ASAM MCD-2 MC

ASAM MCD-2 MC (aka ASAP2) defines a description format for internal ECU variables used for measurement and calibration purposes. 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.
One of the major strengths of ASAM MCD-2 MC is its support for automotive-specific processes and working methods. The standard has extensive support for lookup tables up to 5 dimensions, which includes various ways to store or calculate axis points. Virtual calibration parameters, which are calculated instead of measured, are supported as well. Calibration and measurement variables can be hierarchically grouped via various means to support function-oriented calibration.
The standard allows to fully specify how data is displayed in an MC-system, independent from ECU-internal data formats. This is achieved via computation methods, format definitions and the definition of units. This has the advantage that calibration engineers can work with data formats that they understand and has a meaning for them. The standard furthermore allows to describe the memory segments of the ECU, which includes a description of their location inside the address space, the type of memory and the way it can be accessed from the outside.
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.
The data description is written in a structured ASCII format (*. a2l), which can be easily parsed and imported. The BOM mechanism (byte order mark) allows to also use the UTF-8 character set. Include statements allow to collect descriptions from different sources. The standard is technology- and vendor-independent. Due to its completeness, versatility and maturity, the standard is widely used and supported by virtually all major MC-systems on the market today.

Datasheet

Title	Data Model for ECU Measurement and Calibration
Category	AE
Current Version	1.6.1
Release Date	17.02.2010
Download	ASAM MCD-2 MC V1.6.1
Application Areas	• Calibration of ECU parameters • Measurement of ECU variables • ECU programming
Specification Content	• Programmers guide • Data model • Interface description
File Formats	a2l, aml

Content

History

The first version of the standard was developed and released before the foundation of ASAM e.V. in 1994. The initial name of the standard was "ASAP-2", which was later changed to ASAM MCD-2 MC. The standard belongs to a group of coordinated standards, which are part of a 3-layer base architecture for MCD-systems (Measurement, Calibration, Diagnostics).

ASAM MCD-2 MC includes the description of internal ECU data (characteristics and measurements) and the ECU interface (how to access characteristics and measurements) in one file. This was a deliberate decision to keep all information to describe and access ECU data in one location. Furthermore, ASAM MCD-2 MC describes a very compact ASCII format. When XML became popular in the industry at the early 2000th, the standardization group deliberately did not migrate ASAM MCD-2 MC to an XML-compatible format. This would have otherwise increased the file size by approximately five times, which would have caused severe performance issues with tools that process such files. Furthermore, a lot of tools in the Automotive industry relied already on the non-XML format and would have to be re-written from scratch, - an effort that was deemed to be unjustified.

ASAM MCD-2 MC v1.6 introduced UTF encoding for the A2L-file to support non-European languages such as Japanese or Chinese, 4- and 5- dimensional data objects to extend curves, maps and cuboids, 64-bit integer data types, layouts for measurement objects and more keywords. The latest version 1.6.1 from 2010 includes minor improvements.

Main contributors to the standard are Continental Automotive GmbH, dSPACE GmbH, ETAS GmbH, M&K GmbH, Robert Bosch GmbH, Softing AG and Vector Informatik GmbH.

Motivation

The calibration of parameters is an essential part of ECU software development. Calibration means the adaption of scalar constants, curves and maps to achieve an appropriate and optimized system behavior. Once a new set of parameters has been determined, the next development step is to run tests in order to evaluate the effectiveness of the calibration. For this purpose, internal variables are read from memory and transferred to a system that displays the data in a human-readable format.

In the early days of ECU development, the values of calibration parameters were directly modified in the source code. Variables had to be made available for data logging in the source code as well. Every change to parameters or the list of measureable variables required modifications in the source code, re-compilation and flashing of the ECU. As the control software grew in complexity and the development of the software was split up into several groups of engineers (function developers, software developers, calibration engineers, vehicle test engineers, etc.), this process became too cumbersome and slow. The process of measurement & calibration needed to be separated from the process of software development. In other words, if a calibration engineer needs to change a parameter value or wants to record the values from a measurement variable, he does not need to ask the software developer do compile a new software version for him. He can do this with the existing ECU software version by just reconfiguring his MC-system. This is the fundamental motivation for the group of ASAM MCD standards.

ASAM MCD-2 MC provides an ECU description that is optimized for the calibration engineer. Relevant information such as detailed descriptions of calibration and measurement variables is included. Information that is not needed for calibration (such as code details) is excluded. Furthermore, it contains a description of the device interface to the ECU for read and write access. Such a description of calibration and measurement variables can easily extend to several thousand entries per ECU. Such amounts of data is very typical for engine controllers. Today, software development is highly distributed. Different companies work with MC-systems from different tool vendors. Without standardization, the creation and maintenance of such description files could easily become a major time and cost factor of the overall development process, requiring to maintain several data description files in parallel, or continuously converting them between different formats and making sure that everyone in the development process gets the right format. The variety of tools and description formats would quickly become a hindrance for development progress and a frequent source of errors in the MCD tool chain.

ASAM MCD-2 MC was created to overcome those problems and avoid the waste of time and cost of having to deal with various description formats that more or less contain the same data. The standard defines the syntax and semantics of the data descriptions. It was developed to consider the needs of all involved groups in the calibration process. For example, the description is typically produced by function developers, software engineers, tools & instrumentation experts, and is used by calibration engineers from various disciplines such as mechanical engineering, electrical engineering or controls engineering. They all find data elements and properties within the description format that they need for their work. Furthermore, they can work with the ECU data in a familiar representation without having to understand ECU-internal data formats such as scaled integers, bit-fields or ID-codes.

Application Areas

The primary application area for ASAM MCD-2 MC is the area of measurement & calibration. Virtually all market-leading MC-systems for the Automotive industry know this format and are able to import, process and export a2l-files. The standard is also used in adjacent industries such as in train- and shipbuilding.

Closely related to that is the area of rapid control prototyping systems, where the standard is used for the same purpose. Test automation systems use ASAM MCD-2 MC for automated calibration and data logging. The standard is used furthermore by in-vehicle data loggers and diagnostic tools. Most of the production code generators for embedded software automatically generate a2l-files.

Technical Content

File

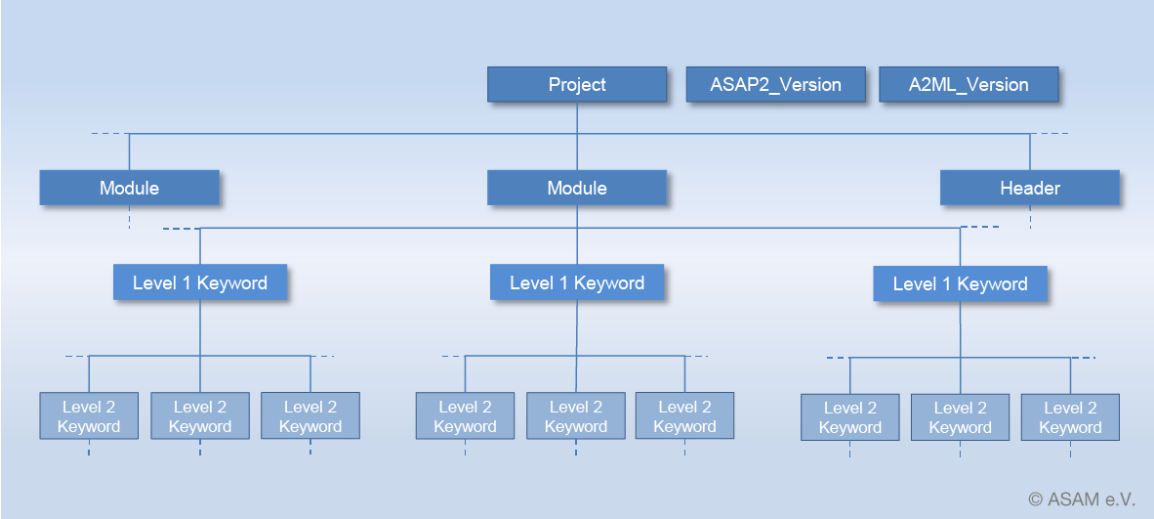
The file extension of ASAM MCD-2 MC compliant description files is "a2l", which is an abbreviation of "ASAM MCD-2 MC Language".

If no encoding is specified in the a2l-file, then ISO-8859-1 (Latin-1) shall be assumed. Otherwise, a2l-files shall use the Unicode Transformation Format at least on the level of UTF-8. Higher levels (UTF-16 and UTF-32) can be used as well. UTF allows the use of non-Latin characters like Chinese or Japanese, which is useful in descriptive texts inside the a2l-file such as descriptions (LongIdentifier), annotations (ANNOTATION) and comments (/* … */).

File Format

The internal format of a2l-files is based upon a non-XML notation. This is because the first version of ASAM MCD-2 MC was created several years before XML became an official standard. The schema described in ASAM MCD-2 MC is simple, efficient and easy to parse. There was never a real need to transform ASAM MCD-2 MC to an XML-compliant schema and the standard continues today to remain a non-XML format.

The content of a2l-files consists of keywords, parameters, delimiters and comments. They form a data model, which describe data semantics and data values. The keywords are the main elements of the ASAM MCD-2 MC data model. Keywords can contain parameters and other keywords. The parameters of a keyword contain the values of the data model. Other keywords underneath a keyword create a hierarchical structure of keywords similar to an aggregation in XML. Parameters and aggregated keywords may be mandatory or optional and may have a multiplicity.



Structure of an a2l-file

Some keywords are delimited with "/begin" and "/end". The delimiters are applied to those keywords that contain optional keywords or list of parameters with variable length. This shall prevent ambiguous interpretation.

The standard clearly defines the list of parameters and aggregated keywords via prototype definitions. The prototype also specifies whether parameters and aggregated keywords are mandatory or optional, their multiplicity and use of delimiters.

Comments in a2l-files follow the same syntactical rules as for the C++ language. Single line comments start with a double forward-slash (i.e. //). Multi-line comments are delimited with a forward-slash and asterisk (i.e. /*) at the beginning and with an asterisk and forward-slash (i.e. */) at the end and shall not be nested.

An a2l-file can have include statements (/include <filename>), which allows to include a2l-file fragments into one a2l-master-file. This is common practice in distributed development processes, where software originates from different suppliers and different tool chains. They all provide partial data descriptions via a2l-file fragments, that have to be merged via include statements into a single file. Furthermore, it is common practice to place the interface description (i.e. the *A2ML* secti on) into its own file with the extension ".aml".

Structure

At the beginning of an a2L-file, the version number (keyword: *ASAP2_VERSION*) of the standard and optionally the version number of the AML language (*A2ML_VERSION*) is stated. The a2l-file consists of four structural levels: Project, Module, primary keywords and secondary keywords. Secondary keywords may actually span across several more levels.

An a2l-file contains one project (*PROJECT*), which describes all calibration and measurement data that belong to one calibration project. A header provides some general information about the project such as project number, version and a description (*HEADER*). A project consists of one or more modules (*MODULE*). Each module represents one ECU. The standard allows to contain several ECU descriptions via the *MODULE* keyword, but current MC-systems typically only support one *MODULE*per a2l-file.

Top-Level Keyword	Description
A2ML_VERSION	Version of the ASAM MCD-2 MC meta language
ASAP2_VERSION	Version of the ASAM MCD-2 MC standard used i
HEADER	Allows to specify a project number and an ECU s which the a2l-file is compatible with.
MODULE	Contains the data description for one ECU.
PROJECT	Keyword on root level of the a2l-file. Contains eve

The third level, which is below the keyword *MODULE*, holds the primary keywords, which contain the actual description of ECU data. The following list contains the keywords of this level.

Primary Keyword	Description
A2ML	Defines the formal description of parameters that communication between the MC- system and the only describes the syntax of communication para (i.e. semantics) depends on the used communica drivers, which is not part of ASAM MCD-2 MC. Th contain the configuration of the protocol stack and messages for measurement & calibration objects *ENT* and *CHARACTERISTIC*. The actual parame the *IF_DATA* blocks. The values in the *IF_DATA* syntax description of the *A2ML* block. The descri the ASAM MCD-2 MC meta language AML.
AXIS_PTS	Axes contain the sample point values for curves a keyword describes the properties of an axis, such memory, references to the input variable (*MEAS* layout and computation method, the maximum nu and further properties.

CHARACTERISTIC	*CHARACTERISTIC* describes the properties of a This parameter can be a scalar, string, array or lo associated axes. The following types of tunable p available: • VALUE: scalar • ASCII: string • VAL_BLK: array (no axes) • CURVE: 1D-table • MAP: 2D-table • CUBOID: 3D-table • CUBE_4: 4D-table • CUBE_5: 5D-table The address, record layout, computation method, calibration limits and further properties are define
COMPU_METHOD	Describes the methods and properties for conver ECU-internal format, which is optimized for imple physical format, which is easily understood by hu Computation methods are typically referenced by and MEASUREMENTs. The majority of automoti uses scaled integers for this data. COMPU_METI from their fixed-point representation into a floatin for display in an MC-system. This representation for signals, or may displays discrete data such as strings. Supported conversion methods are: • IDENTICAL: no conversion • LINEAR: linear, 2-coefficient function with slo • RAT_FUNC: 6-coefficient rational function w numerator and denominator polynomials • TAB_INTP: table with interpolation • TAB_NOINTP: table without interpolation • TAB_VERB: verbal table (i.e. enumeration) • FORM: formula which consists of a specific s functions Please note that the conversion direction is from the physical format, except for RAT_FUNC, whicl conversion from the physical format to the interna Other properties describe the display format (in C the unit. If needed, the *COMPU_METHOD* specif coefficients, references to tables, formulas and u
COMPU_TAB	Conversion tables are used by *COMPU_METHO* cannot be expressed by a formula. They are desc value pairs, i.e. same as a 1D lookup table. Conv without interpolation are supported.
COMPU_VTAB	Verbal conversion tables are used to convert inte human-readable strings. A number or bit-pattern Verbal conversion tables are described by pairs c output-strings. This method is equivalent to enum programming language.
COMPU_VTAB_RANGE	Same as *COMPU_VTAB*, but allows to specify a string.

FRAME	Allows to group MEASUREMENTs to selection lis chosen in an MC-system for selective recording a ECU-internal variables. FRAMEs are typically use which shall be measured and viewed together. Th can also be used to describe the packaging of EC a CAN frame.
FUNCTION	Allows to group MEASUREMENTs, CHARACTEI AXIS_PTSs into selection lists, which can be cho for selective tuning of parameters and recording c FUNCTIONs are typically used to bundle variable which belong to one software function. This supp measurement and calibration. Owned and extern variables can be expressed. Function hierarchies sub-functions can be expressed.
GROUP	Allows to group *MEASUREMENT*s, *CHARACTEI* AXIS_PTSs into selection lists, which can be cho for selective tuning of parameters and recording c are typically used to bundle variables and parame common meaning or are used for a specific view. in conjunction with *USER_RIGHTS* to control use supports function-oriented measurement and cali hierarchies can be expressed that include a root
IF_DATA	List of parametric values that are used to configu communication between the MC-system and ECU keyword on *MODULE* level, the *IF_DATA* section parametric values for the configuration of the prot ections may also be used as secondary keywords parameterize the communication of objects such nd *CHARACTERISTIC*. The list of values needs t syntax definitions in the *A2ML* section of the a2l-f
MEASUREMENT	*MEASUREMENT* describes the properties of a re variable. This variable can be a scalar or an array operations can be applied to the measurement. T order, computation method, upper and lower limit properties are described. Standard allows also to objects, e.g. to stimulate the ECU during runtime. also describe a virtual variable, which is calculate variables, other virtual variables and constants.
MOD_COMMON	Defines default parameters that are common for c module, so they do not have to be repeated for e include the definition of byte alignment, byte orde data in the ECU memory. The parameters of *MO* optional parameters of other keywords. If they are keyword, then the corresponding parameter value used. Otherwise, when a parameter is defined in overrules the parameter value defined in *MOD_C*
MOD_PAR	Specifies general parameters of a module (i.e. EC such as the name of the CPU, customer, version ECU-specific data. Furthermore, this keyword con of the organization of the ECU's memory via the *GMENT* as well as a list of system constants whi conversion methods.
RECORD_LAYOUT	Describes how structures of tunable parameters and axes (*AXIS_PTS*) are stored in memory. It de alignments, order and position of calibration objec rescaling, memory offset and further properties.

UNIT	Defines units that can be referenced by *MEASUR RISTIC*, *AXIS_PTS* and *COMPU_METHOD*. Unit according to the International System of Units (SI based units described by exponents of the seven derived units described by a reference unit and a method.
USER_RIGHTS	Lists *GROUP*s and access rights for named users be read-only or read & write.
VARIANT_CODING	Description of tunable parameters, which have m stored in ECU memory at different addresses. Th variant-coded parameters and non-variant coded differ in the *CHARACTERISTIC* keyword. If a par coded, then *VARIANT_CODING* has a reference specifies additional properties that describe how t variant (i.e. read the selected value from memory might be selected by another tunable parameter (*HARACTERISTIC*) or an ECU-internal variable (.

The fourth and lower levels of an a2l-file consist of secondary keywords. They are aggregated by primary keywords. The secondary keywords are a way to further structure the data and to provide further details. The following list contains the keywords of this level.

Secondary Keyword	Aggregated by	Description
ADDR_EPK	*MOD_PAR*	Address of EPROM ider
ALIGNMENT_BYTE	*MOD_COMMON, RECORD_LAYOUT*	Alignment of byte data i
ALIGNMENT_FLOAT32_ IEEE	*MOD_COMMON, RECORD_LAYOUT*	Alignment of float32 dat
ALIGNMENT_FLOAT64_ IEEE	*MOD_COMMON, RECORD_LAYOUT*	Alignment of float64 dat
ALIGNMENT_INT64	*MOD_COMMON, RECORD_LAYOUT*	Alignment of int64 data
ALIGNMENT_LONG	*MOD_COMMON, RECORD_LAYOUT*	Alignment of long data i
ALIGNMENT_WORD	*MOD_COMMON, RECORD_LAYOUT*	Alignment of word data
ANNOTATION	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC, FUNCTION, GROUP, MEASUREMENT*	Container for annotation
ANNOTATION_LABEL	*ANNOTATION*	Title of annotation.
ANNOTATION_ORIGIN	*ANNOTATION*	Creator of annotation.
ANNOTATION_TEXT	*ANNOTATION*	Explanatory text in an a
ARRAY_SIZE	*MEASUREMENT*	Number of values in the array is supported, only. Please use *MATRIX_DI*

AXIS_DESCR	*CHARACTERISTIC*	Specifies the properties to a tunable curve, map following axis types are • STD_AXIS: Axis sp • FIX_AXIS: Axis sp calculated axis poir stored in ECU mem • COM_AXIS: Axis s tables. • CURVE_AXIS: Axi tables and rescaled curve (*CURVE_AX* • RES_AXIS: Axis sl and rescaled, i.e. n axis (*AXIS_PTS_R*
AXIS_PTS_REF	*AXIS_DESCR*	Reference to *AXIS_PTS* values are stored in a di memory location than th *ACTERISTIC* the axis d
AXIS_PTS_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position, datat and addressing method axis points in memory.
AXIS_RESCALE_X	*RECORD_LAYOUT*	Specifies the rescale ma axis points and used po maps.
BIT_MASK	*CHARACTERISTIC, MEASUREMENT*	Specifies a bit mask to e the object's value.
BIT_OPERATION	*MEASUREMENT*	Specifies an additional b which consists of a bit s extension.
BYTE_ORDER	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC, MEASUREMENT, MOD_COMMON*	Endianness or position bit.
CALIBRATION_ACCESS	*AXIS_PTS, CHARACTERISTIC*	Type of access to the tu full access, offline or no
CALIBRATION_HANDLE	*CALIBRATION_METHOD*	Parameters for *CALIBR*
CALIBRATION_ HANDLE_TEXT	*CALIBRATION_HANDLE*	Text for *CALIBRATION_*
CALIBRATION_ METHOD	*MOD_PAR*	Specifies the memory a the MC-system.
COEFFS	*COMPU_METHOD*	Coefficients for the ratio *C*).
COEFFS_LINEAR	*COMPU_METHOD*	Coefficients for the linea
COMPARISON_ QUANTITY	*CHARACTERISTIC*	Reference to a *MEASU* represents the working
COMPU_TAB_REF	*COMPU_METHOD*	Reference to a conversi
CPU_TYPE	*MOD_PAR*	String that identifies the

CURVE_AXIS_REF	*AXIS_DESCR*	Reference to the curve's hat is used to normalize
CUSTOMER	*MOD_PAR*	String that identifies the
CUSTOMER_NO	*MOD_PAR*	String that provides a nu customer.
DATA_SIZE	*MOD_COMMON*	Number of bits in measu objects. Typically repres the used micro-controlle
DEFAULT_VALUE	*COMPU_TAB, COMPU_VTAB, COMPU_VTAB_RANGE*	Default output string, wh when the measured EC of the defined table.
DEFAULT_VALUE_ NUMERIC	*COMPU_TAB*	Default output float valu display when the measu of range of the defined t
DEF_CHARACTERISTIC	*FUNCTION*	References to *AXIS_PT* *C* that belong to the func
DEPENDENT_ CHARACTERISTIC	*CHARACTERISTIC*	The value of the *CHAR* references this *DEPEN* *TIC*, is calculated instea memory. *DEPENDENT_* pecifies a formula and re parameters (in memory purpose to calculate the changes automatically, referenced parameters l
DEPOSIT	*AXIS_DESCR, AXIS_PTS, MOD_COMMON*	Storage mode for axis p signifies that absolute va stored. "DIFFERENCE" difference-values betwe stored.
DISCRETE	*CHARACTERISTIC, MEASUREMENT*	Indicates that the param be interpolated, e.g. in c post-processing.
DISPLAY_IDENTIFIER	*AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Can be used to specify a is different (e.g. shorter) name.
DIST_OP_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d (i.e. slope) value within t distance value is used to points for the described
ECU	*MOD_PAR*	String that describes the
ECU_ADDRESS	*MEASUREMENT*	Address of the MEASUR memory.
ECU_ADDRESS_ EXTENSION	*AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Specifies additional add instance to distinguish b address spaces of an EC
ECU_CALIBRATION_ OFFSET	*MOD_PAR*	Offset that has to be add absolute address of a *C* Used to resolve near-po or to select the data se variant data sets.

EPK	*MOD_PAR*	String that describes the
ERROR_MASK	*MEASUREMENT*	Bit mask that can be use bits of a *MEASUREMEI* error.
EXTENDED_LIMITS	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC*	Specifies an extended u beyond the normal uppe values. Can be used to out-of-range warnings a messages, or to allow s set calibration values be
FIX_AXIS_PAR	*AXIS_DESCR*	Specifies the value of th power-of-two exponent and total number of sam computing the sample p an equidistant axis of ty
FIX_AXIS_PAR_DIST	*AXIS_DESCR*	Specifies the value of th increment value and the sample points for compu values of an equidistant
FIX_AXIS_PAR_LIST	*AXIS_DESCR*	Explicitly specifies the s the axis of type *FIX_AX*
FIX_NO_AXIS_PTS_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies the number of number is fixed and not
FNC_VALUES	*RECORD_LAYOUT*	Specifies position, datat addressing method to st record layout.
FORMAT	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Specifies the display for C-printf notation. Overru parameter in referenced
FORMULA	*COMPU_METHOD*	Specifies a conversion f physical value from the Expression of the formu C notation. Shall be use rational functions are no
FORMULA_INV	*FORMULA*	Specifies a conversion f ECU-internal value from the inversion of the refe Expression of the formu notation.
FRAME_ MEASUREMENT	*FRAME*	List of *MEASUREMENT* this frame.
FUNCTION_LIST	*AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Lists the *FUNCTION*s in listed. Obsolete keyword *ON* instead.
FUNCTION_VERSION	*FUNCTION*	String that describes the function.
GUARD_RAILS	*AXIS_PTS, CHARACTERISTIC*	Determines that the oute curves and maps are ca adjusted.
IDENTIFICATION	*RECORD_LAYOUT*	Specifies position and d identification number for

IF_DATA	*AXIS_PTS, CHARACTERISTIC, FRAME, FUNCTION, GROUP, MEASUREMENT, MEMORY_LAYOUT, MEMORY_SEGMENT*	*IF_DATA* sections are u device driver for commu MC-system and the ECU keyword, it includes the values that are needed f for this object that the de transfer. The list of value the definitions in the A2l a2l-file. *IF_DATA* sectio as primary keywords (se configure the protocol st
IN_MEASUREMENT	*FUNCTION*	References to *MEASUR* defined as inputs to this
LAYOUT	*MEASUREMENT*	Specifies how multidime arrays are stored in linea "ROW_DIR" signifies ro "COLUMN_DIR" signifie
LEFT_SHIFT	*BIT_OPERATION*	Number of bit positions *IT_OPERATION*.
LOC_MEASUREMENT	*FUNCTION*	References to *MEASUR* defined as local variable
MAP_LIST	*CHARACTERISTIC*	Lists the maps which co
MATRIX_DIM	*CHARACTERISTIC, MEASUREMENT*	Specifies the dimension arrays.
MAX_GRAD	*AXIS_DESCR*	Specifies the maximum for this axis.
MAX_REFRESH	*CHARACTERISTIC, MEASUREMENT*	Maximum refresh rate fo control unit.
MEMORY_LAYOUT	*MOD_PAR*	Description of the memo Obsolete keyword. Plea *GMENT* instead.
MEMORY_SEGMENT	*MOD_PAR*	Description of one mem ECU. Includes program location, address, size a
MONOTONY	*AXIS_DESCR, AXIS_PTS*	Specifies which kind of r sample values is allowe
NO_AXIS_PTS_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d of axis points within the
NO_OF_INTERFACES	*MOD_PAR*	Number of interfaces.
NO_RESCALE_X	*RECORD_LAYOUT*	Specifies position and d of rescaling values withi
NUMBER	*CHARACTERISTIC*	Specifies the number of characters or values) in keyword. Please use *MA*
OFFSET_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d value within the record l is used to calculate the described *FIX_AXIS*.

OUT_MEASUREMENT	*FUNCTION*	References to *MEASUR* defined as the outputs o
PHONE_NO	*MOD_PAR*	Phone number of the re engineer.
PHYS_UNIT	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Specifies the physical u specified in the referenc
PROJECT_NO	*HEADER*	String that describes a p
READ_ONLY	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC, USER_RIGHTS*	Specifies read-only acce parameter or for this use
READ_WRITE	*MEASUREMENT*	Specifies that write-acce *EASUREMENT*.
REF_CHARACTERISTIC	*FUNCTION, GROUP*	References to *AXIS_PT C* that are used but not *ON* or which belong to t
REF_GROUP	*USER_RIGHTS*	Reference to groups of *CHARACTERISTIC*s to access rights for this gro
REF_MEASUREMENT	*GROUP*	Reference to *MEASURE* this group.
REF_MEMORY_ SEGMENT	*AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Reference to a memory address is not unique, e memory segments exis
REF_UNIT	*COMPU_METHOD, UNIT*	Reference to a physical
RESERVED	*RECORD_LAYOUT*	Specifies a position in th shall be ignored (i.e. not
RIGHT_SHIFT	*BIT_OPERATION*	Number of bit positions *BIT_OPERATION*.
RIP_ADDR_W / _X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d result of interpolation fo axis and the look-up tab
ROOT	*GROUP*	Specifies the root of this
SHIFT_OP_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d power-of-two exponent slope) value within the r distance value is used t points for the described
SIGN_EXTEND	*BIT_OPERATION*	Specifies that a right shi extend the leftmost bit t new, shifted number has the two's complement sy
SI_EXPONENTS	*UNIT*	Specifies the exponents units to express this der
SRC_ADDR_X / _Y / _Z / _4 / _5	*RECORD_LAYOUT*	Specifies position and d of the axis' input value w
STATIC_RECORD_ LAYOUT	*RECORD_LAYOUT*	Specifies that a tunable number of axis points d expand in memory when axis points.

STATUS_STRING_REF	*COMPU_METHOD*	Reference to a verbal co *U_VTAB* or *COMPU_V* split up the value range a numerical part and a v contains status informat part such as providing a the quality of the measu
STEP_SIZE	*AXIS_DESCR, AXIS_PTS, CHARACTERISTIC*	Specifies an increment v subtracted when using t calibrating.
SUB_FUNCTION	*FUNCTION*	Reference to other *FUN* sub-functions of this *FU* to reproduce the hierarc ECU software.
SUB_GROUP	*GROUP*	Reference to other *GRO* sub-groups of this *GRO* create a hierarchy of gro
SUPPLIER	*MOD_PAR*	String that describes the supplier of the ECU.
SYMBOL_LINK	*AXIS_PTS, CHARACTERISTIC, MEASUREMENT*	Reference to the symbo within a linker map file. (automatic update of mea a2L-file.
SYSTEM_CONSTANT	*MOD_PAR*	Specifies name and valu can be used in *FORMU*
UNIT_CONVERSION	*UNIT*	Specifies slope and offs convert the referenced u *UNIT*.
USER	*MOD_PAR*	String that describes the
VAR_ADDRESS	*VAR_CHARACTERISTIC*	List of ECU addresses f variant coded tunable pa *ACTERISTIC*). The num this list must match the from the referenced vari *ERION*).
VAR_CHARACTERISTIC	*VARIANT_CODING*	Description of a tunable than one value in ECU r description consists of a tunable parameter (*CHA* references to variant crit *N*) defining the possible reference to the list of E values for each variant (
VAR_CRITERION	*VARIANT_CODING*	Description of a variant description consists of a and a selector variable (*REMENT* or *CHARACT* currently active variant b
VAR_FORBIDDEN_ COMB	*VARIANT_CODING*	Combination of variants

VAR_MEASUREMENT	*VAR_CRITERION*	Reference to an ECU-in selects the active varian corresponding *MEASUI* a *COMPU_TAB*, whose correspond with the vari *AR_CRITERION*.
VAR_NAMING	*VARIANT_CODING*	Indexing method to disti variants, e.g. "NUMERI(
VAR_SELECTION_ CHARACTERISTIC	*VAR_CRITERION*	Reference to a tunable selects the active varian corresponding *CHARA(* to a *COMPU_TAB*, who correspond with the vari *AR_CRITERION*.
VAR_SEPERATOR	*VARIANT_CODING*	Separation symbol betw variant coded paramete extension.
VERSION	*HEADER, MOD_PAR*	String that describes the
VIRTUAL	*MEASUREMENT*	Specifies that this *MEA* measured but calculate measurements via a *CO*
VIRTUAL_ CHARACTERISTIC	*CHARACTERISTIC*	Specifies a formula to ca value of this virtual char referenced *CHARACTE* the virtual characteristic memory. It is typically us *NDENT_CHARACTERI*

Datatypes

ASAM MCD-2 MC supports datatypes that are typically used in ECU software. The standard does not explicitly state the signedess, bit-width and format of those data types. The following table provides a typical interpretation of the datatypes as used in the automotive industry.

Datatype	Signedness	Bit Width	Format
UBYTE	unsigned	8	one's comp
SBYTE	signed	8	two's comp
UWORD	unsigned	16	one's comp
SWORD	signed	16	two's comp
ULONG	unsigned	32	one's comp
SLONG	signed	32	two's comp
A_UINT64	unsigned	64	one's comp
A_INT64	signed	64	two's comp
FLOAT32_IEEE	signed	32	IEEE 754 s
FLOAT64_IEEE	signed	64	IEEE 754 d

Relation to other Standards

The standard belongs to a group of tightly coupled standards that specify interfaces of meas-urement, calibration and diagnostic systems (MCD). The ASAM MCD-1 standards specify cali-bration protocols between the ECU and an external MC-system. ASAM MCD-2 standards specify data models and description file formats for describing internal ECU data and network data. The ASAM MCD-3 standards specify APIs for remote controlling of MCD-systems, e.g. for automated calibration. The diagnostic part of ASAM MCD-3 has also been published as ISO 22900-3. ASAM MCD-2-MC is furthermore associated to ASAM CDF, which is a file format that stores the values of calibration parameters and associated meta data, and ASAM MDF, which is a file format that stores the values of measured variables and associated meta data. When all standards are jointly applied, then the MCD tool-chain achieves a high degree of interoperability, vendor- and technology-independence and allows easy exchange of data between customers and suppliers.

The data model of ASAM MCD-2 MC has been the foundation for other, later standards of the automotive industry. For example, ASAM MDX took the majority of ASAM MCD-2 MC data elements over into its own data model. The same applies to AUTOSAR's software component template, which is almost identical to ASAM MDX with respect to MCD data descriptions.

Benefits & Advantages

The main advantage of ASAM MCD-2 MC is that the standard allows to separate the process of measurement & calibration from the process of software development. Calibration engineers can work independent from software engineers as soon as they get a flashable software version and a matching a2l-file.

The advantages are even more significant when the development process is spread over several companies. Software sources do not have to be shared any longer to allow other parties to tune parameters or change the list of measurable. A software supplier may just provide a flashable executable and the corresponding a2L-file, which is all that is needed to enable his customer to carry out calibration & measurement tasks.

Since a2l-files are standardized and vendor-independent, they do not have to be converted even though every partner in a development project may use different tools and different interfaces. The standard allows to connect software development tools, calibration tools and ECU calibration interfaces with a neutral description format. All tools that support the description format are able to exchange and process the included information, hence there are no vendor-specific or technology-specific dependencies between tools of an ASAM-compliant calibration tool-chain.

Industry Adoption

Due to this comprehensive and complete coverage of data related to measurement and calibration, the standard has been globally accepted in the automotive industry and displaced most of the proprietary formats that were formerly used in the Automotive industry.

Today, ASAM MCD-2 MC is widely used in the automotive industry as the data format to describe measurement variables and calibration parameters. Virtually all market-leading MC-systems know this format and are able to import and export a2l-files. The standard is used furthermore in other tools in the MCD area, such as data loggers, diagnostic tools, rapid control prototyping systems and automated calibration and testing systems. Most of the production code generators for embedded software automatically generate a2l-files along with C-code sources. Special tools are available for most of the Linker map file formats that can update an a2l-file with address and record-layout information.

List of Deliverables

The standard includes the following deliverables:

Standard document

ASAM offers a checker tool for a2l- and aml-files, which is available as a seperate deliverable. The checker verifies that files are syntactically correct, name spaces have unique names, references are resolved and that mandatory parameters are specified and correctly typed, dependencies between parameters and some further plausibility and consistency checks.

Downloads

The data description in this a2l-file matches the downloadable cdfx-file (see ASAM CDF) and mf4-file (see ASAM MDF). You can study with those three files how they work together.

ASAP2_Demo_V161.a2l	Example of an a2l-file, including AML-section for XCP.