Protocol specification
SI Modbus

SolarInvert inverter

Preface

Dear customer,

Thank you for your confidence in our products. SolarInvert GmbH has over 10 years of experience in the development and production of high-performance electronic components and systems for photovoltaics. We have made it our mission to always serve our customers to the highest standards.

The technology and equipment of our products meet the latest national and international requirements in terms of functionality and safety. Further developments and improvements are constantly being taken into account, which is why illustrations, dimensions and technical data as well as general contents listed in this manual may change as a result of adaptation to new findings. We have decided not to test every new development or market trend at the expense of our customers. We therefore only introduce new technologies and concepts after conducting our own in-depth tests and studies.

The naming of products from other manufacturers in this manual is for information purposes only and does not constitute trademark misuse.

Disclaimer

The texts and illustrations have been compiled with the utmost care. Despite all efforts, errors cannot be completely avoided. For the correctness of the content can therefore not be guaranteed. For incorrect information and its consequences we cannot accept any legal responsibility or liability whatsoever.

We are always grateful for suggestions for improvement and information on errors. We endeavor to always adapt these instructions to the latest device version.

However, it is possible that there may be differences between devices and instructions. We reserve the right to make technical changes.

Table of contents

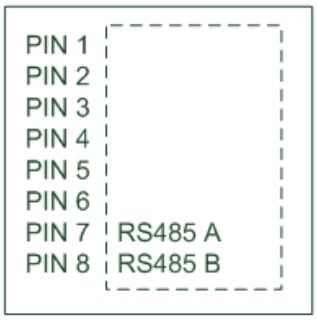
Revision History

Index	Date	Modification
005	01.06.2014	Transfer to new style sheet
009	30.09.2014	Function code 3Bh revised
010	08.10.2014	Value ranges and resolution for all FCs revised
011	02.01.2015	Function code 3Bh revised
012	10.01.2015	Introduction of further function codes
013	21.2.2015	- Removal of the offset for response address - Introduction of CRC16 - Removal of automatic address assignment by the master - Udc step size set to 0.1V - Function code 44h added - Function code 33h revised
014	25.2.2015	- Negative response revised - Change to no-parity - Function code 38h revised - Example for response 33h revised
015	20.3.2015	Response 3Eh revised
016	26.03.2015	Response 33h revised
017	21.04.2015	Addition to 1.4 to include reference to transmission pause during boot process Correction in the identifier list of response 31h Correction in the identifier list of response 33h Correction in the identifier list of response 3Eh Correction in the identifier list of response 3Bh Correction in the identifier list of answer 3Fh Correction in the identifier list of answer 44h
018	05.05.2015	Pin assignment of the RJ45 sockets added Answer 3Fh revised Answer 38h revised
019	12.05.2015	Correction: No termination for wind inverters Jumper assignment for network address added
020	26.05.2015	- Answer 3Eh revised - Assignment matrix for voltage type and inverter model added
021	04.12.2015	Correction of the scale factor for Udc (cmd 33h) only inverters with voltage type 4 (SOL-50, WIN-36, BAT-30) are affecte
022	23.05.2016	Correction in answer 31h
030	05.07.2016	Chap. 4.7 and correction of all checksums
031	03.05.2017	Fundamental revision with extensive error corrections and alignment with device software
032	31.10.2019	Correction of display of negative values for power PAC in response 33h and value range for PMAX in response 3Eh Addition to response 40h for new errors from v60
033	03.10.2021	Correction of number of bytes in response 3Eh and number of reserved bytes
036	18.10.2024	Correction Address Customer service

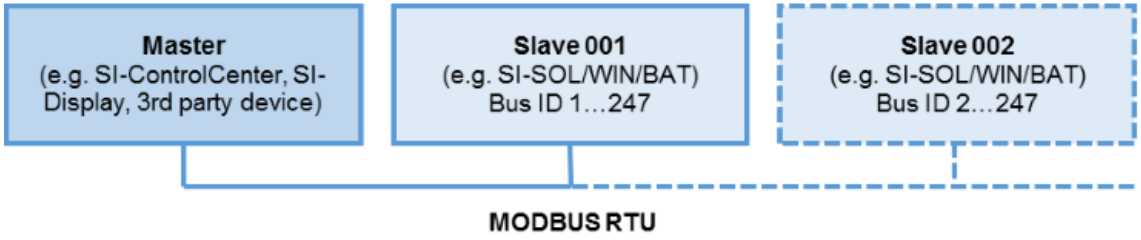
1 Communication via SI-Modbus

SolarInvert inverters offer a serial interface (RS485) and use proprietary radio codes of the Modbus RTU communication protocol. More detailed information on Modbus RTU beyond this manual is available at www.modbus.org.
Modbus is a master-slave protocol that supports up to 247 slaves but only 1 master at a time. The master can be, for example, a PC on which SolarInvert Control Center (SICC) is installed, or another compatible device such as a PowerDog..

1.1 Pin assignment for RJ45 connector



1.2 Bus Layout



1.3 Termination

To avoid signal interference, the two end points of the bus must be terminated with an end-of-line resistor. Use a twisted pair cable to connect the devices and, if necessary, a 120 Ohm resistor for termination.

Devices with only one RJ45 socket, such as WindInverters and BatteryInvert, are supplied with an internal terminating resistor as standard. For all other devices, a terminating resistor can be connected via the free RJ45 socket.

1.4 Galvanic isolation

If devices from other manufacturers are to be integrated into the bus, it is essential to ensure that the RS485 bus is electrically isolated from the circuit of the devices.

1.5 Network management and address assignment

After switching on, the master performs a network scan by querying all available addresses in the network (polling). To ensure that all devices are recognized correctly, it must be ensured that they are switched on and ready to receive.

While an inverter is in the boot process, the inverter must not be addressed by the master. This would be interpreted by the bootloader as an attempt to flash the firmware. The master must therefore give the inverter a few seconds, as otherwise the inverter cannot complete the boot process.

NOTE

SolarInvert inverters switch off without a DC source and can then no longer be addressed via RS485! Polling must therefore take place with sufficient irradiation, wind strength or a closed battery disconnector.

The following conditions apply to the address assignment of the slaves:

- On delivery, the slave responds under its default device ID, which is defined based on the production number between 11 and 20.
- In the event of an address conflict, the SolarInvert Control Center (SICC) can be used to change the device ID. Only the device whose device ID is to be changed may be connected to the PC.

2 Data transmission

- Baud rate 57600, 1 start bit, 1 stop bit, 8 data bits, no parity
- High byte is transmitted before low byte
- For all values longer than 1 byte, the MSB is transmitted first
- The CRC checksum is calculated over the entire message

2.1 Code examples for the checksum calculation

2.1.1 Sample implementation for calculating the checksum:

```
void CALC_ CRC16(unsigned char len)
{
 unsigned char   I;
 unsigned char   I1;
 unsigned short  CRC;
 unsigned short  LSB;
 CRC = 0xFFFF;
 for (I=0; I<len; I++)
    {
     CRC = ((CRC^TXD_TAB[I])| 0xFF00) &(CRC | 0x00FF);
                  //TXD_TAB[I] = OUTGOING DATASTREAM
     for (I1=0; I1<8; I1++)
        {
         LSB=(CRC & 0x0001);
         CRC= CRC/2;
         if (LSB)
            CRC=CRC ^0xA001;
        }
    }
 TXD_TAB[I] = (CRC & 0x00FF);          // HCRC
 TXD_TAB[I+1] = (CRC & 0xFF00)/256;       // LCRC
```

2.1.2 Sample implementation for checking the checksum:

```
 unsigned char CHECK_CRC16(unsigned char len)
{
 unsigned char     I;
 unsigned char     I1;
 unsigned short    CRC;
 unsigned short    LSB;
 unsigned char     RESULT;
 CRC = 0xFFFF;
 for (I=0; I<len; I++)
    {
     CRC = ((CRC^RXD_TAB[I])| 0xFF00) &(CRC | 0x00FF);
                  //RXD_TAB[I] = INCOMMING DATASTREAM
     for (I1=0; I1<8; I1++)
        {
         LSB=(CRC & 0x0001);
         CRC= CRC/2;
         if (LSB)
            CRC=CRC ^0xA001;
        }
     }

 if ((RXD_TAB[I] == (CRC & 0x00FF)) && ((RXD_TAB[I+1] == (CRC &
0xFF00)/256)))
      return RESULT = true;
 else
      return RESULT = false;
}
```

2.1 Request telegrams

Communication on the bus is always initiated by a request from the master. This uses function codes to grant access to information and control options of the slaves. Depending on the definition of the function code, a variable amount of data is sent, which is processed accordingly by the slave.

A request has the following telegram structure:

Slave-address	Function code	Data	Checksum

Slave-address	Device ID of the addressed slave (1 byte)
Function code	Determines the action that the addressed slave performs (1 byte)
Data	Dependent on the function code used (variable length)
Checksum	CRC16 (2 Bytes)

2.2 Response telegrams

Every request sent (with the exception of broadcasts) is acknowledged by a response from the addressed slave. The response depends on the function code used. If the function code does not define a specific response, the slave will send a standard response.

A standard response has the following telegram structure:

Slave-address	Function code	Length (0x 01)	Data (0x 01)	Checksum

If the received request could not be processed, an exception response is sent instead. It contains an exception code that allows conclusions to be drawn about the cause of the error.

An exception response has the following telegram structure:

Slave-address	Function code (+80h)	Exception code (see below)	Checksum

To make the exception response recognizable as such, the function code of the request is incremented by 80h.

Exception code:

Code	Designation	Description
0x 01	Invalid function code	The function code used is not supported
0x 02	Invalid register address	Register not within the permitted memory area
0x 03	Invalid data value	Incorrect number, types or value ranges of the arguments
0x 04	Device error	An internal device error has occurred

3 Standard function codes

The Modbus standard implementation uses function codes to read and write 16-bit registers that are located in the slave's working memory. However, this requires precise information about the addresses and data types of all available registers, as well as their meaning, and is only suitable for experts.

SolarInvert Devices generally support the following standard function codes:

- 0x 03 (reading holding registers) **- as from firmware v.62**
- 0x 16 (writing holding registers) **– as from firmware v.6?**

From firmware v.62, the registers are organized according to the SUNSPEC standard and can be read, but not yet written due to backwards compatibility with the proprietary function codes. More information on the SUNSPEC standard at http://sunspec.org/

4 Proprietary function codes

As an alternative, SolarInvert inverters support special function codes for the most common applications, which can be used without precise knowledge of the register addresses and allow quick access to the basic functions of the devices.

When using the proprietary function codes, only the correct sequence, number and value range of the arguments specified by the function code must be observed. Addressing errors are therefore ruled out.

NOTE

All data values except string are represented as unsigned integers! The sign-and-magnitude representation is used for negative data values. For decimal numbers, the bit value is multiplied by a scale factor on the master side. This and other useful information can be found in the “Arguments used” table of the respective function code.

4.1 Reading out manufacturing data (cmd 31h)

This request provides you with information about the serial number, production date and software versions of your inverter. The model designation cannot be read out individually (see Section 4.2, cmd 3Eh).

Request 31h:

Address	Function code	Length	Data	CRC

Addresse	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 31	Calls function 31h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 90 47	Checksum (2 bytes)

Respond 31h:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 31		Calls function 31h (1 byte)
Length		0x 0C		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	SNR	0x 00 01	1	Serial number (2 bytes)
	PDATE	0x 1E 04 0D	30.04.13	Production date DD-MM-YY (3 bytes)
	BOM	0x 22	v.34	Hardware version number (1 byte)
	FW	0x 32	v.50	Firmware version (1 byte)
	BB-ID	0x 28	v.40	Bootstrapper version number (1 byte)
	RES	0x 00 00 00 00	0	Reserved, is not evaluated (4 bytes)
CRC		… …		Checksum (2 Bytes)

Arguments used 31h:

Identifier	Typ	Format	Value range	Scale factor	Other
SNR	Uint16	0	0… 65535	1	
PDATE	3x Uint8	DD.MM.JJ	1…31 1…12 0…99	1	
BOM	Uint8	0	0…255	1	
FW	Uint8	0	0…255	1	
BB-ID	Uint8	0	0…255	1	

4.2 Reading out operating data (cmd 33h/3Eh)

Two functions (33h and 3Eh) are available for reading out the operating data of the inverter. Select the appropriate function based on the data you are interested in.

Request 33h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 33	Calls function 33h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 31 87	Checksum (2 bytes)

Reply 33h:

Address	Function code	Lenght	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 33		Calls function 33h (1 byte)
Length		0x 15		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	UDC	0x 01 90	74,4 V	DC voltage (2 bytes) Scale factor 0.186 see below
	RES	0x B4	reserved	Reserved for internal purposes (1 byte)
	UAC	0x B4	230V	AC voltage after software correction (1 byte) +32h
	RES	0x FA	reserved	Reserved for internal purposes (1 byte)
	FREQ	0x 13 88	50,00 Hz	Mains frequency in Hz (2 bytes) Scale factor 0.01
	PAC	0x 03 84	+90 W	Actual value of the AC power (2 bytes) Scale factor 0.1
	RES	0x FA	reserved	Reserved for internal purposes (1 byte)
	STATE	0x 03	MPPT active	Inverter status (1 Byte)
	RES	0x 00 00 00 …	Intern	Reserved, is not evaluated (10 bytes)
CRC		.. ..		Checksum (2 bytes)

Arguments used 33h:

Identifier	Typ	Format	Value range	Scale factor	Other
UDC	Uint16	0 V	0…65535	Volt=01: 0,028 V Volt=02: 0,054 V Volt=03: 0,054 V Volt=04: 0,079 V Volt=05: 0,108 V Volt=06: 0,142 V Volt=07: 0,186 V Volt=08: 0,244 V Volt=09: 0,357 V Volt=10: 0,479 V	Scale factor depends on the voltage type of the inverter, see return value VOLT (cmd 3Eh)
UAC	Uint8	0 V	50…305	1	Offset +32h 0x 00 = 50V
FREQ	Uint16	0 Hz	0…65535	0,01	
PAC	Int16 sign-magnitude	0 W	0…32767/-32768…-0	0,1	MSB=1 charge MSB=0 unload
State	Uint8		00: Init level 1 01: Init level 2 02: Init level 3 03: MPPT 04: Reset 05: Constant-voltage 06: Low Power Mode 07: Wind 08: Factory Mode 09: Battery 10: Battery Safety	1	

NOTE

The correct scale factor for UDC must be selected using the return value VOLT. VOLT can be queried with function code 3Eh (see below).

Request 3Eh:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 3E	Calls the 3Eh function (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x A0 44	Checksum (2 bytes)

Reply 3Eh:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 3E		Calls the 3Eh function (1 byte)
Length		0x 13		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	SNR	0x 30 39	12345	Serial number (2 bytes)
	TYP	0x 01	solar	Inverter type (1 byte)
	VOLT	0x 07	120 V	Voltage type (1 byte)
	PNOM	0x 06	600 W	Power class (1 byte)
	COSPHI	0x 78	0,90 (k)	Power factor cos φ (1 byte) Scale factor 0.005
	E	0x 0D 4D C3 00	6200 Wh	Daily total of the grid feed-in (4 bytes)
	PMAX	0x 07 6C	1900 W	Currently valid power limitation in W (2 bytes)
	OPH	0x 00 7E 57 C0	230 h	Total operating hours (4 bytes)
	PHASE	0x 02	2	Connection phase (1 byte)
	JMPR	0x 04	4	Modbus device ID set via jumper (1 byte)
	RES	0x 00	00	Reserved, is not evaluated (1 byte)
CRC		.. ..		Checksum (2 bytes)

Arguments used 3Eh:

Identifier	Typ	Format	Value range	Scale factor	Other
SNR	Uint16	0	0…65535	1	
TYP	Uint8	0	01 = SOL 02 = WIN 03 = BAT	1	
VOLT	Uint8	0	VOLT / SOL / WIN / BAT 01 / 12 V / 12 V / 8 V 02 / 17 V / 18 V / 12 V 03 / 35 V / 24 V / 24 V 04 / 50 V / 36 V / 30 V 05 / 70 V / 48 V / 36 V 06 / 90 V / 72 V / 48 V 07 / 120 V / 96 V / 96 V 08 / 160 V / 120 V / 120 V 09 / 240 V / 160 V / 160 V 10 / 320 V / 240 V / 240 V	1	Voltage type of the inverter depending on the area of application
PNOM	Uint8	0 W	01 … 32	100	
COSPHI	Uint16	0,00 (i/k)	0,9 (i)…1…0,9 (k)	1±(0x64-COSPHI)*0,005	0x 64 = 1 0x 50 = 0,9 (i) 0x 78 = 0,9 (k)
E	Uint32	0,0 Wh	0…4294967295	1/36000	
PMAX	Uint16	0 W	0…32767	0,1	
OPH	Uint32	0 h	0…4294967295	1/36000	
PHASE	Uint8	0	0…3	1	
JMPR	Uint8	0	0…15	1	

NOTE

The exact model designation of the inverter is made up of the return values "TYPE", "PNOM" and "VOLT". Example: SOL 1900-70
TYPE [01] = SOL
PNOM [19] = 1900
VOLT [06] = 70

4.3 Switching off the inverter (cmd 34h/35h)

You can switch off the inverter by switching off the H-bridge and disconnecting the inverter from the household grid via the integrated relays.

Request 34h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 34	Calls function 34h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 80 46	Checksum (2 bytes)

The inverter is now in the INIT (00) operating state. To resume operation you must reactivate the relay and H-bridge with function code 35h or reset the inverter manually (disconnect DC voltage). After disconnecting the DC voltage, the device automatically switches to the standard operating mode, if possible, depending on the device type.

Request 35h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 35	Calls function 35h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x D1 86	Checksum (2 bytes)

After switching on again with function code 35h, the appliance returns to the operating status that existed before it was switched off with function code 34h.

4.4 Force idling (cmd 36h/37h)

You can use function code 36h to set the PWM of the H-bridge to 0% load and thus bring the inverter to idle without affecting the operating status or the relay status.

Request 36h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 36	Calls the function 36h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 21 86	Checksum (2 bytes)

Then make sure that you deactivate the idle mode with 37h again:

Request 37h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 37	Calls function 37h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 70 46	Checksum (2 bytes)

4.5 Switching to constant voltage mode or controlling the voltage (cmd 38h/39h)

SOL and WIN devices can be set to constant voltage mode using function code 38h. The input voltage can be specified or set to 0 in order to use the internal default setting. The mode remains active until it is ended with function code 39h or the inverter is reset by disconnecting the DC voltage. If function code 38h is sent with a default of 0V, the device switches to constant voltage mode and uses the parameter U_SOLCV as the voltage default.

With BatteryInvert, function code 38h is used to send the voltage specification to the inverter in modes 0 and 3. If a voltage specification of 0V is sent to the BatteryInvert, the inverter regulates solely to the power sent with function code 3Fh. If this is also 0, the BatteryInvert regulates in mode 0 to the parameter U_SOLBAT, in mode 3 it switches to safety mode. If U_SOLBAT=0V, the device also switches to safety mode in mode 0. In mode 3, the timeout is also reset when 38h is received.

Request 38h:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 38		Calls function 38h (1 byte)
Length		0x 02		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	USET	0x 02 E8	74,4 V	DC voltage (2 bytes) Scale factor 10
CRC		0x B4 4E		Checksum (2 bytes)

Arguments used 38h:

Identifier	Typ	Format	Value range	Scale factor	Other
USET	Uint16	0 V	0…65535	10	

To end the constant voltage mode for SOL and WIN, use function 39h. With BatteryInvert, function code 39h has no function. If a voltage specification is to be canceled here without replacing it with a new value, a default value of 0V must be sent.

Request 39h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 39	Calls function 39h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x 11 85	Checksum (2 bytes)

4.6 Defining the power factor cos φ (cmd 3Bh)

You can specify a set value for the power factor cos φ for the inverter.

Request 3Bh:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 3B		Calls the 3Bh function (1 byte)
Length		0x 01		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	COSPHI	0x 78	0,90 (k)	Power factor cos φ (1 byte) Scale factor 0.005
CRC		0x 71 A7		Checksum (2 bytes)

Arguments used 3Bh:

Identifier	Typ	Format	Value range	Scale factor	Other
COSPHI	Uint8	0,00 (i/k)	0,9 (i)…1…0,9 (k)	1±(0x64-COSPHI)*0,005	0x 64 = 1 0x 50 = 0,9 (i) 0x 78 = 0,9 (k)

This specification for the power factor is not saved and is therefore only valid until the inverter is re-started.

4.7 Defining the inverter power (cmd 3Fh/44h)

Depending on the device type, you have two different options for controlling or limiting the power of the inverter.

NOTE

The inverter moves to this point immediately (if possible). For battery inverters, a positive value means discharging the battery (storage), while a negative value means charging (mains supply).

4.7.1 Controlling/limiting the power with BatteryInvert in mode 0 and 3

The desired power can be specified for the BatteryInvert in mode 0 or 3 with the function code 3Fh. The Umin and Umax commands are also sent, which can be used to further restrict the voltage window (U_MIN...U_MAX) set on delivery.

The value for the timeout is required in mode 3 to initialize the timeout function. This means that even if the BatteryInvert is only to be controlled via the voltage (with 38h), 3Fh must be sent once when switching on. If the power is not to be limited, PMAX must be sent with 0 or a value above the maximum power of the inverter. The time-out is reset each time a message with 38h or 3Fh is received. This means that in mode 3, a voltage or power specification must be sent regularly within the interval sent with 3Fh.

Request 3Fh:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 3F		Calls the 3Fh function (1 byte)
Length		0x 0D		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	PMAX	0x 012c	300W	discharge 300W
	Umin	0x 016a	36,2V	
	Umax	0x 03e8	100,0V	
	Timeout	0x 0096	30s	30s
	reserviert	00 00 00 00 00		
CRC		0x BE D9		Checksum (2 Bytes)

Arguments used 3Fh:

Identifier	Typ	Format	Value range	Scale factor	Other
PMAX	Int16 sign-magnitude	0 W	0…3200	1,0	MSB=1: charge, MSB=0: unload
Umin	Uint16	0 V	U_MIN…Umax	10	
Umax	Uint16	0 V	Umin…U_MAX	10	
Timeout	Uint16	0 s	15s…12000s	5	

None of these values are saved, i.e. they are only valid until the next restart.

4.7.2 Power limitation of solar inverters

The desired power is specified relative to the PV generator power connected to the inverter (in %):

Request 44h:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 44		Calls function 44h (1 byte)
Length		0x 01		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	Pmax_rel	0x 46	70%	Max. Feed-in power in % (1 byte)
CRC		0x C1 AF		Checksum (2 Bytes)

Arguments used 44h:

Identifier	Typ	Format	Value range	Scale factor	Other
Pmax_rel	Uint8	0 %	0…100	1	0

Pmax_rel is saved in the inverter, i.e. the limitation is valid until it is overwritten with a new value and also applies after a restart. To deactivate the limitation, it is simply set to 100% again.

4.8 Reading out the error logbook (cmd 40h)

SolarInvert inverters have counters that provide information about the number of certain types of faults during their service life. These counters can be read but not manipulated.

Request 40h:

Address	Function code	Length	Data	CRC

Address	0x 01	Device ID of the addressed slave (1 byte)
Function code	0x 40	Calls the function 40h (1 byte)
Length	0x 01	Number of bytes in the data block (1 byte)
Data	0x 01	No data (1 byte)
CRC	0x C0 5C	Checksum (2 bytes)

Reply 40h:

Address	Function code	Length	Data	CRC

Address		0x 01		Device ID of the addressed slave (1 byte)
Function code		0x 40		Calls the function 40h (1 byte)
Length		0x 28 ab v60 0x 2E		Number of bytes in the data block (1 byte)
Data	Identifier	Value	Plain text	
	ACMX	0x 00 03	3	AC overvoltage (2 bytes)
	ACMN	0x 00 03	3	Undervoltage AC (2 bytes)
	DCMX	0x 00 10	16	Overvoltage DC (2 bytes)
	DCMN	0x 01 00	256	DC undervoltage (2 bytes)
	FMX	0x 02 00	512	Overfrequency (2 bytes)
	FMN	0x 01 00	256	Underfrequency (2 bytes)
	OT1	0x 00 F0	15	Temperature dep. power red. 87.5% (2 bytes)
	OT2	0x 00 F0	15	Temperature dev. power red. 75% (2 bytes)
	OT3	0x 00 F0	15	Temperature dev. power red. 62.5% (2 bytes)
	OT4	0x 00 F0	15	Temperature dev. power red. 50% (2 bytes)
	OT5	0x 00 F0	15	Temperature dev. power red. 37.5% (2 bytes)
	COM	0x 00 00	0	not implemented/as of v.60 Error communication between control board and Grid Inspector (2 bytes)
	ISL	0x 00 F0	15	Power failure (2 bytes)
	RES	0x 00 00	0	reserved, for internal purposes (2 bytes)
	RES	0x 00 00	0	reserved, from v60 Undervoltage AC delayed (2 bytes)
	OVRCURR	0x 00 00	0	Over current interrupt triggered (2 bytes)
	RES	0x 00 00	0	reserved, for internal purposes (2 bytes)

	OVRVOLT	0x 00 00	0	Overvoltage interrupt triggered (2 bytes)
	AVRG	0x 00 00	0	10min average mains voltage too high (2 bytes)
	OVRHT	0x 00 00	0	Overtemperature shutdown (2 bytes)
	UACDIF	0x 00 00	0	Mains voltage measurement deviation (2 bytes), only from v60
	RELFLT	0x 00 00	0	Mains relay error (2 bytes), only from v60
	CURFLT	0x 00 00	0	Current sensor error (2 bytes), only from v60
CRC		.. ..		Checksum (2 Bytes)

Arguments used 40h:

Identifier	Typ	Format	Value range	Scale factor	Other
ACMX	Uint16	0	0…65535	1	0
ACMN	Uint16	0	0…65535	1	0
DCMX	Uint16	0	0…65535	1	0
DCMN	Uint16	0	0…65535	1	0
FMX	Uint16	0	0…65535	1	0
FMN	Uint16	0	0…65535	1	0
OT1	Uint16	0	0…65535	1	0
OT2	Uint16	0	0…65535	1	0
OT3	Uint16	0	0…65535	1	0
OT4	Uint16	0	0…65535	1	0
OT5	Uint16	0	0…65535	1	0
COM	Uint16	0	0…65535	1	0
ISL	Uint16	0	0…65535	1	0
RES	Uint16	0	0…65535	1	0
RES	Uint16	0	0…65535	1	0
OVRCURR	Uint16	0	0…65535	1	0
RES	Uint16	0	0…65535	1	0
OVRVOLT	Uint16	0	0…65535	1	0
AVRG	Uint16	0	0…65535	1	0
UACDIF	Uint16	0	0…65535	1	from v60
RELFLT	Uint16	0	0…65535	1	from v60
CURFLT	Uint16	0	0…65535	1	from v60

5 Customer service and warranty

Accessibility

Should a fault occur with one of our products, we will of course be on hand to help and advise you.

SolarInvert GmbH | Customer | Monreposstr. 49 | D-71634 Ludwigsburg
Tel. 07141/299 21 13 | Fax 07141/299 21 21 | info@solarinvert.de

You can reach our customer service by telephone during our business hours.

NOTE

The warranty conditions and technical data can be found at:
www.solarinvert.de