ASAM | Association for standardisation of
automation and measuring systems

Release Presentation

P2016-04_AE_MCD-1_XCP_BS+AS_V1-4-0_FVD

ASAM MCD1-MC V1.4

Release Date: 2017 / 07 / 01

Introduction

ASAM MCD-1 XCP defines a bus-independent, master-slave communication protocol to connect ECUs with calibration systems. XCP is short for Universal Measurement and Calibration Protocol. The primary purpose of XCP is to adjust internal parameters and acquire the current values of internal variables of an ECU. The first letter X in XCP shall express the fact that the protocol is designed for a variety of bus systems. The standard consists of a base standard, which describes memory-oriented protocol services without direct dependencies to specific bus systems. Several associate standards contain the transport layer definitions for CAN, FlexRay, Ethernet (UDP/IP and TCP/IP)and serial links (SPI and SCI).

Introduction (contd.)

XCP accesses parameters and measurement variables in a memory address oriented way. The properties and memory addresses of this data are described in the A2L-file format, which is standardized through ASAM MCD-2 MC. The A2L-file contains all information to carry out such accesses and to correctly interpret the data that is transmitted via the XCP protocol. This means that access to a specific parameter or variable does not need to be hardcoded into the ECU application. In other words, the ECU just contains a generic XCP-protocol stack, which responds to memory access requests from the calibration system. Different calibration and measurement tasks can be performed by different configurations of the calibration system without recompiling and reprogramming the ECU application code.

Introduction (contd.)

XCP was designed with two main objectives. The first is to impact the ECU resources, such as CPU load, RAM consumption and flash memory, for the XCP slave in a minimal way. The second is to achieve a maximal data transmission rate over the communication link and to reduce the impact on bus communication as much as possible. The standard also describes the organization of the ECU memory segments used by the ECU software. This allows memory-type specific access. Additionally, it describes the ECU interface for data read- and write access.

Deliverables

- Specification documents
- AML Sources, A2L example files
- Seed&Key DLL Template

What's New?

Overview

- **Packed DAQ lists**
  This feature highly reduces protocol overhead for cyclic measurement data transmission by omitting timestamps if they can be restored by the MC tool by a well defined formula.
- **Improvements for DAQ data consistency**
  The standard provides means to describe logical DAQ data consistency e.g. for multicore processors
- **Slave enforced calibration page switching and resource state changes**
  This feature is added to the ECU state model to reflect calibration page switching or DAQ resource state changes enforced by the ECU, for example due to a power reset
- **POD standard related requirements**
- **Command space enhancement**
  The number of possible protocol commands is enhanced. This is also required by the new POD standard.
- **GET_VERSION command**
  This command provides the minor protocol version as additional information which was not accessible before.

What’s New?

Overview (contd.)

- **Enhancements of ODT optimization types**
  A „strict mode“ for ODT optimization was introduced.
- **START_STOP_SYNC command enhancement**
  The mode „prepare for start selected“ was added to avoid a race condition when starting DAQ lists for data acquisition
- **XCP on Ethernet packet alignment**
  By introducing fill space to the XCP on ETH tail, it is possible to align XCP on ETH packets within a UDP frame to e.g. WORD boundaries
- **XCP on Ethernet slave IP address configuration**
  The MC tool can actively set the IP address of an XCP slave such as an MC hardware interface
- **XCP on CAN efficient DAQ data transfer support**
  The standard supports efficient data transfer of measurement labels which are larger than the possible CAN payload by allowing splitting in certain cases

What’s New?

Base Specification (Protocol Layer)

- **Packed DAQ lists**
  Chapter 4.1.10, 6.2.3, 7.4, 7.5.4.7, 7.5.4.8, 7.5.4.14, 7.5.4.16, 7.5.4.17, 7.6.3, 12.3.3, 12.3.3.6
- **Improvements for DAQ data consistency**
  Chapter 4.1.5, 4.1.12, 4.1.13, 7.5.4.14
- **Slave enforced calibration page switching and resource state changes**
  Chapter 4.4.4, 7.5.1.4, 7.5.1.8, 7.7.12
- **Command space enhancement**
  Chapter 7.2, 7.4, 7.5
- **POD standard related requirements**
  Chapter 7.2, 7.4, 7.5.1.7
- **GET_VERSION command**
  Chapter 7.4

What’s New?

Base Specification (Protocol Layer) contd.

- **Enhancements of ODT optimization types**
  Chapter 4.1.9, 7.5.4.11, 7.5.4.14
- **START_STOP_SYNC command enhancement**
  Chapter 7.5.4.5, 12.3.3, 12.3.3.4

What's New?

Transport Layer

- **XCP on CAN**
  - Chapter 7.3: Supporting Efficient DAQ Data Transfer (AML: `MEASUREMENT_SPLIT_ALLOWED`)
- **XCP on ETH**
  - Chapter 4.3.2: Packet Alignment: Added Fill space to the XCP on Ethernet message to ensure packet alignment (AML: `PACKET_ALIGNMENT`)
  - Chapter 5.2: Command `GET_SLAVE_ID_EXTENDED`
  - Chapter 5.3: Command `SET_SLAVE_IP_ADDRESS`

Compatibility

- XCP V1.4 is compatible to XCP V1.3 with regard to the communication protocol.
- Each new feature is optional, an ECU may implement a subset.