Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Association for Standardisation of
Automation and Measuring Systems

ASAM MCD-1 (XCP on CAN)

Universal Measurement and Calibration Protocol

CAN Transport Layer

Version 1.5.0

Date: 2017-11-30

Associated Standard

© by ASAM e.V., 2017

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Disclaimer

This document is the copyrighted property of ASAM e.V.
Any use is limited to the scope described in the license terms. The license terms can be viewed at www.asam.net/license

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Table of Contents

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

1 FOREWORD

XCP is short for Universal Measurement and Calibration Protocol. The main purpose is the data acquisition and calibration access from electronic control units. Therefore a generic protocol layer is defined. As transport medium different physical busses and networks can be used. For each authorized transport medium a separated transport layer is defined. This separation is reflected in standard document structure, which looks like follows:

- One Base Standard
- Associated Standards for each physical bus or network type

The Base Standard describes the following content:

- Protocol Layer
- Interface to ASAM MCD-2 MC
- Interface to an external SEED&KEY function
- Interface to an external Checksum function
- Interface to an external A2L Decompression/Decrypting function
- Example Communication sequences

This associated standard describes the XCP on CAN/CAN-FD transport layer.

The "X" inside the term XCP generalizes the "various" transportation layers that are used by the members of the protocol family. Because XCP is based on CCP the "X" shall also show that the XCP protocol functionality is extended in compare with CCP.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

2 INTRODUCTION

This standard describes how XCP is transported on CAN and CAN-FD as transport layer. In the following chapters all statements are valid for CAN and CAN-FD even if only CAN is mentioned. If there are any differences between CAN and CAN-FD both variants are described. It is shown how addressing shall be realized and the usage of the different communication models (see chapter 4.2). Also the content of the control field of the XCP message frame format is described. For details about the frame format structure please refer the base standard [1]. Chapter 5 shows the specific commands which are defined for the transport on CAN. The interface to the ASAM MCD-2 MC description file is described in chapter 9.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

3 RELATIONS TO OTHER STANDARDS

3.1 BACKWARD COMPATIBILITY TO EARLIER RELEASES

3.1.1 CAN TRANSPORT LAYER

This Transport layer uses the version number 1.5. This version number is represented as 16 bit value, where the high byte contains the major version (U) and low byte contains the minor version (V) number.

If this associated standard is modified in such a way that a functional modification in the slave's driver software is needed, the higher byte of its XCP Transport Layer Version Number will be incremented. This could be the case e.g. when modifying the parameters of an existing command or adding a new command to the specification.

If this associated standard is modified in such a way that it has no direct influence on the slave's driver software, the lower byte of its XCP Transport Layer Version Number will be incremented. This could be the case e.g. when rephrasing the explaining text or modifying the AML description.

The slave only returns the most significant byte of the XCP Transport Layer Version Number for the current Transport Layer in the response upon CONNECT.

3.1.2 THE COMPATIBILITY MATRIX

The Compatibility Matrix gives an overview of the allowed combinations of Protocol Layer and Transport Layer parts. For details about the Compatibility Matrix please refer the base standard [1].

3.2 REFERENCES TO OTHER STANDARDS

For details about the References to other standards please refer the base standard [1].

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

4 THE XCP TRANSPORT LAYER FOR CAN

4.1 ADDRESSING

The master can use `GET_SLAVE_ID` to detect all XCP slaves within a CAN network. The master has to send `GET_SLAVE_ID` with the XCP Broadcast CAN identifier.

XCP on CAN uses at least two different CAN identifiers for each independent slave: one identifier for the CMD and STIM packets and one identifier for the RES, ERR, EV, SERV and DAQ packets.
The STIM CAN Identifiers may be the same as the CMD CAN Identifier or may be assigned by the `SET_DAQ_ID` command.
The DAQ CAN Identifiers may be the same as the RES/ERR/EV/SERV CAN Identifier or may be assigned by the `SET_DAQ_ID` command.

The assignment of CAN message identifiers to the XCP objects CMD/STIM and RES/ERR/EV/SERV/DAQ is defined in the slave device description file (e.g. [2]), which is used to configure the master device. It is recommended that the bus priority of the message objects be carefully determined in order to avoid injury to other real-time communication on the bus. Also, the CMD/STIM should obtain higher priority than the RES/ERR/EV/SERV/DAQ.

The most significant bit (of the 32-bit value) set, indicates a 29 bit CAN identifier.
Bit 30 of the CAN-ID indicates CAN-FD frames.

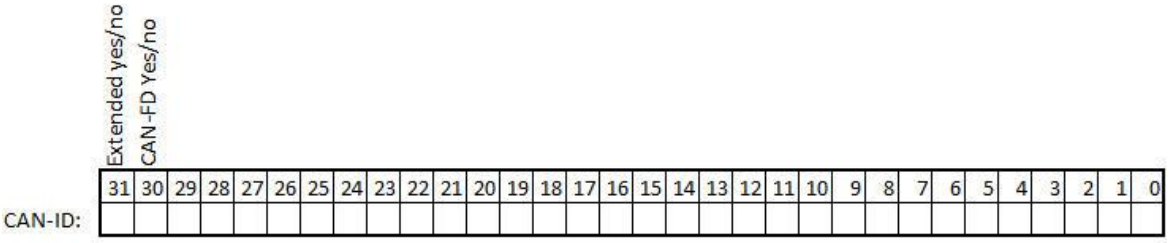

Figure 1 CAN-ID

4.2 COMMUNICATION MODEL

XCP on CAN makes use of the standard communication model.
The block transfer communication model is optional.
The interleaved communication model is not allowed.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

4.2.1 MASTER BLOCK TRANSFER WITH INCREMENTAL CAN-ID

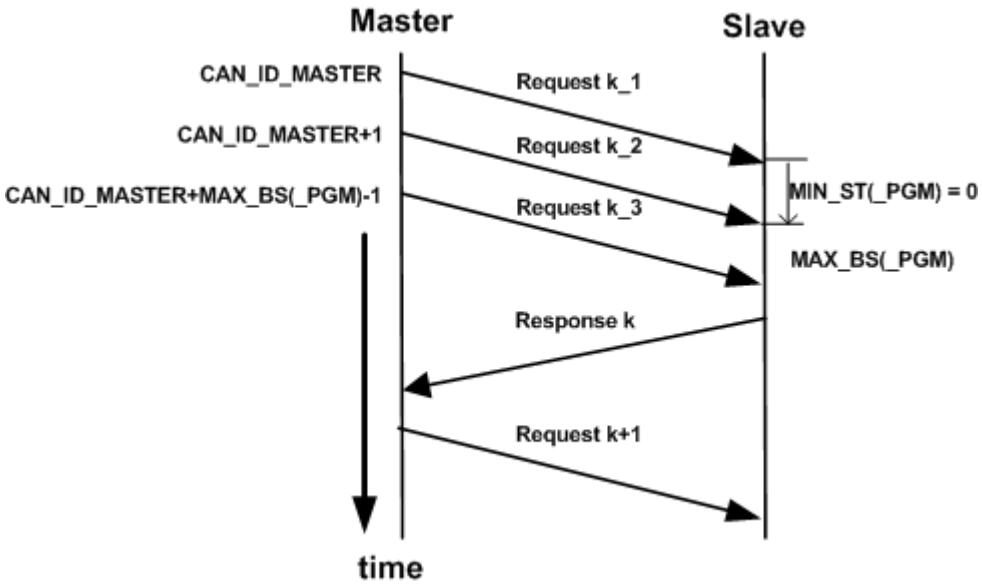

Figure 2 Master block transfer with incremental CAN-ID

For block transfer from master to slave during a download or programming sequence, performance can be increased if the master uses different CAN-IDs for every request and the communication is done with MIN_ST(_PGM) = 0.
With `CAN_ID_MASTER_INCREMENTAL`, the slave can inform the master that for a block transfer sequence it has to use a range of CAN-IDs for the different requests. The master has to send the first request with `CAN_ID_MASTER`. The master has to send consecutive requests by incrementing `CAN_ID_MASTER` for every new request. The master has to send the last request of the block transfer sequence with `CAN_ID_MASTER+MAX_BS(_PGM)-1`.

4.3 HEADER AND TAIL

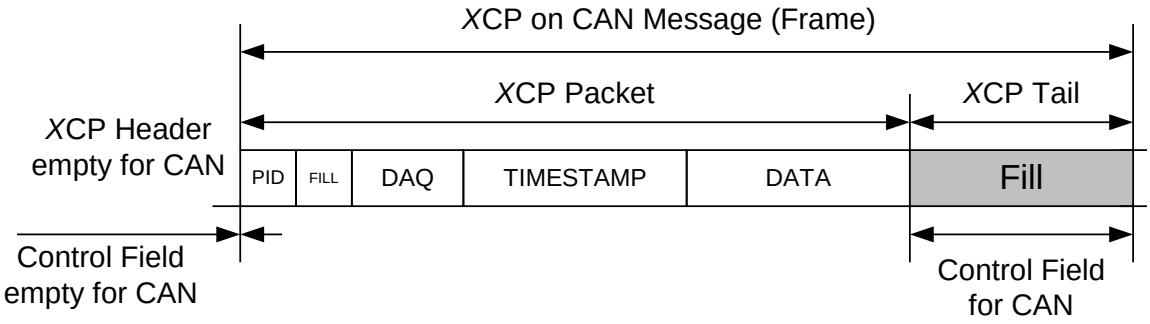

Figure 3 Header and tail for XCP on CAN

4.3.1 HEADER

For XCP on CAN there's no Header (empty Control Field).

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

4.3.2 TAIL

For XCP on CAN, the Tail consists of a Control Field containing optional Fill bytes.
The maximum data length of a CAN message and therefore maximum length of an XCP on CAN message is `MAX_DLC` = 8 for CAN and one of the following values for CAN-FD 8, 12, 20, 16, 24, 32, 48 or 64.
If the length (LEN) of an XCP Packet equals `MAX_DLC`, the Control Field of the XCP Tail is empty and the XCP on CAN Message is the same as the XCP Packet (DLC = LEN = `MAX_DLC`).
If LEN is smaller than `MAX_DLC`, there're 2 possibilities to set the DLC.
A first possibility is to set DLC = LEN. The Control Field of the XCP Tail is empty and the XCP on CAN Message is the same as the XCP Packet.

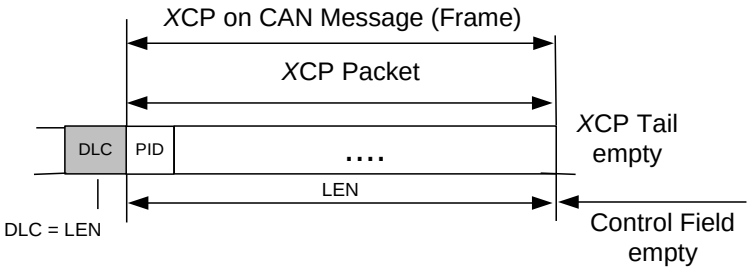

Figure 4 No XCP tail if DLC = LEN (<= MAX_DLC)

A second possibility is to set DLC = `MAX_DLC`. The Control Field of the XCP Tail contains `MAX_DLC` – LEN fill bytes. The content of the FILL bytes is "do not care".

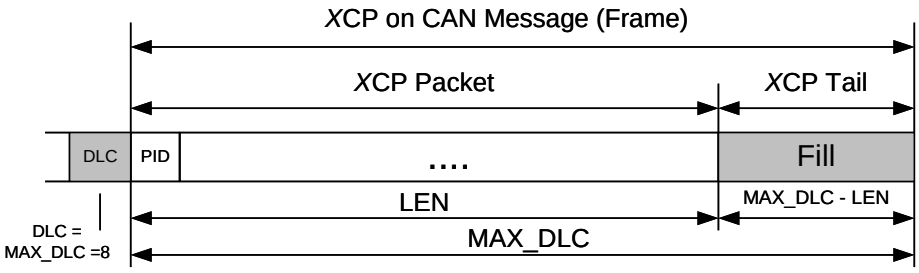

Figure 5 XCP tail if DLC = MAX_DLC (> LEN)

With `MAX_DLC_REQUIRED`, the slave can inform the master that it has to use CAN frames with `DLC = MAX_DLC = 8/64` when sending to the slave.

4.4 THE LIMITS OF PERFORMANCE

The maximum length of a CTO or a DTO packet for CAN is 8.
For CAN-FD only the following values are valid: 8, 12, 16, 20, 24, 32, 48 and 64 depending on the defined MAX_DLC. MAX_CTO and MAX_DTO ≤ MAX_DLC.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Table 1 CTO and DTO parameter

Name	Type	Representation	Range of value CAN	Valid values for CAN-FD
MAX_CTO	Parameter	BYTE	0x08	0x08, 0x0C, 0x10, 0x14, 0x18, 0x20, 0x30, 0x40
MAX_DTO	Parameter	WORD	0x0008	0x0008, 0x000C, 0x0010, 0x0014, 0x0018, 0x0020, 0x0030, 0x0040

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

5 SPECIFIC COMMANDS FOR XCP ON CAN

5.1 COMMAND OVERVIEW

Table 2 Command code overview

Command	Code	Timeout	Remark
GET_SLAVE_ID	0xFF	t_1	optional
GET_DAQ_ID	0xFE	t_1	optional
SET_DAQ_ID	0xFD	t_1	optional
GET_DAQ_CLOCK_MULTICAST	0xFA	irrelevant	optional

If `SET_DAQ_ID` is implemented, `GET_DAQ_ID` is required.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

5.2 GET SLAVE CAN IDENTIFIERS

Category CAN only, optional
Mnemonic GET_SLAVE_ID

Table 3 GET_SLAVE_ID command structure

Position	Type	Description
0	BYTE	Command Code = TRANSPORT_LAYER_CMD = 0xF2
1	BYTE	Sub Command Code = 0xFF
2	BYTE	0x58 (A_ASCII = X)
3	BYTE	0x43 (A_ASCII = C)
4	BYTE	0x50 (A_ASCII = P)
5	BYTE	Mode 0 = identify by echo 1 = confirm by inverse echo

The master can use GET_SLAVE_ID to detect all XCP slaves within a CAN network.
At the same time, the master gets to know the CAN identifier the master has to use when transferring CMD/STIM to a specific slave and the CAN identifier this slave uses for transferring RES/ERR/EV/SERV/DAQ.
The master has to send GET_SLAVE_ID with the XCP Broadcast CAN identifier.
If the master sends an XCP message with the XCP Broadcast CAN identifier, all XCP slaves that are connected to the CAN network have to respond. GET_SLAVE_ID is the only XCP message that can be broadcasted.
A slave always has to respond to GET_SLAVE_ID, even if the slave device is not in connected state yet.
The slave has to send the response with the CAN identifier it uses for transferring RES/ERR/EV/SERV/DAQ. The CAN identifier for CMD/STIM is coded in Intel format (MSB on higher position).
The master sends GET_SLAVE_ID with an Identification Pattern (ASCII for “XCP”). The master uses this Pattern for recognizing answers from XCP slaves.
If the master sends a GET_SLAVE_ID(identify by echo), the slave has to send a response that contains an echo of the Pattern. Additionally the slave informs the master about the CAN identifier the master has to use when transferring CMD/STIM to this slave.

Table 4 GET_SLAVE_ID positive response structure (mode = identify by echo)

Position	Type	Description
0	BYTE	Packet ID: 0xFF
1	BYTE	0x58
2	BYTE	0x43
3	BYTE	0x50
4	DWORD	CAN identifier for CMD/STIM

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

If the master sends a GET_SLAVE_ID (confirm by inverse echo), the slave has to send a response that contains an inversed echo of the Pattern. Additionally the slave repeats the CAN identifier the master has to use when transferring CMD/STIM to this slave.

Table 5 GET_SLAVE_ID positive response structure (mode = confirm by inversed echo)

Position	Type	Description
0	BYTE	Packet ID: 0xFF
1	BYTE	0xA7
2	BYTE	0xBC
3	BYTE	0xAF
4	DWORD	CAN identifier for CMD/STIM

If the master sends a GET_SLAVE_ID (confirm by inverse echo), without a previous GET_SLAVE_ID (identify by echo), the slaves will silently ignore that command.
If the master first sends a GET_SLAVE_ID (identify by echo) and then a GET_SLAVE_ID (confirm by inversed echo), this sequence allows the master to reliably distinguish the responses of the slaves from other communication frames on the CAN network and to reliably detect the CAN identifier pairs for every single slave.

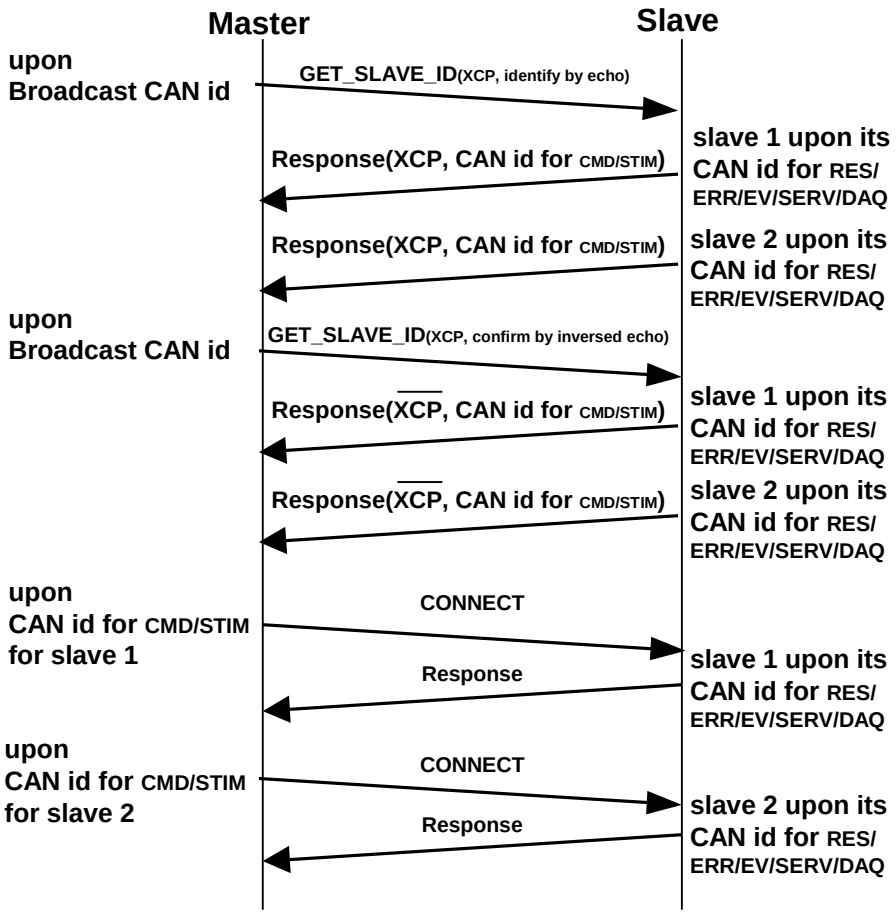

Figure 6 Typical use of GET_SLAVE_ID modes

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

5.3 GET DAQ LIST CAN IDENTIFIER

Category CAN only, optional
Mnemonic GET_DAQ_ID

Table 6 GET_DAQ_ID command structure

Position	Type	Description
0	BYTE	Command Code = TRANSPORT_LAYER_CMD = 0xF2
1	BYTE	Sub Command Code = GET_DAQ_ID = 0xFE
2	WORD	DAQ_LIST_NUMBER [0,1,...MAX_DAQ-1]

Table 7 GET_DAQ_ID positive response structure

Position	Type	Description
0	BYTE	Packet ID: 0xFF
1	BYTE	CAN_ID_FIXED
		0 = CAN-ID can be configured
		1 = CAN-ID is fixed
2	WORD	Reserved
4	DWORD	CAN Identifier of DTO dedicated to list number

As a default, the master transfers all DAQ lists with DIRECTION = STIM on the same CAN Identifier as used for CMD.
Alternatively, the master may have individual CAN Identifiers (other than the one used for CMD) for the DAQ lists with DIRECTION = STIM.

As a default, the slave transfers all DAQ lists with DIRECTION = DAQ on the same CAN Identifier as used for RES/ERR/EV/SERV.

Alternatively, the slave may have individual CAN Identifiers (other than the one used for RES/ERR/EV/SERV) for its DAQ lists with DIRECTION = DAQ.

With GET_DAQ_ID, the master can detect whether a DAQ list uses an individual CAN identifier and whether this Identifier is fixed or configurable.

If the CAN Identifier is configurable, the master can configure the individual Can Identifier for this DAQ list with SET_DAQ_ID.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

5.4 SET DAQ LIST CAN IDENTIFIER

Category CAN only, optional
Mnemonic SET_DAQ_ID

Table 8 SET_DAQ_ID command structure

Position	Type	Description
0	BYTE	Command Code = TRANSPORT_LAYER_CMD = 0xF2
1	BYTE	Sub Command Code = SET_DAQ_ID = 0xFD
2	WORD	DAQ_LIST_NUMBER [0,1,...MAX_DAQ-1]
4	DWORD	CAN Identifier of DTO dedicated to list number

The master can assign an individual CAN Identifier to a DAQ list.
If the given identifier isn't possible, the slave returns an ERR_OUT_OF_RANGE.

5.5 DAQ CLOCK MULTICAST ON CAN

Category CAN only, optional
Mnemonic GET_DAQ_CLOCK_MULTICAST

Table 9 GET_DAQ_CLOCK_MULTICAST command structure

Position	Type	Description
0	BYTE	Command Code = TRANSPORT_LAYER_CMD = 0xF2
1	BYTE	Sub Command Code = 0xFA
2	WORD	Cluster Identifier (Intel byte order = little endian)
4	BYTE	Counter (allows for consistency checks at XCP master)

When an XCP master makes use of GET_DAQ_CLOCK_MULTICAST command on transport layer CAN, a GET_DAQ_CLOCK_MULTICAST command must be sent every two seconds at the latest. The command shall be sent on the CAN-ID "CAN_ID_GET_DAQ_CLOCK_MULTICAST". This CAN-Id may be the same as the CAN-ID "CAN_ID_BROADCAST".

Upon reception of GET_DAQ_CLOCK_MULTICAST command, the XCP slave will respond an EV_TIME_SYNC event packet as defined in XCP Protocol Layer.

Multiple CAN-IDs might be foreseen by the network designer enabling the co-existence of multiple time-synchronization domains in one CAN network.

However, in case that only one CAN-ID is foreseen to be used for GET_DAQ_CLOCK_MULTICAST transmission, only one XCP master out of all XCP masters connected to one CAN-Bus is allowed to generate GET_DAQ_CLOCK_MULTICAST commands.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

5.6 COMMUNICATION ERROR HANDLING

This chapter describes transport layer specific error handling. It extends the error handling concepts specified in the chapter “Communication Error Handling” of the base standard [1]. Please refer to the base standard for obtaining fundamentals on XCP error handling.

5.6.1 ERROR CODE HANDLING

Table 10 Transport Layer CAN subcommands error handling

Command	Error	Pre-Action	Action
GET_SLAVE_ID	timeout t_1	SYNCH	repeat 2 times
	ERR_CMD_SYNTAX	-	retry other syntax
	ERR_SUBCMD_UNKNOWN	-	display error or silently ignore error
GET_DAQ_ID	timeout t_1	SYNCH	repeat 2 times
	ERR_OUT_OF_RANGE	-	retry other parameter
	ERR_SUBCMD_UNKNOWN	-	display error or silently ignore error
SET_DAQ_ID	timeout t_1	SYNCH	repeat 2 times
	ERR_OUT_OF_RANGE	-	retry other parameter
	ERR_SUBCMD_UNKNOWN	-	display error or silently ignore error
GET_DAQ_CLOCK_MULTICAST	–	-	-

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

6 SPECIFIC EVENTS FOR XCP ON CAN

There are no specific events for XCP on CAN.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

7 DATA TRANSFER ON THE CAN BUS

7.1 PRIORITIZATION OF THE MEASUREMENT DATA TRANSFER ON CAN

This chapter describes how the XCP master has to control the priority of the XCP communication on CAN in order not to disturb the mandatory vehicle communications on the network.

An ECU EVENT uses one or more DAQ lists for transmitting the configured measurements over the CAN bus. For distinguishing different DAQs the identification field (see chapter 8.1.2.1 [1]) and the CAN-ID are used and the combination for each DAQ has to be unique.
The prioritization of the data on the CAN bus correlates with the configuration of the CAN-IDs for the different EVENTs. A low CAN-ID indicates a high priority and high CAN-ID indicates a low priority on the CAN bus for the messages. XCP EVENTs may have different priorities. The priority of data on the CAN bus is defined by the CAN-ID used. The XCP on CAN transport layer specification allows to define the CAN-IDs which are used for each EVENT.
For static DAQ lists, the CAN-ID to be used is defined in the IF_DATA as follows:

Static: Assigning CAN-ID to static DAQ list e.g.:

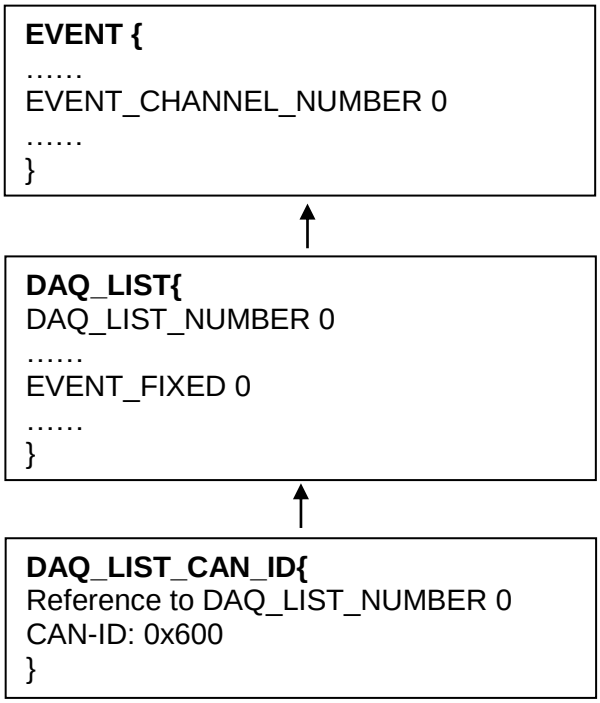

Figure 7 CAN-ID with static DAQ lists

Settings for dynamic DAQ lists cannot be defined in the IF_DATA because dynamic DAQ lists are created by the XCP master at run time.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Therefore, for dynamic DAQ list configuration, CAN IDs can be assigned to EVENTs. This is illustrated below.

Dynamic: Assigning a list of CAN IDs to an event e.g.:

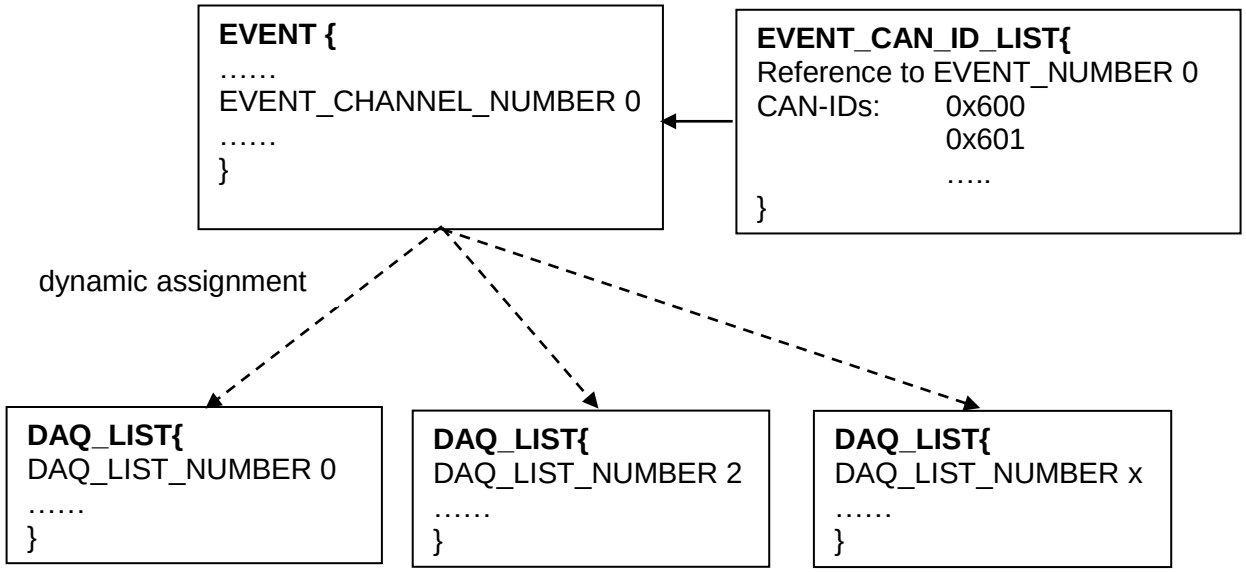

Figure 8 CAN-ID with dynamic DAQ lists

The DAQ_LIST_CAN_ID block shall only be used for static DAQ lists and the EVENT_CAN_ID_LIST block shall only be used for dynamic DAQ lists.

With dynamic DAQ lists the referenced CAN-IDs in the EVENT_CAN_ID_LIST has to be handled as an EVENT specific filter by the XCP master. The XCP slave is allowed to send CAN frames on all defined CAN-IDs of a certain EVENT for all assigned DAQ lists.
The CAN-IDs of all DAQ lists have to be listed in the EVENT_CAN_ID_LIST block of the assigned EVENT. For getting detailed information which CAN-ID is used by which DAQ list the XCP master can read out this information from the XCP slave.

If the XCP slave supports the optional TRANSPORT_LAYER command GET_DAQ_ID (see Chapter 6.3) the XCP master can check if the CAN-ID configuration in the ECU matches to the configuration in the a2l file after all DAQs are assigned to EVENTs with the SET_DAQ_LIST_MODE command**.**

The XCP master can filter the received ODTs by the Identification field to have a clear assignment to the DAQ/EVENT even if the data is transferred over several CAN-IDs from the XCP slave. The identification field guarantees the uniqueness of the data. The FIRST_PID of the DAQ list is part of the positive response to the START_STOP_DAQ_LIST command if the used identification field is ABSOLUTE. With this definition multiple DAQ lists on one CAN-ID are possible to use.
The XCP slave has an algorithm how to distribute the DAQ lists over the available CAN-IDs to optimize certain use cases. Due to this MAX_DAQ_LIST of an EVENT can differ to the number of CAN-IDs of this EVENT. The simplest configuration is to have only one CAN-ID and one DAQ list for each EVENT. It is not possible to configure more than 252 ODTs per DAQ list for the XCP master.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Only with PID_OFF and more than one ODT in one DAQ list per EVENT the assignment is not clear anymore because for each ODT an own DAQ list has to be used.

For using the PID_OFF feature with dynamic DAQ lists it is mandatory for the XCP master to read the used CAN-ID for each DAQ list from the ECU with the optional transport layer command at the end of the DAQ configuration to have a consistent configuration for the XCP master.

7.2 LIMIT OF BUS BANDWIDTH USED FOR DATA TRANSFER

To guarantee a stable CAN bus, it is possible to limit the maximum used bandwidth for data transfer. Especially on a not private CAN bus it is possible that an ECU transfers more data than there is bandwidth left or assigned for data transferring. Due to the prioritization with CAN-IDs, it is possible that other communication on the CAN bus will then be disturbed. For the CAN bus as XCP transport layer, it is possible to define an optional limit for data transfer in percent of the current bus configuration.

The XCP master can monitor this limit while configuring the data transfer. For this, the master has to use the current DAQ configuration in combination with the cycle time of the assigned EVENTs. With this information, the XCP master can calculate the resulting TOTAL_BUSLOAD for the current configuration.

To make it easier for the calculation, the following assumptions are made:

- The CAN frame has always a DLC of 8.
- For CAN-FD, the next larger defined DLC is considered for the calculation if MAX_DLC_REQUIRED is not set and DLC of a frame is not matching one of the defined DLC values.
- If MAX_DLC_REQUIRED is set for CAN-FD, always the corresponding DLC = MAX_DLC parameter is used for the calculation.
- The length of the CAN/CAN-FD frames is not constant because of possible stuffing bits. An assumed average value between the minimum and maximum length has to be used for the calculation of the busload.

Table 11 Frame length in bits for busload calculation

Bus	DLC	Used frame length for Calculation (Std. CAN-ID)	Used frame length for Calculation (Ext. CAN-ID)
CAN	8	120	140
CAN-FD	8	130	150
	12	170	195
	16	210	230
	20	245	265
	24	280	300
	32	320	340
	48	495	515
	64	640	660

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

CAN:

Total_Busload for CAN =

$$\sum_{k=0}^{n} \frac{1}{\text{MIN\_CYCLE\_TIME (k)}} * \#\ \text{of ODT in EVENT(k)} * frame\_length(k)$$

(k = EVENT_CHANNEL_NUMBER, n = MAX_EVENT_CHANNEL)

Example:

1Mbaud CAN with 2 ODT`s in 2ms EVENT and 6 ODT`s in 10ms EVENT with standard CAN_IDs and a MAX_BUS_LOAD of 50%.

Total_Busload =

$$\frac{1}{2\ ms} * 2 * 120\ bit + \frac{1}{10\ ms} * 6 * 120\ bit$$

$$120000\ bit/s + 72000\ bit/s$$

Consumption =

$$\frac{Total\_Busload}{MAX\_BUS\_LOAD * BAUDRATE} = \frac{192000\ bit/s}{500000\ bit/s} = 38{,}4\ \%\ of\ maximum\ available\ bus\ load$$

CAN-FD:

The arbitration part and the data part of a CAN-FD frame are having typically different baud rates on the CAN bus. The frame length for the calculation can change depending on the MAX_DLC_REQUIRED setting. Therefore the calculation for CAN-FD frames is different to the calculation of CAN frames. In consequence the frame length (see Table 11) has to be spitted into arbitration and data bits.
The arbitration part is almost constant independent from the frame length. Only a few stuff bits are possible. Therefore an assumed average length for the arbitration has to be used for the calculation.

- For standard CAN-ID frames the length for the arbitration part is 30 bit
- For extended CAN-ID frames the length for the arbitration part is 50 bit

For the Total_Busload calculation it is important to normalize the data part of the frames to the arbitration part.

$$\#\ of\ bits[CAN] = \#\ of\ bitsArbitration[CAN] + \frac{\#\ of\ bitsData[FD] * BAUDRATE}{CAN\_FD\_DATA\_TRANSFER\_BAUDRATE}$$

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Example:

For a 32 byte CAN-FD frame with extended CAN-ID and BAUDRATE = 1Mbaud and CAN_FD_DATA_TRANSFER_BAUDRATE = 8Mbaud

$\#\ of\ bits[CAN]$ = 50 bit + (340 bit – 50 bit) * 1Mbaud / 8Mbaud = 86,25 bit => 87 bit

Total_Busload for CAN-FD =

$$\sum_{k=0}^{n} \frac{1}{\text{MIN\_CYCLE\_TIME (k)}} * \text{Sum\_of\_Bits\_for\_all\_ODTs\_of Event(k)}$$

(k = EVENT_CHANNEL_NUMBER, n = MAX_EVENT_CHANNEL)

Sum_of_Bits_for_all_ODTs_of Event(k) =

$$\sum_{o=0}^{r} \#\ of\ Bits[CAN]\ of\ ODT(o)\ in\ Event(k)$$

(k = EVENT_CHANNEL_NUMBER)
The index r defines the total number of ODTs in a specific EVENT(k)
The index o identifies a specific ODT for which the NoOfBits[CAN] has to be calculated.

Example:

BAUDRATE 1MBaud, CAN_FD_DATA_TRANSFER_BAUDRATE 8Mbaud, MAX_DLC_REQUIRED, MAX_BUS_LOAD = 30%, std. CAN-IDs, MAX_DLC = 64, 1 ODT with 20 bytes and 2 ODTs with 64 bytes in a 2ms EVENT.

Sum_of_Bits_for_all_ODTs_of Event(2ms) =

3 x NoOfBits[CAN] for a 64 byte frame

3 x (30 bit + (640 bit – 30 bit) * 1Mbaud / 8Mbaud = 3 x 106,25 bit => 321 bit

Total_Busload for CAN-FD =

$$\frac{1}{2\ ms} *\ 321\ bit$$

Total_Busload for CAN-FD = $160500\ bit/s$

Consumption =

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

$$\frac{Total\_Busload}{MAX\_BUS\_LOAD * BAUDRATE} = \frac{160500\ bit/s}{300000\ bit/s} = 53{,}5\ \%\ of\ maximum\ available\ bus\ load$$

MAX_BUSLOAD limits only the traffic on the CAN bus for DAQ and STIM but not for command, response, error or event messages.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

7.3 SUPPORTING EFFICIENT DAQ DATA TRANSFER

With `MEASUREMENT_SPLIT_ALLOWED` at the CAN transport layer of the ASAM MCD-2 MC description file, the slave can indicate that for MEASUREMENT variables, that can be measured consistent within one ODT entry as defined in XCP Protocol Layer, the consistency is also guaranteed by the slave, if they are split into two consecutive ODT entries of two consecutive ODTs.

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

8 CAN-FD TRANSPORT LAYER

For CAN-FD transport layer configuration it is necessary to have additional information in the a2l file.

For CAN–FD a second CAN baud rate is defined which is used for transferring the data area of CAN frame with a different speed. The additional parameters are defined in an optional CAN-FD block in the CAN transport layer AML version 1.2. This block is mandatory for using CAN-FD. If the block is available the transport layer instance becomes incompatible to a standard CAN and cannot be used for a standard CAN bus anymore because it is not possible to send standard CAN and CAN-FD messages in parallel on the same bus.

The DLC for CAN-FD has additional values.

Table 12 DLC for CAN and CAN-FD

ISO		Bus	DLC (4bit)	Number of Data Bytes
ISO 11898-2	ISO 11898-7	CAN/CAN-FD	0 0 0 0	0
		CAN/CAN-FD	0 0 0 1	1
		CAN/CAN-FD	0 0 1 0	2
		CAN/CAN-FD	0 0 1 1	3
		CAN/CAN-FD	0 1 0 0	4
		CAN/CAN-FD	0 1 0 1	5
		CAN/CAN-FD	0 1 1 0	6
		CAN/CAN-FD	0 1 1 1	7
		CAN/CAN-FD	1 0 0 0	8
		CAN-FD	1 0 0 1	12
		CAN-FD	1 0 1 0	16
		CAN-FD	1 0 1 1	20
		CAN-FD	1 1 0 0	24
		CAN-FD	1 1 0 1	32
		CAN-FD	1 1 1 0	48
		CAN-FD	1 1 1 1	64

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

9 INTERFACE TO ASAM MCD-2 MC DESCRIPTION FILE

The following chapter describes the parameters that are specific for XCP on CAN.

9.1 ASAM MCD-2 MC AML FOR XCP ON CAN

The AML for the XCP on CAN transport layer specific properties is defined in the file named XCP_vX_Y_on_CAN.aml where vX_Y is the current transport layer version.

9.2 IF_DATA EXAMPLE FOR XCP ON CAN

The file XCP_vX_Y_IF_DATA_example.a2l where vX_Y is the current protocol layer version gives an IF_DATA example for a XCP on CAN transport layer (see section beginning with "`/begin XCP_ON_CAN`").

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

10 SYMBOLS AND ABBREVIATED TERMS

A2L	ASAM MCD-2 MC Language File
AML	AML
CAN	Controller Area Network
CAN-FD	Controller Area Network – Flexible Data Rate
CCP	CAN Calibration Protocol
CMD	Command
CTO	Command Transfer Object
DAQ	Data Acquisition
DLC	Data Length Code
DTO	Data Transfer Objects
ECU	Electronic Control Unit
ERR	Error
EV	Event
ID	Identifier
LEN	Length
MSB	Most significant Bit
ODT	Object Descriptor Table
PGM	Programming
PID	Packet Identifier
RES	Responses
SERV	Service
STIM	Synchronous Data Stimulation
XCP	Universal Measurement and Calibration Protocol

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

11 BIBLIOGRAPHY

[1] ASAM AE MCD-1 XCP BS Protocol-Layer BS Version V1.3.0
[2] **ASAM MCD-2 MC**/"Measurement and Calibration Data Specification", Version 1.7.x

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Figure Directory

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Table Directory

Licensed to ASAM e.V.
Downloaded by udo.czerwon@continental-corporation.com on 26.03.2018

Association for Standardisation of
Automation and Measuring Systems

E-mail: support@asam.net
Web: www.asam.net

© by ASAM e.V., 2017