15-0078 CAN J1939 Protocol

for

15_0078 elobau Midi Armrest

Document Properties

Status: **Released**

Version: **2.2**

Author: Baur, Felix

Created: 2019-04-09 15:50

Approved Versions

Current Document version **2.2** is **approved.**

Approved Versions:

- **2.0** (2019-05-08 16:41)
- **2.1** (2019-08-08 07:51)
- **2.2** (2022-05-31 14:21)

Document Signatures

Released (2022-05-31 14:21)

Baur, Felix	Signed	2022-05-31 14:18
Kornelius, Mattis (Reviewer)	Signed	2022-05-31 14:20

www.elobau.com | +49 7561 970-0

1 Revision History

Version	Date	Author	Revision
01.00	14.10.16	Bummele	1st draft
01.01	11.11.16	F.Baur	Extended content
01.02	12.01.17	F.Baur	Smaller bugfixes
01.03	12.06.17	F.Baur	- Updated encoder message - Updated message parameter descriptions - Extended message counter and CRC description
01.04	06.08.18	F.Baur	- Updated document according to C-Sample status. - Added MFAUserCmdVehStsMsg message0
01.05	09.04.2019	F.Baur	- Import to Polarion - Update document, add configurable parameters
1.06	16.04.2019	F. Augustin	Review of document done --> No changes regarding functionality, only linguistic improvements
1.07	23.04.2019	F.Baur	- Correct some cross reference links - Correct CAN bus off decrement time parameters - Update default J1939 name parameter values
1.08	30.05.2019	F.Baur	- Update definition of start time of CAN rx timeout supervision
2.0	07.05.2019	F.Baur	- Polation release version- no content modified
2.1	18.7.2019	F.Baur	- Chapter Power-Down Sequence added
2.2	2022	F.Baur	- Vibro Module integration - Documentation of the J1939_76 options

2 Table of Contents

List of Figures

List of Tables

3 Scope

This document defines the CAN communication concept of the elobau MFA.

4 Acronyms and Definitions

Acronym/Definition	Description
ACL	J1939 Address Claim, (see also Ref 5)
ADC	Analogue Digital Converter
BL	Backlight
CRC	Cyclic Redundancy Check
DA	Destination Address
DC	Diagnostic Coverage
DTC	Diagnostic Trouble Code
DM	J1939 Diagnostic Message(s)
DP	Data Page
FBL	Flash-Bootloader
FL	Function Light(s)
FMI	Failure Mode Indicator
MFA	Multifunctional Armrest
NA	Not Available
NACK	Negative Acknowledgement
NVM	Non volatile memory
PDU	Protocol Data Unit
PF	PDU Format
PGN	J1939 Parameter Group Number
PS	PDU Specific
SAE	Society of Automotive Engineers
SA	Source Address
SDG	Safety Data Group
SDM	Safety Data Message
SHM	Safety Header Message
SPN	Suspect Parameter Number
SW	Software

Table 1 Acronyms and definitions

5 Related Documents

Index	Document
Ref_1	SAE J1939 Recommended Practice for a Serial Control and Communications Vehicle Network
Ref_2	SAE J1939-21 Data Link Layer
Ref_3	SAE J1939-71 Vehicle Application Layer
Ref_4	SAE J1939-73 Application Layer - Diagnostics
Ref_5	SAE J1939-81 Network Management
Ref_6	ISO11898-1 First edition 2003-12-01
Ref_7	ISO11898-2 First edition 2003-12-01
Ref_8	ISO25119 2018
Ref_9	ISO 14229-1: Unified diagnostic services (UDS)
Ref_10	SAE J1939-76:2018 Functional Safety Communications Protocol

Table 2 Related documents

6 General Overview

The MFA is a configurable Armrest which can be adapted to a wide range of user requirements.
Therefore the number and types of assembled control elements (Buttons, Joysticks, Encoder, …) can vary in a certain range.
The possibility to create different hardware configurations also has effect on the CAN communication layout.

Figure 1 Example picture of the elobau MFA

7 CAN Communication

7.1 Physical Parameters

The MFA will be connected to a Controller Area Network (ISO 11898, CAN Specification 2.0B) using SAE J1939 protocol with a standard baudrate of 250 kbit/s. The baudrate is (pre) configurable by a parameter in the configuration file (see 11 - Configuration parameters)

7.2 CAN 29 Bit ID Field

The CAN ID contains following filed- as defined in the J1939 standard (Ref_2):

29 Bit Identifier																												
Priority			R	DP	PDU Format (PF)								PDU Specific (PS)								Source Address (SA)							
28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 3 J1939 Identifier fields

Field	Description
Priority	This bits control the priority of the CAN-message during bus arbitration. Therefore the value of the priority bits has a direct influence on the latency of the individual CAN-messages relative to each other. The priority is defined for each message that is transmitted.
R1	Reserved by SAE. This field is set to 0 for all messages that are transmitted.
DP Data Page	This field is set to 0 for all messages that are transmitted.
PF Field	Transmitted message identifier. The PF field is defined for each message transmitted.
PS Field	Transmitted message identifier extension or destination address. The PS field is defined for each message transmitted. See individual Joystick PGN definitions for PF and PS field information.
Source Address	The Source Address field is 8 bits long. There shall only be one device on the network with a given source address. Therefore, the source address field assures that the CAN identifier is unique, as required by CAN. Address management and allocation is detailed in Ref_5. Procedures are defined in Ref_5 to prevent duplication of source addresses. Reference SAE J1939 Appendix B, Tables B2 through B9, for source address assignments. The SA for the MFA is configurable, please refer to 11 - Configuration parameters.

Table 4 J1939 CAN ID field descriptions

8 MFA Messages

There are different types of CAN messages used for the communication with the MFA. All the messages and signals described in the following chapters are also stated in a *.dbc file, which can be processed by appropriate (VECTOR) CAN tools.

8.1 J1939 standard message overview

Name	Direction	Remarks
ACL	TX	J1939 Address Claim Message (see also Ref_5)
RQST	RX	Request Message. Used to request e.g. the ACL message. (see 8.5 - Definition of J1939 request PGN)

Table 5 J1939 CAN messages

8.2 UDS message overview

Name	Direction	Remarks
ISO15765	TX	This message is used for UDS communication with the MFA. PGN: 0xDA00, Priority: 0x6, e.g. MFA SA 0x6, DA: 0xF9: message ID = 0x18DAF906
ISO15765	RX	This message is used for UDS communication with the MFA. PGN: 0xDA00, Priority: 0x6, e.g. Tester SA 0xF9, DA: 0x06: message ID = 0x18DA06F9

Table 6 UDS related CAN messages

8.3 Proprietary MFA message overview

Name	Direction	Remarks
MFAUserCmdPosMsg[1…N]	TX	Msg contains the values of up to three/four Joystick Axis or three/four Thump Wheels or three/four Lever positions. The three/four stated device types can be freely combined.
MFAUserCmdBtnMsg[1…N]		Msg contains the status of up to 24/32 Buttons.
MFAUserCmdEncMsg[1…N]		Msg contains the values of up to two encoder signals and up to one/two position sensor output with corresponding status signals and up to eight button status signals.
MFAUserCmdVibroMsg		Msg controls the VIBRO module.
MFAUserCmdLmpMsg1	RX	Msg contains the dimming value of the BL and FL the MFA shall set. In addition the msg controls the state of up to 15 FL and the state of the BL.
MFAUserCmdLmpMsg[2…N]		Msg controls the state of up to 24 FL.
MFAUserCmdVehStsMsg		Msg contains the vehicle and motor speed.

Table 7 Proprietary CAN messages

Note: Dependent on the number and type of control elements populated/ configured on the MFA it can be necessary to transmit more than one message of the same message type. Therefore every message contains a tailing number from 1 to *N* to achieve a unique identification of the message.

8.4 Detailed proprietary message definition

8.4.1 MFA User Command Position Message (MFAUserCmdPosMsg1…N)

The MFA User Command Position Message is used to transfer information about the state of the axis, thumbwheels or lever controls populated on the MFA. The message type is Proprietary B.

8.4.1.1 Message parameter

Parameter	Definition	Remarks
Cycle Time [ms]	100ms (default)	This parameter is configurable - refer to 11 - Configuration parameters
Send Type	Cyclic (default), On signal change (configurable)	The message will be sent periodically with the defined cycle time, or if configured, the message will be sent on change of at least one contained signal (Refer to chapter 14.1.2 - CAN send modes).
Message Alive Timeout (in multiples of the cycle time)	2	It is recommended to execute message alive timeout supervision with the stated timeout value by the receiving node(s) to ensure safe data communication. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Requestable	False	Cyclic messages are not requestable.
Data Length (DLC)	8	Data length of the message type.
Priority	3	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF00 + N (+0x50 if J1939_76 is active)	Calculation of PGN: 0xFF00 + message index N e.g. MFAUserCmdPosMsg5: 0xFF00 + 5 = 0xFF05. **Note**: if the J1939-76 Safety Protocol will be used. an offset of 0x50 has to be added to the calculated PGN: e.g. for MFAUserCmdPosMsg5: 0xFF05 + 0x50 = 0xFF51
ID	0xCFF0*[N]*00 + SA	e.g. 0xCFF0106 (fMFAUserCmdPosMsg01, SA 0x6).
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication will be used to ensure a safe data comunication. The receiver node is resposible to validate the incomming data.

Table 8 MFAUserCmdPosMsgN Parameters

8.4.1.2 Message layout

	Signal name	Signal type (see section 9-1 for details)	Length [Bits]	Start position [bit number]
Common part	Pos_[N]_01	Position	10	0
	TopRightFwdSts_[N]_01	Status	2	10
	DownLeftBackSts_[N]_01	Status	2	12
	DetentPosSts_[N]_01	Status	2	14
	Pos_[N]_02	Position	10	16
	TopRightFwdSts_[N]_02	Status	2	26
	DownLeftBackSts_[N]_02	Status	2	28
	DetentPosSts_[N]_02	Status	2	30
	Pos_[N]_03	Position	10	32
	TopRightFwdSts_[N]_03	Status	2	42
	DownLeftBackSts_[N]_03	Status	2	44
J1939_76 protocol used	Pos_[N]_04	Position	10	46
	TopRightFwdSts_[N]_04	Status	2	56
	DownLeftBackSts_[N]_04	Status	2	58
	DetentPosSts_[N]_04	Status	2	60
	DetentPosSts_[N]_04	Status	2	62
Proprietary elobau safety protocol used	MsgSeqNum_PM_[N]	Message sequence number	4	48
	CRC_PM_[N]	CRC	12	52

Table 9 MFAUserCmdPosMsg Message Layout

Note: [N] represents the message index of this message type.
E.g., DetentPosSts_***03***_02 signal will be transmitted within the MFAUserCmdPosMsg***3*** message.

8.4.2 MFA User Command Button Message (MFAUserCmdBtnMsg1…N)

The MFA User Command Button Message is used to transfer information about the states of the button controls populated on the MFA. The message type is Proprietary B.

8.4.2.1 Message parameter

Parameter	Definition	Remarks
Cycle Time [ms]	100ms (default)	This parameter is configurable - refer to 11 - Configuration parameters
Send Type	Cyclic (default), On signal change (configurable)	The message will be sent periodically with the defined cycle time, or if configured, the message will be sent on change of at least one contained signal (Refer to chapter 14.1.2 - CAN send modes).
Message Alive Timeout (in multiples of the cycle time)	2	It is recommended to execute message alive timeout supervision with the stated timeout value by the receiving node(s) to ensure safe data communication. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Requestable	False	Cyclic messages are not requestable.
Data Length (DLC)	8	Data length of the message type.
Priority	3	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF10 + N (+0x50 if J1939_76 is active)	PGN will be calculated: 0xFF10 + message number N: e.g. MFAUserCmdBtnMsg2: 0xFF10 + 2 = 0xFF12. **Note**: if the J1939-76 Safety Protocol will be used. an offset of 0x50 has to be added to the calculated PGN: e.g. for MFAUserCmdBtnMsg2: 0xFF12 + 0x50 = 0xFF62
ID	0xCFF1*[N]*00 + SA	e.g. 0xCFF1106 (MFAUserCmdBtnMsg01, SA 0x6).
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication will be used to ensure a safe data comunication. The receiver node is resposible to validate the incomming data.

Table 10 MFAUserCmdBtnMsgN Parameters

8.4.2.2 Message layout

	Signal Name	Signal Type (see section 9-1 for details)	Length [Bits]	Start position [bit number]
Common part	BtnSts_[N]_01	Button	2	0
	BtnSts_[N]_02	Button	2	2
	BtnSts_[N]_03	Button	2	4
	BtnSts_[N]_04	Button	2	6
	BtnSts_[N]_05	Button	2	8
	BtnSts_[N]_06	Button	2	10
	BtnSts_[N]_07	Button	2	12
	BtnSts_[N]_08	Button	2	14
	BtnSts_[N]_09	Button	2	16
	BtnSts_[N]_10	Button	2	18
	BtnSts_[N]_11	Button	2	20
	BtnSts_[N]_12	Button	2	22
	BtnSts_[N]_13	Button	2	24
	BtnSts_[N]_14	Button	2	26
	BtnSts_[N]_15	Button	2	28
	BtnSts_[N]_16	Button	2	30
	BtnSts_[N]_17	Button	2	32

	BtnSts_[N]_18	Button	2	34
	BtnSts_[N]_19	Button	2	36
	BtnSts_[N]_20	Button	2	38
	BtnSts_[N]_21	Button	2	40
	BtnSts_[N]_22	Button	2	42
	BtnSts_[N]_23	Button	2	44
	BtnSts_[N]_24	Button	2	46
J1939_76 protocol used	BtnSts_[N]_25	Button	2	48
	BtnSts_[N]_26	Button	2	50
	BtnSts_[N]_27	Button	2	52
	BtnSts_[N]_28	Button	2	54
	BtnSts_[N]_29	Button	2	56
	BtnSts_[N]_30	Button	2	58
	BtnSts_[N]_31	Button	2	60
	BtnSts_[N]_32	Button	2	62
Proprietary elobau safety protocol used	MsgSeqNum_BM_[N]	Message sequence number	4	48
	CRC_BM_[N]	CRC	12	52

Table 11 MFAUserCmBtnMsg Message Layout

Note: [N] represents the message index of this message type.
E.g., BtnSts_***02***_07 signal will be transmitted within the MFAUserCmdBtnMsg***2*** message.

8.4.3 MFA User Command Encoder Message (MFAUserCmdEncMsg1…N)

The MFA User Command Encoder Message is used to transfer information about state of the encoder and potentiometer controls populated on the MFA. The message type is Proprietary B.

8.4.3.1 Message parameter

Parameter	Definition	Remarks
Cycle Time [ms]	100ms (default)	This parameter is configurable - refer to 11 - Configuration parameters
Send Type	Cyclic (default), On signal change (configurable)	The message will be sent periodically with the defined cycle time, or if configured, the message will be sent on change of at least one contained signal (Refer to chapter 14.1.2 - CAN send modes).
Message Alive Timeout (in multiples of the cycle time)	2	It is recommended to execute message alive timeout supervision with the stated timeout value by the receiving node(s) to ensure safe data communication. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Requestable	False	Cyclic messages are not requestable.
Data Length (DLC)	8	Data length of the message type.
Priority	3	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF20 + N (+0x50 if J1939_76 is active)	Calculation of PGN: 0xFF20 + message number N: e.g. MFAUserCmdEncMsg1: 0xFF20 + 1 = 0xFF21. **Note**: if the J1939-76 Safety Protocol will be used. an offset of 0x50 has to be added to the calculated PGN: e.g. for MFAUserCmdEncMsg1: 0xFF21 + 0x50 = 0xFF71
ID	0xCFF2*[N]*00 + SA	e.g. 0xCFF2106 (MFAUserCmdEncMsg01, SA 0x6).
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication will be used to ensure a safe data comunication. The receiver node is resposible to validate the incomming data.

Table 12 MFAUserCmdEncMsg Parameters

8.4.3.2 Message layout

	Signal Name	Signal Type (see section 9-1 for details)	Length [Bits]	Start position [bit number]
Common part	EncoderCnt_[N]_01	Encoder	8	0
	EncoderCnt_[N]_02	Encoder	8	8
	Pos_EM[N]_1	Position	10	16
	TopRightFwdSts_EM[N]_01	Status	2	26
	DownLeftBackSts_EM[N]_01	Status	2	28
	DetentPosSts_EM[N]_01	Status	2	30
	BtnSts_[N]_01	Button	2	32
	BtnSts_[N]_02	Button	2	34
	BtnSts_[N]_03	Button	2	36
	BtnSts_[N]_04	Button	2	38
	BtnSts_[N]_05	Button	2	40
	BtnSts_[N]_06	Button	2	42
	BtnSts_[N]_07	Button	2	44
	BtnSts_[N]_08	Button	2	46
J1939_76 protocol used	Pos_EM[N]_2	Position	10	48
	TopRightFwdSts_EM[N]_02	Status	2	58
	DownLeftBackSts_EM[N]_02	Status	2	60
	DetentPosSts_EM[N]_02	Status	2	62
Proprietary elobau safety protocol used	MsgSeqNum_EM_[N]	MsgSeqNum	4	48
	CRC_EM_[N]	CRC	12	52

Table 13 MFAUserCmEncMsg Message Layout

Note: [N] represents the message index of this message type.
E.g., BtnSts_***01***_04 signal will be transmitted within the MFAUserCmdEncMsg***1*** message.

8.4.4 MFA User Command Lamp Message (MFAUserCmdLmpMsg1)

The MFA User Command Lamp Message 1 is used to transfer the requested dimming values for BL/FL that are populated on the MFA. In addition it is used to transfer the requested state of 15 Lamps and the BL command to the MFA.. The message type is Proprietary B.

8.4.4.1 Message parameter

Parameter	Definition	Remarks
Cycle Time [ms]	1000ms (default)	This parameter is configurable - refer to 11 - Configuration parameters
Send Type	Cyclic	The message shall be sent periodically with the defined cycle time.
Message Alive Timeout (in multiples of the cycle time)	2	Message alive timeout supervision for this message is applied by MFA. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Requestable	False	Cyclic messages are not requestable.
Data Length (DLC)	8	Data length of the message type.
Default Priority	6	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF31	PGN is 0xFF31.
ID	0x18FF31[xx]	[xx] represents the Sender ID (default: 0xFE, accepts all sender nodes) The sender ID is configurable so that filtering for a specific sender is possible.
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication must be provided by the transmitting node and will be checked by the MFA.

Table 14 MFAUserCmdLmpMsg1 Parameters

8.4.4.2 Message layout

Signal Name	Signal Type (see section 9-1 for details)	Length [Bits]	Start position [bit number]
BLCmd	Lamp	2	0
Dimming_BL	Dimming	8	2
Dimming_FL	Dimming	8	10
FLCmd_01_01	Lamp	2	18
FLCmd_01_02	Lamp	2	20
FLCmd_01_03	Lamp	2	22
FLCmd_01_04	Lamp	2	24
FLCmd_01_05	Lamp	2	26
FLCmd_01_06	Lamp	2	28
FLCmd_01_07	Lamp	2	30
FLCmd_01_08	Lamp	2	32
FLCmd_01_09	Lamp	2	34
FLCmd_01_10	Lamp	2	36
FLCmd_01_11	Lamp	2	38
FLCmd_01_12	Lamp	2	40
FLCmd_01_13	Lamp	2	42
FLCmd_01_14	Lamp	2	44
FLCmd_01_15	Lamp	2	46
MsgSeqNum_LM_01*	Message sequence number	4	48
CRC_LM_01*	CRC	12	52

Table 15 MFAUserCmdLmpMsg1 Message Layout

[*] unused if J1939_76 CAN safety protocol is configured, see 15.1 - J1939-76 safety protocol

8.4.5 MFA User Command Lamp Message (MFAUserCmdLmpMsg2…3)

The MFA User Command Lamp Messages 2…N are used to control up to 24 Lamp states of the MFA. The message type is Proprietary B.

8.4.5.1 Message parameter

The message parameters are the same as for the MFAUserCmdLmpMsg1, please refer to chapter 8.4.4.1 - Message parameter.
The PGNs are 0xFF32 for MFAUserCmdLmpMsg2 and 0xFF33 for MFAUserCmdLmpMsg3.

The signal layout is slightly different compared to the layout of the MFAUserCmdLmpMsg1 message because only the message MFAUserCmdLmpMsg1 provides the control of the dimming values for the BL and FL and the control of the BL lamp state. In contrast the message MFAUserCmdLmpMsg2…3 contains only FL commands.

8.4.5.2 Message layout

Signal Name	Signal Type (see section 9-1 for details)	Length [Bits]	Start position [bit number]
FLCmd_[N]_01	Lamp	2	0
FLCmd_[N]_02	Lamp	2	2
FLCmd_[N]_03	Lamp	2	4
FLCmd_[N]_04	Lamp	2	6
FLCmd_[N]_05	Lamp	2	8
FLCmd_[N]_06	Lamp	2	10
FLCmd_[N]_07	Lamp	2	12
FLCmd_[N]_08	Lamp	2	14
FLCmd_[N]_09	Lamp	2	16
FLCmd_[N]_10	Lamp	2	18
FLCmd_[N]_11	Lamp	2	20
FLCmd_[N]_12	Lamp	2	22
FLCmd_[N]_13	Lamp	2	24
FLCmd_[N]_14	Lamp	2	26
FLCmd_[N]_15	Lamp	2	28
FLCmd_[N]_16	Lamp	2	30
FLCmd_[N]_17	Lamp	2	32
FLCmd_[N]_18	Lamp	2	34
FLCmd_[N]_19	Lamp	2	36
FLCmd_[N]_20	Lamp	2	38
FLCmd_[N]_21	Lamp	2	40
FLCmd_[N]_22	Lamp	2	42
FLCmd_[N]_23	Lamp	2	44
FLCmd_[N]_24	Lamp	2	46
MsgSeqNum_LM_[N] *	Message Sequence Number	4	48
CRC_LM_[N] *	CRC	12	52

Table 16 MFAUserCmdLmpMsg2...3 Message Layout

[*] unused if J1939_76 CAN safety protocol is configured, see 15.1 - J1939-76 safety protocol

Note: [N] represents the message index of this message type.
E.g., FLCmd_***03***_20 signal will be transmitted within the MFAUserCmdLmpMsg***3*** message.

8.4.6 MFA User Command Lamp Message (MFAUserCmdLmpMsg4)

The MFA User Command Lamp Message 4 are used to control possible external connected RGB LEDs. The message type is Proprietary B and the PGN for MFAUserCmdLmpMsg4 is 0xFF34.

8.4.6.1 Message parameter

The message parameters are the same as for the MFAUserCmdLmpMsg1, please refer to chapter 8.4.4.1 - Message parameter . The signal layout is different compared to the layout of the MFAUserCmdLmpMsg1 message because this message contains a simple protocol to control the external RGB LEDs.

8.4.6.2 Message layout

Signal Name	Signal Type	Length [Bits]	Start position [bit number]
LED_ID	multiplexer	8	0
CURR_LED_nn	multiplexed value	8	8
MsgSeqNum_LM_04 *	Message Sequence Number	4	48
CRC_LM_04 *	CRC	12	52

Table 17 MFAUserCmdLmpMsg4 Message Layout

[*] unused if J1939_76 CAN safety protocol is configured, see 15.1 - J1939-76 safety protocol

8.4.6.3 Signal description

Signal Name	Description
LED_ID	Used as multiplexer value; enumerates the used LED output pins of the external LED driver.
CURR_LED_xx	This signal contains the current value to set for the defined LED driver output (see LED_ID). The signal uses following chracteristic: 2Bit width, unsigned, Unit: [%], Gain factor 0.4, 0=0%, 250=100%, 0xFE=Error, 0xFF=N/A.

Table 18 MFAUserCmdLmpMsg4 signal description

8.4.7 MFA User Command Vehicle Status Message (MFAUserCmdVehStsMsg)

In a default J1939 environment the MFA receives the current vehicle status with the following J1939 signals:

1. Signal WhBsdVehSpd (SPN: 84) in message CCVS (PGN: 0xFEF1)
2. Signal EngSpd (SPN: 190) in message EEC1 (PGN: 0xF004)

If those signals are not available in the customer environment the MFA proprietary message MFAUserCmdVehStsMsg is available to transmit these informations to the MFA.
Providing named informations is necessary because it has to be ensured that the vehicle is in a defined state (standstill / no motor speed) when the MFA shall jump to programming / extended diagnostic mode (UDS).

8.4.7.1 Message parameter

Parameter	Definition	Remarks
Cycle Time [ms]	1000 ms (default)	This parameter is configurable - refer to 11 - Configuration parameters
Send Type	Cyclic	The message shall be sent periodically with the defined cycle time.
Message Alive Timeout (in multiples of the cycle time)	2	Message alive timeout supervision for this message is applied by MFA. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Data Length (DLC)	8	Data length of the message type.
Default Priority	3	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF00	Proprietary B PGN.
ID	0xCFF00[xx]	[xx] represents the Sender ID (default: 0xFE, accepts all sender nodes). The sender ID is configurable so that filtering for a specific sender is possible.
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication must be provided by the transmitting node and will be checked by the MFA.

Table 19 MFAUserCmdVehStsMsg Parameters

8.4.7.2 Message layout

Signal Name	Signal Type	Length [Bits]	Start position [bit number]
EngSped	J1939 EngSpd, see 8.4.7.3	16	0
WhBsdVehSpd	J1939 WhBsdVehSpd, see 8.4.7.3	16	16
MsgSeqNum_VS_01	Message sequence number	4	48
CRC_VS_01	CRC	12	52

Table 20 MFAUserCmVehStsMsg Message Layout

8.4.7.3 Definition of J1939 signals for vehicle status detection

Signal name	EngSpd
Width	16 Bit, unsigned
Byteorder	Intel
Unit	rpm
Gain	0.125
Offset	0
J1939 SPN	190

Table 21 EngSpd signal parameters

Signal name	WhBsdVehSpd
Width	16 Bit, unsigned
Byteorder	Intel
Unit	km/h
Gain	0.00390625
Offset	0
J1939 SPN	84

Table 22 WhBsdVehSpd signal parameters

8.4.8 MFA User Command VIBO Command Message (MFAUserCmdVibroMsg)

This message will be used to control the VIBRO module of the MIDI joystick.

The VIBRO message uses a small protocol implemented with CAN signal multiplexing. The "VibroPhase" signal acts as multiplexer signal. Depending on this signal value the other remaining signals have different meanings.

8.4.8.1 Message parameter

Parameter	Definition	Remarks
Send Type	on demand/ non cyclic	The message will be sent for each new demand
Message Alive Timeout (in multiples of the cycle time)	2	Message alive timeout supervision for this message is applied by MFA. Please refer to 15.2.2 - Time expiration (communication timeout / watchdog) for more details.
Data Length (DLC)	8	Data length of the message type.
Default Priority	6	Priority of the message type.
Message type	Proprietary B	User defined PGN in range 0xFF00 – 0xFFFF, refer to Ref_2 for more information about J1939 message types.
Parameter Group Number (PGN)	0xFF39	Proprietary B PGN.
ID	0x8FF39[xx]	[xx] represents the Sender ID (default: 0xFE, accepts all sender nodes). The sender ID is configurable so that filtering for a specific sender is possible.
Safety	Yes	One of the options documented in 15 - Safety-related CAN Communication must be provided by the transmitting node and will be checked by the MFA.

Table 23 MFAUserCmdVibroMsg Parameters

Note: The following constants VIBRO_NUM_PATTERN and VIBRO_NUM_PHASES are fixed to:
VIBRO_NUM_PATTERN: 6
VIBRO_NUM_PHASES: 3
in this project.

8.4.8.2 VIBRO trigger/execution message

VibroPhase value = 0

Signal Name	Description	Startposition [Byte/Bit]	Width [Bits]
VibroPattern	Pattern number [0 .. VIBRO_NUM_PATTERN-1] Defines the 0-based pattern number.	0/0	8
VibroPhase	0	1/0	8
VibroPatternDelay	Defines the delay time between two repetitions of the selected pattern [ms]	2/0	8
VibroPatternRepeat	defines the number of repetitions of the selected pattern [0: one execution, 1: one repetition, .. 255]	3/0	8
MsgSeqNum *	4 Bit message sequence counter like defined in 15.2.3 - Message sequence numbering	6/0	4
CRC *	12 Bit CRC signal like defined in 15.2.4 - CRC	6/4	12

Table 24 VIBRO trigger message

8.4.8.3 VIBRO configuration signal

To configure single phases within a defined pattern, the following message should be used:

VibroPhase value = [1 .. VIBRO_NUM_PHASES]

Signal Name	Description	Startposition [Byte/Bit]	Width [Bits]
VibroPattern	Pattern number [0 ... VIBRO_NUM_PATTERN-1] Defines the 0-based pattern number.	0/0	8
VibroPhase	[1 .. VIBRO_NUM_PHASES]: selects the phase of the selected pattern to configure.	1/0	8
VibroPhaseStartGain	Defines the initial DC of the phase [0-100%]	2/0	8
VibroPhaseTargetGain	Defines the target DC of the phase [0-100%]	3/0	8
VibroPhaseTime	Phase duration [ms]	4/0	8
VibroPhasePolarity	Phase polarity [J1939 status signal, 0=off, 1=on, 0x2=error, 0x3=N/A]	5/0	2
MsgSeqNum *	4 Bit message sequence counter like defined in 15.2.3 - Message sequence numbering	6/0	4
CRC *	12 Bit CRC signal like defined in 15.2.4 - CRC	6/4	12

Table 25 VIBRO config message

8.4.8.4 VIBRO no operation

This is a empty message which do not trigger any reaction of the VIBRo module. This message should be used for to keep alive the timeout monitor for this VIBRO CAN message.

VibroPhase value = 0xFD

Signal Name	Description	Start position [Byte/Bit]	Width [Bits]
VibroPattern	x- don't care	0/0	8
VibroPhase	0xFD	1/0	8
MsgSeqNum *	4 Bit message sequence counter like defined in 15.2.3 - Message sequence numbering	6/0	4
CRC *	12 Bit CRC signal like defined in 15.2.4 - CRC	6/4	12

Table 26 VIBRO no operation message

8.4.8.5 VIBRO stop

To stop the execution of a VIBRO pattern/ phase sequence, following mode/command can be used:

VibroPhase value = 0xFE

Signal Name	Description	Start position [Byte/Bit]	Width [Bits]
VibroPattern	x- don't care	0/0	8
VibroPhase	0xFE	1/0	8
MsgSeqNum *	4 Bit message sequence counter like defined in 15.2.3 - Message sequence numbering	6/0	4
CRC *	12 Bit CRC signal like defined in 15.2.4 - CRC	6/4	12

Table 27 VIBRO stop message

8.4.8.6 VIBRO manual mode

To drive the VIBRO module manually/ directly without using any pattern/ phase sequence, following mode can be used:

VibroPhase value = 0xFF

Signal Name	Description	Start position [Byte/Bit]	Width [Bits]
VibroPattern	x- don't care	0/0	8
VibroPhase	0xFF	1/0	8
VibroDutyCycleManual	Defines the DC to use [0-100%]	2/0	8
VibroPhasePolarity	Phase polarity [J1939 status signal, 0=off, 1=on, 0x2=error, 0x3=N/A]	5/0	2
MsgSeqNum *	4 Bit message sequence counter like defined in 15.2.3 - Message sequence numbering	6/0	4

CRC *	12 Bit CRC signal like defined in 15.2.4 - CRC	6/4	12

Table 28 VIBRO manual mode message

[*] unused if J1939_76 CAN safety protocol is configured, see 15.1 - J1939-76 safety protocol

8.4.8.7 VIBRO module- predefined patterns

Following six preconfigured patterns are available on the MIDI:

Pattern	Phase	Start Gain [%]	Target Gain [%]	Time [ms]	Polarity
0- soft klick	1	60	60	60	0
	2	70	70	20	1
	3	-	-	-	-
1- strong klick	1	70	70	60	0
	2	70	70	20	1
	3	-	-	-	-
2- smooth click	1	70	70	20	0
	2	35	35	50	1
	3	-	-	-	-
3- rising ramp	1	20	40	150	0
	2	40	60	70	0
	3	40	60	30	1
4- falling ramp	1	60	60	40	0
	2	60	10	300	0
	3	-	-	-	-
5- puls alarm	1	20	50	200	0
	2	50	50	30	0
	3	50	20	200	0

Table 29 9.4.7.4 VIBRO module- preconfigured patterns

8.4.8.8 VIBRO module- example pattern

Pattern	Phase	Start Gain [%]	Target Gain [%]	Time [ms]	Polarity
Example	1	20	50	30	0
	2	70	50	60	0
	3	50	0	30	0

Table 30 Example Pattern values

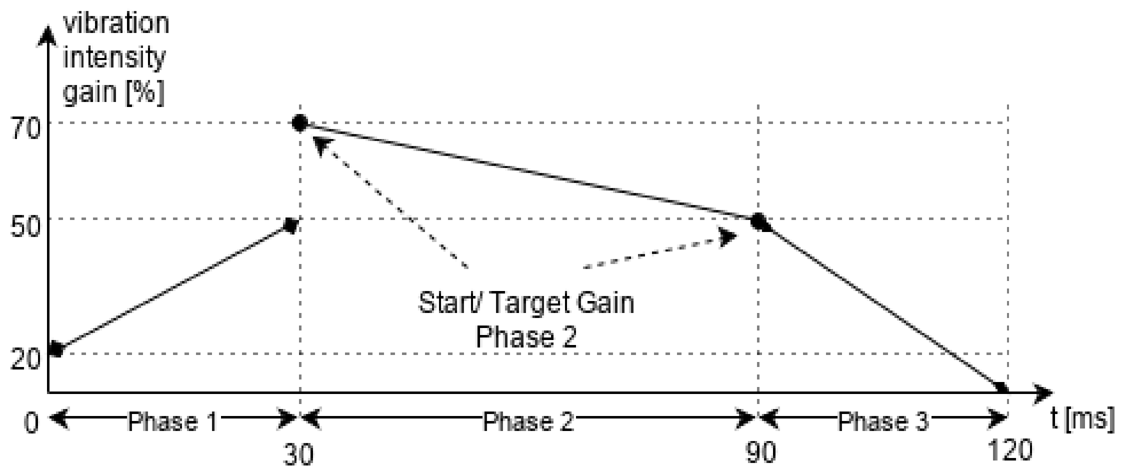

Figure 2 Example Pattern Diagramm

Note: The Polarity is not shown in the figure above. The polarity controls the rotating direction of the vibration motor. This could be used, for example, to actively brake a rotational movement.

If a Pattern repetitions and a pattern delay time (e.g. 40ms in the example below) are configured (see 8.4.8.2 - VIBRO trigger/execution message) the pattern will have the following course:

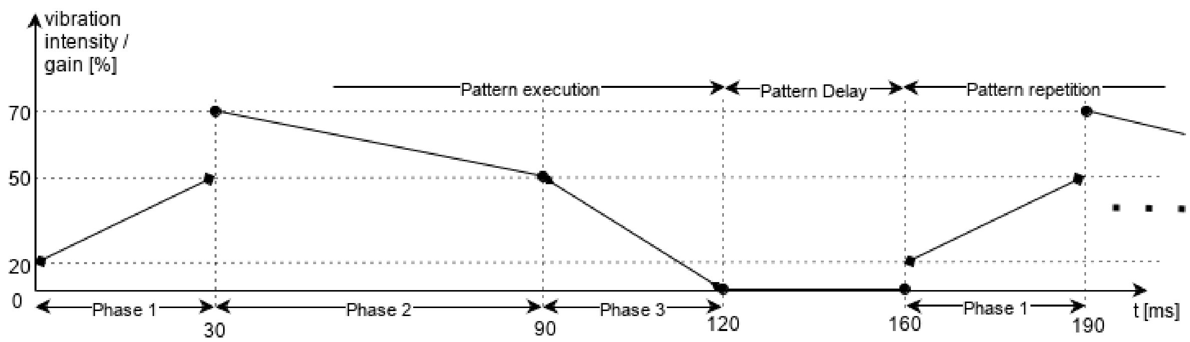

Figure 3 Pattern Repetition with a Patern delay of 40ms

8.5 Definition of J1939 request PGN

8.5.1 PGN definition

This message type, identified by the PGN, provides the capability to request information globally or from a specific destination. Requests specific to one destination are known as destination specific requests. The information in the table below
shows the PGN definition for the "Request PGN" Parameter Group. [Ref 2, 5.4.2]

Name	Description
Parameter Group Name	Request
Definition	Used to request a Parameter Group from a network device or devices.
Transmission repetition	Per user requirements , generally recommended that requests occur no more than 2 or 3

rate	times per second
Data length	3 bytes (The CAN frame for this PG shall set the DLC to 3.)
Extended Data Page	0
Data page	0
PDU Format	234
PDU specific field	Destination Address (global or specific)
Default priority	6
Parameter Group Number	59904 (0x00EA00)
Byte: 1,2,3	Parameter Group Number being requested

Table 31 Request PGN definition

8.5.2 Requestable PGNs of the MIDI ECU

PGN	Description
0x00EE00	Address claimed (see 13.1.2 - Network Management (Address Claiming Process))
0x00FEDA	Soft (see 8.5.4.1 - J1939 Soft PGN)
0x00FDC5	ECU ID (see 8.5.4.2 - J1939 ECUID PGN)
0x00FECA	DM1 (see 17.2.2 - DM1- Active Diagnostic Trouble Codes)
0x00FECB	DM2 (17.2.3 - DM2- Previous active Diagnostic Trouble Codes)
0x00FECC	DM3 (see 17.2.4 - DM3- Diagnostic Data Clear/Reset Of Previously Active DTCs)
0x00FED3	DM11 (see 17.2.5 - DM11- Diagnostic Data Clear/Reset For Active DTCs)

Table 32 Requestable PGNs

8.5.3 Response codes

The MIDI ECU will respond to a destination specific request with following Acknowledgment control bytes: [Ref_2, Figure 9]

Acknowledgment- Control byte	Description
0	Positive Acknowledgment (ACK), if the request was successful
1	Negative Acknowledgment (NACK), if the requested PGN does not exist/ is not implemented
2	Access Denied, not used
3	Cannot Respond, BAM protocol is busy.

Table 33 Acknowledgement control bytes

8.5.4 Requestable PGN definition

8.5.4.1 J1939 Soft PGN

Byte	Description
1	Number of software identification designators represented in the software identification parameter group.
2...16	Vehile Manufacuter Data (DID 0xF187)
17	Field delimiter (*)
18...32	SW Version of Armrest Mainboard (DID 0xF195)

33	Field delimiter (*)
34...48	P/N of the Configuration (DID: 0xF1F2)
49	Field delimiter (*)

Table 34 Soft PGN definition

8.5.4.2 J1939 ECUID PGN

Byte	Description
1...15	Armrest P/N (DID 0xF192)
16	Field delimiter (*)
17...31	Armrest Serialnumber (DID 0xF1FD)
32	Field delimiter (*)
33	Field delimiter (*) (ECULocation is unused)
34	Field delimiter (*) (ECUType is unused)
35...49	ECU Manufacturer Name: "elobau GmbH " (DID: 0xF1F1)
50	Field delimiter (*)
51/52	ECU Hardware Version (DID: 0xF193)
53	Field delimiter (*)

Table 35 ECU ID PGN definition

9 MFA CAN signal definitions

The following table states the available CAN signal types:

Type	Width [bits]	Endianness	Sign	Unit	Default value	Factor	Offset	Min	Max	Table
Button	2	Intel	Unsigned	-	0	1	0	0	3	0: Button not pressed 1: Button pressed 2: Error Indicator 3: Not Available (NA)
Status	2	Intel	Unsigned	-	0	1	0	0	3	0: Not in Left/Right/Fwd/Back/Detent Position 1: On Left/Right/Fwd/Back/Detent Position 2: Error Indicator 3: Not Available (NA)
Position	10	Intel	Unsigned	%	0	0,1	0	0	102	0: 0% 1000: 100% 1001- 1021: Not used 1022 (0x3FE): Error Indicator 1023 (0x3FF): Not Available (NA)
Lamp	2	Intel	Unsigned	-	0	1	0	0	3	0: Lamp Off 1: Lamp On 2: Lamp Flash 3: Not Available (NA)
Dimming	8	Intel	Unsigned	%	0	1	0	0	255	0: 0% 100: 100% 101- 253: Not used 254 (0xFE): Error Indicator 255(0xFF): Not Available (NA)
Encoder	8	Intel	Unsigned	n	0	1	0	0	255	0: Counter 0 250: Max counter value 251- 253: Not used 254 (0xFE): Error Indicator 255(0xFF): Not Available (NA)
MsgSeqNum	4	Intel	Unsigned	n	0	1	0	0	15	0: Counter zero value 11: Counter max value (turn over to 0) 12: Module value (only in first frame) 13: Reserved for future use 14: Error condition 15: Not Available (NA)
CRC	12	Intel	Unsigned	-	0	1	0	0	4095	12 Bit CRC value over the ID and data part of the CAN message Generator Polynomial G(x): x^12+x^11+x^3+x^2+x+1=(x+1)(x^11 +x^2+1)= 0x80F (refer to section 15.2.4 - CRC for detailed information)

Table 36 MFA CAN Signal Types

10 Mapping control elements to CAN signals

The mapping between the populated hardware controls of the MFA and the destination CAN signal(s) is configurable (offline) and can differ - dependent on the hardware configuration, please see 11 - Configuration parameters for more details.

Example Mapping (not conclusive):

Figure 4 CAN mapping example

Notation: MessageName.SignalName

11 Configuration parameters

Elobau is able to create a specific configuration for the MFA armrest to fulfill customer specific requirements. These configurations are created, verified and flashed inhouse at elobau. This means that the configuration is not modifiable during operation of the MFA armrest.

The below stated parameters can be adapted to meet customer requirements:

ID	Parameter	Default value	Min - Max	Description
1000.50	CAN baudrate	250 kBit/s	1 - 1000 kBit/s	CAN baudrate the MFA shall use.
1006.0	CAN send mode	On change inactive	On/Off	Defines the CAN send mode, refer to chapter 14.1.2 - CAN send modes
999.x	SA of the customer ECU	0xFE	1- 0xFE	To avoid PGN/message collisions (if a defined proprietary PGN(s) are used also by other ECUs) the SA definition of the sender ECU allows extended message filtering on the Lamp messages and the vehicle status message.
1006.1	On change minimal cycle time	20 ms	10 - 1000 ms	If CAN send mode "on change" is active this value defines the minimum time period between two CAN-TX messages. This addresses only the specific message (Refer to chapter 14.1.2.2 - On signal change mode).
1000.x	CAN message cycle times	100 ms	5 - 600000 ms	Cycle times of the CAN-messages are configurable in stated range. A customized message cycle time setup must be evaluated in terms of the overall busload by the customer. Note: In case of a activated J1939_76 RX option, all RX CAN messages with a cycle time of 0ms will not be checked according to the safety communication protocol.
1000.0	Device CAN source address (SA)	0x06	1 - 0xFD	Source ID of the MFA (Refer to section 7.2 for detailed information).
1001.x	J1939 ACL name parameters	See 13.1.2 - Network Management (Address Claiming Process)	-	J1939 Name parameters like Identity Number, Manufacturer Code, ECU Instance, Function Instance, Function, Vehicle System, Vehicle System Instance, Industry Group, Arbitrary Address, Capable
 1003.0 1003.1 1003.2	CAN Bus Off parameters - Maximal timeouts - Timeout delay - Decrement time	 32 [n] 10 ms 300000 ms	 0 - 1000 0 - 1000 ms 1 - 3600000 ms	Number of CAN Bus Off resets and timeout values (Refer to section 14.1.1 - CAN Bus off reset)
	UDS	"_CUSTOMER_DATA_"	-	vehicleManufacturerSparePartNumberDataIdentifier (DID 0xF187): Customer specific ID (15 byte) to identify the armrest within the CAN network.
1005.0	Vehicle Status Check	enabled	On/Off	En/Disable the vehicle status check before jump to programming or extended diagnostic session (Refer to chapter 8.4.7).
1020.0	DM1 message	disabled	enabled/ disabled	En/Disable the transmission of the J1939 DM1 message. Also the cyclic transmission in case of no

				active DTCs could be configured.
1050.0	J1939-76Safety protocol TX	disabled	enabled/ disabled	En/Disable the J1939-76 Safety Protocol for the CAN TX messages instead of the elobau proprietary defined CRC/ sequence counter solution.
1050.1	J1939-76Safety protocol RX	disabled	enabled/ disabled	En/Disable the J1939-76 Safety Protocol to the CAN RX messages instead of the elobau proprietary defined CRC/ sequence counter solution. Note: In case of a activated J1939_76 RX option, all RX CAN messages with a configured cycle time (see parameter 1000.x) of 0ms will not be checked according to the safety communication protocol.
1040.0	Global healing disable	disabled	enabled/ disabled	Global DTC healing disable: If enabled no DTCs can change their status bit 0 (DTC_STS_TEST_FAILED) from active to inactive during the same drive cycle
1000.60	Initial RX timeout monitoring delay	0	0-10000 ms	Startup delay for timeout monitoring of RX frames (related to the ACL message)
1004.0	Disable the RX safety checks	0 (=safety checks active)	3 bit bitfield: 0...7	Disable the CAN RX safety measures for the MFAUserCmdLmpMsg*X* and MFAUserCmdVibroMsg: (bit set to 1 = check disabled) bit[0]: timeout monitoring bit[1]: CRC check bit[2]: sequence counter check
1030.3	Lamp test delay	0	0 - 10000 ms	Time delay to start the LampTest (t0, 0=start with ACL)

Table 37 Configuration Parameter List

Component specific parameter	Description
CAN mapping	For each operator control element populated on the MFA the mapping to corresponding CAN signal(s) is configurable. e.g. Joystick X-axis shall appear on CAN signals Pos_04_01, TopRightFwdSts_04_01, DownLeftBackSts_04_01,DetentPosSts_04_01, for a further example please see 10 - Mapping control elements to CAN signals
Mapping control devices to UDS configuration bits.	The UDS protocol can be used to modify a bit array of 128 individual bits and store this configuration to the NVM. Each of these bits can be mapping to one/more configured control elements. Set a bit to "zero" will disable the mapped control element. On this channel the input signal processing and validation will be disabled and the corresponding CAN signal(s) value will set to "N/A". To enable the functionality a "1" has to we written to the defined bit. See UDS service "Variant Coding" in section 17 - Diagnostics or the corresponding diagnostic description file (cdd)
Stuck@Startup test (en/disable)	For each component a stuck at startup test can be enabled. If this check is enabled the considered component will be checked for neutral position during startup. For detailed information please refer to chapter 13.1.5 - Stuck test during system startup. **Note**: This stuck tests are designed to detect defect devices during the startup phase. To disable this test may reduce the defined diagnostic coverage (DC) of the channel.
Stuck close detection threshold time	For buttons and PTOs a stuck close detection can be enabled. If the device is in “close/ pressed” position for more than a defined amount of time the corresponding CAN signal will be set to error state until a key cycle occurs.
General device specific characteristic.	General device specific parameters (debounce times, transmission characteristic, signal inversion, filter settings, voltage thresholds, ...) can be adjusted.

Table 38 Device specific configuration option

12 Flash Bootloader

The MFA microcontroller holds an elobau flash bootloader as a separate software part within its flash memory. This flash bootloader offers the possibility to update the MFA application software via UDS communication. In addition, there is reserved a region in the flash memory of the MFA that is designated to store a customer specific flash bootloader. With this bootloader the MFA application can be updated using a specific update routine.

13 Functional description

13.1 Power-Up sequence

13.1.1 Supply Voltages

For boot up the MFA, constant power supply line and the switched ignition line must be > 8V.

13.1.2 Network Management (Address Claiming Process)

The MFA sends the Address Claim message 500ms after power on with the global destination address 0xFF.

The address claim message (ACL) is requestable with the RQST-message.

13.1.3 J1939 Name Definition

All J1939 Name definitions can be configured by parameters. The following stated values are the defaults:

ID	Value
Identity Number	0
Manufacturer Code	152 / 0x98 (elobau)
ECU Instance	0
Function Instance	0
Function	66 / 0x42 (I/O Controller)
Reserved Filed	0
Vehicle System	0
Vehicle System Instance	0
Industry Group	0 (Global)
Arbitrary Address Capable	0

Table 39 J1939 ACL Name parameter

13.1.4 LED test

During start-up of the MFA a LED test sequence is executed. This test turns on all assembled LEDs for a 1s and turns off all LEDs afterwards. This test will illuminate all assembled LEDs regardless of the configuration installed and should help to detect defect LEDs.

13.1.5 Stuck test during system startup

During power on/startup of the MFA all populated operator control components with enabled Stuck@startup test (see your specific configuration datasheet) must be in zero/neutral position, otherwise the MFA will detect a Stuck@startup error condition.
There are two stuck@startup test policies, depending on the control element type:

type of the control element	Algorithm	Description
Button based component (Buttons, PTOs)	Stuck test	If the component is not in neutral / zero position during initialization the state of the component will switch to signal safe state and the related CAN signals will be set to error state until the next power cycle.
Position based components (Joystick Axis, Thumbwheels, JFTs, Hitchwheels, Potentiometers, ..)	Force to zero	If the component is not in neutral / zero position during initialization the corresponding CAN signal(s) will deliver a zero position until the operator control returns to the neutral position. After crossing the neutral position, the CAN signals will follow the real operator control position.

Table 40 Stuck@Startup modes

Note: Dependent on the vehicle function that is controlled by a control component this test can be switched on/off. Please refer to chapter 11 - Configuration parameters for more information.

13.1.6 Stuck test after system startup

With sending out the ACL message, the MFA switches to "Operational State" and the test window for the stuck@startupenabled control elements ends. During "Operational State" the stuck@operation test is active for all control elements configured accordingly (see datasheet of the used configuration).

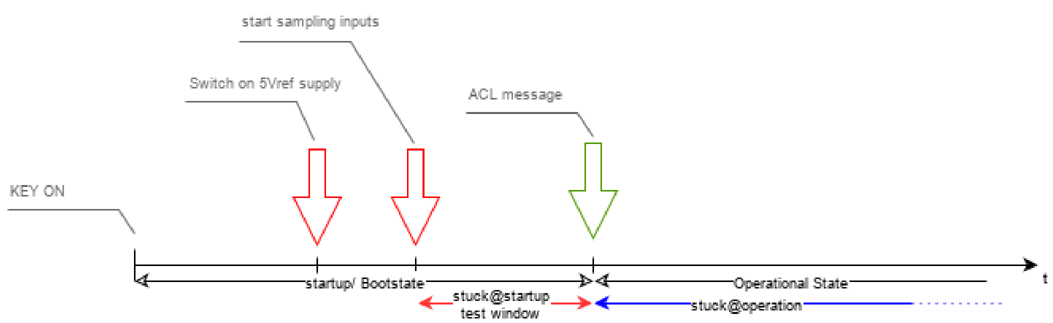

Figure 5 Stuck tests timing conditions

13.2 Operation

During operational mode the status of all activated operator control elements is cyclically sampled and evaluated. In situation that any control element specific error is detected the corresponding CAN signals of the control element will be set to error state (Signal Safe State). All unaffected control elements/signals will stay in operational state.

In case system errors with a possible impact on the integrity of the CAN signals are detected the MFA switches to "System Safe State". In this state the functional CAN communication will be disabled and therefore the armrest will stop sending out functional CAN messages. UDS diagnostic communication is still available.

The MFA have following system states:

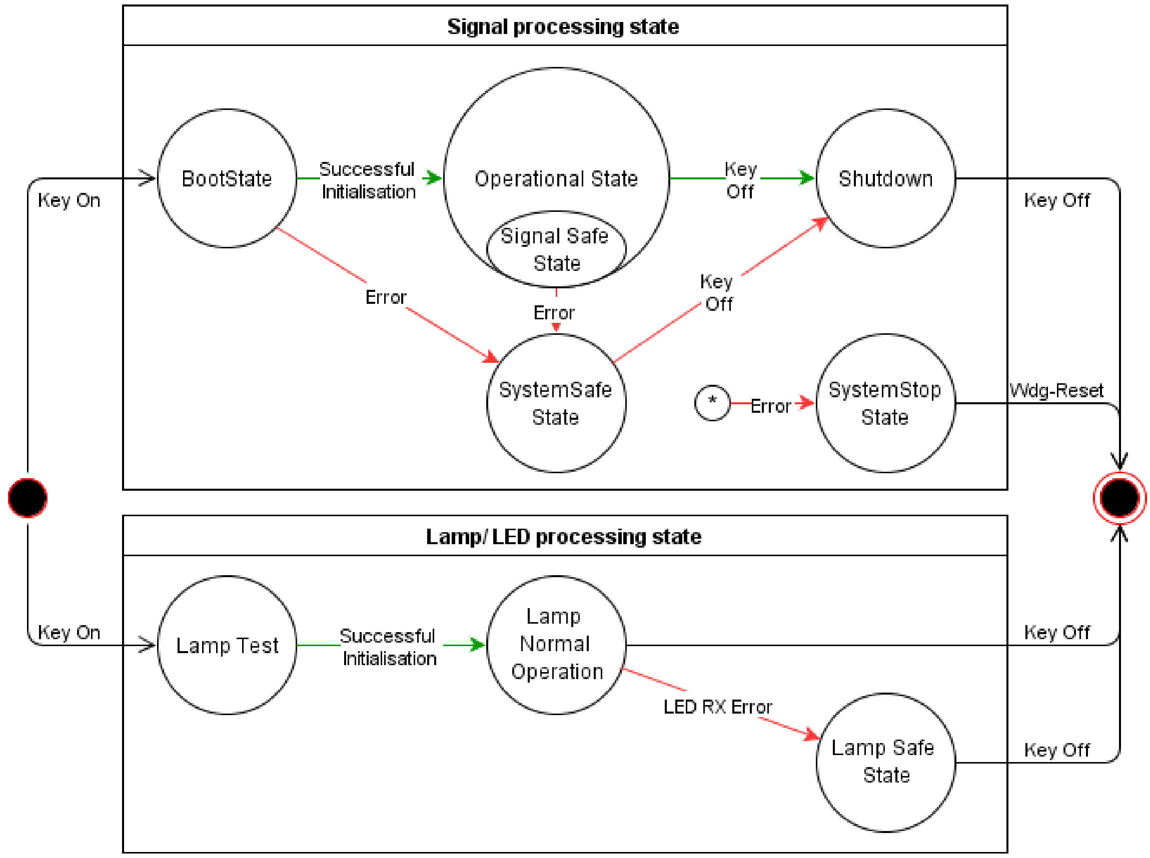

Figure 6 Operating modes overview

Signal processing state	Description	CAN State
Boot State	System starts after power up. - Initialize Hardware/ Software Modules - System Startup Checks (CPU Register, RAM, Application Flash)	Silent
Operational State	Normal operating Mode- no critical System errors active.	Active
Signal Safe State	Signal errors of one or more control elements detected. Corresponding CAN signal(s) set to error state.	Active- Only affected CAN signals in error state
System Safe State	State will be active in case of a critical system error. - Switch OFF functional CAN communication - Flash all connected LEDs (1 Hz, from SW V2.x: 2Hz)	No functional CAN communication. UDS Diagnostic is available (depends on occurred system error)

System Stop State	In case of a critical error in which a further safety signal processing could not guaranteed (RAM errors, CPU register errors), the controller must switch to this state and wait for a hardware watchdog reset signal.	No CAN communication.
Shutdown	- Write Fault Memory to EEPROM - De-initialize software modules	Silent

Figure 7 System states

LED processing State	Description
Lamp Test	Initial lamp test during system startup. Switch On all assembled LEDs for one second- then switch Off again.
Lamp Normal Operation	Normal processing of incoming LED messages.
Lamp Safe State	If incoming LED messages contain errors (CRC, Sequence counter, timeout) the controller switches to this state. All LEDs will blink with 1Hz to signalize this state to the operator.

Figure 8 LED states

13.3 Power-Down sequence

13.3.1 Supply Voltages

For a regular shutdown of the MFA, (only) the ignition line should be switched OFF (<7.5V). This "KEY OFF" event will be detected by the MFA and the device can proceed all necessary steps for a proper shutdown (store data back to the EEPROM, deactivate modules, stop CAN timeout monitoring, hold the CAN TX communication active)
After this sequence, the MFA will switch OFF to reduce the power consumption.

Note: Switch OFF the ignition and the constant power line at the same time could lead to an inconsistent shutdown behavior.

13.3.2 CAN behaviour during shutdown

After KEY OFF event, the CAN RX timeout supervision will be stopped 50ms (debouncing) after detection.

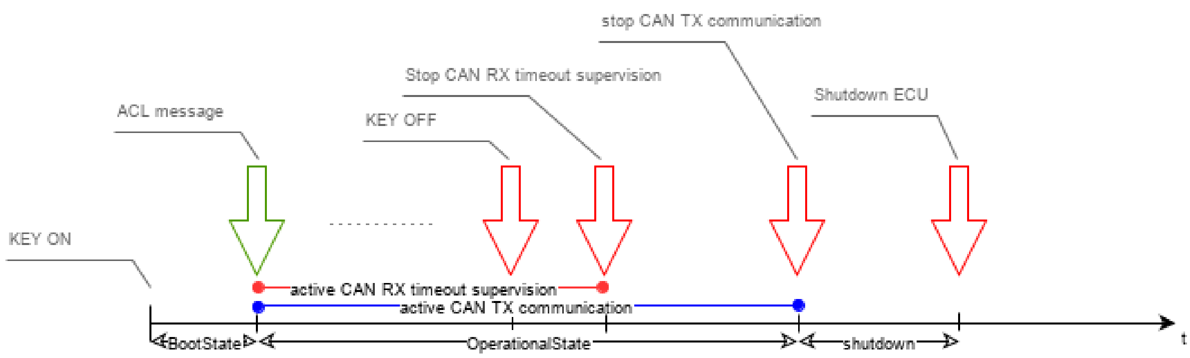

Figure 9 Power Down Sequence

CAN TX communication will be stopped after the RX timeout supervision to avoid timeout events in receiver ECUs.

14 CAN Bus specific

14.1.1 CAN Bus off reset

If the MFA detects a CAN-Bus off state an internal counter is incremented. After a configurable time has elapsed the MFA tries to re-enable the bus interface again. If the bus off counter exceeds a configurable value the CAN node will remain in “bus off” condition and does not try to re-enable the bus interface again. The internal bus off error counter is decremented by one every time a time period of configurable duration without occurrence of a bus off condition has elapsed. The limit of the bus off counter (until which bus recovery is tried), the duration of the time until the next recovery is tried and the "healing time" are configurable. Please refer to chapter 11 - Configuration parameters for further details.

14.1.2 CAN send modes

14.1.2.1 Cyclic mode

The CAN messages are transmitted according to the pre-configured cycle times.

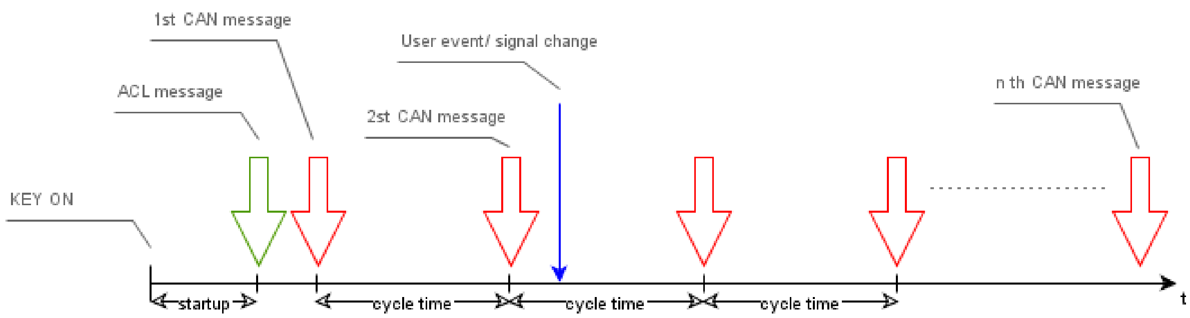

Figure 10 CAN Tx cyclic mode

14.1.2.2 On signal change mode

If this mode is active, CAN signal changes caused by e.g., user press a button, will trigger an immediate send cycle of the related CAN message. After this (asynchronous) send event, the defined cycle time will be respected again. (This allows the receiver of the message to implement a receive timeout control)

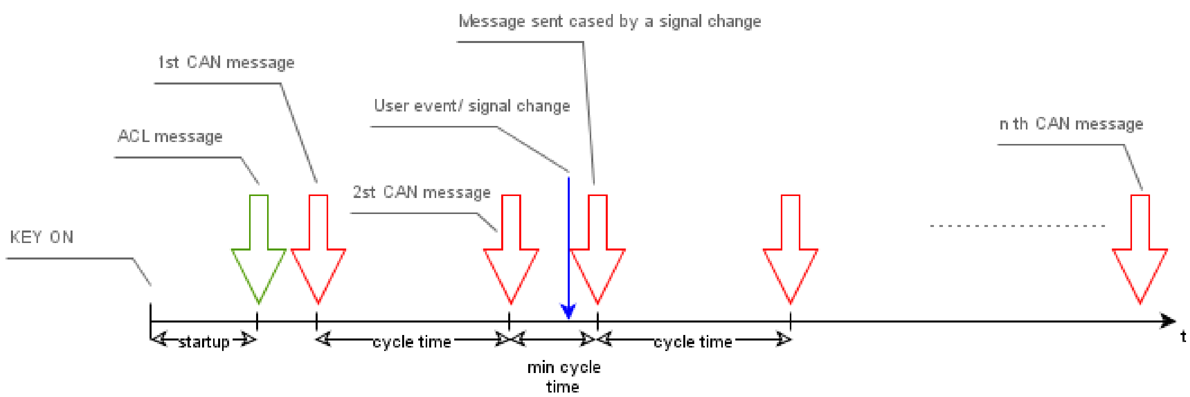

Figure 11 CAN Tx on signal change mode

15 Safety-related CAN Communication

15.1 J1939-76 safety protocol

To ensure a safe data communication, the J1939_76 standard protocol, defined in Ref_10 can be used.
To activate the option, see parameter 1050.0/1 of section 11 - Configuration parameters.

15.1.1 J1939-76 TX

For all cyclic messages which contains safety related operator control information an additional SHM (Safety Header Message) will be sent before sending the related SDM (Safety Data Message)

The receiving device has to validate the correctness of the received SDG (Safety Data Group) according to Ref_10.

15.1.2 1939-76 RX

For incoming messages the J1939_76 protocol could be activated to ensure a safe data transmission to the MIDI armrest.

If parameter 1050.1 is set to "1" the J1939_76 protocol will be active. In this case, all cyclic CAN messages with a configured $cycletime \neq 0$ will be under RX timeout monitoring.

Note: Only MFAUserCmdLmpMsg1 will be initially put under RX supervision, all other (optional) messages will be put unter RX timout supervision after the first reception of that message:

Message	J1939_76 RX supervision
MFAUserCmdLmpMsg1	Initial active
MFAUserCmdLmpMsg2	Starts with reception of the first message
MFAUserCmdLmpMsg3	
MFAUserCmdLmpMsg4	
MFAUserCmdVibroMsg	
MFAUserCmdVehStsMsg	

Table 41 J1939_76 RX supervision activation modes

Note: Only one sender node (SA) per message (PGN) is allowed.

15.1.3 1939-76 configuration

15.2 elobau proprietary safety protocol

15.2.1 Methods and Measures for communication error detection

The below stated methods and measures implemented to control CAN communication errors

Fault Effect	Time expiration/ watchdog	Message sequence number	CRC
Failure of communication peer	X		
Unintended message repetition		X	
Message loss	X	X	
Insertion of messages		X	
Re-sequencing		X	
Message corruption			X
Message delay	X		
Blocking access to data bus	X		
Constant transmission of messages (babbling idiot)	X		

Table 42 Methods and Measures to avoid CAN errors

Note:

1. The message source (ECU) is responsible to calculate the CRC and message sequence number single values and send the message in the defined cycle time.
2. The message sink ECU is responsible to execute the data assurance mechanisms described above to validate the incoming data.

15.2.2 Time expiration (communication timeout / watchdog)

Safety-related CAN-messages shall be transmitted periodically with a defined cycle time. The receiver checks permanently if the time interval between the reception of a safety-related message exceeds a predefined value. If a timeout occurs an error is supposed and therefore the receiving unit will trigger safe state for the related safety function. Of course, the time out shall be greater than the cycle time of the considered message.

Message	Timeout start condition	Timeout factor
MFAUserCmdLmpMsg1	Initial active	2
MFAUserCmdLmpMsg2	Start with first message	
MFAUserCmdLmpMsg3		
MFAUserCmdLmpMsg4		
MFAUserCmdVibroMsg		
MFAUserCmdVehStsMsg		

Table 43 Timeout factors and start conditions

The message timeouts are calculated by multiplying the defined cycle time of a message with the defined timeout factor of the message.

The timeout start condition determines the moment the timeout supervision of a CAN message switches to active:

Initial active: Timeout supervision begins with startup of the armrest, in exact terms with sending out the first functional CAN message after the ACL frame.

Start with first message: Start of the timeout supervision will be triggered when the first message of that kind is received.

15.2.3 Message sequence numbering

A sequence number is included in each safety-related message exchanged between the sender and the receiver. This sequence counter is realized by a running number which is written to an addition data field of the message and which changes in a predetermined way between a safety-related message and the next one of the same kind.

Algorithm/ Description:

- Start with initial value 12 (only once at start-up of the ECU)
- Then the transmitter will increase the counter in every transmit cycle from 0 to 11 and wraparound to 0.
- 13: reserved for future use.
- 14: signalize a signal error state
- 15: signalize N/A.

Example:

Sequence/ time	Value	Description
1	12	Start value - is sent only once at start-up of the sender ECU.
2	0	Lowest correct value of the message sequence number
3	1	Increment value each cycle time
4	2	
5	3	
6	4	
7	5	
8	6	
9	7	
10	8	
11	9	
12	10	
13	11	Highest valid value of the message counter sequence
14	0	Wrap around and start with 0 again
15	1	Increment value
16	2	
17	3	
…	…	…

Table 44 Message sequence counter example

15.2.4 CRC

15.2.4.1 General description

The CRC value is calculated over the data of a CAN frame that is stated below:

- Message ID
- Data field (exclusive CRC signal itself)

The message ID must be aligned to the LSB and has to be padded to 32 bits using 0 (zeros) in the most significant part.

After calculation of the CRC has finished the CRC value is attached to the considered CAN-frame by the transmitter to increase protection. Therefore, the calculated CRC and the signal values over which the CRC was calculated are transmitted within the same CAN-message. The receiver picks up the message (CRC and signals) and calculates the CRC in the same way as the transmitter. After that the receiver must compare the CRC that was received and the calculated one. If they do not match a fault has to be issued.

15.2.4.2 CRC attributes

The MFA uses the following CRC attributes for the 12 Bit CRC calculation:

Attribute	Value	Description
Width	12	CRC width in [Bit]
Polynomial	0x80F	Defines the CRC generator polynomial
Initial value	0xFFF	Start value of CRC calculation
XOR	0xFFF	The result of the CRC calculation will be XOR with this value.
RefIn	False	Input data shall NOT reflected
RefOut	False	Output data shall NOT reflected

Table 45 CRC polynomial attributes

15.2.4.3 CRC calculation example

Example CRC calculation: (CAN ID: 0x18FF32FD)
Data string (sent):

CAN ID (little endian)				Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0xFD	0x32	0xFF	0x18	0x00	0x00	0x00	0x00	0x00	0x00	0xA5	0x71

Note: All underlined data is considered for the CRC checksum.
Calculated CRC checksum value: 0x71A
Message Counter value: 0x5

Data string for CRC calculation of the message receiving unit:

CAN ID (little endian)				Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]
0xFD	0x32	0xFF	0x18	0x00	0x00	0x00	0x00	0x00	0x00	0x05

Note: The receiver must set the higher nibble of the data byte[6] (part of the CRC-value) to “0x0” for the CRC calculation (Highlighted in red).

16 MFA behavior in case of CAN communication errors

16.1 Lamp message

The vehicle controller is able to control the LED states of the MFA with the defined Lamp messages (see 8.4.4). The MFA supervises the incoming Lamp message(s) with following mechanisms:

1. Timeout detection: The MFA checks if the message is received within the expected time range.
2. CRC: The CRC checksum of the frame is checked by the MFA.
3. Sequence counter: The contained 4 bit sequence counter is evaluated by the MFA.

If one of the above stated checks detects a problem the MFA will switch to “Lamp Safe State”. In this state all LED blink with a frequency of 1Hz until a power cycle occurs.

16.2 Vehicle status message

If the J1939-signal “WhBsdVehSpd” or the J1939-signal “EngSpd“ and additionally the “Vehicle Status message” is not available (Refer to chapter 9.4.6 for details) and the configuration option “Vehicle Status Check” is enabled (Refer to chapter 11 - Configuration parameters for details) a jump to "Programming session" or to “Extended Diagnostic session” is not possible.

16.3 MFA TX CAN messages

All proprietary MFA TX-CAN messages contain a CRC- and Sequence counter value.
The receiving unit of named frames is then in responsibility to execute all necessary checks to detect communication errors.

17 Diagnostics

17.1 UDS Diagnostics

The elobau MFA supports UDS off-board diagnostics including

UDS Service	Description
Communication Control	En/Disable Functional CAN communication
Read/Write ECU Parameter	Read/write ECU specific parameters, e.g. HW/SW versions, serial numbers, diagnostic data, ...
Variant Coding	Read/write variant coding bits to en/disable operation of the mapped control element.
I/O control	Set LEDs manually
Control DTC Setting	En/disable logging DTCs to fault memory
Fault Memory	Read/clear the fault memory.

Table 46 UDS Services overview

Note: The complete MFA UDS diagnostics functionality is stated in a *.cdd/ *.odx file.

17.2 J1939 Diagnostics

17.2.1 Available Diagnostic Messages (DM)

ID	Description
DM1	Active Diagnostic Trouble Codes
DM2	Previous active Trouble Codes
DM3	Diagnostic Data Clear/Reset of Previously Active DTCs
DM11	Diagnostic Data Clear/Reset for Active DTCs

Table 47 Overview available J1939 DM

17.2.2 DM1- Active Diagnostic Trouble Codes

17.2.2.1 Overview

DM1 communicates all active diagnostic trouble codes (DTCs).
Active DTCs are all DTCs with status bit

- testFailed (Bit0)

set. Ref_9, D2.3- Table D.3

17.2.2.2 Transmission Rate

The DM1 message will be transmitted, regardless of the presence or absence of any DTC*, once every second and on state change. To prevent a high message rate due to intermittent faults that have a very high frequency, only one state change per DTC per second will be transmitted.

***Note**: In case of absence of any DTC the transmission of DM1 message could be deactivated with a configuration option.

The DM1 can also be requested over the J1939 request PGN, see, 8.5 - Definition of J1939 request PGN

17.2.2.3 PGN Definition

The message/ signal layout is according to [Ref 4, 5.7.1] defined in the project dbc file.

17.2.2.4 Boundary conditions

The maximum number of DTS within one DM1 message are limited to 32. This was done to guarantee the DM1 cycle time of one second. If there are more than 32 faults active- UDS has to be used to read the complete fault memory.

Calculation of the maximal number of DTC which can be sent over BAM within one second:

$$(((((DM1_{TR}/BAM_{\Delta t})-1)*BAM_{PL})-4u)/sizeof(DTC_T)) = \lfloor 32,25 \rfloor = \underline{32}$$
$$with$$
$$DM1_{TR} : DM1TransmissionRate = 1s$$
$$BAM_{\Delta t} = 50ms$$
$$1 = TP.CMmessage$$
$$BAM_{PL} : BAMPayloadofTP.DTframe = 7bytes$$
$$4 = LampStatus(4bytes)$$
$$sizeof(DTC_T) = 4bytes$$

17.2.3 DM2- Previous active Diagnostic Trouble Codes

17.2.3.1 Overview

The DM2 communicate the previous active diagnostic trouble codes (DTCs).
Previous active DTCs are all DTCs with at least ONE of following status bits

- testFailedThisOperationCycle (Bit1)
- pendingDTC (Bit2)
- confirmedDTC (Bit3)

active. [Ref 9, D2.3- Table D.3]

17.2.3.2 PGN Definition

The message/ signal layout is according to [Ref 4, 5.7.1] defined in the project dbc file.

17.2.3.3 Boundary conditions

The maximum number of DTS within one DM2 message are limited to 32. If there are more than 32 previous faults active- UDS has to be used to read the complete fault memory, see also17.2.2.4-1

17.2.4 DM3- Diagnostic Data Clear/Reset Of Previously Active DTCs

To clear the previously active DTCs, DM3 ist requestable- see 8.5 - Definition of J1939 request PGN

17.2.5 DM11- Diagnostic Data Clear/Reset For Active DTCs

To clear the active DTCs, DM11 ist requestable- see 8.5 - Definition of J1939 request PGN

17.3 Fault Memory

17.3.1 Fault Memory Status byte policy

The lower 4bit of the status byte are used by the MIDI:

Status bit position	Description [Ref_9, D2]
0 (LSB): testFailed	This bit shall indicate the result of the most recently performed test. A logical '1' shall indicate that the last test failed.
1: testFailedThisOperationCycle	This bit indicate whether or not a diagnostic test has reported a testFailed result at any time during the current operation cycle (or that a testFailed result has been reported during the current operation cycle and after the last time a call was made to ClearDiagnosticInformation). Reset to logical '0' when a new operation cycle is initiated or after a call to ClearDiagnosticInformation. If this bit is set to logical '1', it remains '1' until a new operation cycle is started.
2: pendingDTC	This bit shall indicate whether or not a diagnostic test has reported a testFailed result at any time during the current or last completed operation cycle. The criteria to set the pendingDTC bit and the TestFailedThisOperationCycle bit are the same. The difference is that the testFailedThisOperationCycle is cleared at the beginning of each operation cycle and the pendingDTC bit is not cleared until an operation cycle has completed where the test has passed at least once and never failed.
3: confirmedDTC	This bit indicates whether a malfunction was detected in the last operating cycle(s). (Read out of the persistence memory during bootup) A confirmedDTC does not always indicate that the malfunction is present at the time of the request. (testFailed can be used to determine if a malfunction is present at the time of the request).

Table 48 Fault Memory Status byte definitions

17.3.2 Occurrence counter policy

To increment the Occurrence Counter the policy of [Ref_4, 5.7.1.12] was implemented:

The 7-bit occurrence count field contains the number of times a fault has been independently detected. The occurrence count is reported as 1 the first time the DTC is detected. The occurrence count is not incremented again, until after the DTC has gone to the previously active state and then back active the DTC state when subsequently detected. At this point the occurrence count would be reported as 2. This continues until the DTC has been independently detected 126 times. The occurrence count shall not be incremented from 126 to 127 - it shall remain at 126 until cleared by DM3 or DM 11. If an occurrence count is not available, then this field should be set to all binary ones (127). The occurrence count is not incremented just due to an ignition key-off and ignition key-on. The diagnostic system shall have monitored the system or component (e.g. DTC) to see that it is no longer malfunctioning in order to declare it previously active.

The MIDI increments the counter whenever a testFailed condition occurs and the pendingDTC bit is set to "0". (se e 17.3.1 - Fault Memory Status byte policy)

17.3.3 Device specific DTCs

Device specific DTCs will be generated with the appropriate SPN of the defined CAN signal(s) of a device plus a FMI which delivers a suitable fault description.

Example:

If the second analogue input track of a Joystick Axis (Device type: Position Sensor, CAN signal configured: Pos_01_01, SPN: 520301) is in undervoltage condition, following DTC will be generated: SPN(520301) + FMI(6) = 0x6DF0E6.

Device	CAN signal	+ FMIs										
		2	3	4	5	6	7	8	11	12	13	14
Balanced Switch	Sig0	x	x	x			x			x		x
	Sig1						x			x		
Switch	Sig0	x					x			x		x
Diagnostic Switch	Sig0	x	x	x			x			x	x	x
Encoder	Sig0	x	x	x		x[e]	x[e]	x[e]				x
PTO	Sig0	x	x	x			x		x	x	x	x
	Sig1	x	x	x								
Position Sensor	Sig0	x[p]	x	x	x[p]	x[p]				x		x

Table 49 Existing SPN/ FMI combinations

[e]: Encoder FMIs 5,6, and 7 have the same meaning for the second input as FMIs 2,3 and 4 for the first line.

[p]: Only available for dual track devices. FMI 5,6 have the same meaning for the second track as FMI 3 and 4 for the first track.

Note: Merged cells means that this FMI will set for both CAN signal SPNs if the fault will be detected.

FMI	Description
2	General: Invalid voltage value
	Position sensor: Coherence check failed
3	Overvoltage
4	Undervoltage
5	([p]) Position sensor: Overvoltage second track
6	([p]) Position sensor: Undervoltage second track
	([e]) Encoder: Invalid voltage value of the second track
7	Stuck close
	([e]) Encoder: Overvoltage value of the second track

8	(e) Encoder: Undervoltage value of the second track
11	PTO: Error caused by the related device
12	Stuck at startup
	Position sensor: Force to Zero (see 13.1.5)
13	Diagnostic Switches: Exceed debounce counter for zone between pressed and released state
	PTO: unexpected condition (4. state)
14	Initialization of component failed. (parameter, curves, ..)

Table 50 FMI descriptions