CAN Network Management Requirement Specification

Version 2.0.0

<Confidential>

GEELY Automobile Research Institute

2014-7

Issue status (Index)	**Document maturity** (Planned / Draft Released)	**Compiled by** Department Name Date	**Review by** Department Name Date	**Approval by** Department Name Date	**Description**
0.1	Planned	E/E Division GEELY R&D IVN Department Cai Weijie Date:2013/01/22			Creation Base on GEELY-NL_HL_BCAN_Network_Specification_v3.0.1.doc
1.0	Draft	E/E Division GEELY R&D IVN Department Cai Weijie Date:2013/05/30			
1.1	Draft	E/E Division GEELY R&D IVN Department Cai Weijie Zhang Hua Date:2013/07/15			
1.2	Draft	E/E Division GEELY R&D IVN Department Cai Weijie Zhang Hua Date:2013/07/15			Update after 1st internal review Update according to the communication voltage meeting with CIXI
1.3	Released	E/E Division GEELY R&D IVN Department Cai Weijie Date:2013/08/29	E/E Division GEELY R&D IVN Department Liu Tong Zhang Hua Meng Na Date:2013/08/29	E/E Division GEELY R&D IVN Department Xiong Xiangtao Date:2013/08/29	
1.3.1	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/09/2	E/E Division GEELY R&D IVN Department Weijie Cai Date:2013/09/3	E/E Division GEELY R&D IVN Department Xiangtao Xiong Date:2013/09/10	Modified the type of message Rx/Tx in network mode. Defined the T_WakeUp Redefined the Start NM Time Deleted the Repeat message concept Added the immediate transmission mechanism Modified the value of the T_REPEAT_MESSAGE
1.3.2	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/09/16	E/E Division GEELY R&D IVN Department Weijie Cai Date:2013/09/17	E/E Division GEELY R&D IVN Department Xiangtao Xiong Date:2013/09/18	Added the NM of fault operation Modified the Bus-Off handling strategy

1.3.3	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/10/7	E/E Division GEELY R&D IVN Department Weijie Cai Date:2013/10/7	E/E Division GEELY R&D IVN Department Xiangtao Xiong Date:2013/10/8	Corrected the Ready Sleep State network behavior
1.4.0	Released	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/10/14	E/E Division GEELY R&D IVN Department Weijie Cai Date:2013/10/14	E/E Division GEELY R&D IVN Department Xiangtao Xiong Date:2013/10/15	Added the other nodes NM ID The version is released
1.4.1	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/11/15			Modified the Bus-Off handling and recovery strategy Refining the NM Repeat Message Request Cleared the user data definition
1.4.2	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2013/12/15			Added the Rolling Counter Mechanism Added the Checksum Mechanism
1.4.3	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2014/1/3			Refined the Condition 9 definition in NM Modified the value of cEorrorChecksum Modified the value of vDiff and cErrorRoll Added the Handling of the Rolling Counter Error Changed the tStartupTimeout definition
1.5.0	Released	E/E Division GEELY R&D IVN Department Hua Zhang Date:2014/1/17	E/E Division GEELY R&D IVN Department Jianglei Wang Weijie Cai Weikang Ying MN Tong Liu Lijun Fu Date:2014/2/10	E/E Division GEELY R&D IVN Department Xiangtao Xiong Wujun Ding Date:2014/2/13	This version is released
1.5.1	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2014/5/17			
1.5.2	Draft	E/E Division GEELY R&D IVN Department Hua Zhang Date:2014/7/3			Deleted the Diagnostic for NM

2.0.0	Released	E/E Division GEELY R&D IVN Department Hua Zhang Date:2014/7/7	E/E Division GEELY R&D IVN Department Jianglei Wang Weijie Cai MN Tong Liu Date:2014/7/7	E/E Division GEELY R&D IVN Department Xiangtao Xiong Wujun Ding Date:2014/7/8	This version is released for NL-3

Change Log

Difference between V1.0 and V1.1

Section	Change Description
3	-Re-defined NM strategy
4.1	-The caption of "GENERAL FAILURE MODE OPERATIONS REQIUREMENTS" is revised to "GENERAL REQUIREMENTS"
4.2	-The caption of "CAN APPLICATION FRAME TRAMSMISSION-DEFAULT VALUE" is revised to "COMMUNICATION DEFAULT VALUE"
4.2	-Delete the content of "Secured RAM Storage" relevant description
4.3	-The caption of "CAN Application Frame Transmission – Lost communication Detection and healing" is revised to "Lost communication Detection and healing"
4.3.3	-Add the caption of "Lost Communication Self Diagnose"
4.4	-The caption of "CAN APPLICATION FRAME TRANSMISSION AND RECEPTION–UNDER/OVER VOLTAGE" is revised to "COMMUNICATION UNDER/OVER VOLTAGE"
4.4.1	-The caption of "CAN Application Frame Transmission – Under/Over voltage" is revised to "Communication Under/Over Voltage Strategy"
4.4.1	-The parameter "uVoltageTxValid" is revised to "uVoltageValid"
4.4.2	-Delete the content of "CAN Application Frame Reception – Under/Over voltage, because the content of this section has been already described in section 4.4.1
4.4.3	-Delete the content of "Under/Over voltage DTC(DTC_Tx_Stop) set strategy",
4.4.2	-Add new section 4.4.2 "Communication Under/Over Voltage Self Diagnosis"
4.5	-The caption of "CAN Application Frame Transmission – Bus-Off" is revised to "CAN BUS-OFF Handling"
4.5.1	-This section is revised to "CAN Application Frame Transmission – Bus-Off"
4.5.2	-This section is revised from "4.6 CAN BUS-OFF Handling And Recovery"
4.5.3	-Add new section 4.5.3 "CAN BUS-OFF Self Diagnosis"

Difference between V1.1 and V1.2

Section	Change Description
4.4.1	- uVoltageComDTCLog minimum value is revised from 8v to 9v -delete parameter tVoltageErrTxStop and tVoltageErrTxRecovery in table 4.4.1-2

Difference between V1.2 and V1.3

Section	Change Description
4.5.2	- Table 4.5.2-2，revised the value of tBusOffRetry from 20ms to 50ms
3.4	- Table 3.4-3, add NM ECU address for IHU and TCM

Difference between V1.3.1 and V1.3

Section	Change Description
3.2.3	-Distinguished the type of message Rx/Tx in network mode
3.2.3.3	-Added the WakeUp Time Definition
3.2.3.3.1	-Redefined the Start NM Time Definition -Deleted the NM Repeat Message-Sequence Definition -Added the immediate transmission mechanism
3.2.3.3.2	-Deleted the NM Normal Message-Sequence Definition -Deleted the Start-Normal NM Time Definition
3.3	-Added the explain of Active Wakeup Bit
3.5	-Changed the meaning of the T_START_NM_TX -Defined the T_WakeUp time for all NM node -Deleted the T_RepeatMessageCycle -Deleted the T_START_NormalNM_TX -Added the T_NM_ImmediateCycleTime

	-Changed the T_NormalMessageCycle to T_NM_MessageCycle -Changed the value of T_REPEAT_MESSAGE from 100ms to 1500ms. -Added the N_ImmediateNM_TIMES for immediate message mechanism.

Difference between V1.3.2 and V1.3.1

Section	Change Description
3.8	-Added the NM of fault operation
4.5	-Modified the Bus-Off handling strategy -Changed the parameter of tBusOffRetry value from 50ms to 100ms

Difference between V1.3.3 and V1.3.2

Section	Change Description
3.2.3.3.3	- Redefined the behavior of Repeat Message Request in the Ready Sleep State

Difference between V1.4.0 and V1.3.3

Section	Change Description
3.4	- Added the NM ID for ABS/ESC, GSM, TCU, GW-Body, GW-Dynamic, GW-Propulsion, GW-Infotainment and GW-Diagnostic.
All	-The version is released

Difference between V1.4.1 and V1.4.0

Section	Change Description
3.2.4	-Refining the NM Repeat Message Request.
3.3	-Cleared the user data 0 and first bit in user data 1 definition.
4.5	-Modified the Bus-Off handling and recovery strategy. -Changed the DTC_BUS_OFF recording strategy.

Difference between V1.4.2 and V1.4.1

Section	Change Description
4.6	-Added the Checksum Mechanism
4.7	-Added the Rolling Counter Mechanism

Difference between V1.4.3 and V1.4.2

Section	Change Description
3.2.4	- Refined the Condition 9: deleted the Bus-Off recovery successful condition as the NM Repeat Message Request.
4.6.1	- Modified the value of cEorrorChecksum from 128 to 10 in Algorithm of Checksum chapter.
4.3.2	- Changed the tStartupTimeout definition.
4.7.1	- Modified the value of vDiff from 2 to 1 in Mechanism of Rolling Counter chapter. - Modified the value of cErrorRoll from 128 to 10 in Mechanism of Rolling Counter chapter. - Added the strategy for Rolling Counter Error.

Difference between V1.5.0 and V1.4.3

Section	Change Description
All	-The version is released

Difference between V1.5.1 and V1.5.0

Section	Change Description
	-Added the Diagnostic requirement
	-Added the T_STARTx_AppFrame timer
	- Added the eCALL NM ID
	~~-Deleted the uVoltageComDTCLog~~
	-Changed the T_WakeUp time

	-Changed the tBusOffDetectionEnable definition

Difference between V1.5.2 and V1.5.1

Section	Change Description
All	-Deleted the diagnostic requirement for NM

Difference between V2.0.0 and V1.5.2

Section	Change Description
All	-This version is released for NL-3

Table of Contents

1. GENERAL

1.1 RELATED DOCUMENT

1. ISO 11898-1(2003) Controller area network (CAN) – Part 1: Data link layer and physical signalling
2. ISO 11898-2(2003) Controller area network (CAN) – Part 2: High-speed medium access unit
3. Road vehicles ISO-15765 – Diagnostics on Controller Area Network (CAN) — Part 2: Network layer services
4. Road vehicles ISO-15765 – Diagnostics on Controller Area Network (CAN) — Part 4: Requirements for emissions-related systems, 2005-01-15 edition.
5. Road vehicles ISO 16845 – Controller Area Network (CAN) Conformance Test Plan (2004)
6. ISO 14229 (2006) Road vehicles—Unified diagnostic services(UDS)—Specification and requirements
7. SAE j2012 – RECOMMENDED PRACTICE FOR DIAGNOSTIC TROUBLE CODE DEFINITIONS
8. GEELY In Vehicle CAN Network Matrix
9. GEELY In Vehicle CAN Bus Diagnostic Requirement Specification
10. GEELY In Vehicle Gateway Functional Requirement Specification
11. GEELY In Vehicle LIN Network Protocol
12. GEELY program In Vehicle Network topology
13. SAE 2284-3 High Speed CAN for Vehicle Applications at 500 KBPS, issued 2002-03
14. ISO 15031-3: Road vehicles – Communication between vehicle and external equipment for emission-related diagnostics – Part 3: Diagnostic connector and related electrical circuits, specification and use
15. GEELY CAN Communication Requirement Specification

1.2 ACRONYMS AND ABBREVIATIONS

Abbr.	Description
HS-CAN	High Speed (@500kbps baud rate) Controller Area Network
ABS	Anti-block Braking System
ACU	Airbag Control Unit
BCM	Body Control Module
IGW	Integrated Gateway
EMS	Engine Control System
ESC	Electronic Stability Controller
ICU	Instrument Control Unit
TCU	Transmission Control Unit
HVAC	Heating, Ventilation, Air-Conditioning

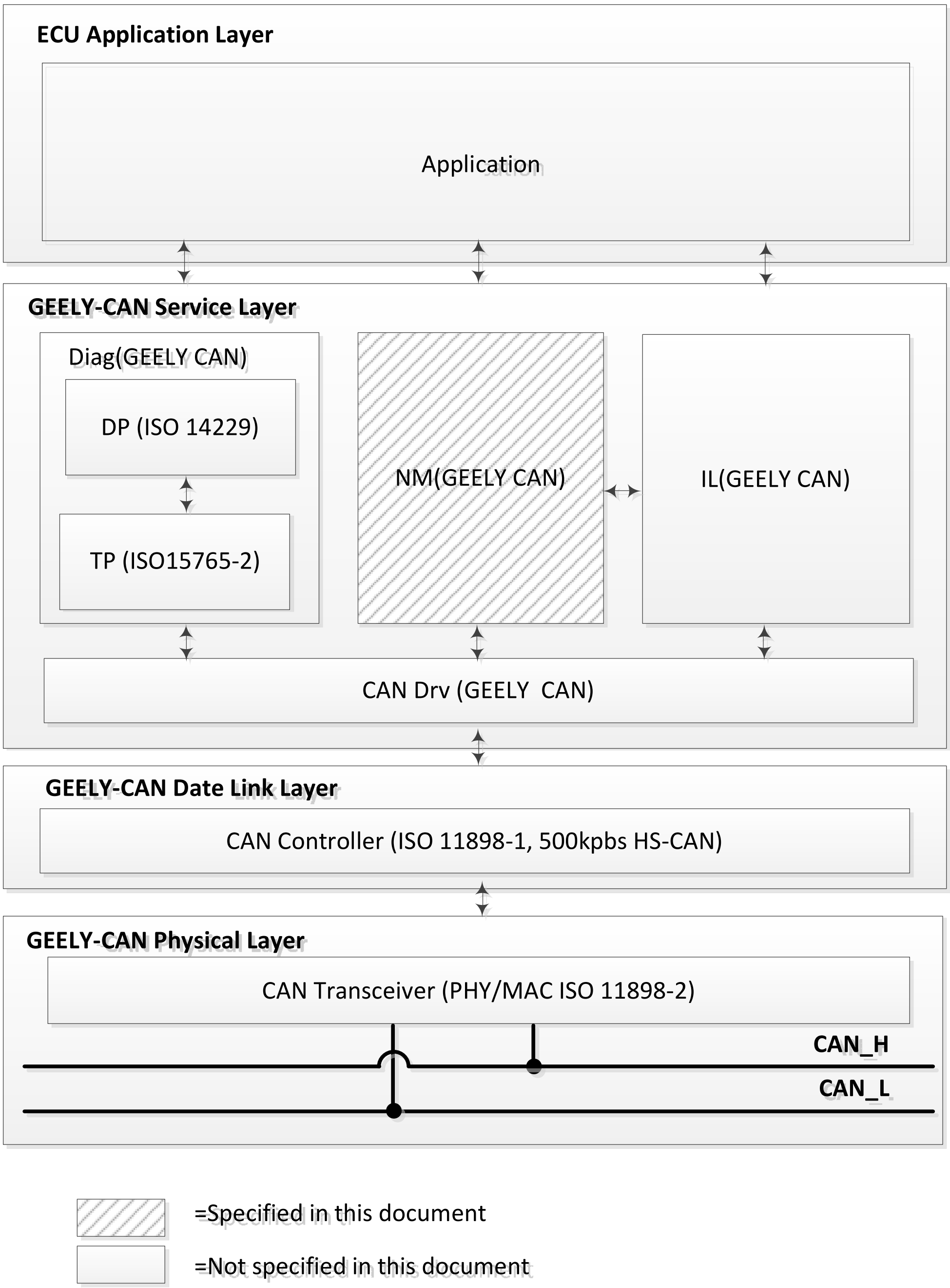
Abbr.	Description
SJW	Synchronization Jump Width
NM	Network Management
PoR	Power On Reset
Tx	Transmission
Rx	Reception
NRZ	None Return To Zero
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
4WD	4 Wheel Drive
TP	Transport Protocol
DP	Diagnostic Protocol
IL	Interaction Layer
GW	Gateway
CMX	CAN Matrix
PDU	Protocol Data Unit
GRI	GEELY Research Institute
CVB	Control Bit Vector in NM-message
OEM	Original Equipment Manufacturer
DTC	Diagnostic Trouble Code
STB	Standby mode control input
……	

2. OBJECTIVE OF SPECIFICATION

This specification defines the CAN **Network Management** requirements of the in-vehicle CAN communication system of GEELY vehicle platform, see GEELY CAN communication stack architecture Figure 2.

Requirements defined hereafter are common to all ECUs connecting to the in-vehicle CAN communication system of GEELY vehicle platform. Exceptions to this shall be explicitly noted and approved by GRI officially.

Note: ECU specific functional requirements are not in the scope of this specification.



Figures 2 GEELY CAN communication stack architecture

3. NETWORK MANAGEMENT

3.1 NETWORK MANAGEMENT OVERVIEW

This chapter defines the Non-ignition switched Nodes Network Management requirements of GEELY vehicle platform.

Non-Ignition switched nodes will still require CAN bus communication after ignition switched OFF.

The Non-ignition switched Node implements a same NM protocol for a system of control units connected through a CAN network.

The GEELY CAN Network Management is a hardware independent protocol that can only be used on CAN. Its main purpose is to coordinate the transition between normal operation and bus-sleep mode of the network.

3.2 NETWORK MANAGEMENT FUNCTIONAL

3.2.1 NM Functional Algorithm

The GEELY NM is based on decentralized direct network management strategy, which means that every network node performs activities self-sufficient depending on the NM PDUs only that are received or transmitted within the communication system.

The NM algorithm is based on periodic NM PDUs, which are received by all nodes in the cluster via broadcast transmission. Reception of NM PDUs, indicates that sending nodes want to keep the network management cluster awake. If any node is ready to go to the Bus Sleep Mode, it stops sending NM PDUs, but as long as NM PDUs from other nodes are received, it postpones transition to the Bus Sleep Mode. Finally, if a dedicated timer elapses because no NM PDUs are received anymore, every node initiates transition to the Bus Sleep Mode.

If any node in the network management cluster requires bus communication, it can wake up the network management cluster from the Bus Sleep Mode by transmitting NM PDUs.

The main concept of the GEELY NM algorithm can be defined by the following two key-requirements:

- Every network node in a NM cluster shall transmit periodic NM PDUs as long as it requires bus-communication; otherwise it shall transmit no NM PDUs.
- If bus communication in a NM cluster is released and there are no NM PDUs on the bus for a amount of time determined by T_NM_TIMEROUT + T_WAIT_BUS_SLEEP transition into the Bus Sleep Mode shall be performed.

3.2.2 NM Transition Diagram

The network management state chart diagram was defined in this section. Refer to in Figure 3.2.2-1.

3.2.3 NM Mode Description

The GEELY network management shall contain three modes:

- Bus Sleep Mode
- Prepare Bus Sleep Mode
- Network Mode

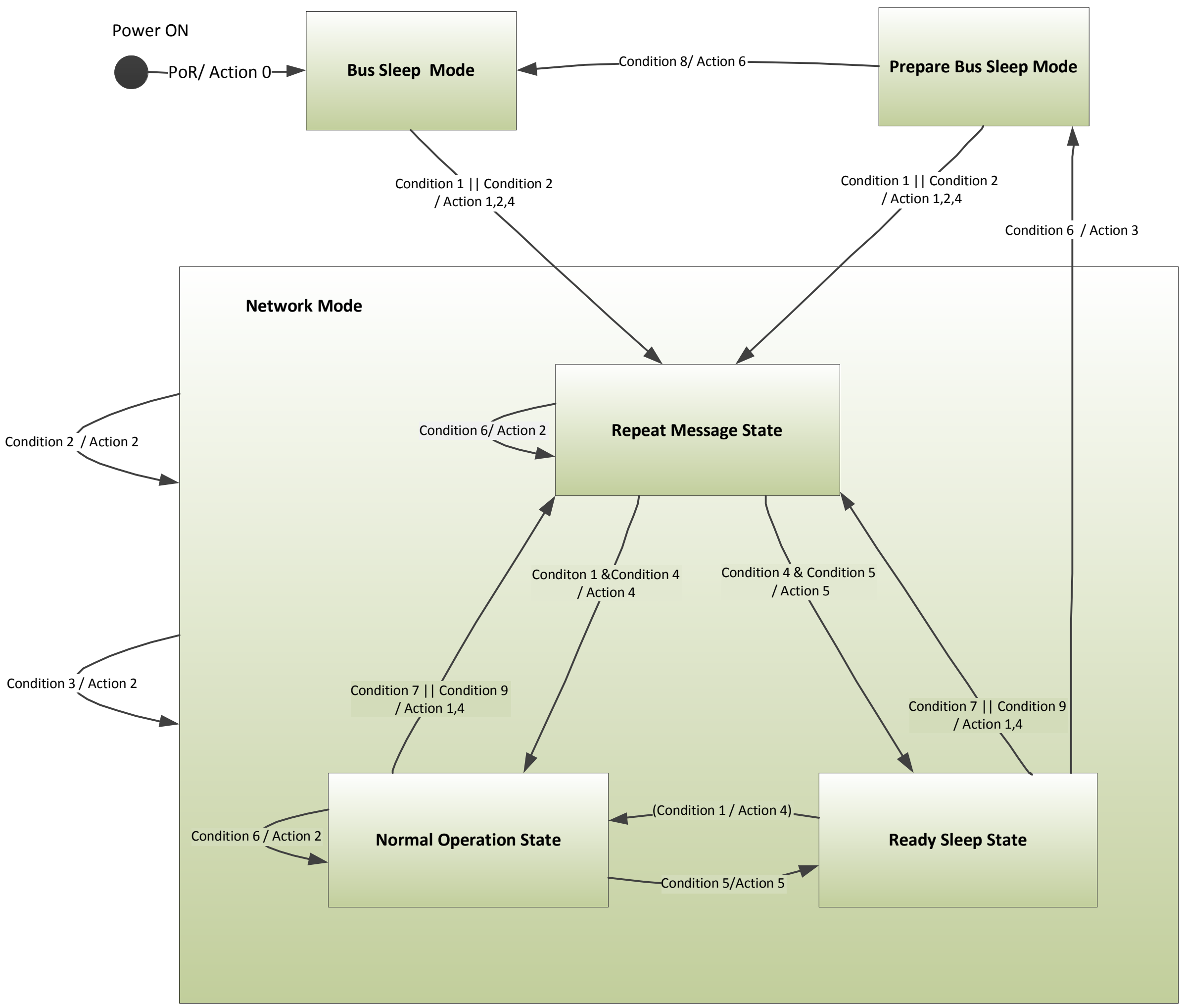


Figure 3.2.2-1.GEELY NM Transition Diagram

The table 3.2.3-1 describes to send different type of message in different mode of network:

NM Mode		NM Frame		App Frame	
		Tx	Rx	Tx	Rx
Bus Sleep Mode		N	Y	N	N
Prepare Bus Sleep Mode		N	Y	—	—
Network Mode	Repeat Message State	Y	Y	Y	Y
	Normal Operation State	Y	Y	Y	Y
	Ready Sleep State	N	Y	Y	Y
'N' denote that frame of Tx/Rx is impossible. 'Y' denote that frame of Tx/Rx is possible. '—' denote that frame of Tx/Rx shall not prohibit. App frames include application messages, diagnosis messages, calibration messages;					

Table 3.2.3-1 different type of message in modes of network

3.2.3.1 Bus Sleep Mode

The purpose of the Bus Sleep Mode is to reduce power consumption in the node when no messages are to be exchanged. The communication controller is switched into the sleep mode, respective wakeup mechanisms are activated and finally power consumption is reduced to the adequate level in the Bus Sleep Mode.

3.2.3.2 Prepare Bus Sleep Mode

The purpose of the Prepare Bus Sleep Mode is to ensure that all nodes have time to stop their network activity before the Bus Sleep Mode is entered. In Prepare Bus Sleep Mode the bus activity is calmed down (i.e. queued messages are transmitted in order to make all Tx-buffers empty) and finally there is no activity on the bus in the Prepare Bus-Sleep Mode.

When the NM module is entered in the Prepare Bus Sleep Mode and start immediately T_WAIT_BUS_SLEEP timer, after that time the Prepare Bus Sleep Mode shall be left and the Bus Sleep Mode shall be entered.

At successful reception of a NM PDU in the Prepare Bus Sleep Mode, the NM Module shall enter the Network Mode; by default the NM Module shall enter the Repeat Message State.

When the network is requested in the Prepare Bus Sleep Mode, the NM module shall immediately enter the Network Mode; by default the NM Module shall enter the Repeat Message State.

3.2.3.3 Network Mode

The Network Mode shall consist of three internal states:

- Repeat Message State
- Normal Operation State
- Ready Sleep State

When the Network Mode is entered from Bus Sleep Mode, by default, the NM module shall enter the Repeat Message State.

When the Network Mode is entered from Prepare Bus Sleep Mode, by default, the NM module shall enter the Repeat Message State.

When the Network Mode is entered, the NM module shall start the T_NM_TIMEROUT Timer.

At successful reception of a NM PDU in the Network Mode, the NM module shall restart the T_NM_TIMEROUT Timer.

At successful transmission of a NM PDU in the Network Mode, the NM module shall restart the T_NM_TIMEROUT Timer.

The NM module shall reset the T_NM_TIMEROUT Timer every time it is started or restarted.

- Wake Up Time Definition

When the sleeping ECU was waked up by local conditions or received NM PDU, ECU will enter Network Mode, by default the NM Module shall enter the Repeat Message State, and start sending the first NM PDU. This process should be complete within specified time that is denoted by ‘t’, where t<=T_WakeUp.

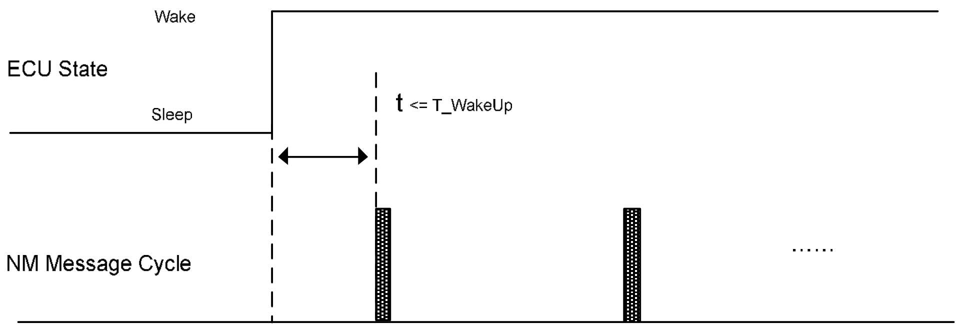


Figure 3.2.3.3-1 Wake Up time definition

3.2.3.3.1 Repeat Message State

The Repeat Message State ensures that any transition from Bus Sleep Mode or Prepare Bus Sleep to the Network Mode becomes visible to the other nodes on the network. Additionally it ensures that any node stays active for a minimum amount of time. It can be used for detection of present nodes.

When the Repeat Message State is entered from Bus Sleep Mode, Prepare Bus Sleep Mode, Normal Operation State or Ready Sleep State, the NM module shall (re-)start transmission of NM PDUs.

When the T_NM_TIMEROUT Timer expires in the Repeat Message State, the NM module shall start the T_NM_TIMEROUT Timer.

The network management state machine shall stay in the Repeat Message State for the amount of time determined by the T_REPEAT_MESSAGE; after that time the NM module shall leave the Repeat Message State.

When Repeat Message State is left and if the network has been requested, the NM module shall enter the Normal Operation State.

When Repeat Message State is left and if the network has been released, the NM module shall enter the Ready Sleep State.

When Repeat Message State is left, the NM module shall clear the Repeat Message Bit.

- **Immediate Transmission Mechanism**

The immediate transmission mechanism is used for nodes which want to immediately wake up the bus or restore communication as fast as possible.

If the Repeat Message State is entered via Network Request or Repeat Message Request, the required number of immediate NM PDUs transmissions shall be started immediately. In other cases, entering the repeat message state caused by successfully received NM PDUs or received the Repeat Message Request Bit Indication, the NM message shall be sent by T_NM_MessageCycle after entering the Repeat Message State. This can be illustrated by Figure 3.2.3.3.1-1, figure 3.2.3.3.1-2.

When entering the Repeat Message State from Bus Sleep State or Prepare Bus Sleep State because of Network Request, the NM PDUs shall be transmitted using T_NM_ImmediateCycleTime as cycle time. The transmission of the first NM PDU shall be triggered as soon as possible. After the transmission the Message Cycle Timer shall be reloaded with T_NM_ImmediateCycleTime. The T_NM_MessageCycle shall not be applied in this case.

When entering the Repeat Message State from Normal Operation State or Ready Sleep State because of Repeat Message Request, the NM PDUs shall be transmitted using T_NM_ImmediateCycleTime as cycle time. The transmission of the first NM PDU shall be triggered as soon as possible. After the transmission the Message Cycle Timer shall be reloaded with T_NM_ImmediateCycleTime. The T_NM_MessageCycle shall not be applied in this case.

The number of NM PDUs transmitted with the cycle time T_NM_ImmediateCycleTime is defined N_ImmediateNM_TIMES. After all immediate NM PDUs have been transmitted the NM shall start transmission using the cycle time T_NM_MessageCycle.

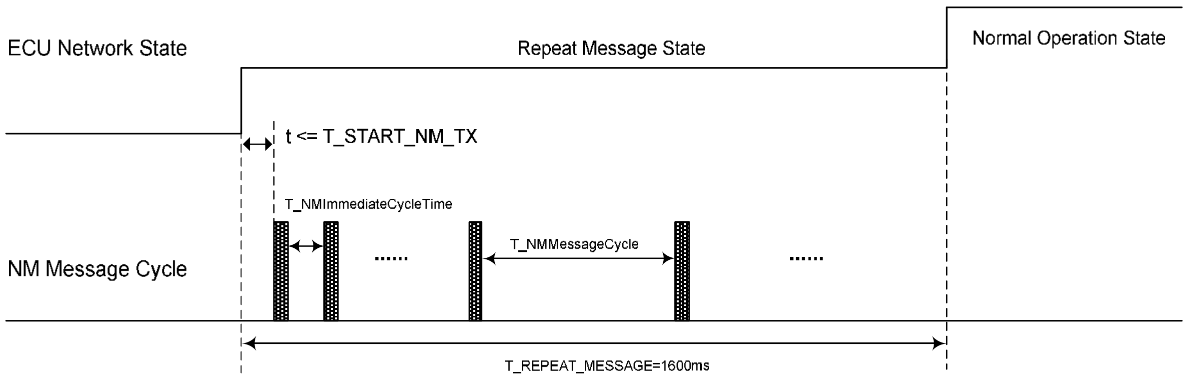


Figure 3.2.3.3.1-1 NM immediate message cycle definition

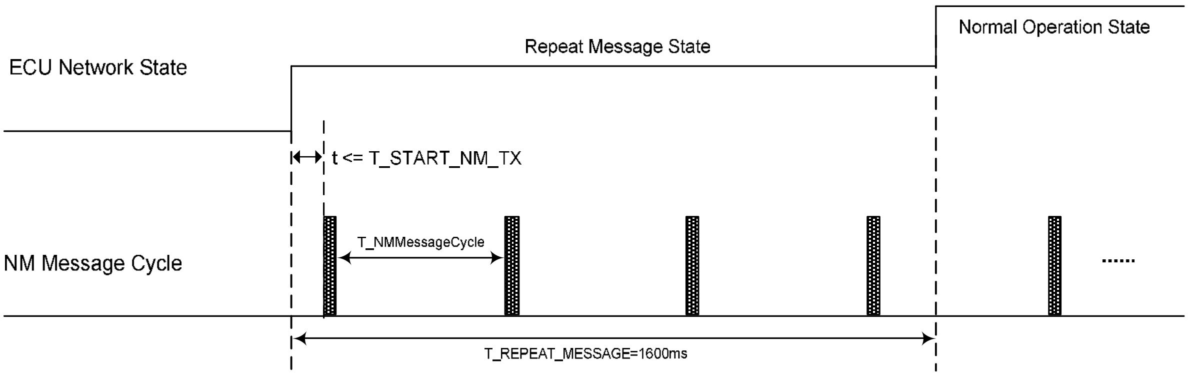


Figure 3.2.3.3.1-2 NM message cycle definition

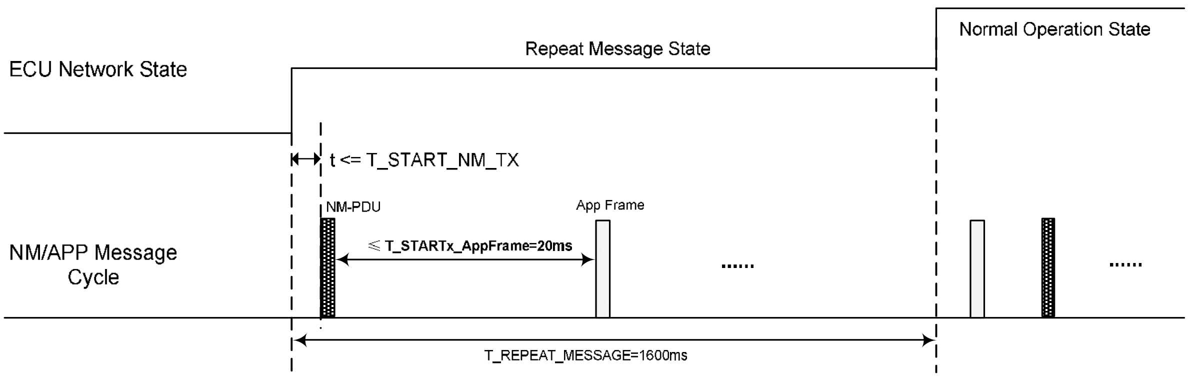


Figure 3.2.3.3.1-3 T_STARTx_AppFrame definition

- Start NM Time Definition

When the Repeat Message State is entered from Bus Sleep Mode, Prepare Bus Sleep Mode, Normal Operation State or Ready Sleep State, the first NM PDUs should be transmitted within ‘t’, where the t<=T_START_NM_TX.

T_REPEAT_MESSAGE is a minimum amount of time during which other nodes on CAN bus can be waked up by receiving NM frame. After this timer elapses, node will transit to Normal Operation State or Ready Sleep State.

CAN application frame should be sent within the T_STARTx_AppFrame in this state after the first NM frame was successfully sent. It can be illustrated by Figure 3.2.3.3.1-3.

3.2.3.3.2 Normal Operation State

The Normal Operation State ensures that any node can keep the network management cluster awake as long as the network is requested.

When the Normal Operation State is entered from Repeat Message State or Ready Sleep State, the NM module shall start transmission of NM PDUs with T_NM_MessageCycle Period.

When the T_NM_TIMEROUT Timer expires in the Normal Operation State, the NM module shall start the T_NM_TIMEROUT Timer.

When the network is released and the current state is Normal Operation State, the NM module shall enter the Ready Sleep State.

At Repeat Message Request Bit Indication in the Normal Operation State, the NM module shall enter the Repeat Message State.

At Repeat Message Request in the Normal Operation State, the NM module shall enter the Repeat Message State.

At Repeat Message Request in the Normal Operation State, the NM module shall set the Repeat Message Bit and start the immediate mechanism.

3.2.3.3.3 Ready Sleep State

The Ready Sleep State ensures that any node in the network management cluster waits with transition to the Prepare Bus Sleep Mode as long as any other node keeps the network management cluster awake.

When the Ready Sleep State is entered from Repeat Message State or Normal Operation State, the NM module shall stop transmission of NM PDUs.

When the T_NM_TIMEROUT Timer expires in the Ready Sleep State, the NM module shall enter the Prepare Bus Sleep Mode.

When the network is requested and the current state is the Ready Sleep State, the NM module shall enter Normal Operation State.

At Repeat Message Request Bit Indication in the Ready Sleep State, the NM module shall enter the Repeat Message State.

At Repeat Message Request in the Ready Sleep State, the NM module shall enter the Repeat Message State.

At Repeat Message Request in the Ready Sleep State, the NM module shall set the Repeat Message Bit and start the immediate mechanism.

3.2.4 NM State Transitions

NM State Transition Conditions:

Condition No.	Description
Condition 1	Network Requested
Condition 2	Successfully Received NM PDU.
Condition 3	Successfully Transmit NM PDU.
Condition 4	T_REPEAT_MESSAGE timer has expired.
Condition 5	Network Released.(e.g. local node sleep condition is true)
Condition 6	T_NM_TIMEROUT has expired.
Condition 7	Repeat Message Bit Received.
Condition 8	T_WAIT_BUS_SLEEP timer has expired.
Condition 9	NM Repeat Message Request. (e.g. Master node request)

Table 3.2.4-1 NM Transition Conditions

NM State Transition Actions:

Action No.	Description
Action 0	Initialization Of NM
Action 1	Start T_REPEAT_MESSAGE Timer.
Action 2	Start T_NM_TIMEROUT Timer.
Action 3	Start T_WAIT_BUS_SLEEP Timer.
Action 4	Node Transmitting NM PDU.
Action 5	Stop Transmit NM PDU
Action 6	Go to Sleep Mode

Table 3.2.4-2 NM Transition Actions

3.3 NM CONTROL FRAME DATA FORMAT

The figure below shows the default format of the NM frame:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	ECU Address							
Byte 1	Control Bit Vector							
Byte 2	User data 0							
Byte 3	User data 1							
Byte 4	User data 2							
Byte 5	User data 3							
Byte 6	User data 4							
Byte 7	User data 5							

Figure 3.3-1: NM Control Frame Data Format

The user data 0 is used to wake up reason, i.e. the network request conditions, that are defined in ECU specification.

The first bit in user data 1 is used to indicate network management state. It's only used to test.

The figure below describes the format of the Control Bit Vector:

Byte 1	Control Bit Vector	Interpretation
bit 0	Repeat Message Request	0: Repeat Message State not requested 1: Repeat Message State requested
bit 1	Res	
bit 2	Res	
bit 3	Res	
bit 4	Active Wakeup Bit	0: Node has not woken up the network (passive wakeup) 1: Node has woken up the network (active Wakeup)
bit 5	Res	
bit 6	Res	
bit 7	Res	

Figure 3.3-2: the format of the Control Bit Vector

When entering the Repeat Message State from Bus Sleep State or Prepare Bus Sleep State because of Network Request (generally local wake up conditions), the NM module should be set Active Wakeup Bit to value of one.

3.4 NM CONTROL FRAME IDENTIFIER

CAN Identifier range from 0x400 ~ 0x4FF, are used for ECU network management control frames as shown in Table 3.4-1.

Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NM Control Frame Base Address			ECU Address							

Table 3.4 -1 NM Control Frame CAN ID Design

Formula: NM Message ID = NM Control Frame Base Address + ECU Address.

All participants of network management ECU must follow this formula to monitor NM message ID, in order to develop a network management platform of GEELY.

NM Control Frame base address:

NM Control Frame Base Address	Description
0x400	NM CAN ID started from this address

Table 3.4 -2 NM Control Frame Base Address

NM ECU Address of GEELY CAN has been defined as in Table 3.4-3.

ECU Name	ECU Address	
Domain	*Bin*	*Hex*
BCM	0000 0001	1
ICU	0000 0010	2
PEPS	0000 0011	3
ESCL	0000 0100	4
GW-Body	0000 0101	5
HVAC	0000 0110	6
IHU(GNL)	0000 0111	7
TCM	0000 1000	8
ABS/ESC	0000 1001	9
GSM	0000 1010	A
TCU	0000 1011	B
GW- Dynamic	0000 1100	C
GW- Propulsion	0000 1101	D
GW-Infotainment	0000 1110	E
GW-Diagnostic	0000 1111	F
eCALL	0001 0000	10
Diag-Tester	1010 1010	AA
Reserved	Reserved	Reserved

Table 3.4-3 GEELY CAN Bus NM ECU Address

3.5 NM TIMING CONTROL PARAMETER

NM time control parameters are defined as in Table 3.5-1.

Timer name	Value	Tolerance	Description
T_REPEAT_MESSAGE	1600ms	+/-10%	Timer for a node becomes visible to the other nodes on the network.
T_NM_TIMEOUT	2000ms	+/-10%	As long as the node enter the network mode and start this timer. When the timer is expired, the node will enter the Prepare Bus Sleep Mode.

T_WAIT_BUS_SLEEP	2000ms	+/-10%	The timer is to ensure that all nodes have time to stop their network activity.
T_START_NM_TX	10ms	+/-10%	The time describes all sent NM message behaviour that the NM node enters network mode and start to transmit the first NM PDU. This time may be zero millisecond, one millisecond, two milliseconds and so on, but ten milliseconds is maximum time.
T_STARTx_AppFrame	20ms	+/-10%	The maximum interval time starts sending application message after send the first NM-PDU successfully.
T_NM_ImmediateCycleTime	20ms	+/-10%	In generally, network request of node should be trigger the immediate transmission mechanism and NM-PDU will be transmitted with this periodic in Repeat Message State.
T_NM_MessageCycle	500ms	+/-10%	The interval time is every two NM frames which are transmitted in Network Mode except Ready Sleep State.
T_WakeUp	100ms	+/-10%	The value is max time that the node from the Sleep Mode to Network Mode and send the first NM PDU, generally transmit in Repeat Message State.
N_ImmediateNM_TIMES	5		The number of NM PDUs transmitted with the cycle time T_NM_ImmediateCycleTime in Repeat Message Mode.

Table 3.5-1 GEELY CAN NM control timing design

3.6 NETWORK REQUEST CONDITION

The specific request conditions of ECUs must be discussed with supplier and defined in ECU specification during the process of development of network functions.

3.7 NETWORK RELEASE CONDITION

The specific release conditions of ECUs must be discussed with supplier and defined in ECU specification during the process of development of network functions.

3.8 NM OF FAULT OPERATION

NM on a node which is or become bus unavailable shall have a deterministic behaviour. NM on a node which is or become bus unavailable shall react such that:

- If a bus becomes unavailable and the node is not ready to sleep, the NM shall not enter bus sleep mode by itself.
- If a bus becomes unavailable and the node is ready to sleep, the NM shall enter bus sleep mode by itself.
- If a bus is unavailable and the node changes its state to ready to sleep, the NM shall enter bus sleep mode by itself.
- If a bus is unavailable and the node changes its state to not ready to sleep, the NM shall not enter bus sleep mode by itself.

The four rules in the description will make sure that the NM of a node that is currently not in bus sleep mode will never enter bus sleep mode while the node itself is not ready to sleep. If the node itself is ready to sleep, the NM shall enter bus sleep mode on its own.

NM of fault operation does not apply for a node that is already in bus sleep mode. In addition, bus unavailability may be hard to check at that time since the bus is not used to communicate in bus sleep mode.

4. FAILURE MODE OPERATIONS

4.1 GENERAL REQUIREMENTS

After Node ECU reset, all CAN transmission and reception of this node shall be disabled until CAN related data initialization is completed (e.g. ECU Network active state entered).

4.2 COMMUNICATION DEFAULT VALUE

For signal reception, default value shall be designed for each signal to be received from CAN network in case physical value is not valid. Default values shall be stored in ROM storage of CAN ECUs. Two kinds of default value shall be defined to ensure the signal transmission:

- Default signal value initialized from ROM storage after PoR before application has physical value evaluated (e.g. sensor inputs), ***refer to /15/section 5.4.*** This kind of default value is specified for each signal in CMX /8/ at column **"Default Value /at Start "**.
- Default signal value initialized from ROM storage in case the needed signal reception is timeout and detected as loss signals. For signal loss detection, refer to section 4.3. This kind of default value is specified for each signal in CMX /8/ at column **"Default Value / at Timeout "**.

4.3 LOST COMMUNICATION DETECTION AND HEALING

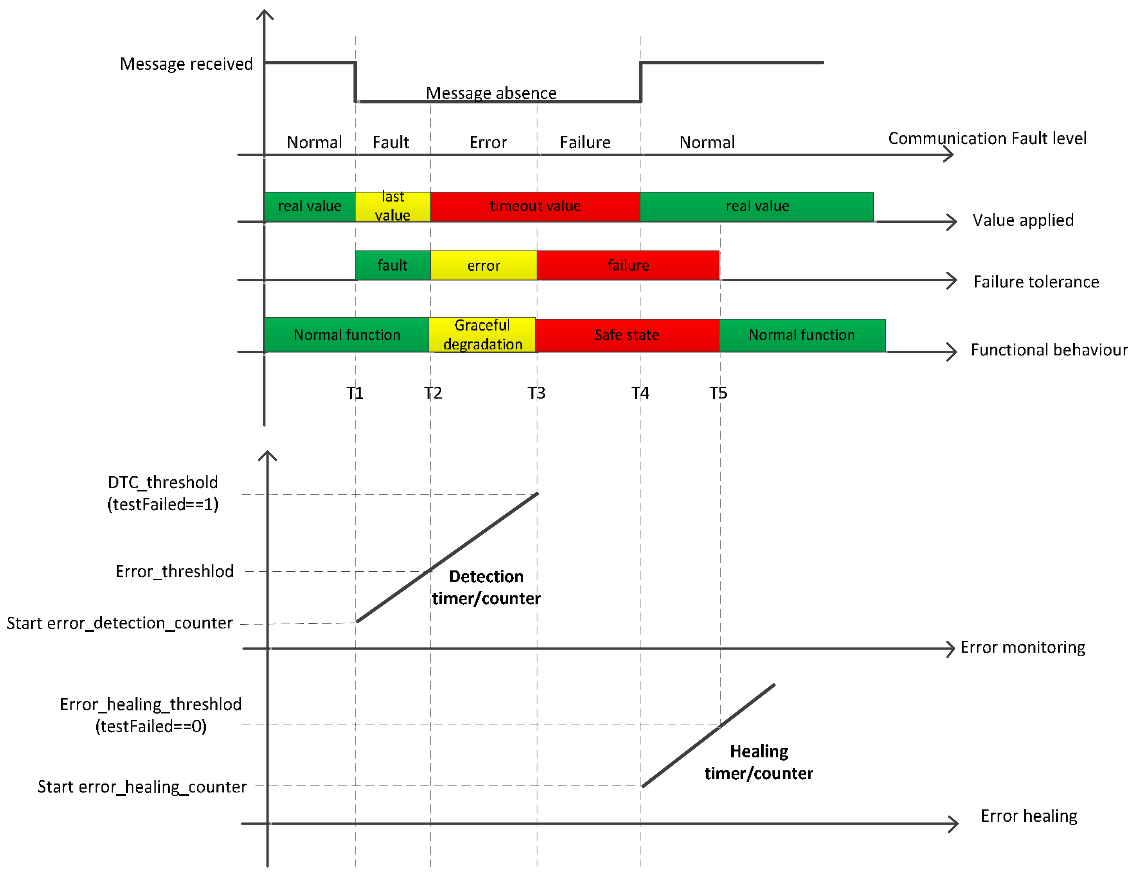
4.3.1 Lost Communication Detection and Healing Strategy

All ECUs in the network should have provision to introduce a message monitoring concept within. Each ECU shall employ two timers /counter (error detection timer/counter and error healing timer/counter) to monitor/heal the message loss. In the case where a particular message sent by a node is lost due to any unexpected reason, all the other nodes in the network which are receivers of the particular message/frame should be able to identify the message loss and recognize this as an error. If the behaviour is repeated continuously, corrective measures have to be taken by the responsible ECU.

Step by step CAN message timeout detection procedure would be as follows:

- A periodic or periodic & event message is continuously getting lost due to unexpected reason.
- Receiver node shall start **error detection timer/counter** to monitor the message loss upon detecting lack of reception of the scheduled message, and **Default Value/ Last Value** should be used as input. All nodes shall assume the last received value is still valid until the frame has been detected as LOST
- If the error detection timer/counter exceeds a **configurable threshold** (**Error_threshold**), the receiver function/functions should switch to limited functionality with function specific **Default Value / at Timeout** as input. For function specific **Default Value / at Timeout**, GEELY IVN team has to be consulted.
- If the detection timer/counter exceeds a second **configurable threshold (DTC_threshold**), a corresponding DTC entry has to be made in the error memory.
- After a random amount of time, receiver nodes start receiving the lost message again.
- Reset the error detection timer/counter.
- Receiver node should start the **healing timer/counter** as soon as the node first receives the lost message for the first time.
- If the healing timer/counter exceeds a **configurable threshold (Error_healing_threshold)**, the receiver function/functions should switch to full functionality mode.
- Reset the healing timer.

Note: Suppliers should take responsibility of healing timer/counter and DTCs erase strategy, and clearly specified in relating ECU Functional Requirement Specification explicably.



NOTE:Above is a general description of fault tolerance mechanism for the failure mode of lost communication, and some detailed measures maybe deviated from this description by the specific ECU which is permitted due to diversity applications best practiced by suppliers.

Figure 4.3-1.GEELY Lost Communication Detection and Healing Strategy

4.3.2 Lost Communication Detection Condition

Frame transmission detection shall be applied to all periodic or periodic &event type of transmissions. CAN frame is detected as LOST with either of following situations:

1) When the IGN switch status is changed from OFF to ON, ~~it is not been received frame since ECU CAN Network enters active state~~, and **tStartupTimeout** has expired.

2) When it has been received once and has not been received any more for continuously ***nbSuccLostFrame*** times of its periodic transmission time – Tx (defined for each periodic and periodic &event frame in / 8/).

NOTE*: Exceptions to* ***tStartupTimeout****, and* ***nbSuccLostFrame*** *are allowed but must be agreed by GEELY officially and clearly specified in relating ECU Functional Requirement Specification explicably. (E.g. maybe there is one very critical signal/message where the LOST signal detection must be more precise than above....)*

Parameter	Symbol	Min	Nominal	Max	Unit	Comment
Offset time after IGN switch OFF->ON before starting timeout mechanism for periodic application frame detected.	tStartupTimeout			2000	ms	*Tolerance 10%*
Time out of reception of needed periodic application frame after last reception of periodic frame	nbSuccLostFrame		5	-	frame	recommend: The minimum timeout value of reception

by target ECU						is **100**ms *Tolerance 10%*

Table 4.3-1: GEELY CAN Parameters on Frame Loss Detection

Frame loss detection shall be only proceed by ECU when it is in Network-active state, which means in Network-inactive and Bus Error/Bus off or Bus Error/Voltage Error, no frame loss detection shall be done.

4.3.3 Lost Communication Self Diagnosis

Once frame loss detection is valid under following conditions, DTC Communication Loss of each reception ECU shall be set in target ECU non-volatile memory:

- Voltage supply of CAN Bus node is in the range of **uVoltageComDTCLog** ,refer to section 4.4.1
- No Bus-off detected

ECU's relevant Communication Loss DTC are described in /9/.

4.4 COMMUNICATION UNDER/OVER VOLTAGE

4.4.1 Communication Under/Over Voltage Strategy

GEELY CAN Bus Voltage supply shall be measured between vehicle ground and KL_30 (for NM node) voltage supply line.

Full functionality (reception and transmission) of GEELY CAN bus shall be guaranteed functional while measured CAN Bus voltage supply stay in range of ***uVoltageValid.***

Network communication (reception and transmission) is permitted outside the specified voltage range of ***uVoltageValid***, provided that the ECU can correctly read and write messages and not cause any disturbance on the network. There shall be **safe margins** in both ends of the operating voltage range in regards to the voltage level where communication becomes unreliable or is no longer possible due hardware constraints. See figure 4.4.1-1.

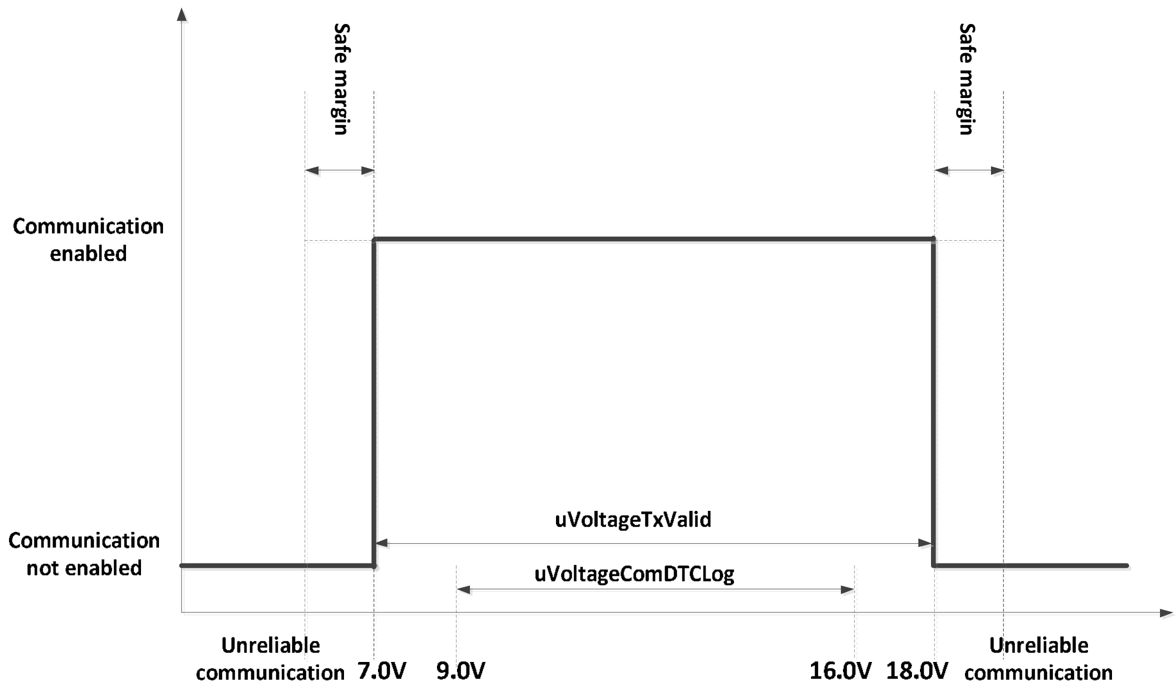


Figure 4.4.1-1 GEELY CAN communication voltage range

Parameter	Symbol	Min	Nominal	Max	Unit	Comment
Normal supply voltage range of ECU message transmission and reception	uVoltageValid	7	-	18	Volts	
Supply voltage range condition for communication DTC log	uVoltageComDTCLog	9		16	Volts	

Table 4.4.1-2 GEELY CAN supply voltage condition parameters

4.4.2 Communication Under/Over Voltage Self Diagnosis

TBD

4.5 CAN BUS-OFF HANDLING

4.5.1 CAN Application Frame Transmission – Bus-Off

In case of ECU enters Bus-Off state, the periodic and event frame shall not be transmit on GEELY CAN Bus. Any remaining periodic and event frame waiting in Tx buffer shall be cleared by ECU application. And no transmission requests are allowed to be pending and resuming if transmission is enabled again.

After Bus-Off recovery and ECU back in network active state, periodic and event frame transmission shall be proceed again with the latest state of the signal.

4.5.2 CAN BUS-OFF Handling And Recovery

For this section, please differentiate between a “bus off state” in the CAN controller and a “bus off handing procedure” that is controlled by the application. A “bus off state” refers to a CAN controller state that ceases message transmission when the error counter exceeds 255 and is controlled by the CAN controller hardware. Whereas a “bus off handing procedure” is an application controlled procedure that shall be implemented through software.

All nodes connecting to GEELY CAN bus shall implement Bus-Off recovery starting after Bus-off detection by node's CAN controller.

In case of the fatal bus error detection, any node shall restart the CAN controller initialization for a BUS recovery using following strategy illustrated in Figure 4.5.2-1 and all Bus-off recovery parameters defined in Table 4.5.2-2 which shall be applied by an ECU on GEELY CAN Bus.

Bus-Off handing procedure describe as below:

- As long as Bus-Off detected by CAN controller and the Bus-Off flag will be set.
- Node will disconnect from the CAN bus within **tBusOffShutdown (a configurable value),** when the Bus-Off event is indicated either by an interrupt or by a status bit that is polled in upper software.
- Reset CAN controller registers.
- All CAN communication should be paused in **tBusOffRecovery (a configurable value).**
- When the timer of **tBusOffRecovery** is expired, the node will (re)start connecting to CAN bus.
- When one of the messages was successfully transmitted, the network state of node should go back to available, i.e. the node will exit Bus-Off handing procedure and return to normal operation.

- If successful transmission of messages are not possible and the Bus-Off event is indicated again, the node should be retry to "bus off handing procedure" until the fault is cleared.

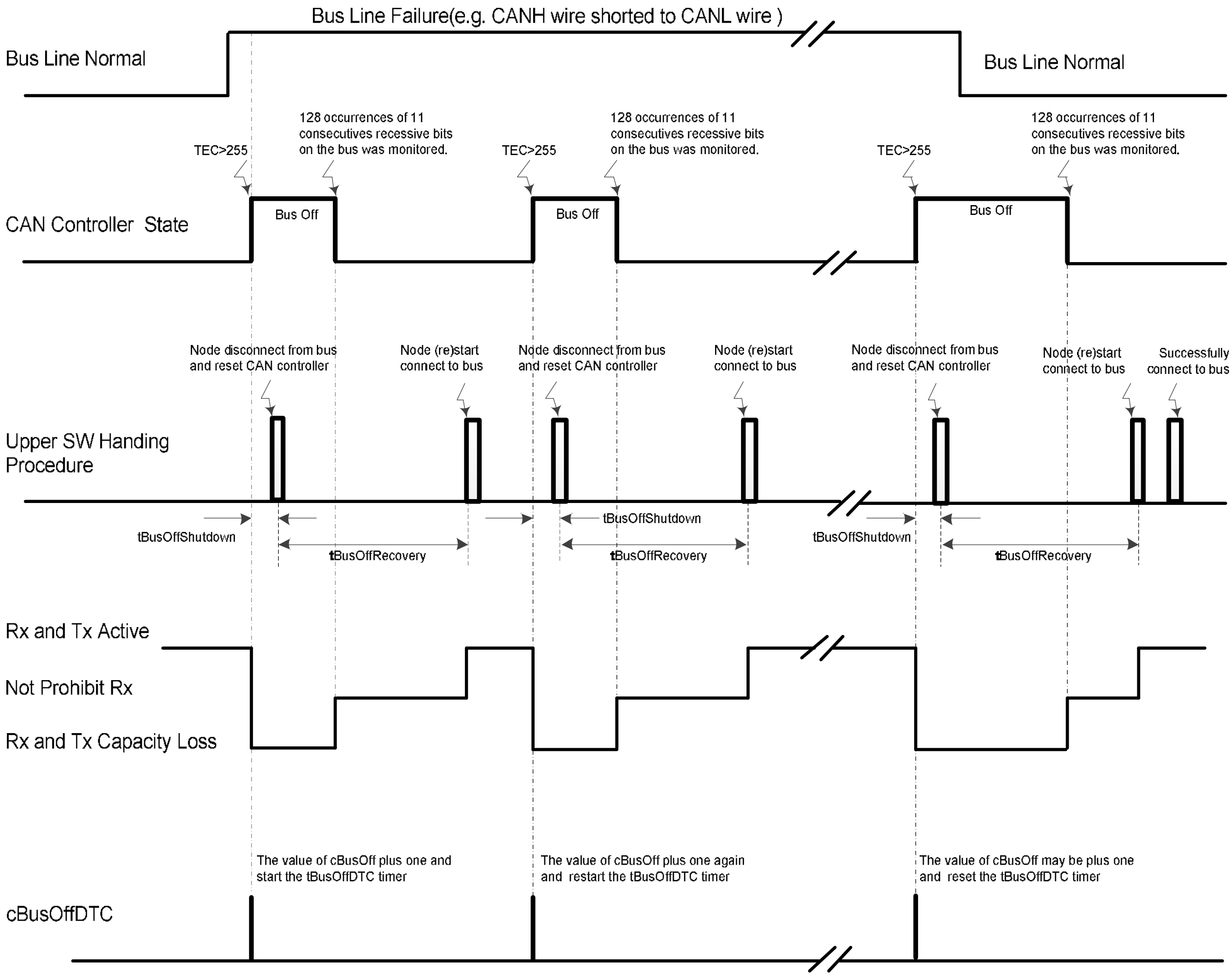


Figure 4.5.2-1 GEELY CAN Bus Bus-Off recovery strategy diagram

The parameters of Bus Off were described in table 4.5.2-2:

Parameter	Symbol	Min	Nominal	Max	Unit	Comment
Bus Off Recovery Time	tBusOffRecovery	-	100	-	ms	Delay time for retrying to activate network after the node was disconnected the bus due to detect bus off flag. *Tolerance 10%*
Bus Off Shutdown Time	tBusOffShutdown	-	-	10	ms	Time to shutdown Network after valid detection of Bus-off error. *Tolerance 10%*

DTC of Bus Off Timer	tBusOffDTC	-	2000	-	ms	The timer is used to capture the number of Bus-Off event, when the timer isn't expired and the value of cBusOff is equal to five, then the DTC_BUS_OFF should be recorded. *Tolerance 10%*
Offset time after IGN switch OFF->ON before DTC storage of Bus Off becomes enable	tBusOffDTCStorageEnable		2000		ms	Time duration after IGN switch OFF->ON to enable DTC of Bus off storage *Tolerance 10%*
Bus Off Counter	cBusOff		5		times	Records of each Bus-Off event times.

Table 4.5.2-2 GEELY CAN Bus Off Recovery Parameters

NOTE:

- The BUS-Off node retried delay timer (tBusOffRecovery) shall be immediately (re)started after the CAN controller was reset by application program.
- Node disconnected from BUS: frame transmission stops and lost frame detection timer stops.
- Node connected to BUS: frame transmission is possible and lost frame detection timer activates.
- Receiving activity is enabled only after the CAN controller has left its internal bus off error state after 128 occurrences of 11 consecutive recessive bits have been seen on the bus, but the periphery environment unreliable, suggested that do not use the received data, the node should enter the safe mode.

4.5.3 CAN BUS-OFF Self Diagnosis

DTC_BUS_OFF will be stored in ECU non-volatile memory for entry of network Bus-Off state under following condition:

- Voltage supply of CAN Bus node is in the range of **uVoltageComDTCLog.**
- Power Mode from IGN OFF-> IGN ON transition has actually happened more than **tBusOffDetectionEnable** before.
- The value of **cBusOff** is equal to **five**.

ECU's relevant DTC_BUS_OFF are described in /9/.

The initial value of **cBusOff** is zero. Once the Bus-Off was detected by upper software, the **cBusOff** must be plus one, as the same time the node will disconnect from the CAN bus, reset CAN controller registers and (re)start the **tBusOffDTC** timer within **tBusOffShutdown (a configurable value)** , if the value of **cBusOff** is equal to five, the DTC_BUS_OFF should be stored in non-volatile memory in ECU, described in Figure 4.5.3-1 , after detected the Bus-Off event again and **tBusOffDTC** timer isn't expired, the value of **cBusOff** will be added plus one and the **tBusOffDTC** timer should be reset. After up to two hundred and fifty-five, the counter no longer increases and need not be cleared also.

When the node reconnects successfully the bus one time and the **tBusOffDTC** timer isn't expired, the value of cBusOff is unchanged.

When the **tBusOffDTC** timer was expired, it should not be restart and the value of **cBusOff** should be cleared.

When the **tBusOffDTC** timer was expired, the Bus-Off event was detected by node, then the value of **cBusOff** is need to plus one until the DTC_BUS_OFF is recorded and the **tBusOffDTC** timer is need to reset. It was described in Figure 4.5.3-2

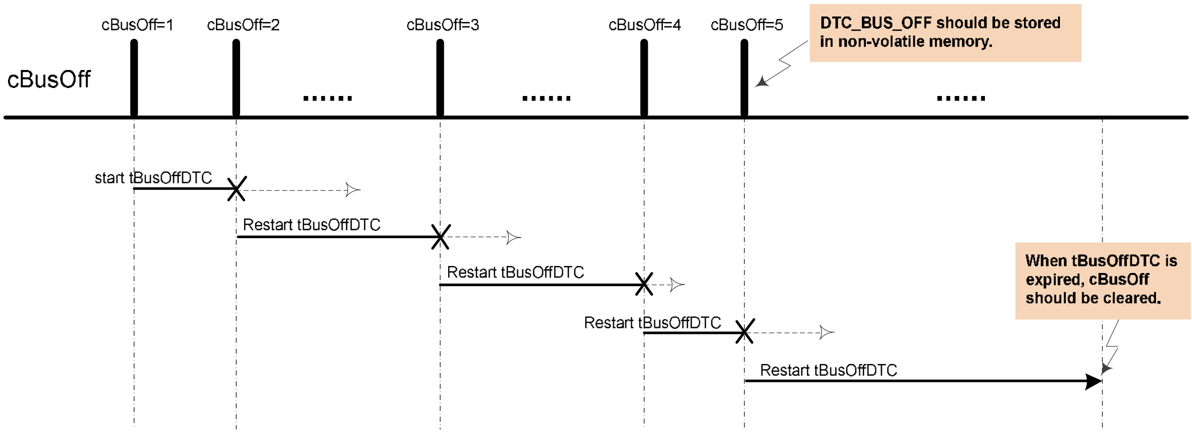


Figure 4.5.3-1 Recording Procedure of DTC_BUS_OFF

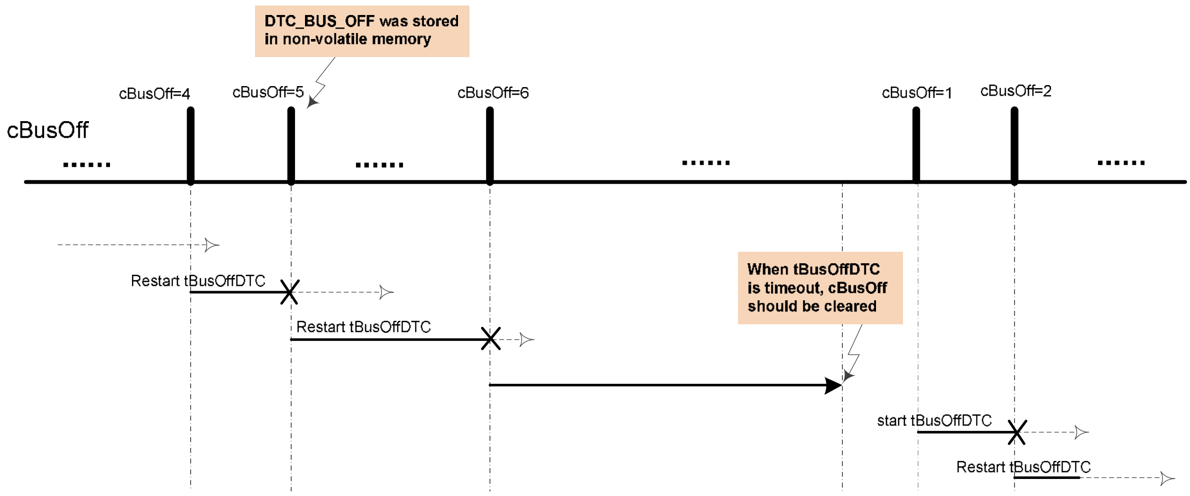


Figure 4.5.3-2 tBusOffDTC Timeout Handling

4.6 CHECKSUM HANDLING

4.6.1 Algorithm of Checksum

The safety code under the control of the safety related process is required to detect message or signals corruption. The function architecture of checksum was described in Figure 4.6.1-1.

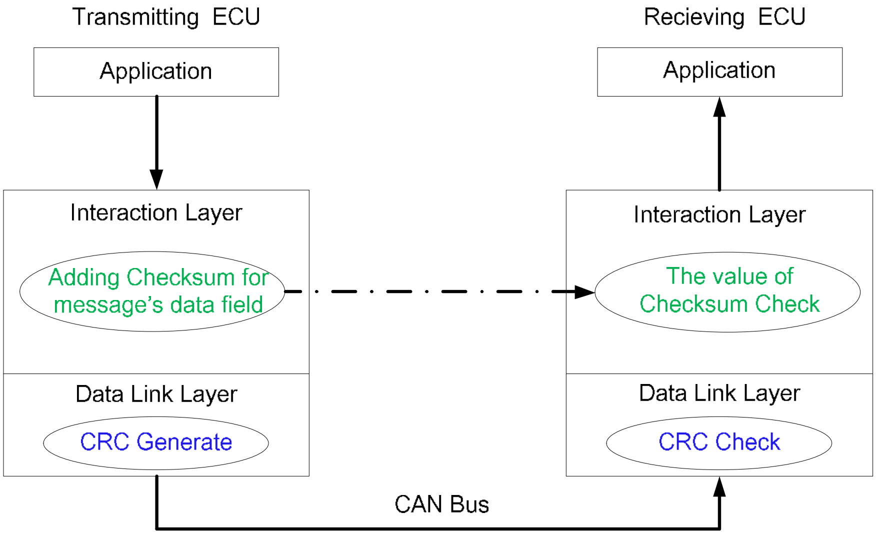


Figure 4.6.1-1 Function Architecture Of Checksum

There are several types of safety coding in GEELY network, as following, if the system have function safety requirements, the CRC algorithm shall be applied.

I. XOR of algorithm based on message

- Length of checksum is 8 bit
- Checksum(Byte7) = (Byte0) XOR (Byte1) XOR (Byte2) XOR (Byte3) XOR (Byte4) XOR (Byte5) XOR (Byte6); The byte sequence of checksum can be determined according to the actual application.

II. 8-bit 0x2F polynomial CRC Calculation

The function of the CRC module shall implement the CRC8 routine based on the generator polynomial 0x2F:

CRC result width:	8 bits
Polynomial:	2Fh
Initial value:	FFh
Input data reflected:	NO
Result data reflected:	NO
XOR value:	FFh
Check:	DFh
Magic check:	42h

Table 4.6.1-2 CRC Algorithm Parameters

- CRC result width: Defines the result data width of the CRC calculation.
- Polynomial: Defines the generator polynomial which is used for the CRC algorithm.
- Initial value: Defines the start condition for the CRC algorithm.
- Input data reflected: Defines whether the bits of each input byte are reflected before being processed (see definition below).
- Result data reflected: Similar to "Input data reflected" this parameter defines whether the bits of the CRC result are reflected (see definition below). The result is reflected over 8-bit for a CRC8.
- XOR value: This Value is XORed to the final register value before the value is returned as the official checksum.

- Check: This field is a check value that can be used as a weak validator of implementations of the algorithm. The field contains the checksum obtained when the ASCII values '1' '2' '3' '4' '5' '6' '7' '8' '9' corresponding to values 31h 32h 33h 34h 35h 36h 37h 38h 39h is fed through the specified algorithm.
- Magic check: The CRC checking process calculates the CRC over the entire data block, including the CRC result. An error-free data block will always result in the unique constant polynomial (magic check) - representing the CRC-result XORed with 'XOR value'- regardless of the data block content.

The CRC module shall provide the following CRC results:

Data bytes (hexadecimal)									CRC
0	0	0	0						12
F2	1	83							C2
0F	AA	0	55						C6
0	FF	55	11						77
33	22	55	AA	BB	CC	DD	EE	FF	11
92	6B	55							33
FF	FF	FF	FF						6C

Table 4.6.1-3 CRC Results

A function has to monitor, whether the sent safety code checksum (transmitting ECU) is equal to the internal calculated safety code checksum (receiving ECU). In case of different values a checksum error will be set, i.e. the **cErrorChecksum** will be added one.

When the checksum was error, and determining whether the data is available, it should be accorded to each ECU function. Suggested that application program do not use the received data, because the periphery communication environment unreliable.

If the current received signal's checksum is not equal to current received message's checksum, the **cErrorChecksum** shall be add one, until the **cErrorChecksum** reach to **10**(Dec), no more increase shall be taken and the DTC for checksum shall be set.

If the current received signal's checksum is equal to current received message's checksum, and the **cErrorChecksum** is not zero, then the **cErrorChecksum** shall be minus one, until the **cErrorChecksum** reach to 0(Dec), no more decrease shall be taken and DTC for checksum shall be unset.

Parameter	Symbol	Min	Nominal	Max	Unit	Comment
Offset time after IGN switch OFF->ON before Checksum detection becomes enable	tChecksumDetectionEnable		2000		ms	Time duration after IGN switch OFF->ON to enable Checksum detection *Tolerance 10%*
Length of Checksum			8		bit	
Value of Error Checksum counter	cEorrorChecksum		10			This is threshold value for recording the DTC_Checksum

Table 4.6.1-4 GEELY Checksum Parameters

4.6.2 DTC OF Checksum Setting

DTC_Checksum will be stored in ECU non-volatile memory under following condition:

- Voltage supply of CAN Bus node is in the range of **uVoltageComDTCLog.**
- Power Mode from IGN OFF-> IGN ON transition has actually happened more than **tChecksumDetectionEnable** before.
- The value of **cErrorChecksum** is equal to **ten (10).**

ECU's relevant DTC_CheckSum are described in /9/.

4.7 ROLLING COUNTER HANDLING

4.7.1 Mechanism of Rolling Counter

Rolling Counter consists of adding a running number to each message exchanged between a transmitter and a receiver. This allows the receiver to check the sequence of message provided by the transmitter. The function architecture was described in Figure 4.7.1-1.

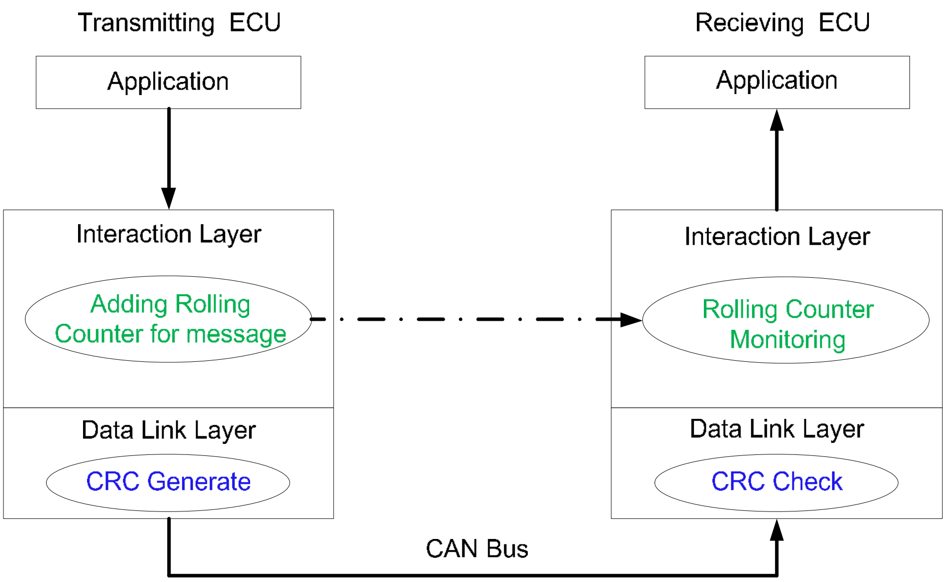


Figure 4.7.1-1 Function Architecture Of Rolling Counter

Length of Rolling Counter is 4bit. Counts from 0 to 15, the signal will be increased by one every message task; if counter is equal to 15 then the counter starts again with 0.

When the ECU initialization or recovery following interruption (e.g. bus off recover successfully), the sequence of message should be counted from 0 to 15.

The parameters of Rolling Counter were described in table 4.7.1-2.

Parameter	Symbol	Min	Nominal	Max	Unit	Comment
Offset time after IGN switch OFF->ON before Rolling Counter detection becomes enable	tRollingCounterDetectionEnable		2000		ms	Time duration after IGN switch OFF->ON to enable Rolling Counter detection *Tolerance 10%*
This is a difference value that the current received rolling counter minus last received rolling counter.	vDiff		1			This is threshold value for setting rolling counter error.

Length of Rolling Counter			4		bit	
Value of Rolling Counter	cRoll	0		15		The value of cRoll should be plus one when the message was transmitted in IL.
Value of Error Rolling Counter	cErrorRoll		10			This is threshold value for recording the DTC_RollingCounter

Table 4.7.1-2 GEELY Rolling Counter Parameters

A function has to monitor, whether the specific CAN signal has been calculated in transmitting ECU and received in ECU; If a value of the cycle counter is missing or the counter is not increased an error of rolling counter will be set. Decision Rolling Counter error strategy should be based on table 4.7.1-3 described. Don 't recommend the value of **vDiff** is greater than one is considered to be rolling counter error.

The strategy of Rolling Counter error was described in table 4.7.1-3.

vDiff	cErrorRoll	Data of reliability	Comment
0	Plus one	Unavailable	The sequence of message is abnormal, the information is repetition.
1	Minus one or unchanged	Available	The sequence of message is normal.
Others value	Plus one	Available	The sequence of message is abnormal, but the information is update.

Table 4.7.1-3 Handling of the Rolling Counter Error

If the current received rolling counter is not equal to last received rolling counter plus **vDiff**, the **cErrorRoll** shall be add one, until the **cErrorRoll** reach to 10(Dec), no more increase shall be taken and the DTC for rolling counter shall be set.

If the current received rolling counter is equal to last received rolling counter plus **vDiff**, and the **cErrorRoll** is not zero, then the **cErrorRoll** shall be minus one, until the **cErrorRoll** reach to 0(Dec), no more decrease shall be taken and DTC for rolling counter shall be unset.

4.7.2 DTC OF Rolling Counter Setting

DTC_RollingCounter will be stored in ECU non-volatile memory under following condition:

- Voltage supply of CAN Bus node is in the range of **uVoltageComDTCLog.**
- Power Mode from IGN OFF-> IGN ON transition has actually happened more than **tRollingCounterDetectionEnable** before.
- The value of **cErrorRoll** is equal to **ten (10).**

ECU's relevant DTC_RollingCounter are described in /9/.

5. APPENDIX

N/A

This page is intentionally blank.