



# CAN communication

for isoPV1685 insulation monitoring devices





# 1 Introduction Introduction

This document describes the communication between internal components and in particular with respect to the connection of the insulation monitoring device isoPV1685.

The overall system consists of several components, which are linked together via a CAN bus.

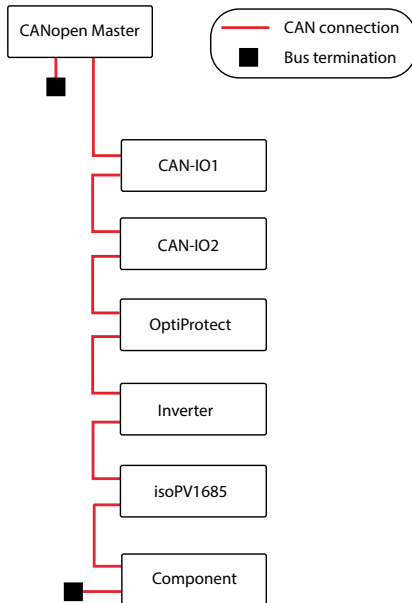
Each component has to two ports, so that an arbitrary arrangement of the devices as the bus is possible in order to take into account local conditions in the cabinet.

Essentially, the communication is tasked with passing on the detected insulation resistance value and differential current value to an evaluating unit that performs error handling measures.

## 1.1 Acronyms and Abbreviations

Acronym	Description
CAN IO	Remote IO units connected via CAN
OptiProtect	String current monitoring via CAN bus connection to the inverter with switching possibility
isoPV1685	Bender insulation monitoring device used for insulation, and depending on the variant, residual current monitoring in large photovoltaic systems.

## 1.2 CAN communication components



## 2 CAN communication concept

### 2.1 CAN communication protocol

Message exchange can be divided into two different categories:

- Process Data Object (PDO) - data which are periodically sent from the device
- Service Data Objects (SDO) - data requested/data to be changed by the communications Master

Depending on the category, the message exchange is handled differently.

In general, each device is assigned a device ID, from which the used CAN message IDs are derived:

- PDOs are sent with the sender ID PDO Tx ID.
- SDO requests go to the SDO Rx ID of the recipient, which in turn responds with its SDO Tx ID

### 2.2 CAN message

A typical CAN message (CAN 2.0 A standard message frame) is shown below:

Frame	Start	Identifier (ID)	RTR	IDE	r0	DLC	Data	CRC	ACK	EOF IF
Bit length	1	11	1	1	1	4	0...64	15	2	10

With this message, the main frames are the CAN ID and the actual data (DATA1-DATA4), which consists of 4 Words:

CAN ID	DATA1	DATA2	DATA3	DATA4
11 Bit	WORD	WORD	WORD	WORD

*Because of physical limitations, the number of bytes in a CAN message is set to a maximum of 8 bytes (4 words).*

To simplify the software (the same structures can be used for SDO (tx) and SDO (rx)), all messages in this specification have a data length of 4 words.

#### 2.2.1 CAN ID

The CAN ID is 11 bits long and consists of a 4-bit Function code, and Device type and Device count (7 bits). The CAN ID structure is as follows:

Function code	Device address
4 BIT	7 BIT

## 2.2.2 Function code

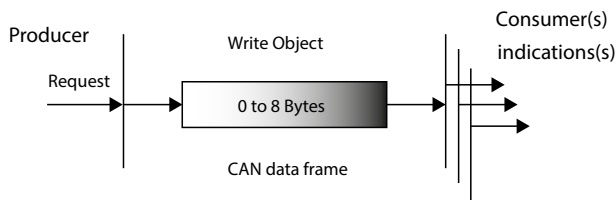
The function code can be modeled on the CANopen protocol and assume the following values:

Service	Function code (binary)	Possible CAN IDs
NMT	0000	-
SYNC	0001	-
TIME STAMP	0010	-
EMCY	0001	129 (081h) - 255 (0FFh)
PDO1 (tx)	001	385 (181h) - 511 (1FFh)
PDO1 (rx)	0100	513 (201h) - 639 (27Fh)
PDO2 (tx)	0101	641 (281h) - 767 (2FFh)
PDO2 (rx)	0110	769 (301h) - 895 (37Fh)
PDO3 (tx)	0111	897 (381h) - 1023 (3FFh)
PDO3 (rx)	1000	1025 (401h) - 1151 (47Fh)
PDO4 (tx)	1001	1153 (481h) - 1279 (4FFh)
PDO4 (rx)	1010	1281 (501h) - 1407 (57Fh)
SDO (tx)	1011	1409 (581h) - 1535 (5FFh)
SDO (rx)	1100	1537 (601h) - 1663 (67F)
NMT Error Control	1110	1792 (700h) - 1919 (77Fh)
RESERVE	1111	1984 (7C0h) - 2047 (7FFh)

**i** For periodic messages, only Function code PDO1 (tx) is used by the devices. For data exchange upon request, Function codes SDO (tx) and SDO (rx) are used. The Master computer can also periodically send messages with PDO1 (tx).

## 2.3 Process Data Object (PDO)

Process data is sent periodically from the device. The PDO mechanism is shown below:



Data transmission is not secured with handshaking or by other similar means. Because the data are sent periodically (PDO (tx)), data security is provided (even in the event of a message failure) no later than after the next period when an actual message is sent.

If the data is to be securely transmitted, other measures must be resorted to, for example:

- Monitoring with Timeout is possible
- Secure transmission with handshaking as an SDO
- Secure the transmitted data by adding information (for example, periodic counter, checksum, bitwise-inverse transfer etc.)

## Multiplex mechanisms

In order to transfer larger amounts of process data, the data are multiplexed for certain PDOs. Therefore, the multiplex number (i.e. index) becomes the first word of each message, which then identifies the actual data. This then reduces the user data to 3 words per CAN message:

Service	INDEX	DATA1	DATA2	DATA3
PDO (tx)	0x0000 - 0xFFFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF

In software, the multiplex mechanism is described as follows:

```

Typedef struct _can_pdo {
    WORD wIndex; // 16-Bit Word
    WORD wData1; // 16-Bit Word
    WORD wData2; // 16-Bit Word
    WORD wData3; // 16-Bit Word
} CAN_PDO;

```

**i** *The Words are stored as Intel hexadecimal format. The definition and PDO structure are defined by CANopen.*

## 2.4 Service Data Objects (SDO)

Requests

Requests are sent to the SDO Rx ID of the requested subscriber. A typical SDO request structure is shown below:

Service	Request	Index	Subindex	DATA1	DATA2
SDO (rx)	0x40	0x2000 - 0x2FFF	0x01 - 0xFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF

The subscriber responds - within 2.5 s - with its SDO Tx ID, the structure of which is:

Service	Response	Index	Subindex	DATA1	DATA2
SDO (tx)	0x80	0x2000 - 0x2FFF	0x01 - 0xFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF

In software, the multiplex mechanism is described as follows:

```
Typedef struct _can_sdo {  
    BYTE bRR;      // 8-Bit Word  
    WORD wIndex;   // 16-Bit Word  
    BYTE bSubindex; // 8-Bit Word  
    WORD wData1;   // 16-Bit Word  
    WORD wData2;   // 16-Bit Word  
} CAN_SDO;
```

**i** *The Words are stored as Intel hexadecimal format. The definition and SDO structure are defined by CANopen.*

## 2.5 Network Management (NMT)

There is only one NMT master in a CANopen network, all other devices are NMT slaves. The NMT structure is as follows:

Service	DATA1
NMT	0x0000 - 0xFFFF

```
Typedef struct _can_nmt {  
    WORD wData1;      // 16-Bit Word  
} CAN_NMT;
```

## 2.6 Emergency (EMCY)

Emergency messages are triggered by the occurrence of a device internal fatal error situation and are transmitted from the concerned application device to the other devices with high priority. An Emergency (EMCY) message has the following structure:

Service	DATA1	DATA2	DATA3	DATA4
EMCY	0x0000 - 0xFFFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF	0x0000 - 0xFFFF

```
Typedef struct _can_emcy {  
    WORD wData1;      // 16-Bit Word  
    WORD wData2;      // 16-Bit Word  
    WORD wData3;      // 16-Bit Word  
    WORD wData4;      // 16-Bit Word  
} CAN_EMCY;
```

## 2.7 Synchronisation (SYNC)

A synchronisation (SYNC) message has the following structure:

Service		Description
SYNC = 0x80	No data	Synchronisation message for synchronous data transmission

**i** *This signal is reserved.*

## 3 Data protocol

The CAN interface uses an 11-bit ID and a speed of 500 kBits. The protocol used is described in Chapter 2.1 CAN communication protocol. The data and commands used are described in this chapter.

### 3.1 CAN IDs

According to the application internal address coding, an address for system components is constructed. The definition of CAN IDs used is:

Function code	Device address
<b>4 BIT</b>	<b>7 BIT</b>
NMT = 0x0000 EMCY = 0x0080 TIME_STAMP = 0x0100 TxPDO1 = 0x0180 SDO(tx) = 0x0580 SDO(rx) = 0x0600 NMT_ERR = 0x0700	Device address 0x27 or decimal 39 for the isoPV1685 module results in: EMCY = 0x00A7 TxPDO1 = 0x01A7 SDO(tx) = 0x05A7 SDO(rx) = 0x0627 NMT_ERR = 0x0727
NMT = 0x0000 EMCY = 0x0080 TIME_STAMP = 0x0100 TxPDO1 = 0x0180 SDO(tx) = 0x0580 SDO(rx) = 0x0600 NMT_ERR = 0x0700	Device address 0x28 or decimal 40 for another system component, for example, results in: NMT = 0x00A8 TxPDO1 = 0x01A8 SDO(tx) = 0x05A8 SDO(rx) = 0x0628 NMT_ERR = 0x0728

### 3.2 Reserved CAN IDs

In addition to the addressing scheme above, addresses for different functions are reserved.

CAN ID	Description
0x100	TIME STAMP
0x080	EMCY
0x000	NMT

**i** *The receiving filter should be set so that only the messages intended for the respective subscriber are received.*



### 3.3 Transmission data (SDO)

Index/Subindex	DATA1 (WORD)	DATA2 (WORD)	Description
0x2011/0x01	CMD_MODULE	STATUS	Commands/status
0x2011/0x02	SPOT_NUM	SPOT_VAL	Request value
0x2011/0x03	PARAM_NUM	PARAM_VAL	Request parameter
0x2011/0x04	PARAM_SET	PARAM_VAL	Set parameter

#### 3.3.1 CMD\_MODULE command

Via the CMD\_MODULE command, specific commands are transmitted to the individual modules. In order not to limit the possible range of commands to 16 bits, a data value is transmitted, i.e. each possible value from 0 ... (216 - 1) corresponds to a command. To keep the structure as clear as possible, each command word is structured the same way. The following table provides an overview of the available commands.

DATA1	Command	Description
0	RESERVE	
1	RESET_ERROR	With this command an error, such as a CAN timeout for example, is acknowledged

#### 3.3.2 SPOT\_NUM (Request measurement value )

The following table provides an overview of the available spot values:

DATA1	Term	Description
0	STATUS	Status of the isoPV1685 module As a WORD (lower 16 Bits) see Chapter 3.8.1
1	RISO	DC side insulation resistance As a WORD in 100 Ohm
2	IDIFF	AC side residual current As a WORD in mA
3	GRIDVTG	Mains voltage - As a WORD in Volts
4	VTGDC+	Voltage DC+/PE As a WORD in Volts
5	VTGDC-	Voltage DC-/PE As a WORD in Volts

### 3.3.3 PARAM\_NUM (Request parameter)

The following table provides an overview of the available parameters:

DATA1	Term	Description
0	STATUS	Set condition of the isoPV1685 module. See Chapter 3.8.2
1	ISOWARN	isoPV1685 insulation resistance warning threshold. As a WORD in 100 Ohm
2	ISOERR	isoPV1685 insulation resistance fault threshold. As a WORD in 100 Ohm
3	IDIFFWARN	isoPV1685 AC residual current warning threshold As a WORD in mA
4	IDIFFERR	isoPV1685 AC residual current fault threshold As a WORD in mA

### 3.3.4 PARAM\_SET (Set parameter)

The same table applies as described in Chapter 3.3.3 (PARAM\_NUM). The Set parameter differs from the Request parameter in the acknowledge command:

#### PARAM\_SET (Acknowledge command)

To change the device parameters, the PARAM SET/ PARAM\_VAL (Index 0x2011 / Subindex 0x04) message is used.

If the parameter could be changed, the device sets PARAM\_SET und PARAM\_VAL to the required value in the message response.

If the device cannot change the parameter, then the current valid value for PARAM\_VAL is sent in the message response.

## 3.4 Network Management (NMT)

Communication control

CAN ID	Byte0	Byte1	Description
0	0x01	0x00	START: periodic communication
0	0x02	0x00	STOP: periodic communication

A bus device starts in the pre-operational state. The operational state occurs when the START signal enables the transmission of the periodic data. SDO communication is permitted in the pre-operational and operational states.

**i** *Attention! PDOs can be periodically transmitted only after communication has been initiated by the START command. Periodic transmission must be terminated after the STOP signal has been received.*

States of a network node

Byte0	Description
0x00	Initialisation
0x01	Pre-operational
0x02	Operational

### 3.5 Synchronisation (SYNC)



*This function must be reserved to synchronously detect and transmit data if necessary*

CAN ID		Description
SYNC = 0x80	No data	Synchronisation message for synchronous data transmission

### 3.6 Sign of life

The CANopen Master sends, among others, a PDO "sign of life" message every 200 ms. This PDO message (See Chapter 2.3) with the index 0 contains no data that can be evaluated.

Index	DATA1	DATA2	DATA3	Description
0	Reserved	Reserved	Reserved	200 ms



*This message can be evaluated by the individual isoPV1685 modules if required. However, the isoPV1685 does not evaluate this information.*

### 3.7 Time stamp (TIME\_STAMP)

Synchronising the component clocks to under one second is desirable in order to enable a finer resolution of diagnostic data for the future. For this purpose, the timestamp format is implemented in CANOpen:

Usually the time stamp object represents an absolute time in milliseconds after midnight (Component ms) and the number of days since January 1, 1984 (Component days).

STRUCT OF

```

    UNSIGNED28    ms
    VOID4         reserved
    UNSIGNED16    days
    
```

TIME\_OF\_DAY

This information is sent with the TIME\_STAMP ID of 0x100.

The CANopen Master periodically sends a timestamp as a PDO to synchronise the clocks available in the system. The timestamp is sent at 60 s intervals.

ID	ms	res	days	Cycle time
0x100	28 Bit	4 Bit	16 Bit	–
	ms of current day	Reserved	Days since 01.01.1984	–

### 3.8 Transmission of insulation monitoring data (PDO)

This data is transmitted periodically by the isoPV1685. This is done with the PDO (tx) message. In the isoPV1685 device variant that does not perform residual current measurements, the message with index 0 is omitted.

Index	Address	DATA1	DATA2	DATA3
0	CANopen Master	STATUS	IDIFF	Redundancy (See Chapter 3.8.2)
1	CANopen Master	STATUS	RISO	Redundancy (See Chapter 3.8.2)

Where:

Data element	Type	Description
STATUS	WORD	Status of the isoPV1685 module encoded in the lower 16 bits in the value range 0x0000 ... 0xFFFF (see table in Chapter 3.8.1)
RISO	WORD	DC-side insulation resistance in 100 Ohm Value range 0x0000 ... 0xFFFF
IDIFF	WORD	AC-side residual current in mA Value range 0x0000 ... 0xFFFF

### 3.8.1 Description of the isoPV1685 module STATUS information

State	Access	Status WORD		Note
		High Byte CODE	Low Byte Flags	
Normal	R/W	0x00	0000 0000	Idle With write access to the status word with 0x0000, errors are acknowledged or the test aborted
			0000 0001	Free running. The device works without time synchronization
			0000 0010	PGH active Insulation fault locator is running
Test	R/W	0x10		Self test
			0x11	Insulation test
			0x12	Residual current test
Alarm	R	0x20	0000 0001	Isometer prewarning
			0000 0010	Isometer Alarm
			0000 0100	Insulation fault location active
			0000 1000	Isometer earth fault
			0001 0000	Residual current prewarning
			0010 0000	Residual current Alarm
			0100 0000	Residual current transformer connection error
			1000 0000	Residual current less than 1 A (out of range)
Device fault	R	0x80	-	Internal device fault Functionality not provided

### 3.8.2 Safety-related transmission - redundancy

The RISO (Insulation resistance on the DC side) and IDIFF (Residual current on the AC side) information should be safely transmitted. As a result, a hardware shutdown may be omitted because of the looping of the isoPV1685 module relay contacts into the safety-shutdown chain. On the one hand, this would result in cost savings. On the other hand, it may be possible in justified cases (possibly restricted) for continued operation, for example, for the purpose of fault isolation. Also, depending on the application, the warning and tripping thresholds can be adjusted without having to reconfigure the isoPV1685.

Furthermore, redundant triggering is also possible. With stricter thresholds within the application and less strict thresholds in the isoPV1685 (which only come into play when the application would fail).

The transmission of the safety level of the communication line need not be considered; instead, the so-called "Black Channel" principle is applied, i.e. the message will be given additional safety-relevant information that is created when the message is generated and which is evaluated together with the message. This then allows the recipient to check the validity of the information before use.

The redundancy information is, in this case, so formed that the source of the message from the multiplex value, and the transmitted safety-relevant information (a 4-byte numerical value) forms the 16-bit sum (summed bitwise together). This value is then bitwise XOR-ed with 0xFFFF in the two last bytes of the message.

**i** *The value 0x1234 should be transmitted with multiplex 0x0002.*

This results in a message with the following structure:

Byte	Value	Description
0	0x02	Multiplex 2 low
1	0x00	Multiplex 2 high
2	0x12	Value of lowest byte
3	0x34	Value of low word high byte
4	0x56	Value of high word low byte
5	0x78	Value of highest byte
6	0xE9	Checksum (0x02 + 0x00 + 0x12 + 0x34 + 0x56 + 0x78 = 0x116) XOR 0xFFFF = 0xFEE9 low byte
7	0xFE	Checksum high byte

### 3.9 Sequence of CAN messages

So that the requirement of error handling with respect to a residual current is carried out within one second, at least 3 measurement readings must be evaluated in order to carry out a signal "debounce".

The insulation monitoring device PDO messages are periodically sent in the following sequence at intervals of 125 m/s:

Sequence	PDO index	Data elements
1	0	Status + residual current + redundancy information
2	1	Status + Insulation resistance + redundancy information

**i** *This differential current and the insulation resistance are transmitted every 250 ms and the status information every 125 ms.*

For the isoPV1685 device variant that does not perform residual current measurement, the PDO message with index 0 is omitted so that the message containing the insulation resistance is transmitted every 250 ms. The message delay must not exceed 250 ms.

### **3.10 Invalid message (EMCY)**

If an isoPV1685 module receives a message it does not recognise, it will reply to this message with an emergency message. The data content of a received "unknown" message is stored in the Emergency message (EMCY) response.

For example:

A subscriber wishes to query a parameter (PARAM\_NUM) - via an SDO message - that does not exist in the device.

The message content is sent back with the EMCY ID (0x0080 + Device ID).

For isoPV1685 variants that do not perform residual current measurement, the EMCY message is sent after any attempt to change or query parameters not available in the variant. Chapter 2.1 CAN communication protocol. The data and commands used are described in this chapter.



Alle Rechte vorbehalten.  
Nachdruck und Vervielfältigung  
nur mit Genehmigung des Herausgebers.

**Bender GmbH & Co. KG**

Postfach 1161 • 35301 Grünberg • Deutschland  
Londorfer Str. 65 • 35305 Grünberg • Deutschland  
Tel.: +49 6401 807-0 • Fax: +49 6401 807-259  
E-Mail: [info@bender.de](mailto:info@bender.de) • [www.bender.de](http://www.bender.de)



All rights reserved.  
Reprinting and duplicating  
only with permission of the publisher.

**Bender GmbH & Co. KG**

PO Box 1161 • 35301 Grünberg • Germany  
Londorfer Str. 65 • 35305 Grünberg • Germany  
Tel.: +49 6401 807-0 • Fax: +49 6401 807-259  
E-Mail: [info@bender.de](mailto:info@bender.de) • [www.bender.de](http://www.bender.de)