



BSI Standards Publication

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

National foreword

This British Standard is the UK implementation of EN 15969-1:2022. It supersedes BS EN 15969-1:2017, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AUE/18, Tanks for the transport of dangerous goods.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 17459 5

ICS 13.300; 23.020.10; 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2022.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

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 destinées au transport de matières
dangereuses - Interface numérique pour le transfert de
données entre des véhicules-citernes et des
installations fixes - Partie 1 : Spécifications du
protocole - Contrôle, données de mesure et
d'événements

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

This European Standard was approved by CEN on 19 September 2022.

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. 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 CEN 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 CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

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



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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

Page

European foreword.....	4
Introduction	6
1 Scope	7
2 Normative references.....	7
3 Terms and definitions, abbreviations and conventions	7
3.1 Terms and definitions	7
3.2 Abbreviations	9
3.3 Conventions.....	9
4 Hardware interface	10
5 Basic protocol layer.....	10
5.1 FTL-frame (frame)	10
5.2 Frame flow (handshake)	11
5.3 Delay and timeout.....	16
5.4 CRC16 Checksum	16
6 Data protocol layer (FTL-data protocol)	16
6.1 Client (OBC) and server (TVE).....	16
6.2 Syntax of data in datagrams	17
6.3 Nodes, subnodes, variables	17
6.4 Format identifiers.....	17
6.5 Types of variable values	20
6.6 Kinds of nodes	21
7 FTL-Data	22
7.1 General.....	22
7.2 Record and field types	22
7.3 Systemwide variables (subnode SYSTEM)	23
7.4 Variables related to global positioning system (subnode GPS)	27
7.5 Accessing a printer on TVE-side (subnode PRN)	27
7.6 Compartment information (subnode COMP)	30
7.7 Notification about changes (subnode NOTIFY)	32
7.8 Information about driver (subnode DRIVER).....	33
7.9 Information about the vehicle (variable VEHICLE_ID)	34
7.10 Information about current operation (subnode OPERATION).....	34
7.11 Access to filesystem on TVE (subnode FS)	36
7.12 Auxiliary (subnode AUX)	41
7.13 Order management (subnode ORDER)	42
7.14 Goods and service database (subnode PRODUCT).....	46
7.15 FTL—logfile (subnodes LOG)	49
7.16 Required variables	81
7.17 NAK ID	82
8 Routing for multiple TVE.....	83
8.1 Purpose	83
8.2 Routing solution	83
8.3 Routing example.....	84
9 Communication with office.....	84

9.1	General	84
9.2	Simple file transfer.....	85
9.3	FTL over TCP/IP	87
10	Communication Examples	89
10.1	Examples for Basic Protocol Layer level	89
10.2	Examples for data protocol layer.....	91
Annex A (normative) Node tree		94
Annex B (normative) Test FTL		95
B.1	Overview	95
B.2	Basic Protocol Layer	95
B.2.1	Frame Tests.....	95
B.2.2	CRC-error	96
B.2.3	Delay and Timeout	96
B.3	Data Protocol Layer	96
B.3.1	Test of Toggling	96
B.3.2	Test of the FTL data layer	98
B.3.3	Test of the required FTL nodes.....	98
B.3.4	Optional System Subnodes.....	101
B.3.5	Optional Node Prn	103
B.3.6	Node Comp	105
B.4	Application Layer.....	111
B.4.1	Test of the L-File.....	111
B.4.2	Test of the LH-File.....	111
B.4.3	Test for the Filling of the NodeList	111
B.4.4	Sequence Test	112
Bibliography		114

European foreword

This document (EN 15969-1:2022) has been prepared by Technical Committee CEN/TC 296 “Tanks for the transport of dangerous goods”, the secretariat of which is held by AFNOR.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by April 2023, and conflicting national standards shall be withdrawn at the latest by April 2023.

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

This document supersedes EN 15969-1:2017.

In comparison with the previous edition, the following technical modifications have been made:

- Figure 11 corrected;
- Subclause 7.3.11 “Trailer coupled (variable TRAILER)” added;
- Subclause 7.5.9 “Duplicate print (DUPLICATE)” added;
- Table 55 “Fields of records of ORDER.PLAN” Index 19 to 21 added;
- In subclause 7.13.4 Value V for planned transaction in process and unplanned transaction in process added;
- Table 67 “L file record types” in Index L1137 Bit 3 added;
- Table 67 “L file record types” Index L1147 added;
- Table 67 “L file record types” Index L1205 added;
- Table 67 “L file record types” in Index L2002 event codes 80 to > 100 added;
- Table 67 “L file record types” Index L4007 and L4008 added;
- Table 67 “L file record types” in Index L4206 delivery path 25 to 36 and 80 to 86 added;
- Table 67 “L file record types” Index L4207 and L4208 added;
- Table 67 “L file record types” Index 94 Diagnose added;
- Node tree in Figure A.1 revised.

EN 15969, *Tanks for transport of dangerous goods — Digital interface for the data transfer between tank vehicle and with stationary facilities*, consists of 2 parts:

- *Part 1: Protocol specification — Control, measurement and event data;*
- *Part 2: Commercial and logistic data.*

This document forms part of a coherent standards programme comprising the following standards:

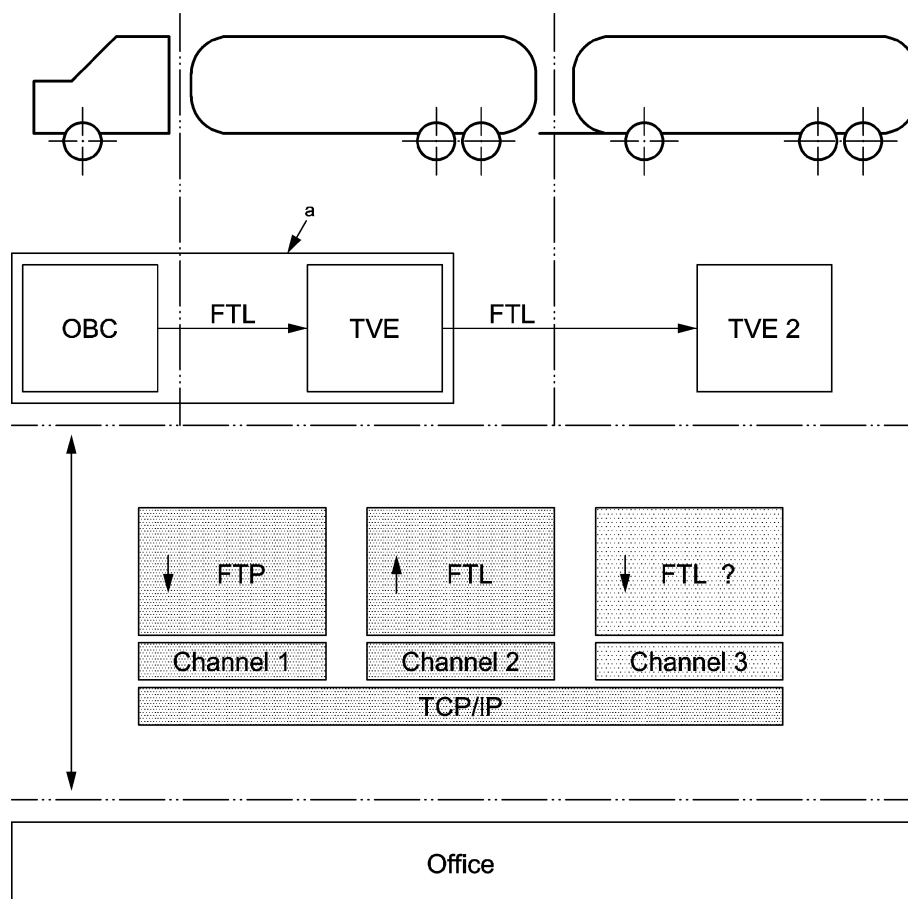
- EN 13616-1, *Overfill prevention devices for static tanks for liquid fuels — Part 1: Overfill prevention devices with closure device*;
- EN 13616-2, *Overfill prevention devices for static tanks for liquid fuels — Part 2: Overfill prevention devices without a closure device*;
- EN 13922, *Tanks for 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 product recognition devices for liquid fuels*;
- EN 15207, *Tanks for the transport of dangerous goods — Plug/socket connection and supply characteristics for service equipment in hazardous areas with 24 V nominal supply voltage*;
- EN 15208, *Tanks for transport of dangerous goods — Sealed parcel delivery systems — Working principles and interface specifications*;
- EN 15969-2, *Tanks for transport of dangerous goods — Digital interface for the data transfer between tank vehicle and with stationary facilities — Part 2: Commercial and logistic data*.

Any feedback and questions on this document should be directed to the users' national standards body. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organisations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Introduction

FTL is an acronym for **Fuel Truck Link**, 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 document specifies data protocols and data format for the communication between electronic equipment (TVE), on-board computer (OBC) of the tank vehicle and stationary equipment.

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.

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

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

DIN 51757:2011, *Testing of mineral oils and related materials — Determination of density*

EN 13616-2, *Overfill prevention devices for static tanks for liquid fuels — Part 2: Overfill prevention devices without a closure device*

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

EN 14116:2012+A2:2018, *Tanks for transport of dangerous goods — Digital interface for product recognition devices for liquid fuels*

EN 15208:2014, *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*

3 Terms and definitions, abbreviations and conventions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <https://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

3.1 Terms and definitions

3.1.1

downgrade

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

3.1.2

answer time

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

3.1.3

array

collection of elements which have the same structure and are able to be accessed individually by means of an index

3.1.4

client

responsible for initiation and control of data exchange

3.1.5

field

element of a datagram delimited by separators

3.1.6

frame

data packet with variable length and specified structure

3.1.7

list

type of variables consisting of a number of records

3.1.8

MaxFrameSize

maximum number of characters in a frame

3.1.9

node

part of an address of a variable

3.1.10

graphic character

character, other than a control function or a format character, that has a visual representation normally handwritten, printed or displayed

[SOURCE: ISO/IEC 10646:2020, 3.28]

3.1.11

record

ordered set of fields, stored contiguously

3.1.12

server

program which provides service to client programs

3.1.13

subnode

subpart of an address of a variable

3.1.14

datagram

instruction or answer to an instruction which comprises an OpCode and operand

3.1.15**transaction**

complete request-answer-cycle

3.1.16**type identifier**

character code for the frame type

3.2 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 1 The OBC is one party in the FTL-communication (the client).

PID	product identification device according to EN 14116
SYN	synchronization controlframe
SPDS	sealed parcel delivery system according to EN 15208
TEF	CRC transmission error controlframe
TVE	tank-vehicle-equipment

NOTE 2 The TVE is one party in the FTL-communication (the server).

OpCode	operation code
--------	----------------

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, it shall be preceded by a comma.

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 be followed by the character “h”.

4 Hardware interface

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

For communication an asynchronous line shall be used (RS232, RS422 or RS485). The OBC and TVE start up and default settings shall be 9 600 baud, 8 data bits, 1 stop bit and no parity.

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.

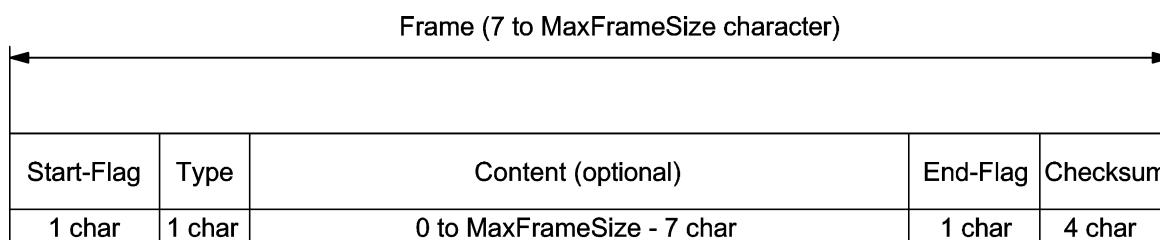


Figure 2

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. A new frame starts upon the receipt of a Start-Flag. Any character received before the Start-Flag shall be ignored. All devices using the FTL-protocol shall be able to receive complete frames of MaxFrameSize characters. 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 minus 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.

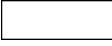
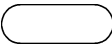
Checksum

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 immediately following the Start-Flag specifies the frame type. The different frame groups and their frame types are described in Table 1.

Table 1 — Frame groups and frame types

Frame group	Frame type	Abbreviation	Additional fields	Type identifier	
				Client to server	Server to client
Dataframe 	end of record frame	EOR	data	R, V	r, v
	additional dataframe following frame	ADF	data	L, P	l, p
	end of transmission frame	EOT	data	E, I	e, i
Controlframe 	acknowledge frame	ACK	no	A	a
	synchronization/wait frame	SYN	no	— ^a	s
	cancel frame	CAN	no	C	c
	CRC transmission error frame	TEF	no	T	t
	not acknowledge frame	NAK	NAK-ID according to Table 71	— ^a	n
^a Not applicable.					

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 from the server shall be answered by the client.

Every frame from the client shall be answered by a frame from the server.

If a dataframe is received by the server when an acknowledge is expected it shall be treated as a cancel frame (CAN) regarding the preceding transaction.

Every data frame on each side, independently, shall be flagged alternatively (toggled) with the secondary (V,P,I) and primary (R,L,E) type identifier. If subsequent dataframes with identical type identifier are received, these shall be treated as a repetition with identical data but shall be answered as the original, see Figure 11. This prevents redundant entries in lists resulting from communication faults.

After the startup of the system the first dataframe on each side shall start with the primary type identifier (R,L,E). The first request after startup shall not be a SET-request to a list.

Examples of frame flows:

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



Figure 3

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

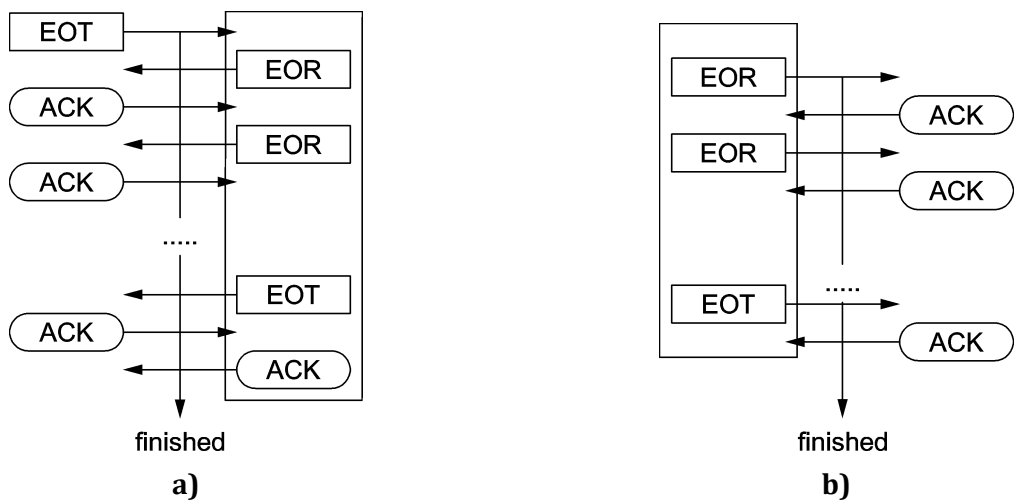


Figure 4

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

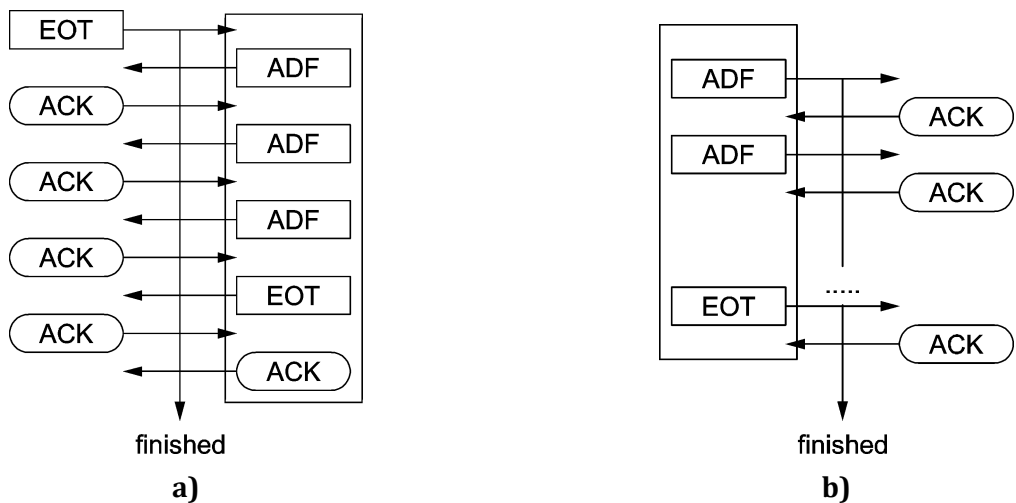


Figure 5

— The preceding examples may be combined as in Figure 6.

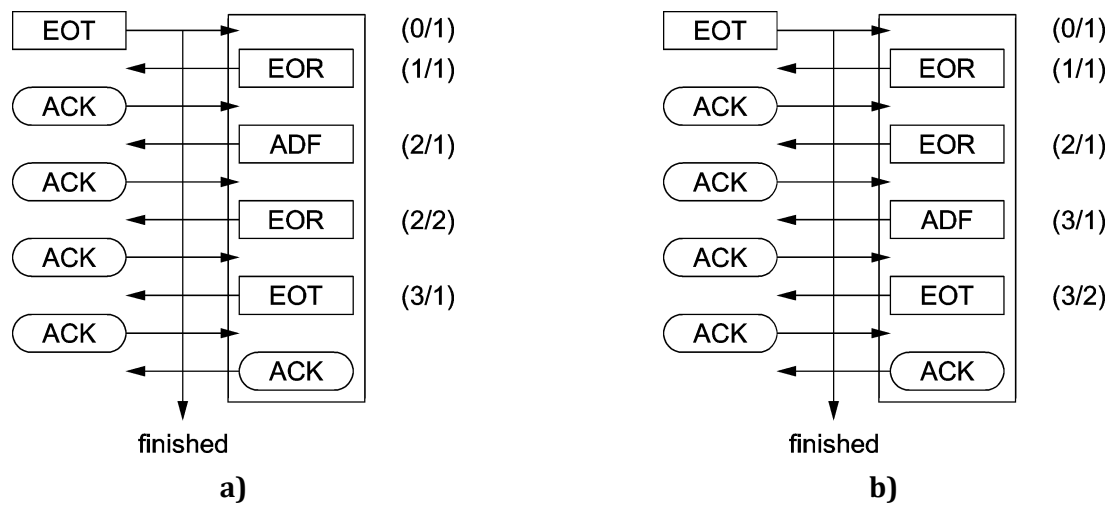


Figure 6

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

SYN—frame

A SYN—frame is not a final acknowledgement. It notifies a busy status of the server 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 R_t , see 5.3.

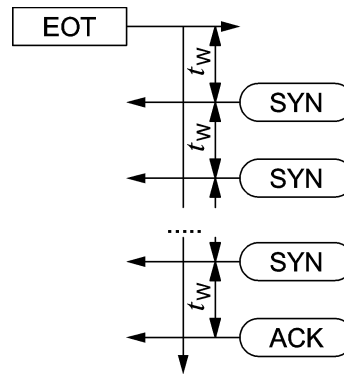


Figure 8

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. It terminates the transaction but does not act as an ACK-frame, see Figure 9.

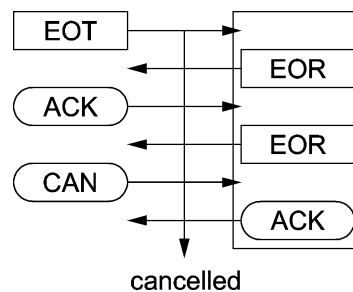


Figure 9

NAK-ID frame

If any problems (except for CRC error) occur with the frame or with the “execution” of the frame a NAK-ID frame shall be sent. The NAK ID, see Table 71, which shall be transmitted in its content, shall identify the reason for the NAK, see Figure 10. The application layer shall decide how to continue.

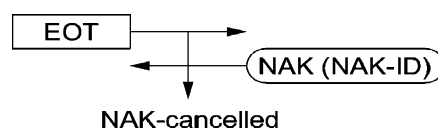


Figure 10



Figure 11

Figure 11 shows the use of the alternating type ID (toggle), which accompanies the record, to identify the receipt of duplicated records.

5.3 Delay and timeout

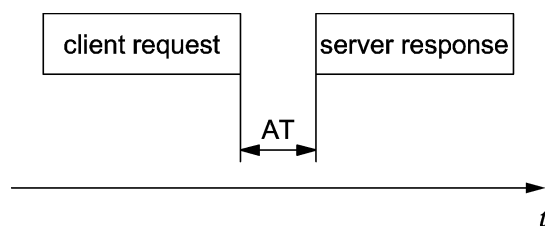


Figure 12

If the answer time (AT), see Figure 12, exceeds R_t , a timeout shall be detected by the client. For short distance (e.g. RS 232, RS485, Low Power Radio) communication “ R_t ” value shall be 1 s.

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

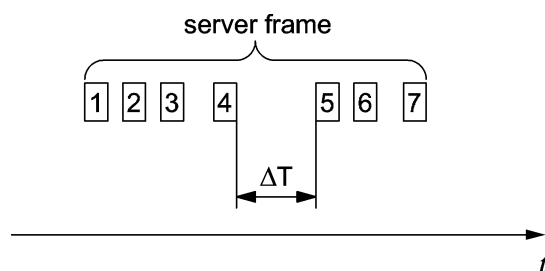


Figure 13

Within a frame transmitted by the server, the delay (ΔT) between two subsequent characters shall not exceed three character times, see Figure 13. 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 three 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 does not know whether 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 EN 14116:2012+A2:2018, Annex B.

6 Data protocol layer (FTL-data protocol)

6.1 Client (OBC) and server (TVE)

The TVE shall act as server, providing data to the client, usually the OBC. Only the client shall initiate data exchange. Each frame from the client shall be followed by one frame from the server.

Thus, the only way for the server to transmit information to the client is to let the client poll the appropriate variables in intervals.

6.2 Syntax of data in datagrams

6.2.1 General

Every datagram shall have the following syntax:

<opcode>,<node>[,<subnode>],<variable>[(<index>)][= <value>].

Except for <value> all other are not case-sensitive. Thus, “ENQ” shall be handled equally to “Enq”.

<subnode> may consist of a number of subnode levels. <value> may be empty.

6.2.2 Operation codes (OpCodes)

The OpCode specifies the type of operation. OpCode according to Table 2 shall be supported:

Table 2 — Operation of FTL protocol

Operation	Opcode	Description
Enquiry	ENQ	Sent by the client to retrieve data from the server (“=” is not allowed)
Set	SET	Sent by the client to set data in the server (“=” shall be present)
Clear	CLR	Sent by the client to delete a list of records (“=” is not allowed)
Report	REP	Sent by the server as a response to a Enquiry (followed by “=” with value). An empty list is identified by a missing equal-sign and no value.
NOTE A further optional Operation “NEW”, which is for channel 3 use only, is given in 9.3.4.		

6.3 Nodes, subnodes, variables

All data provided by server shall be stored in variables which may be accessed by the client (see Clause 7).

Each name identifying <node>, <subnode> and <variable> shall:

- be limited to 10 characters;
- only consist of the characters “A” to “Z”, “0” to “9” and “_”;
- never start with numerical characters “0” to “9”.

Some variables are arrays. By specifying a numerical index enclosed by brackets (), one particular array element shall be accessible.

Indexing shall always start with 1.

6.4 Format identifiers

The formatting of values depends on the corresponding variable. The codes according to Table 3 for format identifiers are used in this document.

Table 3 — Format identifiers

Format identifier	Description
B	Boolean value (0 or 1) Only the values 0 and 1 are valid Examples for valid coding: 0 False 1 True Examples for invalid coding: 2 invalid value A invalid value 0.1 separator not allowed/more than one character not allowed
Nx	Integer decimal value with max x characters (including sign character) Examples for valid coding (for N3): 1 Value: +1 01 Value: +1 123 Value: +123 +12 Value: +12 -12 Value: -12 Examples for invalid coding (for N3): 1234 more than three characters +123 more than three characters 0.1 separator not allowed 1E2 invalid character 'E'
Nx.y	Floating point value max. x digits in front of the period (including ±) max. y digits behind the period Only characters 0 to 9 and period are allowed. At least one digit in front and behind the period shall be used. No exponential expressions. Examples for valid coding (for N3.2): 1.0 Value: +1,0 001.2 Value: +1,2 001.23 Value: +1,23 000.12 Value: +0,12 0.1 Value: +0,1 +01.23 Value: +1,23 -01.23 Value: -1,23 Valid examples for longitude GPS values N4.6 in degrees: +007.512500 7,512500° E = 7° 30' 45" E = 7° 30,750' E 7.5125 7,512500° E = 7° 30' 45" E -007.512500 7,512500° W = 7° 30' 45" W

Format identifier	Description
	Valid examples for latitude GPS values N3.6: +07.512500 7,512500° N = 7° 30' 45" N Examples for invalid coding (for N3.2): 1234.5 more than 3 digits in front of period +123.4 more than 3 digits in front of period -123.4 more than 3 digits in front of period 123.456 more than 2 digits behind period 0,1 separating character not allowed (use period) 123 separating character missing .12 minimum number of digits missing in front of period 123. minimum number of digits missing behind period 1.E5 invalid character
Cx	Text with maximum number of x bytes. UTF-8 according to ISO/IEC 10646:2020 except for codes less than 0x20 and except for comma 0x2C; All other characters are embedded similar to the mechanism of the C programming language using the “\” character followed by a two nibble hex value (examples \n for LF, \x1B for ESC, \, for comma). Following codes shall be implemented: \\ (“\”) \n \r \f \t \, and \xNN (0xNN) Examples for valid coding (for C12): abcdoCD = 12 < abcdoCD = 12 < a\n\x0A a < 0x0A > < 0x0A > “ (<..> .. hex value for control character) a\n\x0A\x1B 12345\,67890 12345,67890 1\,2 Ab\\yz 1,2 Ab\yz Examples for invalid coding (for C12): abcdCD12.3456 more than 12 characters abCD12,345 comma not allowed
S	Time stamp – CCYYMMDDhhmmss Shortened forms of the time stamp string such as (CCYYMMDD) are not allowed. In this case unused fields shall be filled with 00. However a single digit 0 shall be used to represent no time stamp. Is a higher resolution than seconds needed fractions of seconds may be appended at the end of the time stamp string. Examples for valid coding: 19990109230201 09. January 1999 23:02:01 20001224013159.7524. December 2000 1:31:59,75

Format identifier	Description
	Examples for invalid coding: 001224013159 shortened time stamp string 200012240131 seconds are missing 20001224013159,75comma not allowed – use period
D	Date – CCYYMMDD see description for format identifier S above Examples for valid coding: 19990109 09. January 1999 20001224 24. December 2000 Examples for invalid coding: 001224 shortened date string
T	Time – hhmmss see description for format identifier S above Examples for valid coding: 230259 23:02:59 013159.75 1:31:59,75
Hx	Unsigned hexadecimal value with max x characters, representing a bit pattern with $x \cdot 4$ bits Bit 0 is encoded as value 1 Bit 1 is encoded as value 2 Bit 2 is encoded as value 4 and so on Examples for valid coding: 10A8 represents a field of 16 bits where bits 12, 7, 5 and 3 are set. All other Bits are zero. Examples for invalid coding: 10a8 lower case character “a” used 0x10A8 invalid character “x” used \$10A8 invalid character “\$” used

6.5 Types of variable values

6.5.1 Single-Field-Type

<value> represents exactly one field with any one of the variable types given in Table 3.

No separators or special codes for quoting are needed.

6.5.2 CSV Records and quoting

<f1>,<f2>,<f3>,...

In this case, <value> consists of a list of fields separated by the character comma “,”. Each field is of one-field-type.

The formatting of a CSV record shall follow the guidelines below:

- 1) concatenation is done with the separator character comma (“,”);

- 2) numeric field of zero-length (empty field) shall be interpreted as not available;
- 3) trailing commas can optionally be omitted.

6.6 Kinds of nodes

6.6.1 General

When accessing <variable> or <subnode> the behaviour of the server depends on the kind of <variable> or <subnode>. The three kinds in 6.6.2 to 6.6.4 shall be supported.

6.6.2 Values

The simplest kind is a value, which consists of exactly one line, which can either be of type CSV or of any other single-field type, see Table 4.

Table 4 — Operation applicable to simple values

Operation	Effect (if operation allowed in this context)
Enquiry	shall return the current variable contents (<value>)
Set	shall replace the current variable contents (<value>) by the new value given behind the “=”-character
Clear	Not allowed

Examples can be found in Clause 10.

6.6.3 Lists

Some subnodes or variables are accessed as lists, which mean that more than one value or line may be stored or read, see Table 5.

Table 5 — Operation applicable to lists

Operation	Effect (if operation allowed in this context)
Enquiry	shall return the current variable contents, line by line.
Set	shall add the new value given behind the “=”-character to the list
Clear	shall make the list empty

An enquiry of an empty list (list without elements) shall be answered with a report without equal-sign. In contrast, an enquiry of a list with empty element shall be answered with a report with equal-sign without <value>.

If more than one record is to be added to the list it is recommended to send all frames within one transaction using EOR-frames.

EXAMPLE ENQ,FTL,SYSTEM,NodeList

REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,NODELIST

REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,FTL_VERS

REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,FTL_FORMAT

REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,DATETIME

6.6.4 Arrays

Arrays may be accessed according to Table 6 either:

- as value, as described in 6.6.2, when an index is specified in the enquiry or set command (see 6.3);

EXAMPLE 1 ENQ,FTL,AUX,In(1)
 REP,FTL,AUX,In(1) = 2

- or without index, in which case every answer shall contain the requested variable or subnode and the appropriate index.

EXAMPLE 2 ENQ,FTL, AUX,In
 REP,FTL, AUX,In(1) = 2
 REP,FTL, AUX,In(2) = 2
 REP,FTL, AUX,In(3) = 2

Table 6 — Operation applicable to arrays

Operation	Effect (if operation allowed in this context)
Enquiry	with index: shall return the addressed array element without index: return entire array, starting with the lowest, and ending with the highest index
Set	with index: replace the contents of the given array element without index: append the new elements after the last used element
Clear	with index: not allowed without index: clears the contents of all array elements

Examples can be found in Clause 10.

7 FTL-Data

7.1 General

The root node shall be called FTL. (see 8.2 for exceptions). The structure of this node shall be fixed. Thus, subnodes and variables not described in this document shall not be implemented.

However, it is allowed to only implement a subset of the structures mentioned in this specification. Please refer to 7.16 for a list of variables which shall be present in every system.

All mentioned subnodes belong to node FTL, see Annex A.

Unless otherwise noted, when the description of TVE-variable refers to a L—FILE-record type, the first two fields (record type and timestamp) shall be omitted (see example in 7.3.1).

7.2 Record and field types

In this document record types are referred to by #L<recno>_<identifier>. Thus, #L01_DEVICE_ID refers to record type 01, called DEVICE_ID, which can be found in the L_FILE specification in Table 67.

Meaning and formats of fields are referred to by #L<recno><fieldno>[_<identifier>]. Thus, #L0202_veh_type refers to field 2 (called veh_type) in record type 2 (VEHICLE_ID).

7.3 Systemwide variables (subnode SYSTEM)

7.3.1 FTL version (variable FTL_VERS)

Table 7 — Specification of node SYSTEM,FTL_Vers

Variable	FTL,SYSTEM,FTL_Vers = V
Kind	Value, Read Only, required
Enquiry (item)	Returns the version of FTL implemented on the TVE.
Value V	#L00_FTL_VERS

EXAMPLE ENQ,FTL,SYSTEM,FTL_Vers

REP,FTL,SYSTEM,FTL_Vers = 1.00

7.3.2 Format for dataframes (variable FTL_FORMAT)

Table 8 — Specification of node SYSTEM,FTL_Format

Variable	FTL,SYSTEM,FTL_Format = V
Kind	Value, Read Only, optional
Enquiry	Returns the current format used for dataframes.
Value V	"CSV" (default): data are transmitted FTL-formatted

7.3.3 Date and time on TVE (variable DATETIME)

Table 9 — Specification of node SYSTEM,DateTime

Variable	FTL,SYSTEM,DateTime = V
Kind	Value, optional
Enquiry	Returns the timestamp for current date and time on TVE.
Set	Tries to change the timestamp for current date and time on TVE. May also be used to synchronize TVE real time clock
Format	S (see Table 3)

EXAMPLE ENQ,FTL,SYSTEM,DateTime

REP,FTL,SYSTEM,DateTime = 20070820104933

7.3.4 Timeout for OBC alive-test (variable TIMEOUT)

Table 10 — Specification of node SYSTEM,TIMEOUT

Variable	FTL,SYSTEM,TIMEOUT = V
Kind	Value, optional
Enquiry	Returns current timeout setting for alive-test.
Set	Sets a new timeout for alive-test
Value V	Timeout in seconds or 0 to deactivate alive-test. Default value 60 s, maximum time for absence 99999 s (N5)

After each transaction the internal timer shall be reset. When this internal timer exceeds the current timeout setting:

- the connection shall be regarded as timed out;
- the TVE is allowed to initiate processes to be able to work without OBC.

It is recommended to protect the value against power outages to prevent tampering.

7.3.5 Status of OBC alive-test (variable TIMEDOUT)

Table 11 — Specification of node SYSTEM,Timeout

Variable	FTL,SYSTEM,Timeout = V
Kind	Value, optional
Enquiry	Returns the timestamp when connection to OBC was lost.
Set	Reset Timeout condition (V = 0 only)
Value V	Timestamp, when connection was lost. 0 stands for “no timeout since initialization”.

See 7.3.4 for further explanation.

7.3.6 Baudrate for communication (variable BAUD)

Table 12 — Specification of node SYSTEM,Baud

Variable	FTL,SYSTEM,Baud = V
Kind	Value, optional
Enquiry	Returns the current baudrate for FTL communication.
Set	Tries to set a new baudrate for FTL communication.
Value V	Valid baudrate: 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 etc.(N6) Default baudrate for communication is V = 9600.

Since change of communication parameters is a rather sensitive process, several safety procedures shall be implemented:

- TVE shall acknowledge the set using the previous baudrate.

- After the TVE switched to the new baudrate, the OBC shall verify the new setting by requesting the variable again within ten seconds. Otherwise, the TVE shall switch to the default baudrate of 9600.

The OBC should wait at least 5 s before verifying the new baudrate, because the TVE possibly needs some time for switching.

If a baud rate is not supported by the TVE it should answer with NAK-ID 10202.

EXAMPLE ENQ,FTL,SYSTEM,BAUD
 REP,FTL,SYSTEM,BAUD = 9600
 SET,FTL,SYSTEM,BAUD = 115200

Verify new baudrate after 5 s:

 ENQ,FTL,SYSTEM,BAUD
 REP,FTL,SYSTEM,BAUD = 115200

7.3.7 Last system error (variable SYS_ERR)

Table 13 — Specification of node SYSTEM,SYS_ERR

Variable	FTL,SYSTEM,SYS_ERR = V
Kind	Value, read only, cleared on read, optional
Enquiry	Returns the last system error on TVE.
Value V	#L09_SYS_ERR

EXAMPLE (problem with sensor for product code occurred):

 ENQ,FTL,SYSTEM,SYS_ERR
 REP,FTL,SYSTEM,SYS_ERR = 90,10301,44326

7.3.8 List of supported nodes and variables (list NODELIST)

Table 14 — Specification of node SYSTEM,NodeList

Variable	FTL,SYSTEM,NodeList = V
Kind	List, read only, required
Enquiry	Returns a list of nodes and subnodes supported by the answering TVE.
Value V	All supported nodes (including their path)

EXAMPLE ENQ,FTL,SYSTEM,NodeList

 REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,NODELIST
 REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,FTL_VERS
 REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,FTL_FORMAT
 REP,FTL,SYSTEM,NodeList = FTL,SYSTEM,DATETIME

7.3.9 Certificate for electronic signature (list CERTIFICATE)

Table 15 — Specification of node SYSTEM,CERTIFICATE

Variable	FTL,SYSTEM,CERTIFICATE = V
Kind	List, read only, optional
Enquiry	Returns the certificate of the unit used for electronic signature
Format V	Base64 ^a

^a According to 6.8 of RFC 2045; the encoding of Base 64 is explained in the document “Multi purpose internet mail extension” (MIME) subclause 6.8. This document is available at website, accessed through: www.ietf.org/rfc/rfc2045.txt.

7.3.10 Remote popup message (variable MESSAGE)

Table 16 — Specification of node SYSTEM,MESSAGE

Variable	FTL,SYSTEM,MESSAGE = V
Kind	Variable, optional
Set	message number and text to be displayed as popup information on TVE screen
Enquiry	read back the value previously set
Format V	<nr>, <text> where nr is of format N3 and text is of format C255

7.3.11 Trailer coupled (variable TRAILER)

Table 17 — Specification of node SYSTEM,TRAILER

Variable	FTL,SYSTEM,TRAILER = V
Kind	Value, optional
Enquiry	State trailer coupled
Value V	0 = no trailer coupled 1 = trailer coupled, FTL access via TVE 2 = trailer coupled, FTL access directly to TVE 2

7.4 Variables related to global positioning system (subnode GPS)

7.4.1 General

In some systems, both TVE and OBC may use information from a GPS receiver (e.g. time setting, checking for geofencing).

In order to avoid the need to install two GPS systems on one tank vehicle, FTL provides two interfacing alternatives.

7.4.2 GPS on OBC-side (variable OBC)

Table 18 — Specification of node GPS,OBC

Variable	FTL,GPS,OBC = V
Kind	Value, optional
Enquiry	Returns the value, that has been stored there by OBC last time.
Set	Simply stores V in this variable for usage by TVE.
Format	#L08_GPS_INFO

The OBC shall write GPS-data to the TVE.

To avoid excessive traffic on the bus, it is usually sufficient to set the node value only once by the OBC when a phase of movement ends and the tank vehicle comes to rest.

7.4.3 GPS on TVE-side (variable TVE)

Table 19 — Specification of node GPS,TVE

Variable	FTL,GPS,TVE = V
Kind	Value, read only, optional
Enquiry	If TVE includes a GPS receiver, the GPS-data shall be available for use by the OBC.
Format	#L08_GPS_INFO

7.5 Accessing a printer on TVE-side (subnode PRN)

7.5.1 General

When the TVE includes an on board printer (e.g. ticket printer as part of a metrological approved metering system), it may also be used by the OBC to print lists or any other data. In order to avoid conflicts while printing a document, the printer shall be reserved by the OBC before accessing it. Any attempt by the OBC to set a printer node without reserving it first, shall be rejected by the TVE by answering with a NAK-ID frame. If no printer is installed, only the subnode PRN,TYPE is required and shall return 0 (printer not available).

7.5.2 Request for the type of OBC-document to print on TVE (variable PRINTDOC)

Table 20 — Specification of node PRN,PRINTDOC

Variable	FTL,PRN,PRINTDOC = V
Kind	Value, optional
Enquiry	Returns a value representing the document desired to be printed by OBC on TVE.
Set	Set to zero acknowledges that the request has been handled.
Value V	<doctype>[,<arg1>[,...]] Arguments shall be numerical. <doctype> shall be one of 0: no printing request pending 1: delivery note. No argument means “last delivery note”. Argument specifies note number. 2: extended delivery note. Arguments like 1. 3 print parameter data (manufacturer specific data, LP_file) 4 print diagnostic data (manufacturer specific data, LD_file) 99: customer specific request. First argument shall be manufacturer number. Further arguments can freely be used.

This variable may be used as remote control for the OBC, using the display of the TVE, e.g. to avoid climbing into the drivers cabin to initiate a printout. The OBC shall poll this variable during time periods where printing may be desired.

After processing this request, the OBC should clear this variable to signal that the job is done.

7.5.3 Printer port to access (variable PORT)

Table 21 — Specification of node PRN,PORT

Variable	FTL,PRN,PORT = V
Kind	Value, optional
Enquiry	Returns currently selected port used for printing.
Set	Sets printer port, V
Value V	Numeric value from 0 to 9 describing the printer port to use. V = 0 means default printer port.

7.5.4 Type of printer (variable TYPE)

Table 22 — Specification of node PRN,TYPE

Variable	FTL,PRN,TYPE = V
Kind	Value, read only, required
Enquiry	Returns the type of the selected printer.
Value V	Numeric value, one of 0: no printer at this port 1: printer installed but generally not available for OBC

	2: generalized FTL-printer (UTF-8 shall be used) 3: printing to file 4: unknown printer Printers from STAR (10 to 19): 10: STAR SP298 11: STAR DP8340 Printers from EPSON (20 to 29): 20: EPSON TM290 21: EPSON TM295 22: EPSON LQ570 Other printers (90 to 99) manufacturer specific
--	--

7.5.5 Status of selected printer (variable STATUS)

Table 23 — Specification of node PRN,STATUS

Variable	FTL,PRN,STATUS = V
Kind	Value, read only, required if PRN, Type is greater than 0
Enquiry (item)	Returns current status of selected printer.
Value V	Numeric value, one of 0: ready 1: no paper on top (upper sensor) 2: no paper on bottom (lower sensor) 3: no paper (both sensors) 4: printer offline

7.5.6 Reserving the printer (variable RESERVED)

Table 24 — Specification of node PRN,RESERVED

Variable	FTL,PRN,RESERVED = V
Kind	Value, required if PRN, Type is greater than 0
Enquiry	Returns current reservation state of selected printer.
Set	Set to 1 tries to reserve printer for usage by OBC. Set to 0 frees the printer
Value V	Numeric value, one of 0: idle, free for usage 1: currently reserved for OBC > 1: reserved by TVE

The printer could be in use by TVE or another client. In this case, the TVE reserves the printer itself.

If the OBC tries to reserve the printer by writing V = 1, while the printer is already reserved by TVE or another client, the TVE shall answer with a NAK-ID frame.

When the printer was reserved by the OBC and a further request by the OBC for reservation is done, the TVE shall answer with an ACK and shall leave the value at $V = 1$.

7.5.7 Commands to the printer (variable CMD)

Table 25 — Specification of node PRN,CMD

Variable	FTL,PRN,CMD = V
Kind	Value, required if PRN, Type is greater than 0
Enquiry	Result unspecified
Set	Executes a command, returns ACK-frame after the command was successfully executed, NAK-ID frame otherwise
Value V	Numeric value, one of 0: no operation 1: grab paper 2: eject paper

7.5.8 Text to print (variable TX_TEXT)

Table 26 — Specification of node PRN,TX_TEXT

Variable	FTL,PRN,TX_TEXT = V
Kind	Value, required if PRN, Type is greater than 0
Enquiry	Result unspecified
Set	Prints the text, returns ACK-frame after the text was successfully printed, NAK-ID frame otherwise.
Value V	Text to print, see Table 3, Cx.

7.5.9 Duplicate print (variable DUPLICATE)

Table 27 — Specification of node PRN,DUPLICATE

Variable	FTL,PRN,DUPLICATE = V
Kind	Value, optional if PRN, Type is greater than 0
Set	Prints the number of V copies of the last printed ticket
Value V	Number of copies, (N1)

7.6 Compartment information (subnode COMP)

7.6.1 General

First compartment ($n = 1$) shall be next to driver cabin.

7.6.2 Number of compartments (variable COUNT)

Table 28 — Specification of node COMP,COUNT

Variable	FTL,COMP,COUNT = V
Kind	Value, read only, required for SPDS
Enquiry	Returns number of compartments the TVE controls.
Value V	Non negative number (N_2)

7.6.3 Current status of a compartment (array STATUS)

Table 29 — Specification of node COMP,STATUS

Variable	FTL,COMP,STATUS[(n)] = V
Kind	Array, read only, required for SPDS
Enquiry	Returns array with status of all compartments.
Enquiry (n)	Returns status of compartment n.
Format	#L40_COMP_STAT

7.6.4 Current status of load (array CONTENT)

Table 30 — Specification of node COMP, CONTENT

Variable	FTL,COMP, CONTENT [(n)] = V
Kind	Array, read only, optional
Enquiry	Returns array with the current load of all compartments.
Enquiry (n)	Returns the current load of compartment n
Format	#L13_COMP_CONT

7.6.5 Currently connected PIDs (list of PID_INFO)

Table 31 — Specification of node COMP,PID_INFO

Variable	FTL,COMP,PID_INFO = V
Kind	List, read only, optional
Enquiry	Returns a list of CSV records with all detected PIDs for all compartments.
Format	#L45_PID_INFO

7.6.6 Loading information from gantry (array LOADING)

The content of the loading information from the gantry shall be according to the following:

Table 32 — Specification of node COMP,Loading(n)

Variable	FTL,COMP,Loading(n) = V
Kind	Array, optional
Enquiry	Returns array with loading information for each compartment stored by loading gantry (OBC).
Clear	Erases all previous loading information for compartments.
Enquiry (n)	Returns a CSV record with loading information for compartment n stored from loading gantry (OBC).
Set (n)	Sets new loading information for compartment n.(e.g. from gantry)
Format	#L11_Transfer

This node may be interrogated at any time. Note that the data quality may vary, depending on the timing of the request.

7.7 Notification about changes (subnode NOTIFY)

7.7.1 General

There are several subnodes and variables, which are recommended to be checked frequently. This can lead to heavy load on the communication line, and thus, reducing the bandwidth available for transmission of real data.

Subnode NOTIFY provides a simple method to overcome this drawback. Instead of polling several variables, the OBC polls only one subnode (Notify). The answer is a list of all nodes that were changed since the last time they were read.

Since not all nodes are of interest to the OBC at any time, the OBC may set up a list of all nodes that are subject to notification.

7.7.2 List of Supervised subnodes and variables (list NODELIST)

Table 33 — Specification of node NOTIFY,NodeList

Variable	FTL,NOTIFY,NodeList = V
Kind	List, optional
Enquiry	Returns a list of all subnodes and variables, which are subject to notification.
Set	Tries to add a new subnode or variable for notification
Clear	Drops the list with the effect, that no subnode or variable is supervised anymore.
Value V	node (including their path) For command SET, V may be "*" (asterisk). In this case, the list will be set to all subnodes and variables, which are providing the supervision feature.

When a subnode or variable cannot be added to the list, because it does not support supervision, a NAK-ID frame with value 10600 is returned.

Set-commands for already added subnodes or variables shall be accepted and acknowledged.

When an array is added for notification (e.g. FTL,COMP,STATUS), all elements of the array shall be monitored for notification. It shall not be possible to only add items of an array.

When a node of type list is added for notification (e.g. L_FILE), it shall be shown as “changed” as soon as new entries have been added or existing entries have been changed.

EXAMPLE 1 CLR,FTL,NOTIFY,NODELIST
 SET,FTL,NOTIFY,NODELIST = FTL,COMP,STATUS
 SET,FTL,NOTIFY,NODELIST = FTL,PRN,Status
 SET,FTL,NOTIFY,NODELIST = FTL,PRN,Port

or for adding all possible subnodes and variables,

EXAMPLE 2 SET,FTL,NOTIFY,NODELIST = *

Initially the list is empty.

7.7.3 List of changed subnodes and variables (list CHANGES)

Table 34 — Specification of node NOTIFY,Changes

Variable	FTL,NOTIFY,Changes = V
Kind	List, optional
Enquiry	Returns a list of all nodes (including their path), which are on the node list and have changed since the last enquiry of this node.

The TVE shall clear an entry in this list when this node of this entry is read.

EXAMPLE ENQ,FTL,NOTIFY,Changes
 REP,FTL,NOTIFY,Changes = FTL,COMP,STATUS
 REP,FTL,NOTIFY,Changes = FTL,PRN,STATUS
 REP,FTL,NOTIFY,Changes = FTL,L_FILE

7.8 Information about driver (subnode DRIVER)

7.8.1 List of drivers (list DRIVERS)

Table 35 — Specification of node DRIVER,DRIVERS

Variable	FTL,DRIVER,DRIVERS = V
Kind	List, optional
Enquiry	Returns the list of all drivers according to the record structure described in Table 36.
Set	Adds a new driver to the list.
Clear	Removes all drivers from the database on the TVE.
Value V	See Table 36.

The TVE shall be able to exchange information about drivers with OBC. Table 36 contains the structure which the subnode shall be based on.

Table 36 — Format of CSV record “DRIVER.DRIVERS”

No	Name	Type	Description
1	DRV_ID	N6	Identification number of driver #L0302
2	DRV_NAME	C30	Name of driver in recommended format <name> <surname>
3	LNG_ID	C2	ID of driver's language according to ISO 639-1 which shall be used for communication.
4	DRV_PIN	N6	PIN code of driver.

7.8.2 Driver currently logged in (variable CURRENT)**Table 37 — Specification of node DRIVER,CURRENT**

Variable	FTL,DRIVER,CURRENT = V
Kind	Value, optional
Enquiry	Returns a record for the driver currently logged in.
Set	Sets a new record for the driver logged in on the OBC.
Clear	Stores an empty value in this variable.
Value V	See Table 36

This variable affects both directions. When a driver logs into TVE, the TVE shall update this node. Alternatively, when the driver logs into OBC, the OBC shall set this node.

When the driver logs out, this shall be indicated by value 0 in field DRV_ID.

7.9 Information about the vehicle (variable VEHICLE_ID)**Table 38 — Specification of node VEHICLE_ID**

Variable	FTL,VEHICLE_ID
Kind	Value, read only, required
Enquiry	Returns the identification of the vehicle
Format	#L02_VEHICLE_ID

This variable allows the identification of the trailer by the OBC to detect change of trailers.

7.10 Information about current operation (subnode OPERATION)**7.10.1 Information about current meter status (array METER_INFO)****Table 39 — Specification of node OPERATION,METER_INFO**

Variable	FTL,OPERATION,METER_INFO[(n)] = V
Kind	Array, read only, optional
Enquiry	Returns array with current status of all meters.
Enquiry (n)	Returns current status of meter n.
Format	#L11_TRANSFER