SLOVENSKI STANDARD
oSIST prEN 15969-1:2009

01-november-2009

Rezervoarji za prevoz nevarnih snovi - Digitalni vmesniki za prenos podatkov med vozilom-cisterno in nepremiļ no inġtalacijo - 1. del: Specifikacija protokola - Upravljanje, merjenje (meritve) in zajem podatkov

Tanks for transport of dangerous goods - Digital interface for the data transfer between tank vehicle and with stationary facilities - Part 1: Protocol specification - Control, measurement and event data

Tanks für die Beförderung gefährlicher Güter - Digitale Schnittstelle für den Datenaustausch zwischen Tankfahrzeugen und stationären Einrichtungen - Teil 1: Protokollspezifikation - Steuerungs-, Mess- und Ereignisdaten

Citernes pour le transport de matières dangereuses - Interface numérique pour le transfert de données entre le véhicule-citerne et les installations fixes - Partie 1 : Spécification du protocole - Contrôle, mesurage et données d'évènements

Ta slovenski standard je istoveten z: **prEN 15969-1**

ICS:

23.020.10	Nepremiļ ne posode in rezervoarji	Stationary containers and tanks
35.240.60	Uporabniške rešitve IT v transportu in trgovini	IT applications in transport and trade

oSIST prEN 15969-1:2009 **en,de**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
prEN 15969-1

July 2009

ICS 35.240.60

English Version

Tanks for transport of dangerous goods - Digital interface for the data transfer between tank vehicle and with stationary facilities - Part 1: Protocol specification - Control, measurement and event data

Citernes pour le transport de matières dangereuses - Interface numérique pour le transfert de données entre le véhicule-citerne et les installations fixes - Partie 1 : Spécification du protocole - Contrôle, mesurage et données d'évènements

Tanks für die Beförderung gefährlicher Güter - Digitale Schnittstelle für den Datenaustausch zwischen Tankfahrzeugen und stationären Einrichtungen - Teil 1: Protokollspezifikation - Steuerungs-, Mess- und Ereignisdaten

This draft European Standard is submitted to CEN members for enquiry. It has been drawn up by the Technical Committee CEN/TC 296.

If this draft becomes a European Standard, CEN 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.

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

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

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning : This document is not a European Standard. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a European Standard.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

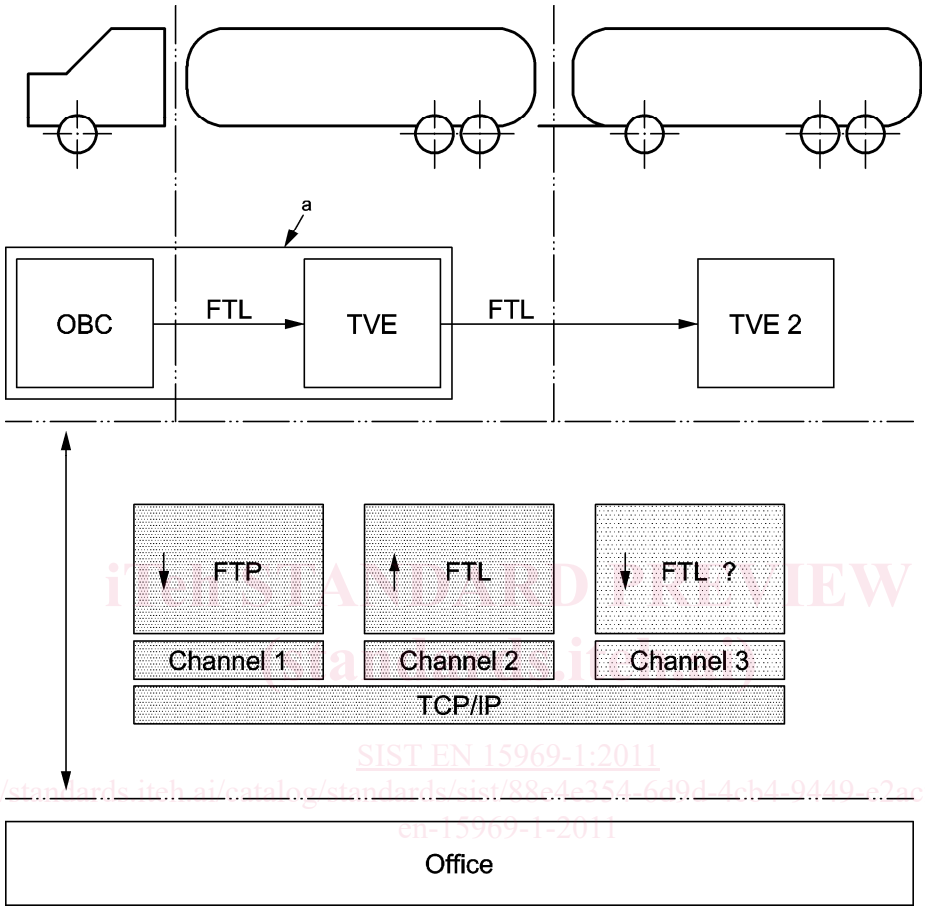
Foreword

This document (prEN 15969-1:2009) has been prepared by Technical Committee CEN/TC 296 “Tanks for transport of dangerous goods”, the secretariat of which is held by AFNOR.

This document is currently submitted to the CEN Enquiry.

Introduction

FTL is an acronym for **F**uel **T**ruck **L**ink, the interface between electronic system(s) on board of a tank vehicle (tank-vehicle-equipment) and any external computer, e.g. an on-board-computer installed in the driver's cabin, for illustration see Figure 1.



Key

→ direction of communication (client → server)
a may be either two independent units or one single unit which incorporates both functions OBC and TVE

Figure 1

1 Scope

This standard specifies data protocols and data format for the interfaces between electronic equipment (TVE), on-board computer (OBC) of the tank vehicle and stationary equipment for all interconnecting communication paths.

This document specifies the basic protocol FTL used in the communication (basic protocol layer), the format and structure of FTL-data to be transmitted (data protocol layer) and describes the content of the FTL-data.

NOTE This data protocol may be used for other application e.g. between stationary tank equipment and offices.

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.

EN 13922, *Tanks for transport of dangerous goods — Service equipment for tanks — Overfill prevention systems for liquid fuels*

EN 14116, *Tanks for transport of dangerous goods — Digital interface for the product recognition device*

EN 15208:2006, *Tanks for transport of dangerous goods — Sealed parcel delivery systems — Working principles and interface specifications*

ISO 639-1, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

ISO 8859-15, *Information technology — 8-bit single-byte coded graphic character sets — Part 15: Latin alphabet No. 9*

DIN 51757, *Testing of mineral oils and related materials; determination of density*

3 Terms and definitions, abbreviations and conventions

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

3.1 Abbreviations

ACK	acknowledge controlframe
ADF	additional dataframe
ASCII	American Standard Code for Information Interchange
CAN	cancel controlframe
CRC	cyclic redundancy checksum
CSV	comma separated variable record
COP	crossover prevention
EOR	end of record dataframe
EOT	end of transmission dataframe
FTL	fuel-truck-link name of the interface
FTP	file transfer protocol
L_FILE	log file

LH_FILE	log file header
NAK	not acknowledge controlframe
OBC	on-board-computer
NOTE	One party in the FTL-communication (the client).
PID	product identification device according to EN 14116
SYN	synchronisation controlframe
TEF	CRC transmission error controlframe
TVE	tank-vehicle-equipment
	NOTE One party in the FTL-communication (the server).
OpCode	operation code

3.2 Terms and definitions

3.2.1
downgrade
intentional loading and discharge of a higher grade product (substance) into a lower grade product of the same group

3.2.2
answer time
time between last frame character transmitted from OBC (client) and first character frame received from TVE (server)

3.2.3
array
collection of elements which have a same structure and able to be accessed individually by means of an index

3.2.4
client
responsible for initiation and control of data exchange

3.2.5
field
element of a datagram delimited by separators

3.2.6
frame
data packet with variable length and defined structure

3.2.7
list
type of variables consisting of a number of records

3.2.8
MaxFrameSize
maximum number of characters in a frame

3.2.9
node
part of an address of a variable

3.2.10
graphic character
according to ISO 8859-15

3.2.11
record
ordered set of fields, stored contiguously

3.2.12
server
program which provides service to client programs

3.2.13
subnode
subpart of an address of a variable

3.2.14
datagram
instruction or answer to an instruction, which comprises an OpCode and operand

3.2.14
transaction
complete request-answer-cycle

3.2.15
type identifier
character code for the frame type

3.3 Conventions

3.3.1 Syntax conventions

When describing the syntax of e.g. a datagram, some parts are required.

Every abstract part shall get a name, which is encapsulated by “<” and “>”. Optional arguments are additionally encapsulated in square brackets.

EXAMPLE <field>[,<value>]

<field> has always to be given (required). <value> is optional, but when given, a comma shall precede.

3.3.2 Presentation of communication exchange

In this document several examples can be found, demonstrating the flow of communication.
To illustrate the direction, data sent by the TVE (server) is shown indented.

EXAMPLE

client request 1
    server response 1
    server response 2
    server response 3
client request 2

This means, that the command "client request n" shall be transmitted by the OBC, whereas the lines "server response n" were transmitted by the TVE.

3.3.3 Numbers

Numbers may either be coded in decimal format (e.g. 12) or in hexadecimal format (e.g. 1Bh). In the latter case, the number shall followed by the character “h“.

4 Hardware interface

Communication shall take place between exactly two parties (point-to-point) the TVE and OBC.

For communication an asynchronous line shall be used. The OBC and TVE start up and default settings shall apply 9600 baud, 8 data bits, 1 stop bit and no parity. It shall be ensured that the two communication points are of the same standard.

The TVE may optionally support other baud rates (switching and switching back see 7.3.6).

5 Basic protocol layer

5.1 FTL-frame (frame)

The FTL-frame shall be according to Figure 2.

Frame (7 to MaxFrameSize character)

Start-Flag	Type	Content (optional)	End-Flag	Checksum
1 char	1 char	0 to MaxFrameSize - 7 char	1 char	4 char

Figure 2

A frame shall have the following minimum requirements:

- always starts with a Start—Flag
- always followed by type identifier

- 1 End-Flag
- 4 character Checksum (valid or invalid)
- frame length limited to MaxFrameSize

Frames which do not fulfil these requirements shall be ignored and not answered. Whenever a Start—Flag is received a new frame started. All possibly received character before a Start-Flag shall be ignored. Only the remaining frame shall be the base for a response.

Each device which takes part off the FTL-protocol shall be able to receive at least one complete frame which may have MaxFrameSize characters. Therefore all kind of low level handshake shall be omitted. A frame shall be answered even if it contains an invalid checksum or incorrect characters (see 5.2).

If the type identifier in a frame is unknown a NAK shall be sent.

MaxFrameSize

The MaxFrameSize shall be 255 characters.

Start—Flag

The ASCII code 02h (start of text <STX>) shall be used as the Start-Flag.

Type identifier

The type identifier shall be according to Table 1.

Content

The content may be empty or shall contain up to MaxFrameSize—7 characters.
All characters in the content shall be printable characters.

End-Flag

The ASCII code 03h (End of Text <ETX>) shall be used as the End—Flag.

Checksum

The Checksum <CRC> verifies the integrity of a frame. It covers all characters from Start—Flag to End—Flag including these flags. A CRC16 (16 bit) value in hexadecimal format (always 4 characters long) is used and shall consist of the printable ASCII character “0”..”9” or “A”..”F” (Example: the value 1AC9h shall be sent with 4 ASCII character “1AC9”). The algorithm for the calculation is described in 5.4.

5.2 Frame flow (handshake)

The character following the Start-Flag describes the type of the frame. Depending on this character 2 different framegroups with different frametypes shall be distinguished according to Table 1.

Table 1 — Frame groups and frame types

Framegroup	Frametype	Abbreviation	Additional fields	Type identifier	
				client to server	server to client
Dataframe	end of record frame	EOR	data	R	r
	additional dataframe following frame	ADF	data	L	l
	end of transmission frame	EOT	data	E	e
Controlframe	acknowledge frame	ACK	no	A	a
	synchronisation/wait frame	SYN	no	S	s
	cancel frame	CAN	no	C	c
	CRC transmission error frame	TEF	no	T	t
	not acknowledge frame	NAK	NAK-ID according to Table 17	N	n

To distinguish the direction of data (client to server or server to client) upper and lower case type character shall be used.

Every communication shall start with a dataframe.

Every dataframe shall be acknowledged. An EOT—frame can be answered by either a controlframe or acknowledged by another dataframe. All other dataframes shall be answered by controlframes only.

Examples of frame flows:

- Transaction that requires only one datagram in either direction, each fitting into a single frame, see Figure 3.

Figure 3

- Transaction that requires more than one datagram (e.g multi record transfer), EOR—frames shall indicate that additional datagrames will follow. An EOT—frame shall be the last dataframe of the transaction, see Figure 4.

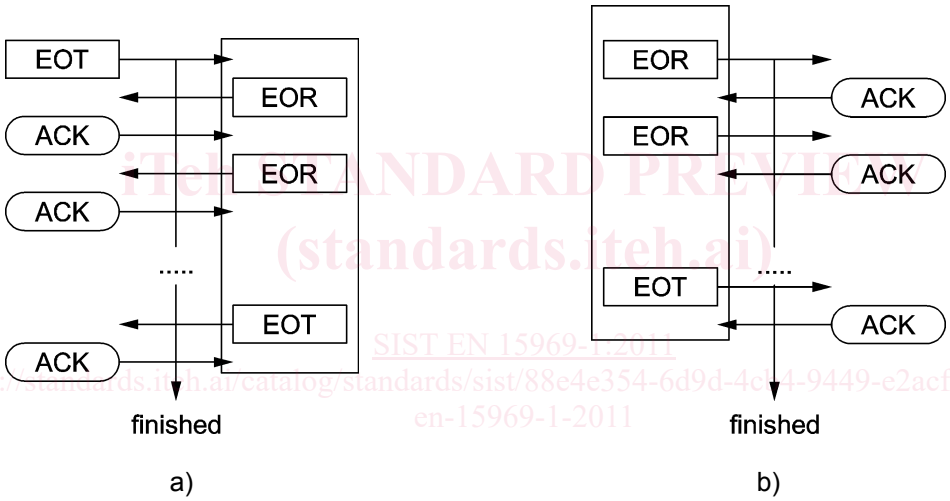


Figure 4

- Datagrams that requires more than one frame, because MaxFrameSize is too small to hold a complete datagram, it shall be split into one or more ADF—frames and an EOT—frame, see Figure 5.

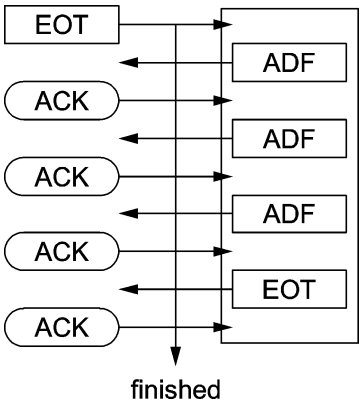
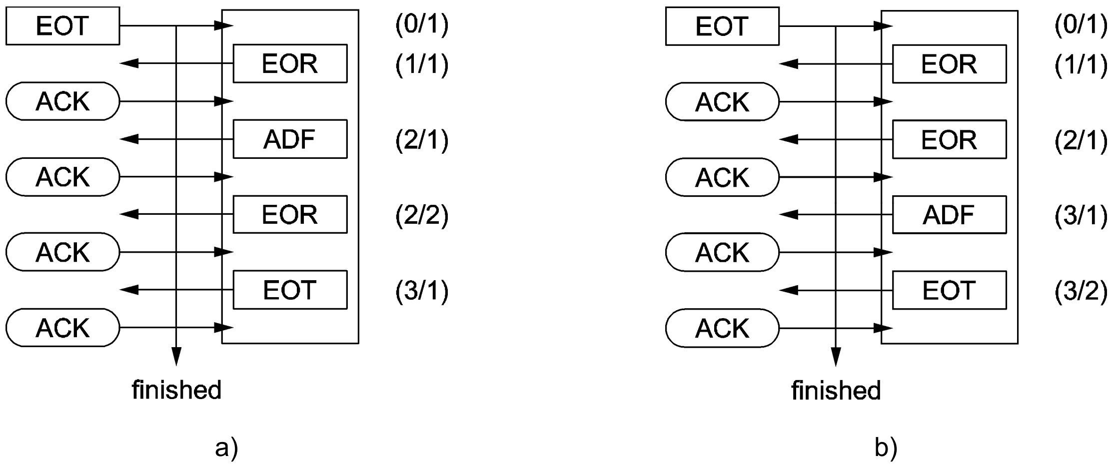


Figure 5

- The preceding examples may be combined as in Figure 6:



NOTE The numbers in the brackets [m/n] have the following meaning:

m datagram sequence within transaction

n indicates the sequence number of the dataframe within the datagram (if splitting is required)

Figure 6

TEF—frame

In case of a CRC error all frametypes shall be answered with a TEF— frame. The frame shall then be repeated, see Figure 7.

Key

1 controlframe is repeated

2 dataframe is repeated

Figure 7

*S*YN-frame

A SYN—frame is not a final acknowledge. It notifies a busy status while preparing the answer to prevent a timeout.

Multiple SYN—frame are possible but always a final acknowledge shall follow, see Figure 8. t_W shall be between 30 % and 90 % of the maximum answer time Rt, see 5.3.

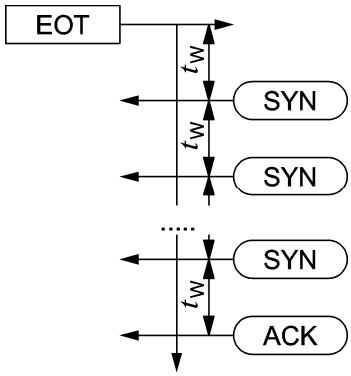


Figure 8

CAN—frame

Transactions can be cancelled if procedures with higher priorities have to be serviced from the higher application layer. Then a CAN—frame shall be sent. Regarding the data it shall act like an ACK—frame but shall terminate the current datagram, see Figure 9.

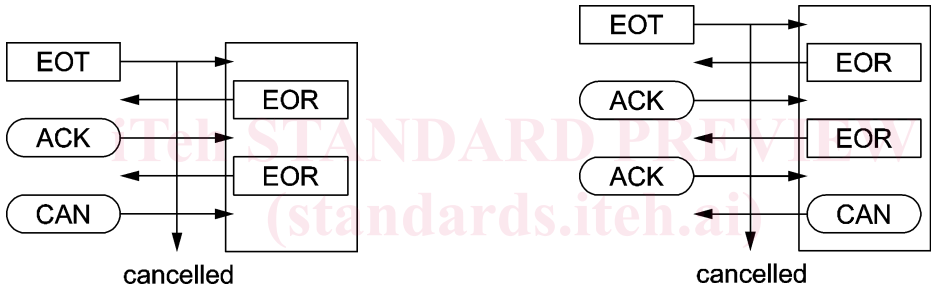


Figure 9

NAK-frame

In case that any problems (except for CRC error) occur with the frame or with the “execution” of the frame a NAK-frame shall be sent. The NAK ID which shall be transmitted in its content shall identify the reason for the NAK, see Figure 10. The application layer has to decide how to continue.

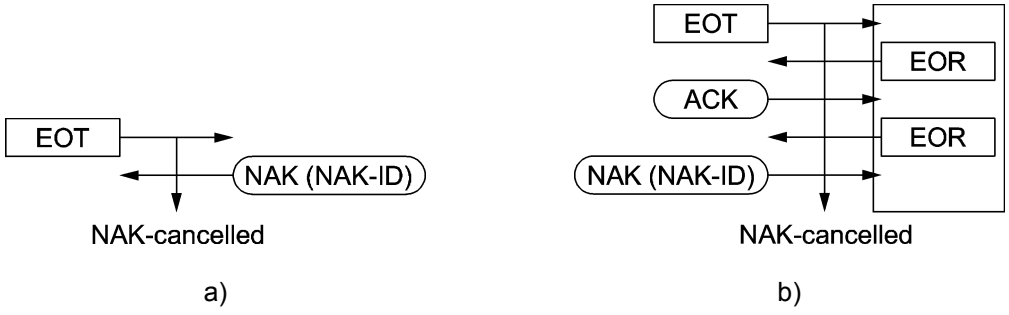


Figure 10

If a NAK-frame is sent the last datagram shall be cancelled. All frames which are acknowledged before with an ACK-frame shall be accepted by this layer. The application layer has to decide whether an incomplete transaction is accepted or not. For NAK ID see Table 17.

5.3 Delay and timeout

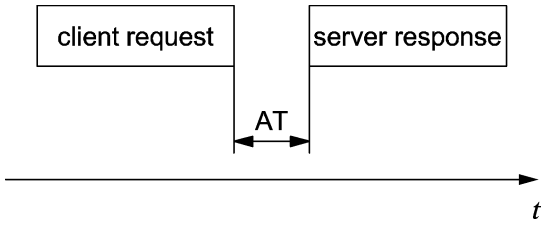


Figure 11

If the answer time (AT), see Figure 11, exceeds Rt, a timeout shall be detected by the client. For short distance (e.g. RS 232, RS485, Low Power Radio) communication “Rt” value shall be 1s.

NOTE This value may not be sufficient for long distance (e.g. GSM/GPRS) communication, where additional routing delays may occur.

The server shall have no timeout.

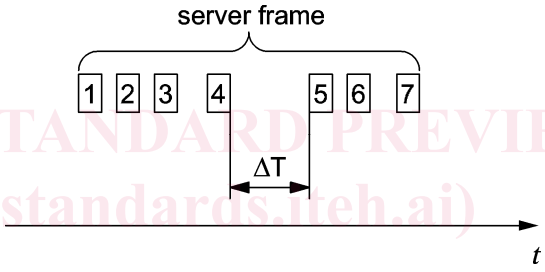


Figure 12

Within a frame transmitted by the server, the delay (ΔT) between two subsequent characters shall not exceed three character times, see Figure 12. For frames transmitted by the client there is no limitation.

If a timeout occurs after a transmission of a dataframe, the previous dataframe shall be repeated by the client. If a timeout occurs after a transmission of a controlframe, a repetition shall be requested by transmission of a TEF-frame.

After 3 consecutive timeouts (1 initial + 2 retries), the transaction shall be cancelled without a CAN-frame (see 5.2). The application layer shall decide what action shall be taken.

As the client is uncertain if the server has received the last ACK-frame before the timeout the last record may be repeated upon the next request to this node.

5.4 CRC16 Checksum

Since a frame might have been corrupted during transmission, a checksum shall be included in each frame. It shall cover all characters from Start—Flag to End—Flag including these flags.

The calculation algorithm for CRC16 shall be according to Annex C of EN 14116:2007+A1:2008.