GEFRAN

KMC Safety

User Manual

Code 80795 **Edition** 08-2024

1. SOMMARIO

1. INTRODUCTION

The GEFRAN KMC Safety transducer is a Digital Pressure Sensor with CANopen interface. It implements the standard CANopen Safety communications protocol defined by CiA (CAN in Automation).
The CANopen and CANopen Safety standards supported by the device are listed in the following table.

CiA standard	Description	Version
CiA 301	CANopen application layer and communication profile	4.2.0
CiA 305	CANopen layer setting services (LSS) and protocols	3.0.1
CiA 404-1	Device profile for measuring devices and closed-loop controllers	1.0.1
CiA 890	Presentation of SI units and prefixes	1.0.0
EN50325-5	Functional safety communication based on EN 50325-4	-
CiA 319	CANopen implementation and configuration specification for safety-related device	1.0.1

Table 1. Supported CANopen and CANopen Safety standards

This document describes the CANopen Safety implementation on the GEFRAN KMC Safety device. It is addressed to CANopen Safety network system integrators and to CANopen Safety device designers who already know the content of the above-mentioned standards defined by CiA.

The details of aspects defined by CANopen Safety do not pertain to the purpose of this text. For further information on the CANopen and CANopen Safety protocol see https://www.can-cia.org/

2. GET STARTED PROCEDURE

2.1. NODE PARAMETERS SETTING

Before connecting the GEFRAN KMC Safety device to a fully configured and working CAN bus, some basic configuration actions have to be performed. The configuration involves the Node-ID and the Baud rate of the CANopen device.

The configuration is mandatory if at least one of these conditions is true:
1) The Node-ID of the GEFRAN KMC Safety device is identical to the Node-ID of another device connected to the CAN bus.

2) The GEFRAN KMC Safety device operates with a baud rate different from the CAN bus baud rate.

If the condition at point 2 in not verified, the configuration can also be performed on that CAN bus, but all the other devices on the CAN bus should be taken in power-off state during the configuration process in order to avoid errors or conflicts.
If the baud rate configuration has to be performed, the GEFRAN KMC Safety device must be connected to a CAN bus that works at the same baud rate of the sensor. The baud rate of the actual CAN bus (with all devices connected to it) can also be temporary set equal to the sensor baud rate until configuration is done. The configuration is made using LSS (Layer Setting Services).

Switching to LSS configuration mode

The first operation is to switch the sensor into LSS configuration mode. If the sensor is the only device on the CAN bus (with the LSS master), the LSS Switch State Global command can be used.

Source	COB-ID	DLC	Data	Destination
Controller	7E5h	08h	04h; 01h; 00h; 00h; 00h; 00h; 00h; 00h	Sensor

Figure 1. LSS Switch State Global command

If there are other devices on the CAN bus (except the LSS master), the LSS Switch State Selective command must be used. Refer to the LSS Services section for details.

Setting the Node-ID

If the Node-ID of the sensor has to be changed, the LSS Configure Node-ID command must be used.

Source	COB-ID	DLC	Data	Destination
Controller	7E5h	08h	11h; **20h(*)**; 00h; 00h; 00h; 00h; 00h; 00h	Sensor
Sensor	7E4h	08h	11h; **00h(**)**; 00h; 00h; 00h; 00h; 00h; 00h	Controller

Figure 2. LSS Configure Node-ID command

(*) the Node-ID value to be configured, within 1..127 (32 in this example).

(**) if value is 1, it means Node-ID out of range, i.e. the command was not accepted.

SAFETY NOTE
For CANopen Safety devices it is recommended to configure the Node-ID value within 1 and 64. In this way the COB-IDs of SRDOs are automatically set by default in the standard forms (FFh + 2 • Node-ID) for COB-ID 1, and (100h + 2 • Node-ID) for COB-ID 2. If the Node-ID is greater than 64 the default values of COB-IDs are set by default to the values 17Fh and 180h for COB-ID1 and COB-ID 2 respectively.

Setting the baud rate

If the baud rate of the sensor has to be changed, the LSS Configure Bit Timing Parameters command must be used.

Source	COB-ID	DLC	Data	Destination
Controller	7E5h	08h	13h; 00h; **02h(*)**; 00h; 00h; 00h; 00h; 00h	Sensor
Sensor	7E4h	08h	13h; **00h(**)**; 00h; 00h; 00h; 00h; 00h; 00h	Controller

Figure 3. LSS Configure Bit Timing Parameters command

(*) the table-index of the corresponding bit rate (500kbit/s in this example).
Refer to the Table index in the LSS Configure Bit Timing Parameters section for details.

(**) If the value is 1 means that the bit timing is not supported; the command was not accepted.

Storing configuration settings

To save the previously configured Node-ID and Baud rate permanently (to non-volatile memory of the device) the LSS Store Configuration command must be used.

Source	COB-ID	DLC	Data	Destination
Controller	7E5h	08h	17h; 00h; 00h; 00h; 00h; 00h; 00h; 00h	Sensor
Sensor	7E4h	08h	17h; **00h(*)**; 00h; 00h; 00h; 00h; 00h; 00h	Controller

Figure 4. LSS Store Configuration command

(*) value other than 0, means store operation failed.

Verifying configuration setting

To check if the configuration settings of the device have been correctly executed and stored, proceed as follows:

1. power off the device
2. set the baud rate of the CAN bus to the correct value
3. power on the device

If the boot-up message is received, it means that the device baud rate setting is correct. The Node-ID of the device is contained inside the COB-ID of the message (boot-up COB-ID = 700h + Node-ID).

The format of the boot-up message is specified in the following figure.

Source	COB-ID	DLC	Data	Destination
Sensor	700h + Node-ID	01h	00h	Controller

Figure 5. Boot-up message format

SAFETY NOTE If the Node-ID is changed, the SRDO configuration is set to invalid at next power on or after a NMT reset command. In this case the device goes in Pre-operational state and the safety communication configuration must be validated again.

2.2. COMMUNICATION PARAMETERS SETTING

After configuring the node parameters, the sensor can be integrated in the CANopen network. When powering on, the sensor transmits the boot-up message, and it goes into the Pre-operational state.

Before requesting process data, configuration of communication parameters of the sensor can be performed. Configuration of communication parameters is made through SDO Services (Service Data Objects). Through SDO services, it is possible for example to change the refresh-time of the SRDO, the transmission type of the PDO (Process Data Object) selecting the synchronous (through SYNC messages) or asynchronous (through event-timer) mode, change the transmission time (event timer) of the asynchronous PDO, change the PDO mapping, etc.

It is possible to save changed parameters in non-volatile memory accessing the Store Parameters object through SDO, or restore default parameters with the Restore Default Parameters object.

It is possible to access all the objects specified in the Object Dictionary of the device (see Object Dictionary section).

SDO Services are available in Pre-operational and Operational states only (see NMT Services section).

SAFETY NOTE By default (out of the factory), after a “restore default parameters”, or after a parameter change, the communication parameter set is invalidated. The device cannot go in Operational state until the validation of the communication parameter set is performed. Please refer to the “Configuration of safe communication parameters” paragraph.

2.3. APPLICATION PARAMETERS SETTING

After configuring the node parameters, the sensor can be integrated in the CANopen network. When powering on, the sensor transmits the boot-up message, and it goes into the Pre-operational state.

Before requesting process data, configuration of operating parameters of the sensor can be performed. Configuration of operating parameters is made through SDO Services (Service Data Objects). Through SDO services, it is possible for example to change the decimal digits of the pressure data (integer32 format) or adjust the zero offset of the device.

It is possible to save changed parameters in non-volatile memory accessing the Store Parameters object through SDO, or restore default parameters with the Restore Default Parameters object.

It is possible to access all the objects specified in the Object Dictionary of the device (see Object Dictionary section).

SDO Services are available in Pre-operational and Operational states only (see NMT Services section).

SAFETY NOTE By default (out of the factory), after a “restore default parameters”, or after a safety-related parameter change, the safety-related application parameter set is invalidated. The device cannot go in Operational state until the validation of the safety-related application parameter set is performed. Please refer to the “Configuration of safety-related application parameters” paragraph.

2.4. REQUESTING PROCESS DATA

The GEFRAN KMC Safety device provides two SRDOs (Safety-related data object): SRDO1 and SRDO2.
By default, only one of them is configured as “valid for transmit”, the other one is set as invalid, so it does not transmits.
SRDO1 is used when the KMC Safety device is ordered with pressure data in integer32 format:

- 1st application mapped object: safety pressure data integer32 – normal (object 5130h, sub-index 1)
- 2nd application mapped object: safety pressure data integer32 – inverted (object 5130h, sub-index 2)
- 3rd application mapped object: safety status - normal (object 5150h, sub-index 1)
- 4th application mapped object: safety status - inverted (object 5150h, sub-index 2)

SRDO2 is used when the KMC Safety device is ordered with pressure data in float format.

- 1st application mapped object: safety pressure data float – normal (object 5030h, sub-index 1)
- 2nd application mapped object: safety pressure data float – inverted (object 5030h, sub-index 2)
- 3rd application mapped object: safety status - normal (object 5150h, sub-index 1)
- 4th application mapped object: safety status - inverted (object 5150h, sub-index 2)

The GEFRAN KMC Safety device also provides one Transmit PDO (TPDO1), with two mapped objects by default:

- 1st application mapped object: pressure data (object 9130h or 6130h)
- 2nd application mapped object: status (object 6150h)

A third object, temperature data, can be mapped (see PDO mapping).
For the KMC Safety device the TPDO1 is disabled by default.

SAFETY NOTE
Only the complete SRDO data (SRDO normal and SRDO inverted data frames) must be evaluated for functional safe application. The TPDO1 data is not safe.

SRDO data format

Safety Pressure and Safety status data are mapped in SRDO as shown in the following figure.

	COB-ID	DLC	D0	D1	D2	D3	D4
SRDO normal	FFh + 2 • Node-ID	5	SAFETY Pressure data (normal)				SAFETY Status (normal)
SRDO inverted	100h + 2 • Node-ID	5	SAFETY Pressure data (inverted)				SAFETY Status (inverted)

Figure 6. SRDO mapped data

The physical unit of the pressure data can be set through the object 6131h (AI physical unit PV).

For the KMC Safety device, if SRDO1 is used, the pressure data mapped in SRDO is of integer type (i.e. mapped object is 5130h), so the value has to be rescaled considering the value of the object 6132h (AI decimal digits).

For the KMC Safety device, if SRDO2 is used, the pressure data mapped in SRDO is of float type (i.e. mapped object is 6130h), so the value has not to be rescaled.

Byte ordering of pressure data inside SRDO follows the LSB..MSB ordering scheme.

TPDO1 data format

Pressure and status data are mapped in TPDO1 as shown in the following figure.

COB-ID	DLC	D0	D1	D2	D3	D4
180h + Node-ID	5	Pressure data				Status

Figure 7. TPDO1 mapped data

The physical unit of the pressure data can be set through the object 6131h (AI physical unit PV). If the pressure data mapped in TPDO1 is of integer type (i.e. mapped objects are 2090h or 9130h), the value has to be rescaled considering the value of the object 6132h (AI decimal digits).

If the pressure data mapped in TPDO1 is of float type, the value has not to be rescaled.

Byte ordering of pressure data inside TPDO1 follows the LSB..MSB ordering scheme.

SRDO and TPDO1 data transmission

The transmission of the Safety-related data object (SRDO) or the Process Data Object (TPDO) is made when the sensor is in Operational state. To start data transmission, the master sends the NMT “Start” command, as shown in the following figure.

Source	COB-ID	DLC	Data	Destination
Controller	000h	02h	01h; **00h(*)**	Sensor

Figure 8. NMT “Start” command

(*) 00h: all nodes, nnh: only the node with Node-ID equal to nnh

To stop data transmission the master sends the “Enter NMT Pre-operational state” command, as shown in the following figure.

Source	COB-ID	DLC	Data	Destination
Controller	000h	02h	80h; **00h(*)**	Sensor

Figure 9. NMT “Enter NMT pre-operational” command

(*) 00h: all nodes, nnh: only the node with Node-ID equal to nnh

2.5. DEFAULT PARAMETERS SETTINGS

The default parameters settings for the KMC Safety device are listed in the following table

Parameter Name/Description	Object (Index,Subindex)	Default Value
Node-ID	-	1(*)
Transmission speed	-	250 kbps(*)
Number of mapped objects	1A00,0	2
PDO mapping, 1st object	1A00,1	9130h (AI input PV, integer32)(*) or 6130h (AI input PV, float)(*)
PDO mapping, 2nd object	1A00,2	6150h (AI status)
PDO mapping, 3rd object	1A00,3	2091h (Temperature)
COB-ID SYNC	1005,0	80h
COB-ID EMCY	1014,0	80h + Node-ID
COB-ID SDO rx	1200,1	600h + Node-ID
COB-ID SDO tx	1200,2	580h + Node-ID
COB-ID TPDO	1800,1	80000180h + Node-ID

Table 2. Parameters default values

(*) Parameter values can be selected during the ordering phase of the GEFRAN KMC Safety sensor. The allowed range of selectable values is listed in the following table.
(**) Parameter values depend on the factory configuration of the device (ordering code of the sensor).

Parameter Name/Description	Selectable values
Transmission speed	20, 50, 125, 250, 500, 800, 1000 kbps
Node-ID	1..127
PDO mapping, 1st object	9130h (AI input PV, integer32) 6130h (AI input PV, float)

Table 3. Selectable parameters during ordering phase

The default safety-related parameters settings for the KMC Safety device are listed in the following table:

Parameter Name/Description	Object (Index,Subindex)	Default Value
SRDO1 Information direction	1301,1	1(**) or 0(**)
SRDO1 Refresh-time	1301,2	25
SRDO1 SRVT	1301,3	20
SRDO1 Transmission type	1301,4	254
SRDO1 COB-ID 1	1301,5	FFh + 2 • Node-ID
SRDO1 COB-ID 2	1301,6	100h + 2 • Node-ID
SRDO2 Information direction	1302,1	0(**) or 1(**)
SRDO2 Refresh-time	1302,2	25
SRDO2 SRVT	1302,3	20
SRDO2 Transmission type	1302,4	254
SRDO2 COB-ID 1	1302,5	FFh + 2 • Node-ID
SRDO2 COB-ID 2	1302,6	100h + 2 • Node-ID
AI ADC sample rate	6114,1	1000
AI input scaling 1 PV (float)	6121,1	0
AI input scaling 2 PV (float)	6123,1	Pressure Full Scale float (**)
AI Input offset (float)	6124,1	0
AI physical unit PV	6131,1	004E0000h (bar)** or 00AB0000h (psi)**
AI decimal digits	6132,1	2 (bar)** or 1 (psi)**
AI filter type	61A0,1	0
AI filter constant	61A1,1	1
AI input scaling 1 PV (integer32)	9121,1	0
AI input scaling 2 PV (integer32)	9123,1	Pressure Full Scale integer32 (**)
AI Input offset (integer32)	9124,1	0

Table 4. Safety-related parameters default values

(**) Parameter values depend on the factory configuration of the device (ordering code of the sensor).

The values of the above listed parameters can also be modified through the SDO services.

3. LSS SERVICES

LSS services and protocols are used to inquire or to change the settings of three parameters of the CANopen device:
- Node-ID of the CANopen device
- Bit timing parameters of the physical layer (bit rate)
- LSS address compliant to the identity object (1018h)

3.1. LSS SWITCH STATE SERVICES

LSS switch state global

By means of this service, the LSS master device switches all LSS slave devices in the network into LSS waiting state or LSS configuration state.

The LSS master sends this message to switch the LSS slave(s) into configuration state:

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	04h	01h	00h	00h	00h	00h	00h	00h

Figure 10. LSS switch state global - configuration state - message

The LSS master sends this message to switch back the LSS slave(s) to waiting state:

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	04h	00h	00h	00h	00h	00h	00h	00h

Figure 11. LSS switch state global - waiting state - message

LSS switch state selective

By means of this service, the LSS master device switches the LSS slave device, whose LSS address equals the LSS address specified by the messages, into LSS configuration state.

The transmitted LSS address shall be equal to the identity object (object 1018h) of the related LSS slave.

The LSS address for the GEFRAN KMC Safety device is specified in the following table.

	Address Field	Value
LSS Address	Vendor-ID	00000093h
	Product code	53434D4Bh(*) for INT32 version or 73434D4Bh(*) for FLOAT version
	Revision Number	Actual KMC Safety r.n.(**)
	Serial Number	Actual KMC Safety s.n. (printed on the label)(***)

Figure 12. KMC LSS Address

(*) If read as string data type, it equals to the signature “KMCS” for 53434D4Bh or "KMCs" for 73434D4Bh (KMC with CANopen Safety Output)

(**) Actual Revision number can vary. The user can inquire the Revision number with LSS Inquire Revision Number command (see LSS Inquire Services) , or through an SDO read command of the object (1018, 3).

(***) Actual Serial number is device specific. It is printed on the label attached to the GEFRAN KMC Safety transducer case, or it can be inquired with the LSS Inquire Serial Number command (see LSS Inquire Services), or through an SDO read command of the object (1018, 4). The value printed on the label must be intended as expressed in hexadecimal format.

The LSS master sends this message sequence to switch the GEFRAN KMC Safety into configuration state (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	40h	93h	00h	00h	00h	00h	00h	00h
7E5h	Rx	8	41h	4Bh	4Dh	43h	53h(***)	00h	00h	00h
7E5h	Rx	8	42h	01h(*)	00h(*)	01h(*)	00h(*)	00h	00h	00h
7E5h	Rx	8	43h	34h(**)	12h(**)	01h(**)	15h(**)	00h	00h	00h
7E4h	Tx	8	44h	00h	00h	00h	00h	00h	00h	00h

Figure 13. LSS switch state selective message sequence

(*) The Revision number used for this example is 00010001h
(**) The Serial number used for this example is: 15011234h
(***) The Product code used for this example is 53434D4Bh ("KMCS")

The Serial Number is assigned by GEFRAN to the KMC Safety sensor in accordance with the following scheme.

SERIAL NUMBER : YY WW NNNN, where:
YY: year of production
WW: week of production
NNNN: progressive number inside the week, starting from 1

3.2. LSS CONFIGURATION SERVICES

LSS configure Node-ID

By means of this service, the LSS master device configures the pending Node-ID of the LSS slave device. The LSS slave device confirms the success or the failure of the service execution.

The allowed Node-ID values are in the range 1..127 (01h..7Fh). The LSS master sends this message to configure the value of the Node-ID (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	11h	Node ID	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	11h	Error code	00h	00h	00h	00h	00h	00h

Figure 14. LSS configure Node-ID message

where Error code: 00h (Protocol successfully completed) or 01h (Node-ID out of range)

The pending Node-ID becomes active only after the master sends a NMT reset communication command. The Node-ID is not automatically saved to the non-volatile memory of the slave device. In order to save the persistent Node-ID, refer to the LSS store configuration service.

When the pending Node-ID becomes active, or when the Node-ID is stored in non volatile memory, the following COB-IDs are automatically updated according to their default values:

- COB-ID EMCY (1014h)
- COB-ID SDO rx (1200h, sub 1)
- COB-ID SDO tx (1200h, sub 2)
- COB-ID TPDO (1800h, sub 1)
- COB-IDs SRDO1 (1301h, sub 5 and 6)
- COB-IDs SRDO2 (1302h, sub 5 and 6)

At the power on, the active Node-ID equals the persistent Node-ID.

SAFETY NOTE

The LSS configure Node-ID command can impact on the safe communcation configuration.
If the command is executed, the safe communication configuration can be set to invalid if the new active Node-ID is different from the previous active Node-ID. In this case the device goes in Pre-operational state and the safe communication configuration must be validated again.

LSS configure bit timing parameters

By means of this service, the LSS master device configures the pending bit rate of the LSS slave device. The LSS slave device confirms the success or the failure of the service execution.

The allowed bit rate values with the associated table index, are specified in the following table.

Table index	Bit rate (kbit/s)
0	1000
1	800
2	500
3	250
4	125
6	50
7	20

Table 5. index for bit timing table

The LSS master sends this message to configure the bit rate (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	13h	00h	Table index	00h	00h	00h	00h	00h
7E4h	Tx	8	13h	Error code	00h	00h	00h	00h	00h	00h

Figure 15. LSS configure bit timing message

where Error code: 00h (Protocol successfully completed) or 01h (Bit timing not supported).

The pending bit rate becomes active only after the master sends the LSS activate bit timing parameter service, or with the next power-on after the execution of the LSS store configuration service.

The bit rate is not automatically saved to the non-volatile memory of the slave device. In order to save the persistent bit rate, refer to the LSS store configuration service.

At the power on, the active bit rate equals the persistent bit rate.

LSS activate bit timing parameters

By means of this service, the LSS master activates simultaneously the bit rate at the LSS communication interface of all CANopen devices in the network.

Therefore the reception of this command triggers at the LSS slave the copying process of the currently pending bit rate to the active bit rate.

The LSS master sends this message to activate the bit timing parameters:

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	15h	Switch delay		00h	00h	00h	00h	00h

Figure 16. LSS activate bit timing parameters message

where Switch delay is the time, in ms, multiplied by 2 when the new bit timing settings becomes active (Intel format byte ordering)

The Switch delay parameter specifies the length of two delay periods of equal length, which are necessary to avoid operating the network with different bit rates.

After “Switch delay” has elapsed the first time after service indication, the slave device stops communicating on the bus.

After “Switch delay” has elapsed one more time, the slave device resume the communication on the bus using the new active bit rate.

LSS store configuration

By means of this service, the LSS master device requests the LSS slave device to store the configured local layer settings (Node-ID and bit rate) to non-volatile memory. On execution of this command the pending Node-ID and bit rate are copied to the persistent Node-ID and bit rate.

The LSS master sends this message to store the LSS configuration (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	17h	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	17h	Error code	00h	00h	00h	00h	00h	00h

Figure 17. LSS store configuration message

where Error code: 00h (Protocol successfully completed) or 02h (Storage media access error).

SAFETY NOTE

The LSS store configuration command can impact on the safe communication configuration.
If the command is executed, the safe communication configuration can be set to invalid if the new persistent Node-ID is different from the previous persistent Node-ID. In this case, after the next reset of the application or communication, or after a power-cycle, the device goes in Pre-operational state. The safety communication configuration must be validated again.
The LSS store configuration command is not allowed when the device is in Operational state.

3.3. LSS INQUIRY SERVICES

LSS inquire Node-ID

By means of this service, the LSS master device inquires the active Node-ID of the LSS slave device that is in LSS configuration state. The LSS slave device responds indicating his active Node-ID.

The LSS master sends this message to inquire the Node-ID (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	5Eh	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	5Eh	Node ID	00h	00h	00h	00h	00h	00h

Figure 18. LSS inquire Node-ID message

where Node-ID is the LSS slave's active Node-ID.

LSS inquire LSS address

By means of this service, the LSS master device inquires the LSS address of the LSS slave device. The LSS slave device responds indicating his LSS address.

The LSS master sends this message to inquire the Vendor-ID (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	5Ah	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	5Ah	Vendor ID				00h	00h	00h

Figure 19. LSS inquire identity Vendor-ID message

where Vendor-ID is the LSS slave's identity Vendor-ID (Intel format byte ordering).

The LSS master sends this message to inquire the Product-code (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	5Bh	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	5Bh	Product code				00h	00h	00h

Figure 20. LSS inquire identity Product-code message

where Product-code is the LSS slave's identity Product-code (Intel format byte ordering).

The LSS master sends this message to inquire the Revision number (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	5Ch	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	5Ch	Revision number				00h	00h	00h

Figure 21. LSS inquire identity Revision number message

where Revision number is the LSS slave's identity Revision number (Intel format byte ordering).

The LSS master sends this message to inquire the Serial number (the slave sends the response message):

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	5Dh	00h	00h	00h	00h	00h	00h	00h
7E4h	Tx	8	5Dh	Serial number				00h	00h	00h

Figure 22. LSS inquire identity Serial number message

where Serial number is the LSS slave's identity Serial number (Intel format byte ordering).

3.4. LSS IDENTIFICATION SERVICES

LSS identify remote slave

By means of this service, the LSS master device requests all LSS slave devices to identify themselves by means of the 'LSS identify slave' service, whose LSS address meets the LSS_Address_sel.

The LSS_Address_sel consists of a single vendor-ID and a single product code and a span of revision and serial numbers determined by a low and high number.

The protocol defined in the following figure implements the LSS identify remote slave service. All LSS slave devices with matching vendor-ID and product-code and whose major revision-number and serial-numbers are located within the given ranges, identify themselves with the LSS identify slave service.

The boundaries are included in the interval.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	46h	Vendor-ID				00h	00h	00h
7E5h	Rx	8	47h	Product-code				00h	00h	00h
7E5h	Rx	8	48h	Revision number low				00h	00h	00h
7E5h	Rx	8	49h	Revision number high				00h	00h	00h
7E5h	Rx	8	4Ah	Serial number low				00h	00h	00h
7E5h	Rx	8	4Bh	Serial number high				00h	00h	00h

Figure 23. LSS identify remote slave message sequence

Where:

Vendor-ID is the LSS slave's identity Vendor-ID (Intel format byte ordering).

Product-code is the LSS slave's identity Product-code (Intel format byte ordering).

Revision number low and Revision number high identity the Revision number span (Intel format byte ordering).

Serial number low and Serial number high identity the Serial number span (Intel format byte ordering).

LSS identify slave

By means of this service, an LSS slave device indicates that it is a slave device with an LSS address within the LSS_ Address_sel given by an LSS identify remote slave service executed prior to this service.

The protocol is defined in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E4h	Tx	8	4Fh	00h	00h	00h	00h	00h	00h	00h

Figure 24. LSS identify slave message

LSS identify non-configured remote slave

By means of this service, the LSS master device requests all LSS slave devices to identify themselves by means of the 'LSS identify non-configured slave' service, who got stuck in NMT Initialization state, whose pending Node-ID is invalid (FFh) and who have no active Node-ID.

The protocol is defined in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E5h	Rx	8	4Ch	00h	00h	00h	00h	00h	00h	00h

Figure 25. LSS identify non-configured slave message

LSS identify non-configured slave

By means of this service, an LSS slave device indicates that it is an LSS slave device that got stuck in NMT Initialization state, owns an invalid (FFh) pending Node-ID and no active Node-ID.

This service is executed in case a LSS identify non-configured remote slave service was initiated by the LSS master device.

The protocol is defined in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
7E4h	Tx	8	50h	00h	00h	00h	00h	00h	00h	00h

Figure 26. LSS identify non-configured slave message

4. SDO SERVICES

SDO services provide direct access to the object entries of a CANopen device's object dictionary. The device initiating the SDO transfer is called the SDO client.

The CANopen device hosting the accessed object dictionary is called the SDO server.

SDO download

The SDO client uses this service for transferring data to the object dictionary of the SDO server. SDO download service is therefore used to configure (write) communication, device and manufacturer parameters of the GEFRAN KMC Safety device.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
600h + Node-ID	Rx	8	Cs	Index		Sub index	Data			
580h + Node-ID	Tx	8	60h	Index		Sub index	00h	00h	00h	00h

Figure 27. SDO download message

where:

Cs is the command specifier of the SDO download request, whose value depends on the number of bytes of Data field:

Cs=23h 4 transmitted data bytes
Cs=27h 3 transmitted data bytes
Cs=2Bh 2 transmitted data bytes
Cs=2Fh 1 transmitted data bytes

Data is the data to be copied in the object dictionary value (Intel format byte ordering)
Index is the object dictionary parameter index (Intel format byte ordering)
Sub index is the object dictionary parameter sub index

SDO upload

The SDO client uses this service for transferring the data from the server (owner of the object dictionary) to the client. SDO upload service is therefore used to check (read) communication, device and manufacturer parameters of the GEFRAN KMC Safety device.

COB-ID	**Rx/Tx**	**DLC**	**Data**							
			D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
600h + Node-ID	Rx	8	40h	Index		Sub index	00h	00h	00h	00h
580h + Node-ID	Tx	8	42h	Index		Sub index	Data			

Figure 28. SDO upload message

where:

Index is the object dictionary parameter index (Intel format byte ordering)
Sub index is the object dictionary parameter sub index
Data is the data value read from object dictionary (Intel format byte ordering)

SDO abort transfer

The SDO abort transfer service aborts the SDO download or the SDO upload service of an SDO referenced by its number.
As result of an SDO abort transfer event, the SDO server sends this message to the SDO client:

COB-ID	**Rx/Tx**	**DLC**	**Data**							
			D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
580h + Node-ID	Tx	8	80h	Index		Sub index	Abort code			

Figure 29. SDO abort response message

where:

Index is the object dictionary parameter index (Intel format byte ordering)
Sub index is the object dictionary parameter sub index
Abort code explain the reason of the SDO abort event.

4.1. OBJECT DICTIONARY

The object dictionary of the GEFRAN KMC Safety device is specified in the following tables.

Communication Profile Area

Index	*Sub index*	*Name*	*Type*	*Access*	*Default value*	*Comment*
1000h	0	Device type	Unsigned32	RO	80020194h	Analogue input with device-specific PDO mapping and ds404 device profile
1001h	0	Error register	Unsigned8	RO	-	0x00: no error 0x81: generic error (manufacturer specific)
1002h	0	Manufacturer status register	Unsigned32	RO	-	Safety-related errors detected by device
1003h	0	Pre-defined error field	Unsigned8	RW	-	Number of errors.
	1..32		Unsigned32	RO	-	Standard error field (1 to 32)
1005h	0	COB-ID SYNC	Unsigned32	RW	00000080h	Configured COB-ID of the synchronization object (SYNC)
1008h	0	Manufacturer device name	Visible string	RO	KMCS	Name of the device
1009h	0	Manufacturer HW version	Visible string	RO	-	Hardware version description
100Ah	0	Manufacturer SW version	Visible string	RO	-	Software version description
100Ch	0	Guard time	Unsigned16	RW	0	Multiplied with object 100Dh gives the lifetime value used by the node guarding protocol
100Dh	0	Life time factor	Unsigned8	RW	0	Multiplied with object 100Ch gives the lifetime value used by the node guarding protocol
1010h	0	Store parameters	Unsigned8	RO	3	Highest sub-index supported
	1		Unsigned32	RW(**)	1	Writing the signature “save” (73h, 61h, 76h, 65h) stores all parameters in flash memory
	2		Unsigned32	RW(**)	1	Writing the signature “save” (73h, 61h, 76h, 65h) stores only communication parameters in flash memory
	3		Unsigned32	RW(**)	1	Writing the signature “save” (73h, 61h, 76h, 65h) stores only application parameters in flash memory
1011h	0	Restore default parameters	Unsigned8	RO	3	Highest sub-index supported
	1		Unsigned32	RW(**)	1	Writing the signature “load” (6Ch, 6Fh, 61h, 64h) restores all parameters in flash to their default values
	2		Unsigned32	RW(**)	1	Writing the signature “load” (6Ch, 6Fh, 61h, 64h) restores only communication parameters in flash to their default values
	3		Unsigned32	RW(**)	1	Writing the signature “load” (6Ch, 6Fh, 61h, 64h) restores only application parameters in flash to their default values
1014h	0	COB-ID EMCY	Unsigned32	RO	00000080h + Node-ID	Configured COB-ID for the EMCY write service
1015h	0	Inhibit time EMCY	Unsigned16	RW	0	Configured inhibit time for the EMCY service
1017h	0	Producer heartbeat time	Unsigned16	RW	0	Configured cycle time of the heartbeat (ms)

Index	Sub index	Name	Type	Access	Default value	Comment
1018h	0	Identity object	Unsigned8	RO	4	Highest sub-index supported
	1		Unsigned32	RO	00000093h	Vendor-ID
	2		Unsigned32	RO	53434D4Bh(*) or 73434D4Bh(*)	Product code
	3		Unsigned32	RO	-	Revision number
	4		Unsigned32	RO	-	Serial number
1200h	0	SDO1 server parameter	Unsigned8	RO	2	Highest sub-index supported
	1		Unsigned32	RO	00000600h + Node-ID	COB-ID client --> server (rx)
	2		Unsigned32	RO	00000580h + Node-ID	COB-ID server --> client (tx)
1301h	0	SRDO1 commu-nication para-meter	Unsigned8	RO	6	Highest sub-index supported
	1		Unsigned8	RW(**)	1(*) or 0(*)	Information direction 0: disabled 1 enabled for transmission
	2		Unsigned16	RW(**)	25	Refresh-time SRDO1 transmission time in milliseconds
	3		Unsigned8	RW(**)	20	SRVT Unused by SRDO producer
	4		Unsigned8	RO	254	Transmission type
	5		Unsigned32	RW(**)	000000FFh + (2 • Node-ID)	COB-ID 1 Used by SRDO1 normal
	6		Unsigned32	RW(**)	000000FFh + (2 • Node-ID)	COB-ID 2 Used by SRDO1 inverted
1302h	0	SRDO2 commu-nication para-meter	Unsigned8	RO	6	Highest sub-index supported
	1		Unsigned8	RW(**)	1(*) or 0(*)	Information direction 0: disabled 1 enabled for transmission
	2		Unsigned16	RW(**)	25	Refresh-time SRDO2 transmission time in milliseconds
	3		Unsigned8	RW(**)	20	SRVT Unused by SRDO producer
	4		Unsigned8	RO	254	Transmission type
	5		Unsigned32	RW(**)	000000FFh + (2 • Node-ID)	COB-ID 1 Used by SRDO2 normal
	6		Unsigned32	RW(**)	00000100h + (2 • Node-ID)	COB-ID 2 Used by SRDO2 inverted
1381h	0	SRDO1 mapping parameter	Unsigned8	RO	4	Highest sub-index supported
	1		Unsigned8	RO	51300120h	Safety-related application data object 1 (plain data)
	2		Unsigned16	RO	51300220h	Safety-related application data object 1 (bitwise inverted data)
	3		Unsigned8	RO	51500108h	Safety-related application data object 2 (plain data)
	4		Unsigned8	RO	51500208h	Safety-related application data object 2 (bitwise inverted data)

Index	Sub index	Name	Type	Access	Default value	Comment
1382h	0	SRDO2 mapping parameter	Unsigned8	RO	4	Highest sub-index supported
	1		Unsigned8	RO	51300120h	Safety-related application data object 1 (plain data)
	2		Unsigned16	RO	51300220h	Safety-related application data object 1 (bitwise inverted data)
	3		Unsigned8	RO	51500108h	Safety-related application data object 2 (plain data)
	4		Unsigned8	RO	51500208h	Safety-related application data object 2 (bitwise inverted data)
13FEh	0	Configuration valid	Unsigned8	RW(**)	00h	A5h: configuration is valid Other values: configuration is not valid
13FFh	0	Safety configuration signature	Unsigned8	RO	2	Highest sub-index supported
	1		Unsigned16	RW(**)	0000h	SRDO1 signature
	2		Unsigned16	RW(**)	0000h	SRDO2 signature
1800h	0	TPDO1 communication parameter	Unsigned8	RO	5	Highest sub-index supported
	1		Unsigned32	RW	80000180h + Node-ID	COB-ID del TPDO1
	2		Unsigned8	RW	254	Transmission type
	5		Unsigned16	RW	10	Event-timer
1A00h	0	TPDO1 mapping parameter	Unsigned8	RW	2	Number of mapped application objects in TPDO1
	1		Unsigned32	RW	91300120h(*) or 61300120h(*)	1st application object
	2		Unsigned32	RW	61500108h	2nd application object
	3		Unsigned32	RW	20910010h	3rd application object

(*) Parameter values depends on the factory configuration of the device (ordering code of the sensor).
(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

Manufacturer Profile Area

Index	Sub index	Name	Type	Access	Default value	Comment
2010h	0	Minimum nominal pressure	Integer16	RO	-	Minimum nominal pressure value
2011h	0	Maximum nominal pressure	Integer16	RO	-	Maximum nominal pressure value
2020h	0	Minimum value storage	Real32	RO	-	Minimum measured pressure value (volatile)
2021h	0	Maximum value storage	Real32	RO	-	Maximum measured pressure value (volatile)
2090h	0	Process value as integer	Integer32	RO	-	AI input PV as 32 bit integer data format. Identical to 9130h
2091h	0	Temperature	Integer16	RO	-	Actual working temperature of the electronic given in 0.5 °C
2100h	0	User device name	Unsigned32	RW	FFFFFFFFh	User defined name for the device
2201h	0	Last calibration date year	Unsigned8	RW	-	Year of the last calibration (last two digits)
2202h	0	Last calibration date month	Unsigned8	RW	-	Month of the last calibration
2203h	0	Last calibration date day	Unsigned8	RW	-	Day of the last calibration
2207h	0	Date of production year	Unsigned8	RO	-	Year of production (last two digits)
2208h	0	Date of production month	Unsigned8	RO	-	Month of production

Index	Sub index	Name	Type	Access	Default value	Comment
2209h	0	Date of production day	Unsigned8	RO	-	Day of production
2320h	0	Persistent Node-ID	Unsigned8	RW	01h(*)	Node-ID stored in non-volatile memory of the device
5030h	0	SAFETY AI_Input_PV (float)	Unsigned8	RO	2	Highest sub-index supported
	1		Real32	RO	-	SAFETY AI Input PV 1 float (normal)
	2		Real32	RO	-	SAFETY AI Input PV 1 float (inverted)
5130h	0	SAFETY AI_Input_PV (integer32)	Unsigned8	RO	2	Highest sub-index supported
	1		Integer32	RO	-	SAFETY AI Input PV 1 integer32 (normal)
	2		Integer32	RO	-	SAFETY AI Input PV 1 integer32 (inverted)
5150h	0	SAFETY AI Status	Unsigned8	RO	2	Highest sub-index supported
	1		Unsigned8	RO	-	SAFETY AI Status 1 (normal)
	2		Unsigned8	RO	-	SAFETY AI Status 1 (inverted)
51FCh	0	SAFETY Application configuration check Password	Unsigned32	RW(**)	00000000h	Writing the password “sfty” (79746673h) in this object allows the modification of the object 51FDh.
51FDh	0	SAFETY Application configuration check enable	Unsigned8	RW(***)	01h	The safety-related application configuration validity: 01h: is checked 00h: is not checked
51FEh	0	SAFETY Application Configuration Valid	Unsigned8	RW(**)	00h	A5h: configuration is valid Other values: configuration is not valid
51FFh	0	SAFETY Application Configuration Signature	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned16	RW(**)	0000h	CRC signature for the configuration data of the safety-related application parameters

(*) Parameter values depends on the factory configuration of the device (ordering code of the sensor).

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

(***) RW if NMT state is Pre-operational and password written in object 51FCh, RO if NMT state is Operational

Device Profile Area

Index	Sub index	Name	Type	Access	Default value	Comment
6110h	0	AI sensor type	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned16	RO	90	AI sensor type 1
6114h	0	AI ADC sample rate	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned32	RW(**)	1000	AI ADC sample rate 1
6121h	0	AI input scaling 1 PV (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RW(**)	-	AI input scaling 1 PV 1 (float)
6123h	0	AI input scaling 2 PV (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RW(**)	-	AI input scaling 2 PV 1 (float)
6124h	0	AI input offset (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RW(**)	-	AI input offset 1 (float)
6125h	0	AI autozero	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned32	WO(***)	-	AI autozero 1
6130h	0	AI input PV (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RO	-	AI input PV 1 (float)
6131h	0	AI physical unit PV	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned32	RW(**)	004E0000h(*) or 00AB0000h(*)	AI physical unit PV 1
6132h	0	AI decimal digits PV	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned8	RW(**)	2(*) or 1(*)	AI decimal digits PV 1
6148h	0	AI span start (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RW	-	AI span start 1 (float)
6149h	0	AI span end (float)	Unsigned8	RO	1	Highest sub-index supported
	1		Real32	RW	-	AI span end 1 (float)
6150h	0	AI status	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned8	RO	-	AI status 1
61A0h	0	AI filter type	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned8	RW(**)	0	AI filter type 1
61A1h	0	AI filter constant	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned8	RW(**)	1	AI filter constant 1
7100h	0	AI input FV (integer16)	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned16	RO	-	AI input FV 1(integer16)
7120h	0	AI input scaling 1 FV (integer16)	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned16	RO	-	AI input scaling 1 FV 1 (integer16)
7122h	0	AI input scaling 2 FV (integer16)	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned16	RO	-	AI input scaling 2 FV 1 (integer16)
9121h	0	AI input scaling 1 PV (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW(**)	-	AI input scaling 1 PV 1 (integer32)
9123h	0	AI input scaling 2 PV (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW(**)	-	AI input scaling 2 PV 1 (integer32)
9124h	0	AI input offset (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW(**)	-	AI input offset 1 (integer32)
9130h	0	AI input PV (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RO	-	AI input PV 1 (integer32)
9148h	0	AI span start (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW	-	AI span start 1 (integer32)
9149h	0	AI span end (integer32)	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW	-	AI span end 1 (integer32)

(*) Parameter values depends on the factory configuration of the device (ordering code of the sensor).

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

(***) WO if NMT state is Pre-operational

4.2. SDO OBJECTS

1000h – Device type

This object describes the type of the device and its functionality. It is composed of a 16-bit field that describes the device profile or the application profile that is used and a second 16-bit field, which gives additional information about optional functionality of the device.

The structure of the device parameter is represented in the following figure.

31 16	**15 0**
Additional Information	Device Profile Number

Figure 30. Structure of the Device type parameter

Additional information = 8002h
Device Profile Number = 0194h

Object description

Index	Name
1000h	Device type

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Device type	RO	Unsigned32	80020194h	80020194h

1001h – Error register

This object provides error information. The CANopen device maps internal errors into this object. It is a part of an emergency object.

For the GEFRAN KMC Safety device the Generic error indication is given when one or more safety-related errors are present. The error register contains one of the error codes described in the following table.

Error code	Description
00h	No error
81h	Generic error (manufacturer specific)

Table 6. Error codes in the Error register

Object description

Index	Name
1001h	Error register

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Error register	RO	Unsigned8	00h, 81h	-

1002h – Manufacturer status register

This object provides a common status register for manufacturer-specific purposes. The device maps safety-related errors into this object.

When a safety-related error is detected, a defined bit inside the Manufacturer status register is set.

An error is set if the error mask applied to the Manufacturer status register is equal to the error mask itself.

The safety-related errors detected and handled by the device are collected in the following table.

Error mask	Bit position	Safety-related error description
00000001h	0	RAM error
00000002h	1	FLASH error
00000004h	2	CPU error
00000008h	3	SW calculation error
00000010h	4	Program temporal sequence monitoring check error
00000020h	5	SW watchdog control error
00000040h	6	HW watchdog control error
00000080h	7	Manufacturer calibration data error
00000100h	8	Internal 5V voltage error
00000200h	9	Internal 3.3V voltage error
00000400h	10	Sensor input stage error
00000800h	11	Sensor over range error
00001000h	12	Sensor under range error
00002000h	13	Sensor broken error
00004000h	14	Temperature over range error
00008000h	15	Power supply out of range error
00010000h	16	SRDO internal timer error
00020000h	17	Flash write error
00040000h	18	SRDO configuration not valid error
00080000h	19	Safety-related application parameters configuration not valid error
00100000h	20	Internal DMA transfer error

Table 7. Safety-related errors in Manufacturer status register

Object description

Index	Name
1002h	Manufacturer status register

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Manufacturer status register	RO	Unsigned32	-	-

1003h – Pre-defined error field

This object provides the error that occurred in the device and were signaled via the emergency object. In doing so it provides an error history.

The object entry at sub-index 00h contains the number of actual errors that are recorded in the array starting at sub-index 01h.

The most recent error is always stored at sub-index 01h.

Writing 00h to sub-index 00h deletes the entire error history

A read access to sub-index N with N > "Number of errors" is responded with an SDO abort message.

The pre-defined error field is composed of a 16-bit error code (lower 2 bytes) and a 16-bit additional error information (upper 2 bytes) which is fixed at 0.

31 … 16	**15 … 0**
Additional Information (0000h)	Error code

Figure 31. Structure of the pre-defined error field

About the error codes see the description of EMCY services.

Object description

Index	Name
1003h	Pre-defined error field

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Number of errors	RW	Unsigned8	0..32	-
1	Standard error field 1	RO	Unsigned32	-	-
..	..	..	..	..	..
32	Standard error field 32	RO	Unsigned32	-	-

1005h – COB-ID SYNC

This object indicates the configured COB-ID of the synchronization object (SYNC). It also defines whether the CANopen device generates the SYNC.

The structure of this object is specified in the following figure.

31	**30**	**29**	**28 … 11**	**10 … 0**
x	gen.	frame	Reserved (0 0000h)	11-bit CAN-ID

Figure 32. Structure of SYNC COB-ID

The value definition is given in the following table.

Field name	Value	Description
x	0	Do not care
gen	0	Device does not generate SYNC message
frame	0	11-bit CAN-ID valid (CAN base frame)
11 bit CAN-ID	80h	11-bit CAN-ID of the CAN base frame

Table 8. COB-ID SYNC message field

Object description

Index	Name
1005h	COB-ID SYNC

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	COB-ID SYNC	RW	Unsigned32	Unsigned32 (*)	00000080h

(*) The 11-bit CAN-ID of the COB-ID must be compliant to the restricted CAN-ID definitions (see Restricted CAN-ID section). A restricted CAN-ID cannot be used.

1008h – Manufacturer device name

This object provides the name of the device as given by the manufacturer.

Object description

Index	Name
1008h	Manufacturer device name

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Manufacturer device name	RO	Visible_string	KMCS	KMCS

1009h – Manufacturer hardware version

This object provides the manufacturer hardware version description.

Object description

Index	Name
1009h	Manufacturer hardware version

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Manufacturer hardware version	RO	Visible_string	Visible_string	-

100Ah – Manufacturer software version

This object provides the manufacturer software version description.

Object description

Index	Name
100Ah	Manufacturer software version

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Manufacturer software version	RO	Visible_string	Visible_string	-

100Ch – Guard time

This object indicates the configured guard time. The guard time multiplied with the lifetime factor gives the lifetime for the life guarding protocol.

The value of 0 disables the lifeguarding, all other values up to 65535 are valid.

Object description

Index	Name
100Ch	Guard time

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Guard time	RW	Unsigned16	0..65535	0

100Dh – Life time factor

The lifetime factor multiplied with the guard time gives the lifetime for the life guarding protocol. The value of 0 disables the life guarding, all other values, up to 255, are valid.

Object description

Index	Name
100Dh	Life time factor

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Life time factor	RW	Unsigned8	0..255	0

1010h – Store parameters

This object controls the saving of parameters in non-volatile memory.
To avoid storage of parameters by mistake, storage is only executed when the signature "save" is written to:

- sub-index 1: all parameters are saved in non-volatile memory
- sub-index 2: only communication parameters are saved in non-volatile memory
- sub-index 3: only application parameters are saved in non-volatile memory

The storage write access structure is specified in the following figure

31			**0**
e (65h)	v (76h)	a (61h)	s (73h)
MSB			LSB

Figure 33. Storage write access structure

By read access the sub-indexes 1, 2 or 3 of this object, the device provides information about its saving capabilities. Giving the value of 1, it means that the device saves parameters on command.

Object description

Index	Name
1010h	Store parameters

Entry description

<table>
<tr><th>SUB Index</th><th>Name</th><th>Access</th><th>Data Type</th><th>Value Range</th><th>Default</th></tr>
<tr><td>0</td><td>Highest sub-index supported</td><td>RO</td><td>Unsigned8</td><td>3</td><td>3</td></tr>
<tr><td>1</td><td>Save all parameters</td><td>RW(**)</td><td>Unsigned32</td><td rowspan="3">Read access: 00000001h
Write access: 65766173h (ASCII: “save”)</td><td rowspan="3">Read access: 00000001h
Write access: 65766173h (ASCII: “save”)</td></tr>
<tr><td>2</td><td>Save communication parameters</td><td>RW(**)</td><td>Unsigned32</td></tr>
<tr><td>3</td><td>Save application parameters</td><td>RW(**)</td><td>Unsigned32</td></tr>
</table>

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

NOTE

The Store Parameters command should not be used if the LSS Configure Node-ID command has been previously used.
If the Node-ID must be changed, this is the correct operating sequence:
1) configure the Node-ID with the LSS Configure Node-ID command
2) save the Node-ID value with the LSS Store Configuration command
3) reset the device with the NMT Reset Application command, or execute a power cycle

SAFETY NOTE

The store parameters command is not allowed when the device is in Operational state

1011h – Restore default parameters

This object controls the restore of parameters in non-volatile memory to their default values, according to the communication and device profile.

In order to avoid restoring of parameters by mistake, restoring is only executed when the signature “load” is written to:
- sub-index 1: all parameters are restored in non-volatile memory
- sub-index 2: only communication parameters are restored in non-volatile memory
- sub-index 3: only application parameters are restored in non-volatile memory

The restore default parameters write access structure is specified in the following figure.

<table>
<tr><th colspan="2" align="left">31</th><th colspan="2" align="right">0</th></tr>
<tr><td>d (64h)</td><td>a (61h)</td><td>o (6Fh)</td><td>l (6Ch)</td></tr>
<tr><td colspan="2" align="left">MSB</td><td colspan="2" align="right">LSB</td></tr>
</table>

Figure 34. Restore write access structure

By read access the sub-index 1, 2 or 3 of this object, the device provides information about its restoring capabilities. Giving the value of 1, it means that the device can restore parameters on command.

The default values are set valid after the device is power cycled.

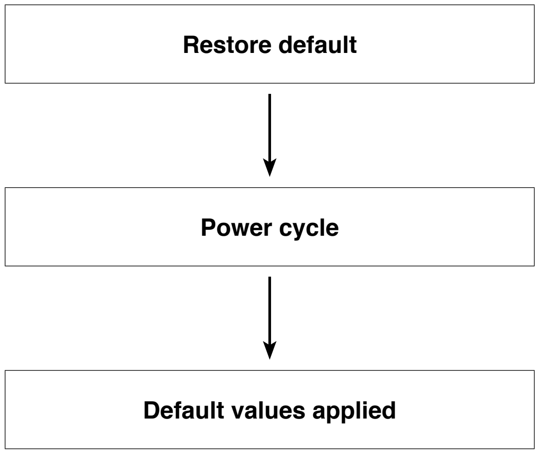


Figure 35. Restore procedure

For the GEFRAN KMC Safety device, the Restore default parameters command does not apply to these objects:

- COB-ID EMCY (1014h)
- COB-ID of TPDO1 (1800h, sub-index 1)
- COB-IDs of 1st SDO (1200h, sub-index 1 and 2)
- COB-IDs of SRDO1 (1301h, sub-index 5 and 6)
- COB-IDs of SRDO2 (1302h, sub-index 5 and 6)

The value of the above listed objects is modified only after a change of the Node-ID value.

Object description

Index	Name
1011h	Restore default parameters

Entry description

<table>
<tr><th>SUB Index</th><th>Name</th><th>Access</th><th>Data Type</th><th>Value Range</th><th>Default</th></tr>
<tr><td>0</td><td>Highest sub-index supported</td><td>RO</td><td>Unsigned8</td><td>3</td><td>3</td></tr>
<tr><td>1</td><td>Restore all default parameters</td><td>RW(**)</td><td>Unsigned32</td><td rowspan="3">Read access: 00000001h
Write access: 64616F6Ch (ASCII: “load”)</td><td rowspan="3">Read access: 00000001h
Write access: 64616F6Ch (ASCII: “load”)</td></tr>
<tr><td>2</td><td>Restore communication parameters</td><td>RW(**)</td><td>Unsigned32</td></tr>
<tr><td>3</td><td>Restore application parameters</td><td>RW(**)</td><td>Unsigned32</td></tr>
</table>

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE
The restore default parameters command invalidates the Safe communication and the Safety-related application parameter configurations. The restore default parameters command is not allowed when the device is in Operational state

1014h – COB-ID EMCY

This object indicates the configured COB-ID of the EMCY write service.
The structure of this object is specified in the following figure.

31	30	29	28 11	10 0
valid	res.	frame	Reserved (0 0000h)	11-bit CAN-ID

Figure 36. Structure of EMCY COB-ID

The value definition is given in the following table.

Field name	Value	Description
valid	0	EMCY exists / is valid
reserved	0	Reserved (always 0)
frame	0	11-bit CAN-ID valid (CAN base frame)
11 bit CAN-ID	80h + Node-ID (default) or user defined	11-bit CAN-ID of the CAN base frame

Table 9. COB-ID EMCY message fields

Object description

Index	Name
1014h	COB-ID EMCY

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	COB-ID EMCY	RO	Unsigned32	00000080h + Node-ID	00000080h + Node-ID

1015h – Inhibit time EMCY

This object indicates the configured inhibit time of the EMCY write service.
The inhibit time EMCY defines the minimum time that elapses between two consecutive invocations of the EMCY service.
The value is given in multiples of 100 us. The accepted values must be multiples of 10, i.e. 1 ms. The value 0 disables the inhibit time.

Object description

Index	Name
1015h	Inhibit time EMCY

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Inhibit time EMCY	RW	Unsigned16	0..65535 as multiples of 10	0

NOTE
When using low baudrate values, setting the Inhibit time EMCY to the proper value can avoid possible bus overloads due to the high frequency rate of transmission of the EMCY messages under certain circumstances.

1017h – Producer heartbeat time

The producer heartbeat time indicates the configured cycle time of the heartbeat, given in 1 ms. The value 0 disables the producer heartbeat.

Object description

Index	Name
1017h	Producer heartbeat time

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Producer heartbeat time	RW	Unsigned16	0..65535	0

1018h – Identity object

This object provides general identification information of the device.

- Sub-index 1: contains the unique value that is allocated uniquely to each vendor of a CANopen device. For GEFRAN s.p.a. manufacturer it is 00000093h.
- Sub-index 2: contains the unique value that identifies a specific type of CANopen device. For the GEFRAN KMC Safety device it is 53434D4Bh for the Integer32 data version or 73434D4Bh for the float data version.
- Sub-index 3: contains the major revision number and the minor revision number of the revision of the device. Its value is device specific.
- Sub-index 4: contains the serial number that identifies uniquely the device. Its value is device specific.

Object description

Index	Name
1018h	Identity object

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	4	4
1	Vendor-ID	RO	Unsigned32	00000093h	00000093h
2	Product code	RO	Unsigned32	53434D4Bh(*) or 73434D4Bh(*)	53434D4Bh(*) or 73434D4Bh(*)
3	Revision number	RO	Unsigned32	-	-
4	Serial number	RO	Unsigned32	-	-

(*) Parameters values depends on the factory configuration of the device (ordering code of the sensor).

The user can also get the identity object values using the LSS inquire identity services (see LSS protocol description section).

The Product code, when read as string data type, equals to the “KMCS” or “KMCs” signature (KMC with CANopen Safety Output).

The Revision number can vary depending on HW/FW updates. The Serial Number is unique for each device.

The Serial number is also printed on the label attached to the case of the device.

1200h – SDO1 server parameter

This object describes the first SDO used on the device.

The values at sub-index 1 and sub-index 2 specify the COB-IDs for the first SDO. The object structure is specified in the following figure.

31	30	29	28 11	10 0
valid	dyn	frame	Reserved (0 0000h)	11-bit CAN-ID

Figure 37. Structure of SDO1 COB-ID

The value definition is given in the following table.

Field name	Value	Description
valid	0	SDO exists / is valid
dyn	0	Value is assigned statically
reserved	0	11-bit CAN-ID valid (CAN base frame)
11 bit CAN-ID	00000600h + Node-ID (default rx) or 00000580h + Node-ID (default tx)	11-bit CAN-ID of the CAN base frame

Table 10. SDO1 COB-ID fields

Object description

Index	Name
1200h	SDO1 server parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	COB-ID client --> server (rx)	RO	Unsigned32	Unsigned32 (*)	00000600h + Node-ID
2	COB-ID server --> client (tx)	RO	Unsigned32	Unsigned32 (*)	00000580h + Node-ID

(*) The 11-bit CAN-ID of the COB-ID must be compliant to the restricted CAN-ID definitions (see Restricted CAN-ID section). A restricted CAN-ID cannot be used.

1301h – SRDO1 communication parameter

This object indicates the communication behavior of the SRDO1.
Sub-index 00h indicates the highest supported sub-index (06h).
Sub-index 01h indicates if the SRDO1 is produced or is not valid for transmission (not transmitted by the device).

Value	Description
00h	SRDO1 is not valid for transmission (no SRDO transmission)
01h	SRDO1 is valid for transmission (SRDO producer)

Sub-index 02h indicates the refresh-time for the SRDO1. The refresh-time defines the cyclic transmission rate of the SRDO1.

Sub-index 03h is reserved for SRDO1 producer devices, as in the KMC Safety. For SRDO1 consumer devices it is called SRVT.

Sub-index 04h indicates the transmission type as defined in EN 50325-4.

Sub-index 05h indicates the COB-ID used by the SRDO1 for the plain (normal) data (first CAN data frame of the SRDO1).

Sub-index 06h indicates the COB-ID used by the SRDO1 for the bitwise inverted data (second CAN data frame of the SRDO1).

Object description

Index	Name
1301h	SRDO1 communication parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	6	6
1	Information direction	RW(**)	Unsigned8	00h, 01h	(*)
2	Refresh-time	RW(**)	Unsigned16	Unsigned16	25
3	SRVT	RW(**)	Unsigned8	Unsigned8	20
4	Transmission type	RO	Unsigned8	254	254
5	COB-ID 1	RW(**)	Unsigned32	101h..17Fh (odd values only)	Node-ID ≤ 64: FFh + 2 • Node-ID Node-ID > 64: 17Fh
6	COB-ID 2	RW(**)	Unsigned32	102h..180h (even values only)	Node-ID ≤ 64: 100h + 2 • Node-ID Node-ID > 64: 180h

(*) Parameter value depends on the factory configuration of the device (ordering code of the sensor).
(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

The transmission validity of the SRDO depends on the ordering code of the KMC Safety device.
If the data type selected for the pressure data is INTEGER32:
• the SRDO1 is set valid for transmission by default (Information direction = 01h)
• the SRDO2 is set not valid for transmission by default (Information direction = 00h)
If the data type selected for the pressure data is REAL32:
• the SRDO1 is set not valid for transmission by default (Information direction = 00h)
• the SRDO2 is set valid for transmission by default (Information direction = 01h)
Since default COB-IDs are identical for SRDO1 and SRDO2, SRDO1 and SRDO2 should not be set both as valid for transmission until COB-ID are changed manually.

1302h - SRDO2 communication parameters

This object indicates the communication behavior of the SRDO2.
Sub-index 00h indicates the highest supported sub-index (06h).
Sub-index 01h indicates if the SRDO2 is produced or is not valid for transmission (not transmitted by the device).

Value	Description
00h	SRDO2 is not valid for transmission (no SRDO transmission)
01h	SRDO2 is valid for transmission (SRDO producer)

Sub-index 02h indicates the refresh-time for the SRDO2. The refresh-time defines the cyclic transmission rate of the SRDO.

Sub-index 03h is reserved for SRDO2 producer devices, as in the KMC Safety. For SRDO2 consumer devices it is called SRVT.

Sub-index 04h indicates the transmission type as defined in EN 50325-4.

Sub-index 05h indicates the COB-ID used by the SRDO2 for the plain (normal) data (first CAN data frame of the SRDO2).

Sub-index 06h indicates the COB-ID used by the SRDO2 for the bitwise inverted data (second CAN data frame of the SRDO2).

Object description

Index	Name
1302h	SRDO2 communication parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	6	6
1	Information direction	RW(**)	Unsigned8	00h, 01h	(*)
2	Refresh-time	RW(**)	Unsigned16	Unsigned16	25
3	SRVT	RW(**)	Unsigned8	Unsigned8	20
4	Transmission type	RO	Unsigned8	254	254
5	COB-ID 1	RW(**)	Unsigned32	101h..17Fh (odd values only)	Node-ID ≤ 64: FFh + 2 • Node-ID Node-ID > 64: 17Fh
6	COB-ID 2	RW(**)	Unsigned32	102h..180h (even values only)	Node-ID ≤ 64: 100h + 2 • Node-ID Node-ID > 64: 180h

(*) Parameter value depends on the factory configuration of the device (ordering code of the sensor).
(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

The transmission validity of the SRDO depends on the ordering code of the KMC Safety device.
If the data type selected for the pressure data is REAL32:
• the SRDO2 is set valid for transmission by default (Information direction = 01h)
• the SRDO1 is set not valid for transmission by default (Information direction = 00h)
If the data type selected for the pressure data is INTEGER32:
• the SRDO2 is set not valid for transmission by default (Information direction = 00h)
• the SRDO1 is set valid for transmission by default (Information direction = 01h)
Since default COB-IDs are identical for SRDO1 and SRDO2, SRDO1 and SRDO2 should not be set both as valid for transmission until COB-ID are changed manually.

1381h – SRDO1 mapping parameter

This object indicates the SR application objects that are mapped into the SRDO1.

Sub-index 1 to sub-index 4 contain the information of the mapped SR application objects. The object describes the content of the SRDO by their index, sub-index and length, as specified in the following figure.

31 16	15 8	7 0
Index	Sub-index	Lenght

Figure 38. Structure of SRDO mapping

The fields definition is given in the following table.

Field name	Description
Index	The content of the SRDO described by the index
Sub-Index	The content of the SRDO described by the sub-index
Lenght	The length of the SR application object in bit

Table 11. SRDO mapping fields

Object description

Index	Name
1381h	SRDO1 mapping parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned32	4	4
1	SR application data object 1 (plain data)	RO	Unsigned32	51300120h	51300120h
2	SR application data object 1 (bitwise inverted data)	RO	Unsigned32	51300220h	51300220h
3	SR application data object 2 (plain data)	RO	Unsigned32	51500108h	51500108h
4	SR application data object 2 (bitwise inverted data)	RO	Unsigned32	51500208h	51500208h

SAFETY NOTE

Data mapped in SRDO1 are fixed and cannot be changed. Data mapped are the following:

- SAFETY AI input PV - Integer32 (normal)
- SAFETY AI input PV - Integer32 (inverted)
- SAFETY AI status (normal)
- SAFETY AI status (invertedl)

1382h – SRDO2 mapping parameter

This object indicates the SR application objects that are mapped into the SRDO2.

Sub-index 1 to sub-index 4 contain the information of the mapped SR application objects, as explained for object 1381h.

Object description

Index	Name
1382h	SRDO2 mapping parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	4	4
1	SR application data object 1 (plain data)	RO	Unsigned32	50300120h	50300120h
2	SR application data object 1 (bitwise inverted data)	RO	Unsigned32	50300220h	50300220h
3	SR application data object 2 (plain data)	RO	Unsigned32	51500108h	51500108h
4	SR application data object 2 (bitwise inverted data)	RO	Unsigned32	51500208h	51500208h

SAFETY NOTE

Data mapped in SRDO2 are fixed and cannot be changed. Data mapped are the following:

- SAFETY AI input PV - Float (normal)
- SAFETY AI input PV - Float (inverted)
- SAFETY AI status (normal)
- SAFETY AI status (invertedl)

13FEh – Configuration valid

This object indicates if the current configuration of the safe communication parameters is valid.

By default, the configuration is not valid (the value of the object is 00h) and the device switches into the safe state.

The configuration is set to not valid also after any change of the content of at least one of the SR communication objects (1301h and 1302h), in particular in these cases:

1) after a direct change of at least one value in objects 1301h or 1302h

2) after a change of the Node-ID

3) after the command 1011h (restore all default parameters or restore communication parameters)

The configuration can be set to valid after the safe communication parameters validation procedure is performed.

Value	Description
00h to A4h	Configuration is not valid
A5h	Configuration is valid
A6h to FFh	Configuration is not valid

Object description

Index	Name
13FEh	Confguration valid

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Configuration valid	RW(**)	Unsigned8	00h..FFh	00h

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

13FFh – Safety configuration signature

This object is used to secure and verify the configuration of the SRDO.

Sub-index 1 contains the CRC signature related to the configuration data of SRDO1.

Sub-index 2 contains the CRC signature related to the configuration data of SRDO2.

For the calculation of the CRCs, communication and mapping parameters data of SRDO1 and SRDO2 are involved.

For details see the dedicated paragraph.

Object description

Index	Name
13FFh	Safety configuration signature

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	SRDO1 signature	RW(**)	Unsigned16	0000h..FFFFh	0000h
2	SRDO2 signature	RW(**)	Unsigned16	0000h..FFFFh	0000h

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

1800h – TPDO1 communication parameter

This object contains the communication parameters for the PDOs the CANopen device can transmit.

Sub-index 1 contains the COB-ID of the TPDO1.
The object structure is specified in the following figure.

31	30	29	28 11	10 0
valid	RTR	frame	Reserved (0 0000h)	11-bit CAN-ID

Figure 39. Structure of TPDO1 COB-ID

The value definition is given in the following table.

Field name	Value	Description
valid	0	PDO exists / is valid
	1	PDO does not exists / is not valid
RTR	0	RTR is processed on this PDO
frame	0	11-bit CAN-ID valid (CAN base frame)
11 bit CAN-ID	00000180h + Node-ID (default) or user defined	11-bit CAN-ID of the CAN base frame

Table 12. TPDO1 COB-ID fields

The user can change the default TPDO1 COB-ID value in the range of the allowed values, ensuring that no conflicts with other COB-IDs are generated.

The value is also automatically changed in accordance with the “default scheme” when changing the Node-ID value.

Sub-index 2 defines the transmission type of the TPDO.

Three types of PDO transmission are defined:

1. Synchronous: means that the PDO is transmitted after the SYNC
2. RTR-only: means that the PDO is not transmitted normally it shall be requested via RTR
3. Event-driven: means that the PDO may be transmitted at any time based on the occurrence of a CANopen device internal event

Transmission type settings are explained in the following table.

Value	Description
0	Synchronous (acyclic)
1	Synchronous (cyclic every 1 SYNC)
2	Synchronous (cyclic every 2 SYNC)
3	Synchronous (cyclic every 3 SYNC)
... ...	... ...
240	Synchronous (cyclic every 240 SYNC)
241	RESERVED
... ...	RESERVED
251	RESERVED
252	RTR-only
253	RTR-only
254	Event-driven (asynchronous)
255	Event-driven (asynchronous)

Table 13. TPDO1 transmission type description

Sub-index 5 contains the event-timer. The time is the maximum interval for PDO transmission if the transmission type is set to FEh and FFh.

Its value is given in multiples of 1 ms. The value of 0 disables the event-timer (no PDO is transmitted).

Object description

Index	Name
1800h	TPDO1 communication parameter

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	5	5
1	COB-ID used by TPDO1	RW	Unsigned32	Unsigned32 (*)	80000180h + Node-ID
2	Transmission type	RW	Unsigned8	0..240 and 252..255	254
5	Event-timer	RW	Unsigned16	0..65535	10

(*) The 11-bit CAN-ID of the COB-ID must be compliant to the restricted CAN-ID definitions (see Restricted CAN-ID section). A restricted CAN-ID cannot be used.

1A00h – TPDO1 mapping parameter

This object contains the mapping for the PDOs the device is able to transmit.

Sub-index 1 and sub-index 2 contain the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length, as specified in the following figure.

31 16	15 8	7 0
Index	Sub-index	Length

Figure 40. Structure of TPDO1 mapping

The value definition is given in the following table.

Field name	Description
Index	The content of the PDO described by the index
Sub-index	The content of the PDO described by the sub-index
Length	The length of the application object in bit

Table 14. TPDO1 mapping fields

Object description

Index	Name
1A00h	TPDO1 mapping parameter

Entry description

Sub index	Name	Access	Data Type	Value Range	Default
0	Number of mapped application objects in TPDO1	RO	Unsigned8	0..3	2
1	1st application object (pressure)	RW	Unsigned32	20900020h, 20910010h, 61300120h, 61500108h, 91300120h 50300120h, 50300220h, 51300120h, 51300220h, 51500108h, 51500208h	91300120h* or 61300120h*
2	2nd application object (status)	RW	Unsigned32	20900020h, 20910010h, 61300120h, 61500108h, 91300120h 50300120h, 50300220h, 51300120h, 51300220h, 51500108h, 51500208h	61500108h
3	3rd application object (temperature)	RW	Unsigned32	20900020h, 20910010h, 61300120h, 61500108h, 91300120h 50300120h, 50300220h, 51300120h, 51300220h, 51500108h, 51500208h	20910010h

(*) Parameter value depends on the ordering code of the sensor.

2010h – Minimum nominal pressure

This object indicates the minimum nominal pressure. The value is given depending on the measure unit of pressure selected during the product ordering (bar or psi).

Object description

Index	Name
2010h	Minimum nominal pressure

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Minimum nominal pressure	RO	Integer16	-	-

2011h – Maximum nominal pressure

This object indicates the maximum nominal pressure. The value is given depending on the measure unit of pressure selected during the product ordering (bar or psi).

Object description

Index	Name
2011h	Maximum nominal pressure

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Maximum nominal pressure	RO	Integer16	-	-

2020h – Minimum value storage

This object indicates the minimum value of the AI input PV (object 6130h) registered since the power-on or reset of the device.

The storage is volatile.

Object description

Index	Name
2020h	Minimum value storage

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Minimum value storage	RO	Real32	Real32	-

2021h – Maximum value storage

This object indicates the maximum value of the AI input PV (object 6130h) registered since the power-on or reset of the device.

The storage is volatile.

Object description

Index	Name
2021h	Maximum value storage

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Maximum value storage	RO	Real32	Real32	-

2090h – Process value as integer

This object gives the value of the measured pressure as integer data type format. This object is the same as the object 9130h.

Object description

Index	Name
2090h	Process value as integer

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Process value as integer	RO	Integer32	Integer32	-

2091h – Temperature

This object gives the value of the actual working temperature of the electronic of the device. The value is given in 0.5 °C unit.

Object description

Index	Name
2091h	Temperature

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Temperature	RO	Integer16	Integer16	-

2100h – User device name

This object contains the value of the device name specified by the user.

Object description

Index	Name
2100h	User device name

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	User device name	RW	Unsigned32	Unsigned32	FFFFFFFFh

2201h – Last calibration date year

This object contains the year of the last calibration date.

Object description

Index	Name
2201h	Last calibration date year

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Last calibration date year	RW	Unsigned8	0..99	-

2202h – Last calibration date month

This object contains the month of the last calibration date.

Object description

Index	Name
2202h	Last calibration date month

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Last calibration date month	RW	Unsigned8	1..12	-

2203h – Last calibration date day

This object contains the day of the last calibration date.

Object description

Index	Name
2203h	Last calibration date day

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Last calibration date day	RW	Unsigned8	1..31	-

2207h – Date of production year

This object contains the year of the production date of the device

Object description

Index	Name
2207h	Date of production year

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Date of production year	RO	Unsigned8	0..99	-

2208h – Date of production month

This object contains the month of the production date of the device

Object description

Index	Name
2208h	Date of production month

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Date of production month	RO	Unsigned8	1..12	-

2209h – Date of production day

This object contains the day of the production date of the device

Object description

Index	Name
2209h	Date of production day

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Date of production day	RO	Unsigned8	1..31	-

5030h – SAFETY AI input PV (float)

This object provides the safety data as result of the input scaling block and gives the measured quantity scaled in the used physical unit of the process value (PV) set by AI physical unit PV (see object 6131h).
The data type is floating point number.
Sub-index 01h contains the plain (normal) data.
Sub-index 02h contains the bitwise inverted data.

Object description

Index	Name
5030h	SAFETY AI input PV (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	SAFETY AI input PV 1 float (normal)	RO	Real32	Real32	-
2	SAFETY AI input PV 1 float (inverted)	RO	Real32	Real32	-

5130h – SAFETY AI input PV (Integer32)

This object provides the safety data as result of the input scaling block and gives the measured quantity scaled in the used physical unit of the process value set by AI physical unit PV (object 6131h), considering the actual number of decimal digits (object 6132h).

The data type is Integer32 number.

Sub-index 01h contains the plain (normal) data.

Sub-index 02h contains the bitwise inverted data.

Object description

Index	Name
5130h	SAFETY AI input PV (Integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	SAFETY AI input PV 1 integer32 (normal)	RO	Unsigned32	Integer32	-
2	SAFETY AI input PV 1 integer32 (inverted)	RO	Unsigned32	Integer32	-

5150h – SAFETY AI Status

This object provides the safety status of the analog input channel as defined for the object 6150h.

Sub-index 01h contains the plain (normal) data.

Sub-index 02h contains the bitwise inverted data.

Object description

Index	Name
5150h	SAFETY AI status

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	SAFETY AI status 1 (normal)	RO	Unsigned8	Unsigned8	-
2	SAFETY AI status 1 (inverted)	RO	Unsigned8	Unsigned8	-

51FCh – SAFETY application configuration check password

This object provides a write protection mechanism for the object 51FDh.

Writing the password “sfty” (79746673h) in this object allows the modification of the object 51FDh.

Object description

Index	Name
51FCh	SAFETY application configuration check password

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	SAFETY application configuration check password	RW(**)	Unsigned32	Write access: 79746673h (ASCII: "sfty")	00000000h

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

51FDh – SAFETY application configuration check enable

This object provides the enabling of the check of the safety-related application configuration validity.
The write access to this object is allowed only after the correct password is written in object 51FCh.

Value	Description
00h	Safety-related application configuration check disabled
01h	Safety-related application configuration check enabled

If the value is set to 00h, the safety-related application configuration validity is not checked.
If the value is set to 01h, the safety-related application configuration validity is checked. In this case, if the configuration is not valid, the device goes in safe state (Pre-operational). It is required to validate the safety-related application configuration to be able to send the device in Operational state.

Object description

Index	Name
51FDh	SAFETY application configuration check enable

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	SAFETY application configuration check enable	RW(***)	Unsigned8	00h..01h	00h

(***) RW if NMT state is Pre-operational and password written in object 51FCh, RO if NMT state is Operational

51FEh – SAFETY application configuration valid

This object indicates if the current configuration of the safety-related application parameters is valid.
By default, the configuration is not valid (the value of the object is 00h) and the device switches into the safe state.
The configuration is set to not valid also after any change of the value of at least one of the safety-related application parameters, listed below:

1) AI ADC sample rate (6114h, 1)
2) AI input scaling 1 PV float (6121h, 1)
3) AI input scaling 2 PV float (6123h, 1)
4) AI input offset float (6124h, 1)
5) AI physical unit PV (6131h, 1)
6) AI decimal digits PV (6132h, 1)
7) AI filter type (61A0h, 1)
8) AI filter constant (61A1h, 1)
9) AI input scaling 1 PV Integer32 (9121h, 1)
10) AI input scaling 2 PV Integer32 (9123h, 1)
11) AI input offset Integer32 (9124h, 1)

The configuration is set to not valid also after the command 1011h (restore all default parameters or restore application parameters)

The configuration can be set to valid after the safety-related application parameters validation procedure is performed.

Value	Description
00h to A4h	Configuration is not valid
A5h	Configuration is valid
A6h to FFh	Configuration is not valid

Object description

Index	Name
51FEh	SAFETY application configuration valid

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	SAFETY application configuration valid	RW(**)	Unsigned8	00h..FFh	00h

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

51FFh – SAFETY application configuration signature

This object is used to secure and verify the configuration of the safety-related application parameters.
Sub-index 1 contains the CRC signature related to the configuration data of the safety-related application parameters.
For details see the dedicated paragraph.

Object description

Index	Name
51FFh	SAFETY application configuration signature

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	Application signa-ture	RW(**)	Unsigned16	0000h..FFFFh	0000h

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

6110h – AI Sensor Type

This object indicates the configured type of sensor, which is connected to the analog input. The value read indicates a pressure sensor.

Object description

Index	Name
6110h	AI sensor type

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI sensor type 1	RO	Unsigned16	90	90

6114h – AI ADC sample rate

This object indicates the configured conversion rate used by the A/D converter. The value is given in multiples of microseconds. Only values multiple of 1000 microseconds are valid.

Object description

Index	Name
6114h	AI ADC sample rate

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI ADC sample rate 1	RW(**)	Unsigned32	1000..10000000 as multiples of 1000	1000

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.

If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

6121h – AI input scaling 1 PV (float)

This object indicates the configured PV of the first calibration point for the analog input channel. It is scaled in physical unit of PV (see object 6131h). The data type is floating point number.

For more details about this object usage see also the section “DS 404 specific functionalities”.

Object description

Index	Name
6121h	AI input scaling 1 PV (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 1 PV 1 (float)	RW(**)	Real32	Real32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.

If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

6123h – AI input scaling 2 PV (float)

This object indicates the configured PV of the second calibration point for the analog input channel. It is scaled in physical unit of PV (see object 6131h). The data type is floating point number.

For more details about this object usage see also the section "DS 404 specific functionalities"..

Object description

Index	Name
6123h	AI input scaling 2 PV (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 2 PV 1 (float)	RW(**)	Real32	Real32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.

If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

6124h – AI input offset (float)

This object indicates the configured additional offset value for the analog input channel. It is scaled in physical unit of PV (see object 6131h). The data type is floating point number.

For more details about this object usage see also the section "DS 404 specific functionalities".

Object description

Index	Name
6124h	AI input offset (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input offset 1 (float)	RW(**)	Real32	Real32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.

If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

6125h – AI autozero

Writing a signature value of “zero“ to this object causes a modification of the AI input offset (objects 6124h and 9124h) in such a way that the actual AI input PV becomes zero.

The autozero write access structure is specified in the following figure.

31			**0**
o (6Fh)	r (72h)	e (65h)	z (7Ah)
MSB			LSB

Figure 41. AI autozero write access structure

For more details about this object usage see also the section “DS 404 specific functionalities”.

Object description

Index	Name
6125h	AI autozero

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI autozero 1	WO	Unsigned32	Write access: 6F72657Ah (ASCII: “zero”)	-

SAFETY NOTE

The autozero command impacts on the safety-related application parameter configuration.

If the command is executed, the safety-related application parameter configuration can be set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The autozero command is not allowed when the device is in Operational state.

6130h – AI input PV (float)

This object provides the result of the input scaling block and gives the measured quantity scaled in the used physical unit of the process value (PV) set by AI physical unit PV (see object 6131h).

The data type is floating point number.

Object description

Index	Name
6130h	AI input PV (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input PV 1 (float)	RO	Real32	Real32	-

6131h – AI physical unit PV

This object indicates the configured SI units and prefixes for the process value within the analog input FB.
The physical units supported by the GEFRAN KMC Safety device are listed in the following table.

Value	Physical unit
004E0000h	bar
00AB0000h	psi
06220000h	Megapascal

Table 15. Physical units supported for the process value

Object description

Index	Name
6131h	AI physical unit PV

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI physical unit PV 1	RW(**)	Unsigned32	(see table)	004E0000h(*) or 00AB0000h(*)

(*) Parameter value depends on the ordering code of the sensor
(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE
This object belongs to the check of the safety-related application parameter configuration. If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again. The change of the value is not allowed when the device is in Operational state.

6132h – AI decimal digits PV

This object indicates the configured number of decimal digits following the decimal point for interpretation of data types INTEGER8, INTEGER16 and INTEGER32.
The objects whose value is affected by the AI decimal digits PV are the following:

2090h: Process value as integer
5130h: SAFETY AI input PV (integer32)
9121h: AI input scaling 1 PV (integer32)
9123h: AI input scaling 2 PV (integer32)
9124h: AI input offset (integer32)
9130h: AI input PV (integer32)
9148h: AI span start (integer32)
9149h: AI span end (integer32)

Example: A FV of 1.23 (REAL32) is coded in INTEGER32 format as:

- 1 if number of decimal digits is set to 0
-12 if number of decimal digits is set to 1
-123 if number of decimal digits is set to 2
-1230 if number of decimal digits is set to 3

In order to avoid overflow conditions, the maximum value of decimal digits that can be set depends on the actual physical unit set for the PV (see object 6131h).

The allowed range of decimal digits values for a specific physical unit, and the default value, is listed in the following table.

Physical unit	Decimal digits range	Decimal digits default value
bar	0..5	2
psi	0..3	1
Megapascal	0..6	3

Table 16. Decimal digits range and default values

Object description

Index	Name
6132h	AI decimal digits PV

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI decimal digits PV 1	RW(**)	Unsigned8	(see table)	2(*) or 1(*)

(*) Parameter value depends on the ordering code of the sensor

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.

If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

6148h – AI span start (float)

This object indicates the configured lower limit of the expected Process Value. When the PV is lower than this limit, it is marked as negative overloaded (see AI status, object 6150h). It is scaled in physical unit of PV (see object 6131h).

The data type is floating point number.

Object description

Index	Name
6148h	AI span start (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI span start 1 (float)	RW	Real32	Real32	-

Note:

The value is set to the minimum nominal pressure by default (see object 2010h).

The user can define a specific value. The value is refused if it is below the 5%FS of the minimum nominal pressure value.

The value can not be higher than the AI span end value (see object 6149h).

6149h – AI span end (float)

This object indicates the configured upper limit of the expected Process Value. When the PV exceeds this limit, it is marked as positive overloaded (see AI status, object 6150h).

It is scaled in physical unit of PV (see object 6131h). The data type is floating point number.

Object description

Index	Name
6149h	AI span end (float)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI span end 1 (float)	RW	Real32	Real32	-

Note:
The value is set to the maximum nominal pressure value by default (see object 2011h).
The user can define a specific value. The value is refused if it is above the 10%FS of the maximum nominal pressure value.
The value can not be lower than the AI span start value (see object 6148h).

6150h – AI status

This object provides the status of the analog input channel as defined in the following figure:

7 … 3	2	1	0
Reserved	Negative overload	Positive overload	Not valid
MSB			LSB

Examples:

Value	Description
00h	Measure is valid, normal working condition
01h	Measure is not valid. Safety-related error detected. The device goes in safe state (pre-operational)
02h	Measure is valid, but the measured value is above AI span end
03h	Measure is not valid because the measured value is over the 10%FS of the maximum nominal pressure value.
04h	Measure is valid, but the measured value is below AI span start
05h	Measure is not valid because the measured value is below the minimum nominal pressure value for more than 5%FS.

> SAFETY NOTE
>
> If the measured value is over the 40%FS or below the minimum nominal pressure value for more than 10%FS, the device goes in safe state (Pre-operational).

Object description

Index	Name
6150h	AI status

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI status 1	RO	Unsigned8	-	-

61A0h – AI filter type

This object indicates the type of filter to be used for calculation.
The filter types used by GEFRAN KMC Safety device are specified in the following table.

Value	Description
0	No filter (measure unfiltered)
1	Moving average
2	Repeating average

Table 17. Filter types

If the selected filter type is not “0”, a proper filter constant value must be specified for the correct filter operation (see object 61A1h).

Object description

Index	Name
61A0h	AI filter type

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI filter type 1	RW(**)	Unsigned8	(see table)	0

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.
If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

61A1h – AI filter constant

This object indicates the configured constant value used for the filter calculation (see object 61A0h).

Object description

Index	Name
61A1h	AI filter constant

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI filter constant 1	RW(**)	Unsigned8	1..64	1

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

The value of the filter constant should be set depending on the type of filter used (see object 61A0h). The same value of the filter constant gives different results with different filter types.

Note:
The calculation result is also influenced by the value of the AI ADC sample rate value (see object 6114h), so the choice of the AI filter constant should be done depending also on the value of that parameter.

SAFETY NOTE
This object belongs to the check of the safety-related application parameter configuration. If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again. The change of the value is not allowed when the device is in Operational state.

7100h – AI input FV

This object provides the converted value of the analog input module, which is not yet scaled to the physical unit of the pressure, called field value (FV).

Object description

Index	Name
7100h	AI input FV

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input FV 1	RO	Unsigned16	Unsigned16	-

7120h – AI input scaling 1 FV

This object indicates the configured FV of the first calibration point for the analog input channel.

Object description

Index	Name
7120h	AI input scaling 1 FV

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 1 FV 1	RO	Unsigned16	Unsigned16	-

7122h – AI input scaling 2 FV

This object indicates the configured FV of the second calibration point for the analog input channel.

Object description

Index	Name
7122h	AI input scaling 2 FV

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 2 FV 1	RO	Unsigned16	Unsigned16	-

9121h – AI input scaling 1 PV (integer32)

This object indicates the configured PV of the first calibration point for the analog input channel. It is scaled in physical unit of PV, considering the actual number of decimal digits (see objects 6131h and 6132h).

The data type is 32 bit signed integer.

For more details about this object usage see also the section “DS 404 specific functionalities”.

Object description

Index	Name
9121h	AI input scaling 1 PV (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 1 PV 1 (integer32)	RW(**)	Integer32	Integer32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE
This object belongs to the check of the safety-related application parameter configuration. If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again. The change of the value is not allowed when the device is in Operational state.

9123h – AI input scaling 2 PV (integer32)

This object indicates the configured PV of the second calibration point for the analog input channel. It is scaled in physical unit of PV, considering the actual number of decimal digits (see objects 6131h and 6132h). The data type is 32 bit signed integer.

For more details about this object usage see also the section “DS 404 specific functionalities”.

Object description

Index	Name
9123h	AI input scaling 2 PV (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input scaling 2 PV 1 (integer32)	RW(**)	Integer32	Integer32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE
This object belongs to the check of the safety-related application parameter configuration. If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Preoperational state; the safety-related application parameter configuration must be validated again. The change of the value is not allowed when the device is in Operational state.

9124h – AI input offset (integer32)

This object indicates the configured additional offset value for the analog input channel. It is scaled in physical unit of PV, considering the actual number of decimal digits (see objects 6131h and 6132h). The data type is 32 bit signed integer.

For more details about this object usage see also the section “DS 404 specific functionalities”.

Object description

Index	Name
9124h	AI input offset (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input offset 1 (integer32)	RW(**)	Integer32	Integer32	-

(**) RW if NMT state is Pre-operational, RO if NMT state is Operational

SAFETY NOTE

This object belongs to the check of the safety-related application parameter configuration.
If the value is changed, the safety-related application parameter configuration is set to invalid and the device goes in Pre-operational state; the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

9130h – AI input PV (integer32)

This object provides the result of the input scaling block and gives the measured quantity scaled in the used physical unit of the process value set by AI physical unit PV (object 6131h), considering the actual number of decimal digits (object 6132h).

The data type is 32 bit signed integer.

Object description

Index	Name
9130h	AI input PV (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI input PV 1 (integer32)	RO	Integer32	Integer32	-

9148h – AI span start (integer32)

This object indicates the configured lower limit of the expected Process Value. When the PV is lower than this limit, it is marked as negative overloaded (see AI status, object 6150h).

It is scaled in physical unit of PV, considering the actual number of decimal digits (see objects 6131h and 6132h).

The data type is 32 bit signed integer.

Object description

Index	Name
9148h	AI span start (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI span start 1 (integer32)	RW	Integer32	Integer32	-

Note:
The value is set at the minimum nominal pressure value by default.
The user can define a specific value. The value is refused if it is below the 5%FS of the minimum nominal pressure value.
The value can not be higher than the AI span end value (see object 9149h).

9149h – AI span end (integer32)

This object indicates the configured upper limit of the expected Process Value. When the PV exceeds this limit, it is marked as positive overloaded (see AI status, object 6150h).

It is scaled in physical unit of PV, considering the actual number of decimal digits (see objects 6131h and 6132h).

The data type is 32 bit signed integer.

Object description

Index	Name
9149h	AI span end (integer32)

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	AI span end 1 (integer32)	RW	Integer32	Integer32	-

Note:
The value is set equal to the maximum nominal pressure value by default.
The user can define a specific value. The value is refused if it is above the 10%FS of the maximum nominal pressure value.
The value can not be higher than the AI span end value (see object 9149h).

5. PDO SERVICES

The real-time data transfer is performed by means of "Process Data Objects (PDO)". Data type and mapping of application objects into a PDO is determined by a corresponding default PDO mapping structure within the object dictionary. For the PDO1 see object 1A00h.

Communication parameters of the PDO, as COB-ID, transmission mode and transmission frequency, are also specified in the object dictionary. For the PDO1 see object 1800h.

Since the GEFRAN KMC Safety device is a PDO producer, its PDO is also called Transmit PDO (TPDO).

SAFETY NOTE

The PDO transmission is not considered Safety.
For Safety applications only use the SRDO (Safety-related Data Object) for the reception and the evaluation of the safety-related data.

5.1. PDO MESSAGE FORMAT

The format of the Transmit PDO message is explained in the following figure.

COB-ID	Rx/Tx	DLC	Data				
			D0	D1	D2	D3	D4
180h + Node-ID	Tx	5	Pressure LSB	Pressure	Pressure	Pressure MSB	Status

Figure 42. Transmit PDO1 (TPDO1) message format

5.2. PDO DATA TYPES

Two types of data are mapped in PDO1 by default: Pressure and Status.
Pressure data can be an INTEGER32 data type or REAL32 data type.
Status data is an UNSIGNED8 data type.
A third type of data can be mapped in PDO1: Temperature (INTEGER16 data type).

Assuming that the data is expressed as a bit sequence of length 32 for INTEGER32 data type and REAL32 data type (b0..b31), and as a bit sequence of length 8 for UNSIGNED8 data type (b0..b7), the transfer syntax is explained in the following figure.

Octet number	1	2	3	4
INTEGER32 REAL32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$
UNSIGNED8	$b_7..b_0$	$b_{15}..b_8$	-	-

Figure 43. Transfer syntax for different data type

Floating point numbers

Data of the basic data types REAL32 have values in the real numbers.
The data type REAL32 is represented as a bit sequence with the length 32.
The IEEE 32 bit implementation of floating point number is represented in the following table.

Bit	b31	b30..b23	b22..b0
Function	S (sign)	E (exponent)	F (mantissa)

The bit sequence b = b0..b31 assigns the following value (finite non-zero number):
$REAL32(b) = (-1)^S \times 2^{E-127} \times (1+F)$
where
$S = b_{31}$, is the sign
$E = b_{30} \times 2^7 + \ldots + b_{23} \times 2^0$, $0 < E < 255$, is the un-biased exponent
$F = 2^{-23} \times (b_{22} \times 2^{22} + \ldots + b_1 \times 2^1 + b_0 \times 2^0)$ is the fractional part of the number (mantissa)

Notes:
E = 0 is used to represent + 0.
E = 255 is used to represent infinities or NaN (not a number).
Example:

Hex: 40C8 0000HEX
Binary: 0100 0000 1100 1000 0000 0000 0000 0000BIN

Calculation of sign, exponent and mantissa:

S = 0
$E = 1000\ 0001BIN = 1 \times 2^7 + 1 \times 2^0 = 129DEC$
$F = 1 \times 2^{-1} + 1 \times 2^{-4} = 0{,}5 + 0{,}0625 = 0{,}5625DEC$

Calculation of the floating point number:

$40C8\ 0000HEX = (-1)^0 \times 2^{129-127} \times (1+0{,}5625) = 6{,}25$

5.3. PDO MAPPING

The GEFRAN KMC Safety device supports a variable PDO mapping. When the device is in NMT state Pre-operational, the following procedure is used for re-mapping:

1. Destroy TPDO1 by setting bit valid of COB-ID used by TPDO1 in TPDO1 communication parameter object (1800h, sub-index 1) to 1b
2. Disable mapping by setting Number of mapped application objects in TPDO1 mapping parameter object (1A00, sub-index 0) to 0
3. Modify mapping by changing the value of
   a. 1st application object in TPDO1 mapping parameter object (1A00, sub-index 1)
   b. 2nd application object in TPDO1 mapping parameter object (1A00, sub-index 2)
   c. 3rd application object in TPDO1 mapping parameter object (1A00, sub-index 3)
   to one of the values listed in the following table.

Mappable object	Value	Number of bytes
2090h: Process value as integer	20900020h	4
6130h: AI input PV (float)	61300120h	4
9130h: AI input PV (integer 32)	91300120h	4
6150h: Ai status	61500108h	1
2091h: Temperature	20910010h	2
5030h,1: Safety AI input PV normal (float)	50300120h	4
5030h,2: Safety AI input PV inverted (float)	50300220h	4
5130h,1: Safety AI input PV normal (integer32)	51300120h	4
5130h,2: Safety AI input PV inverted (integer32)	51300220h	4
5150h,1: Safety AI status normal	51500108h	1
5150h,2: Safety AI status inverted	51500208h	1

Table 18. Mappable objects in TPDO1

4. Enable mapping by setting Number of mapped application objects in TPDO1 mapping parameter object (1A00, sub-index 0) to the desired value (1..3)
5. Create TPDO1 by setting bit valid to 0b of COB-ID used by TPDO1 in TPDO1 communication parameter object (1800h, sub-index 1) to 0b

Note:
the total number of bytes mapped in TPDO1 cannot exceed the value of 8.
Otherwise, a mapping error is given when enabling mapping (step 4).

5.4. PDO TRANSMISSION TYPES

The PDO transmission type for the KMC Safety device can be changed.
There are three types of transmission mode:

1. Synchronous transmission
2. Asynchronous transmission with RTR frames
3. Asynchronous transmission with event-timer

Synchronous Transmission

The transmission of the PDO is performed after the CANopen device receive the n-th SYNC object, when the transmission type is set to n, with n in the range of 1..240.

The SYNC message format is described in the SYNC services description.

Asynchronous Transmission with RTR frames

The transmission of the PDO is performed after the CANopen device receive the PDO remote frame.
The format of the PDO remote frame is explained in the following figure.

COB-ID	**Rx/Tx**	**DLC**	**Data**							
			D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
PDO COB-ID + RTR bit	Rx	0	-	-	-	-	-	-	-	-

Figure 44. RTR message format

Asynchronous Transmission

The transmission of the PDO is performed cyclically after the event-timer has elapsed. The transmission period, expressed in multiples of 1 ms, can be changed through the object 1800h sub-index 5 (PDO event timer) or through the object 6200h (cyclic timer).

6. NMT SERVICES

Through NMT services the NMT master controls the state of the NMT slave devices.
The state attribute is one of these:

- √ Initialization
- √ Pre-operational
- √ Operational
- √ Stopped

6.1. NMT DEVICE STATES

Initialization state

In the NMT state initialization the CANopen device is initialized. The CANopen device parameters are set to their power-on values (last stored parameters in non-volatile memory).

The NMT state initialization own the sub-states Reset application and Reset communication, which are processed automatically one after the other :

1) Reset application: the CANopen device resets all application-related CANopen device parameters and initializes the CANopen Node-ID.
2) Reset communication: the CANopen device reset all communication-related CANopen device parameters and set the CANopen Node-ID.

Pre-operational state

In the pre-operational state the behaviour of the CANopen device at its communication interface can be configured. This can take place by SDO or LSS services. PDO communication is not allowed.

Operational state

In the operational state all communication objects are active. Object Dictionary Access via SDO is possible and the node can handle PDO communication.

Stopped state

In the stopped state the device stops the communication. In this state no communication object is supported, except of Error control services and the reception of NMT commands.

6.2. NMT NODE CONTROL

After power-on, the CANopen device initializes. The initialization state terminates with the transmission of the boot-up message, after which the device enters autonomously the pre-operational state.

In order to change the NMT state of a CANopen device, the NTM master sends the message shown in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
0	Tx	2	CS	Node-ID	-	-	-	-	-	-

Figure 45. NMT message format

The bit fields and their values are explained in the following table.

Bit field	Value range	Description
CS	1	Start. Enter NMT Operational state
	2	Stop. Enter NMT Stopped state
	128	Enter NMT Pre-operational state
	129	Enter NMT Reset application state
	130	Enter NMT Reset communication state
Node-ID	0	All devices must perform the commanded transition
	1 to 127	Only the device that claims the indicated Node-ID must execute the commanded transition

All possible NMT states and state transitions are shown in the following figure.

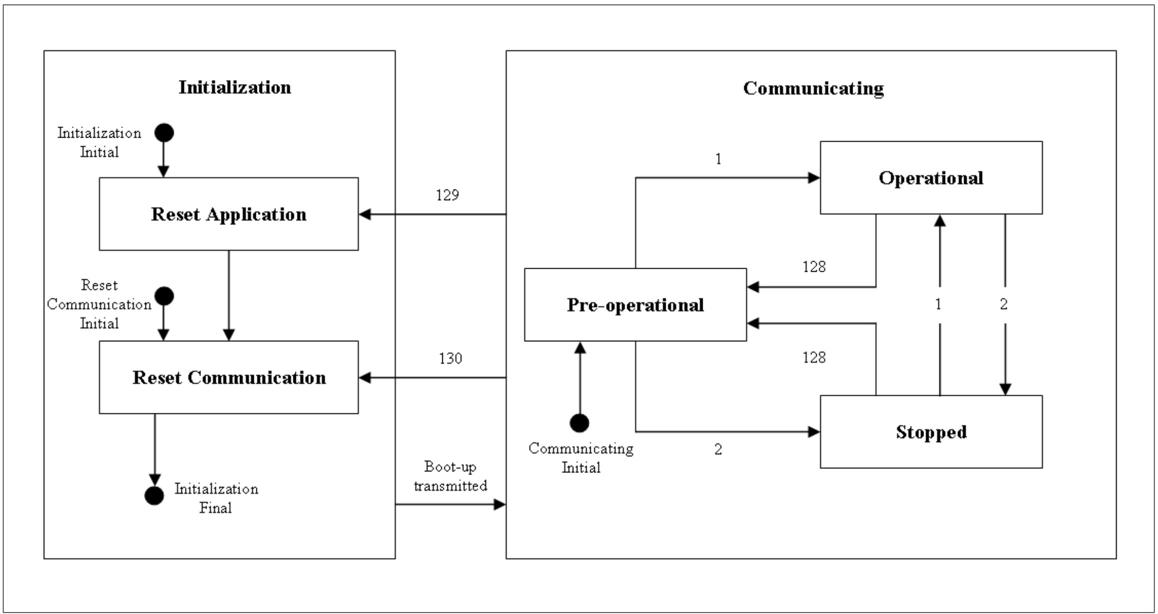


Figure 46. NMT states and state transitions

SAFETY NOTE

If a safety-related error is detected, the device cannot change from the pre-operational state to the operational state.

If the device is in operational state and a safety related error is detected, the device automatically switches to the Pre-operational state (safe state)

6.3. NMT STATES AND COMMUNICATION OBJECTS

Specific services can only be executed if the devices involved in the communication are in the appropriate communication states.

The relationship between communication states and communication objects is shown in the following table.

Object	Reset application	Reset communication	Pre-operational	Operational	Stopped
PDO				X	
SRDO				X	
SDO			X	X	
Boot up		X			
SYNC			X	X	
EMCY			X	X	
NMT error control (Heartbeat and Node guarding)			X	X	X
NMT node control			X	X	

Table 19. NMT states and communication objects

6.4. RESTRICTED CAN-IDS

Restricted CAN-ID can't be used as a CAN-ID by any configurable communication object, neither for SYNC, EMCY, PDO, and SDO. They are listed in the following table.

CAN-ID	used by COB
0 (000h)	NMT
1 (001h) – 127 (07Fh)	reserved
257 (101h) – 384 (180h)	reserved
1409 (581h) – 1535 (5FFh)	default SDO (tx)
1537 (601h) – 1663 (67Fh)	default SDO (rx)
1760 (6E0h) – 1791 (6FFh)	reserved
1793 (701h) – 1919 (77Fh)	NMT error control
1920 (780h) – 2047 (7FFh)	reserved

Table 20. Restricted CAN-IDs

7. BOOT-UP SERVICES

Through this service, the NMT slave indicates that a local state transition occurred from the state Initialization to the state Pre-operational.

The protocol uses the same identifier as the error control protocol. The format of the boot-up message is explained in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
700h + Node-ID	Tx	1	00h	-	-	-	-	-	-	-

Figure 47. Boot-up message format

8. SYNC SERVICES

The SYNC object can be broadcasted periodically by the SYNC producer. This SYNC object provides the basic network synchronization mechanism.

If the CANopen devices operates synchronously (see object 1800h, sub-index 2), it uses the SYNC object to synchronize its own timing, as the PDO transmission, with that of the synchronization object producer.

The format of the SYNC object is explained in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
80h	Rx	0	-	-	-	-	-	-	-	-

Figure 48. SYNC message format

The COB-ID of the SYNC message can be changed by object 1005h (SYNC COB-ID).

9. EMCY SERVICES

Emergency objects are triggered by the occurrence of the device internal error situation. An emergency object is transmitted only once per 'error event'. No further emergency objects are transmitted as long as no new errors occur on the CANopen device. If one or more error conditions change, the device transmits the emergency object with the updated error code. The error register value inside the EMCY object is also updated.

For the GEFRAN KMC Safety device the “Generic error” condition is defined.

The possible EMCY error codes are shown in the following table.

Error code	Description
0000h	Error reset or no error
10XXh	Generic error

Table 21. EMCY error code classes for the KMC Safety device

The generic EMCY error codes are defined in the following table.

Error code	Safety-related error
1000h	RAM error
1001h	FLASH error
1002h	CPU error
1003h	SW calculation error
1004h	Program temporal sequence monitoring check error
1005h	SW watchdog control error
1006h	HW watchdog control error
1007h	Manufacturer calibration data error
1008h	Internal 5V voltage error
1009h	Internal 3.3V voltage error
100Ah	Sensor input stage error
100Bh	Sensor over range error
100Ch	Sensor under range error
100Dh	Sensor broken error
100Eh	Temperature over range error
100Fh	Power supply out of range error
1010h	SRDO internal timer error
1011h	Flash write error
1012h	SRDO configuration not valid error
1013h	Safety-related application parameters configuration not valid error
1014h	Internal DMA transfer error

Table 22. EMCY error codes

About the content of the error register see the description of the object 1001h (Error register).

The format of the EMCY message is explained in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
80h + Node-ID	Tx	8	EMCY error code LSB	EMCY error code MSB	Error register (1001h)	Manufacturer specific error field				

Figure 49. EMCY message format

The COB-ID of the EMCY message can be changed by object 1014h (EMCY COB-ID).

The Manufacturer specific error field, inside the EMCY message, is defined as follows.

D3	D4	D5	D6	D7
Manufacturer status register (1002h) LSB	Manufacturer status register (1002h)	Manufacturer status register (1002h)	Manufacturer status register (1002h) MSB	00h

Figure 50. Manufacturer specific error field

The error control services are used to detect failures within a CAN-based network. Error control services are achieved principally through periodically transmission of messages by a CANopen device.

Two possible mechanism exist to perform the error control: Node guarding and Heartbeat.

The GEFRAN KMC Safety device makes use of both the mechanism.

10.1. NODE GUARDING PROTOCOL

The slave uses the guard time (object 100Ch) and life time factor (object 100Dh) from its object dictionary to calculate the node lifetime, as follows:

node lifetime = guard time x life time factor

If node lifetime is 0, the slave does not handle the guarding mechanism of the NMT master. The guarding is achieved by transmitting guarding requests (node guarding protocol) by the NMT master.

If a NMT slave has not responded within a defined span of time (node life time) or if the NMT slave's communication status has changed, the NMT master informs its NMT master application about that event.

If the NMT slave is not guarded within its lifetime, the NMT slave informs its local application about that event. Guarding starts for the NMT slave when the first RTR for its guarding CAN-ID is received. This may be during the boot-up phase or later.

For the GEFRAN KMC Safety device the node guarding is disabled by default. It can be programmed through objects 100Ch and 100Dh.

10.2. HEARTBEAT PROTOCOL

The heartbeat mechanism is established by cyclically transmitting the heartbeat message. If the heartbeat cycle fails for the heartbeat producer the local application on the heartbeat consumer, aware of this heartbeat message, will be informed about that event.

The format of the heartbeat message is explained in the following figure.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
700h + Node-ID	Tx	1	NMT state	-	-	-	-	-	-	-

Figure 51. Heartbeat message format

The first byte of the heartbeat message data field contains the actual CANopen Network Management State of the CANopen device, as shown in the following table.

Bit field	Value	Description
NMT state	0	Reserved (see boot-up protocol)
	4	Stopped
	5	Operational
	127	Pre-operational

Table 23. NMT state field in heartbeat message

For the GEFRAN KMC Safety device the heartbeat is disabled by default. It can be programmed through object 1017h.

In this section specific functionalities defined in ds404 profile are explained.

11.1. CALIBRATION

The analog input Functional Block (FB) converts Field Values (FVs) into Process Values (PVs).
The unscaled readings from the A/D converter are the Field Values.
The FVs are converted to the physical dimension SI units of the measured quantity.
These converted values are called Process Values. Pressure values in bar, psi, Pa, etc. are Process Values.
The conversion from FVs to PVs is described as a linear transformation. This is defined by two pairs of FVs and corresponding PVs called calibration point 1 and calibration point 2.

Calibration point 1: (Input scaling 1 FV, Input scaling 1 PV)
Calibration point 2: (Input scaling 2 FV, Input scaling 2 PV)

This is illustrated in the following figure.:

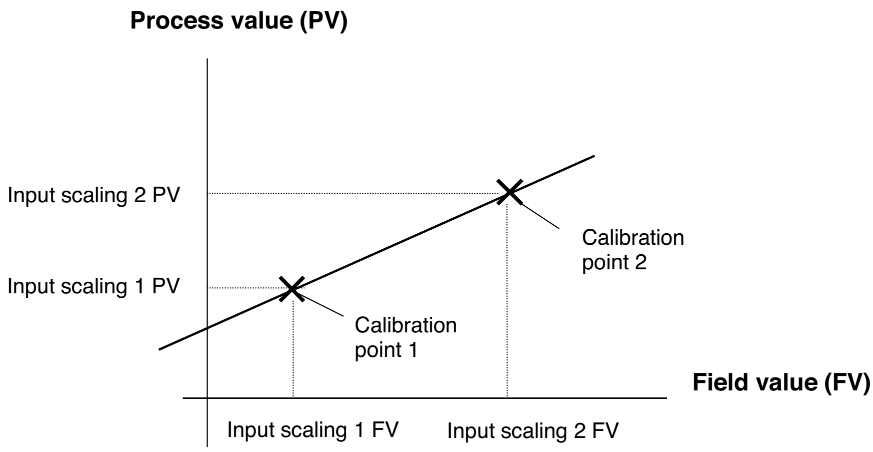


Figure 52. Calibration

The calibration of point 1 is carried out via the objects 6121h (data is float) or 9121h (data is integer32).
The calibration of point 2 is carried out via the objects 6123h (data is float) or 9123h (data is integer32).
The objects 7120h and 7122h are read only.

The GEFRAN KMC transducer is yet calibrated by the manufacturer.
The user can perform his own calibration, if needed. The user-defined calibration can be discarded through a Restore default parameters action (see object 1011h).

The device can be calibrated following the instructions described below.

11.2. PRE-CALIBRATION RECOMMENDATIONS

Before calibrating the device, it is suggested to set to zero the value of the AI input offset (object 6124h or 9124h), so that the user can verify that the pressure value after calibration equals the value set for P1 (object 6121h or 9121h) or P2 (object 6123h or 9123h).

Otherwise the user must remember that the output pressure value is affected by the AI input offset value.

Calibration of point 1

1. The user apply the required pressure value (reference value) of the calibration point 1
2. The user waits until the pressure value is stable at the reference value
3. The user writes the value that the device is supposed to indicate under the pressure currently applied to the object 6121h (float data), or to the object 9121h (32 bit signed integer data)

Calibration of point 2

1. The user apply the required pressure value (reference value) of the calibration point 2
2. The user waits until the pressure value is stable at the reference value
3. The user writes the value that the device is supposed to indicate under the pressure currently applied to the object 6123h (float data), or to the object 9123h (32 bit signed integer data)

Notes:
The value written to the objects 6121h, 6123h, 9121h and 9123h, is expressed with the physical unit currently used (see object 6131h)
The value written in objects 9121h and 9123h has to take in consideration the number of decimal digits currently used (see object 6132h)
The calibration is refused if the calculated characteristic differs too much from the manufacturer calculated characteristic, in particular if the value of the new calculated k coefficient (slope of the characteristic) is 5%FS above the value of the k coefficient set by manufacturer.

Example 1
KMC sensor with nominal pressure range 0..250 bar
The user has set the pressure physical unit in psi unit with a number of decimal digits of 2.
AI physical unit PV (6131h): psi
AI decimal digits (6132h): 2
The user is used to operate with integer values, so has mapped the object 9130h (AI input PV (integer32) in TPDO1.
The nominal pressure range of 0..250 bar corresponds to 0..3625 psi.

Calibration of point 1

The user apply a reference pressure of 0 psi.
When the pressure is stable, a pressure of 0,65 psi is measured by the reference pressure sensor.
Since the number of decimal digits (6132h) is 2, the value that the user writes to object 9121h is 0,65 x 102 = 65 = 00000041h.
The user must send the SDO write command shown in the following image.

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
600h + Node-ID	8	22h	21h	91h	01h	41h	00h	00h	00h

Figure 53. Calibration of point 1 (example 1)

Calibration of point 2

The user apply a reference pressure of 3625 psi.
When the pressure is stable, a pressure of 3624.12 psi is measured by the reference pressure sensor.
Since the number of decimal digits (6132h) is 2, the value that the user writes to object 9123h is 3624.12 x 102 = 362412 = 000587ACh.
The user must send the SDO write command shown in the following image.

COB-ID	**DLC**	**CS**	**index**		**sub-index**	**value**			
		D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
600h + Node-ID	8	22h	23h	91h	01h	ACh	87h	05h	00h

Figure 54. Calibration of point 2 (example 1)

Example 2
KMC sensor with nominal pressure range 0..250 bar
The user has set the pressure physical unit in bar.
AI physical unit PV (6131h): bar
The user is used to operate with floating point numbers, so has mapped the object 6130h (AI input PV (float)) or the object 2090h in TPDO1.

Calibration of point 1

The user apply a reference pressure of 0 bar.
When the pressure is stable, a pressure of 0.3 bar is measured by the reference pressure sensor.
The value 0.3 in floating point data format corresponds to 3ECCCCCDh.
The user must send the SDO write command shown in the following image.

COB-ID	**DLC**	**CS**	**index**		**sub-index**	**value**			
		D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
600h + Node-ID	8	22h	21h	61h	01h	CDh	CCh	CCh	3Eh

Figure 55. Calibration of point 1 (example 2)

Calibration of point 2

The user apply a reference pressure of 250 bar.
When the pressure is stable, a pressure of 250.2 bar is measured by the reference pressure sensor.
The value 250.2 in floating point data format corresponds to 437A3333h.
The user must send the SDO write command shown in the following image.

COB-ID	**DLC**	**CS**	**index**		**sub-index**	**value**			
		D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
600h + Node-ID	8	22h	23h	91h	01h	33h	33h	7Ah	43h

Figure 56. Calibration of point 2 (example 2)

SAFETY NOTE

The calibration adjustment impacts on the safety-related application parameter configuration.

If the value is modified, the safety-related application parameter configuration is set to invalid. The device goes in Pre-operational state and the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

11.3. OFFSET ADJUSTMENT

Using the offset adjustment, the calibration characteristic is shifted by an additional input offset value.
The calibration characteristic is shifted down by positive values of the offset.
The calibration characteristic is shifted up by negative values of the offset.
This is illustrated in the following figure.

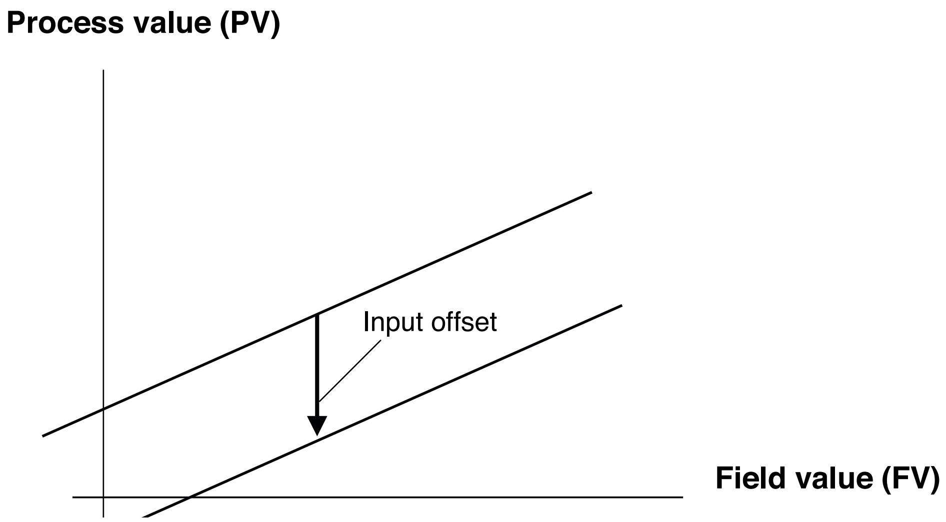


Figure 57. Offset adjustment

The user can use the offset adjustment functionality to get the required exact reading value from the device at a specific pressure level.

Notes:

- The value written to the objects 6124h or 9124h is expressed with the physical unit currently used (see object 6131h)
- The value written to the object 9124h (32 bit signed integer) has to take in consideration the value of the number of decimal digits currently used (see object 6132h)
- The maximum offset value that can be accepted is within a range of ±5%FS.
  If the value set is outside this range, an SDO abort will be given, and the value discarded.

Example
The user has set the pressure physical unit in bar unit with a number of decimal digits of 1.
The user has set the pressure to 100.0 bar, but the device indicates 100.2 bar.
The user wants to get a 100 bar reading from the device.
The offset value to get a 100.0 bar output value from the device is 0.2 bar.

When using floating point numbers, the user has to write the value 0.2 to the object 6124h.
The value 0.2 in floating point data format corresponds to 3E4CCCCDh.
The user must send the following SDO write command:

COB-ID	**DLC**	**CS**	**index**		**sub-index**	**value**			
		D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
600h + Node-ID	8	22h	24h	61h	01h	CDh	CCh	4Ch	3Eh

Figure 58. Offset adjustment request (6124h)

The response message is the following:

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
580h + Node-ID	8	60h	24h	61h	01h	00h	00h	00h	00h

Figure 59. Offset adjustment response (6124h)

When using 32 bit integer numbers, considering the number of decimal digits is 1, the user has to write the value 0.2 x 10 = 2 = 00000002h to the object 9124h.
The user must send the following SDO write command:

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
600h + Node-ID	8	22h	24h	91h	01h	02h	00h	00h	00h

Figure 60. Offset adjustment request (9124h)

The response message is the following:

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
580h + Node-ID	8	60h	24h	91h	01h	00h	00h	00h	00h

Figure 61. Offset adjustment response (9124h)

SAFETY NOTE

The offset adjustment impacts on the safety-related application parameter configuration.

If the value is modified, the safety-related application parameter configuration is set to invalid. The device goes in Pre-operational state and the safety-related application parameter configuration must be validated again.

The change of the value is not allowed when the device is in Operational state.

11.4. AUTO-ZERO

The auto-zero command sets the zero offset value so that the instantaneous measured PV becomes zero.

The autozero command is executed by writing the signature value of “zero“ to the object 6125h.

The offset value (Ai Input offset) is automatically calculated. It can be read through the objects 6124h (float) or 9124h (integer32).

Similarly to the offset adjustment, after the autozero is performed, the whole calibration characteristic is shifted by the calculated offset.

The autozero procedure is described below:

1. The user applies a pressure of zero (e.g. 0 bar)
2. The user launches the autozero command through an SDO write command (see below)
3. The user waits for the correct SDO write response from the device

In order to launch the autozero command, the user must send the following SDO write command:

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
600h + Node-ID	8	22h	25h	61h	01h	7Ah	65h	72h	6Fh

Figure 62. Autozero command request

The response message is the following:

COB-ID	DLC	CS	index		sub-index	value			
		D0	D1	D2	D3	D4	D5	D6	D7
580h + Node-ID	8	60h	25h	61h	01h	00h	00h	00h	00h

Figure 63. Autozero command response

Note:

- The autozero command must be executed when the pressure is near 0 bar (or equivalent pressure value).
  The device automatically detects this condition. The offset value, calculated by the autozero function, that can be accepted must be within a range of ±5%FS.
  Otherwise, an SDO abort will be given, and the autozero procedure will not be completed..

SAFETY NOTE
The autozero command impacts on the safety-related application parameter configuration. If the command is executed, the safety-related application parameter configuration can be set to invalid. The device goes in Pre-operational state and the safety-related application parameter configuration must be validated again. The autozero command is not allowed when the device is in Operational state.

12. SRDO SERVICES

The Safety-related data transfer is performed by means of an SRDO.
An SRDO consists of two CAN data frames with CAN-IDs which are different in at least two bit positions.

SRDO Transmission

The cyclic transmission rate is defined by the refresh-time.

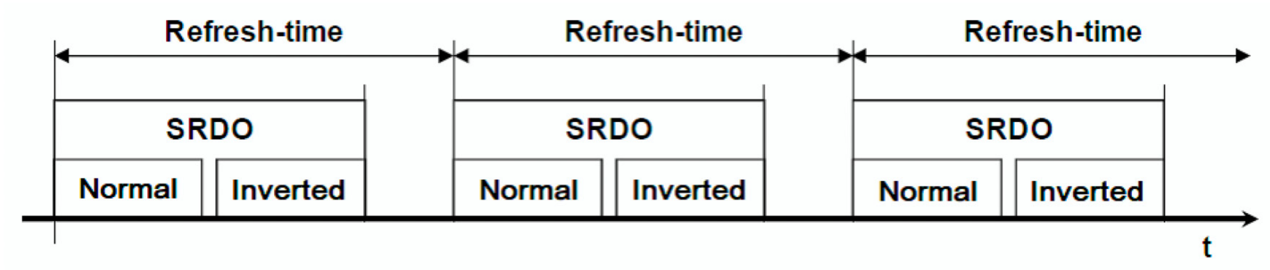


Figure 64. Refresh-time of an SRDO

The second data frame is transmitted immediately after the transmission of the first CAN data frame is finished.
The satety-related application data of the second CAN data frame is the bitwise inverted version of the safety-related data of the first CAN data frame.

SAFETY NOTE

Before configuring the device, the configuration tool shall proof the authenticity of the device to be configured. For that purpose, the identity object (1018h), with the exception of the sub-index 04h (serial number), shall be double-checked with the corresponding EDS by means of SDO read access.

Safety configuration signature

This object is used to secure and verify the configuration of the SRDO (communication and mapping parameters). To each SRDO a safety configuration signature (CRC) is applied.

An external configuration tool downloads the configuration data of an SRDO into the device. The configuration tool then calculates a CRC signature based on the configuration data of the SRDO and downloads this calculated CRC signature into the device.

The device also calculates the CRC based on the configuration data of the SRDO and then compares the downloaded CRC signature with the calculated CRC signature. If both matches the CRC configuration is correct.

The device and the configuration tool use the CRC algorithm with the generator polynomial as defined in (1) and the data as defined in (2), as specified in the EN 50325-5.

$$g(x) = x^{16} + x^{12} + x^5 + 1 \qquad (1)$$

$$\begin{aligned} d(x) = {} & a_{7\ to\ 0} \\ & + b_{7\ to\ 0} + b_{15\ to\ 8} \\ & + c_{7\ to\ 0} \\ & + d_{7\ to\ 0} + d_{15\ to\ 8} + d_{23\ to\ 16} + d_{31\ to\ 24} \\ & + e_{7\ to\ 0} + e_{15\ to\ 8} + e_{23\ to\ 16} + e_{31\ to\ 24} \\ & + f_{7\ to\ 0} \\ & + g^1_{7\ to\ 0} \\ & + h^1_{7\ to\ 0} + h^1_{15\ to\ 8} + h^1_{23\ to\ 16} + h^1_{31\ to\ 24} \\ & + g^2_{7\ to\ 0} \\ & + h^2_{7\ to\ 0} + h^2_{15\ to\ 8} + h^2_{23\ to\ 16} + h^2_{31\ to\ 24} \\ & \qquad to \\ & + g^{128}_{7\ to\ 0} \\ & + h^{128}_{7\ to\ 0} + h^{128}_{15\ to\ 8} + h^{128}_{23\ to\ 16} + h^{128}_{31\ to\ 24} \qquad (2) \end{aligned}$$

The following table shows SRDO parameter data used to calculate the CRC signature of SRDO1.
For the calculation of the CRC for SRDO1 objects 1301h and 1381h are used.

Order	Index	Sub-index	Name	Size	Value
	1301_h		**SRDO communication parameter**		
1		01_h	Information direction	1 octet	a_7 to a_0
2		02_h	Refresh-time / SCT	2 octets	b_{15} to b_0
3		03_h	SRVT	1 octet	c_7 to c_0
4		05_h	COB-ID 1	4 octets	d_{31} to d_0
5		06_h	COB-ID 2	4 octets	e_{31} to e_0
	1381_h		**SRDO mapping parameter**		
6		00_h		1 octet	f_7 to f_0
7		01_h	Sub-index	1 octet	g_7^1 to g_0^1 $(01h)$
8		01_h	SR application data object 1 (plain data)	4 octets	h_{31}^1 to h_0^1
9		02_h	Sub-index	1 octet	g_7^2 to g_0^2 $(02h)$
10		02_h	SR application data object 1 (bitwise inverted data)	4 octets	h_{31}^2 to h_0^2
11		03_h	Sub-index	1 octet	g_7^3 to g_0^3 $(03h)$
12		03_h	SR application data object 2 (plain data)	4 octets	h_{31}^3 to h_0^3
13		04_h	Sub-index	1 octet	g_7^4 to g_0^4 $(04h)$
14		04_h	SR application data object 2 (bitwise inverted data)	4 octets	h_{31}^4 to h_0^4
to		to			
259		$7F_h$	Sub-index	1 octet	g_7^{127} to g_0^{127} $(7Fh)$
260		$7F_h$	SR application data object 64 (plain data)	4 octets	h_{31}^{127} to h_0^{127}
261		80_h	Sub-index	1 octet	g_7^{128} to g_0^{128} $(80h)$
262		80_h	SR application data object 64 (bitwise inverted data)	4 octets	h_{31}^{128} to h_0^{128}

Table 24. data for CRC calculation of SRDO1

The calculation of the CRC signature for the SRDO2 is similar, but objects 1302h and 1382h are used.

The calculated CRC of the SRDO1 must be downloaded into the object 13FFh, sub-index 1 (SRDO1 signature).
The calculated CRC of the SRDO2 must be downloaded into the object 13FFh, sub-index 2 (SRDO2 signature).

Configuration valid

This object is used to validate the SRDO configuration. The configuration tool downloads the value A5h into the object 13FEh, sub-index 0 (configuration valid). If the CRCs of the SRDO have been previously written and are correct, the validation of the SRDO is done. If at least one of the CRCs is wrong, the configuration valid is set to 00h, and the device returns in safe state (pre-operational).

SAFETY NOTE

Before configuring the device, the configuration tool shall proof the authenticity of the device to be configured. For that purpose, the identity object, with the exception of the sub-index 04h (serial number), shall be double-checked with the corresponding EDS by means of SDO read access.

Safety-related application configuration signature

This object is used to secure and verify the configuration of the safety-related application parameters.
For the KMC Safety device the safety-related application parameters are those listed in the following table.

Index / Sub-index	Parameter Name in CiA 404
6114, 1	AI ADC sample rate
6121, 1	AI input scaling 1 PV (Float)
6123, 1	AI input scaling 2 PV (Float)
6124, 1	AI input offset (Float)
6131, 1	AI physical unit PV
6132, 1	AI decimal digits PV
61A0, 1	AI filter type
61A1, 1	AI filter constant
9121, 1	AI input scaling 1 PV (Integer32)
9123, 1	AI input scaling 2 PV (Integer32)
9124, 1	AI input offset (Integer32)

Table 25. Safety-related application parameters list

To this set of safety-related application parameters a safety configuration signature (CRC) is applied.

An external configuration tool downloads the configuration data of the safety-related application parameters into the device. The configuration tool then calculates a CRC signature based on the configuration data of the safety-related application parameters and downloads this calculated CRC signature into the device.

The device also calculates the CRC based on the configuration data of the safety-related application parameters and then compares the downloaded CRC signature with the calculated CRC signature. If both matches the CRC configuration is correct.

The device and the configuration tool use the CRC algorithm with the generator polynomial as defined in (3) and the data as defined in (4).

$$g(x) = x^{16} + x^{12} + x^5 + 1 \qquad (3)$$

$$\begin{aligned} d(x) = {} & a_{7\,to\,0} + a_{15\,to\,8} + a_{23\,to\,16} + a_{31\,to\,24} \\ & + b_{7\,to\,0} + b_{15\,to\,8} + b_{23\,to\,16} + b_{31\,to\,24} \\ & + c_{7\,to\,0} + c_{15\,to\,8} + c_{23\,to\,16} + c_{31\,to\,24} \\ & + d_{7\,to\,0} + d_{15\,to\,8} + d_{23\,to\,16} + d_{31\,to\,24} \\ & + e_{7\,to\,0} + e_{15\,to\,8} + e_{23\,to\,16} + e_{31\,to\,24} \\ & + f_{7\,to\,0} \\ & + g_{7\,to\,0} \\ & + h_{7\,to\,0} + h_{15\,to\,8} \\ & + i_{7\,to\,0} + i_{15\,to\,8} + i_{23\,to\,16} + i_{31\,to\,24} \\ & + j_{7\,to\,0} + j_{15\,to\,8} + j_{23\,to\,16} + j_{31\,to\,24} \\ & + k_{7\,to\,0} + k_{15\,to\,8} + k_{23\,to\,16} + k_{31\,to\,24} \qquad (4) \end{aligned}$$

The following table shows safety-related application parameters data used to calculate the CRC signature of Safety-related application.

Order	Index	Sub-index	Name	Size	Value
1	6114, 1	01h	AI ADC sample rate	4 octets	a_{31} to a_0
2	6121, 1	01h	AI input scaling 1 PV (Float)	4 octets	b_{31} to b_0
3	6123, 1	01h	AI input scaling 2 PV (Float)	4 octets	c_{31} to c_0
4	6124, 1	01h	AI input offset (Float)	4 octets	d_{31} to d_0
5	6131, 1	01h	AI physical unit PV	4 octets	e_{31} to e_0
6	6132, 1	01h	AI decimal digits PV	1 octet	f_7 to f_0
7	61A0, 1	01h	AI filter type	1 octet	g_7 to g_0
8	61A1, 1	01h	AI filter constant	2 octets	h_{15} to h_0
9	9121, 1	01h	AI input scaling 1 PV (Integer32)	4 octets	i_{31} to i_0
10	9123, 1	01h	AI input scaling 2 PV (Integer32)	4 octets	j_{31} to j_0
11	9124, 1	01h	AI input offset (Integer32)	4 octets	k_{31} to k_0

Table 26. Parameter data for CRC calculation of SRDO1

The calculated CRC of the safety-related application parameters must be downloaded into the object 51FFh, sub-index 1.

Configuration valid

This object is used to validate the safety-related application parameter configuration. The configuration tool downloads the value A5h into the object 51FEh, sub-index 0 (configuration valid). If the CRC of the safety-related application parameters has been previously written and is correct, the validation of the safety-related application parameters is done. If the CRC is wrong, the configuration valid is set to 00h, and the device returns in safe state (pre-operational).

15. DISABLING THE SAFETY-RELATED APPLICATION PARAMETERS CHECK

If desired, it is possible to disable the Safety-related application parameters check by the object 51FDh.

If the check is disabled, the procedure for the validation of the Safety-related application parameters will not be required. In this case, only the SRDO validation is required as by EN 50325-5 specification.

For safety reasons, the object 51FDh is write protected. It is possible to change its value only after the password “sfty” is written inside the object 51FCh.

Step 1- Unlock the password protection

The user writes by SDO the password “sfty” inside the object 51FCh, sub-index 0.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	23h	FCh	51h	00h	73h	66h	74h	79h

Step 2- Disable the Safety application configuration check

The user writes by SDO the value “0” inside the object 51FDh, sub- index 0.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Fh	FDh	51h	00h	00h	00h	00h	00h

Step 3- Store configuration

The parameter can be saved in non-volatile memory with the standard command 1010h (“store all parameters”).

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	23h	10h	10h	01h	73h	61h	76h	65h

NOTE: to enable again the Safety-related application parameters check, follow the steps above where at step 2 the value “1” is written instead of “0” inside the object 51FDh, sub-index 0.

SAFETY NOTE

It is not recommended to disable the Safety-related application parameter check.
Doing so, the device cannot detect unintended or not authorized changes to safety-related application parameters.
This could decrease the safety level of the System.

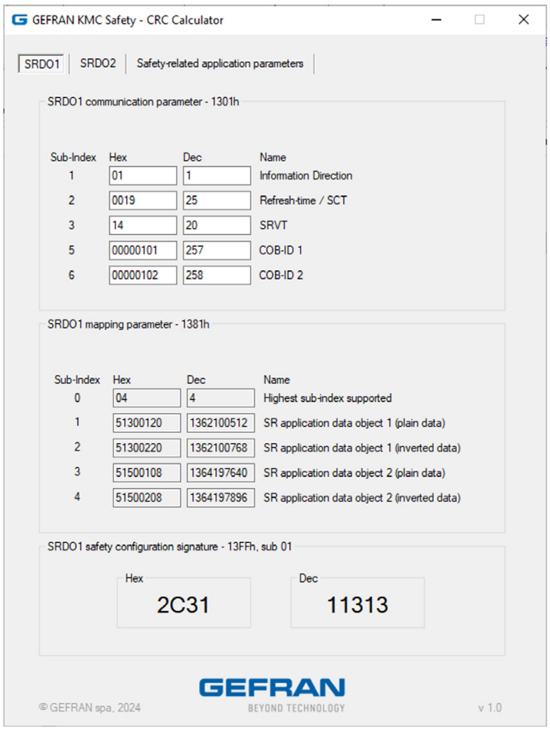
16. CONFIGURATION TOOLS FOR CRC CALCULATION

The CANopen Safety master should integrate a tool for the calculation of the CRCs for the SRDOs and for the validation of the SRDO, as defined in the EN 50325-5. Please refer to the user manual of the specific CANopen Safety master.
The calculation of the CRC for the safety-related application parameters and its validation is not defined inside the CiA DS404 profile. The KMC Safety device uses a manufacturer specific procedure for the safety-related application parameters configuration.
For this purpose, GEFRAN make available a SW tool for the calculation of the CRC for the safety-related application parameters (but also for the calculation of the CRCs for the SRDO). This tool is called “GEFRAN KMC Safety CRC Calculator”. It is a PC-based software and is freely downloadable from the KMC Safety product page of the GEFRAN website (https://www.gefran.com).

Below you can find a practical example that shows the steps followed for the validation of the device’s configuration with the help of the “GEFRAN KMC Safety CRC Calculator”.
The example is for a KMC Safety device with Node-ID = 1, FS = 1000 bar in a default parameters configuration.

Step 1- SRDO Configuration

- **SRDO1 CRC Calculation**

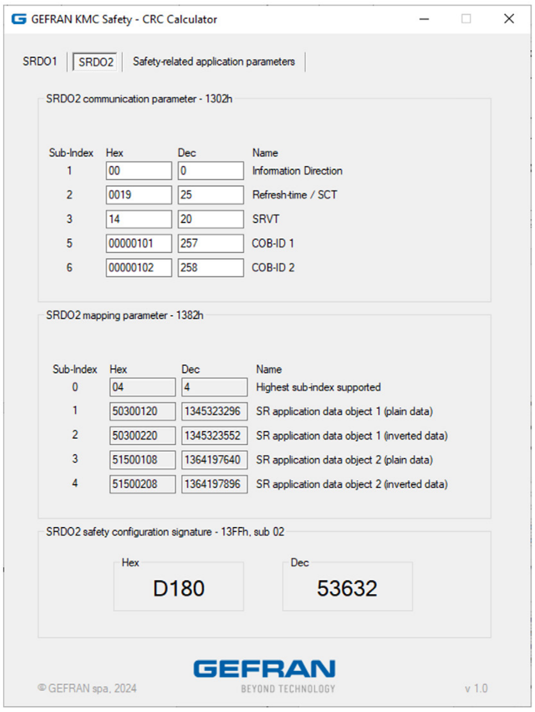


The user sets the required data for SRDO1 as configured in the device (decimal or hexadecimal formats available). The CRC Calculator tool calculates the CRC of the SRDO1.

The user writes by SDO the SRDO1 safety configuration signature (CRC of the SRDO1) inside the object 13FFh, sub-index 1.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Bh	FFh	13h	01h	31h	2Ch	00h	00h

- **SRDO2 CRC Calculation**



The user sets the required data for SRDO2 as configured in the device (decimal or hexadecimal formats available). The CRC Calculator tool calculates the CRC of the SRDO2.

The user writes by SDO the SRDO2 safety configuration signature (CRC of the SRDO2) inside the object 13FFh, sub-index 2.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Bh	FFh	13h	02h	80h	D1h	00h	00h

- **SRDO Validation**

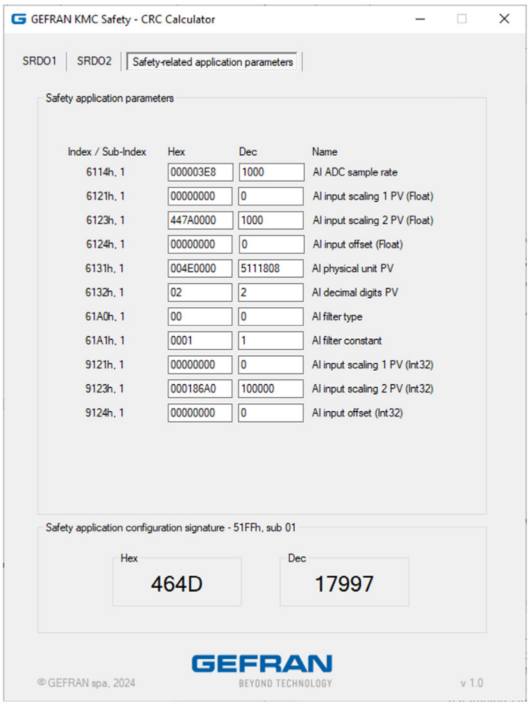
The user writes by SDO the SRDO configuration valid flag (A5h) inside the object 13FEh, sub-index 0.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Fh	FEh	13h	00h	A5h	00h	00h	00h

If the SRDO CRCs previously written are correct an SDO confirm is returned, and the SRDO configuration is validated, otherwise an SDO abort is returned

Step 2- SR Application Configuration

- **SR Application CRC Calculation**



The user sets the required data for the Safety-related application parameters as configured in the device (decimal or hexadecimal formats available). The CRC Calculator tool calculates the CRC of the Safety-related application parameters.

The user writes by SDO the Safety application configuration signature (CRC of the Safety-related application parameters) inside the object 51FFh, sub-index 1.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Bh	FFh	51h	01h	4Dh	46h	00h	00h

- **SR Application Validation**

The user writes by SDO the Safety application configuration valid flag (A5h) inside the object 51FEh, sub-index 0.

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	2Fh	FEh	51h	00h	A5h	00h	00h	00h

If the Safety application CRC previously written is correct an SDO confirm is returned, and the Safety application configuration is validated, otherwise an SDO abort is returned

Step 3- Store configuration

If step 1 and step 2 are executed correctly the validation is completed, and the device can be sent in Operational state. The configuration of the CRCs and the validation can be saved in non-volatile memory with the standard command 1010h ("store all parameters").

COB-ID	Rx/Tx	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
601h	Tx	8	23h	10h	10h	01h	73h	61h	76h	65h

17. C-CODE FOR CRC CALCULATION

Below you can find an example of functions written in C-Code that implements:

- CRC calculation algorithm as defined in the EN 50325-5
- CRC calculation sequence for the SRDO1
- CRC calculation sequence for the SRDO2
- CRC calculation sequence for the Safety-related application parameters

These function can be implemented in a PLC software to automate the process of CRCs calculation.

NOTE:
The CANopen Safety master should integrate a tool for the calculation of the CRCs for the SRDOs and for the validation of the SRDO, as defined in the EN 50325-5. Please refer to the user manual of the specific CANopen Safety master.

CRC calculation algorithm

The CRC calculation algorithm for SRDO1, SRDO2 is compliant with the EN 50325-5 document.
The CRC calculation algorithm for the Safety-related application parameters is the same.
It can be done with the help of this function (C-Code).

```
uint16_t EN50325_5_CrcCalc(uint16_t crc, uint8_t  value)
{
 static const uint16_t EN50325_5_crc_tabccitt[256] = {
 0x0000u, 0x1021u, 0x2042u, 0x3063u, 0x4084u, 0x50a5u, 0x60c6u, 0x70e7u,
 0x8108u, 0x9129u, 0xa14au, 0xb16bu, 0xc18cu, 0xd1adu, 0xe1ceu, 0xf1efu,
 0x1231u, 0x0210u, 0x3273u, 0x2252u, 0x52b5u, 0x4294u, 0x72f7u, 0x62d6u,
 0x9339u, 0x8318u, 0xb37bu, 0xa35au, 0xd3bdu, 0xc39cu, 0xf3ffu, 0xe3deu,
 0x2462u, 0x3443u, 0x0420u, 0x1401u, 0x64e6u, 0x74c7u, 0x44a4u, 0x5485u,
 0xa56au, 0xb54bu, 0x8528u, 0x9509u, 0xe5eeu, 0xf5cfu, 0xc5acu, 0xd58du,
 0x3653u, 0x2672u, 0x1611u, 0x0630u, 0x76d7u, 0x66f6u, 0x5695u, 0x46b4u,
 0xb75bu, 0xa77au, 0x9719u, 0x8738u, 0xf7dfu, 0xe7feu, 0xd79du, 0xc7bcu,
 0x48c4u, 0x58e5u, 0x6886u, 0x78a7u, 0x0840u, 0x1861u, 0x2802u, 0x3823u,
 0xc9ccu, 0xd9edu, 0xe98eu, 0xf9afu, 0x8948u, 0x9969u, 0xa90au, 0xb92bu,
 0x5af5u, 0x4ad4u, 0x7ab7u, 0x6a96u, 0x1a71u, 0x0a50u, 0x3a33u, 0x2a12u,
 0xdbfdu, 0xcbdcu, 0xfbbfu, 0xeb9eu, 0x9b79u, 0x8b58u, 0xbb3bu, 0xab1au,
 0x6ca6u, 0x7c87u, 0x4ce4u, 0x5cc5u, 0x2c22u, 0x3c03u, 0x0c60u, 0x1c41u,
 0xedaeu, 0xfd8fu, 0xcdecu, 0xddcdu, 0xad2au, 0xbd0bu, 0x8d68u, 0x9d49u,
 0x7e97u, 0x6eb6u, 0x5ed5u, 0x4ef4u, 0x3e13u, 0x2e32u, 0x1e51u, 0x0e70u,
 0xff9fu, 0xefbeu, 0xdfddu, 0xcffcu, 0xbf1bu, 0xaf3au, 0x9f59u, 0x8f78u,
 0x9188u, 0x81a9u, 0xb1cau, 0xa1ebu, 0xd10cu, 0xc12du, 0xf14eu, 0xe16fu,
 0x1080u, 0x00a1u, 0x30c2u, 0x20e3u, 0x5004u, 0x4025u, 0x7046u, 0x6067u,
 0x83b9u, 0x9398u, 0xa3fbu, 0xb3dau, 0xc33du, 0xd31cu, 0xe37fu, 0xf35eu,
 0x02b1u, 0x1290u, 0x22f3u, 0x32d2u, 0x4235u, 0x5214u, 0x6277u, 0x7256u,
 0xb5eau, 0xa5cbu, 0x95a8u, 0x8589u, 0xf56eu, 0xe54fu, 0xd52cu, 0xc50du,
 0x34e2u, 0x24c3u, 0x14a0u, 0x0481u, 0x7466u, 0x6447u, 0x5424u, 0x4405u,
 0xa7dbu, 0xb7fau, 0x8799u, 0x97b8u, 0xe75fu, 0xf77eu, 0xc71du, 0xd73cu,
 0x26d3u, 0x36f2u, 0x0691u, 0x16b0u, 0x6657u, 0x7676u, 0x4615u, 0x5634u,
 0xd94cu, 0xc96du, 0xf90eu, 0xe92fu, 0x99c8u, 0x89e9u, 0xb98au, 0xa9abu,
 0x5844u, 0x4865u, 0x7806u, 0x6827u, 0x18c0u, 0x08e1u, 0x3882u, 0x28a3u,
 0xcb7du, 0xdb5cu, 0xeb3fu, 0xfb1eu, 0x8bf9u, 0x9bd8u, 0xabbbu, 0xbb9au,
 0x4a75u, 0x5a54u, 0x6a37u, 0x7a16u, 0x0af1u, 0x1ad0u, 0x2ab3u, 0x3a92u,
 0xfd2eu, 0xed0fu, 0xdd6cu, 0xcd4du, 0xbdaau, 0xad8bu, 0x9de8u, 0x8dc9u,
 0x7c26u, 0x6c07u, 0x5c64u, 0x4c45u, 0x3ca2u, 0x2c83u, 0x1ce0u, 0x0cc1u,
 0xef1fu, 0xff3eu, 0xcf5du, 0xdf7cu, 0xaf9bu, 0xbfbau, 0x8fd9u, 0x9ff8u,
 0x6e17u, 0x7e36u, 0x4e55u, 0x5e74u, 0x2e93u, 0x3eb2u, 0x0ed1u, 0x1ef0u};

 uint16_t tmp, x;

 x = (uint16_t)value;
 x &= 0xffu;

 tmp = (crc >> 8) ^ x;
 crc = ((crc & 0xffu) << 8) ^ EN50325_5_crc_tabccitt[tmp];

 return(crc);
}
```

CRC calculation sequence for SRDO1 or SRDO2

The CRC calculation sequence for SRDO1 and SRDO2 is compliant with the EN 50325-5 document.
It can be done with the help of this function (C-Code).

```
uint16_t SRDO1CrcCalculation(SRDOxData_T SRDOxData)
{

  uint16_t crc;
  uint8_t  tmp;

  crc = 0u;

  /* 1301, 1: SRDO1 Information direction or
     1302, 1: SRDO2 Information direction */
  tmp = SRDOxData.direction;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1301, 2: SRDO1 Refresh time or
     1302, 2: SRDO2 Refresh time */
  tmp = (uint8_t)(SRDOxData.refreshTime & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.refreshTime >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1301, 3: SRDO1 SRVT or
              1302, 3: SRDO2 SRVT */
  tmp = SRDOxData.srvt;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1301, 4: SRDO1 COB-ID 1 or
     1302, 4: SRDO2 COB-ID 1 */
  tmp = (uint8_t)(SRDOxData.cobId1 & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId1 >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId1 >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId1 >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1301, 5: SRDO1 COB-ID 2 */
     1302, 5: SRDO2 COB-ID 2 */
  tmp = (uint8_t)(SRDOxData.cobId2 & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId2 >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId2 >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.cobId2 >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1381, 0: SRDO1 Mapping parameter - Highest sub-index supported or
     1382, 0: SRDO2 Mapping parameter - Highest sub-index supported */
  tmp = 4u;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* SRDO1 Mapping parameter - Sub-index 1 or
     SRDO2 Mapping parameter - Sub-index 1 */
  tmp = 1u;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1381, 1: SRDO1 Mapping parameter - SR application data object 1 (plain data) or
     1382, 1: SRDO2 Mapping parameter - SR application data object 1 (plain data) */
  tmp = (uint8_t)(SRDOxData.SRapplicationDataObject1_plain & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_plain >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
```

```
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_plain >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_plain >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* SRDO1 Mapping parameter - Sub-index 2 or
     SRDO2 Mapping parameter - Sub-index 2 */
  tmp = 2u;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1381, 2: SRDO1 Mapping parameter - SR application data object 1 (inverted data) or
     1382, 2: SRDO2 Mapping parameter - SR application data object 1 (inverted data) */
  tmp = (uint8_t)(SRDOxData.SRapplicationDataObject1_inverted & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_inverted >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_inverted >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject1_inverted >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* SRDO1 Mapping parameter - Sub-index 3 or
     SRDO2 Mapping parameter - Sub-index 3 */
  tmp = 3u;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1381, 3: SRDO1 Mapping parameter - SR application data object 2 (plain data) or
     1382, 3: SRDO2 Mapping parameter - SR application data object 2 (plain data) */
  tmp = (uint8_t)(SRDOxData.SRapplicationDataObject2_plain & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_plain >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_plain >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_plain >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* SRDO1 Mapping parameter - Sub-index 4 or
     SRDO2 Mapping parameter - Sub-index 4 */
  tmp = 4u;
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 1381, 4: SRDO1 Mapping parameter - SR application data object 2 (inverted data) or
     1382, 4: SRDO2 Mapping parameter - SR application data object 2 (inverted data) */
  tmp = (uint8_t)(SRDOxData.SRapplicationDataObject2_inverted & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_inverted >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_inverted >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((SRDOxData.SRapplicationDataObject2_inverted >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  return(crc);

}
```

CRC calculation sequence for SR application parameters

The CRC calculation for the safety-related application parameters is manufacturer specific for the KMC Safety.
It can be done with the help of this function (C-Code).

```
uint16_t SafetyApplicationCrcCalculation(AppData_T AppData)
{

  uint16_t crc;
  uint8_t  tmp;

  crc = 0u;

  /* 6114, 1: AI ADC sample rate */
  tmp = (uint8_t)(AppData.AI_ADC_sample_rate & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_ADC_sample_rate >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_ADC_sample_rate >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_ADC_sample_rate >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 6121, 1: AI input scaling 1 PV (Float) */
  tmp = (uint8_t)(AppData.AI_input_scaling_1_PV_float_IEEE754 & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_1_PV_float_IEEE754 >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_1_PV_float_IEEE754 >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_1_PV_float_IEEE754 >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 6123, 1: AI input scaling 2 PV (Float) */
  tmp = (uint8_t)(AppData.AI_input_scaling_2_PV_float_IEEE754 & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_2_PV_float_IEEE754 >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_2_PV_float_IEEE754 >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_scaling_2_PV_float_IEEE754 >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 6124, 1: AI input offset (Float) */
  tmp = (uint8_t)(AppData.AI_input_offset_float_IEEE754 & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_offset_float_IEEE754 >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_offset_float_IEEE754 >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_input_offset_float_IEEE754 >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 6131, 1: AI physical unit PV */
  tmp = (uint8_t)(AppData.AI_physical_unit_PV & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_physical_unit_PV >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_physical_unit_PV >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_physical_unit_PV >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 6132, 1: AI decimal digits PV */
  crc = EN50325_5_CrcCalc(crc, AppData.AI_decimal_digits_PV);
```

```
  /* 61A0, 1: AI filter type */
  crc = EN50325_5_CrcCalc(crc, AppData.AI_filter_type);

  /* 61A1, 1: AI filter constant */
  tmp = (uint8_t)(AppData.AI_filter_constant & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)((AppData.AI_filter_constant >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 9121, 1: AI input scaling 1 PV (Integer32) */
  tmp = (uint8_t)((uint32_t)AppData.AI_input_scaling_1_PV_integer & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_1_PV_integer >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_1_PV_integer >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_1_PV_integer >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 9123, 1: AI input scaling 2 PV (Integer32) */
  tmp = (uint8_t)((uint32_t)AppData.AI_input_scaling_2_PV_integer & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_2_PV_integer >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_2_PV_integer >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_scaling_2_PV_integer >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  /* 9124, 1: AI input offset (Integer32) */
  tmp = (uint8_t)((uint32_t)AppData.AI_input_offset_integer & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_offset_integer >> 8) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_offset_integer >> 16) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);
  tmp = (uint8_t)(((uint32_t)AppData.AI_input_offset_integer >> 24) & 0xffu);
  crc = EN50325_5_CrcCalc(crc, tmp);

  return(crc);

}
```

GEFRAN

GEFRAN spa
via Sebina, 74 - 25050 PROVAGLIO D'ISEO (BS) - ITALIA
tel. 0309888.1 - fax. 0309839063 Internet: http://www.gefran.com