B Network management (NetMng)

Overview of the component

Network management involves the task of monitoring the different nodes present on the CAN bus and also handling the CAN related errors. The aim of monitoring the nodes is to know about the reliability of the network at any given instant. The Network Manager uses the interface functions of the Data Link Layer to monitor the nodes and to handle the errors. The application software uses the information provided by the Network Manager to know about the status of nodes and also the status of the CAN Bus.

Structure of the component

Monitoring CAN

The local control unit (EDC) is referred to as the Own Node and all the other control units present on the network are referred to as the Remote Nodes. It is not possible to monitor a node if it only receives messages. For monitoring a node it should transmit at least one periodic CAN message.The global network monitoring involves monitoring the Own Node as well as the Remote Nodes. The properties of CAN messages handled by the EDC are shown in the next table.

The Own Node is monitored by checking whether the node is able to transmit the Key Message.

The Remote Node is monitored by checking whether the Key Message from the Remote Node is received and the data length of the Key Message is correct.

Additionally the CAN controller check its bus state and reports it to the network manager.

The current Frame Manager and Net Manager implementation meets the FIAT "Network Specification & Message Map C-CAN Ed.5 - Rev. B" (FLORENCE) with adaptive cruise control (ACC).

Table 356: List of CAN messages and CAN nodes

No. *)	Message name	Short name	Identifier	Trigger	Data length	Sender node
0	Status-C-CAN	STNC	561hex	50 ms	8 byte	NCM
1	MOT1	MOT1	361hex	10 ms	8 byte	NCM
2	MOT2	MOT2	3A1hex	10 ms	8 byte	NCM
3	MOTGEAR	MOGE	061hex	10 ms	8 byte	NCM
4	ASR1	ASR1	146hex	10 ms	8 byte	NFR
5	ASR2	ASR2	166hex	10 ms	8 byte	NFR
6	STATUS-B-CAN	STNB	560hex	100 ms	8 byte	NBC
7	GEARMOT	GEMO	0CBhex	10 ms	8 byte	NCA
8	GEARMOT2	GEM2	0DBhex	10 ms	8 byte	NCA
9	EndOfLineConfig	EOLC	680hex	1000 ms	8 byte	NBC
10	ImmobilizerRequest	IMRQ	041hex	event	7 byte	NCM
11	ImmobilizerResponse	IMRP	040hex	event	7 byte	NBC
12	Adaptive Cruise Control 1	ACC1	3B0hex	20 ms	6 bytes	NAC
13	Brake assistant	BA	433hex	10 ms	8 bytes	NBA
14	not used					
15	not used					

The CAN 4.11 functionality is selected over the switch **FrmMng_swtCANProtocol_C**.

Table 357: List of CAN messages and CAN nodes (CAN 4.11)

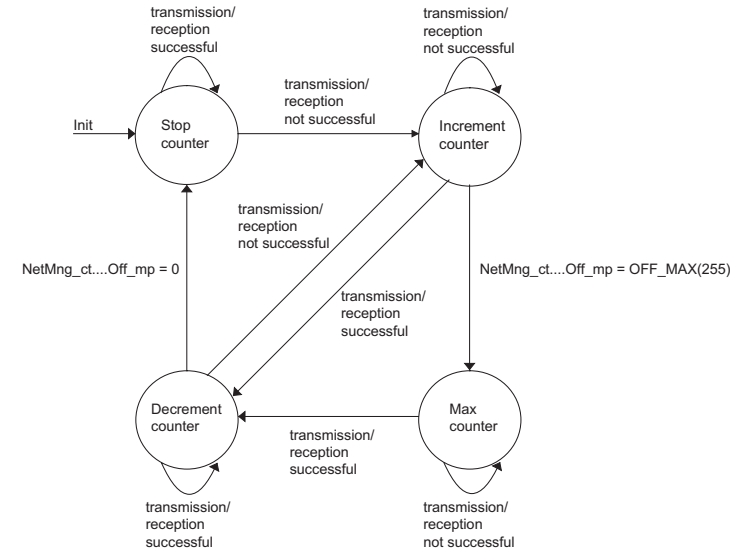
No. *)	Message name	Short name	Identifier	Trigger	Data length	Sender node
0	MOT1	MOT1	316hex	10 ms	8 byte	NCM
1	MOT2	MOT2	329hex	10 ms	8 byte	NCM
2	MOT3	MOT3	235hex	10 ms	8 byte	NCM
3	ASR1	ASR1	153hex	10 ms	8 byte	NFR
4	EGS	GEMO	43Fhex	15 ms	8 byte	EGS
5..15	not used					

*) No. refers to the bit position in **NetMng_stCANMsg**.

Monitoring CAN messages

The correct transmission or reception of CAN messages is monitored by the Data Link Layer in the status message **NetMng_stCANMsg**. Each CAN message is represented by a bit in the status message (for bit position see table above). If the bit is set the corresponding CAN message was not correct received or transmitted. For each CAN message a short term debouncing counter with calibratable increments, decrements and limits is implemented.

Figure 997 : Monitoring CAN messages, states



Stop:	NetMng_ct....Off_mp = 0
Max:	NetMng_ct....Off_mp = OFF_MAX
Increment:	NetMng_ct....Off_mp = MAX (NetMng_ct....Off_mp + NetMng_ct....OffInc_C, OFF_MAX)
Decrement:	NetMng_ct....Off_mp = MIN (NetMng_ct....Off_mp - NetMng_ct....OffDec_C, 0)

NetMng_6.dsf

Figure 998 : Monitoring CAN message, debouncing counter

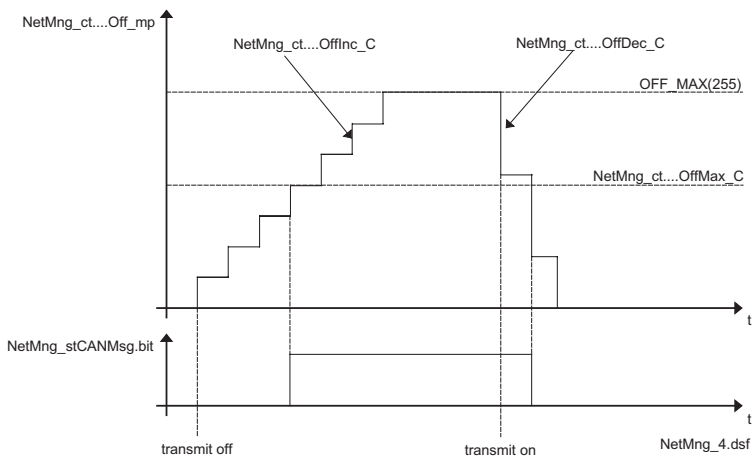


Table 358: Monitoring CAN messages, labels

No *)	Counter	Increment	Decrement	Limit
0	NetMng_ctSTNCOff_mp	NetMng_ctSTNCOffInc_C	NetMng_ctSTNCOffDec_C	NetMng_ctSTNCOffMax_C
1	NetMng_ctMOT1Off_mp	NetMng_ctMOT1OffInc_C	NetMng_ctMOT1OffDec_C	NetMng_ctMOT1OffMax_C
2	NetMng_ctMOT2Off_mp	NetMng_ctMOT2OffInc_C	NetMng_ctMOT2OffDec_C	NetMng_ctMOT2OffMax_C
3	NetMng_ctMOGEOff_mp	NetMng_ctMOGEOffInc_C	NetMng_ctMOGEOffDec_C	NetMng_ctMOGEOffMax_C
4	NetMng_ctASR1Off_mp	NetMng_ctASR1OffInc_C	NetMng_ctASR1OffDec_C	NetMng_ctASR1OffMax_C
5	NetMng_ctASR2Off_mp	NetMng_ctASR2OffInc_C	NetMng_ctASR2OffDec_C	NetMng_ctASR2OffMax_C
6	NetMng_ctSTNBOff_mp	NetMng_ctSTNBOffInc_C	NetMng_ctSTNBOffDec_C	NetMng_ctSTNBOffMax_C
7	NetMng_ctGEMOOff_mp	NetMng_ctGEMOOffInc_C	NetMng_ctGEMOOffDec_C	NetMng_ctGEMOOffMax_C
8	NetMng_ctGEM2Off_mp	NetMng_ctGEM2OffInc_C	NetMng_ctGEM2OffDec_C	NetMng_ctGEM2OffMax_C
9	NetMng_ctEOLCOff_mp	NetMng_ctEOLCOffInc_C	NetMng_ctEOLCOffDec_C	NetMng_ctEOLCOffMax_C
10	reserved			
11	reserved			
12	NetMng_ctACC1Off_mp	NetMng_ctACC1OffInc_C	NetMng_ctACC1OffDec_C	NetMng_ctACC1OffMax_C
13	NetMng_ctBAOff_mp	NetMng_ctBAOffInc_C	NetMng_ctBAOffDec_C	NetMng_ctBAOffMax_C
14	not used			
15	not used			

Table 359: Monitoring CAN 4.11 messages, labels

No *)	Counter	Increment	Decrement	Limit
0	NetMng_ctMOT1Off_mp	NetMng_ctMOT1OffInc_C	NetMng_ctMOT1OffDec_C	NetMng_ctMOT1OffMax_C
1	NetMng_ctMOT2Off_mp	NetMng_ctMOT2OffInc_C	NetMng_ctMOT2OffDec_C	NetMng_ctMOT2OffMax_C
2	NetMng_ctMOT3Off_mp	NetMng_ctMOGEOffInc_C	NetMng_ctMOGEOffDec_C	NetMng_ctMOGEOffMax_C
3	NetMng_ctASR1Off_mp	NetMng_ctASR1OffInc_C	NetMng_ctASR1OffDec_C	NetMng_ctASR1OffMax_C
4	NetMng_ctGEMOOff_mp	NetMng_ctGEMOOffInc_C	NetMng_ctGEMOOffDec_C	NetMng_ctGEMOOffMax_C

No *)	Counter	Increment	Decrement	Limit
5-15	Not used			

*) No. refers to the bit position in **NetMng_stCANMsg**.

Monitoring CAN nodes, Indirect Network Management (INM)

It is possible to monitor a CAN node if at least one periodic CAN message is transmitted. For each node the Key Messages can be chosen from the CAN message status **NetMng_stCANMsg** by a bit mask.

If one masked CAN message is not correctly received/transmitted the corresponding node is marked as Absent/Mute and the error debouncing is started in order to store the error in the error memory.

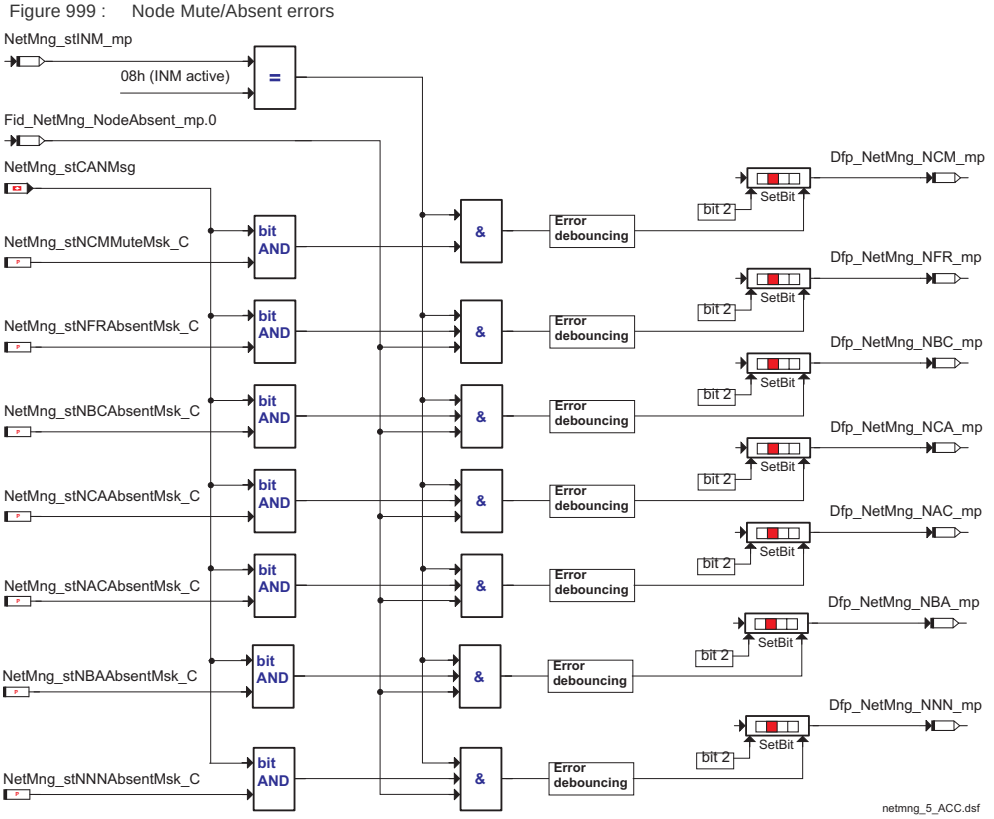
Table 360: Bit mask for node Mute/Absent monitoring

CAN Node	Error	Bit mask
NCM, engine control unit	**Dfp_NetMng_NCM_mp**	**NetMng_stNCMMuteMsk_C**
NFR, brake node	**Dfp_NetMng_NFR_mp**	**NetMng_stNFRAbsentMsk_C**
NBC, body computer	**Dfp_NetMng_NBC_mp**	**NetMng_stNBCAbsentMsk_C**
NCA, gear node	**Dfp_NetMng_NCA_mp**	**NetMng_stNCAAbsentMsk_C**
NAC, ACC node	**Dfp_NetMng_NAC_mp**	**NetMng_stNACAbsentMsk_C**
NBA, brake assistant node	**Dfp_NetMng_NBA_mp**	**NetMng_stNBAAbsentMsk_C**
NNN, not used	**Dfp_NetMng_NNN_mp**	**NetMng_stNNNAbsentMsk_C**

Table 361: Node Mute/Absent error debouncing

CAN Node	Error	Intact debouncing time	Defect debouncing time
NCM, engine control unit	**Dfp_NetMng_NCM_mp**	**NetMng_DebNCMMuteOk_C**	**NetMng_DebNCMMuteDef_C**
NFR, brake node	**Dfp_NetMng_NFR_mp**	**NetMng_DebNFRAbsentOk_C**	**NetMng_DebNFRAbsentDef_C**
NBC, body computer	**Dfp_NetMng_NBC_mp**	**NetMng_DebNBCAbsentOk_C**	**NetMng_DebNBCAbsentDef_C**
NCA, gear node	**Dfp_NetMng_NCA_mp**	**NetMng_DebNCAAbsentOk_C**	**NetMng_DebNCAAbsentDef_C**
NAC, adapt. cruise contr.	**Dfp_NetMng_NAC_mp**	**NetMng_DebNACAbsentOk_C**	**NetMng_DebNACAbsentDef_C**
NBA, brake assistant	**Dfp_NetMng_NBA_mp**	**NetMng_DebNBAAbsentOk_C**	**NetMng_DebNBAAbsentDef_C**
NNN, not used	**Dfp_NetMng_NNN_mp**	**NetMng_DebNNNAbsentOk_C**	**NetMng_DebNNNAbsentDef_C**

Figure 999 : Node Mute/Absent errors



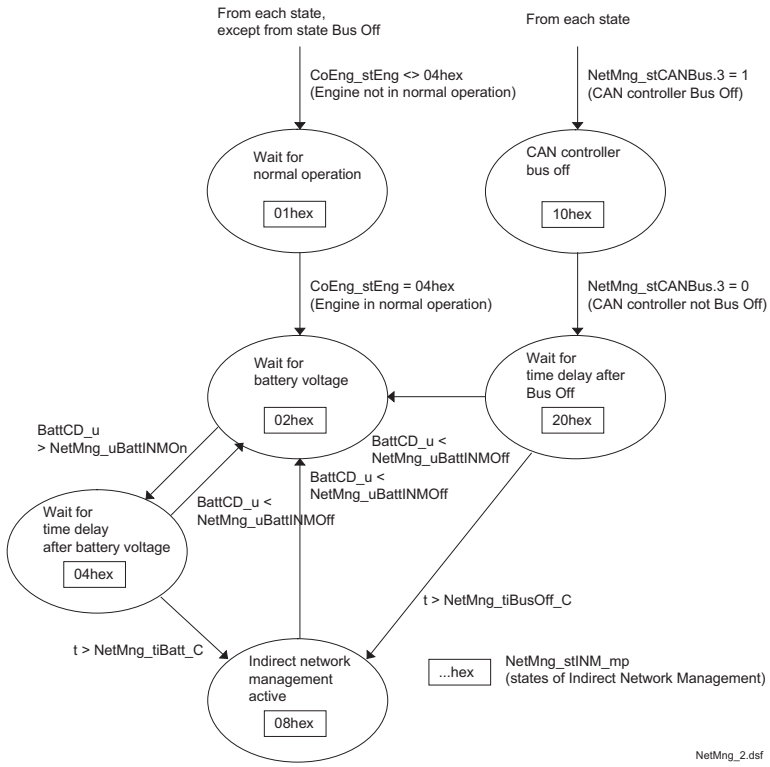
Release of Indirect Network Management

To suppress error entries if battery voltage is too low for some Remote Nodes to work properly, e.g. during engine start phase, the nodes are monitored only after the required conditions are satisfied. The conditions to enable the node monitoring is shown in the following table and figure.

Table 362: Release of Indirect Network Management

NetMng_stINM_mp	
01hex	INM waits for engine in normal operation **CoEng_stEng** = COENG_STNORMAL(04hex)
02hex	INM waits for battery voltage above threshold **BattCD_u** > **NetMng_uBattINMOn_C**.
04hex	INM waits for delay time (**NetMng_tiBatt_C**) after battery voltage threshold has exeeded
08hex	INM is active
10hex	CAN controller is in Bus Off state: **NetMng_stCANBus**.3 = set
20hex	INM waits for delay time (**NetMng_tiBusOff_C**) after CAN controller has left Bus Off state

Figure 1000 : Release of Indirect Network Monitoring



Bus states

Monitoring bus state

The states Error Active, Error Passive and Bus Off are monitored by the CAN controller itself according to the CAN protocol. The state confirmed Bus Off is additionally debounced by the Network Manager.

Table 363: Bus status - **NetMng_stCANBus**

Bit position	State
bit 0	NETMNG_ERRORACTIVE
bit 1	NETMNG_ERRORPASSIVE
bit 3	NETMNG_BUSOFF
bit 7	NETMNG_CONFIRMEDBUSOFF

Figure 1001 : Bus off debouncing, states

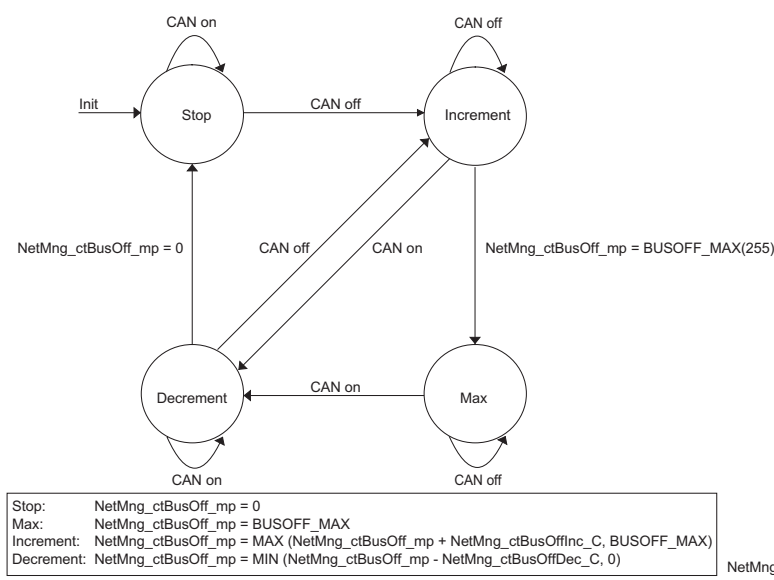


Figure 1002 : Bus Off debouncing, counter

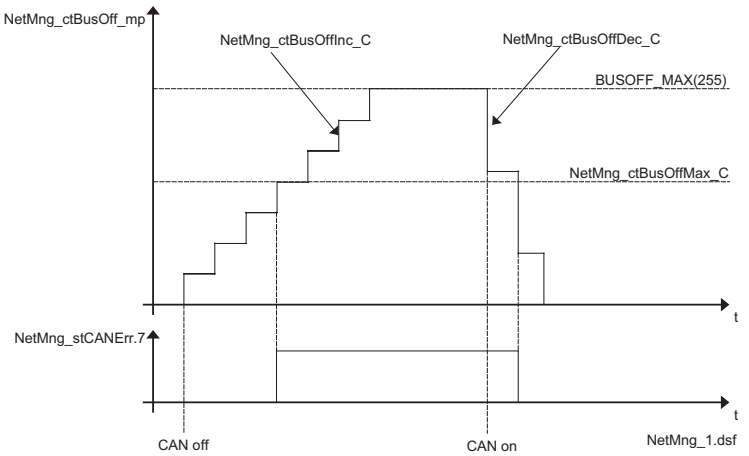
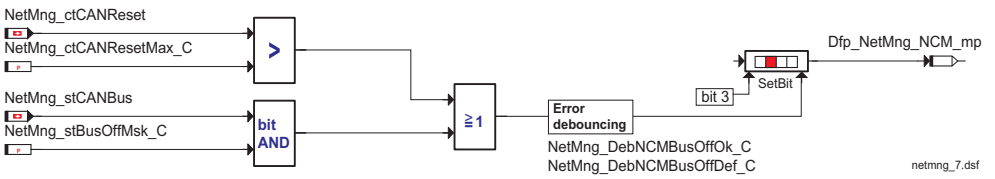


Figure 1003 : Bus Off error, monitoring



Reinitialization

After the state changes to NETMNG_CONFIRMEDBUSOFF, a timer is started. When the delay time **NetMng_tiCANReset_C** is exceeded the CAN chip will be reinitialized. With each reinitialization a counter **NetMng_ctCANReset** is incremented by 1. The counter has a maximum value of 255.

Output values

NetMng_ctCANReset: CAN reinitialisation counter after Bus Off state [-] <UBYTE> --->
NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->

Input values

BattCD_u: battery voltage [mV] <SWORD> --->
CoEng_stEng: current engine state [-] <UBYTE> --->
Eng_nAvrg: average engine speed [rpm] <SWORD> --->

Measuring points

Dfp_NetMng_NAC_mp: Fault path to reports NAC errors [-] <UWORD> --->
Dfp_NetMng_NBA_mp: Fault path to reports NBA errors [-] <UWORD> --->
Dfp_NetMng_NBC_mp: Measurement point for fault path Dfp_NetMng_NBC (body computer) [-] <UWORD> --->
Dfp_NetMng_NCA_mp: Measurement point for fault path Dfp_NetMng_NCA (gear node) [-] <UWORD> --->
Dfp_NetMng_NCM_mp: Measurement point for fault path Dfp_NetMng_NCM (engine control unit) [-] <UWORD> --->
Dfp_NetMng_NFR_mp: Measurement point for fault path Dfp_NetMng_NFR (brake node) [-] <UWORD> --->
Dfp_NetMng_NNN_mp: Measurement point for fault path Dfp_NetMng_NNN [-] <UWORD> --->
Fid_NetMng_NodeAbsent_mp: Measurement point for Function Identifier Fid_NodeAbsent, inhibits NetMng_NodeAbsent errors [-] <UBYTE> --->
NetMng_ctACC1Off_mp: Fail counter of message ACC1 [-] <UBYTE> --->
NetMng_ctASR1Off_mp: Defectcounter of the ASR200-CAN-Message [-] <UBYTE> --->
NetMng_ctASR2Off_mp: Defectcounter of the ASR208-CAN-Message [-] <UBYTE> --->
NetMng_ctBAOff_mp: Fail counter of message BA [-] <UBYTE> --->
NetMng_ctEOLCOff_mp: Fail counter of CAN message EndOfLineConfiguration [-] <UBYTE> --->
NetMng_ctGEM2Off_mp: Fail counter of CAN message GEARMOT2 [-] <UBYTE> --->
NetMng_ctGEMOOff_mp: Fail counter of CAN message GEARMOT [-] <UBYTE> --->
NetMng_ctMOGEOff_mp: Fail counter of CAN message MOTGEAR [-] <UBYTE> --->
NetMng_ctMOT1Off_mp: Fail counter of CAN message MOT1 [-] <UBYTE> --->
NetMng_ctMOT2Off_mp: Fail counter of CAN message MOT2 [-] <UBYTE> --->
NetMng_ctMOT3Off_mp: Timeout-counter of the CAN-Id MOT3 [-] <UBYTE> --->
NetMng_ctSTNBOff_mp: Fail counter of CAN message STATUS-B-CAN [-] <UBYTE> --->
NetMng_ctSTNCOff_mp: Fail counter of CAN message STATUS-C-CAN [-] <UBYTE> --->
NetMng_stINM_mp: Indirect Network Manager (INM) status [-] <UBYTE> --->

Application parameters

NetMng_ctACC1OffDec_C: Decrement of ACC1 counter<value> --->
NetMng_ctACC1OffInc_C: Increment of ACC1 counter<value> --->
NetMng_ctACC1OffMax_C: Defect threshold of ACC1 counter<value> --->
NetMng_ctASR1OffDec_C: Decrement of the ASR1 absent counter<value> --->
NetMng_ctASR1OffInc_C: Increment of the ASR1 absent counter<value> --->
NetMng_ctASR1OffMax_C: Defect threshold of the ASR1 CAN-Message<value> --->
NetMng_ctASR2OffDec_C: Decrement of the ASR2 absent counter<value> --->
NetMng_ctASR2OffInc_C: Increment of the ASR2 absent counter<value> --->
NetMng_ctASR2OffMax_C: Defect threshold of the ASR2 CAN-Message<value> --->
NetMng_ctBAOffDec_C: Decrement of BA counter<value> --->
NetMng_ctBAOffInc_C: Increment of BA counter<value> --->
NetMng_ctBAOffMax_C: Defect threshold of BA counter<value> --->
NetMng_ctBusOffDec_C: Decrement of the Bus-Off-counter<value> --->
NetMng_ctBusOffInc_C: Increment of the Bus-Off-counter<value> --->
NetMng_ctBusOffMax_C: Defect threshold of the Bus-Off-counter<value> --->
NetMng_ctCANResetMax_C: Max. value of CAN reinitialisation counter after Bus Off state<value> --->
NetMng_ctEOLCOffDec_C: Decrement of the END OF LINE CONF absent counter<value> --->
NetMng_ctEOLCOffInc_C: Increment of the END OF LINE CONF absent counter<value> --->
NetMng_ctEOLCOffMax_C: Defect threshold of the END OF LINE CONF absent counter<value> --->
NetMng_ctGEM2OffDec_C: Decrement of the GEARMOT2 absent counter<value> --->
NetMng_ctGEM2OffInc_C: Increment of the GEARMOT2 absent counter<value> --->
NetMng_ctGEM2OffMax_C: Defect threshold of the GEARMOT2 absent counter<value> --->
NetMng_ctGEMOOffDec_C: Decrement of the GEARMOT absent counter<value> --->
NetMng_ctGEMOOffInc_C: Increment of the GEARMOT absent counter<value> --->
NetMng_ctGEMOOffMax_C: Defect threshold of the GEARMOT absent counter<value> --->
NetMng_ctMOGEOffDec_C: Decrement of the MOTGEAR absent counter<value> --->
NetMng_ctMOGEOffInc_C: Increment of the MOTGEAR absent counter<value> --->
NetMng_ctMOGEOffMax_C: Defect threshold of the MOTGEAR absent counter<value> --->
NetMng_ctMOT1OffDec_C: Decrement of the MOT1 absent counter<value> --->
NetMng_ctMOT1OffInc_C: Increment of the MOT1 absent counter<value> --->
NetMng_ctMOT1OffMax_C: Defect threshold of the MOT1 absent counter<value> --->
NetMng_ctMOT2OffDec_C: Decrement of the MOT2 absent counter<value> --->
NetMng_ctMOT2OffInc_C: Increment of the MOT2 absent counter<value> --->
NetMng_ctMOT2OffMax_C: Defect threshold of the MOT2 absent counter<value> --->
NetMng_ctMOT3OffDec_C: Decrement value for the timeout counter of the CAN-Id MOT3<value> --->
NetMng_ctMOT3OffInc_C: Increment value for the timeout counter of the CAN-Id MOT3<value> --->
NetMng_ctMOT3OffMax_C: Timeout counter value of the CAN-Id MOT3<value> --->
NetMng_ctSTNBOffDec_C: Decrement of the STATUS-B-CAN absent counter<value> --->
NetMng_ctSTNBOffInc_C: Increment of the STATUS-B-CANabsent counter<value> --->
NetMng_ctSTNBOffMax_C: Defect threshold of the STATUS-B-CAN absent counter<value> --->
NetMng_ctSTNCOffDec_C: Decrement of the STATUS-C-NCM absent counter<value> --->
NetMng_ctSTNCOffInc_C: Increment of the STATUS-C-NCM absent counter<value> --->
NetMng_ctSTNCOffMax_C: Defect threshold of the STATUS-C-NCM absent counter<value> --->
NetMng_DebNACAbsentDef_C: Defect debounce time for NAC absent<value> --->
NetMng_DebNACAbsentOk_C: Healing debounce time for NAC absent<value> --->
NetMng_DebNBAAbsentDef_C: Defect debounce time for NBA absent<value> --->
NetMng_DebNBAAbsentOk_C: Healing debounce time for NBA absent<value> --->
NetMng_DebNBCAbsentDef_C: Defect debounce time for NBC Absent<value> --->
NetMng_DebNBCAbsentOk_C: Healing debounce time for NBC absent<value> --->
NetMng_DebNCAAbsentDef_C: Defect debounce time for NCA Absent<value> --->
NetMng_DebNCAAbsentOk_C: Healing debounce time for NCA absent<value> --->
NetMng_DebNCMMuteDef_C: Defect debounce time for NCM mute<value> --->
NetMng_DebNCMMuteOk_C: Healing debounce time for NCM mute<value> --->
NetMng_DebNFRAbsentDef_C: Defect debounce time for NFR Absent<value> --->
NetMng_DebNFRAbsentOk_C: Healing debounce time for NFR absent<value> --->
NetMng_DebNNNAbsentDef_C: Defect debounce time for NNN Absent<value> --->
NetMng_DebNNNAbsentOk_C: Healing debounce time for NNN absent<value> --->
NetMng_stNACAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NAC mute<value> --->
NetMng_stNBAAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NBA mute<value> --->
NetMng_stNBCAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NBC absent<value> --->
NetMng_stNCAAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NCA absent<value> --->
NetMng_stNCMMuteMsk_C: Bit mask for NetMng_stCANMsg to serve fault NCM mute<value> --->
NetMng_stNFRAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NFR absent<value> --->
NetMng_stNNNAbsentMsk_C: Bit mask for NetMng_stCANMsg to serve fault NNN absent<value> --->
NetMng_tiBatt_C: Time delay after battery voltage threshold exceeded to activate INM<value> --->
NetMng_tiBusOff_C: Time delay after Bus Off state to activate INM<value> --->
NetMng_tiCANReset_C: Time delay for CAN reset<value> --->
NetMng_uBattINMOff_C: Battery threshold to deactivate INM<value> --->

B.1 Frame Manager (Function) (FrmMng)

Functional overview

In this part of the Frame Manager documentation there is a description of the common features and definitions of the component Frame Manager.

Functional description

Supported CAN messages (FLORENCE)

The current Frame Manager implementation (CAN messages, IDs, time period, data length, bit positions of signals) meets the Fiat-specification for the CAN FLORENCE message map.

Table 364: CAN messages (FLORENCE)

Message name	Short name	Identifier	Trigger	Data length	Sender node
Status-C-CAN	STNC	561hex	50 ms	8 byte	NCM
MOT1	MOT1	361hex	10 ms	8 byte	NCM
MOT2	MOT2	3A1hex	10 ms	8 byte	NCM
MOTGEAR	MOGE	061hex	10 ms	8 byte	NCM
ASR1	ASR1	146hex	10 ms	8 byte	NFR
ASR2	ASR2	166hex	10 ms	8 byte	NFR
STATUS-B-CAN	STNB	560hex	100 ms	8 byte	NBC
GEARMOT	GEMO	0CBhex	10 ms	8 byte	NCA
GEARMOT2	GEM2	0DBhex	10 ms	8 byte	NCA
EndOfLineConfig	EOLC	680hex	1000 ms	8 byte	NBC
ImmobilizerRequest	IMRQ	041hex	event	7 byte	NCM
ImmobilizerResponse	IMRP	040hex	event	7 byte	NBC
ACC1	ACC1	3B0hex	20ms	6 bytes	NAC
BA	BA	433hex	10ms	8 bytes	NBA
not used					
not used					

Definition of bit and byte position in CAN frame

Table 365: Definition of bit position in CAN frame

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FrmMng_st....Ena0_C, FrmMng_st....Dfl0_C, FrmMng_st....Raw0	63	62	61	60	59	58	57	56
FrmMng_st....Ena1_C, FrmMng_st....Dfl1_C, FrmMng_st....Raw1	55	54	53	52	51	50	49	48
FrmMng_st....Ena2_C, FrmMng_st....Dfl2_C, FrmMng_st....Raw2	47	46	45	44	43	42	41	40
FrmMng_st....Ena3_C, FrmMng_st....Dfl3_C, FrmMng_st....Raw3	39	38	37	36	35	34	33	32
FrmMng_st....Ena4_C, FrmMng_st....Dfl4_C, FrmMng_st....Raw4	31	30	29	28	27	26	25	24
FrmMng_st....Ena5_C, FrmMng_st....Dfl5_C, FrmMng_st....Raw5	23	22	21	20	19	18	17	16
FrmMng_st....Ena6_C, FrmMng_st....Dfl6_C, FrmMng_st....Raw6	15	14	13	12	11	10	9	8
FrmMng_st....Ena7_C, FrmMng_st....Dfl7_C, FrmMng_st....Raw7	7	6	5	4	3	2	1	0

Supported CAN messages (CAN 4.11)

The current Frame Manager implementation (CAN messages, IDs, time period, data length, bit positions of signals) meets the Fiat-specification for the CAN 4.11 message map.

Table 366: CAN messages (CAN 4.11)

Message name	Short name	Identifier	Trigger	Data length	Sender node
MOT1	MOT1	316hex	10 ms	8 byte	NCM
MOT2	MOT2	329hex	10 ms	8 byte	NCM
MOT3	MOT3	235hex	10 ms	8 byte	NCM
ASR1	ASR1	153hex	10 ms	8 byte	NFR
GEM0	GEM0	43Fhex	15 ms	8 byte	EGS

Definition of bit and byte position in CAN frame for CAN 4.11

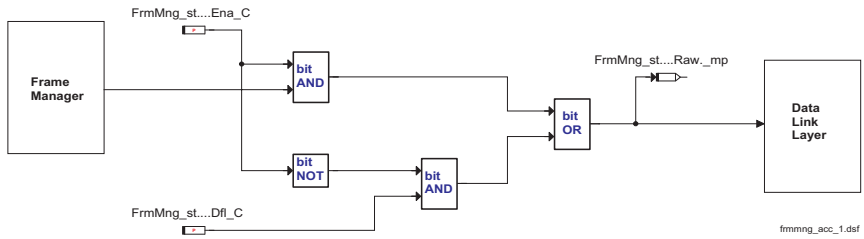
Table 367: Definition of bit position in CAN frame

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FrmMng_st....Ena0_C, FrmMng_st....Dfl0_C, FrmMng_st....Raw0	7	6	5	4	3	2	1	0
FrmMng_st....Ena1_C, FrmMng_st....Dfl1_C, FrmMng_st....Raw1	15	14	13	12	11	10	9	8
FrmMng_st....Ena2_C, FrmMng_st....Dfl2_C, FrmMng_st....Raw2	23	22	21	20	19	18	17	16
FrmMng_st....Ena3_C, FrmMng_st....Dfl3_C, FrmMng_st....Raw3	31	30	29	28	27	26	25	24
FrmMng_st....Ena4_C, FrmMng_st....Dfl4_C, FrmMng_st....Raw4	39	38	37	36	35	34	33	32
FrmMng_st....Ena5_C, FrmMng_st....Dfl5_C, FrmMng_st....Raw5	47	46	45	44	43	42	41	40
FrmMng_st....Ena6_C, FrmMng_st....Dfl6_C, FrmMng_st....Raw6	55	54	53	52	51	50	49	48
FrmMng_st....Ena7_C, FrmMng_st....Dfl7_C, FrmMng_st....Raw7	63	62	61	60	59	58	57	56

Enable mask for transmitted CAN messages

With the enable mask it can be chosen if the transmitted data bytes are calculated by the Frame Manager or taken from the default data set.

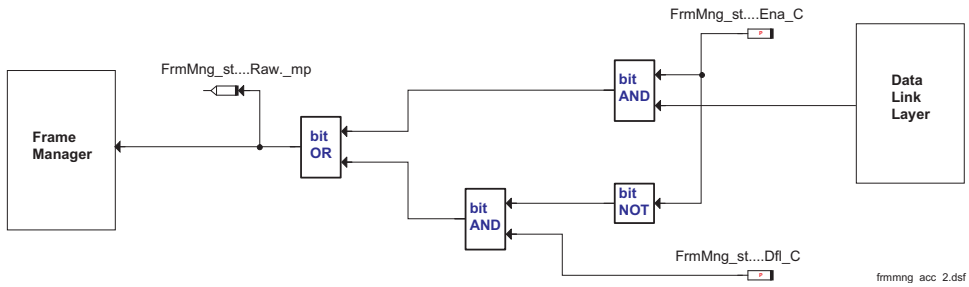
Figure 1004 : Enable mask for transmitted CAN messages



Enable mask for received CAN messages

With the enable mask it can be chosen if the data bytes are received from Data Link Layer or taken from the default data set.

Figure 1005 : Enable mask for received messages



Signal conversion

The signals that are transmitted/received need to be converted to match the ECU physical resolution and the CAN resolution. The conversions are needed only for signals carrying physical quantities like engine speed, coolant temperature, vehicle speed etc. and not for signals carrying logical information.

In common the following equation is used:

Equation 46: Signal conversion

$$CANValue(t) = \frac{(PhysicalValue(t)) + Offset)}{Slope}$$

Torque definitions

All engine torques are normalized torques expressed in percentage referred to the normalization value MOT2_MaxEngineTorqueNorm.

Indicated Torque

Engine torque due to combustion. This torque can have only positive values.

Equation 47: Indicated Torque

$$IndicatedTorque(t) = AbsoluteCrankshaftTorque(t) + FrictionTorque(t)$$

Friction Torque

Sum of engine mechanical losses due to gas exchange and auxiliary loads (oil pump, water pump, high pressure fuel pump, alternator, air condition compressor). This torque can have only positive values.

Absolute Crankshaft Torque

Engine torque at crankshaft. This torque can have positive and negative values.

Delta Torque

Torque difference between two Absolute Crankshaft Torques (or two Indicated Torques).

Send messages (FrmMng_Send)

Functional overview

In this chapter there is the description of CAN messages which are sent by the engine node.

Functional description

CAN-message MOTGEAR

ID: 061h, Data length: 8 byte, Period: 10ms

Table 368: MOTGEAR - frame overview

Signal name	Bit	Description
EngineTorqueForNCAValidData	63	Validation of EngineTorqueForNCA.
EngineTorqueReductNCAValidData	62	Validation of EngineTorqueReductNCA.
StopLightSwitchStsValidData	61	Validation of brake signal. (Not yet implemented.)
StopLightSwitchSts	60	Digital input of brake signal. (Not yet implemented.)
AirconSts	59	Power stage of air condition compressor.
EngineSpeedValidData	58	Not supported because engine speed is not sent in this message.
EngineTorqueForNCA	55-48	Crankshaft torque without external gearbox interventions (theoretical value).
EngineTorqueReductNCA	47-40	Reduction of engine torque due to external gearbox intervention.
EngineWaterTempForNCA	39-32	Analog signal input engine water temperature.
EngineWaterTempForNCAFailSts	31	Fail in the engine water temperature measurement/ plausibility check.
EngineTorqueWOExtReqValidData	30	Validation of EngineTorqueWOExtReq. (Not yet implemented.)
AirPressureValidData	29	Validation of AirPressure. Failure in the air pressure measurement. (Not yet implemented.)
EngineTorqueUnplausible	28	Indicates if calculated torques are not exact because of failures of engine air system, engine temperature or rail pressure. (Not yet implemented.)
EngineTorqueWOExtReq	23-16	Indicated torque without external gearbox or brake node interventions (theoretical value). (Not yet implemented.)
AirPressure	15-8	Analog signal input atmospheric pressure. (Not yet implemented.)

Figure 1006 : CAN-message MOTGEAR (1)

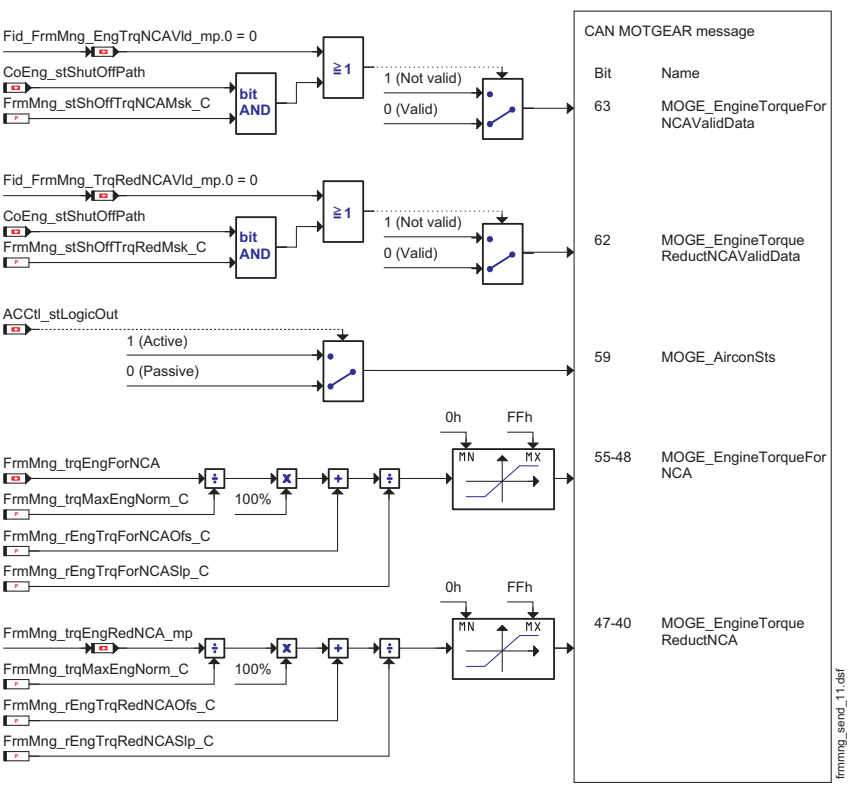


Figure 1007 : CAN-message MOTGEAR (2)

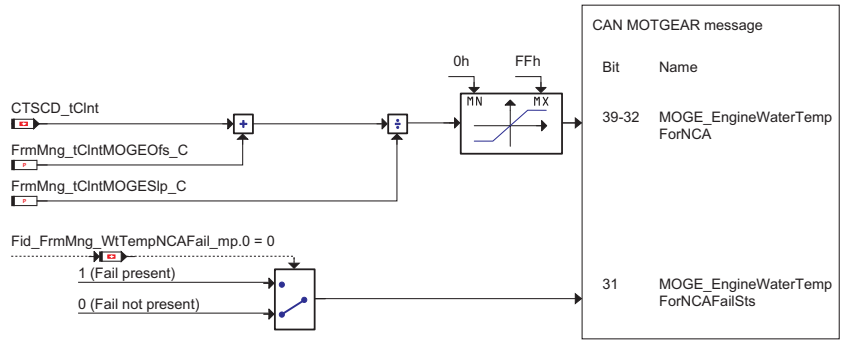
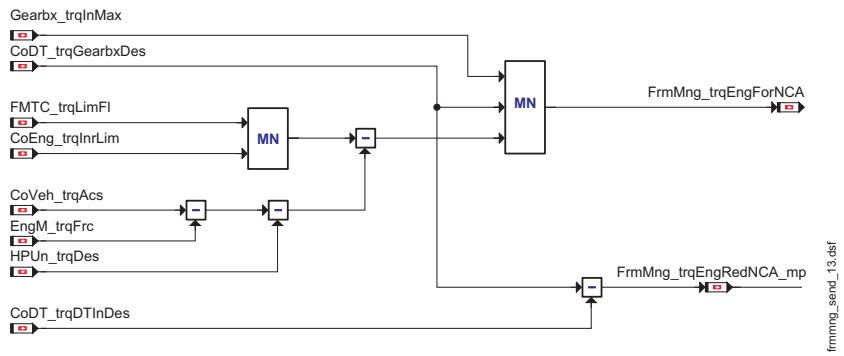


Figure 1008 : Torque calculations for CAN message MOTGEAR



CAN-message MOT1 ID: 361h, Data length: 8 byte, Period: 10ms

Table 369: MOT1 - frame overview

Signal name	Bit	Description
MaxEngineTorqueValidData	63	Validation of MaxEngineTorque.
GasPedalPositionValidData	62	Validation of GasPedalPosition. Fail in gas pedal position measurement/plausibility check.
EngineFrictionTorqueValidData	61	Validation of EngineFrictionTorque.
FeedbackASR/VDCReq	60	CAN communication brake node/ECU fail or ASR1 message counter fail.
EngineTorqueValidData	59	Validation of EngineTorque.
EngineTorqueDriverReqValidData	58	Validation of EngineTorqueDriverReq. Gas pedal position measurement/plausibility check.
TorqueInterventionSts	57-56	0: Torque intervention o.k.. 1: - 2: - 3: Torque intervention will not be completely fulfilled because of: system error or injection shut off.
EngineTorque	55-48	Current indicated engine torque. In case of no engine speed or injection shut-off the engine torque will be zero.
EngineSpeed	47-32	Current engine speed.
EngineTorqueDriverReq	31-24	Indicated torque calculated by maximum selection of accelerator pedal position and cruise control.
EngineFrictionTorque	23-16	Sum of losses due to mechanical friction, gas exchange, oil pump and water pump. Additionally losses due to air condition compressor.
MaxEngineTorque	15-8	Maximum possible indicated engine torque in the current engine state (theoretical value). In cases of no engine speed or injection shut-off MaxEngineTorque will be zero.
GasPedalPosition	7-0	Maximum selection of accelerator pedal position and cruise control.

Figure 1009 : CAN-message MOT1 (1)

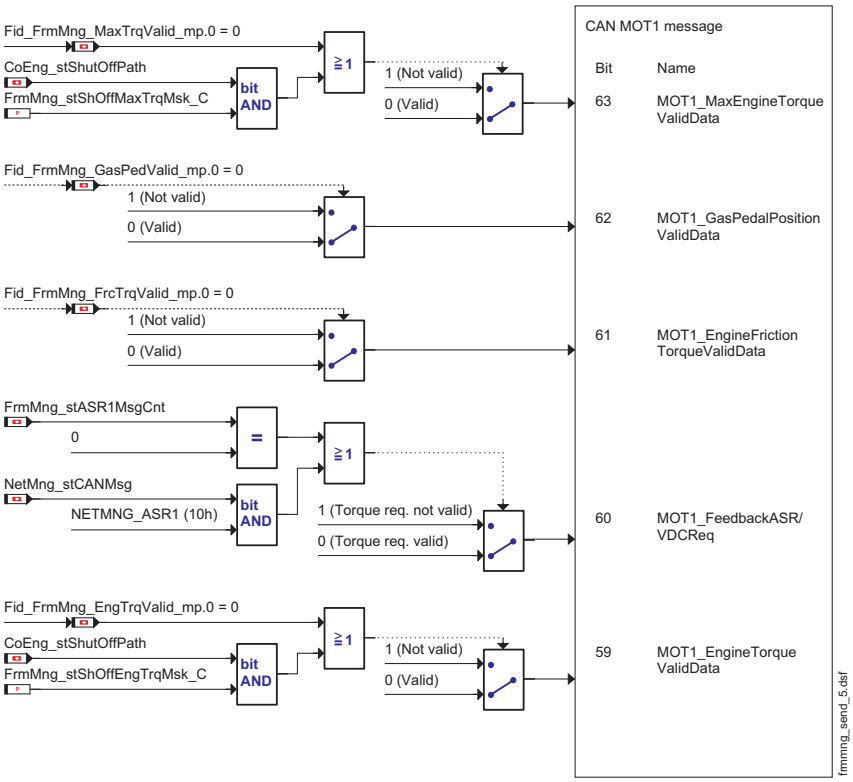


Figure 1010 : CAN-message MOT1 (2)

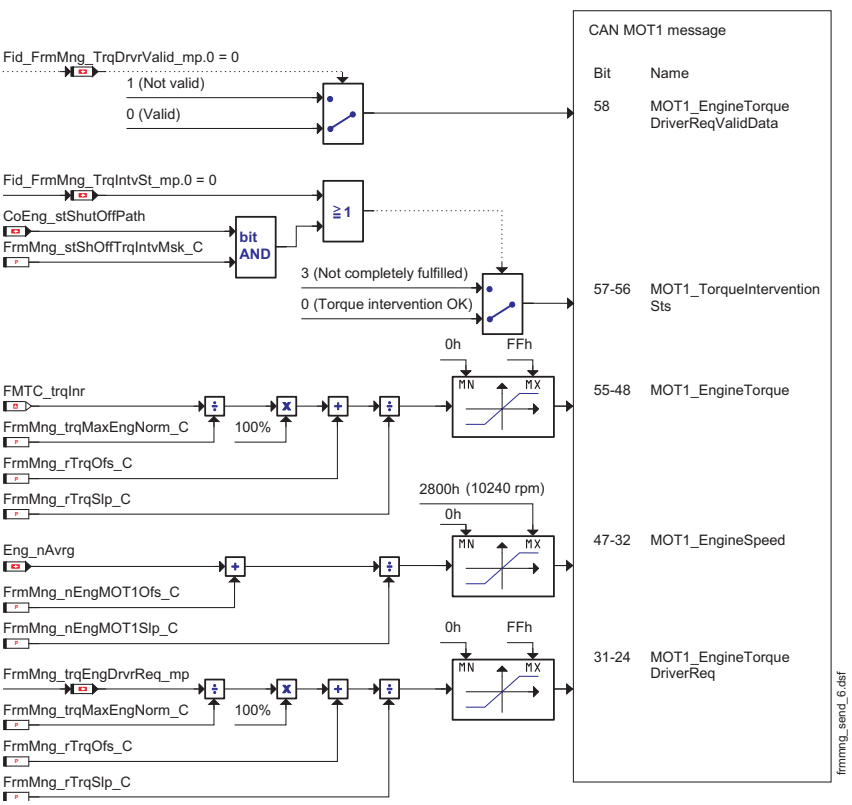


Figure 1011 : CAN-message MOT1 (3)

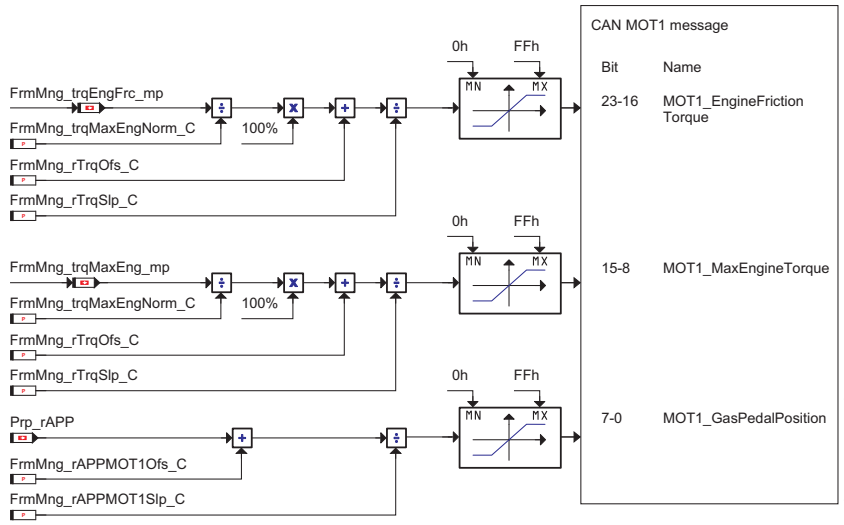
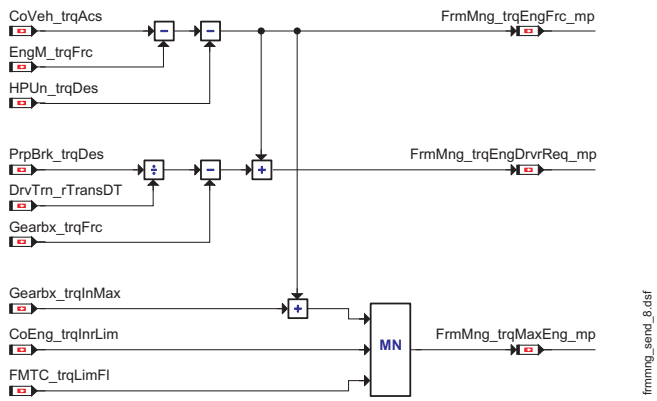


Figure 1012 : Torque calculations for CAN message MOT1



CAN-message MOT2 ID: 3A1h, Data length: 8 byte, Period: 10ms

Table 370: MOT2 - frame overview

Signal name	Bit	Description
CANStand	-	-
MaxEngTorqueNorm	55-48	Normalization torque, indicates the engine in the vehicle.
GasPedalGradientValidData	47	Validation of GasPedalGradient. Failures of gas pedal measurement/plausibility check of cruise control.
ThrottlePositionValidData	46	Should be set to "Not Valid" by application.
StopLightSwitchStsValidData	45	Validation of brake signal.
StopLightSwitchSts	44	Digital input of brake signal.
EngineIntervention	43	Any external torque intervention by brake or gear node possible (no engine speed or injection shut off).
CruiseControlSts	42	Cruise control active.
ECUFail	41	-
Override	40	Gas pedal requested torque is greater than cruise control requested torque.
GasPedalGradient	39-32	Gradient of accelerator pedal position
ThrottlePosition	-	-
ECACCFail	23	Communication between NCM and ACC fails or NCM detects a failure of ACC.
ECACCShutOff	22	Signalize every shut-off condition to the ACC.
ClutchPedalAction	21	Digital input of clutch pedal.
GasPedalAct	20	Gas pedal position above threshold.
EngineTorqueTarget	-	-
MainSwitchACC	11	ACC main switch.
OperationCodeACC	10-8	No ACC Button pressed (MainSwitchACC=1) 0 Set Plus pressed 1 Set Minus pressed 2 Resume pressed 3 Off pressed (MainSwitchACC=0) 4 Failure in ACC-Input detected 7
TypeOfGearbox	7-6	Type of gearbox used.

Figure 1013 : CAN-message MOT2 (1)

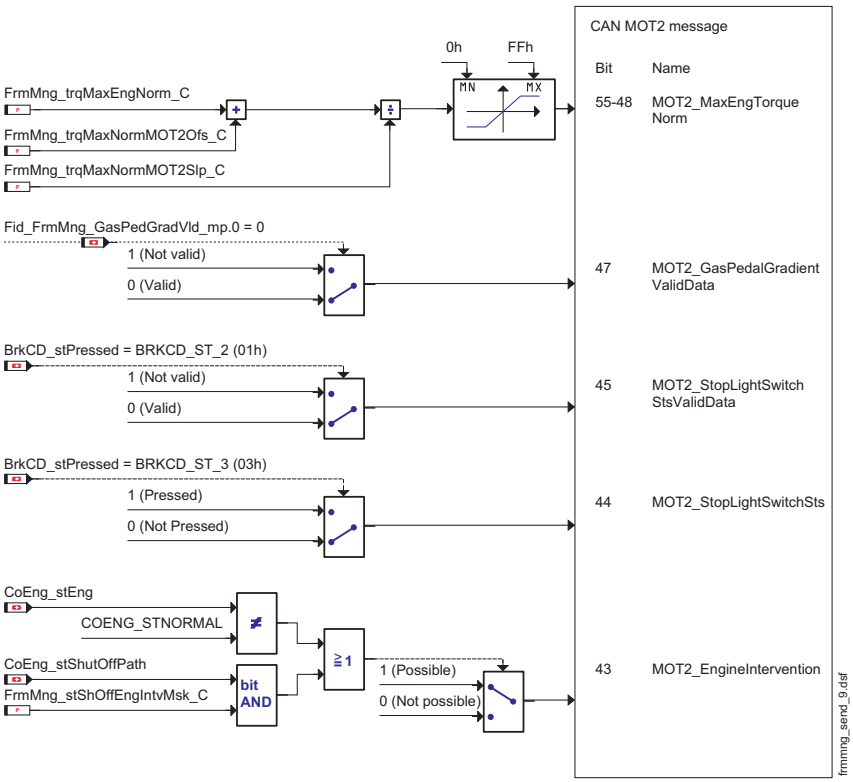


Figure 1014 : CAN-message MOT2 (2)

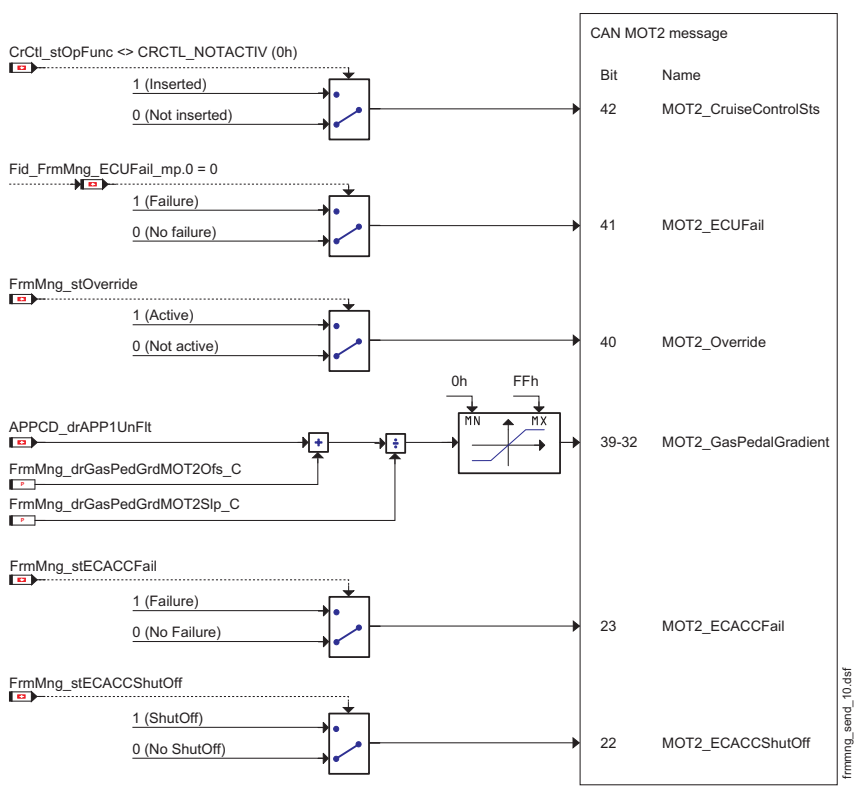


Figure 1015 : CAN-message MOT2 (3)

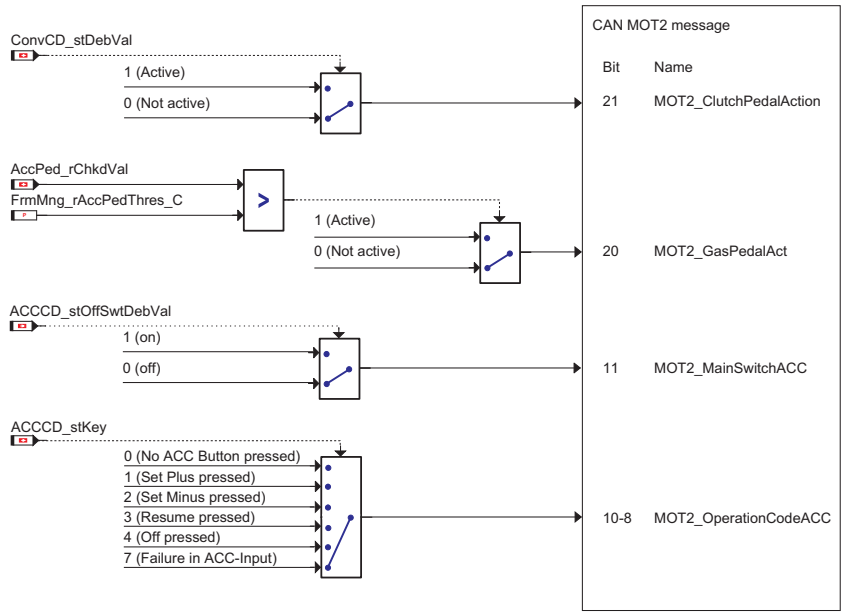
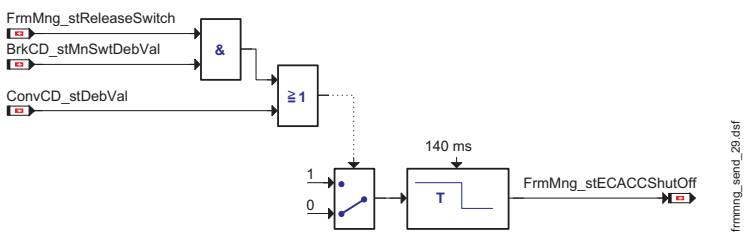


Figure 1016 : Calculation of ECACCShutOff for CAN message MOT2



If the **FrmMng_stECACCShutOff** is set, it will be signalized at least 140 ms duration.

CAN-message STNC (Status C-NCM) ID: 561h, Data length: 8 byte, Period: 50ms

Table 371: Status C-NCM - frame overview

Signal name	Bit	Description
	63-61	Reserved.
D_ES	60-59	CAN node state.
EOL	58	No EOL prog. at FIAT
CurrentFailSts	57	Indicates if any error is present in current driving cycle.
GenericFailSts	56	Indicates if any error is stored in the error memory.
OilPressureSts	55	Bit is set in case of low pressure measured by digital input or sensor fault. It is also set during oil pressure indication lamp in diagnose mode.
OilPressureFailSts	54	Oil pressure plausibility check with the engine speed has failed.
FuelWaterPresentSts	53	Digital input. It indicates water in fuel filter. It is also set during water-in-fuel lamp in diagnose mode.
GlowPlugLampSts	52	Power stage of the glow indication lamp is energized for a minimum time after key15 on or during pre-glow. It is also set during glow indication lamp in diagnose mode.
GlowPlugFailSts	51	Indicates glow system errors.
FuelWaterPresentFailSts	50	Plausibility check after key15 on fails.
CruiseControlLampSts	49	Power stage of the cruise control lamp. It is also set during cruise control lamp in diagnose mode.
EngineSpeedThreshold	-	-
CompressorSts	47-46	Power stage of the air condition compressor and air condition compressor enable state.
EMSFailSts	44-43	Power stage of the diagnosis lamp. The immobilizer in recovery mode occupies diagnosis lamp for blink code.
EngineWaterTempFailSts	42	Fail in the engine water temperature measurement/ plausibility check.
VehicleSpeedPlausibilityChk	41	Fail in the vehicle speed measurement/plausibility check.
EngineWaterTempWarningLightSts	40	Power stage of the engine water temperature warning lamp. It is also set during engine water temperature warning lamp in diagnose mode.
EngineWaterTemp	39-32	Analog signal input engine water temperature.
FuelConsumption	31-16	Current fuel consumption.
EngineSpeed	15-8	Engine speed.

Signal name	Bit	Description
FuelConsumptionValidData	6	Fuel consumption calculation not reliable because of injector failures or shut-off requests.
EngineSpeedValidData	5	Fail in the engine speed measurement.

Figure 1017 : CAN-message STNC(1)

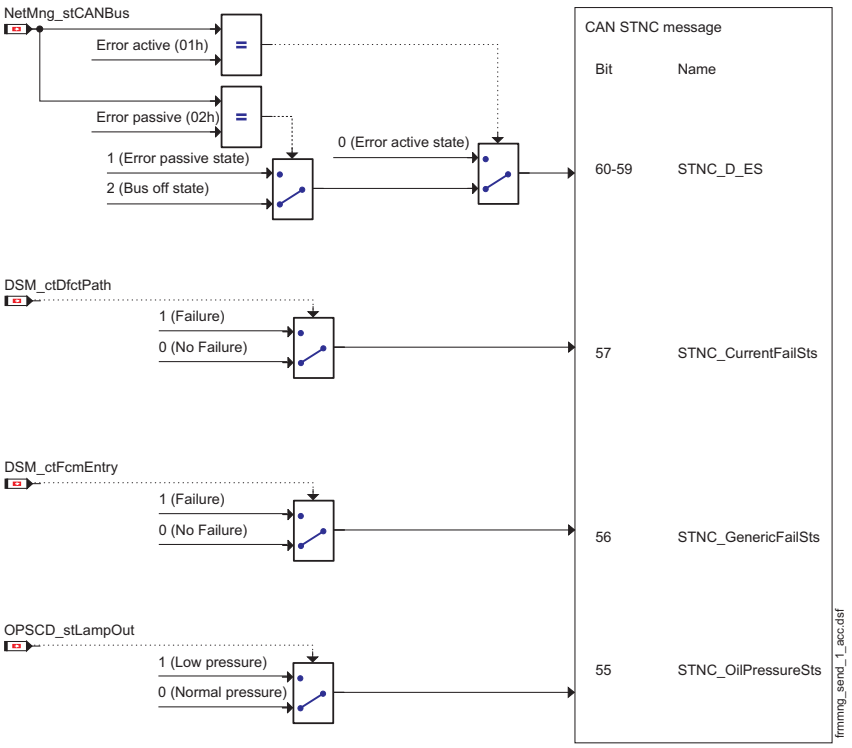


Figure 1018 : CAN-message STNC(2)

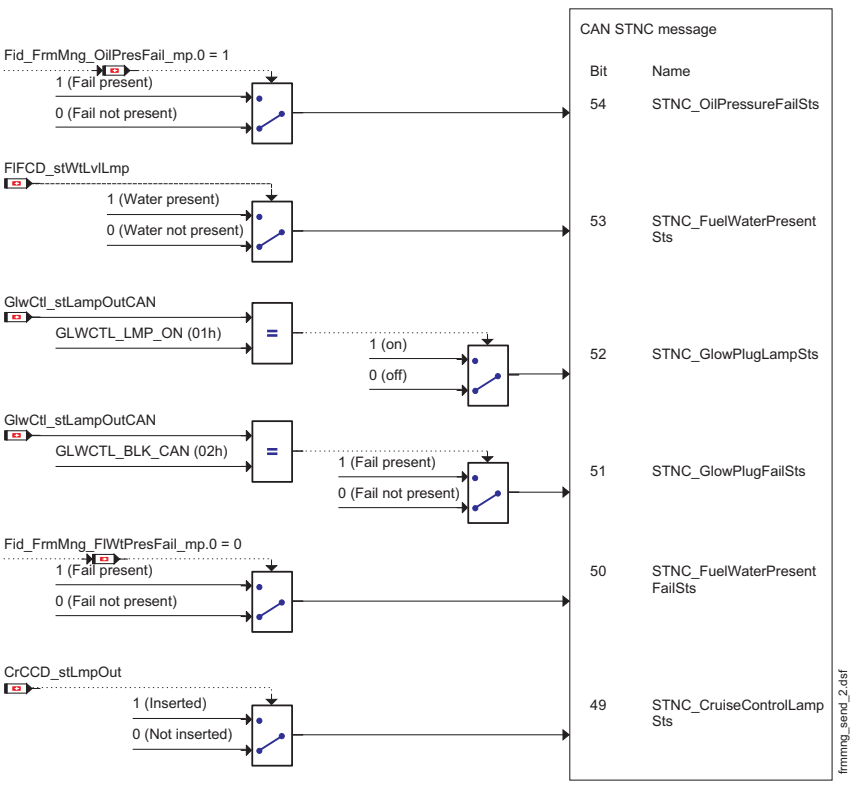


Figure 1019 : CAN-message STNC(3)

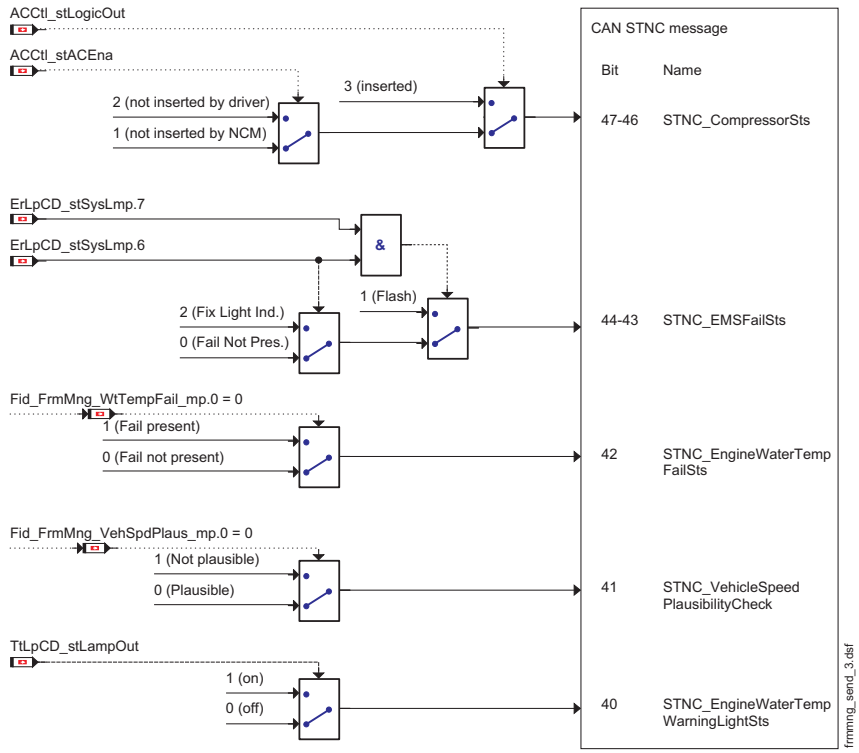
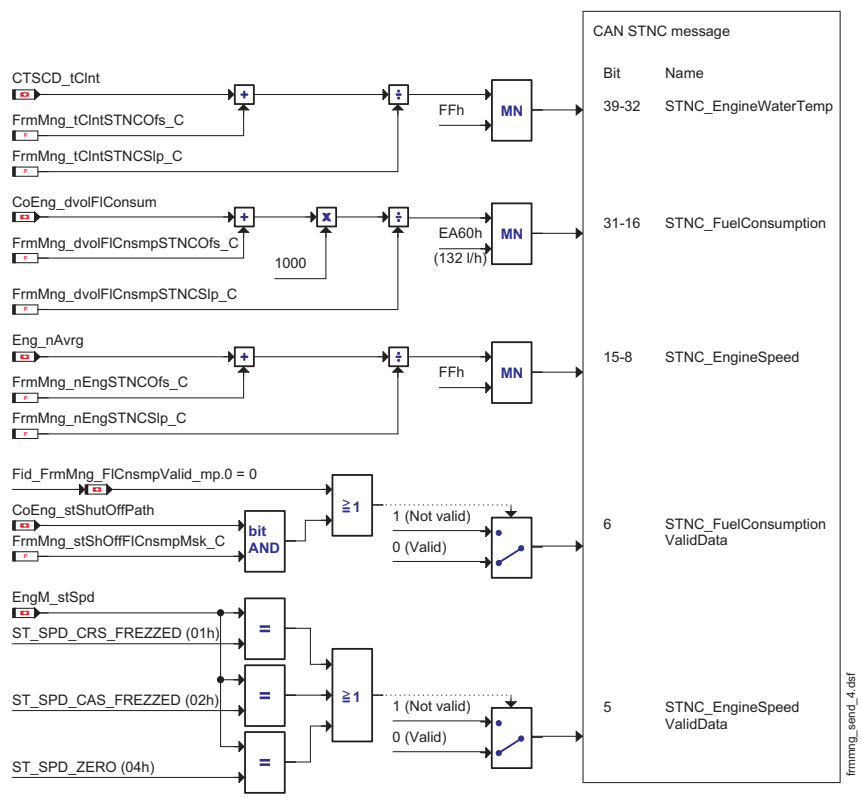


Figure 1020 : CAN-message STNC(4)



CAN-message MOT1 for CAN 4.11 ID: 316h, Data length: 8 byte, Period: 10ms

Table 372: MOT1 - frame overview

Signal name	Bit	Description
S_KL15	0	Status of K15 (K15 OFF=0; K15 ON=1)
N_MOT	16-31	Engine speed.
NF_MD0	40-47	Engine friction torque. Sum of losses due to mechanical friction, accelerator exchange, oil pump and water pump (function of Eng_nAvrg and CTSCD_tClnt) Additional losses due to air condition compressor.
MD_MOT	48-55	Crankshaft torque without external gearbox interventions (theoretical value).
MD_RED	56-63	Reduction of engine torque due to external gearbox intervention.

Figure 1021 : CAN-message MOT1 (1)

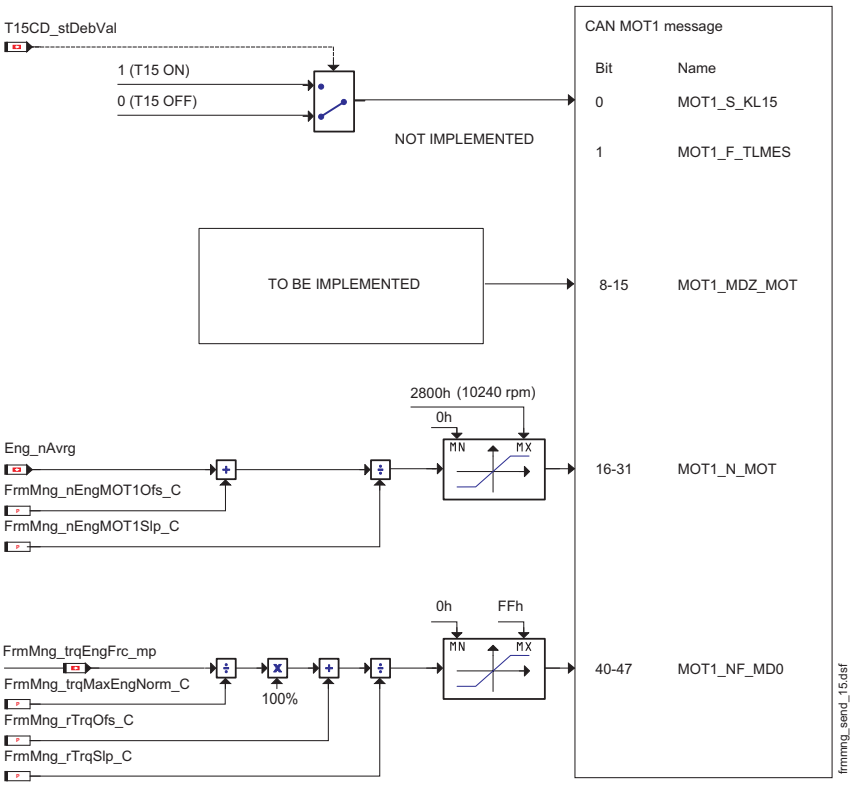


Figure 1022 : CAN-message MOT1 (2)

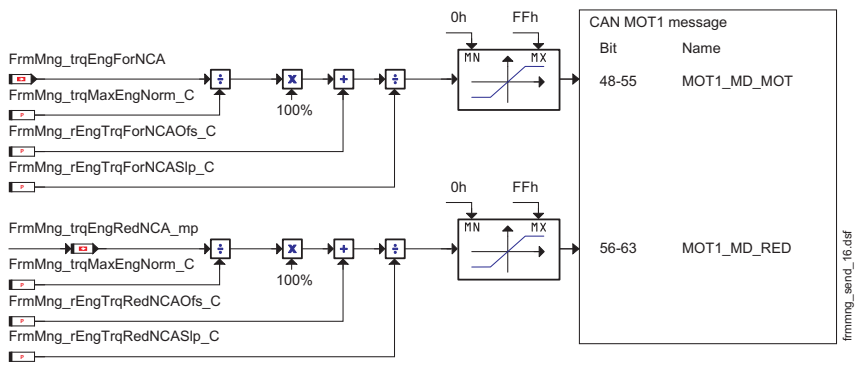
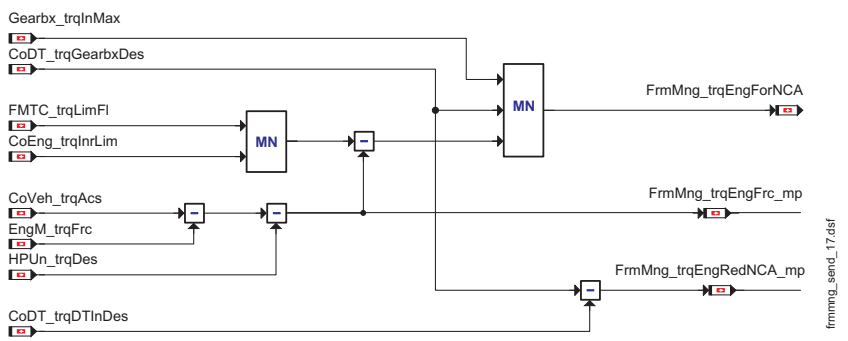


Figure 1023 : Torque calculations for CAN message MOT1



CAN-message MOT2 for CAN 4.11 ID: 329h, Data length: 8 byte, Period: 10ms

Table 373: MOT2 - frame overview

Signal name	Bit	Description
MUL_CODE	7-8	Multiplexer position of MUL_INFO. 00 CAN_STAND 01 MDNORM 10 NODEF (not defined) 11 VAR_COD
MUL_INFO	0-6	CAN_STAND Version of CAN message map MDNORM Normalization torque NODEF - VAR_CODE Gearbox type
T_MOT	8-15	Analog signal input engine water temperature.
N_MOT8	48-55	Engine speed.

Signal name	Bit	Description
U_BATT	56-63	Battery voltage.

Figure 1024 : CAN-message MOT2 (1)

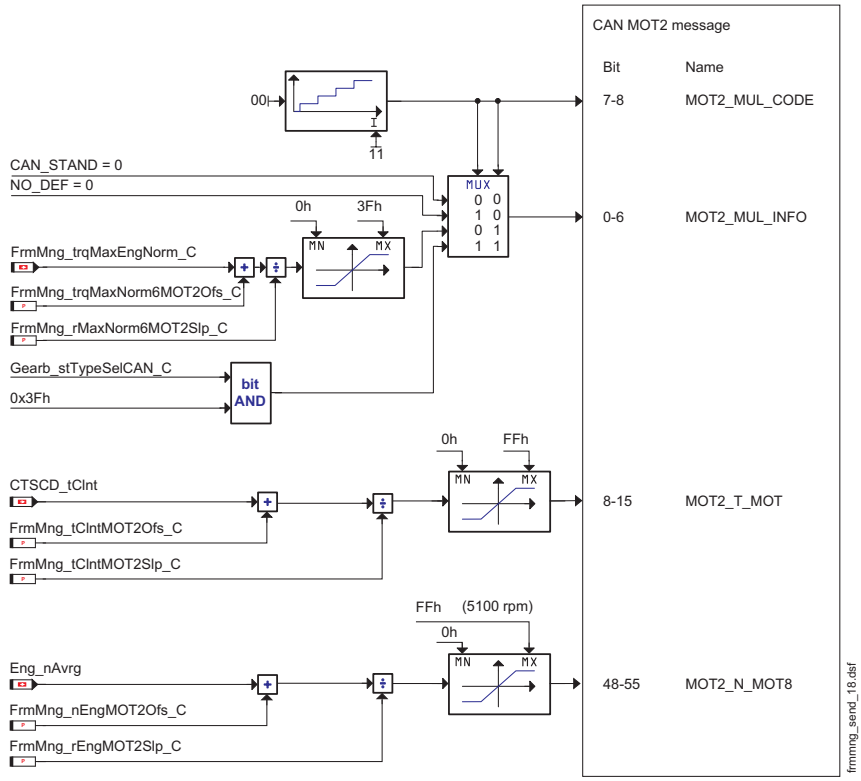
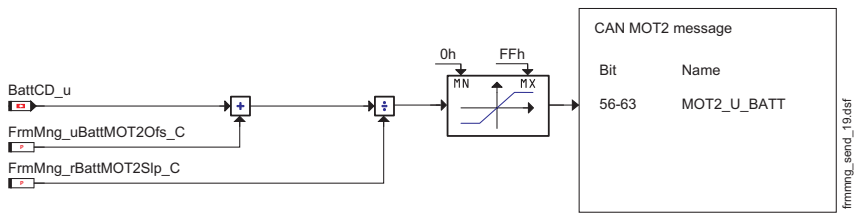


Figure 1025 : CAN-message MOT2 (2)



CAN-message MOT3 for CAN 4.11 ID: 235h, Data length: 6 byte, Period: 10ms

Table 374: MOT3 - frame overview

Signal name	Bit	Description
S_KOREL	7	Power stage of air condition compressor.
S_LL	6	Engine in low idle mode.
B_NFB	4	Limp home mode.
B_FGR	3	Cruise control active.
F_SWPED	2	Validation of GasPedalPosition. Fail in gas pedal position measurement / plausibility check.
NF_SOLL	8-15	Current indicated engine torque. In case of no engine speed or injection shut-off the engine torque will be zero.
NF_FAHRER	16-23	Indicated torque calculated by maximum selection of the accelerator pedal position and cruise control.
SW_PED	40-47	Maximum selection of accelerator pedal position and cruise control.

Figure 1026 : CAN-message MOT3 (1)

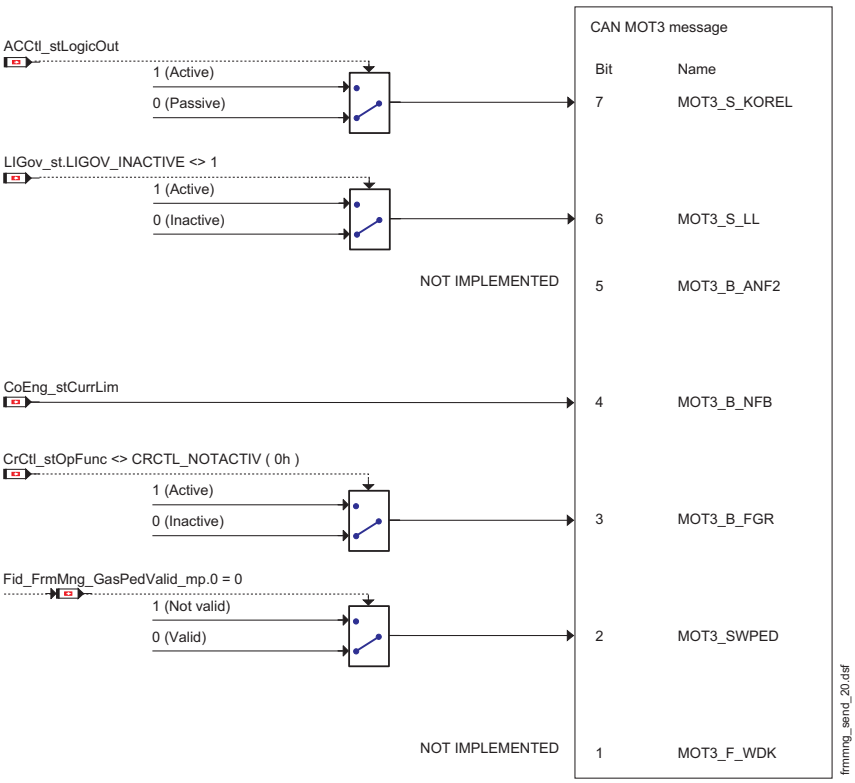


Figure 1027 : CAN-message MOT3 (2)

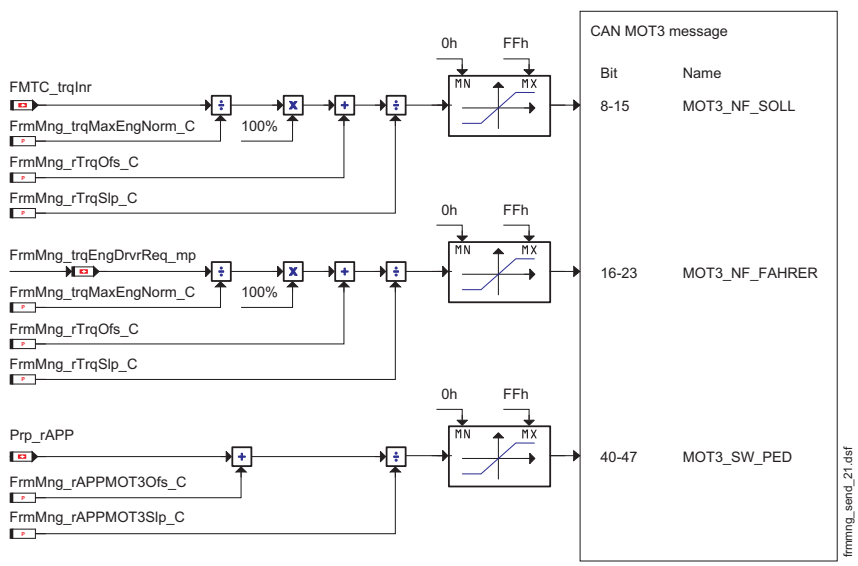
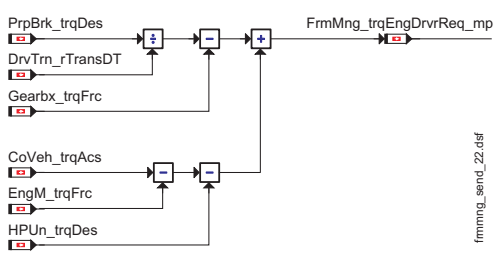


Figure 1028 : Torque calculations for CAN message MOT3



ECU initialization

During initialization phase the CAN messages are written with their default values.

Output values

```
FrmMng_trqEngForNCA: Engine torque for gearbox node (without gearbox intervention) [Nm]
<SWORD> --->
```

Input values

ACC_stECACCFail: Communication status between ECU and ACC.
ACC_stECACCShutOff: Signalize every shut-off condition to the ACC.
ACCCD_stKey: demanded status of adaptive cruise control actuating device [-] <UBYTE> --->
ACCCD_stOffSwtDebVal: Dedounced value from ACC ON/OFF-switch.
AccPed_rChkdVal: AccPed ratio after security check [%] <SWORD> --->
ACCtl_stACEna: air conditioning compressor enable status [-] <UBYTE> --->
ACCtl_stLogicOut: AC compressor control output [-] <UBYTE> --->
APPCD_drAPP1UnFlt: difference of accelerator pedal 1 position unfiltered value [%/s] <SWORD> --->
BattCD_u: battery voltage [mV] <SWORD> --->
BrkCD_stPressed: Brake pressed state [-] <UBYTE> --->
CoDT_trqDTInDes: desired value for drivetrain input (clutch) [Nm] <SWORD> --->
CoDT_trqGearbxDes: desired gearbox input torque (from driver) [Nm] <SWORD> --->
CoEng_dvolFlConsum: fuel consumption [l/h] <SWORD> --->
CoEng_stEng: current engine state [-] <UBYTE> --->
CoEng_stShutOffPath: active shut-off paths resulting from active reversible, irreversible, and afterrun shut-off paths [-] <ULONG> --->
CoEng_trqInrLim: limitation torque (inner engine torque) [Nm] <SWORD> --->
CoVeh_trqAcs: Torque requirement of the accessories [Nm] <SWORD> --->
CrCCD_stLmpOut: lamp out signal for CAN [-] <UBYTE> --->
CrCtl_stOpFunc: state of cruise control operation functionality [-] <UBYTE> --->
CTSCD_tClnt: coolant temperature [deg C] <SWORD> --->
DrvTrn_rTransDT: drivetrain transmission (wheel torque per engine torque) [-] <SWORD> --->
DSM_ctDfctPath: Number of actual defect fault paths [-] <UWORD> --->
DSM_ctFcmEntry: Number of used fault code entries [-] <UBYTE> --->
Eng_nAvrg: average engine speed [rpm] <SWORD> --->
EngM_stSpd: state of engine speed [-] <UBYTE> --->
EngM_trqFrc: current friction torque [Nm] <SWORD> --->
ErLpCD_stSysLmp: Gloabal status of the System Lamp and status of activation sources [-] <UBYTE> --->
FlFCD_stWtLvlLmp: logical status of fuel filter water level warning lamp [-] <UBYTE> --->
FMTC_trqInr: current inner engine torque [Nm] <SWORD> --->
FMTC_trqLimFl: Current fuel quantity limiting torque [Nm] <SWORD> --->
FrmMng_stASR1MsgCnt: ASR1 message counter status [-] <UBYTE> --->
FrmMng_stECACCFail: ECU detects ACC frame errors [-] <UBYTE> --->
FrmMng_stECACCShutOff: Shut off conditions for the ACC [-] <UBYTE> --->
Gearbx_trqFrc: current frictional torque of gearbox [Nm] <SWORD> --->
Gearbx_trqInMax: maximum input torque of gearbox [Nm] <SWORD> --->
GlwCtl_stLampOutCAN: state request of glowlamp for CAN [-] <UBYTE> --->
HPUn_trqDes: torque demand of high pressure pump [Nm] <SWORD> --->
NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->
OPSCD_stLampOut: status request for oilpressure lamp [-] <UBYTE> --->
Prp_rAPP: summerized accelerator pedal position from APP and virtual APP [%] <SWORD> --->
PrpBrk_trqDes: x-movement torque demand [Nm] <SWORD> --->
TtLpCD_stLampOut: state of temperature warning lamp [-] <UBYTE> --->

Measuring points

Fid_FrmMng_ECUFail_mp: function identifier for ECU fail status [-] <UBYTE> --->
Fid_FrmMng_EngTrqNCAVld_mp: function identifier for EngineTorqueForNCAValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_EngTrqValid_mp: function identifier for EngineTorqueValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_FlCnsmpValid_mp: function identifier for fuel consumption validity [-] <UBYTE> --->
Fid_FrmMng_FlWtPresFail_mp: function identifier for fuel water present fail status [-] <UBYTE> --->
Fid_FrmMng_FrcTrqValid_mp: function identifier for EngineFrictionTorqueValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_GasPedGradVld_mp: function identifier for GasPedalGradientValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_GasPedValid_mp: function identifier for GasPedalPositionValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_MaxTrqValid_mp: function identifier for MaxEngineTorqueValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_OilPresFail_mp: function identifier for oil pressure fail [-] <UBYTE> --->
Fid_FrmMng_TrqDrvrValid_mp: function identifier for EngineTorqueDriverReqValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_TrqIntvSt_mp: function identifier for TorqueInterventionSts CAN signal [-] <UBYTE> --->
Fid_FrmMng_TrqRedNCAVld_mp: function identifier for EngineTorqueReductNCAValidData CAN signal [-] <UBYTE> --->
Fid_FrmMng_VehSpdPlaus_mp: function identifier for vehicle speed plausibility check fail [-] <UBYTE> --->
Fid_FrmMng_WtTempFail_mp: function identifier for engine water temperature fail status [-] <UBYTE> --->
Fid_FrmMng_WtTempNCAFail_mp: function identifier for EngineWaterTempForNCAFailSts CAN

signal [-] <UBYTE> --->
FrmMng_stMOGERaw0_mp: CAN message object MOTGEAR byte 0, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw1_mp: CAN message object MOTGEAR byte 1, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw2_mp: CAN message object MOTGEAR byte 2, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw3_mp: CAN message object MOTGEAR byte 3, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw4_mp: CAN message object MOTGEAR byte 4, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw5_mp: CAN message object MOTGEAR byte 5, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw6_mp: CAN message object MOTGEAR byte 6, raw value [-] <UBYTE> --->
FrmMng_stMOGERaw7_mp: CAN message object MOTGEAR byte 7, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw0_mp: CAN message object MOT1 byte 0, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw1_mp: CAN message object MOT1 byte 1, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw2_mp: CAN message object MOT1 byte 2, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw3_mp: CAN message object MOT1 byte 3, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw4_mp: CAN message object MOT1 byte 4, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw5_mp: CAN message object MOT1 byte 5, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw6_mp: CAN message object MOT1 byte 6, raw value [-] <UBYTE> --->
FrmMng_stMOT1Raw7_mp: CAN message object MOT1 byte 7, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw0_mp: CAN message object MOT2 byte 0, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw1_mp: CAN message object MOT2 byte 1, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw2_mp: CAN message object MOT2 byte 2, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw3_mp: CAN message object MOT2 byte 3, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw4_mp: CAN message object MOT2 byte 4, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw5_mp: CAN message object MOT2 byte 5, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw6_mp: CAN message object MOT2 byte 6, raw value [-] <UBYTE> --->
FrmMng_stMOT2Raw7_mp: CAN message object MOT2 byte 7, raw value [-] <UBYTE> --->
FrmMng_stMOT3Raw0_mp: Raw value of CAN message object MOT3 byte 0 [-] <UBYTE> --->
FrmMng_stMOT3Raw1_mp: Raw value of CAN message object MOT3 byte 1 [-] <UBYTE> --->
FrmMng_stMOT3Raw2_mp: Raw value of CAN message object MOT3 byte 2 [-] <UBYTE> --->
FrmMng_stMOT3Raw3_mp: Raw value of CAN message object MOT3 byte 3 [-] <UBYTE> --->
FrmMng_stMOT3Raw4_mp: Raw value of CAN message object MOT3 byte 4 [-] <UBYTE> --->
FrmMng_stMOT3Raw5_mp: Raw value of CAN message object MOT3 byte 5 [-] <UBYTE> --->
FrmMng_stMOT3Raw6_mp: Raw value of CAN message object MOT3 byte 6 [-] <UBYTE> --->
FrmMng_stMOT3Raw7_mp: Raw value of CAN message object MOT3 byte 7 [-] <UBYTE> --->
FrmMng_stSTNCRaw0_mp: CAN message object Status-C-CAN byte 0, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw1_mp: CAN message object Status-C-CAN byte 1, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw2_mp: CAN message object Status-C-CAN byte 2, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw3_mp: CAN message object Status-C-CAN byte 3, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw4_mp: CAN message object Status-C-CAN byte 4, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw5_mp: CAN message object Status-C-CAN byte 5, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw6_mp: CAN message object Status-C-CAN byte 6, raw value [-] <UBYTE> --->
FrmMng_stSTNCRaw7_mp: CAN message object Status-C-CAN byte 7, raw value [-] <UBYTE> --->
FrmMng_trqEngDrvrReq_mp: Engine torque driver request [Nm] <SWORD> --->
FrmMng_trqEngFrc_mp: Engine friction torque (including HPUn and accessories) [Nm] <SWORD> --->
FrmMng_trqEngRedNCA_mp: Engine torque reduction due to gearbox request [Nm] <SWORD> --->
FrmMng_trqMaxEng_mp: Maximum engine torque [Nm] <SWORD> --->

Application parameters

FrmMng_drGasPedGrdMOT2Ofs_C: CAN message object MOT2, conversion gas pedal gradient, offset<value> --->
FrmMng_drGasPedGrdMOT2Slp_C: CAN message object MOT2, conversion gas pedal gradient, slope<value> --->
FrmMng_dvolFlCnsmpSTNCOfs_C: Offset of conversion of fuel consumption to CAN format.<value> --->
FrmMng_dvolFlCnsmpSTNCSlp_C: Slope of conversion of fuel consumption to CAN format.<value> --->
FrmMng_nEngMOT1Ofs_C: CAN message object MOT1, conversion engine speed, offset<value> --->
FrmMng_nEngMOT1Slp_C: CAN message object MOT1, conversion engine speed, slope<value> --->
FrmMng_nEngSTNCOfs_C: CAN message object Status-C-CAN, conversion engine speed, offset<value> --->
FrmMng_nEngSTNCSlp_C: CAN message object Status-C-CAN, conversion engine speed, slope<value> --->
FrmMng_rAccPedThres_C: CAN message object MOT2, AccPed threshold for GasPedalAct signal<value> --->
FrmMng_rAPPMOT1Ofs_C: CAN message object MOT1, conversion gas pedal position, offset<value> --->
FrmMng_rAPPMOT1Slp_C: CAN message object MOT1, conversion gas pedal position, slope<value> --->
FrmMng_rEngTrqForNCAOfs_C: CAN message object MOTGEAR, conversion engine torque for gearbox node, offset<value> --->
FrmMng_rEngTrqForNCASlp_C: CAN message object MOTGEAR, conversion engine torque for gearbox node, slope<value> --->
FrmMng_rEngTrqRedNCAOfs_C: CAN message object MOTGEAR, conversion engine torque reduction by gearbox node, offset<value> --->
FrmMng_rEngTrqRedNCASlp_C: CAN message object MOTGEAR, conversion engine torque reduction

by gearbox node, slope<value> --->
FrmMng_rTrqOfs_C: CAN conversion of ratio of torque, offset<value> --->
FrmMng_rTrqSlp_C: CAN conversion of ratio of torque, slope<value> --->
FrmMng_stMOGEDfl0_C: CAN message object MOTGEAR byte 0, default value<value> --->
FrmMng_stMOGEDfl1_C: CAN message object MOTGEAR byte 1, default value<value> --->
FrmMng_stMOGEDfl2_C: CAN message object MOTGEAR byte 2, default value<value> --->
FrmMng_stMOGEDfl3_C: CAN message object MOTGEAR byte 3, default value<value> --->
FrmMng_stMOGEDfl4_C: CAN message object MOTGEAR byte 4, default value<value> --->
FrmMng_stMOGEDfl5_C: CAN message object MOTGEAR byte 5, default value<value> --->
FrmMng_stMOGEDfl6_C: CAN message object MOTGEAR byte 6, default value<value> --->
FrmMng_stMOGEDfl7_C: CAN message object MOTGEAR byte 7, default value<value> --->
FrmMng_stMOGEEna0_C: CAN message object MOTGEAR byte 0, enable mask<value> --->
FrmMng_stMOGEEna1_C: CAN message object MOTGEAR byte 1, enable mask<value> --->
FrmMng_stMOGEEna2_C: CAN message object MOTGEAR byte 2, enable mask<value> --->
FrmMng_stMOGEEna3_C: CAN message object MOTGEAR byte 3, enable mask<value> --->
FrmMng_stMOGEEna4_C: CAN message object MOTGEAR byte 4, enable mask<value> --->
FrmMng_stMOGEEna5_C: CAN message object MOTGEAR byte 5, enable mask<value> --->
FrmMng_stMOGEEna6_C: CAN message object MOTGEAR byte 6, enable mask<value> --->
FrmMng_stMOGEEna7_C: CAN message object MOTGEAR byte 7, enable mask<value> --->
FrmMng_stMOT1Dfl0_C: CAN message object MOT1 byte 0, default value<value> --->
FrmMng_stMOT1Dfl1_C: CAN message object MOT1 byte 1, default value<value> --->
FrmMng_stMOT1Dfl2_C: CAN message object MOT1 byte 2, default value<value> --->
FrmMng_stMOT1Dfl3_C: CAN message object MOT1 byte 3, default value<value> --->
FrmMng_stMOT1Dfl4_C: CAN message object MOT1 byte 4, default value<value> --->
FrmMng_stMOT1Dfl5_C: CAN message object MOT1 byte 5, default value<value> --->
FrmMng_stMOT1Dfl6_C: CAN message object MOT1 byte 6, default value<value> --->
FrmMng_stMOT1Dfl7_C: CAN message object MOT1 byte 7, default value<value> --->
FrmMng_stMOT1Ena0_C: CAN message object MOT1 byte 0, enable mask<value> --->
FrmMng_stMOT1Ena1_C: CAN message object MOT1 byte 1, enable mask<value> --->
FrmMng_stMOT1Ena2_C: CAN message object MOT1 byte 2, enable mask<value> --->
FrmMng_stMOT1Ena3_C: CAN message object MOT1 byte 3, enable mask<value> --->
FrmMng_stMOT1Ena4_C: CAN message object MOT1 byte 4, enable mask<value> --->
FrmMng_stMOT1Ena5_C: CAN message object MOT1 byte 5, enable mask<value> --->
FrmMng_stMOT1Ena6_C: CAN message object MOT1 byte 6, enable mask<value> --->
FrmMng_stMOT1Ena7_C: CAN message object MOT1 byte 7, enable mask<value> --->
FrmMng_stMOT2Dfl0_C: CAN message object MOT2 byte 0, default value<value> --->
FrmMng_stMOT2Dfl1_C: CAN message object MOT2 byte 1, default value<value> --->
FrmMng_stMOT2Dfl2_C: CAN message object MOT2 byte 2, default value<value> --->
FrmMng_stMOT2Dfl3_C: CAN message object MOT2 byte 3, default value<value> --->
FrmMng_stMOT2Dfl4_C: CAN message object MOT2 byte 4, default value<value> --->
FrmMng_stMOT2Dfl5_C: CAN message object MOT2 byte 5, default value<value> --->
FrmMng_stMOT2Dfl6_C: CAN message object MOT2 byte 6, default value<value> --->
FrmMng_stMOT2Dfl7_C: CAN message object MOT2 byte 7, default value<value> --->
FrmMng_stMOT2Ena0_C: CAN message object MOT2 byte 0, enable mask<value> --->
FrmMng_stMOT2Ena1_C: CAN message object MOT2 byte 1, enable mask<value> --->
FrmMng_stMOT2Ena2_C: CAN message object MOT2 byte 2, enable mask<value> --->
FrmMng_stMOT2Ena3_C: CAN message object MOT2 byte 3, enable mask<value> --->
FrmMng_stMOT2Ena4_C: CAN message object MOT2 byte 4, enable mask<value> --->
FrmMng_stMOT2Ena5_C: CAN message object MOT2 byte 5, enable mask<value> --->
FrmMng_stMOT2Ena6_C: CAN message object MOT2 byte 6, enable mask<value> --->
FrmMng_stMOT2Ena7_C: CAN message object MOT2 byte 7, enable mask<value> --->
FrmMng_stMOT3Dfl0_C: Default value of CAN message object MOT3 byte 0<value> --->
FrmMng_stMOT3Dfl1_C: Default value of CAN message object MOT3 byte 1<value> --->
FrmMng_stMOT3Dfl2_C: Default value of CAN message object MOT3 byte 2<value> --->
FrmMng_stMOT3Dfl3_C: Default value of CAN message object MOT3 byte 3<value> --->
FrmMng_stMOT3Dfl4_C: Default value of CAN message object MOT3 byte 4<value> --->
FrmMng_stMOT3Dfl5_C: Default value of CAN message object MOT3 byte 5<value> --->
FrmMng_stMOT3Dfl6_C: Default value of CAN message object MOT3 byte 6<value> --->
FrmMng_stMOT3Dfl7_C: Default value of CAN message object MOT3 byte 7<value> --->
FrmMng_stMOT3Ena0_C: Enable mask of CAN message object MOT3 byte 0<value> --->
FrmMng_stMOT3Ena1_C: Enable mask of CAN message object MOT3 byte 1<value> --->
FrmMng_stMOT3Ena2_C: Enable mask of CAN message object MOT3 byte 2<value> --->
FrmMng_stMOT3Ena3_C: Enable mask of CAN message object MOT3 byte 3<value> --->
FrmMng_stMOT3Ena4_C: Enable mask of CAN message object MOT3 byte 4<value> --->
FrmMng_stMOT3Ena5_C: Enable mask of CAN message object MOT3 byte 5<value> --->
FrmMng_stMOT3Ena6_C: Enable mask of CAN message object MOT3 byte 6<value> --->
FrmMng_stMOT3Ena7_C: Enable mask of CAN message object MOT3 byte 7<value> --->
FrmMng_stShOffEngIntvMsk_C: Bit mask to set EngineIntervention CAN signal to Not possible by shut-off paths<value> --->
FrmMng_stShOffEngTrqMsk_C: Bit mask for deactivating EngineTorqueValidData CAN signal by shut-off paths<value> --->
FrmMng_stShOffFlCnsmpMsk_C: Bit mask for deactivating FuelConsumptionValid CAN signal by shut-off paths<value> --->
FrmMng_stShOffMaxTrqMsk_C: Bit mask for deactivating MaxEngineTorqueValidData CAN signal by shut-off paths<value> --->
FrmMng_stShOffTrqIntvMsk_C: Bit mask to set TorqueInterventionSts CAN signal to Not completely fulfilled by shut-off paths<value> --->
FrmMng_stShOffTrqNCAMsk_C: Bit mask for deactivating EngineTorqueForNCAValidData CAN

signal by shut-off paths<value> --->
FrmMng_stShOffTrqRedMsk_C: Bit mask for deactivating EngineTorqueReductNCAValidData CAN signal by shut-off paths<value> --->
FrmMng_stSTNCDfl0_C: CAN message object Status-C-CAN byte 0, default value<value> --->
FrmMng_stSTNCDfl1_C: CAN message object Status-C-CAN byte 1, default value<value> --->
FrmMng_stSTNCDfl2_C: CAN message object Status-C-CAN byte 2, default value<value> --->
FrmMng_stSTNCDfl3_C: CAN message object Status-C-CAN byte 3, default value<value> --->
FrmMng_stSTNCDfl4_C: CAN message object Status-C-CAN byte 4, default value<value> --->
FrmMng_stSTNCDfl5_C: CAN message object Status-C-CAN byte 5, default value<value> --->
FrmMng_stSTNCDfl6_C: CAN message object Status-C-CAN byte 6, default value<value> --->
FrmMng_stSTNCDfl7_C: CAN message object Status-C-CAN byte 7, default value<value> --->
FrmMng_stSTNCEna0_C: CAN message object Status-C-CAN byte 0, enable mask<value> --->
FrmMng_stSTNCEna1_C: CAN message object Status-C-CAN byte 1, enable mask<value> --->
FrmMng_stSTNCEna2_C: CAN message object Status-C-CAN byte 2, enable mask<value> --->
FrmMng_stSTNCEna3_C: CAN message object Status-C-CAN byte 3, enable mask<value> --->
FrmMng_stSTNCEna4_C: CAN message object Status-C-CAN byte 4, enable mask<value> --->
FrmMng_stSTNCEna5_C: CAN message object Status-C-CAN byte 5, enable mask<value> --->
FrmMng_stSTNCEna6_C: CAN message object Status-C-CAN byte 6, enable mask<value> --->
FrmMng_stSTNCEna7_C: CAN message object Status-C-CAN byte 7, enable mask<value> --->
FrmMng_tClntMOGEOfs_C: CAN message object MOTGEAR, conversion coolant temperature, offset<value> --->
FrmMng_tClntMOGESlp_C: CAN message object MOTGEAR, conversion coolant temperature, slope<value> --->
FrmMng_tClntSTNCOfs_C: CAN message object Status-C-CAN, conversion coolant temperature, offset<value> --->
FrmMng_tClntSTNCSlp_C: CAN message object Status-C-CAN, conversion coolant temperature, slope<value> --->
FrmMng_trqMaxEngNorm_C: Maximum engine torque for normalization<value> --->
FrmMng_trqMaxNormMOT2Ofs_C: CAN message object MOT2, conversion maximum engine torque for normalization, offset<value> --->
FrmMng_trqMaxNormMOT2Slp_C: CAN message object MOT2, conversion maximum engine torque for normalization, slope<value> --->
FrmMng_uBattMOT2Ofs_C: Offset of MOT2 battery voltage UBATT<value> --->
FrmMng_uBattMOT2Slp_C: Slope of MOT2 battery voltage UBATT<value> --->

Receive messages from brake (FrmMng_RecBrk)

Functional overview

Functional description

In this chapter there is the description of CAN messages which are received from brake node.

Message STATUS C-NFR

ID: 566h, Data Length: 8 byte, Period: 100ms

The message STATUS C-NFR is not supported.

Message ASR0

ID: 046h, Data Length: 8 byte, Period: 5ms

The message ASR0 is not supported.

Message ASR1

ID: 146h, Data Length: 8 byte, Period: 10ms

Table 375: ASR1 - frame overview

Signal name	Bit	Description
TorqueReductASRSlowValidData	63	-
TorqueIncreaseMSRValidData	62	Validation of TorqueIncreaseMSR
TorqueReductASRFastValidData	61	Validation of TorqueReductASRFast
BrakeInterventionSts	60	Brake intervention is active, affect brake state messages
BrakePedalSts	59	Brake pedal switch, affect brake state messages
BrakePedalValidData	58	Validation of BrakePedalSts, affect brake state messages
TorqueReductASRSlow	55-48	MSRActive=0: - MSRActive=1: ASR1 MessageCounter
CANStand	47-40	-
TorqueIncreaseMSR	39-32	MSRActive=0: ASR1 MessageCounter MSRActive=1: Absolute Indicated Torque intervention/ increase
TorqueReductASRFast	31-24	MSRActive=0: Absolute Indicated Torque intervention/reduct MSRActive=1: Bitwise negation of ASR1 MessageCounter+TorqueIncreaseMSR
VehicleSpeedVSOSig	23-11	Vehicle speed
VehicleSpeedVSOSigFailSts	10	Failure of signal VehicleSpeedVSOSig
ABS/ASR/VDCSts	9	-
ASRFailSts	8	-
ABSFailSts	7	-
EBDFailSts	6	-
ABSActive	5	Affect brake state
MSRActive	3	Enable torque intervention/increase, affect brake state messages
ASRActive	2	Enable torque intervention/reduct, affect brake state messages
VDCFailSts	1	-
VDCActive	0	Affect brake state messages

Evaluation of ASR1 Message Counter

To ensure that the signals are not affected by corrupted CAN messages, the ASR1 Message Counter is implemented in the CAN message ASR1. The 4 bit ASR1 Message Counter is incremented by 1 each 10ms.

The ASR1 Message Counter is received at various positions of CAN message ASR1, see table below.

Table 376: Position of ASR1 Message Counter

Status	Position of ASR1 Message Conter in CAN Message ASR1
DCS active (MSRActive=1)	ASR1Byte1 = ASR1 Message Counter ASR1Byte4 = Complement (ASR1 Message Counter + TorqueIncreaseMSR)
TCS active (ASRActive=1)	ASR1Byte3 = ASR1 Message Counter
DCS and TCS not active	ASR1Byte3 = ASR1 Message Counter

To check the continuous incrementation of the ASR1 Message Counter the actual received value **FrmMng_ctASR1MsgCnt0_mp** is compared with the value received three times before **FrmMng_ctASR1MsgCnt3_mp**. The incrementation is correct if the difference is between the two callibatable values **FrmMng_ctASR1MsgCntMin_C** and **FrmMng_ctASR1MsgCntMax_C**. The schedule time of this process is 10ms.

Equation 48: ASR1 Message Counter check

$$ASR1MsgCntMin \le ASR1MsgCnt(n) - ASR1MsgCnt(n-3) \le ASR1MsgCntMax$$

During DCS is active (MSRActive=1) an additionally Complement check is done concerning the ASR1 Message Counter and the Torque request TorqueIncreaseMSR. The complement in this case is a bitwise negation.

Equation 49: ASR1 Complement check (if DCS active)

$$ASR1Byte4 = CMPL(ASR1MsgCnt(n) + TorqueIncreaseMSR)$$

The ASR1 Message Counter check and the Complement check in case of MSRActive=1 is an input of an additional debouncing counter **FrmMng_ctASR1MsgCntOff_mp**. The debouncing counter has got a calibratable increment **FrmMng_ctASR1MsgCntOffInc_C** and decrement **FrmMng_ctASR1MsgCntOffDec_C**. The status **FrmMng_stASR1MsgCnt** is set if the debouncing counter **FrmMng_ctASR1MsgCntOff_mp** is under the limit **FrmMng_ctASR1MsgCntOffMax_C**.

Figure 1029 : ASR1 Message Counter

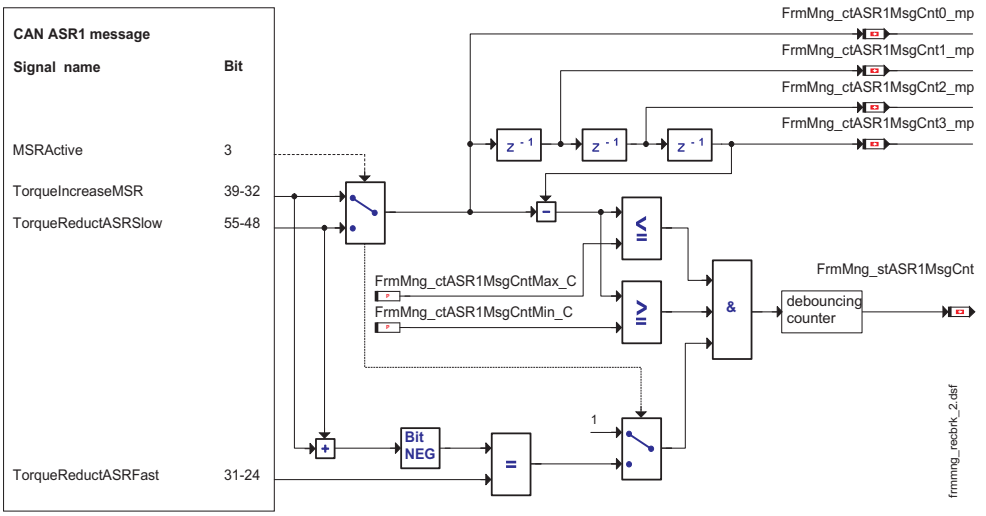
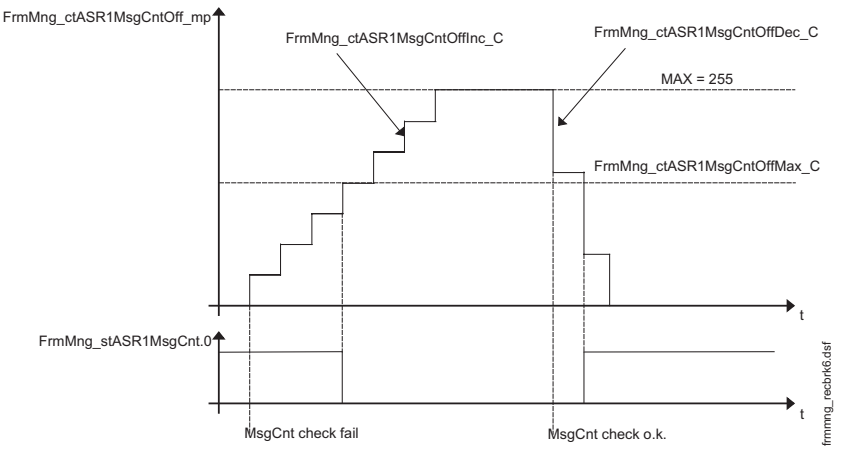


Figure 1030 : ASR1 Message Counter, debouncing

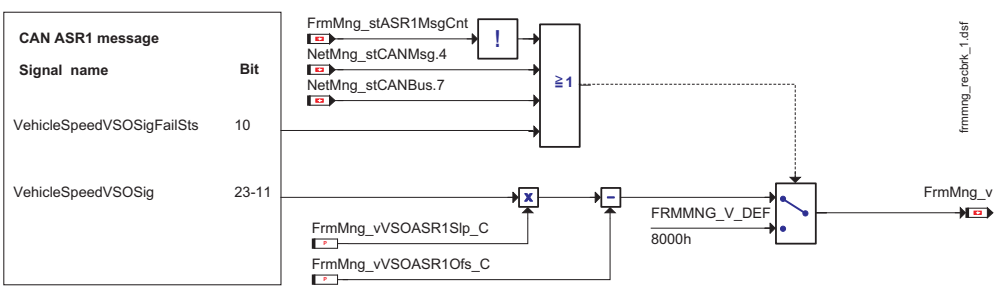


Evaluation of Vehicle Speed

The received CAN vehicle speed is converted by offset **FrmMng_vVSOASR1Ofs_C** and slope **FrmMng_vVSOASR1Slp_C** and is then sent in the message **FrmMng_v**.

If any CAN message error or CAN bus error is present the default value is sent in **FrmMng_v**.

Figure 1031 : Vehicle Speed



Evaluation of Traction Control (TCS/ASR)

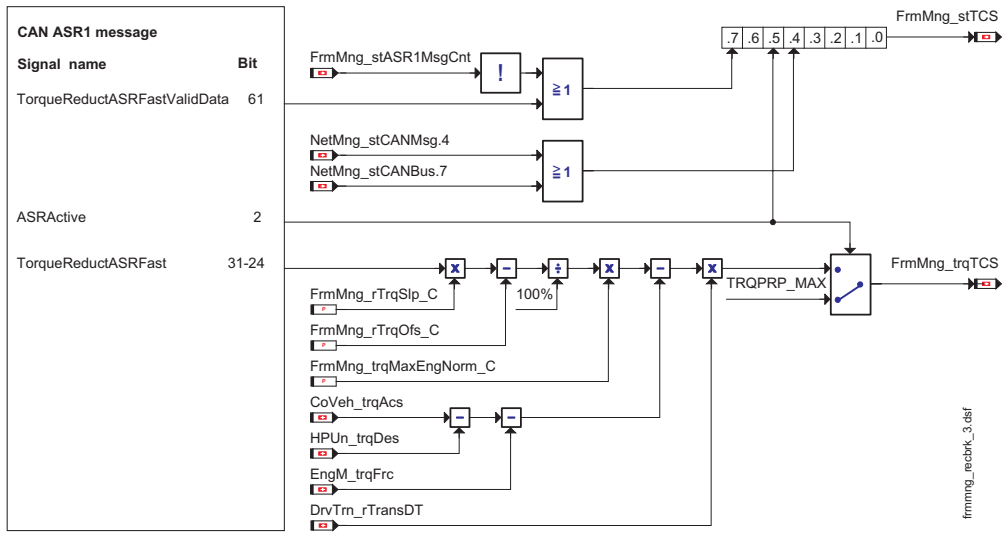
The received torque decrease (traction control torque) is converted by offset **FrmMng_rTrqOfs_C** and slope **FrmMng_rTrqSlp_C**. The measurement point **FrmMng_rTCSRaw_mp** displays the received ratio of torque. Additionally the renormalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done and then the torque losses are subtracted. The TCS torque increase demand is sent in the message **FrmMng_trqTCS**.

If any CAN message error or CAN bus error is present the torque decrease state **FrmMng_stTCS** will be set accordingly.

Table 377: Traction Control, state

FrmMng_stTCS	
Bit 2, FRMMNG_TCS_NORAMP	Shut off TCS intervention without ramp
Bit 4, FRMMNG_TCS_CANERR	TCS affected due to CAN errors
Bit 5, FRMMNG_TCS_DEM	TCS intervention is active
Bit 7, FRMMNG_TCS_ERR	TCS affected due to CAN message errors

Figure 1032 : Traction Control



Evaluation of Drag Torque Control (DCS/MSR)

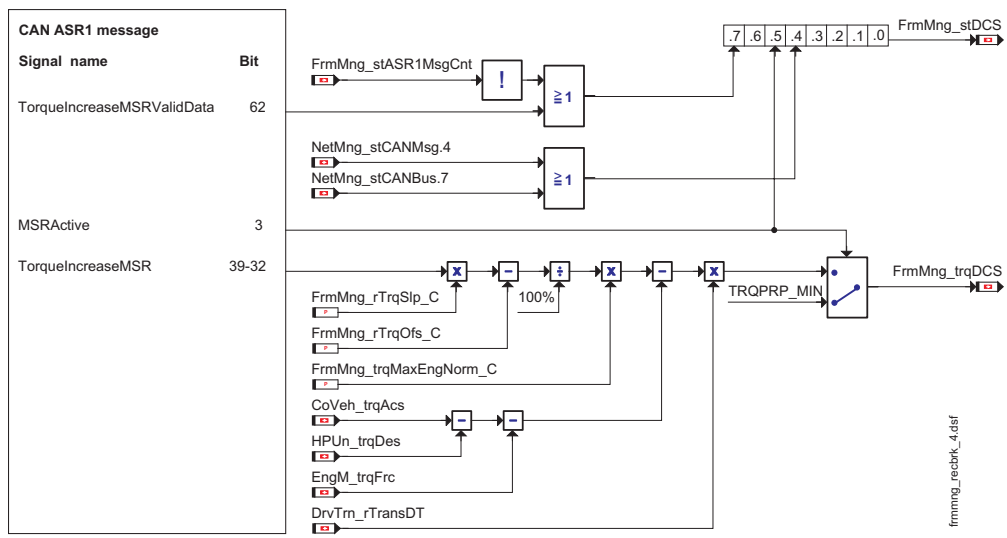
The received torque increase (drag torque control) is converted by offset **FrmMng_rTrqOfs_C** and slope **FrmMng_rTrqSlp_C**. The measurement point **FrmMng_rDCSRaw_mp** displays the received ratio of torque. Additionally the renormalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done and then the torque losses are subtracted. The TCS torque increase demand is sent in the message **FrmMng_trqDCS**.

If any CAN message error or CAN bus error is present the torque increase state **FrmMng_stDCS** will be set accordingly.

Table 378: Drag Torque Control, state

FrmMng_stTCS	
Bit 2, FRMMNG_DCS_NORAMP	Shut off DCS intervention without ramp
Bit 4, FRMMNG_DCS_CANERR	DCS affected due to CAN errors
Bit 5, FRMMNG_DCS_DEM	DCS intervention is active
Bit 7, FRMMNG_DCS_ERR	DCS affected due to CAN message errors

Figure 1033 : Drag Torque Control



Evaluation of Brake Status

The evaluation of the brake status is divided in two parts: Brake pedal switch information **FrmMng_stBrk** and the brake intervention information for vehicle propulsion (e.g. cruise control) **FrmMng_stBrkPrp**. With the bit mask **FrmMng_stBrkMsk_C** single CAN signals can be chosen to take affect on **FrmMng_stBrk**. With the bit mask **FrmMng_stBrkPrpMsk_C** single CAN signals can be chosen to take affect on **FrmMng_stBrkPrp**.

If any CAN message error or CAN bus error is present the default value is sent in **FrmMng_stBrk** and **FrmMng_stBrkPrp**.

Table 379: Bit definition in **FrmMng_stBrk**

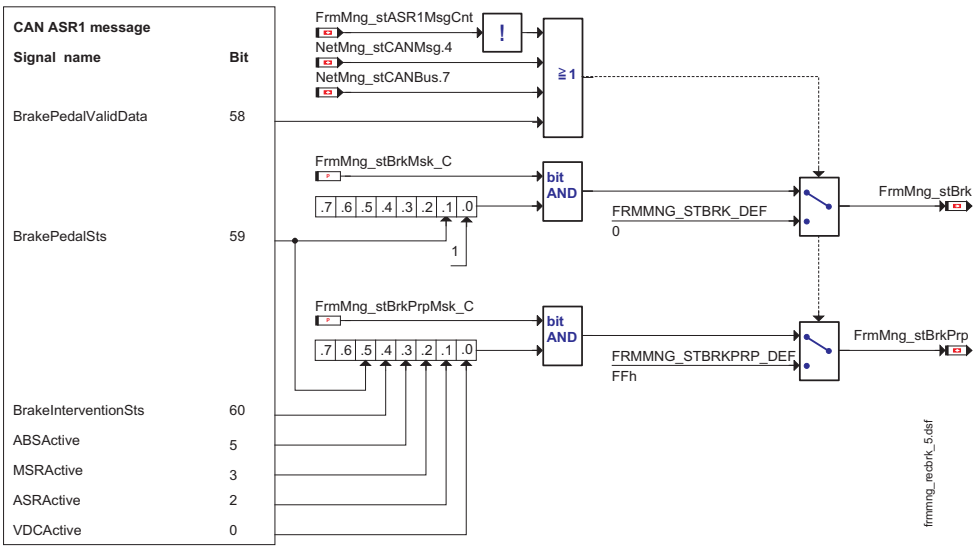
Bit position	
Bit 0, FRMMNG_STBRK_CAN	Enable input of brake pedal via CAN
Bit 1, FRMMNG_STBRK_VAL	Brake switch signal
Bit 2, FRMMNG_STBRK_REDCAN	Enable input of redundant brake switch via CAN
Bit 3, FRMMNG_STBRK_REDVAL	Redundant brake switch signal

Table 380: Bit definition in **FrmMng_stBrkPrp**

Bit position	
Bit 0, FRMMNG_STBRKPRP_VDC	Vehicle Dynamic Control (VDC) is active (VDCActive=1)
Bit 1, FRMMNG_STBRKPRP_TCS	TCS is active (ASRActive=1)

Bit position	
Bit 2, FRMMNG_STBRKPRP_DCS	DCS is active (MSRActive=1)
Bit 3, FRMMNG_STBRKPRP_ABS	ABS is active (ABSActive=1)
Bit 4, FRMMNG_STBRKPRP_INT	BrakeInterventionSts=1
Bit 5, FRMMNG_STBRKPRP_BRK	BrakePedalSts=1

Figure 1034 : Brake Intervention



Message ASR2 ID: 166h, Data Length: 8 byte, Period: 10ms

The message ASR2 is not supported.

Message ASR1 for CAN 4.11 ID: 153h, Data Length: 8 byte, Period: 10ms. This can be selected using the switch **FrmMng_swtCANProtocol_C**.

Table 381: ASR1 - frame overview

Signal name	Bit	Description
B_MSR	1	Enable torque intervention / increase, affect brake status messages.
F_V1	10	Failure of signal V1.
V1	11-15 16-23	Vehicle speed.
NF_ASR	24-31	B_MSR = 0: Absolute indicated torque intervention/ reduct. Plaus. with bitwise negation of NF_MSR B_MSR = 1: -
NF_MSR	32-39	B_MSR = 0: - B_MSR = 1: Absolute Indicated Torque interven-tion/increase. Plaus. with bitwise negation of NF_ASR
MSG_ASR	56-63	ASR1 MessageCounter.

Evaluation of ASR1 Message Counter

To ensure that the signals are not affected by corrupted CAN messages, the ASR1 Message Counter is implemented in the CAN message ASR1. The 8 bit ASR1 Message Counter is incremented by 1 each 10ms.

The ASR1 Message Counter is received at bit positions 7-0 (MSG_ASR) of CAN message ASR1.

To check the continuous incrementation of the ASR1 Message Counter the actual received value **FrmMng_ctASR1MsgCnt0_mp** is compared with the value received three times before **FrmMng_ctASR1MsgCnt3_mp**. The incrementation is correct if the difference is between the two callibratable values **FrmMng_ctASR1MsgCntMin_C** and **FrmMng_ctASR1MsgCntMax_C**. The schedule time of this process is 10ms.

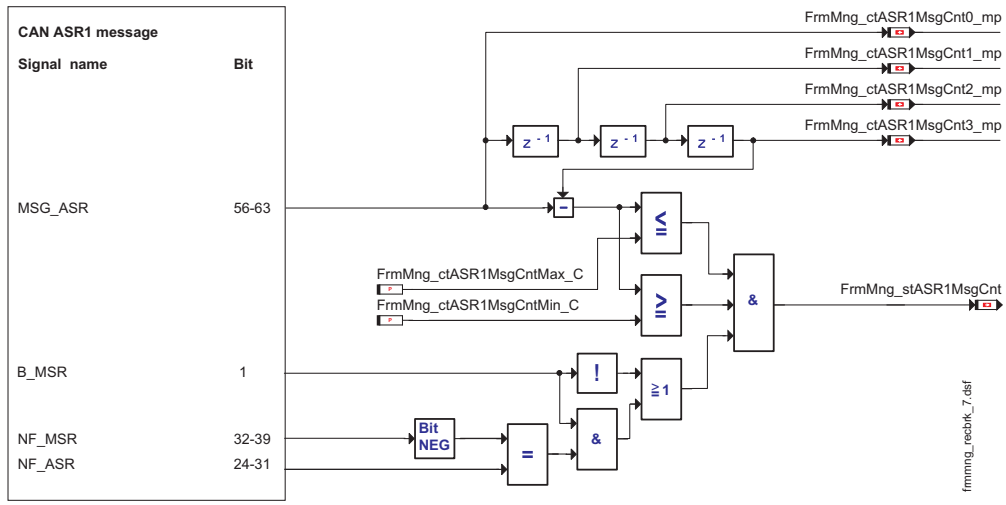
Equation 50: ASR1 Message Counter check

$$ASR1MsgCntMin \le ASR1MsgCnt(n) - ASR1MsgCnt(n-3) \le ASR1MsgCntMax$$

An additionally Complement check is done concerning the Torque request NF_MSR. The complement in this case is a bitwise negation (NF_ASR = CMPL(NF_MSR)).

The ASR1 Message Counter check and the Complement check in case of MSRActive=1 is an input of an additional debouncing counter **FrmMng_ctASR1MsgCntOff_mp**. The debouncing counter has got a callibratable increment **FrmMng_ctASR1MsgCntOffInc_C** and decrement **FrmMng_ctASR1MsgCntOffDec_C**. The status **FrmMng_stASR1MsgCnt** is set if the debouncing counter **FrmMng_ctASR1MsgCntOff_mp** is under the limit **FrmMng_ctASR1MsgCntOffMax_C**.

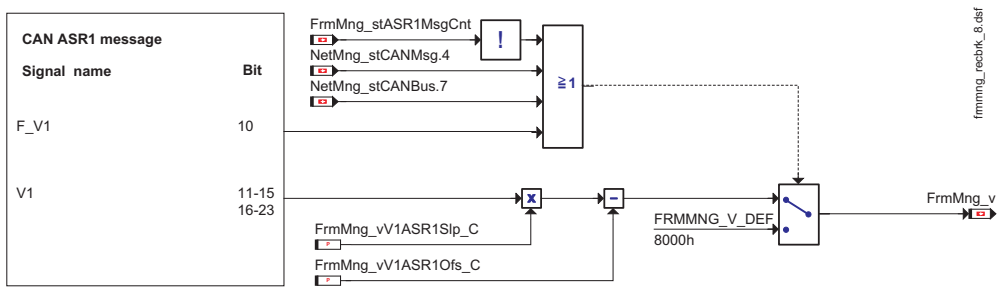
Figure 1035 : ASR1 Message Counter



Evaluation of Vehicle Speed The received CAN vehicle speed is converted by offset **FrmMng_vV1ASR1Ofs_C** and slope **FrmMng_vV1ASR1Slp_C** and is then sent in the message **FrmMng_v**.

If any CAN message error or CAN bus error is present the default value is sent in **FrmMng_v**.

Figure 1036 : Vehicle Speed



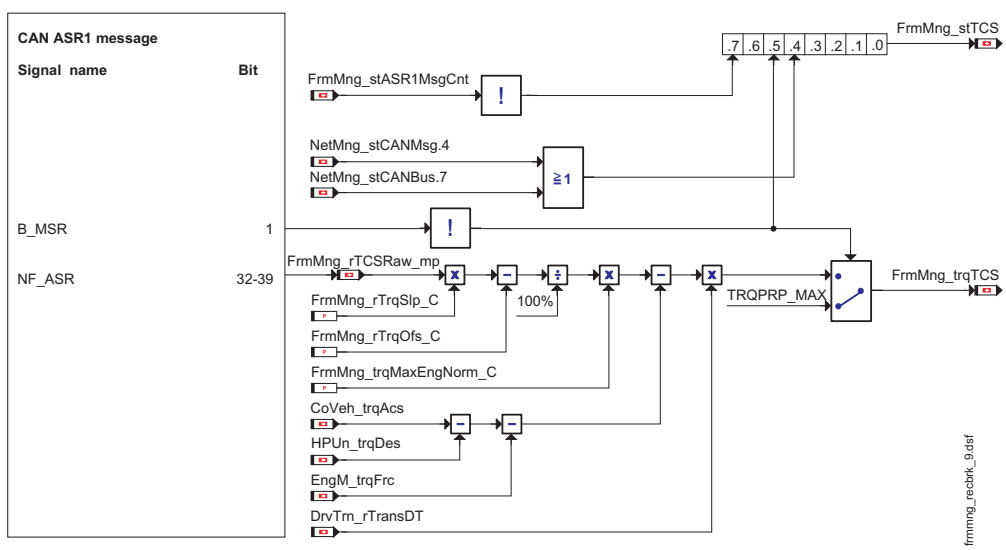
Evaluation of Traction Control (TCS/ASR) The received torque decrease (traction control torque) is converted by offset **FrmMng_rTrqOfs_C** and slope **FrmMng_rTrqSlp_C**. The measurement point **FrmMng_rTCSRaw_mp** displays the received ratio of torque. Additionally the re-normalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done and then the torque losses are subtracted. The TCS torque increase demand is sent in the message **FrmMng_trqTCS**.

If any CAN message error or CAN bus error is present the torque decrease state **FrmMng_stTCS** will be set accordingly.

Table 382: Traction Control, state

FrmMng_stTCS	
Bit 2, FRMMNG_TCS_NORAMP	not used
Bit 4, FRMMNG_TCS_CANERR	TCS affected due to CAN errors
Bit 5, FRMMNG_TCS_DEM	TCS intervention is active
Bit 7, FRMMNG_TCS_ERR	TCS affected due to CAN message errors

Figure 1037 : Traction Control



Evaluation of Drag Torque Control (DCS/MSR)

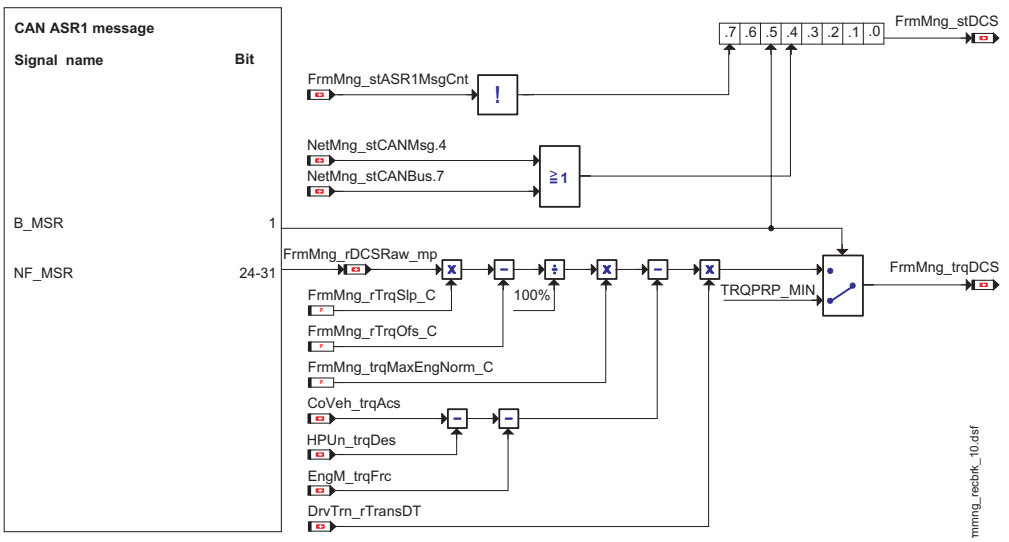
The received torque increase (drag torque control) is converted by offset **FrmMng_rTrqOfs_C** and slope **FrmMng_rTrqSlp_C**. The measurement point **FrmMng_rDCSRaw_mp** displays the received ratio of torque. Additionally the re-normalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done and then the torque losses are subtracted. The TCS torque increase demand is sent in the message **FrmMng_trqDCS**.

If any CAN message error or CAN bus error is present the torque increase state **FrmMng_stDCS** will be set accordingly.

Table 383: Drag Torque Control, state

FrmMng_stTCS	
Bit 2, FRMMNG_DCS_NORAMP	not used
Bit 4, FRMMNG_DCS_CANERR	DCS affected due to CAN errors
Bit 5, FRMMNG_DCS_DEM	DCS intervention is active
Bit 7, FRMMNG_DCS_ERR	DCS affected due to CAN message errors

Figure 1038 : Drag Torque Control



Message GEM0 ID: 43Fh, Data Length: 8 byte, Period: 15ms

The message GEM0 is only received.

ECU initialization

During initialization phase the ID, the Data Length and the Period are set according to the specification. The data bytes are taken from the default data set.

Output values

FrmMng_stASR1MsgCnt: ASR1 message counter status [-] <UBYTE> --->
FrmMng_stBrk: brake switch status from CAN [-] <UBYTE> --->
FrmMng_stBrkPrp: state of cruise control over CAN [-] <UBYTE> --->
FrmMng_stDCS: state of MSR - torque demand [-] <UBYTE> --->
FrmMng_stTCS: state of ASR - torque demand [-] <UBYTE> --->
FrmMng_trqDCS: Drag-torque Control System (DCS) gearbox output torque demand [Nm] <SWORD> --->
FrmMng_trqTCS: Tracktion Control System (TCS) gearbox output torque demand [Nm] <SWORD> --->
FrmMng_v: vehicle speed via CAN [km/h] <SWORD> --->

Input values

CoVeh_trqAcs: Torque requirement of the accessories [Nm] <SWORD> --->
DrvTrn_rTransDT: drivetrain transmission (wheel torque per engine torque) [-] <SWORD> --->
EngM_trqFrc: current friction torque [Nm] <SWORD> --->
HPUn_trqDes: torque demand of high pressure pump [Nm] <SWORD> --->
NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->

Measuring points

FrmMng_ctASR1MsgCnt0_mp: ASR1 message counter (n) [-] <UBYTE> --->
FrmMng_ctASR1MsgCnt1_mp: ASR1 message counter (n-1) [-] <UBYTE> --->
FrmMng_ctASR1MsgCnt2_mp: ASR1 message counter (n-2) [-] <UBYTE> --->
FrmMng_ctASR1MsgCnt3_mp: ASR1 message counter (n-3) [-] <UBYTE> --->
FrmMng_ctASR1MsgCntOff_mp: ASR1 message counter , debouncing counter [-] <UBYTE> --->
FrmMng_rDCSRaw_mp: Raw value of DCS ratio of torque [%] <SWORD> --->
FrmMng_rTCSRaw_mp: Raw value of TCS ratio of torque [%] <SWORD> --->
FrmMng_stASR1Raw0_mp: CAN message object ASR1 byte 0, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw1_mp: CAN message object ASR1 byte 1, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw2_mp: CAN message object ASR1 byte 2, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw3_mp: CAN message object ASR1 byte 3, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw4_mp: CAN message object ASR1 byte 4, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw5_mp: CAN message object ASR1 byte 5, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw6_mp: CAN message object ASR1 byte 6, raw value [-] <UBYTE> --->
FrmMng_stASR1Raw7_mp: CAN message object ASR1 byte 7, raw value [-] <UBYTE> --->

Application parameters

FrmMng_ctASR1MsgCntMax_C: ASR1 message counter, max threshold<value> --->
FrmMng_ctASR1MsgCntMin_C: ASR1 message counter, min threshold<value> --->
FrmMng_ctASR1MsgCntOffDec_C: ASR1 message counter debouncing, decrement<value> --->
FrmMng_ctASR1MsgCntOffInc_C: ASR1 message counter debouncing, increment<value> --->
FrmMng_ctASR1MsgCntOffMax_C: ASR1 message counter debouncing, max threshold<value> --->
FrmMng_rTrqOfs_C: CAN conversion of ratio of torque, offset<value> --->
FrmMng_rTrqSlp_C: CAN conversion of ratio of torque, slope<value> --->
FrmMng_stASR1Dfl0_C: CAN message object ASR1 byte0, default value<value> --->
FrmMng_stASR1Dfl1_C: CAN message object ASR1 byte 1, default value<value> --->
FrmMng_stASR1Dfl2_C: CAN message object ASR1 byte 2, default value<value> --->
FrmMng_stASR1Dfl3_C: CAN message object ASR1 byte 3, default value<value> --->
FrmMng_stASR1Dfl4_C: CAN message object ASR1 byte 4, default value<value> --->
FrmMng_stASR1Dfl5_C: CAN message object ASR1 byte 5, default value<value> --->
FrmMng_stASR1Dfl6_C: CAN message object ASR1 byte 6, default value<value> --->
FrmMng_stASR1Dfl7_C: CAN message object ASR1 byte 7, default value<value> --->
FrmMng_stASR1Ena0_C: CAN message object ASR1 byte 0, enable mask<value> --->
FrmMng_stASR1Ena1_C: CAN message object ASR1 byte 1, enable mask<value> --->
FrmMng_stASR1Ena2_C: CAN message object ASR1 byte 2, enable mask<value> --->
FrmMng_stASR1Ena3_C: CAN message object ASR1 byte 3, enable mask<value> --->
FrmMng_stASR1Ena4_C: CAN message object ASR1 byte 4, enable mask<value> --->
FrmMng_stASR1Ena5_C: CAN message object ASR1 byte 5, enable mask<value> --->
FrmMng_stASR1Ena6_C: CAN message object ASR1 byte 6, enable mask<value> --->
FrmMng_stASR1Ena7_C: CAN message object ASR1 byte 7, enable mask<value> --->
FrmMng_stBrkMsk_C: Bit mask for CAN-brake information<value> --->
FrmMng_stBrkPrpMsk_C: Bit mask for CAN propulsion brake information<value> --->
FrmMng_swtCANProtocol_C: Switch to select the type of CAN protocol - 0= FLORENCE, 1= CAN 4.11<value> --->
FrmMng_trqMaxEngNorm_C: Maximum engine torque for normalization<value> --->
FrmMng_vV1ASR1Ofs_C: Offset of vehicle speed VSO-signal - CAN 4.11<value> --->
FrmMng_vV1ASR1Slp_C: Slope of vehicle speed VSO-signal - CAN 4.11<value> --->
FrmMng_vVS0ASR1Ofs_C: Conversion for ASR1 vehicle speed, offset
FrmMng_vVSOASR1Slp_C: CAN message object ASR1, conversion vehicle speed VSO, slope<value> --->

Receive messages from adaptive cruise control (FrmMng_RecNac)

Functional description

Table 384: Message ACC1 (ID: 3B0h, Data Lenght: 5 Byte, Period: 20 ms)

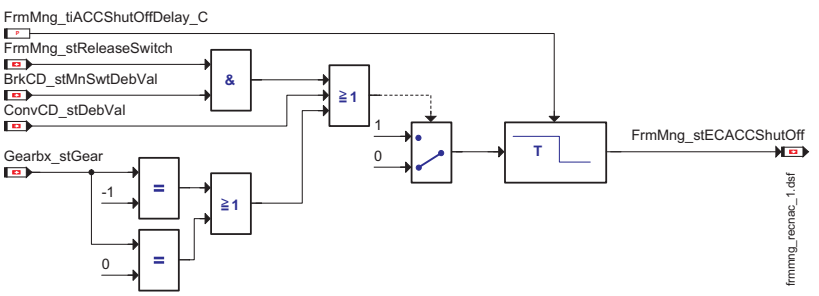
Signal name	Bit)	Measuring point	Description
CANStand_Reserved	63-56	-	Not used by ECU
ACCTorqueCommand	55-48	**FrmMng_dcycACCTrq_mp**	Total indicated torque target.
ACCDecelCommand	47-40	**FrmMng_dcycACCDec_mp**	The signal is used by the ECU only as Complement.
AliveCountACC	39-36	**FrmMng_ctACCAlive_mp**	Message counter.
ACCSts	35-33	**FrmMng_stACCSts_mp**	ACC operating mode.
ACCTrqComValidData	32	**FrmMng_stACCTrqValid_mp**	Torque tracking request is present.
ACCDecComValidData	31	**FrmMng_stACCDecValid_mp**	Evaluate the plausibility conditions
DCBoosterPreControl	30	-	Not used by ECU
DCBoosterHold	29	-	Not used by ECU
-	28-26	-	Reserved for NBA
ACCNoUpshift	25	-	Not used by ECU
-	24	-	-
Checksum_ACC1	23-16	-	Not used by ECU

Shut Off conditions

As soon as the driver activates the brake or the clutch (**ConvCD_stDebVal**) or there is no valid forward gear (**Gearbx_stGear**) and after that for the debouncing time **FrmMng_tiShutOffDelay_C**, no ACC torque intervention may take place.

For braking manoeuvres which are not triggered by the driver (eg. ACC demand to brake assistant), the message **FrmMng_stECACCShutOff** should not be set. Therefore, the state of the brake switch (**BrkCD_stMnSwtDebVal**) is still linked with the status of the brake assistant (**FrmMng_stReleaseSwitch**).

Figure 1039 : Shut Off conditions



Torque intervention

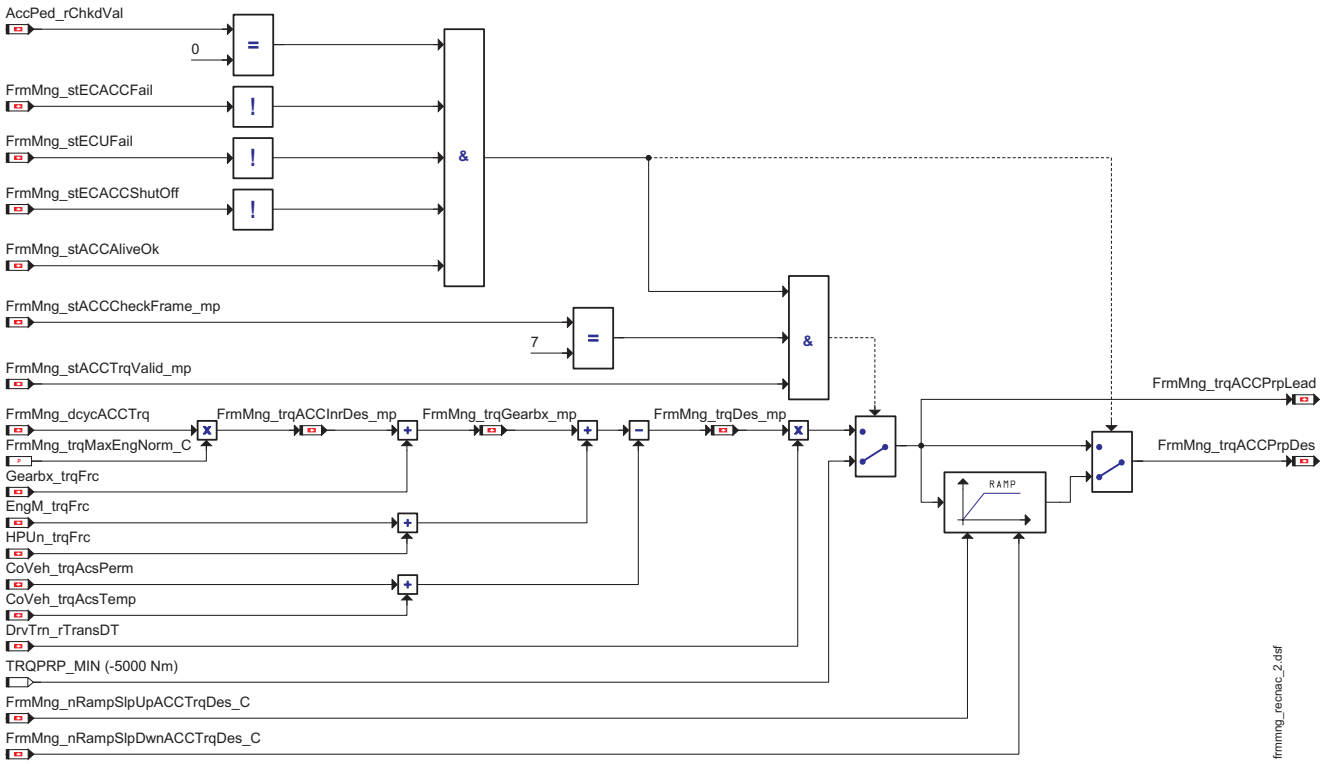
The torque demand sent by the ACC control unit (NAC) is only adopted if the following conditions apply.

- **FrmMng_stECACCFail** = 0 None of the errors assigned to **Fid_FrmMng_ACC_mp** are finally defective.
- **FrmMng_stECUFail** = 0 None of the errors assigned to **Fid_FrmMng_ECUFail_mp** are finally defective.
- **FrmMng_stCheckFrame_mp** = 7 The message was checked for plausibility.
- **FrmMng_stACCAliveOk** = 1 No defect of the ACC alive counter has yet been detected.
- **FrmMng_stECACCShutOff** = 0 At present, the ACC shut-off conditions are not fulfilled.
- **FrmMng_stACCTrqValid_mp** = 1 The ACC control unit demands an increasing torque.

To get the ACC request in [Nm], one has to multiply **FrmMng_dcycACCTrq** with **FrmMng_trqMaxEngNorm_C**. For recalculation of the friction torque one has to add **Gearbx_trqFrc** (result is shown in **FrmMng_trqGearbx_mp**), **EngM_trqFrc** and **HPUn_trqDes**. From this result one subtracts the temporary and permanent accesories torque **CoVeh_trqAcsTemp** and **CoVeh_trqAcsPerm** and gets **FrmMng_trqAcs_mp**. After multiplication with the drive train transformation **DrvTrn_rTransDT**, one gets torque **FrmMng_trqACCPrpDes**, requested from the ECU by ACC. In these calculations the gearshifts aren't take in consideration.

In case of ignoration the torque demand sent by the ACC due to **FrmMng_stECACCFail**, **FrmMng_stECUFail**, **FrmMng_stECACCShutOff** or **FrmMng_stAliveOk** the torque **FrmMng_trqACCPrpDes** will be ramped down to 0. Otherwise there is an instant cutoff.

Figure 1040 : Torque intervention



Monitoring

The plausibility of each ACC1 message received is tested with the following monitoring functions:

- Monitoring of the Alive Counter (**FrmMng_ctACCAlive_mp**)
- Plausibility check of the received ACC status (**FrmMng_stACCSts_mp**)
- Plausibility check of the message contents
- **Fid_FrmMng_ACC**

Alive Counter

Restrictions

The alive counter is only monitored (**FrmMng_stACCCheckFrame_mp**.0 = 1) if the vehicle is not in afterrun (**CoEng_stEng** ≠ 48) and the network manager has not recognized a timeout error (**NetMng_stCANMsg**.12 = 0) or a CAN-BusOff error (**NetMng_stCANBus**.7 = 0).

Monitoring strategy

The ACC1 message contains an alive counter which is continually incremented by the NAC ECU. Therefore, in error-free operation, it is not possible that two successive messages will possess the same value.

Error reaction

If the value of the alive counter does not change, the error counter **FrmMng_ctACCAliveErr_mp** is initially incremented and the ACC torque demand is frozen to the last valid value. However, the message **FrmMng_stACCAliveOk** remains set.

On the other hand, if the error counter reaches the threshold **FrmMng_ctACCAliveErrMin_C**, the message **FrmMng_stACCAliveOk** is set to zero which prevents an ACC torque intervention. At the same time, error debouncing of **Dpf_FrmMng_ACC_mp**.0 is started.

Error debouncing is implemented by a counter. The counter mode (normal, up-down) and the step width can be appplicated with the label **FrmMng_stACCAliveCnt_C**.

Figure 1041 : Error debouncing

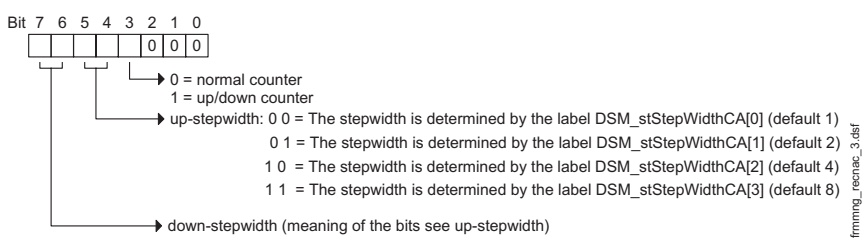


Figure 1042 : Dfp_FrmMng_ACC_mp.o

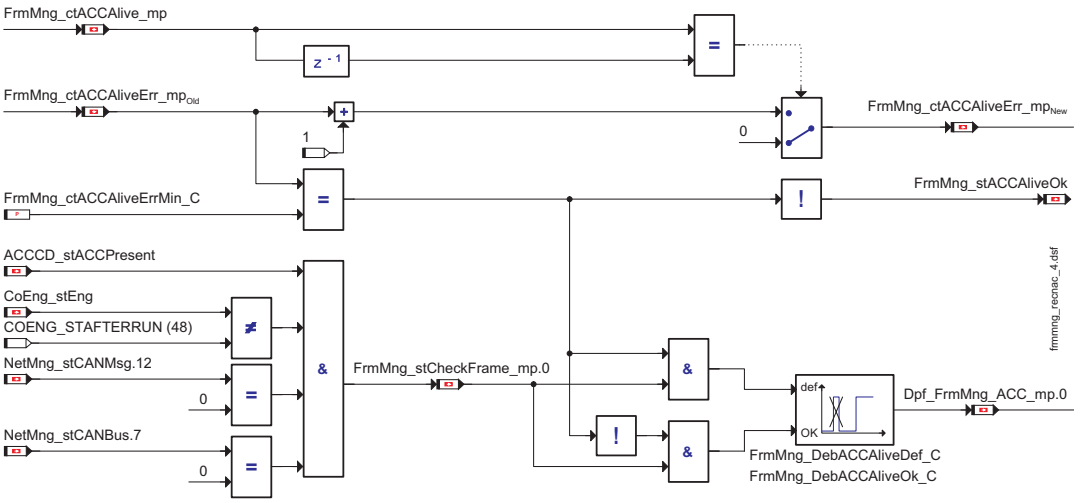
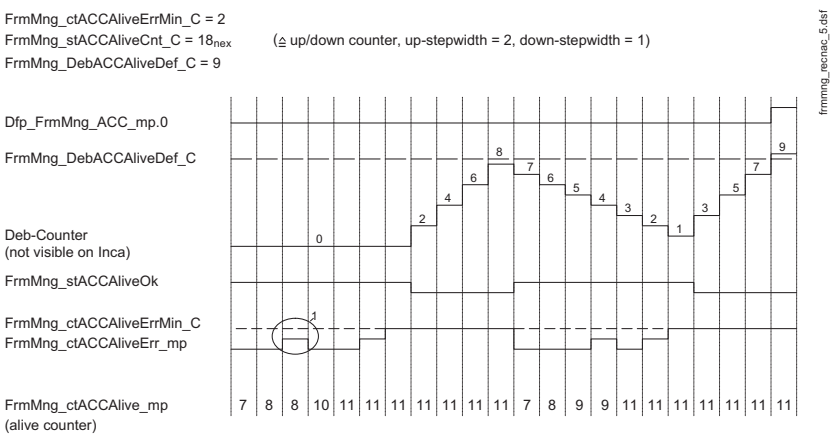


Figure 1043 : Example for the error detection with an up/down counter



[1]sporadic errors in the alive counter, which occur mainly due to sampling problems, are ignored (in these cases the torque demand is frozen at the last valid value)

ACC status

Restrictions

The plausibility checks of the received ACC status are only executed (**FrmMng_stACCCheckFrame_mp**.1 = 1), if the alive counter could be monitored (**FrmMng_stACCCheckFrame_mp**.0 = 1) and if the NAC-ECU reports no error (**FrmMng_stACCSts_mp** ≠ 7).

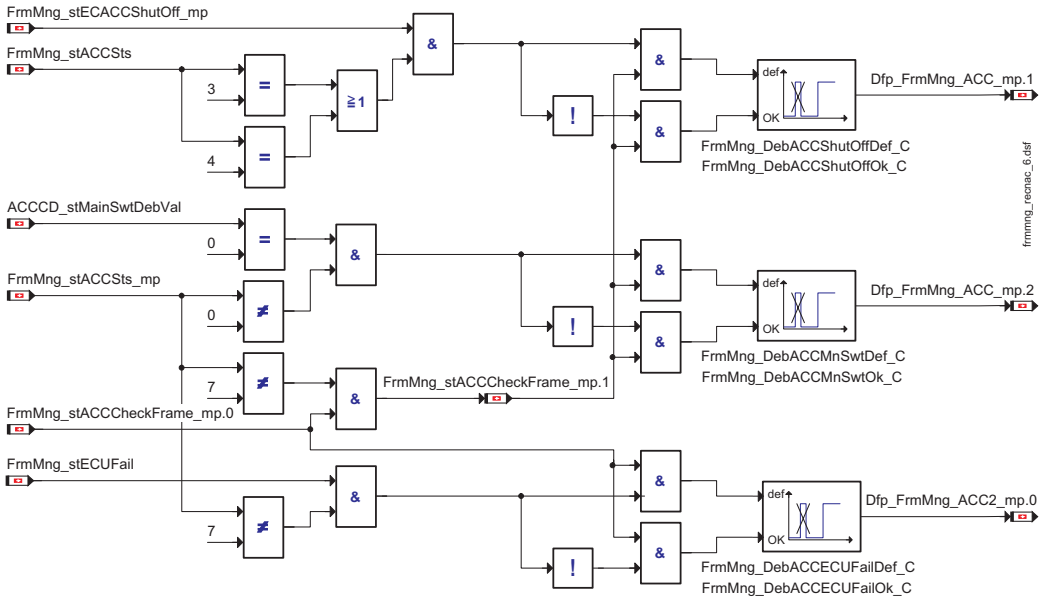
Monitoring strategy

- If the driver would like to brake or operate the clutch (**FrmMng_stECACCShutOff** = 1), the NAC-ECU must not transfer any more demand which increases torque.
- If the main switch of the ACC control device is switched off (**ACCCD_stMainSwtDebVal** = 0), the NAC-ECU must transmit the status 0 after a certain period.

Error reaction

- If the state of the message **FrmMng_stECACCShutOff** does not match the received ACC status, the error **Dpf_FrmMng_ACC**.1 is reported. As soon as the error is finally defective, an ACC torque intervention is prevented.
- If the state of the ACC main switch **ACCCD_stMainSwtDebVal** does not match the received ACC status, the error **Dpf_FrmMng_ACC**.2 is reported. As soon as the error is finally defective, an ACC torque intervention is prevented.
- If the state of the message **FrmMng_stECUFail** = 1 and **FrmMng_stACCSts_mp** is not equal 7, the error **Dfp_FrmMng_ACC_mp**.0 is reported. As soon as the error is finally defective, an ACC torque intervention is prevented.

Figure 1044 : ACC status



Message contents

Restrictions

The plausibility checks of the message contents are only executed (**FrmMng_stACCCheckFrame_mp**.2 = 1), if the alive counter could be monitored (**FrmMng_stACCCheckFrame_mp**.0 = 1) and if the NAC-ECU reports no error (**FrmMng_stACCSts_mp** ≠ 7) and is not located in the Off mode (**FrmMng_stACCSts_mp** ≠ 0) and if at least one of the vaild bits (**FrmMng_stACCTrqValid_mp** or **FrmMng_stACCDecValid_mp**) is set.

Monitoring strategy

In one ACC1 message, the NAC-ECU always transmits only one demand which increases the torque or one which reduces it. Therefore, both the valid bits (**FrmMng_stACCTrqValid_mp** or **FrmMng_stACCDecValid_mp**) must not be set simultaneously.

In addition to the actual torque demand, the NAC-ECU transmits the complementary value of this demand. Depending on the ACC status (increasing or reducing torque demand), a slightly different algorithm is used.

Increasing torque demand (**FrmMng_stACCTrqValid_mp** = 1):

FrmMng_dcycACCDec_mp = BitByBitNegationOf (**FrmMng_dcycACCTrq_mp** + **FrmMng_ctACCAlive_mp**)

Reducing torque demand (**FrmMng_stACCDecValid_mp** = 1):

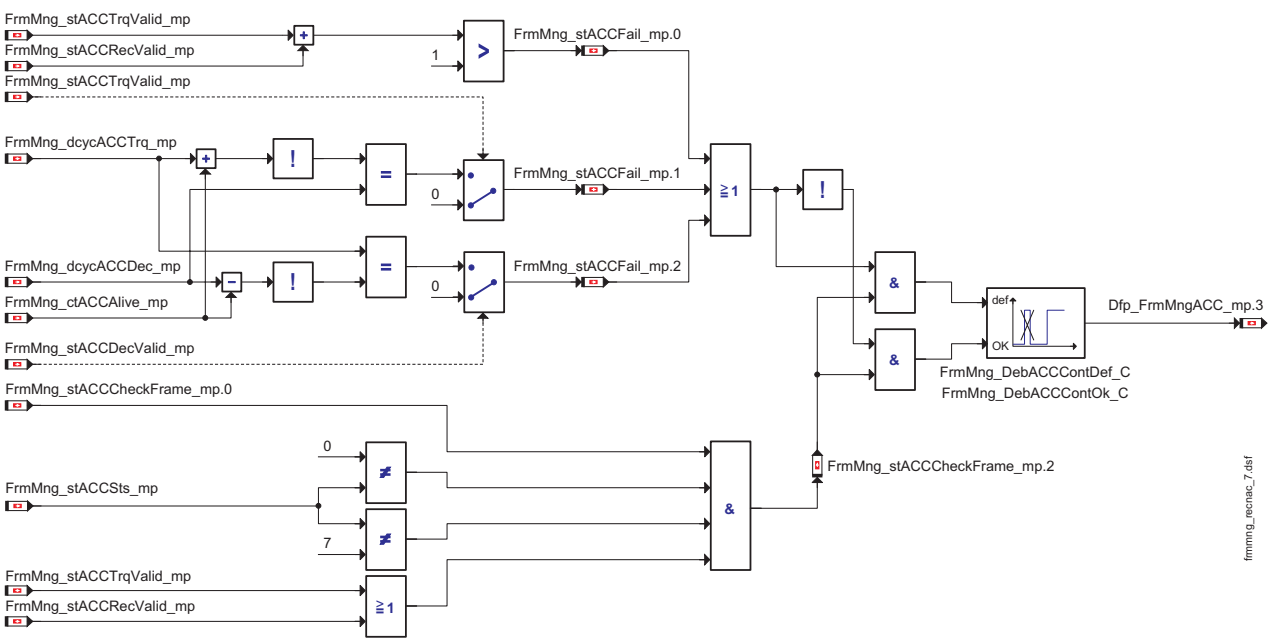
FrmMng_dcycACCTrq_mp = BitByBitNegationOf (**FrmMng_dcycACCDec_mp** - **FrmMng_ctACCAlive_mp**)

Error reaction

If one of the plausibility conditions is not fulfilled, the error **Dfp_FrmMng_ACC_mp**.3 is reported. During the debouncing, the last valid torque demand continues to be used. As soon as the error is finally defective, an ACC torque intervention is prevented.

Error debouncing is implemented by a counter. The counter mode (normal, up-down) and the step width can be appplicated with the label **FrmMng_stACCContCnt_C**. The meaning of the bits is identical to the label **FrmMng_stACCAliveCnt_C**.

Figure 1045 : Dfp_FrmMng_ACC_mp.3



Fid_FrmMng_ACC

All those errors are included in this Fid for which, in the event of an error, the ACC torque demand is set to zero. This Fid should at least contain the error path **Dfp_FrmMng_ACC** (implausible or faulty ACC message) and the error path **Dfp_NetMng_NAC** (ACC message failure).

Whenever the ACC torque demand is set by the Fid to zero, the status bit **FrmMng_stECACCFail** is set.

Figure 1046 : FrmMng_stECACCFail

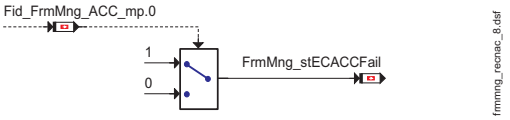
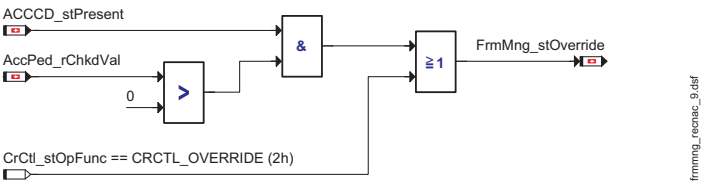


Figure 1047 : FrmMng_stOverride



Output values

FrmMng_stACCAliveOk: ACC Alive counter is ok [-] <UBYTE> --->
FrmMng_stECACCFail: ECU detects ACC frame errors [-] <UBYTE> --->
FrmMng_stECACCShutOff: Shut off conditions for the ACC [-] <UBYTE> --->
FrmMng_trqACCPrpDes: Torque request from ACC [Nm] <SWORD> --->
FrmMng_trqACCPrpLead: received lead ACC-torque [Nm] <SWORD> --->

Input values

ACCCD_stMainSwtDebVal: debounce value of main switch [-] <UBYTE> --->
BrkCD_stMnSwtDebVal: Brake main switch debounced logical value [-] <UBYTE> --->
CoEng_stEng: current engine state [-] <UBYTE> --->
ConvCD_stDebVal: debounced value of clutch signal [-] <UBYTE> --->
CoVeh_trqAcsPerm: torque of permanent active accessories [Nm] <SWORD> --->
CoVeh_trqAcsTemp: torque of temporary active accessories [Nm] <SWORD> --->
DrvTrn_rTransDT: drivetrain transmission (wheel torque per engine torque) [-] <SWORD> --->
EngM_trqFrc: current friction torque [Nm] <SWORD> --->
FrmMng_stRelease:
Gearbx_trqFrc: current frictional torque of gearbox [Nm] <SWORD> --->
HPUn_trqDes: torque demand of high pressure pump [Nm] <SWORD> --->
NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->

Measuring points

Dfp_FrmMng_ACC2_mp: Measurement point for fault path Dfp_FrmMng_ACC2 [-] <UWORD> --->
Dfp_FrmMng_ACC_mp: Measurement point for fault path of Dfp_FrmMng_ACC_mp [-] <UWORD> --->
Fid_FrmMng_ACC_mp: Measurement point for the function identifier Fid_FrmMng_ACC [-] <UBYTE> --->
FrmMng_ctACCAlive_mp: Alive counter [-] <UBYTE> --->
FrmMng_ctACCAliveErr_mp: Alive error counter [-] <UBYTE> --->
FrmMng_dcycACCDec_mp: Complement of indicated torque [[%]] <UBYTE> --->
FrmMng_dcycACCTrq_mp: Total indicated torque target [[%]] <UBYTE> --->
FrmMng_stACCCheckFrame_mp: contents of ACC1 CAN message is checked [-] <UBYTE> --->
FrmMng_stACCDecValid_mp: Evaluate the plausibilit conditions [-] <UBYTE> --->
FrmMng_stACCFail_mp: ACC contents has an failure [-] <UBYTE> --->
FrmMng_stACCSts_mp: ACC operating mode [-] <UBYTE> --->
FrmMng_stACCTrqValid_mp: Torque tracking request is present [-] <UBYTE> --->
FrmMng_trqACCAcs_mp: ACC engine torque [Nm] <SWORD> --->
FrmMng_trqACCGearbx_mp: ACC torque with frictional torque of gearbox [Nm] <SWORD> --->
FrmMng_trqACCInrDes_mp: ACC torque [Nm] <SWORD> --->

Application parameters

DSM_CDCDfp_FrmMng_ACC2_C: CARB specific fault code for fault path Dfp_FrmMng_ACC2<value_block> --->
DSM_CDKDfp_FrmMng_ACC2_C: Diagnostic Trouble Code customer specific for fault path Dfp_FrmMng_ACC2<value> --->
DSM_ClaDfp_FrmMng_ACC2_C: Fault Class for fault path Dfp_FrmMng_ACC2<value> --->
DSM_EnvDfp_FrmMng_ACC2_C: Path specific environment conditions for fault path Dfp_FrmMng_ACC2<value_block> --->
DSM_TsfDfp_FrmMng_ACC2_C: Severity Counter Limit for fault path Dfp_FrmMng_ACC2<value> --->
DSM_TypDfp_FrmMng_ACC2_C: Conversion of fault types for diagnostic interface for fault

path Dfp_FrmMng_ACC2<value_block> --->
FrmMng_ctACCAliveErrMin_C: Threshold for alive counter error<value> --->
FrmMng_DebACCAliveDef_C: defect debounce value of alive counter error<value> --->
FrmMng_DebACCAliveOk_C: healing debounce value of alive counter error<value> --->
FrmMng_DebACCContDef_C: defect debounce value of ACC messages contents error<value> --->
FrmMng_DebACCContOk_C: healing debounce value of ACC messages contents error<value> --->
FrmMng_DebACCECUFailDef_C: defect debounce value of ECU Fail error<value> --->
FrmMng_DebACCECUFailOk_C: heal time for message ECU fail<value> --->
FrmMng_DebACCMnSwtDef_C: defect debounce time of ACC main switch error<value> --->
FrmMng_DebACCMnSwtOk_C: healing debounce time of ACC main switch error<value> --->
FrmMng_DebACCShutOffDef_C: defect debounce time of ACC shut off error<value> --->
FrmMng_DebACCShutOffOk_C: healing debounce time of ACC shut off error<value> --->
FrmMng_nRampSlpDwnACCTrqDes_C: Slope down ACC torque desired<value> --->
FrmMng_nRampSlpUpACCTrqDes_C: Slope up ACC torque desired<value> --->
FrmMng_stACCAliveCnt_C: Debounce counter mode of alive counter error<value> --->
FrmMng_stACCContCnt_C: Debounce counter mode of ACC contents error<value> --->
FrmMng_tiACCShutOffDelay_C: Off delay time of shut off conditions<value> --->

Receive messages from brake assistant (FrmMng_RecNBA)

Functional overview

Functional description

In this chapter there is the description of CAN messages which are received from brake assistant control node.

Message STATUS C-NBA

ID: 573h, Data Length: 1 byte, Period: 500ms

The message STATUS C-NBA is not supported.

Message BA

ID: 433h, Data Length: 8 byte, Period: 10ms only for Fiat engine project 841.

Table 385: BA - frame overview

Signal name	Bit	Description
DCDiagnosisMode	63	Not used by ECU.
DC_ACCFailSts	62	Not used by ECU.
DCIntervention	61	Not used by ECU.
DCPresent	60	Not used by ECU.
-	59-48	-
BoosterReleaseSwitchValidData	47	Not used by ECU.
ReleaseSwitch	46	1, when brake intervention is due to the driver. 0, when brake intervention is due to the ACC.
-	45-44	-
NBA-NFR_StatusFlags (reserved)	43-41	Not used by ECU.
DC-NFR_MaxCommandValidData	40	Not used by ECU.
DCFunctionFailSts	39	Not used by ECU.
VehicleAccelerationNACValidData	38	Not used by ECU.
VehicleSpeedRefNACValidData	37	Not used by ECU.
VehicleSpeedRefNAC	36-24	Not used by ECU.
VehicleAccelerationNAC	23-16	Not used by ECU.
DC-NFR_MaxCommand	15-8	Not used by ECU.
DC-NCA_StatusFlags	7-0	Not used by ECU.

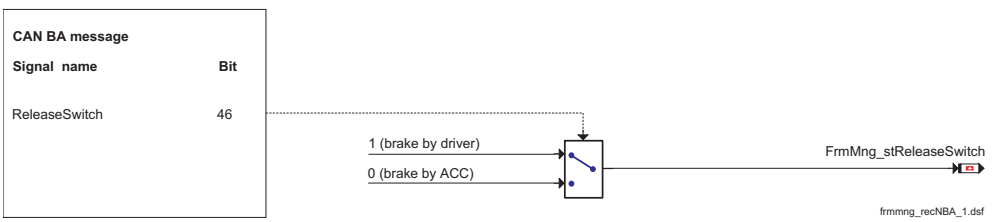
Evaluation of ReleaseSwitch

This signal is necessary because the brake intervention of the ACC cause, in case of systems equiped with a brake assistant, the movement of the brake pedal and the consequent changes of brake switch and stop light switch.

When the brake switch is in "pressed position" and **FrmMng_stReleaseSwitch** is 1 (brake intervention due to the driver), the adaptive cruise control shall be deactivated.

When the brake switch is in "pressed position" and **FrmMng_stReleaseSwitch** is 0 (brake intervention due to the ACC), the adaptive cruise control shall not be deactivated.

Figure 1048 : BA ReleaseSwitch



ECU initialization

During initialization phase the ID, the Data Length and the Period are set according to the specification. The data bytes are taken from the default data set.

Output values

FrmMng_stReleaseSwitch: Distinguish between driver or ACC brake intervention [-] <UBYTE> --->

Measuring points

FrmMng_stBARaw0_mp: CAN message object BA byte 0, raw value [-] <UBYTE> --->
FrmMng_stBARaw1_mp: CAN message object BA byte 1, raw value [-] <UBYTE> --->
FrmMng_stBARaw2_mp: CAN message object BA byte 2, raw value [-] <UBYTE> --->
FrmMng_stBARaw3_mp: CAN message object BA byte 3, raw value [-] <UBYTE> --->
FrmMng_stBARaw4_mp: CAN message object BA byte 4, raw value [-] <UBYTE> --->
FrmMng_stBARaw5_mp: CAN message object BA byte 5, raw value [-] <UBYTE> --->
FrmMng_stBARaw6_mp: CAN message object BA byte 6, raw value [-] <UBYTE> --->
FrmMng_stBARaw7_mp: CAN message object BA byte 7, raw value [-] <UBYTE> --->

Application parameters

FrmMng_stBADfl0_C: CAN message object BA byte 0, default value<value> --->
FrmMng_stBADfl1_C: CAN message object BA byte 1, default value<value> --->
FrmMng_stBADfl2_C: CAN message object BA byte 2, default value<value> --->
FrmMng_stBADfl3_C: CAN message object BA byte 3, default value<value> --->
FrmMng_stBADfl4_C: CAN message object BA byte4, default value<value> --->
FrmMng_stBADfl5_C: CAN message object BA byte5, default value<value> --->
FrmMng_stBADfl6_C: CAN message object BA byte6 default value<value> --->
FrmMng_stBADfl7_C: CAN message object BA byte7 default value<value> --->
FrmMng_stBAEna0_C: CAN message object BA byte0, enable mask<value> --->
FrmMng_stBAEna1_C: CAN message object BA byte1, enable mask<value> --->
FrmMng_stBAEna2_C: CAN message object BA byte2, enable mask<value> --->
FrmMng_stBAEna3_C: CAN message object BA byte3, enable mask<value> --->
FrmMng_stBAEna4_C: CAN message object BA byte4, enable mask<value> --->
FrmMng_stBAEna5_C: CAN message object BA byte5, enable mask<value> --->
FrmMng_stBAEna6_C: CAN message object BA byte6, enable mask<value> --->
FrmMng_stBAEna7_C: CAN message object BA byte7, enable mask<value> --->

Receive messages from body computer (FrmMng_RecNBC)

Functional description

In this chapter there is the description of CAN messages which are received from body computer control node.

Message TIME & DATE (TIDA)

ID: 6A0h, Data Length: 6 byte, Period: 1000 ms

The message TIME & DATE is not supported

Message EOL Configuration (EOLC)

ID: 680h, Data Length: 8 byte, Period: 1000 ms

The message EOLC is not supported

Message STATUS B_CAN (STNB)

ID: 560h, Data Length: 8 byte, Period: 100 ms

Table 386: STNB - frame overview

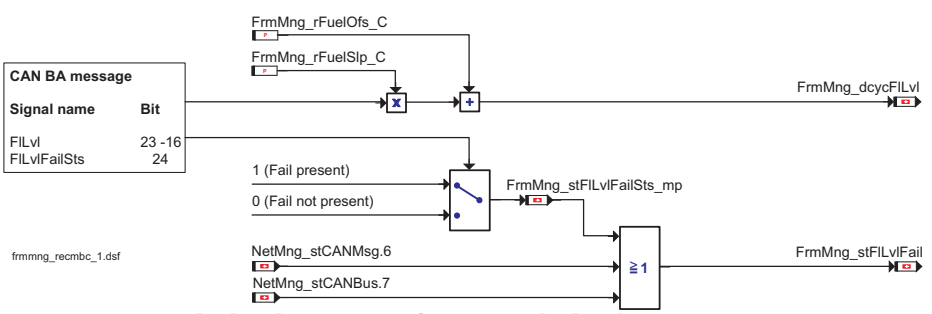
Signal name	Bit	Description
ABS Lamp - Fail Sts	63-62	Not used by ECU
EBD Lamp - Fail Sts	61-60	Not used by ECU
TC/ASR Lamp - Fail Sts	59-58	Not used by ECU
VDC Lamp - Fail Sts	57-56	Not used by ECU
Electric Steering Lamp - Fail Sts	55-54	Not used by ECU
Parking Brake Enable	51	Not used by ECU
Browse Gear BC Sts	50	Not used by ECU
City Mode Sts	49	Not used by ECU
Steering Load	48	Not used by ECU
Key Sts	47-44	Not used by ECU
Driver Door Sts	43	Not used by ECU
NBC Clutch Pedal Sts	42	Not used by ECU
PB Lamp - Fail Sts	41-40	Not used by ECU
External Temperature	39-32	Not used by ECU
External Temperature Fail Sts	31	Not used by ECU
Rain Sensor Fail Sts	30	Not used by ECU
Rain Sensor	29-26	Not used by ECU
Recharge Sts	25	Not used by ECU
Fuel Level Fail Sts	24	Fuel Level fail status
Fuel Level	23-16	Fuel Level
Brake Pad Wear Sts	15	Not used by ECU
Brake Fluid Level Sts	14	Not used by ECU
Hand Brake Sts	13	Not used by ECU
R Heated Window Sts	12	Not used by ECU
LH Turn Signal Sts	11	Not used by ECU
RH Turn Signal Sts	10	Not used by ECU
Display Fail Sts	9-8	Not used by ECU
Travel Distance	7-0	Not used by ECU

Evaluation of fuel level

This signal is needed for additive dosing system for the DPF function. **FrmMng_dcycFlLvl** is 100% if the fuel tank is full and 0% if it is empty. If a fail is present **FrmMng_stFlLvlFail** is 1.

If there occurs a transmission error **NetMng_stCANMsg**.6 or a bus error **NetMng_stCANBus**.7 **FrmMng_stFlLvlFail** is also set.

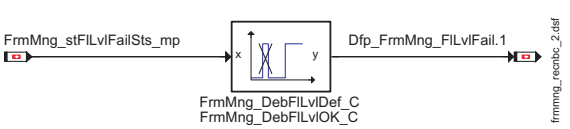
Figure 1049 : STNB status fuel level fail



Error reaction

If **FrmMng_stFlLvlFail** is 1, the error **Dfp_FrmMng_FlLvlFail**.1 is reported.

Figure 1050 :



ECU initialization

During initialization phase the ID, the Data Length and the Period are set according to the specification. the data bytes are taken from the default data set.

Output values

FrmMng_dcycFlLvl: Fuel Level via CAN
FrmMng_stFlLvlFail: Status of Fuel Level signal

Input values

NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->

Measuring points

Dfp_FrmMng_FlLvlFail_mp: Measurement point for fault path Dfp_FrmMng_FlLvlFail [-] <UWORD> --->
FrmMng_stFlLvlFail1_mp: Measurement point FuelLevelFailSts status
FrmMng_stSTNBRaw0_mp: CAN message object Status-B-CAN byte 0, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw1_mp: CAN message object Status-B-CAN byte 1, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw2_mp: CAN message object Status-B-CAN byte 2, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw3_mp: CAN message object Status-B-CAN byte 3, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw4_mp: CAN message object Status-B-CAN byte 4, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw5_mp: CAN message object Status-B-CAN byte 5, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw6_mp: CAN message object Status-B-CAN byte 6, raw value [-] <UBYTE> --->
FrmMng_stSTNBRaw7_mp: CAN message object Status-B-CAN byte 7, raw value [-] <UBYTE> --->

Application parameters

FrmMng_DebFlLvlDef_C: Defect debounce time of Fuel Level error
FrmMng_DebFlLvlOk_C: Healing decounce time of Fuel Level error
FrmMng_rFuelOfs_C: conversion of fuel level, offset<value> --->
FrmMng_rFuelSlp_C: conversion of fuel level, slope<value> --->
FrmMng_stSTNBDfl0_C: CAN message object Status-B-CAN byte 0, default value<value> --->
FrmMng_stSTNBDfl1_C: CAN message object Status-B-CAN byte 1, default value<value> --->
FrmMng_stSTNBDfl2_C: CAN message object Status-B-CAN byte 2, default value<value> --->
FrmMng_stSTNBDfl3_C: CAN message object Status-B-CAN byte 3, default value<value> --->
FrmMng_stSTNBDfl4_C: CAN message object Status-B-CAN byte 4, default value<value> --->
FrmMng_stSTNBDfl5_C: CAN message object Status-B-CAN byte 5, default value<value> --->
FrmMng_stSTNBDfl6_C: CAN message object Status-B-CAN byte 6, default value<value> --->
FrmMng_stSTNBDfl7_C: CAN message object Status-B-CAN byte 7, default value<value> --->
FrmMng_stSTNBEna0_C: CAN message object Status-B-CAN byte 0, enable mask<value> --->
FrmMng_stSTNBEna1_C: CAN message object Status-B-CAN byte 1, enable mask<value> --->
FrmMng_stSTNBEna2_C: CAN message object Status-B-CAN byte 2, enable mask<value> --->
FrmMng_stSTNBEna3_C: CAN message object Status-B-CAN byte 3, enable mask<value> --->
FrmMng_stSTNBEna4_C: CAN message object Status-B-CAN byte 4, enable mask<value> --->
FrmMng_stSTNBEna5_C: CAN message object Status-B-CAN byte 5, enable mask<value> --->
FrmMng_stSTNBEna6_C: CAN message object Status-B-CAN byte 6, enable mask<value> --->
FrmMng_stSTNBEna7_C: CAN message object Status-B-CAN byte 7, enable mask<value> --->

Receive messages from gearbox (FrmMng_RecTsc)

Functional overview

Functional description

In this chapter there is the description of CAN messages which are received from gearbox node (transmission control system).

Message STATUS C-NCA

ID: 056h, Data Length: 8 byte, Period: 100ms

The message STATUS C-NCA is not supported.

Message GEARMOT

ID: 0CBh, Data Length: 8 byte, Period: 10ms

Table 387: GEARMOT - frame overview

Signal name	Bit	Description
FanOnReq	63	Activation of cooling fan(s), (not yet implemented)
TorqueConverterSts	62-61	Formation of converter information
MI_Req	60	MIL request by gearbox
TransmissionTemperatureFailSts	59	Failure of signal TransmissionTemperature
GearSwitchActiveSts	58	Formation of converter information
TorqueAmpTransmissionValidData	57	-
NCAReqForTorqueRedValidData	56	Validation of NCAReqForTorqueRed
NCAReqForTorqueRed	55-48	DeltaTorque intervention/reduction
TorqueAmpTransmission	47-40	-
TransmissionTemperature	39-32	Temperature of transmission for activating cooling fan(s)
ActualGear	31-29	Formation of gear information
GearEngagement	28	Formation of converter information
GearEngagementValidData	27	Validation of GearEngagement
ActualGearValidData	26	Validation of ActualGear
TorqueLimitationValidData	25	Validation of TorqueLimitation
TorqueIncreaseValidData	24	Validation of TorqueIncrease
TorqueLimitation	23-16	Absolute Crankshaft Torque intervention/limitation
TorqueIncrease	15-8	Absolute Crankshaft Torque intervention/increase
MessageCounter	7-4	Enable torque interventions by gearbox
TorqueIncreaseActive	3	Enable torque intervention/increase
TargetGear	2-0	Formation of gear information

Evaluation of GEMO Message Counter

To check the continuous incrementation of the GEMO Message Counter the actual received value **FrmMng_ctGEMOMsgCnt0_mp** is compared with the value received three times before **FrmMng_ctGEMOMsgCnt3_mp**. The incrementation is correct if the difference is between the two calibratable values **FrmMng_ctGEMOMsgCntMin_C** and **FrmMng_ctGEMOMsgCntMax_C**. The schedule time of this process is 10ms.

Equation 51: GEMO Message Counter check

$$GEMOMsgCntMin \le GEMOMsgCnt(n) - GEMOMsgCnt(n-3) \le GEMOMsgCntMax$$

During torque increase is active (TorqueIncreaseActive=1) an additionally Complement check is done concerning the GEMO Message Counter and the torque reduct request NCAReqForTorqueRed. The complement in this case is a bitwise negation.

Equation 52: GEMO Complement check (if torque increase active)

$$GEMOTorqueReduct = CMPL(GEMOMsgCnt(n) + TorqueIncrease)$$

The GEMO Message Counter check is an input of an additional debouncing counter **FrmMng_ctGEMOMsgCntOff_mp**. The debouncing counter has got a calibratable increment **FrmMng_ctGEMOMsgCntOffInc_C** and decrement **FrmMng_ctGEMOMsgCntOffDec_C**. The status **FrmMng_stGEMOMsgCnt** is set if the debouncing counter **FrmMng_ctGEMOMsgCntOff_mp** is under the limit **FrmMng_ctGEMOMsgCntOffMax_C**.

Figure 1051 : GEMO Message Counter

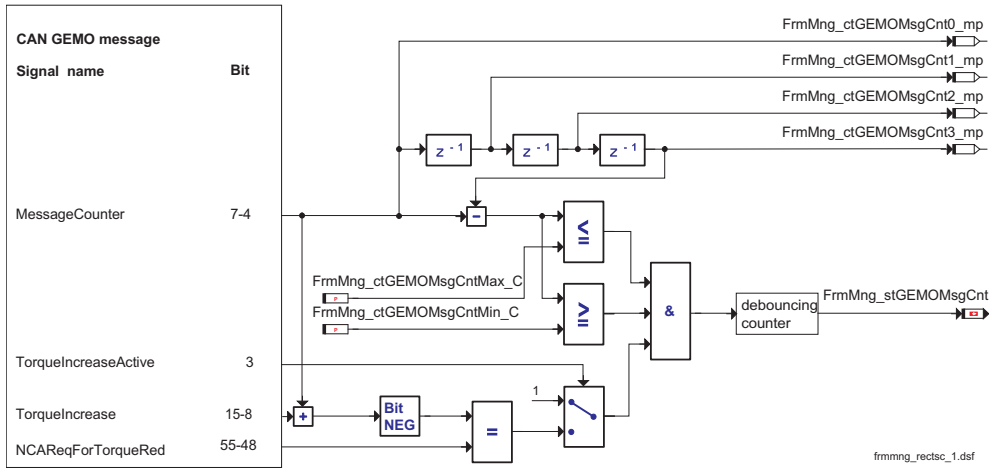
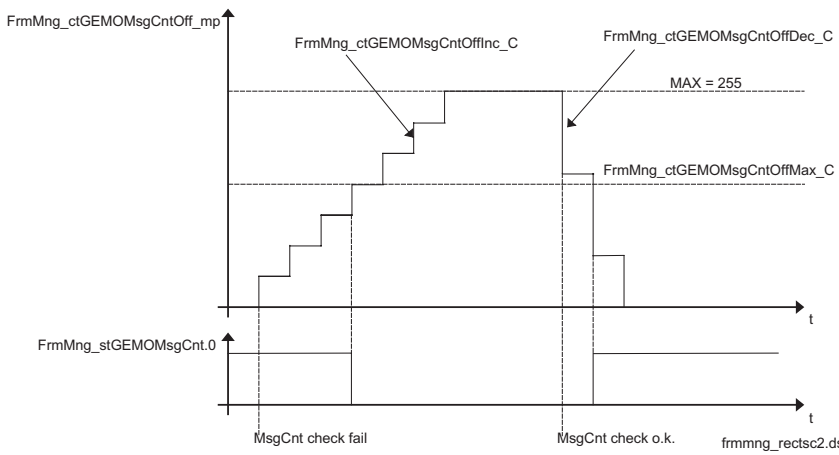


Figure 1052 : GEMO Message Counter, debouncing



Evaluation of Gear Information

The received CAN gear **FrmMng_stGearRaw_mp** is converted as shown in the table and sent in the message **FrmMng_stGear**.

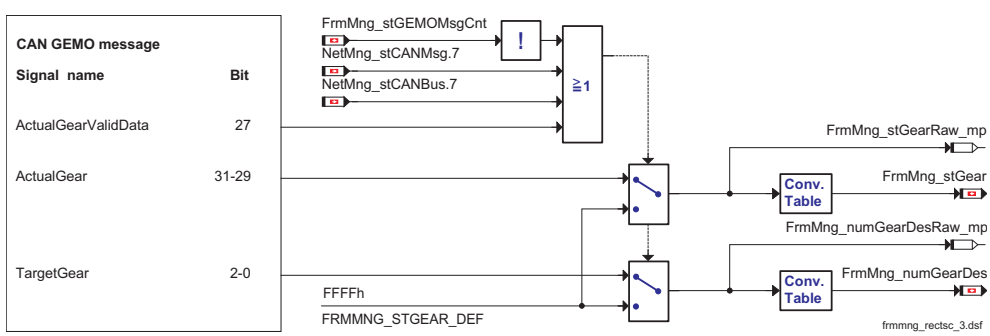
The received CAN target gear **FrmMng_numGearDesRaw_mp** is converted as shown in the table and sent in the message **FrmMng_numGearDes**.

If any CAN message error or CAN bus error is present the default value is sent in **FrmMng_stGear** and **FrmMng_numGearDes**.

Table 388: Conversion Table of Gear Information

FrmMng_stGearRaw_mp, FrmMng_numGearDesRaw_mp	FrmMng_stGear, FrmMng_numGearDes	Description
1	01h	1st gear
2	02h	2nd gear
3	03h	3rd gear
4	04h	4th gear
5	05h	5th gear
6	00h	N or P Position
7	10h	1st reverse gear
All other states	FFFFh	Defect

Figure 1053 : Gear Information



Evaluation of Converter Information

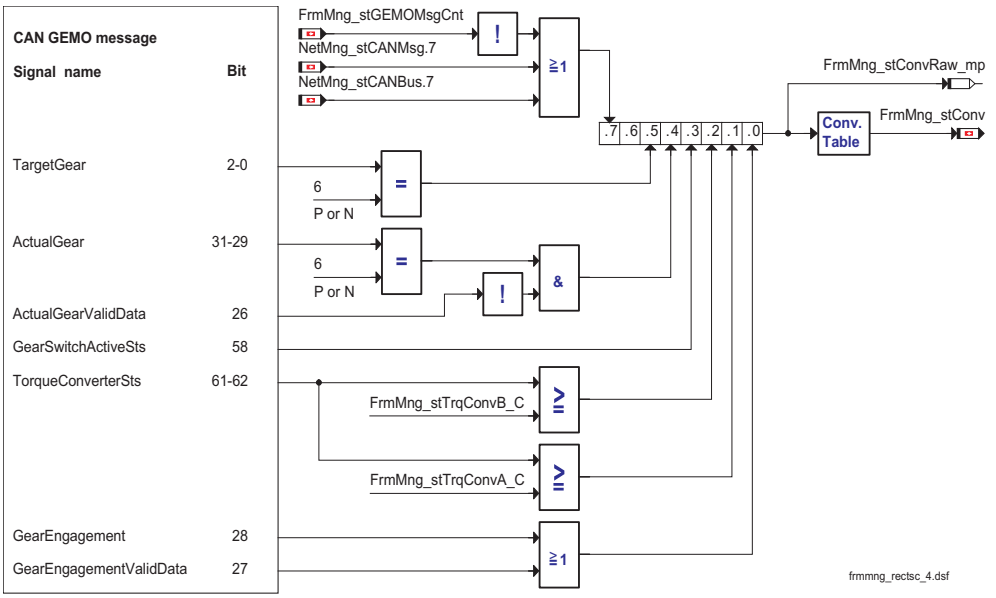
The received CAN converter states are represented in the measurement point **FrmMng_stConvRaw_mp**. The state is converted as shown in the table and sent in the message **FrmMng_stConv**.

If any CAN message error or CAN bus error is present the default value is sent in **FrmMng_stConv**.

Table 389: Conversion Table of Converter Information

FrmMng_stConvRaw_mp	FrmMng_stConv	Description
0xxxx111b	0	Clutch not actuated / disengaged
0xxxx011b	1	Control mode
0xxxxx01b 0xxxxxx0b	2	Clutch actuated / engaged
1xxxxxxb	3	Defect

Figure 1054 : Converter Information

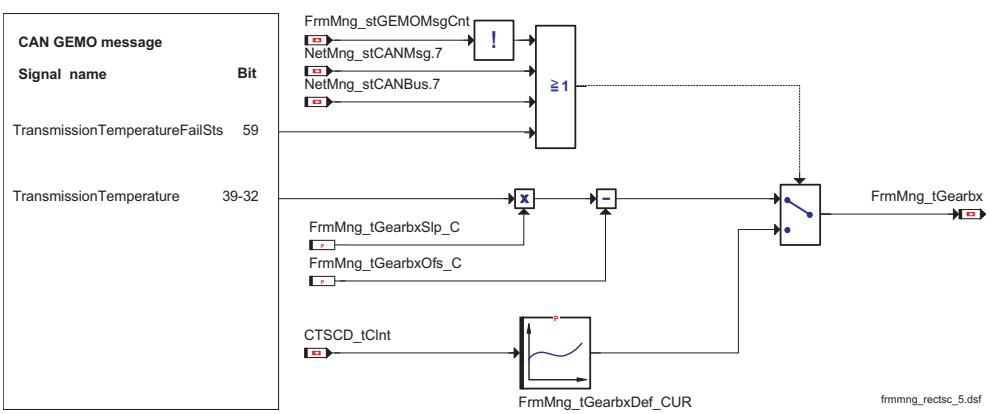


Evaluation of Gearbox Temperature

The received CAN gearbox temperature is converted by offset **FrmMng_tGearbxOfs_C** and slope **FrmMng_tGearbxSlp_C** and is then sent in the message **FrmMng_tGearbx**.

If any CAN message error or CAN bus error is present the gearbox temperature will be calculated as a default gearbox temperature by engine water temperature **CTSCD_tClnt** and the curve **FrmMng_tGearbxDef_CUR**.

Figure 1055 : Gearbox Temperature

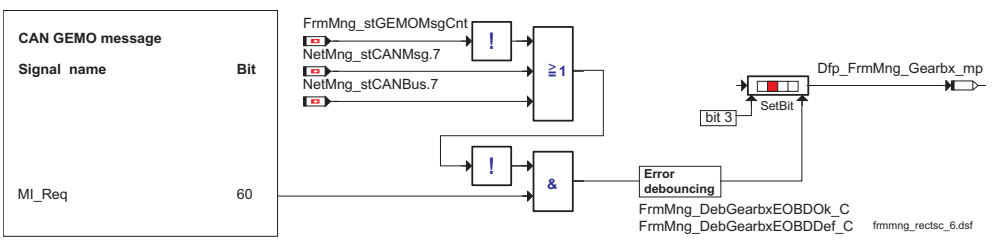


Evaluation of Gearbox EOBD Error

The received gearbox EOBD error will set bit 3 in the fault path **Dfp_FrmMng_Gearbx_mp**. The gearbox EOBD error is debounced by **FrmMng_DebGearbxEOBDDef_C** and **FrmMng_DebGearbxEOBDOk_C**.

If any CAN message error or CAN bus error is present the the error debouncing is stopped.

Figure 1056 : Gearbox EOBD Error



Evaluation of Torque Increase

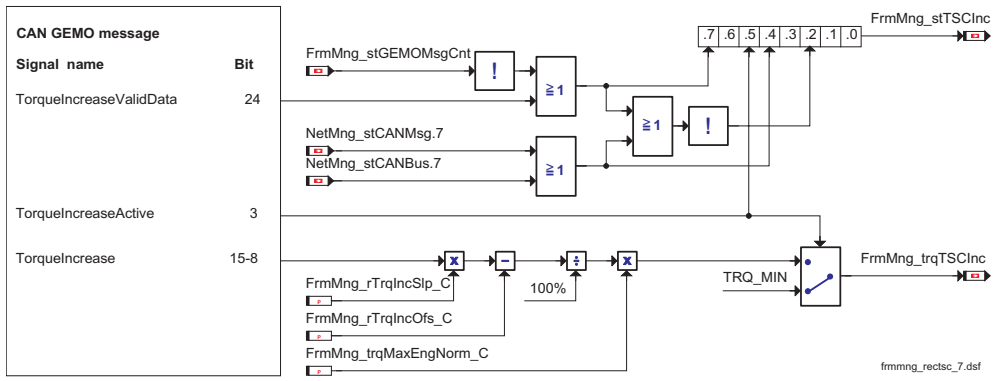
The received torque increase is converted by offset **FrmMng_rTrqIncOfs_C** and slope **FrmMng_rTrqIncSlp_C**. The measurement point **FrmMng_rTSCIncRaw_mp** displays the received ratio of torque. Additionally the renormalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done and then the torque increase will be sent in the message **FrmMng_trqTSCInc**.

If any CAN message error or CAN bus error is present the torque increase state **FrmMng_stTSCInc** will be set accordingly.

Table 390: Gearbox Torque Increase, state

FrmMng_stTSCInc	
Bit 1, FRMMNG_STGBXTRQ_MNMX	Gearbox Increase and Decrease allowed
Bit 2, FRMMNG_STGBXTRQ_NORAMP	Gearbox Increase without ramp
Bit 3, FRMMNG_STGBXTRQ_LIMP	Gearbox in Limp Home Mode
Bit 4, FRMMNG_STGBXTRQ_CANOFF	Gearbox Increase is affected due to CAN errors
Bit 5, FRMMNG_STGBXTRQ_DEMOFF	Gearbox Increase is active
Bit 7, FRMMNG_STGBXTRQ_MSGERR	Gearbox Increase is affected due to CAN message errors

Figure 1057 : Gearbox Torque Increase



Evaluation of Torque Decrease

The received torque limitation is converted by offset **FrmMng_rTrqLimOfs_C** and **FrmMng_rTrqLimSlp_C**. The measurement point **FrmMng_rTSCLimRaw_mp** displays the received ratio of torque. Additionally the renormalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done.

The received torque reduct is converted by offset **FrmMng_rTrqRedOfs_C** and **FrmMng_rTrqRedSlp_C**. The measurement point **FrmMng_rTSCRedRaw_mp** displays the received ratio of torque. Additionally the renormalisation with the norm torque **FrmMng_trqMaxEngNorm_C** is done. The torque reduct is subtracted from the base torque **FrmMng_trqEngForNCA**. During torque increase is active the torque reduct signal is disabled because of the Complement check of GEMO Message Counter.

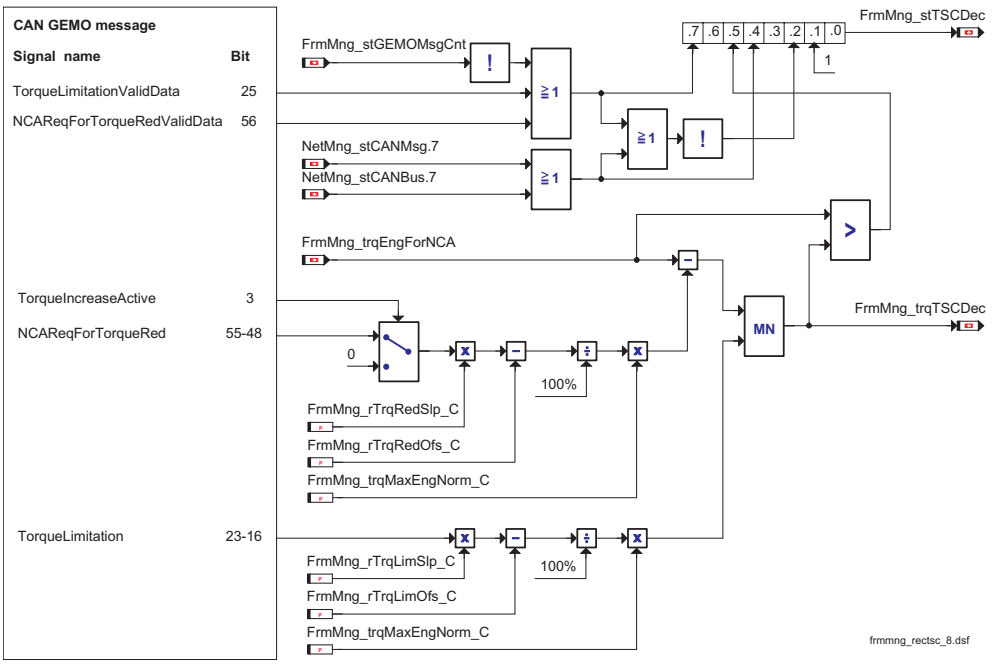
Afterwards the minimum of torque reduction and torque limitation is sent in message **FrmMng_trqTSCDec**. The torque decrease active bit in state **FrmMng_stTSCDec** is only set if **FrmMng_trqEngForNCA** is greater than **FrmMng_trqTSCDec**, therefore a real torque intervention takes place.

If any CAN message error or CAN bus error is present the torque increase state **FrmMng_stTSCDec** will be set accordingly.

Table 391: Gearbox Torque Decrease, state

FrmMng_stTSCDec	
Bit 1, FRMMNG_STGBXTRQ_MNMX	Gearbox Increase and Decrease allowed
Bit 2, FRMMNG_STGBXTRQ_NORAMP	Gearbox Decrease without ramp
Bit 3, FRMMNG_STGBXTRQ_LIMP	Gearbox in Limp Home Mode
Bit 4, FRMMNG_STGBXTRQ_CANOFF	Gearbox Decrease is affected due to CAN errors
Bit 5, FRMMNG_STGBXTRQ_DEMOFF	Gearbox Decrease is active
Bit 7, FRMMNG_STGBXTRQ_MSGERR	Gearbox Decrease is affected due to CAN message errors

Figure 1058 : Torque Decrease



Message GEARMOT2 ID: 0DBh, Data Length: 8 byte, Period: 10ms

The message GEARMOT2 is not supported.

ECU initialization

During initialization phase the ID, the Data Length and the Period are set according to the specification. The data bytes are taken from the default data set.

Output values

FrmMng_numGearDes: gear desire information from CAN [-] <UWORD> --->
FrmMng_stConv: conversion/clutch status from CAN [-] <UBYTE> --->
FrmMng_stGear: Gear information as given by Transmission Controller [-] <UWORD> --->
FrmMng_stGEMOMsgCnt: GEARMOT message counter status [-] <UBYTE> --->
FrmMng_stTSCDec: State of reduced torque demand from transmission shift control (TSC) [-] <UWORD> --->
FrmMng_stTSCInc: State of increased torque demand from transmission shift control (TSC) [-] <UWORD> --->
FrmMng_tGearbx: Gearbox temperature [deg C] <SWORD> --->
FrmMng_trqTSCDec: Reduced clutch torque demand from transmission shift control (TSC) [Nm] <SWORD> --->
FrmMng_trqTSCInc: Increased clutch torque demand from transmission shift control (TSC) [Nm] <SWORD> --->

Input values

CTSCD_tClnt: coolant temperature [deg C] <SWORD> --->
FrmMng_trqEngForNCA: Engine torque for gearbox node (without gearbox intervention) [Nm] <SWORD> --->
NetMng_stCANBus: State of CAN devices [-] <UBYTE> --->
NetMng_stCANMsg: State of CAN messages [-] <UWORD> --->

Measuring points

Dfp_FrmMng_Gearbx_mp: Errorpath for gearbox error [-] <UWORD> --->
FrmMng_ctGEMOMsgCnt0_mp: GEARMOT message counter (n) [-] <UBYTE> --->
FrmMng_ctGEMOMsgCnt1_mp: GEARMOT message counter (n-1) [-] <UBYTE> --->
FrmMng_ctGEMOMsgCnt2_mp: GEARMOT message counter (n-2) [-] <UBYTE> --->
FrmMng_ctGEMOMsgCnt3_mp: GEARMOT message counter (n-3) [-] <UBYTE> --->
FrmMng_ctGEMOMsgCntOff_mp: GEMO message counter , debouncing counter [-] <UBYTE> --->
FrmMng_numGearDesRaw_mp: gear desire information from CAN, raw value [-] <UWORD> --->
FrmMng_rTSCIncRaw_mp: Raw value of TSC increase ratio of torque [%] <SWORD> --->
FrmMng_rTSCLimRaw_mp: Raw value of TSC limitation ratio of torque [%] <SWORD> --->
FrmMng_rTSCRedRaw_mp: Raw value of TSC reduction ratio of torque [%] <SWORD> --->
FrmMng_stConvRaw_mp: conversion/clutch status from CAN, raw value [-] <UBYTE> --->
FrmMng_stGearRaw_mp: Gear information as given by Transmission Controller, raw value [-] <UWORD> --->
FrmMng_stGEMORaw0_mp: CAN message object GEARMOT byte 0, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw1_mp: CAN message object GEARMOT byte 1, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw2_mp: CAN message object GEARMOT byte 2, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw3_mp: CAN message object GEARMOT byte 3, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw4_mp: CAN message object GEARMOT byte 4, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw5_mp: CAN message object GEARMOT byte 5, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw6_mp: CAN message object GEARMOT byte 6, raw value [-] <UBYTE> --->
FrmMng_stGEMORaw7_mp: CAN message object GEARMOT byte 7, raw value [-] <UBYTE> --->

Application parameters

FrmMng_ctGEMOMsgCntMax_C: GEARMOT message counter, max threshold<value> --->
FrmMng_ctGEMOMsgCntMin_C: GEARMOT message counter, min threshold<value> --->
FrmMng_ctGEMOMsgCntOffDec_C: GEARMOT message counter debouncing, decrement<value> --->
FrmMng_ctGEMOMsgCntOffInc_C: GEARMOT message counter debouncing, increment<value> --->
FrmMng_ctGEMOMsgCntOffMax_C: GEARMOT message counter debouncing, max threshold<value> --->
FrmMng_DebGearbxEOBDDef_C: Defect debouncing of gearbox EOBD error<value> --->
FrmMng_DebGearbxEOBDOk_C: Intact debouncing of gearbox EOBD error<value> --->
FrmMng_rTrqIncOfs_C: Conversion of gearbox torque increase, offset<value> --->
FrmMng_rTrqIncSlp_C: Conversion of gearbox torque increase, slope<value> --->
FrmMng_rTrqLimOfs_C: Conversion of gearbox torque limitation, offset<value> --->
FrmMng_rTrqLimSlp_C: Conversion of gearbox torque limitation, slope<value> --->
FrmMng_rTrqRedOfs_C: Conversion of gearbox torque reduction, offset<value> --->
FrmMng_rTrqRedSlp_C: Conversion of gearbox torque reduction, slope<value> --->
FrmMng_stGEMODfl0_C: CAN message object GEARMOT byte0, default value<value> --->
FrmMng_stGEMODfl1_C: CAN message object GEARMOT byte1, default value<value> --->
FrmMng_stGEMODfl2_C: CAN message object GEARMOT byte2, default value<value> --->
FrmMng_stGEMODfl3_C: CAN message object GEARMOT byte3, default value<value> --->
FrmMng_stGEMODfl4_C: CAN message object GEARMOT byte4, default value<value> --->
FrmMng_stGEMODfl5_C: CAN message object GEARMOT byte5, default value<value> --->
FrmMng_stGEMODfl6_C: CAN message object GEARMOT byte6, default value<value> --->
FrmMng_stGEMODfl7_C: CAN message object GEARMOT byte7, default value<value> --->
FrmMng_stGEMOEna0_C: CAN message object GEARMOT byte 0, enable mask<value> --->
FrmMng_stGEMOEna1_C: CAN message object GEARMOT byte 1, enable mask<value> --->
FrmMng_stGEMOEna2_C: CAN message object GEARMOT byte 2, enable mask<value> --->
FrmMng_stGEMOEna3_C: CAN message object GEARMOT byte 3, enable mask<value> --->
FrmMng_stGEMOEna4_C: CAN message object GEARMOT byte 4, enable mask<value> --->
FrmMng_stGEMOEna5_C: CAN message object GEARMOT byte 5, enable mask<value> --->
FrmMng_stGEMOEna6_C: CAN message object GEARMOT byte 6, enable mask<value> --->
FrmMng_stGEMOEna7_C: CAN message object GEARMOT byte 7, enable mask<value> --->
FrmMng_stTrqConvA_C: A-threshold of torque converter information<value> --->
FrmMng_stTrqConvB_C: B-threshold of torque converter information<value> --->
FrmMng_tGearbxDef_CUR: Curve of default gearbox temperature<curve_individual> --->
FrmMng_tGearbxOfs_C: Conversion gearbox temperature, offset<value> --->
FrmMng_tGearbxSlp_C: Conversion gearbox temperature, slope<value> --->
FrmMng_trqMaxEngNorm_C: Maximum engine torque for normalization<value> --->

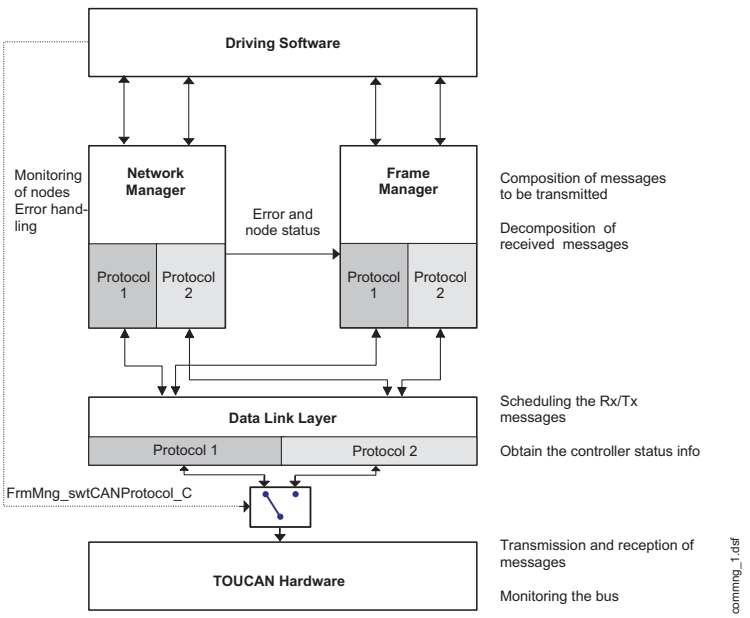
B.2 Communication Manager (ComMng)

Overview of the component

This document contains an overview of the Controller Area Network (CAN) implementation in the EDC16 environment.

Structure of the component

Figure 1059 : CAN overview



CAN Hardware

The Controller Area Network (CAN) protocol is implemented by two on-chip CAN controller (TouCAN). The CAN protocol is used for communication between different Electronic Control Units present on the vehicle and also for communication with external application tools and testers.

The TouCAN transmits and receives messages using the 16 message buffers upon request from the CPU. The communication between the TouCAN and the CPU is done through the TouCAN DPRAM. Apart from the message buffers, the control and status registers are also part of the DPRAM for the CPU to configure and get the status information from the TouCAN respectively.

The two TouCAN controllers can be used for 16 messages each or the two interfaces can be connected together to be used as a controller with 32 messages.

Switch between different CAN protocols

To switch between different CAN protocols there is a applicable software switch **FrmMng_swtCANProtocol_C** implemented in the software.

Table 392: Applicable CAN protocols

FrmMng_swtCANProtocol_C	CAN protocol
0	FLORENCE (default)
1	CAN4.11

CAN Data Link Layer

The CAN Data Link Layer is implemented as part of the Hardware Encapsulation (HWE). This provides an interface between the application software and the TouCAN controller. The following points explain in brief the important functionalities of the Data Link Layer.

Initialisation

The initialisation is done during the control unit initialisation before the normal runtime operation. The Data Link Layer will initialise the TouCAN registers and the message buffers of the TouCAN. It also initialises the global error and status variables.

Message scheduling

The TouCAN transmits or receives messages from the DPRAM depending on the schedule information. The Data lLnk Layer schedules the transmission or reception of a message by modifying the control register for that particular message. The Data Link Layer supports synchronous and asynchronous modes of transmission or reception.

Synchronous mode includes Time synchronous and engine speed synchronous. In the time synchronous mode, it is possible to transmit or receive messages in different time intervals provided the time interval of the message is a multiple of the time interval of the respective time synchronous mode.

Asynchronous message transmission or reception can be triggered by the user within the scheduling mode by an interface function call.

Error handling

The TouCAN constantly monitors the transmission and reception of messages. It updates the status information which reflects any error in transmission or reception of messages. The data link layer acts as an interface in providing the error information to the application software. The data link layer detects Bus Off error and also monitors the messages to check if the Key Messages have been timed out.

Data Link Layer Configuration

To cater to different requirements, certain data have to be configured accordingly. The configurable data includes

- The TouCAN controller related data like number of controllers, interrupt level for CAN device, baudrate for each controller etc.
- Message related data like identifiers, buffer number, data length, schedule time, time out period, direction of transfer (transmit message or receive message) etc.

Network Manager

The different control units in the vehicle are connected via CAN to form a network, with each control unit referred to as a node. Network Management involves monitoring all nodes in the network. The monitoring of nodes gives a picture about the reliability of the network at any given instant. Network Management involves monitoring of Own Node and also the Remote Nodes.

A message can be designated as a Key Message for the node. Monitoring the nodes consists of monitoring Key Messages from Own Node as well as the Remote Node. Monitoring of nodes includes checking for the timeout and data length of the corresponding Key Messages. Another functionality of the Network Manager is to monitor the Bus Off using the interface functions provided by the Data Link Layer.

For a detailed description of the network manager, refer to the network manager document.

Frame Manager

The messages to be transmitted or received are referred to as Frames in the application software. The Frame Manager composes the frames that are to be transmitted and decomposes the frames that are received. The information to be put into a frame is obtained from the application software as signals. The signals are configured to occupy the correct position in a frame. The signals are written into the image buffer from where the data link layer transfers it into the DPRAM for transmission. In case of received frames, the Frame Manager reads the information from the image buffer as various signals. The signals are configured to give the position from where to pickup the information. The Frame Manager processes the signals and makes them available for the driving software.

The Frame Manager also does the conversion of the signals from internal control unit resolution to the CAN resolution for signals to be transmitted and vice-versa for signals received. The signal value limitation is also part of the Frame Manager.

For a detailed description of the Frame Manager refer to the Frame Manager document.