

TECHNICAL REPORT
OF
THE JAPAN ELECTRICAL MANUFACTURERS' ASSOCIATION

JEM-TR 214

FA Control Network [FL-net (OPCN-2)]
Device profile common specification

Established on November 28, 2000



社団法人日本電機工業会

The Japan Electrical Manufacturers' Association

Foreword for the English edition

This document is English translation of JEM-TR 214 that is written in Japanese for convenience to users and developers of FL-net who feel difficult to understand Japanese language. Therefore, the JEM-TR 214 (Japanese) is original while this document is for reference only.

This document may be revised while the original is unchanging to improve English expression without noticeable version change. The working group on English documentation of the special committee for FL-net promotion at JEMA, which worked on the translation, welcomes comments and questions on English expression, which will help it for future improvement.

Foreword

This is a technical report of the Japan Electrical Manufacturers' Association deliberated by the Network Subcommittee, the Programmable Controller Technical Committee and the Standardization Committee, and established by the Steering Technical Committee.

JEM 1479 has the following appendices:

Appendix 1 (Informative) Summary of transfer syntax format ASN.1

Appendix 2 (Informative) Installation of items read by log data read service

CONTENTS

	PAGE
PREFACE	1
1. SCOPE	1
2. REFERRED STANDARD	1
3. DEFINITION	1
4. GENERAL REQUIREMENTS	2
4.1 NETWORK PARAMETERS.....	2
4.2 SYSTEM PARAMETERS	2
4.3 DEVICE COMMUNICATION I/O INFORMATION.....	4
4.4 OTHERS	4
5. DEVICE COMMUNICATION INFORMATION	4
5.1 SYSTEM PARAMETERS.....	4
5.1.1 <i>General format of system parameters</i>	4
5.1.2 <i>System parameters-- Example 1 (PC)</i>	6
5.1.3 <i>System parameters-- Example 2 (CNC)</i>	12
5.2 DEVICE COMMUNICATION I/O INTERFACE	15
5.2.1 <i>Resource of device communication I/O interface</i>	15
5.2.2 <i>Status / mod</i>	17
5.2.3 <i>Upload/download</i>	18
5.2.4 <i>Implementation of message services</i>	21
5.2.5 <i>Transparent mode message</i>	22
5.2.6 <i>Security-related function</i>	22
APPENDIX 1 (INFORMATIVE) SUMMARY OF ASN.1 TRANSFER SYNTAX FORMAT	23
APPENDIX 2 (INFORMATIVE) IMPLEMENTATION OF ITEMS READ BY THE "LOG DATA READ" SERVICE	26
EXPLANATION	28

Device profile common specification for FA control network [FL-net (OPCN-2)]

Preface

This technical report defines common items on device profile for FA control network devices that install JEM 1479 “FA control network [FL-net (OPCN-2)]-- protocol specification” (hereafter referred to as FA link protocol) to interconnect each other by means of the FA link protocol, while the technical report does not define any standard device profile specific for any device type.

Items to be defined in this report are system parameters and I/O information. Items related to the system parameters contain contents of information, document syntax, and transfer syntax. Items related to the I/O information for device communications contain contents of information and document syntax vendors should disclose.

1. Scope

This technical report defines device profile common specifications accompanying the FA link protocol.

Remarks In this report, the network means FA control network (OPCN-2/FL-net).

2. Referred standard

The following standards are referred to in this report (JEM-TR 214), and constitute some portions of this technical report. The latest versions of these standards and technical report shall apply.

JEM 1479	Protocol specification for FA control network [FL-net (OPCN-2)]
JEM-TR 213	Implementation guidelines of FA control network [FL-net (OPCN-2)]
JIS X 5603	Open systems interconnection -- Specification of Abstract Syntax Notation One (ASN.1)
JIS X 5604	Information processing systems -- Open Systems Interconnection -- Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

Remarks Items cited from “ISO 8825 : 1987, Information processing systems-- Open systems interconnection-- Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1) ” are equivalent to this standard.

3. Definition

Definitions of main and related terms used in this document are listed below. Others are defined in 3.

Definition of JEM 1479.

- a) **Device communication information** All the information communicated among devices connected to the FA control network. Classified into the following b)-- d) (refer to Figure 1).
- b) **Network parameter** Setting information required for transfer function
- c) **System parameter** Management information to identify devices in the FA control network. A static parameter that does not change whatever the operating status of applications and device status are. Classified into common parameters and device-specific parameters.
- d) **Device communication I / O information** Information that other devices in the FA control network can access upon requests by applications. It contains dynamic information whose contents change according to the operating status of applications or device status.

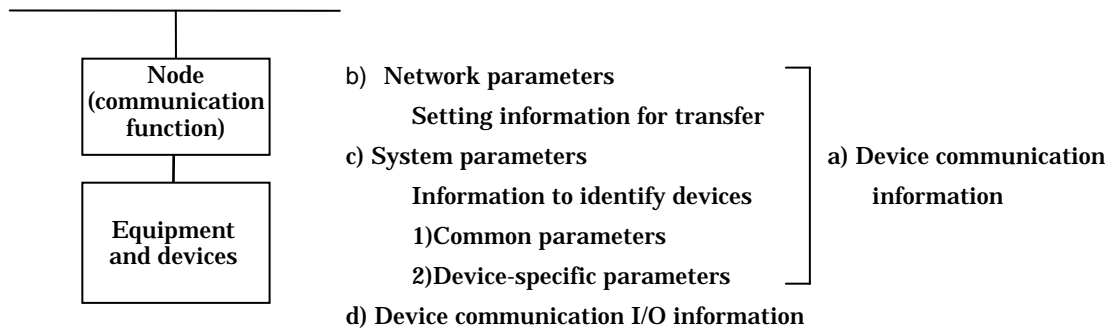


Figure 1 Classification of device information

4. General requirements

General requirements for the device communication information in this document are as follows:

4.1 Network parameters

Network parameters are out of scope of this report as they are defined in JEM 1479.

4.2 System parameters

This document specifies items, document syntax, and transfer syntax of the system parameters and defines the common parameters, while the vendor-specific parameters shall be defined by the vendors.

The transfer syntax shall conform to JIS X 5604 (refer to Appendix 1).

a) Common parameters

This is management information for identifying devices connected to the network.

The use and purpose of the common parameters are as follows:

- Information on network system configuration
- Maintenance information
- Information on system pre-design

Items required for the common parameters are as follows:

- Version of device profile common specification

Remarks The version number takes “1” when the device complies with the first edition of this technical report and increments by one when its implementation is changed.

- System parameter ID characters
- System parameter revision numbers
- Date of system parameter change
- Device type (device category)
- Vendor name
- Product type-form

b) Device-specific parameters

Items and contents specific to the device are defined by vendors. If there is any device-specific parameter, the following item is required:

- Device-specific parameter ID character

Example

- Device configuration in the system
- Functional specifications

c) Data types available as system parameters

Among the general tag class defined in **JIS X 5604** (refer to **Appendix 1**), data types listed below are available as system parameters.

Table 1 Data types available as system parameter

Tag number (hexadecimal)	Data type	Available or not (¹)
00	(Reserved)	N
01	BOOLEAN	Y
02	INTEGER	Y
03	BIT STRING	Y
04	OCTET STRING	Y
05	NULL	Y
06	OBJECT IDENTIFIER	N
07	Object Descriptor	N
08	EXTERNAL	N
09	REAL	N
0A	ENUMERATED	N
0B~0F	(Reserved)	N
10	SEQUENCE and SEQUENCE OF (²)	Y
11	SET and SET OF	N
12	Numeric String	N
13	Printable String	Y
14	Teletex String	N
15	Videotex String	N
16	IA5 String	N
17	UTC Time	N
18	Generalized Time	N

19	Graphic String	N
1A	Visible String	N
1B	General String	N
1C	Character String	N
1D ~ 1E	(Reserved)	N

Note (1) Y: available N: not available
(2) The available structure types are "SEQUENCE"
and "SEQUENCE OF" types only.

4.3 Device communication I/O information

The vendor who implements the FA link protocol into devices shall disclose vendor-specific device communication I/O information with common-formatted documents for convenience of the users.

The device communication I/O information is as follows:

a) Information on resource of device communication I/O information

Device-specific resources accessible from the client side. It contains correspondence of resource and virtual address used for "Read service / Write service" of "Byte block / Word block" in the message transmission.

b) Information on status and mode

Description which status exists, how the status looks, whether the status is controllable, and if controllable, including its relations with the "Stop/Start service" of the message transmission.

c) Information on downloading and uploading

Description whether there exists a program or data to be uploaded or downloaded, and if it exists, how it is uploaded or downloaded

d) Implementation of message services

Description of message services implementation.

e) Information on the transparent mode service of the FA link protocol

Disclosure of the format, meanings, and usage of the transparent mode service if it is implemented.

f) Information on security

Disclosure of security setting and resetting procedures for each resource (if any).

4.4 Others

a) Implementation of items to be read by the log data read service

The vendor shall declare whether each of the items that can be read by the log data read service is implemented or not (refer to **Appendix 2**).

5. Device communication information

5.1 System parameters

5.1.1 General format of system parameters

a) Message transfer service Table 2 shows the message transfer services used to access system parameters:

Table 2 Message transfer service

TCD	Service name	Function
65011	Profile read (request)	Request to read all system parameters at a time
65211	Profile read (response)	Response to the profile read request The response data field is up to 1024 bytes long.

b) Configuration of system parameter System parameters shall be arranged in an order and contents as shown in Table 3.

Table 3 Configuration of system parameters

Name of parameter	Remarks
Common parameters	Parameters defined common to all devices
Device-specific parameters	Parameters (optional) that the vendor freely defines for each device

c) Details of common parameters The requisites for common parameters are shown in Table 4.

Table 4 Details of common parameters

Name of parameter	Characters used [Printable String type] (length), (characters)	Data type [type]	Description example (length),(contents)
Version of device profile common specifications	6, "COMVER"	INTEGER	1, 1
System parameter ID	2, "ID"	Printable String	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	1, 0
System parameter revision date	7, "REVDATE"	[INTEGER], 2, (0001-9999), [INTEGER], 1, (01-12), [INTEGER], 1, (01-31)	2, 1998 1, 9 1, 30
Device type	10, "DVCATEGORY"	Printable String	2, "PC" ⁽³⁾
Vendor name	6, "VENDOR"	Printable String	4, "ABCD"
Product name	7, "DVMODEL"	Printable String	3, "XYZ"

Note ⁽³⁾ Contents of the "Device type" parameter are as follows:

- "PC" : Programmable controller
- "NC" or "CNC" : Numeric controller
- "RC" or "ROBOT" : Robot controller
- "COMPUTER" : Computers such as personal computer, panel computer, workstation, and intelligent display
- "SP-...*" : Specified by the vendor (*...* : alphanumeric characters)
- "OTHER" : Others

Remark 1. The used characters and data type shall conform to JIS X 5603.

Remark 2. The transfer syntax shall be of the SEQUENCE structured type with the whole system parameters, the whole common parameters, the system parameter revision date and the

whole device-specific parameters (optional). (Structuring in the device-specific parameters is optional.)

d) Details of device-specific parameters (when used) Table 5 shows the details of device-specific parameters.

Table 5 Details of device-specific parameters

Name of parameter	Character used	Data type	Content of parameter
Device-specific parameter ID	2, "ID"	Printable String	7, "DEVPARA"
The content of parameters can be freely defined by the vendor.			

5.1.2 System parameters-- Example 1 (PC)

a) Text notation of system parameters with table format Table 6 shows an example of text notation for programmable controller (PC).

Table 6 Text notation example of system parameters with table format

1) SysPara

Name of parameter	Characters used [Printable String type] (length), (characters)	Data type [Type]	Text notation (Length), (content)
Version of device profile common specifications	6, "COMVER"	INTEGER	1, 1
System parameter ID	2, "ID"	Printable String	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	1, 0
System parameter revision date	7, "REVDATE"	[INTEGER],2,(0001-9999), [INTEGER],1,(01-12), [INTEGER],1,(01-31)	2, 1998 1, 9 1, 30
Device type	10, "DVCATEGORY"	Printable String	2, "PLC"
Vendor name	6, "VENDOR"	Printable String	29, "ABCD-XYZ Electric Corporation"
Product name	7, "DVMODEL"	Printable String	5, "PC-M"

2) PcmPara

Name of parameter	Characters used	Data type	Content of parameter
Device-specific parameter ID	2, "ID"	Printable String	7, "DEVPARA"
CPU1 name	8, "CPU1NAME"	Printable String	9, "PMSP35-5N"
CPU2 name	8, "CPU2NAME"	Printable String	9, "PMSP25-2N"
CPU3 name	8, "CPU3NAME"	Printable String	9, "PMSP25-2N"
CPU4 name	8, "CPU4NAME"	Printable String	9, "PMBP20-0N"
Module 105 name	9, "IO105NAME"	Printable String	9, "PMWD64-4N"
Module 106 name	9, "IO106NAME"	Printable String	9, "PMLD01-0N"
Module 107 name	9, "IO107NAME"	Printable String	9, "PMLE01-5N"

b) **Abstract syntax** An abstract syntax notation example for the text notation described in a) is as follows:

1) **Type definition**

```

PcmRecord ::= SEQUENCE
{
  syspara          SysparaType,
  pcmpara          PcmType
}

SysparaType ::= SEQUENCE
{
  nameCOMVER       NameType,
  paraCOMVER       INTEGER,
  nameID           NameType,
  paraID           NameType,
  nameREV          NameType,
  paraREV          INTEGER,
  nameREVDATE     NameType,
  paraREVDATE     DateType,
  nameDVCATEGORY  NameType,
  paraDVCATEGORY  NameType,
  nameVENDOR       NameType,
  paraVENDOR       NameType,
  nameDVMODEL     NameType,
  paraDVMODEL     NameType
}

PcmType ::= SEQUENCE
{
  nameID           NameType,
  paraID           NameType,
  module          SEQUENCE OF ModInfo
                DEFAULT {}
}

NameType ::= PrintableString
DateType ::= SEQUENCE
{
  year            INTEGER,
  month           INTEGER,
  day             INTEGER
}

ModInfo ::= SEQUENCE
{
  nameMODULE      NameType,
  paraMODULE      NameType
}

```

2) Value definition

```

{
  syspara
  {
    nameCOMVER      "COMVER",
    paraCOMVER      1,
    nameID           "ID",
    paraID           "SYSPARA",
    nameREV          "REV",
    paraREV          0,
    nameREVDATE     "REVDATE",
    paraREVDATE     { year 1998,
                    month 9,
                    day 30 },

    nameDVCATEGORY  "DVCATEGORY",
    paraDVCATEGORY  "PLC",
    nameVENDOR       "VENDOR",
    paraVENDOR       "MSTC-JOP Electric Corporation",
    nameDVMODEL      "DVMODEL",
    paraDVMODEL      "PC-M"
  }
  pcmpara
  {
    nameID           "ID",
    paraID           "DEVPARA",
    module
    {
      { nameMODULE   "CPU1NAME",
        paraMODULE   "PMSP35-5N" },
      { nameMODULE   "CPU2NAME",
        paraMODULE   "PMSP25-2N" },
      { nameMODULE   "CPU3NAME",
        paraMODULE   "PMSP25-2N" },
      { nameMODULE   "CPU4NAME",
        paraMODULE   "PMBP20-0N" },
      { nameMODULE   "IO105NAME",
        paraMODULE   "PMWD64-4N" },
      { nameMODULE   "IO106NAME",
        paraMODULE   "PMLD01-0N" },
      { nameMODULE   "IO107NAME",
        paraMODULE   "PMLE01-5N"}
    }
  }
}

```

c) Transfer syntax data array (coding) The following shows a coding example of the abstract syntax described in b).

Identifier Length	Content
30	820131
30	7B
13	06 "COMVER"
02	01 1
13	02 "ID"
13	07 "SYSPARA"
13	03 "REV"
02	01 0
13	07 "REVDATE"
30	0A
02	02 07CE
02	01 09
02	01 1E
13	0A "DVCATEGORY"
13	02 "PC"
13	06 "VENDOR"
13	1D "ABCD-XYZ Electric Corporation"
13	07 "DVMODEL"
13	04 "PC-M"
30	81B1
13	02 "ID"
13	07 "DEVPARA"
30	15
13	08 "CPU1NAME"
13	09 "PMSP35-5N"
30	15
13	08 "CPU2NAME"
13	09 "PMSP25-2N"
30	15
13	08 "CPU3NAME"
13	09 "PMSP25-2N"
30	15
13	08 "CPU4NAME"
13	09 "PMBP20-0N"
30	16
13	09 "IO105NAME"
13	09 "PMWD64-4N"

Identifier	Length	Content									
30	16	<table border="1"> <thead> <tr> <th>Identifier</th> <th>Length</th> <th>Content</th> </tr> </thead> <tbody> <tr> <td>13</td> <td>09</td> <td>"IO106NAME"</td> </tr> <tr> <td>13</td> <td>09</td> <td>"PMLD01-0N"</td> </tr> </tbody> </table>	Identifier	Length	Content	13	09	"IO106NAME"	13	09	"PMLD01-0N"
Identifier	Length	Content									
13	09	"IO106NAME"									
13	09	"PMLD01-0N"									
30	16	<table border="1"> <thead> <tr> <th>Identifier</th> <th>Length</th> <th>Content</th> </tr> </thead> <tbody> <tr> <td>13</td> <td>09</td> <td>"IO107NAME"</td> </tr> <tr> <td>13</td> <td>09</td> <td>"PMLE01-5N"</td> </tr> </tbody> </table>	Identifier	Length	Content	13	09	"IO107NAME"	13	09	"PMLE01-5N"
Identifier	Length	Content									
13	09	"IO107NAME"									
13	09	"PMLE01-5N"									

Remarks The figures are hexadecimal.

d) **Data array on the line** The Data array on the line is shown in **Figure 2**. Data of the relative address 00 are first transmitted sequentially in the order of byte 0 to byte F. Next, data of the relative address 10 are sequentially transmitted in the same way. In this way, finally data of the relative address 130 are sequentially transmitted.

Relative address	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(A)	(B)	(C)	(D)	(E)	(F)
00	30	82	01	31	30	7B	13	06	"C"	"O"	"M"	"V"	"E"	"R"	02	01
10	01	13	02	"I"	"D"	13	07	"S"	"Y"	"S"	"P"	"A"	"R"	"A"	13	03
20	"R"	"E"	"V"	02	01	00	13	07	"R"	"E"	"V"	"D"	"A"	"T"	"E"	30
30	0A	02	02	07	CE	02	01	09	02	01	1E	13	0A	"D"	"V"	"C"
40	"A"	"T"	"E"	"G"	"O"	"R"	"Y"	13	02	"P"	"C"	13	06	"V"	"E"	"N"
50	"D"	"O"	"R"	13	1D	"M"	"S"	"T"	"C"	"."	"J"	"O"	"P"	SP*	"E"	"I"
60	"e"	"c"	"t"	"r"	"i"	"c"	SP*	"C"	"o"	"r"	"p"	"o"	"r"	"a"	"t"	"i"
70	"o"	"n"	13	07	"D"	"V"	"M"	"O"	"D"	"E"	"L"	13	04	"P"	"C"	"."
80	"M"	30	81	B1	13	02	"I"	"D"	13	07	"D"	"E"	"V"	"P"	"A"	"R"
90	"A"	30	15	13	08	"C"	"P"	"U"	"1"	"N"	"A"	"M"	"E"	13	09	"P"
A0	"M"	"S"	"P"	"3"	"5"	"."	"5"	"N"	30	15	13	08	"C"	"P"	"U"	"2"
B0	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"S"	"P"	"2"	"5"	"."	"2"	"C"	"N"
C0	30	15	13	08	"P"	"U"	"3"	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"S"
D0	"P"	"2"	"5"	"."	"2"	"N"	30	15	13	08	"C"	"P"	"U"	"4"	"N"	"A"
E0	"M"	"E"	13	09	"P"	"M"	"B"	"P"	"2"	"0"	"."	"0"	"N"	30	16	13
F0	09	"I"	"O"	"1"	"0"	"5"	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"W"	"D"
100	"6"	"4"	"."	"4"	"N"	30	16	13	09	"I"	"O"	"1"	"0"	"6"	"."	"0"
110	"N"	30	16	13	09	"I"	"O"	"1"	"0"	"7"	"N"	"A"	"M"	"E"	13	09
120	"N"	"A"	"M"	"E"	13	09	"P"	"M"	"L"	"D"	"0"	"1"	"P"	"M"	"L"	"E"
130	"0"	"1"	"."	"5"	"N"											

Remark 1. The numerical data are hexadecimal. " " indicates a character data.

2. Parentheses in the relative address show the last digit of an address.

Note * SP stands for a space character.

Figure 2 Data array on the line

5.1.3 System parameters-- Example 2 (CNC)

a) Text notation of system parameters with table format Table 7 shows an example of text notation for CNC.

Table 7 Text notation example of system parameters with table format

1) SysPara

Name of parameter	Characters used [Printable String type] (length), (characters)	Data type [Type]	Text notation (Length), (content)
Version of device profile common specifications	6, "COMVER"	INTEGER	1, 1
System parameter ID	2, "ID"	Printable String	7, "SYSPARA"
System parameter revision number	3, "REV"	INTEGER	1, 0
System parameter revision date	7, "REVDATA"	[INTEGER],2,(0001-9999) , [INTEGER],1,(01-12), [INTEGER],1,(01-31)	2, 1998 1, 9 1, 30
Device type	10, "DVCATEGORY"	Printable String	3, "CNC"
Vendor name	6, "VENDOR"	Printable String	9, "ABCDJ LTD "
Product name	7, "DVMODEL"	Printable String	16, "MSTCJ Series 16a"

2) CncPara

Name of parameter	Characters used	Data type	Content of parameter
Device-specific parameter ID	2, "ID"	Printable String	7, "DEVPARA"
Model name	5, "MODEL"	Printable String	8, "MS16a-MA"
Series	6, "SERIES"	Printable String	4, "MSF1"
Revision	3, "REV"	INTEGER	1, 0
System	7, "System"	SEQUENCE	(⁴)
System information	7, "SysInfo"	SEQUENCE	(⁴)

Note (⁴) This parameter is of structured type and has the following data.

System

Optional structure flag	5, " SFLAG "	BIT STRING	8, " 00100101 "
Number of control axis	4, " AXES "	INTEGER	2, 4

SysInfo

Virtual address input	2, " IN "	OCTET STRING	6, " 000000 "
Virtual address output	3, " OUT "	OCTET STRING	6, " 040000 "

b) Abstract syntax An abstract syntax notation example for the text notation described in a) is as follows:

1) Type definition

CncRecord ::= SEQUENCE

```

{
  SysPara          SysParaType,
  CncPara          CncParaType,
}
SysParaType ::= SEQUENCE
{
  nameCOMVER       NameType,
  paraCOMVER       INTEGER,
  nameID           NameType,
  paraID           NameType,
  nameREV          NameType,
  paraREV          INTEGER,
  nameREVDATE      NameType,
  paraREVDATE      DateType,
  nameDVCATEGORY   NameType,
  paraDVCATEGORY   NameType,
  nameVENDOR       NameType,
  paraVENDOR       NameType,
  nameDVMODEL      NameType,
  paraDVMODEL      NameType
}
CncParaType ::= SEQUENCE
{
  nameID           NameType,
  paraID           NameType,
  nameMODEL        NameType,
  paraMODEL        NameType,
  nameSERIES       NameType,
  paraSERIES       NameType,
  nameREV          NameType,
  paraREV          INTEGER,
  nameSystem       NameType,
  paraSystem       SystemType,
  nameSysInfo      NameType,
  paraSysInfo      SysInfoType
}

```



```

        paraDVCATEGORY    "CNC",
        nameVENDOR        "VENDOR",
        paraVENDOR        "ABCDJ LTD",
        nameDVMODEL      "DVMODEL",
        paraDVMODEL      "ABCDJ Series 16a"
    }
CncPara {
    nameID                "ID",
    paraID                "DEVPARA",
    nameMODEL            "MODEL",
    paraMODEL            "MS16a-MA",
    nameSERIES           "SERIES",
    paraSERIES           "MSF1",
    nameREV              "REV",
    paraREV              0,
    nameSystem           "System",
    paraSystem           {
        nameINPUT        "SFLAG",
        paraINPUT        '00100101'B,
        nameAXES         "AXES",
        paraAXES         4
    },
    nameSysInfo          "SysInfo",
    paraSysInfo          {
        nameIN           "IN",
        paraIN           '000000000000'H
        nameOUT          "OUT",
        paraOUT          '000400000000'H }
    }
}

```

5.2 Device communication I/O interface

5.2.1 Resource of device communication I/O interface

Resources of device communication I/O information means registers and data memory that are specific to respective devices and accessible from the client side. Although a program area is one of the resources, it does mean just an area for storing programs and its accessibility does not mean the one to a program as a whole. Table 8 shows the format to disclose general resource of device communication I/O interface and Table 9 shows a description example.

Table 8 Format to disclose resource of device communication I/O interface (general)

Item	Description	
Area name	Name of the resource	
Access unit	Minimal unit to read and write the resource (bits, bytes, or words)	
Area size	Size of the resource expressed in access unit	
Access attribute	Read-only or read/write from/to the network If this changes dynamically, its condition must be disclosed.	
Correspondence to virtual address (byte block)	Vendor-specific notation	Virtual address
	Correspondence to virtual address by " byte block read/write " services	-
Data array (byte block)	Contains an explanation pertaining to the data list when data in other than byte length are allocated.	
Correspondence to virtual address (word block)	Vendor-specific notation	Virtual address
	Correspondence to virtual address by the " word block read/write " services	-
Data array (word block)	Contains an explanation pertaining to the data list when data in other than word length are allocated.	
Other accessing method	Describes methods (mainly transparent mode) to access this area other than the "byte block read / write" and "word block read/write" services (if any)	

Table 9 Format to disclose resource of device communication I/O interface (description example)

Item	Description	
Area name	Register 9000 to 9777	
Access unit	Byte, word, double words	
Area size	512 bytes	
Access attribute	Read/write unconditional	
Relationship to virtual address (byte block)	Vendor-specific notation	Virtual address
	9000	16#00000800
	9001	16#00000801

	9777	16#000009FF
Data array (byte block)	1 register byte is equivalent to 1 byte of a byte block.	
Relationship to virtual address (word block)	Vendor-specific notation	Virtual address
	9000, 9001	16#00000400
	9002, 9003	16#00000401

	9776, 9777	16#000004FF
Data array (word block)	2 register bytes are equivalent to 1 word of a word block.	
	MSB	LSB
	register N+1	register N
Other accessing method	None	

5.2.2 Status / mod

Table 10 shows a general format to disclose status and mode information and Table 11 shows a description example.

Table 10 Format to disclose status and mode information (general)

Item	Description
Status type	Shows the status defined on the device.
Status accessing method	Shows a method of accessing the status from the network.
Mode type	Shows a mode defined on the device.
Mode information accessing method	Shows a method of accessing mode information from the network.
Correspondence to "stop command" and "start command" service	Shows correspondence of modes to "stop/start command" services
Correspondence to ULS in the FA link header	Shows correspondence of status/mode to U_ERR, RUN and STOP in the FA link header

Table 11 Format to disclose status and mode information (description example)

Item	Description
Status type	Normal status Abnormal status (in case operation continues) Abnormal status (in case operation stops)
Status accessing method	Word block read (address: 16#10000000)
Mode type	Program mode Setting change mode Monitor mode
Mode information accessing method	Word block read (address: 16#10000000)
Correspondence to "stop command" and "start command" service	1) At reception of "stop command" Transfer to the program mode 2) At reception of "run command" Transfer to the setting change mode when the device is out of service by the "stop command" service.
Correspondence to ULS in the FA link header	Normal status: NORMAL Abnormal status (in case operation continues): WARNING Abnormal status (in case operation stops): ALARM Program mode: STOP Setting change mode: RUN Monitor mode: RUN

5.2.3 Upload/download

Uploading and downloading is to deal with what is meaningful as a whole such as program or data area, which is accessed collectively. Further, the following additional functions should be provided:

- A function to prohibit others from accessing the area during upload or download accessing to the area
- A function to control the sequence of data transfer while the collective access is implemented as plural transfers by dividing the data area

The general format to disclose upload/download information and its description example is shown in Table 12 and Table 13 respectively.

Table 12 Format to disclose upload /download information

Item	Description
Existence of an area to be uploaded or downloaded	Describes area which can be uploaded or downloaded.
Uploading method	<ul style="list-style-type: none"> • Description pertaining to an upload-starting process • Description pertaining to an upload-ending process • Format • Process in abnormal status (timer, etc.)
Downloading method	<ul style="list-style-type: none"> • Description pertaining to a download-starting process • Description pertaining to a download-ending process • Format • Process in abnormal status (timer, etc.)

Table 13 Format to disclose upload /download information (description example)

Item	Description																																																		
Existence of an area to be uploaded or downloaded	Program area																																																		
Uploading method	<p>Define the following functions, using a transparent message TCD = XXX and the first byte of a transmission frame as a subcommand:</p> <p style="padding-left: 40px;">subcom = 16#01 (to start uploading) subcom = 16#02 (uploading) subcom = 16#03 (to end uploading) subcom = 16#99 (to stop uploading)</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 35%;"></td> <td style="width: 30%; text-align: center;">Remote device</td> <td style="width: 30%;"></td> <td style="width: 5%;"></td> <td style="width: 30%; text-align: center;">Local device</td> </tr> <tr> <td>subcom1 start</td> <td style="text-align: center;">→</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;">←</td> <td></td> <td></td> <td>return the number of block</td> </tr> <tr> <td>subcom2 block0 request</td> <td style="text-align: center;">→</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;">←</td> <td></td> <td></td> <td>read block 0 and respond</td> </tr> <tr> <td></td> <td style="text-align: center;">:</td> <td></td> <td></td> <td></td> </tr> <tr> <td>subcom2 block n request</td> <td style="text-align: center;">→</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;">←</td> <td></td> <td></td> <td>read block n and respond</td> </tr> <tr> <td>subcom3 end</td> <td style="text-align: center;">→</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;">←</td> <td></td> <td></td> <td>response</td> </tr> </table> <p>The timeout period of each command should be 500 ms. When receiving "subcom=1," this device occupies the selected area and releases the area in the following cases:</p> <ul style="list-style-type: none"> - When receiving "subcom=16#01" or "subcom=16#99" - When receiving no succeeding request for 5 seconds <p>Remark For its format, see the "implementation format of transparent mode service" provided by the vendor.</p>		Remote device			Local device	subcom1 start	→					←			return the number of block	subcom2 block0 request	→					←			read block 0 and respond		:				subcom2 block n request	→					←			read block n and respond	subcom3 end	→					←			response
	Remote device			Local device																																															
subcom1 start	→																																																		
	←			return the number of block																																															
subcom2 block0 request	→																																																		
	←			read block 0 and respond																																															
	:																																																		
subcom2 block n request	→																																																		
	←			read block n and respond																																															
subcom3 end	→																																																		
	←			response																																															

5.2.4 Implementation of message services

The general format to disclose message service implementation and its description example are shown in Table 14 and Table 15 respectively.

Table 14 Format to disclose message service implementation (general)

Item	Client	Server	Remarks
Byte block data read			
Byte block data write			
Word block data read			
Word block data write			
Network parameter read		*	
Network parameter write			
Stop command			
Start command			
Profile read		*	
Transparent mode			
Log data read		*	
Log data clear		*	
Message echo back		*	

* : implemented (essential)

Table 15 Format to disclose message service implementation (description example)

Item	Client	Server	Remarks
Byte block data read	N	Y	
Byte block data write	N	Y	
Word block data read	N	Y	
Word block data write	N	Y	
Network parameter read	N	Y*	
Network parameter write	N	Y	
Stop command	N	Y	
Stop command	N	Y	
Profile read	N	Y*	
Transparent mode	Y	Y	(details in separate document)
Log data read	N	Y*	
Log data clear	N	Y*	
Message echo back	N	Y*	

Y : implemented Y* : implemented (mandatory) N : not implemented

5.2.5 *Transparent mode message*

If implementing an original protocol on the transparent mode service, its contents shall be disclosed.

Table 16 shows a general format to disclose transparent mode services.

Table 16 Format to disclose transparent mode service (general)

Item	Description
Function	Free format description explaining the function

5.2.6 *Security-related function*

When each resource has a security function, the process (security setting, resetting etc.) shall be disclosed. A general format to disclose the security-related function and its description example are shown in Table 17 and Table 18, respectively.

Table 17 Format to disclose security-related function (general)

Item	Description
Existence of an area to apply security function	Describe areas to apply security function if any.
Security type	Security type (inaccessible, write-in prohibited, and mode unchangeable, etc.)
Security setting method	Describe its procedure, etc.
Security resetting method	Describe its procedure, etc.

Table 18 Format to disclose security-related function (description example)

Item	Description
Existence of an area to apply security function	Program area, register area
Security type	inaccessible to the program area, modes unchangeable, write-in prohibited to register area.
Security setting method	Use a transparent mode service with TCD = 200 (password setting)
Security resetting method	Use a transparent mode service with TCD = 201 (password canceling) If the password is forgotten, use a transparent mode service with TCD = 202 to cancel the password. Note that this service also clears the program area.

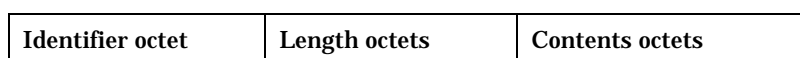
Appendix 1 (Informative) Summary of ASN.1 Transfer Syntax Format

This article gives supplemental explanations for the standards referred in the body, and does not constitute part of the standard.

The part of JIS X 5604 related to this technical report is described.

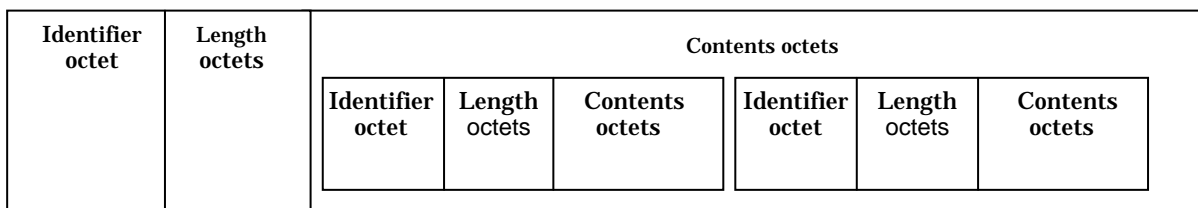
1. Encoding structure of transfer data

a) **Basic ASN.1 type** Appendix 1-Figure 1 shows the basic ASN.1 type.



Appendix 1-Figure 1 Basic ASN.1 type

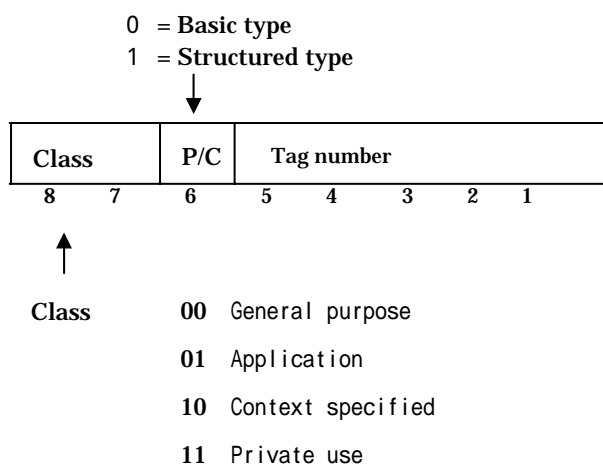
b) **Encoding example of structured ASN.1 type** Appendix 1-Figure 2 shows an encoding example of structured ASN.1.



Appendix 1- Figure 2 Encoding example of structured ASN.1 type

c) Identifier octet

1) **Structure (one octet type)** Appendix1-Figure 3 shows a structure of identifier octets



Appendix 1- Figure 3 Structure of identifier octet (one octet type)

- 2) Tag number (general-purpose tag class) Appendix 1-Table 1 shows a tag number of the identifier octets (general-purpose tag class).

Appendix 1-Table 1 Tag number of identifier octet (general-purpose tag class)

Tag number (hexadecimal)	Type	Tag number (hexadecimal)	Type
00	(Reserved)	11	SET and SET OF
01	BOOLEAN	12	NumericString
02	INTEGER	13	PrintableString
03	BIT STRING	14	TeletexString
04	OCTET STRING	15	VideotexString
05	NULL	16	IA5String
06	OBJECT IDENTIFIER	17	UTCTime
07	ObjectDescriptor	18	GeneralizedTime
08	EXTERNAL	19	GraphicString
09	REAL	1A	VisibleString
0A	ENUMERATED	1B	GeneralString
0B~0F	(Reserved)	1C	CharacterString
10	SEQUENCE and SEQUENCE OF	1D ~ 1E	(Reserved)

- 3) Data type and structured flag Data types of identifier octet corresponding to the basic type and structured type are shown in Appendix 1-Table 2.

Appendix 1-Table 2 Correspondence of data type to basic type and structured type

ASN.1 type	Basic	Structured
BOOLEAN, INTEGER, OBJECT IDENTIFIER, REAL, ENUMERATED	Y	-
BIT STRING	Y	Y
String type such as OCTET STRING and NumericString	Y	Y
NULL (without value field)	Y	-
SEQUENCE, SEQUENCE OF, SET, SET OF	-	Y
EXTERNAL	-	Y
CHOICE	Y	Y
ANY	Y	Y
Tagged type	Y	Y

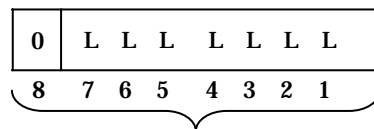
Y : available

4) **Printable strings** Characters and codes available in printable string are shown in **Appendix 1-Table 3**.

Appendix 1-Table 3 Characters and code available in printable string

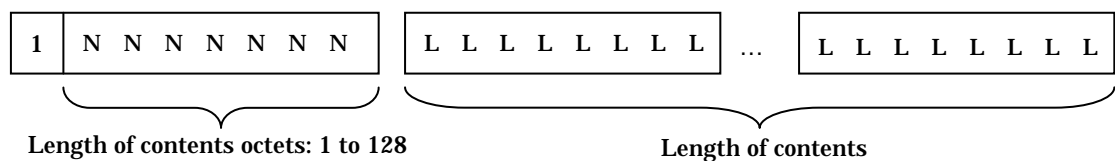
Name	Characters	Code (hexadecimal)
Capital letters	A, B, ..., Z	41, 42, ..., 5A
Small letters	a, b, ..., z	61, 62, ..., 7A
Digits	0, 1, ..., 9	30, 31, ..., 39
Space	(space)	20
Apostrophe	'	27
Left parenthesis	(28
Right parenthesis)	29
Plus sign	+	2B
Comma	,	2C
Hyphen	-	2D
Full stop	.	2E
Solidus	/	2F
Colon	:	3A
Equal sign	=	3D
Question mark	?	3F

d) **Length octets** Appendix 1-Figure 4 and Appendix 1-Figure 5 show the fixed-length short format and long format of the length octets respectively.



Length of contents octet: 1 to 127

Appendix 1-Figure 4 Fixed-length short format of length octet



Appendix 1-Figure 4 Fixed-length long format of length octet

2. Data transmission sequence

The data transmission sequence takes the “Big endian” type that first transmits the highest octet of data.

Appendix 2 (Informative) Implementation of Items Read by the "Log Data Read" Service

This article gives additional explanations for the items related to the body and does not constitute the part of the standard.

The contents described in this article are out of scope of device communication information dealt with in this technical report but desirable to be disclosed by vendors.

Vendors shall declare whether the items read by the "Log Data Read " service and shown in **Appendix 2-Table 1** are implemented or not.

Appendix 2-Table 1 Items read by "Log Data Read" service

Major items	Minor items	Implemented (Y) or not(N)
Transmission/ reception	Total number of transmission at socket unit	*
	Total number of transmission errors at socket unit	*
	Number of Ethernet transmission errors	
	Total number of receptions	*
	Total number of reception errors	*
	Number of Ethernet reception errors	
Frame types	Number of tokens transmitted	
	Number of cyclic frames transmitted	
	Number of peer-to-peer messages transmissions	
	Number of broadcast messages transmissions	
	Number of tokens received	
	Number of cyclic frames received	
	Number of peer-to-peer messages received	
	Number of broadcast messages received	
Cyclic transmission	Number of cyclic reception errors	*
	Number of cyclic address size errors	
	Number of cyclic CBN errors	
	Number of cyclic TBN errors	
	Number of cyclic BSIZE errors	
Message transmission	Number of message retransmissions	*
	Number of message over-retransmissions	*
	Number of message reception errors	*
	Number of message version-of-sequence number errors	
	Number of message sequence number retransmissions recognized	
ACK-related	Number of ACK errors	*
	Number of ACK version-of-sequence number errors	
	Number of ACK sequence number errors	
	Number of ACK node number errors	
	Number of ACK TCD errors	

Appendix 2-Table 1 Items read by "Log Data Read" service (continued)

Major items	Minor items	Implemented (Y) or not (N)
Token-related	Number of token multiplications recognized	*
	Number of tokens discarded	*
	Number of tokens re-issued	*
	Number of token holding timeouts	
	Number of token monitoring timeouts	
Status 1	Total service time	
	Number of frame waiting states	*
	Number of participations	*
	Number of self-exits	*
	Number of exits by skipping	*
	Number of exits of other nodes recognized	*
Status 2	List of participation recognized nodes	
Vendor definable area		

* : implemented (mandatory)

JEM-TR 214 : 2000
Device profile common specification for
FA control network [FL-net (OPCN-2)]

Explanation

This article comments on the contents specified and described in the body and the appendices. This article does not constitute a part of the technical report.

1. History of technical report establishment This technical report is re-edition of “FL-net device profile common specification (JOP-1002 version 1, established 22/Jul./1999)” settled by the FA Control Committee of FA Open Promotion Council of the Manufacturing Science Technical Center (MSTC) to adjust it to the standard document format of the Japan Electrical Manufacturers’ Association (JEMA). This was brought about because standardization tasks of FA control network (FL-net) were transferred from MSTC to JEMA.

2. Item particularly deliberated In the “FL-net device profile common specification”, “PLC” was used for a device type parameter to express the programmable controller. But as “PLC” is a trademark, “PC” is adopted to meet the IEC standard.