blue PiraT 2
Spezifikation Telemotive Trace-File Format
Specification

Version: **3.6.0**

Last Update: **03.02.2014 13:46**

Author: **Markus van Pinxteren**

Change history

Version /Date	Chapter	Change	Author	Versions relevance (Position)		
				1	2	3
3.1.0 11.10.2011		Release Version	MVP			
3.2.0	2.8	CAN-Channel extended to 8 bit	Mvp			
3.3.0 17.02.12	2.13	FlexRay-Message (ID 0x0005) replaced by FlexRay-Extended-Message (ID 0x0015)	MKi		x	
3.3.0		**Release**				
28.03.12	2.11	GPIO Message Text updated	GKl			x
18.05.12	2.4	Extension of the serial Message to DLC	MVP			x
23.05.12	2.19	Correction of the data field sizes	MVP			x
04.06.12	2.9	MOST150 network status message adapted to SMSC Spec	RoSc			x
18.07.12	2.2	ID corrected for FlexRay data (0x0015)	MKi			x
18.07.12	2	Subsection on FlexRay data moves to the ID of the corresponding position	MKi			x
23.07.12	2.2	Message "Data record stopped" (ID 0x0083), "Data record continued" (ID 0x0084), "Start of the of the Data logger" (ID 0x0085) and " Falling asleep of the data logger " (ID 0x0086) removed, are not used in blue PiraT 2.	MKi			x
23.07.12	2.17	Description of the data in the start time of message according to the use in blue PiraT 2	MKi			x
24.07.12	2	MOST50 Message removed MOST25 Streaming Message removed ECL Message removed Added Discarded message for FlexRay Cleanup	RoSc			x
24.07.12	2.14	All types away, do not appear in the trace file.	MKi			x
26.07.12	1.1	New Chapter 1.1	mvp			x
10.08.12	2.1 2.7	Byte order corrected in Table 2 DLC corrected in Table 8	RoSc			x
22.08.12	2.12	Added unique MessageType for FlexRay CAS and MTS symbol in Table 20.	RoSc		x	
04.09.12		**Release 3.4.0** **FW/Client 1.7.2**				
01.08.12	2.5	New Ethernet protocols included	JLi			x
07.03.13	2.17	Timezone Message added	JLi		x	x
14.03.2013		ECL-Message added	mvp		x	

29.04.2013	2.12	Correction of byte description	mvp			**x**
06.06.2013		**Release 3.5.0** **FW/Client 1.8.1**				
23.07.2013		STR ECL-Message description corrected	RoSc			
19.11.2013	2.1	Description of Message Length field in Message Header corrected	mvp			**x**
04.12.2013	2.5	Ethernet Protocol type MII Mode field added	MaHa		**x**	
28.01.2014	2.9.4 2.9.5	MOST150 Streaming Message added MOST150 (De-)Allocation Message added	RoSc			
28.01.2014		Ethernet Protocol type EP_MII Mode type 8 field added	MaHa		**x**	
30.01.2014		LIN Wake Up Message and LIN Status Message added	MaHa		**x**	
03.02.2014		**Release 3.6.0** **FW/Client 1.9.1**				

Table of Contents

1. Telemotive Trace File Format

The Telemotive trace file format is a proprietary, binary file format for storing recorded trace data. The format is based on a sequence of different types of messages (see Section 2) with the exception of the file identifier and version number (see Table 1).

	Description	Field length in Bytes
Header	File-Identifier „TelemotiveLogFile “	32
	Version number of Log file x.y.z.a (see Section 1.1)	4
	FileTimeMessage (see Table 37)	22
	[System-Configuration (multiple SysConfigMessages - Table 35)]	X
	SeparatorMessage (SystemMessage - Table 34)	28
	Tracedata	Y
	Eof-Message	18

Table 1: TMT file structure

A TMT file always starts with the file identifier **„TelemotiveLogFile“**.
Next is the file version, which refers to the valid specification document. After the version number only messages follow as described in Chapter 2.

The first message is always a **FileTimeMessage**. This message contains the data bytes, the UTC start time of the file in µs since 01.01.1970. All other time stamps in the header of the following messages are relative timestamps (µs) to this absolute start time.

The FileTime message follows an **optional** block with **SysConfigMessages**, which indicates which configuration of the data logger was used at the time of recording.

A **SystemMessage** of the type **"Separator"** closes the file header. Then the data recorded by the logger follows the formatted messages of that type. The trace data is always stored in the file in chronological order after it arrives on the logger.

The last message in a file is always an Eof message (see Table 41).

1.1. Versioning

It uses only the first three digits of the version number. The fourth place is re-served and currently always has the value 0.

The first two digits refer to the valid specification document; similarly, the first two points of the version number of the specification are crucial.
The third digit version of the TMT file version is used for bug fixes, which do not require any change in the specification document.
The third digit version of the document version is used for changes in the document, which do not require adjustment of the TMT implementation (bug fixes, editing descriptions etc.).

1.2. Processing with blue PiraT 2 Client Software

The processing of TMT files with the blue PiraT 2 client software is only possible with unchanged and complete offline data records – **created by the client**. Correct processing is dependent on the file name location within the offline data records. Additionally correct processing is dependent on peripheral configuration and database files.

2. The Telemotive Message Format

2.1. General Message Structure

Each message consists of a message header and a message body. The structure of the header is the same for each message type as opposed to the body, which depends on the type (Message-ID) of data to be stored.

If not indicated otherwise, all values are stored in big-endian byte order.

Description	Field length (in Bytes)	
Message length exclusive the length of this field	2	Header
Message-ID	2	
Reserved	2	
Timestamp, relative to the start time in the file header (see chapter 1)	8	
Payload – depending on Message-ID	n	Body

Table 2: General construct of a Message

2.2. Registered Message-ID's

ID	Type	Table
0x0000	Marker set	Tabelle 4
0x0001	[reserved]	-
0x0002	[reserved]	-
0x0003	Serial Data	Table 5
0x0004	Ethernet (Rx) Data	Table 6
0x0005	[reserved]	-
0x0006	LIN Data	Table 10
0x0007	[reserved]	-
0x0008	Ethernet (Tx) Data	Table 6
0x0009	[reserved]	-
0x000A	ECL-Message (MOST50)	-
0x000B	CAN Telegram	Table 19
0x0010	MOST150 Message	Table 20
0x0011	MOST50 Message	-
0x0012	Analog Data	Table 27
0x0013	GPIO Data	Table 28
0x0014	MOST25 Message	Table 29
0x0015	FlexRay Data	Table 33
0x0080	System Message (e.g. Error Message)	Table 34
0x0081	Statement of Configuration Data	Table 35
0x0082	Jump in the Timebase (i.e. Clock renew configured/synchronized)	(*)
0x0083	[reserved]	-
0x0084	[reserved]	-
0x0085	[reserved]	-
0x0086	[reserved]	-
0x0087	Temperature	Table 36
0x0088	Start Time of the Data in the File	Table 37
0x0089	Trigger counter will recessed	(*)
0x008A	Time Zone	
0x0090	[reserved]	-
0x0091	[reserved]	-

0x0092	Message rejected	Table 39
0x0093	Configuration Data CARMEN / Caromee	Table 40
0x0094	Meta Data-Message	see XTMT-specification
0x00FF	End of File	Table 41

Table 3 Message format

(*) These messages consist only of message type, length and time stamp. They require no further data.

2.3. Marker-Message (ID 0x0000)

Description	Length
Marker counter (consecutive Index)	2 Bytes
Absolute 64 Bit UTC Timestamp (µs since 01.01.1970)	8 Bytes

Tabelle 4: Format of a Marker-Message

2.4. Serial Message (ID 0x0003)

The serial Data (see Table 5) will divided into blocks according to the time of arrival, i.e. a block is not necessarily exactly one ASCII line. This is the file format for flexible binary formats.

Description	Length
Channel Number	1 Byte
Type of the Protocol 0x00 no Protocol 0x01 MASK Trace Client 0x02 MASK GN Logger	1 Byte
Status of the Serial Interface for these Data block Bit 0: Overrun Bit 1: Parity Error Bit 2: Framing Error Bit 3: Break	1 Byte
DLC – Number of following bytes	2 Bytes
Serial Data bytes	n Bytes

Table 5: Payload Format of a serial Message (ID0x03)

2.5. Ethernet-Message (ID 0x0004 und 0x0008)

For Ethernet messages, there are two addresses, one for the received data (RX = 0x0004) and one for sent data (TX = 0x0008). For example: Data sent for a GNLog protocol during a dial-up:

Description	Length
Channel number	1 Byte
Protocol Type 0 – GNLogger 1 – RAW 2 – UTF8 3 – DLT_BMW 4 – UDP-Server 5 – Spy Mode 6 – Eso Trace 7 – MII Mode (unused) 8 – EP_MII Mode	1 Byte
Data depending on Protocol Type	n Bytes

Table 6: Format of an Ethernet Message

2.5.1. Ethernet-Message Protocol Type (ID 0,1,2,3,4,5,6)

Description	Length
Ethernet Data	n Bytes

Table 7: Format of GNLogger, RAW, UTF8, DLT_BMW, UDP-Server, Spy Mode and Eso Trace Protocol Type

2.5.2. Ethernet-Message Protocol Type (MII Mode) (ID7)

➔ do not use → replaced by ID8

The “Length of Ethernet Data” field indicates how many bytes of Ethernet Data follow. This length is exclusive “Padding bytes”.

Description	Length
Length of Ethernet Data	2 Bytes
Ethernet Data	n Bytes
Optional: Padding bytes (0x00) for Quadlet alignment	m Bytes

Table 8: Format of MII Mode Protocol Type

2.5.3. Ethernet-Message Protocol Type (MII Mode) (ID8)

The “Status” field contains the status of the Ethernet Phy.

- Bit 0 Phy Error. If the error signal of the Ethernet Phy occurs during receiving a message the Bit is set to 1. The Bit is 0, if no error is detected.
- Bit 1 – 7 are reserved

The “Length of Ethernet Data” field indicates how many bytes of Ethernet Data follow. This length is exclusive “Padding bytes”.

Description	Length
Reserved	3 Bytes
Status	1 Byte
Length of Ethernet Data	2 Bytes
Ethernet Data	n Bytes
Optional: Padding bytes (0x00) for Quadlet alignment m = 1,2,3	m Bytes

Table 9: Format of EP_MII Mode Protocol Type

2.6. LIN-Message (ID 0x0006)

2.6.1. LIN Data Message

Description	Length
Channel number	1 Byte
LIN-Status	1 Byte
Period of Bits (1/Baudrate)	2 Bytes
Transmission time of the whole Frames	2 Bytes
Period of Sync Break	2 Bytes
Period of Break Delimiters	2 Bytes
Period of Headers	2 Bytes
LIN-ID	1 Byte
Number of Data bytes incl. Checksum	1 Byte
LIN Data bytes	n Bytes
Checksum	1 Byte
Padding-Byte	0..1 Bytes

Table 10: Format of a LIN Data Message

The LIN-ID field contains the "Protected Identifier", consisting of six addressing bits and two parity bits. Status information about the validity of the parity will not be stored in the data record.

The "number of data bytes" field can accept values from 0 to 9 and includes the response data and the checksum. The number of bytes in the response field is therefore lower by 1. A value of 0 may result in an "event triggered frame", which remains unanswered, or for exceeding the allowable response time by an addressed slave in case of an Unconditional frame.

The checksum can be formed in different ways depending on the version of the LIN standard. Information about the generation and the validity of the checksum is not stored in the data record.

The 4CAN2LINext can attach a new byte to the record data to maintain a 16-bit alignment. The length specified in the header includes the padding byte, while specifying the data bytes within the LIN data set does not.

The status field contains flags which describe some specific cases when receiving a LIN message and change the meaning of the data fields. These values apply for the v6.0.1 firmware:

- Bit 0 indicates a wake-up frame. The message contains no more data. The byte 0 of the data area may be closed due to the length of the wake-up frame. The fields with the times are not important.
- Bit 7 shows an error status. A record that has this bit is set and should not be represented as a valid LIN message. In addition, individual bits 1..6 trigger could be labeled for this indication.
- Bit 3 is set when data is received, but was not introduced with a Sync-Break (LIN_ERR_SPURIOUS). Because of this format breach, the data cannot be rated as a LIN frame. The individual characters are saved until the maximum record data length, or a timeout, and then stored in the payload of the LIN data message. The additional data fields of the LIN message (LIN identifier, time) do not contain valid values.
- Bit 4 indicates an incomplete LIN message that has already been terminated with the delivery of the sync break (LIN_ERR_BREAK). Then all data fields in the LIN data contain no valid data.

- Bit 5 indicates a LIN message, in which only the Sync-Break was received (LIN_ERR_SYNC). In LIN record data only the total transmission time is stored.
- Bit 6 is set when in a LIN frame the Protected Identifier is not received. Only the sync break and the sync pattern received. (LIN_ERR_IDENTIFIER).
- Bits 1..2 are not yet clearly specified. If any of the bits 3..6 are set, this should be considered as a fault indicator. The record should not be regarded as valid LIN message.

The four fields for the time period include information relative to certain reference points within a LIN frame in the unit µs. Formats can be derived from these data fields for each export. The range of fields includes the range 0 to 65535, where "0" is the designation for unavailable information. The value will never exceed 65535, which will remain as the maximum value.

The time stamp in the header of the message refers to the date on which the receipt of the telegram was completed in 4CAN2LINext (timeout or the beginning of a new frame). The transmission time of the frame or response can be calculated back to the start of transmission of the master or the slave.

2.6.2. LIN Wake Up Message

Description	Length
Channel number	1 Byte
LIN-Status	1 Byte
Period of Bits (1/Baudrate)	2 Bytes
Wake up pulse time	2 Bytes

Table 11: Format of a LIN Wake Up Message

2.6.3. LIN Status Message

Description	Length
Channel number	1 Byte
LIN-Status	1 Byte
Bit time	2 Bytes

Table 12: Format of a LIN Status Message

2.7. ECL-Message (ID 0x000A)

Description	Length
ECL-message type 0x06 Electrical WU Symbol (EWU) 0x07 System Test WU Symbol (STWU) 0x08 System Test Parameters (STP) 0x09 System Test Result (STR) 0x0A Undefined Pulse	1 Byte
Data depending on ECL message type (see following sections)	7 Bytes

Table 13: Format of an ECL message

2.7.1. ECL EWU Message (ECL type 0x06)

Description	Length
Padding Bytes	3 Bytes
Transmission time of EWU symbol (µs)	4 Bytes

Table 14: Format of an ECL EWU Message

2.7.2. ECL STWU Message (ECL type 0x07)

Description	Length
Padding bytes	3 Bytes
Transmission time of STWU symbol (µs)	4 Bytes

Table 15: Format of an ECL STWU message

2.7.3. ECL STP message (ECL type 0x08)

Description	Length
Padding Bytes	2 Bytes
Parameters: Bit 0: Parameter P1 Bit 1: Parameter P2 Bit 2: Parameter P3 Bit 3: Parameter P4 Bit 4: Parameter P5 Bit 5-7: Reserved	1 Byte
Transmission time of the STP sequence from begin of tTestStart to end of SyncBit (µs)	4 Bytes

Table 16: Format of an ECL STP message

2.7.4. ECL STR message (ECL type 0x09)

Description	Length
Padding Bytes	2 Bytes
Results: Bit 0: O_n Bit 1: E_n Bit 2-6: node class (1-20) Bit 7: reserved	1 Byte
Transmission time of STR sequence. From beginning of tTestSync to falling edge within time window of E_n and O_n (µs). If both E_n and O_n remain high the transmission time is marked invalid (0xFFFFFFFF).	4 Bytes

Table 17: Format of an ECL STR message

2.7.5. ECL Undefined Pulse message (ECL type 0x0A)

Description	Length
Padding Bytes	3 Bytes
Transmission time of low pulse (µs)	4 Bytes

Table 18: Format of an ECL Undefined Pulse message

2.8. CAN-Message (ID 0x000B)

Description	Length
Channel number	1 Byte
CAN-Message type 0x00 Standard 0x01 Error frame 0x02 Transmitted frame	1 Byte
CAN-Status (Bit 7..4) 0x00 Status ok 0x01 Stuff Error 0x02 Format Error 0x03 Acknowledge Error 0x04 Bit 1 Error 0x05 Bit 0 Error 0x06 CRC Error 0x07 Overrun	1 Byte
Number of Data bytes (DLC) (Bit 3..0)	1 Byte
Extended ID Flag (2.0b Standard) (Bit 31) CAN ID (Bit 30..0)	4 Bytes
CAN Date bytes	N Bytes

Table 19: Payload Format of a CAN-Message

2.9. MOST150 Message (ID 0x0010)

Description	Length
MOST-Message type 0x00 MOST Control-Message 0x01 MDP (MOST Data Packet) 0x02 MEP (MOST Ethernet Packet) 0x03 MOST Streaming Data (Synchronous) 0x11 MDP reduced 0x12 MEP reduced 0x20 (De-)Allocation Event 0xF0 Network-Status-Message	1 Byte
Status byte Bit 0: MOST Lock Bit 1: Light Bit 6-2: Reserved Bit 7: Message abort A (De-)Allocation Event has a status byte which is always 0x00!	1 Byte
Padding bytes for quadlet alignment (always 0x00)	2 Bytes
Data depends on MOST- Message type (see the following tables)	n Bytes

Table 20: Payload Format of MOST150-Message

2.9.1. MOST150 Channel-Control-Message (MOST-ID 0x00)

Description	Length
Spy Length	2 Bytes
Admin Bytes	2 Bytes
Priority	1 Byte
Receive-Address (low byte first)	2 Bytes
Preemptive Acknowledge (PACK)	1 Byte
Packet length (high byte first)	2 Bytes
Packet number	1 Byte
Source address (high byte first)	2 Bytes
Date bytes	n Bytes
CRC (high byte first)	2 Bytes
CACK	1 Byte

Table 21: Data Format of MOST150 Control-Channel-Message

The packet length-1 indicates how many bytes follow.

2.9.2. MOST150 Data Package (MDP) (MOST-ID 0x01)

Spy Length	2 Bytes
Admin Bytes	2 Bytes
Packet length (high byte first)	2 Bytes
Receive-Address (low byte first)	2 Bytes
PACK (preemptive Acknowledgement)	1 Byte
Packet number	1 Byte
Source-Address (high byte first)	2 Bytes
Data	n Bytes
CRC (high byte first)	2 Bytes
CACK	1 Byte

Table 22: Data Format of MOST Data Packet (MDP)

The length of the data field is calculated from the packet length - 1 minus the other control bytes: (2 + 1 + 1 + 2 + 2 + 1 = 9)

2.9.3. MOST150 Ethernet Packet (MEP) (MOST-ID 0x02)

Spy Length	2 Bytes
Admin Bytes	2 Bytes
Packet length (high byte first)	2 Bytes
Ethernet-Receive-Address (low byte first)	6 Bytes
PACK (preemptive Acknowledgement)	1 Byte
Data	n Bytes
CRC (low byte first)	4 Bytes
CACK	1 Byte

Table 23: Format of MOST Ethernet Packet (MEP)

The length of the data field is calculated from the packet length - 1 minus the other control bytes: (6 + 1 + 4 + 1 = 12)

2.9.4. MOST150 Streamingnachricht (MOST-ID 0x03)

Beschreibung	Länge
reserved (0x00)	2 Bytes
streaming data length [bytes] (header field and padding bytes not included)	2 Bytes
header length [bytes] (Length includes first field of channel label up to and including the last field of channel width; Length field and padding bytes are not included)	2 Bytes
channel width and channel label Bit 0-8: channel label 0 Bit 9-15: channel width 0 [bytes] in case more than one channel is active follows: Bit 16-24: channel label 1 Bit 25-31: channel width 1 [bytes] etc. („n“ refers tot he number of active streaming channels)	n*2 Bytes
padding bytes for quadlet alignment (always 0xff)	0 or 2 Bytes
streaming data	m Bytes

Tabelle 24: Format der Daten eine MOST-Streamingnachricht

2.9.5. MOST150 (De-)Allocation Event (MOST-ID 0x20)

Channel Allocation State value is 12 in case of an Allocation, value is 13 in case of a De-Allocation.	1 Byte
Channel Allocation Information: FreeBytes	2 Bytes
Channel Allocation Information: Channel Width	2 Bytes
Channel Allocation Information: Connection Label	2 Bytes
Reserved	5 Bytes

Table 25: Data Format of a MOST (De-)Allocation Event

2.9.6. MOST150 Status Message (MOST-ID 0xF0)

Length (high byte first)	2 Byte
MPR (low byte first)	2 Byte
MDC (low byte first)	2 Byte
Network status Byte 1 bit 0: System Lock Flag bit 1: Shutdown Flag bit 2-7: Reserved	1 Byte
Network status Byte 2 Reserved	1 Byte
Node Position (low byte first)	2 Byte
Timestamp (low byte first)	4 Byte

Table 26: Data Format of a MOST-Network-Status-Message

2.9.7. MOST150 Data Packet (MDP) – reduced (MOST-ID 0x11)

The MDP message (see 2.9.2) does not contain all the data provided. It ends after the header in the specified length (see Table 22).

2.9.8. MOST150 Ethernet Packet (MEP) – reduced (MOST-ID 0x12)

The MEP message (see 2.9.3) does not contain all the data provided. It ends after the header in specified length (see Table 23).

2.10. Analog Message (ID 0x0012)

Each analogue signal message stores the values of all active analogue interfaces. The payload of an analogue message consists of 1 – n sequences of byte table below.
The exact number of the sequences (n) results from the total length of the message in the header (see Table 2).

Description	Length
Port-Index	2 Bytes
Direction 0x00 – unknown 0x01 – in 0x02 – out	1 Byte
Signal value (int32_t)	4 Bytes
Exponent (of 10) for scaling of the Value (int8_t)	1 Byte
Unit of the Value 0x00 undefined 0x01 RAW 0x01 VOLT 0x02 AMPERE	1 Byte

Table 27: Format of Analog-Signal-Value

2.11. GPIO Message (ID 0x0013)

Each digital message stores one or more active signal values of the GPIO interfaces. The payload of a GPIO message consists of [1 : n] sequences of bytes shown in the table below. The exact number of the sequences (n) results from the total length of the message in the header (see Table 2).

Description	Length
Port-Index (= Channel number)	2 Bytes
Direction 0x00 – unknown 0x01 – in 0x02 – out	1 Byte
Data-Mask	2 Bytes
Signal-Value	2 Bytes

Table 28: Format of GPIO-Signal-Value

2.12. MOST25 Message (ID 0x0014) (Source: SpyNIC)

Description	Length
MOST-Message 0x00 MOST Control-Message 0x01 MDP (MOST Data Packet) 0x02 Reserved 0x03 Reserved 0x11 MDP reduced 0x12 Reserved 0xF0 Network-Status-Message	1 Byte
Status byte Bit 0: MOST Lock Bit 1: Light Bit 2-6: Reserved Bit 7: Message abort	1 Byte
Anzahl der folgenden Bytes bis einschließlich ACK (nur für Kontroll- und MDP-Nachrichten, sonst immer 0x00)	2 Bytes
Data depend on MOST-Message type (see following Table)	n Bytes

Table 29: Payload Format of a MOST-Message

System Lock is set immediately after a lock and without waiting for a stable lock.

2.12.1. MOST25 Control-Channel-Message (Message type 0x00)

Description	Length
MType	1 Byte
Receive- Address (low byte first)	2 Bytes
Reserved	1 Byte
Package length (high byte first)	2 Bytes
Reserved	1 Byte
Source-Address (high byte first)	2 Bytes
Data bytes	n Bytes
CRC (high byte first)	2 Bytes
Reserved	1 Byte
ACK	1 Byte

Table 30: Data Format of a MOST-Control-Channel-Message

The packet length indicates how many bytes follow (including ACK byte).

2.12.2. MOST25 Data Packet (MDP) (Message type 0x01)

Reserved	1 Byte
Received-Address (low byte first)	2 Bytes
Reserved	1 Byte
Package length(high byte first)	2 Bytes
Reserved	1 Byte
Source-Address (high byte first)	2 Bytes
Data	n Bytes
Reserved	4 Bytes

Table 31: Data Format of a MOST Data Packet (MDP)

The packet length indicates how many bytes follow (including ACK byte).

2.12.3. MOST25 Status Message (Message type 0xF0)

Length (high byte first)	2 Bytes
MPR	1 Byte
SBC	1 Byte
Network status Byte 1 Bit 0: MOST Lock Bit 1: Light Bit 2-7: Reserved	1 Byte
Channel Allocation Information: FreeBytes	1 Byte
Channel Allocation Information: Channel Width	1 Byte
Channel Allocation Information: Connection Label	1 Byte
Reserved	4 Bytes

Table 32: Data Format of a MOST-Network-Status-Message

2.12.4. MOST25 Data Package (MDP) – reduced (Message type 0x11)

The MDP message (see 2.12.2) does not contain all the data provided. It ends after the header in the specified length (see Table 2).

2.13. FlexRay-Extended-Message (ID 0x0015)

Description	Length
Message Type *Flexray Symbol:* 0x00 WUS 0x01 Reserved 0x02 Reserved 0x03 undefined Low Pulse (neither Symbol or TSS) 0x04 CAS 0x05 MTS *Flexray Message:* 0x10 = static Message 0x11 = dynamic Message 0x12 = invalid Message (FES/DTS false) *Message structure at 0x12:* MsgT[1] + ChannelNr[1] + RxBytes[2] + 0x0000 + FR-Header[5] + Payload[2*n] + TrailerCRC[3]. Once the end byte is detected during a message and neither BSS nor FES / DTS are detected, all other data bits will issue in raw format. Incorrect BSS / FES / DTS bit sequence completely enclosed.	1 Byte
Channel number 0x00 Channel 1A 0x01 Channel 1B 0x02 Channel 2A etc…	1 Byte
Number of bytes received (corresponds to the number of received BSS)	2 Bytes
indicator bits Bit 0: Startup Frame Indicator Bit 1: Sync Frame Indicator Bit 2: Null Frame Indicator Bit 3: Payload Preamble Indicator Bit 4-7: Reserved	1 Byte
FrameID	2 Bytes
Payload-length (n) (Number of 16-Bit-Worte)	1 Byte
Header-CRC	2 Bytes
Cycle-Count	1 Byte
Payload-Data as 16 Bit-Words	2*n Bytes
Trailer-CRC	3 Bytes

Table 33: Data Format of a FlexRay-Extended-Message

2.14. Data Format of a System-Message (ID 0x0080)

Description	Length
Type: 0x00 Info message 0x01 Version number 0x09 Info of GN-Log/Ethernet recording 0x0E Separator (End of the header in the trace file) 0x80 Warning 0x90 Error-Message	1 Byte
System-Message in ASCII	n Bytes

Table 34: Payload Format of a System-Message

2.15. Data Format of a Configuration-Message (ID 0x0081)

Description	Length
Data configuration as String	n Bytes

Table 35: Format for specifying configuration data

2.16. Temperature-Message (ID 0x0087)

Description	Length
Temperature in °C	2 Bytes

Table 36: Format Temperature-Message

2.17. Start time-Message(ID 0x0088)

The first message of each TMT file is a start time message. The data of the message contains the date on which the data logger has been started, as an absolute 64-bit timestamp. All following messages in this file contain a time relative to this time timestamp in the message header.

Description	Length
Absolute 64 Bit UTC Timestamp (usec since 01.01.1970)	8 Bytes

Table 37: Format of Start-Time-Message

2.18. Timezone(ID 0x008A)

The second message of each TMT file is a timezone message. The data of the message contains the timezone of the logged data, as an UTF8-String.

Description	Length
Timezone as UTF-8 string	n Bytes

Table 38: Format of Timezone-Message

2.19. Message for rejected Bus-Messages (ID 0x0092)

The message is sent when the data bus message will be logged regularly.

Description	Length
Message type of the rejected Message 0x00 MOST150 Control Message 0x01 MOST150 Network-Status-Message 0x02 MOST150 MDP (MOST Data Packet) 0x03 MOST150 MEP (MOST Ethernet Packet) 0x10 Flexray Message	1 Byte
Device ID (0x00)	1 Byte
Padding bytes for Quadlet alignment (0x0000)	2 Bytes
Timestamp of when the message was recorded further (usec relative to file Start time) (**)	8 Bytes
Timestamp of when the message is no longer recorded (usec relative to file Start time) (**)	8 Bytes
Number of rejected messages	4 Bytes

Table 39: Message Format for rejected Messages

(**)The time stamp is a relative time stamp to the absolute timestamp that is stored in the first message of each TMT file. This first message is always a start time message (ID 0x48) (see 2.17). Timestamps are generally stored as UTC (Microseconds since 01.01.1970).

2.20. Format of a Configuration Channel for Channel-Configuration in CARMEN / Caromee (ID 0x0093)

Description	Length
Reserved	1 Byte
Interface-Typ 0x00 - undefined 0x01 - CAN, 0x02 - SERIAL 0x03 - MOST25 0x04 - FLEXRAY 0x05 - LIN 0x06 - ETHERNET 0x07 - CAMERA 0x08 - MOST150 0x09 - MOST50 0x0A - ANALOG 0x0B - CCPXCP 0x0C - GPIO	1 Byte
Channel-Index	1 Byte
Recording active [true/false]	1 Byte
CARMEN / Caromee Container-ID	4 Bytes
Length of the following Channel name	1 Byte
Channel name	n Bytes

Table 40: Format of a Channel-Configuration for CARMEN / Caromee

2.21. EndOfFile-Message (ID 0x00FF)

Each Trace-File ends with an EOF Message.

Description	Length
Reserved	32 Bits

Table 41: Format – End of File (ID0x00FF)