

Unified-E Modbus Adapter Benutzerhandbuch

Modbus-Endpunkte und Datenpunkte konfigurieren

Software-Version 3.1.0.0, zuletzt aktualisiert: Juli 2025

Herausgeber: Unified-E AG, Winterthur, Schweiz



Inhalt

1	Allgemeines	3
1.1	Einführung.....	3
1.2	Speicherbereiche bei Modbus	3
2	Kommunikationsparameter des Adapters	3
2.1	Allgemeine Parameter.....	4
2.2	TCP/IP-spezifische Parameter	5
2.3	Seriell/USB-spezifische Parameter.....	5
3	Adressierung von Datenpunkten	6
3.1	Datenpunkt-Adresse im Dialog konfigurieren	6
3.2	Aufbau der Datenpunkt-Adresse	7

1 Allgemeines

1.1 Einführung

Der Modbus-Adapter ermöglicht den Zugriff auf externe Geräte über das Modbus-Protokoll und stellt die gelesenen oder geschriebenen Werte in der Visualisierung zur Verfügung. Unterstützt werden sowohl Modbus TCP (über Ethernet) als auch Modbus RTU (über serielle Schnittstellen).

1.2 Speicherbereiche bei Modbus

Modbus kennt vier Speicherbereiche, der Zugriff erfolgt über Registernummern:

- **Halte-Register:**
Es handelt sich um 16-bit Register, die sowohl gelesen als auch gesetzt werden können.
- **Digitaler Ausgang:**
Dort sind die Register der digitalen Ausgänge zu finden, die sowohl gelesen als auch gesetzt werden können. Der Register-Wert ist immer 0 oder 1 (BOOL).
- **Eingangs-Register:**
Es handelt sich um 16-bit Register, die jeweils einen analogen Eingang widerspiegeln. Ein Eingangs-Register kann nur gelesen werden.
- **Digitaler Eingang:**
Dort sind die Register der digitalen Eingänge zu finden, die folglich nur gelesen werden können. Der Register-Wert ist immer 0 oder 1 (BOOL).

Auch wenn das Modbus-Protokoll ausschliesslich 16-Bit-Register (WORD) adressiert, können über den Unified-E Modbus-Adapter auch einzelne Bits, 32-Bit-Werte, Zeichenketten oder Array-Strukturen verarbeitet werden (wie in Kapitel 3 beschrieben).

2 Kommunikationsparameter des Adapters

Für jeden Modbus-Endpunkt müssen grundlegende Kommunikationsparameter konfiguriert werden. Die Auswahl der Schnittstelle bestimmt dabei, ob die Kommunikation über TCP/IP oder Seriell/USB erfolgt. Je nach gewählter Schnittstelle werden unterschiedliche Parameter angezeigt.

Endpunkt-Adresse konfigurieren:

- **TCP/IP-Schnittstelle:** Die IP-Adresse, z. B. «192.168.1.40»
- **Serielle Schnittstelle:** Der COM-Port, z. B. «COM6»

Beispiel: Serielle Schnittstelle

Schritt 2: Parameter einstellen

Allgemein	
Schnittstelle	Seriell/USB
Geräte-ID	1
Startadresse	1-basiert
Register-Reihenfolge	Big-Endian (höchstes Register zuerst)
Seriell/USB	
Baudrate	38400
Parität	None
Stoppbits	1
Timeout [ms]	3000

Beispiel: Ethernet-Schnittstelle

Schritt 2: Parameter einstellen

Allgemein	
Schnittstelle	TCP/IP
Geräte-ID	1
Startadresse	1-basiert
Register-Reihenfolge	Big-Endian (höchstes Register zuerst)
TCP/IP	
TCP-Port	502
Timeout [ms]	2000

2.1 Allgemeine Parameter

Diese Parameter sind bei beiden Schnittstellenarten verfügbar:

- Schnittstelle: Auswahl der gewünschten Kommunikationsart:
 - TCP/IP für Modbus-TCP-Verbindungen
 - Seriell/USB für Modbus-RTU über serielle Schnittstellen
- Geräte-ID: Gibt die Slave-ID (auch Unit-ID) des angeschlossenen Modbus-Geräts an
 - Gültige Werte: 1 bis 247 (je nach Gerät)
- Startadresse: Gibt an, ob Registeradressen 0- oder 1-basiert angegeben werden.

- In der Regel wird 1-basiert verwendet, d. h. das erste Register hat die Adresse 1
- **Register-Reihenfolge:** Bestimmt die Reihenfolge mehrwortiger Werte (z. B. 32-Bit-Werte)
 - Big-Endian (höchstes Register zuerst): Das Register mit dem höheren Index enthält die höherwertigen Bits
 - Little-Endian (niederwertigstes Register zuerst): Das Register mit dem niedrigeren Index enthält die höherwertigen Bits
 - Diese Einstellung beeinflusst die Interpretation von mehrwortigen Datentypen wie Int32 oder Float

2.2 TCP/IP-spezifische Parameter

Diese Parameter gelten nur, wenn die Schnittstelle auf TCP/IP gesetzt ist:

- **TCP-Port:** Der zu verwendende Port des Modbus-TCP-Servers
 - Standardwert: 502
- **Timeout [ms]:** Zeit in Millisekunden, wie lange auf eine Antwort vom Modbus-TCP-Server gewartet wird
 - Dieser Timeout gilt für Verbindungsaufbau und Datenzugriffe.
 - Beispiel: 2000 = 2 Sekunden

2.3 Seriell/USB-spezifische Parameter

Diese Parameter gelten nur, wenn die Schnittstelle auf Seriell/USB gesetzt ist:

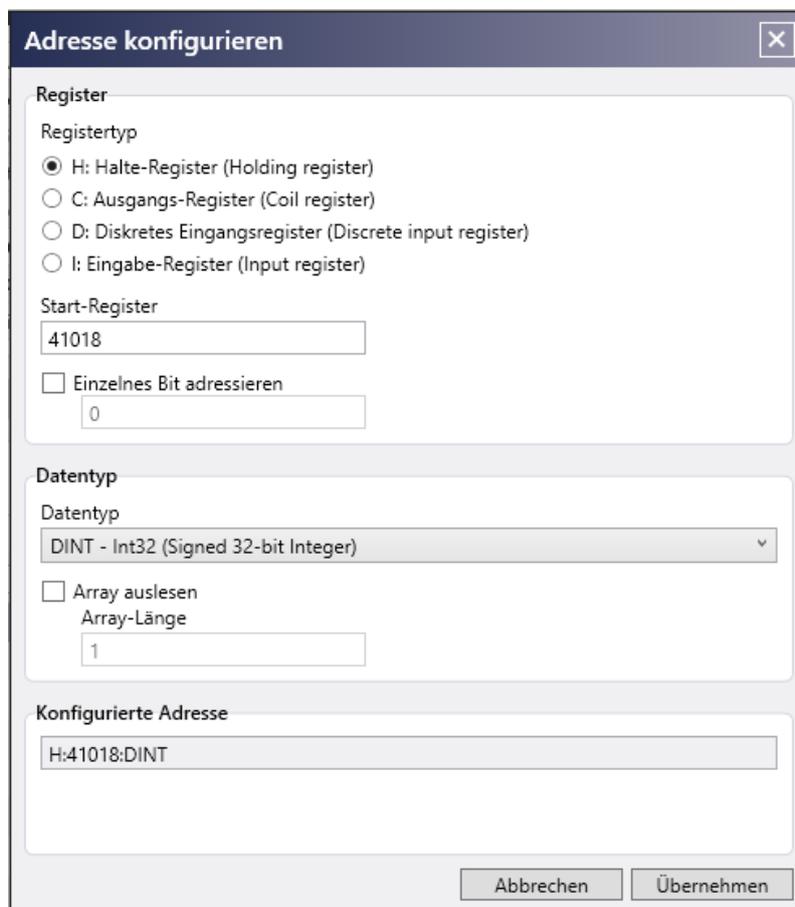
- **Baudrate:** Übertragungsgeschwindigkeit in Bit pro Sekunde.
 - Mögliche Werte: 2400 bis 230400
 - Beispiel: 38400
- **Parität:** Paritätsprüfung zur Erkennung von Übertragungsfehlern
- **Stoppbits:** Gibt an, wie viele Stoppbits bei der Übertragung verwendet werden
 - Gängige Werte: 1 oder 2
- **Timeout [ms]:** Zeit in Millisekunden, wie lange auf eine Antwort beim Lesen oder Schreiben über die serielle Schnittstelle gewartet wird
 - Beispiel: 3000 = 3 Sekunden

3 Adressierung von Datenpunkten

3.1 Datenpunkt-Adresse im Dialog konfigurieren

Im Unified-E App Designer können die Datenpunkt-Adressen von Modbus-Endpunkten direkt über einen Konfigurationsdialog erstellt werden. Die Adresse eines Modbus-Datenpunkts beschreibt dabei, auf welches Register im externen Modbus-Gerät zugegriffen werden soll.

Der Dialog «Adresse konfigurieren» lässt sich in der Datenpunkte-Tabelle in der «Adresse»-Spalte über die Schaltfläche «...» öffnen und bietet eine strukturierte Eingabemaske für die wichtigsten Adressierungsparameter:



- **Registertyp:** Legt fest, auf welchen Modbus-Registerbereich zugegriffen werden soll:
 - H: Halte-Register (Holding register): Lese- und schreibbare 16-Bit-Register (FC 03 / FC 06)
 - C: Ausgangs-Register (Coil register): 1-Bit-Register für digitale Ausgänge (FC 01 / FC 05)
 - D: Diskretes Eingangsregister (Discrete input register): 1-Bit-Register für digitale Eingänge, nur lesbar (FC 02)
 - I: Eingabe-Register (Input register): Nur lesbare 16-Bit-Register (FC 04)

- **Start-Register:** Die numerische Adresse des Registers, auf das zugegriffen werden soll
 - Beispiel: 41018 (entspricht Registeradresse 41018 innerhalb des gewählten Bereichs)
- **Einzelnes Bit adressieren:** Nur sichtbar bei booleschen Datentypen. Aktivieren Sie diese Option, wenn Sie ein einzelnes Bit innerhalb eines 16-Bit-Registers auslesen oder schreiben möchten
 - Bitnummer (0–15): Gibt an, welches Bit innerhalb des Registers verwendet werden soll
- **Datentyp:** Bestimmt, wie der Registerinhalt interpretiert wird. Zur Auswahl stehen verschiedene Ganzzahlen, Gleitkommazahlen und Boolesche Werte:
 - Beispiel: DINT – Int32 (Signed 32-bit Integer)

3.2 Aufbau der Datenpunkt-Adresse

Die Adresse eines Datenpunktes in der Datenpunkt-Tabelle hat folgenden Aufbau:

<Adressbereich>:<Startregister[.<Bitnummer>]>:<Datentyp>

Adressbereich-Kürzel:

- "H": Halte-Register (Holding register)
- "C": Digitaler Ausgang (Coil), Datentyp BOOL muss gesetzt werden
- "I": Eingangs-Register (Input register)
- "D": Digitaler Eingang (Digital input) , Datentyp BOOL muss gesetzt werden

Startregister:

Die Modbus-Startadresse für den Datenpunkt.

Bitnummer:

Bei numerischen Datentypen kann mit ".<Bitnummer>" das jeweilige Bit adressiert werden.

Beispiel: "H:100.3:WORD" adressiert das vierte Bit vom WORD-Wert.

Datentyp:

Bei Datentypen, die grösser sind als die Registergrösse werden beginnend vom Startregister die nachfolgenden Register zusätzlich verwendet.

Mögliche Datentypen:

- BOOL, nur verwenden bei digitalen Ein- und Ausgängen (Adressbereich "C" und "D")
- BYTE, USINT
- SINT
- WORD, UINT
- DWORD, UDINT
- INT
- DINT
- LINT
- ULINT
- REAL
- LREAL
- STRING[<Anzahl>]; die Anzahl muss im Bereich 1 bis 100 liegen, bei der Startadresse (erstes Byte) steht die aktuelle Länge, die einzelnen Zeichen werden als ANSI-Byte interpretiert.
- ARRAY mit numerischen Werten. Syntax: <Basistyp>[<Anzahl Elemente>]

Beispiele:

C:0:BOOL: Adressiert einen digitalen Ausgang 0

H:0:REAL: Adressiert einen REAL-Wert, beginnend bei Startadresse 0

Beispiel: Modbus-Datenpunkte in der Datenpunkte-Tabelle:

Unified-E App Designer 3.1.5.2 - Lounge Unified-E.uep (verändert)

Datei Bearbeiten Simulator Veröffentlichungen Einstellungen Hilfe

Erstellen Bearbeiten Simulieren Online testen Veröffentlichungen

Aktive App-Sprache: Deutsch Aktives Display: Desktop-PC

Projekt-Navigation: Smartfox, MySonne, Jetter, PVA

Ansichten: Endpunkt-Datenpunkte

Endpunkt-Datenpunkte: Smartfox, Jetter, MySonne

Lokale Datenpunkte: Statustexttypen, Bildtypen, Farbtypen, Meldungen, Rezepturen, Vorlagen

Ansichtenrahmen, App-Sprachen, Benutzer-Verwaltung, Skripte, Trigger, Kurven-Aufzeichnungen, Daten-Kommunikation, Einstellungen

Endpunkt-Adapter auswählen und parametrieren

Schritt 1: Adapter auswählen

Endpunkt Objektname: Smartfox

Endpunkt-Adapter für Kommunikation: Modbus Adapter

Endpunkt-Adresse: 192.168.11.11

Schritt 2: Parameter einstellen

Allgemein

Schnittstelle: TCP/IP

Geräte-ID: 1

Startadresse: 1-basiert

Register-Reihenfolge: Big-Endian (höchstes Register zuerst)

TCP/IP

TCP-Port: 502

Timeout [ms]: 2000

Schritt 3: Verbindung testen

Online-Verbindung zum Endpunkt testen

Mit der eingegebenen Adresse und den Parametern wird die Erreichbarkeit des Endpunktes getestet.

Verbindung testen...

Kommentar

Datenpunkte definieren

Datenpunkte

Endpunkt-Datenpunkte beobachten: Starten Beenden

Smartfox: Modbus Adapter - 192.168.11.11

Verbindungs-Status:

Bezeichnung	Adresse	Zugriff	Datentyp	Simulator Startwert	Gruppe
8 Energie lieferung Total	H:41004:ULINT	Lesen	Numerisch		
9 Energie bezug aktueller Tag	H:41012:UDINT	Lesen	Numerisch		
10 Energie lieferung aktueller Tag	H:41014:UDINT	Lesen	Numerisch		
11 Energy Smartfox aktueller Tag	H:41016:UDINT	Lesen	Numerisch		
12 Wechselrichter 1 power	H:41400:UDINT	Lesen	Numerisch		
13 Wechselrichter 1 energie	H:41402:ULINT	Lesen	Numerisch		
14 relay 4 mode	H:42340:UINT	Schreiben, Lesen	Numerisch		
15 SO input	H:41040:UDINT	Lesen	Numerisch		

Script-Datenpunkte definieren

Importieren... Exportieren...