



Association for Standardisation of
Automation and Measuring Systems

この日本語版はオリジナルの英語版の翻訳であり、情報提供のみを目的としている。齟齬がある場合、英語版の ASAM MCD-2 MC 標準ドキュメントが優先される。

The Japanese version is a translation of the original in English for information purposes only. In case of a discrepancy, the English version of the ASAM MCD-2 MC standard document shall prevail.

ASAM MCD-2 MC (ASAP2 / A2L)

Data Model for ECU Measurement and
Calibration

Version 1.7.0

Date: 2015-02-26

Base Standard

© by ASAM e.V., 2015

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

Table of Contents

Foreword 序文	5
1 Introduction はじめに	6
1.1 Overview 概要.....	6
1.2 Motivation 動機.....	6
1.3 Scope スコープ.....	7
1.4 History 歴史.....	9
2 Relations to Other Standards 他規格との関連	10
2.1 Backward Compatibility to Earlier Releases 前リリース版との下位互換性....	10
2.1.1 Incompatibility of Keyword FORMULA 式のキーワードの非互換性.....	10
2.1.2 Restriction for Brackets 括弧の制約.....	11
2.1.3 Win32 APIs for Seed&Key and Checksum Calculation Seed&Key とチ ェックサム計算用 Win32 APIs	12
2.1.4 Optional numbers for MATRIX_DIM MATRIX_DIM の任意の数字	12
2.2 References to Other Standards 他規格の参照.....	12
2.2.1 ASAM MCD-2 MC within the ASAM AE Overall Context ASAM AE 全 体における ASAM MCD-2 MC	12
2.2.2 ASAM MCD-1	13
2.2.3 ASAM MCD-2	14
2.2.3.1 ASAM MCD-2 D ODX.....	14
2.2.3.2 ASAM MCD-2 NET.....	14
2.2.4 ASAM MCD-3	15
2.2.5 More ASAM Standards with an ASAM MCD-2 MC relation ASAM MCD-2 MC に関連した他の ASAM 標準仕様	15
3 Division of the description data 記述データ区分	17
4 Format of the description file 記述ファイル形式	20
4.1 Encoding of the A2L file A2L ファイルのエンコーディング	20
4.1.1 Unicode Transformation Format.....	20
4.1.2 Byte-Order Mark	20
4.2 Hierarchic division of the keywords.....	22
4.3 Name Spaces ネームスペース	31
4.3.1 Functions 関数.....	31
4.3.2 Groups グループ.....	31
4.3.3 Variants バリエーション.....	31
4.3.4 Measure and adjustable objects 測定とアジャスタブルオブジェクト	31
4.3.5 Transformer トランスフォーマー.....	32
4.3.6 Conversion methods 変換規則.....	32

4.3.7	Conversion tables 変換テーブル.....	32
4.3.8	Memory segments メモリーセグメント.....	32
4.3.9	Record layouts レコードレイアウト.....	32
4.3.10	Type definitions 型定義.....	32
4.3.11	Structure components 構造体.....	32
4.4	Predefined data types 定義済みデータ型	33
4.5	Mapping of predefined data Types to ASAM data types 予約データ型と ASAM データ型の関係	38
4.6	Comments コメント	38
4.7	Alphabetical list of keywords キーワードのアルファベット順リスト.....	39
4.7.1	General 概要	39
5	Bibliography 参考文献	41

Foreword 序文

ASAM MCD-2 MC describes a data model for ECU Measurement and Calibration via a non XML format. Such data resides inside the memory of the ECU. The format definition makes the data accessible through application systems for tuning real-time testing. The description contains information about data types, dimensions, record layouts and memory locations of ECU variables. Further information describes how the variable values shall be converted into human-readable quantities and displayed in an MC-system.

In addition to the data description, the standard allows to describe the device interface between the MC-system and the ECU for read- and write access. As a result, the ASAM MCD-2 MC description contains all information in one place that is needed to access, modify, interpret and display ECU-internal variables.

ASAM MCD-2 MC は、非 XML 形式による ECU 測定および適合の為のデータモデルを記述する。このデータは ECU のメモリに属している。この記述フォーマットは、アプリケーションシステムを通じて適合およびリアルタイムテストのデータアクセスを可能にする。この記述にはデータ型、次元、レコードレイアウトおよび ECU 内部値のメモリ位置を含む。さらなる情報として、値がどのように人間に解読可能な数値や測定適合システムで表示される為に変換されるかを記述している。

データの記述において、この標準は測定適合システムと ECU 間のリードおよびライトアクセスのデバイスインターフェイス記述を可能にする。この結果、ASAM MCD-2 MC 記述は ECU 内部値のアクセス、変更、解釈および表示に必要な情報を一つにまとめている。

The format definition describes the following content:

- A2L description file keywords
- AML format description
- AML template
- A2L and AML example

このフォーマット定義は以下の内容を記載する。

- A2L 記述ファイルのキーワード
- AML フォーマット記述
- AML テンプレート
- A2L および AML の例

1 Introduction はじめに

1.1 Overview 概要

ASAM MCD-2 MC defines a description format for internal ECU variables used for measurement and calibration purposes.

Calibration means the adaption of characteristics (scalars, curves and maps) within the functional code of ECUs to achieve and optimize an appropriate system behavior. These calibration operations are either performed manually by a calibration engineer or are executed by external client applications, such as optimization programs or test bed automation systems. This adaption requires a WRITE-access to the ECU to set a new value or a READ-access to retrieve the current one.

Whether this calibration already produces the intended effect or not will normally be checked by inspection of other ECU variables by MEASUREMENT access. Therefore, the host tool configures so-called measurement tasks, transmits them to the ECU and henceforth takes the values which are automatically sent by the ECU.

The description is typically used by measurement & calibration systems (MC-systems) for tuning scalar constants, curves and maps of the ECU software and to record the system's response via measurement variables during real-time testing. The description contains information about data types, dimensions, record layouts and memory locations of ECU variables. Further information describes how the variable values shall be converted into human-readable quantities and displayed in an MC-system. Furthermore, it describes the HW interface of the ECU for device driver configuration of the application system. The standard is also known as "ASAP2".

ASAM MCD-2 MC は、測定および適合目的に使用される ECU 内部変数の記述形式を定義する。適合とは、適切なシステムの動作の実現と最適化を行う ECU の機能コードに付随する特性(スカラー、カーブ、マップ)の適用を意味する。これらの適合作業は、適合エンジニアが手動で行うか最適化プログラムまたはテストベッドの自動化システムなどの外部クライアントアプリケーションによって実行される。

この適合は、すでに意図した効果を生み出しているか否か、他の ECU 変数の測定によるアクセスによってチェックされる。したがってホストとなるツールは、測定タスクと呼ばれるこの作業を設定し、測定する変数を ECU によって後に自動的に送るための設定項目を ECU へ送る。

本標準の記述(A2L)は、ECU ソフトウェアのスカラー・カーブ・マップ定数の適合、およびリアルタイムテスト中の変数測定によるシステムの応答の記録を行う測定・適合システム(MC-System)によって典型的に使用される。A2L には、データの型および次元、レコードレイアウトと ECU 変数のメモリ位置に関する情報が含まれている。詳細としては、変数の値を人間が読める量にどの様に変換され、MC-System でどの様に表示されるかが記述される。さらには、ECU のハードウェアインターフェースの為のアプリケーションシステムのデバイスドライバ設定が記述される。本標準は「ASAP2」としても知られている。

1.2 Motivation 動機

An essential part of ECU software development is the calibration of control strategy's parameters. This means the adaption of scalars, curves and maps to achieve an optimized and appropriate system behavior. Internal variables need to be read from the ECU to evaluate the effectiveness of the calibrated software. Such operations are carried out by tools which need a detailed description of the calibration parameters and internal variables. They furthermore need to have a description of the device interface to the ECU for read and write

access. This description is typically produced by function developers, software engineers, tool & instrumentation experts, and is used by calibration engineers.

The standard allows the connection of software development tools, calibration tools and ECU calibration interfaces with a neutral description format (A2L). Tools that support the description format are able to exchange and process the included information. Hence, no vendor-specific or technology-specific dependencies are required between tools of an ASAM-compliant calibration tool-chain.

ECU ソフトウェア開発の重要な部分は、ECU 内部変数パラメータの適合である。これはスカラー、カーブやマップの値が最適化され、適切なシステム動作を実現することを意味する。

内部変数は、適合されたソフトウェアの有効性を評価するために、ECU から読み取る必要がある。この操作は、適合パラメータおよび内部変数の詳細な評価が出来るツールによって実行される。このツールは、さらに、ECU への読み書きアクセスのためのデバイスインターフェイスを持つ。本標準の記述(A2L)は、典型的には、関数の開発者、ソフトウェアエンジニア、ツール・計装専門家によって生成され、適合エンジニアによって使用される。

本標準では、中立的な記述フォーマット(A2L)でソフトウェア開発ツール、適合ツールと ECU 適合インターフェイスとの接続を可能にする。A2L 記述形式をサポートするツールは、含まれている情報の処理、交換が可能である。したがって、ASAM 準拠の適合ツールチェーンのツール間では、特定ベンダーまたは特定テクノロジーには依存しない。ASAM MCD-2 MC 規格を採用することで、コンバータを必要とせずデータ交換を可能にし、エンジニアやサプライヤとの間に明確な仕様の交換を容易にします (<http://www.asam.net/>より引用)。

1.3 Scope スコープ

The ASAM MCD-2 MC standard (market name ASAP2 / A2L) was developed to take into consideration the needs of all groups involved in the calibration process. The standard defines a description format that describes the calibration parameters (called CHARACTERISTIC) and internal variables (called MEASUREMENT) of ECU software. The description includes elementary information like addresses, data types, dimensions, identifiers and much more. To convert the ECU internal characteristic and measurement implementation values into physical values, ASAM MCD-2 MC describes computation methods for their conversion between both representations. Calibration engineers can work with the ECU data in a familiar format without having to understand ECU-internal data formats. Software engineers can provide this data to them or even get the description files automatically generated from code generators. An include mechanism ensures that description files can originate from different sources.

The ECU normally stores the measurement and calibration quantities internally in an implementation optimized format. This format is very often a fixed-point format. Outside the ECU physical models are used. The ASAM MCD-2 MC standard describes by so-called record layouts how data are stored inside the ECU and which computation methods are needed to transform the ECU internal data representation into the physical one and vice versa.

The ASAM MCD-2 MC standard also allows to describe and configure the ECU interfaces or vendor specific extensions by a meta description language (AML). For ASAM standardized ECU interfaces, such as CCP and XCP the content of these AML parts are also standardized. But there are also a lot of vendor specific instantiations in the market which use this mechanism. Measurement and calibration tools are normally only used during development phase of ECUs. They allow a direct, address-oriented write- and read-access but also a synchronous, continuous measurement access to ECU internal variables.

MC tools also offer features for flashing of new software versions comprising of new code and/or new parameter sets using the relevant ASAM MCD-1 interfaces. For calibration and

flashing purposes the ASAM MCD-2 MC standard describes the memory segment configuration to rebuild code and data externally.

Address-oriented information of ASAM MCD-2 MC files may become obsolete with a new ECU software version running a new compiler/linker run because variables may be relocated in the memory. Therefore, code generator tools but also other utilities very often generate or update ASAM MCD-2 MC files.

The ASAM MCD-2 MC standard is widely used in the automotive industry and supported by every major calibration tool on the market.

It is used by many code generators, calibration and diagnostics tools, rapid control prototyping tools, data loggers, measurement systems, automation systems, etc.

The standard is aligned with other ASAM standards like ASAM MCD-1 XCP/CCP, ASAM CDF and ASAM MCD-3 MC/ASAP3.

ASAM MCD-2 MC 標準 (通称 ASAP2 / A2L) は、適合プロセスに関係する全ての関係者のニーズに対応する為に開発された。本標準は、ECU ソフトウェアの適合パラメータ (CHARACTERISTIC と呼ばれる) と内部変数 (MEASUREMENT と呼ばれる) を記述するフォーマットを定めており、その記述ファイルは、アドレス、データ型、ディメンション、識別子など多くの基本情報を含む。

ECU 内部の適合パラメータと内部変数の実行値を物理値に変換するために、ASAM MCD-2 MC ではその変換についての双方向の変換を記述する。適合エンジニアは ECU 内部のデータフォーマットを理解しなくとも、なじみやすいフォーマットで記述された ECU データを使用して作業を行うことができる。ソフトウェア・エンジニアがこのデータを適合エンジニアへ提供することも、コードジェネレーターから自動生成された記述ファイルを手に入れることもできる。また“インクルード”の仕組みにより、様々なソースからの記述ファイルの作成が可能である。

ECU は通常、測定値と適合値を実行に最適化されたフォーマットで内部に記憶している。このフォーマットは、ほとんどの場合固定少数点フォーマットである。ECU の外では物理モデル (物理式、物理量の世界) が使用される。ASAM MCD-2 MC 標準では、ECU 内部データの記憶方法に関するいわゆる“レコードレイアウト”と、その ECU 内部のデータ表記から物理値への変換、その逆の物理値から内部データ表記への変換に必要な計算方法を記述できる。

ASAM MCD-2 MC 標準は、メタ記述言語 (AML) により、ECU インターフェース又はベンダー特有の拡張仕様を記述、構成することを許容している。CCP や XCP のような ASAM で標準化された ECU インターフェース向けにそれら AML 部分の内容も標準化されている。しかしながら、市場にはこの仕組みを使用したベンダー特有の事例も多く存在する。

測定適合ツール (MC ツール) は通常、ECU の開発フェーズでのみ使用される。MC ツールは ECU 内部変数に対する直接的なアドレス指定による読み書き、かつ連続的な同期測定を可能にする。

また、MC ツールは、関連する ASAM MCD-1 インターフェースを使用して、新しいコードや新しいパラメータセットで構成される新たなソフトウェア・バージョンへフラッシュ・プログラミングする為の機能も提供する。ASAM MCD-2 MC 標準では、適合とフラッシュ・プログラミングを行う目的で、外部からコードとデータを再構築する為のメモリーセグメント構成を記述する。

ASAM MCD-2 MC ファイルのアドレス指定情報は、新たなコンパイラ/リンカを実行している新しい ECU ソフトウェア・バージョンでは古くなっているかもしれない。なぜなら変数はメモリ内で再配置される可能性があるからである。したがって、コード生成ツールや他のユーティリティ・ツールは ASAM MCD-2 MC ファイルを頻繁に生成、または更新する。

ASAM MCD-2 MC 標準は広く自動車産業で使用されており、市場のあらゆる主要な適合ツールでサポートしている。

それは、多くのコードジェネレーター、適合・診断ツール、ラビットプロトタイピングツール、データロガー、測定システム、オートメーションシステムなどで使用される。

この標準は、ASAM MCD-1 XCP/CCP や ASAM CDF、ASAM MCD-3 MC/ASAP3 のような他の ASAM 標準と整合が取られている。

1.4 History 歴史

First versions of the standard were developed already before foundation of ASAM e.V. in 1998. These versions were named ASAP-2. They have been part of a 3-layer base architecture, which is described in chapter [2.2](#).

ASAP-2 was renamed to ASAM MCD-2 MC. MCD stands for Measurement, Calibration and Diagnostics.

この標準の初期バージョンは、1998年のASAM e.V.の設立前に既に開発されていた。それらのバージョンはASAP-2と命名した。それらは2.2章で解説されている3レイヤーのベースアーキテクチャーの一部であった。

ASAP-2はASAM MCD-2 MCに名称を変えた。MCDとは、測定、適合そして診断のことである。

2 Relations to Other Standards 他規格との関連

2.1 Backward Compatibility to Earlier Releases 前リリース版との下位互換性

The version ASAM MCD-2 MC V1.7.0 is in general downward compatible to the former version ASAM MCD-2 MC V1.6.1. This means that the keywords and keyword combinations used to describe ECU software are in the same way supported as in the former version.

The ASAM MCD-2 MC reuses the same notation format of the former ASAP2 standards to ensure a downward compatibility. This is necessary as a wide set of existing ASAP2 tools is in the market and a switch to a different language format (e.g. XML) leads to high efforts to modify these tools.

Because of the maintenance some details are no longer downward compatible. This chapter lists the critical ones.

ASAM MCD-2 MC V1.7.0は、前バージョンのV1.6.1に対して、概ね下位互換性がある。つまり、ECUソフトウェアの説明に使用するキーワードやキーワードの組み合わせが、前バージョンのものと同じくサポートされている。ASAM MCD-2 MCは、下位互換性を保証するために、前バージョンのASAP2規格と同じ表記フォーマットを使用する。ASAP2を取り扱うツールは世の中で広く利用されており、異なる言語フォーマット(XMLなど)に変更すると、既存ツールの改修に多大な労力を必要とする。

しかしながら、メンテナンスのために詳細としては下位互換性がないものもある。本章では、重大なものを述べる。

2.1.1 Incompatibility of Keyword FORMULA 式のキーワードの非互換性

The former definition of the keyword was not compatible to ANSI-C notation. The ANSI-C compatibility is important to convert formula descriptions easily between different systems engineering system (compiler) and software description systems.

In detail these are the logical operators:

Since ASAM MCD-2 MC V1.6.0 (ANSI-C) the meaning of some operators differs from ASAM MCD-2 MC V1.5.

以前のバージョンにおける式の定義は、ANSI-C表記と互換性が無かった。ANSI-C準拠は、コンパイラなどのエンジニアリングツールとソフトウェアを記述するシステム間で、式の記述転換を簡単に行う上で重要である。

詳細には、論理演算子である。具体的には、ASAM MCD-2 MC V1.6.0 (ANSI-C)では、論理演算子の意味をASAM MCD-2 MC V1.5から変更しているものがある。

Table 1 formula operator compatibility 式演算子互換性

Operator	ASAM MCD-2 MC V1.5.1	ASAM MCD-2 MC V1.6.0 (ANSI-C)
&	logical AND	bitwise AND
	logical OR	bitwise OR
XOR	exclusive OR	not supported
~	logical NOT	bitwise NOT
^	Power	bitwise exclusive OR
&&	not supported	logical AND

Operator	ASAM MCD-2 MC V1.5.1	ASAM MCD-2 MC V1.6.0 (ANSI-C)
	not supported	logical OR
!	not supported	logical NOT
ln(x)	supported, but not specified	not supported
log(x)	supported, but not specified	natural logarithm
log10(x)	not supported	decimal logarithm
演算子	ASAM MCD-2 MC V1.5.1	ASAM MCD-2 MC V1.6.0 (ANSI-C)
&	論理積	ビット積
	論理和	ビット和
XOR	排他的論理和	【サポート対象外】
~	否定	ビット否定
^	乗	排他的ビット和
&&	【サポート対象外】	論理積
	【サポート対象外】	論理和
!	【サポート対象外】	否定
ln(x)	【サポート対象だが、詳細は定められていない】	【サポート対象外】
log(x)	【サポート対象だが、詳細は定められていない】	自然対数
log10(x)	【サポート対象外】	常用対数

Since ASAM MCD-2 MC V1.6.0 (ANSI-C) the notation of some operators differs from ASAM MCD-2 MC V1.5.

ASAM MCD-2 MC V1.6.0 (ANSI-C) 以降、演算子の表記法をASAM MCD-2 MC V1.5から変更しているものがある。

Table 2 formula operator notation 式演算子表記法

ASAM MCD-2 MC V1.5.1	ASAM MCD-2 MC V1.6.0 (ANSI-C)
arcsin(x)	asin(x)
arccos(x)	acos(x)
arctan(x)	atan(x)

Note: If the keyword ASAP2_VERSION is missing (former ASAM MCD-2 MC version) or states a version smaller than V1.6.0 tools shall use the formula interpretation of ASAM MCD-2 MC V1.5.1.

注記: キーワード ASAP2_VERSION がない(前 ASAM MCD-2 MC バージョン)、または、V1.6.0 より小さいバージョンを記載の場合、ツールは ASAM MCD-2 MC V1.5.1 の式解釈を使用することとする。

2.1.2 Restriction for Brackets 括弧の制約

Since ASAM MCD-2 MC V1.6.0 always brackets of the form '/begin' '/end' are requested. Curly brackets '{' '}' are no longer supported.

ASAM MCD-2 MC V1.6.0 以降、常に '/begin' '/end' の種類の括弧が要求される。波括弧 '{' '}' はサポートされない。

2.1.3 Win32 APIs for Seed&Key and Checksum Calculation Seed&Key とチェックサム計算用 Win32 APIs

The definition of Win32 APIs for Seed&Key and checksum calculation which was formerly part of the ASAM MCD-2 MC V1.5.1 specification is no longer part of this specification. These definitions are now found at ASAM Seed&Key.

以前 ASAM MCD-2 MC V1.5.1 仕様の一部であった Seed&Key とチェックサム計算用の Win32 APIs の定義はもはや本仕様の一部ではない。これらの定義は現在 ASAM Seed&Key [7] でみられる。

2.1.4 Optional numbers for MATRIX_DIM MATRIX_DIM の任意の数字

ASAM MCD-2 MC V1.7 defines the dimensions as optional numbers. This allows to add only the dimensions used.

In earlier versions (\leq V1.6.1) always 3 dimensions are given. To describe objects with less than 3 dimensions ASAM MCD-2 MC V1.6.1 defined a rule that not used dimensions are stated as '1'.

This leads to inconsistency in case of usage of MATRIX_DIM for 1 or two dimensions. E.g. MATRIX_DIM 4 2 1 describes in V1.7 a 3-dimensional object (z-dimension = 1) and in V1.6.1 a 2-dimensional object.

To avoid conflicts MATRIX_DIM must be handled version dependent.

ASAM MCD-2 MC V1.7 は次元を任意の数字と定義している。これにより使用する次元のみを追加することができる。

初期バージョン (\leq V1.6.1) では常に 3 次元が与えられる。3 次元よりも小さい次元で物体を記述するために、ASAM MCD-2 MC V1.6.1 では、使用されない次元は「1」と記載されるというルールを定めた。

これは 1 次元また 2 次元に MATRIX_DIM を使用する場合、矛盾をもたらす。

例: MATRIX_DIM 4 2 1 は、V1.7 では 3 次元のオブジェクト (Z 次元 = 1)、V1.6.1 では 2 次元のオブジェクトを記述する。

矛盾を避けるために、MATRIX_DIM はバージョン依存で扱われなければならない。

2.2 References to Other Standards 他規格の参照

2.2.1 ASAM MCD-2 MC within the ASAM AE Overall Context ASAM AE 全体における ASAM MCD-2 MC

The MCD standards of ASAM Automotive Electronics are structured in a 3 layer system as follows:

以下のように ASAM 自動車エレクトロニクスの MCD 規格は、3 層システムで構成されている。

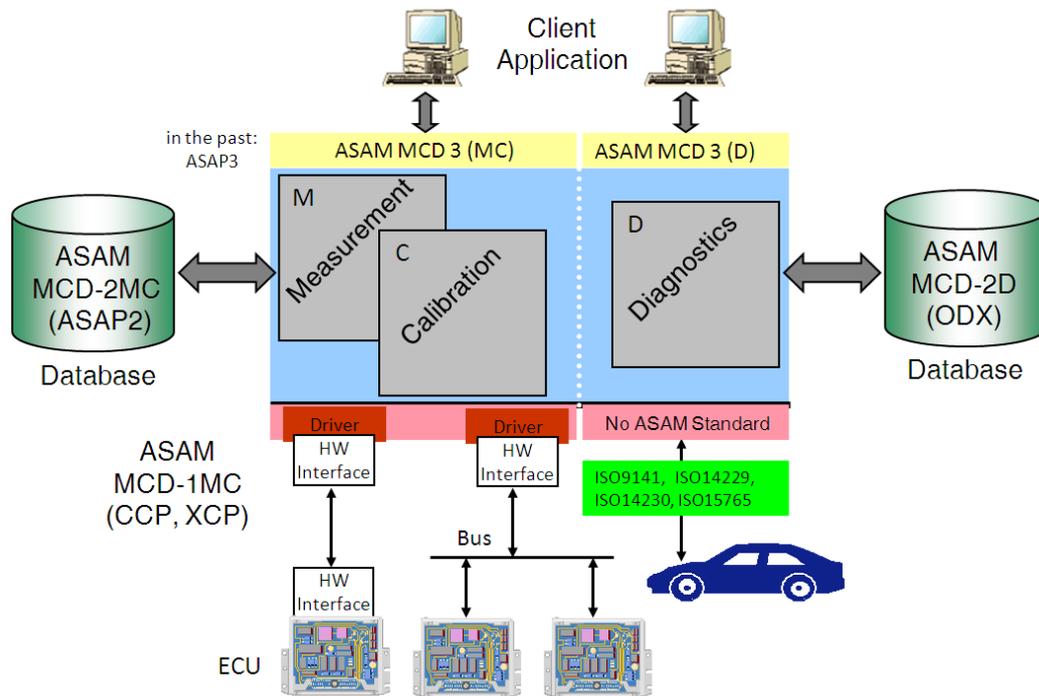


Figure 1 Structure of the MCD standards of ASAM AE

2.2.2 ASAM MCD-1

This summarizing term today denotes a set of standards to define the data interface primarily of ECUs. As shown in Figure 1 the ASAM defined interfaces are used only for measurement and calibration tasks.

Diagnostic protocols and interfaces are not standardized by ASAM but by ISO (ISO 9141 [8], 14229 [9], 14230 [10], 15765 [11] or the PDU-API in ISO/DIS 22900-2 [13]).

The most import standard in this sector, implemented in many ECUs worldwide, is the so-called CAN Calibration Protocol. All market relevant calibration tools support this protocol as a base feature. But also specific measurement equipment, such as data loggers are able to connect to ECUs directly via the CAN bus.

Based on several years of experience with this CAN bus limited protocol ASAM developed the XCP standard, which in contrast to CCP is defined independently from the specific transport layer. Today several layers are available, such as CAN, USB, Ethernet, Flexray and Sxl.

Very important for the success of XCP was the functional extension. As CCP was restricted to measurement and calibration, XCP also added a standardized option to write values back to the ECU in a synchronous manner (stimulation).

CCP as well as XCP allow to flash ECUs during development phase. This flashing access will normally be eliminated when the ECU is given to production and therefore is not available during service phase. Here the corresponding diagnostics services are applied.

この要約用語は、今日の ECU の主なデータインターフェースを定義するための規格の集合を表す。図 1 に示すように、ASAM 定義のインターフェースは、測定と適合タスクに使用されている。診断プロトコルとインターフェースは、ASAM ではなく、ISO によって規格化される (ISO 9141 [ISO 9141]、14229 [ISO 14229]、14230 [ISO 14230]、15765 [ISO 15765] または ISO / DIS 22900-2 PDU-API [ISO 22900-2])。

世界中の多くの ECU に実装される分野のインポート規格は、いわゆる CAN キャリブレーションプロトコルである。適合ツールに関連する全ての市場は、基本機能としてこのプロトコルをサポートする。

しかし、データロガーのような特定の測定機器は、CAN バスを介して直接、ECU に接続することができる。

この CAN バスにおけるプロトコルの数年の経験に基づいて、ASAM は CCP とは対照的に、特定のトランスポート層から独立して定義されている XCP 規格を開発した。今日、いくつかの層は、CAN、USB、イーサネット、FlexRay、SxI として利用できる。

XCP の成功のために非常に重要な機能拡張があった。CCP は、測定および適合に限定されているが、XCP は、同期的に ECU に値を書き込むための標準化されたオプション (Stimulation) を追加した。

CCP と同様に XCP は、開発段階での ECU をフラッシュすることができる。サービス・フェーズでは使用可能でなくても、ECU が生産される場合、このフラッシングアクセスは通常、排除される。この時点で、対応する診断サービスが適用される。

2.2.3 ASAM MCD-2

The different ASAM MCD-2 formats build the data basis of an MCD system. ASAM MCD-2 MC describes the necessary data for an MC oriented access, ASAM MCD-2 D ODX [6] for a service-oriented diagnostics access and ASAM MCD-2 NET [3] for an access via the different bus systems.

各 ASAM MCD-2 のフォーマットは、MCD システムのデータ基盤を構築する。ASAM MCD-2 MC は MC 指向アクセスについて、ASAM MCD-2 D ODX はサービス指向の診断アクセスについて、そして ASAM MCD-2 NET は異なるバスシステム間のアクセスについて、それぞれ必要なデータを記述する。

2.2.3.1 ASAM MCD-2 D ODX

ASAM MCD-2 D ODX in its versions since 2.0 is a unique, open XML exchange format for diagnostics data. The seamless data exchange between different partners along the process chain (suppliers, OEMs or service partners) is a very important process improvement.

Diagnostic tools like service testers or more development oriented tools can be parameterized via this format. The ODX standard defines an object-oriented data model, which is described in UML (Unified Modeling language). Inheritance and associations help to avoid data redundancies.

In contrast to the ASAM MCD-2 MC standard, ODX data describe the parameters and access information for a diagnostic service oriented ECU access.

The ODX standard is also available as ISO 22901-1 [12].

ASAM MCD-2 D ODX はバージョン 2.0 より、診断データのためのユニークでオープンな XML によるデータ交換フォーマットである。プロセスに沿った様々なパートナー (サプライヤ、OEM、サービスパートナー) の間でシームレスなデータ交換を行うことは、プロセスの改善に非常に重要である。

サービステスターのような診断ツールや開発志向ツールはこのフォーマットによってパラメータ化されます。ODX 標準仕様は UML (Unified Modeling Language) で記述されたオブジェクト指向のデータモデルを定義する。継承と関連はデータ冗長の回避に役立つ。

ASAM MCD-2 MC 標準とは対照的に、ODX データは診断サービス指向の ECU アクセスに必要なアクセス情報やパラメータを記述する。

ODX 標準仕様は ISO 22901-1 としても入手できる。

2.2.3.2 ASAM MCD-2 NET

ASAM MCD-2 NET is also an XML format. It is capable to describe the entire vehicle communication network. All relevant bus technologies, such as CAN, MOST, LIN or Flexray are supported. Information exchange between tools of different suppliers is a standard ASAM

MCD-2 NET use case. Partial information can be completed step by step along the development progress.

ASAM MCD-2 NET も XML フォーマットであり、車載通信ネットワーク全体の情報を記述でき、CAN、MOST、LIN、FlexRay などの全ての車載通信技術をサポートする。

異なるサプライヤのツール間における情報交換は、ASAM MCD-2 NET の標準的な使用例であり、開発の進行に沿って段階的に部分情報を補完できる。

2.2.4 ASAM MCD-3

ASAM MCD-3 specifies an object-oriented programming resp. remote-control interface to an MCD-server system.

This standard comprises of a base standard which is coded as a technology independent UML model with corresponding interface implementations for (D)COM, Java and C++.

The functional columns M (measurement), C (calibration) and D (diagnostics) can be applied independently of each other, but also in combined manner as M, MD, MC or entire MCD systems. Common parts, such as project or hardware setup, are defined as single source.

There is a clear market trend that more and more combined and integrated MCD systems will become available.

The diagnostics part of ASAM MCD-3 [4] is available as 22900-3 [14].

It is important to mention the well-established remote-control interface ASAP-3 MC [5]. This standard was developed in the early nineties for RS232 and TCP/IP based communication between test bed automation systems and calibration tools. It is still in usage in many applications until today.

ASAM MCD-3 はオブジェクト指向プログラミングにおける、MCD サーバシステムへのリモート制御インターフェイスを規定する。この標準仕様は COM/DCOM、Java、C++ で実装されたインターフェイスに対応した、テクノロジーに依存しない UML モデルでコード化された基本規格で構成されている。機能カラム M (measurement)、C (calibration)、D (diagnostics) はそれぞれ独立して適用できるが、M、MD、MC または MCD 全体として組み合わせられた方法でも利用可能である。プロジェクトまたはハードウェア設定などの共通部分はシングルソースとして定義される。市場の明確な傾向として、より多くの組み合わせと、統合された MCD システムが利用可能になることが挙げられる。ASAM MCD-3 の診断パートは ISO/DIS 22900-3 として入手可能である。

十分に確立された ASAP-3 MC 遠隔制御インターフェイスに配慮することは重要である。この標準仕様は自動テストベンチシステムと適合ツール間を RS232 や TCP/IP ベースの通信で接続するために、90 年代初頭に開発された。今日に至るまで多くのアプリケーションで利用されている。

2.2.5 More ASAM Standards with an ASAM MCD-2 MC relation ASAM MCD-2 MC に関連した他の ASAM 標準仕様

Beside the MCD standards additional standards either have been developed from scratch or have been modified and adapted based on pre-developments on MSR side.

The ASAM CDF [1] standard specifies an XML format to store calibration data, their level of maturity and other development process related data.

The ASAM MDF [2] standard allows to store measurement data in a very efficient binary format. MCD 規格上に、追加仕様が一から開発されたり、MSR 側の事前の開発に基づいて修正、適応されたりしている。

ASAM CDF 標準仕様は、適合データを格納するための XML フォーマットを規定する。適合状態のレベルやその他の開発プロセス関連データが含まれる。

ASAM MDF 標準仕様は、測定データが格納された非常に効率的なバイナリフォーマットを規定する。

3 Division of the description data 記述データ区分

The definition of the ASAM MCD-2 MC interface and hence the specification of the ASAM MCD-2 MC data base is aimed at defining a database independently of a computer or an operating system in such a way that a transparent and manufacturer-independent standard is established. As exchange format for such ECU descriptions *.a2l files are used.

From the calibration point of view the database in accordance with the ASAM MCD-2 MC interface contains the complete description of all control unit relevant data in a project. A project consists of project specific header, which is typically created by the project manager, data and one or more control unit specific descriptions. These control unit descriptions (= description of an ECU) include all conversion formulas and explanations about the applicable (adjustable) and measurable (non-adjustable) quantities and present a format description of the interface specific parameters. The measurement and calibration system needs only to evaluate the quantities (and their conversion etc.), but not the interface specific parameters. The latter are only passed on to the structures of the driver. To make sure that these structures are correctly filled the MCD must know the parameter type. The type is communicated with the ASAM MCD-2 MC metalanguage. As exchange format for such ECU IF Data descriptions *.aml files are used.

A project may include the control unit descriptions of various control units from different suppliers. The descriptions differ in terms of content, but use a common information storage methodology to allow for a global management of the project components. An INCLUDE mechanism allows to summarize the various control unit descriptions of various projects (Single-Source-Concept).

The ASAM MCD-2 MC database thus consists of a number of different subcomponents structured in accordance with the following diagram. The MODULE keyword denotes an independent ECU or device.

ASAM MCD-2 MC インターフェイス、つまり ASAM MCD-2 MC データベースの定義は、透過的でメーカーに依存しない標準が確立されるような方法でコンピュータまたはオペレーティングシステムから独立したデータベースとして定義することを目的としている。このような ECU を表現するための交換フォーマットとして* .a2l ファイルが使用されている。適合の観点から ASAM MCD-2 MC インターフェイスに従ったデータベースは、プロジェクト内のすべての制御ユニット関連データの完全な記述が含まれている。プロジェクトは、通常、プロジェクトマネージャー、データ、および 1 つ以上の制御ユニットの特定の記述によって作成されたプロジェクト固有のヘッダーで構成されている。これらの制御ユニットの記述(= ECU の記述)は、すべての変換式と適用可能量(調整可能)と測定量(調整不可能)に関する記述が含まれており、インターフェイス固有のパラメータのフォーマットの記述を提示する。測定および適合システムは、量(およびその変換など)だけでなく、インターフェイス固有のパラメータを評価する必要がある。後者(インターフェイス固有のパラメータ)は、ドライバの構成にのみ渡される。これらの構造が正しく記述されていることを確認するために MCD は、パラメータの型を知らなければならない。その型は、ASAM MCD-2 MC のメタ言語で定義されている。このような ECU のインターフェイスデータの記述のための交換フォーマットとして* .aml ファイルが使用されている。

プロジェクトは、異なるサプライヤからの様々な制御ユニットの制御部の説明を含むことができる。その記述は、内容的に異なるがプロジェクトコンポーネントのグローバル管理を可能にするために一般的な情報記憶方法を使用しています。INCLUDE 機構は、様々なプロジェクトの様々な制御ユニットの記述をまとめることができる(単一ソース・コンセプト)。

ASAM MCD-2 MC データベースは、このように、次の図に従って構造の異なるサブコンポーネントの数で構成される。MODULE キーワードは、独立した ECU またはデバイスである。

```
/begin PROJECT
  /begin HEADER
    /* Project description */
  /end HEADER

  /begin MODULE          /* Specific Device description*/
    /begin MOD_PAR
      /* Control unit management data */
    /end MOD_PAR

    /begin MOD_COMMON
      /* Module-wide (ECU specific) definitions */
    /end MOD_COMMON

    /begin CHARACTERISTIC
      /* Adjustable objects */
    /end CHARACTERISTIC

    /begin CHARACTERISTIC
    /end CHARACTERISTIC

    /begin AXIS_PTS
      /* Axis points objects */
    /end AXIS_PTS

    /begin AXIS_PTS
    /end AXIS_PTS

    ...
    /begin MEASUREMENT
      /* Measurement objects */
    /end MEASUREMENT

    /begin MEASUREMENT
    /end MEASUREMENT

    ...
    /begin COMPU_METHOD
      /* Conversion method */
    /end COMPU_METHOD

    /begin COMPU_METHOD
      .
    /end COMPU_METHOD
    /begin COMPU_TAB
      /* Conversion tables */
    /end COMPU_TAB

    /begin COMPU_TAB
    /end COMPU_TAB

    ...
    /begin FUNCTION
      /* Function allocations */
    /end FUNCTION
    /begin FUNCTION
    /end FUNCTION

    ...
```

```
/begin GROUP
    /* Groups */
/end GROUP

/begin GROUP
/end GROUP

...
/begin RECORD_LAYOUT
    /* Record layouts of adjustable objects */
/end RECORD_LAYOUT

/begin RECORD_LAYOUT
/end RECORD_LAYOUT
/end MODULE
...
/end PROJECT          /* further device descriptions can follow */
                    /* END OF PROJECT */
```

The following rules apply for a valid a2l file:

The ASAP2_VERSION keyword is mandatory and expected before the keyword PROJECT. The file must contain exactly one PROJECT. The PROJECT must contain at least one MODULE.

The keywords defined in the ASAM MCD-2 MC database are described in the following chapter.

以下の規則が有効な A2L ファイルに適用される。

ASAP2_VERSION キーワードは必須で PROJECT キーワードの前に必要。

a2l ファイルは、1 つの PROJECT が含まれている必要があり、その PROJECT には、少なくとも一つの MODULE が含まれている必要がある。

4 Format of the description file 記述ファイル形式

4.1 Encoding of the A2L file A2L ファイルのエンコーディング

ASAM MCD-2 MC files are used in different language areas. To support the different character sets used in the different language areas it is necessary to add to the ASAM MCD-2 MC file the information about the used character set.

The ASCII and ISO-8859-x character sets, defined for several language areas are not sufficient. The ASAM MCD-2 MC files needs to be exchangeable world wide.

ASAM MCD-2 MC ファイルは、異なる言語エリアで使用されている。異なる言語エリアで使用されている異なるキャラクターセットをサポートするために、ASAM MCD-2 MCファイルは使用されている言語の情報を追加する必要がある。

いくつもの言語エリアで定義されているASCIIとISO-8859-xのキャラクターセットは、満足いくものではない。ASAM MCD-2 MCファイルは、ワールドワイドに交換可能であることを必要とする。

4.1.1 Unicode Transformation Format

World wide exchange is supported by "Unicode Transformation Format" (UTF). Currently there are 3 relevant versions of UTF available: UTF-8, UTF-16, UTF-32.

UTF-8 is compact and supports nearly every possible character world wide. Therefore UTF-8 is the preferred encoding for ASAM MCD-2 MC files.

Note: Tools shall support at least UTF-8.

ワールドワイドな交換は、“Unicode Transformation Format”(UTF)によってサポートされている。現在、3つのUTFのバージョン、UTF-8, UTF-16, UTF-32が利用可能である。

UTF-8はコンパクトであり、世界中で使用しているキャラクター(文字)をほぼサポートしている。したがって、UTF-8はASAM MCD-2 MCファイルにとって好ましいエンコードである。

注記: ツールは少なくともUTF-8をサポートしなくてはならない。

4.1.2 Byte-Order Mark

The encoding that is used for the ASAM MCD-2 MC file is defined in a Byte-Order Mark (BOM). The BOM is a byte sequence at the beginning of the ASAM MCD-2 MC file.

ASAM MCD-2 MC ファイルのために使用されたエンコーディングは、Byte-Order Mark(BOM)によって定義されている。BOM は、ASAM MCD-2 MC file の当初から使用されているバイト(Byte)シーケンスである。

Currently defined BOM sequences:

現在のBOMシーケンス定義:

Table 3 byte-order mark coding

Bytes	Encoding Form
00 00 FE FF	UTF-32, big-endian
FF FE 00 00	UTF-32, little-endian

FE FF	UTF-16, big-endian
FF FE	UTF-16, little-endian
EF BB BF	UTF-8

If no encoding can be detected, ISO-8859-1 (Latin-1) encoding is used.
もしエンコードが決定できない場合、ISO-8859-1(Latin-1)エンコーディングが使用される。

Note: For data type "ident" the restrictions listed in chapter 3.2 "Predefined data types" are valid.

Note: For user defined tags and enum values in AML the restrictions defined in chapter 5.2 "Format of the ASAM MCD-2 MC Meta Language" are valid.

注記: 4.4章“Predefined data types 定義済みデータ型”に記載されているデータタイプ“ident”制限は、有効。

注記: 5.2章“Format of the ASAM MCD-2 MC metalanguage ASAM MCD-2 メタ言語形式”で定義されているAML制限におけるユーザー定義タグとenum Valueは有効。

4.2 Hierarchic division of the keywords

Table 4 Hierarchic division of the keywords

Keyword	Multiple	Meaning
ASAP2_VERSION		ASAM MCD-2 MC version identification
A2ML_VERSION		Version number of ASAM MCD-2 MC Meta Language
PROJECT		Project description
HEADER		Project header description
PROJECT_NO		Project number
VERSION		Project version number
MODULE	X	Description of the ECU
A2ML		ASAM MCD-2 MC Meta-Language (interface-specific description data)
AXIS_PTS	X	Axis points distribution
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
BYTE_ORDER		Byte order of axis points
CALIBRATION_ACCESS		Access for calibration
DEPOSIT		Absolute or difference axis points
DISPLAY_IDENTIFIER		Optional display name
ECU_ADDRESS_EXTENSION		Address extension of the ECU address
EXTENDED_LIMITS		Extended range of values
FORMAT		Display format of axis points
FUNCTION_LIST		Function orientation
GUARD_RAILS		Indicates the use of guardrails
IF_DATA	X	Interface-specific description data
MAX_REFRESH		Maximum refresh rate
MODEL_LINK		Parameter name in the software model
MONOTONY		Monotony with respect to this axis
PHYS_UNIT		Physical unit of the axis points
READ_ONLY		'Read Only' attribute
REF_MEMORY_SEGMENT		Reference to memory segment
STEP_SIZE		Delta value
SYMBOL_LINK		Reference to symbol of linker map file
BLOB	X	Binary blob object
ADDRESS_TYPE		Address type
ANNOTATION	X	Description
ANNOTATION_LABEL		Title of annotation

Keyword	Multiple	Meaning
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
CALIBRATION_ACCESS		Access for calibration
DISPLAY_IDENTIFIER		Optional display name
ECU_ADDRESS_EXTENSION		Address extension of the ECU address
IF_DATA	X	Interface-specific description data
MAX_REFRESH		Maximum refresh rate
MODEL_LINK		Parameter name in the software model
SYMBOL_LINK		Reference to symbol of linker map file
CHARACTERISTIC	X	Adjustable objects
ANNOTATION	X	Description
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
AXIS_DESCR	X	Axis description
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
AXIS_PTS_REF		Reference to axis point distribution
BYTE_ORDER		Byte order of axis points
CURVE_AXIS_REF		Used to normalize or scale an axis
DEPOSIT		Absolute or difference axis points
EXTENDED_LIMITS		Extended limits, e.g. hard limits
FIX_AXIS_PAR		Fixed axis parameters
FIX_AXIS_PAR_DIST		Fixed axis parameters (variant)
FIX_AXIS_PAR_LIST		Fixed axis values
FORMAT		Display format of axis points
MAX_GRAD		Maximum gradient with respect to this axis
MONOTONY		Monotony with respect to this axis
PHYS_UNIT		Physical unit of the axis points
READ_ONLY		'Read Only' attribute
STEP_SIZE		Delta value
BIT_MASK		Bit mask
BYTE_ORDER		Byte order
CALIBRATION_ACCESS		Access for calibration
COMPARISON_QUANTITY		Comparison quantity
DEPENDENT_CHARACTERISTIC		References to characteristics

Keyword	Multiple	Meaning
DISCRETE		Attribute for discrete object values
DISPLAY_IDENTIFIER		Optional display name
ECU_ADDRESS_EXTENSION		Address extension of the ECU address
ENCODING		Encoding for string interpretation
EXTENDED_LIMITS		Extended limits, e.g. hard limits
FORMAT		Display format of values
FUNCTION_LIST		Function orientation
GUARD_RAILS		Indicates the use of guardrails
IF_DATA	X	Interface-specific description data
MAP_LIST		For cuboids: comprising maps
MATRIX_DIM		Dimensions of multi-dimensional arrays
MAX_REFRESH		Maximum refresh rate
MODEL_LINK		Parameter name in the software model
NUMBER		Number of ASCII characters or fixed values
PHYS_UNIT		Physical unit of the characteristic values
READ_ONLY		'Read Only' attribute
REF_MEMORY_SEGMENT		Reference to memory segment
STEP_SIZE		Delta value
SYMBOL_LINK		Reference to symbol of linker map file
VIRTUAL_CHARACTERISTIC		Mark for being virtual
COMPU_METHOD	X	Conversion method
COEFFS		Coefficients for fractional rational function
COEFFS_LINEAR		Coefficients for linear function
COMPU_TAB_REF		Reference to conversion table
FORMULA		Conversion formula
FORMULA_INV		Inverse conversion formula
REF_UNIT		Reference to a measurement unit
STATUS_STRING_REF		Reference to an additional conversion table with status strings
COMPU_TAB	X	Conversion table
DEFAULT_VALUE		Default output string
DEFAULT_VALUE_NUMERIC		Default value
COMPU_VTAB	X	Verbal conversion table
DEFAULT_VALUE		Default output string
COMPU_VTAB_RANGE	X	Description of range based verbal conversion tables
DEFAULT_VALUE		Default output string
FRAME	X	Frame
FRAME_MEASUREMENT		Frame measurement objects
IF_DATA	X	Interface-specific description data

Keyword	Multiple	Meaning
FUNCTION	X	Function description
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
DEF_CHARACTERISTIC		Defined adjustable objects
FUNCTION_VERSION		Version of the function
IF_DATA	X	Interface-specific description data
IN_MEASUREMENT		Input quantity
LOC_MEASUREMENT		Local quantity
OUT_MEASUREMENT		Output quantity
REF_CHARACTERISTIC		Referenced adjustable objects
SUB_FUNCTION		Sub function of respective function
GROUP	X	Declaration of groups
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
FUNCTION_LIST		Function list
IF_DATA	X	Interface-specific description data
REF_CHARACTERISTIC		Reference to characteristic objects
REF_MEASUREMENT		Reference to measurement objects
ROOT		Flag for root node
SUB_GROUP		Sub group
IF_DATA	X	Interface-specific description data
INSTANCE	X	Instance of a measure or adjustable object
ANNOTATION	X	Description
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
CALIBRATION_ACCESS		Access for calibration
DISPLAY_IDENTIFIER		Optional display name
ECU_ADDRESS_EXTENSION		Address extension of the ECU address
IF_DATA	X	Interface-specific description data
LAYOUT		Layout of multi-dimensional arrays
MATRIX_DIM		Dimensions of multi-dimensional arrays
MAX_REFRESH		Maximum refresh rate
MODEL_LINK		Parameter name in the software model
OVERWRITE	X	Overwrite properties of typedef
CONVERSION		Overwrite conversion method

Keyword	Multiple	Meaning
EXTENDED_LIMITS		Overwrite extended physical limits
FORMAT		Overwrite format
INPUT_QUANTITY		Overwrite input quantity
LIMITS		Overwrite physical limits
MONOTONY		Overwrite monotony
PHYS_UNIT		Overwrite physical unit
READ_WRITE		Marks the instance as writable
SYMBOL_LINK		Reference to symbol of linker map file
MEASUREMENT	X	Measurement object
ADDRESS_TYPE		Address type
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation
ANNOTATION_TEXT		Text of annotation
ARRAY_SIZE		Array size of measurement objects
BIT_MASK		Bit mask to decode single-bit values
BIT_OPERATION		Bit operation
LEFT_SHIFT		Number of bit positions to shift left
RIGHT_SHIFT		Number of bit positions to shift right
SIGN_EXTEND		Sign extension for measurement data
BYTE_ORDER		Byte order of measurement object
DISCRETE		Attribute for discrete object values
DISPLAY_IDENTIFIER		Optional display name
ECU_ADDRESS		Address
ECU_ADDRESS_EXTENSION		Address extension of the ECU address
ERROR_MASK		Mask error bits
FORMAT		Display format of measurement object
FUNCTION_LIST		Function orientation
IF_DATA	X	Interface-specific description data
LAYOUT		Layout of multi-dimensional arrays
MATRIX_DIM		Dimensions of multi-dimensional arrays
MAX_REFRESH		Refresh rate in the control unit
MODEL_LINK		Parameter name in the software model
PHYS_UNIT		Physical unit of the measurement values
READ_WRITE		'Writable'
REF_MEMORY_SEGMENT		Reference to memory segment
SYMBOL_LINK		Reference to symbol of linker map file
VIRTUAL		Virtual measurement
MOD_COMMON		Module-wide (ECU specific) valid definitions

Keyword	Multiple	Meaning
ALIGNMENT_BYTE		Alignment border for byte values
ALIGNMENT_FLOAT32_IEEE		Alignment border for float32 values
ALIGNMENT_FLOAT64_IEEE		Alignment border for float64 values
ALIGNMENT_INT64		Alignment border for int64 values
ALIGNMENT_LONG		Alignment border for long values
ALIGNMENT_WORD		Alignment border for word values
BYTE_ORDER		Byte order
DATA_SIZE		Data size in bits
DEPOSIT		Standard deposit mode for axis
MOD_PAR		Control unit management data
ADDR_EPK	X	Address of EPROM identifier
CALIBRATION_METHOD	X	Access method
CALIBRATION_HANDLE	X	Handle for calibration method
CALIBRATION_HANDLE_TEXT		Additional Text for calibration method
CPU_TYPE		CPU
CUSTOMER		Firm or customer
CUSTOMER_NO		Customer number
ECU		Control unit
ECU_CALIBRATION_OFFSET		Address offset
EPK		EPROM identifier
MEMORY_LAYOUT	X	Memory layout
IF_DATA	X	Interface-specific description data
MEMORY_SEGMENT	X	Memory segment
IF_DATA	X	Interface-specific description data
NO_OF_INTERFACES		Number of interfaces
PHONE_NO		Phone number of calibration engineer responsible
SUPPLIER		Manufacturer or supplier
SYSTEM_CONSTANT	X	System-defined constants
USER		User
VERSION		Module-specific version identifier
RECORD_LAYOUT	X	Description of the record layout
ALIGNMENT_BYTE		Alignment border for byte values
ALIGNMENT_FLOAT32_IEEE		Alignment border for float32 values
ALIGNMENT_FLOAT64_IEEE		Alignment border for float64 values
ALIGNMENT_INT64		Alignment border for int64 values
ALIGNMENT_LONG		Alignment border for long values
ALIGNMENT_WORD		Alignment border for word values
AXIS_PTS_X / _Y / _Z / _4 / _5		Axis points

Keyword	Multiple	Meaning
AXIS_RESCALE_X		Rescaling axis points
DIST_OP_X/_Y/_Z/_4/_5		Parameter 'distance' for fixed characteristics
FIX_NO_AXIS_PTS_X/_Y/_Z/_4/_5		Fixed number of axis points
FNC_VALUES		Table values
IDENTIFICATION		Identification
NO_AXIS_PTS_X/_Y/_Z/_4/_5		Number of X axis points
NO_RESCALE_X		Number of rescale pairs for axis
OFFSET_X/_Y/_Z/_4/_5		Parameter 'offset' for fixed characteristics
RESERVED	X	Parameter is skipped (not interpreted)
RIP_ADDR_W		Table value: Address 'result of interpolation'
RIP_ADDR_X/_Y/_Z/_4/_5		Address 'result of interpolation'
SHIFT_OP_X/_Y/_Z/_4/_5		Parameter 'shift' for fixed characteristics
SRC_ADDR_X/_Y/_Z/_4/_5		Address of input quantity
STATIC_ADDRESS_OFFSETS		Static addresses for non-compact data used, FNC Values handled dynamically
STATIC_RECORD_LAYOUT		Static addresses for non-compact data used, FNC Values handled statically
TRANSFORMER	X	Transformation of ECU calibration BLOBs
TRANSFORMER_IN_OBJECTS		Transformer Input Object List
TRANSFORMER_OUT_OBJECTS		Transformer Output Object List
TYPEDEF_AXIS	X	Type definition for axis points
BYTE_ORDER		Byte order of axis points
DEPOSIT		Absolute or difference axis points
EXTENDED_LIMITS		Extended range of values
FORMAT		Display format of axis points
MONOTONY		Monotony with respect to this axis
PHYS_UNIT		Physical unit of the axis points
STEP_SIZE		Delta value
TYPEDEF_BLOB	X	Type definition for blob objects
ADDRESS_TYPE		Address type
TYPEDEF_CHARACTERISTIC	X	Type definition for an adjustable object
AXIS_DESCR	X	Axis description
ANNOTATION	X	Set of notes
ANNOTATION_LABEL		Title of annotation
ANNOTATION_ORIGIN		Creator of annotation

Keyword	Multiple	Meaning
ANNOTATION_TEXT		Text of annotation
AXIS_PTS_REF		Reference to axis point distribution
BYTE_ORDER		Byte order of axis points
CURVE_AXIS_REF		Used to normalize or scale an axis
DEPOSIT		Absolute or difference axis points
ENCODING		Encoding for string interpretation
EXTENDED_LIMITS		Extended limits, e.g. hard limits
FIX_AXIS_PAR		Fixed axis parameters
FIX_AXIS_PAR_DIST		Fixed axis parameters (variant)
FIX_AXIS_PAR_LIST		Fixed axis values
FORMAT		Display format of axis points
MAX_GRAD		Maximum gradient with respect to this axis
MONOTONY		Monotony with respect to this axis
PHYS_UNIT		Physical unit of the axis points
READ_ONLY		'Read Only' attribute
STEP_SIZE		Delta value
BIT_MASK		Bit mask
BYTE_ORDER		Byte order
DISCRETE		Attribute for discrete object values
EXTENDED_LIMITS		Extended limits, e.g. hard limits
FORMAT		Display format of values
MATRIX_DIM		Dimensions of multi-dimensional arrays
NUMBER		Number of ASCII characters or fixed values
PHYS_UNIT		Physical unit of the characteristic values
STEP_SIZE		Delta value
TYPEDEF_MEASUREMENT	X	Type definition for measurement object
ADDRESS_TYPE		Address Type
BIT_MASK		Bit mask to decode single-bit values
BIT_OPERATION		Bit operation
LEFT_SHIFT		Number of bit positions to shift left
RIGHT_SHIFT		Number of bit positions to shift right
SIGN_EXTEND		Sign extension for measurement data
BYTE_ORDER		Byte order of measurement object
DISCRETE		Attribute for discrete object values
ERROR_MASK		Mask error bits
FORMAT		Display format of measurement object
LAYOUT		Layout of multi-dimensional arrays
MATRIX_DIM		Dimensions of multi-dimensional arrays
PHYS_UNIT		Physical unit of the measurement values

Keyword	Multiple	Meaning
TYPDEF_STRUCTURE	X	Structure definition
ADDRESS_TYPE		Address type
CONSISTENT_EXCHANGE		Ensures consistent access to the structure
STRUCTURE_COMPONENT	X	Definition of a structure component
LAYOUT		Layout of multi-dimensional arrays
MATRIX_DIM		Dimensions of multi-dimensional arrays
UNIT	X	Measurement unit
REF_UNIT		Reference to another unit
SI_EXPONENTS		Exponential of base dimensions
UNIT_CONVERSION		Specifies relationship between two units
USER_RIGHTS	X	Groups with constitute access rights
READ_ONLY		Read only
REF_GROUP	X	List of referenced groups
VARIANT_CODING		Variant coding
VAR_CHARACTERISTIC	X	Definition of variant coded adjustable objects
VAR_ADDRESS		Adjustable objects address list (start address of variants)
VAR_CRITERION	X	Definition of variant criterion
VAR_MEASUREMENT		Measurement object which indicates criterion value
VAR_SELECTION_CHARACTERISTIC		Characteristic object which modifies criterion value
VAR_FORBIDDEN_COMB	X	Forbidden combinations of different variants
VAR_NAMING		Naming of variant coded adjustable objects
VAR_SEPERATOR		Separator of adjustable objects names

4.3 Name Spaces ネームスペース

In an ASAP2 file, there are different object types which can be referenced via their object names. To be able to resolve these references it is necessary to have unique identifiers. The unambiguousness of identifiers is not required in the whole ASAP2 file, but within special name spaces. This chapter defines the name spaces for different keywords.

ASAP2 ファイルには、オブジェクト名から参照できる様々な種類のオブジェクトがある。これらのオブジェクトの参照を解決するには、オブジェクト名はユニークな識別子である必要がある。しかしながら、専用のネームスペースを使えば、A2L ファイル全体においてユニークな識別子を必要としない。この章ではネームスペースの定義をキーワード毎に説明している。

4.3.1 Functions 関数

Within one MODULE, each FUNCTION identifier must be unique. It is not allowed to have two functions with the same identifier.

1 つの MODULE 内において、各 FUNCTION の識別子はユニークであること。2 つの関数が同じ識別子を持つことは許容されていない。

4.3.2 Groups グループ

Within one MODULE, each GROUP identifier must be unique. It is not allowed to have two groups with the same identifier.

1 つの MODULE 内において、各 GROUP の識別子はユニークであること。2 つのグループが同じ識別子を持つことは許容されていない。

4.3.3 Variants バリエーション

Within one VARIANT_CRITERION, each variant identifier must be unique. It is not allowed to have two variants with the same identifier.

1 つの VARIANT_CRITERION 内において、各バリエーションの識別子はユニークであること。2 つのバリエーションが同じ識別子を持つことは許容されていない。

4.3.4 Measure and adjustable objects 測定とアジャスタブルオブジェクト

Within one MODULE, each measure and adjustable object identifier must be unique. It is not allowed to have two objects with the same identifier. Relevant keywords for measure and adjustable objects are MEASUREMENT, AXIS_PTS, CHARACTERISTIC, BLOB and INSTANCE.

Furthermore, the full identifiers of structure components belong to this name space. They are built by adding the component name(s) to the instance name separated by dot(s). See keyword INSTANCE. So it is also not allowed to have e.g. a measure object and a structure component with same identifier (see also Keyword INSTANCE).

1 つの MODULE 内において、各測定とアジャスタブルオブジェクトの識別子は、ユニークであること。測定とアジャスタブルオブジェクトとなるキーワードは、MEASUREMENT, AXIS_PTS, CHARACTERISTIC, BLOB, INSTANCE である。

更に、全ての構造体の識別子は、このネームスペースに属している。これらの識別子は、ドットで区切られたコンポーネント名をインスタンス名に追加していく事で構築される。キーワード INSTANCE キーワードを参照のこと。そのため、例えば同じ識別子を指定した測定オブジェクトと構造体は共存できない(キーワードの INSTANCE 内にある Note も参照のこと)。

4.3.5 Transformer トランスフォーマー

Within one MODULE, each TRANSFORMER identifier must be unique. It is not allowed to have two transformers with the same identifier.

1つのMODULE内において、各TRANSFORMERの識別子はユニークであること。2つのトランスフォーマーが同じ識別子を持つことは許容されていない。

4.3.6 Conversion methods 変換規則

Within one MODULE, each COMPU_METHOD identifier must be unique. It is not allowed to have two conversion methods with the same identifier.

1つのMODULE内において、各COMPU_METHODの識別子はユニークであること。2つの変換メソッドが同じ識別子を持つことは許容されていない。

4.3.7 Conversion tables 変換テーブル

Within one MODULE, each conversion table identifier must be unique. Relevant keywords for conversion tables are COMPU_TAB, COMPU_VTAB and COMPU_VTAB_RANGE.

It is not allowed to have two conversion tables with the same identifier.

1つのMODULE内において、各変換テーブルの識別子はユニークであること。変換テーブルとなるキーワードは、COMPU_TAB, COMPU_VTAB, COMPU_VTAB_RANGEである。2つの変換テーブルが同じ識別子を持つことは許容されていない。

4.3.8 Memory segments メモリーセグメント

Within one MODULE, each MEMORY_SEGMENT identifier must be unique. It is not allowed to have two memory segments with the same identifier.

1つのMODULE内において、各MEMORY_SEGMENTの識別子はユニークであること。2つのメモリーセグメントが同じ識別子を持つことは許容されていない。

4.3.9 Record layouts レコードレイアウト

Within one MODULE, each RECORD_LAYOUT identifier must be unique. It is not allowed to have two record layouts with the same identifier.

1つのMODULE内において、各RECORD_LAYOUTの識別子はユニークであること。2つのレコードレイアウトが同じ識別子を持つことは許容されていない。

4.3.10 Type definitions 型定義

Within one MODULE, each type definition identifier must be unique. Relevant keywords for type definitions are TYPEDEF_MEASUREMENT, TYPEDEF_AXIS, TYPEDEF_CHARACTERISTIC, TYPEDEF_STRUCTURE and TYPEDEF_BLOB.

It is not allowed to have two type definitions with the same identifier.

1つのMODULE内において、各型定義の識別子はユニークであること。型定義となるキーワードは、TYPEDEF_MEASUREMENT, TYPEDEF_AXIS, TYPEDEF_CHARACTERISTIC, TYPEDEF_STRUCTURE, TYPEDEF_BLOBである。

2つの型定義が同じ識別子を持つことは許容されていない。

4.3.11 Structure components 構造体

Within one structure definition, each STRUCTURE_COMPONENT identifier must be unique. It is not allowed to have two components with the same identifier.

1 つの構造体定義内において、各 STRUCTURE_COMPONENT の識別子はユニークであること。2 つのコンポーネントが同じ識別子を持つことは許容されていない。

4.4 Predefined data types 定義済みデータ型

Table 5 Predefined data types

Pre-defined data type 定義済みデータ型		Description 説明
ident	typedef char [MAX_IDENT + 1] ident	<p>String with <i>MAX_IDENT</i> (<i>at present = 1024</i>) alphanumeric characters including points and brackets, interpreted as hierarchical concatenation of partial strings separated by points. Every partial string may not exceed <i>MAX_PARTIAL_IDENT</i> (<i>at present = 128</i>) characters, including the length of an optional array index (numeric or as a symbolic string) in brackets at the end of the partial string. One string without a point in between is also possible, in this case <i>MAX_IDENT = MAX_PARTIAL_IDENT</i>. The number of partial strings within <i>ident</i> is not limited. The character chain must correspond with the identifier laws defined in programming language C. Identifiers can represent instances of array elements or instances of elements of complex C types or nested combinations of these. An instance of the element of a struct type would be represented by the concatenation of the instance name, a point and the element name. An instance of an array element would be represented by an instance name followed by a pair of brackets which contain either a numeric value or a symbolic string which is defined as an enumerator of an ENUM definition of the C program. Identifiers are random names which may contain characters A through Z, a through z, underscore (<code>_</code>), numerals 0 through 9, points (<code>'.'</code>) and brackets (<code>'['</code>, <code>']'</code>). However, the following limitations apply: the first character must be a letter or an underscore, brackets must occur in pairs at the end of a partial string and must contain a number or an alphabetic string (description of the index of an array element).</p> <p><i>MAX_IDENT</i> (現在は= 1024)は、点や括弧などを含む英数字の文字列で、点で区切られた、部分文字列の階層的連結として解釈される。</p> <p>すべての部分文字列は、部分文字列の最後の括弧までのオプションの配列インデックスの長さを含んで(数値または記号文字列)、<i>MAX_PARTIAL_IDENT</i> (=現時点では 128)を超えてはならない。</p> <p>また、点を間に含まない、1 つの文字列も可能である。</p> <p>この場合には、の <i>MAX_IDENT = MAX_PARTIAL_IDENT</i> となる。</p>

		<p>ident 内の部分文字列の数に限定は無い。</p> <p>文字の列挙はプログラミング言語 C の識別子の書式に準拠する必要がある。識別子は、配列要素のインスタンス、または複雑な C タイプ要素のインスタンス、または、これらの組合せとなる。構造型の要素のインスタンスは、インスタンス名の連結、点、エレメント名によって表される。配列エレメントののインスタンスは、C プログラムの列挙型である ENUM 定義で定義されている、インスタンス名、一對の括弧、数値または記号文字列となる。識別子は、文字 A から Z、a から z まで、アンダースコア (_)、数字 0 から 9、点 (.) と括弧 ([,]) が含まれていてもよいランダムな名前である。ただし、以下の制限が存在する。</p> <p>最初の文字は文字またはアンダースコア、番号やアルファベットの文字列 (配列要素のインデックスの記述) を含む部分文字列の最後までの一対の括弧でなければならない。</p> <p><u>Note:</u> Identifiers consisting of partial identifiers separated by points (concatenation of instance name and element name) may be presented by the MCD system in a hierarchical manner (show instance name first, then allow access to an element of the instance). This allows existing MCD systems to restrict the display length of the identifier to MAX_PARTIAL_IDENT.</p> <p><u>Note:</u> Identifiers generally must not match to the following defined ASAM MCD-2 MC keywords and enum values. All keyword and enum values are listed in the Index of Keywords and Enum Values.</p> <p><u>Note:</u> A lower case “x” and a upper case “X” can be used as unique identifiers. In other words, variables are case sensitive so that x and X are different identifiers.</p> <p><u>注記:</u> 部分的識別子からなる点で分割された識別子インスタンス名と要素名の連続)は、MCD システム階層形式 (最初にインスタンス名、続いてアクセス可能なインスタンスのエレメント) により提供される。これは、使用している MCD システムが MAX_PARTIAL_IDENT で定義されている表示長とすることになる。</p> <p><u>注記:</u> 識別子は、原則的に次に定義された ASAM MCD-2 MC キーワードと列挙値に一致してはならない。すべてのキーワードと列挙型の値は、キーワードのインデックスと enum 値に記載されている。</p> <p><u>注記:</u> 小文字の “x” と大文字の “X” を別の識別子として使用します。言い換えると、値は大文字と小文字が区別されますので、x および X は異なる識別子となる。</p>
string	typedef char [MAX_STRING + 1] string	ANSI C compliant ‘C type’ string with maximum MAX_STRING (at present = 255) characters. Begin and end of the string are indicated by a double inverted comma.

		<p>The following escape sequences are allowed:</p> <ul style="list-style-type: none"> \` inverted comma \” quotation mark \\ backslash \n new line \r carriage return \t horizontal tab <p>Additionally, for compatibility with ASAP2 V1.2 and prior, the following is allowed:</p> <ul style="list-style-type: none"> “” quotation mark <p>Examples:</p> <pre>"hello \"world\" how are you ?" "hello ""world"" how are you ?"</pre> <p>MCD systems may ignore the carriage return sequence and/or apply wrapping or scrolling of strings when displayed.</p> <p>Please note that new line can only be inserted with the shown escape sequence.</p> <p>最大データ長 MAX_STRING (現時点では255文字)を持つANSI C準拠の‘C type’文字列を示す。文字列の最初と最後はダブルクォーテーションマークで囲むこと。</p> <p>以下のエスケープシーケンスコードの使用は認められる:</p> <ul style="list-style-type: none"> ¥` シングルクォーテーション ¥” ダブルクォーテーション ¥¥ バックスラッシュ ¥n 改行文字 (ニューライン) ¥r 復帰文字 (キャリッジリターン) ¥t 水平タブ <p>加えて、ASAP2 V1.2以前との互換性を目的として、以下の表記も認められる:</p> <ul style="list-style-type: none"> “” ダブルクォーテーションマーク <p>記述例:</p> <pre>"hello ¥"world¥" how are you ?" "hello ""world"" how are you ?"</pre> <p>MCD システムでは文字列の折り返しやスクロールを適用する目的での復帰文字(キャリッジリターン)の適用をほぼ無視する。改行文字(ニューライン)のエスケープシーケンスのみ文字列に挿入できることに注意。</p>
float	8-byte floating point number (IEEE format) 8バイト浮動小数点 (IEEE フォーマット)	<p>The character for a decimal point is fixed as a dot “.”. A comma “,” is not allowed for use as a decimal point. Alternatively the exponential representation can be used, e.g. 12E-2.</p> <p>Numbers without dot are accepted as well.</p> <p>小数点を表す文字としてドット“.”を使用します。(欧州では一般的な)コンマ“,”は小数点を表す文字として認めません。また指数表記(例:12E-2)も使用する事が出来る。小数点文字を省略した表記も認められる。</p>

int	2-byte signed integer 2 バイト符号付整数型	The notation of hexadecimal values is fixed, e.g. 0xE0, 0xFF, etc. 固定化された 16 進数表記を表す。例: 0xE0, 0xFF, 等
uint	2-byte unsigned integer 2 バイト符号無し整数型	
uint64	8- byte unsigned integer 8 バイト符号無し整数型	
long	4-byte signed integer 4 バイト符号付整数型	
ulong	4-byte unsigned integer 4 バイト符号無し整数型	
datatype	typedef enum datatype { UBYTE, SBYTE, UWORD, SWORD, ULONG, SLONG, A_UINT64, A_INT64, FLOAT32_IEEE, FLOAT64_IEEE }	<p>Enumeration for description of the basic data types in the ECU program (format of FLOAT32/64 IEEE).</p> <p><u>Note:</u> If ECU values of type integer 64 are converted in physical values depending on the computation formula a higher precision in the physical area is necessary. The currently used float 64 format supports less precision than int 64. Therefore the precision of the physical representation is reduced to the precision of float 64. This is relevant for ASAM MCD-3 standards [4] where data transfer is defined as physical. This is additionally relevant for all tools working on PCs / Operating systems that do not support higher precision than float 64. Here the representation is rounded for physical and maybe also for internal representation.</p> <p>ECUプログラム内で使用する基本データ型を列挙している。(FLOAT32/64_IEEEフォーマットについてはAppendix: A参照の事).</p> <p><u>注記:</u> 64ビット整数型の ECU データから計算式を用いて物理値に変換する場合、物理値側でより高い精度が必要となるが 64ビット浮動小数点は 64ビット整数型よりもデータの精度は粗いため、変換される物理値は 64ビット浮動小数点の精度まで減少する。 この事象は ASAM MCD-3 仕様における物理データ転送にも関連する。また 64ビット以上の浮動小数点をサポートしていない PC や OS にも関連する。このシステム的な要因により物理値表現が丸められることがある。</p>
datasize	typedef enum datasize { BYTE, WORD, LONG }	<p>Enumeration for description of the word lengths in the ECU program</p> <p>ECUプログラム内で使用するデータ長を列挙している。</p>
addrtype	typedef enum addrtype { PBYTE	<p>Enumeration for description of the addressing: アドレッシング定義を列挙している。</p> <p>The object address is interpreted as a 1 byte pointer to the object data.</p>

	PWORD PLONG PLONGLONG DIRECT }	オブジェクトのアドレスは 1 バイトのポインタとして解釈される。 The object address is interpreted as a 2 byte pointer to the object data. オブジェクトのアドレスは 2 バイトのポインタとして解釈される。 The object address is interpreted as a 4 byte pointer to the object data. オブジェクトのアドレスは 4 バイトのポインタとして解釈される。 The object address is interpreted as an 8 byte pointer to the object data. オブジェクトのアドレスは 8 バイトのポインタとして解釈される。 The object address is interpreted as object data. オブジェクトのアドレスは、オブジェクトデータをそのまま解釈する。
byteorder	<pre>typedef enum byteorder { LITTLE_ENDIAN, BIG_ENDIAN, MSB_LAST, MSB_FIRST, MSB_FIRST_MSW_ LAST, MSB_LAST_MSW_F IRST }</pre>	Enumeration for description of the byte order in the control unit program. Note: Use of LITTLE_ENDIAN and BIG_ENDIAN defined with keyword BYTE_ORDER leads to mistakes because it is in contradiction to general use of terms „little endian“ and „big endian“. The keywords LITTLE_ENDIAN and BIG_ENDIAN should no longer be used, they should be replaced by MSB_LAST and MSB_FIRST which are equivalent (definition of MSB_LAST and MSB_FIRST: see keyword BYTE_ORDER). ECUプログラムで使用するバイトオーダーの列挙。 注記: BYTE_ORDER のキーワードとしてリトルエンディアン (LITTLE_ENDIAN) とビッグエンディアン (BIG_ENDIAN) の使用は望ましくない。理由として“little endian”と“big endian”の一般的な用語の意味と実態に矛盾があり、間違いを起こしやすくなるためである。 代わりに MSB_LAST および MSB_FIRST のキーワードを使用する。 (MSB_LAST と MSB_FIRST の定義については BYTE_ORDER キーワードを参照のこと)
indexorder	<pre>typedef enum indexorder { INDEX_INCR, INDEX_DECR }</pre>	Enumeration for description of the axis point sequence in the memory. Increasing index with increasing address decreasing index with increasing address ECUメモリ内のマップ・カーブ軸ポインタの順序の定義についての列挙。 アドレスポインタの増加によって、軸インデックスが増加。 アドレスポインタを増加によって、軸インデックスが減少。

4.5 Mapping of predefined data Types to ASAM data types 予約データ型と ASAM データ型の関係

The following table shows, how the ASAM data types are mapped to predefined data types. ASAM データ型と予約データ型との関係は、以下を参照のこと。

Table 6 Map ASAM data types to predefined data types

ASAM data type	predefined data type
A_INT16	int
A_UINT16	uint
A_INT32	long
A_UINT32	ulong
A_FLOAT64	float

4.6 Comments コメント

Single line and multi line comments may be added everywhere in an aml and a2l file.

Single line comments start with the character string `"/"` and end at the end of the same line. Multi line comments start with the character string `"/"` and end with the character string `"/"`. Nested multi line comments are not allowed.

単一行、及び複数行のコメントは、aml/a2l ファイルの任意の場所へ挿入可能である。

単一行のコメント文は`"/"`を先頭に、該当行の終端まで続く。複数行のコメント文は`"/"`を先頭に、終端文字`"/"`まで続く。複数行のネストは許容されない。

Example for a single line comment:

```
// This is a single line comment
```

Example for a multi line comment

```
/*
This is a
multi line comment
*/
```

4.7 Alphabetical list of keywords キーワードのアルファベット順リスト

4.7.1 General 概要

Some individual elements of the database are delimited by '/begin' and '/end' keywords. The delimiters are applied to those elements that contain an optional part, to prevent ambiguous expressions. The delimiters following defined with the ASAM MCD-2 MC keywords are mandatory, i.e. the delimiters have to be used if defined and mustn't be used if not defined.

Optional keywords are shown with help of square brackets, which include an arrow followed by the keyword. If the keyword can be used multiple times this is shown with help of asterisk after the closing bracket, e.g. [-> keyword]*.

The description of the parameters of the keyword you find at the keyword description itself. The parameter description consists of the type and the identifier. If the parameters of a keyword can be used multiple times they are defined inside parenthesis followed by asterisk or plus, i.e. (parameter_type parameter_identifier)* or (parameter_type parameter_identifier)+.

Multiple times means for asterisk that an expression can appear 0 till n times, for plus 1 till n times.

データベースのいくつかの要素は、'/begin'と'/end'キーワードに区切られている。曖昧な表現を避けるため、デリミタ(省略可能なオプションを含む)が各要素へ適用される。ASAM MCD-2 MC キーワードに続いて定義されているデリミタは、省略できない。該当のデリミタは定義をする際には使用する必要があり、定義をしない際は使用してはならない。省略可能なキーワードは表記補助として、角括弧[]で囲い、括弧の中にキーワードへ向けた矢印を付与する(例:[-> Keyword])。複数回使用できる類のキーワードであれば、角括弧を閉じた後にアスタリスク*を付与する(例:[-> Keyword]*)。

キーワードのパラメータについては、該当キーワードを説明している章を参照のこと。パラメータの説明は型と識別子から成る。パラメータが複数回使用できる類なら、丸括弧()で囲い、括弧の外へアスタリスク*かプラス記号+を付与する。パラメータを使用する回数が 0~N 回なら、(パラメータ型 パラメータ識別子)* パラメータを使用する回数が 1~N 回なら、(パラメータ型 パラメータ識別子)+とそれぞれ記述している。

以降のドキュメント内容は削除されている。削除された章については、英語版のオリジナルドキュメントを参照の事。

Content of the original standard document has been removed from hereon. Please refer to the original standard document for the missing chapters.

5 Bibliography 参考文献

- [1] ASAM e.V.; CDF - Calibration Data Format V2.0.0; 2006
- [2] ASAM e.V.; MDF - Measurement Data Format V4.1.0; 2012
- [3] ASAM e.V.; FIBEX - Field Bus Data Exchange Format V4.1.1; 2013
- [4] ASAM e.V.; MCD-3 MC API for Measurement and Calibration server V3.0; 2011
- [5] ASAM e.V. ASAP3 Application Systems Interface Specification; 1999
- [6] ASAM e.V.; ODX - Diagnostic Data Model Specification V2.2; 2008
- [7] ASAM e.V.; Seed&Key and Checksum Calculation API V1.0
- [8] ISO/DIS 9141; Road vehicles -- Diagnostic systems –
Part 1: Requirements for interchange of digital information, 1989
Part 2: CARB requirements for interchange of digital information; 1994
Part 3: Verification of the communication between vehicle and OBD II scan tool; 1998
- [9] ISO/DIS 14229; Road vehicles -- Unified diagnostic services (UDS) – Part 1: Specification and requirements; 2006
- [10] ISO/DIS 14230; Road vehicles -- Diagnostic systems -- Keyword Protocol 2000 – Part 1: Physical layer; 1999
Part 2: Data link layer; 1999
Part 3: Application layer; 1999
Part 4: Requirements for emission-related systems; 2000
- [11] ISO/DIS 15765; Road vehicles -- Diagnostics on Controller Area Networks (CAN) – Part 1: General information; 2004
Part 2: Network layer services; 2004
Part 3: Implementation of unified diagnostic services (UDS on CAN); 2004
Part 4: Requirements for emissions-related systems; 2005
- [12] ISO: ISO/DIS 22901-1; Road vehicles — Open diagnostic data exchange — Part 1: Data model specification; 2008
- [13] ISO: ISO/DIS 22900-2; Road vehicles — Modular vehicle communication interface (MVCI) —Part 2: Diagnostic protocol data unit application programmer interface (D-PDU API); 2006
- [14] ISO: ISO/DIS 22900-3; Road vehicles - Modular vehicle communication interface (MVCI) - Part 3: Diagnostic server application programming interface (D-Server API); 2007
- [15] ASAM e.V.; MDX Meta Data Exchange Format for Software Module Sharing Version 1.2.0; 2013
- [16] ASAM e.V.; Container Catalog XML Model Specification Version 3.0.0; 2004



Association for Standardisation of
Automation and Measuring Systems

E-mail: support@asam.net

Internet: www.asam.net

© by ASAM e.V., 2015