

DIN EN 61158-6-24

ICS 35.100.70; 35.160

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 6-24: Protokollspezifikation des Application Layer
(Anwendungsschicht) – Typ 24-Elemente
(IEC 61158-6-24:2014);
Englische Fassung EN 61158-6-24:2014, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 6-24: Application layer protocol specification – Type-24 Elements
(IEC 61158-6-24:2014);
English version EN 61158-6-24:2014, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 6-24: Spécification du protocole de la couche application – Éléments de type 24
(CEI 61158-6-24:2014);
Version anglaise EN 61158-6-24:2014, seulement en CD-ROM

Gesamtumfang 130 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-09-23 angenommene Europäische Norm als DIN-Norm ist 2015-09-01.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61158-600:2013-08.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in 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 61158-6-24 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als 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 24 fest.

Nationaler Anhang NA (informativ)

1 Anwendungsbereich^{N1)}

1.1 Allgemeines

Der Feldbus-Application Layer (FAL) liefert Benutzerprogramme mit Hilfsmitteln für den Zugriff auf die Feldbuskommunikationsumgebung. In dieser Hinsicht, kann die FAL als ein Fenster zwischen zugehörigen Anwendungsprogrammen betrachtet werden.

Diese Norm stellt für Typ 24 Feldbusse maßgebliche allgemeine Elemente für zeitkritischen und nicht zeitkritischen Datentransfer zwischen Anwendungsprogrammen in einer Automatisierungsumgebung bereit. Der Begriff „zeitkritisch“ (en: time-critical) wird verwendet um die Präsenz eines Zeitfensters darzustellen, in dem eine oder mehrere festgelegte Aktivitäten mit einem vorherbestimmten Grad an Gewissheit erledigt werden. Das Scheitern der Erledigung der festgelegten Aktivität innerhalb des Zeitfensters, trägt das Risiko des Betriebsausfalls der Anwendung mit, und somit das aufgeforderte Eingreifen mit dazugehörigem Risiko für Geräte, Anlage und möglicherweise Menschenleben.

Diese Norm definiert auf eine abstrakte Weise das von außen sichtbare Verhalten, das vom Typ 24 des Feldbus Application Layer bereitgestellt wird, in Form von:

- a) einer abstrakten Syntax zur Festlegung der Application Layer-Protokoll-Dateneinheiten die zwischen kommunizierenden Anwendungseinheiten übermittelt werden,
- b) der übertragenen Syntax zur Festlegung der Application Layer-Protokoll-Dateneinheiten die zwischen kommunizierenden Anwendungseinheiten übermittelt werden,
- c) des Anwendungskontext der Zustandsmaschine zur Festlegung des sichtbaren Dienstverhalten zwischen kommunizierenden Anwendungseinheiten,
- d) der Anwendungsbeziehung der Zustandsmaschine zur Festlegung des sichtbaren Kommunikationsverhalten zwischen kommunizierenden Anwendungseinheiten.

Der Zweck dieser Norm ist, das Protokoll zu definieren, welches bereitgestellt wird zur

- Festlegung der Leitungsdarstellung der Dienstprimitiven, welche in IEC 61158-5-24 beschrieben sind und
- zur Festlegung des von außen sichtbaren Verhaltens verbunden mit ihrer Übertragung.

Diese Norm beschreibt das Protokoll des Typ 24 Feldbus Application Layer in Übereinstimmung mit dem OSI Basis Referenzmodell (ISO/IEC 7498-1) und der OSI Application Layer-Struktur (ISO/IEC 9545).

1.2 Spezifikationen

Das hauptsächliche Ziel dieser Norm, ist das Festlegen der Syntax und des Verhaltens des Application Layer-Protokolls, welcher die in IEC 61158-5-24 beschriebenen Application Layer-Dienste, vermittelt.

Ein zweites Ziel, ist das Bereitstellen von Migrationspfade von bisherigen bestehenden industriellen Kommunikationsprotokollen. Es ist dieses letztere Ziel, welches zur Steigerung von unterschiedlichen Protokollen, die in der Reihe IEC 61158-6 genormt werden, führt.

1.3 Konformität

Diese Norm legt keine individuellen Implementierungen oder Produkte fest, noch beschränkt sie die Implementierungen von Application Layer-Einheiten innerhalb eines industriellen Automatisierungssystems.

Konformität wird durch das Implementieren dieser Application Layer-Protokoll-Spezifikation erreicht.

^{N1)} Die Nummerierung aus IEC 61158-6-24 wurde beibehalten.

3.2 Zusätzliche Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.2.1

Alarm

(en: **alarm**)

Feldgerätestatus, der durch den Dienst zur Feldgerätesteuerung (FDC) des Typ-24-Feldbusses mitteilt, dass das Gerät ein schwerwiegendes Problem erkannt hat, das gelöst werden muss, und dass es den Normalbetrieb nicht fortsetzen kann

Anmerkung 1 zum Begriff: Alle Alarmstatus sind gesperrt und benötigen einige Operationen, um zu erlöschen.

Anmerkung 2 zum Begriff: Alarmer werden in drei Gruppen eingeteilt: Kommunikationsalarmer, Alarmer bei unzulässigen Befehlen und anwendungsspezifische Alarmer. Genaue Definitionen hängen jedoch von der Implementierung der einzelnen Feldgeräte ab.

3.2.2

Anwendungsprozessobjekt

(en: **application process object**)

Netzwerkdarstellung eines bestimmten Aspekts eines Anwendungsprozesses (AP), der als über das Netzwerk zugängliches Objekt, das innerhalb eines AP oder innerhalb eines anderen APO enthalten ist, modelliert wird

Anmerkung 1 zum Begriff: Siehe IEC 61158-1:2014, 9.3.4.

3.2.3

Kontext des Anwendungsprozesses AP-Kontext

(en: **application process context ; AP context**)

gemeinsames Wissen oder allgemeiner Satz von Regeln zur Steuerung der Kommunikation von FAL-Anwendungsentitäten (AEs) und zur Beschreibung des gemeinsamen zulässigen Kommunikationsverhaltens zwischen AEs, die Teil eines bestimmten Satzes von Anwendungsbeziehungen (ARs) sind

Anmerkung 1 zum Begriff: Mit Hilfe der Option, die ein Benutzer während der Nutzung eines Dienstes zur Feldbusverwaltung (FSM) zum Auslesen der Peer-AP-Einrichtung auswählen kann, oder durch die automatische Verhandlungsfunktion des FSM-Systems usw. können Daten innerhalb des AP-Kontextes vom Benutzer im Vorfeld festgelegt werden. Das anzuwendende Verfahren hängt von der Spezifikation der einzelnen Implementierungen ab.

3.2.4

Anwendungsprozessstyp AP-Typ

(en: **application process type; AP type**)

Beschreibung einer Klassifikation von Anwendungsprozessen (APs) in Bezug auf einen Satz von Fähigkeiten für die FAL des Typ 24-Feldbusses

Anmerkung 1 zum Begriff: AP-Typen werden nach ihren Anwendungsrollen im Feldbus-Netzwerk in drei Klassen eingeteilt: C1-Master-AP, C2-Master-AP und Slave-AP.

3.2.5

Async-Befehl

(en: **async command**)

Typ einer Befehls-APDU (Anwendungsprotokolldateneinheit) des FDC-Dienstes der FAL Typ 24, die jederzeit nach der vorausgegangenen Transaktion ohne Berücksichtigung der Synchronisation mit dem Kommunikationszyklus ausgegeben werden kann

Anmerkung 1 zum Begriff: Festlegungen dazu, welcher Befehl asynchron oder nicht asynchron sein sollte, hängen von der Anwendung ab. Diese können in Form eines aufgelisteten Satzes von Befehlen und Antworten oder in Form eines Geräteprofils bereitgestellt werden. Siehe 4.4 und Anhang A.

3.2.6**asynchrone Kommunikation****(en: asynchronous communication)**

Zustand oder Kommunikationsart für den FDC-Dienst der FAL Typ 24, bei dem/der ein Befehl jederzeit nach der vorausgegangenen Transaktion ohne Berücksichtigung der Synchronisation mit dem Kommunikationszyklus ausgegeben werden kann

Anmerkung 1 zum Begriff: In diesem Zustand können keine Sync-Befehle ausgegeben werden; Async-Befehle können aber ausgegeben werden.

3.2.7**Attribut****(en: attribute)**

Information oder Parameter, die/der in veränderlichen Teilen eines Objekts enthalten ist

Anmerkung 1 zum Begriff: Attribute liefern üblicherweise Statusinformationen oder steuern den Betrieb eines Objekts. Attribute können auch das Verhalten eines Objekts beeinflussen.

3.2.8**C1-Master****(en: C1 master)**

AP-Typ, der über Master-Einrichtungen für den FDC-Dienst der FAL Typ 24 verfügt, oder das Gerät, das diesen AP-Typ implementiert

Anmerkung 1 zum Begriff: In einem Netzwerk des Typ 24-Feldbusses ist nur ein einziger C1-Master vorhanden.

3.2.9**C2-Master****(en: C2 master)**

AP-Typ, der nur über Überwachungseinrichtungen für den FDC-Dienst, aber über Anforderer-Einrichtungen für den MSG-Dienst des FAL Typ 24 verfügt, oder das Gerät, das diesen AP-Typ implementiert

Anmerkung 1 zum Begriff: In einem Netzwerk des Typ 24-Feldbusses können weniger als zwei C2-Master vorhanden sein.

3.2.10**Befehl****(en: command)**

PDU, die von einem Anforderer oder einem Master ausgegeben wird, um einen Beantworter oder einen Slave zur Ausführung bestimmter Funktionen zu bringen

3.2.11**Kommunikation, Transfer, Übertragung****(en: communication, transfer, transmission)**

- Kommunikation: Prozess des formalen Austausches von Informationen zwischen zwei oder mehr Geräten, Benutzern, APs oder Entitäten;
- Transfer: Prozess der Übermittlung einer PDU von einem Sender an einen Empfänger;
- Übertragung: Prozess der Aussendung und Übertragung elektrischer Signale oder codierter Daten

3.2.12**Kommunikationszyklus****(en: communication cycle)**

Zeitspanne wiederholter Aktivitäten, die während des Verbindungsaufbaus für das FDC-Protokoll der FAL Typ 24 mit dem Übertragungszyklus synchronisiert wird

Anmerkung 1 zum Begriff: Der Kommunikationszyklus kann mit dem Übertragungszyklus synchronisiert werden, der mit einem festgelegten Skalierungsfaktor multipliziert wird.

3.2.13

Verbindung

(en: connection)

Kontext oder logische Verknüpfung zwischen einem Master- und einem Slave-Objekt unter spezifischen Bedingungen für das FDC-Protokoll für die FAL Typ 24

3.2.14

zyklisch

(en: cyclic)

sich regelmäßig wiederholend

3.2.15

zyklische Kommunikation

(en: cyclic communication)

Übertragungsmodus, bei dem Anforderungs- und Antwort-PDUs wiederholt in den geplanten Zeitschlitz ausgetauscht werden, welche mit einem Übertragungszyklus für das Protokoll der unteren Schicht des Typs 24 synchronisiert werden

Anmerkung 1 zum Begriff: Auf dem Application Layer wird der Kommunikationszyklus in diesem Modus durch den Übertragungszyklus ausgelöst.

3.2.16

Zyklusskalierungszähler

(en: cyclic scale counter)

Zähler zur Erzeugung eines Kommunikationszyklus mittels Skalierung eines primären Zyklus oder eines Übertragungszyklus

3.2.17

Geräte-ID

(en: device ID)

Teil der „Geräte-Informationen“ zur Identifikation des Gerätes in Bezug auf einen spezifischen Produkttyp oder ein spezifisches Modell des Typ 24-Feldbusses

3.2.18

Geräte-Informationen

(en: device information)

formatierte und in das Gerät eingebettete Informationen zur Charakterisierung eines Gerätes, die im Wesentlichen aus Daten zur Identifizierung des Gerätemodells und geräteprofilspezifischen Parametern für den Typ-24-Feldbus bestehen

3.2.19

Geräteprofil

(en: device profile)

Zusammenstellung von gerätemodellspezifischen Informationen und Funktionalitäten, um Konsistenz zwischen unterschiedlichen Gerätemodellen unter derselben Art von Geräten herzustellen

3.2.20

zweifacher Transfer

(en: dual transfer)

Transfermodus für das FDC-Protokoll der FAL Typ 24, bei dem ein Sender dieselbe PDU zweimal je Transaktion sendet und ein Empfänger diese PDUs nutzt, um einen Kommunikationsfehler im zyklischen Kommunikationsmodus, z. B. Datenverfälschung oder Datenverlust, zu erkennen und zu beheben

3.2.21

ereignisgesteuerte Kommunikation

(en: event driven communication)

Übertragungsmodus für das Protokoll der unteren Schicht des Typ 24-Feldbusses, bei dem eine Transaktion in Form eines Befehl-Antwort-Austausches aufgrund von Anforderungen des Benutzers erfolgt

Anmerkung 1 zum Begriff: In diesem Modus treten weder der Übertragungszyklus noch der Kommunikationszyklus auf.

3.2.22**Fehler****(en: error)**

anormaler Zustand oder Fehlfunktion bei der Kommunikation oder einer anderen Aktivität

3.2.23**Dienst zur Feldgerätesteuerung****FDC****(en: field device control (FDC) service)**

zeitkritischer Kommunikationsdienst, der Befehlsdaten von fester Länge zur Steuerung eines Feldgerätes verarbeitet und die entsprechenden zurückgemeldeten Antwortdaten im Rahmen einer strengen Beschränkung der Verzögerung oder des Jitters für die zeitliche Kommunikationssteuerung für die FAL Typ 24 handhabt

3.2.24**Protokoll zur Feldgerätesteuerung****(en: field device control (FDC) protocol)**

zeitkritisches Kommunikationsprotokoll, das Befehlsdaten von fester Länge zur Steuerung eines Feldgerätes verarbeitet und die entsprechenden zurückgemeldeten Antwortdaten im Rahmen einer strengen Beschränkung der Verzögerung oder des Jitters für die zeitliche Kommunikationssteuerung für die FAL Typ 24 handhabt

3.2.25**Master****(en: master)**

Klasse oder das entsprechende Instanzobjekt des FDC-Anwendungsdienstelementes (ASE), die/das die Rolle eines Anforderers von Befehlen für die FAL Typ 24 einnimmt

3.2.26**Nachrichtendienst****MSG-Dienst****(en: message service; MSG service)**

Kommunikationsdienst, der Daten von variabler Länge verarbeitet und keine strenge Beschränkung der Antwortzeit festlegt

3.2.27**Überwachung****(en: monitor)**

Klasse oder das entsprechende Instanzobjekt des FDC-ASE, die/das die Rolle eines Wächters über oder eines Subscribers von Befehlen und Antworten zwischen anderen Kommunikationsknoten für die FAL Typ 24 einnimmt

3.2.28**Überwachungs-Slave****(en: monitor slave)**

Variante des Slave-AP-Typs, die sowohl über die Slave-Klasse als auch über die Überwachungsklasse für das FDC-ASE der FAL Typ 24 verfügt

3.2.29**Netzwerkuhr****(en: network clock)**

synchronisierter und periodisch ausgeführter Zähler, über den alle Knoten in einem Netzwerk verfügen und der als Schwingungsquelle des Übertragungszyklus dient

3.2.30**primärer Zyklus****(en: primary cycle)**

Zeitspanne wiederholter Aktivitäten, die vor dem Verbindungsaufbau für das FDC-Protokoll in der FAL Typ 24 mit dem Übertragungszyklus synchronisiert wird

3.2.31

Protokollautomat

(en: **protocol machine**)

Zustandsautomat, der ein Protokoll als Hauptfunktion der Entität auf jeder Schicht umsetzt

3.2.32

Anforderer

(en: **requester**)

Klasse oder das entsprechende Instanzobjekt des MSG-ASE, die/das die Rolle eines Anforderers oder Senders von Befehlen für die FAL Typ 24 einnimmt

3.2.33

Beantworter

(en: **responder**)

Klasse oder das entsprechende Instanzobjekt des MSG-ASE, die/das die Rolle eines Beantworters oder Empfängers von Befehlen für die FAL Typ 24 einnimmt

3.2.34

Antwort

(en: **response**)

PDU, die von einem Beantworter oder einem Slave ausgegeben wird, um einem Anforderer oder einem Master ein Ergebnis oder einen Status auf einen empfangenen Befehl hin mitzuteilen

3.2.35

Dienst

(en: **service**)

Operation oder Prozess, die/den ein Objekt auf Anforderung eines anderen Objekts hin ausführt

3.2.36

einfacher Transfer

(en: **single transfer**)

normaler Transfermodus für das FDC-Protokoll der FAL Typ 24, bei dem ein Sender eine PDU einmal je Transaktion sendet

3.2.37

Slave-AP

(en: **slave AP**)

AP-Typ, der über Slave-Einrichtungen für den FDC-Dienst der FAL Typ 24 verfügt, oder das Gerät, das diesen AP-Typ implementiert

3.2.38

Slave

(en: **slave**)

Klasse oder das entsprechende Instanzobjekt des FDC-ASE, die/das die Rolle eines Beantworters für die FAL Typ 24 einnimmt

3.2.39

Sync-Befehl

(en: **sync command**)

Typ einer Befehls-APDU des FDC-Dienstes der FAL Typ 24, der bei der synchronisierten Zeitsteuerung mit jedem Kommunikationszyklus ausgegeben wird

Anmerkung 1 zum Begriff: Festlegungen dazu, welcher Befehl synchron oder nicht synchron ist, hängen von der Anwendung ab. Diese können in Form eines aufgelisteten Satzes von Befehlen und Antworten oder in Form eines Geräteprofils bereitgestellt werden. Siehe 4.4 und Anhang A.

3.2.40**synchrone Kommunikation****(en: synchronous communication)**

Zustand oder Kommunikationsart für den FDC-Dienst der FAL Typ 24, bei dem/der ein Befehl bei der synchronisierten Zeitsteuerung mit jedem Kommunikationszyklus ausgegeben wird

Anmerkung 1 zum Begriff: In diesem Zustand können sowohl Sync- als auch Async-Befehle ausgegeben werden.

Anmerkung 2 zum Begriff: In diesem Zustand müssen Synchronisationsfehler von APs mithilfe von Maßnahmen des Wächter-Zählers erkannt werden.

3.2.41**Übertragungszyklus****(en: transmission cycle)**

Zeitspanne wiederholter Aktivitäten für die unteren Schichten des Typ 24-Feldbusses, für die alle Slave-Geräte durch das Protokoll der unteren Schicht mit einem C1-Master-Gerät synchronisiert werden

3.2.42**Übertragungsmodus****(en: transmission mode)**

Zustand oder Kommunikationsart für das Protokoll der unteren Schicht des Type 24-Feldbusses; zyklischer Modus, ereignisgesteuerter Modus

3.2.43**virtueller Speicherplatz****(en: virtual memory space)**

großer Datenblock von APOs für die FAL Typ 24, der mit Pseudo-Speicheradressen zur Wahrung der Konsistenz zwischen unterschiedlichen Gerätemodellen gelesen oder geschrieben werden kann

Anmerkung 1 zum Begriff: Der virtuelle Speicherplatz schließt die Geräte-Informationen sowie einen weiteren anbieter-spezifischen Bereich ein. Siehe Anhang B.

3.2.44**Warnung****(en: warning)**

Feldgerätestatus, der durch den Dienst zur Feldgerätesteuerung (FDC) des Typ 24-Feldbusses mitteilt, dass das Gerät ein unbedeutendes oder vorübergehendes Problem erkannt hat, den Normalbetrieb jedoch fortsetzen kann

Anmerkung 1 zum Begriff: Alle Warnungsstatus sind gesperrt und müssen verarbeitet werden, damit sie erlöschen.

Anmerkung 2 zum Begriff: Warnungen sind in drei Gruppen unterteilt: Kommunikationswarnungen, Warnungen bei unzulässigen Befehlen und anwendungsspezifische Warnungen. Genaue Definitionen hängen jedoch von der Implementierung der einzelnen Feldgeräte ab.

– Leerseite –

ICS 25.040.40; 35.100.70; 35.110

English Version

**Industrial communication networks - Fieldbus specifications -
Part 6-24: Application layer protocol specification - Type-24
Elements
(IEC 61158-6-24:2014)**

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 6-24: Spécification du protocole de la
couche application - Éléments de type 24
(CEI 61158-6-24:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-24:
Protokollspezifikation des Application Layer
(Anwendungsschicht) - Typ 24-Elemente
(IEC 61158-6-24:2014)

This European Standard was approved by CENELEC on 2014-09-23. 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, Former Yugoslav Republic of Macedonia, 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.



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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65C/764/FDIS, future edition 1 of IEC 61158-6-24, 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 61158-6-24:2014.

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) 2015-06-23
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-23

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.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-6-24:2014 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 61158-1:2014	NOTE	Harmonized as EN 61158-1:2014 (not modified).
IEC 61784-1	NOTE	Harmonized as EN 61784-1.
IEC 61784-2	NOTE	Harmonized as EN 61784-2.

CONTENTS

INTRODUCTION.....	6
1 Scope	7
1.1 General.....	7
1.2 Specifications	7
1.3 Conformance.....	8
2 Normative references	8
3 Terms, definitions, abbreviations, symbols and conventions	8
3.1 Referenced terms and definitions	8
3.2 Additional terms and definitions	10
3.3 Abbreviations and symbols	15
3.4 Conventions	16
4 Abstract syntax.....	18
4.1 Basic Data types	18
4.2 FAL PDU types.....	20
4.3 Detailed definitions of _FDCService-PDUs	32
4.4 Device profile	51
5 Transfer syntax	52
5.1 Concepts.....	52
5.2 Encode rules	52
6 Structure of FAL protocol state machine.....	58
7 AP-context state machine (APC SM)	60
7.1 Overview	60
7.2 State descriptions.....	61
7.3 Triggering events.....	62
7.4 Action descriptions at state transitions	62
8 FAL service protocol machines (FSPM).....	63
8.1 Overview	63
8.2 Field Deice Control Protocol Machine (FDC PM)	64
8.3 Message Protocol Machine (MSGPM)	88
9 Application relationship protocol machine (ARPM).....	94
9.1 General.....	94
9.2 ARPM for FDC ASE	94
9.3 ARPM for MSG ASE (ARPM-MSG).....	108
10 DLL mapping protocol machine (DMPM).....	109
Annex A (informative) Device profile and FDC command sets	110
Annex B (normative) Virtual memory space and Device Information	111
B.1 Overview	111
B.2 Device Information.....	112
B.2.1 Device identifier area structure	112
B.2.2 Detail specifications of device IDs	112
Annex C (informative) Basic message function	118
Bibliography	119
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	120

Figure 1 – Tree structure of APDU types	21
Figure 2 – Encode of Integer subtypes	53
Figure 3 – Example of transfer of INTEGER value	53
Figure 4 – Encode of Unsigned subtypes	54
Figure 5 – Float32 type encode	54
Figure 6 – Float64 type encode	55
Figure 7 – Bit field definition example with named bits	56
Figure 8 – Bit field definition example with field size	56
Figure 9 – SEQUENCE type encode	57
Figure 10 – Structure of FAL protocol state machines	59
Figure 11 – Statechart diagram of APCSM	61
Figure 12 – Example communication cycle of FDC master AP	65
Figure 13 – Example communication cycle of FDC slave AP	66
Figure 14 – Synchronous command communication in sync state	67
Figure 15 – Asynchronous command communication in sync state	68
Figure 16 – Asynchronous command communication in async state	69
Figure 17 – Event-driven communication	70
Figure 18 – Statechart diagram of FDCPM-M	71
Figure 19 – Statechart diagram of FDCPM-S	77
Figure 20 – Statechart diagram of FDCPM-MN	84
Figure 21 – PDU transmission flow for user message	88
Figure 22 – PDU transmission flow for one-way message	89
Figure 23 – Statechart diagram of MSGPM-RQ	90
Figure 24 – Statechart diagram of MSGPM-RS	92
Figure 25 – Example of single transfer process	94
Figure 26 – Example of dual transfer process	95
Figure 27 – Statechart diagram of ARPM-FDCM	96
Figure 28 – Statechart diagram of ARPM-FDCS	101
Figure 29 – Statechart diagram of ARPM-FDCMN	106
Figure 30 – Statechart diagram of ARPM-MSG	108
Figure B.1 – Memory map of virtual memory space	111
Figure B.2 – Memory map of device ID area	112
Table 1 – State transition descriptions	17
Table 2 – Description of state machine elements	17
Table 3 – Conventions used in state machines	18
Table 4 – Mapping for Protocol State Machines	59
Table 5 – State descriptions of APC SM	61
Table 6 – Trigger event descriptions of APC SM	62
Table 7 – Transitions of APC SM	63
Table 8 – FDC protocol mode	64

Table 9 – State descriptions of FDCPM-M	72
Table 10 – Trigger event descriptions of FDCPM-M	73
Table 11 – Transitions of main SM of FDCPM-M.....	74
Table 12 – Transitions of submachine of FDCPM-M.....	75
Table 13 – State descriptions of FDCPM-S.....	78
Table 14 – Trigger event descriptions of FDCPM-S.....	79
Table 15 – Transitions of main SM of FDCPM-S	80
Table 16 – Transitions of submachine of FDCPM-S	81
Table 17 – State descriptions of FDCPM-MN.....	84
Table 18 – Trigger event descriptions of FDCPM-MN.....	85
Table 19 – Transitions of main SM of FDCPM-MN	85
Table 20 – Transitions of submachine of FDCPM-MN	85
Table 21 – State descriptions of MSGPM-RQ	90
Table 22 – Trigger event descriptions of MSGPM-RQ	91
Table 23 – Transitions of MSGPM-RQ.....	91
Table 24 – State descriptions of MSGPM-RS.....	92
Table 25 – Trigger event descriptions of MSGPM-RS	93
Table 26 – Transitions of MSGPM-RS	93
Table 27 – State descriptions of ARPM-FDCM.....	96
Table 28 – Trigger event descriptions of ARPM-FDCM	98
Table 29 – Transitions of main SM of ARPM-FDCM	99
Table 30 – Transitions of submachine of ARPM-FDCM	100
Table 31 – State descriptions of ARPM-FDCS	101
Table 32 – Trigger event descriptions of ARPM-FDCS	103
Table 33 – Transitions of main SM of ARPM-FDCS	104
Table 34 – Transitions of submachine of ARPM-FDCS.....	105
Table 35 – State descriptions of ARPM-FDCMN	106
Table 36 – Trigger event descriptions of ARPM-FDCMN	107
Table 37 – Transitions of main SM of ARPM-FDCMN.....	107
Table 38 – Transitions of submachine of ARPM-FDCMN.....	107
Table 39 – State descriptions of ARPM-MSG.....	108
Table 40 – Trigger event descriptions of ARPM-MSG	109
Table 41 – Transitions of ARPM-MSG	109
Table A.1 – Example of registered device profiles.....	110
Table A.2 – Example command list of the profile '00'H.....	110
Table B.1 – Specifications of device IDs.....	113
Table C.1 – Example of message command set.....	118

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 61158-1.

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-24: Application layer protocol specification – Type-24 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 24 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 24 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 machines defining the application service behavior visibly between communicating application entities, and
- d) the application relationship state machines defining the communication behavior visibly between communicating application entities.

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

- a) define the representation-on-wire of the service primitives defined in IEC 61158-5-24, and
- b) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 24 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-24.

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 documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-5-24:2014, *Industrial communication networks – Fieldbus specifications – Part 5-24: Application layer service definition – Type 24 elements*

IEC 61158-6 (all parts), *Industrial communication networks – Fieldbus specifications – Part 6: Application layer protocol specification*

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

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

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

ISO/IEC 9899, *Information technology – Programming languages – C*

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

ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

ISO/IEC/IEEE 60559:2011, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

3 Terms, definitions, abbreviations, symbols and conventions

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

3.1 Referenced terms and definitions

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

3.1.1 Terms and definitions from ISO/IEC 7498-1

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

- a) abstract syntax;

- b) application-entity;
- c) application process;
- d) application protocol data unit;
- e) application-process-invocation;
- f) (N)-facility;
- g) (N)-function;
- h) peer-(N)-entities;
- i) presentation context;
- j) real system;
- k) transfer syntax.

3.1.2 Terms and definitions from ISO/IEC 9545

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

- a) application-association;
- b) application-context;
- c) application-entity-invocation;
- d) application-entity-type;
- e) application-service-element.

3.1.3 Terms and definitions from ISO/IEC 8824-1

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

- a) simple type;
- b) component;
- c) component type;
- d) integer type;
- e) bitstring type;
- f) octetstring type;
- g) null type;
- h) sequence type;
- i) sequence of type;
- j) choice type;
- k) IA5String type;
- l) encoding.

3.1.4 Terms and definitions from ISO/IEC 10731

For the purposes of this document, the following terms as defined in ISO/IEC 10731 apply:

- a) OSI-service-primitive; primitive;
- b) OSI-service-provider; provider;
- c) OSI-service-user; user.

3.1.5 Terms and definitions from ISO/IEC 19501

For the purposes of this document, the following terms as defined in ISO/IEC 19501 apply:

- a) event;

- b) state;
- c) state machine;
- d) substate;
- e) submachine;
- f) transition.

3.2 Additional terms and definitions

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

3.2.1

alarm

field device status to tell that the device has detected a fatal problem to be solved and cannot continue normal working, through the field device control (FDC) service of the Type 24 fieldbus

Note 1 to entry: Any alarm statuses are latched and need some operations to be cleared.

Note 2 to entry: Alarms are classified into three groups; communication alarms, illegal-command-related ones, and application specific ones. But concrete definitions are dependent on implementation of each field devices.

3.2.2

application process object

network representation of a specific aspect of an application process (AP), which is modelled as a network accessible object contained within an AP or within another APO

Note 1 to entry: Refer IEC 61158-1, 9.3.4.

3.2.3

application process context

AP context

shared knowledge or a common set of rules, governing communication of FAL application entities (AEs) and describing the permissible collective communications behavior between the AEs that are party to a specific set of application relationships (ARs)

Note 1 to entry: Data within AP context can be specified by the user in advance, by the option selected while the user uses a field bus management (FSM) service to read out the facility of peer AP, by the automatic negotiation function that the FSM system handles, and so on. The method that is to be adopted depends on the specification of each implementation.

3.2.4

application process type

AP type

description of a classification of application processes (APs) in terms of a set of capabilities for FAL of the Type 24 fieldbus

Note 1 to entry: AP types are classified into three, C1 master AP, C2 master AP and slave AP, by their application roles in the fieldbus network.

3.2.5

async command

type of a command application protocol data unit (APDU) of the FDC service of the Type 24 FAL, which can be issued any time after the previous transaction without consideration of synchronization with the communication cycle

Note 1 to entry: Definitions, which command should be async one or not, are dependent on an application. They may be provided as a registered set of commands and responses or a device profiles. See 4.4 and Annex A.

3.2.6

asynchronous communication

state or a way of communication for the FDC service of the Type 24 FAL, in which a command can be issued any time after the previous transaction without consideration of synchronization with the communication cycle

Note 1 to entry: In this state, sync commands cannot be issued, but async commands can.

3.2.7

attribute

information or parameter contained in variable portions of an object

Note 1 to entry: Typically, they provide status information or govern the operation of an object. Attributes may also affect the behavior of an object.

3.2.8

C1 master

AP type that has master facilities for the FDC service of the Type 24 FAL, or the device implementing that AP type

Note 1 to entry: Only one C1 master exists in a network of the Type 24 fieldbus

3.2.9

C2 master

AP type that has only monitor facilities for the FDC service but requester facilities for message (MSG) service of the Type 24 FAL, or the device implementing that AP type

Note 1 to entry: Less than two C2 masters can exist in a network of the Type 24 fieldbus

3.2.10

command

PDU issued by a requester or a master to make a responder or a slave execute some functions

3.2.11

communication

transfer

transmission

- communication: process to exchange information in a formal manner between two or more devices, users, APs or entities
- transfer: process to convey a PDU from a sender to a receiver
- transmission: process to send out and propagate electrical signals or encoded data

3.2.12

communication cycle

period of repetitive activities synchronized with the transmission cycle while the connection establishing for the FDC protocol of the Type 24 FAL

Note 1 to entry: Communication cycle may synchronize with the transmission cycle multiplied by a specified scaling factor.

3.2.13

connection

context or logical binding under specific conditions for the FDC protocol between a master object and a slave object for the Type 24 FAL

3.2.14

cyclic

repetitive in a regular manner

3.2.15

cyclic communication

transmission mode in which request PDUs and response PDUs are exchanged repetitively in the scheduled time slots synchronized with a transmission cycle for the lower layer protocol of the Type 24

Note 1 to entry: In the AL, the communication cycle arises from the transmission cycle in this mode.

3.2.16

cycle scale counter

counter to generate a communication cycle by means of scaling a primary cycle or a transmission cycle

3.2.17

device ID

part of "Device Information" to identify the device for a specific product type or model of the Type 24 fieldbus

3.2.18

device information

formatted and device-embedded information to characterize a device, which mainly consists of data for device model identification and device-profile specific parameters for the Type 24 fieldbus

3.2.19

device profile

collection of device-model-common information and functionality providing consistency between different device models among the same kind of devices

3.2.20

dual transfer

transfer mode for the FDC protocol of the Type 24 FAL, in which a sender sends a same PDU twice a transaction and a receiver uses them to detect and recover a communication error such as data-corruption or data-loss in cyclic communication mode

3.2.21

event driven communication

transmission mode for the lower layer protocol of the Type 24 fieldbus in which a transaction of command-response-exchanging arises as user's demands

Note 1 to entry: Both the transmission cycle and the communication cycle don't arise in this mode.

3.2.22

error

abnormal condition or malfunction for communication or any other activities

3.2.23

field device control

FDC service

time-critical communication service that handles a fixed length command data to control a field device and the corresponding feedback response data in a severe restriction on delay or jitter for the communication timing for the Type 24 FAL

3.2.24

field device protocol

FDC protocol

time-critical communication protocol that handles a fixed length command data to control a field device and the corresponding feedback response data in a severe restriction on delay or jitter f

3.2.25

master

class or its instance object of FDC application service element (ASE) who plays a role of a command requester for the Type 24 FAL

3.2.26

message service

MSG service

communication service that handles the variable length data and not required a severe restriction on response time

3.2.27

monitor

class or its instance object of FDC ASE who plays a role of a watcher or subscriber of commands and response between other communication nodes for the Type 24 FAL

3.2.28

monitor slave

variant of slave AP type who has both slave class and monitor class for FDC ASE of the Type 24 FAL

3.2.29

network clock

synchronized and periodically running counter that each nodes in a same network have, which becomes an oscillation source of the transmission cycle

3.2.30

primary cycle

period of repetitive activities synchronized with the transmission cycle before the connection establishing for the FDC protocol in Type 24 FAL

3.2.31

protocol machine

state machine that realizes a protocol as the main function of the entity in each layer

3.2.32

requester

class or its instance object of MSG ASE who plays a role of a command requester or sender for the Type 24 FAL

3.2.33

responder

class or its instance object of MSG ASE who plays a role of a command responder or receiver for the Type 24 FAL

3.2.34

response

PDU issued by a responder or a slave to inform a result or some status for the received command to a requester or a master

3.2.35

service

operation or process that an object performs upon request from another object

3.2.36

single transfer

normal transfer mode for the FDC protocol of the Type 24 FAL in which a sender sends a PDU once a transaction

3.2.37

slave AP

AP type that has slave facilities for the FDC service of the Type 24 FAL, or the device implementing that AP type

3.2.38

slave

class or its instance object of FDC ASE who plays a role of a responder for the Type 24 FAL

3.2.39

sync command

type of command APDU of the FDC service of the Type 24 FAL, which is issued at the synchronized timing with every communication cycle

Note 1 to entry: The definitions, which command is sync one or not, are dependent on an application. They may be provided as a registered set of commands and responses or a device profiles. See 4.4 and Annex A.

3.2.40

synchronous communication

state or a way of communication for the FDC service of the Type 24 FAL, in which a command is issued at the synchronized timing with every communication cycle

Note 1 to entry: In this state, both sync commands and async ones can be issued.

Note 2 to entry: In this state, an out-of-synchronization error of APs shall be detected by measures of the watchdog counter.

3.2.41

transmission cycle

period of repetitive activities for the lower layers of the Type 24 fieldbus, which of all the slave devices are synchronized with that of a C1 master device by the lower layer protocol

3.2.42

transmission mode

state or a way of transmission for the lower layer protocol of the Type 24 fieldbus; cyclic mode, event driven mode

3.2.43

virtual memory space

large data block of APOs for the Type 24 FAL that can be read and written with pseudo-memory-addresses to provide consistency between different device models

Note 1 to entry: The virtual memory space includes the device Information and other vender specific area. See Annex B.

3.2.44

warning

field device status to tell that the device has detected a slight or passing problem but still working normally through the field device control (FDC) service of the Type 24 fieldbus

Note 1 to entry: Any warning statuses are latched and need to be operated to clear them.

Note 2 to entry: Warnings are classified into three groups, communication warnings, illegal-command-related ones, and application specific ones. But concrete definitions are dependent on a implementation of each field devices.

3.3 Abbreviations and symbols

For the purposes of this document, the following abbreviations and symbols apply.

AE	Application Entity
AL	Application Layer
A-, AL-	Application Layer (as a prefix)
ALME	Application Layer Management Entity
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Invocation
APO	Application Process Object
APC	Application Process Context (as prefix of a protocol for Type 24 fieldbus)
APC SM	Application Process Context State Machine (for Type 24 fieldbus)
AR	Application Relationship
AR ASE	Application Relationship Application Service Element
AREP	Application Relationship End Point
ARPM	Application Relationship Protocol Machine (for Type 24 fieldbus)
ARPM-FDCM	ARPM for Field Device Control service Master (for Type 24 fieldbus)
ARPM-FDCMN	ARPM for Field Device Control service Monitor (for Type 24 fieldbus)
ARPM-FDCS	ARPM for Field Device Control service Slave (for Type 24 fieldbus)
ARPM-MSG	ARPM for Message service (for Type 24 fieldbus)
ASCII	American Standard Code for Information Interchange
ASDU	Application Service Data Unit
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
CMD	Command PDU for FDC service (for Type 24 fieldbus)
Cnf	confirm primitive
DL-	(as a prefix) Data Link-
DLL	Data Link Layer
DLM	Data Link-management
DLPDU	Data Link-Protocol Data Unit
DLSAP	Data Link Service Access Point
DLSDU	Data Link Service Data Unit
DMPM	Data Link Mapping Protocol Machine
E2PROM	Electrically erasable programmable read only memory
FAL	Fieldbus Application Layer
FCS	Frame check sequence
FDC-	Field Device Control (as prefix of a service or a protocol for Type 24 fieldbus)
FDC ASE	Field Device Control Application Service Element (for Type 24 fieldbus)
FDCPM	Field Device Control Protocol Machine (for Type 24 fieldbus)
FDCPM-M	Field Device Control Protocol Machine for Master (for Type 24 fieldbus)
FDCPM-MN	Field Device Control Protocol Machine for Monitor (for Type 24 fieldbus)
FDCPM-S	Field Device Control Protocol Machine for Slave (for Type 24 fieldbus)
FIFO	First In First Out
FSPM	FAL service protocol machine
FSM-	Fieldbus System Management (as prefix of a service for Type 24 fieldbus)
FSM ASE	Fieldbus System Management Application Service Element for Type 24 fieldbus
HMI	Human-machine Interface
I/O	Input/output
ID	Identifier
Ind	indication primitive
LME	Layer Management Entity
Lsb	least significant bit

MAC	Media Access Control
Msb	most significant bit
MSG	Message (as prefix of a service or a protocol for Type 24 fieldbus)
MSG ASE	Message Application Service Element for Type 24 fieldbus
MSGPM	Message Protocol Machine for Type 24 fieldbus
MSGPM-RQ	MSGPM for Requester for Type 24 fieldbus
MSGPM-RS	MSGPM for Responder for Type 24 fieldbus
OSI	Open Systems Interconnection
PM	Protocol machine
PDU	Protocol Data Unit
PhL	Ph-layer
QoS	Quality of Service
RAM	Random access memory
Req	request primitive
Rsp	response primitive
RSP	Response PDU for FDC service (for Type 24 fieldbus)
SAP	Service Access Point
SDN	Send Data with no Acknowledge
SDU	Service Data Unit
SM	State Machine
SMIB	System Management Information Base
UML	Unified Modelling Language

3.4 Conventions

3.4.1 General conventions

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-24. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. In this standard, it's assumed that every attributes are accessible only from the owner's instance object itself directly or through services of the class.

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

3.4.2 PDU data type conventions

The data types of FAL PDUs are defined with the notations of ASN.1.

Character strings that the first character is “_” (low line) are used as data type symbols of FAL PDUs or their fields. While a same string as a PDU type but lacked the first “_” means a PDU instance of the corresponding data type.

Example CMD-PDU means an instance of _CMD-PDU and a following statement is omitted.

```
CMD-PDU _CMD-PDU ::= { value }
```

3.4.3 State machine conventions

The protocol sequences are described by means of State Machines.

In statechart diagrams, states are represented as boxes, and state transitions are shown as arrows. Their conventions are given in ISO/IEC 19501 as Universal Modeling Language (UML).

Names of states and transitions of the state diagram correspond to the names in the textual listing of the state transitions.

The textual listing of the state transitions is structured as Table 1.

The first row contains the name of the transition by an index number.

The second row defines the source state or the current state before the transition.

The third row contains an event and actions. The event is followed by optional arguments in parentheses, "(" and ")", and guard conditions in brackets, "[" and "]", as the first line. And the actions with starting character "/" follow the event line.

The last row contains the target state or the next state after the transition.

If the event occurs and the conditions are fulfilled the transition fires, i.e. the actions are executed and the next state is entered.

Table 1 – State transition descriptions

T#	Source State	Event (arguments) [conditions] / Action	Target state

A meaning of each element in Table 1 is shown in Table 2, based on UML "transition" definition (ISO/IEC 19501).

Table 2 – Description of state machine elements

Description element	Meaning
Source state Target state	Names of the originating state and the target state of transition.
T#	Name or number of the transition.
Event	Name or description of the trigger event that fire the transition.
(arguments)	A parameter value, an expression, or a sequence of those divided by "," for the event. The preceding "(" and the succeeding ")" are not part of the argument list.
[conditions]	Boolean expression, which is true for the transition to be fired. The preceding "[" and the succeeding "]" are not part of the condition.
/ Action	List of assignments and service or function invocations. The action should be atomic. The preceding "/" is not part of the action.

NOTE "(arguments)" and "[condition]" can be omitted if not necessary.

The conventions used in the state machines are shown in Table 3.

Table 3 – Conventions used in state machines

Convention	Meaning
/*-- description --*/	Describes and explains conditions and/or procedure by using normal sentence between "/*--" and "--*/" instead of using the pseudo code notation. Example: /*-- The PM examines an occurrence of any alarms for the corresponding remote device. -- If an alarm is detected, the PM notifies it to the FSM ASE. -- If not, the PM transfers the MSGService-PDU to the DLL. --*/
=	Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event
Axx	Parameter name if 'a' is a letter Example: Identifier = reason; means value of a 'reason' parameter is assigned to a parameter called 'Identifier'
"xxx"	Fixed visible string Example: Identifier = "abc"; means value "abc" is assigned to a parameter named 'Identifier'
Nnn	If all elements are digits, the item represents a numerical constant shown in decimal representation.
0xnn	If all elements nn are digits, the item represents a numerical constant shown in hexadecimal representation.
==	Logical condition to indicate an item on the left is equal to an item on the right
<	Logical condition to indicate an item on the left is less than the item on the right
>	Logical condition to indicate an item on the left is greater than the item on the right
!=	Logical condition to indicate an item on the left is not equal to an item on the right
&&	Logical "AND"
	Logical "OR"
!	Logical "NOT"
+ - * /	Arithmetic operators
;	Separator of expressions

Further notations as defined in C language (ISO/IEC 9899) can be used to describe conditions and actions.

4 Abstract syntax

4.1 Basic Data types

The following data types may be used in the Type 24 FAL.

a) Basic types as simple types of ASN.1:

- i) INTEGER;
- ii) REAL;
- iii) BIT STRING;
- iv) OCTET STRING;
- v) NULL.

b) Specific basic types as subtypes from ASN.1:

- vi) Integer8 ::= INTEGER (-128..127);
- vii) Integer16 ::= INTEGER (-32 768 .. 32 767);
- viii) Integer32 ::= INTEGER (-2 147 483 648 .. 2 147 483 647);
- ix) Integer64 ::= INTEGER ((-1)×'8000 0000 0000 0000'H..'7FFF FFFF FFFF FFFF'H);
- x) Unsigned8 ::= INTEGER (0..'FF'H);
- xi) Unsigned16 ::= INTEGER (0..'FFFF'H);
- xii) Unsigned32 ::= INTEGER (0..'FFFF FFFF'H);
- xiii) Unsigned64 ::= INTEGER (0..'FFFF FFFF FFFF FFFF'H);
- xiv) Float32 ::= 0 | negative infinity | positive infinity
| REAL (WITH COMPONENTS { mantissa ((-1)× c₃₂), base (2), exponent (q₃₂) })
| REAL (WITH COMPONENTS { mantissa (c₃₂), base (2), exponent (q₃₂) })
where,

$$c_{32} ::= \text{REAL} \left(1.. \left(\sum_{i=0}^{23} \frac{1}{2^i} \right) \right) = \text{REAL} (1..(2-2^{-23})), \text{ and}$$

$$q_{32} ::= \text{INTEGER} (-127..127).$$

NOTE 1 The range of the real value is covered from negative infinity to positive infinity. But the representable precision conforms to binary32 or the single precision of ISO/IEC/IEEE 60559. See 5.2.2.

- xv) Float64 ::= 0 | negative infinity | positive infinity
| REAL (WITH COMPONENTS { mantissa ((-1)× c₆₄), base (2), exponent (q₆₄) })
| REAL (WITH COMPONENTS { mantissa (c₆₄), base (2), exponent (q₆₄) })
where,

$$c_{64} ::= \text{REAL} \left(1.. \left(\sum_{i=0}^{52} \frac{1}{2^i} \right) \right) = \text{REAL} (1.. (2-2^{-52})), \text{ and}$$

$$q_{64} ::= \text{INTEGER} (-1\ 023..1\ 023).$$

NOTE 2 The range of the real value is covered from negative infinity to positive infinity. But the representable precision conforms to binary64 or the double precision of ISO/IEC/IEEE 60559. See 5.2.2.

- xvi) IA5String

c) Structure types with component types of ASN.1:

Structure type is defined by a combination of SEQUENCE type, CHOICE type of ASN.1, basic types and subtype elements defined in ASN.1. The FAL PDU described in 4.2 and 4.3 that follows this subclause correspond this Structure type.

- xvii) SEQUENCE
- xviii) CHOICE

d) Array types with a component type of ASN.1:

Array types are a list of any data type defined with SEQUENCE OF type of ASN.1. Typical array types are shown with SEQUENCE OF type, below:

- xix) SEQUENCE OF;
- xx) BitArray ::= SEQUENCE OF (BIT STRING SIZE(8));
- xxi) Array8 ::= SEQUENCE OF Integer8;
- xxii) Array16 ::= SEQUENCE OF Integer16;
- xxiii) Array32 ::= SEQUENCE OF Integer32;
- xxiv) Array64 ::= SEQUENCE OF Integer64.

4.2 FAL PDU types

4.2.1 Top of APDU types: _APDU

The APDU type: _APDU shall consist of two groups; _FDCServicePDU and _MSGServicePDU.

And _FDCServicePDU shall consist of a pair of PDUs; _CMD-PDU and _RSP-PDU, used for FDC commands and its responses respectively.

Furthermore, _MSGServicePDU shall consist of a pair of PDUs; _MSGREQ-PDU and _MSGRSP-PDU, used for MSG requests and its responses respectively.

_APDU	::=	_FDCServicePDU	
			_MSGServicePDU
_FDCServicePDU	::=	_CMD-PDU	-- FDC command PDU type
			_RSP-PDU
			-- FDC response PDU type
_MSGServicePDU	::=	_MSGREQ-PDU	-- MSG request PDU type
			_MSGRSP-PDU
			-- MSG response PDU type

Figure 1 shows the overview of APDU type definitions by the tree structure chart.

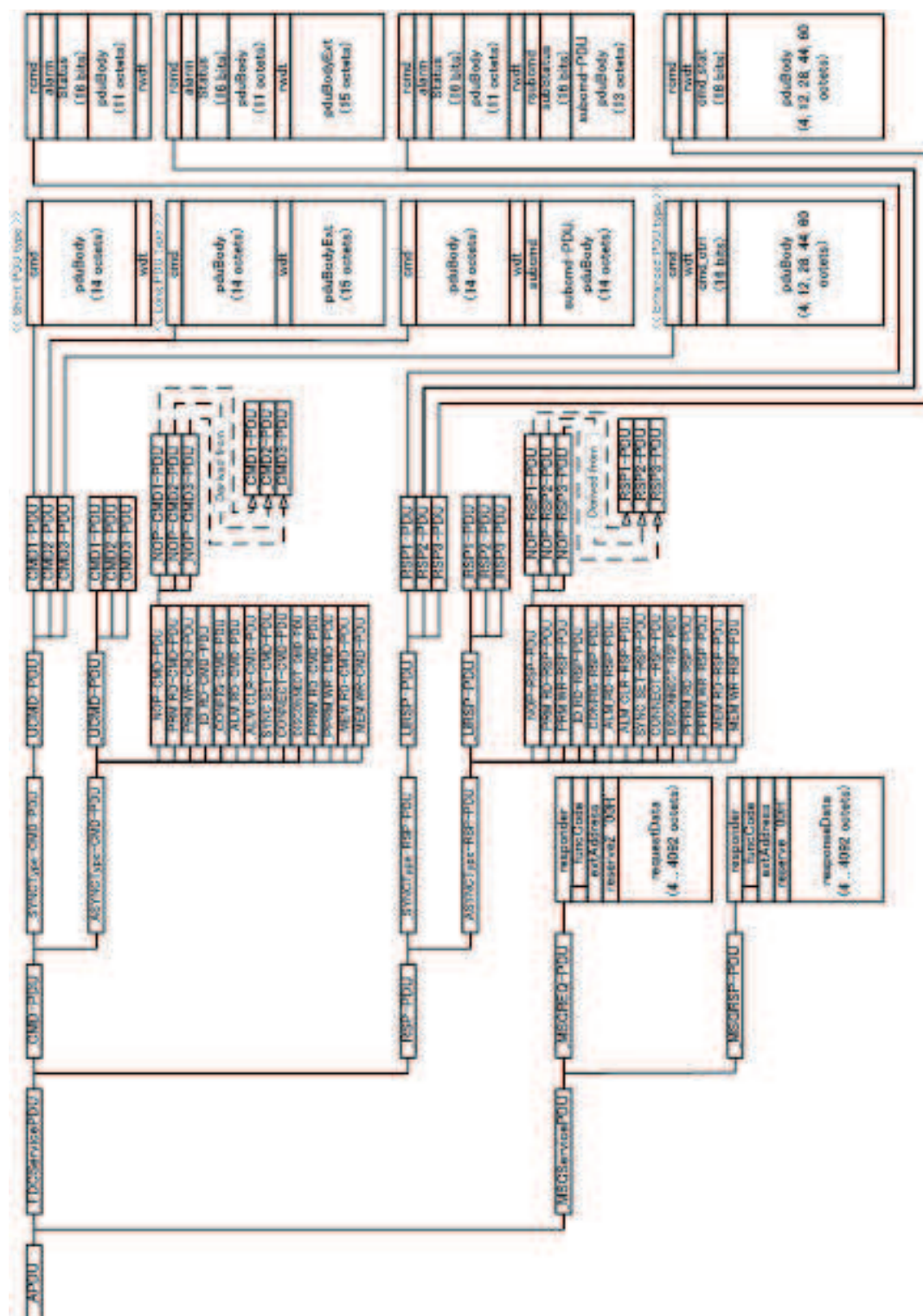


Figure 1 – Tree structure of APDU types

4.2.2 PDUs for field device control service

4.2.2.1 Overview

FDCServicePDU shall be edited with a SDU in the FDC service primitive and be sent via AR. And a FDC ASE receives the PDU from the peer ASE via AR.

The FDC command PDU type: `_CMD_PDU` shall be classified into sync type command (sync command): `_SYNCType-CMD-PDU` and async type command (async command): `_ASYNCType-CMD-PDU`. Both types are the identical PDU formats: `_UCMD-PDU`. Therefore, the FDC PM cannot distinguish their type's PDUs. The user of FDC ASE should know which commands belong to the sync commands or not, and can distinguish the types by using `cmd` field as a key code in the PDU header part.

The async command PDU: `_ASYNCType-CMD_PDU` shall include `_UCMD-PDU` and other fourteen PDU types as common commands, which are also variants of `_UCMD-PDU`.

Derived from the `_UCMD-PDU` and the `_URSP-PDU` explained later, a set of commands and responses can be defined for a specific application, such as motor-drive control, servo control, etc., by FAL users, and the set may be registered as a device profile (see 4.4 and Annex A.). In addition, a FAL user can select any device profiles supported by a target field device, when a connection is established by FDC-Connect service (see 8.2.4.3, and IEC 61158-5-24, 6.4.1.2.3.4).

```
_CMD-PDU                ::= _SYNCType-CMD-PDU
                           | _ASYNCType-CMD-PDU
_SYNCType-CMD-PDU       ::= _UCMD-PDU
_ASYNCType-CMD-PDU      ::= _UCMD-PDU
                           | _NOP-CMD-PDU
                           | _PRM_RD-CMD-PDU
                           | _PRM_WR-CMD-PDU
                           | _ID_RD-CMD-PDU
                           | _CONFIG-CMD-PDU
                           | _ALM_RD-CMD-PDU
                           | _ALM_CLR-CMD-PDU
                           | _SYNC_SET-CMD-PDU
                           | _CONNECT-CMD-PDU
                           | _DISCONNECT-CMD-PDU
                           | _PPRM_RD-CMD-PDU
                           | _PPRM_WR-CMD-PDU
                           | _MEM_RD-CMD-PDU
                           | _MEM_WR-CMD-PDU
_UCMD-PDU                ::= _CMD1-PDU | _CMD2-PDU | _CMD3-PDU
_NOP-CMD-PDU             ::= _NOP-CMD1-PDU | _NOP-CMD2-PDU | _NOP-CMD3-PDU
_PRM_RD-CMD-PDU          ::= _PRM_RD-CMD1-PDU | _PRM_RD-CMD2-PDU | _PRM_RD-CMD3-PDU
_PRM_WR-CMD-PDU          ::= _PRM_WR-CMD1-PDU | _PRM_WR-CMD2-PDU | _PRM_WR-CMD3-PDU
_ID_RD-CMD-PDU           ::= _ID_RD-CMD1-PDU | _ID_RD-CMD2-PDU | _ID_RD-CMD3-PDU
_CONFIG-CMD-PDU          ::= _CONFIG-CMD1-PDU | _CONFIG-CMD2-PDU | _CONFIG-CMD3-PDU
_ALM_RD-CMD-PDU          ::= _ALM_RD-CMD1-PDU | _ALM_RD-CMD2-PDU | _ALM_RD-CMD3-PDU
_ALM_CLR-CMD-PDU         ::= _ALM_CLR-CMD1-PDU | _ALM_CLR-CMD2-PDU | _ALM_CLR-CMD3-PDU
_SYNC_SET-CMD-PDU        ::= _SYNC_SET-CMD1-PDU
```

	_SYNC_SET-CMD2-PDU
	_SYNC_SET-CMD3-PDU
_CONNECT-CMD-PDU	::= _CONNECT-CMD1-PDU
	_CONNECT-CMD2-PDU
	_CONNECT-CMD3-PDU
_DISCONNECT-CMD-PDU	::= _DISCONNECT-CMD1-PDU
	_DISCONNECT-CMD2-PDU
	_DISCONNECT-CMD3-PDU
_PPRM_RD-CMD-PDU	::= _PPRM_RD-CMD1-PDU _PPRM_RD-CMD2-PDU _PPRM_RD-CMD3-PDU
_PPRM_WR-CMD-PDU	::= _PPRM_WR-CMD1-PDU _PPRM_WR-CMD2-PDU _PPRM_WR-CMD3-PDU
_MEM_RD-CMD-PDU	::= _MEM_RD-CMD3-PDU
_MEM_WR-CMD-PDU	::= _MEM_WR-CMD3-PDU

The FDC response PDU type: `_RSP_PDU` shall be also classified into sync type response (sync response): `_SYNCType-RSP-PDU` and async type response (async response): `_ASYNCType-RSP-PDU`. Both types shall be derived from the identical PDU formats: `_URSP-PDU`. Therefore, the FDC PM cannot distinguish their type's PDUs. The user of FDC ASE should know which responses belong to the sync responses or not, and can distinguish the types by using `rcmd` field as a key code in the PDU header part. Moreover, this prefix code of a response PDU corresponds to one of a command PDU, since a response shall be exchanged for a command.

The async type response PDU: `_ASYNCType-RSP_PDU` shall include `_URSP-PDU` and other fourteen PDU types as common responses to any application, which are also variants of `_URSP-PDU`.

<code>_RSP-PDU</code>	::= <code>_SYNCType-RSP-PDU</code>
	<code>_ASYNCType-RSP-PDU</code>
<code>_SYNCType-RSP-PDU</code>	::= <code>_URSP-PDU</code>
<code>_ASYNCType-RSP-PDU</code>	::= <code>_URSP-PDU</code>
	<code>_NOP-RSP-PDU</code>
	<code>_PRM_RD-RSP-PDU</code>
	<code>_PRM_WR-RSP-PDU</code>
	<code>_ID_RD-RSP-PDU</code>
	<code>_CONFIG-RSP-PDU</code>
	<code>_ALM_RD-RSP-PDU</code>
	<code>_ALM_CLR-RSP-PDU</code>
	<code>_SYNC_SET-RSP-PDU</code>
	<code>_CONNECT-RSP-PDU</code>
	<code>_DISCONNECT-RSP-PDU</code>
	<code>_PPRM_RD-RSP-PDU</code>
	<code>_PPRM_WR-RSP-PDU</code>
	<code>_MEM_RD-RSP-PDU</code>
	<code>_MEM_WR-RSP-PDU</code>
<code>_URSP-PDU</code>	::= <code>_RSP1-PDU</code> <code>_RSP2-PDU</code> <code>_RSP3-PDU</code>
<code>_NOP-RSP-PDU</code>	::= <code>_NOP-RSP1-PDU</code> <code>_NOP-RSP2-PDU</code> <code>_NOP-RSP3-PDU</code>
<code>_PRM_RD-RSP-PDU</code>	::= <code>_PRM_RD-RSP1-PDU</code> <code>_PRM_RD-RSP2-PDU</code> <code>_PRM_RD-RSP3-PDU</code>
<code>_PRM_WR-RSP-PDU</code>	::= <code>_PRM_WR-RSP1-PDU</code> <code>_PRM_WR-RSP2-PDU</code> <code>_PRM_WR-RSP3-PDU</code>
<code>_ID_RD-RSP-PDU</code>	::= <code>_ID_RD-RSP1-PDU</code> <code>_ID_RD-RSP2-PDU</code> <code>_ID_RD-RSP3-PDU</code>

```

_CONFIG-RSP-PDU      ::= _CONFIG-RSP1-PDU | _CONFIG-RSP2-PDU | _CONFIG-RSP3-PDU
_ALM_RD-RSP-PDU      ::= _ALM_RD-RSP1-PDU | _ALM_RD-RSP2-PDU | _ALM_RD-RSP3-PDU
_ALM_CLR-RSP-PDU      ::= _ALM_CLR-RSP1-PDU | _ALM_CLR-RSP2-PDU | _ALM_CLR-RSP3-PDU
_SYNC_SET-RSP-PDU     ::= _SYNC_SET-RSP1-PDU
                        | _SYNC_SET-RSP2-PDU
                        | _SYNC_SET-RSP3-PDU
_CONNECT-RSP-PDU      ::= _CONNECT-RSP1-PDU
                        | _CONNECT-RSP2-PDU
                        | _CONNECT-RSP3-PDU
_DISCONNECT-RSP-PDU    ::= _DISCONNECT-RSP1-PDU
                        | _DISCONNECT-RSP2-PDU
                        | _DISCONNECT-RSP3-PDU
_PPRM_RD-RSP-PDU      ::= _PPRM_RD-RSP1-PDU | _PPRM_RD-RSP2-PDU | _PPRM_RD-RSP3-PDU
_PPRM_WR-RSP-PDU      ::= _PPRM_WR-RSP1-PDU | _PPRM_WR-RSP2-PDU | _PPRM_WR-RSP3-PDU
_MEM_RD-RSP-PDU       ::= _MEM_RD-RSP3-PDU
_MEM_WR-RSP-PDU       ::= _MEM_WR-RSP3-PDU

```

4.2.2.2 Common component of PDU

4.2.2.2.1 FDC command PDU

_UCMD-PDU may have three types, short type: _CMD1-PDU, long type: _CMD2-PDU, and enhanced type: _CMD3-PDU. Both _CMD1-PDU and _CMD2-PDU shall have same header fields, but _CMD3-PDU shall have enhanced one.

The pduBody length of _CMD1-PDU shall be 14 octets when it encoded, and so as for _CMD2-PDU. This part is called main-command, and _CMD2-PDU shall have another extended pduBodyExt field too, and can piggyback an additional command PDU called sub-command: _SUBCMD-PDU on that field.

The pduBody length of _CMD3-PDU shall be able to be select from 4, 12, 28, 44, and 60 octets, but need not be switched on the fly.

```

_CMD1-PDU      ::= SEQUENCE{                                     -- short PDU type
    cmd      _CMDCode,
    pduBody  OCTET STRING SIZE (14) ,
    wdt      _WDT }

_CMD2-PDU      ::= SEQUENCE{                                     -- long PDU type
    maincmd-PDU  COMPONENTS OF _CMD1-PDU,                       -- main-command
    CHOICE { subcmd-PDU _SUBCMD-PDU,                             -- sub-command
        pduBodyExt  OCTET STRING SIZE (15) } }

_SUBCMD-PDU    ::= _USUBCMD-PDU SEQUENCE{                       -- sub command
                                                                PDU type
    subcmd      _SubCMD,
    pduBody     OCTET STRING SIZE (14)}

    | _NOP-SUBCMD-PDU
    | _PRM_RD-SUBCMD-PDU
    | _PRM_WR-SUBCMD-PDU
    | _ALM_RD-SUBCMD-PDU
    | _PPRM_RD-SUBCMD-PDU

```

```

| _PPRM_WR-SUBCMD-PDU
_CMD3-PDU ::= SEQUENCE{
    cmd      _CMDCode,
    wdt      _WDT,
    cmd_ctrl _CMD_CTRL,
    pduBody  OCTET STRING SIZE (4|12|28|44|60) }
-- enhanced PDU
type

```

a) cmd field

cmd

data field which contains the code to identify contents of each command

This field shall be coded as _CMDCode, subtype of Unsigned8, with the following value range. The predefined common commands are shown as _PrimaryCMD below.

- '00'H to '1F'H: used or reserved for common commands;
- '20'H to 'BF'H: reserved for application specific commands;
- 'C0'H to 'FF'H: reserved for vendors.

```

_CMDCode ::= Unsigned8
_PrimaryCMD ::= _CMDCode {
    nop ('00'H),
    prm_rd ('01'H), prm_wr ('02'H),
    id_rd ('03'H),
    config ('04'H),
    alm_rd ('05'H), alm_clr ('06'H),
    sync_set ('0D'H),
    connect ('0E'H),
    disconnect ('0F'H),
    pprm_rd ('1B'H), pprm_wr ('1C'H),
    mem_rd ('1D'H), mem_wr ('1E'H) }

```

b) wdt field

wdt

watchdog counter field

This field shall be used for the slave to detect out-of synchronous activity or the WDT error of the corresponding master. The master shall count the wdt field of sending PDUs every communication cycle in case of the synchronous communication state. In addition, the slave shall examine the field value of receiving PDUs every communication cycle to detect the wdt error.

The Data type is _WDT. It shall have two sub fields, mn and sn.

```

_WDT ::= SEQUENCE { mn BIT STRING SIZE (4),
                    sn BIT STRING SIZE (4)}

```

mn (Master count value)

The value range shall be from 0 to 15.

The mn field in a next CMD-PDU to be sent shall be counted up by one. An FDC master shall do that every communication cycle. On the other hand, when an FDC slave receives the PDU, it shall examine the counter every cycle. And, if the counter isn't match the last mn value plus one, it shall alert the watchdog counter error.

sn (Slave count value)

The value range shall be from 0 to 15.

The sn field in a next CMD-PDU to be sent shall be set the same value of the rsn field in the last received RSP-PDU. An FDC master shall copy it from the last RSP-PDU every communication cycle.

c) subcmd field

subcmd

data field which contains the code to identify contents of each sub-command

The Data type is `_SubCMD`. The code definition for the sub-commands shall be same as main cmd field in principle, but some codes may be restricted to use, if they have any difficulties in processing parallel commands. The predefined common sub-commands are shown as `_PrimarySubCMD` below.

```
_SubCMD          ::= Unsigned8
_PrimarySubCMD    ::= _SubCMD {
    nop ('00'H),
    prm_rd ('01'H), prm_wr ('02'H),
    alm_rd ('05'H),
    pprm_rd ('1B'H), pprm_wr ('1C'H) }
```

d) cmd_ctrl field

cmd_ctrl

set of control bits that is independent of any command transactions, from FDC master to FDC slave

The Data type is `_CMD_CTRL`. It shall have two meaningful sub-fields, `alm_clr` and `cmd_id`.

```
_CMD_CTRL        ::= SEQUENCE { reserve1 BIT STRING SIZE (3),
    alm_clr BIT STRING SIZE (1),
    reserve2 BIT STRING SIZE (2),
    cmd_id BIT STRING SIZE (2),
    reserve3 BIT STRING SIZE (8) }
```

alm_clr

This field shall be used to command to clear alarm/warning status of the slave device.

value 0: to execute nothing

value changes to 1: to execute the status clear

cmd_id

When the master device continuously issues identical contents of a CMD-PDU to the slave, it can be used to make the slave device recognizing the command as another command.

The value range shall be from 0 to 3.

The FDC master may use the `cmd_id` to have a FDC slave acknowledge that the command is a new command when the master sends the same command repeatedly to the slave.

Since the slave shall respond the echo of the `cmd_id` of every command through RSP-PDU.`cmd_stat.rcmd_id` field explained later, the master can also judge the command for which the slave station sent the response.

Use of the cmd_id is not mandatory for the master.

While RSP-PDU.cmd_stat.cmdRdy = 0 in response PDUs, the slave shall disregard commands that have a different cmd_id and continues to execute the command which has been already accepted.

4.2.2.2.2 FDC response PDU

_URSP-PDU may have three types, short type: _RSP1-PDU, long type: _RSP2-PDU, and enhanced type: _RSP3-PDU. Both _RSP1-PDU and _RSP2-PDU shall have common header fields, but _RSP3-PDU shall have another enhanced one.

The pduBody length of _RSP-PDU shall be 11 octets, and so as for _RSP2-PDU. This is called a main response or a response of the main command, and _RSP2-PDU shall have another extended pduBodyExt field too, and can piggyback an additional command PDU called sub-response or a response of a sub command: _SUBRS-PDU on that field.

The pduBody length of _RSP3-PDU shall be able to be select from 4, 12, 28, 44, and 60 octets, but need not be switched on the fly.

```

_RSP1-PDU ::= SEQUENCE{                                     -- short PDU type
    rcmd      _CMDCode,
    alarm     _ALARM,
    status    _STATUS,
    pduBody   OCTET STRING SIZE (11),
    rwdt      _RWDT}

_RSP2-PDU ::= SEQUENCE{                                     -- long PDU type
    rsp1-PDU COMPONENTS OF _RSP1-PDU,                     -- main response
    CHOICE { subrsp-PDU _SUBRS-PDU,                       -- sub response
        pduBodyExt OCTET STRING SIZE (15) } }

_SUBRSP-PDU ::= _USUBRSP-PDU SEQUENCE{                   -- sub response
    rsubcmd   _SubCMD,                                     PDU type
    subStatus _SUBSTATUS,
    pduBody   OCTET STRING SIZE (13)}

| _NOP-SUBRSP-PDU
| _PRM_RD-SUBRSP-PDU
| _PRM_WR-SUBRSP-PDU
| _ALM_RD-SUBRSP-PDU
| _PPRM_RD-SUBRSP-PDU
| _PPRM_WR-SUBRSP-PDU

_RSP3-PDU ::= SEQUENCE{                                     -- enhanced PDU
    rcmd      _CMDCode,                                     type
    rwdt      _RWDT,
    cmd_stat  _CMD_STAT,
    pduBody   OCTET STRING SIZE (4|12|28|44|60) }

```

a) rcmd field

rcmd

data field which contains the same code as a received and processing command

So, this field shall be coded as _CMDCode. See 4.2.2.2.1.

b) alarm field

alarm

This field shall contain the alarm code.

The Data type is `_ALARM`. The value range shall be from 0 to 255.

`_ALARM` ::= Unsigned8

c) status field

status

This field shall indicate the status of the slave device.

The Data type is `_STATUS`. It shall have three meaningful sub-fields as follows.

```
_STATUS ::= SEQUENCE { alarm BIT STRING SIZE (1),  
warning BIT STRING SIZE (1),  
cmdRdy BIT STRING SIZE (1),  
reserve BIT STRING SIZE (13) }
```

alarm

This bit field shall indicate the alarm status in the slave device.

- 0: No alarm,
- 1: any alarms occur.

warning

This bit field shall indicate the warning status in the slave device.

- 0: No warning,
- 1: any warnings occur.

cmdRdy

This bit field shall indicate the command progress status of the FDC slave.

The slave device shall keep the bit `cmdRdy` 0, while it is processing the command. The slave device shall change the bit from 0 to 1, when it has completed the processing.

To notice the completion of the specific command the FDC master shall not only examine the bit, but also collate some other RSP-PDU fields with the corresponding CMD-PDU fields to discriminate from the preceding transaction.

When the retention time of `cmdRdy` = 0 is expired, the master shall raise a command time out error. The timer value depends on each product specification for slave devices.

A change of this bit status shall be independent of the alarm or warning status.

- 0: busy with command execution in progress
- 1: ready for new command

d) `rwdt` field

rwdt

watchdog counter field

This field shall be used for the master to detect out-of synchronous activity or the WDT error of the corresponding slave. The slave shall count and update the `rwdt` field of sending PDUs every communication cycle in case of the synchronous communication state. In addition, the master shall examine the field value of receiving PDUs every communication cycle to detect the `rwdt` error.

The Data type is `_RWDT`. It shall have two sub fields, `rmn` and `rsn`.

```
_RWDT ::= SEQUENCE {
    rmn BIT STRING SIZE (4),
    rsn BIT STRING SIZE (4)}
```

rmn (Master count value)

The value range shall be from 0 to 15.

The `rmn` field in a next RSP-PDU to be sent shall be set the same value of the `mn` field in the last received CMD-PDU. An FDC slave shall copy it from the CMD-PDU every communication cycle.

rsn (Slave count value)

The value range shall be from 0 to 15.

The `rsn` field in a next RSP-PDU to be sent shall be counted up by one. An FDC slave shall do that every communication cycle. On the other hand, when an FDC master receives the PDU, it shall examine the counter every cycle. And, if the counter isn't match the last `rsn` value plus one, it shall alert the watchdog counter error.

e) `rsubcmd`

rsubcmd

data field which contains the same code as the received and processing sub-command

This field shall be coded as `_SubCMD`. See 4.2.2.2.1.

f) `subStatus`

subStatus

This field shall indicate the status of the slave station in the sub response PDU part.

The Data type is `_SubSTATUS`. This field shall be coded same as status field.

```
_SUBSTATUS ::= _STATUS
```

g) `cmd_stat` field

cmd_stat

This field shall indicate the status of the slave device.

The Data type is `_CMD_STAT`. It shall have seven sub fields as follows.

```
_CMD_STAT ::= SEQUENCE {
    d_alm BIT STRING SIZE (1),
    d_war BIT STRING SIZE (1),
    cmdRdy BIT STRING SIZE (1),
    alm_clr_cmp BIT STRING SIZE (1),
    reserve BIT STRING SIZE (2),
    rcmd_id BIT STRING SIZE (2),
    cmd_alm BIT STRING SIZE (4),
    comm_alm BIT STRING SIZE (4) }
```

d_alm

This bit field shall indicate occurrence status of device-specific alarm in the slave device.

0: the slave is not in device specific alarm status;