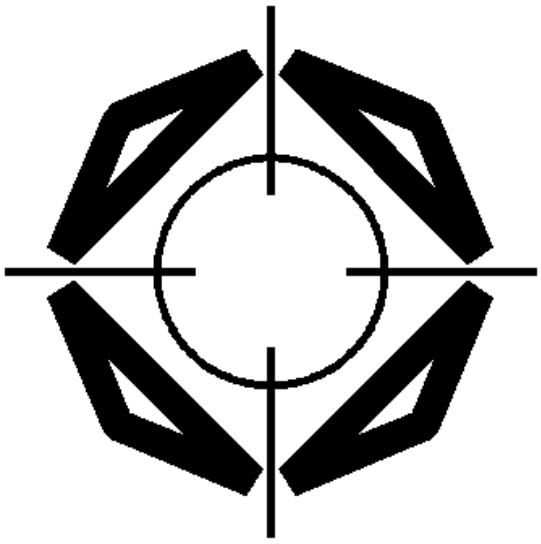
XCP

Version 1.0

"The Universal Measurement and Calibration Protocol Family"

Part 3

*X*CP on Ethernet - Transport Layer Specification



Association for Standardization of Automation and Measuring Systems

Dated:2003-04-08

Status of Document

Date:	2003-04-08
Authors:	Roel Schuermans, Vector Informatik GmbH Rainer Zaiser, Vector Informatik GmbH Frank Hepperle, DaimlerChrysler AG Hans Schröter, DaimlerChrysler AG Reiner Motz, Robert Bosch GmbH Andreas Aberfeld, Robert Bosch GmbH Hans-Georg Kunz, Siemens VDO Automotive AG Thomas Tyl, Siemens VDO Automotive AG Robert Leinfellner, dSPACE GmbH Hendirk Amsbeck, dSPACE GmbH Harald Styrsky, Compact Dynamics GmbH Boris Ruoff, ETAS GmbH Lars Wahlmann, Accurate Technologies Inc.
Version:	1.0
Doc-ID:	XCP -Part 3- Transport Layer Specification XCP on Ethernet (TCP/IP and UDP) -1.0
Status:	Released
Type	Final

Revision History

This revision history shows only major modifications between release versions.

Date	Author	Filename	Comments
2003-04-08	R.Schuermans		Released document

Table of contents

Table of diagrams:

0 Introduction

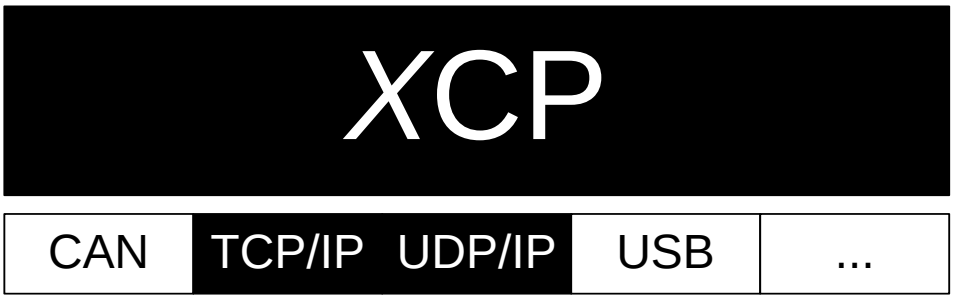
0.1 The *X*CP Protocol Family

This document is based on experiences with the **C**AN **C**alibration **P**rotocol (CCP) version 2.1 as described in feedback from the companies Accurate Technologies Inc., Compact Dynamics GmbH, DaimlerChrysler AG, dSPACE GmbH, ETAS GmbH, Kleinknecht Automotive GmbH, Robert Bosch GmbH, Siemens VDO Automotive AG and Vector Informatik GmbH.

The *X*CP Specification documents describe an improved and generalized version of CCP.

The generalized protocol definition serves as standard for a protocol family and is called "*X*CP" (Universal Measurement and **C**alibration **P**rotocol).

The **"*X*"** generalizes the "various" transportation layers that are used by the members of the protocol family e.g "*X*CP on CAN", "*X*CP on TCP/IP", "*X*CP on UDP/IP", "*X*CP on USB" and so on.



0.2 Documentation Overview

The *X*CP specification consists of 5 parts. Each part is a separate document and has the following contents:

Part 1 "Overview" gives an overview over the *X*CP protocol family, the *X*CP features and the fundamental protocol definitions.

Part 2 "Protocol Layer Specification" defines the generic protocol, which is independent from the transportation layer used.

Part 3 "Transport Layer Specification" defines the way how the *X*CP protocol is transported by a particular transportation layer like CAN, TCP/IP and UDP/IP.

This document describes the way how the *X*CP protocol is transported on Ethernet (TCP/IP and UDP/IP)

Part 4 "Interface Specification" defines the interfaces from an *X*CP master to an ASAM MCD 2MC description file and for calculating Seed & Key algorithms and checksums.

Part 5 "Example Communication Sequences" gives example sequences for typical actions performed with *X*CP.

Everything not explicitly mentioned in this document, should be considered as implementation specific.

0.3 Definitions and Abbreviations

The following table gives an overview about the most commonly used definitions and abbreviations throughout this document.

Abbreviation	Description
A2L	File Extension for an **A**SAM **2**MC **L**anguage File
AML	**A**SAM 2 **M**eta **L**anguage
ASAM	**A**ssociation for **S**tandardization of **A**utomation and **M**easuring Systems
BYP	**BYP**assing
CAL	**CAL**ibration
CAN	**C**ontroller **A**rea **N**etwork
CCP	**C**an **C**alibration **P**rotocol
CMD	**C**o**M**man**D**
CS	**C**heck**S**um
CTO	**C**ommand **T**ransfer **O**bject
CTR	**C**oun**T**e**R**
DAQ	**D**ata **A**c**Q**uisition, **D**ata **A**c**Q**uisition Packet
DTO	**D**ata **T**ransfer **O**bject
ECU	**E**lectronic **C**ontrol **U**nit
ERR	**ERR**or Packet
EV	**EV**ent Packet
LEN	**LEN**gth
MCD	**M**easurement **C**alibration and **D**iagnostics
MTA	**M**emory **T**ransfer **A**ddress
ODT	**O**bject **D**escriptor **T**able
PAG	**PAG**ing
PGM	**P**ro**G**ra**M**ming
PID	**P**acket **ID**entifier
RES	command **RES**ponse packet
SERV	**SERV**ice request packet
SPI	**S**erial **P**eripheral **I**nterface
STD	**ST**an**D**ard
STIM	Data **STIM**ulation packet
TCP/IP	**T**ransfer **C**ontrol **P**rotocol **/ I**nternet **P**rotocol
TS	**T**ime Stamp
UDP/IP	**U**nified **D**ata **P**rotocol **/ I**nternet **P**rotocol
USB	**U**niversal **S**erial **B**us
XCP	Universal **C**alibration **P**rotocol

Table 1: Definitions and Abbreviations

1 The *X*CP Transport Layer for Ethernet (TCP/IP and UDP/IP)

1.1 Addressing

A slave device connected by Ethernet and TCP/IP or UDP/IP protocol is addressed by its IP Address and Port number.

TCP/IP :

The slave device is the listener. It will only accept one connection at the time. If the socket is closed while in *X*CP connected state, the slave device will perform an *X*CP disconnect, which means that all data acquisition will be stopped.

UDP/IP:

While not connected, the slave device will answer upon a CONNECT command by sending the response to the IP address and port of the sender of the command. It will continue to answer to this IP address and port for all subsequent responses. When connected, it will respond only to telegrams from the IP address which has sent the CONNECT command even if another port is used . All other command packets will not be responded.

1.2 Communication Model

*X*CP on TCP/IP and UDP/IP makes use of the standard communication model.

The block transfer communication is optional.
The interleaved communication model is optional.

1.3 Header and Tail

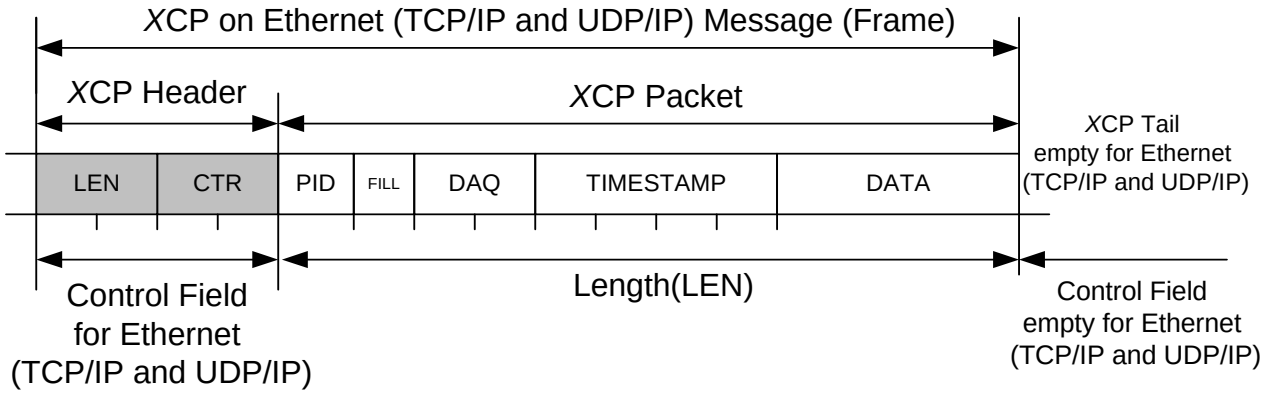


Diagram 1 : Header and Tail for *X*CP on Ethernet (TCP/IP and UDP/IP)

1.3.1 Header

For *X*CP on Ethernet (TCP/IP and UDP/IP) the Header consists of a Control Field containing a **LEN**gth (LEN) and a **C**oun**T**e**R** (CTR).

Both LEN and CTR always are WORDs in Intel format.

To make optimal use of UDP/IP, multiple *X*CP Frames may be combined into a single UDP/IP frame, but an *X*CP Frame may not cross a UDP/IP frame boundary.

The same *X*CP Frame format is used for the stream oriented protocol TCP/IP to simplify decoding the original *X*CP messages.

1.3.1.1 Length

LEN is the number of bytes in the original *X*CP Packet.

1.3.1.2 Counter

The CTR value in the *X*CP Header allows to detect missing Packets.

The master has to generate a CTR value when sending a CMD or STIM. The master has to increment the CTR value for each new packet sent from master to slave.

The slave has to generate a (second independent) CTR value when sending a RES, ERR_EV, SRM or DAQ. The slave has to increment the CTR value for each new packet sent from slave to master.

1.3.2 Tail

For *X*CP on Ethernet (TCP/IP and UDP/IP) there's no Tail (empty Control Field).

1.4 The Limits of performance

The upper limit of MAX_CTO and MAX_DTO depends on protocol stack (TCP/IP and UDP/IP) of the host system.

Name	Type	Representation	Range of value
MAX_CTO	Parameter	BYTE	0x08 – 0xFF
MAX_DTO	Parameter	WORD	0x0008 – 0xFFFF

2 Specific commands for *X*CP on Ethernet (TCP/IP and UDP/IP)

There are no specific commands for *X*CP on Ethernet (TCP/IP and UDP/IP) at the moment.

3 Specific events for *X*CP on Ethernet (TCP/IP and UDP/IP)

There are no specific events for *X*CP on Ethernet (TCP/IP and UDP/IP) at the moment.

4 Interface to ASAM MCD 2MC description file

The following chapter describes the parameters that are specific for *X*CP on TCP/IP and for *X*CP on UDP/IP.

4.1 ASAM MCD 2MC AML for *X*CP on Ethernet (TCP/IP and UDP/IP)

```
/*************************************************************************************/
/*                                                                                   */
/*   ASAP2 meta language for XCP on TCP_IP V1.0                                      */
/*                                                                                   */
/*   2003-03-03                                                                      */
/*                                                                                   */
/*   Vector Informatik, Schuermans                                                   */
/*                                                                                   */
/*   Datatypes:                                                                      */
/*                                                                                   */
/*   A2ML      ASAP2           Windows     description                               */
/*   ------------------------------------------------------------------------------*/
/*   uchar     UBYTE           BYTE        unsigned 8 Bit                            */
/*   char      SBYTE           char        signed 8 Bit                              */
/*   uint      UWORD           WORD        unsigned integer 16 Bit                   */
/*   int       SWORD           int         signed integer 16 Bit                     */
/*   ulong     ULONG           DWORD       unsigned integer 32 Bit                   */
/*   long      SLONG           LONG        signed integer 32 Bit                     */
/*   float     FLOAT32_IEEE                float 32 Bit                              */
/*                                                                                   */
/*************************************************************************************/

/************************ start of TCP_IP ***********************************/

  struct TCP_IP_Parameters { /* at MODULE */

    uint;                       /* XCP on TCP_IP version */
                                /* e.g. "1.0" = 0x0100 */

    uint; /* PORT */

    taggedunion {
      "HOST_NAME" char[256];
      "ADDRESS" char[15];
    };

  };/************************ end of TCP_IP ***********************************/
```

```
/*****************************************************************************************/
/*                                                                                       */
/*   ASAP2 meta language for XCP on UDP_IP V1.0                                          */
/*                                                                                       */
/*   2003-03-03                                                                          */
/*                                                                                       */
/*   Vector Informatik, Schuermans                                                       */
/*                                                                                       */
/*   Datatypes:                                                                          */
/*                                                                                       */
/*   A2ML      ASAP2           Windows     description                                   */
/*   ------------------------------------------------------------------------------------*/
/*   uchar     UBYTE           BYTE        unsigned 8 Bit                                */
/*   char      SBYTE           char        signed 8 Bit                                  */
/*   uint      UWORD           WORD        unsigned integer 16 Bit                       */
/*   int       SWORD           int         signed integer 16 Bit                         */
/*   ulong     ULONG           DWORD       unsigned integer 32 Bit                       */
/*   long      SLONG           LONG        signed integer 32 Bit                         */
/*   float     FLOAT32_IEEE                float 32 Bit                                  */
/*                                                                                       */
/*****************************************************************************************/
/************************** start of UDP_IP ************************************/

  struct UDP_IP_Parameters {  /* at MODULE */

    uint;                       /* XCP on UDP_IP version */
                                /* e.g. "1.0" = 0x0100 */

    uint; /* PORT */

    taggedunion {
      "HOST_NAME" char[256];
      "ADDRESS" char[15];
    };

};/************************** end of UDP_IP ************************************/
```

4.2 IF_DATA example for *X*CP on Ethernet (TCP/IP and UDP/IP)

```
/begin XCP_ON_TCP_IP
    0x0100          /* XCP on TCP_IP version */
     0x5555         /* PORT      */
    "127.0.0.1"     /* ADDRESS   */
/end XCP_ON_TCP_IP

/begin XCP_ON_UDP_IP
    0x0100          /* XCP on UDP_IP version */
     0x5555         /* PORT      */
    "127.0.0.1"     /* ADDRESS   */
/end XCP_ON_UDP_IP
```

ASAM e. V.
Arnikastraße 2
D - 85635 Hoehenkirchen
Germany

Tel.: (+49) 8102 / 895317
Fax.: (+49) 8102 / 895310
E-mail: info@asam.net
Internet: www.asam.net