GMLAN 3.1

Technical Reference

Calibration with GENy

Version 2.01.01

Authors	Gunnar Meiss, Markus Schwarz, Jason Wolbers, Heiko Hübler, Frank Triem, Marco Pfalzgraf
Versions:	2.01.01
Status:	Released

1 Document Information

1.1 History

Author	Date	Version	Remarks
Gunnar Meiss	2007-04-13	1.0	Creation
Gunnar Meiss	2008-01-16	2.0	Added GENy Support
Markus Schwarz	2009-03-26	2.00.01	Corrected generation rules for nmVNMFStartSendCalCnt
Jason Wolbers	2012-03-27	2.00.02	Added descriptions for IlVnRxMessageEnabled, IlVnTxMessageEnabled Fixed Init Message description
Heiko Hübler, Marco Pfalzgraf, Frank Triem	2012-10-27	2.01.00	Added description for Rx Timeout Time Chapter 3 added Added descriptions for ‘Sleep Transition Time’, ‘Supervision Stability Time’, ‘Max No Sleep Confirmation’
Frank Triem	2013-01-28	2.01.01	ESCAN00064578: Update GMLAN version from GMLAN 3.0 to GMLAN 3.1

Table 1-1 History of the Document

1.2 Reference Documents

Index	Document
[1]	Vector’s Interaction Layer User Manual
[2]	Vector’s Interaction Layer Technical Reference for GENy
[3]	Vector’s Interaction Layer Technical Reference for GM
[4]	Vector’s Network Management Technical Reference

Table 1-2 References Documents

Please note
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector’s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Tables

2 Introduction

This document describes the calibration (post build configuration) parameters of the GMLAN Handler that is configured with GENy. It does not describe the process how the calibration of the GMLAN Handler is carried out.

	Please note This document is valid for GMLAN 3.1 ▶ Il_Vector_Gm version 1.01.00 and higher ▶ Nm_Gmlan_Gm version 4.03.00 and higher Changes to previous module version can be found in chapter 3.

3 Module History

This chapter describes the calibration implementation of the Vector Interaction Layer and Network Management for General Motors in GENy.

3.1 Il_Vector_Gm Version 1.01.00

3.1.1 What is new?

- The Rx Timeout Time for each message is calibrated (chapter 5.2.1).

3.1.2 What has changed?

- There are no changes in this version.

3.2 Nm_Gmlan_Gm Version 4.03.00

3.2.1 What is new?

- New calibrateable values for 'Sleep Transition Time', 'Supervision Stability Time' and 'Max No Sleep Confirmation' (chapters 6.5, 6.6 and 6.7).

3.2.2 What has changed?

- There are no changes in this version.

4 GMLAN Handler Calibration

The calibration of the GMLAN Handler is done by modification of the post build time configuration parameters (constant variables and tables) that can be found in `gmlcal.c`.

The following chapter describes in detail the configuration parameters (constant variables and tables) of the generated file `gmlcal.c`.

	Info If the calculated Table Values have a remainder, the value has to be rounded depending on your needs.

5 Il_Vector_Gm : Interaction Layer

This chapter describes the calibration capabilities of the Interaction Layer.

5.1 Transmit Messages

The GMLAN Handler provides the ability to enable and disable the transmit process for each transmitted functional message with the function `IlSetTxMessageEnable(..)`. The default for this calibration is ‘enabled’.

This can also be calibrated directly using the table `IlVnTxMessageEnabled[]`. Note that using the function `IlSetTxMessageEnable` has the same effect as modifying `IlVnTxMessageEnabled[]`, so it is not necessary to both calibrate the table and call the function. The table is a bit-packed field where each bit represents one IL transmit message. If the bit is a 1, the message is calibrated on and can be sent whenever a relevant VN is active; if the bit is a 0, the message is calibrated off and will never be sent by the IL at runtime regardless of VN activity. The IL messages are ordered least significant bit to most significant bit in each byte. The first byte contains IL messages 7-0, the second byte 15-8, and so on. For example, consider a single CAN channel configuration with 10 IL messages:

`{0xFF, 0x03}` – All 10 IL messages are calibrated on

`{0xFE, 0x03}` – IL message 0 is calibrated off

`{0x7F, 0x03}` – IL message 7 is calibrated off

`{0xFF, 0x01}` – IL message 9 is calibrated off

When a configuration contains more than one CAN channel, a new byte in `IlVnTxMessageEnabled` is started to represent the IL messages for the next channel. The remaining bits in the byte for the previous channel are skipped and have no meaning. For example, consider a configuration with 10 IL messages on channel 0 and 12 IL messages on channel 1:

`{0xFF, 0x03, 0xFF, 0x0F}` – All 22 IL messages are calibrated on

`{0xFF, 0x03, 0xFE, 0x0F}` – IL message 10 (the first IL message on channel 1) is calibrated off

`{0xFF, 0x03, 0x7F, 0x0F}` – IL message 17 (the eighth IL message on channel 1) is calibrated off

The numbering of the IL messages can be found at the top of the generated file `il_par.h` in the format `#define IlTxMsgHnd<message name> <number>`. Non-IL messages (VNMF, HLVW, diagnostics, etc.) cannot be calibrated using this method.

5.1.1 Message Delay Time

The start delay time for each transmit message (Interaction Layer Tx Handle) is configured in the table `IlTxStartCycles[]`.

Symbol	Description
`GenMsgStartDelayTime`	The start delay time of a message in ms. This is also the corresponding database attribute.
`kIlTxCycleTime`	The call cycle time of the `IlTxTask` of the dependent channel in ms.
`TableValue`	The value in the table for the corresponding IL Tx handle.

The Formula for the Value Calculation is:

$$\left\lfloor \frac{GenMsgStartDelayTime}{kIlTxCycleTime} \right\rfloor + 1 = TableValue$$

If the following condition matches, the value 0 has to be used for the IL Tx handle.

$$\left\lfloor \frac{GenMsgStartDelayTime}{kIlTxCycleTime} \right\rfloor = 0$$

5.1.2 Minimum Update Time

The minimum update time for each transmit message (Interaction Layer Tx Handle) is configured in the table `IlTxUpdateCycles[]`.

Symbol	Description
`GenMsgDelayTime`	The delay time of a message in ms. This is also the corresponding database attribute.
`kIlTxCycleTime`	The call cycle time of the `IlTxTask` of the dependent channel in ms.
`TableValue`	The value in the table for the dependent IL Tx handle.

The Formula for the Value Calculation is:

$$\left\lfloor \frac{GenMsgDelayTime}{kIlTxCycleTime} \right\rfloor + 1 = TableValue$$

If the following condition matches, the value 0 has to be used for the IL Tx handle.

$$\left\lfloor \frac{GenMsgDelayTime}{kIlTxCycleTime} \right\rfloor = 0$$

5.1.3 Periodic Rate

The periodic rate for each transmit message (Interaction Layer Tx Handle) is configured in the table `IlTxCyclicCycles[]`.

Symbol	Description
`GenMsgCycleTime`	The cycle time of a cyclic message in ms. This is also the corresponding database attribute.
`kIlTxCycleTime`	The call cycle time of the `IlTxTask` of the dependent channel in ms.
`TableValue`	The value in the table for the dependent IL Tx handle.

The Formula for the Value Calculation is:

$$\left\lceil \frac{GenMsgCycleTime}{kIlTxCycleTime} \right\rceil = TableValue$$

5.1.4 Fast Periodic Rate

The fast periodic rate for each transmit message (Interaction Layer Tx Handle) is configured in the table `IlTxEventCycles[]`.

Symbol	Description
`GenMsgCycleTimeFast`	The fast cycle time of a message in ms. This is also the corresponding database attribute.
`kIlTxCycleTime`	The call cycle time of the `IlTxTask` of the dependent channel in ms.
`TableValue`	The value in the table for the dependent IL Tx handle.

The Formula for the Value Calculation is:

$$\left\lceil \frac{GenMsgCycleTimeFast}{kIlTxCycleTime} \right\rceil = TableValue$$

5.1.5 Init Message

The Init Messages are enabled/disabled in the table `IlVnTxSendOnInit[]`. The table contains one bit for each transmit message (Interaction Layer Tx Handle):

- 0: The transmit message is not an Init Message
- 1: The transmit message is an Init Message
- The layout of the table follows the same pattern as `IlVnTxMessageEnabled[]` for transmit messages. Please see section 4.1 for a description and examples.

> **Please note**
> The implementation of the table has changed between CANGen and GENy.

Any message that is transmitted via the Interaction Layer (usually all messages with extended IDs) can be configured as Init Messages. These messages are transmitted according to the configured transmission type ‘cyclic’, ‘on event’ or ‘cyclic and on event’. The Init Messages are additional transmitted upon:

- Reception or transmission of an I-VNMF that initializes at least one VN that is associated to the VN
- Start of a Shared Local VN, which the message is associated to
- All Initial Messages that are associated to any Initially Active VN are transmitted upon reception of a HLVW.

The transmission of the Init Message is delayed according to the calibrated message delay time. Refer to 5.1.1.

5.2 Receive Messages

The GMLAN Handler provides the ability to enable and disable the receive process for each received functional message with the function `IlSetRxMessageEnable(..)`.
The default for this calibration is ‘enabled’.

This can also be calibrated directly using the table `IlVnRxMessageEnabled[]`. Note that using the function `IlSetRxMessageEnable` has the same effect as modifying `IlVnRxMessageEnabled[]`, so it is not necessary to both calibrate the table and call the function. The layout of the table follows the same pattern as `IlVnTxMessageEnabled[]` for transmit messages. Please see section 4.1 for a description and examples.

5.2.1 Rx Timeout Time

The Rx Timeout Time for each message is configured in the table `IlRxTimeoutTbl[]`.

Symbol	Description
`GenSigTimeoutTime_<ECU>`	Timeout time of a message in ms. This is also the corresponding database attribute.
`kIlRxCycleTime`	The call cycle time of the `IlRxTask` of the dependent channel in ms.
`TableValue`	The value in the table for the dependent IL Rx handle.

The Formula for the Value Calculation is:

$$\left\lfloor \frac{GenSigTimeoutTime}{kIlRxCycleTime} \right\rfloor = TableValue$$

5.3 Transmit Signals

5.3.1 Initial Transmit Value

This chapter describes the configuration of Tx signal default values that are set are set in the Interaction Layer state transitions `IlInit` / `IlTxStart` / `IlTxStop`.

The following message layout is used as example in the following chapters.

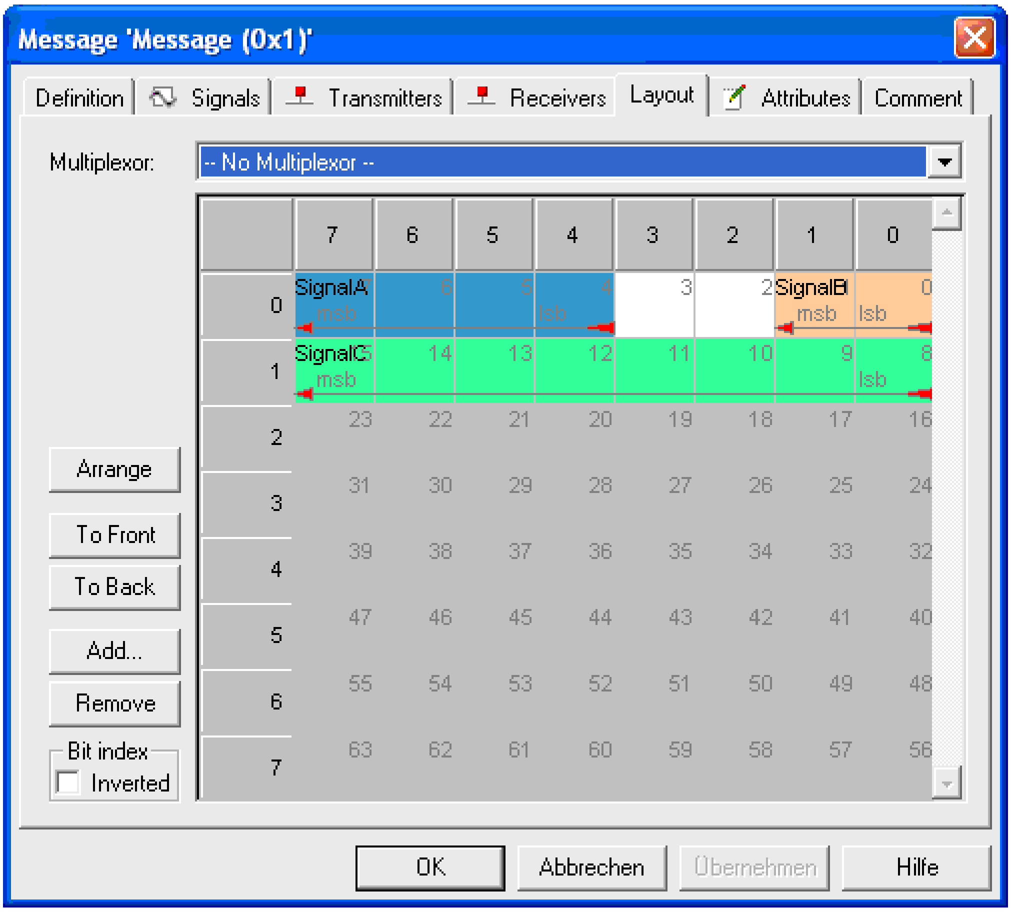

Figure 5-1 Signal layout of the example message

5.3.1.1 GENy Configuration

In order to have the calibration (post build) capability of default values for all transmit signals the ‘Init default’, ‘Start default’ and ‘Stop default’ have to be activated in GENy as shown in the following figures.

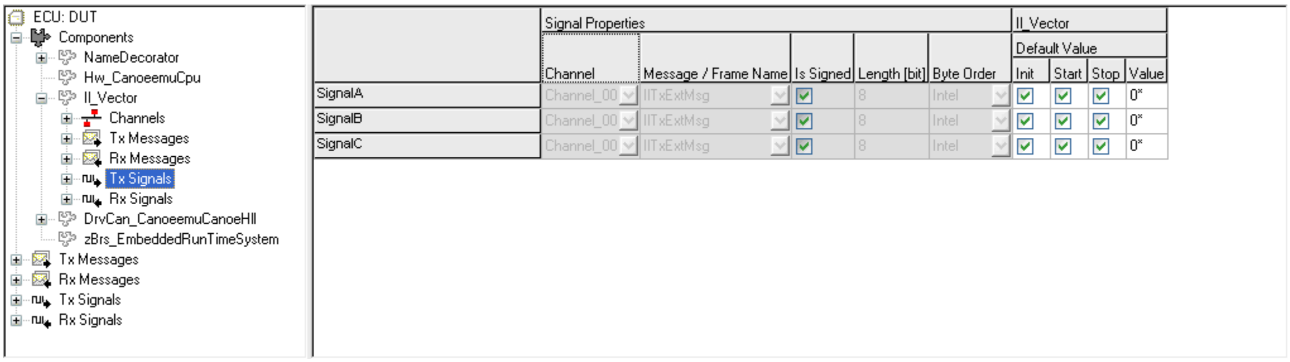

Figure 5-2 GENy configuration of the ‚Tx Signals’ view with the example message’s signals

5.3.1.2 Initial Default Values of Transmit Signals

The generated file `gmlcal.c` contains for each transmit message a table named `<MessageName>IlTxDefaultInitValue` with the data type `vuint8 []`.
The size of the table depends on the length of the data of the corresponding message that is relevant for your node. The table contains the default values of the message.

Example

The SignalA is located in the first byte of the array. The signal starts in bit 4 and ends in bit 7 as shown in Figure 5-1.

If for example the default value for the SignalA in the Intel format is 1 and for the SignalB and SignalC is 0, the table would contain the following values:

```
GMLCAL_MEMROM0 GMLCAL_MEMROM1 vuint8 GMLCAL_MEMROM2
MessageIlTxDefaultInitValue[] = {
   0x10
  ,0x00
};
```

Caution

The byte order (Little Endian / Big Endian) has to be taken in account when setting up the default values for the signals.

5.3.1.3 Start and Stop Default Values of Transmit Signals

The generated file `gmlcal.c` contains for each transmit message a table named `<*MessageName*>IlTxDefaultStartMask` for the `IlTxStart` transition and `<*MessageName*>IlTxDefaultStopMask` for the `IlTxStop` transition with the data type `vuint8 []`.
The size of these tables depends on the length of the data of the corresponding message that is relevant for your node. These tables contain masks that are applied on the corresponding default value of the table `<*MessageName*>IlTxDefaultInitValue`.

The mask table contains a 'set bit' at each bit position, where the default value is applied.

Example
If for example the default value for the SignalA shall be set at `IlTxStart` and not for SignalB and SignalC, the table would contain the following values:

```
GMLCAL_MEMROM0 GMLCAL_MEMROM1 vuint8 GMLCAL_MEMROM2
MessageIlTxDefaultStartMask[] = {
   0xF0
  ,0x00
};
```

5.4 Receive Signals

Receive signal calibration is not supported by the GMLAN Handler.

6 Nm_Gmlan_Gm : Network Management

This chapter describes the calibration capabilities of the Network Management.

6.1 nmVNMFStartSendCalCnt (Bus Wakeup Delay Time)

This is the time between transmission of a HLVW message and a VNMF-Init message when activating a Virtual Network. Also known as the Bus Wakeup Delay Time. Time is measured in multiples of the Nm Cycle Time. Default value is 100ms.

Symbol	Description
`BusWakeupDelayTime`	Delay time in ms.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left\lceil \frac{NmInitDelayTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right\rceil = TableValue$$

	Info For a single channel configuration, there is only a constant generated. For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.2 nmVNMFSendTimeCalCnt (VMNMF Periodic rate)

Time between sending continue VNMFs. Time is measured as multiples of the Nm Cycle Time. Default value is 3000ms if the attribute GenMsgCycleTime for this VNMF message is not set to a different value in the database.

Symbol	Description
`VNMFPeriodicRate`	Cycle of the VNMF message time in ms.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left\lceil \frac{VNMFPeriodicRate + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right\rceil = TableValue$$

	Info For a single channel configuration, there is only a constant generated. For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.3 nmBusoffRecoveryTimeCalCnt (BusOff Recovery Delay Time)

The ‘BusOff recovery Delay Time’ is the time to wait after a BUS-OFF event to reset the CAN controller and attempt recovery. The time is measured as multiples of the Nm Cycle Time. The value corresponds to the “BusOff Recovery Time” field in the channel properties of GENy.

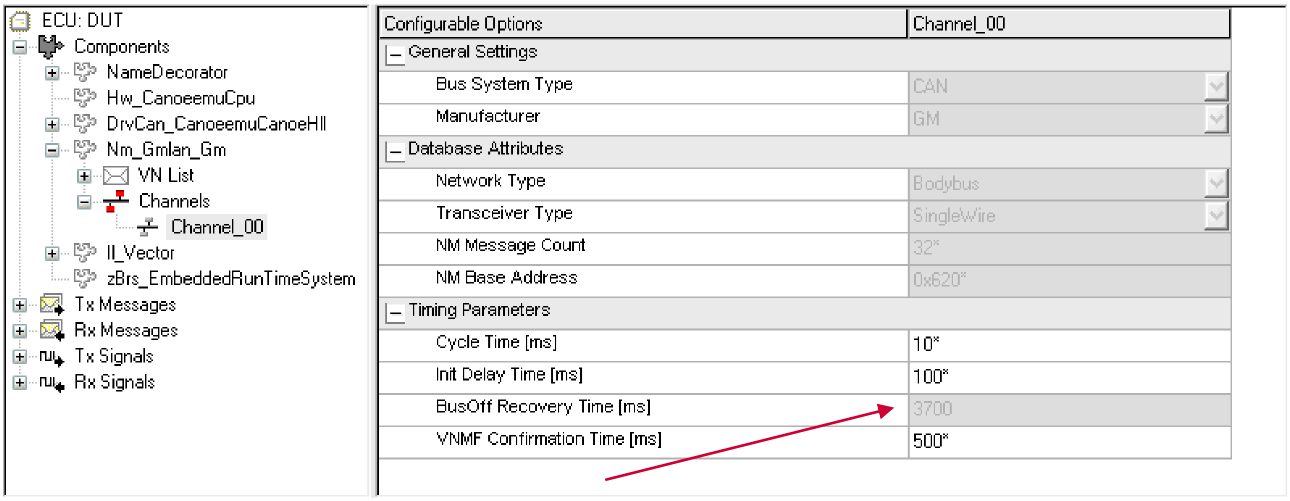

Figure 6-1 “BusOff Recovery Time” configuration in GENy

Symbol	Description
`BusOffRecoveryDelayTime`	Delay time in ms for Bus-Off recovery.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left[\frac{BusOffRecoveryDelayTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right] = TableValue$$

	Info For a single channel configuration, there is only a constant generated. For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.4 nmInitDelayTimeCalCnt (Init Delay Time)

For initially-active Virtual Networks, the ‘Init DelayTime’ defines the time between reception of a HLVW message and transmission of Node Communication Active (NCA), periodic, and send on-init messages. The time is measured as multiples of the Nm Cycle Time. The value corresponds to the “Init Delay Time” field in the channel properties of GENy.

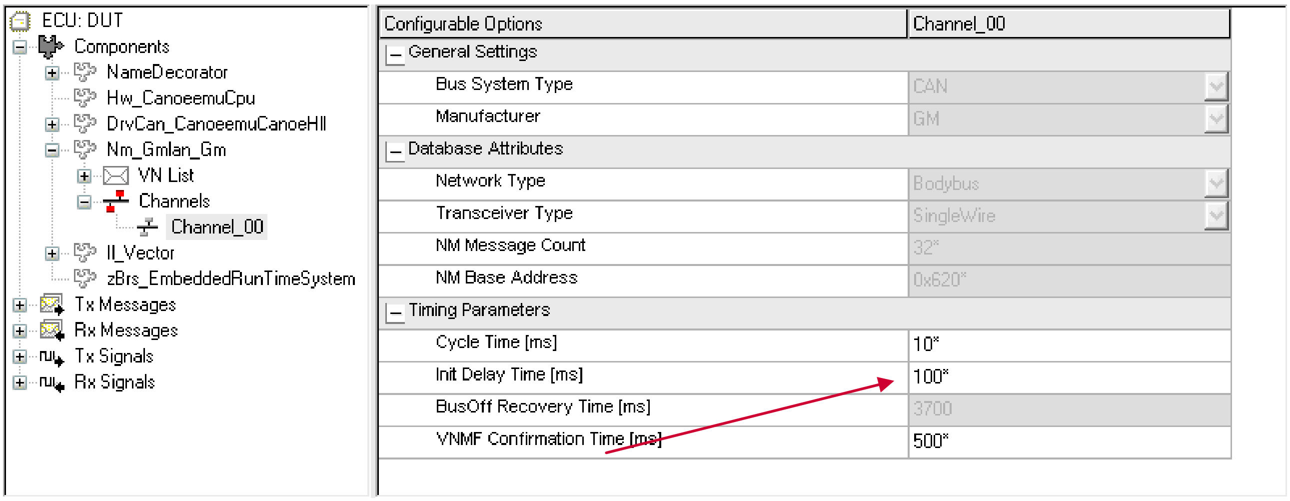

Figure 6-2 “Init Delay Time” configuration in GENy

Symbol	Description
`NmInitDelayTime`	Delay time in ms.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left[\frac{NmInitDelayTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right] = TableValue$$

Info
For a single channel configuration, there is only a constant generated.
For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.5 nmSleepTransitionDelayTimeCalCnt (Sleep Transition Time)

The ‘Sleep Transition Time’ defines an extra delay time between CAN driver initialization and setting the transceiver into sleep mode during shut down. This extra time gap between `CanInit()` and `ApplTrcvrSleepMode()`provides additional protection against missing of wake-up messages (HLVW).

The total time is calculated by adding the 'Bus Wakeup Delay Time' (chapter 6.1) and 'Sleep Transition Time' (this value). Note: The total time must not exceed 4 seconds!

The value corresponds to the “Init Delay Time” field in the channel properties of GENy. The default value is 50ms.

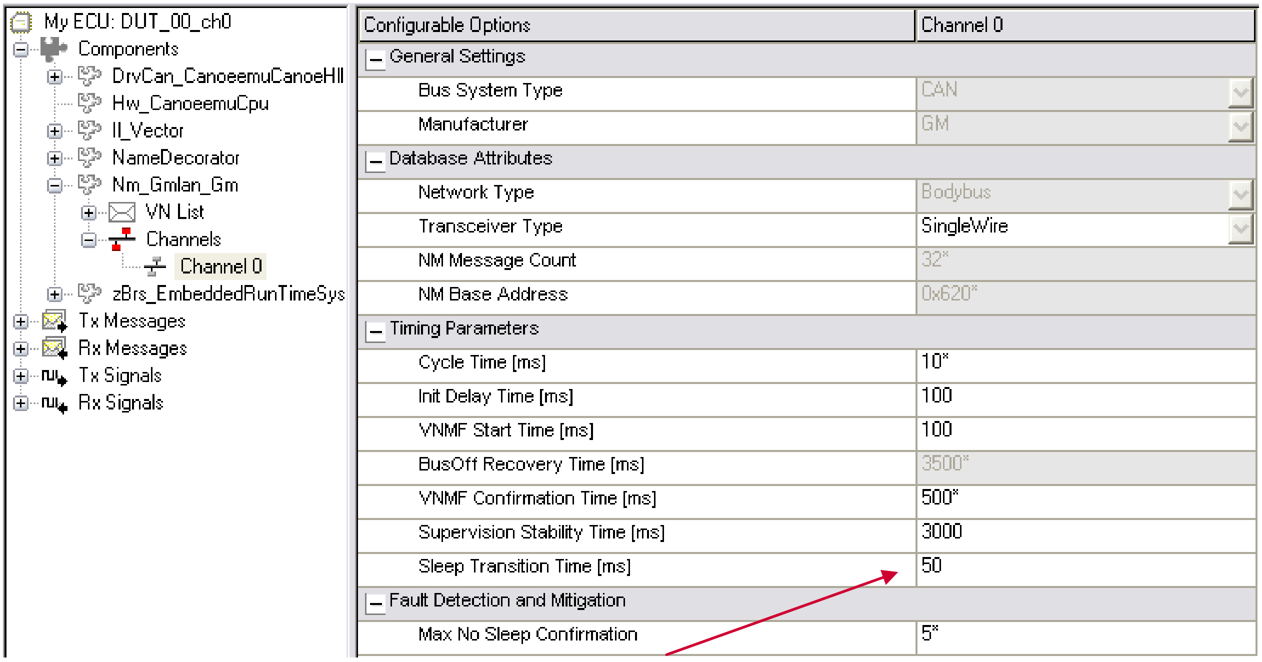

Figure 6-3 "Sleep Transition Time" configuration in GENy

Time is measured as multiples of the Nm Cycle Time.

Symbol	Description
`NmSleepTransitionTime`	Delay time in ms.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left[\frac{NmSleepTransitionTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME}\right] = TableValue$$

Info
For a single channel configuration, there is only a constant generated.
For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.6 nmSupervisionStabilityTimeCalCnt (Supervision Stability Time)

The ‘Supervision Stability Time’ defines a delay time between activation of a VN and start of Rx supervision of the corresponding signals. It is used to avoid 'Loss of Communication' DTCs due to transient conditions after VN activation.

The value corresponds to the “Init Delay Time” field in the channel properties of GENy. It is derived by the dbc attribute 'NodeSuprvStabilityTime'. If attribute does not exist, a default value of 5000ms is used.

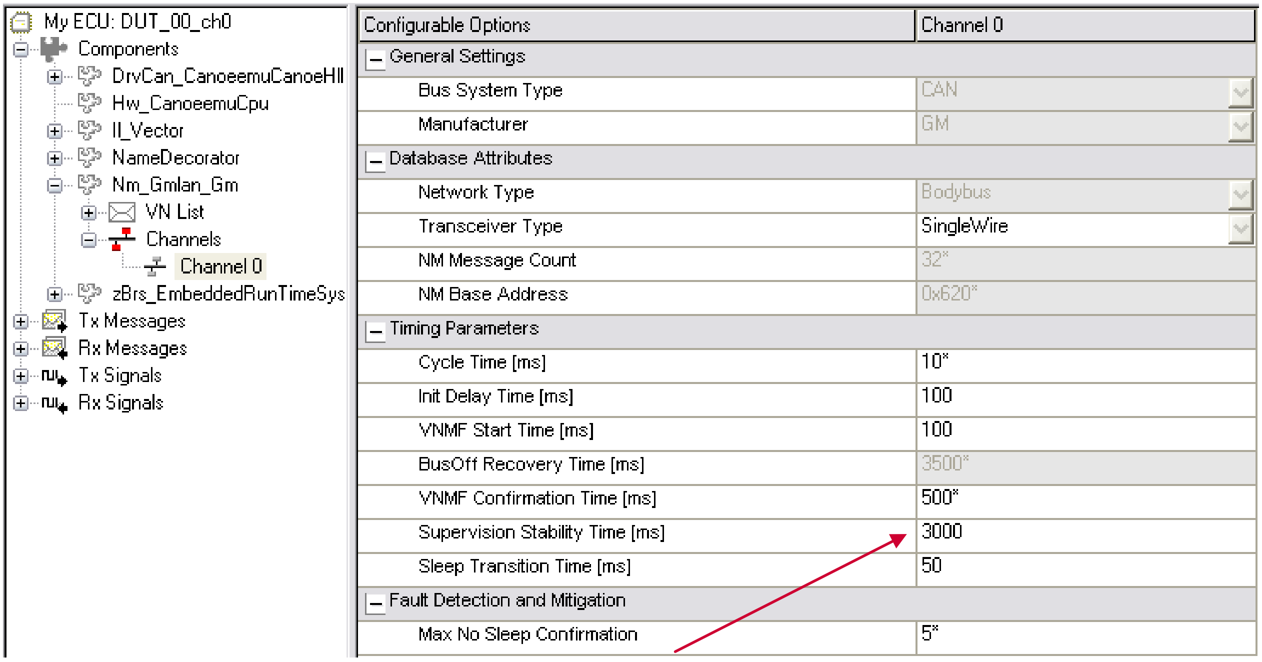

Figure 6-4 "Supervision Stability Time" configuration in GENy

Time is measured as multiples of the Nm Cycle Time.

Symbol	Description
`NmSuprvStabilityTime`	Delay time in ms.
`NM_CYCLETIME`	The call cycle time of the IlNwmTask of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left[\frac{NmSuprvStabilityTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right] = TableValue$$

Info

For a single channel configuration, there is only a constant generated.
For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.7 nmMaxApplShutDownDenyCnt (Max No Sleep Confirmation)

This value is only used if ‘Fault Detection and Mitigation’ and ‘Sleep Confirmation’ are both enabled in GENy.

The value defines a threshold for number of times the application may deny the transition to sleep mode within `ApplNwmSleepConfirmation()`. For detailed description of Fault Detection and Mitigation Algorithm see chapter 3.11 in [4].

The value corresponds to the “Max No Sleep Confirmation” field in the channel properties of GENy. The default value is 5.

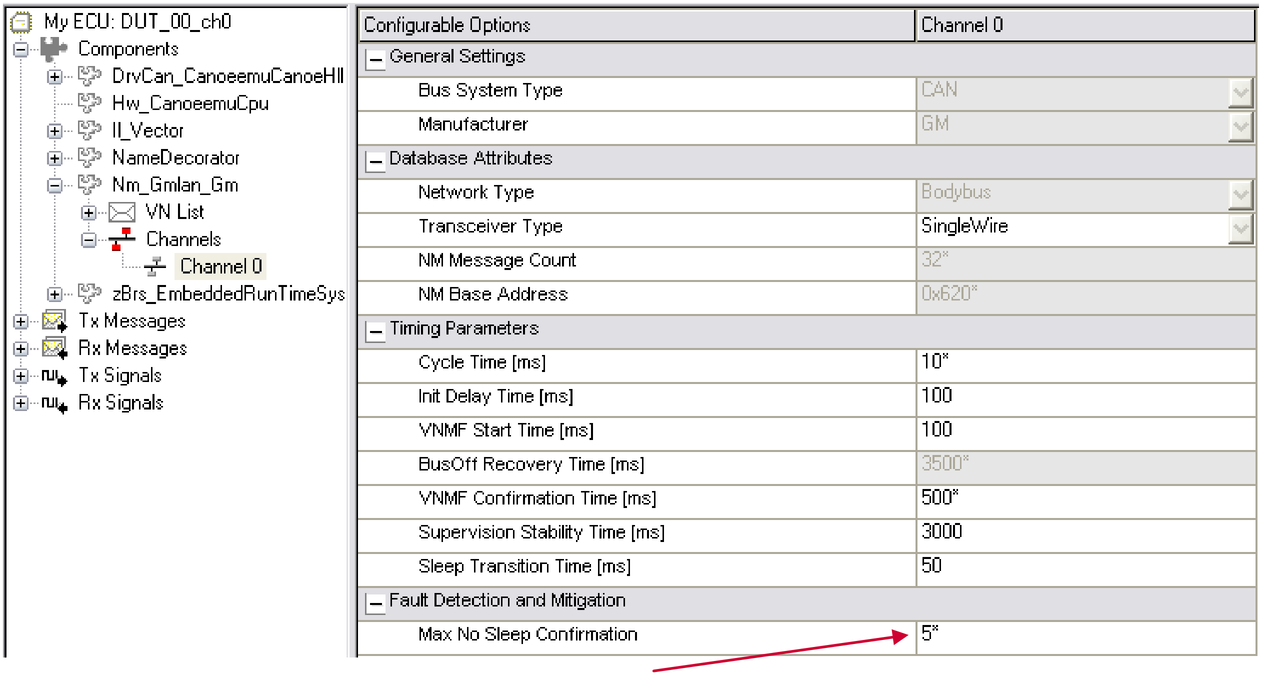

Figure 6-5 "Max No Sleep Confirmation" configuration in GENy

The value is generated directly as defined in GENy.

Info
For a single channel configuration, there is only a constant generated.
For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

6.8 GMLANNodeStatusTimeoutTimeCalCnt (Node Communication Active Frame Timeout)

The timeout for incoming Node Communication Active (NCA) messages is configured with `kGMLANNodeStatusTimeoutTimeCalCnt`. Failure to receive a NCA message in this period indicates that a node has failed. Time is measured as multiples of the Nm Cycle Time. The value corresponds to the value of the NodeStatusMsgTimeoutTime attribute in the database.

Symbol	Description
`NodeStatusMsgTimeoutTime`	Timeout time in ms.
`NM_CYCLETIME`	The call cycle time of the `IlNwmTask` of the dependent channel in ms.
`TableValue`	The value in the table for the dependent NM channel.

$$\left\lfloor \frac{NoteStatusMsgTimeoutTime + NM\_CYCLETIME - 1}{NM\_CYCLETIME} \right\rfloor = TableValue$$

	Info For a single channel configuration, there is only a constant generated. For a multi channel configuration there will be an array generated which will have as much entries as CAN channels are configured. The first entry is used for the first CAN channel, the second for the second CAN channel and so on.

7 Memory Definition file: MemDef.h

In order to allow calibration of the GMLAN Handler without modification of the generated configuration files all calibration parameters are located in the same file (`gmlcal.c`). Since the calibration parameters are stored in non-volatile memory (flash or EEPROM) the possibility of linking/locating these parameters in a separate memory section is provided.

There are mechanisms in order to support various compilers:

- Memory Mapping via pre-processor directives (`#pragma`)
- Linking of tables with memory qualifiers

If necessary both mechanism may be combined.

7.1 Memory Mapping

The memory mapping with pre-processor directives is done with the definition of sections that are embraced with a start definition that is followed by MemDef.h and a stop definition that is followed by MemDef.h. By adding `#pragma` definitions at the beginning and end of a section the parameters (tables) in-between may be linked in to a defined memory section.

Example

The following code shows a partial extract of gmlcal.c and a the mapping of the calibration parameters to the section `CALIBRATION`:

gmlcal.c:

```
#define GMLCAL_START_SEC_CONST
#include "MemDef.h"

GMLCAL_MEMROM0 GMLCAL_MEMROM1 canuint16 GMLCAL_MEMROM2
nmBusoffRecoverTimeCalCnt = (NM_BUSOFF_RECOVER_TIME + NM_CYCLETIME-
1)/NM_CYCLETIME;

#define GMLCAL_STOP_SEC_CONST
#include "MemDef.h"
```

MemDef.h:

```
/* Definition of section for calibration parameters. */
#if defined ( GMLCAL_START_SEC_CONST )
  #undef GMLCAL_START_SEC_CONST
  #pragma section .CALIBRATION
#endif

/* Definition of section for default parameters. */
#if defined ( GMLCAL_STOP_SEC_CONST )
  #undef GMLCAL_STOP_SEC_CONST
  #pragma section .DEFAULT
#endif
```

7.2 Memory Qualifiers

Separate memory qualifiers are used for all calibration parameters instead of the GMLAN Handler's Standard memory qualifiers in order to support linking these parameters to a separate memory section. These memory qualifiers have to be adapted to the user's needs.
The following table provides a list of these memory qualifiers that are defined in MemDef.h.

Memory Qualifier	Default definition
GMLCAL_MEMROM0	V_MEMROM0
GMLCAL_MEMROM1	V_MEMROM1
GMLCAL_MEMROM2	V_MEMROM2 (usually defined to const)
GMLCAL_MEMROM3	V_MEMROM3

The memory qualifiers defined in MemDef.h are exclusive used in gmlcal.c and gmlcal.h

The following example shows how the memory qualifiers are used in gmlcal.c.

Example

The following example shows how the memory qualifiers are used in gmlcal.c / gmlcal.h:

```
GMLCAL_MEMROM0 extern GMLCAL_MEMROM1 canuint16 GMLCAL_MEMROM2
nmBusoffRecoverTimeCalCnt;
```

8 Contact

Visit our website for more information on

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

www.vector.com