



USER MANUAL



ASC-4 Modbus server



1. General information

1.1 ASC Modbus server.....	3
1.2 Software version.....	3
1.3 Disclaimer.....	3

2. Hardware information

2.1 Communication.....	4
2.1.1 ASC Solar communication.....	4
2.1.2 ASC Battery communication.....	5
2.2 Connections for Modbus server.....	5
2.3 Wiring.....	5

3. Data tables

3.1 Configurable area (read only) (function code 04h).....	6
3.1.1 Modbus configurator.....	6
3.2 Reference tables.....	7
3.3 Data type.....	7
3.4 Data format.....	7
3.5 Data scaling.....	8

4. Parameter setting

4.1 Introduction.....	9
4.2 Address areas.....	9
4.2.1 Read coil (Function code 01).....	9
4.2.2 Read discrete inputs (Function code 02).....	9
4.2.3 Read holding registers (Function code 03).....	9
4.2.4 Read input registers (Function code 04).....	10
4.2.5 Write single/multiple coils (Function code 05/15).....	10
4.2.6 Write single/multiple holding registers (Function code 06/16).....	10
4.3 Modbus addresses and examples for parameters.....	11

5. DEIF open protocol

5.1 Using the DEIF open protocol for control.....	13
---	----

6. Other Modbus data

6.1 Engine communication data.....	14
------------------------------------	----

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-4 Modbus server tables** (an Excel spreadsheet) for all the Modbus server addresses.

Use the ASC parameters 7500 to 7520 to configure the Modbus server communication.

NOTE The ASC Modbus client is described in the **ASC-4 Modbus client User manual**.

1.2 Software version

This document is based on ASC-4 software version 4.16 and 4.17.

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.

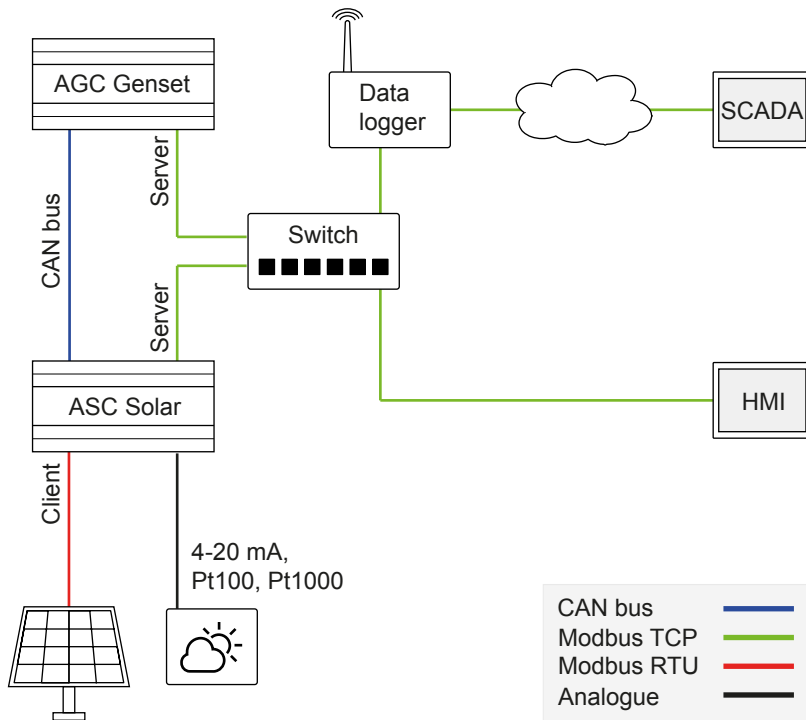
2. Hardware information

2.1 Communication

2.1.1 ASC Solar communication

The ASC can communicate as a Modbus client and/or Modbus server.

Example of ASC Solar communication



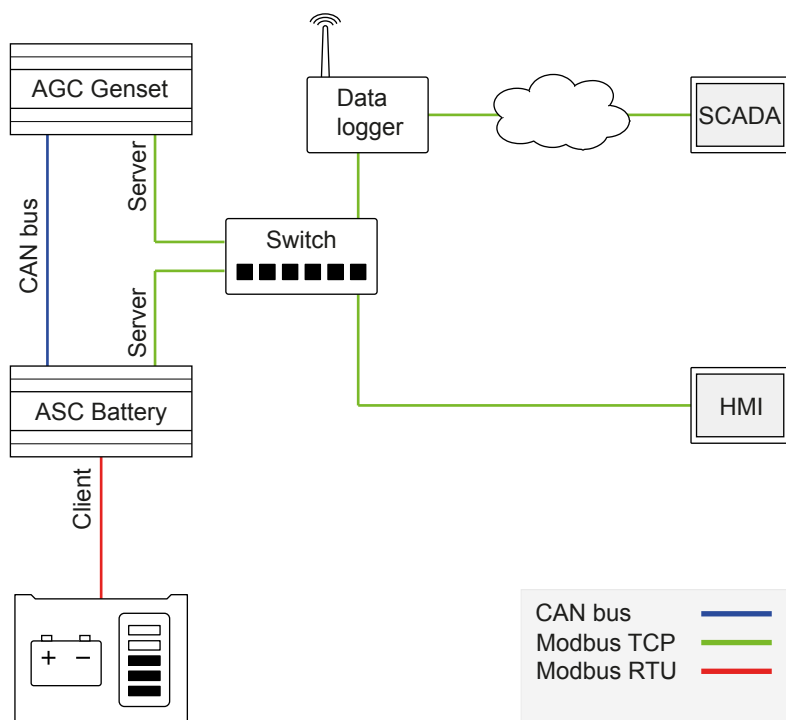
NOTE The ASC can get analogue weather data and/or use a Modbus RTU interface.

NOTE If required, you can use an RTU to TCP/IP converter for the communication between the ASC and the inverters. See the **ASC-4 Commissioning guidelines** for more information.

2.1.2 ASC Battery communication

The ASC can communicate as a Modbus client and/or Modbus server.

Example of ASC Battery communication



NOTE You can use an RTU to TCP/IP converter for the communication between the ASC and the battery. See the **ASC-4 Commissioning guidelines** for more information.

2.2 Connections for Modbus server

Option N (Modbus TCP/IP) is required for the ASC Modbus server. That is, 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.

3. Data tables

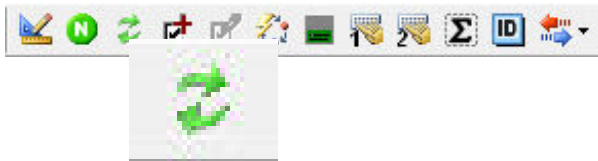
3.1 Configurable area (read only) (function code 04h)

3.1.1 Modbus configurator

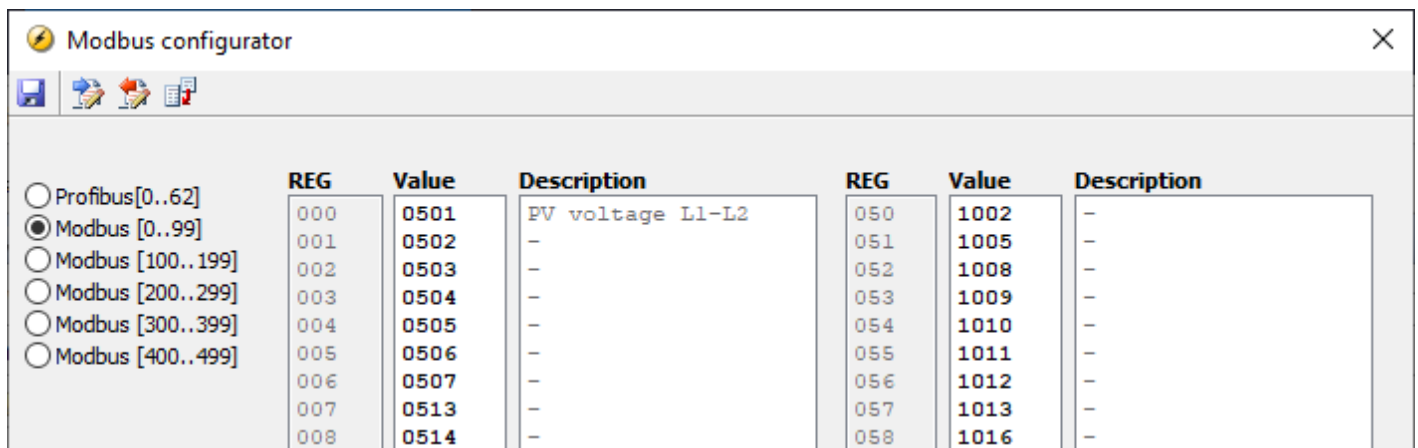
The Modbus configurator lets the user select which data should be available in the first 500 Modbus addresses for Modbus function 04.

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, click the icon shown below in the 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.



	REG	Value	Description	REG	Value	Description
☐ Profibus[0..62]	000	0501	PV voltage L1-L2	050	1002	-
☒ Modbus [0..99]	001	0502	-	051	1005	-
☐ Modbus [100..199]	002	0503	-	052	1008	-
☐ Modbus [200..299]	003	0504	-	053	1009	-
☐ Modbus [300..399]	004	0505	-	054	1010	-
☐ Modbus [400..499]	005	0506	-	055	1011	-
	006	0507	-	056	1012	-
	007	0513	-	057	1013	-
	008	0514	-	058	1016	-

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 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 www.deif.com/documentation/asc-4/ under **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 **40000**.

Modbus address 40000 contains HI16 for *PV active power reference*.

Modbus address 40001 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 0x03, 40342) 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
Values maximum	8000-9999

Data to request	Address area
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

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
Output B	12000-13999
Fail class used	14000-15999

Data to request	Address area
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.

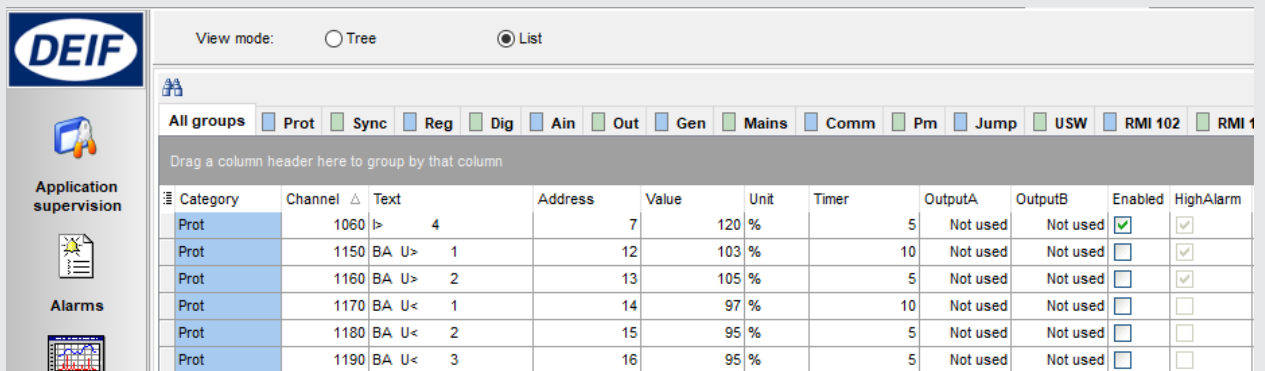
Alarm example



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.

- 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
Prot	1060	I> 4	7	120	%	5	Not used	Not used	☒	☒
Prot	1150	BA U> 1	12	103	%	10	Not used	Not used	☐	☒
Prot	1160	BA U> 2	13	105	%	5	Not used	Not used	☐	☒
Prot	1170	BA U< 1	14	97	%	10	Not used	Not used	☐	☐
Prot	1180	BA U< 2	15	95	%	5	Not used	Not used	☐	☐
Prot	1190	BA U< 3	16	95	%	5	Not used	Not used	☐	☐

- 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.
- The Modbus address to read the parameter is: Parameter **Address** + Address area start = 12 + 6000 = 6012.
- 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.
- 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.
- The Modbus address to read the parameter is: Parameter **Address** + Address area start = 12 + 6000 = 6012.
- 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.

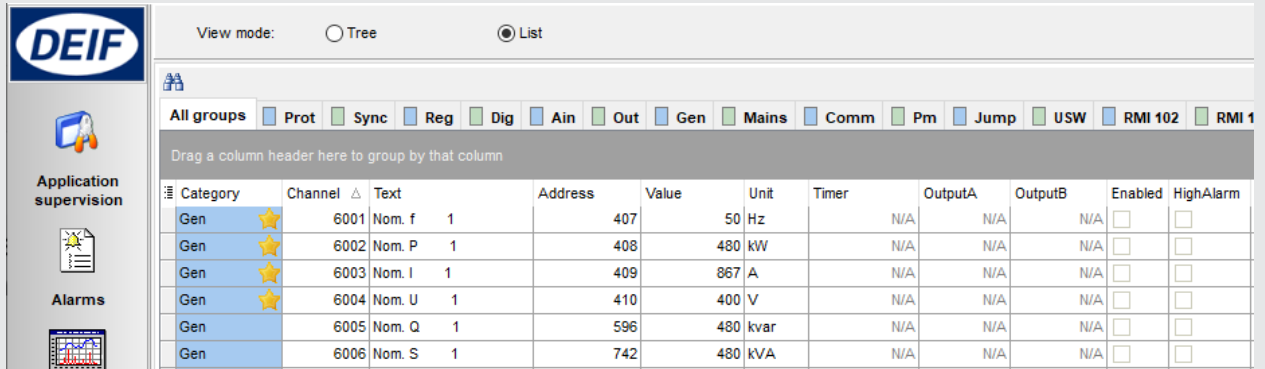
Nominal setting example



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
Gen	6001	Nom. f 1	407	50	Hz	N/A	N/A	N/A	☐	☐
Gen	6002	Nom. P 1	408	480	kW	N/A	N/A	N/A	☐	☐
Gen	6003	Nom. I 1	409	867	A	N/A	N/A	N/A	☐	☐
Gen	6004	Nom. U 1	410	400	V	N/A	N/A	N/A	☐	☐
Gen	6005	Nom. Q 1	596	480	kvar	N/A	N/A	N/A	☐	☐
Gen	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.

5. DEIF open protocol

5.1 Using the DEIF open protocol for control

The ASC can control the PV using its Modbus server interface. References for the PV plant are calculated by the ASC and made available for the PV controller. PV plant statuses are measured or calculated by the PV controller and read by the ASC. Modbus addresses 40000 to 40999 (function code 0x03) are used for this.

The DEIF open protocol includes the following function groups:

- Read area 1 (ASC to PV): The ASC puts data here for the PV controller to read.
- Read area 2 (ASC to PV): The ASC puts data here for the PV controller to read.
- Write area 1 (PV to ASC): The PV controller writes data here for the ASC to read.

NOTE The DEIF open protocol can still be used if the *PV protocol* selection in parameter 7561 is not **DEIF Open**.

6. Other Modbus data

6.1 Engine communication data

The ASC Modbus server includes 25 sets of engine communication data, from up to 16 gensets controlled by AGC controllers. This data uses function code 0x04, and is in addresses 31240 to 31751. If the data is not supported, it has the value 0x8000.

Contact DEIF if you need more information.