

Modbus-TCP Schnittstelle

SAX Power Home 5, SAX Power Home 7

Die Steuerung über Modbus-TCP bietet die Möglichkeit, verschiedene Parameter für andere Anwendungen nutzbar zu machen und eine Fernwirkung auf die Speicher zu realisieren. Hierbei kann auch das mit dem SAX Home ausgelieferte Smartmeter umfänglich ausgelesen werden, was die Verwendung eines weiteren Zählers für Steuer- und Regelanwendungen verschiedener Systeme häufig überflüssig macht.

Sollte kein SAX-Smartmeter verwendet werden, kann die Sollwertsteuerung des Speichers über Modbus-TCP übernommen werden.

Es stehen für jedes Gerät zwei Modbus-Entitäten zur Verfügung, welche parallel verwendet werden können. Der Server steht in beiden Fällen auf **Port 502** zur Verfügung.

Die Anwendung der Schnittstelle soll im Folgenden erläutert werden.

Modbus Mode 1 – Simple

In dieser Entität stehen grundlegende Überwachungsparameter zur Verfügung.

Hierbei ist die Slave-ID dieser Entität **64_{Dec}** bzw. **0x40_{Hex}**. Alle Werte als Integer.

Wird der Speicher ohne Smartmeter verwendet, so stehen nach Freischaltung durch den Kundendienst von SAX Power folgende Register zur Verfügung:

Register Dezimal	Register Hexadezimal	Länge in Bytes	Beschreibung	Eigenschaft	Hinweis
41	0x29	2	Sollwert Leistung P	nur schreiben	Negativ: Laden, Positiv: Entladen
42	0x2a	2	Sollwert cos(phi)	nur schreiben	Faktor 1000: 950 → cos(phi) = 0,95

Mit Smartmeter stehen nach Freischaltung durch den Kundenservice folgende Register zur Verfügung:

Register Dezimal	Register Hexadezimal	Länge in Bytes	Beschreibung	Eigenschaft	Hinweis
43	0x2b	2	Leistungsgrenzwert für Entladung	nur lesen	Freischaltung Kundenservice
44	0x2c	2	Leistungsgrenzwert für Ladung	nur lesen	Freischaltung Kundenservice
45	0x2d	2	Schaltzustand Speicher	lesen/schreiben	Lesen: 01 „Aus“, 02 „Ein“, 03 „Verbund“ Schreiben: 01 „Aus“, 02 „Ein“

Lesbare Register, welche Informationen über das SAX Power Speichersystem beinhalten, sind folgende:

Register Dezimal	Register Hexadezimal	Länge in Bytes	Beschreibung	Eigenschaft	Hinweis
46	0x2e	2	SOC Speicher	nur lesen	In Prozent
47	0x2f	2	Messwert Speicherleistung	nur lesen	Offset: 16384
48	0x30	2	Messwert Smartmeter Leistung	nur lesen	Offset: 16384

Modbus Mode 2 – SunSpec

In dieser Entität steht eine an SunSpec angelehnte Registertabelle zur Verfügung, über welche detailliertere Werte des Speichers und des verbundenen Smartmeters ausgelesen werden können.

Hierbei ist die Slave-ID dieser Entität 40_{Dec} bzw. $0x28_{Hex}$

Address Start / End	Size	R/W	Name	Type	Units	SF	Description	Value Range	
40071	40071	1	R	SunSpec ID	uint16	-	-	Uniquely identifies this as a SunSpec Inverter (Three Phase) Model	103 (dec)
40072	40072	1	R	SunSpec Length	uint16	-	-	Well-known # of 16 bit registers to follow : 50	50 (dec)
40073	40073	1	R	Amps	uint16	A	A_SF	Sum of active phases Current	-
40074	40074	1	R	Amps PhaseA	uint16	A	A_SF	Phase A Current	-
40075	40075	1	R	Amps PhaseB	uint16	A	A_SF	Phase B Current	-
40076	40076	1	R	Amps PhaseC	uint16	A	A_SF	Phase C Current	-
40077	40077	1	R	A_SF	sunssf	-	-	Amps scale factor	-2
40078	40078	1	R	Phase Voltage AB	uint16	V	V_SF	Optional / not supported	
40079	40079	1	R	Phase Voltage BC	uint16	V	V_SF	Optional / not supported	
40080	40080	1	R	Phase Voltage CA	uint16	V	V_SF	Optional / not supported	
40081	40081	1	R	Phase Voltage AN	uint16	V	V_SF	Voltage phase A to N	-
40082	40082	1	R	Phase Voltage BN	uint16	V	V_SF	Voltage phase B to N	-
40083	40083	1	R	Phase Voltage CN	uint16	V	V_SF	Voltage phase C to N	-
40084	40084	1	R	V_SF	sunssf	-	-	Voltage scale factor	-1
40085	40085	1	R	Watts	int16	W	W_SF	Total AC Power	-
40086	40086	1	R	W_SF	sunssf	-	-	AC Power Scale Factor	1
40087	40087	1	R	Hz	uint16	Hz	Hz_SF	Line Frequency	-
40088	40088	1	R	Hz_SF	sunssf	-	-	Line Frequency Scale Factor	-1
40089	40089	1	R	VA	int16	VA	VA_SF	AC Apparent Power	-
40090	40090	1	R	VA_SF	sunssf	-	-	AC Apparent Power Scale Factor	1
40091	40091	1	R	VAr	int16	var	VAr_SF	AC Reactive Power	
40092	40092	1	R	VAr_SF	sunssf			AC Reactive Power Scale Factor	1
40093	40093	1	R	PF	int16	Pct	PF_SF	AC Power Factor	
40094	40094	1	R	PF_SF	sunssf			AC Power Factor Scale Factor	-1
40095	40095	1	R	SOH	uint16		Optional	SOH of speicher	
40096	40096	1	R	Energy produced	uint16	Wh	Energy factor	Toltal Energy produced	
40097	40097	1	R	Energy consumed	uint16	Wh	Energy factor	Total Energy consumed	
40098	40098	1	R	Energy factor	sunssf				1
40099	40099	1	R	Speicher state	uint16			Speicher state(on off standby)	
40100	40100	1	R	smartmeter Phase A current	int16	A		smartmeter Phase A current	Factor -2

40101	40101	1	R	smartmeter Phase B current	int16	A		smartmeter Phase B current	Factor -2
40102	40102	1	R	smartmeter Phase Ccurrent	int16	A		smartmeter Phase C current	Factor -2
40103	40103	1	R	Power phase A from smartmeter	int16	W		Power phase A(smartermeter)	
40104	40104	1	R	Power phase B from smartmeter	int16	W		Power phase B(smartermeter)	
40105	40105	1	R	Power phase C from smartmeter	int16	W		Power phase C(smartermeter)	
40106	40106	1	R	Power factor(smartermeter)	sunssf			Power factor(smartermeter)	1
40107	40107	1	R	smartmeter Phase A voltage	int16	V		smartmeter Phase A voltage	Factor -1
40108	40108	1	R	smartmeter Phase B voltage	int16	V		smartmeter Phase B voltage	Factor -1
40109	40109	1	R	smartmeter Phase C voltage	int16	V		smartmeter Phase C voltage	Factor -1
40110	40110	1	R	total Smartmeter Power	int16	W	PF(smartermeter)	Toltal smartermeter Power	
40111	40111	1	R	Power factor	sunssf	W	-	Smartermeter Power factor	1
40112	40112	1	R	PV power from Smartmeter	uint16	W	PV power factor	PV power from Smartermeter	
40113	40113	1	R	PV power factor	sunssf		-	PV Power factor	1
40114	40114	1	R	SOC	int16	PCT	-	SOC from speicher(Phase A)	-
40115	40115	1	R	capacity	uint16	Wh	-	capacity	Factor 1
40116	40116	1	R	cycles	uint16	-	-	Battery cycles	
40117	40117	1	R	Temperatur	int16	C	-	Battery Temperatur	

Modbus Lese- und Schreibfunktionen

Das Lesen der Parameter muss mit der **Funktion 03H : READ HOLDING REGISTERS** ausgeführt werden.

Schreibbare Parameter müssen mit der **Funktion 10H: PRESET MULTIPLE REGISTERS** beschrieben werden.

Beispielhafte Implementation in Python

Lesen

Eine Funktion zum Lesen von Werten in Python3 kann wie folgt implementiert werden:

```
## Read Registers from SAX Storage.
# IP is given as a string. Returns value as an integer.
def connectAndReadFromSax(ip,hex_adress):
    c = ModbusClient(host=ip, port=502, unit_id=64)
    try:
        c.open()
    except:
        return 65535
    value = c.read_holding_registers(hex_adress,1)
    if value is None:
        value1 = 65535
    else:
        value1 = value[0]
    c.close()
    #Subtract offset value for two's complement representation
    if(hex_adress == 0x30 or hex_adress == 0x2f):
        value1 = value1 - 16384
    return value1
```

Aufbauend können dann folgende Funktionen implementiert werden:

```
def getSOC(ip):
    soc = connectAndReadFromSax(ip, 46)
    return soc

def getGridPower(ip):
    p = connectAndReadFromSax(ip, 0x30)
    return p

def getStoragePower(ip):
    p = connectAndReadFromSax(ip, 0x2f)
    return p
```

Schreiben

Das Schreiben von Parametern kann beispielsweise über folgende Funktion implementiert werden:

```
## Set modbus register on SAX Storage to any integer value.
# IP is given as a string. Returns 1 if vaulue was written succesfully.
def connectAndWriteToSax(ip,hex_adress,value):
    #All SAX devices use the Device Identifier 0x40 and Port 3600 for TCP.
    c = ModbusClient(host=ip, port=502, unit_id=0x40)
    try:
        c.open()
    except:
        return 0

    #Use two's complement for negative representation
    if(value < 0):
        value = int((bin(value & 0xFFFF)[2:]),2)
    c.write_multiple_registers(hex_adress,[value])
    c.close()
    return 1
```

Aufbauend können dann folgende Funktionen implementiert werden:

```
def setStoragePowerLimit(ip,value):
    if(connectAndWriteToSax(ip,0x2b,value)):
        print(f'Storage power set to {value}W')

def setStoragePower(ip,value):
    if(value>0):
        term = 'discharging'
        direction = 'to'
    else:
        term = 'charging'
        direction = 'from'
    if(connectAndWriteToSax(ip,0x29,value)):
        print(f'Storage power set to {value}W, {term} {direction} Grid')
```

