:complexType>
</xsd:element>

<xsd:element name="MethodEntry">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="CmdRegNoCode">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element ref="RegNoCode" maxOccurs="unbounded"/>
               </xsd:sequence>

            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ReqRY" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="EndRX" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ErrEnd" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">

                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ReplyRW" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="16"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="TrueCnd" minOccurs="0">
            <xsd:annotation>
               <xsd:documentation>
                  lt: less than
                  le: less than or equal
                  gt: greater than
                  ge: greater than or equal
                  ==: equality
                  !=: inequality
               </xsd:documentation>
            </xsd:annotation>
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="16"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="ErrReg" minOccurs="0">
```

```
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="8"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="TrueCnd2" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ILock" type="ILockTYPE" minOccurs="0"/>
         <xsd:element name="InterReq" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="InterAns" minOccurs="0">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:restriction base="StringWithOnOffTYPE">
                     <xsd:maxLength value="9"/>
                  </xsd:restriction>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>

<xsd:element name="RegNoCode">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="RegNo" type="xsd:string"/>
         <xsd:choice>
            <xsd:element name="Code" type="xsd:string"/>
            <xsd:sequence>
               <xsd:element name="ReadCode" type="xsd:string"/>
               <xsd:element name="WriteCode" type="xsd:string"/>
            </xsd:sequence>
         </xsd:choice>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>

<!-- ApplicationProcess - CmdPtrn -->
<xsd:element name="CmdPtrn">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Comment" minOccurs="0">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="65"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:sequence>
            <xsd:element ref="WEntry" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
      <xsd:attribute name="EntryNum" type="xsd:short" use="required"/>
   </xsd:complexType>
   <xsd:key name="Key_CmdPtrn_WEntry">
      <xsd:selector xpath="WEntry"/>
      <xsd:field xpath="@id"/>
   </xsd:key>
```

```
</xsd:element>


<xsd:annotation>
   <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:complexType name="ProfileHandle_DataType">
   <xsd:sequence>
      <xsd:element name="ProfileIdentification" type="xsd:string" />
      <xsd:element name="ProfileRevision" type="xsd:string" />
      <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
   </xsd:sequence>
</xsd:complexType>


<xsd:annotation>
   <xsd:documentation>* ISO 15745 CC-LINK DEFINED DATA TYPES *</xsd:documentation>
</xsd:annotation>

<!-- ProfileBody -->
<xsd:complexType name="DeviceIdentityTYPE">
   <xsd:sequence>
      <xsd:element ref="CcLinkVer"/>
      <xsd:element ref="VendName"/>
      <xsd:element ref="VendID"/>
      <xsd:element ref="DevModel"/>
      <xsd:element ref="DevVer"/>
      <xsd:element ref="StationNumber" minOccurs="0"/>
      <xsd:element ref="StationType"/>
      <xsd:element ref="RemDevType" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeviceManagerTYPE">
   <xsd:sequence>
      <xsd:element ref="NumOccupiedStations"/>
      <xsd:element ref="ExtendedCycleConf" minOccurs="0"/>
      <xsd:element ref="BmpFile" minOccurs="0"/>
      <xsd:element ref="ErrReg" minOccurs="0"/>
      <xsd:element ref="UpDownLoadF" minOccurs="0"/>
      <xsd:element ref="MasterFlg" minOccurs="0"/>
      <xsd:element ref="Mode"/>
      <xsd:element ref="TotalIO" minOccurs="0"/>
      <xsd:element ref="IOType" minOccurs="0"/>
      <xsd:element ref="SEMISupport" minOccurs="0"/>
      <xsd:element ref="LEDList" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeviceFunctionTYPE">
   <xsd:sequence>
      <xsd:element name="DevFuncData" type="xsd:string" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ApplicationProcessTYPE">
   <xsd:choice>
      <xsd:element ref="RemoteIOMemoryMap" minOccurs="0"/>
      <xsd:element ref="RemoteDeviceMemoryMap" minOccurs="0"/>
   </xsd:choice>
</xsd:complexType>


<!-- Utility types -->
<xsd:complexType name="StringWithOptOnOffTYPE">
   <xsd:simpleContent>
      <xsd:extension base="xsd:string">
         <xsd:attribute name="onoff" type="OnOffTYPE"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="StringWithOnOffTYPE">
   <xsd:simpleContent>
      <xsd:extension base="xsd:string">
```

```
            <xsd:attribute name="onoff" type="OnOffTYPE" use="required"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>

   <xsd:complexType name="ILockTYPE">
      <xsd:sequence>
         <xsd:element name="DevName" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:extension base="xsd:string">
                     <xsd:attribute name="onoff" type="OnOffTYPE" use="required"/>
                  </xsd:extension>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="OnOffTYPE">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="on"/>
         <xsd:enumeration value="off"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:schema>
```

A.2.2 Device profile template description – XML encapsulation of CSP files

A.2.2.1 General

The device profile XML files used to encapsulate CSP files shall comply with the device profile XML schema as specified in A.2.2.2.

The semantics of the sub-elements of the ExternalProfileHandle element, used to reference an existing CSP file, are specified in Table A.14.

Table A.14 — ExternalProfileHandle elements

XML schema elements	CSP Elements
ProfileIdentification	CSP File Name (see A.4.2)
ProfileRevision	Version (see CSP file version in Table A.35)

If present, the DeviceIdentity, DeviceManager, DeviceFunction and ApplicationProcess elements should be compatible with the formats specified in A.2.1.3.

A.2.2.2 XML schema : CSP_DeviceProfile_wrapper.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileHeader">
```

```
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="DeviceIdentity" minOccurs="0" maxOccurs="1">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="DeviceManager" minOccurs="0" maxOccurs="1">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="DeviceFunction" minOccurs="1" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ApplicationProcess" minOccurs="0" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>


   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP" />
         <xsd:enumeration value="Process" />
         <xsd:enumeration value="InformationExchange" />
         <xsd:enumeration value="Resource" />
         <xsd:enumeration value="Device" />
         <xsd:enumeration value="CommunicationNetwork" />
         <xsd:enumeration value="Equipment" />
         <xsd:enumeration value="Human" />
         <xsd:enumeration value="Material" />
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
                <xsd:element name="ProfileTechnology" type="xsd:string" />
             </xsd:sequence>
          </xsd:complexType>

          <xsd:simpleType name="IASInterface_DataType">
             <xsd:union>
                <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="CSI" />
                      <xsd:enumeration value="HCI" />
                      <xsd:enumeration value="ISI" />
                      <xsd:enumeration value="API" />
                      <xsd:enumeration value="CMI" />
                      <xsd:enumeration value="ESI" />
                      <xsd:enumeration value="FSI" />
                      <xsd:enumeration value="MTI" />
                      <xsd:enumeration value="SEI" />
                      <xsd:enumeration value="USI" />
                   </xsd:restriction>
                </xsd:simpleType>
                <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                      <xsd:length value="4" />
                   </xsd:restriction>
                </xsd:simpleType>
             </xsd:union>
          </xsd:simpleType>

          <xsd:annotation>
             <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
          </xsd:annotation>

          <xsd:complexType name="ProfileHandle_DataType">
             <xsd:sequence>
                <xsd:element name="ProfileIdentification" type="xsd:string" />
                <xsd:element name="ProfileRevision" type="xsd:string" />
                <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
             </xsd:sequence>
          </xsd:complexType>
       </xsd:schema>
```

A.3 Communication network profile template description

A.3.1 Communication network profile template description – XML based

A.3.1.1 General

The communication network profile XML files shall comply with the communication network profile XML schema as specified in A.3.1.3.

Contents of this XML schema are derived from the communication network profile class diagrams shown in 6.2.2, and extended with additional elements to allow full description of communication network requirements or capabilities.

A.3.1.2 Semantics of XML schema elements

A.3.1.2.1 ApplicationLayers

The semantics of the OptionCommand elements are specified in Table A.15.

NOTE Further details of the semantics are described in BAP-05026.

Table A.15 — OptionCommand elements

Elements	Description and semantics
CommandType	Type of command (0x60h ~ 0x7Fh)
Comment	Meaning of the command
BroadcastSupport	Availability of broadcast
ReqRes	Availability of request, response, or both

A.3.1.2.2 TransportLayers

A.3.1.2.2.1 PhysicalLayerInterface

The semantics of the PhysicalLayerInterface elements are specified in Table A.16. PhysicalLayerInterface class is used by all versions of CC-Link.

NOTE Further details of the semantics are described in BAP-05027 and BAP-05031.

Table A.16 — PhysicalLayerInterface elements

Elements	Description and semantics
ConnectorType	Type of connector
InterfaceSpeed	Speeds supported by the physical layer interface Combination of "156kbps”, "625kbps”, "2.5Mbps”, "5Mbps”, and "10Mbps”
Speed	Current setting of speed of the physical layer interface Selection of "156kbps” , "625kbps”, "2.5Mbps”, "5Mbps”, or "10Mbps”

A.3.1.2.2.2 DatalinkLayerInterface

The semantics of the DLConfigParams elements are specified in Table A.17. Table A.18 shows the constraints of the DLConfigParams elements.

NOTE Further details of the semantics are described in BAP-05026.

Table A.17 — DLConfigParams elements

Elements	Description and semantics
NumModules	Number of slave stations connected to the master station (including any reserved station[s])
NumIntelligentDevs	Number of local stations and intelligent device stations
NumAutoRecoveryModules	Number of slave stations that can be returned within 1 link scan.
RetryCount	Retry count when communication is abnormal.
DelayTime	Specify the link scan interval.
OperationInAbnormal	Specify the data link status when a user application program is abnormal.
DataClearInAbnormal	Master station: specifies to retain/clear input data when the communication is abnormal Slave station: specifies to retain/clear output to outside when the communication is abnormal
ScanMode	Specify the link scan to be synchronous/asynchronous with respect to the user application program
ReservedStation	Although the reserved station is counted as a connected station, no unconnected slave station will actually be treated as a data link error station.
ErrInvalidStation	The master and local stations do not treat the slave stations that become unable to perform data link as data link error stations.
ExtensionCycle	Extension cycle 0x0: 1 time 0x1: 2 times 0x2: 4 times 0x3: 8 times
PointsMode	The number of the input/output points of each slave station. 0: 4 points 1: 8 points 2: 16 points
IOPointsNumber	The number of input/output points controlled by a master station

Table A.18 — Constraints of DLConfigParams elements

Element	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
NumModules	Optional	Optional	Optional	Optional
NumIntelligentDevs	Optional	Optional	Optional	Not applicable
NumAutoRecoveryModules	Optional	Optional	Optional	Not applicable
RetryCount	Optional	Optional	Optional	Not applicable
DelayTime	Optional	Optional	Optional	Not applicable
OperationInAbnormal	Optional	Optional	Optional	Not applicable
DataClearInAbnormal	Optional	Optional	Optional	Optional
ScanMode	Optional	Optional	Optional	Not applicable
ReservedStation	Optional	Optional	Optional	Not applicable
ErrInvalidStation	Optional	Optional	Optional	Not applicable
ExtensionCycle	Not applicable	Not applicable	Optional	Not applicable
PointsMode	Not applicable	Not applicable	Not applicable	Optional
IOPointsNumber	Not applicable	Not applicable	Not applicable	Optional

NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.

The semantics of the DLVariables elements are specified in Table A.19. Table A.20 shows the constraints of the DLVariables elements.

NOTE Further details of the semantics are described in BAP-05026.

Table A.19 — DLVariables elements

Elements	Description and semantics
Address	Source address in HDLC frame
FrameIntervals	Current transmission frame interval
RefreshIntervals	Current refresh interval
NumRYInfoPoints	Number of RY Information transmission points 0x0: 0 0x1: 256 0x2: 512 0x3: 768 0x4: 1024 0x5: 1280 0x6: 1536 0x7: 1792 0x8: 2048
NumRWwInfoPoints	Number of RWw information transmission points 0x0: 0 0x1: 32 0x2: 64 0x3: 96 0x4: 128 0x5: 160 0x6: 192 0x7: 224 0x8: 256
StationInfo	Station information (station type attributes, number of occupied stations)
CurrentLinkScanTime	Current value of link scan time
MinLinkScanTime	Minimum value of link scan time
MaxLinkScanTime	Maximum value of link scan time

Table A.20 — Constraints of DLVariables elements

Element	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
Address	Optional	Optional	Optional	Optional
FrameIntervals	Optional	Optional	Optional	Not applicable
RefreshIntervals	Optional	Optional	Optional	Not applicable
NumRYInfoPoints	Optional	Optional	Optional	Not applicable
NumRWwInfoPoints	Optional	Optional	Optional	Not applicable
StationInfo	Optional	Optional	Optional	Not applicable
CurrentLinkScanTime	Optional	Optional	Optional	Not applicable
MinLinkScanTime	Optional	Optional	Optional	Not applicable
MaxLinkScanTime	Optional	Optional	Optional	Not applicable
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

A.3.1.2.3 NetworkManagement

A.3.1.2.3.1 NetworkStatusInfo

Table A.21 shows the constraints of the child classes of NetworkStatusInfo.

Table A.21 — Constraints of the child classes of NetworkStatusInfo

Class	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
SelfStationStatus	Optional	Optional	Optional	Not applicable
FarStationStatus	Optional	Optional	Optional	Not applicable
MasterToSlaveST	Optional	Optional	Optional	Not applicable
SlaveToMasterST	Optional	Optional	Optional	Not applicable
ReceivingInfoLT	Not applicable	Not applicable	Not applicable	Optional
SlaveStationLT	Not applicable	Not applicable	Not applicable	Optional
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

The semantics of the SelfStationStatus elements are specified in Table A.22.

NOTE Further details of the semantics are described in BAP-05026.

Table A.22 — SelfStationStatus elements

Elements	Description and semantics
FrameIntervalAbnormalityFlag	Normal reception of ‘Frames addressed to the self-station’ interval timed out. 0: No 1: Yes
RefreshFrameAbnormalityFlag	‘Refresh frame reception’ interval timed out. 0: No 1: Yes
SuccessiveTransTimeMonitoring	Successive transmission time monitoring timed out. 0: No 1: Yes
SuccessiveReceivingMonitoring	Successive reception monitoring timed out. 0: No 1: Yes
TransPathMonitoring1Flag	Transmission path monitoring (1) timed out. 0: No 1: Yes
TransPathMonitoring2Flag	Transmission path monitoring (2) timed out. 0: No 1: Yes
MasterSwitchingRequest	Request to switch the master function to the standby master station. 0: No 1: Yes

The semantics of the FarStationStatus elements are specified in Table A.23.

NOTE Further details of the semantics are described in BAP-05026.

Table A.23 — FarStationStatus elements

Elements	Description and semantics
PollingStatusBit	Slave station polling response. 0: Abnormal 1: Normal
CRCErrFlag	A CRC error occurred. 0: No 1: Yes
AbortErrFlag	7 or more “1” bits were received consecutively. 0: No 1: Yes
TimeOutErrFlag	Polling response cannot be received within the predetermined time. 0: No 1: Yes
BufferOverErrFlag	Data was received beyond the reception buffer size. 0: No 1: Yes
ReceivingFrameAddrFlag	Address information of the received frame format is invalid. 0: No 1: Yes
RetryFlag	A retry occurred. 0: No 1: Yes

The semantics of the MasterToSlaveST elements are specified in Table A.24 and Table A.25.

NOTE Further details of the semantics are described in BAP-05026.

Table A.24 — MasterToSlaveST elements

Elements	Description and semantics
MasterStationUserApp	Master station’s user application program operational status. 0: Stop 1: Run
MasterStationUserAppErr	Master station’s user application programs. 0: Normal 1: Abnormal
RefreshStartup	Link refresh status 0: Stop 1: Start
TransientData	Transient data 0: Not present 1: Present
TransientDataEnable	Transient data reception capability 0: Disabled 1: Enabled
MasterStationType	Station type of the master station 0: Master station 1: Standby / master station
NumRYInfoPoints	Number of RY Information transmission points 0x0: 0 0x1: 256 0x2: 512 0x3: 768 0x4: 1024 0x5: 1280 0x6: 1536 0x7: 1792 0x8: 2048
NumRWwInfoPoints	Number of RWw information transmission points 0x0: 0 0x1: 32 0x2: 64 0x3: 96 0x4: 128 0x5: 160 0x6: 192 0x7: 224 0x8: 256
ST3	ST3 information

Table A.25 — Constraints of MasterToSlaveST elements

Element	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
MasterStationUserApp	Optional	Optional	Optional	Not applicable
MasterStationUserAppErr	Optional	Optional	Optional	Not applicable
RefreshStartup	Optional	Optional	Optional	Not applicable
TransientData	Optional	Optional	Optional	Not applicable
TransientDataEnable	Optional	Optional	Optional	Not applicable
MasterStationType	Optional	Optional	Optional	Not applicable
NumRYInfoPoints	Optional	Optional	Optional	Not applicable
NumRWwInfoPoints	Optional	Optional	Optional	Not applicable
ST3	Not applicable	Not applicable	Optional	Not applicable
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

The semantics of the SlaveToMasterST elements are specified in Table A.26 and Table A.27.

NOTE Further details of the semantics are described in BAP-05026.

Table A.26 — SlaveToMasterST elements

Elements	Description and semantics
FuseBlowout	Slave station’s fuse blow out 0: Not present 1: Present
UnitErr	Unit error of remote I/O stations or invalid number of points presence flag of slave stations excluding remote I/O stations 0: Not present 1: Present
RefreshNotReceived	Reception status of refresh transmission from the master station. 0: Received 1: Not received
ParamNotReceived	Reception status of parameter information from the master station. 0: Received 1: Not received
SwitchChangeDetection	Switch setting change detection after the power is turned on or reset is cancelled. 0: Not present 1: Present
CyclicComm	Cyclic communication capability 0: Enabled 1: Disabled
WDTErr	The watch dog timer error status 0: Not present 1: Present
TransientData	Transient data 0: Not present 1: Present
TransientCommEnable	Transient data reception capability 0: Disabled 1: Enabled
TransientType	0: 1:n communication (intelligent device station) 1: n:n communication (local station)
TransmissionPathStatus	Transmission path status 0: Normal 1: Error
ExtensionCycle	Extension cycle 0x0: 1 time 0x1: 2 times 0x2: 4 times 0x3: 8 times
ST3	ST3 information

Table A.27 — Constraints of SlaveToMasterST elements

Element	Version 1.00 / 1.10	Version 1.11	Version 2.00	Version LT
FuseBlowout	Optional	Optional	Optional	Not applicable
UnitErr	Optional	Optional	Optional	Not applicable
RefreshNotReceived	Optional	Optional	Optional	Not applicable
ParamNotReceived	Optional	Optional	Optional	Not applicable
SwitchChangeDetection	Optional	Optional	Optional	Not applicable
CyclicComm	Optional	Optional	Optional	Not applicable
WDTErr	Optional	Optional	Optional	Not applicable
TransientData	Optional	Optional	Optional	Not applicable
TransientCommEnable	Optional	Optional	Optional	Not applicable
TransientType	Optional	Optional	Optional	Not applicable
TransmissionPathStatus	Optional	Optional	Optional	Not applicable
ExtensionCycle	Not applicable	Not applicable	Optional	Not applicable
ST3	Not applicable	Not applicable	Optional	Not applicable
NOTE The four right hand columns indicate whether a certain child class is mandatory, optional, or not applicable in a device profile of a specified CC-Link version.				

The semantics of the ReceivingInfoLT elements are specified in Table A.28.

NOTE Further details of the semantics are described in BAP-05031.

Table A.28 — ReceivingInfoLT elements

Elements	Description and semantics
CRCErrFlag	A CRC error occurred. 0: No 1: Yes
ParityErrFlag	A parity error occurred 0: No 1: Yes
AbortErrFlag	7 or more “1” bits were received consecutively. 0: No 1: Yes
NoResponseErrFlag	Master station did not receive a response in the specified period. 0: No 1: Yes

The semantics of the SlaveStatusLT elements are specified in Table A.29.

NOTE Further details of the semantics are described in BAP-05031.

Table A.29 — SlaveStatusLT elements

Elements	Description and semantics
ConnectionInfo	A test response to the station information gathering test is received. 0: Offline 1: Online
InAbnormal	A CRC error, a parity error, or timeout occurs, or no response is received after 31 successive attempts. 0: Normal 1: Abnormal

A.3.1.2.3.2 OtherStationsMInfoList

The semantics of the OtherStationManagementInfo elements are specified in Table A.30.

Table A.30 — OtherStationManagementInfo elements

Elements	Description and semantics
StationNumber	A unique identifier of the device.
VendorCode	Vendor code
ModelCode	Model code
SoftwareVersion	Version of software

A.3.1.2.3.3 SlaveTransInfoLT

The semantics of the SlaveStationInfo elements are specified in Table A.31.

NOTE Further details of the semantics are described in BAP-05031.

Table A.31 — SlaveStationInfo elements

Elements	Description and semantics
OutputHldClr	Indicates to retain/clear output when the communication is abnormal
InputConstant	Indicates whether input time constant is high speed or normal
HeadStationFlag	Indicates whether or not a station is the head station when one slave station contains several stations.
WordDataStationFlag	Indicates whether a slave station is a word data station or not
InputFlag	Indicates whether or not data is remote input or remote input/output
OutputFlag	Indicates whether or not data is remote output or remote input/output
IOPoints	The number of input/output points of slave stations

The semantics of the STInfo elements are specified in Table A.32.

Table A.32 — STInfo elements

Elements	Description and semantics
Transmitted	Indicates whether the station information has been transmitted or not
RemoteIOErr	Indicates whether slave station input/output error occurs or not

A.3.1.3 XML schema : CC-Link_CommNetworkProfile.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.cc-link.org/ISO15745/CommNetworkProfile"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns="http://www.cc-link.org/ISO15745/CommNetworkProfile"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>


   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"
fixed="CommunicationNetwork"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>


   <xsd:annotation>
      <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
   </xsd:annotation>

   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP" />
         <xsd:enumeration value="Process" />
         <xsd:enumeration value="InformationExchange" />
         <xsd:enumeration value="Resource" />
         <xsd:enumeration value="Device" />
         <xsd:enumeration value="CommunicationNetwork" />
         <xsd:enumeration value="Equipment" />
         <xsd:enumeration value="Human" />
         <xsd:enumeration value="Material" />
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
         <xsd:element name="ProfileTechnology" type="xsd:string" />
      </xsd:sequence>
   </xsd:complexType>
```

```
<xsd:simpleType name="IASInterface_DataType">
   <xsd:union>
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="CSI" />
            <xsd:enumeration value="HCI" />
            <xsd:enumeration value="ISI" />
            <xsd:enumeration value="API" />
            <xsd:enumeration value="CMI" />
            <xsd:enumeration value="ESI" />
            <xsd:enumeration value="FSI" />
            <xsd:enumeration value="MTI" />
            <xsd:enumeration value="SEI" />
            <xsd:enumeration value="USI" />
         </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:length value="4" />
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:union>
</xsd:simpleType>


<xsd:annotation>
   <xsd:documentation>* BODY SECTION *</xsd:documentation>
</xsd:annotation>

<xsd:element name="ProfileBody">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="ApplicationLayers"/>
         <xsd:element ref="TransportLayers"/>
         <xsd:element ref="NetworkManagement" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>

<xsd:annotation>
   <xsd:documentation>* ISO 15745 CC-LINK DEFINED ELEMENTS *</xsd:documentation>
</xsd:annotation>

<!-- ProfileBody -->

<xsd:element name="ApplicationLayers" type="ApplicationLayersTYPE"/>
<xsd:element name="TransportLayers" type="TransportLayersTYPE"/>
<xsd:element name="NetworkManagement" type="NetworkManagementTYPE"/>

<!-- ApplicationLayers -->
<xsd:element name="Cyclic"/>
<xsd:element name="ExtendedCyclic"/>
<xsd:element name="Transient" type="TransientTYPE">
   <xsd:unique name="OptionCommand">
      <xsd:selector xpath="OptionCommand"/>
      <xsd:field xpath="CommandType"/>
   </xsd:unique>
</xsd:element>
<xsd:element name="Message"/>

<xsd:annotation>
   <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:complexType name="ProfileHandle_DataType">
   <xsd:sequence>
      <xsd:element name="ProfileIdentification" type="xsd:string" />
      <xsd:element name="ProfileRevision" type="xsd:string" />
      <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
   </xsd:sequence>
</xsd:complexType>


<xsd:annotation>
   <xsd:documentation>* ISO 15745 CC-LINK DEFINED DATA TYPES *</xsd:documentation>
   </xsd:annotation>

   <!-- ApplicationLayers -->
   <xsd:complexType name="ApplicationLayersTYPE">
      <xsd:sequence>
         <xsd:choice>
            <xsd:element name="Master"/>
            <xsd:element name="Slave" type="StationTYPE"/>
         </xsd:choice>
         <xsd:element ref="Cyclic" minOccurs="0"/>
         <xsd:element ref="ExtendedCyclic" minOccurs="0"/>
         <xsd:element ref="Transient" minOccurs="0"/>
         <xsd:element ref="Message" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="TransientTYPE">
      <xsd:sequence>
         <xsd:element name="OptionCommand" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="CommandType" type="xsd:hexBinary">
                     <xsd:annotation>
                        <xsd:documentation>The range of CommandType is from 60h to
7Fh</xsd:documentation>
                     </xsd:annotation>
                  </xsd:element>
                  <xsd:element name="Comment" type="xsd:string"/>
                  <xsd:element name="BroadcastSupport" type="xsd:boolean"/>
                  <xsd:element name="ReqRes" type="xsd:short">
                     <xsd:annotation>
                        <xsd:documentation>
                           0: request
                           1: response
                           2: both request and response
                        </xsd:documentation>
                     </xsd:annotation>
                  </xsd:element>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="StationTYPE">
      <xsd:restriction base="xsd:short">
         <xsd:annotation>
            <xsd:documentation>
               0: Remote I/O station
               1: Remote device station
               2: Intelligent device station (includes local stations and standby master station)
            </xsd:documentation>
         </xsd:annotation>
      </xsd:restriction>
   </xsd:simpleType>

   <!-- TransportLayers -->
   <xsd:complexType name="TransportLayersTYPE">
      <xsd:sequence>
         <xsd:element name="PhysicalLayerInterface" type="PhysicalLayerInterfaceTYPE"/>
         <xsd:element name="DatalinkLayerInterface" type="DatalinkLayerInterfaceTYPE"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="PhysicalLayerInterfaceTYPE">
      <xsd:sequence>
         <xsd:element name="ConnectorType" type="xsd:string" minOccurs="0"/>
         <xsd:element name="InterfaceSpeed">
            <xsd:simpleType>
               <xsd:list itemType="SpeedTYPE"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Speed" type="SpeedTYPE" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>
```

```
<xsd:complexType name="DatalinkLayerInterfaceTYPE">
   <xsd:sequence>
      <xsd:element name="DLConfigParams" type="DLConfigParamsTYPE" minOccurs="0"/>
      <xsd:element name="DLVariables" type="DLVariablesTYPE" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DLConfigParamsTYPE">
   <xsd:sequence>
      <xsd:element name="NumModules" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:unsignedShort">
               <xsd:minInclusive value="1"/>
               <xsd:maxInclusive value="64"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>

      <xsd:element name="NumIntelligentDevs" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:unsignedShort">
               <xsd:minInclusive value="0"/>
               <xsd:maxInclusive value="26"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="NumAutoRecoveryModules" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:unsignedShort">
               <xsd:minInclusive value="1"/>
               <xsd:maxInclusive value="10"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="RetryCount" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:unsignedShort">
               <xsd:minInclusive value="1"/>
               <xsd:maxInclusive value="7"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="DelayTime" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:unsignedShort">
               <xsd:minInclusive value="0"/>
               <xsd:maxInclusive value="5000"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="OperationInAbnormal" type="xsd:unsignedShort" minOccurs="0"/>
      <xsd:element name="DataClearInAbnormal" type="xsd:unsignedShort" minOccurs="0"/>
      <xsd:element name="ScanMode" type="xsd:unsignedShort" minOccurs="0"/>
      <xsd:element name="ReservedStation" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:hexBinary">
               <xsd:maxLength value="16"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="ErrInvalidStation" minOccurs="0">
         <xsd:simpleType>
            <xsd:restriction base="xsd:hexBinary">
               <xsd:maxLength value="16"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="ExtendedCycle" type="ExtendedCycleTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="PointsMode" type="xsd:unsignedShort" minOccurs="0">
         <xsd:annotation>
            <xsd:documentation>
               0: 4 points
               1: 8 points
               2: 16 points
            </xsd:documentation>
         </xsd:annotation>
         </xsd:element>
         <xsd:element name="IOPointsNumber" type="xsd:unsignedShort" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="DLVariablesTYPE">
      <xsd:sequence>
         <xsd:element name="Address" minOccurs="0" nillable="true">
            <xsd:simpleType>
               <xsd:restriction base="xsd:hexBinary">
                  <xsd:maxLength value="2"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="FrameIntervals" type="xsd:float" minOccurs="0" nillable="true"/>
         <xsd:element name="RefreshIntervals" type="xsd:float" minOccurs="0" nillable="true"/>
         <xsd:element name="NumRYInfoPoints" type="NumRYInfoPointsTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="NumRWwInfoPoints" type="NumRWwInfoPointsTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="StationInfo" minOccurs="0" nillable="true">
            <xsd:simpleType>
               <xsd:restriction>
                  <xsd:simpleType>
                     <xsd:list itemType="xsd:unsignedShort"/>
                  </xsd:simpleType>
                  <xsd:maxLength value="64">
                  </xsd:maxLength>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="CurrentLinkScanTime" type="xsd:unsignedShort" minOccurs="0"
nillable="true"/>
         <xsd:element name="MinLinkScanTime" type="xsd:unsignedShort" minOccurs="0" nillable="true"/>
         <xsd:element name="MaxLinkScanTime" type="xsd:unsignedShort" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>

   <!-- NetworkManagment -->
   <xsd:complexType name="NetworkManagementTYPE">
      <xsd:sequence minOccurs="0">
         <xsd:element name="NetworkStatusInfo" type="NetworkStatusInfoTYPE" minOccurs="0"/>
         <xsd:element name="OtherStationsMInfoList" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="OtherStationManagementInfo"
type="OtherStationManagementInfoTYPE" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="SlaveTransInfoLT" type="SlaveTransInfoLTTYPE" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="NetworkStatusInfoTYPE">
      <xsd:sequence>
         <xsd:element name="SelfStationStatus" type="SelfStationStatusTYPE" minOccurs="0"/>
         <xsd:element name="FarStationStatus" type="FarStationStatusTYPE" minOccurs="0"/>
         <xsd:element name="MasterToSlaveST" type="MasterToSlaveSTTYPE" minOccurs="0"/>
         <xsd:element name="SlaveToMasterST" type="SlaveToMasterSTTYPE" minOccurs="0"/>
         <xsd:element name="ReceivingInfoLT" type="ReceivingInfoLTTYPE" minOccurs="0"
maxOccurs="64"/>
         <xsd:element name="SlaveStatusLT" type="SlaveStatusLTTYPE" minOccurs="0" maxOccurs="64"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="SlaveTransInfoLTTYPE">
      <xsd:sequence>
         <xsd:element name="SlaveStationInfo" type="SlaveStationInfoTYPE" minOccurs="0"
maxOccurs="64" nillable="true"/>
         <xsd:element name="STInfo" type="STInfoTYPE" minOccurs="0" maxOccurs="64" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="bitTYPE">
      <xsd:restriction base="xsd:int">
```

```
         <xsd:minInclusive value="0"/>
         <xsd:maxInclusive value="1"/>
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="OtherStationManagementInfoTYPE">
      <xsd:sequence>
         <xsd:element name="StationNumber" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
         <xsd:element name="VendorCode" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
         <xsd:element name="ModelCode" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
         <xsd:element name="SoftwareVersion" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>


   <xsd:complexType name="SelfStationStatusTYPE">
      <xsd:sequence>
         <xsd:element name="FrameIntervalAbnormalityFlag" type="bitTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="RefreshFrameAbnormalityFlag" type="bitTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="SuccessiveTransTimeMonitoring" type="bitTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="SuccessiveReceivingMonitoring" type="bitTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="TransPathMonitoring1Flag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransPathMonitoring2Flag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="MasterSwitchingRequest" type="bitTYPE" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="FarStationStatusTYPE">
      <xsd:sequence>
         <xsd:element name="PollingStatusBit" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="CRCErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="AbortErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TimeOutErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="BufferOverErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="ReceivingFrameAddrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="RetryFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="MasterToSlaveSTTYPE">
      <xsd:sequence>
         <xsd:element name="MasterStationUserApp" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="MasterStationUserAppErr" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="RefreshStartup" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransientData" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransientDataEnable" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="MasterStationType" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="NumRYInfoPoints" type="NumRYInfoPointsTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="NumRWwInfoPoints" type="NumRWwInfoPointsTYPE" minOccurs="0"
nillable="true"/>
         <xsd:element name="ST3" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="SlaveToMasterSTTYPE">
      <xsd:sequence>
         <xsd:element name="FuseBlowout" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="UnitErr" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="RefreshNotReceived" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="ParamNotReceived" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="SwitchChangeDetection" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="CyclicComm" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="WDTErr" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransientData" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransientCommEnable" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransientType" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="TransmissionPathStatus" type="bitTYPE" minOccurs="0" nillable="true"/>
         <xsd:element name="ST3" type="xsd:hexBinary" minOccurs="0" nillable="true"/>
         <xsd:element name="ExtendedCycle" type="ExtendedCycleTYPE" minOccurs="0" nillable="true"/>
      </xsd:sequence>
   </xsd:complexType>
```

```
<xsd:complexType name="ReceivingInfoLTTYPE">
   <xsd:sequence>
      <xsd:element name="CRCErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="ParityErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="AbortErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="NoResponseErrFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SlaveStatusLTTYPE">
   <xsd:sequence>
      <xsd:element name="ConnectionInfo" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="InAbnormal" type="bitTYPE" minOccurs="0" nillable="true"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SlaveStationInfoTYPE">
   <xsd:sequence>
      <xsd:element name="OutputHldClr" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="InputConstant" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="HeadStationFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="WordDataStationFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="InputFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="OutputFlag" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="IOPoints" type="bitTYPE" minOccurs="0" nillable="true"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="STInfoTYPE">
   <xsd:sequence>
      <xsd:element name="Transmitted" type="bitTYPE" minOccurs="0" nillable="true"/>
      <xsd:element name="RemoteIOErr" type="bitTYPE" minOccurs="0" nillable="true"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SpeedTYPE">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="156kbps"/>
      <xsd:enumeration value="625kbps"/>
      <xsd:enumeration value="2.5Mbps"/>
      <xsd:enumeration value="5Mbps"/>
      <xsd:enumeration value="10Mbps"/>
   </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="NumRYInfoPointsTYPE">
   <xsd:annotation>
      <xsd:documentation>
         0: 0 point
         1: 256 points (32 bytes)
         2: 512 points (64 bytes)
         3: 768 points (96 bytes)
         4: 1024 points (128 bytes)
         5: 1280 points (160 bytes)
         6: 1536 points (192 bytes)
         7: 1792 points (224 bytes)
         8: 2048 points (256 bytes)
      </xsd:documentation>
   </xsd:annotation>
   <xsd:restriction base="xsd:hexBinary">
      <xsd:maxLength value="1">
      </xsd:maxLength>
   </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="NumRWwInfoPointsTYPE">
   <xsd:annotation>
      <xsd:documentation>
         0: 0 point
         1: 32 points (64 bytes)
         2: 64 points (128 bytes)
         3: 96 points (192 bytes)
         4: 128 points (256 bytes)
         5: 160 points (320 bytes)
         6: 192 points (384 bytes)
```

```
            7: 224 points (448 bytes)
            8: 256 points (512 bytes)
         </xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="1">
         </xsd:maxLength>
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:simpleType name="ExtendedCycleTYPE">
      <xsd:annotation>
         <xsd:documentation>
            0: Single
            1: Double
            2: Quadruple
            3: Octuple
         </xsd:documentation>
      </xsd:annotation>
      <xsd:restriction base="xsd:hexBinary">
         <xsd:maxLength value="1"/>
      </xsd:restriction>
   </xsd:simpleType>

</xsd:schema>
```

A.3.2 Communication network profile template description – XML encapsulation of CSP files

A.3.2.1 General

The communication network profile XML files used to encapsulate CSP files shall comply with the communication network profile XML schema as specified in A.3.2.2.

The semantics of the sub-elements of the ExternalProfileHandle element, used to reference an existing CSP file, are specified in Table A.14.

A.3.2.2 XML schema : CSP_CommNetworkProfile_wrapper.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>


   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP" />
         <xsd:enumeration value="Process" />
         <xsd:enumeration value="InformationExchange" />
         <xsd:enumeration value="Resource" />
         <xsd:enumeration value="Device" />
         <xsd:enumeration value="CommunicationNetwork" />
         <xsd:enumeration value="Equipment" />
         <xsd:enumeration value="Human" />
         <xsd:enumeration value="Material" />
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
         <xsd:element name="ProfileTechnology" type="xsd:string" />
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI" />
               <xsd:enumeration value="HCI" />
               <xsd:enumeration value="ISI" />
               <xsd:enumeration value="API" />
               <xsd:enumeration value="CMI" />
               <xsd:enumeration value="ESI" />
               <xsd:enumeration value="FSI" />
               <xsd:enumeration value="MTI" />
               <xsd:enumeration value="SEI" />
               <xsd:enumeration value="USI" />
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4" />
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>

   <xsd:annotation>
      <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
   </xsd:annotation>

   <xsd:complexType name="ProfileHandle_DataType">
      <xsd:sequence>
         <xsd:element name="ProfileIdentification" type="xsd:string" />
         <xsd:element name="ProfileRevision" type="xsd:string" />
         <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

A.4 CSP (CC-Link System Profile)

A.4.1 Generic

This subclause describes the structure of the CSP (CC-Link System Profile) file. When supplying a CC-Link device, a CSP file is created based on the requirements explained in this subclause. A CSP file is a text file that can be created by generic text editors.

A.4.2 CSP Filename

Names for CSP files shall be of the form:

[DistinguishableName]DeviceModelName_NumberOfOccupiedStations.csp

DistinguishableName is optional and used only if the DeviceModelName is the same as another DeviceModelName.

The number of occupied stations can be variable for some devices rather than fixed. Some devices have more than one valid "number of occupied stations" value. In such case, a separate CSP file shall be created for each valid "number of occupied stations" value.

EXAMPLE1 Example of CSP file name for an analog-digital converter module named “ADCONV2000-64AD” with two occupied stations
ADCONV2000-64AD_2.csp

EXAMPLE2 Example of CSP file name for a positioning module named “ADCONV2001-D75P2-S3” with the number of occupied stations, 2 or 4:
ADCONV2001-D75P2-S3_2.csp
ADCONV2001-D75P2-S3_4.csp

EXAMPLE3 Example of CSP file name with DistinguishableName "S-ABC-" for the device model name “CC1” to distinguish it from other devices having the same device model name:
S-ABC-CC1_1.csp

Maximum length of a CSP filename is 64 bytes (excluding the terminal NULL character). The DistinguishableName and DeviceModelName shall not contain the underscore character (‘_’).

A.4.3 CSP Content

A.4.3.1 CSP File Structure

A single file shall contain the entire CSP. A CSP consists of the sections shown in Table A.33. The placement of the sections shall be in the order shown in Table A.33.

Table A.33 — CSP file structure

Section name	Delimiter	Required/Optional	
		Remote I/O station	**Remote device station**
File section	[File]	Required	Required
Device	[Device]	Required	Required
RX section	[IO_Info_RX]	Optional	Required
RY section	[IO_Info_RY]	Optional	Required
RWw section	[IO_Info_RWw]	Not applicable	Required
RWw bit register section	[IO_RWw_Bit]	Not applicable	Optional
RWr section	[IO_Info_RWr]	Not applicable	Required
RWr bit register section	[IO_RWr_Bit]	Not applicable	Optional
Select parameter section	[SelectPrm]	Not applicable	Optional
Parameter setting section	[PrmSet*n*][a]	Not applicable	Optional
Group item section	[GrpItem*n*][a]	Not applicable	Optional
Method section	[Method]	Not applicable	Optional
Command pattern section	[CmdPtrn*n*][a]	Not applicable	Optional

[a] The delimiter shall consist of the combination of the character array prior to *n* representing Arabic numerals (decimal number). The decimal numbers shall start at one and be incremented by one for each successive common section.

If a CSP file for a Remote I/O station contains sections that are not applicable to Remote I/O stations, those sections shall be ignored and have no meaning.

EXAMPLE

```
[File]
FileComment = Analog to digital converter module (number of occupied stations: 2);
CreateDate = 1999/01/19;
CreateTime = 18:50:00;
ModDate = 1999/01/20;
ModTime = 13:30:00
Version = 1.0;

[Device]
VendName = A name of vendor
VendID = 0x0;
StationType = 1;
RemDevType = 0x4;
DevModel = ADCONV1999;
DevVer = A;
Senyuu = 2;
BmpFile = remote_dv.bmp;
ErrReg =  ;
UpDownLoadF = 1;
MasterFlg = 0;

[IO_Info_RX]
EntryNum = 8;
BEntry1 = CH.1 A/D conversion complete flag,      RX0,  , A/D conversion complete;
BEntry2 = CH.2 A/D conversion complete flag,      RX1,  , A/D conversion complete;
BEntry3 = CH.3 A/D conversion complete flag,      RX2,  , A/D conversion complete;
BEntry4 = CH.4 A/D conversion complete flag,      RX3,  , A/D conversion complete;
BEntry5 = Initial data processing request flag,   RX18, , Initial data processing request;
BEntry6 = Initial data setting complete flag,    RX19, , Initial data setting complete;
BEntry7 = Error status flag,     RX1A, No error, Error;
```

```
BEntry8 = Remote READY,    RX1B, , Ready;

[IO_Info_RY]
EntryNum = 5;
BEntry1 = Offset/gain value selection, RY0, User setting, factory setting;
BEntry2 = Voltage/current selection, RY1, Voltage, current;
BEntry3 = Initial data processing request flag, RY18, , initial data processing complete;
BEntry4 = Initial data setting request flag, RY19, , initial data setting;
BEntry5 = Error reset request flag, RY1A, , Error reset;

[IO_Info_RWw]
Comment = Analog to digital converter module (number of occupied stations: 2);
EntryNum = 4;
WEntry1 = CH1, RWw1, 2, PrmSet1, 0, , -, -, , , , ;
WEntry2 = CH2, RWw2, 2, PrmSet2, 0, , -, -, , , , ;
WEntry3 = Data format, RWw5, 3, , 0, , -, -, , , , ;
WEntry4 = A/D conversion enable/disable specification, RWw6, 3, , 0, , -, -, ;

[IO_RWw_Bit]
RegNo = 5;
EntryNum = 4;
BEntry1 = Setting data format CH1, RWwBit0, 0~4000, -2000 to 2000;
BEntry2 = Setting data format CH2, RWwBit1, 0~4000, -2000 to 2000;
BEntry3 = Setting data format CH3, RWwBit2, 0~4000, -2000 to 2000;
BEntry4 = Setting data format CH4, RWwBit3, 0~4000, -2000 to 2000;

RegNo = 6;
EntryNum = 4;
BEntry1 = A/D conversion enable/disable specification CH1, RWwBit0, A/DA/D conversion disable, A/DA/D conversion enable;
BEntry2 = A/D conversion enable/disable specification CH2, RWwBit1, A/DA/D conversion disable, A/DA/D conversion enable;
BEntry3 = A/D conversion enable/disable specification CH3, RWwBit1, A/DA/D conversion disable, A/DA/D conversion enable;
BEntry4 = A/D conversion enable/disable specification CH4, RWwBit1, A/DA/D conversion disable, A/DA/D conversion enable;

[IO_Info_RWr]
Comment = Analog to digital converter module;
EntryNum = 5;
WEntry1 = CH.1 Digital output value, RWr0, 1, , 0, , -, -, , , , ;
WEntry2 = CH.2 Digital output value, RWr1, 1, , 0, , -, -, , , , ;
WEntry3 = CH.3 Digital output value, RWr2, 1, , 0, , -, -, , , , ;
WEntry4 = CH.4 Digital output value, RWr3, 1, , 0, , -, -, , , , ;
WEntry5 = CH.5 Digital output value, RWr4, 1, , 0, , -, -, , , , ;

[PrmSet1]
Default = PrmSetGrp1;
GrpNum = 1;
PrmSetGrp1 = CH1, 3, GrpItem1, ;

[PrmSet2]
Default = PrmSetGrp2;
GrpNum = 1;
PrmSetGrp2 = CH2, 3, GrpItem2, ;

[GrpItem1]
PEntry1 = Sampling processing, , <-> | <RWw1>, 2, 0, , , Method1;
PEntry2 = Average processing/count average, , <-> | <RWw1>, 2, 0, 1 to 10000, ms, Method2;
PEntry3 = Average processing/time average, , <-> | <RWw1>, 2, 0, 4 to 10000, times, Method3;

[GrpItem2]
PEntry1 = Sampling processing, , <-> | <RWw1>, 2, 0, , , Method4;
PEntry2 = Average processing/count average, , <-> | <RWw1>, 2, 0, 1 to 10000, ms, Method5;
PEntry3 = Average processing/time average, , <-> | <RWw1>, 2, 0, 4 to 10000, times, Method6;
```

```
[Method]
Method1 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <0> <0> <0>, , , , , , , , , , ;
Method2 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <1> <0> <Input>, , , , , , , , , , ;
Method3 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <1> <1> <Input>, , , , , , , , , , ;
Method4 = <RWw0Bit9> <RWw0Bit1> <RWw2>, <0> <0> <0>, , , , , , , , , , ;
Method5 = <RWw0Bit9> <RWw0Bit1> <RWw2>, <1> <0> <Input>, , , , , , , , , , ;
```

A.4.3.2 CSP editing rules

A CSP file shall consist of sections, entries, fields, comments, and blank characters.

A.4.3.2.1 Section

Each section in a CSP file shall be separated by the delimiters defined in Table A.33. A section shall contain one or more entries.

A.4.3.2.2 Entry

Each entry shall start with an entry keyword followed by an equal sign and one or more fields. An entry shall end with a semicolon. Entry keywords are specified in A.4.4. A single line of a CSP file shall not contain multiple entries. Acceptable maximum length of a line is 512 bytes, including the Carriage Return and Linefeed characters at the end. Acceptable maximum length of a single entry is 1,024 bytes. Characters beyond the maximum length are discarded. An entry may spread over multiple lines.

A.4.3.2.3 Field

Fields shall be delimited with a comma. An optional field shall be filled with a field data, space characters, tab characters, or nothing. A field containing a single hyphen ("-") indicates a required field that does not contain a specific setup value.

A.4.3.2.4 Comment

Comments shall be preceded by a hash sign character (“#”) and ended by the end of the line. A hash sign character following the backslash character ("\") shall be treated as part of the comment rather than a character representing the beginning of the comment.

A.4.3.2.5 Blank characters

The blank characters used in the CSP file shall be:

- Space character, and
- Tab character.

A.4.3.3 CSP data types

A.4.3.3.1 String

String is a character array with NULL character termination. The length of the string includes the NULL character.

A.4.3.3.2 CSPDate

CSPDate type shall be of the format yyyy/mm/dd, where yyyy is the four-digit value for year, mm is the two-digit value for month, and dd is the two-digit value for day. This string value shall have a length of 11 bytes.

EXAMPLE The date January 19th of 1999 is represented as 1999/01/19.

A.4.3.3.3 CSPTime

CSPTime type shall be of the format hh:mm:ss, where hh is the two-digit value for hours in a 24-hour format, mm is the two-digit value for minutes, and ss is the two-digit value for seconds. This string value shall have a length of 11 bytes.

EXAMPLE The time 6:23:44 PM is represented as 18:23:44.

A.4.3.3.4 INT

INT is a 16-bit integer value either in decimal, in hexadecimal notation, or in hexadecimal2 notation. The character array "0x" shall precede the integer values in hexadecimal notation. The hexadecimal2 notation does not use the preceding character array "0x".

A.4.3.3.5 CSPFileVersion

CSPFileVersion type shall be in the format major_version.minor_version. Both major_version and minor_version shall be positive integers in decimal notation.

File version having a larger major_version value shall indicate a newer version file than one having a smaller major_version value. Where the major_version values are the same, the file version having a larger minor_version value shall indicate a newer file than one having a smaller minor_version value.

A.4.4 CSP requirements

A.4.4.1 File Section

The file section contains information about the CSP file. The section delimiter of the file section shall be [File]. Table A.34 shows file section format.

Table A.34 — File section format

Entry name	Entry keyword	Data type	Required/optional
File comment	FileComment	String (65 bytes maximum)	Optional
File creation date	CreateDate	CSPDate	Required
File creation time	CreateTime	CSPTime	Required
Last modified date	ModDate	CSPDate	Optional
Last modified time	ModTime	CSPTime	Optional
CSP file version	Version	CSPFileVersion	Required

Table A.35 shows the information provided by the each entry.

Table A.35 — File section entries

Entry name	Description
File comment	Text displayed by the CC-Link tools
File creation date	CSP file creation date is set by the CSP file developer. The CSP tool uses this value to display the file creation date.
File creation time	CSP file creation time is set by the CSP file developer. The CSP tool uses this value to display the file creation time.
Last modified date	This date defines when the last modification of the CSP file was made. The time may be modified by both the CSP tools and generic text editors.
Last modified time	This time defines when the last modification of the CSP file was made. The time may be modified by both the CSP tools and generic text editors.
CSP file version	CSP file version is set by the CSP file developer. This data is used for file management. Version comparison rule is described in A.4.3.3.5.

EXAMPLE

```
[File]
  FileComment = AD Convert Unit CSP File;
  CreateDate = 1999/01/19;
  CreateTime = 18:50:00;
  ModDate = 1999/01/20;
  ModTime = 13:30:00;
  Version = 1.1;
```

A.4.4.2 Device Section

The device section contains information about the slave station device. Section keyword for the device section shall be [Device]. Table A.36 shows device section format.

Table A.36 — Device section format

Entry name	Entry keyword	Data type	Required/optional
Vendor name	VendName	String (65 bytes maximum)	Required
Vendor identification	VendID	INT (hexadecimal)	Required
Station type	StationType	INT (decimal)	Required
Remote device type	RemDevType	INT (hexadecimal)	Required / Optional[a]
Device model name	DevModel	String (65 bytes maximum)	Required
Version	DevVer	String (8 bytes maximum)	Required
Number of occupied stations	Senyuu	INT (decimal)	Required
Bitmap file	BmpFile	String (257 bytes maximum)	Required
Error code storage register	ErrReg	String (8 bytes maximum)	Optional
Upload/download flag	UpDownLoadF	INT (decimal)	Required
Standby master flag	MasterFlg	INT (decimal)	Optional

[a] Required if the station type is remote device station; Optional if otherwise.

Table A.37 shows the representation of the section entries.

Table A.37 — Device section entries

Entry name	Description
Vendor name	Vendor name
Vendor identification	Vendor identification shall be assigned to each manufacturer.
Station type	Slave station type code. 0: Remote I/O station 1: Remote device station 2: Intelligent device station including a local station and a standby master station
Remote device type	Remote device station type code. The code shall start from 0x1 and 0x0 shall not be used. The code used is shown in Table A.38.
Device model name	Device model name
Version	Device version number
Number of occupied stations	Number of occupied stations.
Bitmap file	Bitmap file related to the CSP file.
Error code storage register	Register storing the error code
Upload / download flag	The value of this entry indicates whether the device supports upload / download capability. 0: Unsupported 1: Supported
Standby master flag	The value of this entry indicates whether the station can be a standby master station or not. 0: Capable of being a standby master station 1: Not capable of being a standby master station

Table A.38 shows the list of remote device types.

Table A.38 — List of remote device types

Remote device type	Description
0x1	PLC
0x2	Personal computer
0x3	Digital I/O
0x4	Analog I/O
0x5	Positioning
0x6	Thermostat
0x7	HMI
0x8	ID
0x09	Serial converting device
0x1D	CC-Link – CC-Link/LT bridge
0x1F	Protocol converting device
0x20	Inverter
0x21	Servo
0x22	NC
0x23	Robot
0x24	Power distribution control device
0x30	Sensor
0x31	Actuator
0x32	Barcode
0x33	Indicator (weight)
0x34	High-speed counter
0x35	Key switch
0x36	Protocol analyzer
0x37	Aerial transmitter
0x38	Conveyor control device
0x39	Power supply control device
0x3A	Welding machine control device
0x3B	Gas detector
0x3C	Solenoid valve
0x3D	Robot (multi-purpose I/O)
0x3E	Printer control device
0x3F	Motor control device
0x40	Vacuum pump
0x41	Multi-axis controller
0x42	Multi-purpose VME board
0x43	Power supply (for mass flow controller)
0x44	Mass flow controller
0x45	Power reception and distribution device

Remote device type	Description
0x46	Control centre
0x47	Welding control device
0x48	Indicator (multi-purpose)
0x49	PID controller
0x4A	Vacuum gauge
0x4B	Wireless device
0x4C	Digital / Analog I/O

EXAMPLE

```
[Device]
VendName = X Corporation;      # Vendor name
VendID = 0x0;                  # Vendor code
StationType = 1;               # Station type (0: I/O, 1: Remote device, 2: Intelligent device)
RemDevType = 0x4;              # Remote device type
DevModel = AJ65BT-64AD;        # Device model name
DevVer = A;                    # Device version data
Senyuu = 2;                    # Number of occupied stations
BmpFile = remote_dv.bmp;       # Applicable Bmp filename
ErrReg = ;                     # For error display
UpDownLoadF = 1;               # Load applicable
MasterFlg = 0;                 # Cannot function as a standby master station
```

A.4.4.3 RX and RY sections

The RX section contains parameters related to RX. The section delimiter shall be [IO_Info_RX]. The RY section contains parameters related to RY. The section delimiter shall be [IO_Info_RY]. Both sections have the same format. Table A.39 shows the section format.

Table A.39 — RX section and RY section format

Entry name	Entry keyword	Data type	Required/optional
Number of entries	EntryNum	INT (decimal)	Required
Register	BEntry*n*[a]	Composed of fields (see Table A.40)	Required / optional[b]

[a] *n* shall be Arabic numerals in decimal. The decimal numbers shall start at one and be incremented by one for each successive entry. The maximum length of the entry keyword shall be 10 bytes.

[b] If the number of entries is 0, the Register entry may be omitted. Otherwise, required.

The field of the entry "Number of entries" shall contain the number of "Register" entries in decimal notation. The entry "Register" defines the usage of each bit register. If a bit register is not used, the corresponding entry of the register may be omitted.

Table A.40 shows the field format of the Register entry.

Table A.40 — Field format of the Register entry in RX and RY section

Field number	Field name	Data type	Required / optional
1	PrmName	String (65 bytes maximum)	Required
2	RegNo	String (9 bytes maximum)	Required
3	OffName	String (65 bytes maximum)	Optional
4	OnName	String (65 bytes maximum)	Optional

Table A.41 shows the field representations of the Register entry.

Table A.41 — Fields of the Register entry in RX and RY sections

Field name	Description
PrmName	Signal name of RX or RY.
RegNo	Register number of RX or RY. The field for RX shall consist of the string, "RX", and a bit register number in hexadecimal. The field for RY shall start with the string, "RY", rather than "RX."
OffName	The semantics of a bit register when the value of the register is 0.
OnName	The semantics of a bit register when the value of the register is 1.

EXAMPLE

```
[IO_Info_RX]
EntryNum = 5;
BEntry1 = CH.1 A/D conversion complete flag, RX0, ,A/D conversion complete;
BEntry2 = CH.2 A/D conversion complete flag, RX1, ,A/D conversion complete;
BEntry3 = CH.3 A/D conversion complete flag, RX2, ,A/D conversion complete;
BEntry4 = CH.4 A/D conversion complete flag, RX3, ,A/D conversion complete;
BEntry5 = Initial data processing request flag, RX18, , initial data processing request;

[IO_Info_RY]
EntryNum = 4;
BEntry1 = Offset/gain value, RYO, user setting, factory setting;
BEntry2 = Voltage/current value, RY1, voltage, current;
BEntry3 = Initial data processing complete flag, RY18, , initial data processing complete;
BEntry4 = Initial data setting complete flag, RY19, , initial data setting complete;
```

A.4.4.4 RWr and RWw sections

A.4.4.4.1 General

The RWr section contains parameters related to RWr. The section delimiter shall be [IO_Info_RWr]. The RWw section contains parameters related to RWw. The section delimiter shall be [IO_Info_RWw].

The RWr section and the RWw section may be used with other sections according to the usage of RWr or RWw. The relations between the sections are represented with reference by name.

NOTE The further usage is described in BAP-05028.

The following combinations of the sections are used:

— RWr or RWw section;

— RWr or RWw section, Bit register section;

- RWr or RWw section, Select parameter section;
- RWr or RWw section, Select parameter section, Select parameter section;
- RWr or RWw section, Select parameter section, Parameter set section, Group item section;
- RWr or RWw section, Select parameter section, Parameter set section, Group item section, Method section;
- RWr or RWw section, Select parameter section, Command pattern section;
- RWr or RWw section, Select parameter section, Command pattern section, Bit register section;
- RWr or RWw section, Parameter set section, Group item section;
- RWr or RWw section, Parameter set section, Group item section, Method section.

A.4.4.4.2 RWr and RWw section format

Both RWr section and RWw sections have the same format. Table A.42 shows the section format for RWr and RWw.

Table A.42 — RWr and RWw section format

Entry name	Entry keyword	Data type	Required/optional
Comment	Comment	String (65 bytes maximum)	Optional
Number of entries	EntryNum	INT (decimal)	Required
Register	WEntry*n*[a]	Composed of fields (see Table A.43)	Required / optional[b]

[a] The entry keyword shall consist of the combination of the character array prior to *n* representing Arabic numerals (decimal number). The decimal numbers shall start at one and be incremented by one for each successive entry. The maximum length of the entry keyword shall be 10 bytes.

[b] If the number of entries is 0, the Register entry may be omitted. Otherwise, it is required.

The entry "Comment" is used for commenting. The field of the entry "Number of entries" shall contain the number of "Register" entries in decimal notation. The entry "Register" defines the usage of each word register. If a word register is not used, the corresponding entry of the register may be omitted. If there is more than one entry which has the same entry keyword, the first entry shall be considered valid and the others shall be ignored.

Table A.43 shows the field format of the Register entry.

Table A.43 — Field format of the register entry in RWr and RWw section

Field number	Field	Data type	Required / optional
1	PrmName	String (65 bytes maximum)	Required
2	RWNo	String (129 bytes maximum)	Required
3	DatTypFlg	INT (decimal)	Required
4	SecName	String (16 bytes maximum)	Optional
5	Default	Depends on DatTypFlg field	Optional
6	Range	String (129 bytes maximum)	Optional
7	UnitStr	String (17 bytes maximum)	Optional
8	HandReq	String (9 bytes maximum)	Required
9	HandAns	String (9 bytes maximum)	Required
10	ReadReg	String (8 bytes maximum)	Optional
11	ILock	String (129 bytes maximum)	Optional
12	InterReq	String (9 bytes maximum)	Optional
13	InterAns	String (9 bytes maximum)	Optional

Table A.44 shows the field descriptions of the Register entry.

Table A.44 — Field descriptions of the Register entry of RWr and RWw sections

Field	Description
PrmName	Signal name of RWw or RWr
RWNo	Register number of RWw or RWr. The field for RWw shall consist of the string, "RWw", and a word register number in hexadecimal. The field for RWr shall consist of the string, "RWr", and a word register number in hexadecimal. Two consecutive word registers may be used as a single 32-bit data value (value 6 to 9 of DatTypFlg). For this purpose, the field shall consist of the first register number with the preceding string, a plus character ("+"), and the second register number with the preceding string. A plus character and the second register number with the preceding string may be omitted. If entries have a duplicate register number, a duplicating entry shall be invalid and ignored.
DatTypFlg	The semantics of data type for RWw or RWr register. The values from 0 to 16 shall be used, and 17 to 65535 shall not be used. 0: 16-bit unsigned hexadecimal data 1: 16-bit signed decimal data 2: 16-bit unsigned decimal data 3: Bit data used in conjunction with Bit register section (see A.4.4.5). 4: 8-bit unsigned hexadecimal data for the upper 8 bits (bit8 to bit15)of a word register 5: 8-bit unsigned hexadecimal data for the lower 8 bits (bit0 to bit 7) of a word register 6: 32-bit signed decimal data 7: 32-bit unsigned decimal data 8: 32-bit unsigned hexadecimal data 9: 32-bit floating point data 10: 4-bit (bit0 to bit3) unsigned hexadecimal data 11: 4-bit (bit4 to bit7) unsigned hexadecimal data 12: 4-bit (bit8 to bit11) unsigned hexadecimal data 13: 4-bit (bit12 to bit15) unsigned hexadecimal data 14: Byte array 15: Word array 16: Dummy data type
SecName	A relating section name. The field shall be a select parameter section name, a parameter set section name or nothing. A select parameter section name shall be expressed by the combination of the string array "SelectPrm" and the decimal number. A parameter set section name shall be expressed by the combination of the string array "PrmSet" and the decimal number. If no data is present, the register entry does not have relating sections.
Default	The default value specified for a word register. Its setting unit depends on the data type indicated in DatTypFlag.
Range	A setting range A range shall be expressed as a value, the starting point and the ending point separated by a tilde character ("~"), or multiple ranges. Blank characters shall be used as a delimiter if multiple ranges are specified.

Field	Description
UnitStr	Unit of the data value to be displayed
HandReq	The RY handshake device name and its on/off status. The field shall consist of the string, "RY", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off". The field shall contain a hyphen character ("-") when no handshake device exists.
HandAns	The RX handshake device name and its on/off status. The field shall consist of the string, "RX", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off". The field shall contain a hyphen character ("-") when no handshake device exists.
ReadReg	The word register for reading. The field shall consist of the string, "RWr" and a RWr number in hexadecimal.
ILock	The interlock device name and its on/off status. The interlock device name and status shall be represented in the format RY*n*:*status* or RX*n*:*status* where *n* is the register number in hexadecimal2 notation and *status* is on or off. The field may have more than one interlock device name and status. If multiple interlock devices exist, each interlock device name and status shall be in angle brackets ("<" and ">") and be listed in the field.
InterReq	The request device name used during a sequence and its on/off status. The field shall consist of the string, "RY", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".
InterAns	The reply device name used during a sequence and its on/off status. The field shall consist of the string, "RX", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".

EXAMPLE1 Example field values of RWNo

```
RWw2
RWw2+RWw3
```

EXAMPLE2 Example entries for 32 bit data

```
WEntry1 = A, RWw1+RWw2, 6, , 0, , , -, -, , , , ;
WEntry2 = B, RWw2, 0, , 0, , , -, -, , , , ;           # This entry shall be ignored.
```

EXAMPLE3 Example entries for 32 bit data

```
WEntry1 = A, RWw1, 0, , 0, , , -, -, , , , ;
WEntry2 = B, RWw1+RWw2, 6, , 0, , , -, -, , , , ;      # This entry shall be ignored.
```

EXAMPLE4 Example entries for 32 bit data

```
WEntry1 = A, RWw1, 0, , 0, , , -, -, , , , ;
WEntry2 = B, RWw1+RWw2, 6, , 0, , , -, -, , , , ;      # This entry shall be ignored.
WEntry3 = C, RWw2, 0, , 0, , , -, -, , , , ;           # This entry shall not be ignored.
```

EXAMPLE5 Example field values of Range

```
100
100~200
100 150~200 300
```

EXAMPLE6 Example field values of ILock

```
RYC:on
RXC:off
<RYC:on><RXE:off>
```

EXAMPLE7 Example of RWw section

```
[IO_Info_RWw]
Comment = Positioning module (number of occupied stations: 4) ;     # Comment
EntryNum = 5 ;    # Number of entries
WEntry1 = One axis positioning start number,     # Signal (parameter) name
        RWw0,       # Register number "RWw" + Arabic numerals "0"
        2,          # As 16-bit unsigned decimal data
         ,          # Parameter selection usage is not used
        0, ,        # Default value, no specific setting range
         ,          # String for unit display
        -, -,       # no handshake registers for RX and RY
         ,          # Word register for read
         ,          # Interlock
         , ;        # Inter-sequence request and answer registers
WEntry2 = One axis overwrite, RWw1, 0, , 100, 0 to 300, %, -, -, , , , ;
WEntry3 = One axis current position change value, RWw2+RWw3, 6, ,  0,    , , -, -, , , , ;
WEntry4 = One axis speed change value, RWw4+RWw5, 7, ,  0,    , , -, -, , , , ;
WEntry5 = One axis JOG speed, RWw6+RWw7, 7, ,  0,    , , -, -, , , , ;
```

A.4.4.5 RWw and RWr bit register section

The RWw bit register section contains parameters related to RWw in bitwise use. The section delimiter shall be [IO_RWw_Bit]. The RWr bit register section contains parameters related to RWr in bitwise use. The section delimiter shall be [IO_RWr_Bit]. These sections shall be used in conjunction with the DatTypFlg field with value 3 of the WEntry shown in Table A.43. Both RWw and RWr bit register sections have the same format. Table A.45 shows the section format.

Table A.45 — RWr and RWw bit register section format

Entry name	Entry keyword	Data type	Required/optional
Pattern number	CmdPtrn	INT (decimal)	Optional
Register number	RegNo	INT (hexadecimal2)	Required
Number of entries	EntryNum	INT (decimal)	Required
Bit register	BEntry*n*[a]	Composed of fields (see Table A.46)	Required

[a] The entry keyword shall consist of the combination of the character array prior to *n* representing Arabic numerals (decimal number). The decimal numbers shall start at one and be incremented by one for each successive entry. The maximum length of the entry keyword shall be 10 bytes.

The field of "Pattern number" contains the name of a command parameter section. The field is used in conjunction with the command parameter section when the usage of the section is bitwise.

The field of "Register number" shall contain a word register number of RWw for RWw bit register section and RWr for RWr bit register section. The value of the field shall be from 0 to F.

The field of "Number of entries" shall contain the number of "Bit register" entries in decimal notation.

The fields of "Bit register" define the usage of a bit in a word register. If a bit in a word register is not used, the corresponding entry of the bit may be omitted.

Table A.46 shows the field format of the Bit register entry.

Table A.46 — Field format of the Bit register entry in RWr and RWw bit register section

Field number	Field	Data type	Required / optional
1	PrmName	String (65 bytes maximum)	Required
2	RWBit	String (9 bytes maximum)	Required
3	OffName	String (65 bytes maximum)	Optional
4	OnName	String (65 bytes maximum)	Optional

Table A.47 shows the field descriptions of the Bit register entry.

Table A.47 — Field descriptions of the bit register entry in RWr and RWw bit register section

Field	Description
PrmName	Signal name of the bit
RWBit	Bit number The field shall consist of the string "RWwBit" for RWw bit register section and a bit number in hexadecimal. For RWr bit register section, the field shall consist of the string "RWrBit" and a bit number in hexadecimal. The hexadecimal number shall start at one and increment by one until F. If entries have a duplicated bit number, a duplicating entry is invalid and ignored.
OffName	The semantics of a bit when the bit value is 0.
OnName	The semantics of a bit when the bit value is 1.

EXAMPLE

```
[IO_Info_RWw]
Comment = AD converter module (number of occupied stations: 2);     # Comment
EntryNum = 2;                # Number of entries
WEntry1 = Data format,       # Signal (parameter) name
        RWw1,                # Register number
        3,                   # Data type flag
         ,                   # Section name
        0, ,                 # Default vale, setting range
         ,                   # String for unit display
        -, -,                # Handshake RX, RY
         ,                   # Word register for read
         ,                   # Interlock
         ,  ;                # Inter-sequence request and answer registers
WEntry2 = A/D conversion enable/disable specification, RWw2, 3, , 0, , , -, -, , , , ;

[IO_RWw_Bit]
RegNo = 1;
EntryNum = 4;
BEntry1 = Data format setting CH1, RWwBit0, 0 to 4000, -2000 to 2000;
BEntry2 = Data format setting CH2, RWwBit1, 0 to 4000, -2000 to 2000;
BEntry3 = Data format setting CH3, RWwBit2, 0 to 4000, -2000 to 2000;
BEntry4 = Data format setting CH4, RWwBit3, 0 to 4000, -2000 to 2000;

RegNo = 2;
EntryNum = 4;
BEntry1 = A/D conversion enable/disable CH1, RWwBit0, A/D conversion disable, A/D conversion enable;
BEntry2 = A/D conversion enable/disable CH2, RWwBit1, A/D conversion disable, A/D conversion enable;
BEntry3 = A/D conversion enable/disable CH3, RWwBit2, A/D conversion disable, A/D conversion enable;
BEntry4 = A/D conversion enable/disable CH4, RWwBit3, A/D conversion disable, A/D conversion enable;
```

A.4.4.6 Select parameter section

The select parameter section contains the definitions of the remote register usage. The section delimiter shall be [SelectPrm*n*] where *n* is Arabic numerals in decimal. The decimal numbers shall start at one and be incremented by one for each successive select parameter section. Table A.48 shows the section format.

Table A.48 — Select parameter section format

Entry name	Entry keyword	Data type	Required/optional
Number of entries	EntryNum	INT (decimal)	Required
Select parameter	SEntry*n*[a]	Composed of fields (see Table A.49)	Required

[a] The entry keyword shall consist of the combination of the character array prior to *n* representing Arabic numerals (decimal number). The decimal numbers shall start at one and be incremented by one for each successive entry. The maximum length of the entry keyword shall be 16 bytes.

The field of the entry "Number of entries" shall be the number of "Select parameter" entries in decimal notation. The entry "Select parameter" defines the choices of the remote register usage.

Table A.49 shows the field format of the Select parameter entry.

Table A.49 — Field format of Select parameter entry

Field number	Field	Data type	Required / optional
1	PrmName	String (65 bytes maximum)	Required
2	CodeNo	INT	Optional
3	UnitStr	String (17 bytes maximum)	Optional
4	SecName	String (16 bytes maximum)	Optional
5	UpDwnTyp	INT (decimal)	Optional

Table A.50 shows the field description of the Select parameter entry.

Table A.50 — Field description of Select parameter entry

Field	Description
PrmName	Selection name
CodeNo	Code number Value of the remote register either in decimal notation or in hexadecimal notation.
UnitStr	Unit of the data value to be displayed
SecName	A relating section name. The field contains a select parameter section name, a parameter set section name, a command pattern section name or nothing. A select parameter section name shall be expressed by the combination of the string array "SelectPrm" and the decimal number. A parameter set section name shall be expressed by the combination of the string array "PrmSet" and the decimal number. A command pattern section name shall be expressed by the combination of the string array "CmdPtrn" and the decimal number. If no data is present, the register entry does not have relating sections.
UpDwnTyp	Upload/download support indication The field indicates whether or not the entry is included in the targets of the actions, upload and download. The value shall be interpreted as follows: 0: Not applicable 1: Upload 2: Download 3: Both upload and download

EXAMPLE Example of the select parameter section

```
#Example Remote Register Select Parameter section
[SelectPrm1]
EntryNum = 3;
SEntry1 = No monitoring,       # Selection name
0x0,                        # Code number
,                           # Unit
,                           # Relating section name
0;                          # Upload/download indicator
SEntry2 = Output frequency, 0x1, 0.01Hz, , 0;
SEntry3 = Output current, 0x2, 0.01A, , 0;
```

A.4.4.7 Parameter set section

The parameter set section contains a group of parameter settings used with a remote register. The section delimiter shall be [PrmSet*n*] where *n* is Arabic numerals in decimal. The decimal numbers shall start at one and be incremented by one for each successive parameter set section exists. Table A.51 shows the section format.

Table A.51 — Parameter set section format

Entry name	Entry keyword	Data type	Required/optional
Default parameter set	Default	String (16 bytes maximum)	Required
Number of parameter set groups	GrpNum	INT (decimal)	Required
Parameter set	PrmSetGrp*n*[a]	Composed of fields (see Table A.52)	Required

[a] The entry keyword shall consist of the combination of the character array prior to *n* representing Arabic numerals (decimal number). The decimal numbers shall start at one and be incremented by one for each successive entry exists. The maximum length of the entry keyword shall be 16 bytes.

The field of the entry "Default parameter set" shall contain a name of default parameter setting. The field of the entry "Number of parameter set groups" shall contain the number of "Parameter set" entries in decimal notation. The entry "Parameter set" shows relating sections information where the actual parameter setting is defined.

Table A.52 shows the field format of the Parameter set entry.

Table A.52 — Field format of Parameter set entry

Field number	Field	Data type	Required / optional
1	GrpName	String (65 bytes maximum)	Required
2	GrpNum	INT	Required
3	GrpItem	String (16 bytes maximum)	Required
4	Method	String (16 bytes maximum)	Optional

Table A.53 shows the field descriptions of the Parameter set entry.

Table A.53 — Field description of Parameter set entry

Field	Description
GrpName	Group name
GrpNum	Number of entries defined in the group item section specified in the field "GrpItem"
GrpItem	Group item section name The field contains a group item section name. The section name shall be expressed by the combination of the string "GrpItem" and the decimal number.
Method	Method name used to change parameter sets The field contains a method entry name. The entry name shall be expressed by the combination of the string "Method" and the decimal number.

EXAMPLE

```
[PrmSet1]
Default = PrmSetGrp1;   # Default parameter set
GrpNum = 4;             # Number of parameter set groups to be defined
PrmSetGrp1 =
  Pr.0 to 99,           # Item name
  89,                   # Number of entries defined in the section GrpItem1
  GrpItem1,             # Group item section name
  Method1;              # Method name of changing parameter sets
```

PrmSetGrp2 = Pr.100 to 159/Pr.200 to 231/Pr.900 to 905, 91, GrpItem2, Method2;
PrmSetGrp3 = Pr.160 to 199/Pr.232 to 285, 59, GrpItem3, Method3;

A.4.4.8 Group item section

The group item section contains parameters related to the parameter set section. The section delimiter shall be [GrpItem*n*] where *n* is Arabic numerals in decimal. The decimal numbers shall start at one and be incremented by one for each successive group item section that exists.

The section has one type of required entry called "parameter entry". The entry keyword shall consist of the combination of the character array "PEntry" and a number in decimal. The decimal number shall start at one and increment by one as the successive parameter entry exists. The maximum length of the entry keyword shall be 16 bytes.

Table A.54 shows the field format of the parameter entry.

Table A.54 — Field format of Group item section parameter entry

Field number	Field	Data type	Required / optional
1	ItemName	String (65 bytes maximum)	Required
2	PrmNo	String (17 bytes maximum)	Optional
3	RegNo	String (257 bytes maximum)	Optional
4	DatTypFlg	INT (decimal)	Required
5	Default	Depends on the data type.	Optional
6	Range	String (129 bytes maximum)	Optional
7	UnitStr	String (17 bytes maximum)	Optional
8	Method	String (16 bytes maximum)	Optional

Table A.55 shows the field descriptions of the Parameter entry.

Table A.55 — Field description of Group item section parameter entry

Field	Descriptions
ItemName	Parameter name
PrmNo	Parameter number
RegNo	Read data register number(s) and write data register number(s). The field shall contain a list of read data register and a list of write data register Those two lists shall be delimited by a vertical bar character ("|"). The lists shall contain one or more data registers. A read data register shall consist of the string "RWr" and the read data register number in hexadecimal, in angle brackets ("<" and ">"). A write data register shall consist of the string "RWw" followed by the write data register in hexadecimal, in angle brackets. A hyphen character in angle brackets shall represent no data register used.
DatTypFlg	The semantics of data type for the parameter (see DatTypFlg in Table A.44).
Default	The default value specified for the parameter. Its setting unit depends on the data type indicated in DatTypFlag.
Range	A setting range (see Range in Table A.44).
UnitStr	Unit of the data value to be displayed
Method	The name of the method used to change parameter set. The field contains an entry keyword used in the method section (see A.4.4.9).

EXAMPLE1 Example of RegNo field values

```
<RWr3>|<RWw3>
<RWr3><RWr4>|<RWw3>
<RWr3>|<->
```

EXAMPLE2 Example of group item section

```
[GrpItem1]
PEntry1 =
  Torque boost,  # parameter name
  0,                             # parameter number
  <RWr3> | <RWw3>,               # register number (<read data register> | <write data register>)
  2,                             # data type flag 2: 16-bit unsigned decimal data
  6, 0~30,                       # default value, setting range
  %,                             # unit to be displayed
  Method1001;                    # method name to change parameter set
PEntry2 = High limit frequency,     1, <RWr3> | <RWw3>, 1, 120, 0~120, Hz, Method1002;
PEntry3 = Low limit frequency,      2, <RWr3> | <RWw3>, 1, 0, 0~120, Hz, Method1003;
```

A.4.4.9 Method section

The method section contains parameters to define a procedure used to change parameter sets. The section delimiter shall be [Method].

The section has one type of required entry named method entry. The entry keyword shall consist of the combination of the character array "Method" and a number in decimal. The maximum length of the entry keyword is 16 bytes.

Table A.56 shows the field format of the Method entry.

Table A.56 — Field format of Method entry

Field number	Field name	Data type	Required / optional
1	RegNo	String (256 bytes maximum)	Required
2	Code	String (32 bytes maximum)	Required
3	ReqRY	String (9 bytes maximum)	Optional
4	EndRX	String (9 bytes maximum)	Optional
5	ErrEnd	String (9 bytes maximum)	Optional
6	ReplyRW	String (16 bytes maximum)	Optional
7	TrueCnd	String (16 bytes maximum)	Optional
8	ErrReg	String (8 bytes maximum)	Optional
9	TrueCnd2	String (9 bytes maximum)	Optional
10	ILock	String (128 bytes maximum)	Optional
11	InterReq	String (9 bytes maximum)	Optional
12	InterAns	String (9 bytes maximum)	Optional

Table A.57 shows the field descriptions of the Method entry.

Table A.57 — Field description of Method entry

Field name	Descriptions
RegNo	Register name used to set a command The register name shall consist of a string "RWw" or "RWr" and a register number in hexadecimal. The field may have more than one register name. If multiple registers exist, each register name shall be in angle brackets ("<" and ">") and be listed in the field. When only a bit of a register is used for this purpose, the register name shall consist of a string "RWw" or "RWr", a register number in decimal, a string "Bit" and a bit register number.
Code	Command code The command codes in the field correspond to the registers specified in RegNo. Command code shall be either a hexadecimal number preceded by a string "0x" or a string "Input". A string "Input" shall be interpreted as a representation that the command code is entered by the user. The field may have more than one command code. If multiple command codes exist, each command code shall be in angle brackets ("<" and ">") and be listed in the field. A register may have both a command for read and a command for write. These commands shall be delimited by a vertical bar ("|").
ReqRY	Command operation request RY The field shall consist of a string "RY", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".
EndRX	Command operation complete RX The field shall consist of a string "RX", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".
ErrEnd	Command operation abnormal completion RX The field shall consist of a string "RX", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".
ReplyRW	Reply code storage register The field shall consist of a string, RWr or RX, and a register number in hexadecimal.
TrueCnd	Success judgement conditions The field corresponding to the ReplyRW in word register shall consist of a comparison operator and a value. The comparison operator shall be either == (equal), != (not equal), < (less than), > (greater than), <= (less than or equal), or >= (greater than or equal). The field corresponding to the ReplyRW in bit register shall consist of a comparison operator "==" and a string "on" or "off". The field with no contents shall be interpreted as a success of the judgement.
ErrReg	Error code storage register The field shall consist of a string, RWr, and a register number in hexadecimal.
TrueCnd2	Success judgement condition upon command normal completion The field shall consist of a string "RX", a bit register number in hexadecimal, a colon character (":"), and the string either "on" or "off".
ILock	The interlock device name and its on/off status (see Table A.44).
InterReq	The request device name used during a sequence and its on/off status (see Table A.44).
InterAns	The reply device name used during a sequence and its on/off status (see Table A.44).

EXAMPLE

```
[Method]
Method1 =
  <RWw2> <RWw3>,              # Command setting register
  <0x7F | 0xFF> <0x00>,       # <Read code | Write code> setting for RWw2, code for RWw3
  RYF,                        # Read request RY | Write request RY
  RXF,                        # Read completion RX | Write completion RX
      ,                       # Command operation abnormal completion RX
  RWr2,                       # Read result storage (RWr) | Write result storage (RWr)
  == 0,                       # Success judgement condition
      ,                       # Error code storage register
      ,                       # Success judgement condition upon command normal completion
      ,                       # Interlock
      ,   ;                   # Inter-sequence request and answer registers
Method2 = <RWw2> <RWw3>, <0x7F | 0xFF> <0x01>, RXF:ON, RYF:ON, , RWr2, == 0, , , , , ;
Method3 = <RWw2> <RWw3>, <0x7F | 0xFF> <0x02>, RXF:ON, RYF:ON, , RWr2, == 0, , , , , ;
Method101 = <RWw2> <RWw3>, <0x00 | 0x80> <0 | Input>, RXF:ON, RYF:ON, , RWr2, == 0, , , , , ;
Method102 = <RWw2> <RWw3>, <0x01 | 0x81> <0 | Input>, RXF:ON, RYF:ON, , RWr2, == 0, , , , , ;
Method103 = <RWw2> <RWw3>, <0x02 | 0x82> <0 | Input>, RXF:ON, RYF:ON, , RWr2, == 0, , , , , ;
Method201 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <0> <0> <0>, , , , , , , , , , ;
Method202 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <1> <0> <Input>, , , , , , , , , , ;
Method203 = <RWw0Bit8> <RWw0Bit0> <RWw1>, <1> <1> <Input>, , , , , , , , , , ;
Method204 = <RWw0Bit9> <RWw0Bit1> <RWw2>, <0> <0> <0>, , , , , , , , , , ;
Method205 = <RWw0Bit9> <RWw0Bit1> <RWw2>, <1> <0> <Input>, , , , , , , , , , ;
Method206 = <RWw0Bit9> <RWw0Bit1> <RWw2>, <1> <1> <Input>, , , , , , , , , , ;
```

A.4.4.10 Command pattern section

The Command pattern section contains a set of the semantics of remote word registers. The section delimiter shall be [CmdPtrn*n*] where *n* is Arabic numerals in decimal. The decimal numbers shall start at one and be incremented by one as each successive command pattern sections are made.

The format of the Command pattern section is the same as RWr and RWw sections (see A.4.4.4.2).

EXAMPLE

```
[CmdPtrn1]
Comment = Initial setting;
EntryNum = 2;
WEntry1 = Initial setting, RWw0, 4, , 0, , , -, -, , , , ;
WEntry2 = HMI internal device use enable/disable, RWw0, 3, , 0, , , -, -, , , , ;
[CmdPtrn2]
Comment = Continuous read setting;
EntryNum = 3;
WEntry1 = Continuous read setting,  -, 16, PrmSet2, , , , RY39:ON, RX39:ON, , , , ;
WEntry2 = Number of HMI internal device points to read from,     RWw0, 5, , , 1~6, points, -, -, , , , ;
WEntry3 = Number for the head HMI internal device to read from,       RWw1, 2, , , 0~1023, ,-, -, , , , ;
```

Annex B
(informative)

UML terminology and notation

B.1 General

This annex explains the UML notation used in this part of ISO 15745 but not explained in ISO 15745-1. It is provided to aid the reader in understanding this part of ISO 15745.

NOTE Additional notations are explained in ISO 15745-1:2003, Amendment 1.

B.2 UML notation

The following notation described in Table B.1 is used in the UML class diagrams (see clause 6).

Table B.1 — UML notation

Symbol	Description
→	Navigability The model element at the tail of the arrow has a direct relationship with the model element at the head of the arrow.

Bibliography

[1] BAP-05026, CC-Link specifications (Overview, Protocol) – CC-Link Partner Association

[2] BAP-05027, CC-Link specifications (Implementation) – CC-Link Partner Association

[3] BAP-05028, CC-Link specifications (Profile) – CC-Link Partner Association

[4] BAP-05031, CC-Link/LT specifications – CC-Link Partner Association

[5] SEMI E54.12-0701, Specification for Sensor/Actuator Network Communications for CC-Link – Semiconductor Equipment and Materials International (SEMI), SEMI standard

ICS 25.040.40

Price based on 88 pages