ASAM | Association for standardisation of automation and measuring systems

Release Presentation

P2016-09_AE_MCD-2_MC_BS_V1-7-1_MAIN

ASAM MCD2-MC V1.7.1

Release Date: 2018-02-06

Introduction

- The focus of the standard is still the description of the ECU functionality from a measurement & calibration point of view
- The maintenance development was based on several user requests
- Main Topics
  - Complex data types for dependent parameters. A dedicated set of data types is needed to cover use cases in the field
  - Improvement of explanation for several keywords
  - An attribute for functions to transfer Autosar information to the calibration engineer
  - Description of data size to detect a safe data exchange with XCP
  - Float16 data type for special applications
  - Add a symbol link for typedefs
  - Increase of the max string length to 1024

Content

- **Deliverables**
- Marketing
- Main Topics
- Compatibility

Deliverables

- Specification document
  - New functionality included
  - Bug fixes / Optimized descriptions
- Revision History
- A2L example file (ECU Container)
  - Updated with new examples
- ASAP2 Checker (planned for End of 2018)
  - Update checker rules
  - Checker checks V1.5.1, V1.6.1, V1.7.0, V1.7.1

Content

- Deliverables
- **Marketing**
- Main Topics
- Compatibility

Marketing

- The new version can be used without modifying existing tool chains
- The complex dependent parameters allow to describe a basic set of dependencies for all data types
- The ASAP2 Checker strategy is continued

Content

- Deliverables
- Marketing
- **Main Topics**
- Compatibility

What's New?

AML – Language Description

- New general description
  - Describes with an example the usage of AML sections
  - Explains with several steps the IF_DATA section and the usage together with other keywords
- declaration_list
  - New AML terminal to allow a list of declarations
- Value ranges for predefined data types
  - Definition of concrete width in byte for all predefined data types

What's New?

FUNCTION – Autosar Attributes

- Harmonization with Autosar

- AR_COMPONENT describes the category of the function
  - Attribute "ComponentType" describes the related usage in Autosar
  - Possible values are
    - "ApplicationSwComponentType"
    - "ParameterSwComponentType"
    - "NvBlockSwComponentType"
    - "CompositionSwComponentType

- The calibration engineer may use this information when working with the function

What's New?

CONTAINER – Unique IDs

- Not all information in a container is new
  - When the user of the container already owns some parts of the container he can skip these parts
- Attribute S
  - The container description file uses the attribute S from ASAM Container Catalog to add a unique ID
- Usage
  - The tool that creates the container may use an ID that allows identifying the source of the file (e.g. location in enterprise database)
  - The operating tool may use this ID to check whether the file is already available locally. This allows prohibiting doublets of the same files.

What's New?

FLOAT16

- For high performance computing it makes sense to use compact data types
- FLOAT16_IEEE is based on the IEEE Standard
  - 16bit floating point
  - Used for e.g. graphic processors
- ALIGNMENT_FLOAT16_IEEE
  - Alignment in the memory

What's New?

White Space handling

- Whitespace is defined as the separator between all language elements e.g. data types or keywords.

- Allowed white spaces

Ascii Code	Code	designation
0x09	HT	Horizontal Tabulation
0x0a	LF/NL	Line Feed / New Line
0x0b	VT	Vertical Tabulation
0x0c	FF/NP	Form Feed / New Page
0x0d	CR	Carriage Return
0x20	SPC	Space

What's New?

Precedence Rules for AXIS_DESCR and AXIS_PTS

- AXIS_DESCR and AXIS_PTS support partly the same keyword parameters

- The description is extended to cover also the optional keywords
  - If a CHARACTERISTC (Curve, Map …) uses in AXIS_DESCR a reference to AXIS_PTS (COM_AXIS or RES_AXIS), then the parameters from AXIS_DESCR shall take precedence.
  - Exceptions to this rule are the parameters MaxAxisPoints, DEPOSIT, BYTE_ORDER and MONOTONY, which shall be taken from AXIS_PTS.

What's New?

Dependent Parameters – Complex Data Types

- For an observer component it is important that it uses the same values as the main component.
  - The observer runs the same code and checks whether there are differences to the main controller
  - All characteristics must have the same values to get the same result

- Dependent parameter support now a dedicated set of rules for complex data types.
  - Part of the rules as example

Only the output values are calculated with the formula. The axis values are copied identically. The formula is applied to each cell of the CURVE.
The dependent curve must have the same number of axis points. Each cell of the 1st input CURVE is calculated with the related cell of the 2nd CURVE.
The dependent array must have the same size as the input. Each cell of the 1st input VAL_BLK is calculated with the related 2nd input VAL_BLK.

What's New?

SYMBOL_TYPE_LINK

- SYMBOL_TYPE_LINK can be used in the context of TYPEDEF_STRUCTURE or STRUCTURE_COMPONENT.
- It specifies the name of a symbol within a linker map file or debug file that describes a class, class member, structure or structure component and corresponds to the respective TYPEDEF_STRUCTURE or STRUCTURE_COMPONENT of the A2L file.
- Using this information, an automatic update of data type, size and offsets can be performed according to a provided linker map file or debug file.

What's New?

ADDRESS_TYPE

- ADDRESS_TYPE can be used in the context of INSTANCE and STRUCTURE_COMPONENT.

- It specifies the whether it is a direct address or a pointer to the address.

Content

- Deliverables
- Marketing
- Main Topics
- **Compatibility**

Compatibility

- Downward compatible to earlier ASAP2 versions
  - All former keywords are supported
  - Only compatible extensions
- Checker (release planned End of 2018)
  - Checker for new ASAM MCD-2 MC V1.7.1
  - Extensions for ASAM MCD-2 MC V1.7.1