TECMP

Technically Enhanced Capture Module Protocol

USER MANUAL

July 2022

Document version: 1.7

CONTENT

1 GENERAL INFORMATION

1.1 Introduction

The following protocol description is used by the Capture Modules (CMs) 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

Not all features are implemented in all CMs. CM individual restrictions are documented inside the user manuals.
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.

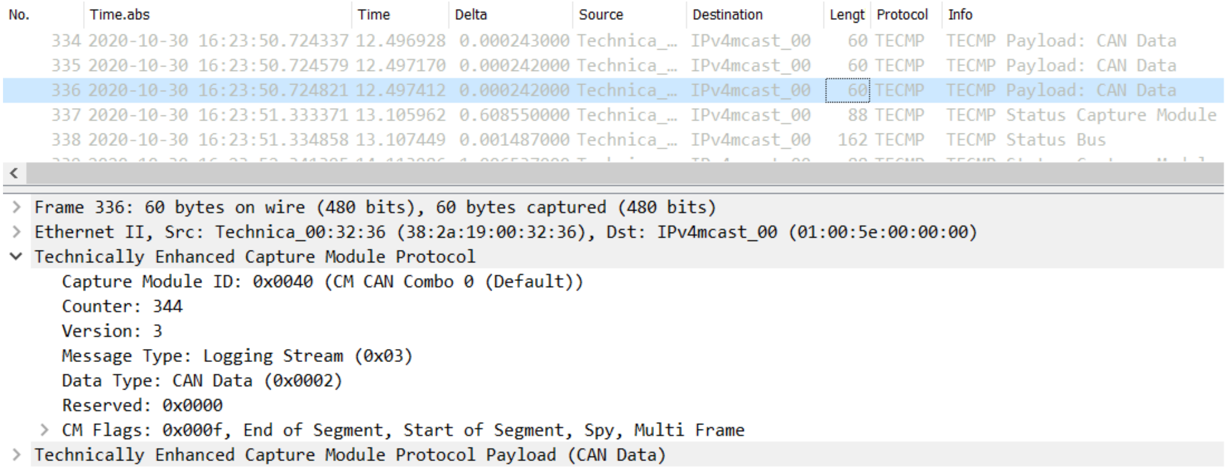
1.2 Purpose of the document

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	Lengt	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)
v 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.

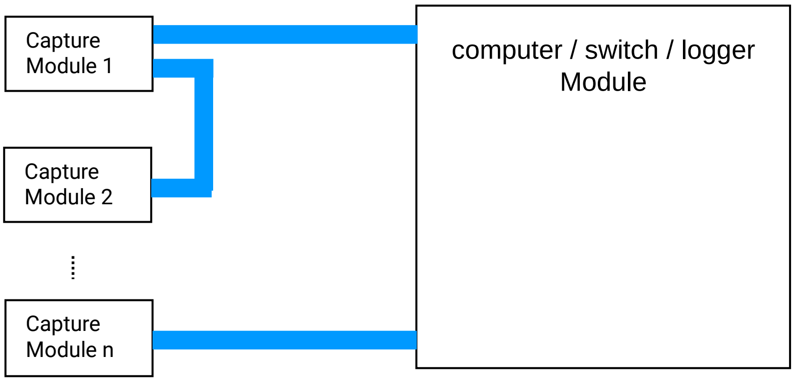


Figure 2: Overview of a system setup

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
CAN	Controller Area Network
CM	Capture Module
Control Message	A method to trigger events. Events can be triggered by a CM or can be received by a CM. A Control Message is identified by a given Control Message ID (see 2.7.4).
CRC	Cyclic redundancy check
EoS	End of Segment
Eth	Ethernet
FCS	Frame Check Sequence
FR	FlexRay
HW	Hardware
ID	Identifier

Abbreviation	Definition / Explanation
IVN	In-Vehicle Network
LIN	Local Interconnect Network
LSB	Least Significant Byte(s)
MSB	Most Significant Byte(s)
POC	Protocol Operation Control
PRE	Preamble
PTP	Precision Time Protocol
Rx	Reception of an IVN data message by the CM resulting in the generation and transmission of a message of type “Logging Stream” by the CM to the sink. Also described as capturing or recording (IVN) data.
SFD	Start of Frame Delimiter
SoS	Start of Segment
Tx	Transmission of an IVN data message by the CM as a result of the reception of a message of type “Replay Data” by the CM from the sink. Also described as sending or replaying (IVN) data.
Status Message	A dedicated message sent by the CM in direction of the Ethernet Capture interface (network, computer, logger).
TECMP	Technically Enhanced Capture Module Protocol

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.

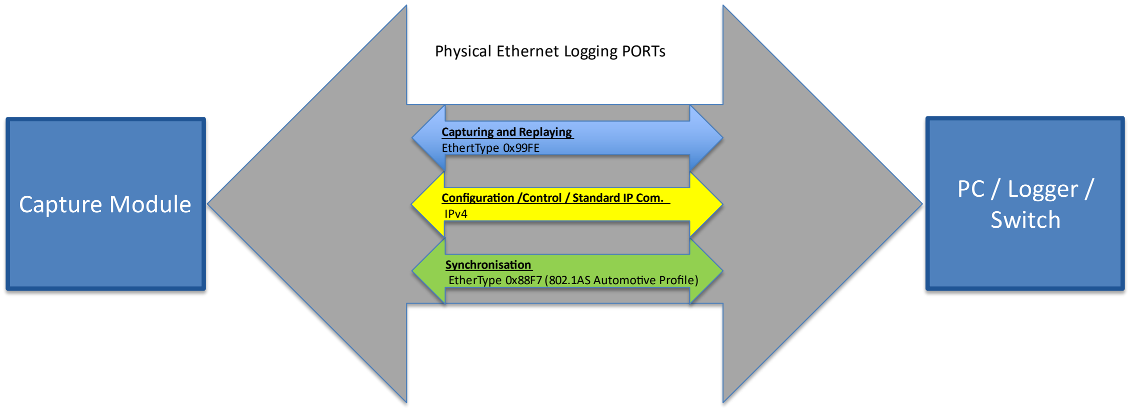


Figure 3: Logical/physical Interfaces

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).

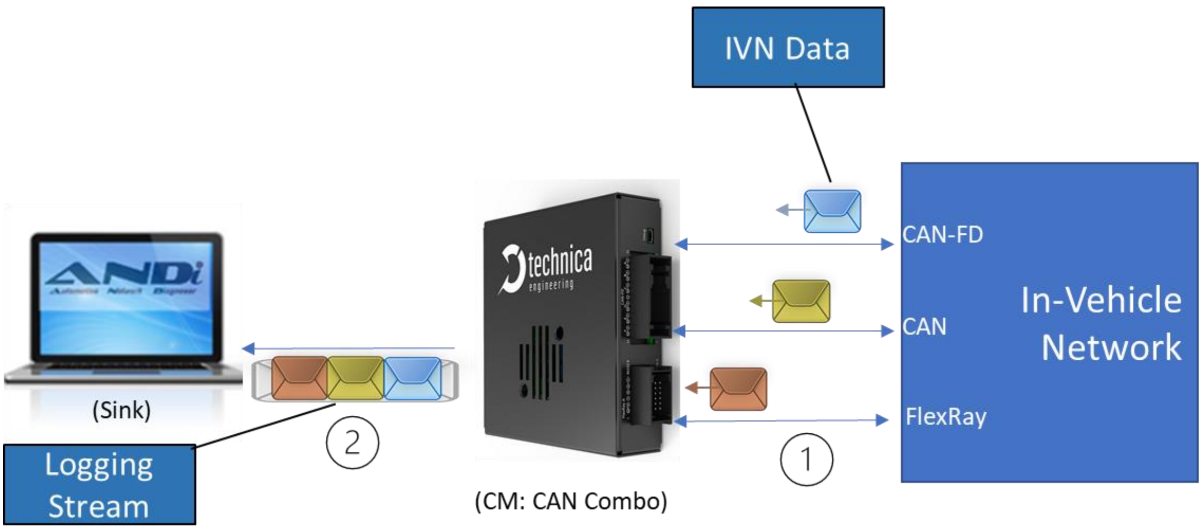


Figure 4: Capturing IVN Data

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…)

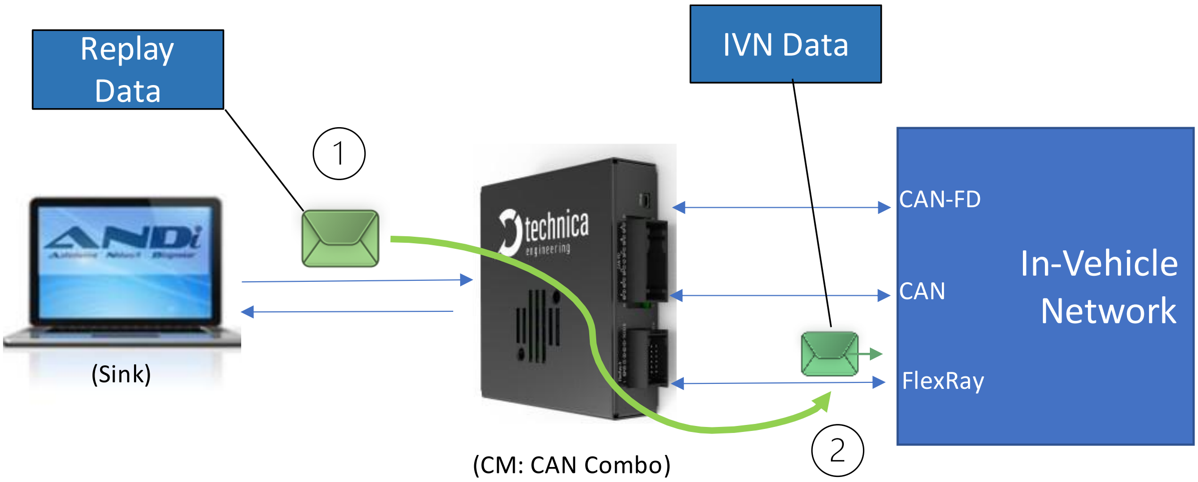


Figure 5: Replaying data onto the IVN

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 change of their payload or header. 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	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.

Field Name	Description
	• 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
Data	see 2.4.3, 2.4.4 and 2.4.5
FCS	CRC checksum, calculated according to IEEE Standard 802.3

Table 2: TECMP message fields description

2.4.1 TECMP Header

The header structure is identical for all TECMP messages exchanged between CM and sink. Figure 7 shows the position of the header compared to the IVN data position (Data). The header can be separated in two parts:

- A global header section (12 Bytes) from the field Device ID to the field Device Flags.
- A data header section (16 Bytes) from the field Interface ID to the field Data Flags.

<table>
<tr><td colspan="4">4 Bytes</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="2">Device ID</td><td colspan="2">Counter</td></tr>
<tr><td>Version</td><td>Message Type</td><td colspan="2">Data Type</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">Device Flags</td></tr>
<tr><td colspan="4">Interface ID</td></tr>
<tr><td colspan="4">Timestamp Bit 63..32</td></tr>
<tr><td colspan="4">Timestamp Bit 31..0</td></tr>
<tr><td colspan="2">Length</td><td colspan="2">Data Flags</td></tr>
<tr><td colspan="4">Data</td></tr>
<tr><td colspan="4">…</td></tr>
</table>

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

Figure 8 shows the use of both sections when multiple Data are transmitted into a single TECMP message. In this case, the global header section is used as in Figure 7 while the data header section is repeated prior to every new Data. For more information, refer to the Chapter 2.4.5,

<table>
<tr><td colspan="4">4 Bytes</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="2">Device ID</td><td colspan="2">Counter</td></tr>
<tr><td>Version</td><td>Message Type</td><td colspan="2">Data Type</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">Device Flags</td></tr>
<tr><td colspan="4">Interface ID</td></tr>
<tr><td colspan="4">Timestamp Bit 63..32</td></tr>
<tr><td colspan="4">Timestamp Bit 31..0</td></tr>
<tr><td colspan="2">Length</td><td colspan="2">Data Flags</td></tr>
<tr><td colspan="4">Data</td></tr>
<tr><td colspan="4">…</td></tr>
<tr><td colspan="4">Interface ID</td></tr>
<tr><td colspan="4">Timestamp Bit 63..32</td></tr>
</table>

<table>
<tr><td colspan="2">Timestamp Bit 31..0</td></tr>
<tr><td>Length</td><td>Data Flags</td></tr>
<tr><td colspan="2">Data</td></tr>
<tr><td colspan="2">…</td></tr>
</table>

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.*

NOTE: All number types are UINT and encoded in big-endian, if not specified otherwise.

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.

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>Device ID</td><td>UINT16</td><td>The first upper Byte is set to 0x00.
The second Byte defines the CM unique ID. This ID can be used to identify different devices if they are used in the same measurement network (see Table 25). Up to 256 different devices can be uniquely identified inside the measurement network.

When capturing data (Message Type = “Logging Stream”):
This field identifies the CM from which the TECMP frame was sent: its source.
When replaying data (Message Type = “Replay data”):
Unused. This field is set to: “0x0000” for every message.</td></tr>
<tr><td>Counter</td><td>UINT16</td><td>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.
When replaying data (Message Type = “Replay data”):
This field is set to: “0x000” for every message. Received requests are no counted by the CM.</td></tr>
<tr><td>Version</td><td>UINT8</td><td>Version of the TECMP protocol used. This value is set to 3.</td></tr>
<tr><td>Message Type</td><td>UINT8</td><td>Type of TECMP Message: Status, Control, …. See Table 23.</td></tr>
<tr><td>Data Type</td><td>UINT16</td><td>Description of the Type of Data (e.g., CAN, LIN, Flexray, Ethernet, Voltage, Current, …). See Table 24.</td></tr>
<tr><td>Reserved</td><td>UINT16</td><td>Reserved for future extension of the protocol: Set to 0x00 and ignored on receiving.</td></tr>
<tr><td>Device Flags</td><td>UINT16</td><td>When capturing data (Message Type = “Logging Stream”):
This field provides additional information on the device.
<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 frames.</td></tr>
<tr><td>1</td><td>SoS (Start of Segment)</td><td>• 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.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>• 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>
<tr><td>15</td><td>Device overflow</td><td>The CM has lost data during transmission to the Sink.</td></tr>
</table>
When replaying data (Message Type = “Replay data”): Reserved: 0x0000.</td></tr>
<tr><td>Interface ID</td><td>UINT32</td><td>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.</td></tr>
<tr><td>Timestamp</td><td>UINT64</td><td>This field provides the 62-bits timestamp information reference of the TECMP frame, as well as a 2-bits timestamp flag section.
<table>
<tr><th>Length (bits)</th><th>Description</th></tr>
<tr><td>2</td><td>Timestamp flag section.</td></tr>
<tr><td>62</td><td>Timestamp information.</td></tr>
</table>
<table>
<tr><th>Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>61 - 0</td><td>Timestamp</td><td>The time since Thursday, January 1, 1970, 00:00 UTC. The resolution is 1ns.
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.
When replaying data (Message Type = “Replay data”):
• 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.</td></tr>
<tr><td>62</td><td>Timestamp Recalculation flag</td><td>When capturing data (Message Type = “Logging Stream”):
• ‘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.
When replaying data (Message Type = “Replay data”): Reserved: ‘0’</td></tr>
</table></td></tr>
</table>

Field Name	Data Type	Description
		63 — Synchronization flag — **When capturing data** (Message Type = "Logging Stream"): • '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 375 ms. • '0': ongoing synchronization with the time master. **When replaying data** (Message Type = "Replay data"): Reserved: '0'
Length	UINT16	Length of the Data field. The length of the Length and Data Flags fields are not included. Any additional 0-padding 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	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…). See Data type specific definition
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 SFD and FCS).

<table>
<tr><td colspan="4">4 Bytes</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="4">DST MAC (MSB)</td></tr>
<tr><td colspan="2">DST MAC (LSB)</td><td colspan="2">SRC MAC (MSB)</td></tr>
<tr><td colspan="4">SRC MAC (LSB)</td></tr>
<tr><td colspan="4">Outer VLAN (Optional)</td></tr>
<tr><td colspan="4">Inner VLAN (Optional)</td></tr>
<tr><td colspan="2">EtherType</td><td colspan="2">Device ID</td></tr>
<tr><td colspan="2">Counter</td><td>Version</td><td>Message Type</td></tr>
<tr><td colspan="2">Data Type</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="2">Device Flags</td><td colspan="2">Interface ID (MSB)</td></tr>
<tr><td colspan="2">Interface ID (LSB)</td><td colspan="2">Timestamp (MSB)</td></tr>
<tr><td colspan="4">Timestamp</td></tr>
<tr><td colspan="2">Timestamp (LSB)</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Data Flags</td><td colspan="2">Data</td></tr>
<tr><td colspan="4">…</td></tr>
</table>

Figure 9: TECMP frame overview

2.4.3 Capture specification (Message Type = 0x3 “Logging Stream”)

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.

In this chapter, the Message Type value is 0x3.
In the following, if a special reception condition like an error for which no valid payload can be captured or like a given pattern (WUP for LIN or WUS for FlexRay or instance) then all the field of the Data section are set to 0x0 (for instance the CAN ID for CAN, any payload length…).

2.4.3.1 TECMP CAN Rx (Data Type = 0x2)

(a) Data Flags

These flags provide information regarding the CAN bus and reception condition.

Bit position	15	14	13	12 - 8	7	6
Description	Overflow	Tx	CRC error	RESERVED	EOF error	ACK_DEL error

Bit position	5	4	3	2	1	0
Description	CRC_DEL error	BIT_STUFF error	ERR	IDE	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.
- IDE
  - ‘1’: 29 bit ID.
  - ‘0’: 11 bit ID.
- 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.
- TX
    - ‘1’: the frame logged on the interface was transmitted by the CM itself.
    - ‘0’: the frame logged on the interface was not transmitted by the CM itself.
- Overflow
    - ‘1’: one or more messages on this interface were lost during recording.
    - ‘0’: no message loss was detected on this interface during recording.

(b) Data

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>CAN ID</td><td>UINT32</td><td>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></td></tr>
<tr><td>Payload Length</td><td>UINT8</td><td>Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8.</td></tr>
<tr><td>Payload</td><td>UINT8 [0-8]</td><td>Payload Bytes (up to 8 Bytes), depending on the Payload Length.</td></tr>
<tr><td>CRC</td><td>UINT8 [2 Bytes]</td><td>CRC (15-bits).
The unused upper bit is filled with ‘0’..</td></tr>
</table>

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

(a) Data Flags

These flags provide information regarding the CAN-FD bus and reception condition.

Bit position	15	14	13	12 - 9	8	7
Description	Overflow	Tx	CRC error	RESERVED	EOF error	ACK_DEL error

Bit position	6	5	4	3	2	1	0
Description	CRC_DEL error	BIT_STUFF error	BRS	ERR	IDE	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.
- IDE
    - ‘1’: 29 bit ID.
    - ‘0’: 11 bit ID.
- 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.
- RESERVED
  - '0': unused.
- CRC error
  - '1': a CRC error was found.
  - '0': no CRC error was found.
- TX
  - '1': the frame logged on the interface was transmitted by the CM itself.
  - '0': the frame logged on the interface was not transmitted by the CM itself.
- Overflow
  - '1': one or more messages on this interface were lost during recording.
  - '0': no message loss was detected on this interface during recording.

(b) Data

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>CAN ID</td><td>UINT32</td><td>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></td></tr>
<tr><td>Payload Length</td><td>UINT8</td><td>Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8, 12, 16, 20, 24, 32, 48, 64.</td></tr>
<tr><td>Payload</td><td>UINT8 [0-64]</td><td>Bytes of the payload (up to 64 Bytes), depending on the Payload Length.</td></tr>
<tr><td>CRC</td><td>UINT8 [3 Bytes]</td><td>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',</td></tr>
</table>

2.4.3.3 TECMP LIN Rx (Data Type = 0x4)

(a) Data Flags

These flags provide information regarding the LIN bus and reception condition.

Bit position	15	14	13	12 – 11	10	9	8
Description	Overflow	Tx	Checksum error	RESERVED	SLEEP	Short WUP	WUP

Bit position	7 - 3	2	1	0
Description	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’: no issue detected.
- RESERVED:
  - ‘0’: unused.
- WUP:
  - ‘1’: detection of a wake up signal (including a dominant state between 250 µs and 5 ms for LIN 2.0 or exactly 8 dominant bits for LIN 1.3).
  - ‘0’: no wake up detection.
- Short WUP:
  - ‘1’: detection of a short wake up signal (including a dominant state between 150 µs and 249 µs for LIN 2.0).
  - ‘0’: no short wake up detection. This is always the case for LIN 1.3.
- SLEEP:
  - ‘1’: detection of a go-to-sleep command.
  - ‘0’: no go-to-sleep command detection.
- 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.
- TX
  - ‘1’: the frame logged on the interface was transmitted by the CM itself.
  - ‘0’: the frame logged on the interface was not transmitted by the CM itself.
- Overflow
  - ‘1’: one or more messages on this interface were lost during recording.
  - ‘0’: no message loss was detected on this interface during recording.

(b) Data

Field Name	Data Type	Description
LIN ID	UINT8	Frame identifier. • 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).
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 Rx (Data Type = 0x8)

(a) Data Flags

These flags provide information regarding the FlexRay bus and reception condition.

Bit position	15	14	13	12	11 - 6
Description	Overflow	Tx	Frame CRC error	Header CRC error	RESERVED

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.
- TX
  - ‘1’: the frame logged on the interface was transmitted by the CM itself.
  - ‘0’: the frame logged on the interface was not transmitted by the CM itself.
- Overflow
  - ‘1’: one or more messages on this interface were lost during recording.
  - ‘0’: no message loss was detected on this interface during recording.

(b) 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	Header CRC.
Frame CRC	UINT8 [3 Bytes]	Frame CRC.

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

(a) Data Flags

These flags detail the UART / RS232 bus and reception condition.

Bit position	15	14	13 - 4	3 - 1	0
Description	RESERVED	Tx	RESERVED	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.
- RESERVED:
  - '0': unused.
- TX
  - '1': the frame logged on the interface was transmitted by the CM itself.
  - '0': the frame logged on the interface was not transmitted by the CM itself.

(b) Data

Several data symbols of an RS232 interface are collected inside the 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 Rx (Data Type = 0x20)

(a) Data Flags

These flags provide information regarding the Analog bus.

Bit position	15	14 - 11	10 - 9	8 – 7	6 – 5	4 – 2	1 - 0
Description	RESERVED	Sample Time	RESERVED	Factor	RESERVED	Unit	RESERVED

- RESERVED:
  - '0': unused.
- Unit:
  - "000": Volt (V).
  - "001": Ampere (A).
  - "010": Watt (W).
  - "011": Ampere hour (Ah).
  - "100": Degree Celsius (°C).
- Factor:
  - "00": 0,1.
  - "01": 0,01.
  - "10": 0,001.
  - "11": 0,0001.

- Sample Time
  - "0000": Reserved
  - "0001": 2500ms.
  - "0010": 1000ms.
  - "0011": 500ms.
  - "0100": 250ms.
  - "0101": 100ms.
  - "0110": 50ms.
  - "0111": 25ms.
  - "1000": 10ms.
  - "1001": 5ms.
  - "1010": 2,5ms.
  - "1011": 1ms.
  - Others: undefined.

(b) 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	INT16	Contains the first value of the recording.
…	INT16	…
Sample N	INT16	Contains the N-th value of the recording.

2.4.3.7 TECMP Ethernet Rx (Data Type = 0x80)

(a) Data Flags

These flags provide information regarding the reception condition on the Ethernet interface.

Bit position	15	14	13	12 - 0
Description	Overflow	Tx	CRC error	RESERVED

- RESERVED:
  - '0': unused.
- CRC error:
  - '1': a CRC error was detected.
  - '0': no CRC error was detected.
- TX
  - '1': the frame logged on the interface was transmitted by the CM itself.
  - '0': the frame logged on the interface was not transmitted by the CM itself.
- Overflow
  - '1': one or more messages on this interface were lost during recording.
  - '0': no message loss was detected on this interface during recording.

(b) 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.

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

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.
FCS	UINT32	Frame Check Sequence.

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

2.4.4 Replay specification (Message Type = 0xA "Replay Data")

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 (0x0)			
DST MAC (LSB): unused (0x0)		SRC MAC (MSB)	
SRC MAC (LSB)			
Outer VLAN (Optional)			
Inner VLAN (Optional)			
EtherType = 0x99FE		Device ID: unused (0x0)	
Counter = 0x0		Version = 0x3	Message Type = 0xA
Data Type = 0x2 (example)		Reserved	
Device 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		Data	
...			

Figure 11: Replay Data example

In this example, a CAN frame (Data Type = 0x2) shall be replayed (Message Type = 0xA) given the current protocol Version (0x3). The frame shall be sent as soon as possible (Timestamp = 0x0). The supported values of the Data Flags and Data for every Data Type are defined in the next subchapters.

In this chapter, the Message Type value is 0xA.

2.4.4.1 TECMP CAN Tx (Data Type = 0x2)

(a) Data Flags

These flags control the transmission properties on the CAN bus.

Bit position	15 – 14	13	12 - 8	7
Description	RESERVED	Use CRC	RESERVED	EOF error
Value	0x0	See below	0x0	See below

Bit position	6	5	4	3 – 0
Description	ACK_DEL error	CRC_DEL error	BIT_STUFF error	RESERVED
Value	See below	See below	See below	0x0

- RESERVED:
    - ‘0’: unused.
- BIT_STUFF error
    - 1’: a bit stuff error will be generated.
    - ‘0’: bit-stuffing will not be interrupted.
- CRC_DEL error:
    - ‘1’: a CRC delimiter error will be injected.
    - ‘0’: no CRC delimiter error will be injected.
- ACK_DEL error:
    - ‘1’: an acknowledgment delimiter error will be injected.
    - ‘0’: no acknowledgment delimiter error will be injected.
- EOF_error:
    - ‘1’: an end of frame field error will be injected.
    - ‘0’: no end of frame field error will be injected.
- Use CRC
    - ‘1’: the provided Checksum value will be used for the transmission.
    - ‘0’: the provided Checksum value will not be used. The device will compute the correct value for the transmission.

(b) Data

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>CAN ID (inclusive IDE bit)</td><td>UINT32</td><td>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></td></tr>
<tr><td>Payload Length</td><td>UINT8</td><td>Length of the CAN payload (in Bytes): 0, 1, 2, 3, 4, 5, 6, 7, 8.</td></tr>
<tr><td>Payload</td><td>UINT8 [0-8]</td><td>Payload Bytes (up to 8 Bytes), depending on the Payload Length.</td></tr>
<tr><td>CRC</td><td>UINT8 [2 Bytes]</td><td>Depending on the value of the Use CRC bit (13) inside the Data Flags:
• ‘0’</td></tr>
</table>

		○ The provided CRC value will not be used. The device will compute the correct value for the transmission. • '1' ○ The provided CRC (15-bits) value will be used for the transmission. 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 = 0x3)

(a) Data Flags

These flags control the transmission properties on the CAN-FD bus.

Bit position	15 - 14	13	12 - 9	8	7
Description	RESERVED	Use CRC	RESERVED	EOF error	ACK_DEL error
Value	0x0	See below	0x0	See below	See below

Bit position	6	5	4	3 - 0
Description	CRC_DEL error	BIT_STUFF error	BRS	RESERVED
Value	See below	See below	See below	0x0

- RESERVED:
  - '0': unused.
- 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 is not switched inside the FD Frame.
- BIT_STUFF_error
  - 1': a bit stuff error will be generated.
  - '0': bit-stuffing will not be interrupted.
- CRC_DEL error:
  - '1': a CRC delimiter error will be injected.
  - '0': no CRC delimiter error will be injected.
- ACK_DEL error:
  - '1': an acknowledgment delimiter error will be injected.
  - '0': no acknowledgment delimiter error will be injected.
- EOF error:
  - '1': an end of frame field error will be injected.
  - '0': no end of frame field error will be injected.
- Use CRC
  - '1': the provided CRC value will be used for the transmission.
  - '0': the provided CRC value will not be used. It will be computed by the device.

(b) Data

<table>
<tr><th>Field Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>CAN ID (inclusive IDE bit)</td><td>UINT32</td><td>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></td></tr>
</table>

Field Name	Data Type	Description
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]	Depending on the value of the Use CRC bit (13) inside the Data Flags: • '0' ○ The provided CRC value will not be used. The device will compute the correct value for the transmission. • '1' ○ The provided CRC (17/21-bits) value will be used for the transmission. The unused upper bit shall be filled with '0'. 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 = 0x4)

(a) Data Flags

These flags control the transmission properties on the LIN bus.

Bit position	15 - 14	13	12 - 10	9
Description	RESERVED	Use Checksum	RESERVED	Short WUP
Value	0x0	See Below	0x0	See below

Bit position	8	7 – 2	1	0
Description	WUP	RESERVED	Use Parity	RESERVED
Value	See below	0x0	See below	0x0

- RESERVED:
  - '0': unused.
- Use Parity
  - '1': The provided parity bits will be used for transmission.
  - '0': The parity bits will be computed by the device.
- WUP:
  - '1': Transmission of a wake up signal (including a dominant state between 250 µs and 5 ms for LIN 2.0 or exactly 8 dominant bits for LIN 1.3).
  - '0': No wake up transmission.
- Short WUP:
  - '1': Transmission of a short wake up signal (including a dominant state between 150 µs and 249 µs). Only supported for LIN 2.0.
  - '0': No short wake up transmission.
- Use Checksum
  - '1': The provided Checksum will be used for transmission.
  - '0': The Checksum will be computed by the device.

(b) Data

Field Name	Data Type	Description
LIN ID	UINT8	Frame identifier: • Bit 7 – Bit 6 : Depending on the value of the "Use Parity" bit (1) inside the Data Flags: ○ '0': The Parity bits will be computed by the device.

		○ '1': The provided Parity bits will be used for the transmission. • 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	Depending on the value of the Use Checksum bit (13) inside the Data Flags: • '0': The Checksum will be computed by the device. • '1': The provided Checksum value (0x00-0xFF) will be used for the transmission. The unused upper bit shall be filled with '0'.

2.4.4.4 TECMP Flexray Tx (Data Type = 0x8)

(a) Data Flags

These flags control the transmission properties on the FlexRay bus.

Bit position	15 - 13	12	11 - 10	9 - 7	6 - 0
Description	RESERVED	Use 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 – 0x7: Undefined.
- Use Header CRC
  - '1': the provided Checksum value (0x0000-0x07FF) will be used for the transmission. Any other value (0x800 – 0xFFF) 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) 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	Depending on the value of the Use Header CRC bit (12) inside the Data Flags:
		• ‘0’
		  ○ The provided Header CRC value will not be used. The device will compute the correct value for the transmission.
		• ‘1’
		  ○ 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.

2.4.5 Packetizing IVN messages

To optimize the performance of the communication, it is possible to limit and reuse the first header section (from the field Device ID to the field Device Flags) only once over multiple IVN Data. As introduced in 2.4.1, the solution consists in transfering at once several IVN messages packetized in a single TECMP frame. The principle applies for both the transfer to the attached sinks (for capturing data) and to the CM (for replaying data). The first 12 Bytes of the TECMP Header are inserted only once, at the beginning of the packetized IVN messages. The remaining header section (16 Bytes, from the field Interface ID to the field Data Flags) is inserted once before each captured IVN message. No padding is used between the successive messages inside the TECMP message.

Figure 12 shows a TECMP frame with a single CAN message contained. Here, paketization is not used.

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

Figure 12: Transport of a single CAN message without packetization.

By opposition to Figure 12, Figure 13 shows the paketization of four CAN messages into a TECMP frame.

Device ID	Counter	Version	Message Type	Data Type	Reserved	Device 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: Transport of a multiple CAN messages with packetization.

As the identification of the Message Type belongs to the first header section, it applies globally to all the packetized IVN messages. It means that only messages of the same Message Type can be packetized together: Logging stream with Logging Stream or Replay Data with Replay Data. There is no packetization of Control Messages.
Similarly, only messages of the same Data Type (CAN with CAN for instance) can be packetized together.

When capturing data:

- If, despite the packetization 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 packetized because this information can be retrieved based on the value of the length (header field) of the Data and the overall buffered length of the TECMP message.

When replaying data:

- Only FlexRay messages can be packetized. Other protocols are not supported. Successive individual TECMP messages shall be used instead.

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.

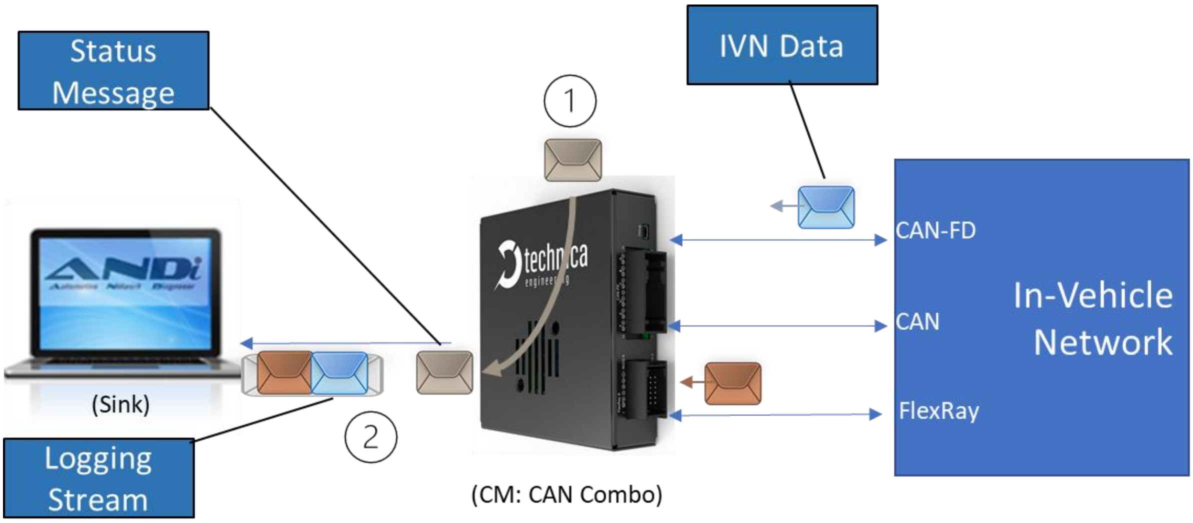


Figure 14: Status Message generation

Three different Status Messages are currently supported by CMs:

- Status Configuration
- Status Bus/IVN
- Status Device

The Status Messages use the same TECMP Header as other TECMP messages presented in Figure 7. Figure 15 shows where the Status Message payload is located at he end of the header.

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

Figure 15: Status Messages TECMP overview

Note: As introduced in Table 3, the value of the length field correspond to the length of the entire status message payload.

2.5.1 Status Configuration message

2.5.1.1 Transmission condition

Status Configuration messages are generated cyclically every 60 seconds.

2.5.1.2 Length limitation and Segmentation

Due to the potentially large CM configuration, a dedicated segmentation feature is supported. The maximum length of any Status configuration Message or segmented Status configuration Message is 800 Bytes. If the payload to be transferred does not fit in a single Message, the payload shall be segmented and transferred in multiple and successive Status Messages, each carrying one segment.

2.5.1.3 Format

The payload is presented in the next Tables.

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

Table 5: Status Configuration message payload

Field Name	Length	Description
Vendor ID	1 Byte	Vendor ID - Technica Engineering: 0x0C
Device Version	1 Byte	Versioning of the device
Device Type	1 Byte	Type of device (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 device ID
Serial Number	4 Bytes	Serial number of the device
Vendor Data	n Byte(s)	Technica Engineering specific device configuration. The length n of this field is given by the value of the Vendor Data Length field.

Table 6: Status Configuration message 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	Device Flags	Interface ID	Time stamp	Length	Data Flags
Vendor ID	Device Version	Device Type	Reserved	Vendor Data Length	Device ID	Serial Number	Vendor Data			

Table 7: Status Configuration message overview

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

Vendor ID:

Technica Engineering	0x0C

Device 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 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

2.5.2.1 Transmission condition

Status Bus/IVN messages are generated cyclically every second. CMs with multiple IVN interfaces send the status of all active IVN interfaces.

2.5.2.2 Length limitation

The maximum length is 1400 Bytes.

2.5.2.3 Format

The payload is depicted in the next illustration.

4 Bytes			
0	1	2	3
Vendor ID	Device Version	Device 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
Device Version	1 Byte	Versioning of the device
Device Type	1 Byte	Type of device (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 device ID
Serial Number	4 Bytes	Serial number of the device.
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

Figure 16 respectively Figure 17 shows an overview for one respectively multiple bus/IVN interfaces:

Device ID	COUNTER	VERSION	Message Type	DATA TYPE	RESERVED	Device Flags	Interface ID	TIME STAMP	LENGTH	Data Flags
VENDOR ID	Device VERSION	Device 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

Device ID	COUNTER	VERSION	Message Type	DATA TYPE	RESERVED	Device Flags	Interface ID	TIME STAMP	LENGTH	Data Flags
VENDOR ID	Device VERSION	Device 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 the CM Ethernet Combo, CM 100 High and CM 1000 High variants:

2 Bytes	
0	1
Link Status	Link Quality
Linkup Time	

Table 14: Status Bus/IVN message Vendor Data format for the CM Ethernet Combo, CM 100 High and CM 1000 High variants.

Field Name	Data Type	Description
Link Status	UINT8	Flag to indicate whether there is linkUp or not in the corresponding channel during generation of the status message: x00: No x01: Yes
Link Quality	UINT8	Quality of the link of the corresponding channel during generation of the status message, 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 500 ms after power up. • 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 Device message

2.5.3.1 Transmission condition

Status Device messages are generated cyclically every second.

2.5.3.2 Length limitation

The maximum length is 1400 Bytes.

2.5.3.3 Format

The payload is depicted in Table 16 and described in Figure 18.

<table>
<tr><td colspan="4">4 Bytes</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Vendor ID</td><td>Device Version</td><td>Device Type</td><td>Reserved</td></tr>
<tr><td colspan="2">Vendor Data length (VDL)</td><td colspan="2">Device ID</td></tr>
<tr><td colspan="4">Serial Number</td></tr>
<tr><td colspan="4">Vendor Data</td></tr>
</table>

Table 16: Status Device Message payload overview

Field Name	Length	Description
Vendor ID	1 Byte	Vendor ID – Technica Engineering: 0x0C
Device Version	1 Byte	Versioning of the device.
Device 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 device.
Serial Number	4 Bytes	Serial number of the device.
Vendor Data	n Byte(s)	Specific status information (e.g.: temperature processor, or device’s error conditions)

Figure 18: Description of the Status device message payload fields

Vendor Data - for all device Variants:

<table>
<tr><td colspan="4">4 Bytes</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Reserved</td><td colspan="3">SW Version</td></tr>
<tr><td colspan="2">HW Version</td><td>Buffer fill level</td><td>Buffer overflow</td></tr>
<tr><td colspan="4">Buffer Size</td></tr>
<tr><td colspan="4">Lifecycle</td></tr>
<tr><td colspan="4">Lifecycle</td></tr>
<tr><td>Voltage Integer part</td><td>Voltage fractional part</td><td>Chassis Temperature</td><td>Silicon Temperature</td></tr>
</table>

Table 17: Status Device message Vendor Data overview

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. First Byte is reprecented as decimal Second Byte is reprecented as hexadecimal E.g. 0x03.0x1A -> v3.1a
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. • 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. • 0x00 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. • 0x00 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 18: Status Device message Vendor Data fields description

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).

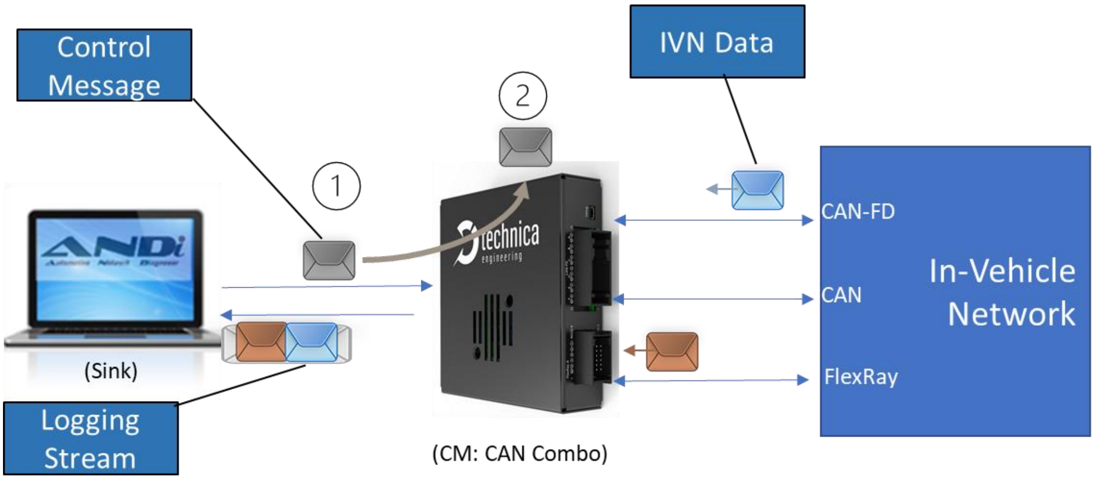


Figure 19: Reception of a Control Message by the CM.

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).

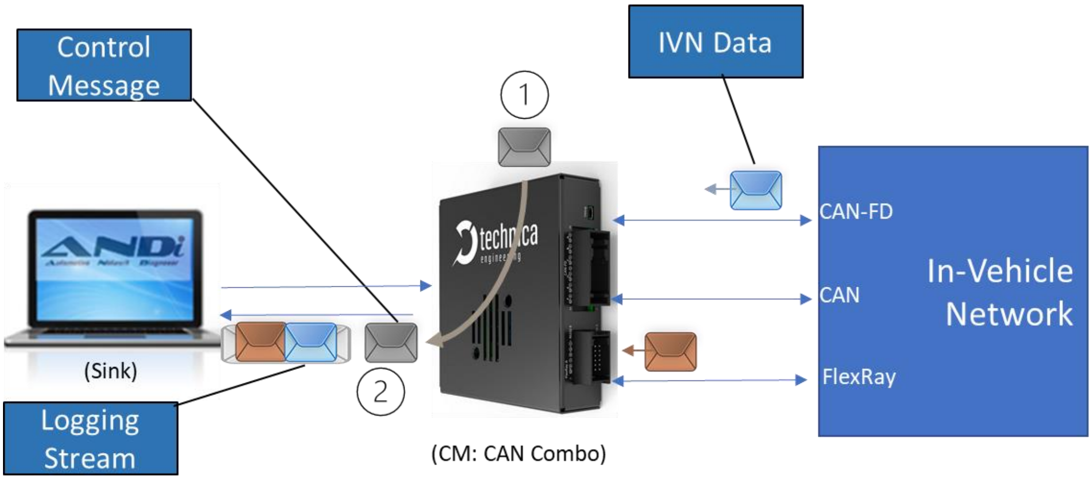


Figure 20: Generation of a Control Message by the CM.

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 Control Messages use the same TECMP Header as other TECMP messages presented in Figure 7. Figure 21 shows where the Control Message payload is located at he end of the header.

Device ID	Counter	Version	Message Type	Data Type	Reserved	Device Flags	Interface ID	Time stamp	Length	Data Flags	Control Message payload

Figure 21: Control Messages TECMP overview

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	
Payload Data (optional)			

Table 19: Control message payload overview

Note: As introduced in Table 3, the value of the length field correspond to the length of the entire control message payload: Device ID, Control Message ID and the optional Payload Data field.

Field Name	Length	Description
Device ID	1 Byte	Identifies the sender of the Control Message (see Table 25).
Control Message ID	1 Byte	Identifier of the specific Control Message (see Table 26).
Payload Data	n Byte(s)	Optional additional information, whose length can be computed based on the value of the Length field.

Table 20: Control message fields description

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

Device ID	Counter	Version	Message Type	Data Type	Reserved	Device Flags	Interface ID	Time-stamp	Length	Data Flags	Device ID	Control Message ID	Payload 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.

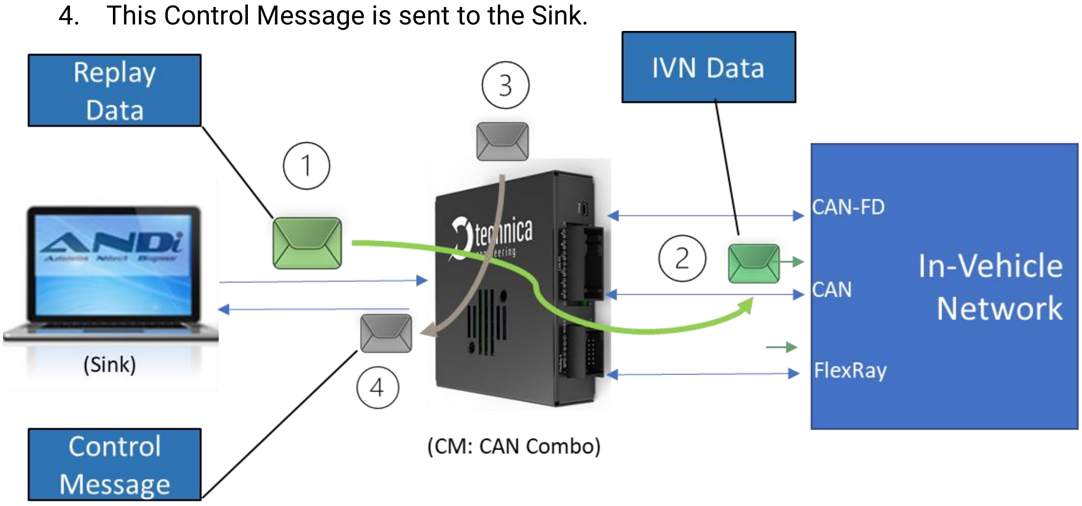


Figure 22: Generation of the CAN Replay fill level Control Message

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

Name	Length	Description
Device 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, 100.
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

2.6.2 FlexRay POC State

This Control Message is generated by the CM towards the Sink upon POC transition of an active FlexRay Interface.

Name	Length	Description
Device ID	UINT16	Sender of the Control Message.
Control Message ID	UINT16	Identifier of the specific Control Message.
Interface ID	UINT32	ID of the FlexRay Interface.
State	UINT8	0x0: CONFIG 0x1: DEFAULT_CONFIG 0x2: HALT 0x3: NORMAL_ACTIVE 0x4: NORMAL_PASSIVE 0x5: READY 0x6: STARTUP 0x7: WAKEUP Else: unknown

2.7 TECMP fields and IDs

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 Device	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 IVN data messages which were received by the CM. “Replay Data” identifies IVN data messages which are sent by the CM.

2.7.2 Data Type

Definition	Dec.	Hex.	Comment
CAN	02	0x0002	Data-Logging in combination with the "Logging Stream" Message Type 0x3 or Data-Replaying in combination with the "Replay Data" Message Type 0xA.
CAN-FD	03	0x0003	
LIN	04	0x0004	
Flexray	08	0x0008	
UART/RS232	16	0x0010	Data-Logging in combination with the "Logging Stream" Message Type 0x3.
Analog	32	0x0020	
Ethernet II (classical Ethernet frame)	128	0x0080	

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 dec. 2 Bytes	Device ID hex. 2 Bytes	Rotary switch position (0x)
LIN Combo				
LIN Combo 0	10.104.3.48	48	0x30	0
LIN Combo 1	10.104.3.49	49	0x31	1
LIN Combo 2	10.104.3.50	50	0x32	2
...	...	...	...	...
LIN Combo 15	10.104.3.63	63	0x3F	F
CAN Combo				
CAN Combo 0	10.104.3.64	64	0x40	0
CAN Combo 1	10.104.3.65	65	0x41	1
CAN Combo 2	10.104.3.66	66	0x42	2
...	...	...	...	...
CAN Combo 15	10.104.3.79	79	0x4F	F
100 High				
100 High 0	10.104.3.96	96	0x50	0
100 High 1	10.104.3.97	97	0x51	1
100 High 2	10.104.3.98	98	0x52	2
...	...	...	...	...
100 High 15	10.104.3.111	111	0x5F	F
Eth Combo				
Eth Combo 0	10.104.3.128	128	0x60	0
Eth Combo 1	10.104.3.129	129	0x61	1
Eth Combo 2	10.104.3.130	130	0x62	2
...	...	...	...	...
Eth Combo 15	10.104.3.143	143	0x6F	F
1000 High				
1000 High 0	10.104.3.144	144	0x70	0
1000 High 1	10.104.3.145	145	0x71	1
1000 High 2	10.104.3.146	145	0x72	2
...	...	...	...	...
1000 High 15	10.104.3.159	159	0x7F	F
Other				
Reserved	n/a	256 - 65535	0x100 – 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**	**Payload Data**	**Description**
Logger Ready	0x02	Logger	None	This control message notifies the logger readiness to receive data. It shall be sent cyclically every second.
CAN Replay fill level	0xE0	CM	See 2.6.1.	Fill level of the CAN transmission feature.
FlexRay POC State	0xE1	CM	See 2.6.2	Notification of the current FlexRay POC State.
Reserved	0xE2 – 0xFF	n/a	n/a	Reserved for future use.
Linkup detected	User-configurable	CM	None	This control message notifies the occurence of a Linkup event on the Ethernet link. It is a single-shot message.
Linkdown detection	User-configurable	CM	None	This control message notifies the occurence of a Linkdown event on the Ethernet link. It is a single-shot message.
Time synchronization acquired	User-configurable	CM	None	This control message notifies the acquisition of the time-synchronization. It is a single-shot message.
Time synchronization lost	User-configurable	CM	None	This control message notifies the loss of the time-synchronization. It is a single-shot message.

Table 26: Control Message ID

2.7.5 Device Type

The following table describes the value of the Device Type (1 Byte) field.

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: Device Type

2.7.6 Interface ID

The following table describes the value of the Interface ID (4 Bytes) field for the Status and Control messages.

Name	ID
Status Configuration message	0x000FFF00
Status Bus/IVN message	0x000FFF01
Status Device 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: • 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
1.6.1	2.4.3.3	Fixed LIN Data Flags section (bits 4-3).	15.12.2021
1.7	Whole document	Numerous changes, including the following: Removed wrong reference to “serialized JSON”. Added new FlexRay POC Status Control Message	29.07.2022

Version	Chapter	Description	Date
		Description of value 0 for “No Slave Response” in LIN Rx corrected. Rework (Maximum length of Status Configuration Message updated from 1400 to 800 Bytes, Segmentation) Extended functionalities for CAN(-FD) Rx and Tx, LIN Rx and Tx. Removed “Threshold” field for Analog Rx. Data Type of Samples corrected from UINT to INT. Size of the “Unit” field updated from two to three bits. Corrected timeout for “clock asynchronous” from 250ms to 375ms. “Source Data” renamed to Data. Fields containing “CM” or “Capture Module” renamed with “Device”. Status Capture Module message renamed to Status Device message. Description of HW Version in Status Device Message updated. Data Type and Control Message ID summary table updated.	