

TECMP

Technically Enhanced Capture Module Protocol

USER MANUAL

December 2021

CONTENT

1	GENERAL INFORMATION	5
1.1	Introduction	5
1.2	Purpose of the document	5
1.3	Available resources	5
1.4	TECMP functions and scope.	6
1.5	Abbreviations	6
2	Feature Specification	8
2.1	Foreword	8
2.2	Capturing IVN data	8
2.3	Replaying IVN data	9
2.4	Data format	9
2.4.1	TECMP Header.....	10
2.4.2	Frame overview.....	14
2.4.3	Recording specification (“Logging Stream” Message Type)	14
2.4.3.1	TECMP CAN Data (Data Type = 0x2)	14
(a)	Data Flags	14
(b)	Source Data.....	15
2.4.3.2	TECMP CAN-FD Data (Data Type = 0x3)	15
(a)	Data Flags	15
(b)	Source Data.....	16
2.4.3.3	TECMP LIN Data (Data Type = 0x4)	17
(a)	Data Flags	17
(b)	Source Data.....	17
2.4.3.4	TECMP Flexray (Data Type = 0x8).....	17
(a)	Data Flags	17
(b)	Source Data.....	18
2.4.3.5	TECMP UART / RS232 ASCII (Data Type = 0x10)	18
(a)	Data Flags	18
(b)	Source Data.....	19
2.4.3.6	TECMP Analog (Data Type = 0x20).....	19
(a)	Data Flags	19
(b)	Source Data.....	20

2.4.3.7	TECMP Analog Extended (Data Type = 0x22)	20
(a)	Data Flags	20
(b)	Source Data	20
2.4.3.8	TECMP Ethernet Data (Data Type = x80)	22
(a)	Data Flags	22
(b)	Source Data	22
2.4.4	Replay specification ("Replay Data" Message Type)	23
2.4.4.1	TECMP CAN Tx (Data Type = 0xE002)	23
(a)	Data Flags	23
(b)	Source Data	23
2.4.4.2	TECMP CAN-FD Tx (Data Type = 0xE003)	24
(a)	Data Flags	24
(b)	Source Data	24
2.4.4.3	TECMP LIN Tx (Data Type = 0xE004)	24
(a)	Data Flags	24
(b)	Source Data	25
2.4.4.4	TECMP Flexray Tx (Data Type = 0xE008)	25
(a)	Data Flags	25
(b)	Source Data	25
2.4.4.5	TECMP GPIO Tx (Data Type = 0xE00A)	26
(a)	Data Flags	26
(b)	Source Data	26
(i)	Channel status	27
(ii)	Potentiometer	28
(iii)	Signal generator	28
(iv)	DC Load	29
2.4.5	Packetizing IVN messages	29
2.5	Status Message	29
2.5.1	Status Configuration message	30
2.5.2	Status Bus/IVN message	32
2.5.3	Status Capture Module message	34
2.6	Control message	36
2.6.1	CAN Replay fill level	37
2.7	TECMP Header fields summary	38

2.7.1	Message Type.....	38
2.7.2	Data Type	38
2.7.3	Device ID.....	38
2.7.4	Control Message ID	39
2.7.5	CM Type.....	40
2.7.6	Interface ID	40
3	Contact.....	41
4	Changelog.....	42

1 GENERAL INFORMATION

1.1 Introduction

The following protocol description targets the (CM) from Technica Engineering. The CM is currently available in five different optimized products for which dedicated user manuals are also available:

- CM Ethernet Combo
- CM CAN Combo
- CM LIN Combo
- CM 100 High
- CM 1000 High

CMs support the official TECMP EtherType (0x99FE). This EtherType is used as reference to transmit the IVN-data collected by the CM to an endpoint device connected via Ethernet. The device can be a computer, an Ethernet switch for network connection, or a data logger module. These devices are called "sinks" later in this document.

CMs are used as building blocks in cutting-edge distributed data logging networks made of one or more sinks aggregating the TECMP messages generated inside the measurement network. This logging network is scalable and is based on IT standards.

1.2 Purpose of the document

This document describes the functionality of the Technica Engineering Capture Module interface. The functions are accessible via the Technically Enhanced Capture Module Protocol (TECMP) published by Technica Engineering and described in this document.

This document defines the data format in which the IVN captured data are transmitted to data sink and how data can be replayed on an IVN. The position of metadata like timestamp and other information related to the source or direction of data are also covered in this document.

1.3 Available resources

Wireshark (3.4.0 or newer) provides a native TECMP dissection. An overview of the TECMP packet dissection inside Wireshark is provided in the next illustration.

No.	Time.abs	Time	Delta	Source	Destination	Length	Protocol	Info
334	2020-10-30 16:23:50.724337	12.496928	0.000243000	Technica_...	IPv4mcast_00	60	TECMP	TECMP Payload: CAN Data
335	2020-10-30 16:23:50.724579	12.497170	0.000242000	Technica_...	IPv4mcast_00	60	TECMP	TECMP Payload: CAN Data
336	2020-10-30 16:23:50.724821	12.497412	0.000242000	Technica_...	IPv4mcast_00	60	TECMP	TECMP Payload: CAN Data
337	2020-10-30 16:23:51.333371	13.105962	0.608550000	Technica_...	IPv4mcast_00	88	TECMP	TECMP Status Capture Module
338	2020-10-30 16:23:51.334858	13.107449	0.001487000	Technica_...	IPv4mcast_00	162	TECMP	TECMP Status Bus

<
> Frame 336: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: Technica_00:32:36 (38:2a:19:00:32:36), Dst: IPv4mcast_00 (01:00:5e:00:00:00)
▼ Technically Enhanced Capture Module Protocol
Capture Module ID: 0x0040 (CM CAN Combo 0 (Default))
Counter: 344
Version: 3
Message Type: Logging Stream (0x03)
Data Type: CAN Data (0x0002)
Reserved: 0x0000
> CM Flags: 0x000f, End of Segment, Start of Segment, Spy, Multi Frame
> Technically Enhanced Capture Module Protocol Payload (CAN Data)

Figure 1: Example of TECMP packet dissection inside Wireshark (CAN)

1.4 TECMP functions and scope.

The TECMP protocol is defined by the following keypoints:

- A specific data format.
- The support of IVN data capturing.
- The support of IVN data replaying.
- Specific status and control messages.
- A time synchronization mechanism.

An overview of a system setup comprising multiple CMs and a logger module is represented in Figure 2.

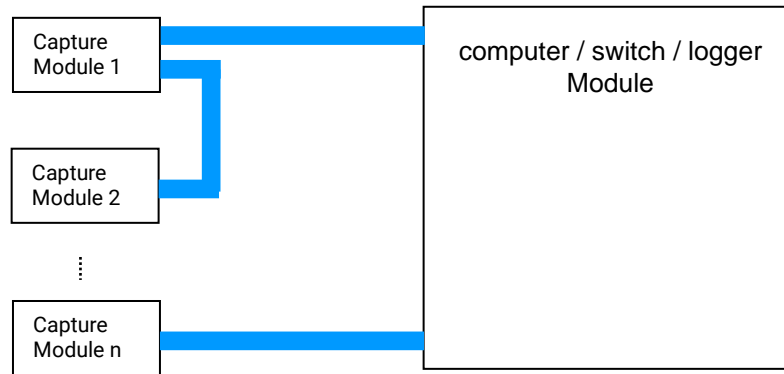


Figure 2: Overview of a system setup

The following TECMP functionalities are covered by this specification:

- Data transport from the CM to an Ethernet network sink (computer, logger...)
- Data transport to the CM from an Ethernet network sink.
- Remote monitoring and control.
- Time synchronization of the CMs: 802.1AS Automotive Profile (AVNU).

1.5 Abbreviations

Abbreviation	Definition / Explanation
CM	Capture Module
TECMP	Technically Enhanced Capture Module Protocol
CAN	Controller Area Network
EoS	End of Segment
SoS	Start of Segment
Eth	Ethernet
FR	FlexRay
HW	Hardware
LIN	Local Interconnect Network
PTP	Precision Time Protocol

Abbreviation	Definition / Explanation
Control Message	A method to trigger events. Events can be triggered by a CM or can be received by a CM. Some Control Message can be configured
Status Message	A dedicated message sent by the CM in direction of the Ethernet Capture interface (network, computer, logger)
IVN	In-Vehicle Network
FCS	Frame Check Sequence

Table 1: Abbreviations

2 Feature Specification

2.1 Foreword

The TECMP protocol enables the recording and replaying of data via a decentralized network based on Ethernet as physical layer. Figure 3 shows an overview of the physical and logical interfaces between the CM and a computer or another Ethernet network components like a datalogger or a switch. For details of the physical interfaces consult the user manual of the specific CM.

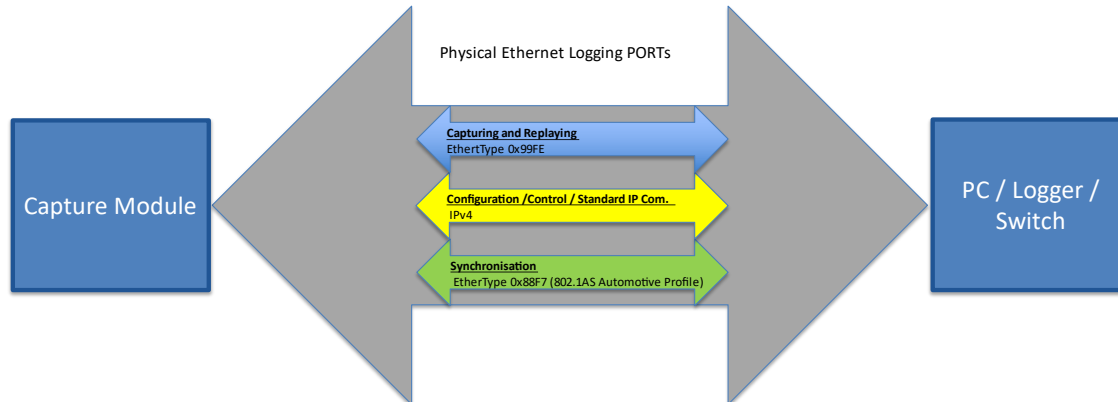


Figure 3: Logical/physical Interfaces

CMs may provide one or more additional interfaces which can be used to connect CMs with each other in a cascading structure. Based on this feature, the functionality of a CM can be extended with another CM.

2.2 Capturing IVN data

The main feature of the TECMP protocol is the recording of data of in-vehicle communication systems on a logging network based on Ethernet. Figure 4 presents a simple Logging network made of one CM and one desktop PC:

1. The IVN data (here originating from a CAN, CAN-FD and FlexRay buses) are captured by the CM
2. The captured data is transmitted to the sink in the form of one or more TECMP messages (a.k.a Logging Stream).

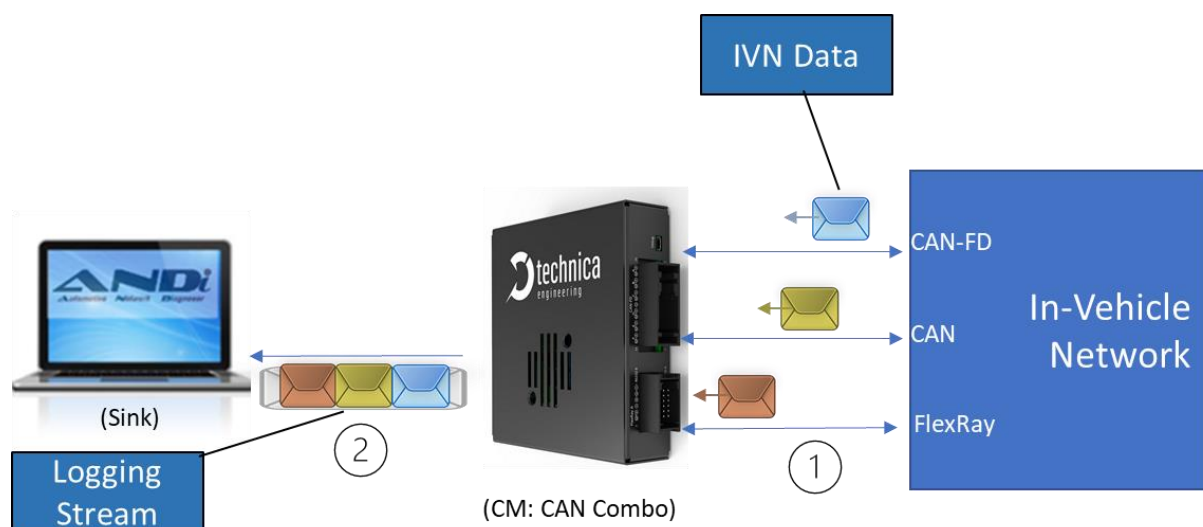


Figure 4: Capturing IVN Data

For more information on the capturing of data, consult the Chapter 2.4.3.

2.3 Replaying IVN data

Additionally, the TECMP protocol supports the transmission of messages into in-vehicle communication systems (Figure 5):

1. The data to transmit on the IVN is received by the CM in the form of one or more TECMP messages (a.k.a Replay Data).
2. The data to transmit on the IVN is sent out by the CM accordingly to the bus technology (Arbitration on CAN(-FD), scheduling policy on FlexRay...)

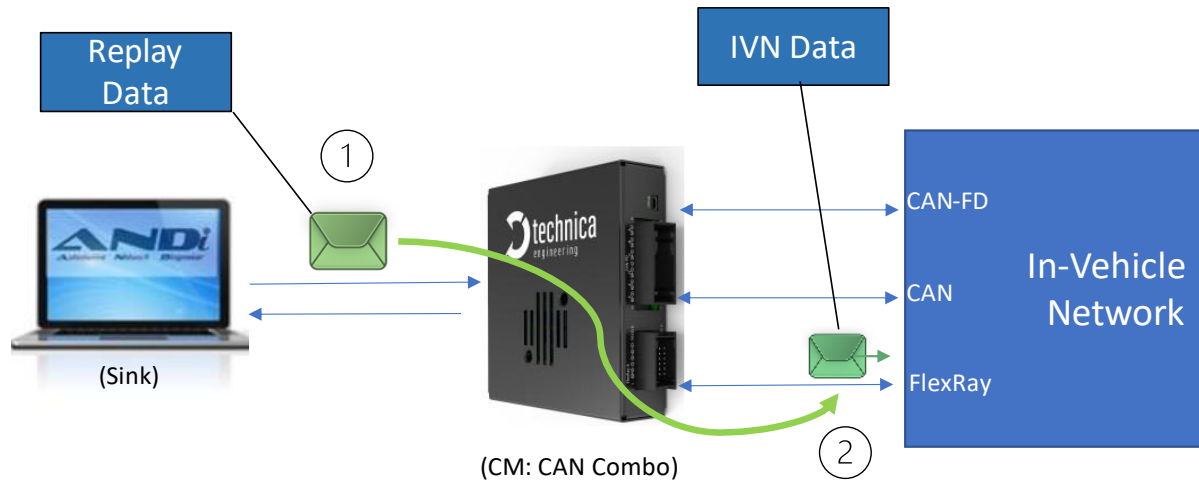


Figure 5: Replaying data onto the IVN

For more information on the replaying of data, consult the Chapter 2.4.4.

2.4 Data format

A TECMP frame is a standard Ethernet II transport layer frame (IEEE 802.3) with the EtherType 0x99FE and containing a TECMP Header. No additional transport protocol is required.

Data of non-Ethernet communication systems as well as of Ethernet-based systems are transported in Ethernet frames and identified using TECMP. Ethernet frames are transported without changes to them or their headers. All messages are left intact. This ensures that analysis tools can easily access the unchanged data, if support of TECMP was added or data was converted.

The data format is defined as follows:

PRE/SFD	DST MAC	SRC MAC	Outer 802.1Q Header (optional)	Inner 802.1Q Header (optional)	Ether Typ 0x99FE	TECMP Header	Source Data	FCS
8 Bytes	6 Bytes	6 Bytes	4 Bytes	4 Bytes	2 Bytes	28 Bytes	Up to n Bytes (n <= 1500 Bytes)	4 Bytes

Figure 6: Data Format of a TECMP message

Field Name	Description
PRE / SFD	Preamble and Start Frame Delimiter according to IEEE Standard 802.3.
DST MAC	The destination MAC address is by default assigned to the multicast address: 01: 00: 5E: 00: 00: 00. The value can be updated during configuration.
SRC MAC	The source MAC address carries the MAC address of the CM. The MAC addresses are unique for each delivered hardware.
VLAN 802.1Q	Optional VLAN fields are provided for possible future extensions.

	<ul style="list-style-type: none"> Outer 802.1Q header (optional): Tagging according to IEEE 802.1Q must be supported by all components. The value of the tag must be configurable in the CM. Inner 802.1Q header (optional): Tagging according to IEEE 802.1Q, must be supported by all components. The value of the tag must be configurable in the CM.
EtherType	0x99FE: TECMP - Identifies this packet as TECMP packet
TECMP header	see 2.4.1
Source Data	see 2.4.3
FSC - CRC	CRC checksum, calculated according to IEEE Standard 802.3

Table 2: TECMP message fields description

2.4.1 TECMP Header

NOTE: In general, all unspecified number types are UINT and encoded in big-endian. The header structure is identical for all TECMP messages exchanged between CM and sink. Figure 7 and Figure 8 show the position of the header compared to the IVN data position.

4 Bytes			
1	2	3	4
Device ID		Counter	
Version	Message Type	Data Type	
Reserved		CM Flags	
Interface ID			
Timestamp Bit 63..32			
Timestamp Bit 31..0			
Length		Data Flags	
Source Data			
...			

Figure 7: Overview of a vehicle data frame in a TECMP frame

4 Bytes			
1	2	3	4
Device ID		Counter	
Version	Message Type	Data Type	
Reserved		CM Flags	
Interface ID			
Timestamp Bit 63..32			
Timestamp Bit 31..0			
Length		Data Flags	
Source Data			
...			
Interface ID			
Timestamp Bit 63..32			
Timestamp Bit 31..0			
Length		Data Flags	
Source Data			
...			

Figure 8: Overview of multiple vehicle data frames packetized in a TECMP frame

Note: Packing multiple vehicle frames is only done for the same Data Type.

The structure and description of the TECMP Header is presented in Table 3. A distinction between the two cases capturing data/replaying data is made whenever a field of the header has a different description in both cases.

Field Name	Data Type	Description																		
Device ID	UINT16	<p>The first upper Byte is reserved : 0x00.</p> <p>The second Byte defines the CM unique ID. This ID can be used to identify different Modules if they are used in the same network (see Table 25). Up to 256 different devices can be uniquely identified inside the network.</p> <p>When capturing data (Message Type = "Logging Stream"): This field identifies the CM from which the TECMP frame was sent: its source.</p> <p>When replaying data (Message Type = "Replay data"): Unused. This field should be set to: "0x0000" for every message.</p>																		
Counter	UINT16	<p>When capturing data (Message Type = "Logging Stream"): This field provides a mean to check the non interruption (loss) of messages sent by the CM. If necessary, the exact number of lost messages can be computed. The counter value is incremented monotonically for each TECMP frame sent from a CM. The counter starts at 0x0001 when the first TECMP frame after power-up is transmitted and increases by 1 at every new transmission. The counter restarts from 0x0000 when reaching the maximum value 0xFFFF. If a CM has several physical interfaces to the sink (Ethernet links), a separate counter is implemented on each interface. The value of this field is only valid for a given Device ID or a given HW interface of the Device ID.</p> <p>When replaying data (Message Type = "Replay data"): This field should be set to: "0x0000" for every message. Received requests are no counted by the CM.</p>																		
Version	UINT8	Version of the TECMP protocol used. This value is set to 3.																		
Message Type	UINT8	Description of the payload function: whether status, control, logging stream or replay data. See Table 23.																		
Data Type	UINT16	Description of the Type of Source Data (e.g., CAN, LIN, Flexray, Ethernet, Voltage, Current, ...). See Table 24.																		
Reserved	UINT16	Reserved for future extension of the protocol: 0x00.																		
CM Flags	UINT16	<p>When capturing data (Message Type = "Logging Stream"): This field provides additional information on the CM.</p> <table> <tr> <th>Offset</th><th>Name</th><th>Description</th></tr> <tr> <td>0</td><td>EoS (End of Segment)</td><td>Segmentation indication for a long information split over multiple TECMP frame.</td></tr> <tr> <td>1</td><td>SoS (Start of Segment)</td><td> <ul style="list-style-type: none"> SoS = 1, EoS = 0: Start of a segmented message. SoS = 0, EoS = 0: Within a segmented message. SoS = 0, EoS = 1: End of a segmented message. SoS = 1, EoS = 1: Unsegmented message. </td></tr> <tr> <td>2</td><td>SPY</td><td>Trigger additional actions on the sink (sink-dependent). Standard value: '1' (none).</td></tr> <tr> <td>3</td><td>Multi-Frame</td><td>'1': Indicates that the CM can send multiple Interface IDs in a TECMP frame. See Figure 13 in Chapter 2.4.5.</td></tr> <tr> <td>14 - 4</td><td>Reserved</td><td>Reserved for future use.</td></tr> </table>	Offset	Name	Description	0	EoS (End of Segment)	Segmentation indication for a long information split over multiple TECMP frame.	1	SoS (Start of Segment)	<ul style="list-style-type: none"> SoS = 1, EoS = 0: Start of a segmented message. SoS = 0, EoS = 0: Within a segmented message. SoS = 0, EoS = 1: End of a segmented message. SoS = 1, EoS = 1: Unsegmented message. 	2	SPY	Trigger additional actions on the sink (sink-dependent). Standard value: '1' (none).	3	Multi-Frame	'1': Indicates that the CM can send multiple Interface IDs in a TECMP frame. See Figure 13 in Chapter 2.4.5.	14 - 4	Reserved	Reserved for future use.
Offset	Name	Description																		
0	EoS (End of Segment)	Segmentation indication for a long information split over multiple TECMP frame.																		
1	SoS (Start of Segment)	<ul style="list-style-type: none"> SoS = 1, EoS = 0: Start of a segmented message. SoS = 0, EoS = 0: Within a segmented message. SoS = 0, EoS = 1: End of a segmented message. SoS = 1, EoS = 1: Unsegmented message. 																		
2	SPY	Trigger additional actions on the sink (sink-dependent). Standard value: '1' (none).																		
3	Multi-Frame	'1': Indicates that the CM can send multiple Interface IDs in a TECMP frame. See Figure 13 in Chapter 2.4.5.																		
14 - 4	Reserved	Reserved for future use.																		

Field Name	Data Type	Description		
		15	CM overflow	The Capture Module has lost data during transmission to the Sink.
		When replaying data (Message Type = "Replay data"): Reserved: 0x0000.		
Interface ID	UINT32	ID that uniquely identifies the log data / bus / link on the IVN. Status and Control messages have dedicated Interface ID value (Table 28) to ease filtering at this level.		
Timestamp	UINT64	This field provides the 62-bits timestamp information reference of the TECMP frame, as well as a 2-bits timestamp flag section.		
		Length (bits)	Description	
		2	Timestamp flag section.	
		62	Timestamp information.	
		Offset	Name	Description
		61 - 0	Timestamp	<p>The time since Thursday, January 1, 1970, 00:00 UTC. The resolution is 1ns.</p> <p>When capturing data (Message Type = "Logging Stream"): This is the time reference of the captured data (when the frame was generated by the CM). The value 0x0 is not allowed.</p> <p>When replaying data (Message Type = "Replay data"): <ul style="list-style-type: none"> • 0x0: This enables sending frames instantly (also bursts possible). Frames are transmitted as soon as possible • >0x0 and > CM Wall Clock: This enables transmitting frames with respect to delta times and relative to CM wall clock. • >0x0 and ≤ CM Wall Clock: Frames are transmitted as soon as possible. </p>
		62	Timestamp Recalculation flag	<p>When capturing data (Message Type = "Logging Stream"):</p> <ul style="list-style-type: none"> • '1': the timestamp value (bits 61..0) was recalculated by the CM before transmission. This happens if the PTP reference time is received by the CM only after it started recording data on the corresponding IVN/Bus. • '0': no timestamp recalculation took place. The PTP reference time was received before the corresponding IVN/Bus recording started. <p>When replaying data (Message Type = "Replay data"): Reserved: '0'</p>
		63	Synchronization flag	When capturing data (Message Type = "Logging Stream"):

Field Name	Data Type	Description																			
			<ul style="list-style-type: none">• '1': lost synchronization with the time master. This bit is set if a time slave does not receive a 1AS packet from the master for more than 250 ms.• '0': ongoing synchronization with the time master. <p>When replaying data (Message Type = "Replay data"): Reserved: '0'</p>																		
Length	UINT16	Length of the Source Data field. The length of the Length and Data Flags fields are not included. Any additional 0-padding bit/Byte added at the end of the TECMP message is not part of the TECMP message itself. It is consequently not taken into account by the Length field.																			
Data Flags	16 Bitfield	<p>This field provides additional information regarding the recording of the source network when capturing data (Message Type = "Logging Stream") or regarding the transmission request when replaying data (Message Type = "Replay Data"). The Data Flags refer to the messages transmitted on the IVN (Ethernet, FlexRay, CAN, CAN-FD, LIN...). This field is split into an interface-specific flag part (described separately in every interface chapter) and a global flag part (described below) which is the same for all interfaces.</p> <table><tr><th>Length (bits)</th><th>Description</th></tr><tr><td>14</td><td>Interface-specific flags.</td></tr><tr><td>2</td><td>Global flags.</td></tr></table> <table><tr><th>Offset</th><th>Name</th><th>Description</th></tr><tr><td>13 - 0</td><td>Interface specific</td><td>Check the corresponding Data Type Interface section (from Chapter 2.4.3.1 to 2.4.4.5).</td></tr><tr><td>14</td><td>Tx</td><td><p>When capturing data (Message Type = "Logging Stream"): This bit indicates that the frame logged on the interface was transmitted by the CM itself. This bit will not be set in Status Messages as this bit is not related to the transmission to the sink but on the logged interface.</p><p>Only supported for Bus/IVN interfaces (neither GPIO nor Analog for instance).</p><p>When replaying data (Message Type = "Replay data"): This bit is unused: '0'.</p></td></tr><tr><td>15</td><td>Overflow</td><td><p>When capturing data (Message Type = "Logging Stream"): This bit indicates that one or more messages on the active IVN/Bus were lost during recording.</p><p>When replaying data (Message Type = "Replay data"): This bit is reserved: '0'.</p></td></tr></table>		Length (bits)	Description	14	Interface-specific flags.	2	Global flags.	Offset	Name	Description	13 - 0	Interface specific	Check the corresponding Data Type Interface section (from Chapter 2.4.3.1 to 2.4.4.5).	14	Tx	<p>When capturing data (Message Type = "Logging Stream"): This bit indicates that the frame logged on the interface was transmitted by the CM itself. This bit will not be set in Status Messages as this bit is not related to the transmission to the sink but on the logged interface.</p> <p>Only supported for Bus/IVN interfaces (neither GPIO nor Analog for instance).</p> <p>When replaying data (Message Type = "Replay data"): This bit is unused: '0'.</p>	15	Overflow	<p>When capturing data (Message Type = "Logging Stream"): This bit indicates that one or more messages on the active IVN/Bus were lost during recording.</p> <p>When replaying data (Message Type = "Replay data"): This bit is reserved: '0'.</p>
Length (bits)	Description																				
14	Interface-specific flags.																				
2	Global flags.																				
Offset	Name	Description																			
13 - 0	Interface specific	Check the corresponding Data Type Interface section (from Chapter 2.4.3.1 to 2.4.4.5).																			
14	Tx	<p>When capturing data (Message Type = "Logging Stream"): This bit indicates that the frame logged on the interface was transmitted by the CM itself. This bit will not be set in Status Messages as this bit is not related to the transmission to the sink but on the logged interface.</p> <p>Only supported for Bus/IVN interfaces (neither GPIO nor Analog for instance).</p> <p>When replaying data (Message Type = "Replay data"): This bit is unused: '0'.</p>																			
15	Overflow	<p>When capturing data (Message Type = "Logging Stream"): This bit indicates that one or more messages on the active IVN/Bus were lost during recording.</p> <p>When replaying data (Message Type = "Replay data"): This bit is reserved: '0'.</p>																			
Source Data	See 2.4.3 / 2.4.4	Data transferred from the vehicle IVN when capturing data (Message Type = "Logging Stream") or to the vehicle IVN when replaying data (Message Type = "Replay Data")..																			

Table 3: Structure and description of the TECMP Header

2.4.2 Frame overview

Figure 9 represents the overall frame sent towards the sink (between FCS and CRC).

4 Bytes			
1	2	3	4
DST MAC (MSB)			
DST MAC (LSB)		SRC MAC (MSB)	
SRC MAC (LSB)			
Outer VLAN (Optional)			
Inner VLAN (Optional)			
EtherType		Device ID	
Counter		Version	Message Type
Data Type		Reserved	
CM Flags		Interface ID (MSB)	
Interface ID (LSB)		Timestamp (MSB)	
Timestamp			
Timestamp (LSB)		Length	
Data Flags		Source Data	
...			

Figure 9: TECMP frame overview

2.4.3 Recording specification (“Logging Stream” Message Type)

In order to simplify the data processing and to transmit data as efficiently as possible, data are transferred in the same coding as in the “original” vehicle's internal buses (IVN). The recorded data are marked with a corresponding timestamp and transferred to the other sinks (e.g. data logger) or to the logging network. The coding details of the data are specified in the next chapters.

For efficient transmission, multiple recorded messages can be aggregated into a larger packet (see 2.4.5). The maximum payload length is limited to 1500 Bytes.

2.4.3.1 TECMP CAN Data (Data Type = 0x2) (a) Data Flags

The interface-specific flags provide information regarding the CAN bus and reception condition.

Categorie	Global flags		Interface-specific flags			
Bit position	15	14	13	12 - 8	7	6
Description	Overflow	Tx	CRC error	RESERVED	EOF error	ACK_DEL error

Categorie	Interface-specific flags					
Bit position	5	4	3	2	1	0
Description	CRC_DEL error	BIT_STUFF error	ERR	RESERVED	RTR	ACK

- ACK:
 - '1': there was an acknowledgement during the reception of the CAN frame.
 - '0': there was no acknowledgement during the reception of the CAN frame.
- RTR:
 - '1': the CAN frame is a remote frame (frame that requests the transmission of a dedicated data Frame). The LENGTH field will be 0x00, and there is no payload.
 - '0': the CAN frame is not a remote frame but a data frame.

- **ERR:**
 - '1': an "Error Frame" was received. In this case no payload is transported and the Payload length will be set to 0.
 - '0': no "Error Frame" was received.
- **BIT_STUFF error:**
 - '1': a bit stuff error was detected. This is for instance the case if an "Error Frame" is received.
 - '0': no bit stuff error was detected..
- **CRC_DEL error:**
 - '1': CRC delimiter error.
 - '0': no CRC delimiter error.
- **ACK_DEL error:**
 - '1': acknowledgment delimiter error.
 - '0': no acknowledgment delimiter error.
- **EOF error:**
 - '1': end of frame field error.
 - '0': no end of frame field error.
- **RESERVED:**
 - '0': unused.
- **CRC error**
 - '1': a CRC error was found.
 - '0': no CRC error was found.

(b) Source Data

Field Name	Data Type	Description												
CAN ID	UINT32	32-bit value containing the CAN identifier together with the Identifier Extension (IDE) bit:												
		<table><tr><th>Bit(s)</th><th>Name</th><th>Description</th></tr><tr><td>28 - 0</td><td>CAN ID</td><td>CAN Identifier (11 – 29 bits). Unused bits are set to '0'.</td></tr><tr><td>30 - 29</td><td>RESERVED</td><td>'0': unused.</td></tr><tr><td>31</td><td>Identifier Extension (IDE)</td><td>'0': 11 bit ID. '1': 29 bit ID.</td></tr></table>	Bit(s)	Name	Description	28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.	30 - 29	RESERVED	'0': unused.	31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.
		Bit(s)	Name	Description										
		28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.										
30 - 29	RESERVED	'0': unused.												
31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.												
Payload Length	UINT8	Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8.												
Payload	UINT8 [0-8]	Payload Bytes (up to 8 Bytes), depending on the Payload Length.												
CRC	UINT8 [2 Bytes]	CRC (15-bits). The unused upper bit is filled with '0'..												

2.4.3.2 TECMP CAN-FD Data (Data Type = 0x3)

(a) Data Flags

The interface-specific flags provide information regarding the CAN bus and reception condition.

Categorie	Global flags		Interface-specific flags				
Bit position	15	14	13	12 - 9	8	7	6
Description	Overflow	Tx	CRC error	RESERVED	EOF error	ACK_DEL error	CRC_DEL error

Categorie	Interface-specific flags					
Bit position	5	4	3	2	1	0
Description	BIT_STUFF error	BRS	ERR	RESERVED	ESI	ACK

- ACK:
 - '1': there was an acknowledgement during the reception of the CAN-FD frame.
 - '0': there was no acknowledgement during the reception of the CAN-FD frame.
- ESI:
 - '1': the error node is active.
 - '0': the error node is passive.
- ERR:
 - '1': an "Error Frame" was received. In this case no payload is transported and the Payload length will be set to 0.
 - '0': no "Error Frame" was received.
- BRS:
 - '1': the bit rate is switched inside the FD Frame from the nominal bit rate of the arbitration phase to the preconfigured data bit rate of the data phase.
 - '0': the bit rate was not switched inside the FD Frame.
- BIT_STUFF error:
 - '1': a bit stuff error was detected. This is for instance the case if an "Error Frame" is received.
 - '0': no bit stuff error was detected..
- CRC_DEL error: ('0' when not supported).
 - '1': CRC delimiter error.
 - '0': no CRC delimiter error.
- ACK_DEL error: ('0' when not supported).
 - '1': acknowledgment delimiter error.
 - '0': no acknowledgment delimiter error.
- EOF error: ('0' when not supported).
 - '1': end of frame field error.
 - '0': no end of frame field error.
- CRC error
 - '1': a CRC error was found.
 - '0': no CRC error was found.
- RESERVED
 - '0': unused.

(b) Source Data

Field Name	Data Type	Description		
CAN ID	UINT32	32-bit value containing the received CAN identifier together with the Identifier Extension (IDE) bit:		
		Bit(s)	Name	Description
		28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.
		30 - 29	RESERVED	'0': unused.
		31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.
Payload Length	UINT8	Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8, 12, 16, 20, 24, 32, 48, 64.		
Payload	UINT8 [0-64]	Bytes of the payload (up to 64 Bytes), depending on the Payload Length.		
CRC	UINT8 [3 Bytes]	Received CRC. <u>Note:</u> The 17-bits long CRC is used for FD Frames with a data field up to 16 Bytes while the 21-bits long CRC is used for FD Frames with a data field longer than 16 Bytes. The unused upper bits are filled with '0',		

2.4.3.3 TECMP LIN Data (Data Type = 0x4)

(a) Data Flags

The interface-specific flags provide information regarding the LIN bus and reception condition.

Categorie	Global flags		Interface-specific flags				
Bit position	15	14	13	12 - 5	2	1	0
Description	Overflow	Tx	Checksum error	RESERVED	No Slave Response	Parity error	Collision error

- Collision error:
 - '1': there was a collision during the reception of the LIN frame. In this case, the received payload is not valid.
 - '0': there was no collision during the reception of the LIN frame. In this case, the received payload is valid.
- Parity error:
 - '1': there is a parity error in the Protected ID.
 - '0': there is no parity error in the Protected ID.
- No Slave Response:
 - '1': there was no response from a slave on a Request from a master.
 - '0': there was no response from a slave on a Request from a master.
- RESERVED:
 - '0': unused
- Checksum error
 - '1': a Checksum error was found. It can be that the frame on the bus was corrupted or that the wrong checksum model was used in the frame.
 - '0': no Checksum error was found.

(b) Source Data

Field Name	Data Type	Description
LIN ID	UINT8	Frame identifier. <ul style="list-style-type: none"> • Bit 7 – Bit 6 : unused (0x0) • Bit 5 – Bit 0 : LIN ID
Payload Length	UINT8	LIN Length of the LIN payload (1,2,4 or 8 Bytes)
Payload	UINT8 [0-8]	Payload Bytes (up to 8 Bytes), depending on the Payload Length.
Checksum	UINT8	LIN checksum.

2.4.3.4 TECMP Flexray (Data Type = 0x8)

(a) Data Flags

The interface-specific flags provide information regarding the FlexRay bus and reception condition.

Categorie	Global flags		Interface-specific flags			
Bit position	15	14	13	12	11 - 6	
Description	Overflow	Tx	Frame CRC error	Header CRC error	RESERVED	
Categorie	Interface-specific flags					
Bit position	5	4	3	2	1	0
Description	CAS	PPI	WUS	Sync	SF	NF

- Null Frame (NF):
 - '1': the Null frame indicator bit is set.
 - '0': the Null frame indicator bit is not set.
- Startup Frame (SF)
 - '1': the Startup frame indicator bit is set.
 - '0': the Startup frame indicator bit is not set.

- Sync Frame (Sync):
 - '1': the Sync frame indicator bit is set.
 - '0': the Sync frame indicator bit is not set.
- Wakeup Symbol (WUS):
 - '1': it is a WUS. In this case, no data payload is carried, the payload length is set to 0.
 - '0': it is not a WUS.
- Payload Preamble Indicator (PPI):
 - '1': the PPI bit inside the frame is set.
 - '0': the PPI bit inside the frame is not set.
- Collision Avoidance Symbol (CAS):
 - '1': it is a CAS.
 - '0': it is not a CAS.
- RESERVED:
 - '0': unused.
- Header CRC error
 - '1': a Header CRC error was found.
 - '0': no Header CRC error was found.
- Frame CRC error:
 - '1': a Frame CRC error was found.
 - '0': no Frame CRC error was found.

(b) Source Data

Field Name	Data Type	Description
Cycle	UINT8	Cycle in which the frame was received.
Frame ID	UINT16	Frame ID.
Payload Length	UINT8	Length of Flexray Frame Payload for a specific ID.
Payload	UINT8 [0-254]	Bytes of the payload (up to 254 Bytes), depending on the Payload Length.
Header CRC	UINT16 [0-2047]	Header CRC.
Frame CRC	UINT8 [3 Bytes]	Frame CRC.

2.4.3.5 TECMP UART / RS232 ASCII (Data Type = 0x10)

(a) Data Flags

The interface-specific flags detail the UART / RS232 bus and reception condition.

Category	Global flags		Interface-specific flags			
Bit position	15	14	13 - 5	4	3 - 1	0
Description	Overflow	Tx	RESERVED	Framing error	DL	Parity error

- Parity error:
 - '1': a parity error occurs on the RS232 interface. The bit Parity error is set, the message is written completely and then the TECMP packet is sent.
 - '0': no parity error occurred.
- DL (indicates the symbol length on RS232):
 - "010": RS232 with 7 bit.
 - "011": RS232 with 8 bit.
- Framing error:
 - '1': a framing error was detected.
 - '0': no framing error was detected.
- RESERVED:
 - '0': unused.

(b) Source Data

Several data symbols of an RS232 interface are collected inside the Source Data. In this case, only the user data (7 - 8 bit length per data symbol) is transmitted in one Byte (possibly filled with '0 bits').

- A maximum of 1000 RS232 data symbols may be transmitted in one TECMP frame
- The timestamping takes place at the first symbol reception on the RS232 line. Configuration of the interface (bitrate, symbol length, parity bit, ...) takes place in the CM. Settings relevant to the recording are stored in the Data Flags.

Field Name	Data Type	Description
Payload	UINT8	RS-232 message (1 Byte per RS-232 symbol)
...
Payload n	UINT8	

2.4.3.6 TECMP Analog (Data Type = 0x20)

(a) Data Flags

The interface-specific flags provide information regarding the Analog bus.

Category	Global flags		Interface-specific flags					
Bit position	15	14	13 - 10	9	8 - 7	6 - 4	3 - 2	1 - 0
Description	RESERVED	Tx	Sample Time	RESERVED	Factor	RESERVED	Unit	Threshold

- Threshold:
 - "00": no threshold notification.
 - "01": set upper threshold exceeded. At least one sample has exceeded the reference points (setting required in the sample).
 - "10": set lower threshold exceeded. At least one sample has fallen below the reference points (setting required in the sample).
 - "11": undefined.
- Unit:
 - "00": Volt (V).
 - Others: undefined.
- RESERVED:
 - '0': unused.
- Factor:
 - "00": 0,1.
 - "01": 0,01.
 - "10": 0,001.
 - "11": 0,0001.
- Sample Time
 - "0000": 0 (Configuration sample as comparator).
 - "0001": 2500ms.
 - "0010": 1000ms.
 - "0011": 500ms.
 - "0100": 250ms.
 - "0101": 100ms.
 - "0110": 50ms.
 - "0111": 25ms.
 - "1000": 10ms.
 - "1001": 5ms.
 - "1010": 2,5ms.

(b) Source Data

- A maximum of 700 measured values may be transmitted in one TECMP frame
- The timestamping takes place when the first analog value is acquired. Configuration of the interface (voltage, current, sample time, ...) takes place in the sample. Settings relevant to the recording are stored in the Data Flags.

Field Name	Data Type	Description
Sample 0	UINT16	Contains the first value of the recording.
...	UINT16	...
Sample N	UINT16	Contains the N-th value of the recording.

2.4.3.7 TECMP Analog Extended (Data Type = 0x22)

This Data Type supports the capturing of one or more Analog samples.

(a) Data Flags

The interface-specific flags provide information regarding the Analog bus.

Categorie	Global flags		Interface-specific flags		
Bit position	15	14	13 - 10	9 - 6	5 - 0
Description	RESERVED	Tx	Version = 0x1	RESERVED	Sample data length

- Sample data length: number of Bytes defining a sample. Example: 0x2 means two Bytes (UINT16).
- RESERVED:
 - '0': unused.
- Version: reference value for the format of the bits 9 - 0 of the Data Flags section.
 - 0x0 : reserved.
 - 0x1 : this is the format presented inside this document.
 - 0x2 – 0xF : reserved.

(b) Source Data

The format is given by interleaving an Interface ID offset (UINT8), a Timestamp offset (UINT16) and a dedicated Sample configuration (UINT16) between successive samples.

- Samples may not be all taken from the same Interface ID. A 1-Byte long Offset field is interleaved between successive samples. If the offset is null, then the associated sample was taken from the same Interface ID as the Sample 0 (TECMP Header).
- Samples may not be all taken at the same Timestamp. A 2-Byte long Offset field is interleaved between successive samples to provide the time gap (in ns) between the first Sample inside the message (Sample 0) and the following sample. If the offset is null, then the associated sample has the same Timestamp as the Sample 0 (TECMP Header).
- Every Sample has a dedicated Configuration (2-Byte long field).

Field Name	Data Type	Description
Configuration 0	UINT16	Configuration of the Sample 0.
Sample 0	UINT8[Sample data length]	Contains the first value of the measurement, taken on the Interface ID indicated inside the header, at the Timestamp indicated inside the header, with the configuration indicated inside the header.
Interface ID offset 1	UINT8	The Interface ID on which the Sample 1 is measured is given by the sum of the Interface ID offset 1 and the value of the reference Interface ID inside the header.

Timestamp offset 1	UINT16	The Timestamp at which the Sample 1 is measured is given by the sum of the Timestamp offset 1 (ns) and the value of the reference Timestamp ID inside the header.
Configuration 1	UINT16	Configuration of the Sample 1.
Sample 1	UINT8[Sample data length]	Contains the second value of the measurement (Sample 1).
...	(N-2)* (UINT16 + UINT8 + UINT8 + UINT16)	...
Interface ID offset N	UINT8	The Interface ID on which the Sample N is measured is given by the sum of the Interface ID offset N and the value of the reference Interface ID inside the header.
Timestamp offset N	UINT16	The Timestamp at which the Sample N is measured is given by the sum of the Timestamp offset N (ns) and the value of the reference Timestamp ID inside the header.
Configuration N	UINT16	Configuration of the Sample N.
Sample N	UINT8[Sample data length]	Contains the N-th value of the measurement.

For each Sample X, the associated Configuration X field has the following format:

Field Name	Data Type	Description				
Configuration X	UINT16	Bit position	15 – 14	13 - 10	9	8 – 7
		Description	RESERVED	Sample Time	RESERVED	Factor
		Bit position	6 – 4	3 – 2	1 - 0	
		Description	RESERVED	Unit	Threshold	
<ul style="list-style-type: none">Threshold:<ul style="list-style-type: none">"00": no threshold notification."01": set upper threshold exceeded. At least one sample has exceeded the reference points (setting required in the sample)."10": set lower threshold exceeded. At least one sample has fallen below the reference points (setting required in the sample)."11": undefined.Unit:<ul style="list-style-type: none">"00": Volt (V)."01": Ampere (A).Others: undefined.RESERVED:<ul style="list-style-type: none">'0': unused.Factor:<ul style="list-style-type: none">"00": 0,1."01": 0,01."10": 0,001."11": 0,0001.Sample Time<ul style="list-style-type: none">"0000": 0 (Configuration sample as comparator)."0001": 2500ms."0010": 1000ms."0011": 500ms."0100": 250ms.						

		<ul style="list-style-type: none"> ○ "0101": 100ms. ○ "0110": 50ms. ○ "0111": 25ms. ○ "1000": 10ms. ○ "1001": 5ms. ○ "1010": 2,5ms.
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.4.3.8 TECMP Ethernet Data (Data Type = x80)

(a) Data Flags

The interface-specific flags provide the CRC error flag.

Category	Global flags		Interface-specific flags	
Bit position	15	14	13	12 - 0
Description	Overflow	Tx	CRC error	RESERVED

- CRC error:
 - '1': a CRC error was detected.
 - '0': no CRC error was detected.
- RESERVED:
 - '0': unused.

(b) Source Data

Ethernet packets of the size of standard frames are completely transferred to the Ethernet payload of the newly generated packet, from MAC Destination up to FCS (Figure 10 and Table 4).

Example: A recorded Ethernet Message on an IVN Link.

"Source data" comes from a transmission of an Automotive Ethernet Packet (Ethernet II):
The payload is an standard Ethernet frame.

Source Data						
DST MAC	SRC MAC	Outer 802.1Q Header (optional)	Inner 802.1Q Header (optional)	Ether Type	Payload	FCS
6 Bytes	6 Bytes	4 Bytes	4 Bytes	2 Bytes	x Bytes	4 Bytes

Figure 10: Structure of an Ethernet Packet (Ethernet II)

In this case the EtherType of the Ethernet Header is the entry of the message to be recorded from the source network.

Field Name	Data Type	Description
DST MAC	UINT8[6]	Destination MAC address.
SRC MAC	UINT8[6]	Source MAC address.
Outer VLAN Tag (optional)	UINT32	Outer 802.1Q Header.
Inner VLAN Tag (optional)	UINT32	Inner 802.1Q Header.
EtherType	UINT16	Ethernet protocol Type.
Payload	UINT8[x]	Payload Bytes.
Checksum	UINT32	Frame Check Sequence.

Table 4: Description of the fields of an Ethernet Packet (Ethernet II)

2.4.4 Replay specification (“Replay Data” Message Type)

As introduced in Chapter 2.3, the TECMP protocol supports the replaying of data into an in-vehicle network. A Replay Data example is represented in the next Figure.

4 Bytes			
1	2	3	4
DST MAC (MSB) : Unused			
DST MAC (LSB) : Unused		SRC MAC (MSB)	
SRC MAC (LSB)			
Outer VLAN (Optional)			
Inner VLAN (Optional)			
EtherType = 0x99FE		Device ID : Unused	
Counter = 0x0		Version = 0x3	Message Type = 0xA
Data Type = 0xE002 (example)		Reserved	
CM Flags = 0x0		Interface ID (MSB) of the target Interface	
Interface ID (LSB) of the target Interface		Timestamp (MSB) = 0x0 (example)	
Timestamp = 0x0 (example)			
Timestamp (LSB) = 0x0 (example)		Length	
Data Flags		Source Data	
...			

Figure 11: Replay Data example

In this example, a CAN frame (Data Type = 0xE002) shall be replayed (Message Type = 0xA) given the current protocol Version (0x3). Th frame shall be sent as soon as possible (Timestamp = 0x0). The possible values of the Data Flags and Source Data for every Data Type are defined in the next subchapters.

2.4.4.1 TECMP CAN Tx (Data Type = 0xE002)

(a) Data Flags

The Data Flags field is unused: 0x0000.

Bit position	15 – 0
Description	RESERVED
Replaying	Reserved: 0x0

- RESERVED:
 - '0': unused.

(b) Source Data

Field Name	Data Type	Description												
CAN ID (inclusive IDE bit)	UINT32	32-bit value containing the CAN identifier together with the Identifier Extension (IDE) bit:												
		<table><tr><th>Bit(s)</th><th>Name</th><th>Description</th></tr><tr><td>28 - 0</td><td>CAN ID</td><td>CAN Identifier (11 – 29 bits). Unused bits are set to '0'.</td></tr><tr><td>30 - 29</td><td>RESERVED</td><td>'0': unused</td></tr><tr><td>31</td><td>Identifier Extension (IDE)</td><td>'0': 11 bit ID. '1': 29 bit ID.</td></tr></table>	Bit(s)	Name	Description	28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.	30 - 29	RESERVED	'0': unused	31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.
		Bit(s)	Name	Description										
		28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.										
		30 - 29	RESERVED	'0': unused										
31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.												

Payload Length	UINT8	Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8.
Payload	UINT8 [0-8]	Payload Bytes (up to 8 Bytes), depending on the Payload Length.
CRC	UINT8 [2 Bytes]	The provided CRC (15-bits) value will be used for the transmission. Note: The unused upper bit shall be filled with '0': the range 0x8000-0xFFFF is undefined.

2.4.4.2 TECMP CAN-FD Tx (Data Type = 0xE003)

(a) Data Flags

The Data Flags field is unused: 0x0000.

Bit position	15 - 0
Description	RESERVED
Replaying	0x0

- RESERVED:
 - '0': unused.

(b) Source Data

Field Name	Data Type	Description												
CAN ID (inclusive IDE bit)	UINT32	32-bit value containing the received CAN identifier together with the Identifier Extension (IDE) bit:												
		<table><tr><th>Bit(s)</th><th>Name</th><th>Description</th></tr><tr><td>28 - 0</td><td>CAN ID</td><td>CAN Identifier (11 – 29 bits). Unused bits are set to '0'.</td></tr><tr><td>30 - 29</td><td>RESERVED</td><td>'0': unused</td></tr><tr><td>31</td><td>Identifier Extension (IDE)</td><td>'0': 11 bit ID. '1': 29 bit ID.</td></tr></table>	Bit(s)	Name	Description	28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.	30 - 29	RESERVED	'0': unused	31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.
		Bit(s)	Name	Description										
		28 - 0	CAN ID	CAN Identifier (11 – 29 bits). Unused bits are set to '0'.										
30 - 29	RESERVED	'0': unused												
31	Identifier Extension (IDE)	'0': 11 bit ID. '1': 29 bit ID.												
Payload Length	UINT8	Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8, 12, 16, 20, 24, 32, 48, 64.												
Payload	UINT8 [0-64]	Bytes of the payload (up to 64 Bytes), depending on the Payload Length.												
CRC	UINT8 [3 Bytes]	The provided CRC (17/21-bits) value will be used for the transmission. Note: The unused upper bits shall be filled with '0'. Note: The 17-bits long CRC is used for FD Frames with a data field up to 16 Bytes long while the 21-bits long CRC is used for FD Frames with a data field longer than 16 Bytes.												

2.4.4.3 TECMP LIN Tx (Data Type = 0xE004)

(a) Data Flags

The Data Flags field is unused: 0x0000.

Bit position	15 - 0
Description	RESERVED
Replaying	0x0

- RESERVED:
 - '0': unused.

(b) Source Data

Field Name	Data Type	Description
LIN ID	UINT8	Frame identifier: <ul style="list-style-type: none"> • Bit 7 – Bit 6 : reserved (0x0) • Bit 5 – Bit 0 : LIN ID
Payload Length	UINT8	LIN Length of the LIN payload (1,2,4 or 8 Bytes). A 0-length value can be used to remove the slave response configuration entry associated with a given LIN ID, from the CM memory.
Payload	UINT8 [0-8]	Payload Bytes (up to 8 Bytes), depending on the Payload Length.
Checksum	UINT8	LIN checksum: the provided Checksum value (0x00-0xFF) will be used for the transmission.

2.4.4.4 TECMP Flexray Tx (Data Type = 0xE008)

(a) Data Flags

The interface-specific flags provide information regarding the FlexRay bus and transmission condition.

Bit position	15 - 13	12	11 - 10	9 - 7	6 - 0
Description	RESERVED	Explicit Header CRC	RESERVED	Tx Mode	RESERVED
Replaying	0x0	X	0x0	X	0x0

- RESERVED:
 - '0': unused.
- Tx Mode (Transmission Mode):
 - 0x0: Reserved.
 - 0x1: Single Shot transmission mode.
 - The configured frame (Cycle Offset, Cycle Repetition, Frame ID...) is transmitted exactly once.
 - 0x2: Continuous transmission mode.
 - The transmission of the configured frame (Cycle Offset, Cycle Repetition, Frame ID...) remains valid until the host explicitly marks it as invalid (TX-NONE).
 - 0x3: TX-NONE.
 - The configured frame (Cycle Offset, Cycle Repetition, Frame ID...) is explicitly marked as invalid.
 - 0x4 – 0x15: Undefined.
- Explicit Header CRC
 - '1': the provided Checksum value (0x0000-0x07FF) will be used for the transmission. Any other value (0x800 – 0xFFFF) will not be used. Instead, the device will compute the correct value for the transmission.
 - '0': the provided Checksum value will not be used. The device will compute the correct value for the transmission.

(b) Source Data

Field Name	Data Type	Description
Cycle	UINT8	Cycle Offset
Frame ID	UINT16	Frame ID
Payload Length	UINT8	Length of Flexray Frame Payload for a specific ID
Payload	UINT8 [0-254]	Bytes of the payload (up to 254 Bytes), depending on the Payload Length.

Cycle Repetition	UINT8	Valid Values: 1-2-4-8-16-32-64
Header CRC	UINT16 [0-2047]	Depending on the value of the Explicit Header CRC bit (12) inside the Data Flags: <ul style="list-style-type: none"> • '0' <ul style="list-style-type: none"> ○ The provided Header CRC value will not be used. The device will compute the correct value for the transmission. • '1' <ul style="list-style-type: none"> ○ 0x0000-0x07FF: The provided Header CRC value will be used for the transmission. ○ 0x0800-0xFFFF: Undefined: The device will compute the correct value for the transmission.

Note: The Frame CRC will be computed by the CM and added to the frame. Its value is not user-defined.

2.4.4.5 TECMP GPIO Tx (Data Type = 0xE00A)

This Data Type supports four different types of operation:

Type of operation	Subtype	Current mode	Description	Operation code (4Bytes)
Channel status	Simulation	Normal Current Channel	Open Load	0x0000 0001
			Short to Vbat	0x0000 0002
			Short to Gnd	0x0000 0004
		High Current Channel	Open Load	0x0000 0008
			Short to Vbat	0x0000 0010
			Short to Gnd	0x0000 0020
	Enable / Disable		Voltage Sensing	0x0000 0040
			High Voltage Sensing	0x0000 0080
			PWM Measurements	0x0000 0100
Potentiometer	Simulation	(not applicable)	Resistive Sensor or resistance coded switch	0x0001 0000
Signal generator	Simulation		Signal generator	0x0002 0000
DC Load	Simulation		DC Load	0x0004 0000

(a) Data Flags

The Data Flags field is unused: 0x0000.

Bit position	15 - 0
Description	RESERVED
Replaying	0x0

(b) Source Data

To each type of operation corresponds a specific Source Data field format covered by one or multiple dedicated "Operation" values (first field of the Source Data).

(i) Channel status

Field Name	Data Type	Description																														
Operation	UINT32	<ul style="list-style-type: none"> • 0x0000 0000 – reserved • 0x0000 0001 – Simulate Open Load • 0x0000 0002 – Simulate Short to Vbat • 0x0000 0004 – Simulate Short to Gnd • 0x0000 0008 – Simulate Open Load on High Current Channel • 0x0000 0010 – Simulate Short to Vbat on High Current Channel • 0x0000 0020 – Simulate Short to Gnd on High Current Channel • 0x0000 0040 – Enable/Disable Voltage Sensing • 0x0000 0080 – Enable/Disable High Voltage Sensing • 0x0000 0100 – Enable/Disable PWM Measurements • 0x0000 0200 – reserved • 0x0000 0400 – reserved • 0x0000 0800 – reserved • 0x0000 1000 – reserved • 0x0000 2000 – reserved • 0x0000 4000 – reserved • 0x0000 8000 – reserved 																														
Length	UINT16	Length of the following fields, aggregated (in Bytes). The value shall be an even number.																														
Field 1 (Mask)	UINT8 [Length/2] Array of length „Length/2”	<ul style="list-style-type: none"> • Mask to select the channels to be updated • Bit 0(LSB) – Bit 23 corresponds to Channels 0-23 • Bits 24-47 are reserved • 1 = Channel will be updated • 0 = No Update 																														
Field 2 (Value)	UINT8 [Length/2] Array of length „Length/2”	<ul style="list-style-type: none"> • New state or value of channels selected in Field 1 • Only the channels selected in the channel mask will be updated. • Bit 0(LSB) – Bit 23 corresponds to Channel 0-23 • Bits 24-31 are reserved <table border="1"> <thead> <tr> <th>Type</th><th>Bit_n = 0</th><th>Bit_n = 1</th></tr> </thead> <tbody> <tr> <td>O/L</td><td>Channel is open</td><td>Channel is closed</td></tr> <tr> <td>Vbat</td><td>Vbatt is OFF</td><td>Short to Vbatt</td></tr> <tr> <td>Gnd</td><td>Gnd is OFF</td><td>Short to GND</td></tr> <tr> <td>HC_O/L</td><td>Channel is open</td><td>Channel is closed</td></tr> <tr> <td>HC_Vbat</td><td>Vbatt is OFF</td><td>Short to Vbatt</td></tr> <tr> <td>HC_Gnd</td><td>Gnd is OFF</td><td>Short to GND</td></tr> <tr> <td>VSense</td><td>Disable Vsense</td><td>Enable Vsense</td></tr> <tr> <td>HV_Sense</td><td>Disable HV_Sense</td><td>Enable HV_Sense</td></tr> <tr> <td>PWM_Sense</td><td>Disable PWM Sense</td><td>Enable PWM Sense</td></tr> </tbody> </table>	Type	Bit_n = 0	Bit_n = 1	O/L	Channel is open	Channel is closed	Vbat	Vbatt is OFF	Short to Vbatt	Gnd	Gnd is OFF	Short to GND	HC_O/L	Channel is open	Channel is closed	HC_Vbat	Vbatt is OFF	Short to Vbatt	HC_Gnd	Gnd is OFF	Short to GND	VSense	Disable Vsense	Enable Vsense	HV_Sense	Disable HV_Sense	Enable HV_Sense	PWM_Sense	Disable PWM Sense	Enable PWM Sense
Type	Bit_n = 0	Bit_n = 1																														
O/L	Channel is open	Channel is closed																														
Vbat	Vbatt is OFF	Short to Vbatt																														
Gnd	Gnd is OFF	Short to GND																														
HC_O/L	Channel is open	Channel is closed																														
HC_Vbat	Vbatt is OFF	Short to Vbatt																														
HC_Gnd	Gnd is OFF	Short to GND																														
VSense	Disable Vsense	Enable Vsense																														
HV_Sense	Disable HV_Sense	Enable HV_Sense																														
PWM_Sense	Disable PWM Sense	Enable PWM Sense																														

Example:

Req 1: Set channels 0,2,4 20,22 to Gnd
 Send TECMP packet with following information:
 Data Type = GPIO Tx

DataFlags = 0x0000 or reserved value
 (SourceData) Operation = 0x0000 0004
 (SourceData) Length = 6
 (SourceData) Field 1 = 0x555555 → Select Channel 0,2,4 20,22 for an update
 (SourceData) Field 2 = 0x555555 → Activate short to GND on channels 0,2,4 20,22

(ii) Potentiometer

Field Name	Data Type	Description
Operation	UINT32	0x0001 0000 – Simulate Resistive Sensor or resistance coded switch.
Length	UINT16	Length of the following fields, aggregated (in Bytes). The value shall be a multiple of 9.
Field 1	UINT8	ID of the output. The maximum value depends on the device used.
Field 2	UINT64	Resistor Value in Ohms.
...		
...		
Field 1	UINT8	ID of the output.
Field 2	UINT64	Resistor Value in Ohms.

(iii) Signal generator

Field Name	Data Type	Description				
Operation	UINT32	0x0002 0000 – Generate Patterns				
Length	UINT16	Length of the following fields, aggregated (in Bytes) Field 1 until Field n + 1.				
Field 1	UINT16	Bit position	15..13	12..11	10..8	7..0
		Description	Reserved	Unit	Mode	Output ID
Output ID: 0-7, ID of the output pin						
Mode: 0b000 = Stop 0b001 = Periodic 0b010 = Non-Periodic						
Unit: 0b00 = mSec 0b01 = uSec 0b10 = nSec 0b11 = Reserved						
Field 2	UINT64	Bit 63=1 (ON TIME) Bit 63=0(OFF TIME) Bits 62..0 = Time is mS, uS or nS as per data flags				
...				
Field n	UINT64	Bit 63=1 (ON TIME) Bit 63=0(OFF TIME) Bits 62..0 = Time is mS, uS or nS as per data flags				
Field n +1	UINT64	Bit 63=1 (ON TIME) Bit 63=0(OFF TIME) Bits 62..0 = Time is mS, uS or nS as per data flags				

(iv) DC Load

Field Name	Data Type	Description
Operation	UINT32	0x0004 0000 – Simulate DC Load
Length	UINT16	Length of the following fields, aggregated (in Bytes)
Field 1	UINT16	Output ID.
Field 2	UINT64	DC Load in mAmps (Min/Max values TBD)

2.4.5 Packetizing IVN messages

To optimize the performance of the communication, it is possible to limit and reuse the same header (12 Bytes) only once over multiple messages. The solution consists in transferring at once several IVN messages packetized in one frame to the attached sinks (for capturing data) or to the CM (for replaying data). The first 12 Bytes of the TECMP Header are sent only once at the beginning of the packetized IVN messages. The remaining header is appended to each message logged.

Format (Example with a single CAN-message):

Device ID	Counter	Version	Message Type	Data Type	Reserved	CM Flags	Interface ID	Time stamp	Length	Data Flags	CAN-Frame
-----------	---------	---------	--------------	-----------	----------	----------	--------------	------------	--------	------------	-----------

Figure 12: Packaging of a single CAN message

Format (Example with multiple CAN-messages):

Device ID	Counter	Version	Message Type	Data Type	Reserved	CM Flags	Interface ID	Time stamp	Length	Data Flags	CAN-Frame 0
							Interface ID	Time stamp	Length	Data Flags	CAN-Frame 1
							Interface ID	Time stamp	Length	Data Flags	CAN-Frame 2
							Interface ID	Time stamp	Length	Data Flags	CAN-Frame 3

Figure 13: Packaging of multiple CAN messages

When capturing data:

- If, despite the packetizing of several messages, the minimum size of a 64-Byte Ethernet frame cannot be reached, the CM inserts appropriate zero padding at the end of the TECMP message. There is no padding between the successive messages.
- The protocol does not provide a dedicated field indicating how many messages are packaged because this information can be retrieved based on the value of the length (header field) of the Source Data and the overall buffered length of the TECMP message.

When replaying data:

- Multiple FlexRay frames can be replayed out of the same TECMP message.
- No padding shall be used between the successive FlexRay frames inside the TECMP message.
- Other protocols are not supported. Successive individual TECMP messages shall be used.

2.5 Status Message

Specific Status Messages can be generated by CMs. Those messages are sent in parallel with the Logging Stream and can be used by the sink to identify or debug the status of the capturing interface of the CM or the CM itself during the recording. Additionally, the configuration of the CM during the recording can also be retrieved:

1. Following the occurrence of an internal event (Timeout period), a Status Message is internally generated.
2. This Status Message is sent towards the sink.

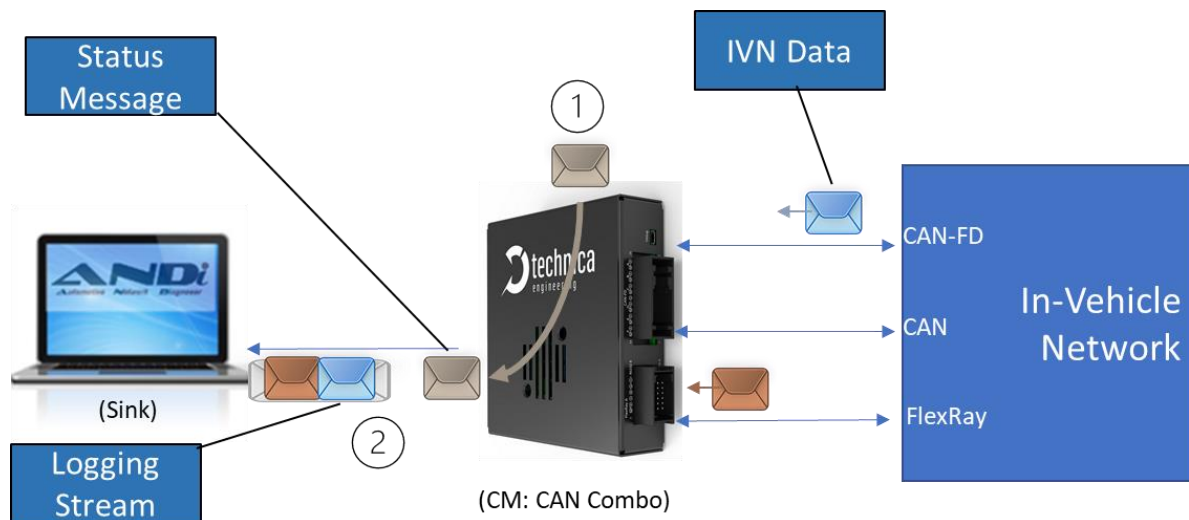


Figure 14: Status Message generation

Three different Status Messages are currently supported by CMs:

- Status Configuration
- Status Bus/IVN
- Status CM

The Status Messages use the following TECMP Header.

Device ID	Counter	Version	Message Type	Data Type	Reserved	CM Flags	Interface ID	Time stamp	Length	Data Flags	Status Message
-----------	---------	---------	--------------	-----------	----------	----------	--------------	------------	--------	------------	----------------

Figure 15: Status Messages TECMP overview

The maximal length of Status Messages is 1400 Bytes. If the payload to be transferred does not fit in a single Status Message, the payload will be segmented and transferred in multiple and successive Status Messages of the same type, each carrying one segment.

2.5.1 Status Configuration message

Status Configuration messages are sent cyclically every 60 seconds after the logging link has been established. The payload is described in the next Tables.

4 Bytes			
0	1	2	3
Vendor ID	CM Version	CM Type	Reserved
Vendor Data Length (VDL)		Device ID	
Serial Number			
Vendor Data			

Table 5: Status Configuration message payload

Field Name	Length	Description
Vendor ID	1 Byte	Vendor ID - Technica Engineering: 0x0C
CM Version	1 Byte	Versioning of the CM
CM Type	1 Byte	Type of CM (e.g.: CAN,FR,Ethernet,LIN) , see Table 27
Reserved	1 Byte	For future extensions
Vendor Data Length	2 Bytes	Length of the "Vendor Data" field (in Bytes)
Device ID	2 Bytes	Configured Capture Module ID
Serial Number	4 Bytes	Serial number of the CM
Vendor Data	n Byte(s)	Technica Engineering specific CM configuration

Table 6: Status Configuration message fields description

Any Capture Module HW variant can be easily identified based on the notified Serial Number and Capture Module version.

Device ID	Counter	Version	Message Type	Data Type	Reserved	CM Flags	Interface ID	Time stamp	Length	Data Flags
Vendor ID	CM Version	CM Type	Reserved	Vendor Data Length	Device ID	Serial Number	Vendor Data			

Table 7: Status Configuration message overview

CM Version:

Version of the Capture Module. Used to distinguish different Capture Module variants / revisions.

Vendor ID:

Technica Engineering	0x0C
----------------------	------

CM Type:

See Table 27.

Capture Module Serial Number:

This field is 4 Bytes long. In case that the correct serial number for the device is not saved in the memory, the last 4 Bytes of the MAC address will be shown as serial number

Vendor Data:

Multiple configuration parameters are exchanged with the CM in the form of a serialized JSON configuration file. The format of the Vendor Data field is depicted and described in the next two illustrations.

4 Bytes			
0	1	2	3
Version	Reserved	Config Message ID	
Total Length (Bytes)			
Total number of segments		Segment number	
Length of segment (Bytes)		Segment Data	...
...			

Table 8: Vendor Data overview

Field Name	Data Type	Description
Version	UINT8	Version number of the Status Configuration message vendor data format
Reserved	UINT8	Reserved field: 0x00
Config Message ID	UINT16	Identifier incremented by one at every new cyclic transmission
Total Length	UINT32	Sum of the lengths of the individual segments
Total number of segments	UINT16	Number of segments onto which the information is distributed. As many number of Status Configuration messages as segments will be generated per cyclic transmission
Segment number	UINT16	Identification of the present segment: between 0 and the value of the field "Total number of segments"-1
Length of segment	UINT16	Length (x) of the following "Segment Data" field in Bytes
Segment Data	x * UINT8	Serialized configuration

Table 9: Status Configuration message Vendor Data fields description

2.5.2 Status Bus/IVN message

Status Bus/IVN messages are sent cyclically every second after the logging link has been established. CMs with multiple interfaces send the status of all connected IVN's. The payload is depicted in the next illustration.

4 Bytes			
0	1	2	3
Vendor ID	CM Version	CM Type	Reserved
Vendor Data Length (VDL)		Device ID	
Serial Number			
Bus Data			

Table 10: Status Bus/IVN message payload overview

4 Bytes	
Bus Data Entry 1	
Bus Data Entry 2 (optional)	
Bus Data Entry 3 (optional)	
Bus Data Entry 4 (optional)	
Optional further Entries ...	

Table 11: Bus Data

4 Bytes	
Interface ID	
Messages Total	
Errors Total	
Vendor Data	

Table 12: Bus Data Entry

Field Name	Length	Description
Vendor ID	1 Byte	Vendor ID - Technica Engineering: 0x0C
CM Version	1 Byte	Versioning of the CM
CM Type	1 Byte	Type of CM (e.g.: CAN, FR, Ethernet, LIN)
Reserved	1 Byte	For future extensions

Vendor Data Length	2 Bytes	Length of the "Vendor Data" field per Bus (in Bytes)
Device ID	2 Bytes	Configured Capture Module ID
Serial Number	4 Bytes	Serial number of the CM
Interface ID	4 Bytes	ID that uniquely identifies the log data / bus / link of the IVN on which the Messages/frames were received or sent
Messages Total	4 Bytes	Messages received on the bus/IVN (Interface ID) since the startup of the CM
Errors Total	4 Bytes	Erroneous messages received on the bus/IVN (Interface ID) since the start of the Capture Module
Vendor Data	n Bytes	See description below

Table 13: Description of the Status Bus/IVN message payload fields

For one bus/IVN interface:

Device ID	COUNTER	VERSION	Message Type	DATA TYPE	RESERVED	CM Flags	Interface ID	TIME STAMP	LENGTH	Data Flags
VENDOR ID	CM VERSION	CM TYPE	RESERVED	LENGTH	Device ID	Serial Number	Interface ID	MESSAGES TOTAL	ERRORS TOTAL	VENDOR DATA

Figure 16: Packetizing a single Status Bus/IVN in one Status Bus/IVN message

For multiple bus/IVN interfaces:

Device ID	COUNTER	VERSION	Message Type	DATA TYPE	RESERVED	CM Flags	Interface ID	TIME STAMP	LENGTH	Data Flags
VENDOR ID	CM VERSION	CM TYPE	RESERVED	VENDOR DATA LENGTH	Device ID	Serial Number	Interface ID	MESSAGES TOTAL	ERRORS TOTAL	VENDOR DATA
							Interface ID	MESSAGES TOTAL	ERRORS TOTAL	VENDOR DATA
							Interface ID	MESSAGES TOTAL	ERRORS TOTAL	VENDOR DATA

Figure 17: Packetizing multiple Status Bus/IVN in one Status Bus/IVN message

Messages Total and Errors Total:

The value of the absolute received messages and respectively messages with errors are available in the respective fields. An error increases both the Messages Total and the Errors Total value. Further information can be transmitted in the Vendor Data.

Vendor Data

The Vendor Data field is used to convey specific information in every Capture Module variant. Consequently, the field has different meaning in different Capture Module types. As an example, for both the CM Ethernet Combo and CM 100 High variants:

Vendor Data for CM Eth Combo & CM 100 High	
Bytes	
0	1
Link Status	Link Quality
Linkup Time	

Table 14: Status Bus/IVN message Vendor Data example

Field Name	Data Type	Description
Link Status	UINT8	Flag to indicate whether there is linkUp or not in the corresponding channel: x00: No x01: Yes

Link Quality	UINT8	Quality of the link of the corresponding channel, from 0 to 5: 0x00: LinkDown 0x01: Lowest Link quality ... 0x05: Highest Link quality
Linkup Time	UINT16	Time (in ms) measured between power up of the board and link up on this port. Two default reference values are defined based on an internal reference timeout elapsing 500ms after power up. <ul style="list-style-type: none"> 0x0000: No linkup detected and the reference timeout did not elapse yet. 0xFFFF: No linkup detected and the reference timeout already elapsed. For instance: 0x153 means 339 ms. Note: This value is updated only once (at startup). The value is not updated by disconnecting and reconnecting the cable.

Table 15: Link Status, Link Quality and Linkup Time fields description

2.5.3 Status Capture Module message

Status Capture Module messages are sent cyclically every second after the logging link has been established. The payload is depicted in Table 16 and described in Figure 18.

4 Bytes			
0	1	2	3
Vendor ID	CM Version	CM Type	Reserved
Vendor Data length (VDL)		Device ID	
Serial Number			
Vendor Data			

Table 16: Status CM Message payload overview

Field Name	Length	Description
Vendor ID	1 Byte	Vendor ID – Technica Engineering: 0x0C
CM Version	1 Byte	Versioning of the CM
CM Type	1 Byte	Type of Capture Module (eg: CAN-Combo)
Reserved	1 Byte	For future extensions
Vendor Data Length	2 Bytes	Length of the “Vendor Data” field (in Bytes)
Device ID	2 Bytes	Configured ID of the Capture Module
Serial Number	4 Bytes	Serial number of the Capture Module
Vendor Data	n Byte(s)	Specific status information (e.g.: temperature processor, or CM’s error conditions)

Figure 18: Description of the Status CM message payload fields

Vendor Data - for all CM Variants:

4 Bytes			
0	1	2	3
Reserved	SW Version		
HW Version		Buffer fill level	Buffer overflow
Buffer Size			
Lifecycle			
Lifecycle			
Voltage Integer part	Voltage fractional part	Temperature	

Table 17: Status CM message Vendor Data overview prior to SW version v015.X.010

4 Bytes			
0	1	2	3
Reserved	SW Version		
HW Version		Buffer fill level	Buffer overflow
Buffer Size			
Lifecycle			
Lifecycle			
Voltage Integer part	Voltage fractional part	Chassis Temperature	Silicon Temperature

Table 18: Status CM message Vendor Data overview from SW version v015.X.010 onwards

Field Name	Data Type	Description
Reserved	UINT8	Reserved
SW Version	3x UINT8	vX.Y.Z, one Byte for each part. E.g. v8.6.40 → 0x08 0x06 0x28
HW Version	2x UINT8	vX.Y, one Byte for each part. E.g. v3.1 → 0x03 0x01
Buffer fill level	UINT8	Percentage of internal memory for reception currently used. Possible values: 0, 10, 20, 30, 40, 50, 60, 70, 80, 85, 90, 95. <ul style="list-style-type: none"> For the Rx Buffer, 100% is not possible because the memory is emptied before it reaches that level. A value of 0% can also indicate that no memory is available.
Buffer overflow	UINT8	Flag to indicate whether the memory reached its upper internal memory limit while storing input packets from IVN/Bus inside the internal memory of the CM. x00: No, x01: Yes In case this limit is reached, the CM takes internal countermeasures, stopping recording until the internal memory returns to a functioning level. Once recording is restarted, the flag is reset. Any message received during this time was lost.
Buffer size	UINT32	Size of the internal memory in Gigabits.
Lifecycle	UINT64	Time since startup, in ns.
Voltage integer part	UINT8	Voltage value, the part before the comma. E.g: 12,30 V → 12 = 0x0C
Voltage fractional part	UINT8	Voltage value, the part after the comma. E.g: 12,30 V → 30 = 0x1E
Chassis Temperature	SINT8	Temperature of the CM chassis in °C. <ul style="list-style-type: none"> x00 up to 0x7E → 0°C up to 126°C 0x7F → temp ≥ 127°C 0x80 → Not Available 0xFF down to 0x81 → -1°C down to -127°C
Silicon Temperature	SINT8	Internal temperature (junction temperature) of the CM processing unit in °C. <ul style="list-style-type: none"> x00 up to 0x7E → 0°C up to 126°C 0x7F → temp ≥ 127°C 0x80 → Not Available 0xFF down to 0x81 → -1°C down to -127°C

Table 19: Status CM message Vendor Data fields description from SW version v015.X.010 onwards

2.6 Control message

To control the recording system and all possible connected systems on the logging network (consisting of several loggers, Captures Module and instrumentation adapters) and signaling events in the recorded trace, specific Control messages are defined by the TECMP protocol. Which Control Message is active in the measurement components depends on their configuration. All CMs can be informed about the occurrence of a configurable Control message. Received Control messages are passed on by all components inside the measurement network. Control Messages are either received by a CM (Figure 19) or generated by it (Figure 20).

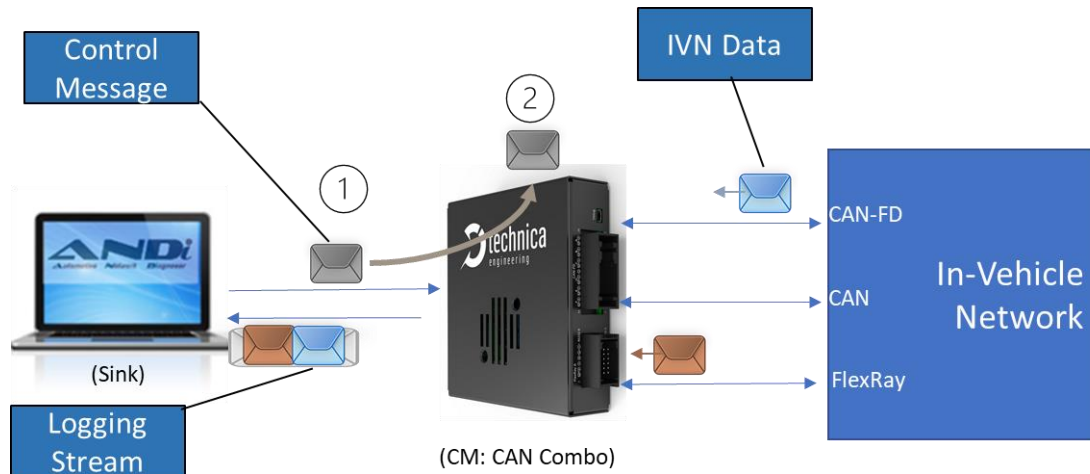


Figure 19: Reception of a Control Message by the CM.

In Figure 19:

1. The CM receives a Control Message.
2. It is internally checked and processed according to the Control Message ID (Table 26).

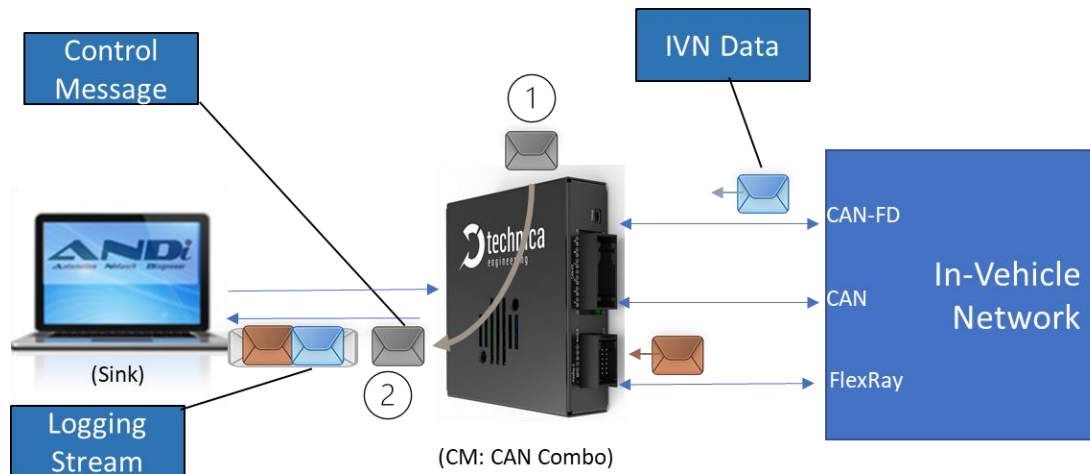


Figure 20: Generation of a Control Message by the CM.

In Figure 20:

1. Following the occurrence of an event (Ethernet Link down detected, time synchronization acquired...), a Control Message is internally generated.
2. This Control Message is sent towards the sink.

The format of the payload of a Control Message is defined by the following table:

4Byte			
0	1	2	3
Device ID		Control Message ID	
Vendor Data (optional)			

Table 20: Control message payload overview

- **Device ID:** sender of the Control Message (see Table 25).
- **Control Message ID:** Identifier of the specific Control Message (see Table 26).

The format of the TECMP Control Message is defined by the following table:

CM ID	Counter	Version	Message Type	Data Type	Reserved	CM Flags	Channel ID	Time-stamp	Length	Data Flags	Device ID	Control Message ID	Vendor Data (Optional)
-------	---------	---------	--------------	-----------	----------	----------	------------	------------	--------	------------	-----------	--------------------	------------------------

Table 21: Control message overview

2.6.1 CAN Replay fill level

The CAN Replay fill level is a special Control Message, supporting the CAN Replay feature:

1. The CAN(-FD) data to transmit on the IVN is received by the CM in the form of one or more TECMP messages.
2. The CAN(-FD) data to transmit on the IVN is sent out by the CM accordingly to the CAN Arbitration principle.
3. Depending on the Busload and the replaying scheme, it might happen that the memory resource of the CM is put under too much pressure. This case coincides with the generation of the CAN Replay fill level Control Message.
4. This Control Message is sent to the Sink.

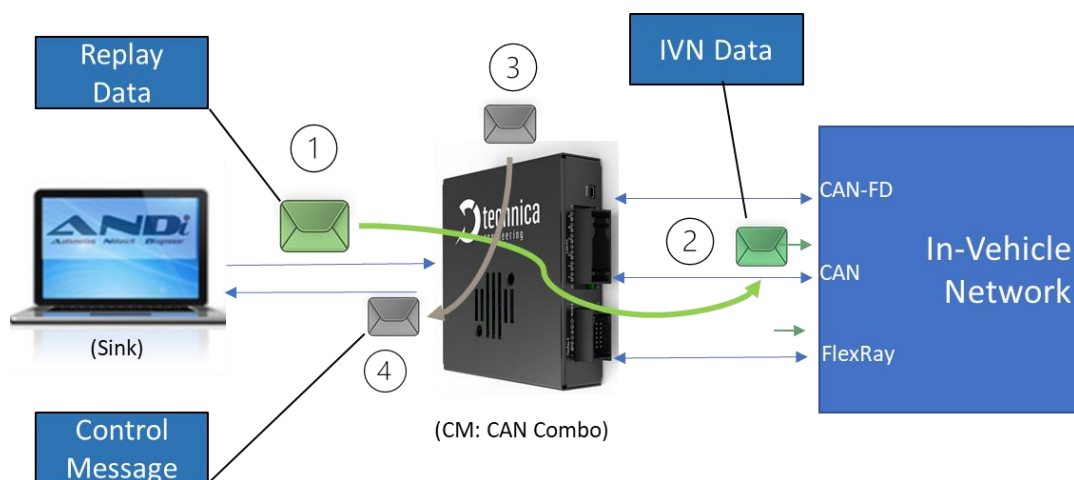


Figure 21: Generation of the CAN Replay fill level Control Message

The Vendor Data format of the CAN Replay fill level is defined as follows:

Name	Length	Description
Capture Module ID	UINT16	Sender of the Control Message.
Control Message ID	UINT16	Identifier of the specific Control Message.
DDR fill Level	UINT8	Percentage of the DDR memory currently used. Possible values: 0, 10, 20, 30, 40, 50, 60, 70, 80, 85, 90, 95.
DDR Buffer Overflow	UINT8	Flag to indicate whether there is a memory overflow or not.

		x00: No, x01: Yes
CAN Queue size	UINT8	Maximum number of frames which can be stored in one CAN Queue.
CAN Queue fill level	UINT8[x]	Number of frames currently stored in the Queue x: between 0 and Max number.

Table 22: CAN Replay fill level

2.7 TECMP Header fields summary

The supported values of the TECMP header fields are documented in this chapter.

2.7.1 Message Type

Name	Dec.	Hex.
Control Message	0	0x0
Status CM	1	0x1
Status Bus/IVN	2	0x2
Logging Stream	3	0x3
Status Configuration	4	0x4
Replay Data	10	0xA

Table 23: Message Type

Note: "Logging Stream" identifies data messages which are received by the CM. "Replay Data" identifies data messages which are sent by the CM.

2.7.2 Data Type

Feature	Definition	Dec.	Hex.	Comment
Capturing	None (UNDEFINED)	00	0x0000	Data-Logging in combination with the "Logging Stream" Message Type (0x3).
	CAN	02	0x0002	
	CAN-FD	03	0x0003	
	LIN	04	0x0004	
	Flexray	08	0x0008	
	UART/RS232	16	0x0010	
	Analog	32	0x0020	
	Analog extended	34	0x0022	
	Ethernet II (classical Ethernet frame)	128	0x0080	
Replay	Reserved	57344	0xE000	Data-Replaying in combination with the "Replay Data" Message Type (0xA).
	CAN Tx	57346	0xE002	
	CAN-FD Tx	57347	0xE003	
	LIN Tx	57348	0xE004	
	Flexray Tx	57352	0xE008	
	GPIO Tx	57354	0xE00A	

Table 24: Data Type

2.7.3 Device ID

In order to enforce interoperability, the IP addresses of the CMs is defined based on the IP-network 10.104.3.xxx.

Capture Module	IP Address 10.104.3.xx	Device ID 2 Bytes	Rotary switch position (0x)
LIN Combo			
LIN Combo 0	10.104.3.48	48	0
LIN Combo 1	10.104.3.49	49	1
LIN Combo 2	10.104.3.50	50	2
...
LIN Combo 15	10.104.3.63	63	F
CAN Combo			
CAN Combo 0	10.104.3.64	64	0
CAN Combo 1	10.104.3.65	65	1
CAN Combo 2	10.104.3.66	66	2
...
CAN Combo 15	10.104.3.79	79	F
100 High			
100 High 0	10.104.3.96	96	0
100 High 1	10.104.3.97	97	1
100 High 2	10.104.3.98	98	2
...
100 High 15	10.104.3.111	111	F
Eth Combo			
Eth Combo 0	10.104.3.128	128	0
Eth Combo 1	10.104.3.129	129	1
Eth Combo 2	10.104.3.130	130	2
...
Eth Combo 15	10.104.3.143	143	F
1000 High			
1000 High 0	10.104.3.144	144	0
1000 High 1	10.104.3.145	145	1
1000 High 2	10.104.3.146	145	2
...
1000 High 15	10.104.3.159	159	F
User-defined	User-defined	0xF000 – 0xFFFF	n/a
Reserved	Reserved	0xFFFF	n/a

Table 25: Device ID

2.7.4 Control Message ID

The type of control message is defined via a 2 Bytes long Control Message ID field.

Control Message			
Name	ID	Source	Description
Logger Ready	0x00 02	Logger	This control message notifies the logger readiness to receive data. It shall be sent cyclically every second.
CAN Replay fill level	0x00 10	CM	See 2.6.1.
Linkup detected	User-configurable	CM	This control message notifies the occurrence of a Linkup event on the Ethernet link. It is a single-shot message.
Linkdown detection	User-configurable	CM	This control message notifies the occurrence of a Linkdown event on the Ethernet link. It is a single-shot message.

Control Message			
Name	ID	Source	Description
Time synchronization acquired	User-configurable	CM	This control message notifies the acquisition of the time-synchronization. It is a single-shot message.
Time synchronization lost	User-configurable	CM	This control message notifies the loss of the time-synchronization. It is a single-shot message.

Table 26: Control Message ID

2.7.5 CM Type

Name	ID
CM LIN Combo	0x02
CM CAN Combo	0x04
CM 100 High	0x06
CM Eth Combo	0x08
CM 1000 High	0x0A
Other	0xFF

Table 27: CM Type

2.7.6 Interface ID

The following table describes the value of the Interface ID field for the Status and Control messages.

Name	ID
Status Bus message	0x000FFF00
Status Bus/IVN message	0x000FFF01
Status Capture Module message	0x000FFF02
Control message	0x000FFF03

Table 28: Interface ID

3 Contact

If you have any questions regarding this document or our products, please contact us:

Technica Engineering GmbH
Leopoldstr. 236
80807 München
Germany

Info@technica-engineering.de

www.technica-engineering.de

4 Changelog

Version	Chapter	Description	Date
1.0	All	First release.	10.02.2020
1.1	2.4 and 2.6.1	References to "User Event" replaced with "Control Message".	12.02.2020
1.2	2.3.2, 2.3.3, 2.6.3 2.2.2.5 2.6.1 2.3	Description of the fields of Vendor Data (Status Bus/IVN messages and Status Capture Module Message) added. CM IP-address list generalized. References to "Length of vendor data" replaced with "Vendor Data Length" (VDL). Added value description to the TECMP Flexray Data flags. Added "Replay Data" Message type and frame format description. Aligned cyclic transmission of Status Message.	06.04.2020
1.3	2.6.3 2.6.5 2.3 2.3.2 2.3.1	Device ID table (switch position) updated with hexadecimal value. Capture Module naming updated. Cyclic time of Status Messages updated. Added upper bound length of 1400 Bytes. Added Linkup time to Status bus/IVN message payload. Added description of Status Configuration message vendor data. Added 1000 High Device ID.	27.05.2020
1.4	2.2.1 2.2.2.4 2.2.2.5	Added restriction on Timestamp. "CRC Error" Data Flag bit detailed for FlexRay. "Tx" Data Flag bit detailed. Added "BRS" bit to the CAN-FD Data frame. Added "CAS" bit to the FlexRay Data frame.	25.08.2020
1.5	1.1 – 2.1 2.3.1 2.3.3	Editorial changes. Status Configuration message cycle time updated from 1s to 60s. Field "Temperature" renamed to "Chassis Temperature". Update from UINT8 to SINT8. Added "Silicon Temperature" field.	13.11.2020
1.6	Whole document	The major following changes were performed: <ul style="list-style-type: none"> Added new "Replay Data" feature. "Channel ID" renamed to "Interface ID". "CM ID" renamed to "Device ID". Logging feature extended: Source Data and Data Flags updated for CAN(-FD) and FlexRay. 	13.12.2021