Baumer

Passion for Sensors

Manual

GK473 - Gateway RS485 - CANopen

Version 1.00 and up

Applicable for:
Spindle position display N 140 / N 141 / N 142 / N 150 / N 152 / N 153 / N 155

Baumer IVO GmbH & Co. KG
Dauchinger Strasse 58-62
DE-78056 Villingen-Schwenningen
Phone +49 (0)7720 942-0
Fax +49 (0)7720 942-900
info.de@baumerivo.com
www.baumer.com

05.11 · 174.02.047/4

Contents

Disclaimer of liability
The present manual was compiled with utmost care, errors and omissions reserved. For this reason Baumer IVO GmbH & Co. KG rejects any liability for the information compiled in the present manual. Baumer IVO nor the author will accept any liability for direct or indirect damages resulting from the use of the present information.

At any time we should be pleased receiving your comments and proposals for further improvement of the present document.

1. Introduction

1.1. Scope of delivery

Please check the delivery upon completeness prior to commissioning.
Depending on encoder configuration and part number delivery is including:

- Encoder
- CD with describing file and manual (also available as download in the Internet)

1.2. Product assignment

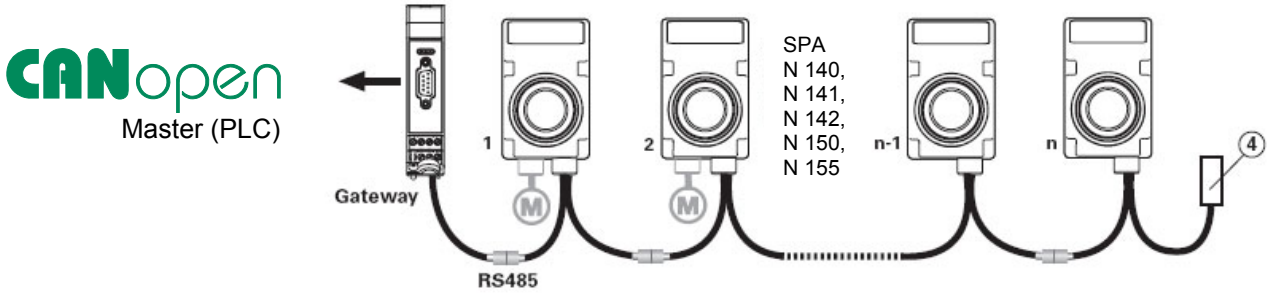
Product	Product-Code	Device Name	Eds-File	Product famiy
GK473.0075PDx	0x2D	GK473 CANopen	GK473_CANopen.eds	multicon

2. Overview

Gateway GK473 (interface converter) serves as „translator“ in data communication between the spindle position displays (SPAs) of the Baumer IVO Multicon system (for example N 140 / N 142 / N 150) and a CANopen capable PLC. The gateway converts the RS485 data blocks of the PLC into CANopen protocol (and vice-versa).

Some supplementary gateway functions considerably improve and facilitate PLC programming.

The gateway enables the communication of max. 32 spindle position displays (SPAs) with CAN bus.
The gateway is compatible to the SPAs of the Multicon series.



3. Safety and operating instructions

Supplementary information
- The present manual is intended as a supplement to already existing documentation (catalogues, product data sheets and mounting instructions).
- The manual must be studied carefully prior to initial commissioning of the equipment.

Intended purpose of the equipment
- The gateway is an interface converter applied in data communication between RS485 interface and CANopen. It comprises special modifications relating to the IVO Multicon spindle positioning system and must only be used for this purpose.

Commissioning
- The gateway must be initialised and mounted only by a qualified expert.
- Observe the operating instructions of the machine manufacturer.

Safety instructions
- Check all electrical connections prior to commissioning of the equipment.
- If mounting, electrical connections or any other work performed at the gateway and the equipment is not correctly executed this can result in malfunction or failure of the gateway.
- Corresponding safety precautions must be provided and observed to exclude any risk of personal injury, damage to material or operating equipment as a result of gateway failure or malfunction.
- The gateway must not be operated beyond the limits specified in the data sheet.

Failure to observe these safety instructions can result in malfunctions, material damage or personal injury.

Transport and storage
- Only ever transport or store the gateway in its original packaging.
- Never drop the gateway nor expose it to major shocks.

Mounting
- Avoid impacts or shocks on the housing.

Disposal
- Dispose of gateway and components in accordance with the legal regulations prevailing in the respective country.

4. Commissioning

4.1. Mechanical mounting

The gateway housing is designed for DIN rail mounting. Use a small screw driver to open the housing and pull out the gateway halfway until stop for setting of Node ID, baud rate and terminating resistor.

- Grid 22.5 mm, height approx. 100 mm and depth approx. 110 mm
- For mounting on 35 mm DIN rail

For mounting on DIN rail place the gateway rear onto the rail and press down until lock snaps into place.

4.2. Electrical connection

4.2.1. Electrical commissioning

- Do not perform any electrical modifications at the gateway.
- Do not carry out any wiring work when the gateway is live.
- Ensure that the entire equipment is installed in line with EMC requirements. Both ambient installations and wiring have an impact on the electromagnetic compatibility of the gateway. Install the gateway and supply cables separately or remote from cables with high interference emissions (frequency converters, contactors etc).
- When working with consumers with high interference emission provide separate power supply for the gateway.
- Connect gateway to protective earth (PE) conductor using shielded cables. The braided shield must be connected to the cable gland or plug. Ideally, aim at bilateral connection to protective earth (PE), i.e. the housing via the mechanical assembly and the cable shield via the downstream devices. In case of earth loop problems earth on one side only as a minimum requirement.

Failure to observe these instructions can result in malfunctions, material damage or personal injury.

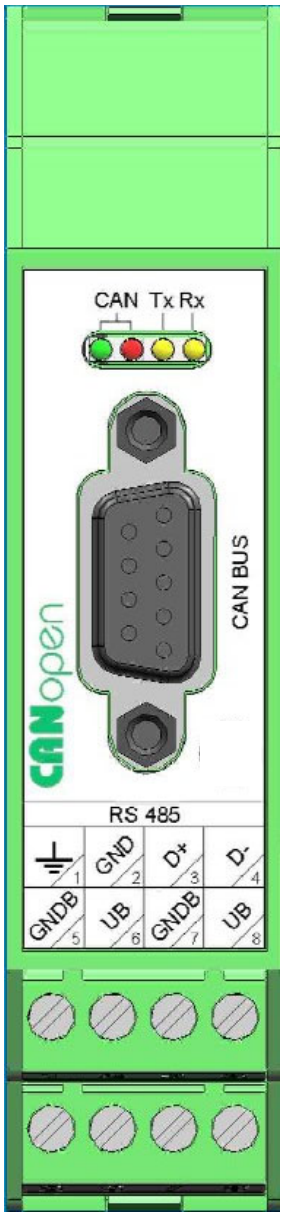
4.2.2. Supply voltage connection

Supply voltage connection of gateway and SPA is provided by screw terminals 5 (GNDB) and 6 (UB) or by screw terminals 7 (GNDB) and 8 (UB). Terminals 5 and 7 as well as 6 and 8 are internally connected providing a loop for power supply to further consumers (for example the spindle position display SPA).

4.2.3. Shield connection / functional earth

Protective earth may be assigned to terminal 1 that at the same time provides shield connection of the RS485 bus. There is a link to terminal 1 and to the housing of the 9-pin SUB-D connector (CANopen). Additional connection to protective earth is via DIN rail mounting. For this purpose the DIN rail must be connected to functional earth by low-impedance.

4.2.4. CAN-Bus - RS485 connection



CAN Bus

CAN Bus connection is provided by 9-pin D-SUB connector (female) that is integrated in the front panel.

PIN	Function
1	Shield/functional earth
2	CAN _L
3	
4	
5	
6	
7	CAN_H
8	
9	

Connector housing: Shield /functional earth

RS485 Bus

RS485 Bus connection and power supply is provided by 2 x 4-pin screw terminal connectors

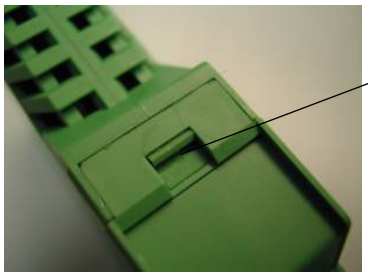
Pin	Funktion
1	Protective earth
2	Signal-GND
3	Rx/Tx +
4	Rx/Tx –
5	GNDB
6	UB
7	GNDB
8	UB

UB:12..24 VDC, ripple 5%, 40 mA

Supplementary consumption of each spindle position display (with no motive drive): approx. 30 mA.

4.3. Device parameter setting

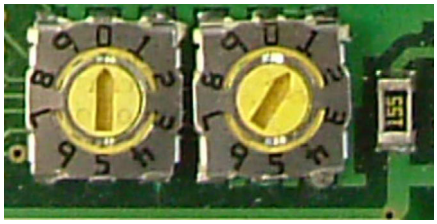
4.3.1. How to open the housing



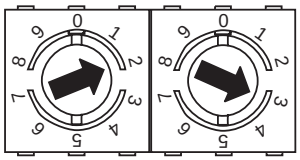
Setting of user address (Node ID) and baud rate requires partial opening of the housing by using a small screw-driver. After having pressed the snap locks on top and behind the screw terminal connector the housing can be pulled out by half until the stop for access to the operating elements. The housing is locked again by pushing it back to its original position until both snap locks click into place.

4.3.2. Setting the CANopen user address (Node ID)

The CANopen Node ID can be set by two BDC rotary switches in a decimal way within the range from 01 to 99. 00 is not permitted -> automatically Node ID 1 will be applied.
Default: Node ID 1.



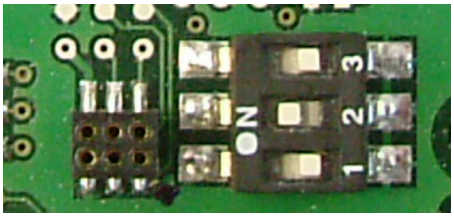
Example: 23



4.3.3. Setting the baud rate

Setting the baud rate is in binary way by poles 1 to 3 of the DIP switch.

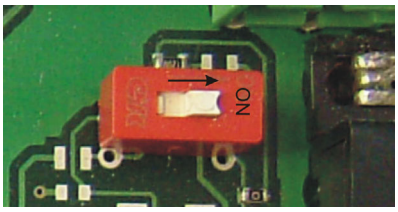
Baudrate	DIP pole position		
	1	2	3
10 kBit/s	OFF	OFF	OFF
20 kBit/s	OFF	OFF	ON
50 kBit/s *	OFF	ON	OFF
100 kBit/s	OFF	ON	ON
125 kBit/s	ON	OF	OF
250 kBit/s	ON	OFF	ON
500 kBit/s	ON	ON	OFF



* default

4.3.4. Activation of CANopen terminating resistor

If GK473 CANopen is the final device in the CAN-Bus bus line it must be terminated by a resistor. The terminating resistor is activated by setting the one-pole DIP to “ON”. Default is “OFF”.



ON = final user
OFF = user X

4.3.5. Gateway projecting in the CANopen project

For gateway projecting in CANopen project please refer to the attached EDS file.
GK473_CANopen.eds

5. CANopen interface description

5.1. General information

The gateway enables data communication between the SPA's RS485 protocol and the CAN bus protocol.

CAN bus (CAN: Controller Area Network) was originally designed by Bosch and Intel as rapid and cheap means of data transmission in automotive industry. Today, CAN bus is also applied in industrial automation.

CAN bus is a fieldbus (with standards defined by the CAN in Automation Association CiA) where devices, actuators and sensors of different manufacturers communicate with each other.

The CANopen profile based on layer 7 of the CAL (CAN-Application Layer) was developed under the technical administration of the Steinbeis Transfer Centre for Automation. Compared to CAL, CANopen provides only the functions appropriate for applications in automation. This way CANopen is a CAL excerpt optimized for use in automation. It enables improved system init as well as utilizing simplified devices. CANopen is optimized for rapid data exchange in real-time systems.

CANopen means:
• Easy access to all device and communication parameters
• Synchronisation of several devices
• Automatic network configuration
• Cyclic and event-triggered process data traffic

CANopen comprises four communication objects (COB) with different characteristics:
• Process data objects for real-time data (PDO)
• Service data objects for parameter and program transmission (SDO)
• Network management (NMT, Heartbeat)
• Pre-defined objects (for synchronisation, emergency signals)

Since there is no standard profile for this kind of gateway it is manufacturer-specific.

5.2. Data transfer structure

The PLC data message (by the CANopen Master) output to the gateway implies the complete command for the intended spindle position display. The command is completed by header and footer info as well as CRC check sum and transmitted by the gateway via RS485.

Vice-versa, header and footer info together with the CRC check sum are deleted from the RS485 message. All other contents remain unchanged and are transmitted as data block of a CANopen message available at the PLC as consistent input data.

The gateway further provides some supplementary functions to relieve the master PLC and to simplify programming operations.

Object Auto Scan SPA 0x3203
Parameter: ON
During operation the gateway is performing automatic and continuous scanning operations of all SPAs connected to verify whether they are “in position”. Upon request this information can be gathered under a special command for CANopen read-in. Thus the AutoScan function relieves the CANopen master from proceeding cyclic SPA scanning operations itself.

This operating mode is imperative for providing void data under „collective information“ objects.

However, there might be situations where this is not desired.

Several commands to the SPA are only valid for the time being until another command is received (for example indicate SPA identifier in the lower display line). The AutoScan object would make these commands ineffective. Furthermore, the scanning procedure keeps the RS485 bus busy and consequently the commands received from the CANopen master can be processed by the gateway faster with AutoScan off.

Objects collective information 0x3204, 0x3205, 0x3206
These objects enable comfortable reading of the SUB bus status.
Precondition for updated data is **Auto Scan SPA (0x3203) ON.**

Ideally, the SUB bus users (SPA) are given an identifier from 0..31.
This way one SDO only is enough for scanning the complete SUB bus status with corresponding SPA.

The data on SDO Upload subindex 2..5 of the respective collective information are in binary code in a 32 bit double word.

Subindex 2	$2^0..2^{31}$	SPA identifier 0..31
Subindex 3	$2^0..2^{31}$	SPA identifier 32..63
Subindex 4	$2^0..2^{31}$	SPA identifier 64..95
Subindex 5	$2^0..2^{31}$	SPA identifier 96..99

Object 0x3204 Collective information „Check Position“

0x3204_0	SDO Sub Index count	
0x3204_1	scan total of SPAs ‚not in position’	
0x3204_2	scan SPA with SUB bus identifier 0..31	‚not in position’
0x3204_3	scan SPA with SUB bus identifier 32..63	‚not in position’
0x3204_4	scan SPA with SUB bus identifier 64..95	‚not in position’
0x3204_5	scan SPA with SUB bus identifier 96..99	‚not in position’

0x3204_2.. 0x3204_5 The information SPA ‚not in position’ is in binary code in a 32 bit double word.

A bit entry at location 2^n is representing a SPA with identifier **n** ‚not in position’

Example: read SPA with SUB- bus identifier 0..31 ‚not in position’

Nr	Time [100µs] (rel)	ID (hex)	Name	Data (hex)	ASCII
0	03:43:22.934.7	601	SDO	40 04 32 02 00 00 00 00	@.2.....
1	00:00:00.004.3	581		43 04 32 02 3C 00 80 00	C.2.<.▮.

0x 00 80 00 3C

0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								x																		x	x	x	x		

Result: SPA with user identifiers 2, 3, 4, 5 and 23 are not in position

Example: scan SPA with SUB bus identifier 64..95 ‚not in position’

Nr	Time [100µs] (rel)	ID (hex)	Name	Data (hex)	ASCII
0	04:08:50.613.1	601	SDO	40 04 32 04 00 00 00 00	@.2.....
1	00:00:00.004.4	581		43 04 32 04 04 00 00 00	C.2.....

0x00 00 00 04

0x00

95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
																													x		

Result: SPA with user identifier 66 is ‚not in position’

Object 0x3205 Collective information „SPAs connected“

0x3205_0	SDO Sub Index count
0x3205_1	reading total of SUB bus users connected (SPA RS485 Bus)
0x3205_2	reading SPA with SUB bus identifier 0..31
0x3205_3	reading SPA with SUB bus identifier 32..63
0x3205_4	reading SPA with SUB bus identifier 64..95
0x3205_5	reading SPA with SUB bus identifier 96..99

0x3205_2.. 0x3205_5 The information ‚SPAs connected' is in binary code in a 32 bit double word.

A bit entry at location 2^n is representing a SPA connected with identifier **n**

Object 0x3206 Collective information „Device status“

0x3206_0	SDO Sub Index count	
0x3206_1	reading total of SPA's in error status ‚e'	
0x3206_2	reading SPA with SUB bus identifier 0..31	error
0x3206_3	reading SPA with SUB bus identifier 32..63	error
0x3206_4	reading SPA with SUB bus identifier 64..95	error
0x3206_5	reading SPA with SUB bus identifier 96..99	error

0x3206_2.. 0x3206_5 The information SPA in error status ‚e' is in binary code in a 32 bit double word.

A bit entry at location 2^n is representing a SPA connected in error status ‚e' with identifier **n**

5.3. Transparent gateway function

5.3.1. General information

Basically, there is a transparent transfer of all commands with code „A“...“z“ from the CANopen master to the SPA and vice-versa. Thus, the PLC is capable of addressing each SPA individually or globally by broadcast command. The gateway does not verify whether the SPA with the respective identifier is connected or not, neither is the command code.

Basically, every command received from CANopen master (PLC) is given a reply by the gateway. The CANopen master must always wait for a reply prior to sending another command to the gateway. This procedure allows the CANopen master to always verify when a command has been accomplished.

Replied is by means of

- transparent transfer of the SPA reply after having addressed directly an individual SPA and after having duly received a reply.
- abort error signal if the SPA did not answer after having elapsed the timeout (approx. 100 ms)

Any timeout required in the CANopen master for monitoring of gateway replies should not be less than 400 ms.

Reading the SUB bus (SPA)
After ‚initiate upload' the CANopen master must wait for at least 100 ms prior to start transmission of the spindle positioning data by the first SDO segment (during this time the SPA is addressed by the SUB bus).

Writing on SUB bus (SPA)
After ‚download complete' by CANopen Master the gateways confirmation is delayed (max. 100 ms, during this time the SPA is addressed by the SUB bus)

Access to RS485 user (SPA) by SDO object 0x3000 and 0x3100
Accessing an SPA either in reading or writing requires two SDO's being transmitted by CANopen.

(1) 0x3000 by SDO an identifier/command pointer is set on the addressed SPA.
(2) 0x3100 by SDO the SPA data are read or written

Note: 0x3000 has to be transmitted once only in order to get reading or writing access as many times as required to the same SPA by the same command.

Example: Check Position SPA with RS485 SUB bus identifier 2
Gateway CANopen user address 1 (Node ID)

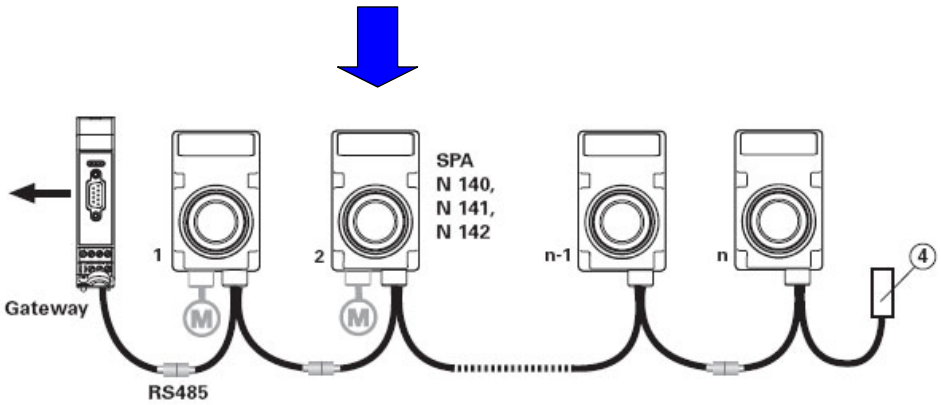
SPA identifier 2 (0x20 +2 = 0x22)
command = „C" =0x43 = check position

Object 0x3000 Setting an identifier/command pointer to SUB bus user (SPA)

Nr	Time [100µs] (abs)	ID (hex)	Name	Data (hex)	ASCII
0	00:01:05.240.4	601	SDO	23 00 30 00 22 43 00 00	#.0."C..
1	00:01:05.243.2	581		60 00 30 00 00 00 00 00	`.0.....
2	00:01:05.252.3	601	SDO	40 00 31 00 00 00 00 00	@.1.....
3	00:01:05.300.8	581		41 00 31 00 0E 00 00 00	A.1.....
4	00:01:05.451.5	601	SDO	60 00 00 00 00 00 00 00	`.....
5	00:01:05.454.1	581		00 22 43 78 31 37 00 00	."Cx17..
6	00:01:05.463.2	601	SDO	70 00 00 00 00 00 00 00	p.....
7	00:01:05.467.4	581		11 00 00 00 00 00 00 00	.....

SPA identifier 2 (0x20 +2 = 0x22)
Command = „C" =0x43 = check position
„x" =0x78 = not in position, profile 0x31,0x37 = 17

The internal identifier pointer for RS485 bus now is on SPA with identifier 2 (0x22= 0x20 + identifier)
The internal command pointer is on command ‚C' (0x43) read Check Position



Object 0x3100 SUB-Bus transparent transfer

This object enables the transparent transfer of SPA commands through the gateway.
Transfer is via CAN Bus as array of **14 bytes as Octet String** (Array of 14 Byte).
CANopen will consistently transfer this string one after the other by a so-called „segmented SDO“. In the following example there are three messages.

Example: Programming the target –12.50, profile number 17, SPA with RS485 SUB-bus identifier 2
Gateway CANopen user address (Node ID) 1

SPA identifier 2 (0x20 +2 = 0x22)
command = „S“ =0x53 = target

SDO download = write on RS485 SUB-bus (SPA)

CAN telegrams CANopen master <-> GK473 Gateway

Nr	Time [100µs] (abs)	ID (hex)	Name	Data (hex)	ASCII
0	00:21:16.027.4	601	SDO	23 00 30 00 22 53 00 00	#.0."S..
1	00:21:16.030.1	581		60 00 30 00 00 00 00 00	`.0.....
2	00:21:16.037.3	601	SDO	21 00 31 00 0E 00 00 00	!.1.....
3	00:21:16.039.6	601	SDO	00 31 37 2D 30 31 32 35	.17-0125
4	00:21:16.042.1	581		60 00 31 00 00 00 00 00	`.1.....
5	00:21:16.044.6	581		20 00 31 00 00 00 00 00	.1.....
6	00:21:16.047.1	601	SDO	11 30 00 00 00 00 00 00	.0.....
7	00:21:16.116.9	581		30 00 31 00 00 00 00 00	0.1.....

First the identifier/command pointer is set as 0x3000 on the SPA to be addressed.
Second the SPA command is transferred by 0x3100.
The data to be written (Octet String) are marked in blue.

For further details regarding data contents and format please refer to the respective manual.
More examples in the annex.

Example: Read target of SPA with RS485 SUB-bus identifier 2
Gateway CANopen user address (Node ID) 1

SPA identifier 2 (0x20 +2 = 0x22)
command = „S“ =0x53 = target

SDO upload = read from RS485 SUB-bus (SPA)

CAN telegrams CANopen Master <-> GK473 Gateway

Nr	Time [100µs] (abs)	ID (hex)	Name	Data (hex)	ASCII
0	00:02:25.642.8	601	SDO	23 00 30 00 22 53 00 00	#.0."S..
1	00:02:25.645.5	581		60 00 30 00 00 00 00 00	`.0.....
2	00:02:25.654.6	601	SDO	40 00 31 00 00 00 00 00	@.1.....
3	00:02:25.730.4	581		41 00 31 00 0E 00 00 00	A.1.....
4	00:02:25.853.8	601	SDO	60 00 00 00 00 00 00 00	`.....
5	00:02:25.856.7	581		00 22 53 31 37 2D 30 31	."S17-01
6	00:02:25.865.6	601	SDO	70 00 00 00 00 00 00 00	p.....
7	00:02:25.868.2	581		11 32 35 30 00 00 00 00	.250....

First the identifier/ command pointer 0x3000 is set on the SPA to be addressed.
After reading command 0x3100 the read SPA data (Octet String) are marked in blue.
Result: SPA with SUB-bus user identifier 2 provides profile no. 17 with target –12.50

For further details regarding data contents and format please refer to the respective manual.
More examples in the annex.

5.3.2. Broadcast commands

Due to the bus conflict on the RS485 interface the spindle position displays are generally not capable of replying to broadcast commands (commands addressed simultaneously to all SPAs, identifier 99, 83h). Instead, the gateway will reply to the CANopen master, but only confirming having transmitted the broadcast command to the RS485 interface of all SPAs. It is NOT a confirmation that the command has been accepted and accomplished by all SPAs.
More examples in the annex.

5.3.3. Interface documentation

The SPA interface is completely documented in the respective SPA interface description which is absolutely imperative for utilizing the gateway.

5.4. Automatic gateway functions

5.4.1. Automatic SPA recognition

After power on the gateway will proceed an automatic SPA identifier scan operation and will store the identifiers encountered in the internal memory. Later, the identifiers of all SPAs connected can be read in by CANopen master by help of SDO collective information.

The function „automatic SPA recognition“ takes for granted that all SPAs are being switched on previously or at the same time with the gateway. Reliable information on the SPAs connected is provided upon gateway init at the CAN bus by bootup message.
Refer also to NMT status diagram.

5.4.2. Continous scanning operations of all SPAs connected

During operation the gateway is automatically scanning all SPAs connected, checking whether they are „in position“ or giving an error signal. Upon request this information can be requested by SDO collective information (see 0x3204, 0x3205, 0x3206) or by PDO.

5.5. Transmit PDO 1..4 „check position“

Objects 0x3204_2.. 0x3204_5 are mapped to Transmit PDO's 1..4 in a static way.

0x3204_2	TXPDO1	SPA with SUB bus identifier 0..31	‚not in position'
0x3204_3	TXPDO2	SPA with SUB bus identifier 32..63	‚not in position'
0x3204_4	TXPDO3	SPA with SUB bus identifier 64..95	‚not in position'
0x3204_5	TXPDO4	SPA with SUB bus identifier 96..99	‚not in position'

The information SPA ‚not in position' is in binary code in a 32 bit double word.
(see object 0x3204)
A bit entry on location 2^n represents a SPA with identifier **n** ‚not in position'

Ideally the SUB bus users (SPA) are given an identifier between 0..31.
This way one PDO only is enough for a complete SPA status request.

Possible PDO definition as time triggered (default) or sync-triggered.

Please note that a cycle time of less than 1 second does not make sense since the SUB bus status has to be scanned first once per cycle. This will take approx. 70 ms per user.

5.6. Error signals

The messages relating to SPA communication are processed as CANopen Abort Messages.

These are subdivided into SPA messages and gateway messages.

- One SPA is giving an error signal after transparent command transfer. For details relating to the respective messages please refer to the relevant SPA interface description. Present state of the art are messages indicating “format error” and „CRC error”.

- Time out SUB-Bus A SUB-bus user did not answer within a time window of 100 ms.
  Abort Code 0x08000020

- Time Out Gateway The gateway did not reply within one second during SDO transfer.
  Abort Code 0x05040000

5.7. CANopen Object directory

Overview on gateway objects

According to CiA (CAN in Automation) the objects are subdivided into three groups:

- **Standard objects**:
  1000h, 1001h, 1018h
- **Manufacturer-specific objects**:
  2000h - 5FFFh
- **Device-specific objects**:
  All remaining objects from 1000h - 1FFFh, 6000h - FFFFh

The following chart is a summary of all SDO objects supported by the gateway.

Objekt object number in Hex
Name **---**
Format U/I = Unsigned/Integer, Figure = number Bit, ARR = Array, REC = Record
Zugriff ro = ReadOnly, wo = WriteOnly, rw = ReadWrite
Default Default at first init or restore default
Significance supplementary description

Object Sub-Index	**Name**	**Format**	**Access**	**Default**		**Significance**
1000h	Device Type	U32	ro	00000000h		Gateway not standardized and not conform to any CiA Device Profile. GK473 provides connection between CANopen Master to the RS485 devices of the IVO Multicon product family.
1001h	Error Register	U8	ro	0h		Bit0 = Generic error Bit4 = Communication error (overrun, …) Bit7 = Manufacturer-specific
1005h	Sync COB-ID	U32	rw	80h		COB-ID of Sync object
1008h	DeviceName	U32	ro	GK473		Gateway CANopen for the IVO Multicon product family
1009h	Hardware Version	U32	ro	device-specific		Product Hardware Version in ASCII
100Ah	Software Version	U32	ro	device-specific		Product Software Version in ASCII
1017h	Producer Heartbeat Time	U16	rw	0h		Producer Heartbeat time in ms
1018h	Identity Object	REC	ro			
00h	Maximum Subindex	U8	ro	4h		
01h	VendoID	U32	ro	ECh		IVO Vendor ID designated by CiA
02h	Product Code	U32	ro	2Dh		Product Code
03h	Revision Number	U32	ro	Manufacturer-specific		Product revision number
04h	Serial Number	U32	ro	0		

PDO Objects

1800h	Transmit PDO1 Parameter	REC				
00h	Maximum Subindex	U8	ro	5h		
01h	COB-ID	U32	ro	180h+id		PDO ID = 180h + Node-ID
02h	PDO Type	U8	rw	FEh		Asynchronous cyclic, time-triggered
05h	EventTimer	U16	rw	3E8h		Cycle time in ms
1801h	Transmit PDO2 Parameter	REC				
00h	Maximum Subindex	U8	ro	5h		
01h	COB-ID	U32	ro	280h+id		PDO ID = 280h + Node-ID
02h	PDO Type	U8	rw	FEh		Asynchronous cyclic, time-triggered
05h	EventTimer	U16	rw	3E8h		Cycle time in ms
1802h	Transmit PDO3 Parameter	REC				
00h	Maximum Subindex	U8	ro	5h		
01h	COB-ID	U32	ro	380h+id		PDO ID = 380h + Node-ID
02h	PDO Type	U8	rw	FEh		Asynchronous cyclic, time-triggered
05h	EventTimer	U16	rw	3E8h		Cycle time in ms
1804h	Transmit PDO4 Parameter	REC				
00h	Maximum Subindex	U8	ro	5h		
01h	COB-ID	U32	ro	480h+id		PDO ID = 480h + Node-ID
02h	PDO Type	U8	rw	FEh		Asynchronous cyclic, time-triggered
05h	EventTimer	U16	rw	3E8h		Cycle time in ms
1A00h	Transmit PDO1 Mapping	ARR				
00h	Maximum Subindex	U8	ro	1h		
01h	PDO1 Contents	U32	ro	32040220h		SPA with SUB-bus identifier 0..31 ‚not in position'
1A01h	Transmit PDO2 Mapping	ARR				
00h	Maximum Subindex	U8	ro	1h		
01h	PDO2 contents	U32	ro	32040320h		SPA with SUB-bus identifier 32..63 ‚not in position'
1A02h	Transmit PDO3 Mapping	ARR				
00h	Maximum Subindex	U8	ro	1h		
01h	PDO3 Contents	U32	ro	32040420h		SPA with SUB-bus identifier 64..95 ‚not in position'
1A03h	Transmit PDO4 Mapping	ARR				
00h	Maximum Subindex	U8	ro	1h		
01h	PDO4 Contents	U32	ro	32040520h		SPA with SUB-bus identifier 96..99 ‚not in position'

Manufacturer-specific objects (GK437 objects)

3000h	Identifier/command pointer on SUB-bus	U32	rw			Pointing on a SUB-bus user, Pointing on a SUB-bus identifier Pointing on a SPA command
3100h	SUB-Bus Transparent transfer	Octet String 14 Byte	rw			Read/write SPA command Array of [14] Bytes
3203h	Auto Scan SPA	U08	rw	1h=on		1 = Automatic scan of SPA On 0 = Automatic scan of SPA Off
3204h	Check Position	REC				
00h	Maximum Subindex	U8	ro			
01h	Total of SPAs ‚not in position'	U8	ro			Total of all RS485 SUB-bus user recognised being ‚not in position'
02h	SPA SUB-bus identifier 0..31	U8	ro			SPA with SUB-bus identifier 0..31 ‚not in position'
03h	SPA SUB-bus identifier 32..63	U32	ro			SPA with SUB-bus identifier 32..63 ‚not in position'
04h	SPA SUB-bus identifier 64..95	U32	ro			SPA with SUB-bus identifier 64..95 ‚not in position'
05h	SPA SUB-bus identifier 96..99	U32	ro			SPA with SUB-bus identifier 96..99 ‚not in position'
3205h	SPAs connected	REC				
00h	Maximum Subindex	U8	ro			
01h	Total of all SPAs connected	U8	ro			Total of all RS485 SUB-bus users recognised
02h	SPA SUB-bus identifier 0..31	U32	ro			SPA with SUB-bus identifier 0..31
03h	SPA SUB-bus identifier 32..63	U32	ro			SPA with SUB-bus identifier 32..63
04h	SPA SUB-bus identifier 64..95	U32	ro			SPA with SUB-bus identifier 64..95
05h	SPA SUB-bus identifier 96..99	U32	ro			SPA with SUB-bus identifier 96..99
3206h	SPA with error signal	REC				
00h	Maximum Subindex	U8	ro			
01h	Total of all SPAs connected in ‚e' error status	U8	ro			Total of all RS485 SUB-bus users recognised being in error status ‚e'
02h	SPA SUB-bus identifier 0..31	U32	ro			SPA with SUB-bus identifier 0..31 being in error status ‚e'
03h	SPA SUB-bus identifier 32..63	U32	ro			SPA with SUB-bus identifier 32..63 being in error status ‚e'
04h	SPA SUB-bus identifier 64..95	U32	ro			SPA with SUB-bus identifier 64..95 being in error status ‚e'
05h	SPA SUB-bus identifier 96..99	U32	ro			SPA with SUB-bus identifier 96..99 being in error status ‚e'

5.8. Front face indicators (LEDs)

The gateway face provides four LEDs indicating the RS485 and CANopen operating status.

RS485 TX (yellow)
Data token transmitted on RS485 interface

RS485 RX (yellow)
Data token received on RS485 interface

CANopen status LED (green)

CAN RUN LED (green)	Status	Significance
Off	INIT	Device is switched off or in INIT status
Flashing	PREOPERATIONAL	Device status PREOPERATIONAL
Single flash	STOPPED	Device status STOPPED
On	OPERATIONAL	Device status OPERATIONAL

CANopen Error LED (red)

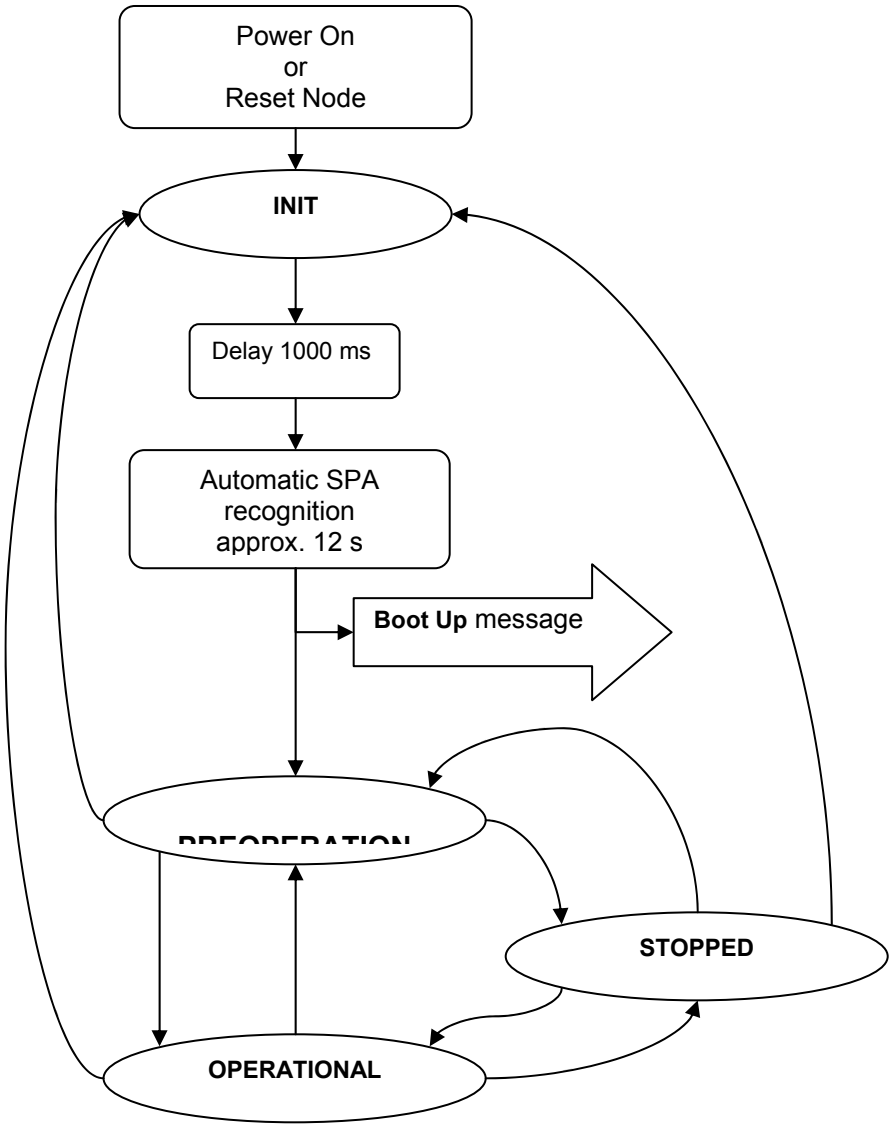
CAN Error LED (red)	Status	Significance
Off	No error	Device in due operation
Flashing	Error in SPA Auto Scan	After switch on while proceeding the SPA auto scan operation. During SUB-bus auto scan the gateway did not encounter some previously recognised users.
On	Bus Off	CAN Controller is in status bus off

The different LED status are in line with DR 303-3 V1.2 CANopen indicator specification

LED on	LED statically on
LED off	LED permanently off
LED flashing	200 ms on, 200 ms off
LED single flash	short on interval (approx. 200 ms) followed by extended off interval (approx. 1000 ms)

CANopen NMT status diagram
Upon gateway switch-on there will be a pause of 1000 ms (SPA init).
Second the gateway will proceed an automatic SPA identifier scan operation and will store all encountered identifiers in the internal memory. The storing operation may take up to 12 seconds.
Later the gateway will give a boot up message to CAN bus and is in operational status.

This status enables reading and writing of SDO parameters. For PDO parameter request the gateway has to be put in operational status first.



6. Technical data

Technical data – electrical ratings

Supply voltage	12...24 VDC
Current consumption	≤200 mA
Display	4 LED´s as operating status indicators
LED operating status	DP (green) fieldbus operating status
	DP (red) operating and error status
	RX / TX (yellow) receiving / transmitting a token (RS485)
Interfaces	CANopen, DeviceNet, Profibus-DPV0
Interface to SPA	EIA RS485 (ASCII protocol)
Data memory	>10 years (flash / EEPROM)
Programmable	Auto-Check functions
parameters	Auto Scan SPA
	Auto Message SPA Position
	Auto Message SPA Error
Emitted interference	DIN EN 61000-6-4
Interference immunity	DIN EN 61000-6-2

Technical data – mechanical design

Operating temperature	-10...+50 °C
Storing temperature	-20...+70 °C
Relative humidity	80 % non-condensing
Protection DIN EN 60529	IP 20
E-connection	D-SUB-female connector, 9-pins
	Plug-in screw terminals
Housing type	DIN rail housing EN 50022
Dimensions W x H x L	22.6 x 99 x 114.5 mm
Mounting	DIN rail housing EN 50022
Weight approx.	120 g
Material	Polyamide green, UL 94V-0

7. Annex

Transfer example for SPA commands via gateway

Gateway Node ID 0x01; CAN Identifier SDO Request =0x600+1=0x601

Generated for IXXAT CANanalyser Sequenzer Client (CAN Layer 2)

```
void R_read (void)                  // "R"    read current value
{
//Step 1: Identifier/command pointer on SUB-bus user
td 1 10          0x601   0x23 0x00 0x30 0x00 0x23 0x52 0x00 0x00              ;0x3000
//                                                |     |      |__ data
//                                                |     |_______ command 'R'
//                                                |_____________ SUB-BUS identifier
//                                                               identifier 03 - > 03 + 0x20 = 0x23
//
//Step 2: SUB-bus transparent transfer
// segmented SDO upload, after Init wait for SUB-bus user response
td 1 200         0x601   0x40 0x00 0x31 0x00 0x0E 0x00 0x00 0x00     ;0x3100
td 1 10          0x601   0x60 0x00 0x00 0x00 0x00 0x00 0x00 0x00     ;
td 1 10          0x601   0x70 0x00 0x00 0x00 0x00 0x00 0x00 0x00     ;
}

void S_write (void)                 // "S"    write target
{
//Step 1:  Identifier/Command pointer on SUB-bus user
td 1 10          0x601   0x23 0x00 0x30 0x00 0x23 0x53 0x00 0x00              ;0x3000
//                                                |     |      |__ data
//                                                |     |_______ command 'S'
//                                                |_____________ SUB-BUS identifier
//                                                               identifier 03 - > 03 + 0x20 = 0x23
//
//Step 2: SUB-Bus transparent transfer
//Example: profile number =17; target -12,50
//  Segmented Download
td 1 1           0x601   0x21 0x00 0x31 0x00 0x0E 0x00 0x00 0x00     ;0x3100
td 1 1           0x601   0x00 0x31 0x37 0x2D 0x30 0x31 0x32 0x35     ;
td 1 1           0x601   0x11 0x30 0x00 0x00 0x00 0x00 0x00 0x00     ;
}
```

```
void A_read_Broadcast (void)   // "A"   Read SUB-b device identifier
{
// First the auto scan function must be „OFF“
td 1 10          0x601   0x2F 0x03 0x32 0x00 0x00 0x00 0x00 0x00      ;0x3203

//Step 1: Address/command pointer on SUB-bus user
td 1 10          0x601   0x23 0x00 0x30 0x00 0x83 0x41 0x00 0x00              ;0x3000
//                                                |     |      |__ data
//                                                |     |_______ command 'A'
//                                                |_____________ SUB-BUS identifier

//                                                              identifier 99 - > 0x63 +0x20 = 0x83
//                                                              = Broadcast
//Step 2: SUB-Bus transparent transfer
// segmented SDO upload, after Init wait for SUB-bus user response
td 1 200         0x601   0x40 0x00 0x31 0x00 0x0E 0x00 0x00 0x00      ;0x3100
td 1 10          0x601   0x60 0x00 0x00 0x00 0x00 0x00 0x00 0x00      ;
td 1 10          0x601   0x70 0x00 0x00 0x00 0x00 0x00 0x00 0x00      ;

break    Each SPA is now showing its identifier, continue with OK

// auto scan on
td 1 10          0x601   0x2F 0x03 0x32 0x00 0x01 0x00 0x00 0x00      ;0x3203
}


void A_write_Broadcast (void)   // "A"   Write SUB-bus device identifier
{
// First the auto scan function must be „OFF”
td 1 10          0x601   0x2F 0x03 0x32 0x00 0x00 0x00 0x00 0x00      ;0x3203

//Step 1: Identifier/command pointer on SUB-bus user
td 1 10          0x601   0x23 0x00 0x30 0x00 0x83 0x41 0x00 0x00              ;0x3000
//                                                |     |      |__ data
//                                                |     |_______ command 'A'
//                                                |_____________ SUB-BUS identifier
//                                                              identifier 99 - > 0x63 +0x20 = 0x83
//                                                              = Broadcast
//
//Step 2: SUB-bus transparent transfer
// Example: New SPA identifier to be placed is 22
td 1 1           0x601   0x21 0x00 0x31 0x00 0x0E 0x00 0x00 0x00      ;0x3100
td 1 1           0x601   0x00 0x32 0x32 0x00 0x00 0x00 0x00 0x00      ;here 32 32 = identifier 22
td 1 1           0x601   0x11 0x00 0x00 0x00 0x00 0x00 0x00 0x00      ;

break Enter at SPA by turning the shaft or by soft key, continue with OK

// Auto Scan on
td 1 10          0x601   0x2F 0x03 0x32 0x00 0x01 0x00 0x00 0x00      ;0x3203
}
```