VECTOR

Rules for Legacy Communication Descriptions

Technical Reference

CAN, LIN and FlexRay

Version 1.8

Authors	Hannes Haas, Matthias Gette
Status	Released

Document Information

History

Author	Date	Version	Remarks
Hannes Haas	2010-09-24	1.0	creation
Hannes Haas	2011-01-11	1.1	Added LIN Improvements on CAN
Hannes Haas	2012-01-30	1.2	Added additional CAN baud rate attributes
Hannes Haas	2013-01-09	1.3	Extended for MICROSAR 4
Matthias Gette	2013-05-10	1.4	Corrections
Matthias Gette	2013-10-23	1.5	Added CANNM info Added Fibex description
Matthias Gette	2014-11-10	1.6	Added CAN-FD info
Matthias Gette	2015-03-19	1.7	Added CAN-FD info for Diag Added NM-OSEK info
Matthias Gette	2017-04-04	1.8	Added SecOC attributes
Matthias Gette	2018-03-14	1.8.1	Corrected Signal Attributes

Reference Documents

No.	Source	Title	Version
[1]	Vector	TechnicalReference_Asr_CanNm.pdf	
[2]	Vector	TechnicalReference_Asr_Com.pdf	
[3]	Vector	TechnicalReference_Asr_CanTp.pdf	
[4]	LIN	LIN Standard	2.1
[5]	Vector	Application Note AN-AND-1-106: Basic CAN Bit Timing Available from the vector.com download center	2.0
[6]	Vector	TechnicalReference_GenTool_CsAsrLegacyDb2SystemDescr_Vector.pdf	1.5

Contents

1 Overview 5
1.1 Supported Legacy Files 5

2 Introduction to the CAN DBC Format 6
2.1 CAN Data Base Editor 6
2.2 Data Base Attributes 6

3 DBC Attributes and Conventions 8
3.1 General Attributes 8
3.2 Attributes for COM Module 9
3.3 Attributes for SecOC Module 10
3.4 Attributes for AUTOSAR Network Management 11
3.5 Attributes for OSEK Network Management (direct) 13
3.6 Attributes for Diagnostic (DCM) 13
3.7 Definition of XCP and application messages 14
3.8 Definition of Update Bits 14
3.9 Definition of Invalid Values 15

4 LDF Rules and Conventions 16
4.1 Definition of LIN Channel Name 16
4.2 Definition of Update Bits 16
4.3 Definition of invalid values 16
4.3.1 Example 17

5 Fibex Rules and Conventions 18

6 General Rules 19
6.1 Definition of AUTOSAR System Signals 19

7 Contact 20

Tables

Table 3-1 General Attributes 9
Table 3-2 List of COM Data Base Attributes 10
Table 3-3 List of SecOC Data Base Attributes 11
Table 3-3 List of NM Data Base Attributes 12
Table 3-4 List of NM Data Base Attributes 13
Table 3-5 List of DCM Data Base Attributes 14

1 Overview

This document describes the legacy data base rules for the Vector AUTOSAR stack based on MICROSAR 3 and MICROSAR 4. Legacy file formats provide an alternative way to configure the MICROSAR communication stack in case no AUTOSAR System Description is available.

The Vector AUTOSAR tooling will convert the legacy files to an AUTOSAR System Description which will be used for further processing in the tool chain.

1.1 Supported Legacy Files

CAN: DBC with attributes as defined in chapter 3

> Authoring Tools: Vector CANdb++ or DaVinci Network Designer

LIN: LDF according to LIN 2.1. Additional rules are defined in chapter 4

> Authoring Tools: Vector LIN File Editor or DaVinci Network Designer

FlexRay: ASAM Fibex according to version 3.1

> Authoring Tool: Vector Fibex Explorer Pro

2 Introduction to the CAN DBC Format

The CAN data base (DBC for short) describes the CAN communication between all ECUs in one network. For the description of multiple CAN networks in a vehicle a separate DBC file for each CAN bus is necessary.

Additionally the DBC format supports the possibility to describe any additional information via attributes, e.g. baud rate of the CAN bus.

The described CAN communication and the attributes within a DBC file are used to configure the embedded communication stack via the configuration tool GENy or DaVinci Configurator Pro. Other Vector tools also use the DBC file as input format, e.g. the network simulation and analyzing tool CANoe.

This document focuses on the necessary attributes and their values for correct configuration and code generation.

2.1 CAN Data Base Editor

Info
When exporting a DBC from CANdb++ Admin or DaVinci Network Designer the option “Export GENy compatible message attributes” has to be selected.

2.2 Data Base Attributes

Data base attributes contain additional configuration-relevant information, e.g. the baud rate of a CAN network. As such information has always a specific context (e.g. baud rate is network specific) at creation time of an attribute the context (object type) has to be defined. The following list contains the relevant object types.

- **Network**:
  Attribute is network-specific and has to be configured only once in the DBC (as a DBC contains only a single network)
- **Node**:
  Attribute is node-specific and has to be set for each node in the network
- **Message**:
  Attribute is message-specific and has to be set for each message in the network
- **Signal**:
  Attribute is signal-specific and has to be set for each signal in the network
- **Node – Mapped Rx Signal**:
  Attribute describes an rx signal property for a specific node and has to be set for each Node-RxSignal relation in the network

Info
The default value of a database attribute is automatically set to all relevant objects at creation time of the attribute.

There are four relevant different value types for data base attributes that are described in the following list:

- **Enumeration (Enum)**:
  Attribute can adopt a limited list of non-numeric values, e.g. No / Yes.

Caution
The names and the sorting order must be considered as defined in this document for each attribute (e.g. No = 0, Yes = 1).

- **Hex**:
  Numeric attribute in a hexadecimal format
- **Integer**:
  Numeric attribute in an integer format
- **String**:
  Attribute can be any string

3 DBC Attributes and Conventions

This chapter describes the definition and usage of the attributes used for the Vector AUTOSAR stack. Please also refer to the attribute documentation in the component-specific technical references.

The described attribute rules are backward compatible with the existing CANbedded solution.

Info
This document just gives an overview of DBC attributes for a data base. For more detailed description on the attributes, please also refer to the attribute documentation in the MICROSAR BSW specific technical references.

3.1 General Attributes

Attribute Name	Object Type	Type	Values and Ranges (**Bold =** default)	Description
Baud rate	Network	Integer	0..**500000**..1000000	CAN Baud rate of the network in bits/s.
SamplePointMin	Network	Integer	50..**75**..100	For details please refer to [6]: “Sample Point”. This parameter is optional.
SamplePointMax	Network	Integer	50..**90**..100	For details please refer to [6]: “Sample Point”. This parameter is optional.
SyncJumpWidthMin	Network	Integer	1..4	For details please refer to [6]: “Resynchronization Jump Width”. This parameter is optional.
SyncJumpWidthMax	Network	Integer	1..4	For details please refer to [6]: “Resynchronization Jump Width”. This parameter is optional.
NBTMin	Network	Integer	6..**8**..25	For details please refer to [6]: “Number of Time Quanta”. This parameter is optional.
NBTMax	Network	Integer	8..**21**..25	For details please refer to [6]: “Number of Time Quanta”. This parameter is optional.
Manufacturer	Network	String	**Vector**	Indicates the OEM. Value must be “Vector”.
DBName	Network	String	**CAN**	Specifies the name of the network. Must be different for any CAN, LIN and FlexRay network within one ECU.
BusType	Network	String	**CAN** CAN FD	Definition of CAN-2.0 and CAN-FD networks. Must be set to “CAN FD”, if there is at least one CAN-FD message.
VFrameFormat	Message	Enum	**CAN Standard**	Indicates kind of CAN message. Hint:

Attribute Name	Object Type	Type	Values and Ranges (**Bold =** default)	Description
			CAN Extended CAN FD Standard CAN FD Extended	This attribute is available for each message without declaration in attribute definitions. Its display text is “ID-Format” or “Type”.

Table 3-1 General Attributes

3.2 Attributes for COM Module

The following table describes the relevant attributes for the AUTOSAR COM module.

Attribute Name	Object Type	Type	Values and Ranges (**Bold =** default)	Description
GenMsgILSupport	Message	Enum	No: 0 **Yes**: 1	Indicates that a message shall be handled by COM. If “Yes” is chosen the message will be handled by COM otherwise not.
GenMsgSendType	Message	Enum	Cyclic: 0, NotUsed, NotUsed, NotUsed, NotUsed, NotUsed, NotUsed, NotUsed, **NoMsgSendType: 8**	Specifies the Tx behavior of the I-PDU. Can be combined with any kind of GenSigSendType.
GenSigSendType	Signal	Enum	Cyclic: 0, OnWrite: 1, OnWriteWithRepetition: 2, OnChange: 3, OnChangeWithRepetition: 4, NotUsed, NotUsed, **NoSigSendType: 7**	Specifies the Tx behavior of a Signal. OnChange is only supported for signals <= 4 Byte. Please note: The combination of transmission types with repetition and without repetition will result in the message being transmitted with repetition at any time.
GenMsgCycleTime	Message	Integer	**0**..65535	Time in ms between each cyclic transmission of a message.
GenMsgCycleTimeFast	Message	Integer	**0**..65535	Time in ms between each cyclic transmission of a message if at least one IfActiveSignal has a different value as its default value. Also relevant for messages with repetitions (i.e.

Attribute Name	Object Type	Type	Values and Ranges (**Bold =** default)	Description
				GenMsgNrOfRepetition > 0): Time between each repetition.
GenSigStartValue	Signal	**Integer** Float	**0**..2147483647	This Value is the default value for the signal. The string value type can represent hexadecimal and integer values.
GenSigInactiveValue	Signal	Integer	**0**..2147483647	Indicates the invalid value of a signal.
GenMsgDelayTime	Message	Integer	**0**..65535	This is the minimum time in ms between the transmissions of different messages with the same identifier.
GenMsgStartDelayTime	Message	Integer	**0**..65535	This defines the time in ms between Com_IpduGroupStart and the first transmission of the cyclic part of this I-PDU.
GenMsgNrOfRepetition	Message	Integer	**0**..255	Number of transmit repetitions after one initial transmission request. The time between the repetitions has to be defined using the dbc attribute GenMsgCycleTimeFast.
GenSigTimeoutTime_<Ecu>	Signal	Integer	**0**..65535	Timeout time in ms used for this signal received by a specific node. If different GenSigTimeoutTime values are configured for a message and update bits are not used, the lowest timeout time (strongest definition) is used for timeout monitoring. A dedicated attribute definition (GenSigTimeoutTime_<Ecu>) has to be provided for every ECU receiving this signal.

Table 3-2 List of COM Data Base Attributes

3.3 Attributes for SecOC Module

The following table describes the relevant attributes for the AUTOSAR COM module.

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
SC_Message	Message	Enum	**No** = 0, Yes = 1	If set to “true” this message is an authenticated message and all the SC* attributes will be considered. It relates to an ISecuredPdu in AUTOSAR System Description.
SC_Verification	Node – Mapped Rx Signal	Enum	No = 0, **Yes** = 1	If set to “true” in any signal of one authenticated message (SC_Verification=”true”) then the complete message is verified upon receiving. If all signals are set to “false” then the message is not verified. It relates to the RxSecurityVerification parameter of an IPduPort in AUTOSAR System Description.
SCP_AuthInfoTxLength	Message	Integer	0..**28**..512	Truncated MAC size of the authenticated message. This size together with SCP_FreshnessValueTxLength must fit to byte boundaries.
SCP_DataId	Message	Integer	**0**..65535	Data Id of the authenticated message
SCP_FreshnessValueId	Message	Integer	**0**..65535	Freshness Value Id of the authenticated message
SCP_FreshnessValueLength	Message	Integer	0..**48**..512	Complete length of the freshness value
SCP_FreshnessValueTxLength	Message	Integer	0..**4**..512	Truncated freshness value size of the authenticated message

Table 3-3 List of SecOC Data Base Attributes

3.4 Attributes for AUTOSAR Network Management

The following table provides a list of data base attributes that shall be used if the CANNM is used on a network.

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
NmType	Network	String	NmAsr	This attribute defines the type of the NM. Must be set to “NmAsr” for AUTOSAR NM networks.
NmAsrNode	Node	Enum	No = 0, **Yes** = 1	This attribute defines if the corresponding node uses the AUTOSAR NM or not
NmAsrTimeoutTime	Network	Integer	1..**1000**..65535	This attribute defines the NM Network Timeout Time
NmAsrWaitBusSleepTime	Network	Integer	1..**750**..65535	This attribute defines the Wait Bus Sleep Time
NmAsrRepeatMessageTime	Network	Integer	1..**400**..65535	This attribute defines the Repeat

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
				MessageTime
NmAsrMessage	Message	Enum	**No** = 0, Yes = 1	This attribute defines whether the corresponding message is an AUTOSAR NM message
NmAsrMessageCount	Network	Integer	1..**64**..256	This attribute defines the maximum number of AUTOSAR NM messages received by the NM (message range) Value must be 2 to the power of n and n has to be natural number
NmAsrBaseAddress	Network	Hex	0..**0x400**.. 0x1FFFFFFF	Base address of the NM messages; Identifies together with NmAsrMessageCount the NM message range: <NmAsrBaseAddress>.. <NmAsrBaseAddress + NmAsrMessageCount - 1> Value must be an integer multiple of NmAsrMessageCount
NmAsrCanMsgCycleTime	Network	Integer	1..**100**..65535	NM message cycle time
NmAsrCanMsgReducedTime	Node	Integer	1..**50**..65535	Message time for Bus Load Reduction. This attribute is node specific and has to be greater or equal to ½ * NmAsrCanMsgCycleTime but less than NmAsrCanMsgCycleTime
NmAsrCanMsgCycleOffset	Node	Integer	**0**..65535	NM message transmission offset. Has to be set less than NmAsrCanMsgCycleTime.
NmAsrNodeIdentifier	Node	Hex	**0**..255	Default address of Arbitrary Address Capable nodes. Node ID as used in the NID of the NM message (if used)
-	<SignalName>_CBV	String	**-**	In a DBC where each NM message contains an 8 bit signal with suffix “_CBV” mapped onto byte position N (N=0 or N=1), N is interpreted as the network wide position of the Control Bit Vector
-	<SignalName>_SNI	String	**-**	In a DBC where each NM message contains an 8 bit signal with suffix “_SNI” mapped onto byte position N (N=0 or N=1), N is interpreted as the network wide position of the Node Identifier

Table 3-4 List of NM Data Base Attributes

3.5 Attributes for OSEK Network Management (direct)

The following table provides a list of data base attributes that shall be used if OSEK-NM is used on a network.

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
NmType	Network	String	Vector	This attribute defines the type of the NM. Must be set to “Vector” for OSEK-NM networks without OEM specific features.
NmNode	Node	Enum	**No** = 0, Yes = 1	This attribute defines if the corresponding node takes part in OSEK-NM communication or not
NmMessage	Message	Enum	**No** = 0, Yes = 1	This attribute defines whether the corresponding message is an OSEK-NM message
NmMessageCount	Network	Integer	16..**128**..256	This attribute defines the maximum number of OSEK-NM messages received by the ECU (message range) Value must be 2 to the power of n and n has to be natural number
NmBaseAddress	Network	Hex	0..**0x400**.. 0x1FFFFFFF	Base address of the NM messages; The NM message range is computed from this attribute and NmMessageCount: <NmBaseAddress>.. <NmBaseAddress + NmMessageCount - 1> Value must be an integer multiple of NmMessageCount
NmStationAddress	Node	Hex	**0**..0xFF	The ECU’s own address used by OSEK-NM.

Table 3-5 List of NM Data Base Attributes

3.6 Attributes for Diagnostic (DCM)

The following data base attributes are utilized to configure the DCM and CANTP modules.

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
DiagState	Message	Enum	**No** = 0, Yes = 1	Set to yes for: > Functional (UDS) request CanTp Normal addressing will be used.
DiagRequest	Message	Enum	**No** = 0, Yes = 1	Set to yes for: > Physical Request CanTp Normal addressing will be used.
DiagResponse	Message	Enum	**No** = 0, Yes = 1	Set to yes for: > Physical Response CanTp Normal addressing will be used.
DiagConnection	Message	Integer	**0**..0xFFFF	Groups DiagRequest and DiagResponse to one Connection by a similar value of this attribute.

Attribute Name	Object Type	Type	Values and Ranges (**Bold** = default)	Description
				The value shall be derived as follows: <TesterAddress><ServerAddr>.
DiagFdOnly	Message	Enum	**No** = 0, Yes = 1	Specifies, to which kind of diag requests the ECU will respond. > No (or not available) ECU will respond to diag requests in both formats CAN 2.0 and CAN-FD > Yes ECU will respond to diag requests in CAN-FD format only Note: both messages request and response must be given the same attribute value This attribute has no effect on CAN-2.0 frames.

Table 3-6 List of DCM Data Base Attributes

3.7 Definition of XCP and application messages

If a message shall be handled by a non AUTOSAR module – such as XCP or another CDD – the message shall not be assigned to any of the following layers.

The following attributes must be set as defined:

> GenMsgILSupport = No

> NmMessage = No (or attribute not available)

> NmAsrMessage = No (or attribute not available)

> DiagState = No (or not available)

> DiagRequest = No

> DiagResponse = No

Furthermore one signal shall be defined in the message that spans over the complete message. The signal has to be used for the Rx mapping of the message in the DBC file.

If the message contains “XCP” or “CCP” in its name (case insensitive) it will be treated as an XCP message.

3.8 Definition of Update Bits

In order to use update bits in the DBC file a naming convention has been defined. The update bit is therefore defined as an independent signal with some specific characteristics. Update bits can be used for signals and signal groups.

> GenMsgILSupport = Yes

> The update bit of the signal (or signal group) with the name <X> is configured as a further signal with the name <X>_UB

> The update bit <X>_UB signal must be in the same message as the signal (or signal group) <X>

> The update bit “signal” shall have a bit size of 1bit

> GenSigSendType shall be NoSigSendType for the update bit

In the system description this update bit signal will be modelled as an AUTOSAR update bit (i.e. the signal definition will be removed and an update bit will be added).

3.9 Definition of Invalid Values

If the AUTOSAR mechanism of an invalid value shall be used for a signal, the DBC value table has to be extended by a value description called “SNA”. The value defined for this description will be used as invalid value by all ECUs sending or receiving this signal.

If a signal has no SNA description in the value table, the invalid value feature will not be used for this signal.

4 LDF Rules and Conventions

The recommended LIN LDF version is LIN 2.1. In the LDF file, the following elements have to be set as follows (see [4]):

LIN_protocol_version = "2.1";

LIN_language_version = "2.1";

This chapter defines additional requirements to this standard in order to be able to configure some AUTOSAR related features that are not available in the LDF.

4.1 Definition of LIN Channel Name

The LIN Channel Name shall be defined in all LDF files by using the LDF description element ‘Channel_name’ (see [4]). The channel name must be unique over all CAN (DBName attribute) and LIN channels.

4.2 Definition of Update Bits

The LIN Master Node (AUTOSAR ECU) shall be able to support update bits (e.g. for CAN to LIN signal routing). In order to use update bits in the LDF file a naming convention has been defined. The update bit is defined as an independent signal with some specific characteristics.

> The update bit of the signal with the name <X> is configured as a further signal with the name <X>_UB

> The update bit <X>_UB signal must be in the same message as the signal <X>

> The update bit “signal” shall have a bit size of 1bit.

> The default value shall be 0 (required for the LIN Slaves that handle update bits as ordinary signals)

In the system description, this update bit signal will be modelled as an AUTOSAR update bit (i.e. the signal definition will be removed and an update bit will be added).

LIN Slaves will read the LDF directly and will handle the Update Bits as an ordinary signal which has to be evaluated manually before reading the actual signal.

4.3 Definition of invalid values

In order to allow the invalidation of LIN Signals by the LIN Master node the LDF element ‘Signal_encoding_types’ (see [4]) shall be used. The value that shall be used as invalid value shall have the name ‘SNA’. This is similar to the solution on CAN where value tables are used.

The LIN slaves will handle this “SNA” value as a normal signal value that must be considered as invalid. This will be handled by the LIN Slave application manually.

4.3.1 Example

```
Signal_encoding_types {
   FrontLeftSwitchStatus_Sig_Type {
      physical_value, 0, 2, 1, 0;
      logical_value, 2, "down";
      logical_value, 3, "SNA";
   }
}

Signal_representation {
   FrontLeftSwitchStatus_Sig_Type : FrontLeftSwitchStatus;
}
```

5 Fibex Rules and Conventions

The recommended Fibex version is Fibex 3.1.0. There are no special attributes or other extensions necessary to create a Flexray communication based on a Fibex file. For limitations refer to see [6].

6 General Rules

These rules apply for LDF and DBC files in the same way.

6.1 Definition of AUTOSAR System Signals

AUTOSAR specifies a so called “Signal Fan-out” which is implemented by the RTE.

Using this mechanism allows the application to write to a single SWC data element which is then transmitted using several COM Signals. I.e. the signal is transmitted on e.g. within several messages on several networks.

In order to configure this mechanism using DBC and LDF files a naming convention is used:

If a node transmits several signals (n) with the same signal name, the same signal size and (if available) the same value table, n COM signals will be created but only a single system signal. If the RTE accesses the system signal a fan-out resp. fan-in will be generated that writes the signal value to all n COM signals.

The mechanism is supported for LDF and DBC (GenMsgILSupport = Yes) signals.

7 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com