

DIN EN 61158-6-10

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-6-10:2008-09
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 6-10: Protokollspezifikation des Application Layer
(Anwendungsschicht) – Typ 10-Elemente
(IEC 61158-6-10:2010);
Englische Fassung EN 61158-6-10:2012, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 6-10: Application layer protocol specification – Type 10 elements
(IEC 61158-6-10:2010);
English version EN 61158-6-10:2012, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 6-10: Spécification des protocoles des couches d'application – Eléments de type 10
(CEI 61158-6-10:2010);
Version anglaise EN 61158-6-10:2012, seulement en CD-ROM

Gesamtumfang 731 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2012-03-28 angenommene Europäische Norm als DIN-Norm ist 2013-05-01.

Für DIN EN 61158-6-10:2008-09 gilt eine Übergangsfrist bis zum 2015-03-28.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN IEC 61158-600:2009-06.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65C „Industrial networks“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die englische Fassung der EN 61158-6-10 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt das Protokoll des Application Layer für Typ 10 fest.

Erklärung zu Schutzrechten

Die Internationale Elektrotechnische Kommission macht darauf aufmerksam, dass ihr gegenüber erklärt wurde, dass die Übereinstimmung mit dem vorgesehenen Norm-Inhalt die Anwendung von Patenten erforderlich machen könnte. Nähere Informationen hierzu befinden sich im Vorwort der Internationalen Norm.

Es wird darauf aufmerksam gemacht, dass Bestandteile der vorliegenden Norm Gegenstand von weiteren Patentrechten sein können. Es liegt nicht im Verantwortungsbereich der DKE, einige oder alle dieser Patentrechte zu benennen.

Der Leser der vorliegenden Norm wird jedoch gebeten, ihm bekannte derartige Patentrechte der Geschäftsstelle mitzuteilen und entsprechende Informationen zur Verfügung zu stellen.

Änderungen

Gegenüber DIN EN 61158-6-10:2008-09 wurden folgende Änderungen vorgenommen:

- a) Berichtigungen;
- b) Verbesserungen;
- c) Verbesserung der Synchronisation;
- d) Verbesserung der Hochlaufzeit nach Spannungswiederkehr.

Frühere Ausgaben

DIN V 19245-1: 1988-01
 DIN EN 61158-6: 2005-05
 DIN EN 61158-6-10: 2008-09

Nationaler Anhang NA (informativ)

3.2 Zusätzliche Begriffe zur verteilten Automatisierung

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.2.1

ACCO

(en: **active connection control object**)

Instanz einer bestimmten FAL-Klasse, die die Verbindungseinrichtung eines Automatisierungsgerätes (als Konsument und Provider) abstrahiert

3.2.2

Konfigurationsdatenbank

(en: **configuration data base**)

Verbindungsinformationen, die von dem ACCO-ASE verwaltet werden

3.2.3

Verbindung

(en: **connection**)

logischer Verbindungszweig zwischen Senke und Quelle von Attributen und Diensten an unterschiedlichen benutzerdefinierten Schnittstellen von benutzerdefinierten RT-Auto-Objekten

3.2.4

Verbindungskanal

(en: **connection channel**)

Beschreibung einer Verbindung zwischen einer Senke und einer Quelle von Datenelementen

3.2.5

Konsument

(en: **consumer**)

Knoten oder Senke, der/die Daten von einem Produzenten empfängt

3.2.6

Konsumenten-ID

(en: **consumerID**)

eindeutige Kennung innerhalb des Anwendungsbereiches des ACCO, die vom Konsumenten zugewiesen wird, um die internen Daten einer konfigurierten Verbindungsenke zu erkennen

3.2.7

Daten-Marshalling

(en: **data marshaling**)

Codierung von Parametern der FAL-Dienstelemente in Bezug auf deren Schnittstellendefinition

ANMERKUNG Daten-Marshalling ist Teil des abstrakten ORPC-Modells.

3.2.8

Engineering

(en: **engineering**)

abstrakte Bezeichnung, welche die Client-Anwendung oder das Gerät kennzeichnet, die/das für die Konfiguration eines Automatisierungssystems durch Verbinden von Datenelementen verantwortlich ist

3.2.9

Ereignis

(en: **event**)

Instanz einer Änderung von Zuständen

3.2.10

Schnittstelle

(en: **interface**)

Zusammenstellung von FAL-Klassenattributen und -diensten, die eine spezifische Sicht auf die FAL-Klasse darstellt

3.2.11

Schnittstellendefinitionssprache

(en: **interface definition language**)

Syntax und Semantik zur formalen Beschreibung von Dienstparametern

ANMERKUNG Diese Beschreibung ist die Eingabe für das ORPC-Modell, insbesondere für das ORPC-Netzwerkprotokoll.

3.2.12

Schnittstellenzeiger

(en: **interface pointer**)

Schlüsselattribut, das eine Objektschnittstelleninstanz eindeutig adressiert

3.2.13

logisches Gerät

(en: **logical device**)

eine bestimmte FAL-Klasse, die eine Softwarekomponente oder eine Firmwarekomponente als eigenständige Einrichtung eines Automatisierungsgerätes abstrahiert

3.2.14

Methode

(en: **method**)

<Objekt> Synonym für einen funktionsfähigen Dienst, der von dem Server-ASE bereitgestellt und von einem Client aufgerufen wird

3.2.15

ORPC

(en: **object remote procedure call**)

Modell für den objektorientierten oder komponentenbasierten Aufruf von entfernten Methoden

3.2.16**physikalisches Gerät****(en: physical device)**

bestimmte FAL-Klasse, die die Hardware-Einrichtungen eines Automatisierungsgerätes abstrahiert

3.2.17**Eigenschaft****(en: property)**

Synonym für ASE-Attribute, die mithilfe von funktionsfähigen ASE-Diensten gelesen oder geschrieben werden können

ANMERKUNG Diese Dienste werden im Allgemeinen mit „get_<Attributname>“ oder „set_<Attributname>“ bezeichnet und entsprechen den IDL-Schlüsselwörtern „propget“ und „propput“.

3.2.18**Provider****(en: provider)**

Quelle einer Datenverbindung

3.2.19**Provider-ID****(en: providerID)**

eindeutige Kennung innerhalb des Anwendungsbereiches des ACCO, die vom Provider zugewiesen wird, um die internen Daten einer konfigurierten Verbindungsquelle zu erkennen

3.2.20**Qualitätscode****(en: quality code)**

zusätzliche Statusinformation eines Datenelementes

3.2.21**Qualitätscode bekannt****(en: quality code aware)**

Attribut der RT-Auto-Klasse, welches angibt, dass ein RT-Auto-Objekt einen Statuscode für seine Datenelemente verwendet

3.2.22**Qualitätscode unbekannt****(en: quality code unaware)**

Gegenteil von „Qualitätscode bekannt“

3.2.23**RT-Auto****(en: RT-Auto)**

FAL-Klasse, die die Automatisierungsfunktion als prozessbezogene Komponente eines Automatisierungsgerätes abstrahiert

3.2.24**Laufzeitobjektmodell****(en: runtime object model)**

Objekte, die in einem Gerät zusammen mit deren Schnittstellen und Methoden vorhanden sind, auf welche zugegriffen werden kann

3.3 Zusätzliche Begriffe zur dezentralisierten Peripherie

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.3.1

Alarm

(en: alarm)

Aktivierung eines Ereignisses, das einen kritischen Zustand aufweist

3.3.2

Alarmbestätigung

(en: alarm ack)

Bestätigung eines Ereignisses, das einen kritischen Zustand aufweist

3.3.3

Alarmdatenobjekt

(en: alarm data object)

Objekte, die kritische Zustände repräsentieren, welche als Geräte/Slot/Subslot/Alarmtyp referenziert werden

3.3.4

zuweisen

(en: allocate)

eine Ressource aus einem allgemeinen Bereich für die ausschließliche Nutzung durch eine spezifische Einheit zuweisen

3.3.5

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten verarbeitet oder erzeugt werden

3.3.6

Interoperabilität auf Anwendungsschicht

(en: application layer interoperability)

Fähigkeit von Anwendungseinheiten, mithilfe der Dienste der FAL koordinierte und kooperative Operationen auszuführen

3.3.7

Anwendungsobjekte

(en: application objects)

mehrere Objektklassen, die einen Austausch von PDUs über das Netzwerk und innerhalb des Netzwerkgerätes zur Laufzeit verwalten und bereitstellen

3.3.8

Anwendungsprozess

(en: application process)

Teil einer verteilten Anwendung in einem Netzwerk, der sich auf einem Gerät befindet und eindeutig adressiert wird

3.3.9

Anwendungsprozesskennung

(en: application process identifier)

unterscheidet mehrere in einem Gerät verwendete Anwendungsprozesse

3.3.10

Anwendungsprozessobjekt

(en: application process object)

Komponente eines Anwendungsprozesses, die über eine FAL-Anwendungsbeziehung identifizierbar und zugänglich ist

ANMERKUNG Definitionen von Anwendungsprozessobjekten bestehen aus einem Satz von Werten für die Attribute ihrer Klasse (siehe Definition von „Klasse von Anwendungsprozessobjekten“). Auf die Definitionen von Anwendungsprozessobjekten kann mithilfe der Dienste des Object-Management-ASE der FAL von einem entfernten Standort aus zugegriffen werden. Object-Management-Dienste der FAL können zum Laden und Aktualisieren von Objektdefinitionen,

zum Lesen von Objektdefinitionen und zur dynamischen Erstellung und Löschung von Anwendungsobjekten und deren Definitionen genutzt werden.

3.3.11

Klasse von Anwendungsprozessobjekten

(en: application process object class)

Klasse von Anwendungsprozessobjekten, die in Bezug auf den Satz der über das Netzwerk zugänglichen Attribute und Dienste der Anwendungsprozessobjekte definiert ist

3.3.12

Anwendungsbeziehung

(en: application relationship)

kooperative Verbindung zwischen zwei oder mehr Aufrufen von Anwendungseinheiten zum Zweck des Informationsaustausches und der Koordination des gemeinsamen Betriebs. Diese Beziehung wird entweder durch den Austausch von Protokolldateneinheiten der Anwendung oder als Ergebnis von Vorkonfigurationsaktivitäten aktiviert

3.3.13

Anwendungsdienstelement für Anwendungsbeziehungen

(en: application relationship application service element)

Anwendungsdienstelement, das die alleinigen Mittel für den Aufbau und die Beendigung aller Anwendungsbeziehungen bereitstellt

3.3.14

Endpunkt der Anwendungsbeziehung

(en: application relationship endpoint)

Kontext und Verhalten einer Anwendungsbeziehung entsprechend der Sicht und Verwaltung durch einen der in die Anwendungsbeziehung eingeschlossenen Anwendungsprozesse

ANMERKUNG Jeder Anwendungsprozess, der in die Anwendungsbeziehung eingeschlossen ist, verwaltet seinen eigenen Endpunkt der Anwendungsbeziehung.

3.3.15

Attribut

(en: attribute)

Beschreibung eines von außen sichtbaren Merkmals oder einer von außen sichtbaren Eigenschaft eines Objekts

ANMERKUNG Die Attribute eines Objekts enthalten Informationen zu veränderlichen Teilen eines Objekts. Üblicherweise stellen sie Statusinformationen bereit oder steuern den Betrieb eines Objekts. Attribute können auch das Verhalten eines Objekts beeinflussen. Attribute werden in Klassenattribute und Instanzattribute unterteilt.

3.3.16

backup

(en: backup)

Status der E/A-AR, der angibt, dass sich diese in Bereitschaft befindet

3.3.17

Verhalten

(en: behavior)

Angabe, wie ein Objekt auf bestimmte Ereignisse reagiert

3.3.18

Kanal

(en: channel)

Darstellung eines einzelnen physikalischen oder logischen Verbindungszweiges eines Eingabe- oder Ausgabe-Anwendungsobjektes eines Servers zum Prozess, um die Adressierung von Diagnose-Informationen zu unterstützen

ANMERKUNG Der Kanal stellt typischerweise einen einzelnen Stecker oder eine einzelne Klemme als reale Schnittstelle eines Moduls oder Untermoduls dar. Diese Referenz wird verwendet, um Ausfallpunkte innerhalb von Diagnose-PDUs zu identifizieren.

3.3.19

kanalbezogene Diagnose

(en: channel related diagnosis)

Informationen in Bezug auf ein spezifisches Element eines Eingabe- oder Ausgabe-Anwendungsobjektes, die zu Wartungszwecken bereitgestellt werden

BEISPIEL Offene Schleife.

3.3.20

Klasse

(en: class)

Satz von Objekten, die allesamt den gleichen Typ von Systemkomponente repräsentieren

ANMERKUNG Eine Klasse ist eine Verallgemeinerung eines Objektes; eine Vorlage für die Festlegung von Variablen und Methoden. Alle Objekte in einer Klasse sind in ihrer Form und in ihrem Verhalten identisch, enthalten aber üblicherweise unterschiedliche Daten in ihren Attributen.

3.3.21

Klassenattribute

(en: class attributes)

Attribut, das von allen Objekten innerhalb derselben Klasse gemeinsam genutzt wird

3.3.22

Klassencode

(en: class code)

eindeutige Kennung, die jeder Objektklasse zugewiesen ist

3.3.23

klassenspezifischer Dienst

(en: class specific service)

Dienst, der durch eine bestimmte Objektklasse definiert ist, eine erforderliche Funktion, die nicht von einem allgemeinen Dienst ausgeführt wird, auszuführen

ANMERKUNG Ein klassenspezifisches Objekt ist für die Objektklasse, die es definiert, eindeutig.

3.3.24

clear

(en: clear)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit nicht ausgeführt wird

3.3.25

Client

(en: client)

- a) Objekt, das die Dienste eines anderen Objekts (Servers) nutzt, um eine Aufgabe auszuführen
- b) Initiator einer PDU, auf die ein Server reagiert

3.3.26

gemeinsames Profil

(en: common profile)

Zusammenstellung geräteunabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen allen Geräten herzustellen

3.3.27**Kommunikationsdatenobjekt****(en: communication data object)**

Objekte, die Parameter von Kommunikationsbeziehungen sind und durch Gerät/Slot/Subslot/Index referenziert werden

3.3.28**Konfigurationsprüfung****(en: configuration check)**

Vergleich der erwarteten Strukturierung des E/A-Datenobjekts des Clients mit der tatsächlichen Strukturierung des E/A-Datenobjekts beim Server in der Startphase

3.3.29**Konfigurationsfehler****(en: configuration fault)**

unzulässiger Unterschied zwischen der erwarteten Strukturierung des E/A-Datenobjekts und der tatsächlichen Strukturierung des E/A-Datenobjekts, erkannt durch den Server

3.3.30**Konfigurationskennung****(en: configuration identifier)**

Darstellung eines Teils von E/A-Daten eines einzelnen Eingabe- und/oder Ausgabemoduls eines Servers

3.3.31**konsumieren****(en: consume)**

Vorgang des Empfangs von Daten von einem Provider

3.3.32**Konsument****(en: consumer)**

Knoten oder Senke, der/die Daten von einem Provider empfängt

3.3.33**Kontextmanagement****(en: context management)**

über das Netzwerk zugängliche Informationen (Kommunikationsobjekte), die die Verwaltung des Betriebs des Feldbussystems, einschließlich der Anwendungsschicht, unterstützen

ANMERKUNG Zur Verwaltung gehören Funktionen wie die Steuerung, die Überwachung und die Diagnose.

3.3.34**Übertragungspfad****(en: conveyance path)**

unidirektionaler Strom von APDUs über eine Anwendungsbeziehung

3.3.35**zyklisch****(en: cyclic)**

sich regelmäßig wiederholend

3.3.36**Datenkonsistenz****(en: data consistency)**

Mittel für die kohärente Übertragung des Eingabe- oder Ausgabedatenobjekts zwischen Client und Server und für den kohärenten Zugriff auf das Eingabe- oder Ausgabedatenobjekt innerhalb von Client und Server

3.3.37

Gerät

(en: device)

physische Hardware, die mit dem Verbindungszweig verbunden ist

ANMERKUNG Ein Gerät kann mehr als einen Knoten umfassen.

3.3.38

Geräte-ID

(en: device ID)

vom Anbieter zugewiesene Kennung des Gerätetyps

3.3.39

Geräteprofil

(en: device profile)

Zusammenstellung geräteabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps herzustellen

3.3.40

Diagnosedatenobjekt

(en: diagnosis data object)

Objekt, das Diagnose-Informationen enthält, die durch Gerät/Slot/Subslot/Index referenziert werden

3.3.41

Diagnose-Informationen

(en: diagnosis information)

alle Daten, die serverseitig für Wartungszwecke verfügbar sind

3.3.42

dynamische Neukonfiguration

(en: dynamic reconfiguration)

Änderung von E/A-Datenobjekten ohne Unterbrechung einer bestehenden Anwendungsbeziehung und fortlaufende Aktualisierung der nicht geänderten E/A-Datenobjekte

3.3.43

Endpunkt

(en: endpoint)

eine der Kommunikationseinheiten, die zu einer Verbindung gehören

3.3.44

Engineering

(en: engineering)

abstrakte Bezeichnung, welche die Client-Anwendung oder das Gerät kennzeichnet, die/das für die Konfiguration eines Automatisierungssystems verantwortlich ist

3.3.45

Fehler

(en: error)

Nichtübereinstimmung zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem festgelegten oder theoretisch richtigen Wert oder Zustand

3.3.46

Fehlerklasse

(en: error class)

allgemeine Gruppierung für zugehörige Fehlerdefinitionen und entsprechende Fehlercodes

3.3.47

Fehlercode

(en: error code)

Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

3.3.48**Ereignis****(en: event)**

Instanz einer Änderung von Zuständen

3.3.49**erweiterte kanalbezogene Diagnose****(en: extended channel related diagnosis)**

Informationen in Bezug auf ein spezifisches Element eines spezifischen Anwendungsobjektes, die zu Wartungszwecken bereitgestellt werden

BEISPIEL Ausfall des Verbindungszweiges.

3.3.50**Rahmen****(en: frame)**

Einheit des Datentransfers auf einem Verbindungsweig

3.3.51**Identifikationsdatenobjekt****(en: identification data object)**

Objekte, die Informationen zum Hersteller und Typ des Gerätes, Moduls und Submoduls enthalten, welche durch Gerät/Slot/Subslot/Index referenziert werden

3.3.52**impliziter AR-Endpunkt****(en: implicit AR endpoint)**

AR-Endpunkt, der lokal innerhalb eines Gerätes ohne Nutzung des Erstellungsdienstes definiert ist

3.3.53**Index****(en: index)**

Adresse eines Datensatzobjektes innerhalb eines Anwendungsprozesses

3.3.54**Instanz****(en: instance)**

tatsächliches physikalisches Vorkommen eines Objektes innerhalb einer Klasse, das eines von mehreren Objekten derselben Objektklasse identifiziert

3.3.55**Instanzattribut****(en: instance attributes)**

Attribut, das für eine Objektinstanz eindeutig ist und von der Objektklasse nicht gemeinsam genutzt wird

3.3.56**instanziiert****(en: instantiated)**

Objekt, das in einem Gerät erzeugt wurde

3.3.57**Aufruf****(en: invocation)**

Vorgang der Nutzung eines Dienstes oder einer anderen Ressource eines Anwendungsprozesses

ANMERKUNG Jeder Aufruf stellt einen separaten Kontrollthread dar, der durch seinen Kontext beschrieben werden kann. Sobald der Dienst ausgeführt oder die Nutzung der Ressource beendet ist, erlischt der Aufruf. Bei Dienstaufrufen wird ein Dienst, der initiiert, aber noch nicht ausgeführt wurde, als ausstehender Dienstaufruf bezeichnet. Bei Dienstaufrufen kann außerdem eine Aufruf-ID verwendet werden, um den Dienstaufruf eindeutig zu identifizieren und ihn von anderen ausstehenden Dienstaufrufen zu unterscheiden.

3.3.58

E/A-Controller

(en: **IO controller**)

Steuergerät, das für mehrere E/A-Geräte (Feldgeräte) als Client agiert

ANMERKUNG Hierbei handelt es sich üblicherweise um eine speicherprogrammierbare Steuerung oder ein verteiltes Steuerungssystem.

3.3.59

E/A-Datenobjekt

(en: **IO data object**)

Objekt, das zu Verarbeitungszwecken zyklisch übertragen wird und das durch Gerät/Slot/Subslot referenziert wird

3.3.60

E/A-Gerät

(en: **IO device**)

Feldgerät, das für den E/A-Betrieb als Server agiert

3.3.61

E/A-Parameterserver

(en: **IO parameter server**)

Server für Anwendungsparameter von E/A-Geräten (Client)

ANMERKUNG Hierbei handelt es sich üblicherweise um ein Gerät zur Sicherung von Parameterdaten und zur Protokollierung von Online-Änderungen von Geräteparametern.

3.3.62

E/A-Subsystem

(en: **IO subsystem**)

Subsystem, das einen E/A-Controller und alle zugehörigen E/A-Geräte umfasst

3.3.63

E/A-Supervisor

(en: **IO supervisor**)

Engineering-Gerät, das die Inbetriebnahme und Diagnose eines E/A-Systems verwaltet

3.3.64

E/A-System

(en: **IO system**)

System, das aus allen seinen E/A-Subsystemen besteht

ANMERKUNG Eine SPS mit mehr als einem E/A-Controller (Netzwerkschnittstelle) steuert beispielsweise ein E/A-System, das aus einem E/A-Subsystem für jeden E/A-Controller besteht.

3.3.65

isochroner Modus

(en: **Isochronous mode**)

E/A-System, das eng getaktet mit einem Jitter von weniger als 1 µs ausgeführt wird

3.3.66

Member

(en: **member**)

Teil eines Attributes, das als Element eines Feldes strukturiert ist

3.3.67

Nachricht

(en: **message**)

Synonym für Rahmen

3.3.68**Methode****(en: method)**

Synonym für einen funktionsfähigen Dienst, der von dem Server-ASE bereitgestellt und von einem Client aufgerufen wird

3.3.69**Modul****(en: module)**

Hardware oder logische Komponente eines physischen Gerätes

3.3.70**Netzwerk****(en: network)**

Menge von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind, einschließlich zwischengeschalteter Repeater, Bridges, Router und Gateways unterer Schichten

3.3.71**Objekt****(en: object)**

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes, normalerweise eine Zusammenstellung zugehöriger Daten (in Form von Variablen) und Methoden (Prozeduren) für Operationen in Bezug auf diese Daten mit klar definierter Schnittstelle und klar definiertem Verhalten

3.3.72**objektspezifischer Dienst****(en: object specific service)**

Dienst, der für die Objektklasse, die ihn definiert, eindeutig ist

3.3.73**operate****(en: operate)**

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit ausgeführt wird

3.3.74**Paket****(en: packet)**

Rahmen

3.3.75**Peer****(en: peer)**

Rolle eines AR-Endpunktes, in der dieser in der Lage ist, sowohl als Client als auch als Server zu agieren

3.3.76**physisches Gerät****(en: physical device)**

Automatisierungsgerät oder anderes Netzwerkgerät

3.3.77**Punkt-zu-Punkt-Verbindung****(en: point-to-point connection)**

Verbindung, die zwischen genau zwei Anwendungsobjekten besteht

3.3.78**primary****(en: primary)**

Status der E/A-AR, der angibt, dass sich diese im Betriebszustand befindet

ANMERKUNG Neben der primären E/A-AR kann auch eine E/A-AR zur Sicherung vorhanden sein. Zum Beispiel für Redundanz und dynamische Neukonfiguration von E/A-Daten verwendet.

3.3.79

Provider

(en: provider)

Knoten oder Quelle, der/die Daten an einen oder viele Konsumenten sendet

3.3.80

PTCP-Unterlandäne

(en: PTCP subdomain)

bestimmte Menge von DTEs mit synchronisierten Uhren

3.3.81

Datensatzobjekt

(en: record data object)

Objekte, die bereits vorverarbeitet sind und zu Informationszwecken oder zum Zweck der weiteren Verarbeitung azyklisch übertragen werden und die durch Gerät/Slot/Subslot/Index referenziert werden

3.3.82

Ressource

(en: resource)

Verarbeitungs- oder Informationsfähigkeit

3.3.83

run

(en: run)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit in Betrieb ist

3.3.84

Server

(en: server)

- a) Rolle eines AREP, in der dieser eine bestätigte Dienstantwort-APDU an den Client, der die Anforderung initiiert hat, zurückgibt
- b) Objekt, das einem anderen Objekt (Client) Dienste bereitstellt

3.3.85

Dienst

(en: service)

Operation oder Funktion, die ein Objekt und/oder eine Objektklasse auf Aufforderung eines anderen Objekts und/oder einer anderen Objektklasse ausführt

3.3.86

Slot

(en: slot)

Adresse einer strukturellen Einheit innerhalb eines E/A-Gerätes

ANMERKUNG Innerhalb eines modularen Gerätes adressiert ein Slot typischerweise ein physisches Modul. Innerhalb kompakter Geräte adressiert ein Slot typischerweise eine logische Funktion oder ein virtuelles Modul.

3.3.87

stop

(en: stop)

Status des E/A-Controllers, der angibt, dass der Steuerungsalgorithmus derzeit nicht ausgeführt wird

3.3.88

Untermodule

(en: submodule)

Hardware oder logische Komponente eines Moduls

3.3.89

Subslot

(en: subslot)

Adresse einer strukturellen Einheit innerhalb eines Slots

ANMERKUNG Ein Subslot kann eine physische Schnittstelle für Untermodule innerhalb eines Moduls adressieren. Im Allgemeinen stellt ein Subslot eine zweite Ebene für die Strukturierung von Daten innerhalb eines Gerätes dar.

3.3.90

Anbieter-ID

(en: vendor ID)

zentrale Verwaltungsnummer, die als Herstellerkennung verwendet wird

– Leerseite –

**Industrial communication networks -
Fieldbus specifications -
Part 6-10: Application layer protocol specification -
Type 10 elements
(IEC 61158-6-10:2010)**

Réseaux de communication industriels -
Spécifications des bus de terrain -
Partie 6-10: Spécification des protocoles
des couches d'application -
Éléments de type 10
(CEI 61158-6-10:2010)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 6-10: Protokollspezifikation des
Application Layer (Anwendungsschicht) -
Typ 10-Elemente
(IEC 61158-6-10:2010)

This European Standard was approved by CENELEC on 2012-03-28. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65C/607/FDIS, future edition 2 of IEC 61158-6-10, prepared by SC 65C, "Industrial networks", of IEC/TC 65, "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-6-10:2012.

The following dates are fixed:

... latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2012-12-28
... latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2015-03-28

This document supersedes EN 61158-6-10:2008.

EN 61158-6-10:2012 includes the following significant technical changes with respect to EN 61158-6-10:2008:

- corrections;
- improvements;
- optimization of the synchronization;
- optimization of the startup time from power down.

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

Endorsement notice

The text of the International Standard IEC 61158-6-10:2010 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC/TR 61158-1	NOTE Harmonized as CLC/TR 61158-1.
IEC 61158-6-3:2010	NOTE Harmonized as EN 61158-6-3:2012.

CONTENTS

INTRODUCTION	21
1 Scope	23
1.1 General	23
1.2 Specifications	23
1.3 Conformance	23
2 Normative references	24
3 Terms, definitions, abbreviations, symbols and conventions	26
3.1 Referenced terms and definitions	26
3.2 Additional terms and definitions for distributed automation	26
3.3 Additional terms and definitions for decentralized periphery	28
3.4 Additional abbreviations and symbols for distributed automation	36
3.5 Additional abbreviations and symbols for decentralized periphery	36
3.6 Additional abbreviations and symbols for media redundancy	37
3.7 Conventions	37
3.8 Conventions used in state machines	44
4 Application layer protocol specification for common protocols	46
4.1 FAL syntax description	46
4.2 Transfer syntax	49
4.3 Discovery and basic configuration	60
4.4 Precision time control	88
4.5 Media redundancy	149
4.6 Real-time cyclic	172
4.7 Real-time acyclic	189
4.8 Remote procedure call	201
4.9 Link layer discovery	216
4.10 MAC bridges	224
4.11 Virtual bridges	238
4.12 IP suite	247
4.13 Domain name system	250
4.14 Dynamic host configuration	250
4.15 Simple network management	250
4.16 Common DLL Mapping Protocol Machines	260
5 Application layer protocol specification for distributed automation	266
5.1 FAL syntax description	266
5.2 Transfer syntax	289
5.3 FAL protocol state machines	291
5.4 AP Context State Machine	292
5.5 FAL Service Protocol Machines	292
5.6 Application Relationship Protocol Machine (ARPM)	386
5.7 DLL Mapping Protocol Machines	390
5.8 Protocol options	394
6 Application layer protocol specification for decentralized periphery	394
6.1 FAL syntax description	394
6.2 Transfer syntax	404

6.3	FAL protocol state machines	526
6.4	AP-Context state machine	528
6.5	FAL Service Protocol Machines	528
6.6	Application Relationship Protocol Machines	609
6.7	DLL Mapping Protocol Machines	683
Annex A (informative)	Filtering Data Base	683
Annex B (informative)	Establishing of a companion AR	686
Annex C (informative)	Establishing of a device access AR	687
Annex D (informative)	Establishing of an AR (simple procedure)	688
Annex E (informative)	Establishing of an AR (accelerated procedure)	689
Annex F (informative)	Establishing of an AR (fast startup procedure)	691
Annex G (informative)	Example of the upload, storage, and retrieval procedure	692
Annex H (informative)	Establishing of an AR using RT_CLASS_3 CRs	694
Annex I (informative)	Overview of the AlarmCRs	695
Annex J (informative)	OSI reference model layers	697
Annex K (informative)	Overview of the IO device state machines	698
Annex L (informative)	Overview of the IO controller state machines	699
Annex M (informative)	Priority regeneration	700
Annex N (informative)	Overview of the synchronization master hierarchy	701
Annex O (informative)	Adapted baggy pants model of the IEEE 802.1D	703
Annex P (informative)	Optimization of bandwidth usage	706
Annex Q (informative)	Time constraints for bandwidth allocation	708
Annex R (informative)	Optimization for fast startup without autonegotiation	710
Annex S (informative)	TX-error handling	712
Bibliography		713
Annex ZA (normative)	Normative references to international publications with their corresponding European publications	714
Figure 1	– Common structure of specific fields	41
Figure 2	– Common structure of specific fields for octet 1 (high)	42
Figure 3	– Common structure of specific fields for octet 2 (low)	42
Figure 4	– Common structure of specific fields for octet 1 (high)	43
Figure 5	– Common structure of specific fields for octet 2	43
Figure 6	– Common structure of specific fields for octet 3	43
Figure 7	– Common structure of specific fields for octet 4 (low)	44
Figure 8	– Coding of the data type BinaryDate	51
Figure 9	– Encoding of Time Of Day value	51
Figure 10	– Encoding of Time Difference value	52
Figure 11	– Encoding of Network Time value	52
Figure 12	– Encoding of Network Time Difference value	53
Figure 13	– Timescale correspondence between PTCP_Time and CycleCounter	94
Figure 14	– Message timestamp point	99
Figure 15	– Four message timestamps	100
Figure 16	– Line delay protocol with follow up	101
Figure 17	– Line delay protocol without follow up	101

Figure 18 – Line delay measurement.....	103
Figure 19 – Model parameter for GSDML usage	104
Figure 20 – Bridge delay measurement	105
Figure 21 – Delay accumulation.....	105
Figure 22 – Worst case accumulated time deviation of synchronization	106
Figure 23 – Scheme for measurement of deviation	107
Figure 24 – Measurement of deviation	107
Figure 25 – Sending Sync-Frame without Follow Up-Frame	108
Figure 26 – Sending Sync- Frame with FollowUp-Frame	108
Figure 27 – Forwarding Sync- and FollowUp-Frame	109
Figure 28 – Transition between Synchronization Variants.....	110
Figure 29 – State transition diagram of delay request	111
Figure 30 – State transition diagram of delay response.....	117
Figure 31 – Overview of PTCP.....	121
Figure 32 – State transition diagram of BMA	122
Figure 33 – State transition diagram of MPSM	132
Figure 34 – State transition diagram of SPSM.....	138
Figure 35 – State transition diagram of SRPM.....	143
Figure 36 – State transition diagram of SCHEDULER.....	148
Figure 37 – MRM protocol machine for MRP	153
Figure 38 – MRC protocol machine.....	160
Figure 39 – State transition diagram of RRT_RELAY	170
Figure 40 – CycleCounter value range.....	174
Figure 41 – Structure of the CycleCounter.....	175
Figure 42 – Optimized CycleCounter setting.....	175
Figure 43 – State transition diagram for generating events	231
Figure 44 – State transition diagram of RED RELAY	235
Figure 45 – State transition diagram of MUX	240
Figure 46 – State transition diagram of DEMUX	244
Figure 47 – Structuring of the protocol machines within the DMPM (bridge).....	261
Figure 48 – Error message structure.....	266
Figure 49 – Coding scheme of ITEMQUALITYDEF	270
Figure 50 – Relationship among protocol machines	292
Figure 51 – State transition diagram of FSPM	305
Figure 52 – State transition diagram of ARPM.....	388
Figure 53 – State transition diagram of DMPM	393
Figure 54 – FrameSendOffset vs. duration of a cycle	439
Figure 55 – Classification of diagnosis, maintenance and qualified.....	470
Figure 56 – Definition of the reserved interval.....	487
Figure 57 – Definition of PLL window	490
Figure 58 – Detection of dropped frames — appear	498
Figure 59 – Detection of dropped frames — disappear	498
Figure 60 – Relationship among Protocol Machines.....	527

Figure B.1 – Establishing of a companion AR	686
Figure C.1 – Establishing of a device access AR.....	687
Figure D.1 – Accelerated establishing of an IOAR (simple procedure).....	688
Figure E.1 – Accelerated establishing of an IOAR without error	689
Figure E.2 – Accelerated establishing of an IOAR with “late” error.....	690
Figure F.1 – Establishing of an IOAR using fast startup.....	691
Figure G.1 – Example of upload with storage	692
Figure G.2 – Example of retrieval with storage	693
Figure H.1 – Establishing of an AR using RT_CLASS_3 CRs	694
Figure I.1 – Overview of an AlarmCR using high priority	695
Figure I.2 – Overview of an AlarmCR using low priority	696
Figure J.1 – Assignment of the OSI reference model layers.....	697
Figure K.1 – Overview of the IO device state machines.....	698
Figure L.1 – Overview of the IO controller state machines	699
Figure N.1 – Level model for synchronization master hierarchy	701
Figure N.2 – Two level variant of the synchronization master hierarchy.....	702
Figure O.1 – Adapted baggy pants model of the IEEE 802.1D	703
Figure O.2 – Adapted baggy pants model of the IEEE 802.1D for a frame transmitter	704
Figure O.3 – Adapted baggy pants model of the IEEE 802.1D for a frame receiver	705
Figure P.1 – Devices build up in a linear structure	706
Figure P.2 – Propagation of frames in linear transmit direction	706
Figure P.3 – Propagation of a frames in receive direction.....	707
Figure Q.1 – Overview of time constraints for bandwidth allocation	708
Figure Q.2 – Caculation of the length of a period.....	708
Figure R.1 – Scheme of a 2-port switch	710
Figure R.2 – Scheme of 2-ports.....	710
 Table 1 – State machine description elements.....	 44
Table 2 – Description of state machine elements.....	45
Table 3 – Conventions used in state machines.....	45
Table 4 – IEEE 802.3 DLPDU syntax.....	47
Table 5 – IEEE 802.11 DLPDU syntax	47
Table 6 – IEEE 802.15.1 DLPDU syntax.....	48
Table 7 – SourceAddress	53
Table 8 – DCP_MulticastMACAdd.....	54
Table 9 – PTCP_MulticastMACAdd range 1	54
Table 10 – PTCP_MulticastMACAdd range 2	54
Table 11 – PTCP_MulticastMACAdd range 3	55
Table 12 – PTCP_MulticastMACAdd range 4	55
Table 13 – PTCP_MulticastMACAdd range 5	55
Table 14 – PTCP_MulticastMACAdd range 6	55
Table 15 – PTCP_MulticastMACAdd range 7	56
Table 16 – PTCP_MulticastMACAdd range 8	56

Table 17 – MRP Organizationally Unique Identifier (OUI)	56
Table 18 – MRPMulticastMACAdd	56
Table 19 – LT (Length/Type)	57
Table 20 – TagControlInformation.Priority	57
Table 21 – FrameID range 1	58
Table 22 – FrameID range 2	58
Table 23 – FrameID range 3	58
Table 24 – FrameID range 4	58
Table 25 – FrameID range 5	59
Table 26 – FrameID range 6	59
Table 27 – FrameID range 7	59
Table 28 – FrameID range 8	59
Table 29 – FrameID range 9	59
Table 30 – FrameID range 10	60
Table 31 – FrameID range 11	60
Table 32 – FrameID range 12	60
Table 33 – FrameID range 13	60
Table 34 – DCP APDU syntax	61
Table 35 – DCP substitutions	61
Table 36 – ServiceID	63
Table 37 – ServiceType for request	64
Table 38 – ServiceType for response	64
Table 39 – ResponseDelayFactor	64
Table 40 – List of options	65
Table 41 – List of suboptions for option IPOption	65
Table 42 – List of suboptions for option DevicePropertiesOption	66
Table 43 – List of suboptions for option DHCPOption	66
Table 44 – List of suboptions for option ControlOption	66
Table 45 – List of suboptions for option DeviceInitiativeOption	66
Table 46 – List of suboptions for option AllSelectorOption	66
Table 47 – List of suboptions for option ManufacturerSpecificOption	66
Table 48 – SuboptionDHCP	68
Table 49 – Coding of DCPBlocklength in conjunction with SuboptionStart	68
Table 50 – Coding of DCPBlocklength in conjunction with SuboptionStop	69
Table 51 – Coding of DCPBlocklength in conjunction with SuboptionSignal	69
Table 52 – Coding of DCPBlocklength in conjunction with SuboptionFactoryReset	69
Table 53 – Coding of DCPBlocklength in conjunction with SuboptionDeviceInitiative	70
Table 54 – BlockQualifier with option IP	71
Table 55 – BlockQualifier with option DeviceProperties, DHCP, and ManufacturerSpecific	71
Table 56 – BlockError	71
Table 57 – BlockInfo for SuboptionIPParameter	72
Table 58 – Bit 1 and Bit 0 of BlockInfo for SuboptionIPParameter	72
Table 59 – Bit 7 of BlockInfo for SuboptionIPParameter	72

Table 60 – BlockInfo for all other suboptions	72
Table 61 – DeviceInitiativeValue	72
Table 62 – SignalValue.....	73
Table 63 – DeviceRoleDetails	74
Table 64 – IPAddress	75
Table 65 – Subnetmask.....	75
Table 66 – StandardGateway	76
Table 67 – DCPUCS state table	79
Table 68 – DCPUCR state table.....	82
Table 69 – DCPMCS state table.....	84
Table 70 – DCPMCR state table	85
Table 71 – DCPHMCS state table	87
Table 72 – DCPHMCR state table.....	88
Table 73 – PTCP APDU syntax.....	89
Table 74 – PTCP substitutions	89
Table 75 – PTCP_TLVHeader.Type.....	90
Table 76 – PTCP_Delay10ns	91
Table 77 – PTCP_Delay1ns_Byte.Value.....	91
Table 78 – PTCP_Delay1ns	91
Table 79 – PTCP_Delay1ns_FUP	92
Table 80 – PTCP_SequenceID	92
Table 81 – PTCP_SubType for OUI (=00-0E-CF).....	92
Table 82 – PTCP_NanoSeconds.....	93
Table 83 – PTCP_Flags.LeapSecond	93
Table 84 – Timescale correspondence between MJD, UTC, and PTCP_EPOCHNumber	93
Table 85 – Timescale correspondence between PTCP_EPOCHNumber, PTCP_Second, PTCP_Nanosecond, CycleCounter, and SendClockFactor	94
Table 86 – PTCP_MasterPriority1.Priority for SyncID == 0 and SyncProperties.Role == 2	95
Table 87 – PTCP_MasterPriority1.Priority for SyncID == 0 and SyncProperties.Role == 1	95
Table 88 – PTCP_MasterPriority1.Priority for SyncID == 1 and SyncProperties.Role == 2	95
Table 89 – PTCP_MasterPriority1.Priority for SyncID == 1 and SyncProperties.Role == 1	96
Table 90 – PTCP_MasterPriority1.Level	96
Table 91 – PTCP_MasterPriority2	96
Table 92 – PTCP_ClockClass for SyncID == 0 (clock synchronization)	96
Table 93 – PTCP_ClockClass for SyncID == 1 (time synchronization).....	97
Table 94 – PTCP_ClockAccuracy	98
Table 95 – PTCP_ClockVariance	98
Table 96 – PTCP_T2TimeStamp.....	99
Table 97 – DelayRequest state table.....	112
Table 98 – Macros used by DelayRequest.....	115

Table 99 – Functions used by DelayRequest.....	116
Table 100 – DelayResponse state table	118
Table 101 – Macros used by DelayResponse.....	119
Table 102 – Functions used by DelayResponse.....	119
Table 103 – BMA state table	123
Table 104 – BMA best remote sync master (RSM) state table	127
Table 105 – BMA get best sync master (GBSM) state table.....	129
Table 106 – Macros used by BMA	130
Table 107 – Functions used by BMA	131
Table 108 – MPSM state table	133
Table 109 – Macros used by MPSM	136
Table 110 – Functions used by MPSM	136
Table 111 – SPSM state table.....	139
Table 112 – Macros used by SPSM.....	142
Table 113 – Functions used by SPSM	142
Table 114 – SRPM state table	144
Table 115 – Macros used by SRPM.....	145
Table 116 – Functions used by the SRPM.....	146
Table 117 – Truth table for one SyncID	147
Table 118 – SCHEDULER state table.....	148
Table 119 – Functions used by the SCHEDULER.....	149
Table 120 – MRP APDU syntax	150
Table 121 – MRP substitutions	150
Table 122 – MRP_TLVHeader.Type	150
Table 123 – MRP_Prio	151
Table 124 – MRP_PortRole.....	151
Table 125 – MRP_RingState.....	152
Table 126 – MRP_Interval.....	152
Table 127 – MRP_Transition.....	152
Table 128 – MRP_TimeStamp	152
Table 129 – MRP_DomainUUID	153
Table 130 – Local variables of MRM protocol machine	154
Table 131 – MRM state machine	155
Table 132 – Local variables of MRC protocol machine.....	161
Table 133 – MRC state machine	161
Table 134 – Functions	166
Table 135 – FDB Clear Timer.....	169
Table 136 – Topology Change Timer.....	169
Table 137 – RRT_RELAY state table.....	170
Table 138 – Macros used by RRT_RELAY	172
Table 139 – RTC APDU syntax.....	173
Table 140 – RTC substitutions	173
Table 141 – CycleCounter Difference	174

Table 142 – DataStatus.State.....	176
Table 143 – DataStatus.Redundancy.....	176
Table 144 – DataStatus.DataValid	176
Table 145 – DataStatus.ProviderState	177
Table 146 – DataStatus.StationProblemIndicator	177
Table 147 – DataStatus.reserved_3.....	177
Table 148 – DataStatus.Ignore of a frame	177
Table 149 – TransferStatus for RT_CLASS_3	177
Table 150 – IOxS.Extension	178
Table 151 – IOCS.Instance	179
Table 152 – IOxS.DataState.....	179
Table 153 – PPM state table	181
Table 154 – Functions used by the PPM.....	184
Table 155 – CPM state table	185
Table 156 – Functions used by the CPM	189
Table 157 – RTA APDU syntax	190
Table 158 – RTA substitutions.....	190
Table 159 – PDUType.Type	191
Table 160 – PDUType.Version	191
Table 161 – VarPartLen.....	192
Table 162 – APMS state table	193
Table 163 – Functions used by the APMS and APMR.....	196
Table 164 – A_Timer_add	197
Table 165 – A_Timer_event	197
Table 166 – A_Timer_remove	197
Table 167 – APMR state table.....	199
Table 168 – RPC APDU syntax.....	201
Table 169 – RPC substitutions	201
Table 170 – RPCPacketType	202
Table 171 – RPCFlags	203
Table 172 – RPCFlags2	203
Table 173 – RPCDRep.Character- and IntegerEncoding	204
Table 174 – RPCDRep Octet 2 – Floating Point Representation.....	204
Table 175 – RPCObjectUUID.Data4	205
Table 176 – RPCObjectUUID for PNIO	205
Table 177 – RPCInterfaceUUID for PNIO	205
Table 178 – RPCInterfaceUUID for the RPC end point mapper	206
Table 179 – RPCOperationNmb (IO device, controller and supervisor)	207
Table 180 – RPCOperationNmb for endpoint mapper	207
Table 181 – RPCDataRepresentationUUID – defined values	209
Table 182 – RPCInquiryType	210
Table 183 – RPCEPMapStatus	212
Table 184 – Values of NCAFaultStatus	214

Table 185 – Values of NCARrejectStatus	215
Table 186 – LLDP APDU syntax	216
Table 187 – LLDP substitutions	217
Table 188 – LLDP_PNIO_SubType	218
Table 189 – PTCP_PortRxDelayLocal	219
Table 190 – PTCP_PortRxDelayRemote	219
Table 191 – PTCP_PortTxDelayLocal	219
Table 192 – PTCP_PortTxDelayRemote	219
Table 193 – CableDelayLocal	220
Table 194 – RTClass2_PortStatus.State	220
Table 195 – RTClass3_PortStatus.State	220
Table 196 – MRRT_PortStatus.State	221
Table 197 – LLDP_RedOrangePeriodBegin.Offset	221
Table 198 – LLDP_RedOrangePeriodBegin.Valid	221
Table 199 – LLDP_OrangePeriodBegin.Offset	222
Table 200 – LLDP_OrangePeriodBegin.Valid	222
Table 201 – LLDP_GreenPeriodBegin.Offset	222
Table 202 – LLDP_GreenPeriodBegin.Valid	223
Table 203 – LLDP_LengthOfPeriod.Length	223
Table 204 – LLDP_LengthOfPeriod.Valid	223
Table 205 – Primitives issued by LLDP to RTC3PSM	225
Table 206 – Primitives issued by IEEE 802.3 to RTC3PSM	225
Table 207 – Primitives issued by PTCP to RTC3PSM	225
Table 208 – Primitives issued by IEEE 802.1D to RTC3PSM	225
Table 209 – Primitives issued by RTC3PSM to IEEE 802.1D	225
Table 210 – RTC3PSM state table	225
Table 211 – RTC3PSM function table	230
Table 212 – Event function table	231
Table 213 – Primitives issued by LLDP to RTC2PSM	232
Table 214 – Primitives issued by IEEE 802.3 to RTC2PSM	232
Table 215 – Primitives issued by PTCP to RTC2PSM	232
Table 216 – Primitives issued by IEEE 802.1D to RTC2PSM	232
Table 217 – Primitives issued by RTC2PSM to IEEE 802.1D	232
Table 218 – RTC2PSM state table	232
Table 219 – RTC2PSM function table	234
Table 220 – RED RELAY state table	236
Table 221 – RED RELAY function table	238
Table 222 – RED RELAY macro table	238
Table 223 – Primitives issued by MUX to MAC	239
Table 224 – Primitives issued by MAC to MUX	239
Table 225 – MUX state table	240
Table 226 – MUX function table	243
Table 227 – Primitives issued by MAC to DEMUX	243

Table 228 – DEMUX state table	244
Table 229 – DEMUX function table	246
Table 230 – DEMUX macro table	246
Table 231 – IP/UDP APDU syntax	247
Table 232 – IP/UDP substitutions	247
Table 233 – UDP_SrcPort	248
Table 234 – UDP_DstPort	248
Table 235 – IP_DstIPAddress	248
Table 236 – IP Multicast DstIPAddress according to RFC 2365	249
Table 237 – Enterprise number	250
Table 238 – LMPM state table	262
Table 239 – LMPM macros table	265
Table 240 – LMPM function table	265
Table 241 – Error messages	266
Table 242 – VARTYPE values	269
Table 243 – ITEMQUALITYDEF values	270
Table 244 – STATEDEF values	274
Table 245 – GROUPEXCEPTIONDEF values	274
Table 246 – ACCESSRIGHTSDEF values	274
Table 247 – PERSISTDEF values	274
Table 248 – UUID values	277
Table 249 – Data format for serialized connection data	289
Table 250 – Calculation of the RT reference data size	291
Table 251 – Primitives issued by FAL User to FSPM	293
Table 252 – Primitives issued by FSPM to FAL User	299
Table 253 – FSPM state descriptions	305
Table 254 – FSPM state table	306
Table 255 – Primitives issued by FSPM to ARPM	387
Table 256 – Primitives issued by ARPM to FSPM	387
Table 257 – Parameters used with primitives exchanged between FSPM and ARPM	388
Table 258 – ARPM state descriptions	388
Table 259 – ARPM state table	389
Table 260 – Primitives issued by ARPM to DMPM	390
Table 261 – Primitives issued by DMPM to ARPM	391
Table 262 – Parameters used with primitives exchanged between ARPM and DMPM	391
Table 263 – Primitives issued by DMPM to ORPC model	392
Table 264 – Primitives issued by ORPC model to DMPM	392
Table 265 – Parameters used with primitives exchanged between DMPM and ORPC model	392
Table 266 – DMPM state descriptions	393
Table 267 – DMPM state table	393
Table 268 – IO APDU substitutions	394
Table 269 – BlockType	404
Table 270 – BlockVersionHigh	408

Table 271 – BlockVersionLow	409
Table 272 – AlarmType	409
Table 273 – AlarmSpecifier.ChannelDiagnosis	410
Table 274 – AlarmSpecifier.ManufacturerSpecificDiagnosis	410
Table 275 – AlarmSpecifier.SubmoduleDiagnosisState	411
Table 276 – AlarmSpecifier.ARDiagnosisState	411
Table 277 – API	411
Table 278 – SlotNumber	412
Table 279 – SubslotNumber	412
Table 280 – Grouping of DiagnosisData	414
Table 281 – Index (user specific)	415
Table 282 – Index (subslot specific)	415
Table 283 – Index (slot specific)	417
Table 284 – Index (AR specific)	417
Table 285 – Index (API specific)	418
Table 286 – Index (device specific)	419
Table 287 – ARType	419
Table 288 – IOCRMulticastMACAdd using RT_CLASS_UDP	420
Table 289 – IOCRMulticastMACAdd using RT_CLASS_2 or RT_CLASS_3	420
Table 290 – Type 10 OUI	421
Table 291 – ARProperties.State	421
Table 292 – ARProperties.SupervisorTakeoverAllowed	422
Table 293 – ARProperties.ParametrizationServer	422
Table 294 – ARProperties.DeviceAccess	422
Table 295 – ARProperties.CompanionAR	422
Table 296 – ARProperties.AcknowledgeCompanionAR	422
Table 297 – ARProperties.PullModuleAlarmAllowed	423
Table 298 – IOCRProperties.RTClass	423
Table 299 – IOCRProperties.MediaRedundancy	423
Table 300 – IOCRTagHeader.IOCRVLANID	424
Table 301 – IOCRTagHeader.IOUserPriority	424
Table 302 – IOCRTYPE	424
Table 303 – CMInitiatorActivityTimeoutFactor with ARProperties.DeviceAccess:=0	425
Table 304 – CMInitiatorActivityTimeoutFactor with ARProperties.DeviceAccess:=1	425
Table 305 – LengthIOCS	426
Table 306 – LengthIOPS	426
Table 307 – AlarmCRProperties.Priority	426
Table 308 – AlarmCRProperties.Transport	427
Table 309 – AlarmCRTagHeaderHigh.AlarmCRVLANID	427
Table 310 – AlarmCRTagHeaderHigh.AlarmUserPriority	427
Table 311 – AlarmCRTagHeaderLow.AlarmCRVLANID	427
Table 312 – AlarmCRTagHeaderLow.AlarmUserPriority	428
Table 313 – AlarmSequenceNumber	428

Table 314 – AlarmCRType	428
Table 315 – RTATimeoutFactor	429
Table 316 – RTARetries	429
Table 317 – AddressResolutionProperties.Protocol	429
Table 318 – AddressResolutionProperties.Factor	430
Table 319 – MCITimeoutFactor	430
Table 320 – ModuleIdentNumber	431
Table 321 – SubmoduleIdentNumber	431
Table 322 – ControlBlockProperties in conjunction with ControlCommand.ApplicationReady	433
Table 323 – ControlBlockProperties in conjunction with the other values of the field ControlCommand	433
Table 324 – ControlCommand.PrmEnd	433
Table 325 – ControlCommand.ApplicationReady	433
Table 326 – ControlCommand.Release	433
Table 327 – ControlCommand.Done	434
Table 328 – ControlCommand.ReadyForCompanion	434
Table 329 – ControlCommand.ReadyForRT_CLASS_3	434
Table 330 – DataDescription.Type	434
Table 331 – Values of SendClockFactor	435
Table 332 – Values of ReductionRatio	436
Table 333 – Values of Phase	437
Table 334 – Values of Sequence	437
Table 335 – DataHoldFactor of a frame	437
Table 336 – WatchdogFactor of a frame	438
Table 337 – Values of FrameSendOffset	438
Table 338 – Values of ErrorCode for negative responses	439
Table 339 – Values of ErrorDecode	440
Table 340 – Coding of ErrorCode1 with ErrorDecode PNORW	440
Table 341 – Values of ErrorCode1 and ErrorCode2 for ErrorDecode with the value PNIO	442
Table 342 – Values of ErrorCode2 for ErrorCode1 = RPC	448
Table 343 – ModuleState	448
Table 344 – SubmoduleState.AddInfo	449
Table 345 – SubmoduleState.QualifiedInfo	449
Table 346 – SubmoduleState.MaintenanceRequired	449
Table 347 – SubmoduleState.MaintenanceDemanded	449
Table 348 – SubmoduleState.DiagInfo	449
Table 349 – SubmoduleState.ARInfo	450
Table 350 – SubmoduleState.IdentInfo	450
Table 351 – SubmoduleState.FormatIndicator	450
Table 352 – SubmoduleState.Detail	451
Table 353 – SubmoduleProperties.Type	451
Table 354 – SubmoduleProperties.SharedInput	451
Table 355 – SubmoduleProperties.ReduceInputSubmoduleDataLength	452
Table 356 – SubmoduleProperties.ReduceOutputSubmoduleDataLength	452

Table 357 – SubmoduleProperties.DiscardIOXS	452
Table 358 – SubstitutionMode	453
Table 359 – SubstituteActiveFlag	453
Table 360 – InitiatorUDPRTPort	453
Table 361 – ResponderUDPRTPort	454
Table 362 – InitiatorRPCServerPort	454
Table 363 – ResponderRPCServerPort	454
Table 364 – IM_Hardware_Revision	455
Table 365 – IM_SWRevision_Functional_Enhancement	455
Table 366 – IM_SWRevision_Bug_Fix	455
Table 367 – IM_SWRevision_Internal_Change	455
Table 368 – IM_Revision_Counter	456
Table 369 – IM_Profile_ID	456
Table 370 – IM_Profile_Specific_Type	456
Table 371 – IM_Version_Major	456
Table 372 – IM_Version_Minor	457
Table 373 – IM_Date	458
Table 374 – UserStructureIdentifier	459
Table 375 – ChannelErrorType	460
Table 376 – ChannelNumber	462
Table 377 – ChannelProperties.Type	462
Table 378 – Valid combinations within ChannelProperties	463
Table 379 – Valid combinations for Alarmnotification and RecordDataRead(DiagnosisData)	463
Table 380 – ChannelProperties.Specifier	464
Table 381 – ChannelProperties.Direction	464
Table 382 – ExtChannelErrorType	465
Table 383 – ExtChannelErrorType for ChannelErrorType 0 – 0x7FFF	465
Table 384 – ExtChannelErrorType for ChannelErrorType “Data transmission impossible”	465
Table 385 – ExtChannelErrorType for ChannelErrorType “Remote mismatch”	465
Table 386 – ExtChannelErrorType for ChannelErrorType “Media redundancy mismatch”	466
Table 387 – ExtChannelErrorType for ChannelErrorType “Sync mismatch” and for ChannelErrorType “Time mismatch”	466
Table 388 – ExtChannelErrorType for ChannelErrorType “Isochronous mode mismatch”	466
Table 389 – ExtChannelErrorType for ChannelErrorType “Multicast CR mismatch”	467
Table 390 – ExtChannelErrorType for ChannelErrorType “Fiber optic mismatch”	467
Table 391 – ExtChannelErrorType for ChannelErrorType “Network component function mismatch”	467
Table 392 – Values for Accumulative Info	468
Table 393 – Values for “Fiber optic mismatch” – “Power Budget”	468
Table 394 – Values for “Network component function mismatch” – “Frame dropped”	468
Table 395 – Values for “Remote mismatch” – “Peer CableDelay mismatch”	469
Table 396 – Values for QualifiedChannelQualifier	469
Table 397 – Values for MaintenanceStatus	469
Table 398 – URRecordIndex	470

Table 399 – URRecordLength	470
Table 400 – LineDelay.Value with LineDelay.FormatIndicator == 0	472
Table 401 – LineDelay.Value with LineDelay.FormatIndicator == 1	472
Table 402 – LineDelay.FormatIndicator	472
Table 403 – RxPort	472
Table 404 – NumberOfTxPortGroups	473
Table 405 – TxPortEntry	473
Table 406 – FrameDetails.SyncFrame	474
Table 407 – FrameDetails.MeaningFrameSendOffset	474
Table 408 – MAUType	475
Table 409 – Valid combinations between MAUType and LinkState	476
Table 410 – CheckSyncMode.CableDelay	476
Table 411 – CheckSyncMode.SyncMaster	477
Table 412 – MAUTypeMode.Check	477
Table 413 – DomainBoundaryIngress	477
Table 414 – DomainBoundaryEgress	478
Table 415 – MulticastBoundary	478
Table 416 – PeerToPeerBoundary	479
Table 417 – DCPBoundary	479
Table 418 – LinkState.Link	480
Table 419 – LinkState.Port	480
Table 420 – MediaType	480
Table 421 – MaxBridgeDelay	481
Table 422 – NumberOfPorts	481
Table 423 – MaxPortTxDelay	481
Table 424 – MaxPortRxDelay	481
Table 425 – MaxLineRxDelay	482
Table 426 – YellowTime	482
Table 427 – StartOfRedFrameID	482
Table 428 – EndOfRedFrameID	483
Table 429 – Dependencies of StartOfRedFrameID and EndOfRedFrameID	483
Table 430 – NumberOfAssignments	483
Table 431 – NumberOfPhases	483
Table 432 – AssignedValueForReservedBegin	484
Table 433 – AssignedValueForOrangeBegin	484
Table 434 – AssignedValueForReservedEnd	484
Table 435 – Values of RedOrangePeriodBegin	485
Table 436 – Dependencies of RedOrangePeriodBegin, OrangePeriodBegin, and GreenPeriodBegin	485
Table 437 – Values of OrangePeriodBegin	485
Table 438 – Values of GreenPeriodBegin	485
Table 439 – EtherType	486
Table 440 – SyncProperties.Role	486
Table 441 – SyncProperties.SyncID	486

Table 442 – Dependencies of ReservedIntervalBegin and ReservedIntervalEnd	487
Table 443 – SyncSendFactor	487
Table 444 – PTCPTimeoutFactor.....	488
Table 445 – PTCPTakeoverTimeoutFactor.....	488
Table 446 – PTCPMasterStartupTime	489
Table 447 – PLLWindow	489
Table 448 – TimeDataCycle	490
Table 449 – TimelOInput.....	490
Table 450 – TimelOOutput	491
Table 451 – TimelOInputValid	491
Table 452 – TimelOOutputValid	491
Table 453 – ControllerApplicationCycleFactor	491
Table 454 – MRP_Role	491
Table 455 – MRP_RTMode.RTClass1_2.....	492
Table 456 – MRP_TOPchgT	492
Table 457 – MRP_TOPNRmax	492
Table 458 – MRP_TSTshortT.....	493
Table 459 – MRP_TSTdefaultT	493
Table 460 – MRP_TSTNRmax.....	493
Table 461 – MRP_LNKdownT	494
Table 462 – MRP_LNKupT	494
Table 463 – MRP_LNKNRmax.....	494
Table 464 – MRP_RTState	495
Table 465 – MRP_Check.MediaRedundancyManager	495
Table 466 – MRP_Check.MRP_DomainUUID	495
Table 467 – VendorBlockType	495
Table 468 – FiberOpticType	496
Table 469 – FiberOpticCableType	496
Table 470 – FiberOpticPowerBudgetType.Value	496
Table 471 – FiberOpticPowerBudgetType.CheckEnable.....	497
Table 472 – NCDropBudgetType.Value	497
Table 473 – NCDropBudgetType.CheckEnable.....	497
Table 474 – FSHelloMode.Mode	498
Table 475 – FSHelloInterval	499
Table 476 – FSHelloRetry	499
Table 477 – FSHelloDelay.....	499
Table 478 – FSPParameterMode.Mode	500
Table 479 – FSPParameterUUID	500
Table 480 – ArgsLength check.....	501
Table 481 – ARBlockReq – request check.....	501
Table 482 – IOCRBlockReq – request check.....	502
Table 483 – AlarmCRBlockReq – request check.....	505
Table 484 – ExpectedSubmoduleBlockReq – request check	506

Table 485 – PrmServerBlock – request check	508
Table 486 – MCRBlockReq – request check.....	508
Table 487 – ARRPCBlockReq – request check	509
Table 488 – ArgsLength check.....	509
Table 489 – ARBlockRes – response check	509
Table 490 – IOCRBlockRes – response check	510
Table 491 – AlarmCRBlockRes – response check.....	511
Table 492 – ModuleDiffBlock – response check	511
Table 493 – ArgsLength check.....	513
Table 494 – ControlBlockConnect – request check	513
Table 495 – ControlBlockPlug – request check.....	513
Table 496 – ArgsLength check.....	514
Table 497 – ControlBlockConnect – response check.....	514
Table 498 – ControlBlockPlug – response check.....	515
Table 499 – ArgsLength check.....	516
Table 500 – ControlBlockConnect – request check	516
Table 501 – ControlBlockPlug – request check.....	516
Table 502 – ArgsLength check.....	517
Table 503 – ControlBlockConnect – response check.....	517
Table 504 – ControlBlockPlug – response check.....	518
Table 505 – ArgsLength check.....	519
Table 506 – ReleaseBlock – request check	519
Table 507 – ArgsLength check.....	520
Table 508 – ReleaseBlock – response check	520
Table 509 – ArgsLength check.....	521
Table 510 – IODWriteReqHeader – request check.....	521
Table 511 – ArgsLength check.....	522
Table 512 – IODWriteResHeader – response check	522
Table 513 – ArgsLength check.....	523
Table 514 – ArgsLength check.....	523
Table 515 – ArgsLength check.....	524
Table 516 – IODReadReqHeader – request check.....	525
Table 517 – RecordDataReadQuery – request check.....	525
Table 518 – ArgsLength check.....	525
Table 519 – IODReadResHeader – response check	526
Table 520 – Primitives issued by AP-Context (FAL user) to FSPMDEV	529
Table 521 – Primitives issued by FSPMDEV to AP-Context (FAL user).....	538
Table 522 – FSPMDEV protocol machine for multicast communication	549
Table 523 – Functions used by AP-Context (FAL user) to FSPMDEV	558
Table 524 – Function used by FSPMDEV to AP-Context (FAL user)	562
Table 525 – Primitives issued by AP-Context (FAL user) to FSPMCTL	570
Table 526 – Primitives issued by FSPMCTL to AP-Context (FAL user)	575
Table 527 – Function used by AP-Context (FAL user) to FSPMCTL.....	585

Table 528 – Functions used by FSPMCTL to AP-Context (FAL user).....	600
Table 529 – Primitives issued by FSPMDEV or FSPMCTL to ALPMI	609
Table 530 – Primitives issued by ALPMI to FSPMDEV or FSPMCTL	610
Table 531 – Primitives issued by CMDEV or CMCTL to ALPMI	610
Table 532 – Primitives issued by ALPMI to CMCTL or CMDEV	610
Table 533 – Primitives issued by APMR to ALPMI	611
Table 534 – Primitives issued by ALPMI to APMR	611
Table 535 – Primitives issued by APMS to ALPMI	611
Table 536 – Primitives issued by ALPMI to APMS	612
Table 537 – ALPMI state table	612
Table 538 – Primitives issued by FSPMDEV or FSPMCTL to ALPMR	616
Table 539 – Primitives issued by ALPMR to FSPMDEV or FSPMCTL	616
Table 540 – Primitives issued by CMDEV or CMCTL to ALPMR	617
Table 541 – Primitives issued by ALPMR to CMCTL or CMDEV	617
Table 542 – Primitives issued by APMR to ALPMR.....	617
Table 543 – Primitives issued by ALPMR to APMR.....	618
Table 544 – Primitives issued by APMS to ALPMR.....	618
Table 545 – Primitives issued by ALPMR to APMS	618
Table 546 – ALPMR state table	619
Table 547 – Primitives issued by CMCTL to NRPM	624
Table 548 – Primitives issued by NRPM to CMCTL	625
Table 549 – Primitives issued by other machines to NRPM	626
Table 550 – Primitives issued by NRPM to other machines	627
Table 551 – NRPM state table	628
Table 552 – Functions used by the NRPM and RMPM	632
Table 553 – Primitives issued by CMDEV to RMPM	633
Table 554 – Primitives issued by RMPM to CMDEV	634
Table 555 – Primitives issued by RPC to RMPM.....	635
Table 556 – Primitives issued by RMPM to RPC.....	635
Table 557 – Primitives issued by other machines to RMPM.....	635
Table 558 – Primitives issued by RMPM to other machines.....	636
Table 559 – RMPM state table	637
Table 560 – Macros used by RMPM	645
Table 561 – Primitives issued by FSPMDEV to CMDEV	645
Table 562 – Primitives issued by CMDEV to FSPMDEV	646
Table 563 – CMDEV state table.....	647
Table 564 – Macros used by CMDEV	659
Table 565 – Variables used by CMDEV.....	660
Table 566 – CMDEV(DA) state table	660
Table 567 – Functions used by CMDEV(DA).....	662
Table 568 – Macros used by CMDEV(DA).....	662
Table 569 – Primitives issued by CMDEV to NRMC.....	663
Table 570 – Primitives issued by NRMC to CMDEV.....	663

Table 571 – Primitives issued by CPM to NRMC	663
Table 572 – Primitives issued by NRMC to CPM	664
Table 573 – Primitives issued by other machines to NRMC	664
Table 574 – Primitives issued by NRMC to other machines	664
Table 575 – NRMC state table	665
Table 576 – Primitives issued by FSPMCTL to CMCTL.....	670
Table 577 – Primitives issued by CMCTL to FSPMCTL.....	671
Table 578 – CMCTL state table.....	672
Table A.1 – Unicast FDB entries	683
Table A.2 – Multicast FDB entries	683
Table A.3 – Broadcast FDB entry	685
Table M.1 – Priority regeneration and queue usage	700
Table Q.1 – Parameter for the equation	709
Table R.1 – Truth table	710
Table S.1 – TX-error	712

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 implementors 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 IEC 61784 series. Use of the various protocol types in other combinations may require permission of 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 10 elements and possibly other types given in the normative elements of this standard.

The following patent rights for Type 10 have been announced by [HI]:

WO publication	Title (WO)
WO 99/046908	Local network, especially ethernet network, with redundancy properties and redundancy manager

The following patent rights for Type 10 have been announced by [SI]:

WO publication	Title (WO)
WO 99/046908	Local network, especially ethernet network, with redundancy properties and redundancy manager
WO 00/026731	Automation system and method for accessing the functionality of hardware components
WO 02/043336	System and method for the parallel transmission of real-time critical and non real-time critical data via switched data networks especially ethernet
WO 02/076033	Synchronous, clocked communication system with local input/output components and method for integrating local input/output components into such a system
WO 03/028258	Method for synchronising nodes of a communication system
WO 03/028259	Communications system and method for synchronising a communications cycle
WO 04/030284	Method for permanent redundant transmission of data telegrams in communication systems

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:

[HI]: Hirschmann Automation and Control GmbH
Stuttgarter Straße 45-51
D-72654 Neckartenzlingen
Germany

[SI]: Siemens AG
CT IP L&T
Hr. Hans-Jörg Müller
Otto-Hahn-Ring 6
D-81739 Munich
Germany

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 6-10: Application layer protocol specification – Type 10 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 10 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 defines in an abstract way the externally visible behavior provided by the Type 10 fieldbus application layer in terms of

- a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
- b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
- c) the application context state machine defining the application service behavior visible between communicating application entities; and
- d) the application relationship state machines defining the communication behavior visible between communicating application entities; and.

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

- a) define the wire-representation of the service primitives defined in IEC 61158-5-10, and
- b) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 10 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) 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 61158-5-10.

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 IEC 61158–6.

1.3 Conformance

This standard does not specify individual implementations or products, nor does it 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 60559, *Binary floating-point arithmetic for microprocessor systems*

IEC 61158-5-10:2010¹, *Industrial communication networks – Fieldbus specifications – Application layer service definition – Type 10 elements*

IEC 61784-3-3, *Industrial communication networks – Profiles – Part 3-3: Functional safety fieldbuses – Additional specifications for CPF 3*

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*

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*

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

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

IEEE 802-2001, *IEEE Standard for Local and metropolitan area networks: Overview and Architecture*, available at <<http://www.ieee.org>>

IEEE 802.1AB-2005, *IEEE Standard for Local and metropolitan area networks: Station and Media Access Control Connectivity Discovery*, available at <<http://www.ieee.org>>

IEEE 802.1D-2004, *IEEE Standard for Local and metropolitan area networks – Media access control (MAC) Bridges*, available at <<http://www.ieee.org>>

IEEE 802.1Q-2005, *IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks*, available at <<http://www.ieee.org>>

IEEE 802.3-2005, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and Physical Layer specifications*, available at <<http://www.ieee.org>>

IEEE 802.11-1999, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access, Control (MAC) and Physical Layer, (PHY) Specifications*, available at <<http://www.ieee.org>>

¹ To be published.

IEEE 802.15.1-2005, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*, available at <<http://www.ieee.org>>

IETF RFC 768, *User Datagram Protocol*; available at <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*; available at <<http://www.ietf.org>>

IETF RFC 792, *Internet Control Message Protocol*; available at <<http://www.ietf.org>>

IETF RFC 826, *An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*; available at <<http://www.ietf.org>>

IETF RFC 1034, *Domain names – concepts and facilities*; available at <<http://www.ietf.org>>

IETF RFC 1112, *Host Extensions for IP Multicasting*; available at <<http://www.ietf.org>>

IETF RFC 1518, *An Architecture for IP Address Allocation with CIDR*, available at <<http://www.ietf.org>>

IETF RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*, available at <<http://www.ietf.org>>

IETF RFC 2131, *Dynamic Host Configuration Protocol*; available at <<http://www.ietf.org>>

IETF RFC 2132, *DHCP Options and BOOTP Vendor Extensions*; available at <<http://www.ietf.org>>

IETF RFC 2365, *Administratively Scoped IP Multicast*; available at <<http://www.ietf.org>>

IETF RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*; available at <<http://www.ietf.org>>

IETF RFC 2674, *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*, available at <<http://www.ietf.org>>

IETF RFC 2737, *Entity MIB (Version 2)*, available at <<http://www.ietf.org>>

IETF RFC 2863, *The Interfaces Group MIB*, available at <<http://www.ietf.org>>

IETF RFC 3330, *Special-Use IPv4 Addresses*, available at <<http://www.ietf.org>>

IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, available at <<http://www.ietf.org>>

IETF RFC 3490, *Internationalizing Domain Names in Applications (IDNA)*; available at <<http://www.ietf.org>>

IETF RFC 3621, *Power Ethernet MIB*, available at <<http://www.ietf.org>>

IETF RFC 4836, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, available at <<http://www.ietf.org>>

The Open Group — Publication C706, *Technical standard DCE1.1: Remote Procedure Call*, available at <<http://www.opengroup.org/onlinepubs/9629399/toc.htm>>

3 Terms, definitions, abbreviations, symbols and conventions

3.1 Referenced terms and definitions

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:

- a) abstract syntax
- b) presentation context

3.1.3 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

3.1.4 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

3.2 Additional terms and definitions for distributed automation

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

3.2.1

active connection control object

instance of a certain FAL class that abstracts the interconnection facility (as Consumer and Provider) of an automation device

3.2.2

configuration data base

interconnection information maintained by the ACCO ASE

3.2.3

connection

the logical link between sink and source of attributes and services at different custom interfaces of Custom RT-Auto objects

3.2.4

connection channel

description of a connection between a sink and a source of data items

3.2.5

consumer

node or sink that is receiving data from a producer

3.2.6

consumerID

unambiguous identifier within the scope of the ACCO assigned by the consumer to recognize the internal data of a configured interconnection sink.

3.2.7

data marshaling

the encoding of parameters of the FAL service primitives with respect to their interface definition

NOTE This is part of the abstract ORPC model.

3.2.8

engineering

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

3.2.9

event

an instance of a change of conditions

3.2.10

interface

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

3.2.11

interface definition language

syntax and semantic of describing service parameters in a formal way

NOTE This description is the input for the ORPC model, especially for the ORPC wire protocol.

3.2.12

interface pointer

key attribute that unambiguously addresses an object interface instance

3.2.13

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

3.2.14

method

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

3.2.15

object remote procedure call

model for object oriented or component based remote method invocation

3.2.16

physical device

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

3.2.17

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".

3.2.18

provider

source of a data connection.

3.2.19

providerID

an unambiguous identifier within the scope of the ACCO assigned by the provider to recognize the internal data of a configured connection source

3.2.20

quality code

additional status information of a data item

3.2.21

quality code aware

attribute of the RT-Auto class that indicates that an RT-Auto object uses a status code for its data items

3.2.22

quality code unaware

opposite of quality code aware

3.2.23

RT-Auto

an FAL class that abstracts the automation function as a process-related component of an automation device

3.2.24

runtime object model

objects that exist in a device together with their interfaces and methods that are accessible

3.3 Additional terms and definitions for decentralized periphery

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

3.3.1

alarm

activation of an event that shows a critical state

3.3.2

alarm ack

acknowledgment of an event that shows a critical state

3.3.3

alarm data object

object(s) which represent critical states referenced by device/slot/subslot/alarm type

3.3.4

allocate

take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.3.5

application

function or data structure for which data is consumed or produced

3.3.6

application layer interoperability

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

3.3.7

application objects

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

3.3.8

application process

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

3.3.9

application process identifier

distinguishes multiple application processes used in a device

3.3.10

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.

3.3.11

application process object class

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

3.3.12

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

3.3.13

application relationship application service element

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

3.3.14

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.

3.3.15

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.

3.3.16

backup

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

3.3.17

behavior

indication of how an object responds to particular events

3.3.18

channel

representation of a single physical or logical link of an input or output application object of a server to the process in order to support addressing of diagnosis information

NOTE The channel typically represents a single connector or clamp as a real interface of a module or sub-module. This reference is used to identify points of failure within diagnosis PDUs.

3.3.19

channel related diagnosis

information concerning a specific element of an input or output application object, provided for maintenance purposes

EXAMPLE open loop

3.3.20

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.

3.3.21

class attributes

attribute that is shared by all objects within the same class

3.3.22

class code

unique identifier assigned to each object class

3.3.23

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.

3.3.24

clear

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

3.3.25

client

a) object which uses the services of another (server) object to perform a task

b) initiator of a PDU to which a server reacts

3.3.26

common profile

a collection of device independent information and functionality providing consistency between all devices

3.3.27

communication data object

object(s) which are parameter of communication relationships and referenced by device/ slot/ subslot/ index

3.3.28

configuration check

comparison of the expected IO-Data object structuring of the client with the real IO-Data object structuring to the server in the start-up phase

3.3.29

configuration fault

an unacceptable difference between the expected IO-Data object structuring and the real IO-Data object structuring, as detected by the server

3.3.30

configuration identifier

representation of a portion of IO Data of a single input- and/or output-module of a server

3.3.31

consume

act of receiving data from a provider

3.3.32

consumer

node or sink receiving data from a provider

3.3.33

context management

network-accessible information (communication objects) that supports managing the operation of the fieldbus system, including the application layer

NOTE Managing includes functions such as controlling, monitoring, and diagnosing.

3.3.34

conveyance path

unidirectional flow of APDUs across an application relationship

3.3.35

cyclic

repetitive in a regular manner

3.3.36

data consistency

means for coherent transmission and access of the input- or output-data object between and within client and server

3.3.37

device

physical hardware connected to the link

NOTE A device may contain more than one node.

3.3.38

device ID

a vendor assigned device type identification

3.3.39

device profile

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

3.3.40

diagnosis data object

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

3.3.41

diagnosis information

all data available at the server for maintenance purposes

3.3.42

dynamic reconfiguration

change of IO data objects without interruption of an established application relationship and continuous updating of non-changed IO data objects

3.3.43

endpoint

one of the communicating entities involved in a connection

3.3.44

engineering

abstract term that characterizes the client application or device responsible for configuring an automation system

3.3.45

error

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

3.3.46

error class

general grouping for related error definitions and corresponding error codes

3.3.47

error code

identification of a specific type of error within an error class

3.3.48

event

instance of a change of conditions

3.3.49

extended channel related diagnosis

information concerning a specific element of a specific application object, provided for maintenance purposes

EXAMPLE Link Fail

3.3.50

frame

unit of data transfer on a link

3.3.51

identification data object

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

3.3.52**implicit AR endpoint**

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

3.3.53**index**

address of a record data object within an application process

3.3.54**instance**

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

3.3.55**instance attributes**

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

3.3.56**instantiated**

object that has been created in a device

3.3.57**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.

3.3.58**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.

3.3.59**IO data object**

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

3.3.60**IO device**

field device which acts as server for IO operation

3.3.61**IO parameter server**

server for application parameter of IO devices (client)

NOTE This is usually a device to backup parameter data and to log online changes of device parameter.

3.3.62**IO subsystem**

subsystem composed of one IO controller and all its associated IO devices

3.3.63

IO supervisor

engineering device which manages commissioning and diagnosis of an IO system

3.3.64

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.

3.3.65

Isochronous mode

IO system operating tightly synchronized with a jitter of less than 1 ps

3.3.66

member

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

3.3.67

message

synonym for frame

3.3.68

method

a synonym for an operational service which is provided by the server ASE and invoked by a client

3.3.69

module

hardware or logical component of a physical device

3.3.70

network

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

3.3.71

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

3.3.72

object specific service

service unique to the object class which defines it

3.3.73

operate

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

3.3.74

packet

frame

3.3.75

peer

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

3.3.76