SOLAR ELECTRIC

KOSTAL

Smart connections.

KOSTAL Interface MODBUS (TCP) & SunSpec

PIKO IQ / PLENTICORE plus

Version

Release date: 2019-05-02

Revision version: 1.8

Inhalt

1. Introduction

1.1 MODBUS Protocol

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, which provides client/server communication between devices connected on different types of buses or networks.

The industry's serial de facto standard since 1979, MODBUS continues to enable millions of automation devices to communicate. Today, support for the simple and elegant structure of MODBUS continues to grow. The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.

MODBUS is a request/reply protocol and offers services specified by function codes. MODBUS function codes are elements of MODBUS request/reply PDUs. The objective of this document is to describe the function codes used within the framework of MODBUS

transactions.

The MODBUS Application Protocol is currently used in the solar sector mainly for system communication in PV power plants. The MODBUS protocol has been developed for reading data from- or writing data to clearly defined data areas.

1.2 Data Formats

The following data formats describe how data is to be interpreted. The data formats are important, for example, for the display of data or for its further processing. The data formats are listed in the Format column of the assignment tables.

U16	An unsigned word (16-bit).
U32	An unsigned double word (32-bit).
S16	A signed word (16-bit).
S32	A signed double word (32-bit).
MBD	Multiple bytes data.

2. MODBUS protocol description

2.1 Application Layer

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, which provides client/server communication between devices connected on different types of buses or networks.

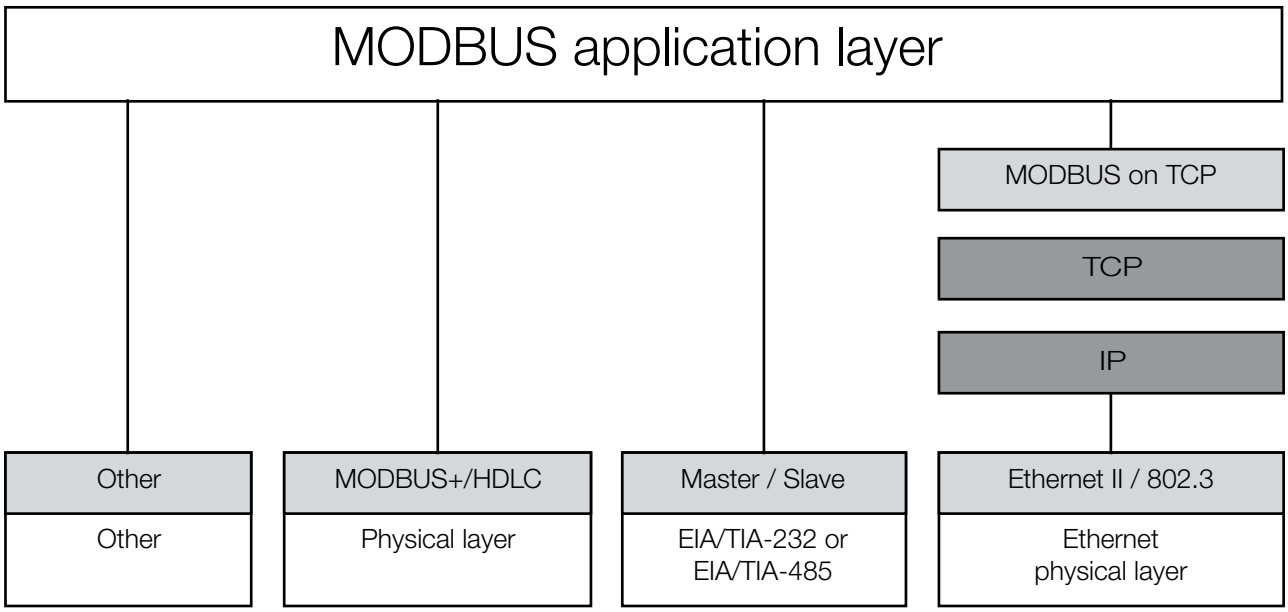

Abb. 1: MODBUS communication stack

MODBUS is an application layer messaging protocol for client/server communication between devices connected on different types of buses or networks.

Scope of this document is the implementation TCP/IP over Ethernet. See MODBUS Messaging Implementation Guide V1.0a.

2.1.1 MODBUS frame

The MODBUS protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers. The mapping of MODBUS protocol on specific buses or network can introduce some additional fields on the application data unit (ADU).

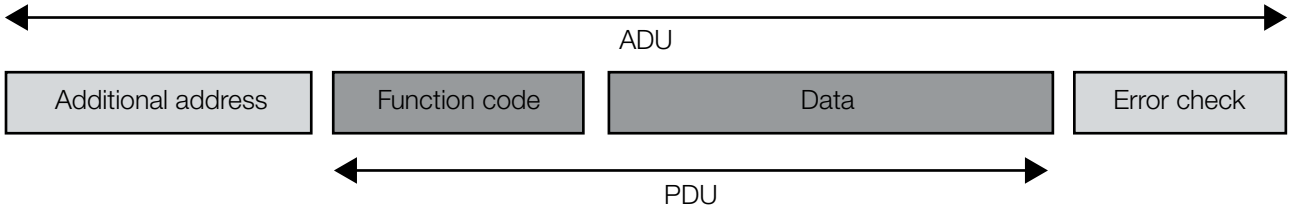

Abb. 2: General MODBUS frame

Abbreviations:

ADU	Application Data Unit
HDLC	High Level Data Link Control
HMI	Human Machine Interface
IETF	Internet Engineering Task Force
I/O	input/Output
IP	Internet Protokoll
MAC	Media Access Control
MB	MODBUS Protokol
MBAP	MODBUS Apllication Protokol
PDU	Protocol Data Unit
PLC	Programmable Logic Control
TCP	Transmission Control Protocol

The MODBUS application data unit is built by the client that initiates a MODBUS transaction. The function indicates to the server what kind of action to perform.

The function code field of a MODBUS data unit is coded in one byte. Valid codes are in the range of

1 ... 255 decimal (the range 128 – 255 is reserved and used for exception responses). When a message is sent from a Client to a Server device the function code field tells the server what kind of action to perform. Function code “0” is not valid.

2.1.2 Data Encoding

MODBUS uses a ‘big-Endian’ representation for addresses and data items. This means that when a numerical quantity larger than a single byte is transmitted, the most significant byte is sent first. So for example

Register size	value	
16 - bits	0x1234	the first byte sent is 0x12, then 0x34

2.1.3 Function code list

The following MODBUS commands are supported by the implemented MODBUS interface:

MODBUS command	Function code	Quantity of Registers[1]
Read Holding Registers	0x03	1 to 125
Write Single Register	0x06	1

[1] Register content is 16-bits width.

2.1.4 Read Holding Registers (0x03)

This function code is used to read the contents of a contiguous block of holding registers in the inverter. The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore registers numbered1-16 are addressed as 0-15.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	0x03
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	0x03
Byte count	1 Bytes	0x0000 to 0xFFFF
Register value	N[1] x 2 Bytes	1 to 125 (0x7D)

[1]N = Quantity of Registers

Error

Error code	1 Byte	0x83
Exception code	1 Bytes	01 or 02 or 03 or 04

Here is an example of a request to read registers 108 – 110:

Request		Response	
Field Name	(hex)	Field Name	(hex)
Function	03	Function	03
Starting Address Hi	00	Byte Count	06
Starting Address Low	6B	Register value Hi (108)	02
No. of Registers Hi	00	Register value Low (108)	2B
No. of Registers Low	03	Register value Hi (109)	00
		Register value Low (109)	00
		Register value Hi (110)	00
		Register value Low (110)	64

The contents of register 108 are shown as the two byte values of 0x022B. The contents of registers 109 –110 are 0x0000 and 0x0064.

2.1.5 Write Single Register (0x06)

This function code is used to write a single holding register in the inverter.

The Request PDU specifies the address of the register to be written.

The normal response is an echo of the request, returned after the register contents have been written.

Request

Function code	1 Byte	0x06
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	0x0000 to 0xFFFF

Response

Function code	1 Byte	0x06
Byte count	1 Bytes	0x0000 to 0xFFFF
Register value	2 Bytes	0x0000 to 0xFFFF

Error

Error code	1 Byte	0x86
Exception code	1 Bytes	01 or 02 or 03 or 04

Here is an example of a request to write register 2 to 0x0003:

Request		Response	
Field Name	(hex)	Field Name	(hex)
Function	06	Function	06
Register Address Hi	00	Register Address Hi	00
Register Address Low	01	Register Address Low	01
Register Value Hi	00	Register Value Hi	00
Register Value Low	03	Register Value Low	03

2.1.6 Exception Responses

When a client device sends a request to a server device it expects a normal response. One of four possible events can occur from the client's query:

- If the server device receives the request without a communication error, and can handle the query normally, it returns a normal response.
- If the server does not receive the request due to a communication error, no response is returned. The client program will eventually process a timeout condition for the request.
- If the server receives the request, but detects a communication error (parity, CRC...), no response is returned. The client program will eventually process a timeout condition for the request.
- If the server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non –existent output or register), the server will return an exception response informing the client of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most – significant bit (MSB) of 0 (their values are all below 80 hex). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hex higher than the value would be for a normal response.

With the function code's MSB set, the client's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

The exception codes are listed:

MODBUS Exception Codes

Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is un-configured and is being asked to return register values.

MODBUS Exception Codes		
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (address -wise at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 “Illegal Data Address” since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	SERVER DEVICE FAILURE	An unrecoverable error occurred while the server was attempting to perform the requested action.
05	ACKNOW LEDGE	Specialized use in conjunction with programming commands.
06	SERVER DEVICE BUSY	Specialized use in conjunction with programming commands. The server is engaged in processing a long – duration program command. The client should retransmit the message later when the server is free.
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server attempted to read record file, but detected a parity error in the memory. The client can retry the request, but service may be required on the server device.
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

2.2 Data Link Layer

2.2.1 Overview

The MODBUS TCP protocol is used in this interface.

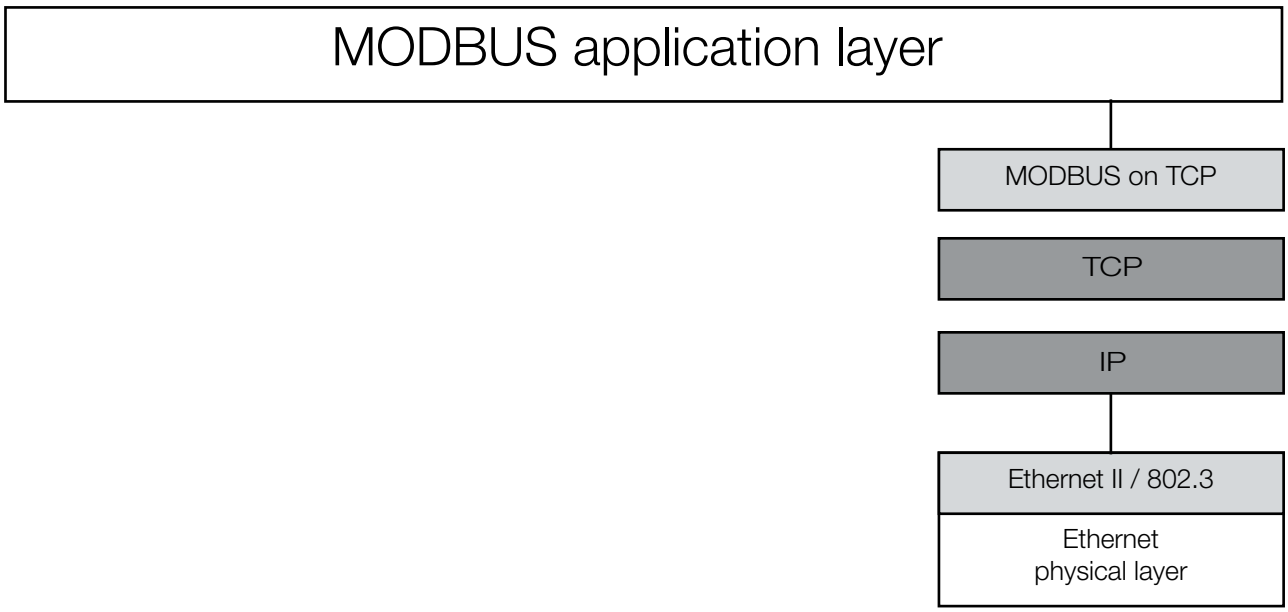

Abb. 3: MODBUS Protocols and ISO/OSI Model

Layer	ISO/OSI Layer	
7	Application	MODBUS / TCP
6	Presentation	MODBUS / TCP
5	Session	MODBUS / TCP
4	Transport	TCP
3	Network	IP
2	Data Link	IEEE 802.3 (Ethernet)
1	Physical	IEEE 802.3 (Ethernet)

2.3 Physical Layer

2.3.1 Ethernet port

A electrical interface in accordance with IEEE 802.3 standard is used for the interface. A RJ45 connector is used for connection.

2.3.2 Electrical interface

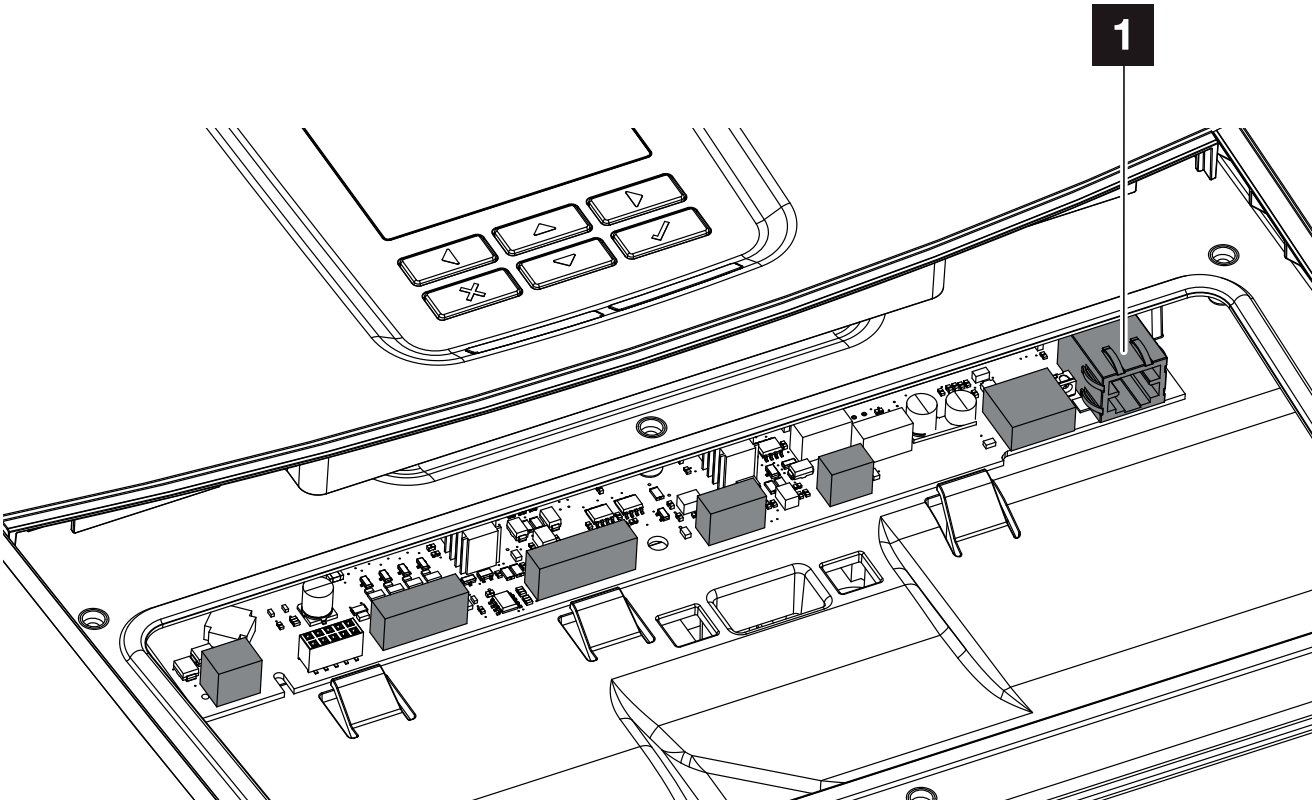

Abb. 4: Smart Communication Board - Interface

Item	Designation	terminal	Explanation
1	Ethernet connection (RJ45)	X206	RJ45 max. 100 Mbit (LAN connection for linking to a router, for example)

3. MODBUS Register table

3.1 TCP-Port and Unit-ID

To access the inverter via MODBUS / TCP, the following TCP-Port and MODBUS-Unit-ID are used as default values:

TCP-Port	1502 (dec)
Unit-ID[1]	71 (dec)

[1] The Unit-ID is modifiable.

3.2 Query the operating data

Addr (hex)	Addr (dec)	Description	Unit	Format	N[1)]	Access	Function Code
0x02	2	MODBUS Enable	-	Bool	1	R/W	0x03/0x06
0x04	4	MODBUS Unit-ID	-	U16	1	R/W	0x03/0x06
0x06	6	Inverter article number	-	String	8	RO	0x03
0x0E	14	Inverter serial number	-	String	8	RO	0x03
0x1E	30	Number of bidirectional converter	-	U16	1	RO	0x03
0x20	32	Number of AC phases	-	U16	1	RO	0x03
0x22	34	Number of PV strings	-	U16	1	RO	0x03
0x24	36	Hardware-Version	-	U16	2	RO	0x03
0x26	38	Software-Version Maincontroller (MC)	-	String	8	RO	0x03
0x2E	46	Software-Version IO-Controller (IOC)	-	String	8	RO	0x03
0x36	54	Power-ID	-	U16	2	RO	0x03
0x38	56	Inverter state[2]	-	U16	2	RO	0x03
0x64	100	Total DC power	W	Float	2	RO	0x03
0x68	104	State of energy manager[3]	-	U32	2	RO	0x03
0x6A	106	Home own consumption from battery	W	Float	2	RO	0x03
0x6C	108	Home own consumption from grid	W	Float	2	RO	0x03
0x6E	110	Total home consumption Battery	Wh	Float	2	RO	0x03
0x70	112	Total home consumption Grid	Wh	Float	2	RO	0x03
0x72	114	Total home consumption PV	Wh	Float	2	RO	0x03
0x74	116	Home own consumption from PV	W	Float	2	RO	0x03
0x76	118	Total home consumption	Wh	Float	2	RO	0x03
0x78	120	Isolation resistance	Ohm	Float	2	RO	0x03
0x7A	122	Power limit from EVU	%	Float	2	RO	0x03
0x7C	124	Total home consumption rate	%	Float	2	RO	0x03
0x90	144	Worktime	s	Float	2	RO	0x03
0x96	150	Actual cos ϕ	-	Float	2	RO	0x03
0x98	152	Grid frequency	Hz	Float	2	RO	0x03

Addr (hex)	Addr (dec)	Description	Unit	Format	N[1)]	Access	Function Code
0x9A	154	Current Phase 1	A	Float	2	RO	0x03
0x9C	156	Active power Phase 1	W	Float	2	RO	0x03
0x9E	158	Voltage Phase 1	V	Float	2	RO	0x03
0xA0	160	Current Phase 2	A	Float	2	RO	0x03
0xA2	162	Active power Phase 2	W	Float	2	RO	0x03
0xA4	164	Voltage Phase 2	V	Float	2	RO	0x03
0xA6	166	Current Phase 3	A	Float	2	RO	0x03
0xA8	168	Active power Phase 3	W	Float	2	RO	0x03
0xAA	170	Voltage Phase 3	V	Float	2	RO	0x03
0xAC	172	Total AC active power	W	Float	2	RO	0x03
0xAE	174	Total AC reactive power	Var	Float	2	RO	0x03
0xB2	178	Total AC apparent power	VA	Float	2	RO	0x03
0xBE	190	Battery charge current	A	Float	2	RO	0x03
0xC2	194	Number of battery cycles	-	Float	2	RO	0x03
0xC8	200	Actual battery charge (-) / discharge (+) current	A	Float	2	RO	0x03
0xCA	202	PSSB fuse state[5]	-	Float	2	RO	0x03
0xD0	208	Battery ready flag	-	Float	2	RO	0x03
0xD2	210	Act. state of charge	%	Float	2	RO	0x03
0xD6	214	Battery temperature	°C	Float	2	RO	0x03
0xD8	216	Battery voltage	V	Float	2	RO	0x03
0xDA	218	Cos ϕ (powermeter)	-	Float	2	RO	0x03
0xDC	220	Frequency (powermeter)	Hz	Float	2	RO	0x03
0xDE	222	Current phase 1 (powermeter)	A	Float	2	RO	0x03
0xE0	224	Active power phase 1 (powermeter)	W	Float	2	RO	0x03
0xE2	226	Reactive power phase 1 (powermeter)	Var	Float	2	RO	0x03
0xE4	228	Apparent power phase 1 (powermeter)	VA	Float	2	RO	0x03
0xE6	230	Voltage phase 1 (powermeter)	V	Float	2	RO	0x03
0xE8	232	Current phase 2 (powermeter)	A	Float	2	RO	0x03
0xEA	234	Active power phase 2 (powermeter)	W	Float	2	RO	0x03
0xEC	236	Reactive power phase 2 (powermeter)	Var	Float	2	RO	0x03
0xEE	238	Apparent power phase 2 (powermeter)	VA	Float	2	RO	0x03
0xF0	240	Voltage phase 2 (powermeter)	V	Float	2	RO	0x03
0xF2	242	Current phase 3 (powermeter)	A	Float	2	RO	0x03
0xF4	244	Active power phase 3 (powermeter)	W	Float	2	RO	0x03
0xF6	246	Reactive power phase 3 (powermeter)	Var	Float	2	RO	0x03
0xF8	248	Apparent power phase 3 (powermeter)	VA	Float	2	RO	0x03
0xFA	250	Voltage phase 3 (powermeter)	V	Float	2	RO	0x03

Addr (hex)	Addr (dec)	Description	Unit	Format	N[1)]	Access	Function Code
0xFC	252	Total active power (powermeter) Sensor position 1 (home consumption): (+) House consumption, (-) generation Sensor position 2 (grid connection): (+) Power supply, (-) feed-in	W	Float	2	RO	0x03
0xFE	254	Total reactive power (powermeter) Sensor position 2 (grid connection): (+) Power supply, (-) feed-in Sensor position 1 (home consumption): (+) House consumption, (-) generation	Var	Float	2	RO	0x03
0x100	256	Total apparent power (powermeter) Sensor position 2 (grid connection): (+) Power supply, (-) feed-in Sensor position 1 (home consumption): (+) House consumption, (-) generation	VA	Float	2	RO	0x03
0x102	258	Current DC1	A	Float	2	RO	0x03
0x104	260	Power DC1	W	Float	2	RO	0x03
0x10A	266	Voltage DC1	V	Float	2	RO	0x03
0x10C	268	Current DC2	A	Float	2	RO	0x03
0x10E	270	Power DC2	W	Float	2	RO	0x03
0x114	276	Voltage DC2	V	Float	2	RO	0x03
0x116	278	Current DC3	A	Float	2	RO	0x03
0x118	280	Power DC3	W	Float	2	RO	0x03
0x11E	286	Voltage DC3	V	Float	2	RO	0x03
0x140	320	Total yield	Wh	Float	2	RO	0x03
0x142	322	Daily yield	Wh	Float	2	RO	0x03
0x144	324	Yearly yield	Wh	Float	2	RO	0x03
0x146	326	Monthly yield	Wh	Float	2	RO	0x03
0x180	384	Inverter network name	-	String	32	RO	0x03
0x1A0	416	IP enable	-	U16	1	RO	0x03
0x1A2	418	Manual IP / Auto-IP	-	U16	1	RO	0x03
0x1A4	420	IP-address	-	String	8	RO	0x03
0x1AC	428	IP-subnetmask	-	String	8	RO	0x03
0x1B4	436	IP-gateway	-	String	8	RO	0x03
0x1BC	444	IP-auto-DNS	-	U16	1	RO	0x03
0x1BE	446	IP-DNS1	-	String	8	RO	0x03
0x1C6	454	IP-DNS2	-	String	8	RO	0x03
0x200	512	Battery gross capacity	Ah	U32	2	RO	0x03
0x202	514	Battery actual SOC	%	U16	1	RO	0x03
0x203	515	Firmware Maincontroller (MC)	-	U32	2	RO	0x03
0x205	517	Battery Manufacturer	-	String	8	RO	0x03

Addr (hex)	Addr (dec)	Description	Unit	Format	N[1)]	Access	Function Code
0x20D	525	Battery Model ID	-	U32	2	RO	0x03
0x20F	527	Battery Serial Number	-	U32	2	RO	0x03
0x211	529	Work Capacity	Wh	U32	2	RO	0x03
0x213	531	Inverter Max Power	W	U16	1	RO	0x03
0x214	532	Inverter Peak Generation Power Scale Factor[4]	-	-	1	RO	0x03
0x217	535	Inverter Manufacturer	-	String	16	RO	0x03
0x227	551	Inverter Model ID	-	String	8	RO	0x03
0x22F	559	Inverter Serial Number	-	String	16	RO	0x03
0x23F	575	Inverter Generation Power (actual)	W	S16	1	RO	0x03
0x240	576	Power Scale Factor[4]	-	-	1	RO	0x03
0x241	577	Generation Energy	Wh	U32	2	RO	0x03
0x243	579	Energy Scale Factor[4]	-	-	1	RO	0x03
0x246	582	Actual battery charge/discharge power	W	S16	1	RO	0x03
0x24A	586	Battery Firmware	-	U32	1	RO	0x03
0x24C	588	Battery Type[6]	-	U16	1	RO	0x03
0x300	768	Productname (e.g. PLENTICORE plus)	-	String	32	RO	0x03
0x320	800	Power class (e.g. 10)	-	String	32	RO	0x03

Notes:

[1] N = Quantity of Registers

[2] Inverter States

0	Off
1	Init
2	IsoMeas
3	GridCheck
4	StartUp
5	-
6	FeedIn
7	Throttled
8	ExtSwitchOff
9	Update
10	Standby
11	GridSync
12	GridPreCheck
13	GridSwitchOff
14	Overheating
15	Shutdown
16	ImproperDcVoltage
17	ESB
18	Unknown

[3] States of energy manager

0x00	Idle
0x01	n/a
0x02	Emergency Battery Charge
0x04	n/a
0x08	Winter Mode Step 1
0x10	Winter Mode Step 2

[4] Scale factors: As an alternative to floating point format, values are represented by integer values with a signed scale factor applied. The scale factor explicitly shifts the decimal point to the left (negative value) or the right (positive value). Scale factors are 16 bit two's compliment integer, the signed range is -10 … 10.

[5] PSSB-fuse-state

0x00	Fuse fail
0x01	Fuse ok
0xFF	Unchecked

[6] Battery type

0x00	No battery (PV-functionality)
0x02	Li-Io battery SONY / MURATA
0x04	Li-Io battery BYD / BBOX

4. SunSpec Interface

4.1 Overview

Information in SunSpec is defined through a set of 'Information Models' representing functionality implemented by devices or plants. SunSpec Alliance Interoperability Specifications describe these information models, data exchange formats and communication protocols used in distributed energy resource systems.

SunSpec information Models are defined using the SunSpec Model Definition XML (SMDX) encoding. Please reference the SMDX file for the definitive version of any SunSpec Information Model, at http://sunspec.org/download.

SunSpec information Models are communication protocol agnostic, but MODBUS is currently the most popular transport protocol in use.

For further information refer to www.sunspec.org.

4.2 Implemented SunSpec Models

Currently the following SunSpec-Models are implemented:

Model-No	Model-Name
1	Common
103	Three Phase Inverter
113	Three Phase Inverter, float
120	Nameplate
123	Immediate Controls
160	Multiple MPPT
802	Battery Base Model

4.3 Startaddresses

Model-No	Startaddress (dec)
1	40003
103	40071
113	40123
120	40185
123	40213
160	40239
802	40309

KOSTAL

KOSTAL Solar Electric GmbH
Hanferstr. 6
79108 Freiburg i. Br.
Deutschland
Telefon: +49 761 47744 - 100
Fax: +49 761 47744 - 111

KOSTAL Solar Electric Ibérica S.L.
Edificio abm
Ronda Narciso Monturiol y Estarriol, 3
Torre B, despachos 2 y 3
Parque Tecnológico de Valencia
46980 Valencia
España
Teléfono: +34 961 824 - 934
Fax: +34 961 824 - 931

KOSTAL Solar Electric France SARL
11, rue Jacques Cartier
78280 Guyancourt
France
Téléphone: +33 1 61 38 - 4117
Fax: +33 1 61 38 - 3940

KOSTAL Solar Electric Hellas E.Π.E.
47 Steliou Kazantzidi st., P.O. Box: 60080
1st building – 2nd entrance
55535, Pilea, Thessaloniki
Ελλάδα
Τηλέφωνο: +30 2310 477 - 550
Φαξ: +30 2310 477 - 551

KOSTAL Solar Electric Italia Srl
Via Genova, 57
10098 Rivoli (TO)
Italia
Telefono: +39 011 97 82 - 420
Fax: +39 011 97 82 - 432

KOSTAL Solar Electric Turkey
Mahmutbey Mah. Taşocağı Yolu
No:3 (B Blok), Ağaoğlu My Office212,
Kat:16, Ofis No: 269
Bağcılar - İstanbul / Türkiye
Telefon: +90 212 803 06 24
Faks: +90 212 803 06 25

www.kostal-solar-electric.com

05 / 2019 - EN - DOC02308911-0003 - Technische Änderungen und Irrtümer vorbehalten.
Aktuelle Informationen finden Sie unter www.kostal-solar-electric.com. Hersteller: KOSTAL Industrie Elektrik GmbH, Hagen, Deutschland