

# **Unified-E Modbus Adapter User Manual**

## **Configure Modbus Endpoints and Datapoints**

Software version 3.1.0.0, last updated: July 2025

Publisher: Unified-E AG, Winterthur, Switzerland



## Content

<b>1</b>	<b>General.....</b>	<b>3</b>
1.1	Introduction .....	3
1.2	Storage Areas in Modbus .....	3
<b>2</b>	<b>Communication Parameters of the Adapter .....</b>	<b>3</b>
2.1	General Parameters .....	4
2.2	TCP/IP-specific Parameters .....	5
2.3	Serial/USB-specific Parameters .....	5
<b>3</b>	<b>Configure Datapoint Addresses .....</b>	<b>5</b>
3.1	Configure the Datapoint Address in the Dialog .....	5
3.2	Format of the Datapoint Address.....	7

# 1 General

## 1.1 Introduction

The Modbus adapter allows access to external devices via the Modbus protocol and makes the read or written values available in the visualization. Both Modbus TCP (via Ethernet) and Modbus RTU (via serial ports) are supported.

## 1.2 Storage Areas in Modbus

Modbus has four memory areas, which can be accessed via register numbers:

- **Hold registers:**  
These are 16-bit registers that can be read as well as set
- **Digital output:**  
This is where the registers of the digital outputs can be found, which can be read as well as set. The register value is always 0 or 1 (BOOL)
- **Input registers:**  
These are 16-bit registers, each reflecting an analog input. An input register can only be read
- **Digital Input:**  
This is where the registers of digital inputs can be found, which can therefore only be read. The register value is always 0 or 1 (BOOL)

Even though the Modbus protocol only addresses 16-bit registers (WORD), the Unified-E Modbus adapter can also be used to process individual bits, 32-bit values, strings or array structures (as described in Chapter 3).

# 2 Communication Parameters of the Adapter

Basic communication parameters must be configured for each Modbus endpoint. The selection of the interface determines whether communication takes place via TCP/IP or Serial/USB. Depending on the interface you choose, different parameters are displayed.

### Configure endpoint address:

- **TCP/IP interface:** The IP address, e.g. "192.168.1.40"
- **Serial port:** The COM port, e.g. "COM6"

### Example: Serial port:

Step 2: Set parameters

Allgemein	
Interface	Serial/USB port
Device ID	1
Start address	1-based
Register ordering	Big-endian (most significant register first)
Serial/USB port	
Baud rate	38400
Parity	None
Stop bits	1
Timeout [ms]	3000

Example: Ethernet interface:

Step 2: Set parameters

Allgemein	
Interface	TCP/IP
Device ID	1
Start address	1-based
Register ordering	Big-endian (most significant register first)
TCP/IP	
TCP port	502
Timeout [ms]	2000

## 2.1 General Parameters

These parameters are available for both types of interfaces:

- Interface: Selection of the desired type of communication:
  - TCP/IP for Modbus TCP Connections
  - Serial/USB for Modbus RTU via serial ports
- Device ID: Specifies the slave ID (also unit ID) of the connected Modbus device
  - Valid values: 1 to 247 (depending on the device)
- Start Address: Specifies whether register addresses are specified 0 or 1 based.
  - Typically, 1-based is used, meaning the first register has address 1
- Register Order: Determines the order of multi-word values (for example, 32-bit values)

- Big-endian (highest register first): The register with the higher index contains the higher-order bits
- Little-endian (lowest register first): The register with the lower index contains the higher-order bits
- This setting affects the interpretation of multi-word data types such as Int32 or Float

## 2.2 TCP/IP-specific Parameters

These parameters only apply if the interface is set to TCP/IP:

- TCP port: The port of the Modbus TCP server to use
  - Default value: 502
- Timeout [ms]: Time in milliseconds how long it waits for a response from the Modbus TCP server
  - This timeout applies to connection establishment and data access.
  - Example: 2000 = 2 seconds

## 2.3 Serial/USB-specific Parameters

These parameters only apply if the interface is set to Serial/USB:

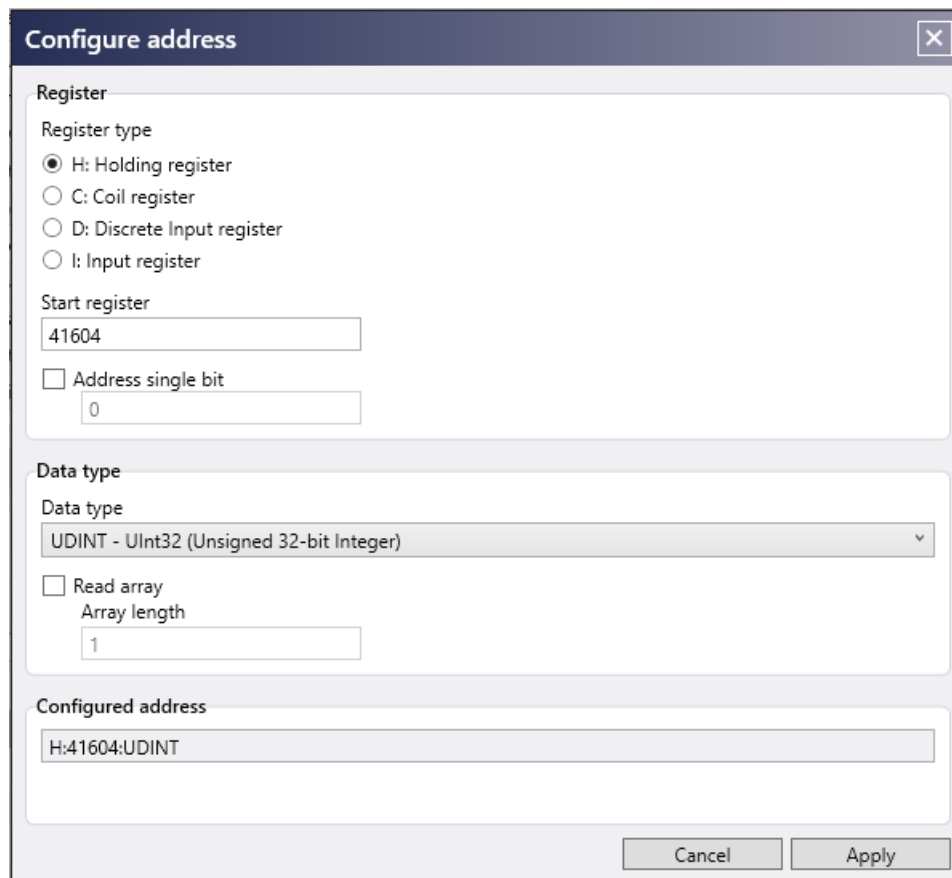
- Baud rate: Transmission speed in bits per second.
  - Possible values: 2400 to 230400
  - Example: 38400
- Parity: Parity check to detect transmission errors
- Stop Bits: Specifies how many stop bits are used in the transfer
  - Common values: 1 or 2
- Timeout [ms]: Time in milliseconds how long it waits for a response when reading or writing through the serial port
  - Example: 3000 = 3 seconds

# 3 Configure Datapoint Addresses

## 3.1 Configure the Datapoint Address in the Dialog

In the Unified-E App Designer, the Datapoint addresses of Modbus endpoints can be created directly via a configuration dialog. The address of a Modbus Datapoint describes which register in the external Modbus device is to be accessed.

The "Configure address" dialog can be accessed in the Datapoints table in the "Address" column via the "..." and offers a structured input mask for the most important addressing parameters:



**Configure address**

**Register**

Register type

- ☒ H: Holding register
- ☐ C: Coil register
- ☐ D: Discrete Input register
- ☐ I: Input register

Start register

41604

☐ Address single bit

0

**Data type**

Data type

UDINT - UInt32 (Unsigned 32-bit Integer)

☐ Read array

Array length

1

**Configured address**

H:41604:UDINT

Cancel Apply

- **Register type:** Specifies which Modbus tab area to access:
  - H: Holding register: Read and write 16-bit registers (FC 03 / FC 06)
  - C: Output register (coil register): 1-bit register for digital outputs (FC 01 / FC 05)
  - D: Discrete input register: 1-bit register for digital inputs, read-only (FC 02)
  - I: Input register: Readable 16-bit registers only (FC 04)
- **Start register:** The numeric address of the register to be accessed
  - Example: 41018 (corresponds to register address 41018 within the selected range)
- **Address single bit:** Only visible on Boolean data types. Select this option if you want to read or write a single bit within a 16-bit register
  - Bit number (0-15): Specifies which bit to use within the register
- **Data type:** Determines how the register contents are interpreted. There are various integers, floating point numbers and Boolean values to choose from:
  - Example: DINT – Int32 (Signed 32-bit Integer)

## 3.2 Format of the Datapoint Address

The address of a Datapoint in the Datapoint table has the following format:

**<Address range>:<Start register[.<Bit number>]>:<Data rype>**

### Address range abbreviations:

- "H": Holding register
- "C": Digital output (coil), data type BOOL must be set
- "I": Input register
- "D": Digital input, data type BOOL must be set

### Start register:

The Modbus start address for the Datapoint.

### Bit number:

For numeric data types, "<bit number>" can be used to address the respective bit.

For example, "H:100.3:WORD" addresses the fourth bit of the WORD value.

### Data type:

For data types that are larger than the register size, the following registers are also used, starting from the start register.

Possible data types:

- BOOL, use only for digital inputs and outputs (address range "C" and "D")
- BYTE, USINT
- SINT
- WORD, UINT
- DWORD, UDINT
- INT
- DINT
- LINT
- ULINT
- REAL
- LREAL

- STRING[<number>]; the number must be in the range of 1 to 100, the starting address (first byte) is the current length, the individual characters are interpreted as ANSI bytes.
- ARRAY with numeric values. Syntax: <Base Type>[<Number of Elements>]

### Examples:

C:0:BOOL: Addresses a digital output 0

H:0:REAL: Addresses a REAL value, starting at start address 0

### Example: Modbus Datapoints in the Datapoints table:

The screenshot shows the Unified-E App Designer 3.1.5.2 interface. The main window is titled "Unified-E App Designer 3.1.5.2 - Lounge Unified-E.uep (modified)". The menu bar includes File, Edit, Simulator, Publish, Settings, and Help. The toolbar contains icons for file operations and simulation. The "Active App Language" is set to German, and the "Active Display" is Desktop-PC.

The left sidebar shows the "Project Navigation" tree with categories like Views, Endpoints, Local Datapoints, Status Text Types, Image Types, Color Types, Alarm Messages, Recipes, Templates, View Frames, App Languages, User Management, Scripts, Triggers, Curve Recordings, Data Communication, and Settings.

The main workspace is divided into three steps for configuring the "Smartfox" endpoint:

- Step 1: Select adapter**: The "Endpoint object name" is "Smartfox". The "Endpoint adapter for communication" is "Modbus Adapter". The "Endpoint address" is "192.168.11.11".
- Step 2: Set parameters**: The "Allgemein" tab is selected. Parameters include:
  - Interface: TCP/IP
  - Device ID: 1
  - Start address: 1-based
  - Register ordering: Big-endian (most significant register first)
  - TCP/IP: TCP port 502, Timeout [ms] 2000
- Step 3: Test connection**: A "Test online connection to endpoint" button is available. A comment field is also present.

Below the configuration steps is the "Define Endpoint Datapoints" section. It shows a table of datapoints for the "Smartfox: Modbus Adapter - 192.168.11.11". The table has columns for Name, Address, Access, Data type, Simulator start value, and Group.

Name	Address	Access	Data type	Simulator start value	Group
3 car charge 1 energy total	H41600:ULINT	Read	Numeric		
4 car charge 1 energy pres	H41604:UDINT	Read	Numeric		
5 car charge 1 charge mode	H41608:UINT	Write, Read	Numeric		
6 car charge 1 man. charging value	H41609:UINT	Read	Numeric		
7 Energie bezug Total	H41000:ULINT	Read	Numeric		
8 Energie lieferung Total	H41004:ULINT	Read	Numeric		

At the bottom of the "Define Endpoint Datapoints" section, there are buttons for "Import..." and "Export..." and a "Define script datapoints" link.