



blue Pirat

BY MAGNA



Specification Telemotive Trace (TMT) File Format / 15.10.2020

TMT-Version 3.9.3
Dokumentenversion 1.2

Table of contents

1	LICENSE AGREEMENT	6
2	PRODUCT LIABILITY	7
2.1	Authors:	8
3	Telemotive Trace File Format	9
3.1	Versioning.....	9
3.2	Processing with blue PiraT 2 Client Software	10
4	The Telemotive Message Format.....	11
4.1	General Message Structure	11
4.2	Registered Message-ID's.....	12
4.3	Marker-Message (ID 0x0000)	12
4.4	Serial Message (ID 0x0003)	13
4.5	Ethernet-Message (ID 0x0004 und 0x0008)	13
4.5.1	Ethernet-Message Protocol Type (ID 0,1,2,4,5,6).....	14
4.5.2	Ethernet-Message Protocol Type DLT (ID 3).....	14
4.5.3	Ethernet-Message Protocol Type MII Mode (ID 7).....	14
4.5.4	Ethernet-Message Protocol Type EP_MII Mode (ID 8)	14
4.6	LIN-Message (ID 0x0006).....	16
4.6.1	LIN Data Message.....	16
4.6.2	LIN Wake Up Message.....	17
4.6.3	LIN Status Message	17
4.7	ECL-Message (ID 0x000A)	19
4.7.1	ECL EWU Message (ECL type 0x06)	19
4.7.2	ECL STWU Message (ECL type 0x07)	19
4.7.3	ECL STP message (ECL type 0x08)	19
4.7.4	ECL STR message (ECL type 0x09)	20
4.7.5	ECL Undefined Pulse message (ECL type 0x0A)	20
4.8	CAN-Message (ID 0x000B)	20
4.9	Container Message (ID 0x000C)	22
4.10	TTY-Message (ID 0x000D)	22
4.11	MII-Message (ID 0x000E)	22
4.12	MOST150 Message (ID 0x0010)	23

4.12.1 MOST150 Channel-Control-Message (MOST-ID 0x00)	24
4.12.2 MOST150 Data Package (MDP) (MOST-ID 0x01)	24
4.12.3 MOST150 Ethernet Packet (MEP) (MOST-ID 0x02)	24
4.12.4 MOST150 Streaming message (MOST-ID 0x03)	25
4.12.5 MOST150 (De-)Allocation Event (MOST-ID 0x20)	25
4.12.6 MOST150 Status Message (MOST-ID 0xF0)	26
4.12.7 MOST150 Data Packet (MDP) – reduced (MOST-ID 0x11)	26
4.12.8 MOST150 Ethernet Packet (MEP) – reduced (MOST-ID 0x12)	26
4.13	Analog Message (ID 0x0012)	26
4.14	GPIO Message (ID 0x0013)	28
4.15	MOST25 Message (ID 0x0014) (Source: SpyNIC)	28
4.15.1 MOST25 Control-Channel-Message (Message type 0x00)	29
4.15.2 MOST25 Data Packet (MDP) (Message type 0x01)	29
4.15.3 MOST25 Status Message (Message type 0xF0)	29
4.15.4 MOST25 Data Package (MDP) – reduced (Message type 0x11)	30
4.16	FlexRay-Extended-Message (ID 0x0015)	31
4.17	PHY-Status-Message (ID 0x001F)	32
4.17.1 Register Number – Value Pairs (general)	32
4.17.2 Register Number – Value Pairs (TJA1100 specific)	33
4.18	Data Format of a System-Message (ID 0x0080)	33
4.19	Data Format of a Configuration-Message (ID 0x0081)	33
4.20	Temperature-Message (ID 0x0087)	34
4.21	Start time-Message (ID 0x0088)	34

4.22	Timezone(ID 0x008A).....	34
4.23	Message for rejected Bus-Messages (ID 0x0092).....	34
4.24	Extended message for rejected Bus-Messages (ID 0x0095).....	36
4.25	Format of a Configuration Channel for Channel-Configuration in CARMEN / Caromee (ID 0x0093)	37
4.26	EndOfFile-Message (ID 0x00FF)	37
5	Support.....	38
5.1	Service Center	38
5.2	OTRS Ticket system.....	38
6	List of tables	40
7	Open Source	42
8	Version history.....	43
9	Contact	46

1 LICENSE AGREEMENT

Please read the license agreement of this license contract carefully, before you install the software. By the installation of the software you agree to the conditions of this license contract.

This software-license agreement, in the following called “license”, contains all rights and restrictions for final users that regulate the use of the accompanying software, operating instructions and other documents, in the following called as “software”.

1. This license contract is an agreement between licensor and licensee, who is being licensed to use the named software.
2. Licensee acknowledges that this is only a limited nonexclusive license. This means, that the licensee has no right to allocate sublicenses. Licensor is and remains the owner of all titles, rights and interests in the software.
3. The software is a copyright property of the MAGNA Telemotive GmbH. The program or parts of it may not be further licensed to third parts, rented, sold or be further marketed in any form without explicit written approval by MAGNA Telemotive GmbH. The user may neither change the software and their components, nor modify, nor redevelop or decompile otherwise in any form.
4. This software is subject to no warranty. This software is sold as is, without any warranty. If at any time, a user changes his system, we hold no responsibility to change our software to make it work again.
5. This license permits licensee to install the software on more than one computer system, as long as the software will not be used on more than one computer system simultaneously. Licensee will not make copies of the software or allow copies of the software to be made by others, unless authorized by this license agreement. Licensee may make copies of the software for backup purposes only. Licensee is not entitled to transmit or to transfer the software or its rights from this license agreement.
6. Licensor is not liable to licensee for any damages, including compensatory, special, incidental, exemplary, punitive or consequential damages, connected with or resulting from this license agreement or licensee's use of this software.
7. Licensee agrees to defend and indemnify licensor and hold licensor harmless from all claims, losses, damages, complaints or expenses connected with or resulting from licensee's business operations.
8. Licensor has the right to terminate this license agreement and licensee's right to use this software upon any material breach by licensee. The duration of the license contract is indefinitely determined.
9. Licensee agrees to return all copies of the software to licensor or to destroy them upon termination of the license contract.
10. This license agreement replaces and supersedes all prior negotiations, dealings and agreements between licensor and licensee regarding this software.
11. This license contract is subject to German law.
12. If a regulation of this license contract is void by law, the validity of the remaining regulations is not affected. If there is such a regulation it will be replaced by a valid, according to the legal regulations and enforceable regulation with similar intention and similar economic consequence.
13. The license contract is effective by delivery of the software of the licensor to the licensee and/or by usage of the software by the licensee. This license contract is also valid without licensor's signature.
14. The license automatically goes out if the licensee does not agree to the license regulations described here or offend against the license regulations of this license contract. With ending the license contract the licensee is obliged to extinguish or to destroy the software and all copies of it no matter if installed or stored on disk or to hand all of it back to MAGNA Telemotive GmbH.
15. The licensee is liable for all damages caused to the licensor by the violation of these license regulations.

2 PRODUCT LIABILITY

The General Terms and Conditions of Sale and Delivery of MAGNA Telemotive GmbH can be found on our website (www.telemotive.de) under imprint.



2.1 Authors:

- Markus van Pinxteren
- Gerhard Klügl
- Robert Schwabe
- Javier Lipiz
- Daniel Scheib

3 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 3.1)

	Description	Field length in Bytes
Header	File-Identifier „TelemotiveLogFile“	32
	Version number of Log file x.y.z.a (see section 3.1)	4
	FileTimeMessage (see section 4.18)	22
	[System-Configuration (multiple SysConfigMessages – see section 4.17.2)]	X
	SeparatorMessage (SystemMessage – see section 4.17.1)	28
	Tracedata	Y
	Eof-Message	18

Table 3.1: 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 μ s since 01.01.1970. All other time stamps in the header of the following messages are relative timestamps (μ s) 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 4.46)

3.1 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 digits of the version number of the specification are crucial.

The third digit of the TMT file version is used for bug fixes, which do not require any change in the specification document.

The third digit 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.).

3.2 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.

4 The Telemotive Message Format

4.1 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	
Bit 15: 0: Message passed through filters with preserve indication 1: Message passed through filters with discard indication Bit 14: 0: Mii-Message gets stored in Tracefile 1: Mii-Message gets filtered because of customer filter configuration Bit 13: 0: Mii-Message gets forwarded to PS 1: Mii-Message gets filtered for PS Bit 12 – 0: Reserved	2	
Timestamp, relative to the start time in the file header (see chapter 3)	8	
Payload – depending on Message-ID	n	

Table 4.1: General construct of a Message

4.2 Registered Message-ID's

ID	Type	Table
0x0000	Marker set	Table 4.3
0x0001	[reserved]	-
0x0002	[reserved]	-
0x0003	Serial Data	Table 4.4
0x0004	Ethernet (Rx) Data	Table 4.5
0x0005	[reserved]	-
0x0006	LIN Data	Table 4.9
0x0007	[reserved]	-
0x0008	Ethernet (Tx) Data	Table 4.5
0x0009	[reserved]	-
0x000A	ECL-Message (MOST50)	Table 4.12
0x000B	CAN Telegram	Table 4.18
0x000C	Container Message	Table 4.19
0x000D	TTY Message	Table 4.20
0x000E	MII Message	Table 4.21
0x0010	MOST150 Message	Table 4.22
0x0011	MOST50 Message	-
0x0012	Analog Data	Table 4.29
0x0013	GPIO Data	Table 4.30
0x0014	MOST25 Message	Table 4.31
0x0015	FlexRay Data	Table 4.32
0x0080	System Message (e.g. Error Message)	Table 4.33
0x0081	Statement of Configuration Data	Table 4.34
0x0082	Jump in the Timebase (i.e. Clock renew configured/synchronized)	(*)
0x0083	[reserved]	-
0x0084	[reserved]	-
0x0085	[reserved]	-
0x0086	[reserved]	-
0x0087	Temperature	Table 4.40
0x0088	Start Time of the Data in the File	Table 4.41
0x0089	Trigger counter recessed	(*)
0x008A	Time Zone	
0x0090	[reserved]	-
0x0091	[reserved]	-
0x0092	Message rejected	Table 4.44
0x0093	Configuration Data CARMEN / Caromee	Table 4.45
0x0094	Meta Data-Message	see XTMT-specification
0x00FF	End of File	Table 4.46

Table 4.2: Message format

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

4.3 Marker-Message (ID 0x0000)

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

Table 4.3: Format of a Marker-Message

4.4 Serial Message (ID 0x0003)

The serial Data 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 4.4: Payload Format of a serial Message (ID0x03)

4.5 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 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 4.5: Format of an Ethernet Message

4.5.1 Ethernet-Message Protocol Type (ID 0,1,2,4,5,6)

Description	Length
Ethernet Data	n Bytes

Table 4.6: Format of GNLogger, RAW, UTF8, DLT_BMW, UDP-Server, Spy Mode and Eso Trace Protocol Type

4.5.2 Ethernet-Message Protocol Type DLT (ID 3)

An ethernet message with protocol type DLT contains four byte ECU-ID followed by one or more DLT messages.

Description	Length
ECU-ID	4 Bytes
DLT message ^{*)}	n Bytes

To iterate through the DLT messages the DLT message length must be read which is located in byte 2 and 3 (zero based index) of the DLT header (see document referenced below).

^{*)} DLT message corresponding to https://www.autosar.org/fileadmin/user_upload/standards/classic/4-0/AU-TOSAR_SWS_DiagnosticLogAndTrace.pdf (Chapter 7.7.1)

4.5.3 Ethernet-Message Protocol Type MII Mode (ID 7)

→ do not use → replaced by ID 8

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
<i>Optional:</i> Padding bytes (0x00) for Quadlet alignment	m Bytes

Table 4.7: Format of MII Mode Protocol Type

4.5.4 Ethernet-Message Protocol Type EP_MII Mode (ID 8)

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
<i>Optional:</i> Padding bytes (0x00) for Quadlet alignment m = 1,2,3	m Bytes

Table 4.8: Format of EP_MII Mode Protocol Type

4.6 LIN-Message (ID 0x0006)

4.6.1 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 4.9: 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.

- 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 μ s. 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.

4.6.2 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 4.10: Format of a LIN Wake Up Message

4.6.3 LIN Status Message

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

Table 4.11: Format of a LIN Status Message

4.7 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 4.12: Format of an ECL message

4.7.1 ECL EWU Message (ECL type 0x06)

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

Table 4.13: Format of an ECL EWU Message

4.7.2 ECL STWU Message (ECL type 0x07)

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

Table 4.14: Format of an ECL STWU message

4.7.3 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 (μ s)	4 Bytes

Table 4.15: Format of an ECL STP message

4.7.4 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 (μ s). If both E_n and O_n remain high the transmission time is marked invalid (0xFFFFFFFF).	4 Bytes

Table 4.16: Format of an ECL STR message

4.7.5 ECL Undefined Pulse message (ECL type 0x0A)

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

Table 4.17: Format of an ECL Undefined Pulse message

4.8 CAN-Message (ID 0x000B)

Description	Length
Channel number	1 Byte
CAN-Message type 0x00 Standard 0x01 Error frame 0x02 Transmitted frame 0x03 Remote Transmission Request	1 Byte
CAN-FD Status (Bit 7..6) Bit 7 ESI (error state indicator) Bit 6 BRS (bit rate switch) Reserved (Bit 5..4) CAN-Status (Bit 3..0) 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
Date Length Code (DLC) in Bytes 0..64 = 0..64 Bytes	1 Byte
Extended ID Flag (2.0b Standard) (Bit 31) CAN-FD Flag (Bit 30) Reserved (Bit 29) CAN ID (Bit 28..0)	4 Bytes
CAN Data bytes	N Bytes

Table 4.18: Payload Format of a CAN-Message

4.9 Container Message (ID 0x000C)

Description	Length
Message ID of message in payload	2 Byte
Number of messages in payload	1 Byte
Compression (Bit 0) 0x00 payload is not compressed 0x01 payload is compressed	1 Byte
Timestamp of last message in payload	8 Bytes
Size of payload (uncompressed) [bytes]	2 Bytes
Size of channelArray (N)	1 Byte
Array of channel numbers of messages in payload (channel #255 to be ignored)	N Bytes
payload	M Bytes

Table 4.19: Payload Format of a Container-Message

Currently compression is always deactivated (byte = 0x00). The timestamp of the first message in payload is identical to the timestamp of the container message (see header). The payload consists of a sequence of message with the same message id and with a leading sync word (0x55AA).

4.10 TTY-Message (ID 0x000D)

Description	Length
Channel number	1 Byte
Protocol 0x00 none 0x01 QXDM	1 Byte
Number of Data bytes	2 Bytes
TTY Data bytes	N Bytes

Table 4.20: Payload Format of a TTY-Message

4.11 MII-Message (ID 0x000E)

Description	Length
Nanosecond part of timestamp (0.999)	2 Bytes
Channel number	1 Byte
Subtype 0x00 MII Standard	1 Byte
Direction Bit 0: 0 = receive, 1 = transmit Bit 1..7 reserved	1 Byte
Link quality	1 Byte
Reserved	3 Bytes
Status Bit 0: 0 = OK, 1 = Error Bit 1: 0 = Preamble OK 1 = Preamble shortened/elongated Bit 2: 0 = Normal Frame 1 = Raw Frame (SFD not found) Bit 3: 0 = Frame Length OK 1 = Frame Length <64 Bytes Bit 4..7 reserved	1 Byte
Number of bytes in Ethernet-Frame	2 Bytes
Ethernet-Frame	N Bytes

Table 4.21: Payload Format of a MII-Message

4.12 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 4.22: Payload Format of MOST150-Message

4.12.1 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
CRC ACK	1 Byte

Table 4.23: Data Format of MOST150 Control-Channel-Message

The packet length-1 indicates how many bytes follow.

4.12.2 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
Preemptive Acknowledge (PACK)	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 4.24: 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: $(2 + 1 + 1 + 2 + 2 + 1 = 9)$

4.12.3 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 4.25: 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)

4.12.4 MOST150 Streaming message (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 to the number of active streaming channels)	n*2 Bytes
padding bytes for quadlet alignment (always 0xff)	0 or 2 Bytes
streaming data	m Bytes

Table 4.26: Format der Daten eine MOST-Streaming message

4.12.5 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 4.27: Data Format of a MOST (De-)Allocation Event

4.12.6 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 4.28: Data Format of a MOST-Network-Status-Message

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

The MDP message (see 4.12.2) does not contain all the data provided. It ends after the header in the specified length (see Table 24).

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

The MEP message (see 4.12.3) does not contain all the data provided. It ends after the header in specified length (see Table 25).

4.13 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 4.1)

Description	Length
Port-Index	2 Bytes
Error Bit 3 – invalid analog value Bit 2..0 - reserved	4 Bit
Direction 0x00 – unknown 0x01 – in 0x02 – out	4 Bit
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 0x02 VOLT 0x02 AMPERE	1 Byte

Table 4.29: Format of Analog-Signal-Value

4.14 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 4.1).

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 4.30: Format of GPIO-Signal-Value

4.15 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 4.31: Payload Format of a MOST-Message

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

4.15.1 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 4.32: Data Format of a MOST-Control-Channel-Message

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

4.15.2 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 4.33: Data Format of a MOST Data Packet (MDP)

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

4.15.3 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 4.34: Data Format of a MOST-Network-Status-Message

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

The MDP message (see 4.15.2) does not contain all the data provided. It ends after the header in the specified length (see Table 4.1).

4.16 FlexRay-Extended-Message (ID 0x0015)

Description	Length
Message Type <i>Flexray Symbol:</i> 0x00 WUS 0x01 Reserved 0x02 Reserved 0x03 undefined Low Pulse (neither Symbol or TSS) 0x04 CAS 0x05 MTS <i>Flexray Message:</i> 0x10 = static Message 0x11 = dynamic Message 0x12 = invalid Message (FES/DTS false) <i>Message structure at 0x12:</i> 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 4.35: Data Format of a FlexRay-Extended-Message

4.17 PHY-Status-Message (ID 0x001F)

Description	Length
Bustype 0x00 - BUSID_UNKNOWN 0x13 - BUSID_MII	1 Byte
Channel number	1 Byte
Phytype 0x00 – unknown 0x01 – TJA1110 0x02 – KSZ9031RNX 0x03 – BCM54810	2 Bytes
Array of pairs: register number (2Bytes) + register value (2Bytes)	N x 4Bytes

Table 4.36: Payload Format of a PHY-Status-Message

The number of register number – register value pairs is not fixed and depends on PHY type. The order of the register number-value pairs is also not fixed. For detailed information, please see the data sheet of the specific PHY.

4.17.1 Register Number – Value Pairs (general)

Description	Length
Register number: 2 Bytes [15:0] 0x1 (Basis Status Register)	2+2 Bytes
Register value: 2 Bytes [15:0] Bit 2: Link Status 0: Link down 1: Link up	

Table 4.37: Payload Format of a PHY-Status-Message

4.17.2 Register Number – Value Pairs (TJA1100 specific)

<p>For detailed information, please see TJA1100 100BASE-T1 PHY for Automotive Ethernet Product Data Sheet, Rev.3 – 23 May 2017, NXP Semiconductors.</p> <p>Example (Chapter 6.12, Table 21)</p> <p>Register number: 2 Bytes [15:0] 0x21 (Interrupt Status Register)</p> <p>Register value: 2 Bytes [15:0]</p> <p>Bit 0: SLEEP_ABORT (transition from sleep request back to normal mode)</p> <p>Bit 1: TEMP_ERR (Overtemperature error detected)</p> <p>Bit 2: UV_RECOVERY (Undervoltage recovery detected)</p> <p>Bit 3: UV_ERR (undervoltage detected)</p> <p>Bit 4: reserved</p> <p>Bit 5: CONTROL_ERR (SMI Control Error detected)</p> <p>Bit 6: SQI_WARNING (SQI value below warning limit and bit LINK_UP is set)</p> <p>Bit 7: TRAINING_FAILED (training phase failure detected)</p> <p>Bit 8: SYM_ERR (symbol error detected)</p> <p>Bit 9: LINK_STATUS_UP (link status bit LINK_UP changed from 'link fail' to 'link ok')</p> <p>Bit 10: LINK_STATUS_FAIL (link status bit LINK_UP changed from 'link ok' to 'link fail')</p> <p>Bit 11: PHY_INIT_FAIL (PHY initialization error detected)</p> <p>Bit 12: reserved</p> <p>Bit 13: reserved</p> <p>Bit 14: WAKEUP (local or remote wake-up detected)</p> <p>Bit 15: PWON (power-on detected)</p>	2+2 Bytes
---	-----------

Table 4.38: Payload Format of a PHY-Status-Message

4.18 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 4.39: Payload Format of a System-Message

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

Description	Length
Data configuration as String	n Bytes

Table 4.40: Format for specifying configuration data

4.20 Temperature-Message (ID 0x0087)

Description	Length
Temperature in °C	2 Bytes

Table 4.41: Format Temperature-Message

4.21 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 4.42: Format of Start-Time-Message

4.22 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 4.43: Format of Timezone-Message

4.23 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 4.44: 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 4.21). Timestamps are generally stored as UTC (Microseconds since 01.01.1970).

4.24 Extended message for rejected Bus-Messages (ID 0x0095)

The message is sent when the data bus message is logged regularly and replaces the prior message with ID 0x92 (see 4.23).

Description	Length
Bus-ID of the affected messages 0x00 - BUSID_UNKNOWN 0x01 - BUSID_CAN 0x02 - BUSID_SERIAL 0x03 - BUSID_MOST25 0x04 - BUSID_FLEXRAY 0x05 - BUSID_LIN 0x06 - BUSID_ETHERNET 0x07 - BUSID_CAMERA 0x08 - BUSID_MOST150 0x09 - BUSID_MOST50 0x0a - BUSID_ANALOG 0x0b - BUSID_CCPXCP 0x0c - BUSID_GPIO 0x0d - BUSID_GPS 0x0e - BUSID_ECL 0x0f - BUSID_COMPLEXFILTER 0x10 - BUSID_TTY 0x11 - BUSID_TSA	1 Byte
Channel index of the affected messages	1 Byte
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 4.45: 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 4.21). Timestamps are generally stored as UTC (Microseconds since 01.01.1970).

4.25 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 4.46: Format of a Channel-Configuration for CARMEN / Caromee

4.26 EndOfFile-Message (ID 0x00FF)

Each Trace-File ends with an EOF Message.

Description	Length
Reserved	32 Bits

Table 4.47: Format – End of File (ID 0x00FF)

5 Support

If problems occur with the hard- or software from Magna Telemotive GmbH, please take following steps:

- Read the User Manual
- Please check if you are using an up-to-date software
- Please check if all cables are correctly attached to the data logger
- If you are able to establish a connection to the data logger, run the program "*Bug Reporter*". This program creates a zip file, which you should please put into a ticket into our [OTRS Ticket system](#)
- Contact Customer Support at TMO.productsupport@magna.com (+49 89 357186-518)

5.1 Service Center

In our Service Center you will find the newest firmware versions and the latest version of the System Client as well as older versions for download. In addition, we offer detailed documentations and specifications for our current products.

There are two ways to reach the service center:

1. Using the current link:

<https://sc.telemotive.de/4/index.php?id=154&L=1>

2. Go to the Telemotive homepage and use the login link top right.

<http://www.telemotive.de>

Note: *If you do not have an account for our service center and OTRS ticket system, please send a mail to TMO.Produktsupport@magna.com and we will generate an account for you.*

5.2 OTRS Ticket system

With the login data for the service center you have access to our OTRS-Ticket system too. Every email sent to TMO.productsupport@magna.com generates automatically a ticket and can therefore be forwarded to the responsible person promptly.

At <https://produktsupport.telemotive.de> the status of your tickets can be checked fast and easily.

You can log in using your access data above. Creation of new tickets is also possible as upload an bugreport. The most important steps are described in a manual that can be found in the upper right corner of the website or under this link directly:

https://sc.telemotive.de/4/uploads/media/OTRS_Kurzanleitung.pdf

Note: *If you want to upload more than 20 MB please create a ticket first and upload the file in a second step without the limitation.*

[Index](#)

6 List of tables

Table 3.1: TMT file structure.....	9
Table 4.1: General construct of a Message	11
Table 4.2: Message format.....	12
Table 4.3: Format of a Marker-Message.....	13
Table 4.4: Payload Format of a serial Message (ID0x03)	13
Table 4.5: Format of an Ethernet Message.....	13
Table 4.6: Format of GNLogger, RAW, UTF8, DLT_BMW, UDP-Server, Spy Mode and Eso Trace Protocol Type	14
Table 4.7: Format of MII Mode Protocol Type.....	14
Table 4.8: Format of EP_MII Mode Protocol Type	15
Table 4.9: Format of a LIN Data Message	16
Table 4.10: Format of a LIN Wake Up Message	17
Table 4.11: Format of a LIN Status Message	17
Table 4.12: Format of an ECL message	19
Table 4.13: Format of an ECL EWU Message.....	19
Table 4.14: Format of an ECL STWU message.....	19
Table 4.15: Format of an ECL STP message	19
Table 4.16: Format of an ECL STR message	20
Table 4.17: Format of an ECL Undefined Pulse message	20
Table 4.18: Payload Format of a CAN-Message.....	21
Table 4.19: Payload Format of a Container-Message.....	22
Table 4.20: Payload Format of a TTY-Message	22
Table 4.21: Payload Format of a MII-Message	23
Table 4.22: Payload Format of MOST150-Message.....	23
Table 4.23: Data Format of MOST150 Control-Channel-Message	24
Table 4.24: Data Format of MOST Data Packet (MDP)	24
Table 4.25: Format of MOST Ethernet Packet (MEP)	24
Table 4.26: Format der Daten eine MOST-Streaming message	25
Table 4.27: Data Format of a MOST (De-)Allocation Event	25
Table 4.28: Data Format of a MOST-Network-Status-Message.....	26
Table 4.29: Format of Analog-Signal-Value	26

Table 4.30: Format of GPIO-Signal-Value	28
Table 4.31: Payload Format of a MOST-Message.....	28
Table 4.32: Data Format of a MOST-Control-Channel-Message	29
Table 4.33: Data Format of a MOST Data Packet (MDP)	29
Table 4.34: Data Format of a MOST-Network-Status-Message.....	29
Table 4.35: Data Format of a FlexRay-Extended-Message	31
Table 4.36: Payload Format of a PHY-Status-Message.....	32
Table 4.37: Payload Format of a PHY-Status-Message.....	32
Table 4.38: Payload Format of a PHY-Status-Message.....	33
Table 4.39: Payload Format of a System-Message	33
Table 4.40: Format for specifying configuration data	33
Table 4.41: Format Temperature-Message	34
Table 4.42: Format of Start-Time-Message	34
Table 4.43: Format of Timezone-Message	34
Table 4.44: Message Format for rejected Messages	34
Table 4.45: Message Format for rejected Messages	36
Table 4.46: Format of a Channel-Configuration for CARMEN / Caromee	37
Table 4.47: Format – End of File (ID 0x00FF)	37
Table 8.1: Version history.....	45

7 Open Source

License terms for open source software used in the firmware or System Client can be found by following this link:

⇒ https://sc.telemotive.de/4/uploads/media/BLUEPIRAT_3rd_Party_Software_Licenses.pdf

8 Version history

Version /Date	Chapter	Change	Author	Versions relevance (Position)			Doc
				1	2	3	
3.1.0 11.10.2011		Release Version	MVP				
3.2.0	4.8	CAN-Channel extended to 8 bit	Mvp				
3.3.0 17.02.12	4.16	FlexRay-Message (ID 0x0005) replaced by FlexRay-Extended-Message (ID 0x0015)	MKi		x		
3.3.0		Release					
28.03.12	4.14	GPIO Message Text updated	GKI				x
18.05.12	4.4	Extension of the serial Message to DLC	MVP				x
23.05.12	4.18	Correction of the data field sizes	MVP				x
04.06.12	4.12	MOST150 network status message adapted to SMSC Spec	RoSc				x
18.07.12	4.2	ID corrected for FlexRay data (0x0015)	MKi				x
18.07.12	4	Subsection on FlexRay data moves to the ID of the corresponding position	MKi				x
23.07.12	4.2	Message "Data record stopped" (ID 0x0083), "Data record continued" (ID 0x0084), "Start 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	4.21	Description of the data in the start time of message according to the use in blue PiraT 2	MKi				x
24.07.12	4	MOST50 Message removed MOST25 Streaming Message removed ECL Message removed Added Discarded message for FlexRay Cleanup	RoSc				x
24.07.12		All types away, do not appear in the trace file.	MKi				x
26.07.12	3.1	New Chapter 1.1	mvp				x
10.08.12	4.1 4.7	Byte order corrected in Table 2 DLC corrected in Table 8	RoSc				x

Version /Date	Chapter	Change	Author	Versions relevance (Position)			Doc
				1	2	3	
22.08.12	4.12	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	4.5	New Ethernet protocols included	JLi				x
07.03.13	4.17	Timezone Message added	JLi		x		x
14.03.2013		ECL-Message added	mvp		x		
29.04.2013	4.12	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	4.1	Description of Message Length field in Message Header corrected	mvp				x
04.12.2013	4.5	Ethernet Protocol type MII Mode field added	MaHa		x		
28.01.2014	4.9.4	MOST150 Streaming Message added	RoSc				
	4.9.5	MOST150 (De-)Allocation Message added					
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					
14.02.2014	4.10	Fixed Table 27, Analog-Message (Volt = 0x02)	mvp				x
17.12.2014		ContainerMessage, TTYMessage added	JLi		x		
27.04.2015	4.9	ContainerMessage updated	RoSc				x
04.09.2015		Release 3.7.0 FW/Client 2.1.1					
22.7.2016	4.8	CAN-FD Message added	JLi		x		

Version /Date	Chapter	Change	Author	Versions relevance (Position)			Doc
				1	2	3	
18.10.2016	4.23	LostExtMessage added	mvp				x
13.12.2016		Release 3.8.0 FW/Client 3.0.1					
08.06.2017	4.11, 4.17	MII Message, PhyStatus Message added	JLi		x		
09.02.2018	4.8	Updated DLC Field description	DanS				x
08.06.2017		Release 3.9.1 FW/Client 3.4.x					
23.04.2018	4.0	Chapter "General Message Structure" Field Reserved changed.	GKI				
17.10.2019	4.17	PhyStatus Message updated	TPf				
19.02.2020	4.1	Chapter "General Message Structure" Field Reserved changed.	StSc				
29.06.2020	4.11	Status bit of MII-Message (ID 0x000E) expanded with Preamble, SFD and frame length status	RoSc				x
15.07.2020	4.13	Analog-Message direction field splitted in two 4 bit fields: Error and Direction	mvp			x	
04.08.2020		TMT-Version 3.9.3 – Dokumentenversion 1.1 FW 4.2.7.40					
15.10.2020	4.5.2	Chapter "Ethernet-Message Protocol Type DLT introduced	mvp				X
15.10.2020		TMT-Version 3.9.3 – Dokumentenversion 1.2					

Table 8.1: Version history

9 Contact



DRIVING **EXCELLENCE.**
INSPIRING **INNOVATION.**

MAGNA Telemotive GmbH

Office München
Frankfurter Ring 115a
80807 München / Germany

Tel.: +49 89 357186-0
Fax.: +49 89 357186-520
E-Mail: TMO.info@magna.com
Web: www.telemotive.de

Sales

Tel.: +49 89 357186-550
Fax.: +49 89 357186-520
E-Mail: TMO.Sales@magna.com

Support

Tel.: +49 89 357186-518

E-Mail: TMO.productsupport@magna.com

ServiceCenter: <https://sc.telemotive.de/bluepirat>

© by MAGNA Telemotive GmbH, 2019

Subject to errors and to technical changes as part of product improvement.