ICA 011

S-DIAS Interface Module
CAN

Instruction Manual

Date of creation: 26.05.2014 | Version date: 26.07.2023 | Article number: 20-102-011-E

Publisher: SIGMATEK GmbH & Co KG
A-5112 Lamprechtshausen
Tel.: +43/6274/4321
Fax: +43/6274/4321-18
Email: office@sigmatek.at
WWW.SIGMATEK-AUTOMATION.COM

Translation of the Original Instructions

S-DIAS Interface Module ICA 011

with 1 CAN bus

1 termination circuit switchable

The S-DIAS ICA 011 interface module has a CAN interface.
The internal CAN termination resistor can be deactivated at the connector via software or wire jumper.

Contents

1 Introduction

1.1 Target Group/Purpose of this Operating Manual

This operating manual contains all information required for the operation of the product.

This operating manual is intended for:

- Project planners
- Technicians
- Commissioning engineers
- Machine operators
- Maintenance/test technicians

General knowledge of automation technology is required.

Further help and training information, as well as the appropriate accessories can be found on our website www.sigmatek-automation.com.

Our support team is happily available to answer your questions.
Please see our website for our hotline number and business hours.

1.2 Important Reference Documentation

This and additional documents can be downloaded from our website or obtained through support.

1.3 Contents of Delivery

1x ICA 011

2 Basic Safety Directives

2.1 Symbols Used

The following symbols are used in the operator documentation for warning and danger messages, as well as informational notes:

DANGER

Danger indicates that death or serious injury **will occur**, if the specified measures are not taken.

⇨ To avoid death or serious injuries, observe all guidelines.

Danger indique une situation dangereuse qui, faute de prendre les mesures adéquates, **entraînera** des blessures graves, voire mortelles.

⇨ Respectez toutes les consignes pour éviter des blessures graves, voire mortelles.

WARNING

Warning indicates that death or serious injury **can** occur, if the specified measures are not taken.

⇨ To avoid death or serious injuries, observe all guidelines.

Avertissement d'une situation dangereuse qui, faute de prendre les mesures adéquates, **entraînera** des blessures graves, voire mortelles.

⇨ Respectez toutes les consignes pour éviter des blessures graves, voire mortelles.

CAUTION

Caution indicates that moderate to slight injury **can** occur, if the specified measures are not taken.

⇨ To avoid moderate to slight injuries, observe all guidelines.

Attention indique une situation dangereuse qui, faute de prendre les mesures adéquates, **peut** entraîner des blessures assez graves ou légères.

⇨ Respectez toutes les consignes pour éviter des blessures graves, voire mortelles.

INFORMATION

Information

⇨ Provides important information on the product, handling or relevant sections of the documentation, which require attention.

2.2 Disclaimer

INFORMATION

The contents of this operating manual were prepared with the greatest care. However, deviations cannot be ruled out. This operating manual is regularly checked and required corrections are included in the subsequent versions. The machine manufacturer is responsible for the proper assembly, as well as device configuration. The machine operator is responsible for safe handling, as well as proper operation.

The current operating manual can be found on our website. If necessary, contact our support.

Subject to technical changes, which improve the performance of the devices. The following operating manual is purely a product description. It does not guarantee properties under the warranty.

Please thoroughly read the corresponding documents and this operating manual before handling a product.

SIGMATEK GmbH & Co KG is not liable for damages caused through, non-compliance with these instructions or applicable regulations.

2.3 General Safety Directives

The Safety Directives in the other sections of this operating manual must be observed. These instructions are visually emphasized by symbols.

INFORMATION

According to EU Directives, the operating manual is a component of a product.

This operating manual must therefore be accessible in the vicinity of the machine since it contains important instructions.

This operating manual should be included in the sale, rental or transfer of the product, or its online availability indicated.

Regarding the requirements for Safety and health connected to the use of machines, the manufacturer must perform a risk assessment in accordance with machine directives 2006/42/EG before introducing a machine to the market.

Operate the unit with devices and accessories approved by SIGMATEK only.

CAUTION

Handle the device with care and do not drop or let fall.

Prevent foreign bodies and fluids from entering the device.

The device must not be opened!

Manipulez l’appareil avec précaution et ne le laissez pas tomber.

Empêchez les corps étrangers et les liquides de pénétrer dans l’appareil.

L’appareil ne doit pas être ouvert!

If the device does not function as intended or has damage that could pose a danger, it must be replaced!

En cas de fonctionnement non conforme ou de dommages pouvant entraîner des risques, l'appareil doit être remplacé!

The module complies with EN 61131-2.

In combination with a facility, the system integrator must comply with EN 60204-1 standards.

For your own safety and that of others, compliance with the environmental conditions is essential.

Le module est conforme à la norme EN 61131-2.

En combinaison avec une équipement, l’intégrateur de système doit respecter la norme EN 60204-1.

Pour votre propre sécurité et celle des autres, le respect des conditions environnementales est essential.

2.4 Software/Training

The application is created with the software LASAL CLASS 2 and LASAL SCREEN Editor.

Training for the LASAL development environment, with which the product can be configured, is provided. Information on our training schedule can be found on our website.

3 Standards and Directives

3.1 Directives

The product was constructed in compliance with the following European Union directives and tested for conformity.

3.1.1 EU Conformity Declaration

EU Declaration of Conformity

The product ICA 011 conforms to the following European directives:

- **2014/35/EU** Low-voltage Directive
- **2014/30/EU** Electromagnetic Compatibility (EMC Directive)
- **2011/65/EU** “Restricted use of certain hazardous substances in electrical and electronic equipment” (RoHS Directive)

The EU Conformity Declarations are provided on the SIGMATEK website. See Products/Downloads or use the search function and the keyword “EU Declaration of Conformity”.

4 Type Plate

Exemplary nameplate (symbol image)

HW: Hardware version
SW: Software version

5 Technical Data

5.1 Performance Data

Interfaces	1x CAN 1x Termination connection		
Adjustable data transfer rates	CAN	20,000 Baud, 50,000 Baud, 100,000 Baud, 125,000 Baud, 250,000 Baud, 500,000 Baud, 615,000 Baud, 1,000.000 Baud	
Over voltage protection	CAN	Pin CAN H	±30 V
		Pin CAN L	±30 V
	Termination	Pin TERM+	+30 V -0 V
		Pin TERM-	0
Maximum connectible CAN participants	100		
Short-circuit proof	yes		
Status LEDs	yes		

5.2 Electrical Requirements

Power supply +24 V	+18-30 V DC	
Voltage supply from the S-DIAS bus	+5 V	
Current consumption on the S-DIAS bus (+5 V supply)	typically 60 mA	maximum 70 mA
Voltage supply from the S-DIAS bus	+24 V	
Current consumption on the S-DIAS bus (+24 V supply)	typically 20 mA	maximum 40 mA

INFORMATION

If this S-DIAS module is connected to an S-DIAS supply module with several S-DIAS modules, the total current of the modules used must be determined and checked.

The total current of the +24 V S-DIAS supply cannot exceed 1.6 A!

The total current of the +5 V S-DIAS supply cannot exceed 1.6 A!

The specification for the current can be found in the module-specific documentation under "Electrical Requirements".

UL Certification

INFORMATION

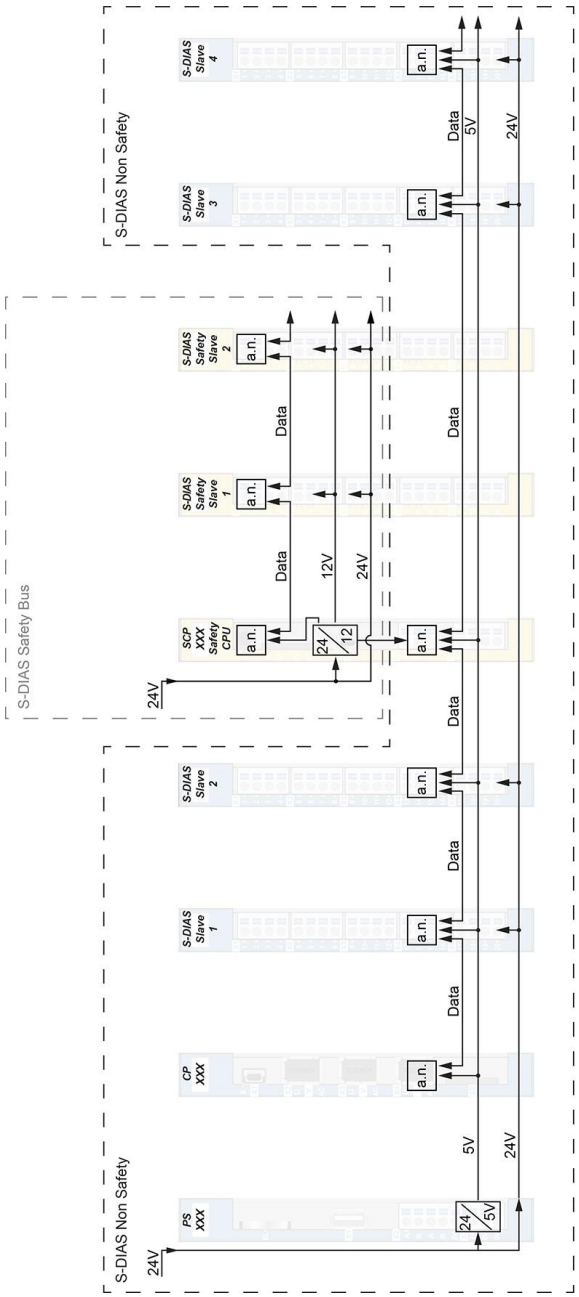
For USA and Canada:

The supply must be limited to:

a) max. 5 A at voltages from 0-20 V DC, or

b) 100 W at voltages from 20-60 V DC

The limiting component (e.g. transformer, power supply or fuse) must be certified by an NRTL (Nationally Recognized Testing Laboratory).


S-DIAS Non Safety
S-DIAS Safety Bus
S-DIAS Non Safety
24V
PS XXX
CP XXX
S-DIAS Slave 1
S-DIAS Slave 2
SCP XXX Safety CPU
S-DIAS Safety Slave 1
S-DIAS Safety Slave 2
S-DIAS Slave 3
S-DIAS Slave 4
a.n.
Data
5V
12V
24V
24/5V
24/12
a.n. = active node
Wiring S-DIAS Safety in S-DIAS System
- each S-DIAS module is an active module (active node)
- Safety CPU is connected to the S-DIAS bus (incl. +5 V supply)
- Safety bus is independent and separated from the S-DIAS bus

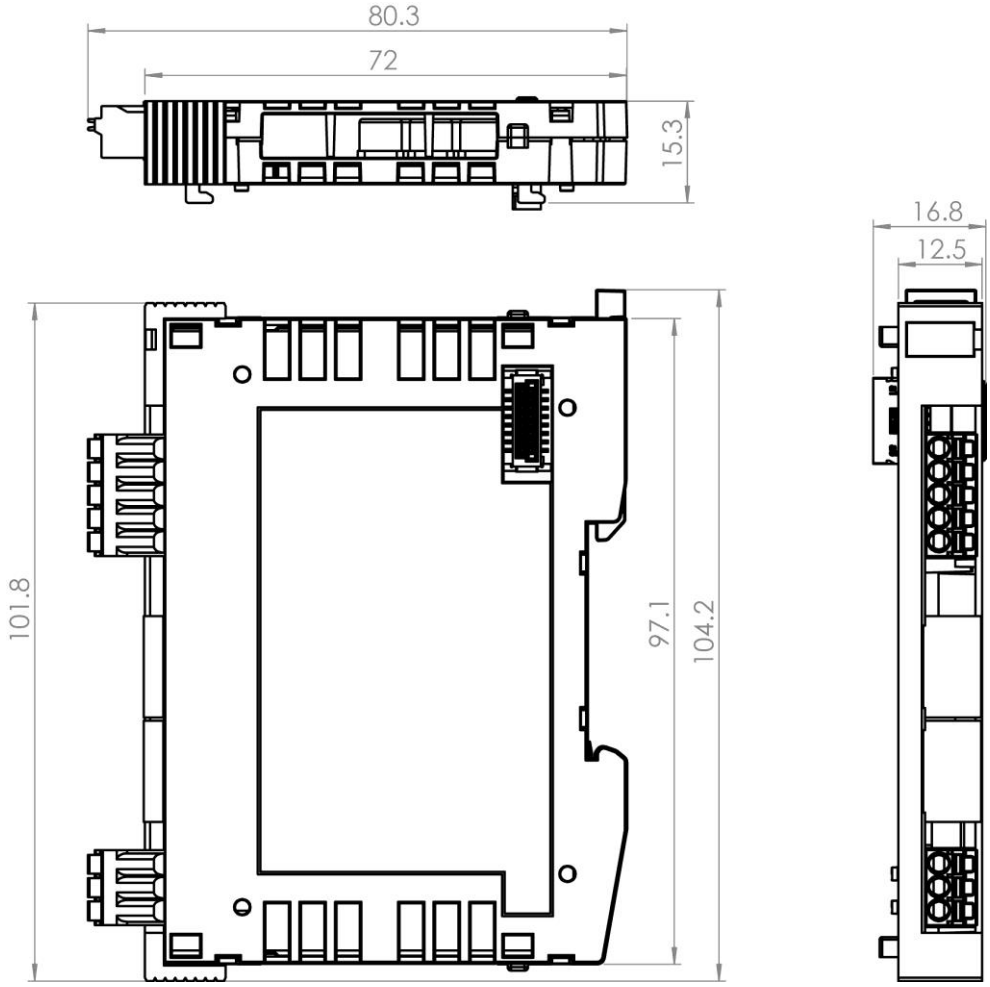
5.3 Miscellaneous

Article number	20-102-011
Standard	UL 508 (E247993)
Approbations	UL, cUL, CE, UKCA

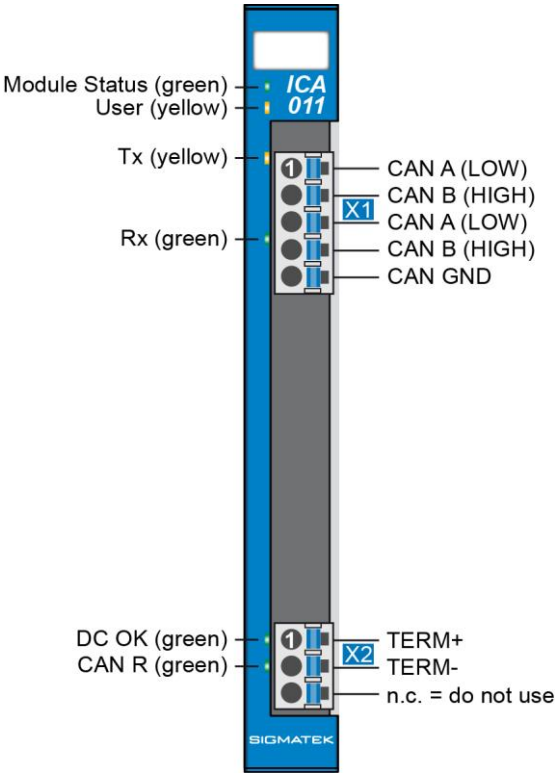
5.4 Environmental Conditions

Storage temperature	-20 ... +85 °C	
Environmental temperature	0 ... +55 °C	
Humidity	0-95 %, non-condensing	
Installation altitude above sea level	0-2000 m without derating > 2000 m up to a maximum of 5000 m with derating of the maximum environmental temperature by 0.5 °C per 100 m	
Operating conditions	pollution degree 2	
EMC resistance	in accordance with EN 61000-6-2 (industrial area)	
EMC noise generation	in accordance with EN 61000-6-4 (industrial area)	
Vibration resistance	EN 60068-2-6	3.5 mm from 5-8.4 Hz 1 g from 8.4-150 Hz
Shock resistance	EN 60068-2-27	15 g
Protection type	EN 60529	IP20

6 Mechanical Dimensions



7 Connector Layout



7.1 Status LEDs

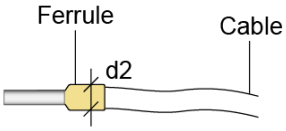
Module Status	green	ON	module active
		OFF	no supply available
		BLINKING (5 Hz)	no communication
User	yellow	ON	can be set from the application (e.g. the module LED can be set to blinking through the visualization so that the module is easily found in the control cabinet)
		OFF	
		BLINKING (2 Hz)	
		BLINKING (4 Hz)	
CAN Tx	yellow	BLINKS	sending data
CAN Rx	green	BLINKS	receiving data
DC OK	green	ON	module is supplied with +24 V
CAN R	green	ON	internal terminator activated
		OFF	Internal terminator deactivated

7.2 Applicable Connectors

Connectors:
X1-X2: Connectors with spring terminals (included in delivery)
The spring terminals are suitable connecting ultrasonically compacted (ultrasonically welded) strands.

Connections:

Stripping length/Sleeve length:	10 mm
Plug-in direction:	parallel to conductor axis or to PCB
Conductor cross section, rigid:	0.2-1.5 mm^2
Conductor cross section, flexible:	0.2-1.5 mm^2
Conductor cross section, ultrasonically compacted:	0.2-1.5 mm^2
Conductor cross section AWG/kcmil:	24-16
Conductor cross section flexible, with ferrule without plastic sleeve:	0.25-1.5 mm^2
Conductor cross section flexible, with ferrule with plastic sleeve:	0.25-0.75 mm^2 (ground for reducing d2 of the ferrule)

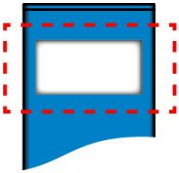


d2 = max. 2.8 mm

INFORMATION

The S-DIAS module CANNOT be connected or disconnected while voltage is applied!

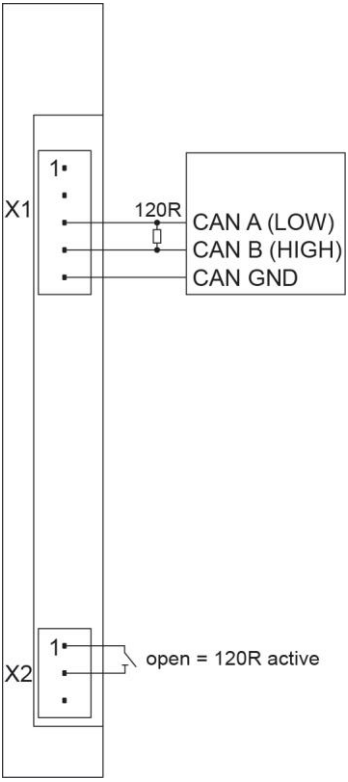
7.3 Label Field



Manufacturer	Weidmüller
Type	MF 10/5 CABUR MC NE WS
Weidmüller article number	1854510000
Compatible printer	Weidmüller
Type	Printjet Advanced 230V
Weidmüller article number	1324380000

8 Wiring

8.1 Example Connection



INFORMATION

To use the CAN termination resistor, X2 must be open and the CanTermination client of the ICA011 HW class set to 1.

9 CAN Bus Setup

This section explains how to correctly configure the CAN bus. The following parameters must first be set: Station number and data transfer rate.

9.1 CAN Bus Station Number

Each CAN bus station is assigned its own station number. With this station number, data can be exchanged with other stations connected to the bus. In a CAN bus system however, each station number can only be assigned once!

9.2 Number of CAN Bus Participants

The maximum number of participants on the CAN bus depends on the cable length, termination resistance, data transfer rate and the drivers used in the participants.

With a termination resistance of 120 Ω, at least 100 participants are possible.

9.3 CAN Bus Data Transfer Rate

Various data transfer rates (baud rates) can be set on the CAN bus. The longer the bus line is, the lower the data transfer rate that must be selected.

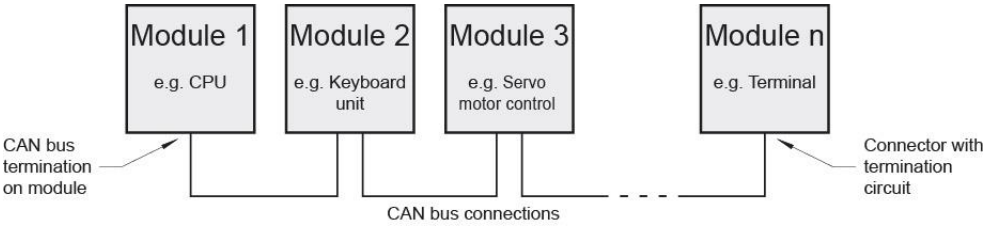
Value	Baud Rate	Maximum Length
0	615 kbits / s	60 m
1	500 kbits / s	80 m
2	250 kbits / s	160 m
3	125 kbits / s	320 m
4	100 kbits / s	400 m
5	50 kbits / s	800 m
6	20 kbits / s	1200 m
7	1 Mbit / s	30 m

These values apply to the following cable: 120 Ω Twisted Pair.

Note: For the CAN bus protocol: 1 kbit/ s = 1 kBaud.

9.4 CAN Bus Termination

In a CAN bus system, both end modules must be terminated. This is necessary to avoid transmission errors caused by reflections in the line.



9.5 Connection to the ICA 011

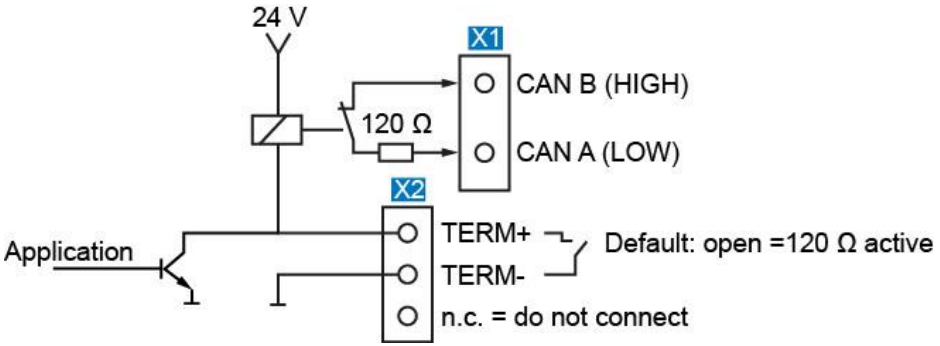
A terminator is located in the ICA 011, which is activated when connector X2 is open (=> no jumper from X2 TERM+ to TERM-).

This means that if the ICA 011 is connected at the end of the CAN bus, no external 120 Ω terminator on X1 is required.

On X2, the internal terminator can be deactivated with a jumper wire or switch (TERM+/ TERM-).

If the jumper wire is set between TERM+ and TERM-, the internal terminator is disabled (open / deactivated).

If no jumper wire is set, the internal 120 Ω CAN bus terminator is activated (located between CAN A (LOW) and CAN B (HIGH)).



10 Wiring

- The 120 Ω terminating resistors must be placed at each bus end, whereby the ICA 011 already has a built-in terminator (with TERM +/- for deactivation).
- Star wiring must be avoided

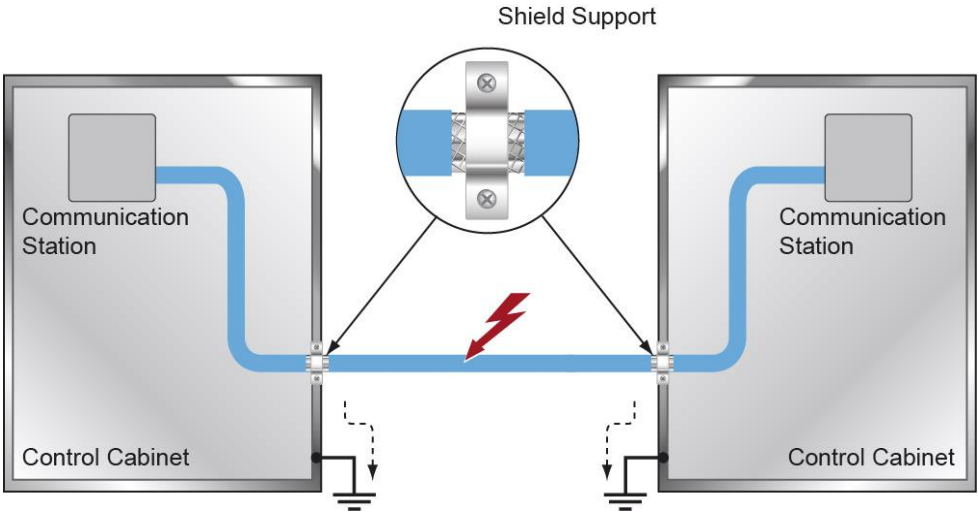
10.1 Shielding

The CAN bus wiring must be shielded.

The low-ohm shielding is either connected at the entry to the control cabinet or directly before the ICA 011 over a large, low-ohm surface (cable grommets, grounding clamps)!

Noise signals can therefore be prohibited from reaching the electronics and affecting the function.

To avoid compensating currents from the PE, which flow over the shielding the conductors, it is recommended that the system components have low Ohm and low impedance connections to one another.



11 Assembly/Installation

11.1 Check Contents of Delivery

Ensure that the contents of the delivery are complete and intact. See chapter 1.3 Contents of Delivery.

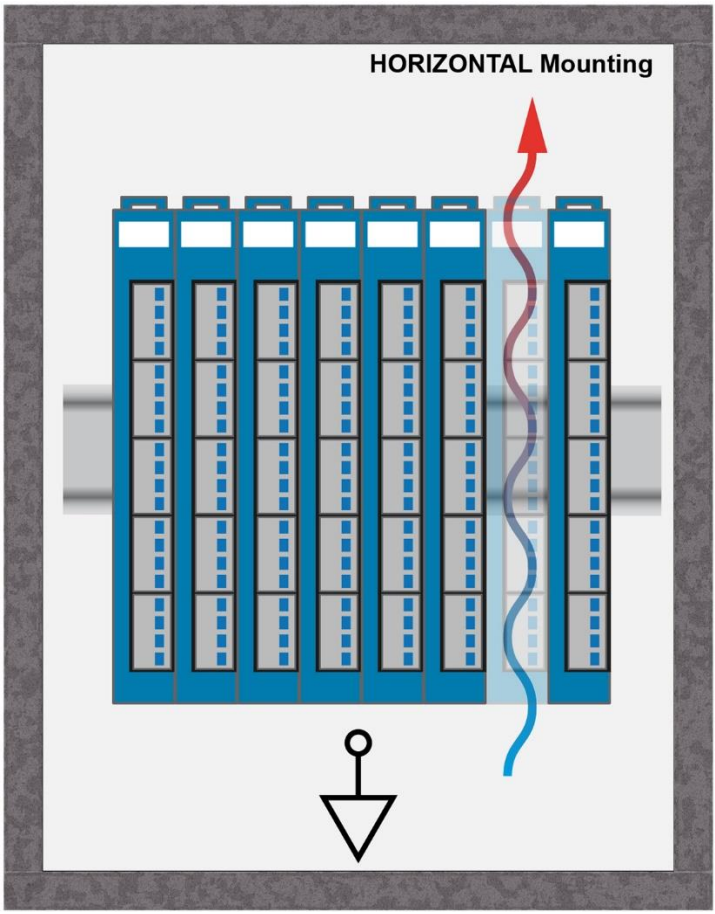
INFORMATION

On receipt and before initial use, check the device for damage. If the device is damaged, contact our customer service and do not install the device in your system.

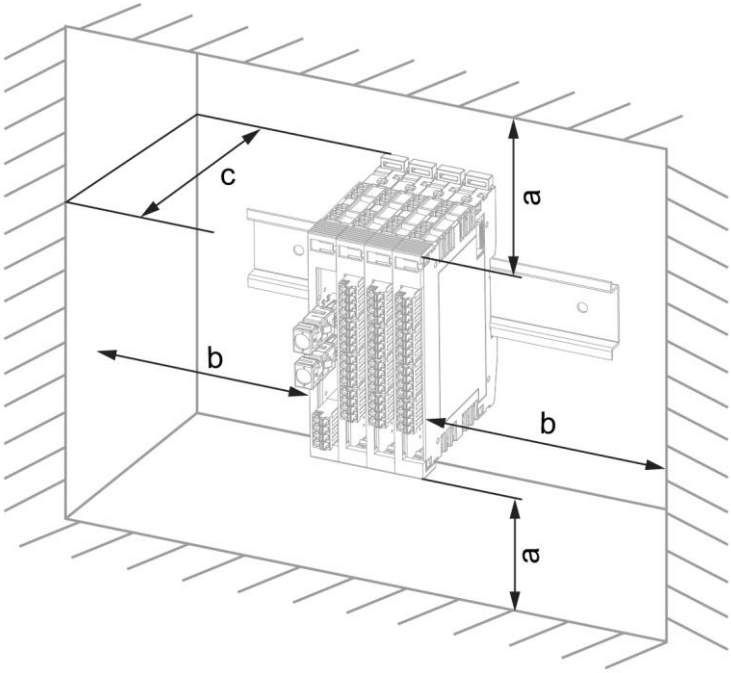
Damaged components can disrupt or damage the system.

11.2 Mounting

The S-DIAS modules are designed for installation into the control cabinet. To mount the modules a DIN-rail is required. The DIN rail must establish a conductive connection with the back wall of the control cabinet. The individual S-DIAS modules are mounted on the DIN rail as a block and secured with latches. The functional ground connection from the module to the DIN rail is made via the grounding clamp on the back of the S-DIAS modules. The modules must be mounted horizontally (module label up) with sufficient clearance between the ventilation slots of the S-DIAS module blocks and nearby components and/or the control cabinet wall. This is necessary for optimal cooling and air circulation, so that proper function up to the maximum operating temperature is ensured.



Recommended minimum distances of the S-DIAS modules to the surrounding components or control cabinet wall:



a	b	c
30 mm (1.18″)	**30 mm** (1.18″)	**100 mm** (3.94″)

a, b, c … distances in mm (inches)

12 Supported Cycle Times

12.1 Cycle Times below 1 ms (in µs)

50	100	125	200	250	500
					x

x= supported

12.2 Cycle Times equal to or higher than 1 ms (in ms)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

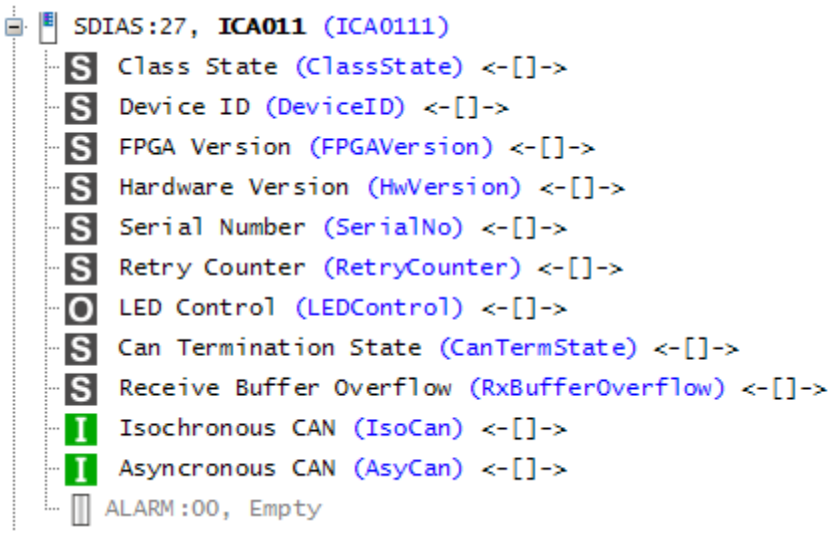
x= supported

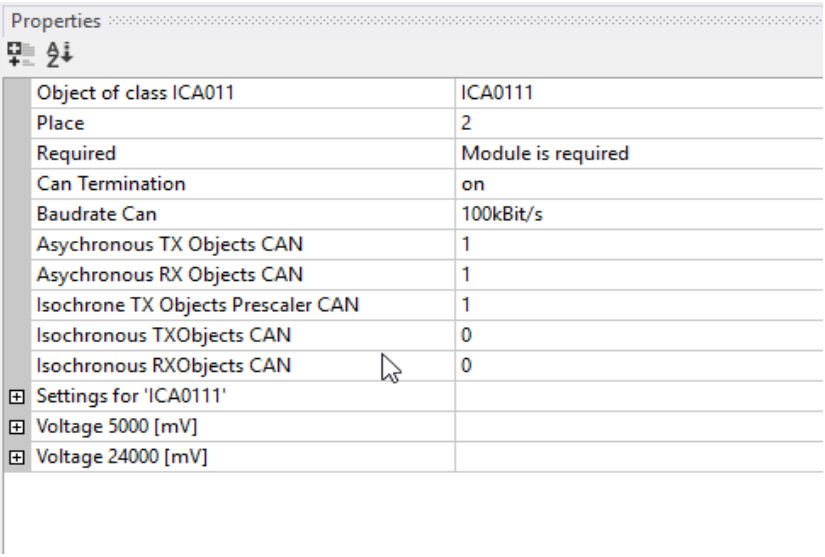
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x= supported

13 Hardware Class ICA011

Hardware Class ICA011 for the S-DIAS ICA011 CAN module





Object of class ICA011	ICA0111
Place	2
Required	Module is required
Can Termination	on
Baudrate Can	100kBit/s
Asychronous TX Objects CAN	1
Asychronous RX Objects CAN	1
Isochrone TX Objects Prescaler CAN	1
Isochronous TXObjects CAN	0
Isochronous RXObjects CAN	0
⊞ Settings for 'ICA0111'	
⊞ Voltage 5000 [mV]	
⊞ Voltage 24000 [mV]	

This hardware class is used to control the ICA 011 hardware module with a CAN interface. More information on the hardware can be found in the module documentation.

13.1 General Information

Class State	State	Shows the actual status of the hardware class.
Device ID	State	Shows the Device ID of the hardware module
FPGA version	State	FPGA version of the module in 16#XY (e.g. 16#10 = version 1.0).
Hardware Version	State	Hardware version of the module in format 16#XXYY (e.g. 16#0120 = Version 1.20).
Serial Number	State	Shows the serial number of the hardware module.
Retry Counter	State	Increments when a transfer fails on the SDIAS bus.
LED Control	Output	Used to control the application LED of the S-DIAS module, in order to find the module in the network more quickly. The following statuses are possible: 0 LED off 1 LED on 2 blinks slowly 3 blinks rapidly
Required	Property	This setting is active by default, which means that the S-DIAS hardware module at this position is mandatory for the system and can under no circumstances be disconnected or return an error. Otherwise, the entire hardware deactivated. If the hardware module is missing or removed, an S-DIAS error is triggered. If his client is initialized with 0, the hardware module located in this position is not mandatory. This means that it can be inserted or removed at any time. However, which components identified as "not required" should be selected with regard to the safety of the system.

13.2 CAN Interface

<table>
<tr><td>Can Termination</td><td>Property</td><td>Activates the internal CAN terminator.
<table><tr><td>0</td><td>deactivate internal terminating resistance</td></tr><tr><td>1</td><td>activate internal terminating resistance</td></tr></table>as an initialization value</td></tr>
<tr><td>Baud rate CAN</td><td>Property</td><td>The baud rate setting for the 1st CAN bus interface:
<table><tr><th>Client</th><th>Baud rate kBit/s</th><th>Sample point</th></tr><tr><td>0</td><td>615</td><td>86</td></tr><tr><td>1</td><td>500</td><td>86</td></tr><tr><td>2</td><td>250</td><td>86</td></tr><tr><td>3</td><td>125</td><td>86</td></tr><tr><td>4</td><td>100</td><td>86</td></tr><tr><td>5</td><td>50</td><td>86</td></tr><tr><td>6</td><td>20</td><td>86</td></tr><tr><td>7</td><td>1000</td><td>86</td></tr><tr><td>8</td><td>83.33</td><td>75</td></tr></table>Alternatively, the baud rate can be set with the SetBaudrate() function.
as initialization value</td></tr>
<tr><td>Asy Tx Objects Can</td><td>Property</td><td>Here, the number of asynchronous Send objects sent per cycle can be set. ()
For detailed description, see "Asynchronous sending and receiving through VARAN“
Value range:
<table><tr><td>0-5</td><td>CAN objects per cycle</td></tr><tr><td>0</td><td>no asynchronous objects are sent</td></tr><tr><td>1-5</td><td>maximum number of objects that are sent</td></tr></table>as initialization value</td></tr>
<tr><td>Asy Rx Objects Can</td><td>Property</td><td>Here, the number of asynchronous Send objects sent per cycle can be set.
For detailed description, see "Asynchronous sending and receiving through VARAN“
Value range:
<table><tr><td>0-9</td><td>CAN objects per cycle</td></tr><tr><td>0</td><td>no asynchronous objects are received</td></tr><tr><td>1-9</td><td>maximum number of objects that are received</td></tr></table>as initialization value</td></tr>
<tr><td>Iso Tx Prescaler Can</td><td>Property</td><td>Using this client, the send time for the isochronous TX CAN objects can be increased by a multiple of the cycle time.</td></tr>
<tr><td>Iso Can</td><td>Interface</td><td>Interface for calling the isochronous send and receive methods. An Object channel to _CanLib is required (when using Extended-CanFrames, an object channel of the _SdiasCanIsoInterface class is required) The available functions are explained under "Isochronous Interface".</td></tr>
<tr><td>Asy Can</td><td>Interface</td><td>Interface for calling the asynchronous send and receive methods. An Object channel to _CanLib is required (when using Extended-CanFrames, an object channel of the _SdiasCanAsyInterface class is required) The available functions are explained under "Asynchronous Interface".</td></tr>
</table>

Can Termination State	State	Shows the status of the CAN termination resistance 0 termination resistance off 1 termination resistance on
Receive Buffer Overflow	State	Indicates whether the asynchronous CAN receive buffer has overflowed. Occurs when more asynchronous CAN objects were received than the available space in the buffer. To counter this problem, the "Asy Rx Objects Can" property must be increased or the bus cycle reduced.

13.3 Communication Interfaces

ALARM	Downlink	With this downlink the corresponding alarm class can be placed via the hardware editor.

13.4 Isochroneous Interface

13.4.1 IsInstalled

Returns whether the CAN interface is installed.

Return parameters	Type	Description
retval	DINT	0 CAN Interface is installed -1 the interface in this variant of the hardware is not available

13.4.2 GetBaudrate

This function provides the baud rate of the CAN interface.

Transfer parameters	Type	Description
Baudrate	^USINT	The target address to which the baud rate should be written 0 615 kBit/s 1 500 kBit/s 2 250 kBit/s 3 125 kBit/s 4 100 kBit/s 5 50 kBit/s 6 20 kBit/s 7 1 MBit/s 8 83,33 kBit/s
Return parameters	**Type**	**Description**
retval	DINT	0 function was successful ≠0 ErrorCode

13.4.3 SetBaudrate

This function is used to set the baud rate of the CAN interface.

Transfer parameters	Type	Description
us_BaudRate	USINT	0 = 615 kBit/s 1 = 500 kBit/s 2 = 250 kBit/s 3 = 125 kBit/s 4 = 100 kBit/s 5 = 50 kBit/s 6 = 20 kBit/s 7 = 1 MBit/s 8 = 83,33 kBit/s
Return parameters	**Type**	**Description**
retval	DINT	0 = function was successful ≠0 = ErrorCode

13.4.4 AddCanObj

Adds a Send or Receive object to the CAN bus.

Transfer parameters	Type	Description
ObjNr	INT	Object number (max. 2047)
length	USINT	Data length (max. 8)
Mode	USINT	1 = RX with callback 2 = TX
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0 = Handle of the CAN object <0 = ErrorCode

13.4.5 CanTxObject

Sends an object to the CAN bus.

Transfer parameters	Type	Description
ObjNr	INT	Object number (max. 2047)
length	USINT	Data length (max. 8)
data	PVOID	Pointer to the data
Return parameters	**Type**	**Description**
retval	INT	0 Function successful ≠0 ErrorCode

13.4.6 AddCanObjExtended

Adds a Send or Receive object to the CAN bus, which uses an Extend identifier (29-bit ObjNr instead of the 11-bit ObjNr). An object channel to the _SdiasCanIsoInterface class is required therefore.

Transfer parameters	Type	Description
ObjNr	DINT	Object number (max. 536870911)
length	USINT	Data length (max. 8)
Mode	USINT	1 RX with callback 2 TX
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0 Handle of the CAN object <0 ErrorCode

13.4.7 CanTxObjectExtended

Sends an object to the CAN bus, which uses an Extend identifier (29-bit ObjNr instead of the 11-bit ObjNr). An object channel to the _SdiasCanIsoInterface class is required therefore.

Transfer parameters	Type	Description
ObjNr	DINT	Object number (max. 536870911)
length	USINT	Data length (max. 8)
data	PVOID	Pointer to the data
Return parameters	**Type**	**Description**
retval	INT	0 Function successful ≠0 ErrorCode

13.4.8 GetMyStation

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.9 SetMyStation

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.10 LoginIntoCanNew

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.11 InitBasicCanObject

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.12 Set_RTR_Flag

Sets the Remote Transmit Request flag of the receiving object. So a request is sent to the according sending object. After receiving the object the RTR flag is reset.

Transfer parameters	Type	Description
handle	DINT	Number of the used CAN object (it is the return value of the function AddCanObj)
Return parameters	**Type**	**Description**
retval	DINT	0 OK -1 wrong handle

13.4.13 CanTxObjHandle

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.14 CanRxObjHandle

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.15 RedefCanObj

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.16 ChkObjExists

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.17 DelCanObj

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.4.18 DelBasicCanObj

Not supported by the isochronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5 Asynchrones Interface

13.5.1 IsInstalled

Returns whether the CAN interface is installed.

Return parameters	Type	Description
retval	DINT	0 CAN Interface is installed -1 the interface in this variant of the hardware is not available

13.5.2 GetBaudrate

This function provides the baud rate of the CAN interface.

Transfer parameters	Type	Description
us_BaudRate	^USINT	The target address to which the baud rate should be written 0 615 kBit/s 1 500 kBit/s 2 250 kBit/s 3 125 kBit/s 4 100 kBit/s 5 50 kBit/s 6 20 kBit/s 7 1 MBit/s 8 83,33 kBit/s
Return parameters	**Type**	**Description**
retval	DINT	0 function was successful ≠0 ErrorCode

13.5.3 SetBaudrate

This function is used to set the baud rate of the CAN interface.

Transfer parameters	Type	Description
us_BaudRate	USINT	0 615 kBit/s 1 500 kBit/s 2 250 kBit/s 3 125 kBit/s 4 100 kBit/s 5 50 kBit/s 6 20 kBit/s 7 1 MBit/s 8 83,33 kBit/s
Return parameters	**Type**	**Description**
retval	DINT	0 function was successful ≠0 ErrorCode

13.5.4 AddCanObj

Adds a Receive object to the CAN bus.

Transfer parameters	Type	Description
ObjNr	INT	Object number (max. 2047)
length	USINT	Data length (max. 8)
Mode	USINT	1 RX with callback
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0 Handle of the CAN object <0 ErrorCode

13.5.5 InitBasicCanObject

Adds a Receive object to the CAN bus.
This function also installs the filter to detect more than one CAN object.

Transfer parameters	Typ	Description
ObjNr	INT	Object number (max. 2047)
mask	USINT	Bit mask to select CAN objects 16#7FF: No objects except ObjNr are received 16#000: All objects are received
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0: Handle of the CAN object <0: ErrorCode

13.5.6 CanTxObject

Sends an object to the CAN bus.

Transfer parameters	Type	Description
ObjNr	INT	Object number (max. 2047)
length	USINT	Data length (max. 8)
data	PVOID	Pointer to the data
Return parameters	**Type**	**Description**
retval	INT	0: Function successful ≠0: ErrorCode

13.5.7 AddCanObjExtended

Adds a Send or Receive object to the CAN bus, which uses an Extend identifier (29-bit ObjNr instead of the 11-bit ObjNr). An object channel to the _SdiasCanAsyInterface class is required therefore.

Transfer parameters	Type	Description
ObjNr	DINT	Object number (max. 536870911)
length	USINT	Data length (max. 8)
Mode	USINT	1 RX with callback 2 TX
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0 Handle of the CAN object <0 ErrorCode

13.5.8 InitBasicCanObjectExtended

Adds a Receive object to the CAN bus, which uses an Extend identifier (29-bit ObjNr instead of the 11-bit ObjNr). An object channel to the _SdiasCanAsyInterface class is required therefore. This function also installs the filter to detect more than one CAN object.

Transfer parameters	Type	Description
ObjNr	DINT	Object number (max. 536870911)
mask	UDINT	Bit mask to select CAN objects 16#1FFFFFFF no objects except ObjNr are received 16#00000000 are received16#000: All objects are received
ActionPtr	PVOID	Pointer to the function called by the callback
ThisPointer	PVOID	Thispointer
Return parameters	**Type**	**Description**
retval	DINT	≥0 Handle of the CAN object <0 ErrorCode

13.5.9 CanTxObjectExtended

Sends an object to the CAN bus, which uses an Extend identifier (29-bit ObjNr instead of the 11-bit ObjNr). An object channel to the _SdiasCanAsyInterface class is required therefore.

Transfer parameters	Type	Description
ObjNr	DINT	Object number (max. 536870911)
length	USINT	Data length (max. 8)
data	PVOID	Pointer to the data
Return parameters	**Type**	**Description**
retval	INT	0 Function successful ≠0 ErrorCode

13.5.10 GetMyStation

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.11 SetMyStation

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.12 LoginIntoCanNew

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.13 Set_RTR_Flag

Sets the Remote Transmit Request flag of the receiving object. So a request is sent to the according sending object. After receiving the object the RTR flag is reset.

Transfer parameters	Type	Description
handle	DINT	Number of the used CAN object (it is the return value of the function AddCanObj)
Return parameters	**Type**	**Description**
retval	DINT	0 OK -1 wrong handle

13.5.14 CanTxObjHandle

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.15 CanRxObjHandle

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.16 RedefCanObj

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.17 ChkObjExists

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.18 DelCanObj

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.5.19 DelBasicCanObj

Not supported by the asynchronous interface.

Return parameters	Type	Description
retval	DINT	-1 function not supported

13.6 Example of the interrupt function

Transfer parameter	Type	Description
pCanPtr	^_Basic-CanObj	Pointer to the data in the CAN controller. These data are hardware dependent.
rxdataptr	^_Basic-CanObj	Pointer to the data read from the CAN controller before the interrupt.
thisp	^CanBase (Name of the Class)	Pointer to the class.

13.7 Type Description

_**BasicCanObj:** is a structure, which can contain CAN data. In addition, two entries are also available, in which the data length of the object and the object number are given.
CanBase: As shown in the following example, the name of the class is used in InitBasicCanObj.

13.8 _BasicCanObj

Label	Data type	Description
CanData	ARRAY [0..7] of USINT	Contains received CAN data
length	USINT	Contains the data length of the receive object
objnumber	UDINT	Contains the object number of the receive object

```
FUNCTION VIRTUAL GLOBAL CanBase::Init
VAR
retval : DINT;
END_VAR
if _firstscan then
if toAsyCan/toIsoCan.IsInstalled() = 0 then
toAsyCan/toIsoCan.SetBaudrate(1);
retval   :=   toAsyCan/toIsoCan.InitBasicCanObject(to_int(objNr),   16#01A0,   #CanIrq(),
this);
if retval < 0 then
ErrorCode := CANOBJ_ERROR; // If CAN Object couldn't be added
else
ErrorCode := 0;
end_if;
end_if;
end_if;
END_FUNCTION

FUNCTION CanBase::CanIrq
VAR_INPUT
pCanPtr : ^_BasicCanObj;
rxdataptr : ^_BasicCanObj;
thisp : ^CanBase;
END_VAR
this := thisp;
// This pointer correctly otherwise not be allowed to access variables of the class!
RXData := rxdataptr^; // Copies the data to the RXData struct
END_FUNCTION

FUNCTION VIRTUAL GLOBAL CanBase::CyWork
VAR_INPUT
EAX : UDINT;
END_VAR
VAR_OUTPUT
state (EAX) : UDINT;
END_VAR
toAsyCan/toIsoCan.CanTxObj(to_int(objNr), 8, #Data); // Daten werden zyklisch versendet
state := READY;
END_FUNCTION
```

13.9 Internal Properties

13.9.1 Initialization of CAN Objects

The CAN object must be created before the Init phase.

13.9.2 Isochronous Communication

Both Send and Receive objects must be assigned using AddCanObj (AddCanObjectExtended). The input parameters for the Callback must be valid for the Receive objects, these are not used for Send objects. To send, the CanTxObject function (CanTxObjectExtendet) is called. Up to 56 ISO-RX and 56 ISO-TX CAN objects can be created.

13.9.3 Asynchronous Communication

Receive objects must be created using AddCanObject or InitBasicCanObject (AddCanObjectExtended or InitBasicCanObjectExtended) only. The input parameters for the Callback must be valid. Send objects do not have to be predefined. To send, the CanTxObject function (CanTxObjectExtendet) is called.

13.9.4 Error Codes

-1	Interface not available / function not supported
-2	Incorrect length
-3	Invalid object number
-4	Invalid mode
-5	No CAN object free
-11	Object number already available

13.9.5 Asynchronous Sending and Receiving

For asynchronous communication, the number of CAN objects to send per cycle can be set or changed in the "AsyRxObjectsCanX" and "AsyTxObjectsCanX" clients.

Per cycle:
- when using at the local S-DIAS bus, this corresponds to the S-DIAS bus time
- when using a VI021/022, this corresponds to the VARAN bus time

- **AsyTxObjectsCan:** Per cycle, as many CAN objects are sent as are entered in this client (hardware classes buffer size = 32).

- **AsyRxObjectsCan:** Per cycle, as many CAN objects are received as are entered in this client (ICA buffer size = 157).

Due to the expected CAN messages, the values should be chosen so that there is no buffer overflow. In order to keep the bus load (SDIAS / VARAN) low, the values should not be selected unnecessarily large.

e.g.:
ICA on local S-DIAS bus – S-DIAS cylce time 2 ms
6 CAN send messages all 10 ms
3 CAN receive messages all 10 ms

AsyTxObjectsCan
=> 6 messages / 10 ms * 2 ms bus time = 1,2 messages per cycle
=> setting at least 2

AsyRxObjectsCan
3 messages / 10 ms * 2 ms bus time = 0,6 messages per cycle
=> setting at least 1

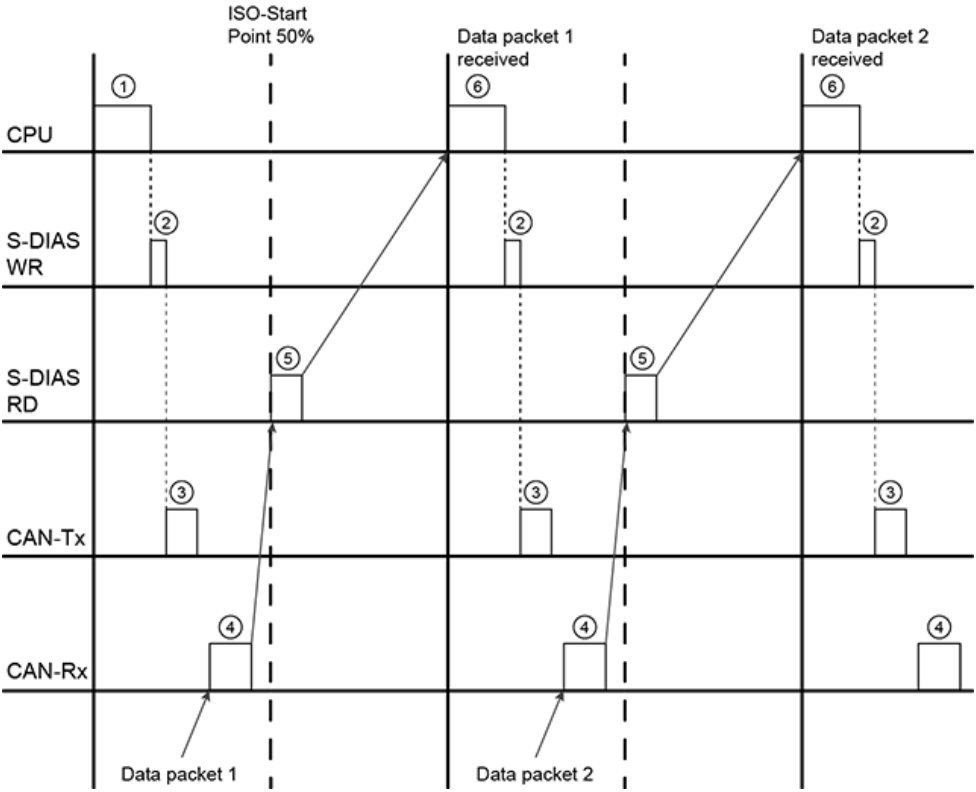
ICA via VI021/022– VARAN cycle time 4 ms
8 CAN send messages all 8 ms
16 CAN receive messages all 8 ms

AsyTxObjectsCan
8 messages / 8 ms * 4 ms bus time = 4 messages per cylce
=> setting at least 4

AsyRxObjectsCan
16 messages / 8 ms * 4 ms bus time = 8 messages per cycle
=> setting at least 8

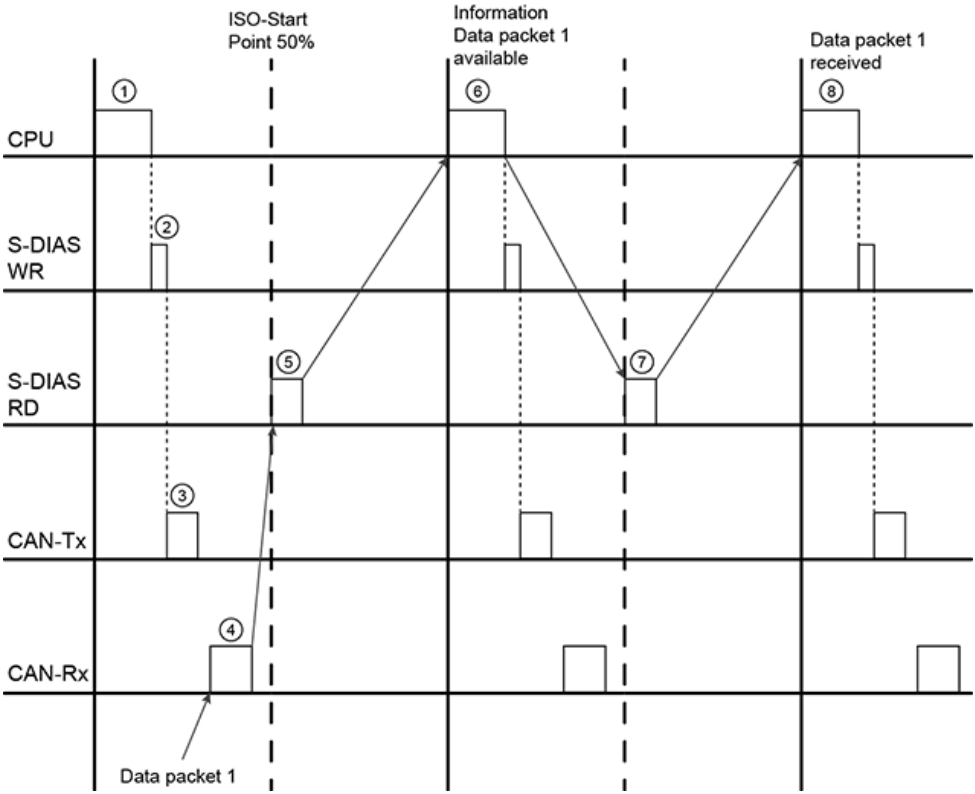
13.9.6 Time response sending/receiving

13.9.6.1 S-DIAS Local ISO



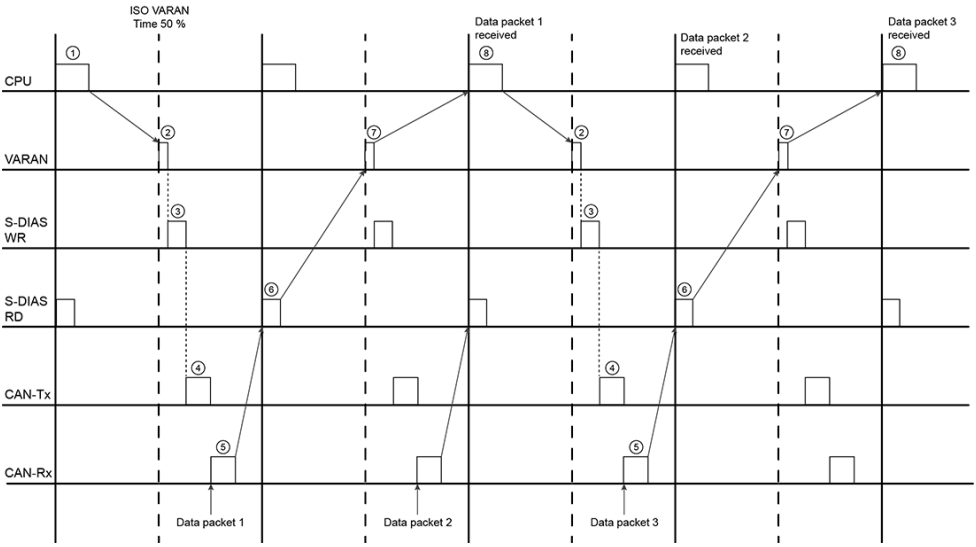
1. User class sends data.
2. Data is written in the CCA module.
3. Data are sent on the CAN bus.
4. Data packet 1 is received at the CAN bus.
5. ICA module reads data packet 1 from the receive buffer.
6. Data packet 1 is available and a callback is triggered. Data processed by user class.

13.9.6.2 S-DIAS Lokal ASY



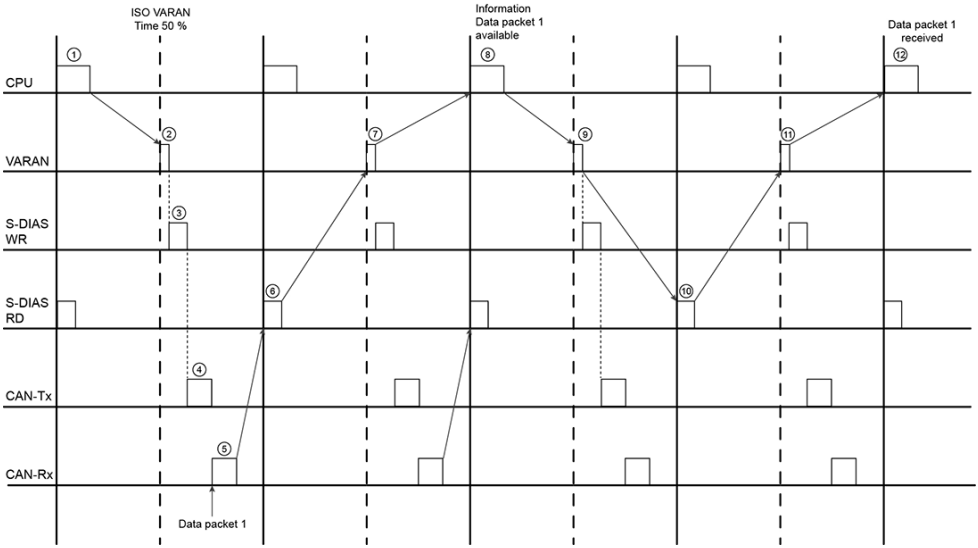
1. User class sends data.
2. Data is written in the CCA module.
3. Data are sent on the CAN bus.
4. Data packet 1 is received at the CAN bus.
5. ICA module registers that data are in the receive buffer.
6. HW class starts reading the data.
7. Data are read by the module via S-DIAS.
8. Data packet 1 is available and a Callback is triggered. User class processes the files.

13.9.6.3 S-DIAS via VARAN ISO



1. User class processes data.
2. Data are set by the VARAN bus.
3. Data is written in the CCA module.
4. Data are sent on the CAN bus.
5. Data packet 1 is received at the CAN bus.
6. ICA module reads data packet 1 from the receive buffer.
7. Data are transferred via the VARAN bus.
8. Data packet 1 is available and a callback is triggered. Data processed by user class.

13.9.6.4 S-DIAS via VARAN ASY



1. User class processes data.
2. Data are sent via VARAN bus.
3. Data is written in the CCA module.
4. Data are sent at CAN-Bus.
5. Data packet 1 is received at the CAN bus.
6. ICA module registers that data packet 1 is in the receive buffer.
7. The information that data are in receive buffer, is transferred by the VARAN bus.
8. Hardware class starts reading the data.
9. Request to read the data is transferred via VARAN.
10. Data are read by the module via S-DIAS.
11. Data are transferred at the VARAN bus.
12. Data packet 1 is available and a callback is triggered. Data processed by user class.

13.10 Appendix

Examples of bit masks for InitBasicCanObject:
Mask = 16#FFF0, ObjNr = 16#0000
=> Received objects: 16#0000 – 16#000F

Mask = 16#FFF0, ObjNr = 16#00F0
=> Received objects: 16#00F0 – 16#00FF

Mask = 16#0000, ObjNr = nicht benötigt
=> Received objects: all CAN-Objekte

Mask = 16#FEF0, Objnr = 16#0000
=> Received objects: 16#0000 –16#000F, 16#0100 – 16#010F

14 Transport/Storage

INFORMATION

This device contains sensitive electronics. During transport and storage, high mechanical stress must therefore be avoided.

For storage and transport, the same values for humidity and vibration as for operation must be maintained!

Temperature and humidity fluctuations may occur during transport. Ensure that no moisture condenses in or on the device, by allowing the device to acclimate to the room temperature while turned off.

When sent, the device should be transported in the original packaging if possible. Otherwise, packaging should be selected that sufficiently protects the product from external mechanical influences. Such as cardboard filled with air cushioning.

15 Storage

INFORMATION

When not in use, store the operating panel according to the storage conditions. See chapter 14.

During storage, ensure that all protective covers (if available) are placed correctly, so that no contamination, foreign bodies or fluids enter the device.

16 Maintenance

INFORMATION

During maintenance as well as servicing, observe the safety instructions from chapter 2 Basic Safety Directives.

16.1 Service

This product was constructed for low-maintenance operation.

16.2 Repair

INFORMATION

In the event of a defect/repair, send the device with a detailed error description to the address listed at the beginning of this document.

For transport conditions, see chapter 14 Transport/Storage.

17 Disposal

INFORMATION

Should you need to dispose of the device, the national regulations for disposal must be followed.

The device appliance must not be disposed of as household waste.

Documentation Changes

Change date	Affected page(s)	Chapter	Note
25.07.2014	1 5 6 8	 2 Mechanical Dimensions 3 Connector Layout 4.1 Example Connection	Plug changed
05.11.2014	8	4.1 Example Connection	CAN allocation
30.01.2015	8	4.2 Applicable Connectors	Added note concerning connecting the S-DIAS module while voltage is applied
26.03.2015	8	3.2 Applicable Connectors	Added connections
21.01.2016	4	1.3 Miscellaneous	Standards changed
25.01.2015	3	1.2 Electrical Requirements	Graphics added
28.04.2016	17	7 Mounting	Graphics distances
06.09.2016	1 7	 2 Mechanical Dimensions	Text changed Graphic corrected
21.09.2016	11	4.1 Example Connection	Note added
17.02.2017	8	3 Connector Layout	LED colors
17.08.2017	7 11	1.4 Environmental Conditions 3.2 Applicable Connectors	Pollution Degree Sleeve length added Added info regarding ultrasonically welded strands
18.10.2017	13 20	3.3 Label Field 7 Mounting	Added chapter Graphic replaced
25.06.2018	5	1.2 Electrical Requirements	Note UL conditions
14.11.2019	23	8 Supported Cycle Times	Chapter added
28.02.2020	23	8 Supported Cycle Times	Text adapted
08.09.2020	24	9 Hardware Class ICA011	Chapter added
04.11.2020	21	7 Mounting	Expansion functional ground connection
11.07.2022	7	1.2 Electrical Requirements	Current consumption +5 V corrected
06.12.2022	10	1.3 Miscellaneous	UKCA conformity

26.07.2023		Document	General chapters added, design