Development engineers and technicians benefit from high-performance standards and tools for optimal parameterization of ECUs. Such standards and tools must enable flexible read and write access to variables or memory contents. The CAN Calibration Protocol (CCP) – an OEM-independent standard – was developed for this purpose back in the 1990s. At that time, CAN was the sole dominant system for in-vehicle networking.
As automotive electronics continued to develop, additional bus systems such as LIN, MOST® and FlexRay came into use. Because it was restricted to the CAN bus, CCP reached its limits, and this led to the development of the XCP protocol.

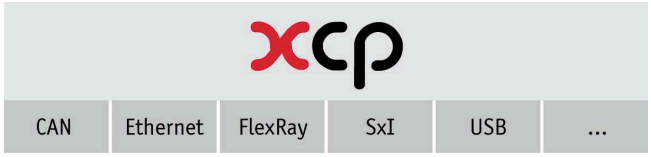
Like CCP, the "Universal Measurement and Calibration Protocol" (XCP) originated from the "Association for Standardization of Automation and Measuring Systems" (ASAM) and was standardized in the year 2003. Vector played a key role in its release.
The "X" stands for the variable and interchangeable transport layer. To satisfy needs for a universal communication solution in a wide variety of applications, the ASAM working group gave special attention to the following criteria:

- Support of a variety of transport layers such as CAN, FlexRay, Ethernet, SPI, SCI, USB, etc.
- Better resource utilization in the ECU (dynamic DAQ lists)
- Synchronous data stimulation (bypassing)
- Support of start-up measurement (Resume mode)
- Optimized communication by block transfer
- Polling of XCP functions available in the ECU (plug & play)
- More precise measurement data acquisition by measuring time stamps in the ECU (Slave)

As a two-layer protocol, it consistently splits the XCP protocol from the transport layers, and it takes a Single-Master/Multi-Slave approach.

Protocol Layer and Transport Layers
XCP is capable of utilizing the same protocol layer based on different transport layers. This protocol is a universal measurement and calibration protocol that operates independently of the type of network that is used. ASAM defines the following standard transport layers (Status May 2010):

- XCP on CAN
- XCP on SxI (SPI, SCI)
- XCP on Ethernet (TCP/IP and UDP/IP)
- XCP on USB
- XCP on FlexRay

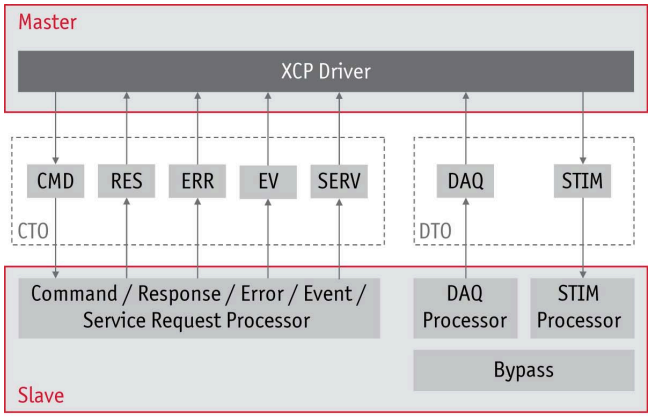


The latest member of this protocol family is XCP on FlexRay, which has been specified since early 2006. As a participant in the ASAM working group for standardizing XCP on FlexRay, Vector helped to define basic principles and contributed its extensive XCP know-how to the standardization process. Users benefit from early implementation of the new standard in CANape.

Single-Master/Multi-Slave Concept

The measurement and calibration system assumes the role of XCP master, while the ECU operates as a slave. The master and slave each communicate over the integrated XCP driver. There is an ECU description file (in A2L format) for each slave, which – among other things – specifies associations between symbolic variable names and their address ranges, physical meanings of the data and the checksum method used. The XCP master can read out all necessary information from these A2L description files.

In communication via XCP, a distinction is made between the "Command Transfer Object" (CTO) and the "Data Transfer Object" (DTO). For example, the master might send a command to the ECU over the bus by CTO, and the ECU acknowledges over the same pathway after executing the requested service. Available CTOs are: CMD (Command), RES (Response), ERR (Error), EV (Event) and SERV (Service Request Processor). The DAQ (Data Acquisition) and STIM (Stimulation) data transfer objects are used for event-driven reading of measurement variables from the XCP slave's memory or writing values to memory.



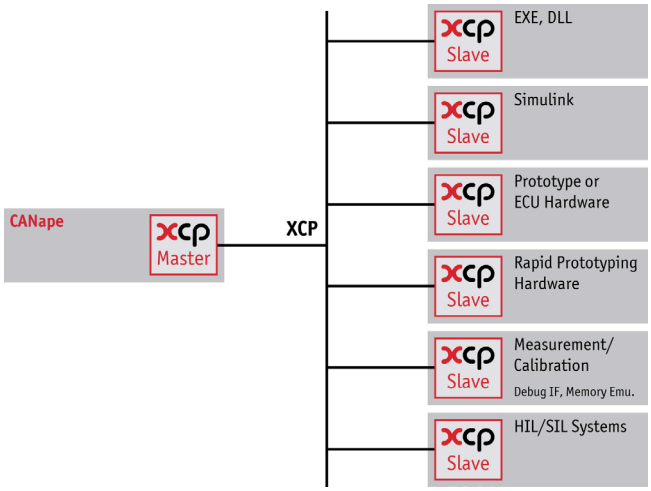
CTO: Command Transfer Object DTO: Data Transfer Object

Measuring and Calibrating Throughout the Development Process

In measuring and calibrating with XCP, the specific environment in which the code or model (Simulink, rapid prototyping hardware, HIL, SIL, etc.) is run is irrelevant. That is, in optimizing the control algorithms for the ECU, you can use one and the same tool, always performing similar work steps. This makes configurations, measurement data and description files usable and interchangeable throughout the development process.

An XCP Master, such as CANape, is capable of communicating with different XCP slaves simultaneously. They include:

- ECUs or ECU prototypes
- Measurement and calibration hardware such as debug interfaces or memory emulators
- Rapid prototyping hardware
- HiL/SiL systems



The XCP standard was developed with crucial input by Vector, whose extensive know-how and experience also flowed into comprehensive XCP support:

Tools

- **CANape** is primarily used for optimal parameterization (calibration) of electronic control units. You can calibrate parameter values during the system's runtime and simultaneously acquire measured signals. The physical connection between CANape and the ECU is over XCP (for all standardized transport protocols) or over CCP.
- A complete tool chain for generating and managing the necessary A2L description files (**ASAP2 Tool-Set** and **CANape** with **ASAP2 Editor** – also available as a stand-alone tool).
- **CANoe.XCP** lets you access internal ECU values for testing and analysis tasks.

Embedded Software

Communication modules with separate transport layers for CAN, FlexRay and Ethernet:

- **XCP Basic** – free download, only contains basic XCP functions
- **XCP Professional** – available for CANbedded and AUTOSAR, along with the entire range of ASAM functional features, it includes many other useful extensions

Services

- **Consultation** on using XCP on your projects
- **Integration** of XCP in your ECU

Training

- You can learn about the protocol's underlying mechanisms and models in the „**XCP Fundamentals Seminar**".
- The „**CANape mit XCP on FlexRay**" workshop explains the fundamentals of FlexRay and special aspects of XCP on FlexRay, especially dynamic bandwidth management.

Special XCP Support in CANape

CANape was the first MCD tool to support the XCP 1.0 specification and was also the first XCP on FlexRay master on the market:

- See technical article "A Multifaceted Standard – Optimize ECU Parameters with XCP"
- See technical article "XCP on FlexRay at BMW".

One special technical aspect of **XCP on FlexRay** is its dynamic bandwidth management. CANape utilizes this to identify the bandwidth available for XCP in the FlexRay cluster, and it allocates this bandwidth – dynamically and very efficiently – to the current calibration data traffic. This method optimally exploits the bandwidth available for XCP communication. [more...]

CANape also has a **DLL interface**. It enables support of XCP on any (user-defined) transport layer. This lets you integrate any desired measurement technology or proprietary protocols in CANape. A code generator supports you in creating the XCP-specific content for such a driver.