GEFRAN

RK-5 C

(CANopen OUTPUT)

User Manual

Code 85201A **Edition** 02-2017

SUMMARY

1. INTRODUCTION

The GEFRAN RK5C is a Digital Linear Position Sensor with CANopen interface. It implements the standard CANopen communications protocol defined by CiA (CAN in Automation).

The CANopen standards supported by the device are listed in the following table.

CiA standard	Description	Version
DS 301	CANopen application layer and communication profile	4.2.0
DS 305	Layer setting services (LSS) and protocols	3.0.1
DS 406	Device profile for encoders	3.2.0

Table 1 - Supported CANopen standards

This document describes the CANopen implementation on the GEFRAN RK5C CANopen device. It is addressed to CANopen network system integrators and to CANopen device designers who already know the content of the above-mentioned standards defined by CiA.

The details of aspects defined by CANopen do not pertain to the purpose of this text. For further information on the CANopen protocol see www.can-cia.de

2. GET STARTED PROCEDURE

2.1 NODE PARAMETERS SETTING

Before connecting the GEFRAN RK5C sensor 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 RK5C sensor is identical to the Node-ID of another CANopen device connected to the CAN bus.
2) The GEFRAN RK5C sensor 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 CANopen 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 RK5C sensor 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; **7Eh***; 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 (126 in this example).
** if value is 1, it means Node-ID out of range, i.e. the command was not accepted.

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
Controller	700h + Node-ID	01h	00h	Controller

Figure 5 – Boot-up message format

2.2 OPERATING 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 transmission type of the PDO (Process Data Object) selecting the synchronous (through SYNC messages) or asynchronous (through event-timer) mode, or transmission time (event timer) of the asynchronous PDO.

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).

2.3 REQUESTING PROCESS DATA

The GEFRAN RK5C CANopen position sensor provides one Transmit PDO (TPDO1), that includes position and speed data measured by the sensor.

TPDO1 data format

Position and speed data are mapped in TPDO1 as shown in the following figure.

COB-ID	DLC	D0	D1	D2	D3	D4	D5
180h + Node-ID	6	Position value				Speed value	

Figure 6 - TPDO1 mapped data

The position data is expressed with a fixed resolution corresponding to 100 µm, in INTEGER32 data format. The speed data in expressed with a 1mm/s resolution, in INTEGER16 data format. Byte ordering of position and speed data inside TPDO1 follows the LSB..MSB ordering scheme.

Position and speed values are calculated as follows:

Position [µm]= Position value * 100 µm

Speed [mm/s] = Speed value * 1 mm/s

TPDO1 data transmission

The transmission of the Process Data Object 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 7 - 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 8 - NMT “Enter NMT pre-operational” command

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

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							
7E5h	Rx	8	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
			04h	01h	00h	00h	00h	00h	00h	00h

Figure 9 - 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							
7E5h	Rx	8	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
			04h	00h	00h	00h	00h	00h	00h	00h

Figure 10 - 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 RK5C CANopen device is specified in the following table.

	Address Field	Value
LSS Address	Vendor-ID	00000093h
	Product code	43354B52h
	Revision Number	Actual RK5C r.n.*
	Serial Number	Actual RK5C s.n. (printed on the label)**

Figure 11 - RK5C LSS Address

* Actual Revision number can vary.
The user can inquire the Revision number with LSS Inquire Revision Number command (see LSS Inquire Services).

** Actual Serial number is device specific.
It is printed on the label attached to the GEFRAN RK5C transducer case, or it can be inquired with the LSS Inquire Serial Number command (see LSS Inquire Services).

The LSS master sends this message sequence to switch the GEFRAN RK5C CANopen device 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	52h	4Bh	35h	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 12 - 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 Serial Number is assigned by GEFRAN to the RK5C 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 13 - 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 SYNC (1005h)
- COB-ID EMCY (1014h)
- COB-ID SDO rx (1200h, sub 1)
- COB-ID SDO tx (1200h, sub 2)
- COB-ID TPDO (1800h, sub 1)

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

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
5	Reserved
6	50
7	20
8	10

Table 2 - Table 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 14 - 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 15 - 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 16 - LSS store configuration message

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

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 17 - 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 18 - 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 19 - 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 20 - 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 21 - 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				Reserved		
7E5h	Rx	8	47h	Product-code				Reserved		
7E5h	Rx	8	48h	Revision number low				Reserved		
7E5h	Rx	8	49h	Revision number high				Reserved		
7E5h	Rx	8	4Ah	Serial number low				Reserved		
7E5h	Rx	8	4Bh	Serial number high				Reserved		

Figure 22 - 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	Reserved						

Figure 23 - 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	Reserved						

Figure 24 - 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	Reserved						

Figure 25 - 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 RK5C CANopen 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 26 - 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 RK5C CANopen 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 27 - 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 28 - 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.

The following table contains the abort codes provided by the protocol SDO abort transfer of the GEFRAN RK5C CANopen device.

Abort code	Description
05040001h	Client/server command specifier not valid or unknown
05040005h	Out of memory
06010001h	Attempt to read a write only object
06010002h	Attempt to write a read only object
06020000h	Object does not exist in the object dictionary
06040041h	Object cannot be mapped to the PDO
06070010h	Data type does not match, length of service parameter does not match
06090011h	Sub-index does not exist
06090030h	Invalid value for parameter (download only)
08000020h	Data cannot be transferred or stored to the application.

Figure 29 - SDO abort codes

4.1 OBJECT DICTIONARY

The object dictionary of the GEFRAN RK5C CANopen 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	000A0196h	Multi-sensor encoder interface with ds406 device profile
1001h	0	Error register	Unsigned8	RO	-	00h: no error 81h: transducer error
1002h	0	Manufacturer status register	Unsigned32	RO	-	Common status register for manufacturer-specific purposes
1005h	0	COB-ID SYNC	Unsigned32	RW	00000080h	Configured COB-ID of the synchronization object (SYNC)
1008h	0	Manufacturer device name	Visible string	RO	RK5C	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
1010h	0	Store parameters	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned32	RW	00000001h	Writing the signature “save” (73h, 61h, 76h, 65h) stores all parameters in flash memory
1011h	0	Restore default parameters	Unsigned8	RO	1	Highest sub-index supported
	1		Unsigned32	RW	00000001h	Writing the signature “load” (6Ch, 6Fh, 61h, 64h) restores all parameters in flash to their default values
1014h	0	COB-ID EMCY	Unsigned32	RW	00000080h + Node-ID	Configured COB-ID for the EMCY write service
1015h	0	*Inhibit time EMCY*	Unsigned16	RW	0000h	*Configured inhibit time for the EMCY service*
1017h	0	Producer heartbeat time	Unsigned16	RW	0000h	Configured cycle time of the heartbeat (ms)
1018h	0	Identity object	Unsigned8	RO	4	Highest sub-index supported
	1		Unsigned32	RO	00000093h	Vendor-ID
	2		Unsigned32	RO	43354B52h	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)
1800h	0	TPDO1 communication parameter	Unsigned8	RO	5	Highest sub-index supported
	1		Unsigned32	RW	00000180h + Node-ID	COB-ID of the TPDO1
	2		Unsigned8	RW	FEh	Transmission type
	5		Unsigned16	RW	0001h	Event-timer
1A00h	0	TPDO1 mapping parameter	Unsigned8	RO	2	Number of mapped application objects in TPDO1
	1		Unsigned32	RO	60200120h	1st application object (position)
	2		Unsigned32	RO	60300110h	2nd application object (speed)

Manufacturer Profile Area

Index	Sub index	Name	Type	Access	Default value	Comment
2000h	0	Number of cursors	Unsigned8	RO	1	Number of cursors set for position and speed measuring

Device Profile Area

Index	Sub index	Name	Type	Access	Default value	Comment
6000h	0	Operating parameters	Unsigned16	RW	-	Configuration of the operating parameters of the encoder
6005h	0	Linear encoder measuring step settings	Unsigned8	RO	2	Highest sub-index supported
	1		Unsigned32	RO	100000	Position measuring step given in multiples of 0,001µm
	2		Unsigned32	RO	100	Speed measuring step given in multiples of 0,01mm/s
6010h	0	Preset values for multi-sensor devices	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RW	-	Preset value channel 1
6020h	0	Position values for multi-sensor devices	Unsigned8	RO	1	Highest sub-index supported
	1		Integer32	RO	-	Position value channel 1
6030h	0	Speed value	Unsigned8	RO	1	Highest sub-index supported
	1		Integer16	RO	-	Speed value channel 1
6200h	0	Cyclic timer	Unsigned16	RW	0001h	Transmission period for TPDO1 given in multiples of 1ms
6500h	0	Operating status	Unsigned16	RO	-	Operating status of the encoder functions configured in the object 6000h
6501h	0	Measuring step	Unsigned32	RO	100000	Position measuring step given in multiples of 0,001µm

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 = 000Ah
Device Profile Number = 0196h

Object description

Index	Name
1000h	Device type

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Device type	RO	Unsigned32	000A0196h	000A0196h

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 RK5C CANopen device two types of error conditions are defined: Device hardware error and Data set error.

The “Data Set” type error occurs when a mismatch between the stored checksum and the calculated checksum during a reading operation from the non volatile memory of the device is detected. The checksum is verified during the start-up phase after power-on, during initialization state, after a reset communication or reset device NMT command, and after a SDO write in object 1011h (load default parameters). This error can only be cleared by a hardware device reset.

The “Device Hardware” type error is set when the microcontroller detects anomalies during the sensor low-level measuring routines. The error register contains one of the error codes described in the following table.

Error code	Name
00h	No error
01h	Data set error
81h	Device hardware error or Device hardware error and Data set error

Table 3 - 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,01h,81h	00h

1002h – Manufacturer status register

This object provides a common status register for manufacturer-specific purposes.

The structure of the register is represented in the following figure.

31 24	23 16	15 8	7 0							
0	0	0	0	T	E3	E2	E1	E0	S	N

Figure 31 Structure of the Manufacturer status register

Where

N: Status 0 = sensor in error state
1 = normal running state, valid position and velocity data transmitted

S: Working state 0 = normal running state
1 = start up or internal test mode

E0: Magnet error 0 = one magnet detected
1 = no or more than one magnet detected

E1: Range error 0 = no error
1 = the calculated position is out of range when also the position and velocity values are set to zero
1 = the velocity value may be not correct

E2: Data Flash error 0 = no error
1 = non volatile memory checksum not correct

E3: Controller error 0 = no error
1 = error detected by microcontroller

T: µC temperature state 0 = ok, µC temperature <= 105°C
1 = warning, µC temperature > 105°C

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	00h..7Fh	01h

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 (default) or user defined	11-bit CAN-ID of the CAN base frame

Table 4 - COB-ID SYNC message field

The user can change the default COB-ID SYNC 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.

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	Visible_string	RK5C

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	-

1010h – Store parameters

This object controls the saving of parameters in non-volatile memory.

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

Figure 33 - Storage write access structure

By read access the sub-index 1 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

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	Save all parameters	RW	Unsigned32	Read access: 00000001h Write access: 65766173h (ASCII: “save”)	Read access: 00000001h Write access: 65766173h (ASCII: “save”)

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 the sub-index 1, so that all parameters are restored in non-volatile memory.

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

31			**0**
d (64h)	a (61h)	o (6Fh)	l (6Ch)
MSB			LSB

Figure 34 - Restore write access structure

By read access the sub-index 1 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.

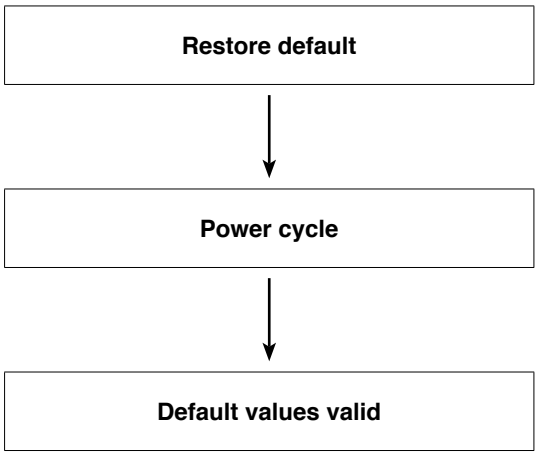

Figure 35 - Restore procedure

For the GEFRAN RK5C CANopen device, the Restore default parameters command does not apply to these objects:

- COB-ID SYNC (1005h)
- COB-ID EMCY (1014h)
- COB-ID of TPDO1 (1800h, sub-index 1)
- COB-IDs of 1st SDO (1200h, sub-index 1 and 2)

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

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	Restore all default parameters	RW	Unsigned32	Read access: 00000001h Write access: 64616F6Ch (ASCII: “load”)	Read access: 00000001h Write access: 64616F6Ch (ASCII: “load”)

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
	1	EMCY does not exists / is not 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 5 - COB-ID EMCY message fields

The user can change the default COB-ID EMCY 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.
By setting bit 31 (valid) of the EMCY COB-ID to 1 the user can disable the EMCY service.

Object description

Index	Name
1014h	COB-ID EMCY

Entry description

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

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 100us. The accepted values must be multiples of 10, i.e. 1ms. 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	0000h..FFFFh as multiples of 10	0000h

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	0000h..FFFFh	0000h

1018h – Producer heartbeat time

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: contain the unique value that identifies a specific type of CANopen device. For the GEFRAN RK5C CANopen device it is 43354B52h.
- 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	43354B52h	43354B52h
3	Revision number	RO	Unsigned32	-	-
4	Serial number	RO	Unsigned32	-	-

The user can also get the identity object values using the LSS inquire identity services (see LSS protocol description section). The specific serial number of the device 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 6 - 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	00000601h..0000067Fh	00000600h + Node-ID
2	COB-ID server --> client (tx)	RO	Unsigned32	00000581h..0000057Fh	00000580h + Node-ID

1800h – TPDO1 communication parameter

This object contains the communication parameters for the PDOs the CANopen device is able to 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 38 - 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 7 - 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 8 - 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 (*)	00000180h + Node-ID
2	Transmission type	RW	Unsigned8	0..240 and 252..255	254
5	Event-timer	RW	Unsigned16	0000h..FFFFh	0001h

(*) 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 39 - 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 9 - 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	2	2
1	1st application object (position)	RW	Unsigned32	60200120h	60200120h
2	2nd application object (speed)	RW	Unsigned32	60300110h	60300110h

2000h – Number of cursors

This object indicates the number of cursors set for position and speed measuring.

Object description

Index	Name
2000h	Number of cursors

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Number of cursors	RO	Unsigned8	01h	01h

6000h – Operating parameters

This object indicates the configuration of the operating parameters of the encoder.

Object description

Index	Name
6000h	Operating parameters

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Operating parameters	RO	Unsigned16	0000h	0000h

This object is not supported by the GEFRAN RK5C CANopen device.

6005h – Linear encoder measuring step settings

This object indicates the measuring step settings for position and speed for linear encoders.

The value of position step setting (sub-index 1) is given in multiples of 0,001 µm.
The value of speed step setting (sub-index 2) is given in multiples of 0,01 mm/s

Object description

Index	Name
6005h	Linear encoder measuring step settings

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	2	2
1	Position step setting	RO	Unsigned32	100000	100000
2	Speed step setting	RO	Unsigned32	100	100

6010h – Preset values for multi-sensor devices

This object indicates the preset value for the output position value of the device. Using the preset function, the user can set the actual output position value to the value specified in sub-index 1 of the object.

The preset value is accepted only if the measuring function of the device is in normal running state (see Manufacturer Status Register, object 1002h), i.e. the position measure value is valid. After a successful execution of the write preset command, the new output position value is updated since the next measuring cycle. The offset from the position is calculated.

Using the “Store parameters” command (see object 1010h), the user can save the offset to the non-volatile memory of the device, so that the output measure value is permanently affected by the preset value. Using the “Restore default parameters” (see object 1011h) the preset value is cancelled, i.e. the offset is set to zero.

The preset value is not saved in non-volatile memory, but it can be read until the next reset of the device after a write preset command. In other cases, a read operation of the sub-index 1 doesn’t give the preset value.

Object description

Index	Name
6010h	Preset values for multi-sensor devices

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	Preset value channel 1	RW	Integer32	Integer32	-

6020h – Position values for multi-sensor devices

This object provides the output position value for the multi-sensor device.

This object is also mapped into sub-index 1 of the PDO1 (see object 1A00h).

Object description

Index	Name
6020h	Position values for multi-sensor devices

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	Position value channel 1	RO	Integer32	Integer32	-

6030h – Speed value

This object provides the output speed value. This object is also mapped into sub-index 2 of the PDO1 (see object 1A00h).

Object description

Index	Name
6030h	Speed value

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Highest sub-index supported	RO	Unsigned8	1	1
1	Preset value channel 1	RO	Integer16	Integer16	-

6200h – Cyclic timer

This object indicates the transmission period for TPDO 1. The values are given in multiples of 1 ms. It is hard-wired to the PDO event timer (see object 1800h). A cyclic transmission of the TPDO1 is set, when the cyclic timer is programmed unequal 0000h and the TPDO1 transmission type is set to 254 or 255.

Cyclic timer settings are explained in the following table.

Value	Description
0	Event-driven TPDO1 transmission disabled
1-65535	TPDO1 event-timer in ms

Figure 40 - Cyclic timer settings

Object description

Index	Name
6200h	Cyclic timer

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Cyclic timer	RW	Unsigned16	0000h..FFFFh	0001h

6500h – Operating status

This object provides the operating status of the encoder functions configured in the object 6000h..

Object description

Index	Name
6500h	Operating status

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Operating status	RO	Unsigned16	0000h	0000h

This object is not supported by the GEFRAN RK5C CANopen device.

6501h – Measuring step

This object provides the position measuring step that is an output of the encoder. The measuring step is given in multiples of 0,001 µm.

This object is similar to the object 6005h, sub-index 1.

Object description

Index	Name
6501h	Measuring step

Entry description

SUB Index	Name	Access	Data Type	Value Range	Default
0	Measuring step	RO	Unsigned32	100000	100000

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 RK5C CANopen device is a PDO producer, its PDO is also called Transmit PDO (TPDO).

5.1 PDO MESSAGE FORMAT

COB-ID	Rx/Tx	DLC	Data					
			D0	D1	D2	D3	D4	D5
180h + Node-ID	Tx	6	Pos LSB	Pos	Pos	Pos MSB	Speed LSB	Speed MSB

Figure 41 - Transmit PDO1 (TPDO1) message format

5.2 PDO DATA TYPES

Two types of data are mapped in PDO1: Position and Speed.
Position data is an INTEGER32 data type.
Speed data is an INTEGER16 data type.

Assuming that the data is expressed as a bit sequence of length 16 for INTEGER16 data type (b0..b15), and as a bit sequence of length 32 for INTEGER32 data type (b0..b31), the transfer syntax is explained in the following figure.

Octet number	1	2	3	4
INTEGER16	$b_7..b_0$	$b_{15}..b_8$	-	-
INTEGER32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$

Figure 42 - Transfer syntax for INTEGERn data type

5.3 PDO MAPPING

The PDO mapping type for the GEFRAN RK5C CANopen device is fixed. See object 1A00h description.

5.4 PDO TRANSMISSION TYPES

The PDO transmission type for the RK5 CANopen 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 43 - 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 44 - 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.

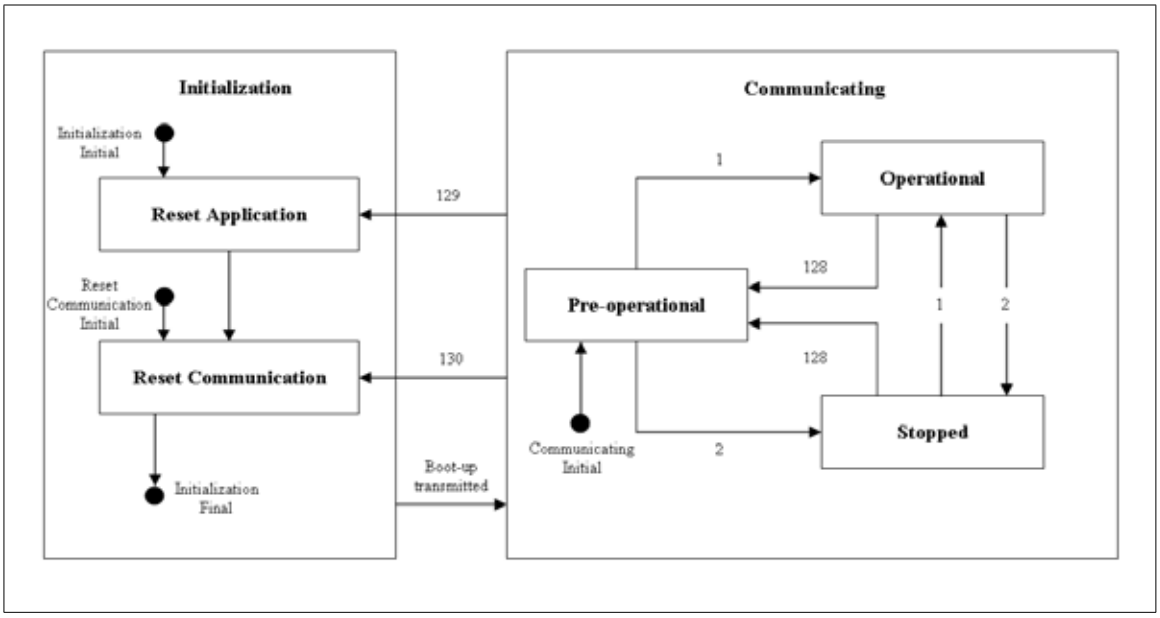

Figure 45 - NMT states and state transitions

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	
SDO			X	X	
Boot up		X			
SYNC			X	X	
EMCY			X	X	
NMT error control (Heartbeat)			X	X	X
NMT node control			X	X	

Table 10 - 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 11 - 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 46 – 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 1800, 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 47 - 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 CANopen 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 CANopen device transmits the emergency object with the updated error code. The error register value inside the EMCY object is also updated. For the GEFRAN RK5C CANopen device two types of error conditions are defined: Device hardware error and Data set error.

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

Error code	Description
0000h	Error reset or no error
5000h	Device hardware
6300h	Data set

Table 12 - EMCY error codes for the RK5C CANopen device

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)	00h	00h	00h	00h	00h

Figure 48 - EMCY message format

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

The error control services are used to detect failures within a CAN-based network. The RK5 CANopen device makes use of the heartbeat mechanism.

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 49 - 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 13 - NMT state field in heartbeat message

For the RK5 CANopen device the heartbeat is disabled by default. It can be programmed through object 1017h.