ERSA

ERTMS / ETCS DMI

DMI - EVC Interface Specification

Document Characterisation

File Name	Reference	Version and date	N° of Pages
ERSA_DMI_SPEC_DMI-EVC Interface_v4.4.2b.doc	ERSA_DMI_SPEC_DMI-EVC_Interface	4.4.2b of 03/11/2014	105

Approval

Last Approved Document Version *: 4.4.10			
Name	**Function / Company**	**Date**	**Signature**
Michel MEHL	Project Leader / E.R.S.A.		

Circulation

Company	Addressee/Function		Reason
ERSA	Michel MEHL	DMI Project Leader	A
	Nicolas VAN LANDEGHEM	EVC Project Leader	R
	Didier WECKMANN	EVC Software Engineer	R

***A** = **A**pproval, **R** = **R**eview, **Q** = **Q**uality, **C** = a**C**tion, **I** = **I**nformation*

Change Description

Version	Date	Author	Comment	A*
4.4.2b	03/11/14	Didier WECKMANN	Update and review	

**A(Approved) if this box is checked (✓) then the corresponding version of the document was approved and released.*

Referenced Material

No	Author Company	File Name	Reference	Version and date	Title/Comment(s)
/1/	UNISIG		Subset-026	2.3.0 of 24/02/2006	ERTMS/ETCS Class 1 System Requirement Specifications
/2/	UNISIG		Subset-026	3.3.0 of 07/03/2012	ERTMS/ETCS Class 1 System Requirement Specifications
/3/	ERA	ETCS driver machine interface v3.3.0.doc	ERA_ERTMS_015560	3.3.0 of 01/03/2012	ETCS Driver Machine Interface
/4/	ERA	ETCS driver machine interface v3.0rev.doc	ERA_ERTMS_015560	3.0 of 19/02/2010	ETCS Driver Machine Interface
/5/	ERA	ERA-ERTMS-015560 v2.3.pdf	ERA_ERTMS_015560	2.3 of 14/4/2009	ETCS Driver Machine Interface
/6/	IEEE/ACM		COBS, Consistent Overhead Byte Stuffing, ToN paper	VOL.7, NO. 2, APRIL 1999	TRANSACTIONS ON NETWORKING
/7/	ERSA	ERA02_DMI_MAN_System Configuration and Installation_v2.0.doc	ERA02_DMI_MAN_SystemSCI	2.0 of 9/6/2010	ERTMS/ETCS DMI System Configuration and Installation
/8/	ERA		Subset-108	1.2.0 of 17/01/2008	Interoperability-related consolidation on TSI annex A documents
/9/	ERA		ERA_ERTMS_040001	1.12 of 22/03/2013	Assignment of values to ETCS variables
/10/	UNISIG		Subset-058	3.0.0 of 02/03/2012	FFFIS STM Application Layer

Table of Contents

Figure index

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to specify and describe the communication interface between the ERTMS/ETCS DMI and the EVC ERTMS baseline 3 (SRS 3.3.0).

1.2 SCOPE

The scope of this communication is limited to the DMI strictly developed according to the ERA specification versions for the ETCS Driver Machine Interface for the respective ERTMS baseline version.

The description covers the following issues:

- Hardware requirements.
- Functional requirements.
- Communication protocol.
- Communication data using packets and variables.

1.3 PROTOCOL VERSION MANAGEMENT

As the major and minor version of the protocol are transmitted during the communication establishment and are used to check communication compatibility, it is important to increment them at each modification of this specification which has an impact on the format of the exchanged data.

1.4 UTILISATION

This document and its content have to be kept confidential inside ERSA and shall not be distributed outside.

This document may be provided as information delivery to a third party only depending on specific contractual arrangements.

1.5 PROPERTY AND CONFIDENTIALITY

2 HARDWARE REQUIREMENTS

The DMI and EVC may communicate through the following media:

- Named pipe (system FIFO).
- Serial interface.

The requirements are the same on Windows and on Linux operating systems.

The named pipes are mostly intended to be used in simulation environment where EVC and DMI run on the same machine.

Serial interfaces are used in industrial environments when DMI and EVC are two physically separated devices.

Since the communication protocol is half-duplex (see §4), either half-duplex serial links (RS485) or full-duplex serial links (RS232, RS422) can be used.

The characteristics of the serial link are configurable parameters both on DMI and EVC side.

The following values are commonly used: 38400 b/s baud rate, 8 data bits, no parity, and 1 stop bit.

3 FUNCTIONAL REQUIREMENTS

3.1 COMMUNICATION MONITORING

The DMI display becomes dark when no cyclic telegram can be received within 3 seconds.

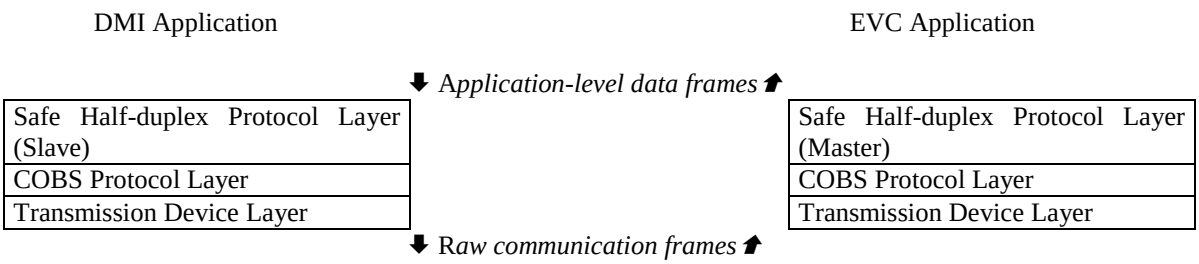
3.2 APPLICATION AVAILABILITY MONITORING (HARDWARE WATCHDOG)

When running on an industrial hardware display, the DMI integrates the management of a hardware watchdog. This watchdog is re-triggered periodically every 6 seconds. This mechanism ensures the whole system is restarted (machine re-boot) if the DMI application is not alive anymore after this time.

4 COMMUNICATION PROTOCOL

4.1 PROTOCOL STACK OVERVIEW

The communication between EVC and DMI relies on a stack of protocol layers which combined ensure robustness and integrity of the exchanged data.
The following diagram gives an overview of the protocol stack in OSI-like representation:



On the uppermost protocol level, a half-duplex protocol layer enables the controlling of a half-duplex communication flow. This layer wraps output data frames coming from the upper application level (see packets specification given hereafter) with additional control flow data and process these later to synchronize the communication flow.

The half-duplex layer also ensures integrity of the data frames (including the control flow data) by appending a CRC computed for the PDU frame and checking the CRC for input frames coming from the lower layer.

The COBS protocol layer is used to transport data frames passed on by the CRC layer, ensuring identification of the start and the end of a data frame.

On the lowest protocol level, any transmission medium featuring a sufficient throughput can virtually be used for the data exchange, using the appropriate hardware and system routines. Currently, the DMI and EVC may communicate through the following media:

- Named pipe (system FIFO).
- Serial interface.

4.2 TRANSMISSION DEVICE LAYER

Named pipes (system FIFOs) are mostly intended to be used in simulation environment where EVC and DMI run on the same machine and are not physically separated.

Serial interfaces are used in industrial environments when DMI and EVC are two physically separated devices. Since the communication protocol is half-duplex (see §4), either half-duplex serial links (RS485) or full-duplex serial links (RS232, RS422) can be used.

When RS485 is used, note that for sending data, the DMI sets the RTS signal (Request To Send) and clears it once the data were actually sent through the cable. The EVC cannot send any data as long as the RTS signal is set by the DMI.

The DMI also supports the following device layer:

- UDP/IP
- TCP/IP
- Raw Ethernet (802.1Q)
- RS232
- RS485

4.3 COBS PROTOCOL LAYER

Data frames are transported using a byte stuffing algorithm, called Consistent Overhead Byte Stuffing (COBS) [/6/].

COBS performs a reversible operation to eliminate zero bytes from the data. Once eliminated, the value 0x00 can be used as a frame marker delimiter.

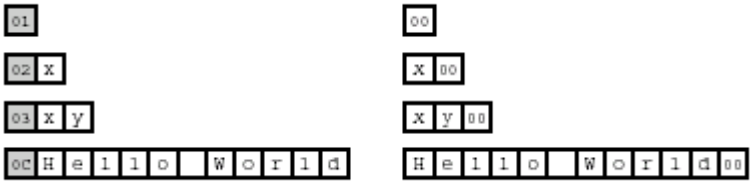
The following steps are performed:

- Logically append a single zero byte to the input data. (It is not necessary actually to add this zero byte to the end of the packet in memory; the encoding routine simply has to behave as if the added zero were there.)
- Locate all the zero bytes in the packet (including the added one), and divides the packet at these boundaries into one or more *zero-terminated chunks*. Every zero-terminated chunk contains exactly one zero byte, and that zero is always the last byte of the chunk. A chunk may be as short as one byte (i.e. a chunk containing just a solitary zero byte) or as long as an entire packet.
- Encode each zero-terminated chunk using one or more variable length *COBS code blocks*. Chunks of 254 bytes or fewer are encoded as a single COBS code block (0x01 to 0xFE). Chunks longer than 254 bytes are encoded using multiple code blocks (0xFF). After all chunks have been encoded, the entire resulting aggregate block of data is completely free of zero bytes, so zeroes can then be placed around the encoded packet to mark clearly where it begins and ends.

A *COBS code block* consists of a single code byte, followed by zero or more data bytes. The number of data bytes is determined by the code byte:

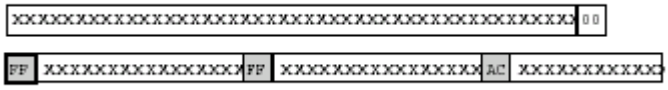
Code	Followed by	Meaning
0x00	(not applicable)	(not allowed)
0x01	no data bytes	A single zero byte
n	(*n*–1) data bytes	The (*n*–1) data bytes, followed by a single zero
0xFF	254 data bytes	The 254 data bytes, **NOT** followed by a zero

Examples:



For codes 0x01 to 0xFE, the meaning of each code block is that it represents the sequence of data bytes contained within the code block, *followed by an implicit zero byte*. The zero byte is implicit — it is not actually contained within the sequence of data bytes in the code block.

If a chunk is greater than 254 characters, code 0xFF represents the sequence of 254 data bytes contained within the code block, *without any implicit zero*. Chunks longer than 254 bytes are encoded using one or more of these special maximum length code blocks, followed by a single normal code block. Each special maximum length code block has no implied trailing zero, but the final (normal) code block does include an implied trailing zero at the end, so this aggregate sequence of code blocks correctly encodes the required long chunk of non-zero data with a single zero byte at the end.



4.4 SAFE HALF-DUPLEX PROTOCOL LAYER

4.4.1 Overview

This protocol layer provides a way to arbitrate half-duplex communications over the transmission layer, in particular when a RS485 link is used to avoid data collisions. It also provides control flow for ensuring data reception and data re-transmission (data sequence numbers).

The protocol relies on a master/slave scheme, whereby the slave can only send data as response to a frame received from the EVC. The EVC plays the role of the master and the DMI is the slave.

The half-duplex protocol also provides basic means based on sequence numbers in order to guarantee data reception. Each data frame sent is assigned and conveys a unique sequence number which shall be returned as acknowledgement in the response message.

4.4.2 Protocol frames definition

A set of protocol frames, also called PDU (Protocol Data Unit), are designed to implement the half-duplex protocol.

A PDU contains control information as well as the payload data if any.

PDUs are numbered so that each PDU can be distinguished from each other. There are two types of PDUs:

- DATA: this PDU used to convey some data and acknowledge last received PDU
- NO_DATA: this PDU is sent when no data needs to be sent and acknowledge last received PDU. On Master side, this PDU is used to allow DMI to send something even when EVC has nothing to send. On slave side, it uses to acknowledge last received PDU when this is no data to send.

All PDUs consist of at least 3-bytes protocol header and the application data to transport with the protocol. The protocol header itself consists of:

- A unique 1-byte frame type identifier.
- 1-byte sequence number identifying the PDU.
- 1-byte sequence number of the last received PDU, which is used as acknowledgement for the last received PDU.

The following tables describe the structure of each PDU type:

DATA PDU			
Field name	Size (bits)	Value	Meaning
Id	8	0	Frame identifier
SN	8	[0-255]	Sequence number
AckN	8	[0-255]	Sequence number of last received PDU
Length	16	*Application dependent*	Size of payload (bytes)
Payload	8 * n bytes	*Application dependent*	Conveyed payload data. 0 to N bytes
CRC	32		CRC32 computed over header and payload

NO_DATA PDU			
Field name	Size (bits)	Value	Meaning
Id	8	1	Frame identifier
SN	8	[0-255]	Sequence number
AckN	8	[0-255]	Sequence number of last received PDU
CRC	32		CRC32 computed over header and payload

For SN and AckN permitted values are in the range 1-255. For re-synchronizing purpose, a special SN=0 value is defined.

Note: It is important that this layer runs above the COBS protocol layer in order to ensure a minimum integrity in the data frames exchanged on the lowest layer, i.e. for detecting whether frames are complete and re-synchronising communication following an interruption.

4.4.3 Principles

Any PDU is assigned and conveys a unique sequence number which shall be returned as acknowledgement in the next received PDU.

The Master and Slaves respectively manages standalone a sequence number counter which is incremented and assigned to each successive sent PDU. Sequence numbers start counting from 1 and are monotonic increasing by 1, with byte wrap from 255 to 1. Sequence number 0 shall be reserved for synchronization purposes initiated by the Master. The Master and Slaves respectively keep track of the last received PDU sequence number, which is stored in field AckN in the next sent PDU.

The sequence number returned as acknowledgement in each sent PDU shall be the last valid received PDU sequence number, regardless of whether a PDU of this sequence number was already received or not.

If a frame with an unexpected SN is received, the sequence number returned as acknowledgement in each sent PDU shall be the last valid received PDU sequence number, not the received unexpected SN.

Master and Slave can transmit the PDU numbered SN+1 if and only if PDU numbered SN was acknowledged (AckN=SN)

The Master always takes the initiative to transmit a PDU. The Master associates the PDU with a sequence number SN and starts a timeout T to get the response. The Master ensures that a PDU of any type (i.e. DATA or NO_DATA) is sent at regular configurable pace, which is 300ms by default, to allow the DMI to send any data even regardless of whether the EVC has any data to transmit or not.

The Slave sends back a proper PDU with AckN=SN in order to acknowledge the PDU previously received from the Master. In case the Slave has some data ready to be sent to the Master, it adds those data to the response PDU.

When no response was received, the Master re-sends the same PDU after a time T=300ms. The timeout has to be set greater or equal to the sum of the transmission time, the receiver elaboration time, and the send back time.

Master and Slave shall ignore PDUs having a sequence number different from the expected one.

Received frames with the same sequence number as the previous received one are discarded.

The Master uses the special frame with SN=0 and AckN=0 to synchronize with Slave. The communication is synchronised when the Master receives as response a PDU with SN=0 and Ack=0. Following this exchange, the SN counting on both Master and Slave side starts from 1 and initial AckN is 0.

Synchronisation typically occurs at initialisation for the first frame exchange. Synchronisation is re-performed in case of multiple failures of transmission.

The maximum number of faulty transmissions to be tolerated before re-synchronizing the channel is fixed to 3.

This means approximately 1 s of communication latency in case of unrecoverable misalignment of sequence numbering.

On both sides of the communication channel the CRC errors on received PDUs must be treated in the same way as for the loss of PDUs. This means that the last sent PDU must be re-sent.

4.4.4 Protocol frames sequencing

A) SYNCHRONISATION

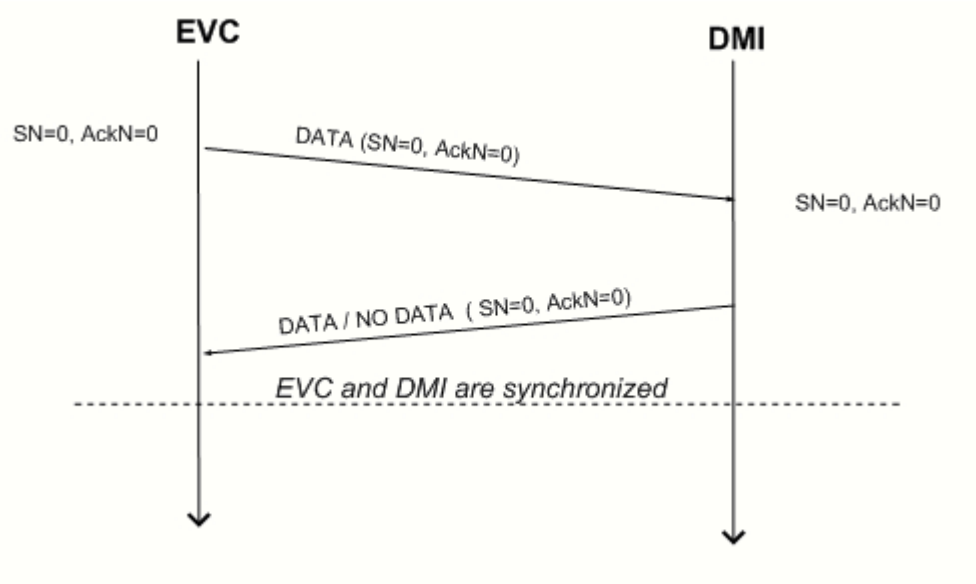


Figure 1 Half-duplex protocol synchronisation sequence

B) NORMAL COMMUNICATION

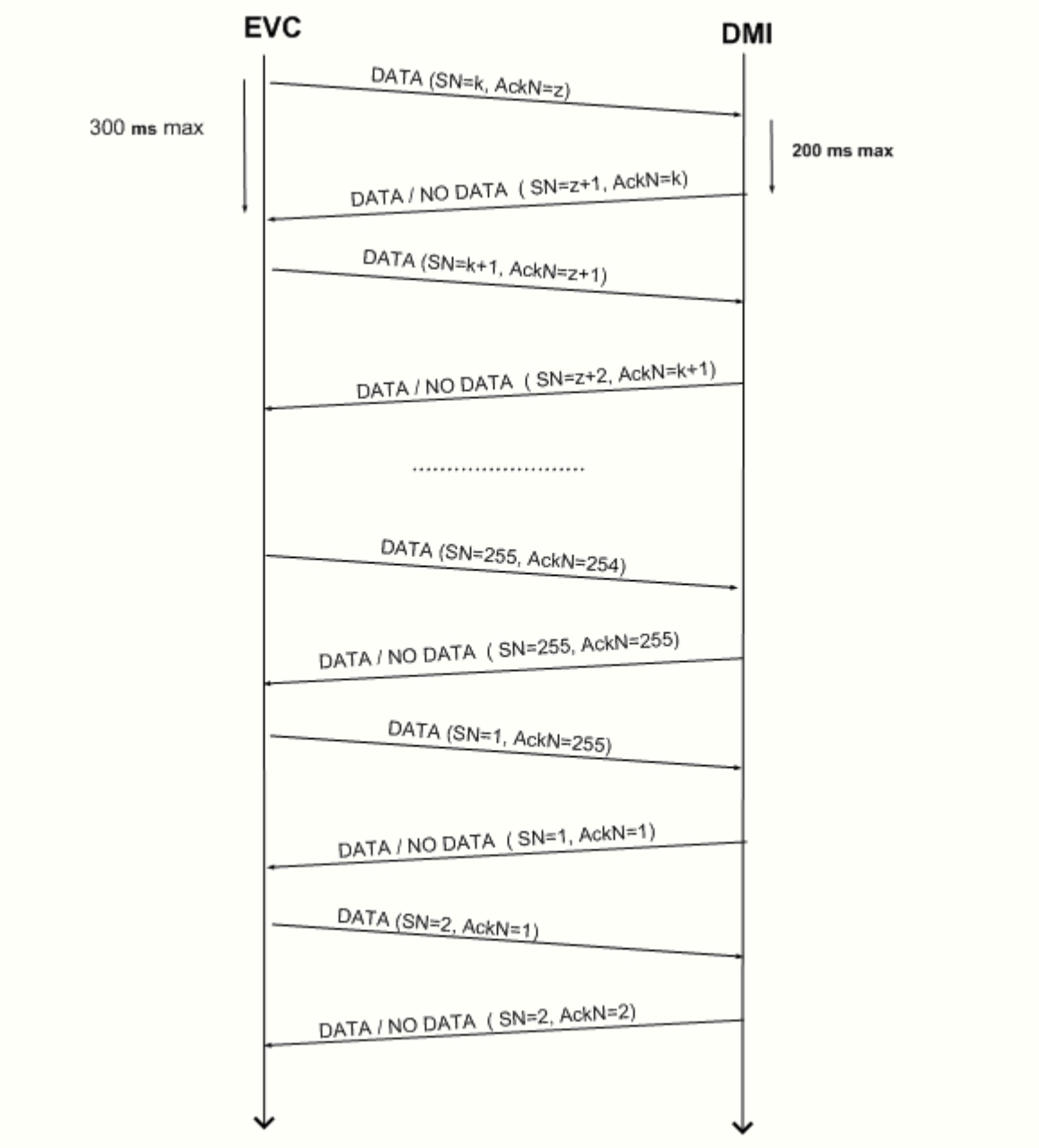


Figure 2 Half-duplex normal frame sequence

C) ERRONEOUS SEQUENCE DETECTED BY EVC: TIMEOUT FOR DATA ACK

The following sequence diagram describes the case in which a PDU from DMI to EVC gets lost. The timeout by EVC side expires and EVC re-sends the previous PDU. In order to avoid collision on the half-duplex link, DMI must respect the response time of 200 ms. Otherwise the retry done by EVC could be in conflict with DMI response

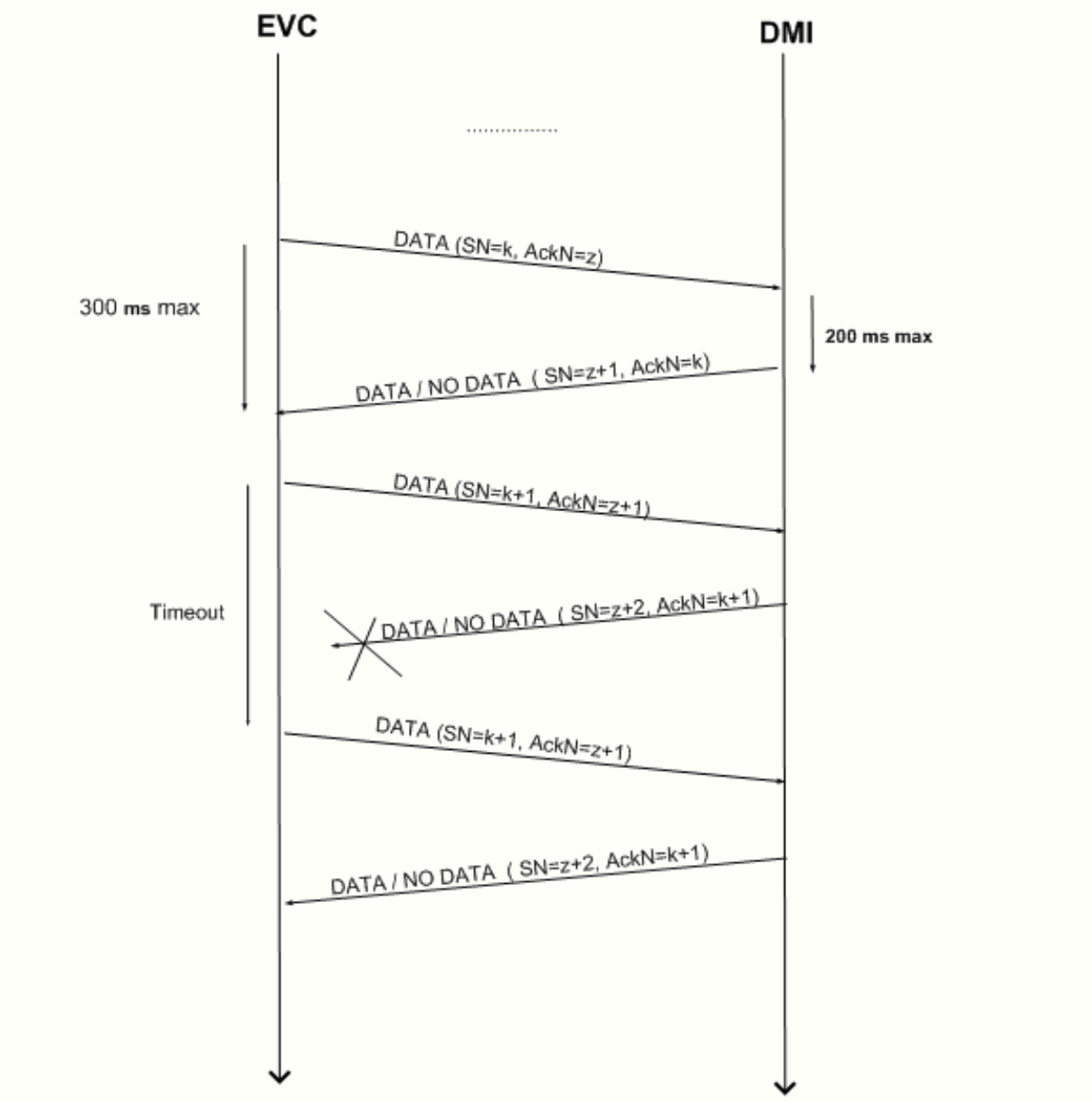


Figure 3 Half-duplex Erroneous sequence detected by EVC: timeout for data ack

D) ERRONEOUS SEQUENCE DETECTED BY EVC: TIMEOUT FOR DATA ACK (DATA NOT ARRIVED TO DMI)

The following sequence diagram describes the case in which a PDU from EVC to DMI gets lost. The timeout on EVC side expires and EVC re-sends the previous PDU.
No timeout requirements are defined for DMI. Slave DMI only transmits as a consequence of the reception of a PDU from the Master EVC.

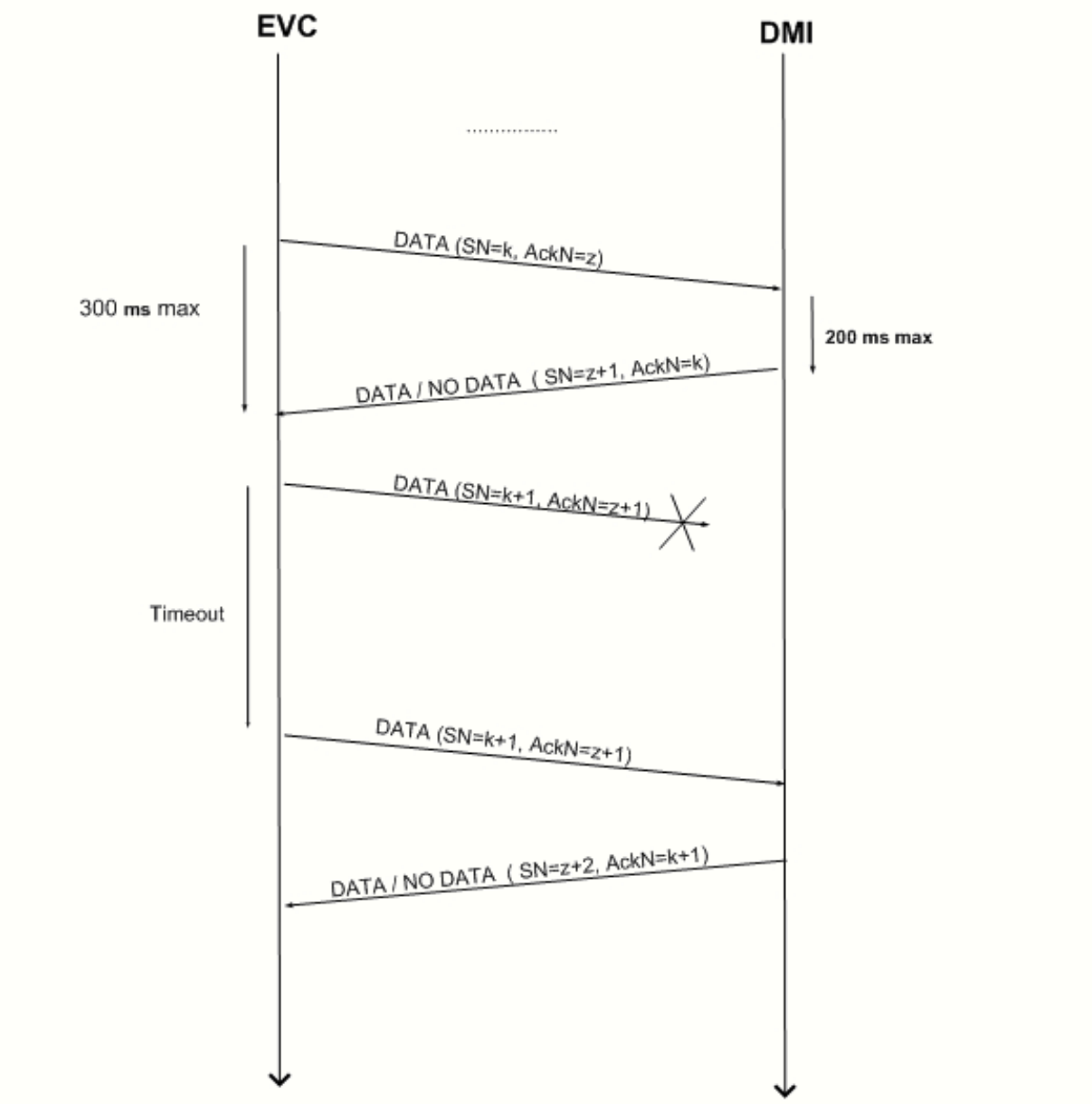


Figure 4 Half-duplex Erroneous sequence detected by EVC: timeout for data ack (DATA not arrived to DMI)

E) PDU CORRUPTION FROM EVC TO DMI - NEGATIVE ACKNOWLEDGEMENT

The following sequence diagram describes the case in which a PDU from EVC to DMI is corrupted (noise or cable malfunction). DMI checks the CRC of the PDU and immediately is able to inform EVC about the invalid reception. This is done simply by confirming the sequence number of the last correctly received PDU.
No timeout is needed in this case, since EVC immediately is able to re-transmit the corrupted PDU, as soon as it receives the "negative acknowledgement".

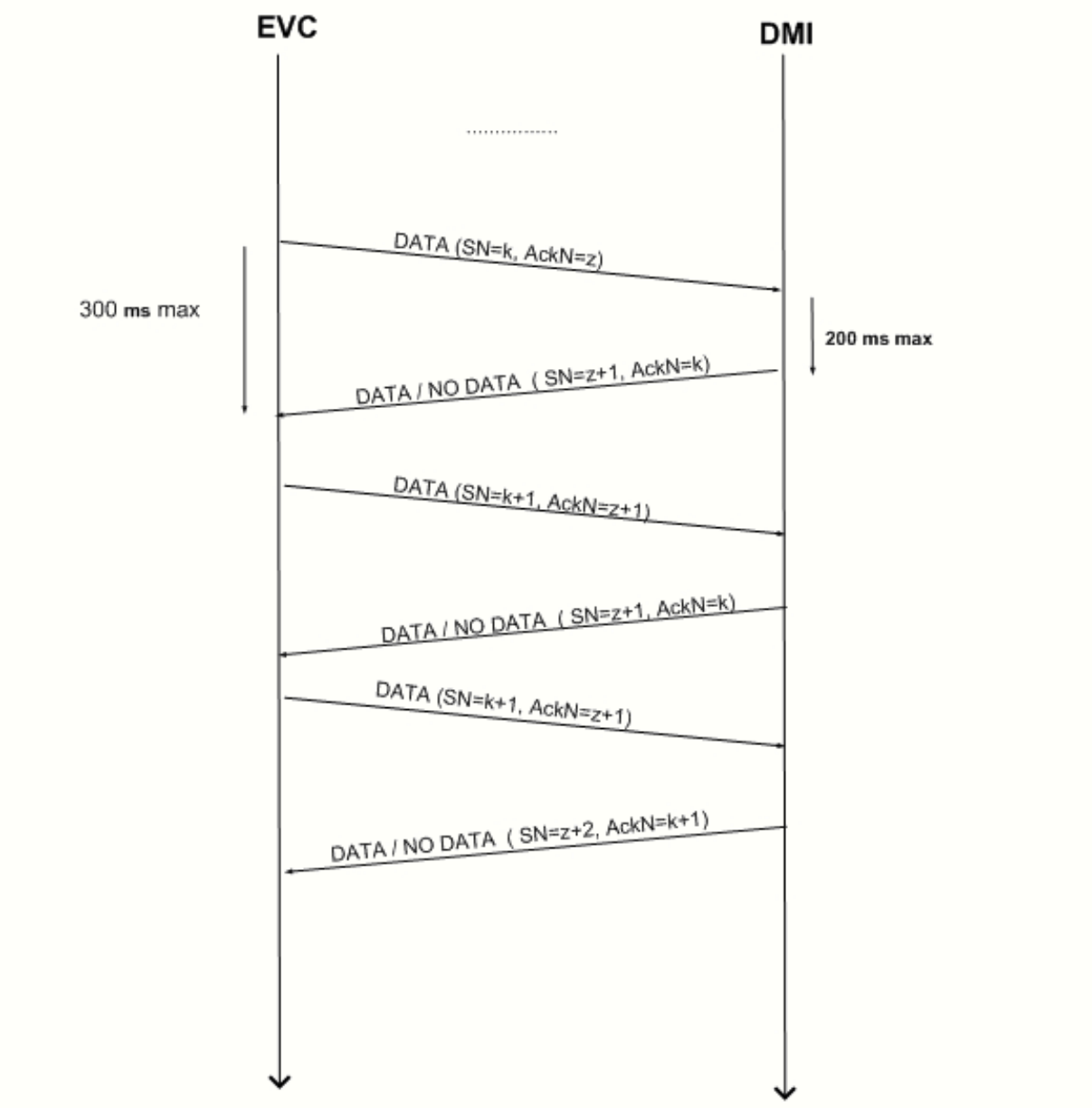


Figure 5 Half-duplex PDU corruption from EVC to DMI - Negative Acknowledgement

Situation: CRC error detected for first PDU SN=k+1.Last transmitted frame is re-transmitted.
On reception of negative acknowledgement, PDU SN=k+1 is resent immediately by EVC.

F) PDU CORRUPTION FROM DMI TO EVC – REQUEST FOR RETRANSMISSION

The following sequence diagram describes the case in which a PDU from DMI to EVC is corrupted, due to noise or cable malfunction. EVC checks the CRC of the PDU and immediately is able to inform DMI about the invalid reception. This is done simply by confirming the sequence number of the last correctly received PDU.

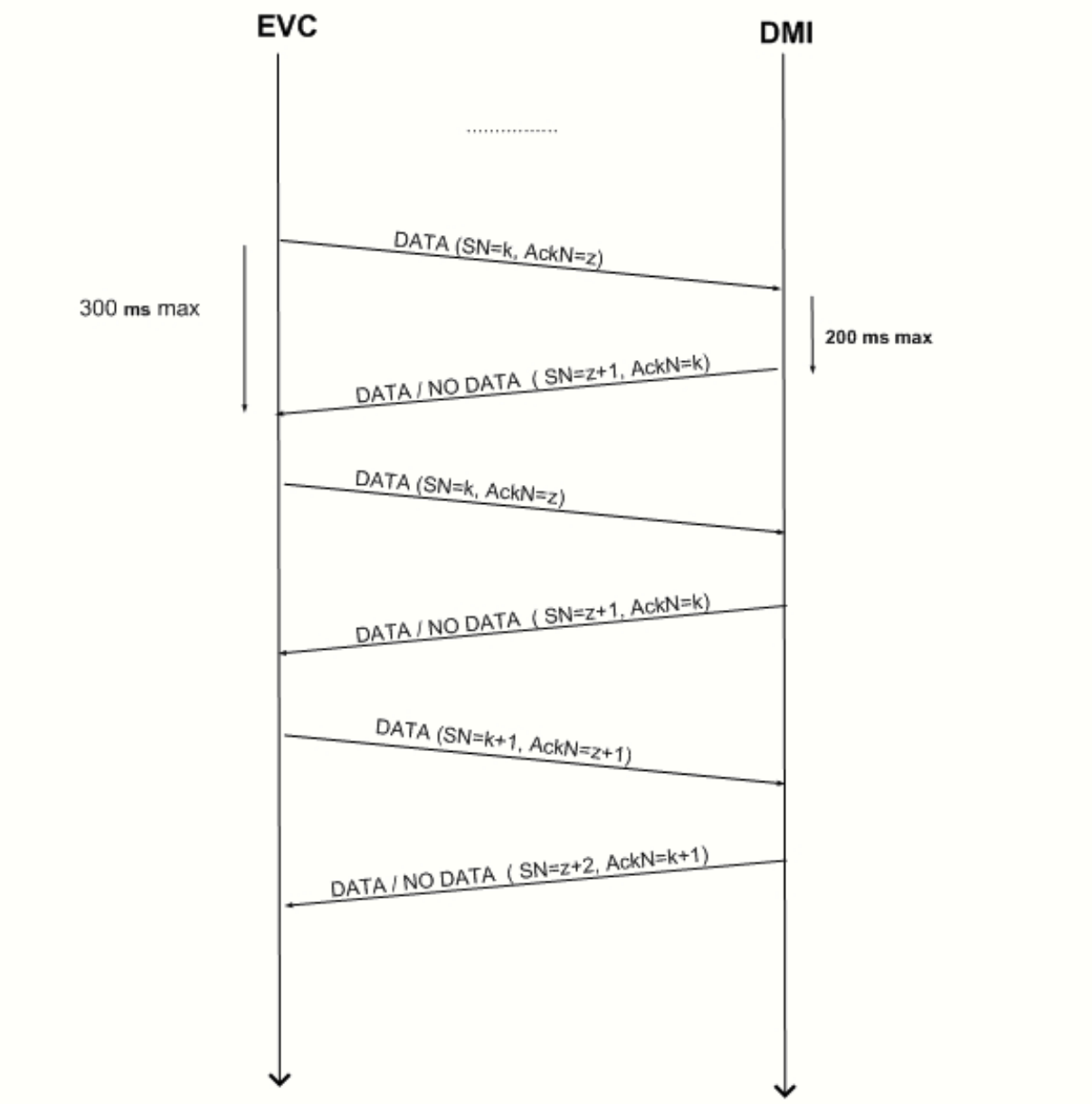


Figure 6 Half-duplex PDU corruption from DMI to EVC – Request for retransmission

G) SLAVE ALGORITHM

The DMI implements the slave algorithm.

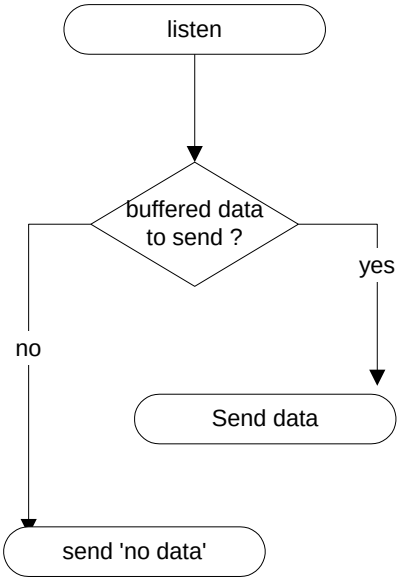


Figure 7 Half-duplex protocol slave flowchart

By default, the slave blocks all send request made by the upper layer and endlessly attempts to read data from the lower layer.

When **DATA** (0x00) is received: the heading bytes are removed from the frame and the rest of the frame is buffered. It stops receiving data from the lower protocol layer and allows exactly one send request made by the upper layer to be fulfilled. If there is no send request pending at that precise point in time, the slave sends **NO_DATA** (0x01) to the underlying layer and switches back to reception mode. If there is a blocked send request pending, it constructs and sends a **DATA** protocol frame starting with byte 0x00 followed by the data frame to be sent.

Timeouts:
They are associated to the sending of the protocol frames. The Master expects getting them within 300 milliseconds from the sending of the matching DATA frame. The Slave has to ensure these data are not sent outside this time window as the Master will not expect these data any longer (when serial links are used, that means, the data may then collide on the RS485 link with these unexpected data).

To address this issue, the Slave associates a use-by date to any outgoing protocol frame sent. For protocol frames the use-by date is equal to the DATA frame reception time plus 100 milliseconds. There is 200 milliseconds large safety margin to regard transmission time over the lower layer and the estimated processing time between the time of reception of the data by the lower layer and the time of the response sending by the lower layer.

For example, considering a frame of 120 bits (15 bytes) and a serial link speed of 19200 bauds (about 2 bytes/ms), a transmission time of 7 ms shall be considered.

H) MASTER ALGORITHM

The master can send and receive data at any moment.

When a send request is done by the upper layer, the master constructs and sends the **DATA** protocol frame starting followed by the data frame to be sent. In the meantime any receive request made by the upper layer is blocked.

The Master receives new data as response to **DATA** protocol frame using the lower protocol layer.

Whenever a **DATA** frame is received, the heading byte is removed from the frame and the rest of the frame is made available for the upper layer.

The Master expects receiving any answer within 300 milliseconds from the sending of the **DATA** frame. If the timeout expires and no data has been received, the Master assumes that the answer will never arrive and any following unexpected frame will be ignored.

I) SAFETY AND INTEGRITY FUNCTIONALITY

A CRC 32 is computed from and appended to the application-level data frame.

The CRC used is the 32-Bits CCITT-CRC computed with the polynomial used for IEEE 802.3, 4, 5, 6 (Ethernet, Token Passing Bus, Token Ring, Metropolitan Area Network standard protocols, as well as in the FDDDI protocol.
$g(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1 = 0x04C11DB7$

The initialization register for CRC computation (32 Bits) is set to FFFFFFFFH. The CRC is used as an error detection code.

5 COMMUNICATION DATA

5.1 PACKETS OVERVIEW

5.1.1 From EVC to DMI

NID	Packet Name	Periodicity	Short description
1	DMI_DYNAMIC	300 ms	Dynamic Data, like current train speed, target data…
2	DMI_MENU_REQUEST	Sporadic	Request to enable/disable driver menus and buttons
3	DMI_ENTRY_REQUEST	Sporadic	Request to input certain data (driver id, train data…)
4	DMI_EVC_CODED_TRAIN_DATA	Sporadic	EVC Coded Train Data to be validated by EVC
5	DMI_TEXT_MESSAGE	Sporadic	Predefined or Plain Text Message
6	DMI_TRACK_DESCRIPTION	Sporadic	Description of track (speed and gradient profile…)
7	DMI_IDENTIFIER_REQUEST	Sporadic	Request for the DMI version information
8	DMI_ICONS	Sporadic	Request to display one or more icon(s) in any area.
9	DMI_SYSTEM_VERSION	Sporadic	Display the EVC operated system version
10	DMI_DISPLAY_CONTROL	Sporadic	Enables to control state of DMI display (cabin activation).
11	DMI_EVC_LEVEL_DATA	Sporadic	Gives the list of available levels.
12	DMI_EVC_RADIO_NET_DATA	Sporadic	Gives the list of available radio network
13	DMI_EVC_VBC_DATA	Sporadic	Gives the list of VBCs stored on-board
14	DMI_USER_FUNCTION	Sporadic	Provide user specific function
15	DMI_EVC_CODED_SETVBC_DATA	Sporadic	Coded VBC data (set) to be validated by driver
16	DMI_EVC_CODED_RMVBC_DATA	Sporadic	Coded VBC data (remove) to be validated by driver
17	DMI_NTC_INPUT	Sporadic	Input information related to NTC
18	DMI_NTC_DATA_ENTRY	Sporadic	Description of NTC data entry window

5.1.2 From DMI to EVC

NID	Packet Name	Periodicity	Short description
128	DMI_STATUS	600 ms	Activity status of the DMI
129	DMI_DRIVER_REQUEST	Sporadic	Driver Request or Acknowledgement (other than text)
130	DMI_TEXT_MESSAGE_ACK	Sporadic	Text Message Acknowledgment
131	DMI_TRAIN_DATA_ACK	Sporadic	Train Data Acknowledgment (Validation)
132	DMI_IDENTIFIER	Sporadic	Version information of the DMI
133	DMI_ICON_ACK	Sporadic	Icon Acknowledgment
134	DMI_SOUND_STATUS	Sporadic	Indication of audible information on DMI
135	DMI_SET_VBC_DATA	Sporadic	Set Virtual Balise Cover
136	DMI_RM_VBC_DATA	Sporadic	Remove Virtual Balise Cover
137	DMI_RADIO_NET_DATA	Sporadic	Entered radio network
138	DMI_SETVBC_DATA_ACK	Sporadic	VBC data (set) acknowledgement
139	DMI_RMVBC_DATA_ACK	Sporadic	VBC data (remove) acknowledgement
140	DMI_NTC_OUTPUT	Sporadic	Output information related to NTC

5.1.3 Two-way packets

NID	Packet Name	Periodicity	Short description
192	DMI_DRIVER_IDENTIFIER	Sporadic	Default or Entered Driver Identifier
193	DMI_TRAIN_RUNNING_NUMBER	Sporadic	Default or Entered Train Running Number
194	DMI_SR_DATA	Sporadic	Default or Entered Staff Responsible Data
195	DMI_TRAIN_DATA	Sporadic	Default or Entered Train Data
196	DMI_ADHESION_FACTOR_DATA	Sporadic	Default or Entered Adhesion Factor Data

197	DMI_LEVEL_DATA	Sporadic	Default or entered ETCS Level
199	DMI_RBC_DATA	Sporadic	Default or entered RBC contact info (RBC data and radio network ID)

5.2 VARIABLES OVERVIEW

5.2.1 Naming Convention

The naming of the variables follows a logical pattern. All names are starting with DMI_ and followed with a prefix in relation to the type of data. The following types are used:

A_ acceleration
D_ distance
G_ gradient in per mill
L_ length
M_ miscellaneous
N_ number
NC_ class number
NID_ identity number
Q_ qualifier
O_ location in odometer co-ordinates
T_ time/date
V_ speed
X_ text

5.2.2 DMI_NID_STM

The DMI_NID_STM (NID_NTC) is one of the most important variables for the management of STMs. It is used to uniquely identify a STM. The following table describes the values associated to each STM.

Values	Country	System	Comments	Confirmed by
0	Spain	ASFA	Used in Zaragoza-Huesca ERTMS TRK-L1	Alstom
1	Netherlands	ATB	(The two versions of ATB (EG and NG) can be seen as a group, not to be activated individually. But this number shall also be used by an STM with the EG function only)	Alstom, Siemens, Ansaldo, CER, Bombardier
2	Spain	ASFA AVE		Unisig
3	Spain	LZB Spain (C)	Including ASFA AVE function for conventional lines	Unisig
5	Belgium	TBL 1	Includes Crocodile	Alstom, Siemens, Ansaldo, CER
6	Germany, Austria	INDUSI		Siemens, CER, Thales, Bombardier
7	Belgium	TBL 2/3	Including crocodile and TBL 1	Alstom, Siemens, Ansaldo, CER
8	France	KVB		Alstom, Ansaldo, CER
9	Germany, Austria	LZB	PZB/LZB Classic and LZB CE (national group)	Ansaldo, CER
10	Spain	LZB	Includes AFSA, AVE for high speed lines	Siemens, CER,

				Thales, Bombardier
11	Italy	SCMT	Used to manage level transitions	Alstom, Ansaldo, CER
12	Luxembourg	MEMOR II+	crocodile based system for Luxemburg	Unisig
14	France, Belgium, UK, China, Korea	TVM		Ansaldo, CER
15	Italy	BACC		Unisig
16	Italy	RSDD		Unisig
17	Hungary	EVM		Thales
18	Belgium	Crocodile		Ansaldo, Siemens, Unisig
19	Spain	EBICAB 900	TBS includes ASFA function	
20	UK	TPWS/AWS	Assumed that both names used for identical systems or can be seen as a group, not to be activated individually	Unisig
22	Norway, Sweden	ATC2	Ebicab 700 (N+S) , L10000 (N+S)	Ansaldo, Unisig
23	Finland	EBICAB 900		Unisig
24	Poland	EBICAB 900 (PL)		Unisig
25	Korea	KNR ATS		Bombardier
26	Poland	SHP		Unisig
28	Belgium	TBL1+		Service public fédéral Mobilité et Transports
30	Denmark	ZUB 123		Unisig
31	Switzerland	ZUB 121 / Signum		Ansaldo
32	France	RPS	French version of Crocodile	Unisig
33	Czech Republic / Slovakia	LS		Ansaldo, Unisig, CER
34	Portugal	EBICAB 700 (P)	not same system as e.g Spain and Portugal	Unisig
35	UK	selcab	for conventional lines	Unisig
36	Romania and others	INDUSI I 60		Unisig
37	UK	TBL		Unisig
39	Balticum	ALSN	It is also installed in Russian federation and Belarus	Unisig
40	Bulgaria	EBICAB		Unisig
45	China	CTCS-2	Used for the line between cities Wuhan and Guangzhou	Bombardier
255	Reserved for multicast	All	Unisig Subset 035, time and odometer multicast	Alstom, Ansaldo, Unisig

Note: Updated lists are maintained in /9/.

5.3 VARIABLES / AREAS CORRELATION TABLE

Section	Title	Field	Information	Variables
8.2.1.1	Speed dial	B0	Scale speed	
8.2.1.2	Current train speed pointer	B1	Train speed	DMI_V_TRAIN DMI_NID_C DMI_NID_C_UNKNOWN
			State	DMI_M_MODE DMI_M_SUPSTATUS DMI_M_WARNING DMI_V_RELEASE DMI_V_PERMITTED DMI_V_TARGET
8.2.1.3	Current train speed digital	B1	Train speed	DMI_V_TRAIN DMI_NID_C DMI_NID_C_UNKNOWN
			State	DMI_M_MODE DMI_M_SUPSTATUS DMI_M_WARNING DMI_V_RELEASE DMI_V_PERMITTED DMI_V_TARGET
8.2.1.4	Circular Speed Gauge	B2	Permitted speed	DMI_V_PERMITTED
			Intervention speed	DMI_V_INTERVENTION
			Target speed	DMI_V_TARGET
			State	DMI_M_MODE DMI_M_SUPSTATUS DMI_M_WARNING DMI_V_RELEASE DMI_V_PERMITTED DMI_V_TARGET
8.2.1.5	Basic Speed Hook(s)	B2	Hook at v-perm	DMI_V_PERMITTED
			Hook at v-target	DMI_V_TARGET
			Visibility	DMI_M_MODE DMI_M_SUPSTATUS DMI_V_PERMITTED DMI_V_TARGET
8.2.1.6	Release speed	B2	Speed	DMI_V_RELEASE
		B6	State	DMI_M_MODE DMI_M_SUPSTATUS DMI_V_RELEASE DMI_V_PERMITTED
8.2.2.1	Distance to target bar	A3	Distance	MMI_O_BRAKETARGET
8.2.2.2	Distance to target digital	A2	Distance	MMI_O_BRAKETARGET
8.2.2.3	Emergency/Service Brake Intervention	C9	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.2.3.1	Mode information	B7	Icon	DMI_M_MODE
8.2.3.1	Mode acknowledgement	C1	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.2.3.1	Override symbol	C1	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.2.3.2	Level information	C8	Icon	DMI_M_LEVEL DMI_NID_STM
8.2.3.3	Track Ahead Free information	D	Icon	DMI_M_ENTRY_REQUEST

8.2.3.4	Text messages	E5-E9	Message	DMI_Q_TEXT DMI_X_TEXT DMI_Q_TEXTCLASS DMI_Q_TEXTCONFIRM
8.2.3.4	Text messages scroll buttons	E10-E11	Activation	DMI_Q_TEXT DMI_X_TEXT DMI_Q_TEXTCLASS DMI_Q_TEXTCONFIRM
8.2.3.5	Actual order and announcement	B3-B5	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.2.3.6	Adhesion Factor Indication	A4	Icon	DMI_X_ICON_REF DMI_X_AREA_NAME
8.2.3.7	Level Crossing “not protected” Indication	B3-B5	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.3.3	Distance scale	D1-D7	Visibility	DMI_M_MODE
8.3.4	Orders and announcement	D2	Icon	DMI_M_TRACKCOND
			Item distance	DMI_O_TRACKCOND DMI_O_TRAIN
			Visibility	DMI_M_MODE
		D3	Icon	DMI_M_TRACKCOND
			Item distance	DMI_O_TRACKCOND DMI_O_TRAIN
			Visibility	DMI_M_MODE
		D4	Icon	DMI_M_TRACKCOND
			Item distance	DMI_O_TRACKCOND DMI_O_TRAIN
			Visibility	DMI_M_MODE
8.3.5	Gradient Profile	D5	Gradient	DMI_G_A DMI_Q_GDIR
			Item distance	DMI_O_GRADIENT
			Visibility	DMI_M_MODE
8.3.6	Speed profile discontinuity symbols	D6 / D7	Speed discontinuity	DMI_V_STATIC DMI_M_MODE DMI_O_VLOA (speed limit a EOA)
			Item distance	DMI_O_STATIC
			Visibility	DMI_M_MODE
8.3.7	Planning Area Speed Profile (PASP)	D7	Speed discontinuity	DMI_V_STATIC DMI_M_MODE
			Item distance	DMI_O_STATIC DMI_O_LOA (end limit of PASP) DMI_M_MODE
			Visibility	DMI_M_MODE
8.3.8	Indication Marker	D7	Distance	DMI_O_BCSP
			Visibility	DMI_M_MODE DMI_O_LOA DMI_O_TRAIN
8.4.1	Communication session indication	E1	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME
8.4.2	Reversing permitted indication	C1	Icon	DMI_M_ICON_CONTROL_FLAG DMI_M_ICON_FLASHING_FREQ DMI_X_ICON_REF DMI_X_AREA_NAME

8.4.3	Local time	G13	Time	DMI_T_CLOCK (DMI_DYNAMIC)
8.4.4	Geographical position	G12	Distance	DMI_O_KP_BALISE_TRACK_KILOMETER DMI_O_KP_DIST_TO_BALISE DMI_M_KP_FLAG
			Visibility	DMI_M_MODE
	"Main" menu buttons	DFG DGI	State	DMI_M_AVAILABLE_MENU
	"Override" menu buttons	DFG DGI	State	DMI_M_AVAILABLE_MENU
	"Spec" menu buttons	DFG DGI	State	DMI_M_AVAILABLE_MENU
	"Settings" menu buttons	DFG DGI	State	DMI_M_AVAILABLE_MENU
	"RBC contact" menu buttons	DFG DGI	State	DMI_M_AVAILABLE_MENU
	Hourglass ("Main" menu)	DFG DGI	State	DMI_M_AVAILABLE_MENU
	"Enter Data"/"Select type" button in train data entry page		Visibility	DMI_M_AVAILABLE_MENU
	Main window	DFG DGI	Visibility	DMI_M_ENTRY_REQUEST
	Driver ID entry page	DFG DGI	Visibility	DMI_M_ENTRY_REQUEST
	Train running number entry page	DFG DGI	Visibility	DMI_M_ENTRY_REQUEST
	ETCS level entry page	DFG DGI	Visibility	DMI_M_ENTRY_REQUEST
	RBC contact menu page	DFG DGI	Visibility	DMI_M_ENTRY_REQUEST
	Train data default values	DFG DGI	Field values	DMI_TRAIN_DATA packet
	Level data default values	DFG DGI	Field values	DMI_LEVEL_DATA packet
	Level buttons in level entry page		Visibility	DMI_M_LEVEL DMI_NID_STM
			State	DMI_Q_LEVEL_INHIBITED
	RBC data default values	DFG DGI	Field values	DMI_RBC_DATA packet
	SR data default values	DFG DGI	Field values	DMI_SR_DATA packet
	Driver ID default value	DFG DGI	Field values	DMI_DRIVER_IDENTIFIER packet
	Train running number default value	DFG DGI	Field values	DMI_TRAIN_RUNNING_NUMBER packet
	Adhesion factor default value	DFG DGI	Field values	DMI_ADHESION_FACTOR_DATA packet

5.4 VARIABLES / SOUNDS CORRELATION TABLE

§ ERA v2.3 specification	Packet	Sound	Variable
7.2.3.3	DMI_DYNAMIC	S2 - Warning	DMI_M_WARNING
7.3.1.2	DMI_DYNAMIC	S-Info	DMI_M_SUPSTATUS
7.4.3.3	DMI_DYNAMIC	S1 - Overspeed	DMI_M_WARNING
7.4.4.3	DMI_DYNAMIC	S2 - Warning	DMI_M_WARNING
5.4.1.5	DMI_ICONS	S-Info	DMI_M_ICON_CONTROL_FLAG DMI_X_ICON_REF DMI_X_AREA_NAME
5.4.1.5	DMI_TEXT_MESSAGE	S-Info	DMI_QTEXTCONFIRM DMI_QTEXTCLASS
8.2.2.3.6	DMI_DYNAMIC	S-Info	DMI_M_WARNING
8.2.3.4.7	DMI_TEXT_MESSAGE	S-Info	DMI_QTEXTCONFIRM DMI_QTEXTCLASS
12.3.2.1	DMI_DYNAMIC	S1 - Overspeed	DMI_M_WARNING
12.3.3.1	DMI_DYNAMIC	S2 - Warning	DMI_M_WARNING

5.5 IDENTIFICATION PROCEDURE

The first packet send by EVC shall be DMI_IDENTIFIER_REQUEST.

The DMI sends DMI_IDENTIFIER as response to DMI_IDENTIFIER_REQUEST.

The DMI does not send any data before it has sent DMI_IDENTIFIER to the EVC.

The EVC does send any data before it has received DMI_IDENTIFIER from the DMI and the information hold in DMI_IDENTIFIER are compatible with a further operation and communication with the DMI according to the Onboard system requirements.

In order to allow version management the format of DMI_IDENTIFIER_REQUEST and DMI_IDENTIFIER must be fixed. The communication protocol version (version of this document) is indicated by:

- DMI_M_IMPL_VERSION in DMI_IDENTIFIER_REQUEST.
- DMI_M_PROTO_VERSION_MAJOR_CODE.
- DMI_M_PROTO_VERSION_MINOR_CODE in DMI_IDENTIFIER.

5.6 ERROR MANAGEMENT

5.6.1 Invalid packets and variables

Packets with invalid DMI_L_PACKET are discarded and error code is returned to EVC.
Packets containing variable values which are out of the valid range are discarded and error code is returned to EVC.
Packets which cannot be decoded (inconsistent structure, i.e. DMI_N_ITER not compatible with the amount of received bytes) are discarded.

5.6.2 Sending timeouts

On application level, a configurable timeout, which is 1s by default, is associated by the DMI with the sending of a packet.

When a packet could not be send over the lower protocol layers, this packet is discarded without any error reported to EVC.

5.7 DMI_DYNAMIC

This packet is transmitted every 300 milliseconds by the EVC to the DMI. It contains all dynamic data, such as the current train speed, position and target data.

	VARIABLE	Name	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	1	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	16	24	65535	1 bit		
3	DMI_T_CLOCK		Clock time Seconds elapsed in day since 00:00	32	0	4294967.295 s	1 ms		
4	DMI_V_TRAIN		Estimated train speed at the actual time, without tolerance added. Used to set the needle speed position.	10	0	600	1 km/h	1023	speed unknown
5	DMI_X_VTRAIN_DIGIT1	Text String Element	One character representing the first speed digit. It is a space char if speed is less than 100 km/h. Used to display the current train speed value.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
6	DMI_X_VTRAIN_DIGIT2	Text String Element	One character representing the second speed digit. It is a space char if speed is less than 10 km/h. Used to display the current train speed value.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
7	DMI_X_VTRAIN_DIGIT3	Text String Element	One character representing the third speed digit.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet		

			Used to display the current train speed value.				#1)		
8	DMI_O_TRAIN	Location of the train	Front train location in odometer co-ordinates, estimated without tolerance added. The odometer co-ordinates are always counted upwards (positive) and start at 0 when powering on the EVC	32	0	42949672.95 m	0.01m		
9	DMI_O_BRAKETARGET	Distance to brake target (A2)	Distance to brake target (A2). Distance from the train front to the position, without corrected tolerance, of the next brake target point, which has influence on the braking curve. Used to display the target distance bar information.	32	0	42949672.94 m	0.01m	4294967295	Target unknown
10	DMI_X_OBRAKETARGET_DIGIT1	Text String Element	One character representing the first distance digit. It is a space char if distance is less than 10000m. Used to display the target digital bar information.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
11	DMI_X_OBRAKETARGET_DIGIT2	Text String Element	One character representing the second distance digit. It is a space char if distance is less than 1000m. Used to display the target digital bar information.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
12	DMI_X_OBRAKETARGET_DIGIT3	Text String Element	One character representing the third distance digit. It is a space char if distance is less than 100 m. Used to display the target digital bar information.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
13	DMI_X_OBRAKETARGET_DIGIT4	Text String Element	One character representing the fourth distance digit. It is a space char if distance is less than 10 m. Used to display the target digital bar information.	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
14	DMI_X_OBRAKETARGET_DIGIT5	Text String Element	One character representing the fifth distance digit. Used to display the target digital bar	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet		

			information.				#1)		
15	DMI_V_TARGET	Target speed	Speed to be applied at the next restrictive static speed profile discontinuity which has influence on the braking curve.	10	0	600	1 km/h	1023	no target
16	DMI_V_PERMITTED	Permitted speed	Permitted speed according to current ATP rules	10	0	600	1 km/h	1023	speed unknown
17	DMI_V_RELEASE	Release speed	Release speed at the EOA, shown when release speed is lower than permitted speed	10	0	600	1 km/h	1023	speed unknown
18	DMI_O_BCSP	Brake curve starting point	Position, in odometer co-ordinates, without tolerance correction, of the next brake starting point for the next brake target.	32	0	42949672.94 m	0.01m	4294967295	no target
19	DMI_V_INTERVENTION	Intervention speed	Intervention speed	10	0	600	1 km/h	1023	speed unknown
20	DMI_M_MODE	ETCS mode	Onboard operating mode.	4	0	14	enum.	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15	Full Supervision On Sight Staff Responsible Shunting Unfitted Sleeping Standby Trip Post Trip System Failure Isolation Non-Leading Limited Supervision STM National Reversing Passive Shunting
21	DMI_M_LEVEL	ETCS level	Onboard level of operation	3	0	4	enum.	0 1	Level 0 Level STM specified by

								2 3 4 5-6 7	NID_STM Level 1 Level 2 Level 3 Spare unknown
22	DMI_NID_STM	NID STM	Number identifier of STM. ONLY TRANSMITTED if Level STM specified in DMI_M_LEVEL.	8	0	255	enum.	See §6.2.2	See §6.2.2
23	DMI_NID_C	Country Id		10	0	1023	enum	See SRS	See SRS
24	DMI_NID_C_UNKNOWN	Country Id Unknown Flag	Tells whether the value of the country Id (DMI_NID_C) is known or not. Unknown value of DMI_NID_C shall be ignored	1	0	1	enum	0 1	DMI_NID_C is valid DMI_NID_C is unknown
25	DMI_M_WARNING	Warning indicator	Status of indication, warning and intervention	4	0	15	bit mask	0 bit 1 bit 2 bit 3 bit 4	no warning intervention warning over speed warning indication
26	DMI_M_SUPSTATUS	Supervision status	Current supervision status	4	0	15	bit mask	0 1 2 3 4-14 15	CSM PIM TSM RSM <spare> Unknown
27	DMI_O_LOA	Position of LOA (0.01m)	Position, in absolute odometer co-ordinates, without tolerance correction, of the limit of authority.	32	0	42949672.94 m	0.01m	4294967295	no movement authority
28	DMI_V_LOA	Speed at LOA	Speed at the Limit Of Authority	10	0	600	1 km/h	1023	speed unknown
29	DMI_O_KP_BALISE_TRACK_KILOMETER	Kilometer point of the reference balise on the track	Kilometer point of the reference balise on the track.	32	0	42949672.94 m	0.01m	4294967295	Unknown kilometer point
30	DMI_O_KP_DIST_TO_BALISE	Distance of the train to the reference	Distance to the reference balise.	32	0	42949672.94 m	0.01m	4294967295	Unknown kilometer point.

		balise.							
31	DMI_M_KP_FLAG	Direction to balise		2	0	3	enum.	0 1 2 3	Nominal Reverse Unknown direction Unknown kilometer point
32	DMI_O_DIST_TO_TSA	Distance between the front of the train to the tunnel area	ONLY TRANSMITTED if M_ERTMS_VERSION >= 3.2	32	1	2^32-2	0.01m	2^32-1 0	Unknown distance Inside tunnel area

Notes

- DMI_V_TRAIN: If DMI_NID_C is 2 and DMI_NID_C_UNKNOWN is 0, the speed needle displays the value of the passed speed converted to miles per hour (MPH). In all other cases, the displayed speed needle corresponds to the value passed in DMI_V_TRAIN.

- DMI_X_VTRAIN_DIGIT1, DMI_X_VTRAIN_DIGIT2, DMI_X_VTRAIN_DIGIT3 : These variables are mainly aimed for embedded safety-related systems for the sake of diverse storing of safety related data. When the current speed digits are all spaces, the displayed digital speed value is derived from DMI_V_TRAIN converted to the national-specific speed unit according to DMI_NID_C when DMI_NID_C_UNKNOWN is 0. In all other cases, the displayed digital speed corresponds to the value passed in the characters MI_X_VTRAIN_DIGIT1, DMI_X_VTRAIN_DIGIT2, DMI_X_VTRAIN_DIGIT3.

Note: Miles per hour is the only national-specific speed unit currently supported by the DMI (DMI_NID_C = 2).

- DMI_M_WARNING: The respective bit values of M_WARNING are interpreted as follows:
  - Indication: this bit is set whenever v-train > v-indication (see indication curve calculation).
  - Over speed warning: this bit is additionally set whenever v-train > v-permitted.
  - Warning: this bit is set whenever v-train > v-warning. When this bit is set, the over speed warning bit is clear and the indication bit is still set.
  - Intervention: this bit is set whenever v-train > v-intervention. When this bit is set, the over speed warning and warning bits are clear and the indication bit is still set.

Thus, only one bit can be set at a time, except the indication bit which can be set together with any other bit.

The display behaviour is undefined in any combination of bits not specified above.

5.8 DMI_MENU_REQUEST

This packet is sent by the EVC to enable and disable access to input fields and menus on the DMI, according to the actions allowed to be done by the driver. The request options are indicated through a bit pattern in the DMI_M_AVAILABLE_MENU variable.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	2	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_AVAILABLE_MENU	Enabled Driver Requests Mask	Mask giving the list of enabled or disabled menus/buttons		32	0	65535	bit mask	see table below	

DMI_M_AVAILABLE_MENU	bit 1	Menu button start of mission	X
	bit 2	Menu button shunting	X
	bit 3	Menu button shunting exit	X
	bit 4	Menu button non-leading	X
	bit 5	Menu button exit-non leading	Spare
	bit 6	Menu button maintain shunting	X
	bit 7	Menu button driver ID	X
	bit 8	Menu button train running number	X
	bit 9	Menu button ETCS level	X
	bit 10	Menu button train data modification	X
	bit 11	Menu button train data view	X
	bit 12	Menu button staff responsible data	X
	bit 13	Menu button language selection	X

	bit 14	Menu button override EOA	X
	bit 15	Menu button override route suitability	Spare
	bit 16	Menu button adhesion factor	X
	bit 17	Menu button system version	X
	bit 18	Menu button volume	X
	bit 19	Menu button luminance	X
	bit 20	Menu button train integrity	X
	bit 21	Menu button isolation	X
	bit 22	Show hourglass (hide hourglass if not set)	X
	bit 23	Menu button use short number	X
	bit 24	Menu button enter RBC data	X
	bit 25	Menu button radio network ID	X
	bit 26	Menu button contact last RBC	X
	bit 27	Button “switch” for train data	X
	bit 28	Fix train data entry (1) flexible train data (0)	X
	bit 29	Menu button Set VBC	X
	bit 30	Menu button Remove VBC	X
	bit 31		Spare
	bit 32		Spare

Table 1 DMI_AVAILABLE_MENU

5.9 DMI_ENTRY_REQUEST

This packet is sent by the EVC to force the driver to input certain information.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	3	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_ENTRY_REQUEST	Action required	Tells which data have to be input by the driver. The request has to be carried out immediately by the DMI / driver		8	0	255	enum.		See second table below

DMI_M_ENTRY_REQUEST	Special / Reserved Values	Meaning of S/R Values
	0	Enter / revalidate driver identifier
	1	Enter / revalidate train running number
	2	Enter / revalidate ETCS level
	3	Enter RBC contact menu
	4	Validate train data
	5	Enter NTC data
	6	spare
	7	Enable the track ahead free by the driver
	8	Disable the Track Ahead free page
	9	Show “main window”
	10	Hide “main window”
	11	Hide adhesion factor entry window
	12	Hide staff responsible entry window
	13	Show “Set VBC validation” window
	14	Show “Remove VBC validation” window
	15-255	spare

Table 2 DMI_M_ENTRY_REQUEST

DMI_EVC_CODED_TRAIN_DATA

This packet contains the validated train data transmitted by the EVC in response to the original train data entered by the driver. The coding of each train data fields consists in converting each digit of the numerical field value into its BCD representation stored on 1 byte.

Train data are configurable and therefore the content of this packet depends on the train data configuration. Any static description of this packet is provided as example only. Here is a generic description of this packet format:

- The train data packet always starts with the DMI_NID_PACKET and DMI_L_PACKET as any other packet exchanged between EVC and DMI.
- This packet header is followed by each train data field defined within the train data configuration selected at DMI start-up.
- The fields appear in the sequence order as they are defined within the configuration.
- Each digit of the field numerical value is coded in BCD and stored on one byte. E.g "100" will be stored on 3 bytes whose values are respectively 1, 0 and 0.
- The number of integer digits and the number of decimal digits to be coded are given within the train data configuration via the <format-7-segments>. Each 'X' stands for one digit an each 'x' stands for one decimal digit.

There is one exception: for a train data field defined as a set of predefined values, the number of digits is computed from the maximum number of digits which could be represented according to bit coding of that train data field inside DMI_TRAIN_DATA packet. For example, if a train data field is coded on 8 bits, maximum number of digits will be '3' as the maximum coded value is 255 and is represented on 3 digits.

The following packet description is the default implementation provided for the train data as described in document /1/ and /8/ for ERTMS.

	VARIABLE	Name	Description	Remk	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	4	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_X_NCTRAIN1	NCTRAIN digit 1	Train Category digit 1		8	0	9	BCD value	> 9	unused / invalid

5	DMI_X_NCTRAIN2	NCTRAIN digit 2	Train Category digit 2		8	0	9	BCD value	> 9	unused / invalid
6	DMI_X_NCTRAIN3	NCTRAIN digit 2	Train Category digit 2		8	0	9	BCD value	> 9	unused / invalid
7	DMI_X_NCTRAIN4	NCTRAIN digit 2	Train Category digit 2		8	0	9	BCD value	> 9	unused / invalid
8	DMI_X_NCTRAIN5	NCTRAIN digit 2	Train Category digit 2		8	0	9	BCD value	> 9	unused / invalid
9	DMI_X_LTRAIN1	LTRAIN digit 1	Train Length digit 1		8	0	9	BCD value	> 9	unused / invalid
10	DMI_X_LTRAIN 2	LTRAIN digit 2	Train Length digit 2		8	0	9	BCD value	> 9	unused / invalid
11	DMI_X_LTRAIN 3	LTRAIN digit 3	Train Length digit 3		8	0	9	BCD value	> 9	unused / invalid
12	DMI_X_LTRAIN 4	LTRAIN digit 4	Train Length digit 4		8	0	9	BCD value	> 9	unused / invalid
13	DMI_X_BRAKE_MODEL1	BRAKE MODEL digit 1	Brake model digit 1		8	0	9	BCD value	> 9	unused / invalid
14	DMI_X_BRAKE_MODEL2	BRAKE MODEL digit 2	Brake model digit 2		8	0	9	BCD value	> 9	unused / invalid
15	DMI_X_BRAKE_MODEL3	BRAKE MODEL digit 3	Brake model digit 3		8	0	9	BCD value	> 9	unused / invalid
16	DMI_X_VMAX_TRAIN1	VMAX _TRAIN digit 1	Max. permitted Train Speed digit1		8	0	9	BCD value	> 9	unused / invalid
17	DMI_X_VMAX_TRAIN2	VMAX _TRAIN digit 2	Max. permitted Train Speed digit 2		8	0	9	BCD value	> 9	unused / invalid
18	DMI_X_VMAX_TRAIN3	VMAX _TRAIN digit 3	Max. permitted Train Speed digit 3		8	0	9	BCD value	> 9	unused / invalid
19	DMI_X_MAXLELOAD1	MAXLELOAD digit 1	Axle Load in 0,5 tons (tens, before decimal pt.)		8	0	9	BCD value	> 9	unused / invalid
20	DMI_X_MAXLELOAD2	MAXLELOAD digit 2	Axle Load in 0,5 tons (ones, before decimal pt.)		8	0	9	BCD value	> 9	unused / invalid
21	DMI_X_MAXLELOAD3	MAXLELOAD digit 3	Axle Load in 0,5 tons (hundredth, after decimal pt.)		8	0	9	BCD value	> 9	unused / invalid
22	DMI_X_MAIRTIGHT1	MAIRTIGHT digit 1	Train fitted with Airtight System		8	0	9	BCD value	> 9	unused / invalid
23	DMI_X_MAIRTIGHT2	MAIRTIGHT digit 2	Train fitted with Airtight System		8	0	9	BCD value	> 9	unused / invalid

24	DMI_X_MAIRTIGHT3	MAIRTIGHT digit 3	Train fitted with Airtight System		8	0	9	BCD value	> 9	unused / invalid
25	DMI_X_MLOADINGGAUGE1	MLOADINGGAUGE digit 1	Loading gauge		8	0	9	BCD value	> 9	unused / invalid
26	DMI_X_MLOADINGGAUGE 2	MLOADINGGAUGE digit 2	Loading gauge		8	0	9	BCD value	> 9	unused / invalid
27	DMI_X_MLOADINGGAUGE 3	MLOADINGGAUGE digit 3	Loading gauge		8	0	9	BCD value	> 9	unused / invalid

5.10 DMI_EVC_CODED_SETVBC_DATA

This packet contains the validated VBC data transmitted by the EVC in response to the original VBC data entered by the driver. The coding of VBC data fields consists in converting each digit of the numerical field value into its BCD representation stored on 1 byte.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	15	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_X_VBC1	VBC CODE digit 1	Code for Virtual Balise Cover digit 1		8	0	9	BCD value	> 9	unused / invalid
5	DMI_X_VBC2	VBC CODE digit 2	Code for Virtual Balise Cover digit 2		8	0	9	BCD value	> 9	unused / invalid
6	DMI_X_VBC3	VBC CODE digit 3	Code for Virtual Balise Cover digit 3		8	0	9	BCD value	> 9	unused / invalid
7	DMI_X_VBC4	VBC CODE	Code for Virtual Balise Cover		8	0	9	BCD value	> 9	unused / invalid

		digit 4	digit 4							
8	DMI_X_VBC5	VBC CODE digit 5	Code for Virtual Balise Cover digit 5		8	0	9	BCD value	> 9	unused / invalid
9	DMI_X_VBC6	VBC CODE digit 6	Code for Virtual Balise Cover digit 6		8	0	9	BCD value	> 9	unused / invalid
10	DMI_X_VBC7	VBC CODE digit 7	Code for Virtual Balise Cover digit 7		8	0	9	BCD value	> 9	unused / invalid
11	DMI_X_VBC8	VBC CODE digit 8	Code for Virtual Balise Cover digit 8		8	0	9	BCD value	> 9	unused / invalid

5.11 DMI_EVC_CODED_RMVBC_DATA

This packet contains the validated VBC data transmitted by the EVC in response to the original VBC data entered by the driver. The coding of VBC data fields consists in converting each digit of the numerical field value into its BCD representation stored on 1 byte.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	16	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_X_VBC1	VBC CODE digit 1	Code for Virtual Balise Cover digit 1		8	0	9	BCD value	> 9	unused / invalid
5	DMI_X_VBC2	VBC CODE digit 2	Code for Virtual Balise Cover digit 2		8	0	9	BCD value	> 9	unused / invalid
6	DMI_X_VBC3	VBC CODE	Code for Virtual Balise Cover		8	0	9	BCD value	> 9	unused / invalid

		digit 3	digit 3							
7	DMI_X_VBC4	VBC CODE digit 4	Code for Virtual Balise Cover digit 4		8	0	9	BCD value	> 9	unused / invalid
8	DMI_X_VBC5	VBC CODE digit 5	Code for Virtual Balise Cover digit 5		8	0	9	BCD value	> 9	unused / invalid
9	DMI_X_VBC6	VBC CODE digit 6	Code for Virtual Balise Cover digit 6		8	0	9	BCD value	> 9	unused / invalid
10	DMI_X_VBC7	VBC CODE digit 7	Code for Virtual Balise Cover digit 7		8	0	9	BCD value	> 9	unused / invalid
11	DMI_X_VBC8	VBC CODE digit 8	Code for Virtual Balise Cover digit 8		8	0	9	BCD value	> 9	unused / invalid

DMI_TEXT_MESSAGE

The EVC sends this packet when a text message shall be displayed to the driver. The text message can be either predefined (DMI_Q_TEXT used to identify predefined text message) or contained in the packet itself (DMI_X_TEXT used for that purpose), but not both within one packet. When there is no predefined text, DMI_Q_TEXT is set to zero and when there is no in plain text, DMI_L_TEXT is set to zero.
DMI_Q_TEXTCONFIRM indicates how the DMI shall handle the message, i.e. whether it is just information, or requires a mandatory acknowledgement, or either an acknowledgement or an invalidating.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.	SRS 2.2.2	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	5	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_NID_MESSAGE_IDENTIFIER	Identifier of the text message	Counter used to identify text messages sent for further acknowledgment or removal.	FFFIS	8	0	255	binary coded		
5	DMI_Q_TEXT	Predefined text message identifier	Predefined messages to be displayed by the DMI.		8	0	255	enum.	see table below	
6	DMI_L_TEXT	Length of text string	L_TEXT defines the length of a text string (L_TEXT * X_TEXT)	SRS 2.2.2	8	0	255	binary value	1 Text String Element	If L_TEXT > 0, there is a free text and Q_TEXT shall be ignored.
7	DMI_X_TEXT (* L_TEXT)	Text String Element	One character of the plain text messages	SRS 2.2.2	8	32	255	ASCII (8 Bits), UTF-8		
8	DMI_Q_TEXTCLASS	Class of		SRS	4	0	1	enum.	0	Eurocab operation text

		message		2.2.2					1 2 3-15	message (important information) Auxiliary information from trackside Important information from trackside spare
9	DMI_Q_TEXTCONFIRM	Reaction to text	Qualifies the need / reaction of text confirmation		4	0	5	enum.	0 1 2 3 4-15	Informative Ack. required Spare Remove text spare

DMI_Q_TEXT Q_TEXT is only relevant if L_TEXT = 0 When Q_TEXTCLASS=0, Q_TEXT gives the index of the operational text message in table 50 of ERA specification as here:	0	Balise read error	X
	1	Communication error	X
	2	Entering FS	X
	3	Entering OS	X
	4	No track condition will be received	Spare
	5	Runaway movement	X
	6	SH refused	X
	7	Trackside not compatible	X
	8	Train data changed	X
	9	Train is rejected	X
	10	Unauthorized passing of EOA / LOA	X
	11	No MA received at level transition	X
	12	SR distance exceeded	X
	13	SH stop order	X
	14	SR stop order	X
	15	Emergency stop	X
	16	Trackside malfunction	X
	17	SH request failed	X
	18	RV distance exceeded	X
	19	No track description	X
	20	STM brake demand	X
	21	Route unsuitable - axle load category	X

	22 Route unsuitable - loading gauge	X
	23 Route unsuitable - traction system	X
	24 Radio network registration failed	X
	25-255	Spare
	0	Level crossing not protected
When Q_TEXTCLASS=1, Q_TEXT gives the id of an auxiliary information from trackside.	1	Acknowledgment
	2-255	Spare
When Q_TEXTCLASS=2, Q_TEXT gives the id of an important information from trackside		

Table 3 DMI_Q_TEXT

5.12 DMI_TRACK_DESCRIPTION

This packet contains the description of the track in front of the train and it size depends on the content of the profiles. It is forwarded to the DMI by the EVC whenever it gets new information from the trackside about the static speed profile, the gradient profile and the track condition profile. This information is presented to the driver in the planning area. When the train reaches the position of a track condition, the according symbol is displayed additionally in the actual order and announcement area (B3/4/5).

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	6	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_N_ITER	Number of static speed discontinuities	Number of (DMI_O_STATIC, DMI_V_STATIC) couples.	SRS 2.2.2	8	0	255			
4	DMI_O_STATIC	Position of new static speed	Position, in absolute odometer co-ordinates, without tolerance correction, of the start location of a speed discontinuity in the most restrictive static speed profile		32	0	42949672.95 m	0.01m		
5	DMI_V_STATIC	New static speed	Speed value at a speed profile continuity		16	0	600	1 km/h		
6	DMI_N_ITER	Number of gradient changes	Number of (DMI_O_GRADIENT, DMI_G_A, DMI_Q_GDIR) triplets.	SRS 2.2.2	8	0	255	binary value		
7	DMI_O_GRADIENT	Position of gradient change	Position, in absolute odometer co-ordinates, without tolerance correction, of the start location of a gradient change		32	0	42949672.95 m	0.01m		

8	DMI_G_A	New gradient value	Gradient value of a part of the track.	SRS 2.2.2	8	0	255	1 ‰	255	Non numerical value telling that the current gradient description ends at O_GRADIENT[N_ITER-1]
9	DMI_Q_GDIR	Up-/Downhill indicator for gradient value	Indicates the direction of the gradient	SRS 2.2.2	8	0	1	enum.	0 1	downhill uphill
10	DMI_N_ITER	Number of track condition information	Number of (DMI_O_TRACKCOND, DMI_M_TRACKCOND)	SRS 2.2.2	8	0	255	binary value		
11	DMI_O_TRACKCOND	Location of the point on the track	Position, in absolute odometer co-ordinates, of the start location of a point where an announcement of a track condition must be shown on the DMI.		32	0	42949672.95 m	0.01m		
12	DMI_M_TRACKCOND	Type of track condition	Type of track condition	SRS 2.2.2	8	0	11	enum.	see table below	
13	DMI_Q_STATIC_EXTEND	Replace / Extend indicator for the current profile.	Replaces or extends the current shown profile with the items in this packet	SRS 2.2.2	1	0	1	enum.	0 Replace 1 Extend	

DMI_M_TRACKCOND Identity of planning information symbol.	1	PL01 (lower pantograph)	X
	2	PL02 (lower pantograph warning)	X
	3	PL03 (Raise pantograph)	X
	4	PL04 (Raise pantograph warning)	X
	5	PL05 (Neutral section)	X
	6	PL06 (Neutral section warning)	X
	7	PL07 (End of neutral section)	X
	8	PL08 (End of neutral section warning)	X
	9	PL09 (Non stopping area)	X
	10	PL10 (Radio hole)	X

	11 PL11 (Inhibition of magnetic shoe brakes)	X
	12 PL12 (Inhibition of magnetic shoe brakes warning)	X
	13 PL13 (Inhibition of eddy current brakes)	X
	14 PL14 (Inhibition of eddy brakes warning)	X
	15 PL15 (Inhibition of regenerative brakes)	X
	16 PL16 (Inhibition of regenerative brakes warning)	X
	17 PL17 (Close air conditioning intake)	X
	18 PL18 (Open air conditioning intake)	X
	19 PL19 (Close air conditioning intake warning)	X
	20 PL20 (Open air conditioning intake warning)	X
	21-23	Spare
	24 PL24 (sound horn)	X
	25 PL25 (change of traction system)	X
	26 PL26 (change of traction system warning)	X
	27 PL27 (change of traction system – AC 25kV 50Hz)	X
	28 PL28 (change of traction system warning – AC 25kV 50Hz)	X
	29 PL29 (change of traction system – AC 15kV 16.7Hz)	X
	30 PL30 (change of traction system warning – AC 15kV 16.7Hz)	X
	31 PL31 (change of traction system – DC 3kV)	X
	32 PL32 (change of traction system warning – DC 3kV)	X
	33 PL33 (change of traction system – DV 1.5kV)	X
	34 PL34 (change of traction system warning – DV 1.5kV)	X
	35 PL35 (change of traction system – DC 600/750V)	X
	36 PL36 (change of traction system warning – DC 600/750V)	X
	27-255	Spare

Table 4 DMI_M_TRACKCOND

DMI_IDENTIFIER_REQUEST

The EVC sends this packet to request version information about the DMI, and expects to receive the DMI_IDENTIFIER packet containing the requested information. The DMI_IDENTIFIER packet is detailed later in this specification.
At the moment the DMI does not use or perform any check on the content of this packet. It only returns the DMI_ VERSION packet when it receives this one.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	7	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_ERTMS_VERSION				8			BCD		MSB 4 bits gives SRS major version number LSB 4 bits gives SRS minor version number
4	DMI_M_IMPL_VERSION				8			BCD	Fix value specific to the onboard supplier	MSB 4 bits give implementation major version number. LSB 4 bits give implementation minor version number. It should correspond to the version of the communication protocol.
3	DMI_M_EVC_VERSION_MAJOR_CODE	EVC Major Version Code	EVC Major Version Code. The complete version number of the EVC is *major.minor.update*.		8	0	254		255	unknown
4	DMI_M_EVC_VERSION_MINOR_CODE	EVC Minor Version	EVC Minor Version Code. The complete version number of the		8	0	254		255	unknown

		Code	EVC is *major.minor.update*.							
5	DMI_M_EVC_VERSION_UPDATE_CODE	EVC Update Version Code	EVC Update Version Code. The complete version number of the EVC is *major.minor.update*.		8	0	254		255	unknown
6	DMI_L_NAME	Length of EVC Name	Length of EVC name.		8	0	255			
7	DMI_X_NAME (* DMI_L_NAME)	EVC Name	Name used to identify the EVC.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
8	DMI_L_EXTRA	Length of extra information string	Length of extra information string.		8	0	255			
9	DMI_X_EXTRA (* DMI_L_EXTRA)	Extra information string	String that can contain any extra information about the EVC.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		

5.13 DMI_ICONS

This packet allows requesting the display of one or several icon(s) in any area.

Only one icon is allowed to request an acknowledgement (only one icon with DMI_M_ICON_CONTROL_FLAG =3)

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	8	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_N_ITER	Number of TIU events	Number of icons.		8	0	255			
4	…DMI_M_ICON_CONTROL_FLAG	Control flag for the icon display.	Control flag for the icon(s) display.		2	0	3	enum	00: show icon in area 01: clear area 02: show icon flashing in area 03: show icon with yellow flashing frame in area	
5	…DMI_M_ICON_FLASHING_FREQ	Flashing frequencies when flashing requested.	Flashing frequencies.		8	0	25,5 Hz	0.1 Hz	0	0 means no flashing.
6	…DMI_NID_ICON_GROUP	Icon group Identifier	Id of the icon group as defined in ERA. DMI_NID_ICON_GROUP + DMI_NID_ICON_RANK is a diverse representation of DMI_X_ICON_REF		4	0	7	Enum	0 level symbols 1 mode symbols 2 status symbols 3 Orders an announcements symbols 4 Planning information symbols 5 Navigation symbols 6 Settings symbols	

									7 Driver requests symbol

7	…DMI_NID_ICON_RANK	Icon rank within icon group	Rank of the icon within the icon group the icon is defined in in ERA.
			DMI_NID_ICON_GROUP + DMI_NID_ICON_RANK is a diverse representation of DMI_X_ICON_REF
					6	0	63	enum	Examples
									If DMI_NID_ICON_GROUP=0, then
									1 is LE01 symbol
									2 is LE02 symbol
									etc.
									
									If DMI_NID_ICON_GROUP=1, then
									1 is MO01 symbol
									2 is MO02 symbol

8	…DMI_L_ICON_REF	Length of ERA Icon Reference(s)	Length of the ERA Reference of the icon to display.		8	0	255		
9	……DMI_X_ICON_REF (*DMI_L_ICON_REF)	ERA Icon Reference	Plain text representing the ERA reference of the icon to display.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)	
10	…DMI_NID_AREA_GROUP	Area group Identifier	Number identifying one of the main areas as defined in ERA. Increasing numbers correspond to increasing alphabetical letters. DMI_NID_AREA_GROUP + DMI_NID_AREA_RANK is a diverse representation of DMI_X_AREA_REF		4	0	7	Enum	0 area A 1 area B 2 area C 3 area D 4 area E 5 area F 6 area G 7 area H (softkeys)

11	…DMI_NID_AREA_RANK	Coded ERA Area Id	Rank of the area within the area group the area is part of as defined by ERA.
			DMI_NID_AREA_GROUP + DMI_NID_AREA_RANK is a diverse representation of DMI_X_AREA_REF
					6	0	63	enum	Examples
									If DMI_NID_AREA_GROUP=0, then
									1 is A1 area
									2 is A2 area
									etc.
									
									If DMI_NID_AREAGROUP=1, then
									1 is B1 area
									2 is B2 area

									etc.	
12	…DMI_L_AREA_NAME	Length of ERA Area Name	Length of the ERA Area Name where to display the icon.		8	0	255			
13	……DMI_X_AREA_NAME (*DMI_L_AREA_NAME)	ERA Area Name	Plain text representing the ERA Area Name where to display the icon.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
14	DMI_NID_ICON_IDENTIFIER	Identifier of icon to be acknowledged	Identifier of icon to be acknowledged		8	0	254		255	No icon to be acknowledged.

DMI_SYSTEM_VERSION

This packet informs DMI about the operated system version used by EVC. These data are sent by EVC on driver request when the system version menu is opened.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	9	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.		16	24	65535	1 bit		
3	DMI_M_VERSION	Operated system version	Code for EVC operated system version represented by a minor and a major code.		7	0	127		001 0000 001 0001 010 0000 Others	Version 1.0 Version 1.1 Version 2.0 Unused

5.14 DMI_DISPLAY_CONTROL

This packet is sent by the EVC when the cabin is opened or closed, what shall respectively trigger activation of deactivation of the ETCS display on the DMI.

A deactivated DMI shall neither show any information to the driver nor capture any data from the driver, i.e.:

- The screen shall remain in blank.
- No input shall be allowed. All input performed by the driver on the input medium (e.g. Touchscreen or/and soft keys) shall be ignored and shall not affect the input performed when the DMI is activated again.
- All packets received from the EVC shall be ignored and discarded whilst the DMI is not active, with exception of the DMI_VERSION_REQUEST packet

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	10	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_ACTIVE_DMI	Active cab(s)	Status of cab activation.		1	0	1	enum.	0 1	cab not active (off and reset) cab active (on)

5.15 DMI_EVC_VBC_DATA

This packet is sent by the EVC to provide the list of Virtual Balise Covers stored on-board.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	13	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_N_ITER		Number of VBC		8	0	255			
4	DMI_NID_VBC_CODE		Code for Virtual Balise Cover		24	0	16777216			

5.16 DMI_USER_FUNCTION

EVC can transmit this packet to the DMI in order to perform user specific function.
Format and content of "other data" will depend on the value of "DMI_NID_XUSER".

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	14	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NID_XUSER		Identifier of the user functionality		8	0	255			
4	Other data		Depends on DMI_NID_XUSER							

5.17 DMI_STATUS

The DMI transmits cyclically its activity status to the EVC through this packet. Error code range from 128 to 192 is reserved for reporting hardware failures.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	128	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Clock value of the last received dynamic data.		32	0	4294967.295 s	1 ms		
4	DMI_Q_STATUS (*)	Activity status of the MMI	Status code of the DMI at time T_CLOCK.	FFFIS	8	0	3	enum.		see table below
5	DMI_N_ALIVE (*)	Alive counter	DMI alive counter. The DMI shall increment this value every second. The EVC shall consider the DMI as not working any longer and trigger a safe reaction as soon as this value is detected stuck		8	0	255	enum.		See note

List of error codes is extensible depending on specific functional and safety requirements.

The DMI shall report about the activity and aliveness of all its safety relevant parts (additional safety self-test functions) through the alive counter sent along the operating status. The alive counter shall only be incremented when all system safety functions are proven to be running safe, i.e. when all additional safety self-test functions are detected to be working and are not stuck between each status reporting cycle. All threads and functions shall be checked somehow for being functional. The EVC shall trigger a safe reaction when the alive counter is stuck. Nevertheless all detectable errors shall be reported via the DMI-status.

Meaning of status code value range is given by the following table:

Range (decimal)	Range (hexadecimal)	Meaning
0-0	0x00-0x00	No error.
1-127	0x01-0x7F	EVC interface faults. All faults related to the communication with EVC and the packet content.
128-191	0x80-0xBF	Hardware failures. All hardware failures which can be detected with the Environment Controller.
192-219	0xC0-0xDB	Maintenance local fault. All faults encountered while operating with the System Maintenance Interface.
220-255	0xDC-0xFF	DMI application unrecoverable faults. All faults (e.g. unexpected software state) that cannot be classified in any other value range.
230-240	0xE6-0xF0	DMI application unrecoverable faults related to safety. All unsafe system states detected by the extra safety monitoring functions.

List of status codes:

DMI_Q_STATUS (*)	0	Running state
	1	Starting state
	2	Failure state (unable to run)
	3	Running not active state (cabin off)
	92	Train Speed Overflow (v-train > v-max scale)
	100	Invalid track condition
	101	Invalid predefined text message (Q_TEXT)
	102	Invalid text message (Q_TEXTCONFIRM)
	128	HW warning temp. reached
	129	TFT OFF temp. reached
	130	Device OFF temp. reached
	131	Over temperature
	132	Backlight on/off
	133	FAN blocked
	134	Power supply key switch off
	136	Watchdog not running
	4 – 127	spare
	134-255	spare

Table 5 DMI_Q_STATUS

5.18 DMI_DRIVER_REQUEST

The packet contains the action requested by the driver when activating a menu button on the DMI.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128]: Source is EVC [128, 255]: Source is DMI.	129	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Clock value of the last received dynamic data.		32	0	4294967.295 s	1 ms		
4	DMI_M_REQUEST	Driver request	This variable contains a request by the driver.		8	0	255	Resolution: enum.	see table below	
5	DMI_NID_STM	NID STM	Number identifier of STM. ONLY TRANSMITTED if DMI_M_REQUEST = “NTC data entry”.	8	0	255	enum.	See §6.2.2	See §6.2.2	

DMI_M_REQUEST	0-9	Spare
	10 Start of mission	X
	11 Shunting entry	X
	12 Shunting exit	X
	13 Non-leading	X
	14 Non-leading exit	Spare
	15 Maintain shunting	X
	16 Level entry request	X
	17-19	Spare
	20 Override EOA	X

	21 Override route unsuitability	Spare
	22-28	Spare
	29 Request for radio network entry	X
	30 Request for train data	X
	31 Request for Adhesion factor data	X
	32 Request for SR data	X
	33 Request for system version	X
	34 Request for switching train data entry	X
	35 Request for train data view	X
	36 Request to show geographical position	X
	37 Language changed	X
	38 Train data entry aborted	X
	39 Train running number entry aborted	X
	40 SR data entry aborted	X
	41 Radio network entry aborted	X
	42 Request to hide geographical information	X
	43 Request to show supervision data	X
	44 Request to hide supervision data	X
	45 Scroll text up	X
	46 Scroll text down	X
	47 Request to contact last known RBC	X
	48 Request to use short number	X
	49 Request isolation	X
	50 -63	Spare
	64 Track Ahead Free is validated	X
	65 The Train Integrity request	X
	66 Set VBC request	X
	67 Remove VBC request	X
	68 Show tunnel stopping information	X
	69 Hide tunnel stopping information	X
	70 NTC data entry request	X
	71 End of NTC data entry	X
	72-255	Spare

Table 6 DMI_M_REQUEST

5.19 DMI_TEXT_MESSAGE_ACK

This packet contains the acknowledgement of the driver for the text message identified by DMI_NID_MESSAGE_IDENTIFIER. This acknowledgement is either positive, i.e. the message is accepted by the driver, or negative, in which case the message is rejected.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	130	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Mirrored EVC clock timestamp	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_NID_MESSAGE_IDENTIFIER	Identifier of the text message	Counter used to identify text messages sent for further acknowledgment or removal.		8	0	255			
5	DMI_M_ACKNOWLEDGE	Result of acknowledgement	Result of driver's acknowledgement		8	0	1	enum.	0 1 2 - 15	acknowledged not acknowledged spare

5.20 DMI_TRAIN_DATA_ACK

This packet contains the acknowledgement of the driver for the validated train data transmitted by the EVC in response to the original train data entered by the driver. This acknowledgement can either be positive, i.e. the validated train data are accepted, or negative, in which case the data are rejected.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	131	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Mirrored EVC clock timestamp	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_M_ACKNOWLEDGE	Result of acknowledgement	Result of driver's acknowledgement		8	0	1	enum.	0 1 2 - 15	acknowledged not acknowledged spare

5.21 DMI_IDENTIFIER

The DMI sends this packet in response to the DMI_IDENTIFIER_REQUEST sent by the EVC.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	132	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_IDENTIFIER		Identifier of the DMI		1	0	1	enum		0 DMI 1 1 DMI 2
4	DMI_NID_CABIN		Identifier of the cabin		1	0	1	enum		0 Cabin A 1 Cabin B
5	DMI_L_NAME	Length of DMI Name	Length of DMI name		8	0	64			
6	DMI_X_NAME (* DMI_L_NAME)	DMI Name	Name used to identify the DMI.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
7	DMI_M_VERSION_MAJOR_CODE	DMI Major Version Code	DMI Major Version Code. The complete version number of the DMI is *major.minor.update*.		8	0	254		255	unknown
8	DMI_M_VERSION_MINOR_CODE	DMI Minor Version Code	DMI Minor Version Code. The complete version		8	0	254		255	unknown

			number of the DMI is *major.minor.update*.							
9	DMI_M_VERSION_UPDATE_CODE	DMI Update Version Code	DMI Update Version Code. The complete version number of the DMI is *major.minor.update*.		8	0	254		255	unknown
10	DMI_M_CONFIG_VERSION_MAJOR_CODE	DMI Configuration Major Version Code	DMI Configuration Major Version Code. This code is related to the configuration identifier of the DMI.		8	0	254		255	unknown
11	DMI_M_CONFIG_VERSION_MINOR_CODE	DMI Configuration Minor Version Code	DMI Configuration Minor Version Code. This code is related to the configuration identifier of the DMI.		8	0	254		255	unknown
12	DMI_M_CONFIG_VERSION_UPDATE_CODE	DMI Configuration Update Version Code	DMI Configuration Update Version Code. This code is related to the configuration identifier of the DMI.		8	0	254		255	unknown
13	DMI_M_PROTO_VERSION_MAJOR_CODE	Communication Protocol Major Version Code	Communication Protocol Major Version Code. The complete version number of the protocol is *major.minor*. It corresponds to the major version of this specification.		8	0	254		255	unknown
14	DMI_M_PROTO_VERSION_MINOR_CODE	Communication Protocol Minor Version Code	Communication Protocol Minor Version Code. The complete version number of the protocol is *major.minor*. It corresponds to the minor version of this specification.		8	0	254		255	unknown
15	DMI_BOARD_NUMBER	Hardware board number	Hardware board number		16	0	65535			
16	DMI_L_FIRMWARE_VERSION	Length of textual firmware number	Length of the string containing the firmware number		8	0	255			
17	DMI_X_FIRMWARE_VERSION (*DMI_L_FIRMWARE_VERSION)	Textual firmware number	String containing the firmware number		8	32	255	ASCII (8 Bits), ISO8859-		

								1 (Latin Alphabet #1)		
18	DMI_L_DATE_TEXT	Length of textual date	Length of the string containing the textual build date of the DMI		8	0	255			
19	DMI_X_DATE_TEXT (* DMI_L_DATE_TEXT)	Textual date.	String that contains the textual build date of the DMI		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		The extra string shall be of the form: "<*short_month*> <*month_day*> <*year*> <*hour*>:<*min*>:<*sec*>" e.g: Jul 13 2005 09:43:03
20	DMI_L_EXTRA	Length of extra information string	Length of extra information string		8	0	255			
21	DMI_X_EXTRA (* DMI_L_EXTRA)	Extra information string	String that can contain additional information about the DMI		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		

5.22 DMI_ICON_ACK

This packet is sent when an icon is acknowledged.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	133	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Clock time	Clock value of the last received dynamic data.		32	0	4294967.295 s	1 ms		
4	DMI_L_ICON_REF	Length of ERA Icon Reference	Length of the ERA Reference of the acknowledged icon		8	0	255			
5	DMI_X_ICON_REF (*DMI_L_ICON_REF)	ERA Icon Reference	ERA Reference of the acknowledged icon.		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
6	DMI_L_AREA_NAME	Length of ERA Area Name	Length of the ERA Area Name where to icon is displayed.		8	0	255			
7	DMI_X_AREA_NAME (*DMI_L_ICON_REF)	ERA Area Name	ERA Area Name where to display the icon..		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin		

								Alphabet #1)		
8	DMI_NID_ICON_IDENTIFIER	Acknowledged icon identifier	Acknowledged icon identifier		8	0	254		255	No icon to be acknowledged.

5.23 DMI_SOUND_STATUS

This packet is sent by the DMI to indicate a change of the audible information on DMI (sent when the sound is played and when it is stopped).

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	134	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_Q_SOUND		Status of the audible information played to the driver. A bit set to '1' means that the corresponding sound is generated.		3	0	7		xx1 x1x 1xx	Sound Sinfo Sound S1 Sound S2

5.24 DMI_SET_VBC_DATA

This packet is sent by the DMI when a Virtual Balise Cover is set.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	135	Valid id for this packet
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
4	DMI_NID_VBC_CODE		Code for Virtual Balise Cover		24	0	16777216			

5.25 DMI_RM_VBC_DATA

This packet is sent by the DMI when a Virtual Balise Cover is removed.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	136	Valid id for this packet
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
4	DMI_NID_VBC_CODE		Code for Virtual Balise Cover		24	0	16777216			

5.26 DMI_DRIVER_IDENTIFIER

It is a two-way packet. EVC supplies the default driver ID using this packet. Reversely, this packet supplies the EVC with the driver identifier entered by the driver.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	192	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NID_DRIVER	Driver identity	Entered driver identity		80	0	0	10 ASCII characters		

5.27 DMI_DRIVER_IDENTIFIER

It is a two-way packet. EVC supplies the default train running number using this packet. This packet supplies the EVC with the train running number entered by the driver.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	193	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NID_OPERATIONAL	Train running number	Entered operational train running number	SRS 2.2.2	32	0	9999 9999	4 bits BCD	0-9 15	digit no digit (if shorter than 8 digits)

5.28 DMI SR_DATA

It is a two-way packet. This packet supplies the DMI with the default Staff Responsible data stored onboard (maximum speed and distance). These data are sent by EVC on driver request when the SR data entry page is opened. This packet is sent by the DMI when new SR data have been entered.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	194	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_V_STFF	Max Staff Responsible speed	Speed value to override the default max Staff Responsible speed in the system		16	0	600	1 km/h	other	unused
4	DMI_D_STFF	Max distance in Staff Responsible	Distance on which the train is allowed to run in Staff Responsible mode (overrides the preset value in the system).		32	0	42949672.95 m	0.01m		

5.29 DMI_TRAIN_DATA

It is a two-way packet. This packet contains the default train data sent by the EVC and is also returned by the DMI when the train data has been entered by the driver.

The content of the packet, i.e. the individual train data information, are configurable and therefore static description of this packet are provided as example only. Here is a generic description of this packet format:

- The train data packet always starts with the DMI_NID_PACKET et DMI_L_PACKET as any other packet exchanged between EVC and DMI.
- This packet header is followed by each train data field defined within the train data configuration selected at DMI start-up.
- The fields appear in the sequence order as they are defined within the configuration.
- The fields are coded on the bit length specified within the configuration.
- The fields are coded using the resolution specified within the configuration. E.g. if the resolution is 5, the value coded within the packet will be the raw value divided by 5.
- The minimum and maximum values that can be coded for a field are also determined within the configuration.
- Fields for which an enumeration of values is defined within the configuration can't be set to other values.

The following packet description is the default implementation provided for the train data as described in document /1/ and /8/ for ERTMS.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	195	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NC_TRAIN	International category to which the train belongs	Train category used for the static speed profile calculation. Thanks to NC_TRAIN, the train knows the SSP it must obey. by	SRS 2.2.2	16	See below				

			receiving a list of static speed profile, thanks to NC_TRAIN, the train can select the SSP it must obey. Each bit represents one category. A train can belong to various categories.							
4	DMI_L_TRAIN	Train Length	Length of the train.	SRS 2.2.2	16	0	4094	1m	4095	unknown
5	DMI_M_BRAKEPERCT	Braking percentage	Braking percentage		8	30 (30%)	250 (250%)	1%	Any other	unknown
6	DMI_V_MAXTRAIN	Maximum permitted Speed for the Train			16	0	600 (600 km/h)	1 km/h	other	unused
7	DMI_M_AXLELOAD	Axle Load	Maximum axle load of the train.	SRS 2.2.2	8	See below				
8	DMI_M_AIRTIGHT	Airtight system presence	Indicates whether the train is fitted with an airtight system or not.	SRS 2.2.2	8	0	1	enum.	0 1 2 3 – 15	not fitted fitted unknown spare
9	DMI_M_LOADINGGAUGE	Loading gauge	Define the loading gauge profile of a train		8	0	255		0 1 2 3 4 5-255	Out of GC G1 GA GB GC spare

DMI_NC_TRAIN	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
	0	17	enum	0 1 2 3 4 5 6 7 8	PASS 1 PASS 2 PASS 3 TILT 1 TILT 2 TILT 3 TILT 4 TILT 5 TILT 6

				9 10 11 12 13 14 15 16 17 18-65535	TILT 7 FP 1 FP 2 FP 3 FP 4 FG 1 FG 2 FG 3 FG 4 Spare

DMI_M_AXLELOAD	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
	0	11	enum	0 1 2 3 4 5 6 7 8 9 10 11 12-255	A B1 B2 C2 C3 C4 D2 D3 D4 D4XL E4 E5 Spare

5.30 DMI_ADHESION_FACTOR_DATA

It is a two-way packet. EVC supplies the default adhesion factor using this packet. Reversely, this packet supplies the EVC with the new adhesion factor entered by the driver for replacing the value pre-set in the system.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	196	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_ADHESIONFACTOR	Entered adhesion factor	Entered adhesion percentage for track due to environmental influences.		8	0	100	enum.	70 100 other	slippery rail (70% grip) not slippery rail (100% grip) unused

5.31 DMI_LEVEL_DATA

Two-way packet, supplies the DMI with the default ETCS level. Reversely, supplies the EVC with the data related to the ETCS level selected by the driver.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	197	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_M_LEVEL	ETCS level		SRS 2.2.2	3	0	4	enum.	0 1 2 3 4 5-7	Level 0 Level STM specified by NID_STM Level 1 Level 2 Level 3 Spare
4	DMI_NID_STM	NID STM	Number identifier of STM. ONLY TRANSMITTED if Level STM specified in DMI_M_LEVEL.	SRS 2.2.2	8	0	255	enum.	See §6.2.2	See §6.2.2

5.32 DMI_EVC_LEVEL_DATA

EVC supplies the DMI with the list of active level.

These data are sent by EVC on driver request when the level entry page is opened.

	VARIABLE	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128]: Source is EVC [128, 255]: Source is DMI.	11	Valid id for this packet.
2	DMI_L_PACKET	Length of packet in bits, inclusive packet header.		16	0	65535			
3	DMI_N_ITER	Number of iteration of active levels		8	0	31			
4	DMI_M_LEVEL		If DMI_N_ITER > 0	3	0	4		0 1 2 3 4 5-7	Level 0 Level STM specified by NID_STM Level 1 Level 2 Level 3 Spare
5	DMI_NID_STM	Number identifier of STM.	SRS 2.2.2 If DMI_M_LEVEL = 1	8	0	255	enum.		

5.33 DMI_RBC_DATA

It is a two-way packet. This packet supplies the EVC with the data related to entered RBC data and the radio network ID when the level 2/3 has been selected. Reversely, the EVC sends the default RBC data with this packet.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	199	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NID_RBC	RBC ETCS identity number	Entered RBC ETCS identity number (including NID_C and NID_BC defined according to ERTMS/ETCS – Class 1, SRS, Subset-026-7)	SRS 2.0.0	24	0	16777214	binary value	16777215	not unknown
4	DMI_NID_RADIO	RBC phone no	Entered Radio subscriber number (phone number)	SRS 2.2.2	64	0	9999 9999 9999 9999	4 bits BCD	0-9 15	digit indicates no digit (if shorter than 16 digits)

5.34 DMI_RADIO_NET_DATA

DMI supplies the EVC with the entered radio network.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	137	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_NETWORK_ID		Radio network identity	SRS 2.3.0	24	0	999999	4 bits BCD		For each digit: Values A-E Not used Use value F for digit to indicate no digit (if number shorter than 6 digits)

5.35 DMI_SETVBC_DATA_ACK

This packet contains the acknowledgement of the driver for the VBC data transmitted by the EVC in response to the original VBC data entered by the driver. This acknowledgement can either be positive, i.e. the VBC data are accepted, or negative, in which case the data are rejected.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	138	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Mirrored EVC clock timestamp	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_M_ACKNOWLEDGE	Result of acknowledgement	Result of driver's acknowledgement		8	0	1	enum.	0 1 2 - 15	acknowledged not acknowledged spare

5.36 DMI_RMVBC_DATA_ACK

This packet contains the acknowledgement of the driver for the VBC data transmitted by the EVC in response to the original VBC data entered by the driver. This acknowledgement can either be positive, i.e. the VBC data are accepted, or negative, in which case the data are rejected.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	139	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_T_CLOCK	Mirrored EVC clock timestamp	Seconds elapsed in day since 00:00		32	0	4294967.295 s	1 ms		
4	DMI_M_ACKNOWLEDGE	Result of acknowledgement	Result of driver's acknowledgement		8	0	1	enum.	0 1 2 - 15	acknowledged not acknowledged spare

5.37 DMI_EVC_RADIO_NET_DATA

EVC supplies the DMI with the list of available radio networks.

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	12	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_N_ITER		Number of iteration of radio networks		8	0	255			
4	DMI_NETWORK_ID	Network	Radio network identity	SRS 2.3.0	24	0	999999	4 bits BCD		For each digit: Values A-E Not used Use value F for digit to indicate no digit (if number shorter than 6 digits)
5	DMI_L_LABEL		Length of network label string		8	0	255	1 label string element		
6	DMI_X_LABEL (*DMI_L_LABEL)		One character of the label of radio network		8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		
7	DMI_Q_DEFAULT		Indicator of default radio network		1	0	1		0 1	Not default network Default radio network

5.38 DMI_NTC_DATA_ENTRY

EVC supplies the DMI with description for the NTC data entry window (list of STM requesting data)

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	18	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
3	DMI_Q_ENABLED		Indicator of button enabled		1	0	1		0 1	All buttons disabled (except “end data entry”) All buttons enabled
4	DMI_N_ITER		Number of STM requesting data		8	0	255			
5	DMI_NID_STM	NID STM	Identifier of STM (NTC)	SRS 2.2.2	8	0	255	enum.	See §6.2.2	See §6.2.2

5.39 DMI_NTC_INPUT

EVC supplies the DMI with input information related to a NTC (STM)

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	17	Valid id for this packet.
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
4	DMI_NID_STM	NID STM	Identifier of STM (NTC)	SRS 2.2.2	8	0	255	enum.	See §6.2.2	See §6.2.2
5	NTC sub packet									See § ???

5.40 DMI_NTC_OUTPUT

DMI supplies the EVC with output information related to a NTC (STM)

	VARIABLE	Name	Description	Remark	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	DMI_NID_PACKET		Header for each packet, allowing the receiving equipment to identify the data that follows.		8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	140	Valid id for this packet
2	DMI_L_PACKET		Length of packet in bits, inclusive packet header.	SRS 2.2.2	16	24	65535	1 bit		
4	DMI_NID_STM	NID STM	Identifier of STM (NTC)	SRS 2.2.2	8	0	255	enum.	See §6.2.2	See §6.2.2
5	NTC sub packet									See § ???

5.40.1 NTC sub packet

The NTC sub-packets correspond to a subset of packets/variables defined in /10/ with prefix “NTC_”.

NID	Packet Name	From	Short description
32	NTC_BUTTON_REQUEST	EVC	Request to display a NTC button
34	NTC_BUTTON_EVENT_REPORT	DMI	Report event on a NTC button
35	NTC_INDICATOR_REQUEST	EVC	Request to display a NTC indicator
38	NTC_TEXT_MESSAGE	EVC	Request to display a NTC text message
39	NTC_DEL_TEXT_MESSAGE	EVC	Request to remove a NTC text message
40	NTC_ACK_REPLY	DMI	Report an acknowledgement of an NTC text message
43	NTC_SUPERVISION_DATA	EVC	Speed and distance supervision information
46	NTC_SOUND_CMD	EVC	Request to play a NTC sound
179	NTC_DATA_ENTRY_RQST	EVC	Request entry of specific NTC data
180	NTC_DATA_VALUES	DMI	Provide the specific NTC data values
183	NTC_DATA_VIEW	EVC	Provide the specific NTC data view

A) NTC_BUTTON_REQUEST

EVC supplies the DMI with a request to display a NTC button.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	32	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_N_ITER	Number of iteration	5	0	24			
4	NTC_NID_BUTTON(k)	Button identifier	8	0	255			
5	NTC_NID_BUTPOS(k)	Button position identifier	5	1	24		1 2	Softkey Touchscreen F8 F8

							3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18-24	F9 F10 H2 H3 H4 Spare Spare Spare Spare Spare Spare Spare Spare Spare Spare Spare	F9 C2 C3 C4 C5 C6 G1 G2 G3 G4 G5 G6 G7 G8 G9 G10 spare
6	NTC_NID_ICON(k)	Icon identifier	8	1	255		0	No icon reference	
7	NTC_M_BUT_ATTRIB(k)	Attributes of the button	10				0xxxxxxxxx 10xxxxxxxx 11xxxxxxxx 1x00xxxxxx 1x01xxxxxx 1x10xxxxxx 1x11xxxxxx 1xxx000xxx 1xxx001xxx 1xxx010xxx 1xxx011xxx 1xxx100xxx 1xxx101xxx 1xxx110xxx 1xxx111xxx 1xxxxxx000 1xxxxxx001 1xxxxxx010 1xxxxxx011 1xxxxxx100 1xxxxxx101	Not displayed Button Normal flashing Button Counterphase flashing Button No flashing Button Slow flashing Button Fast flashing Reserved Dark blue button background White button background Red button background Blue button background Green button background Yellow button background Light red button background Light green button background Black text label White text label Red text label Blue text label Green text label Yellow text label	

							1xxxxxx110 1xxxxxx111	Light red text label Light green text label
8	NTC_L_CAPTION(k)	Length of X_CAPTION in bytes	6	0	24			
9	NTC_X_CAPTION(k,j)	Caption text	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		

B) NTC_BUTTON_EVENT_REPORT

DMI supplies the EVC with a report about event on a NTC button.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	34	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_N_ITER	Number of iteration	5	0	24			
4	NTC_NID_BUTTON(k)	Button identifier	8	0	255			
5	NTC_Q_BUTTON(k)	Button event	1				0 1	Push event Release event
6	NTC_T_BUTTON_EVENT(k)	Event timestamp	32	0	4294967295	1ms		

C) NTC_INDICATOR_REQUEST

EVC supplies the DMI with a request to display a NTC indicator.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	35	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_N_ITER	Number of iteration	5	0	24			
4	NTC_NID_INDICATOR(k)	Indicator identifier	8	0	255			
5	NTC_NID_INDPOS(k)	Indicator position identifier	5	1	24		1 2 3 4 5 6 7 8 9 10 11 12 13 14 14 16 17 18 19 20-24	B3 B4 B5 H1 (Reserved in touch screen technology) C2 C3 C4 C5 C6 G1 G2 G3 G4 G5 G6 G7 G8 G9 G10 spare

6	NTC_NID_ICON(k)	Icon identifier	8	1	255		0	No icon reference
7	NTC_M_IND_ATTRIB(k)	Attributes of the indicator	10				0xxxxxxxxx 10xxxxxxxx 11xxxxxxxx 1x00xxxxxx 1x01xxxxxx 1x10xxxxxx 1x11xxxxxx 1xxx000xxx 1xxx001xxx 1xxx010xxx 1xxx011xxx 1xxx100xxx 1xxx101xxx 1xxx110xxx 1xxx111xxx 1xxxxxx000 1xxxxxx001 1xxxxxx010 1xxxxxx011 1xxxxxx100 1xxxxxx101 1xxxxxx110 1xxxxxx111	Not displayed Indicator Normal flashing Indicator Counterphase flashing Indicator No flashing Indicator Slow flashing Indicator Fast flashing Reserved Dark blue indicator background White indicator background Red indicator background Blue indicator background Green indicator background Yellow indicator background Light red indicator background Light green indicator background Black text label White text label Red text label Blue text label Green text label Yellow text label Light red text label Light green text label
8	NTC_L_CAPTION(k)	Length of X_CAPTION in bytes	6	0	24			
9	NTC_X_CAPTION(k,j)	Caption text	8	32	255	ASCII (8 Bits), ISO8859-1 (Latin Alphabet #1)		

D) NTC_TEXT_MESSAGE

EVC supplies the DMI with a request to display a NTC text message.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values

1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	38	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_NID_XMESSAGE	Text message identifier	8	0	255			
4	NTC_M_XATTRIBUTE	Attributes of text	10				0xxxxxxxxx 1xxxxxxxxx x0xxxxxxx x1xxxxxxx xx00xxxxxx xx01xxxxxx xx10xxxxxx xx11xxxxxx xxxx000xxx xxxx001xxx xxxx010xxx xxxx011xxx xxxx100xxx xxxx101xxx xxxx110xxx xxxx111xxx xxxxxxx000 xxxxxxx001 xxxxxxx010 xxxxxxx011 xxxxxxx100 xxxxxxx101 xxxxxxx110 xxxxxxx111	Text message shall be displayed in group 1 Text message shall be displayed in group 2 Normal flashing. Counterphase No flashing Slow flashing Fast flashing Reserved Dark blue text background White text background Red text background Blue text background Green text background Yellow text background Light red text background Light green text background Black text White text Red text Blue text Green text Yellow text Light red text Light green text
5	NTC_Q_ACK	Acknowledgement qualifier	1	0	1		0 1	No acknowledgement required Acknowledgement required
6	NTC_L_TEXT	Text length in bytes	8	0	80			
7	NTC_X_TEXT(k)	Text character	8	32	255	ASCII (8 Bits), UTF-8		

E) NTC_DEL_TEXT_MESSAGE

EVC supplies the DMI with a request to delete a NTC text message.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	39	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_NID_XMESSAGE	Text message identifier	8	0	255			

F) NTC_ACK_REPLY

DMI supplies EVC with information about acknowledgement of a NTC text message.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	40	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_NID_XMESSAGE	Text message identifier	8	0	255			

G) NTC_SUPERVISION_DATA

EVC supplies the DMI with speed and distance supervision information for related NTC.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	43	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_Q_SCALE		2					
4	NTC_V_PERMIT	Permitted speed	10					
5	NTC_V_TARGET	Target speed	7					
5	NTC_V_RELEASE	Release speed	10					
6	NTC_V_INTERV	Intervention speed	10					
7	NTC_D_TARGET	Target distance	15					
8	NTC_M_COLOUR_SP	Color of speed pointer	3					
9	NTC_M_COLOUR_PS	Color of permitted speed	3					
10	NTC_Q_DISPLAY_PS	Display of permitted speed	2					
11	NTC_M_COLOUR_TS	Color of target speed	3					
12	NTC_Q_DISPLAY_TS	Display of target speed	2					
13	NTC_M_COLOUR_RS	Color of release speed	3					
14	NTC_Q_DISPLAY_RS	Display of release speed	2					
15	NTC_M_COLOUR_IS	Color of intervention speed	3					
16	NTC_Q_DISPLAY_IS	Display of intervention speed	2					
17	NTC_Q_DISPLAY_TD	Display of target distance	2					

H) NTC_SOUND_CMD

EVC supplies the DMI with sound information for related NTC.

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	46	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_N_ITER	Number of sound to be generated	5	0	2			
4	NTC_NID_SOUND(k)	Sound identifier	8	1	255		0	No sound identifier
5	NTC_Q_SOUND(k)	Continuous / not continuous / stop	2				0 1 2	Stop sound generation One shot play (Sound is played once) Continuous play
6	NTC_N_ITER(k)	Number of sound segment	5	0	32			
7	NTC_M_FREQ(k,j)	Frequency of a segment	8	4 (128Hz)	255 (8160Hz)	32Hz	0	Silence
8	NTC_T_SOUND(k,j)	Duration of segment	8	1 (100ms)	100 (10s)	100ms		

I) NTC_DATA_ENTRY_RQST

EVC supplies the DMI with specific data entry request

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	179	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_Q_FOLLOWING		1					
4	NTC_N_ITER(k)		5	0	15			
5	NTC_NID_DATA(j)		8					
6	NTC_L_CAPTION(j)		6	0	40			
7	NTC_X_CAPTION(j,q)		8					
8	NTC_L_VALUE(j)		5	0	20			
9	NTC_X_VALUE(j,i)		8					
10	NTC_N_ITER(j)		5					
11	NTC_L_VALUE(j,i)		5	0	20			
12	NTC_X_VALUE(j,I,k)		8					

J) NTC_DATA_VALUES

DMI supplies the EVC with specific data values

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	180	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_N_ITER(k)		5	0	15			
4	NTC_NID_DATA(j)		8					
5	NTC_L_VALUE(j)		5	0	20			
6	NTC_X_VALUE(j,i)		8					

K) NTC_DATA_VIEW

EVC supplies the DMI with specific data view

	VARIABLE	Description	Length (bits)	Min Value	Max Value	Resolution / Formula	Special / Reserved Values	Meaning of S/R Values
1	NTC_NID_PACKET	Header for each packet, allowing the receiving equipment to identify the data that follows.	8	0	255	enum. [0, 128[: Source is EVC [128, 255]: Source is DMI.	183	Valid id for this packet.
2	NTC_L_PACKET	Length of packet in bits, inclusive packet header	13			1 bit		
3	NTC_Q_FOLLOWING		1					
4	NTC_N_ITER(k)		5	0	32			
5	NTC_NID_DATA(j)		8					
6	NTC_L_CAPTION(j)		6	0	40			
7	NTC_X_CAPTION(j,q)		8					
8	NTC_L_VALUE(j)		5					
9	NTC_X_VALUE(j,i)		8					

6 DMI AUTONOMOUS FUNCTIONS

6.1 OVERVIEW

Except specified otherwise in the following section, all DMI functions described in documents /3/, /4/ shall be assumed as being carried out standalone by the DMI application.

6.2 DEFAULT WINDOWS

6.2.1 Brake information (area A)

The display of the distance to target digital and the distance to target bar is fully controlled by the DMI according to:

- the current mode
- the distance value transmitted in the DMI_DYNAMIC packet

Requests of the driver to display additional information in some modes (distance to target digital is managed standalone by the DMI).

6.2.2 Speedometer (area B)

A) B3/B4/B5 INDICATORS (TRACKSIDE ORDERS)

Display of icons in the B3/B4/B5 is fully controlled by EVC via DMI_ICONS packet.

B) B6 INDICATOR (DIGITAL RELEASE SPEED)

The DMI displays the appropriate release speed in B6 according to:

- The current ETCS mode (DMI_M_MODE)
- The current release speed transmitted in the DMI_DYNAMIC packet (DMI_V_RELEASE)
- The current toggle status depending on the driver request input to show this information

C) B7 INDICATOR (CURRENT ETCS MODE)

The DMI displays the appropriate icon in B7 according to the current mode (DMI_M_MODE) transmitted in the DMI_DYNAMIC packet.

D) CSG AND SPEED HOOKS

The display of CSG and hooks (position, colour, aspect) is controlled by the DMI according to:

- The curent ETCS mode (DMI_M_MODE).
- The current supervision status (DMI_M_SUPSTATUS).
- The current warning and intervention status according to the supervision curves (DMI_M_WARNING).
- The current speed values transmitted in the DMI_DYNAMIC packet; i.e. current train speed (DMI_V_TRAIN), current permitted speed (DMI_V_PERMITTED), current target speed (DMI_V_TARGET), current intervention speed (DMI_V_INTERVENTION) and current release speed (DMI_V_RELEASE).

Responsibilities of the DMI:

- For each applicable combination of ETCS mode, supervision status and status information according to ETCS, DMI bases on the transmitted speed values, i.e. permitted speed, target speed, release speed and intervention speed for displaying the various parts of the CSG according to the color scheme defined by ERA for the respective speed values.

- The DMI tests the current MODE for displaying the CSG or not.
- The DMI tests the current speed values for displaying the extended CSG part of the intervention speed only when current speed is greater than permitted speed.

Note: DMI won't use colors which are not supposed be visible in a given supervision status and status information (e.g. won't use yellow if in CSM but indication status is set) if mode, supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING) are not consistent between each other.

Responsibilities of the EVC:

- The EVC ensures consistency between ETCS mode, supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING)
- The EVC ensures that the values passed for permitted speed, release speed and target are consistent with regard to ETCS, i.e.:
  - If a target speed is defined, target speed shall be less than permitted speed.
  - If an intervention speed is defined, intervention speed shall be greater than permitted speed.
  - If a release speed is defined, target speed shall be 0.

With regard to the first point, it must be noted that the DMI is also to detect inconsistencies between ETCS mode, supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING) in the scope of the current speed display function.

The CSG display is undefined and cannot be relied upon by the driver in case the EVC fails to fulfill these requirements.

E) SPEED NEEDLE AND DIGITAL SPEED

The display of the speed needle/digital (position, color, aspect) is controlled by the DMI according to:

- The curent ETCS mode (DMI_M_MODE)
- The current supervision status (DMI_M_SUPSTATUS)
- The current warning and intervention status according to the supervision curves (DMI_M_WARNING)
- The current speed values transmitted in the DMI_DYNAMIC packet; i.e. current train speed (DMI_V_TRAIN), current permitted speed (DMI_V_PERMITTED), current target speed (DMI_V_TARGET), current intervention speed (DMI_V_INTERVENTION) and current release speed (DMI_V_RELEASE).
- The current dial scale.

Responsibilities of the DMI:

- For each applicable combination of ETCS mode, supervision status and status information according to ETCS, DMI tests the current speed values for displaying the suitable needle color, with one exception: When over speed status or warning status is active, the orange is always shown regardless of the speed values received by the DMI.
- The speed needle display is blocked at the dial speed limit when DMI_V_TRAIN is beyond the maximum speed of the dial scale.

Note: Even the EVC is responsible to ensure consistency between ETCS mode, supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING), the DMI is able to:

- Detect inconsistencies between ETCS mode and supervision status (DMI_M_SUPSTATUS) and triggers safe reaction [1]. In TR, PT, NL, SB, the DMI expects M_SUPSTATUS to be set to 'UNKNOWN' (15).
- Detect inconsistencies between supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING) and triggers safe reaction [1].

Responsibilities of the EVC:

- The EVC ensures consistency between ETCS mode, supervision status (DMI_M_SUPSTATUS) and status information (M_WARNING)

- The EVC shall not transmit combinations of status information (M_WARNING) and speed values which are not applicable according to ETCS. For these cases, the DMI display is undefined and the display cannot be relied upon by the driver.
- In TR, PT, NL, SB, M_SUPSTATUS shall be set to 'UNKNOWN' (15). Otherwise the DMI performs a safe reaction [1]

List of safe reactions on inconsistent EVC inputs (consistency check): [1] DMI shows a black needle and informs the EVC with a specific error code.

F) DISPLAY ON DRIVER REQUEST

All requests of the driver to display additional information in some modes (basic speed hooks, release speed) are managed standalone by the DMI.

6.2.3 Planning area (area D)

The DMI manages standalone all functions related to the planning area (including zooming, hiding) according to:

- the profiles sent in the DMI_TRACK_DESCRIPTION packet
- the current train position transmitted in the DMI_DYNAMIC packet
- the current ETCS mode transmitted in the DMI_DYNAMIC packet

6.2.4 Text message area (area E)

The arrangement of the text message as well as the scrolling in the list of received text message fully managed standalone by the DMI.

6.2.5 Other

6.2.6 A4 Adhesion Factor

The display of the A4 area is controlled by EVC.

6.2.7 E1 status of the communication sessions

The display of the E1 area is controlled by EVC.

6.2.8 Driver menu (area F)

The enabling/disabling of all menu buttons is fully controlled by DMI, except of the data view (F3) button which enabling status is controlled by EVC.

The activation of any of these button gives not rise to any data exchange with the EVC, with exception of the data view (F3) button for which a request to receive the train data stored onboard.

6.2.9 Geographical position (area G)

The display of the geographical position and its visibility status according to the driver input is fully controlled by the DMI according to the position information transmitted by the EVC in the DMI_DYNAMIC frame.

6.2.10 Current time (area G)

The display of the current time is updated in real time according to the time information transmitted by the EVC in the DMI_DYNAMIC frame.

6.2.11 Track Head Free acknowledgment (area D)

The display and removal of the Track Head Free window is triggered by the EVC.

6.3 MENU WINDOWS

6.3.1 Navigation (opening, closing)

The DMI manages standalone the opening and closing of menus as well as navigation between menus, except in the following cases:

- The opening of the RBC contact menu is commanded by EVC
- The opening of Main window is commanded by EVC start-up dialogue sequence (Figure 132, S0, A31, S10 document /3/).
- The opening of Main window is commanded by EVC main dialogue sequence (Figure 133, document /3/) after a new train data entry procedure has been performed by the driver
- The opening of Main window is commanded by EVC main dialogue sequence (Figure 133, document /3/) after a new level entry procedure has been performed by the driver
- The Closing of Main window is commanded by EVC main window dialogue sequence (Figure 133, document /3/) after activation of the "start" button.

6.3.2 Buttons enabling/disabling

The enabling/disabling of all menu buttons is fully controlled by EVC.

When EVC triggers the display of hourglass (main window), the DMI ensures standalone all buttons of the main window are disabled.

When EVC removes display of hourglass, it controls itself re-enabling of buttons.

The enabling of the "close" button in the main window depending on activation of hourglass is controlled standalone by DMI.

6.3.3 Buttons visibility

The visibility of the following buttons is managed standalone by the DMI depending on the current ETCS mode transmitted in DMI_M_MODE:

Button Shunting (Main window)	Shown when DMI_M_MODE != SH
Button Exit Shunting (Main window)	Shown when DMI_M_MODE == SH
Button Exit leading (Main window)	Shown when DMI_M_MODE != NL

6.3.4 Hourglass (main window)

Display and removal of hourglass is commanded by EVC.

6.4 DATA ENTRY, DATA VALIDATION AND DATA VIEW

6.4.1 Navigation (opening, closing)

The DMI manages standalone the opening and closing of data entry page, data validation page and data view page, except in the following cases:

- The opening of the driver ID entry page is commanded by EVC in startup dialogue sequence (Figure 132, D2 condition, document /3/).

- The opening of the level entry page is commanded by EVC in startup dialogue sequence (Figure 132, D2 condition, document /3/).
- The opening of the train running number page is commanded by EVC in main window dialogue sequence (Figure 133, D6 condition, document /3/).

6.4.2 Storing and display of data field values

The DMI does not store any data. The default values shown in data fields when an entry or a validation page is opened are those transmitted by the EVC prior to the page opening.

There is only one exception: For the train data view, the DMI displays the last entered data (except the train data which are retransmitted by EVC).

6.4.3 Button enabling/disabling

The disabling of the "close" buttons during Startup sequence is managed standalone by DMI. Driver ID window, Level window, RBC window.

The availability (visibility) of the "selection type" button in flexible train data entry page and the "enter data" in the fix train data entry page is controlled by EVC (see 5.3).

6.5 ACKNOWLEDGEMENTS

6.5.1 Overview

All acknowledgements are commanded by EVC.

6.5.2 Icons

Acknowledgements by means of icons are requested via DMI_ICONS.

The EVC fully controls the state of the C1 and C9 areas. Therefore, for icons to be acknowledged there, the EVC orders the display of the icon with a flashing frame and removes the icon and flashing frame once the related item has been acknowledged by the driver.

When the drivers activates an area of the DMI showing an icon for which an acknowledgement is required (i.e. icon shown with flashing frame) the DMI sends DMI_ICON_ACK to EVC to inform it of the name of activated area and the name of the icon shown in the area at the precise time the area was activated.

When the drivers activates an area of the DMI showing an icon or not and there was NO acknowledgement required (i.e. no icon shown or no flashing frame shown for icon) , the DMI ensures that no DMI_ICON_ACK is sent. Therefore, DMI_ICON_ACK is only sent when actually an activated area was showing an icon with a flashing frame.

The DMI detects that an acknowledgement for an icon is required from the driver when it receives an order to display that icon with a flashing frame.

Only the following areas are considered by the DMI for the detection of an acknowledgement request and the sequential processing of that acknowledgement:

- C1
- C9

That means that even when an icon with a flashing frame is ordered in a different area as C1/C9, i.e. e.g. B3/4/5, the DMI will not recognize it as an acknowledgement request.

6.5.3 Text messages

Acknowledgements by means of text messages are requested via the packet DMI_TEXT_MESSAGE.

For text to be acknowledged, the EVC orders the display of the message with a flashing frame. The DMI transmits the information that the text area been pressed by the driver via DMI_TEXT_MESSAGE_ACK.

The DMI manages standalone the aspect of the text message once the message acknowledged.

6.6 LANGUAGES

The language selection is managed standalone by the DMI. The display of all indicated texts is done in the appropriate language by the DMI via the language dictionary.

The managed texts include all predefined text messages which are stored in the DMI, with exception of the free text messages shall be sent by the EVC using the matching Unicode characters (UTF-8).

6.7 ICONS

The display of icons is managed standalone by DMI except for the following areas for which EVC controls the display of icon via DMI_ICONS:

- A4
- B3
- B4
- B5
- C1
- C7
- C9
- E1
- E2

The EVC is responsible for the content displayed in those areas.

When the DMI receives a request to display an icon in area different of the aforementioned ones, it ignores that requests and returns an error status code to the EVC (DMI_STATUS).

6.8 RESET OF DISPLAY ON CAB ACTIVATION

The DMI ensures standalone the display is reset when cab is open, at startup or after a previous cab closing:

- It clears the text message area.
- Pending acknowledgements, either icons or text messages, are removed.
- Any information of planning area is removed (i.e. any track condition, any speed profile, any gradient profile).
- It clears all areas where icons can be displayed, both those areas controlled by DMI (B6, B7, C8) and those areas controlled by EVC in normal operation (A4, B3, B4, B5, C1, C9, C7 (override EOA), E1, E2).
- It ensures no sub window is open, i.e. no data entry, no any data view, no data validation, no menu window, no track ahead free confirmation window.
- All menu buttons are disabled.
- No geographical information is available.
- No ETCS mode is shown (area B7 is empty).
- No ETCS level is shown (area C8 is empty).
- Any pending sounding is stopped.
- No level is available anymore for the level entry.

6.9 DISPLAY DEPENDING ON ETCS MODE

The DMI ensures consistency of default window layout depending on the current ETCS mode. E.g. It shows planning area only when in FS mode.

The display for the following specific ETCS mode is managed as follows:

- SLEEPING (SL):
  - The cabin is switched off (as if DMI_DISPLAY_CONTROL with DMI_M_ACTIVE_DMI =0 was sent).
- ISOLATION (IS): Only areas A, B and text message areas are shown.
  - All subareas within are cleared.
  - No speed information (no needle) is shown on the speedometer (only the speed dial).
  - Text message "ISOLATION" is added to the text message area.

Main window dialogue sequence (figure 119, p160)			
ID	Trigger / controlled by	Trigger packet or event	Precondition
Default window	EVC	EVC sends DMI_ENTRY_REQUEST=hide main window (coming from S7) and DMI_MENU_REQUEST= no hourglass	Ma or SR authorisation received from RBC
	EVC	Driver presses start. EVC sends DMI_ENTRY_REQUEST=hide main window as response to DMI_DRIVER_REQUEST=start (coming from S1)	start menu was enabled by EVC with DMI_MENU_REQUEST level is 0/1/STM
	EVC	EVC sends DMI_ENTRY_REQUEST=hide main window as response to DMI_DRIVER_REQUEST (coming from S1), after pressing NL	NL menu was enabled by EVC with DMI_MENU_REQUEST
S1		Coming from start-up	
	DMI	Driver presses main (coming from default window)	
	DMI	Driver has entered ID (coming from S2) and DMI_DRIVER_IDENTIFIER was sent to EVC	
	EVC	EVC sends DMI_ENTRY_REQUEST=show main in response to DMI_TRAIN_RUNNING_NUMBER (coming from D1)	Driver has entered train running number while level is 0/1/STM
	EVC	EVC sends DMI_ENTRY_REQUEST=show main and DMI_MENU_REQUEST=hourglass in response to DMI_TRAIN_RUNNING_NUMBER (coming from D2)	Driver has entered train running number while level is 2/3
	EVC	EVC sends DMI_MENU_REQUEST= no hourglass (coming from S9)	EVC received train data ack by RBC or session terminated
	EVC	EVC sends	Driver has entered level 0/1/STM

		DMI_ENTRY_REQUEST=show main in response to DMI_LEVEL_DATA (coming from S4)	
	EVC	EVC sends DMI_MENU_REQUEST=no hourglass (coming from D3)	Session with RBC cannot be opened by EVC DMI_ICONS (ST04) DMI_TEXT_MESSAGE ("trackside not compatible") sent by EVC
	EVC	EVC sends DMI_MENU_REQUEST=no hourglass (coming from D4)	Session with RBC can be opened by EVC DMI_ICONS (ST03) sent by EVC D32 to S10 states are managed by EVC if start of mission
S2	DMI	Driver presses driver id menu (coming from S1)	Driver ID menu was enabled by EVC with DMI_MENU_REQUEST
S3-1	DMI	Driver presses train data menu, DMI_TRAIN_DATA received by DMI in response to DMI_DRIVER_REQUEST=30	Train data menu was enabled by EVC with DMI_MENU_REQUEST
	DMI	Driver does not validate train data(no) (coming from S3-2)	
S3-2	DMI	Driver has entered train data (coming from S3-1) and DMI_EVC_TRAIN_DATA was received by DMI in response to DMI_TRAIN_DATA sent to EVC	
S3-3	DMI	Driver has validated train data (coming from S3-2) and DMI_TRAIN_DATA_ACK was sent to EVC	DMI_TRAIN_RUNNING_ NUMBER received from EVC (default value)
	DMI	Drivers presses train running number (coming from S1)	Train running number menu was enabled by EVC with DMI_MENU_REQUEST
D1	EVC	(coming from S3-3)	Driver has entered train running number and DMI_TRAIN_RUNNING_NUMBER was received by EVC
D2	EVC	(coming from D1)	EVC detects level 2/3
S9	EVC	EVC sends DMI_MENU_REQUEST =hourglass and DMI_ENTRY_REQUEST=show main in response to DMI_TRAIN_RUNNING_NUMBER	
S4	DMI	Driver presses level menu (coming from S1)	Level menu enabled by EVC with DMI_MENU_REQUEST DMI_LEVEL_DATA (default value) received from EVC
S5	DMI	Driver has entered level 2/3 (coming from S4)	DMI_RBC_DATA (default value) received from EVC
S8	EVC	Driver has entered RBC contact data EVC sends DMI_MENU_REQUEST =hourglass and DMI_ENTRY_REQUEST=show main in response to DMI_RBC_DATA	
D3	EVC		
D4	EVC		
S7	DMI	Driver presses start EVC activates the hourglass with DMI_MENU_REQUEST as response to DMI_DRIVER_REQUEST=start	Level is 2/3

Shunting dialogue sequence (figure 120, p166)			
ID	Trigger / controlled by	Trigger packet or event	Precondition
S0	EVC	Coming from S1 of main window dialogue sequence	Driver presses shunting Shunting menu was enabled by EVC with DMI_MENU_REQUEST
		EVC sends DMI_MENU_REQUEST=no hourglass when coming from S1.	DMI_ICONS (remove ST04) sent by EVC or DMI_TEXT_MESSAGE="SH refused" sent by EVC
D1	EVC	DMI_DRIVER_REQUEST=shunting received by EVC	
Default window	EVC	EVC sends DMI_ENTRY_REQUEST=hide main window (coming from D1)	Level is 0/1 DMI_ICONS (MO01) sent by EVC
	EVC	EVC sends DMI_ENTRY_REQUEST=hide main and DMI_MENU_REQUEST =no hourglass when coming from S1.	Onboard receives message 28 (shunting authorised)
S1	EVC	EVC sends DMI_MENU_REQUEST =hourglass when coming from D1	Level is 2/3

Override window dialogue sequence (figure 121, p168)			
ID	Trigger / controlled by	Trigger packet or event	Precondition
S0	DMI	Coming from default window from main window dialogue sequence	
	DMI	Drivers presses "EoA" and DMI_DRIVER_REQUEST=override EOA sent to EVC (coming from S1)	EOA menu was enabled by EVC with DMI_MENU_REQUEST
	DMI	Drivers presses "EoA" and DMI_DRIVER_REQUEST=override route suitability sent to EVC (coming from S3)	Route suitability menu was enabled by EVC with DMI_MENU_REQUEST
S1	DMI	Driver presses "override"	
S2	DMI	Driver presses "override route suitability"	

Note: S2 was introduced by ERSA and is not defined in ERSA specification.

Special window dialogue sequence (figure 122, p170)			
ID	Trigger / controlled by	Trigger packet or event	Precondition
S0	DMI	Coming from default window from main window dialogue sequence	
	DMI	Drivers presses "Train integrity" and DMI_DRIVER_REQUEST=train integrity sent to EVC (coming from S1)	Train integrity menu was enabled by EVC with DMI_MENU_REQUEST
S1	DMI	Driver presses "special"	
	DMI	Driver has entered new adhesion factor and DMI_ADHESION_FACTOR_DATA was sent to EVC (coming from S2)	DMI_ICONS (ST02 or remove ST02) sent by EVC
	DMI	Driver has entered new SR data and DMI_SR_DATA was sent to EVC (coming from S3)	
S2	DMI	Driver presses "adhesion"	Adhesion factor menu was enabled by EVC with DMI_MENU_REQUEST DMI_ADHESION_FACTOR_DATA (default value) was received by DMI
D1	EVC	DMI_ADHESION_FACTOR_DATA received by EVC	
S3	DMI	Driver presses "SR speed"	SR menu was enabled by EVC with DMI_MENU_REQUEST DMI_SR_DATA (default value) was received by DMI

Settings window dialogue sequence (figure 123, p172)			
ID	Trigger / controlled by	Trigger packet or event	Precondition
S0	DMI	Coming from default window from main window dialogue sequence	
	DMI	Drivers presses "Train integrity" and DMI_DRIVER_REQUEST=train integrity sent to EVC (coming from S1)	Train integrity menu was enabled by EVC with DMI_MENU_REQUEST
S1	DMI	Driver presses "settings"	
	DMI	Driver has selected new language or closed language window (coming from S2)	DMI has updated current text information according to selected language
	DMI	Driver has changed volume or closed volume window (coming from S3)	DMI has updated current sound volume according to the driver's entry
	DMI	Driver has changed brightness or closed brightness window (coming from S4)	DMI has updated current brightness according to the driver's entry
S2	DMI	Driver presses "language"	Language menu was enabled by EVC with DMI_MENU_REQUEST
S3	DMI	Driver presses "volume"	Volume menu was enabled by EVC with DMI_MENU_REQUEST
S4	DMI	Driver presses "brightness"	Brightness menu was enabled by EVC with DMI_MENU_REQUEST