Modbus TCP – Schnittstellenbeschreibung
Wallbox eM4 Controller / Extender

Inhalt

1. Konfiguration

- Die Kommunikation erfolgt über LAN / WLAN
- Wallbox eM4 ist Server / Slave
- Server Port: 502 (Standard Modbus TCP)

2. Aufbau Telegramm

2.1. Daten von Server lesen

2.1.1. Anfrage Client

Byte	Inhalt	Beschreibung
0	transaction identifier 1	Identifikations-Nr. bei mehreren gleichzeitig aktiven Anfragen
1	transaction identifier 2	
2	protocol identifier 1	0x00 (MODBUS)
3	protocol identifier 2	0x00 (MODBUS)
4	Anzahl nachfolgende Telegrambytes (high byte)	0x00 (immer 0x00, da max 255 Datenbytes)
5	Anzahl nachfolgende Telegrambytes (low byte)	0x06 (Byte 6...11)
6	unit identifier	0xFF (Identifikation erfolgt über IP-Adresse)
7	Function Code	0x03 (Lesen)
8	Adresse Startregister (high byte)	
9	Adresse Startregister (low byte)	
10	Anzahl der zu lesenden Register (high Byte)	0x00
11	Anzahl der zu lesenden Register (low Byte)	0x01 ... 0x7E (Server kann max. 126 Register liefern)

2.1.2. Antwort Server

Byte	Inhalt	Beschreibung
0	transaction identifier 1	Identifikations-Nr. der zugehörigen Anfrage
1	transaction identifier 2	
2	protocol identifier 1	0x00 (MODBUS)
3	protocol identifier 2	0x00 (MODBUS)
4	Anzahl nachfolgende Telegrambytes (high byte)	0x00 (immer 0x00, da max 255 Datenbytes)
5	Anzahl nachfolgende Telegrambytes (low byte)	0x05 ... 0xFF (Byte 6...n; min 5 Bytes, max 255 Bytes)
6	unit identifier	0xFF (Identifikation erfolgt über IP-Adresse)
7	Function Code	0x03 (Lesen)
8	Anzahl Datenbyte	0x02 ... 0xFC (min 2, max 252 Bytes)
9	Inhalt Adresse Startregister (high Byte)	
10	Inhalt Adresse Startregister (low Byte)	
11..n	weitere Register	max 125 weitere Register

2.2. Daten auf Server schreiben

2.2.1. Anfrage Client

Byte	Inhalt	Beschreibung
0	transaction identifier 1	Identifikations-Nr. bei mehreren gleichzeitig aktiven Anfragen
1	transaction identifier 2	
2	protocol identifier 1	0x00 (MODBUS)
3	protocol identifier 2	0x00 (MODBUS)
4	Anzahl nachfolgende Telegrambytes (high byte)	0x00 (immer 0x00, da max 255 Datenbytes)
5	Anzahl nachfolgende Telegrambytes (low byte)	0x09 ... 0xFF (Byte 6...n; min 9 Byte, max 255 Byte)
6	unit identifier	0xFF (Identifikation erfolgt über IP-Adresse)
7	Function Code	0x10 (Schreiben)
8	Adresse Startregister (high byte)	
9	Adresse Startregister (low byte)	
10	Anzahl der zu schreibenden Register (high Byte)	0x00

Byte	Inhalt	Beschreibung
11	Anzahl der zu schreibenden Register (low Byte)	0x01 ... 0x7C (min 1 Register, max 124 Register)
12	Anzahl der zu schreibenden Datenbyte	0x02 ... 0xF8 (min 2 Byte, max 248 Byte)
13	Datenbyte für Adresse Startregister (high Byte)	
14	Datenbyte für Adresse Startregister (low Byte)	
15...n	Datenbyte für weitere Register	

2.2.2. Antwort Server

Byte	Inhalt	Beschreibung
0	transaction identifier 1	Identifikations-Nr. der zugehörigen Anfrage
1	transaction identifier 2	
2	protocol identifier 1	0x00 (MODBUS)
3	protocol identifier 2	0x00 (MODBUS)
4	Anzahl nachfolgende Telegrambytes (high byte)	0x00 (immer 0x00, da max 255 Datenbytes)
5	Anzahl nachfolgende Telegrambytes (low byte)	0x06 (Byte 6...11)
6	unit identifier	0xFF (Identifikation erfolgt über IP-Adresse)
7	Function Code	0x10 (Schreiben)
8	Adresse Startregister (high byte)	
9	Adresse Startregister (low byte)	
10	Anzahl der geschriebenen Register (high Byte)	0x00
11	Anzahl der geschriebenen Register (low Byte)	0x01 ... 0x7C (Client kann max. 124 Register schreiben)

2.3. Fehler (Server)

Fehlercode	Beschreibung
0x01	Verwendung eines nicht unterstützten Funktionscodes
0x02	Verwendung einer ungültigen Registeradresse
0x03	Verwendung unerlaubter Datenwerte, z.B. eine unerlaubte Anzahl Register
0x06	Server busy (max. Anzahl gleichzeitiger Transaktionen erreicht)

Die Fehlercodes werden nicht unterstützt. Im Fehlerfall sowie bei Übertragungsfehler erfolgt keine Antwort durch den Server.

3. Register

Parameters / Values whose content extends over more than one register are handled in big endian order (MSR first).

3.1. API-Adressraum Register

Start	Ende	Start	Ende	Inhalt
0x0001	0x00FF			Allgemeine Eigenschaften des MODBUS-Endpunktes (Revision API, Controller (ESP32, SBC), Server/Client)
0x0100	0x20FF			Eigenschaften und Funktion Produkte
		0x0100	0x01FF	Standalone: Produkt Gruppe: Produkt 1
		0x0200	0x02FF	Gruppe: Produkt 2
		0x0300	0x03FF	Gruppe: Produkt 3
		…	…	Gruppe: Produkt n
		0x2000	0x20FF	Gruppe: Produkt 32
0x2100	0x2FFF			(reserviert)
0x3000	0x4FFF			Eigenschaften und Funktion Outlet
		0x3000	0x30FF	Standalone: Outlet 1 = Outlet links Gruppe: Outlet 1
		0x3100	0x31FF	Standalone: Outlet 2 = Outlet rechts Gruppe: Outlet 2
		0x3200	0x32FF	Gruppe: Outlet 3
		...	...	Gruppe: Outlet n
		0x4F00	0x4FFF	Gruppe: Outlet 32

3.2. Allgemeine Eigenschaften des Modbus-Endpunktes

Beschreibt die Allgemeinen Eigenschaften des MODBUS-Endpunktes, mit dem die MODBUS-Kommunikation stattfindet

Adressraum (MSB) 0x00..

Startadresse (LSB)	**Anzahl Register**	**Inhalt**		
0x..01	1	Revision API	R	HighByte → Major (z.B. 0x01) LowByte → Minor (z.B. 0x05)
0x..02	1	Typ Controller	R	0x0000 → ESP32 0x0001 → SBC
0x..03	1	Node type	R	0x0000 → Server

R → Read only access W → Write only access RW → Read and write access

3.3. Produkte

Beschreibt die Eigenschaften von Produkten mit Outlet (Ebene Ladepunktsteuerung).

Adressraum (MSB) 0x01.. - 0x20..

Startadresse (LSB)	**Anzahl Register**	**Inhalt**		
0x..00	16	Typ	R	Materialnummer SAP der Wallbox 32 ASCII-Zeichen, zwei Zeichen je Register
0x..10	16	S/N	R	S/N der Wallbox 32 ASCII-Zeichen, zwei Zeichen je Register
0x..20	1	Ausführung	R	bit 15...12 → 0x0 (reserviert) bit 11... 8 → 0x0=ein outlet; 0x1=zwei outlets bit 7... 4 → 0x0=Kabel; 0x1=socket bit 3... 0 → 0x0=einphasig; 0x1=dreiphasig
0x..21	1	Outlet#	R	Nummer der zugeordneten outlets (1...32) bit 15 ... 8 → Outlet links (standalone 0x1) bit 7 ... 0 → Outlet rechts (standalone 0x2)
0x..22	1	FW-Revision	R	bit 15...12 → Major bit 11... 8 → Minor bit 7 ... 0 → Patch

Startadresse (LSB)	Anzahl Register	Inhalt		
0x..23	1	I_{rated}	R	Nennstrom des Produktes 6 ... 32A 0x003C ... 0x0140 → 10x Strom in (A)
0x..24	1	$I_{default}$ $I_{default} \leq I_{rated}$	R	Maximalstrom des Produktes 6 ...32A aufgrund der Installation 0x003C ... 0x0140 → 10x Strom in (A)
0x..26	1	Spannung AR4100	R	Spannung 0 ... 12V am Steuereingang AR4100 0x0000 ... 0x0078 → 10x Spannung in (V) In Gruppe: nur für Controller auszulesen

R → Read only access W → Write only access RW → Read and write access

Beispiel:

1. $I_{default}$ standalone → Register 0x0124
2. Ausführung outlets Produkt 6 einer Gruppe → Register 0x0620

3.4. Outlet

Beschreibt die Eigenschaften von Outlets.

Adressraum (MSB) 0x30.. - 0x4f..

Startadresse (LSB)	Anzahl Register	Inhalt		
0x..00	1	Produkt#	R	Nummer des zugeordneten Produktes (1...32) standalone → 0x0001
0x..01	6	Strom Zähler (3x uint32)	R	10x Strom Phase in A; Auflösung 0.1A bit 95 ... 64 → Phase 1 bit 63 ... 32 → Phase 2 bit 31 ... 0 → Phase 3
0x..07	6	Spannung Zähler (3x uint32)	R	10x Spannung Phase in V; Auflösung 0.1V bit 95 ... 64 → Phase 1 bit 63 ... 32 → Phase 2 bit 31 ... 0 → Phase 3
0x..0D	2	Wirkleistung Zähler (unit32)	R	Wirkleistung outlet in W 0x00000000 ... 0xFFFFFFFF → 0 ... 4294967.295 kW

Startadresse (LSB)	Anzahl Register	Inhalt		
0x..0F	2	Zählerstand Wirkenergie (unit32)	R	100x Zählerstand outlet in kWh; Auflösung 0.01kWh0 ... 999999,99kWh 0x00000000 ... 0xFFFFFFFF → 0 ... 42949672.95 kWh
0x..31	1	Status outlet (unit16)	R	0x00A0 → Outlet geblockt, EV wird erkannt 0x00A1 → Outlet wartet auf EV 0x00A2 → Outlet reserviert 0x00B0 → EV erkannt, Authentifizierung gescheitert 0x00B1 → EV erkannt, Authentifizierung 0x00B2 → Outlet kann Energie für Ladung bereitstellen 0x00B3 → EV hat Ladung beendet oder unterbrochen 0x00C2 → Outlet stellt Energie für Ladung bereit (Anforderung durch EV) 0x00E0 → Outlet geblockt, EV wird nicht erkannt 0x00E2 → Outlet im Bootvorgang 0x00Fx → Fehler
0x..32	1	Icmax (unit16)	RW	EMS-Vorgabe maximaler Strom für den outlet; Auflösung 0.1A Icmax ≤Idefault 0x0000, 0x003C ... 0x0140 → 0.0A, 6.0 ... 32.0A
0x..33	1	Ic (unit16)	R	Max Strom, den das EV aufnehmen darf; Auflösung 0.1A 0x0000, 0x003C ... 0x0140 → 0.0A, 6.0 ... 32.0A

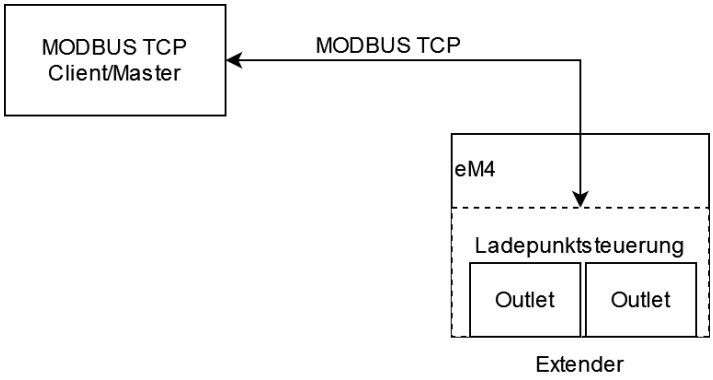
R → Read only access W → Write only access RW → Read and write access

Beispiel:

1. Abfrage Spannung Zähler rechts TWIN standalone (d.h. outlet 2) → Register 0x3107
2. Abfrage der Produktenummer, der dieser outlet (hier outlet 6) zugeordnete ist → Register 0x3500

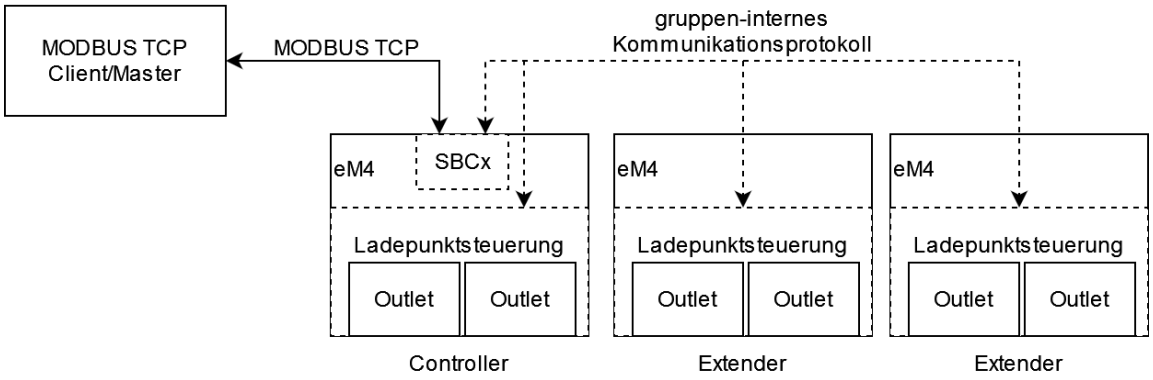
4. Anwendungsbeispiele

4.1. Extender als Stand-Alone



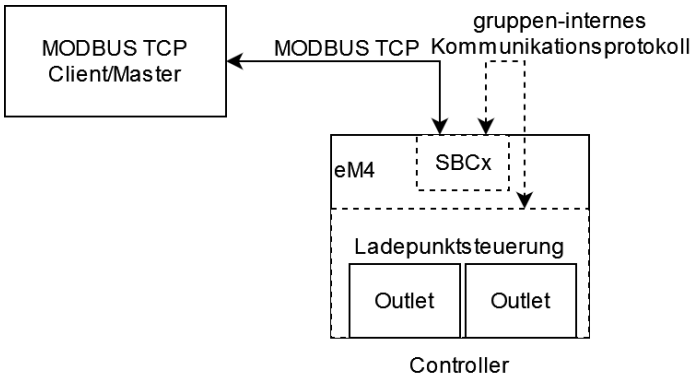
- Ladepunktsteuerung ist MODBUS-Endpunkt
- Endpunkt unterstützt 3.2 (Eigenschaften Endpunkt), 3.3 für das Produkt und 3.4 für jeden outlet

4.2. Gruppe mit Controller und Extender



- SBCx der Controller ist MODBUS-Endpunkt
- Endpunkt unterstützt 3.2 (Eigenschaften Endpunkt), 3.3 für jedes Produkt und 3.4 für jeden outlet

4.3. Controller als „Stand-Alone“



- SBCx der Controller ist MODBUS-Endpunkt
- Endpunkt unterstützt 3.2 (Eigenschaften Endpunkt), 3.3 für das Produkt und 3.4 für jeden outlet