ASC 150

4189341366A

Modbus server

User manual

Improve Tomorrow

1. General information

1.1 ASC Modbus server

The ASC Modbus server includes:

- A large proprietary protocol, including live data, and parameter reading and writing
- Inverter monitoring
- DEIF open protocol
- A SunSpec interface

More information
See the **ASC 150 Modbus server tables** (an Excel spreadsheet) for all the Modbus server addresses.

NOTE The ASC Modbus client is described in the **ASC 150 Modbus client User manual**.

1.2 Software version

This document is based on ASC 150 software version 1.17.

1.3 Disclaimer

DEIF A/S reserves the right to change any of the contents of this document without prior notice.

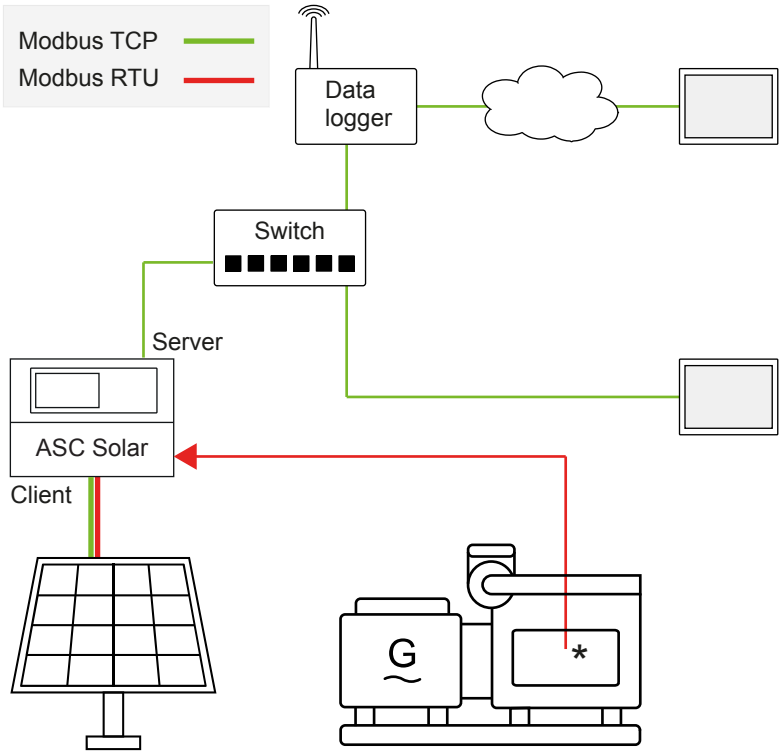
The English version of this document always contains the most recent and up-to-date information about the product. DEIF does not take responsibility for the accuracy of translations, and translations might not be updated at the same time as the English document. If there is a discrepancy, the English version prevails.

2. Hardware information

2.1 Communication

2.1.1 ASC Solar communication

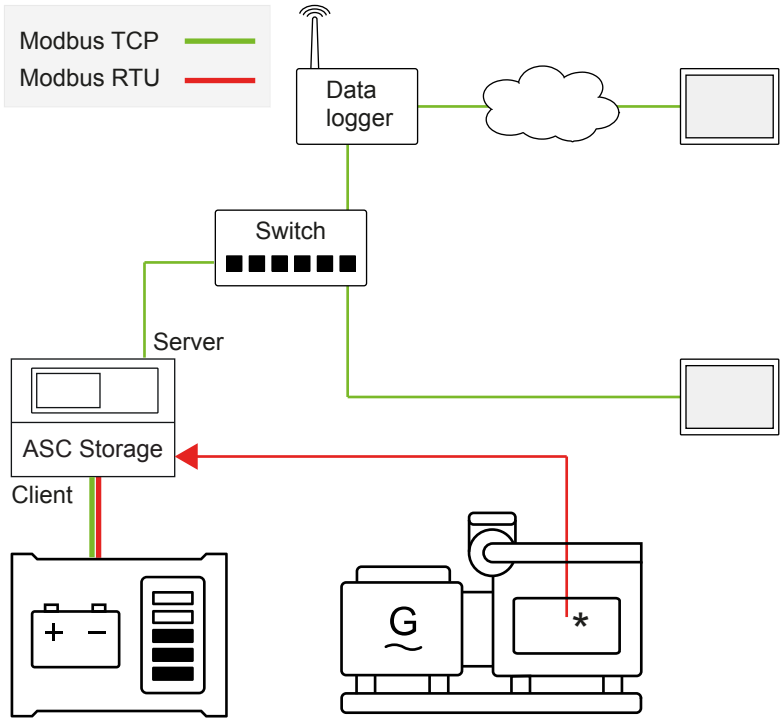
Example of ASC Solar communication for a single controller application



The ASC can communicate over Modbus as the client and/or server device. The ASC can communicate with the PV system using Modbus TCP or Modbus RTU. The ASC reads from power meters or genset controllers using Modbus RTU.

2.1.2 ASC Storage communication

Example of ASC Storage communication for a single controller application



The ASC can communicate over Modbus as the client and/or server device. The ASC can communicate with the ESS using Modbus TCP and/or Modbus RTU. * The ASC reads from power meters or genset controllers using Modbus RTU.

2.2 Connections for Modbus server

The external controllers and/or SCADA system are the Modbus clients, and the ASC is the Modbus server.

2.3 Wiring

More information
See the **Installation instructions** for wiring information.

2.4 Configuring Modbus connections

2.4.1 RS-485

`Communication > RS485 > RS485 [1 or 2] > Settings`

Parameter	Name	Range	Default	Details
7511 or 7521	Ext. comm. ID [1 or 2]	1 to 247	3	Modbus communication ID for the ASC Modbus server. The external Modbus client uses this ID to communicate with the ASC.
7512 or 7522	Ext. comm.speed[1 or 2]	9600 Baud 19200 Baud 38400 Baud 115200 Baud	9600 Baud	Communication speed for RS-485 [1 or 2]

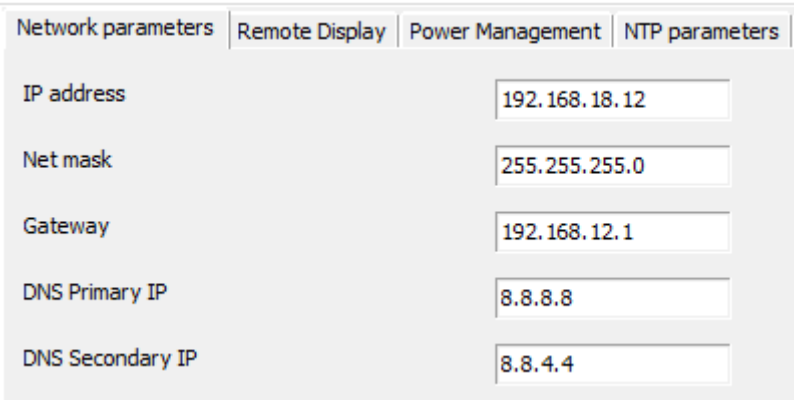
Communication > RS485 > RS485 [1 or 2] > Communication error

Parameter	Name	Range	Default	Details
7513 or 7523	Timer	1 to 100 s	10 s	Communication error for RS-485 [1 or 2]
7516 or 7526	Enable	OFF ON	OFF	
7517 or 7527	Fail class	Fail classes	Warning	

2.4.2 Modbus server TCP connection

The controller's Ethernet connection is used for remote or local monitoring.

You can use the utility software to see (or set up) the controller. On *the Ethernet setting (TCP/IP)* page, open *Network parameters*.



Alternatively, use the display: `Settings > Communication > Ethernet setup`

3. Data tables

3.1 Configurable area (read only) (function code 04h)

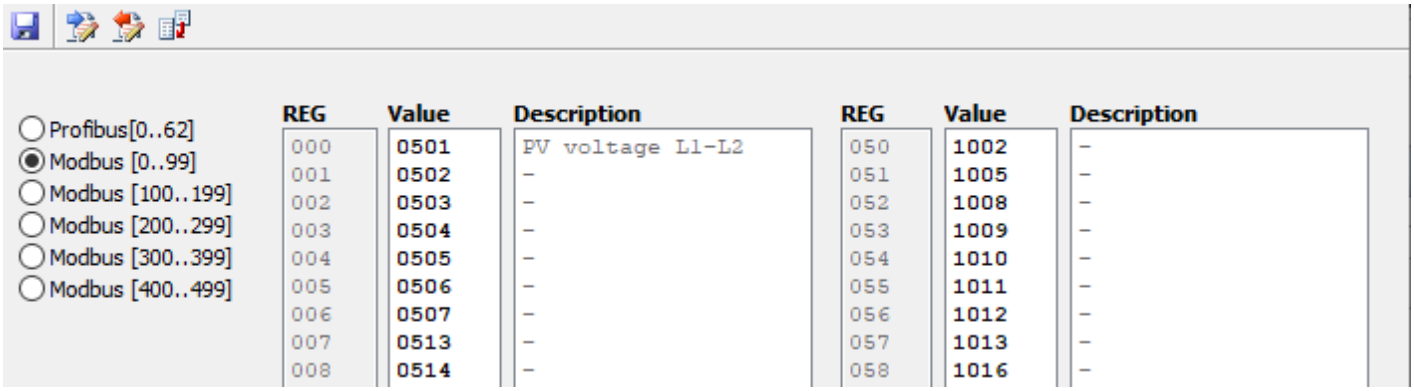
3.1.1 Modbus configurator

The Modbus configurator lets the user select which data should be available in the first 500 Modbus addresses for Modbus function 04.

Profibus users often use this function to select the data they can use. That is why the first address range is called Profibus. This first address range is shared by Profibus and Modbus users, as they share the same data.

To open the Modbus configurator, select *Modbus Configurator* from the left toolbar in the utility software.

Using the Modbus configuration, you can configure Modbus addresses 0 to 499 to any of the Modbus addresses from 500 to 1999, as shown below.



The window has four different columns that are described below:

Range: Each Modbus range contains 100 Modbus addresses (63 for the Profibus range). In the screenshot above, the Modbus address range 0 to 99 is selected.

REG: The information in the specific address REG is duplicated from the Modbus address configured in Value. The number changes when another range is selected (0 to 499).

Value: The Modbus address (500 to 1999) of the information that will be copied to the corresponding REG Modbus address.

Description: Free text for the user to fill in. The text is only saved in the parameter file. In the screenshot, Modbus address 000 duplicates the information of Modbus address 501, which displays the PV voltage between L1 and L2, as the description shows.

NOTE The popup window has its own dedicated *Read/write* and *Copy description* buttons, which must be used for manual configuration.

Configurable Modbus example

For the screenshot, to check what is assigned to Modbus address 001, look up Modbus address 502 in the **Modbus server tables**, under *Input register (04)*.

502 is an *AC measurement*, with function name *PV/ESS voltage L2-L3*. Therefore, reading Modbus address 001 from the input register (04) returns the *PV/ESS voltage L2-L3*.

	To assign a different function to Modbus address 001, change the number under *Value*. For example, change the number 507. Reading Modbus address 001 from the input register (04) then returns the *PV/ESS Frequency L1*.

3.2 Reference tables

The Modbus server tables can be downloaded from the documentation page under **Communication, Modbus server tables**. The Modbus server tables are stored in an .xlsx file that contains:

- Command flags (01; 05; 15)
- Digital inputs and outputs (02)
- Control registers (03; 06; 16)
- Read areas (ASC to PV) and write area (PV to ASC) (03; 06; 16)
- Read/write (Battery) (03; 06; 16)
- SunSpec Modbus server support (03; 06; 16)
- Weather area (03; 06; 16)
- A wide range of readable functions (04), including:
  - Profibus configurable area
  - Modbus configurable area
  - Alarms, measurements, statuses, states, and so on

The number in brackets refers to the Modbus function code (as a decimal value) for the information, and corresponds to the sheet names in the spreadsheet.

3.3 Data type

The **Modbus server tables** include the following data types:

- acc32: SunSpec
- acc64: SunSpec
- bitfield16: SunSpec
- bitfield32: SunSpec
- BOOL: Boolean data
- enum16: SunSpec
- INT16s: Signed 16-bit integers
- INT16u: Unsigned 16-bit integers (SunSpec)
- INT32s: Signed 32-bit integers
- INT32u: Unsigned 32-bit integers (SunSpec)
- string: Text data
- sunssf: SunSpec

3.4 Data format

The ASC uses these data formats:

- **AB**: 16-bit values.
- **ABCD**: 32-bit values. These are represented as with HI16 on lowest address and LO16 on highest address.

32-bit value format example

The *PV active power reference* is **INT32s** data, with data format **ABCD**, in Modbus address **46000** under *Holding register* (03;06;16).

Modbus address 46000 contains HI16 for *PV active power reference*.
Modbus address 46001 contains LO16 for *PV active power reference*.

3.5 Data scaling

Modbus data is processed as data bytes. This data cannot directly process decimal values. Therefore scaling is defined to convert decimal values to a form that can be sent using Modbus, or to correctly interpret values received from Modbus. Data in the *Holding register* and *Input register* is scaled according to the formula:

$$\text{Actual value} = \text{Value in register} * 10^{-\text{Scaling}}$$

For each Modbus address, the scaling is given in the **Modbus server tables**.

Scaling example

The Modbus address for battery frequency (holding register 0×03, 46342) has a scaling of 2. When the frequency is read from the controller using Modbus, the Modbus register returns 5000. The actual frequency is:

$$\begin{aligned} \text{Actual value} &= \text{Value in register} * 10^{-\text{Scaling}} \\ &= 5000 * 10^{-2} \\ &= 50.00 \end{aligned}$$

To write a new frequency of 60.00 Hz using Modbus, the value to write to the register is:

$$\begin{aligned} \text{Value in register} &= \text{Actual value} / 10^{-\text{Scaling}} \\ &= 60.00 / 10^{-2} \\ &= 6000 \end{aligned}$$

4. Parameter setting

4.1 Introduction

Modbus communication can read parameter data from the controller and write parameter data to the controller. The parameter Modbus addresses are not listed in the Modbus tables. The Modbus address for a parameter is calculated by adding an offset to the first number in the address area. The offset is the parameter *Address* in the Utility Software (USW).

Modbus communication can also read and write alarm data for IOs and CIOs. The IO Modbus addresses are not listed in the Modbus tables. The IO Modbus addresses are also calculated by adding an offset to the first number in the address area.

This chapter lists the address areas, and gives examples of Modbus address calculations.

NOTE The DEIF controller is the Modbus server.

4.2 Address areas

4.2.1 Read coil (Function code 01)

Reads the ON/OFF status of discrete output coils. The controller returns **0** (FALSE) when the coil is not activated, and **1** (TRUE) when the coil is activated.

Address area for reading status flags

Data to request	Address area
Enable	2000-3999

4.2.2 Read discrete inputs (Function code 02)

Reads the ON/OFF status of discrete input contacts. The controller returns **0** (FALSE) when the discrete input is not activated, and **1** (TRUE) when the discrete input is activated.

Address areas for reading status flags

Data to request	Address area
Alarm active	4000-5999
Alarm acknowledge	6000-7999
Timer output	8000-9999
Timer running	10000-11999

4.2.3 Read holding registers (Function code 03)

Reads the data value contained in the holding registers. The data can be signed integers (16 or 32 bit) or boolean values. The controller returns the value stored in the holding register. Note that you need the scaling to interpret the value correctly.

Address areas for reading holding registers

Data to request	Address area
Timers used	2000-3999
Values used	4000-5999
Values minimum	6000-7999

Data to request	Address area
Values maximum	8000-9999
Output A	10000-11999
Output B	12000-13999
Fail class used	14000-15999
Enable	16000-17999
Inhibit	18000-19999

4.2.4 Read input registers (Function code 04)

Reads the data value contained in the input registers. The data can be signed integers (16- or 32-bit) or boolean values. The controller returns the value stored in the input register. Note that you need the scaling to interpret the value correctly.

Address areas for reading input registers

Data to request	Address area
Timers minimum	2000-3999
Timers maximum	4000-5999
Output A minimum	6000-7999
Output A maximum	8000-9999
Output B minimum	10000-11999
Output B maximum	12000-13999
Fail class minimum	14000-15999
Fail class maximum	16000-17999
Timers elapsed time	20000-21999
Actual values	22000-23999

4.2.5 Write single/multiple coils (Function code 05/15)

Change the ON/OFF status of a single or multiple discrete output coils. Write **0** (FALSE) to deactivate the coil, or **1** (TRUE) to activate the coil.

Address areas for writing status flags

Data to request	Address area
Enable	2000-3999
Acknowledge alarm	6000-7999

4.2.6 Write single/multiple holding registers (Function code 06/16)

Change the value of a single or multiple holding registers. The data can be signed integers (16 or 32 bit) or boolean values. When writing values to holding registers, you need to use the correct scaling and data type.

Address area for writing holding registers

Data to request	Address area
Timers used	2000-3999
Values used	4000-4999
Output A	10000-11999

Data to request	Address area
Output B	12000-13999
Fail class used	14000-15999
Enable	16000-17999
Inhibit	18000-19999

4.3 Modbus addresses and examples for parameters

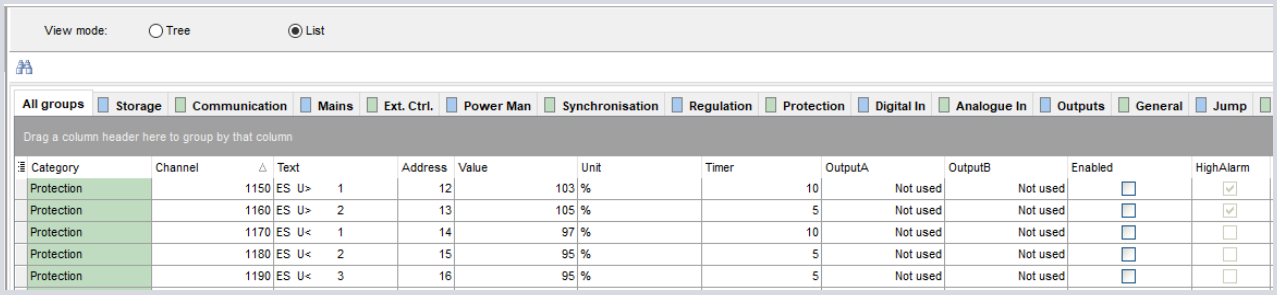
The Modbus address for a parameter is the sum of the **Address** in the Utility Software (USW) and the first value of the address area. To find the address of a specific parameter, go to the **Parameters** tab in the USW, then find the parameter using the parameter's name or parameter number (**Channel** column). The parameter address is located in the **Address** column.

Alarm example

Acknowledge alarm

In this example an over-voltage 1 alarm is active and unacknowledged. First we will check if the alarm is already acknowledged, then acknowledge the alarm using Modbus.

1. The parameter number for the over-voltage 1 alarm is 1150. Find the parameter in the USW in the **Parameters** tab and note the **Address** value for the parameter (12 for over-voltage 1).
   - The parameter numbers are listed in the **Channel** column.



Category	Channel	Text	Address	Value	Unit	Timer	OutputA	OutputB	Enabled	HighAlarm
Protection	1150	ES U> 1	12	103	%	10	Not used	Not used	☐	☑
Protection	1160	ES U> 2	13	105	%	5	Not used	Not used	☐	☑
Protection	1170	ES U< 1	14	97	%	10	Not used	Not used	☐	☐
Protection	1180	ES U< 2	15	95	%	5	Not used	Not used	☐	☐
Protection	1190	ES U< 3	16	95	%	5	Not used	Not used	☐	☐

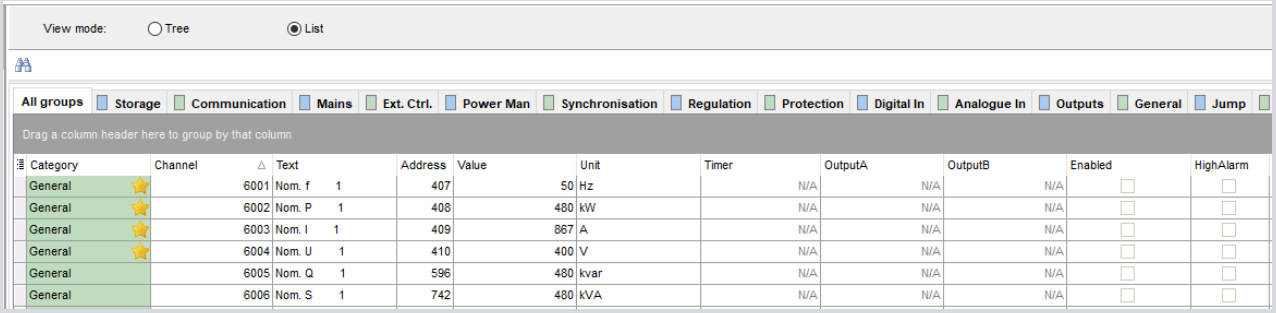
2. To read if the alarm is acknowledged, go to the table in **Address areas > Read discrete input (Function code 02)**. The address area for *Alarm acknowledge* starts at 6000.
3. The Modbus address to read the parameter is: Parameter **Address** + Address area start = 12 + 6000 = 6012.
4. Use function code 02 to read address 6012.
   - For this example when the address is read, the controller returns **0** (FALSE). This means that the alarm is not acknowledged.
5. To acknowledge the alarm, go to the table in **Address areas > Write single/multiple coils (Function code 05/15)**. The address area for *Acknowledge alarm* starts at 6000.
6. The Modbus address to read the parameter is: Parameter **Address** + Address area start = 12 + 6000 = 6012.
7. Use function code 05 to write **1** (TRUE) to address 6012.
   - The alarm is now acknowledged and reading address 6012 using function code 02 returns **1** (TRUE). This means the alarm is acknowledged.

Nominal setting example

Change nominal frequency

In this example nominal frequency 1 is changed from 50 Hz to 60 Hz.

1. The parameter number for the nominal frequency 1 is 6001. Find the parameter in the USW in the **Parameters** tab and note the **Address** value for the parameter (407 for nominal frequency 1).

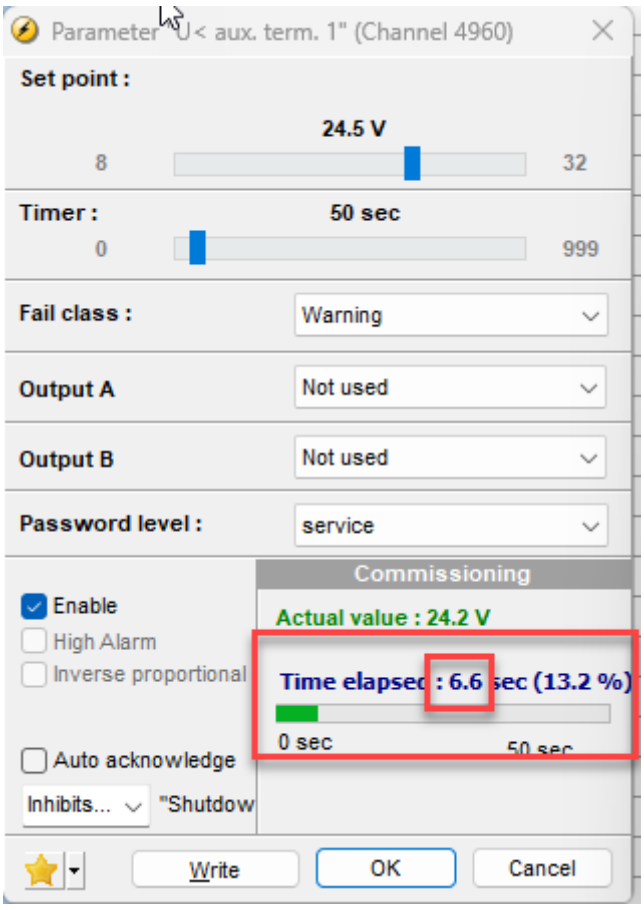


Category	Channel	Text	Address	Value	Unit	Timer	OutputA	OutputB	Enabled	HighAlarm
General	6001	Nom. f 1	407	50	Hz	N/A	N/A	N/A		
General	6002	Nom. P 1	408	480	kW	N/A	N/A	N/A		
General	6003	Nom. I 1	409	867	A	N/A	N/A	N/A		
General	6004	Nom. U 1	410	400	V	N/A	N/A	N/A		
General	6005	Nom. Q 1	596	480	kvar	N/A	N/A	N/A		
General	6006	Nom. S 1	742	480	kVA	N/A	N/A	N/A		

2. To change the nominal frequency, go to the table in **Address areas > Write single/multiple holding registers (Function code 06/16)**. The address area for *Values used* starts at 4000.
3. The Modbus address to write the new value to is: Parameter **Address** + Address area start = 407 + 4000 = 4407.
4. The nominal frequency has a scaling of 1. See **Data scaling*** for the exponential scaling formula and examples. To write 60 Hz to the address, a value of 600 must be written to the address. Use function code 06 to write 600 to address 4407.
   - Nominal frequency 1 is now 60.0 Hz. To confirm the change use function code 03 to read address 4407. The address returns 600. (The scaling is also 1.)

NOTE * A measurement's scaling in the Modbus tables is not necessarily the same as the scaling for the corresponding parameter. For example, the scaling for the frequency measurement is 2, while the scaling for the nominal frequency parameter is 1.

4.3.1 Examples for commissioning



Alarm timer elapsed time

The low supply voltage alarm parameter number is **4960** (*U< aux. term. 1*), and the address is **315**.

1. To see the alarm timer elapsed time, go to the table in **Address areas > Read input registers (Function code 04)**. The address area for *Timers elapsed time* starts at 20000.
2. The Modbus address to read the alarm timer elapsed time is: **Address** + Address area start = 315 + 20000 = 20315.
3. Use function code 04 to read address 20315.
4. Example: When the alarm timer elapsed time shown in the utility software is 6.6 seconds, the Modbus value is 66.
   - The alarm timer elapsed time thus has a scaling of 1.

Actual battery voltage

The operator wants to monitor the actual value of the supply voltage. The parameter number is **4960** (*U< aux. term. 1*), and the address is **315**.

1. To see the actual value, go to the table in **Address areas > Read input registers (Function code 04)**. The address area for *Actual values* starts at 22000.
2. The Modbus address to read the elapsed time is: **Address** + Address area start = 315 + 22000 = 22315.
3. Use function code 04 to read address 22315.
4. Example: When the actual value shown in the utility software is 24.2 V, the Modbus value is 242.
   - The actual value thus has a scaling of 1.

5. DEIF Open protocol

5.1 Using the DEIF Open protocol

The ASC can use its Modbus server interface to control the PV/ESS. The ASC calculates references for the PV/ESS plant and makes them available for the PV/ESS controller. The PV controller measures or calculates PV/ESS plant statuses and these are read by the ASC. Modbus addresses 46000 to 46999 (function code 0×03) are used for this.

The DEIF open protocol includes the following function groups:

- Read area 1 (ASC to PLC/external system) (from 46000): The ASC puts data here for the PV/ESS controller to read.
- Write area 1 (PLC/external system to ASC) (from 46100): The PV/ESS controller writes data here for the ASC to read.
- Read area 2 (ASC to PLC/external system) (from 46200): The ASC puts data here for the PV/ESS controller to read.
- Read/Write (Battery) (from 46300)

NOTE The ASC Solar controller can still use the DEIF Open protocol even if the *PV protocol* selection in parameter 7561 is not **DEIF Open**.

NOTE The ASC Storage controller can still use the DEIF Open protocol even if the *ESS prot.* selection in parameter 7561 is not **DEIF Open**.