IEC 61158-6-18

Edition 2.0 2010-08

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 6-18: Application layer protocol specification – Type 18 elements

IEC 61158-6-18:2010(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 61158-6-18

Edition 2.0 2010-08

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 6-18: Application layer protocol specification – Type 18 elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XA

ICS 25.04.40; 35.100.70; 35.110

ISBN 978-2-88912-132-8

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-18: Application layer protocol specification – Type 18 elements

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

International Standard IEC 61158-6-18 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision.

The main changes with respect to the previous edition are listed below:

- editorial corrections;
- addition of cyclic data segmenting.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/607/FDIS	65C/621/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61158 series, published under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC/TR 61158-1.

The application protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementers and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the profile parts. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning Type 18 elements and possibly other types given in subclause 4.5, 5.5, 6.9.2 and 9.2.2 as follows:

3343036/Japan	[MEC]	“Network System for a Programmable Controller”
5896509/USA	[MEC]	“Network System for a Programmable Controller”
246906/Korea	[MEC]	“Network System for a Programmable Controller”
19650753/Germany	[MEC]	“Network System for a Programmable Controller”

IEC takes no position concerning the evidence, validity and scope of these patent rights.

The holder of these patent rights has assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of these patent rights is registered with IEC. Information may be obtained from:

[MEC] Mitsubishi Electric Corporation
Corporate Licensing Division
7-3, Marunouchi 2-chome, Chiyoda-ku,
Tokyo 100-8310, Japan

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

ISO (www.iso.org/patents) and IEC (http://www.iec.ch/tctools/patent_decl.htm) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-18: Application layer protocol specification – Type 18 elements

1 Scope

1.1 General

The Fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs”.

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 18 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard specifies interactions between remote applications and defines the externally visible behavior provided by the Type 18 fieldbus application layer in terms of

a) the formal abstract syntax defining the application layer protocol data units conveyed between communicating application entities;
b) the transfer syntax defining encoding rules that are applied to the application layer protocol data units;
c) the application context state machine defining the application service behavior visible between communicating application entities;
d) the application relationship state machines defining the communication behavior visible between communicating application entities.

The purpose of this standard is to define the protocol provided to

a) define the wire-representation of the service primitives defined in IEC 61158-5-18, and
b) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 18 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI application layer structure (ISO/IEC 9545).

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-18.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of protocols standardized in the IEC 61158-6 series.

1.3 Conformance

This standard does not specify individual implementations or products, nor do they constrain the implementations of application layer entities within industrial automation systems. Conformance is achieved through implementation of this application layer protocol specification.

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.

IEC 61158-5-18:2010[1], *Industrial communication networks – Fieldbus specifications – Part 5-18: Application layer service definition – Type 18 elements*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

3 Terms and definitions

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 ISO/IEC 7498-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 7498-1 apply:

a) application entity
b) application process
c) application protocol data unit
d) application service element
e) application entity invocation
f) application process invocation
g) application transaction
h) real open system
i) transfer syntax

3.1.2 ISO/IEC 8822 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8822 apply:

1 o be published.

a) abstract syntax
b) presentation context

3.1.3 ISO/IEC 9545 terms

For the purposes of this document, the following terms as defined in ISO/IEC 9545 apply:

a) application-association
b) application-context
c) application context name
d) application-entity-invocation
e) application-entity-type
f) application-process-invocation
g) application-process-type
h) application-service-element
i) application control service element

3.1.4 ISO/IEC 8824-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8824-1 apply:

a) object identifier
b) type

3.2 Other terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.2.1
alarm
activation of an event that shows a critical state

3.2.2
alarm ack
acknowledgment of an event that shows a critical state

3.2.3
application
function or data structure for which data is consumed or produced

3.2.4
application layer interoperability
capability of application entities to perform coordinated and cooperative operations using the services of the FAL

3.2.5
application objects
multiple object classes that manage and provide a run time exchange of PDUs across the network and within the network device

3.2.6
application process
part of a distributed application on a network, which is located on one device and unambiguously addressed

3.2.7
application process identifier
distinguishes multiple application processes used in a device

NOTE Application process identifier is assigned by PROFIBUS International (PI).

3.2.8
application process object
component of an application process that is identifiable and accessible through an FAL application relationship

NOTE Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application objects and their corresponding definitions.

3.2.9
application process object class
a class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.10
application relationship
cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.2.11
application relationship application service element
application-service-element that provides the exclusive means for establishing and terminating all application relationships

3.2.12
application relationship endpoint
context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

NOTE Each application process involved in the application relationship maintains its own application relationship endpoint.

3.2.13
attribute
description of an externally visible characteristic or feature of an object

NOTE The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behavior of an object. Attributes are divided into class attributes and instance attributes.

3.2.14
backup
status of the IO AR, which indicates that it, is in the standby state

3.2.15
behavior
indication of how an object responds to particular events

3.2.16
class
a set of objects, all of which represent the same kind of system component

NOTE A class is a generalization of an object; a template for defining variables and methods. All objects in a class are identical in form and behavior, but usually contain different data in their attributes.

3.2.17
class attributes
attribute that is shared by all objects within the same class

3.2.18
class code
unique identifier assigned to each object class

3.2.19
class specific service
service defined by a particular object class to perform a required function which is not performed by a common service

NOTE A class specific object is unique to the object class which defines it.

3.2.20
clear
status of the IO controller, which indicates that the control algorithm is currently not running

3.2.21
cyclic
repetitive in a regular manner

3.2.22
device
physical hardware connected to the link

NOTE A device may contain more than one node.

3.2.23
device ID
a vendor assigned device type identification

3.2.24
device profile
a collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.2.25
diagnosis data object
object(s) which contains diagnosis information referenced by device/slot/subslot/index

3.2.26
diagnosis information
all data available at the server for maintenance purposes

3.2.27
endpoint
one of the communicating entities involved in a connection

3.2.28
engineering
abstract term that characterizes the client application or device responsible for configuring an automation system via interconnecting data items

3.2.29
error
discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.2.30
error class
general grouping for related error definitions and corresponding error codes

3.2.31
error code
identification of a specific type of error within an error class

3.2.32
event
an instance of a change of conditions

3.2.33
frame
denigrated term for DLPDU, unit of data transfer on a link

3.2.34
identification data object
object(s) that contain information about device, module and sub-module manufacturer and type referenced by device/slot/subslot/index

3.2.35
implicit AR endpoint
AR endpoint that is defined locally within a device without use of the create service

3.2.36
index
address of a record data object within an application process

3.2.37
instance
the actual physical occurrence of an object within a class that identifies one of many objects within the same object class

3.2.38
instance attributes
attribute that is unique to an object instance and not shared by the object class

3.2.39
instantiated
object that has been created in a device

3.2.40
interface
collection of FAL class attributes and services that represents a specific view on the FAL class

3.2.41
invocation
act of using a service or other resource of an application process

NOTE Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

3.2.42
IO controller
controlling device, which acts as client for several IO devices (field devices)

NOTE This is usually a programmable controller or a distributed control system.

3.2.43
IO data object
object designated to be transferred cyclically for the purpose of processing and referenced by device/slot/subslot

3.2.44
IO device
field device which acts as server for IO operation

3.2.45
IO supervisor
engineering device which manages commissioning and diagnosis of an IO system

3.2.46
IO system
system composed of all its IO subsystems

NOTE As an example a PLC with more than one IO controller (network interface) controls one IO system composed of an IO subsystems for each IO controller.

3.2.47
logical device
a certain FAL class that abstracts a software component or a firmware component as an autonomous self-contained facility of an automation device

3.2.48
member
piece of an attribute that is structured as an element of an array

3.2.49
method
<object> a synonym for an operational service which is provided by the server ASE and invoked by a client

3.2.50
module
hardware or logical component of a physical device

3.2.51
network
a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.2.52
object
abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior

3.2.53
object specific service
service unique to the object class which defines it

3.2.54
operate
status of the IO controller that indicates that the control algorithm is currently running

3.2.55
peer
role of an AR endpoint in which it is capable of acting as both client and server

3.2.56
physical device
a certain FAL class that abstracts the hardware facilities of an automation device

3.2.57
point-to-point connection
connection that exists between exactly two application objects

3.2.58
primary
status of the IO AR that indicates that it is in the operating state

NOTE Besides a primary IO AR a backup IO AR may exist. In example used for redundancy and dynamic reconfiguration of IO data.

3.2.59
property
a synonym for ASE attributes which are readable or writeable via operational ASE services

NOTE These services are generally named “get_<Attribute Name>” or “set_<Attribute Name>” and correspond with the IDL keywords “propget” and “propput”.

3.2.60
provider
a) source of a data connection
b) node or source sending data to one or many consumer

3.2.61
Register X
register containing bit-oriented cyclic data of type input data that is transmitted from a slave to a master

3.2.62
Register Y
register containing bit-oriented cyclic data of type output data that is transmitted from a master to a slave

3.2.63
Register Wr
register containing word-oriented cyclic data of type input data that is transmitted from a slave to a master

3.2.64
Register Ww
register containing word-oriented cyclic data of type output data that is transmitted from a master to a slave

3.2.65
resource
processing or information capability

3.2.66
run
status of the IO controller which indicates that the control algorithm is currently operating

3.2.67
runtime object model
objects that exist in a device together with their interfaces and methods that are accessible

3.2.68
service
operation or function than an object and/or object class performs upon request from another object and/or object class

3.2.69
slot
address of a structural unit within an IO device

NOTE Within a modular device, a slot typically addresses a physical module. Within compact devices, a slot typically addresses a logical function or virtual module.

3.2.70
stop
status of the IO controller which indicates that the control algorithm is currently not running

3.2.71
vendor ID
central administrative number used as manufacturer identification

NOTE The vendor ID is assigned by PROFIBUS International (PI).

3.3 Abbreviations and symbols

AE	Application entity
AL	Application layer
AP	Application process
APDU	Application protocol data unit
AR	Application relationship
AREP	Application relationship endpoint
ASE	Application service element
DL-	(as a prefix) data-link-
DLL	Data-link layer
DLPDU	Data-link protocol data unit
DLSDU	DL-service data unit
FAL	Fieldbus application layer
ID	Identifier
PDU	Protocol data unit
PL	Physical layer
SDU	Service Data Unit

3.4 Additional abbreviations and symbols for type 18

RX	Register X

RY	Register Y
RWr	Register Wr
RWw	Register Ww

3.5 Conventions

3.5.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-18. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5-18 standard. The service specification defines the services that are provided by the ASE.

This standard uses the descriptive conventions given in ISO/IEC 10731.

4 Abstract syntax

4.1 M1 device manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 1.

Table 1 – M1 device manager attribute format

Attribute	Format	Size (bits)
Management information	Structure of 6 elements:	9 octets
Transmission speed	Unsigned8	8
Number of occupied stations	Unsigned8	8
Station number	Unsigned8	8
Vendor code	Unsigned16	16
Model code	3 octets, bit mapped	24
Software/protocol version	1 octet, bit mapped	8
Connected Slave management information	Array of 64 members:	640 octets
Slave information 1 - 64	Structure of 5 elements:	10 octets
Station number	Unsigned8	8
Vendor code	Unsigned16	16
Model code	3 octets, bit mapped	24
Software/protocol version	1 octet, bit mapped	8
Reserved field	3 octets	24

4.2 M2 device manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 2.

Table 2 – M2 device manager attribute format

Attribute	Format	Size (bits)
Slave station information	Array of 64 members:	128 octets

Attribute	Format	Size (bits)
Slave station information 1 - 64	Word	16
Slave station status information	Array of 64 members:	32 octets
Slave station status information 1 - 64	4 Bits	4

4.3 S1 device manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 3.

Table 3 – S1 device manager attribute format

Attribute	Format	Size (bits)
Management information	Structure of 5 elements:	10 octets
Station number	Unsigned8	8
Vendor code	Unsigned16	16
Model code	3 octets, bit mapped	24
Software/protocol version	1 octet, bit mapped	8
Reserved field	3 octets	24

4.4 S2 device manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 4.

Table 4 – S2 device manager attribute format

Attribute	Format	Size (bits)
Slave station information	Word	16
Slave station status information	4 Bits	4

4.5 M1 connection manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 5.

Table 5 – M1 connection manager attribute format

Attribute	Format	Size (bits)
Parameter information	Structure of 14 elements:	452 octets
Number of connected modules	Unsigned16	16
Number of intelligent devices	Unsigned16	16
Station information	Array of 64 members:	128 octets
Station information 1 - 64	Word	16
Number of automatic return modules	Unsigned16	16
Number of retries	Unsigned16	16
Delay time setting	Unsigned16	16
Standby Master station specification	Unsigned16	16
Operation during Master error state	Unsigned16	16
Data-link during Master error state	Unsigned16	16
Scan mode specification	Unsigned16	16

Attribute	Format	Size (bits)
Reserved station specification	Array of 8 members:	8 octets
Reserved station number	Unsigned8	8
Error invalid station specification	Array of 8 members:	8 octets
Invalid station number	Unsigned8	8
Reserved 1	4 Words	8 octets
Reserved 2	78 Words	156 octets
Network status information	Structure of 4 elements:	260 octets
Master status information	Word	16
Slave status information	Array of 64 members:	128 octets
Slave status information 1 - 64	Word	16
Master transmitted status field	Word	16
Slave transmitted status field	Array of 64 members:	128 octets
Slave transmitted status field 1 - 64	Word	16
Network information	Structure of 3 elements:	6 octets
Current link scan time	Unsigned16	16
Minimum link scan time	Unsigned16	16
Maximum link scan time	Unsigned16	16

4.6 M2 connection manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 6.

Table 6 – M2 connection manager attribute format

Attribute	Format	Size (bits)
Parameter information	Structure of 4 elements:	15
Transmission speed	2 Bits	2
Last station number	Unsigned8	8
Point mode setting	2 Bits	2
Master station i/o point mode setting	3 Bits	3
Network status information	Structure of 2 elements:	48 octets
Reception status information	Structure of 64 elements:	32 octets
Reception status 1 - 64	4 Bits	4
Slave status information	Structure of 64 elements:	16 octets
Slave status 1 - 64	2 Bits	2

4.7 S1 connection manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 7.

Table 7 – S1 connection manager attribute format

Attribute	Format	Size (bits)
Process data support level	2 Bits	2
Network status information	Structure of 1 element:	16
Master transmitted status field	Word	16

4.8 S2 connection manager PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 8.

Table 8 – S2 connection manager attribute format

Attribute	Format	Size (bits)
Parameter information	Structure of 2 elements:	11
Slave station number	Unsigned8	8
Number of occupied slots	3 Bits	3

4.9 M1 cyclic transmission PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 9.

Table 9 – M1 cyclic transmission attribute format

Attribute	Format	Size (bits)
Master status	Word	16
Data out	Structure of 2 elements:	x + y [a]
RY data	Bit-oriented data structure	x [a]
RWw data	Word-oriented data structure	y [a]
Data in	Structure of 2 elements	z + 16 [a]
Number of Modules	Unsigned16	16
Slave input data	Array of up to 64 members:	z [a]
Station number	Unsigned16	16
Slave status	Word	16
RX data	Bit-oriented data structure	x [a]
RWr data	Word-oriented data structure	y [a]

[a] The values of x, y and z are dependent upon the values of the corresponding configuration settings in the Master status.

4.10 M2 cyclic transmission PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 10.

Table 10 – M2 cyclic transmission attribute format

Attribute	Format	Size (bits)
RY data	Bit-oriented data structure	64 n [a]
RX data	Bit-oriented data structure	64 n [a]

[a] n = the number of points specified by the point-mode-setting value of the M2 connection manager.

4.11 S1 cyclic transmission PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 11.

Table 11 – S1 cyclic transmission attribute format

Attribute	Format	Size (bits)
Slave status	Word	16
Data out	Structure of 2 elements:	x + y [a]
RY data	Bit-oriented data structure	x [a]
RWw data	Word-oriented data structure	y [a]
Master status	Word	16
Data in	Structure of 2 elements:	x + y [a]
RX data	Bit-oriented data structure	x [a]
RWr data	Word-oriented data structure	y [a]

[a] The values of x and y are dependent upon the values of the corresponding configuration settings in the Master status.

4.12 S2 cyclic transmission PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 12.

Table 12 – S2 cyclic transmission attribute format

Attribute	Format	Size (bits)
RY data	Bit-oriented data structure	n [a]
RX data	Bit-oriented data structure	n [a]

[a] n = the number of points specified by the point-mode-setting value of the corresponding M2 connection manager times the number of occupied Slots as specified by the S2 connection manager.

4.13 Acyclic transmission PDU abstract syntax

The abstract syntax for attributes belonging to this class is described in Table 13.

Table 13 – Acyclic transmission attribute format

Attribute	Format	Size (bits)
Length	Unsigned16	16
Reserved	Octet	8
Type and sequence	Bit-oriented data structure	8
Segment number	Octet	8
Data type	Bit-oriented data structure	8

Attribute	Format	Size (bits)
Destination address	Unsigned8	8
Source address	Unsigned8	8
Destination app type	Unsigned8	8
Source app type	Unsigned8	8
Destination app module	Unsigned8	8
Source app module	Unsigned8	8
Destination network id	Unsigned8	8
Destination address	Unsigned8	8
Destination id	Bit-oriented data structure	16
Source network id	Unsigned8	8
Source address	Unsigned8	8
Source id	Bit-oriented data structure	16
Command header	Octet-oriented data structure	64
Command parameters	Variable length parameter field	0-960 octets

5 Transfer syntax

5.1 M1 device manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 14.

Table 14 – M1 device manager attribute encoding

<table>
<tr><th>Attribute</th><th colspan="4">Encoding</th></tr>
<tr><td>Management information</td><td colspan="4">Specifies the configuration of the Master device</td></tr>
<tr><td>Transmission speed</td><td colspan="4">Enumerated list of baud rate values in kbit/s
0 = 156
1 = 625
2 = 2 500
3 = 5 000
4 = 10 000
5 - 255 = reserved</td></tr>
<tr><td>Number of occupied stations</td><td colspan="4">1 - 4 = allowable range</td></tr>
<tr><td>Station number</td><td colspan="4">0 = Master
1 - 127 = reserved
128 = Standby Master
129 - 255 = reserved</td></tr>
<tr><td>Vendor code</td><td colspan="4">The assignment and management of vendor codes is best handled by applicable trade organizations and is therefore beyond the scope of this specification</td></tr>
<tr><td rowspan="11">Model code</td><td>Octet</td><td>Bit</td><td>Description</td><td>Value</td></tr>
<tr><td rowspan="4">0</td><td>1 – 0</td><td>Total number of used bit-oriented data bits (both RX and RY combined)</td><td>0 = full use
1 = 8 used
2 = 32 used
3 = 16 used</td></tr>
<tr><td>3 – 2</td><td>Distribution of used bit-oriented data bits</td><td>0 = RX and RY in equal sizes
1 = RX only
2 = RY only
3 = other RX / RY mix</td></tr>
<tr><td>5 – 4</td><td>Number of occupied station slots</td><td>0 = 1 slot
1 = 2 slots
2 = 3 slots
3 = 4 slots</td></tr>
<tr><td>7 – 6</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td rowspan="4">1</td><td>0</td><td>FAL-user switch setting</td><td>0 = normal
1 = abnormal</td></tr>
<tr><td>1</td><td>FAL-user output upon fault</td><td>0 = clear
1 = hold</td></tr>
<tr><td>5 – 2</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7 – 6</td><td>DLE support level</td><td>0 = level A
1 = level B
2 = level C
3 = reserved</td></tr>
<tr><td rowspan="2">2</td><td>6 – 0</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7</td><td>Acyclic data format supports messaging type commands.</td><td>0 = not supported
1 = supported</td></tr>
<tr><td rowspan="3">Software/protocol version</td><td>Bit</td><td colspan="2">Description</td><td>Value</td></tr>
<tr><td>5 - 0</td><td colspan="2">Software version</td><td>1 - 63 = allowable range</td></tr>
<tr><td>7 - 6</td><td colspan="2">Protocol version</td><td>0 = Version 1
1 = Version 2
2 = Version 3
3 = Version 4</td></tr>
</table>

<table>
<tr><th>Attribute</th><th colspan="4">Encoding</th></tr>
<tr><td>Connected Slave management information</td><td colspan="4">Specifies the configuration of the connected Slaves</td></tr>
<tr><td>Slave information 1 - 64</td><td colspan="4">Array of 64 elements, each encoded as:</td></tr>
<tr><td>Station number</td><td colspan="4">1 - 64 = allowable range</td></tr>
<tr><td>Vendor code</td><td colspan="4">The assignment and management of vendor codes is best handled by applicable trade organizations and is therefore beyond the scope of this specification</td></tr>
<tr><td rowspan="12">Model code</td><td>Octet</td><td>Bit</td><td>Description</td><td>Value</td></tr>
<tr><td rowspan="4">0</td><td>1 – 0</td><td>Total number of used bit-oriented data bits (both RX and RY combined)</td><td>0 = full use
1 = 8 used
2 = 32 used
3 = 16 used</td></tr>
<tr><td>3 – 2</td><td>Distribution of used bit-oriented data bits</td><td>0 = RX and RY in equal sizes
1 = RX only
2 = RY only
3 = other RX / RY mix</td></tr>
<tr><td>5 – 4</td><td>Number of occupied station slots</td><td>0 = 1 slot
1 = 2 slots
2 = 3 slots
3 = 4 slots</td></tr>
<tr><td>7 – 6</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td rowspan="4">1</td><td>0</td><td>FAL-user switch setting</td><td>0 = normal
1 = abnormal</td></tr>
<tr><td>1</td><td>FAL-user output upon fault</td><td>0 = clear
1 = hold</td></tr>
<tr><td>5 – 2</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7 – 6</td><td>DLE support level</td><td>0 = level A
1 = level B
2 = level C
3 = reserved</td></tr>
<tr><td rowspan="2">2</td><td>6 – 0</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7</td><td>Acyclic data format supports messaging type commands.</td><td>0 = not supported
1 = supported</td></tr>
<tr><td rowspan="3">Software/protocol version</td><td>Bit</td><td colspan="2">Description</td><td>Value</td></tr>
<tr><td>5 - 0</td><td colspan="2">Software version</td><td>1 - 63 = allowable range</td></tr>
<tr><td>7 - 6</td><td colspan="2">Protocol version</td><td>0 = Version 1
1 = Version 2
2 = Version 3
3 = Version 4</td></tr>
<tr><td>Reserved field</td><td colspan="4">Reserved for future specification definition</td></tr>
</table>

5.2 M2 device manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 15.

Table 15 – M2 device manager attribute encoding

Attribute	Encoding	
Slave station information	Specifies the configuration of the connected Slaves	
Slave station information 1 - 64	Array of 64 Words, each encoded as:	
	Bit	**Definition**
	2 – 0	0 = 1 i/o point 1 = 2 i/o points 2 = 4 i/o points 3 = 8 i/o points 4 = 16 i/o points 5 = 12 i/o points 6 = reserved 7 = reserved
	3	output i/o type present (0 = false; 1 = true)
	4	input i/o type present (0 = false; 1 = true)
	5	device type: (0 = remote i/o station; 1 = remote device station)
	6	configured as a head station (for number of occupied DLE station slots > 1) (0 = false; 1 = true)
	7	input time constant (0 = normal; 1 = high speed)
	8	output state for abnormal operating states (0 = clear; 1 = hold)
	15 – 9	reserved
Slave station status information	The status fields transmitted from each connected Slave	
Slave station status information 1 - 64	Array of 64, 4-Bit fields, each encoded as:	
	Bit	**Definition**
	0	Slave status (0 = Normal; 1 = Error)
	1	Configuration data transmitted since connected (0 = false; 1 = true)
	2	Parity (provides even parity for status field)
	3	Reserved (set = 0)

5.3 S1 device manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 16.

Table 16 – S1 device manager attribute encoding

<table>
<tr><th>Attribute</th><th colspan="4">Encoding</th></tr>
<tr><td>Management information</td><td colspan="4">Specifies the configuration of the Slave device</td></tr>
<tr><td>Station number</td><td colspan="4">1 - 64 = allowable range</td></tr>
<tr><td>Vendor code</td><td colspan="4">The assignment and management of vendor codes is FAL user specific and beyond the scope of this specification</td></tr>
<tr><td rowspan="11">Model code</td><td>Octet</td><td>Bit</td><td>Description</td><td>Value</td></tr>
<tr><td rowspan="4">0</td><td>1 – 0</td><td>Total number of used bit-oriented data bits (both RX and RY combined)</td><td>0 = full use
1 = 8 used
2 = 32 used
3 = 16 used</td></tr>
<tr><td>3 – 2</td><td>Distribution of used bit-oriented data bits</td><td>0 = RX and RY in equal sizes
1 = RX only
2 = RY only
3 = other RX / RY mix</td></tr>
<tr><td>5 – 4</td><td>Number of occupied station slots</td><td>0 = 1 slot
1 = 2 slots
2 = 3 slots
3 = 4 slots</td></tr>
<tr><td>7 – 6</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td rowspan="4">1</td><td>0</td><td>FAL-user switch setting</td><td>0 = normal
1 = abnormal</td></tr>
<tr><td>1</td><td>FAL-user output upon fault</td><td>0 = clear
1 = hold</td></tr>
<tr><td>5 – 2</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7 – 6</td><td>DLE support level</td><td>0 = level A
1 = level B
2 = level C
3 = reserved</td></tr>
<tr><td rowspan="2">2</td><td>6 – 0</td><td>reserved</td><td>(set = 0)</td></tr>
<tr><td>7</td><td>Acyclic data format supports messaging type commands.</td><td>0 = not supported
1 = supported</td></tr>
<tr><td rowspan="3">Software/protocol version</td><td>Bit</td><td colspan="2">Description</td><td>Value</td></tr>
<tr><td>5 - 0</td><td colspan="2">Software version</td><td>1 - 63 = allowable range</td></tr>
<tr><td>7 - 6</td><td colspan="2">Protocol version</td><td>0 = Version 1
1 = Version 2
2 = Version 3
3 = Version 4</td></tr>
<tr><td>Reserved field</td><td colspan="4">Reserved for future specification definition</td></tr>
</table>

5.4 S2 device manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 17.

Table 17 – S2 device manager attribute encoding

Attribute	Encoding	
Slave station information	**Bit**	**Definition**
	2 – 0	0 = 1 i/o point 1 = 2 i/o points 2 = 4 i/o points 3 = 8 i/o points 4 = 16 i/o points 5 = 12 i/o points 6 = reserved 7 = reserved
	3	output i/o type present (0 = false; 1 = true)
	4	input i/o type present (0 = false; 1 = true)
	5	device type: (0 = remote i/o station; 1 = remote device station)
	6	configured as a head station (for number of occupied DLE station slots > 1) (0 = false; 1 = true)
	7	input time constant (0 = normal; 1 = high speed)
	8	output state for abnormal operating states (0 = clear; 1 = hold)
	15 – 9	reserved
Slave station status information	**Bit**	**Definition**
	0	Slave status (0 = Normal; 1 = Error)
	1	Configuration data transmitted since connected (0 = false; 1 = true)
	2	Parity (provides even parity for status field)
	3	Reserved (set = 0)

5.5 M1 connection manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 18.

Table 18 – M1 connection manager attribute encoding

<table>
<tr><th>Attribute</th><th colspan="3">Encoding</th></tr>
<tr><td>Parameter information</td><td colspan="3">Specifies the connection configuration</td></tr>
<tr><td>Number of connected modules</td><td colspan="3">1 - 64 = allowable range</td></tr>
<tr><td>Number of intelligent devices</td><td colspan="3">Specifies the number of Slave devices that support process data support level C

0 - 26 = allowable range</td></tr>
<tr><td>Station information</td><td colspan="3">Indicates the connection configurations of the connected Slave devices.</td></tr>
<tr><td rowspan="4">Station information 1 - 64</td><td>Bit</td><td>Description</td><td>Value</td></tr>
<tr><td>7 – 0</td><td>Station number</td><td>1 - 64 = allowable range</td></tr>
<tr><td>11 – 8</td><td>Number of occupied station slots</td><td>1 - 4 = allowable range</td></tr>
<tr><td>15 – 12</td><td>Station type in terms of process data support level</td><td>0 = Level A
1 = Level B
2 = Level C
3 = reserved
4 = reserved
5 = Level B 1x cyclic seg
6 = Level C 1x cyclic seg
7 = reserved
8 = Level B 2x cyclic seg
9 = Level C 2x cyclic seg
10 = reserved
11 = Level B 4x cyclic
12 = Level C 4x cyclic
13 = reserved
14 = Level B 8x cyclic
15 = Level C 8x cyclic</td></tr>
<tr><td>Number of automatic return modules</td><td colspan="3">Specifies the number Slave devices that are able to resume connections.

1 - 10 = allowable range</td></tr>
<tr><td>Number of retries</td><td colspan="3">Specifies the number of times the M1 connection manager object will retry communications scans with consecutive DL errors.

1 - 7 = allowable range</td></tr>
<tr><td>Delay time setting</td><td colspan="3">Specifies the data-link Scan interval in µs.

0 - 5000 = allowable range</td></tr>
<tr><td>Standby Master station specification</td><td colspan="3">0 - 64 = allowable range

0 = no standby Master</td></tr>
<tr><td>Operation during Master error state</td><td colspan="3">Specifies M1 connection manager object control over the DL during an FAL error state.

0 = Stop Scan
1 = Continue Scan
2 - 65535 = reserved</td></tr>
<tr><td>Data-link during Master error state</td><td colspan="3">Specifies the state for process data during an FAL error state.

0 = Hold data states
1 = Clear data states
2 - 65535 = reserved</td></tr>
<tr><td>Scan mode specification</td><td colspan="3">Specifies the scanning behavior.

0 = free-running (continuous loop)
1 = triggered (loop once)
2 - 65535 = reserved</td></tr>
<tr><td>Reserved station specification</td><td colspan="3">Specifies the reserved Slave station numbers. Although reserved stations are counted as connected stations, a data-link error will not occur for reserved stations not connected.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th colspan="3">Encoding</th></tr>
<tr><td>Reserved station number</td><td colspan="3">1 - 64 = allowable range</td></tr>
<tr><td>Error invalid station specification</td><td colspan="3">Specifies the error invalid Slave station numbers. Slave station errors (at the Master) will not occur for error invalid Slave stations.</td></tr>
<tr><td>Invalid station number</td><td colspan="3">1 - 64 = allowable range</td></tr>
<tr><td>Network status information</td><td colspan="3">Master reception status</td></tr>
<tr><td rowspan="9">Master status information</td><td>Bit</td><td>Description</td><td>Values</td></tr>
<tr><td>0</td><td>Master frame interval timeout occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>1</td><td>Refresh frame reception timeout occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>2</td><td>Consecutive transmission timeout occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>3</td><td>Consecutive reception timeout occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>4</td><td>Transmission route monitor 1 error occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>5</td><td>Transmission route monitor 2 error occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>6</td><td>Standby Master activation requested</td><td>0 = False
1 = True</td></tr>
<tr><td>15 – 7</td><td>reserved</td><td></td></tr>
<tr><td>Slave status information</td><td colspan="3">Scanning results</td></tr>
<tr><td rowspan="9">Slave status info 1 - 64</td><td>Bit</td><td>Description</td><td>Values</td></tr>
<tr><td>0</td><td>Polling status error occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>1</td><td>CRC error occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>2</td><td>An abort frame was reported by the DL</td><td>0 = False
1 = True</td></tr>
<tr><td>3</td><td>A polling response timeout occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>4</td><td>A received data buffer overflow occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>5</td><td>An invalid reception address filed was received</td><td>0 = False
1 = True</td></tr>
<tr><td>6</td><td>At least one retry has occurred</td><td>0 = False
1 = True</td></tr>
<tr><td>15 – 7</td><td>reserved</td><td></td></tr>
</table>

Attribute	Encoding		
Master transmitted status field	**Bit**	**Description**	**Values**
	0	FAL-user state	0 = Stop 1 = Run
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Stop 1 = Run
	3	Acyclic status	0 = Normal 1 = Error
	4	Acyclic enabled	0 = Disable 1 = Enable
	6 – 5	reserved	
	7	Master type	0 = Active 1 = Standby
	11 – 8	Size of bit-oriented data fields: RX, RY in Slots. A Slot is 4 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
	15 – 12	Size of word-oriented data fields: RWr, RWw in Slots. A Slot is 8 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
Slave transmitted status field	Transmitted status fields received from each Slave		
Slave status field 1 - 64	**Bit**	**Description**	**Values**
	0	FAL-user fuse status	0 = Normal 1 = Abnormal
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Complete 1 = Not received
	3	Slave parameter receive status	0 = Complete 1 = Not received
	4	Slave's FAL-user switch status	0 = No change 1 = Changed
	5	Cyclic transmission enabled	0 = Enable 1 = Disable
	6	reserved	
	7	FAL-user watchdog timer error detected	0 = Normal 1 = Error
	8	Acyclic status	0 = Normal 1 = Error
	9	Acyclic enabled	0 = Disabled 1 = Enabled

Attribute	Encoding		
	10	Acyclic type: Master/Slave or Peer/Peer	0 = M/S 1 = P/P
	11	reserved	
	12	Transmission status	0 = Normal 1 = Fault
	13	reserved	1
	15 – 14	Cyclic segmenting configuration	0 = 1x or not supported 1 = 2x 3 = 4x 4 = 8x
Network information	Real time performance parameters		
Current link scan time	ms		
Minimum link scan time	ms		
Maximum link scan time	ms		

5.6 M2 connection manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 19.

Table 19 – M2 connection manager attribute encoding

Attribute	Encoding		
Parameter information			
Transmission speed	0 = 156 (baud rate in kbit/s) 1 = 625 2 = 2 500 3 = reserved		
Last station number	1 - 64 = allowable range		
Point mode setting	0 = undefined 1 = 4 (points) 2 = 8 3 = 16		
Master station i/o point mode setting	0 = 16 (total points controlled by Master) 1 = 32 2 = 48 3 = 64 4 = 128 5 = 256 6 = 512 7 = 1024		
Network status information	Connection status		
Reception status information	Reception and monitoring timer status		
Reception status 1 - 64	**Bit**	**Description**	**Values**
	0	CRC error occurred	0 = Normal 1 = Error
	1	Parity error occurred	0 = Normal 1 = Error
	2	No response error occurred	0 = Normal 1 = Error
	3	reserved	
Slave status information	Slaves connection status		

Attribute	Encoding		
Slave status 1 - 64	**Bit**	**Description**	**Values**
	0	Slave station connection status	0 = Not connected 1 = Connected
	1	31 or more consecutively received errors including: CRC, parity or timeout	0 = Normal 1 = Fault

5.7 S1 connection manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 20.

Table 20 – S1 connection manager attribute encoding

Attribute	Encoding
Process data support level	0 = Level A 1 = Level B 2 = Level C 3 = reserved 4 = reserved 5 = Level B 1x cyclic seg 6 = Level C 1x cyclic seg 7 = reserved 8 = Level B 2x cyclic seg 9 = Level C 2x cyclic seg 10 = reserved 11 = Level B 4x cyclic 12 = Level C 4x cyclic 13 = reserved 14 = Level B 8x cyclic 15 = Level C 8x cyclic
Network status information	Status field transmitted by the connected Master.

Attribute	Encoding		
Master transmitted status field	**Bit**	**Description**	**Values**
	0	FAL-user state	0 = Stop 1 = Run
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Stop 1 = Run
	3	Acyclic status	0 = Normal 1 = Error
	4	Acyclic enabled	0 = Disable 1 = Enable
	6 – 5	reserved	
	7	Master type	0 = Active 1 = Standby
	11 – 8	Size of bit-oriented data fields: RX, RY in Slots. A Slot is 4 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
	15 – 12	Size of word-oriented data fields: RWr, RWw in Slots. A Slot is 8 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved

5.8 S2 connection manager PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 21.

Table 21 – S2 connection manager attribute encoding

Attribute	Encoding
Parameter information	Specifies the connection configuration of the Slave
Slave station number	1 - 64 = range of allowable values
Number of occupied slots	3 bits encoded as an unsigned integer 1 - 4 = range of allowable values

5.9 M1 cyclic transmission PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 22.

Table 22 – M1 cyclic transmission attribute encoding

Attribute	Encoding		
Master status	Bit	Description	Values
	0	FAL-user state	0 = Stop 1 = Run
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Stop 1 = Run
	3	Acyclic status	0 = Normal 1 = Error
	4	Acyclic enabled	0 = Disable 1 = Enable
	6 – 5	reserved	
	7	Master type	0 = Active 1 = Standby
	11 – 8	Size of bit-oriented data fields: RX, RY in Slots. A Slot is 4 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
	15 – 12	Size of word-oriented data fields: RWr, RWw in Slots. A Slot is 8 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
Data out	Process data registers set by the Master for Slave device output.		
RY data	A position mapped field of bit-oriented output data for all connected Slave devices ordered by Slot with 32 bits per Slot.		
RWw data	A position mapped field of word-oriented output data for all connected Slave devices ordered by Slot with 4 words per Slot.		
Data in	Process data registers read by the Master representing Slave device inputs.		
Number of Modules	1 - 64 = range of allowable values		
Slave input data			
Station number	1 - 64 = range of allowable values		

Attribute	Encoding		
Slave status	**Bit**	**Description**	**Values**
	0	FAL-user fuse status	0 = Normal 1 = Abnormal
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Complete 1 = Not received
	3	Slave parameter receive status	0 = Complete 1 = Not received
	4	Slave’s FAL-user switch status	0 = No change 1 = Changed
	5	Cyclic transmission enabled	0 = Enable 1 = Disable
	6	reserved	
	7	FAL-user watchdog timer error detected	0 = Normal 1 = Error
	8	Acyclic status	0 = Normal 1 = Error
	9	Acyclic enabled	0 = Disabled 1 = Enabled
	10	Acyclic type: Master/Slave or Peer/Peer	0 = M/S 1 = P/P
	11	reserved	
	12	Transmission status	0 = Normal 1 = Fault
	13	reserved	1
	15 – 14	Cyclic segmenting configuration	0 = 1x or not supported 1 = 2x 3 = 4x 4 = 8x
RX data	A field containing the bit-oriented input data from Slave device n ordered by Slot with 32 bits per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		
RWr data	A field containing the word-oriented input data from Slave device n ordered by Slot with 4 words per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		

5.10 M2 cyclic transmission PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 23.

Table 23 – M2 cyclic transmission attribute encoding

Attribute	Encoding
RY data	A field containing the bit-oriented output data for the Slave devices ordered by Slot with n bits per Slot. Where n = 4, 8 or 16 based upon the point-mode-setting value of the M2 connection manager.
RX data	A field containing the bit-oriented input data from the Slave devices ordered by Slot with n bits per Slot. Where n = 4, 8 or 16 based upon the point-mode-setting value of the M2 connection manager.

5.11 S1 cyclic transmission PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 24.

Table 24 – S1 cyclic transmission attribute encoding

Attribute	Encoding		
Slave status	**Bit**	**Description**	**Values**
	0	FAL-user fuse status	0 = Normal 1 = Abnormal
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Complete 1 = Not received
	3	Slave parameter receive status	0 = Complete 1 = Not received
	4	Slave's FAL-user switch status	0 = No change 1 = Changed
	5	Cyclic transmission enabled	0 = Enable 1 = Disable
	6	reserved	
	7	FAL-user watchdog timer error detected	0 = Normal 1 = Error
	8	Acyclic status	0 = Normal 1 = Error
	9	Acyclic enabled	0 = Disabled 1 = Enabled
	10	Acyclic type: Master/Slave or Peer/Peer	0 = M/S 1 = P/P
	11	reserved	
	12	Transmission status	0 = Normal 1 = Fault
	13	reserved	1
	15 – 14	Cyclic segmenting configuration	0 = 1x or not supported 1 = 2x 3 = 4x 4 = 8x
Data out	The process data received from the Master		
RY data	A field containing the bit-oriented input data ordered by Slot with 32 bits per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		

Attribute	Encoding		
RWw data	A field containing the word-oriented input data ordered by Slot with 4 words per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		
Master status	**Bit**	**Description**	**Values**
	0	FAL-user state	0 = Stop 1 = Run
	1	FAL-user status	0 = Normal 1 = Fault
	2	Cyclic refresh status	0 = Stop 1 = Run
	3	Acyclic status	0 = Normal 1 = Error
	4	Acyclic enabled	0 = Disable 1 = Enable
	6 – 5	reserved	
	7	Master type	0 = Active 1 = Standby
	11 – 8	Size of bit-oriented data fields: RX, RY in Slots. A Slot is 4 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
	15 – 12	Size of word-oriented data fields: RWr, RWw in Slots. A Slot is 8 octets in length	0 = 0 1 = 8 2 = 16 3 = 24 4 = 32 5 = 40 6 = 48 7 = 56 8 = 64 9 - 15 = reserved
Data in	The process data transmitted to the Master		
RX data	A field containing the bit-oriented input data ordered by Slot with 32 bits per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		
RWr data	A field containing the word-oriented input data ordered by Slot with 4 words per Slot. The number of Slots occupied by the Slave device determines the total length of this field.		

5.12 S2 cyclic transmission PDU encoding

The specific PDU encoding for attributes belonging to this class is described in Table 25.

Table 25 – S2 cyclic transmission attribute encoding

Attribute	Encoding
RY data	A field containing the bit-oriented output data from the Master device ordered by Slot with n bits per Slot. Where n = 4, 8 or 16 based upon the point-mode-setting value of the associated M2 connection manager. The total length of this field is determined by n times the number Slots occupied by the Slave device as specified by the S2 connection manager object.
RX data	A field containing the bit-oriented input data for the Master device ordered by Slot with n bits per Slot. Where n = 4, 8 or 16 based upon the point-mode-setting value of the associated M2 connection manager. The total length of this field is determined by n times the number Slots occupied by the Slave device as specified by the S2 connection manager object.

5.13 Acyclic transmission PDU encoding

5.13.1 Acyclic message encoding

The specific PDU encoding for acyclic messages is specified in Table 26.

Table 26 – Acyclic transmission – message data encoding

Field	Size (octets)	Value
Length	2	Length of message (in octets) not including the first 4 octets. (i.e., set to a value four less than the total length of the message)
reserved	1	for future use (set = 0)
Type and sequence	1	bits 3 – 0 = type (set = 0) <u>M2 type:</u> bits 7 – 4 = sequence number in the range 1-7. FAL user increments this field by 1 upon each successive request, rolling back to 1 after 7. <u>S2 type:</u> bits 6 – 4 = reserved field – used by DL-protocol for segmenting and reassembly Bit 7 = sequence flag, alternating 0 and 1. FAL user toggles this bit on each successive request
Segment number	1	reserved field – used by DL-protocol for segmenting and reassembly
Data type	1	b7 = priority (0 = low; 1 = high) b6 = response required (0 = true; 1 = false) b5 – b0 = reserved
Destination address	1	Station number of the intended destination AREP
Source address	1	Station number of the originating device.
Destination app type	1	FAL user target application type (set = 33)
Source app type	1	FAL user source application type (set = 33)
Destination app module	1	FAL-user target application handler module (0 = network, 1-255 = FAL user specific)
Source app module	1	DLS-user source application handler module (0 = network, 1-255 = FAL user specific)
Destination network id	1	for future extension (set = 0)
Destination address	1	for future extension (set = 0)

Field	Size (octets)	Value
Destination id	2	bits 9 – 0 = fixed (set = 1023) bits 15 – 10 = Station number of the intended destination AREP
Source network id	1	for future extension (set = 0)
Source address	1	for future extension (set = 0)
Source id	2	bits 9 – 0 = fixed (set = 1 23) bits 15 – 10 = Station number of the originating device
Command header	8	Acyclic command header as specified in Table 27
Command parameters	0 – 960	Parameter field for the command as specified in 5.13.2

Table 27 – Command header format

Field	Size (octets)	Value
Length	2	length of command plus command data not including this field. (i.e., set to a value six more than the length of command data)
Command type	1	command code as specified in Table 28
reserved	1	(set = 0)
SAP	2	FAL user specific Service Access Point identifier
Response code	2	bits 7 – 0 = error fields bit 8 = priority (0 = Warning; 1 = Major error) bits 11 – 9 = error location (set by DLS-user) bits 15 – 12 = FAL user specific field

Table 28 – Command codes

Code	Command
0	undefined
1	Parameter block 1
2	Parameter block 2
3	System information acquisition
4	Memory access information acquisition
5 – 7	reserved
8	Run
9	Stop
10 – 14	reserved
15	Line test
16	Memory read
17	reserved
18	Memory write
19 – 31	reserved
32	Special command reserved for compliance with the semiconductor manufacturing industry the specifics of which are beyond the scope of this specification
33 – 95	reserved
96 – 127	open area for FAL user definition

5.13.2 Acyclic command parameter encoding

5.13.2.1 Parameter block 1

The encoding of the request parameter field for this command is specified in Figure 1.

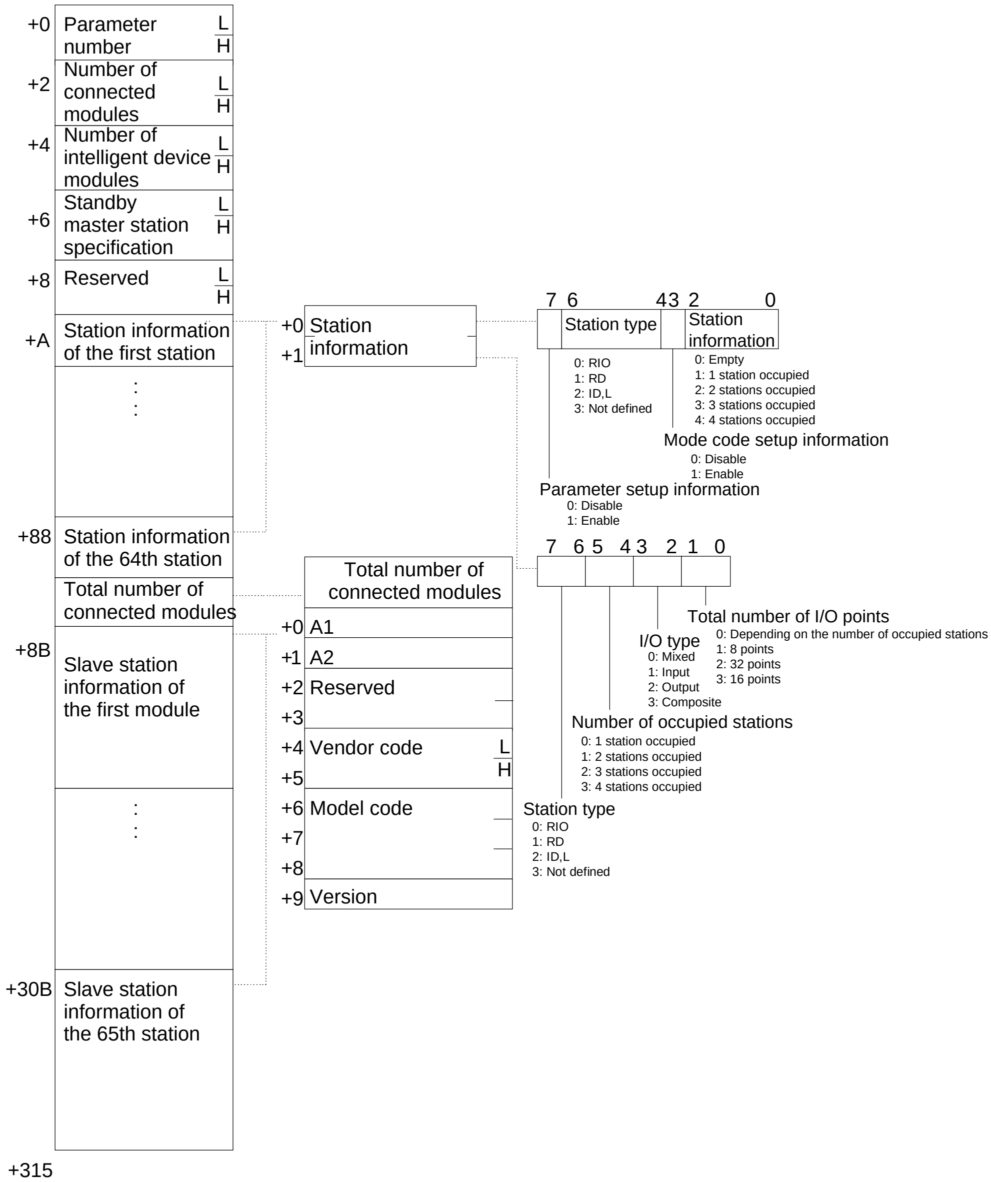

Figure 1 – Parameter block 1 command parameter field

5.13.2.2 Parameter block 2

The encoding of the request parameter field for this command is specified in Figure 2.

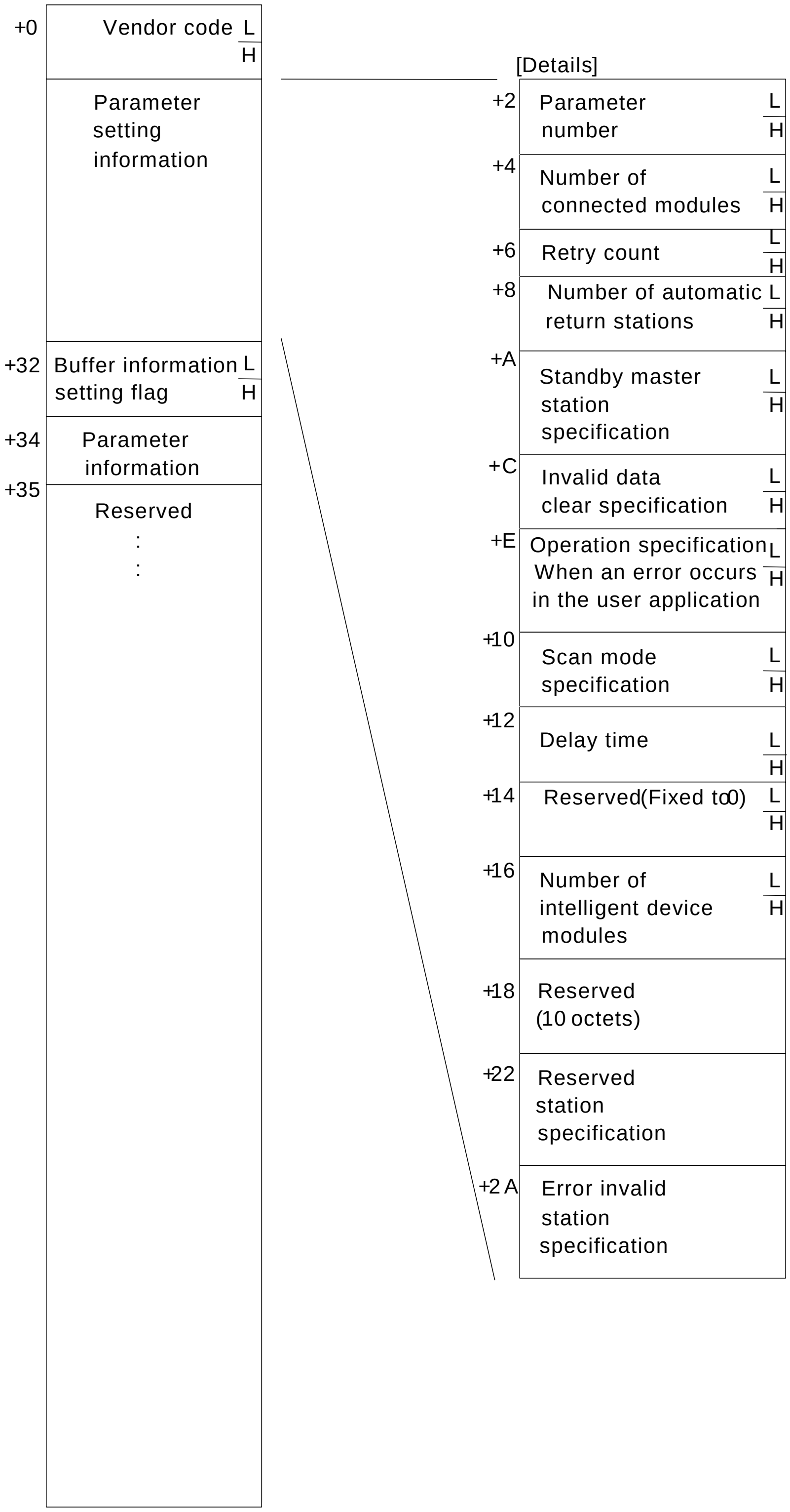

Figure 2 – Parameter block 2 command parameter field

5.13.2.3 System information acquisition

The encoding of the response parameter field for this command is specified in Table 29.

Table 29 – System information command parameter field

Octet	Field	Encoding
0 – 1	Vendor code	Field specified in M2 device manager
2 – 5	Model code	Field specified in M2 device manager
6 – 7	Software version	Field specified in M2 device manager
8 – 23	Supported commands	128 bit, bit-mapped field where the bit location corresponds to a command number (0 - 127). The values contained in the bit indicates if the command is supported. 0 = not supported 1 = supported
24	Segmenting limit	Specifies the maximum number of segments supported for acyclic messaging 1 - 7 = range of allowable values
25 – 26	Data buffer limit	Specifies the maximum length for acyclic messaging 0 - 960 = range of allowable values

5.13.2.4 Memory access information acquisition

The encoding of the response parameter field for this command is specified in Table 30.

Table 30 – System information command parameter field

Octet	Field	Encoding
0 – 31	List of available access codes	256 bit, bit-mapped field where the bit location corresponds to an access code (0 - 255). The values contained in the bit indicates if the access code is supported. 0 = supported 1 = not supported
32 – 35	Device name	Coded value to represent the device type. Specific value are FAL user specific and beyond the scope of this specification.
36 – 37	Number of access points	Specifies the number memory points accessible over the network.

5.13.2.5 Run

The encoding of the request parameter field for this command is specified in Table 31.

Table 31 – System information command parameter field

Octet	Field	Encoding
0 – 1	Run mode	0 = undefined 1 = forced run mode 2 = undefined 3 = normal run mode 4 - 65535 = reserved
2	Clear mode	0 = no not clear device 1 = clear all but locked range 2 = clear complete device 3 - 255 = reserved
3	Signal flow mode	reserved (set = 0)

5.13.2.6 Stop

The encoding of the request parameter field for this command is specified in Table 32.

Table 32 – System information command parameter field

Octet	Field	Encoding
0 – 1	Run mode	0 = undefined 1 = forced run mode 2 = undefined 3 = normal run mode 4 - 65535 = reserved

5.13.2.7 Line test

The encoding of the response parameter field for this command is specified in Table 33.

Table 33 – Line test command parameter field

Octet	bit	description	value
0 – 1	15 – 0	Vendor code	The assignment and management of vendor codes is FAL user specific and beyond the scope of this specification.
2	1 – 0	Total number of used bit-oriented data bits (both RX and RY combined)	0 = full use 1 = 8 used 2 = 32 used 3 = 16 used
	3 – 2	Distribution of used bit-oriented data bits	0 = RX and RY in equal sizes 1 = RX only 2 = RY only 3 = other RX / RY mix
	5 – 4	Number of occupied station slots	0 = 1 slot 1 = 2 slots 2 = 3 slots 3 = 4 slots
	7 – 6	reserved	(set = 0)
3	0	FAL user switch setting	0 = normal 1 = abnormal
	1	FAL user output upon fault	0 = clear 1 = hold
	5 – 2	reserved	(set = 0)
	7 – 6	Process data support level	0 = level A 1 = level B 2 = level C 3 = reserved
4	6 – 0	reserved	(set = 0)
	7	Messaging system support. Acyclic data format supports messaging type commands.	0 = not supported 1 = supported
5	5 – 0	software version	1 – 63
	7 – 6	Cyclic data segmenting support	0 = does not support cyclic data segmenting 1 = supports cyclic data segmenting 2 = reserved 3 = reserved
6 – 9	-	Master’s test data	The 4 octets of test data receive by the Slave device form the Master during the connection process.

5.13.2.8 Memory read

The encoding of the request parameter field for this command is specified in Table 34.

Table 34 – Memory read command parameter field

Octet	Field	Encoding		
0 – 1	Quantity	Specifies the number of attribute specifications that follow 1 - 160 = range of allowable values		
2	Attribute type	**Bit**	**Description**	**Values**
		0	Memory location with respect to device	0 = internal 1 = external
		2 – 1	Access type	0 = bit 1 = octet 2 = word 3 = double word
		7 – 3	reserved	0
3	Access code	**Bit**	**Description**	**Values**
		0	Bit input data	0 = False 1 = True
		1	Bit output data	0 = False 1 = True
		2	Word data	0 = False 1 = True
		3	Timer	0 = False 1 = True
		4	Counter	0 = False 1 = True
		5	Link	0 = False 1 = True
		6	Status	0 = False 1 = True
		7	Retain	0 = False 1 = True
4 – 5	Address	Specifies the memory address		
6 – 7	Number of points to read	for Bit type attribute 0 - 7680 = range of allowable values for octet type attribute 0 - 960 = range of allowable values for Word type attribute 0 - 480 = range of allowable values		
…	… Blocks of 6 octets Attribute Specifications as specified in Quantity …			

The encoding of the response parameter field for this command is the data field specified by the request.

5.13.2.9 Memory write

The encoding of the request parameter field for this command is specified in Table 35.

Table 35 – Memory write command parameter field

<table>
<tr><th>Octet</th><th>Field</th><th colspan="3">Encoding</th></tr>
<tr><td>0 – 1</td><td>Quantity</td><td colspan="3">Specifies the number of attribute specifications that follow
1 - 160 = range of allowable values</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">Attribute type</td><th>Bit</th><th>Description</th><th>Values</th></tr>
<tr><td>0</td><td>Memory location with respect to device</td><td>0 = internal
1 = external</td></tr>
<tr><td>2 – 1</td><td>Access type</td><td>0 = bit
1 = octet
2 = word
3 = double word</td></tr>
<tr><td>7 – 3</td><td>reserved</td><td>0</td></tr>
<tr><td rowspan="9">3</td><td rowspan="9">Access code and memory type</td><th>Bit</th><th>Description</th><th>Values</th></tr>
<tr><td>0</td><td>Bit input data</td><td>0 = False
1 = True</td></tr>
<tr><td>1</td><td>Bit output data</td><td>0 = False
1 = True</td></tr>
<tr><td>2</td><td>Word data</td><td>0 = False
1 = True</td></tr>
<tr><td>3</td><td>Timer</td><td>0 = False
1 = True</td></tr>
<tr><td>4</td><td>Counter</td><td>0 = False
1 = True</td></tr>
<tr><td>5</td><td>Link</td><td>0 = False
1 = True</td></tr>
<tr><td>6</td><td>Status</td><td>0 = False
1 = True</td></tr>
<tr><td>7</td><td>Retain</td><td>0 = False
1 = True</td></tr>
<tr><td>4 – 5</td><td>Address</td><td colspan="3">Specifies the memory address</td></tr>
<tr><td>6 – 7</td><td>Number of points to read</td><td colspan="3">for Bit type attribute
0 - 7680 = range of allowable values
for octet type attribute
0 - 960 = range of allowable values
for Word type attribute
0 - 480 = range of allowable values</td></tr>
<tr><td>…</td><td colspan="4">…
Blocks of 6 octets Attribute Specifications as specified in Quantity
…</td></tr>
<tr><td>…</td><td colspan="4">…
Data to be written, encoded as specified in the request
…</td></tr>
</table>

6 Structure of FAL protocol state machines

Interface to FAL services and protocol machines are specified in this subclause.

NOTE The state machines specified in this subclause and ARPMs defined in the following sections only define the valid events for each. It is a local matter to handle these invalid events.

The behavior of the FAL is described by three integrated protocol machines. The three protocol machines are: FAL Service Protocol Machine (FSPM), the Application Relationship Protocol Machine (ARPM), and the Data-link layer Mapping Protocol Machine (DMPM). The relationship among these protocol machines as well as primitives exchanged among them are depicted in Figure 3.

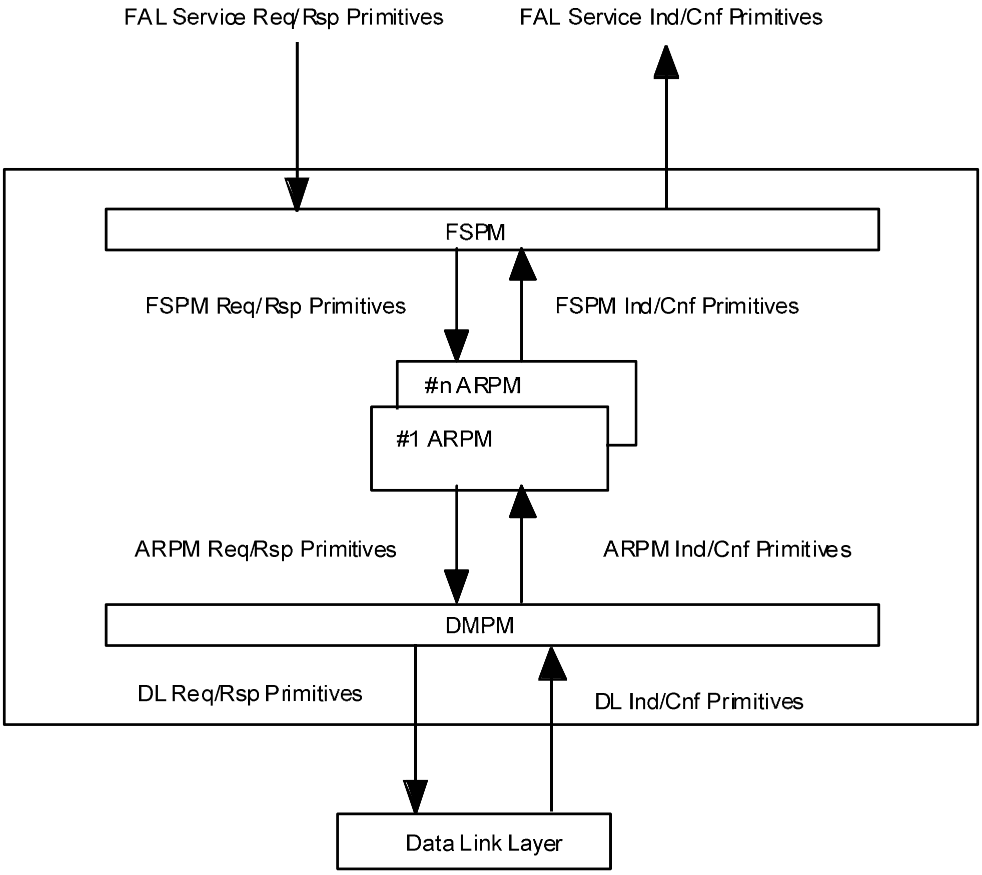

Figure 3 – Relationships among protocol machines and adjacent layers

The FSPM describes the service interface between the FAL-user and a particular AREP. The FSPM is common to all the AREP classes and does not have any state changes. The FSPM is responsible for the following activities:

a) to accept service primitives from the FAL service user and convert them into FAL internal primitives;
b) to select an appropriate ARPM state machine based on the AREP Identifier parameter supplied by the FAL-user and send FAL internal primitives to the selected ARPM;
c) to accept FAL internal primitives from the ARPM and convert them into service primitives for the FAL-user.

The ARPM describes the establishment and release of an AR and exchange of FAL-PDUs with a remote ARPM(s). The ARPM is responsible for the following activities:

a) to accept FAL internal primitives from the FSPM and create and send other FAL internal primitives to either the FSPM or the DMPM, based on the AREP and primitive types;
b) to accept FAL internal primitives from the DMPM and send them to the FSPM as a form of FAL internal primitives;
c) if the primitives are for the Establish or Release service, it shall try to establish or release the specified AR.

The DMPM describes the mapping between the FAL and the DLL. It is common to all the AREP types and does not have any state changes. The DMPM is responsible for the following activities:

a) to accept FAL internal primitives from the ARPM, prepare DLL service primitives, and send them to the DLL;

b) to receive DLL indication or confirmation primitives from the DLL and send them to the ARPM in a form of FAL internal primitives.

7 AP-context state machine

The type 18 FAL does not implement an AP-context state machine.

8 FAL service protocol machine (FSPM)

8.1 Overview

The FSPM provides the interface to the FAL user in the form of service handlers which convert service parameters into APDUs and process service requests from the FAL user or convert APDUs into service parameters and deliver service indications to the FAL user.

8.2 FAL service primitives

The FSPM operates in a single state with events defined by the receipt of service primitives. The descriptions of these receipt events are specified in Table 36.

Table 36 – FSPM events

Primitive	Description
Get	Retrieves the value of the attribute identified.
Set	Sets the value of the attribute identified as specified.
Error	Upon receipt of an error indication from the ARPM, an analogous error notification is delivered to the FAL user.
Connect	Upon receipt of a connect service request from the FAL user, a connect service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Disconnect	Upon receipt of a disconnect service request from the FAL user, a disconnect service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Start scan	Upon receipt of a start scan service request from the FAL user, a start scan service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Stop scan	Upon receipt of a stop scan service request from the FAL user, a stop scan service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Activate standby	Upon receipt of an activate standby service request from the FAL user, an activate standby service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Verify slave configuration	Upon receipt of a verify slave configuration service request from the FAL user, a verify slave configuration service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Trigger transmission	Upon receipt of a trigger transmissions service request from the FAL user, a trigger transmission service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Data received	Upon receipt of a data received indication from the ARPM, a data received indication is delivered to the FAL user.
Send message	Upon receipt of a send message service request from the FAL user, a send message service request is delivered to the ARPM and the confirmation is returned to the FAL user.
Message received	Upon receipt of a message received indication from the ARPM, a message received indication is delivered to the FAL user.

9 AR protocol machine (ARPM)

9.1 Overview

The ARPM manages the functions and behaviors of the ARs by

a) receiving, decoding and processing service primitives from the FSPM,

b) preparing, encoding and delivering service primitives to the DMPM,

c) receiving, decoding and processing service primitives from the DMPM,

d) preparing, encoding and delivering service primitives to the FSPM,

e) monitoring critical functions of the ARs including timeout times and other fault conditions,

f) delivering event notifications to the APCSM.

There are three types of AR: M1 Master, M2 Master and Slave.

9.2 M1 master ARPM

9.2.1 Overview

The M1 Master ARPM manages the behavioral states, transitions and interactions of an M1 Master AR. As shown in Figure 4, there are two states.

Sub-states are not represented as definitive states, but exist as abstractions used to identify a set of behaviors with a state. The M1 Master ARPM defines two sub-states within the running state, scanning and not scanning.

The ARPM is initiated in the Idle state.

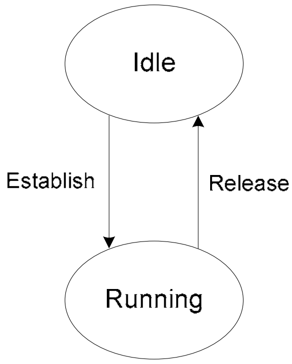

Figure 4 – ARPM M1 master AR state diagram

9.2.2 State descriptions

9.2.2.1 Idle

The FAL is not connected to the network. The only behavior associated with the Idle state is a determination of errors or other fault conditions that are specified as preventing the transition to the Running state.

If configured for automatic network connection, the ARPM will automatically invoke its connect service request resulting in an ARP-connect M1 type primitive being delivered to the DMPM.

9.2.2.2 Running state – not scanning

In this sub-state the FAL is connected to the network, but not performing the scanning function. A Master Class 1 whose M1 connection manager object is configured with a Station number of Standby Master (128) exhibits all the behavior of a Slave Class 1.

9.2.2.3 Running state – scanning

9.2.2.3.1 General description

The initiation mechanism of a scan cycle depends on the setting of the M1 connection manager object's Scan mode specification. For Triggered (single loop) type, the cycle is started upon receipt of an trigger transmission service request and is therefore under the timing control of the FAL user. For Free-running (continuous loop) type, the cycle shall be started after a stack processing time of between 0.1 and 500 mSec based upon processing capabilities. The one exception to this is for ARs over connections configured for 156 kbit/s baud rate, in which case the minimum processing time must be greater than 180 uSec.

Whether configured for Triggered or Free-running Scan mode specification, a scan cycle is initiated with an ARP-trigger primitive being delivered to the DMPM.

The ARPM creates and maintains a list of active Slaves, originally compiled from the data in the M1 connection manager’s Parameter information structure Slave information array, and then updated based upon Slaves’ status as determined by monitor timers, error processing and ARP-error indications. This list of active Slaves identifies which connected Slaves are active and which are in the suspended state.

If there are one or more Slaves in the list of active Slaves identified as being in the suspended state, then every other scan cycle is replaced with a reestablish cycle which is performed by the ARPM delivering an ARP-resume primitive to the DMPM for each connected Slave in the suspended state. This may result in a transition of one or more Slaves’ status form suspended to running.

9.2.2.3.2 Buffer management

The ARPM shall ensure that sufficient buffer management techniques are implemented to effect absolute consistency in the transmission and receipt of cyclic data as well as acyclic messages. Due to the nature of fragmentation at the lower levels, this may involve monitoring the fragment count and/or interlocking request/response pairs.

Another consideration in the determination of data consistency is the timing of a free-running M1 type ARPM. This is specified as synchronous with the network, and therefore asynchronous with the FAL user. Hence methods may be required, such as double buffering, to ensure data consistency.

Both the cyclic data buffer and acyclic message buffer are accessed by the FAL via the Get and Set services. The FSPM is signaled that fresh data is available for the FSPM via the data received and message received indications. The presence of a fresh acyclic message for delivery is signaled to the ARPM from the FSPM via receipt of the trigger transmission primitive.

The ARPM manages the consistency of the transmissions by controlling the delivery of the ARP-data update and the ARP-message update primitives to the DMPM.

9.2.2.3.3 Error processing

Receipt of an ARP-error indication of type:

a) Frame-error
b) CRC-error
c) Abort-error
d) Invalid-address, or
e) DLE-Slave-timeout

results in a comparison of the number of times during the current scan attempt this error has occurred against the value of the M1 connection manager’s Number of retries setting. If not exceeded, the current Slave scan attempt is repeated and the retry is noted. If exceeded, the status of the station number of the Slave being scanned is changed in the list of active Slaves from active to suspended.

Receipt of an ARP-error indication of type:

f) Buffer-overflow

results in a Release event.

Each time the list of active Slaves is accessed, it is checked for the condition of a suspended state for all Slave station numbers in the list of active Slaves, whereupon appropriate updates are made to the list.

All error conditions, faults and changes in status are reflected by the ARPM update of the M1 connection manager's Network-status-information attribute.

9.2.3 State tables

The state tables for the M1 Master ARPM is shown in Table 37, Table 38 and Table 39.

NOTE The get, set and error service primitives, in and of themselves, have no effect on states. Their corresponding new attribute values or indicated error conditions may result in effected events as specified.

It is specified that the receipt of any primitive not included in the tables results in an error response.

Table 37 – M1 master state-event table 1 – events

Current State	Event	Status	Action	Next State
Idle	Establish	All conditions	Initiate the methods specified for the Running state	Running
Idle	Release	All conditions	Return an error	SAME
Running	Establish	Sub-state == scanning	Return an error	SAME
Running	Establish	Sub-state == not scanning	Return an error	SAME
Running	Release	Sub-state == scanning	Invoke a stop scan service request, then initiate the methods specified for the Idle state	Idle
Running	Release	Sub-state == not scanning	Initiate the methods specified for the Idle state	Idle

Table 38 – M1 master state-event table 2 – receipt of FSPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	Connect	All conditions	Deliver an ARP-connect service request to the DMPM Start a 1.67 sec timer An ***establish event*** results if the connect is successful prior to expiration of the timer Otherwise, repeat action	SAME
Idle	Disconnect	All conditions	Return an error	SAME
Idle	Start scan	All conditions	Return an error	SAME
Idle	Stop scan	All conditions	Return an error	SAME
Idle	Activate standby	All conditions	Return an error	SAME
Idle	Verify salve configuration	All conditions	Return an error	SAME
Idle	Trigger transmission	All conditions	Return an error	SAME
Idle	Send message	All conditions	Return an error	SAME
Running	Connect	All conditions	Return an error	SAME
Running	Disconnect	All conditions	Deliver an ARP-disconnect A ***release event*** results	SAME
Running	Start scan	Sub-state == not scanning	Initiate the methods specified for the running state –	SAME

Current State	Primitive received	Status	Action	Next State
			scanning. See 9.2.2.3 Transition to: sub-state = scanning	
Running	Start scan	Sub-state == scanning	Return an error	SAME
Running	Stop scan	Sub-state == not scanning	Return an error	SAME
Running	Stop scan	Sub-state == scanning	Initiate the methods specified for the running state – not scanning. See 9.2.2.2 Transition to: sub-state = not scanning	SAME
Running	Activate standby	Sub-state == not scanning and station number == 128	Deliver an ARP-activate standby service request to the DMPM Transition to: sub-state = scanning	SAME
Running	Activate standby	Sub-state == scanning or station number ≠ 128	Return an error	SAME
Running	Verify salve configuration	All conditions	Compare data collected from connected slaves against the configuration specified by the connection manager object. If highest slave station number collected > last station number (parameter information) Then return an error If highest slave station number collected < last station number (parameter information) Then modify last station number value accordingly If number of occupied slots and station type collected ≠ number of occupied slots and station type (parameter information) Then return an error If number of occupied slots (parameter information) results in overlap with subsequent slave station Then return an error	SAME
Running	Trigger transmission	Sub-state == scanning and scan mode == triggered	Deliver an ARP-trigger service request to the DMPM	SAME
Running	Trigger transmission	Sub-state == not scanning or scan mode == free-running	Return an error	SAME
Running	Send message	Sub-state == scanning	Deliver an ARP-message update service request to the DMPM	SAME
Running	Send message	Sub-state == not scanning	Return an error	SAME

Table 39 – M1 master state-event table 3 – receipt of DMPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	ARP-cyclic received	All conditions	Ignored	SAME
Idle	ARP-acyclic received	All conditions	Ignored	SAME
Running	ARP-cyclic received	All conditions	deliver a data received primitive to the FSPM	SAME
Running	ARP-acyclic received	All conditions	deliver a message received primitive to the FSPM	SAME

9.3 M2 master ARPM

9.3.1 Overview

The M2 Master ARPM manages the behavioral states, transitions and interactions of an M2 Master AR. As shown in Figure 5, there are two states. The ARPM is initiated in the Idle state.

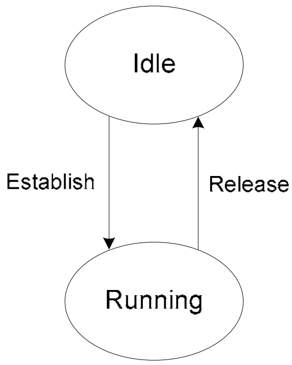

Figure 5 – ARPM M2 master AR state diagram

9.3.2 State descriptions

9.3.2.1 Idle

The FAL is not connected to the network. The only behavior associated with the Idle state is a determination of errors or other fault conditions that are specified as preventing the transition to the Running state.

If configured for automatic network connection, the ARPM will automatically invoke its connect service request resulting in an ARP-connect M2 type primitive being delivered to the DMPM.

9.3.2.2 Running

9.3.2.2.1 General description

While in the running, the M2 Master ARPM is always scanning. Scanning for the Master Class 2 FAL is performed automatically by the protocol layers below the FAL. All connected Slave stations are scanned from station number 1 through the station number specified by the Last station number. This is the value specified by the M2 connection manager and possibly updated by the FSPM during the execution of the Verify slave configuration service as specified.

The scanning of Slaves connected is initiated by an ARP-trigger primitive being delivered to the DMPM. This happens automatically at a rate that sustains the scan cycle rate as determined by the Transmission speed, Point mode setting and Last station number of the Parameter information structure of the M2 connection manager.

9.3.2.2.2 Buffer management

The ARPM shall ensure that sufficient buffer management techniques are implemented to effect absolute consistency in the transmission and receipt of cyclic data. Due to the nature of fragmentation at the lower levels, this may involve monitoring the fragment count. Because scanning is synchronous with the network, and therefore asynchronous with the FAL user, methods may be required, such as double buffering, to ensure data consistency.

The cyclic data buffer is accessed by the FAL via the Get and Set services. The FAL is signaled that fresh data is available for the FAL via the FSP-cyclic-received indication.

The ARPM manages the consistency of the transmissions by controlling the delivery of the ARP-data update primitive to the DMPM.

9.3.2.2.3 Error processing

Receipt of an ARP-error indication of type:

a) Frame-error
b) CRC-error
c) DLE-Slave-timeout

results in discarding the data received from the Slave and a prevention of the associated data received primitive from being delivered to the FSPM.

The ARPM also checks the parity of the M2 connection manager's Reception status information and if determined inconsistent, discards the data received from the Slave and prevents the associated data received primitive from being delivered to the FSPM.

9.3.3 State tables

The state tables for the M2 Master ARPM is shown in Table 39, Table 41 and Table 42.

NOTE The get, set and error service primitives, in and of themselves, have no effect on states. Their corresponding new attribute values or indicated error conditions may result in effected events as specified.

It is specified that the receipt of any primitive not included in the tables results in an error response.

Table 40 – M2 master state-event table 1 – events

Current State	Event	Status	Action	Next State
Idle	Establish	All conditions	Initiate the methods specified for the Running state	Running
Idle	Release	All conditions	Return an error	SAME
Running	Establish	All conditions	Return an error	SAME
Running	Release	All conditions	Initiate the methods specified for the Idle state	Idle

Table 41 – M2 master state-event table 2 – receipt of FSPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	Connect	All conditions	Deliver an ARP-connect service request to the DMPM Start a 1.67 sec timer An ***establish event*** results if the connect is successful prior to expiration of the timer Otherwise, repeat action	SAME
Idle	Disconnect	All conditions	Return an error	SAME
Idle	Verify salve configuration	All conditions	Return an error	SAME
Running	Connect	All conditions	Return an error	SAME
Running	Disconnect	All conditions	Deliver an ARP-disconnect A ***release event*** results	SAME
Running	Verify salve configuration	All conditions	Compare data collected from connected slaves against the configuration specified by the connection manager object. If highest slave station number collected > last station number (parameter information) Then return an error If highest slave station number collected < last station number (parameter information) Then modify last station number value accordingly If number of occupied slots and station type collected ≠ number of occupied slots and station type (parameter information) Then return an error If number of occupied slots (parameter information) results in overlap with subsequent slave station Then return an error	SAME

Table 42 – M2 master state-event table 3 – receipt of DMPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	ARP-cyclic received	All conditions	Ignored	SAME
Running	ARP-cyclic received	All conditions	deliver a data received primitive to the FSPM	SAME

9.4 Slave ARPM

9.4.1 Overview

The Slave ARPM manages the behavioral states, transitions and interactions of a Slave AR. As shown in Figure 6, there are three states. The ARPM is initiated in the Idle state.

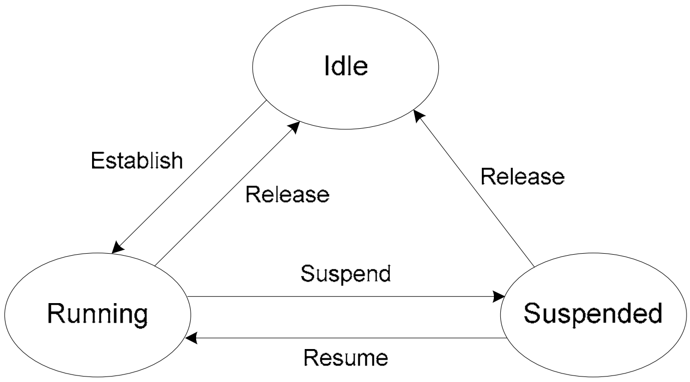

Figure 6 – ARPM slave AR state diagram

9.4.2 State descriptions

9.4.2.1 Idle

The FAL is not connected to the network. The only behavior associated with the Idle state is a determination of errors or other fault conditions that are specified as preventing the transition to the Running state.

If configured for automatic network connection, the ARPM will automatically invoke its connect service request resulting in an ARP-connect M2 type primitive being delivered to the DMPM.

9.4.2.2 Running

9.4.2.2.1 General description

Because the scanning behavior for a Slave type FAL is performed in response to receipt of scanning transmissions from the connected Master, this behavior is handled entirely at levels below the FAL.

9.4.2.2.2 Buffer management

The ARPM shall ensure that sufficient buffer management techniques are implemented to effect absolute consistency in the transmission and receipt of cyclic data. Due to the nature of fragmentation at the lower levels, this may involve monitoring the fragment count. Because scanning is synchronous with the network, and therefore asynchronous with the FAL user, methods may be required, such as double buffering, to ensure data consistency.

The cyclic data buffer is accessed by the FAL via the Get and Set services. The FAL is signaled that fresh data is available for the FAL via the FSP-cyclic-received indication.

In the case of S1 type Slaves with a process data support level C, the presence of a fresh acyclic message for delivery is signaled to the ARPM from the FSPM via receipt of the ARP-send primitive and the FSPM is signaled that a fresh acyclic message is available for the FSPM via the message received indication.

The ARPM manages the consistency of the transmissions by controlling the delivery of the ARP-data update and the ARP-message update primitives to the DMPM.

9.4.2.2.3 Monitor timer

The Slave type ARPM implements a timer to monitor the connection with the Master. Upon expiration of this timer, the ARPM is subjected to a Suspend event. The values used for these timers are specified in Table 43 and Table 43 for S1 and S2 type Slaves respectively.

Table 43 – S1 connect monitoring time

Baud rate (kbit/s)	Time (ms)
156	839
625	420
2500	105
5000	52.4
10000	52.4

Table 44 – S2 connect monitoring time

Baud rate (kbit/s)	Time (ms)
156	858
625	230
2500	66

9.4.2.2.4 Error processing

Receipt of an ARP-error indication of type:

a) Frame-error
b) CRC-error
c) Abort-error

results in discarding the data received from the Master and a prevention of the associated data received service indication or a message received service indication from being delivered to the FSPM.

Receipt of an ARP-error indication of type:

d) DLE-Master-timeout

results in a Suspend event for the ARPM.

9.4.2.3 Suspended

Unless prevented by a fault condition, the ARPM repeatedly attempts to recover from the suspend state by delivering ARP-resume service requests to the DMPM. Acknowledgement of a successful resume event is detected upon receipt of a ARP-cyclic-received service indication.

9.4.3 State tables

The state tables for the Slave ARPM is shown in Table 44, Table 46 and Table 47.

NOTE The get, set and error service primitives, in and of themselves, have no effect on states. Their corresponding new attribute values or indicated error conditions may result in effected events as specified.

It is specified that the receipt of any primitive not included in the tables results in an error response.

Table 45 – Slave state-event table 1 – events

Current State	Event	Status	Action	Next State
Idle	Establish	All conditions	Initiate the methods specified for the Running state	Running
Idle	Release	All conditions	Return an error	SAME
Running	Establish	All conditions	Return an error	SAME
Running	Release	All conditions	Initiate the methods specified for the Idle state	Idle

Table 46 – Slave state-event table 2 – receipt of FSPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	Connect	All conditions	Deliver an ARP-connect service request to the DMPM Start a 1.67 sec timer An ***establish event*** results if the connect is successful prior to expiration of the timer Otherwise, repeat action	SAME
Idle	Disconnect	All conditions	Return an error	SAME
Idle	Send message	All conditions	Return an error	SAME
Running	Connect	All conditions	Return an error	SAME
Running	Disconnect	All conditions	Deliver an ARP-disconnect A ***release event*** results	SAME
Running	Send message	Slave type == S1 and Configured process data support level == C and Sub-state == scanning	Deliver an ARP-message update service request to the DMPM	SAME
Running	Send message	Sub-state == not scanning	Return an error	SAME

Table 47 – Slave state-event table 3 – receipt of DMPM service primitives

Current State	Primitive received	Status	Action	Next State
Idle	ARP-cyclic received	All conditions	Ignored	SAME
Idle	ARP-acyclic received	All conditions	Ignored	SAME
Running	ARP-cyclic received	All conditions	deliver a data received primitive to the FSPM	SAME
Running	ARP-acyclic received	All conditions	deliver a message received primitive to the FSPM	SAME

10 DLL mapping protocol machine (DMPM)

10.1 Overview

The DMPM maps the ARPM service requests to DL service requests (converting APDUs to DLSDUs) and DL service indications to ARPM service indications (converting DLSDUs to APDUs).

10.2 Primitives received from the ARPM

The mapping of ARPM primitives to DL service requests is specified in Table 48.

Table 48 – ARPM to DL mapping

ARPM primitive	DL service
ARP-connect M1 type	Establish-master-polled request
ARP-connect M2 type	Establish-master-packed request
ARP-connect S1 type	Establish-slave-polled request
ARP-connect S2 type	Establish-slave-packed request
ARP-disconnect	Release-connection request
ARP-suspend	Suspend-connection request
ARP-resume	Resume-connection request
ARP-trigger	Master-transmission-trigger request
ARP-activate standby	Activate-standby-master request
ARP-data update	Cyclic-data-update request
ARP-message update	Acyclic-data-send request

10.3 Indications received from the DL

The mapping of DL service indications to ARPM indications is specified in Table 49.

Table 49 – DL to ARPM mapping

DL service indication	ARPM indication
Cyclic-data-update indication	ARP-cyclic-received
Acyclic-data-received indication	ARP-acyclic-received
Error indication	ARP-error

Bibliography

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

IEC/TR 61158-1:2010[2], *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-18:2007, *Industrial communication networks – Fieldbus specifications – Part 3-18: Data-link layer service definition – Type 18 elements*

IEC 61158-4-18:2010[2], *Industrial communication networks – Fieldbus specifications – Part 4-18: Data-link layer protocol specification – Type 18 elements*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

____

2 To be published.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch