

# iE 150

Battery and Solar Modbus server

**User manual**



<b>1. General information</b>	
1.1 Modbus server.....	3
1.2 Software version.....	3
1.3 Disclaimer.....	3
<b>2. Hardware information</b>	
2.1 Battery/Storage controller communication.....	5
2.2 Solar controller communication.....	5
2.3 Connections for Modbus server.....	6
2.4 Wiring.....	6
2.5 Configuring Modbus connections.....	6
2.5.1 RS-485.....	6
2.5.2 Modbus server TCP connection.....	6
<b>3. Data tables</b>	
3.1 Configurable area (read only) (function code 04h).....	8
3.1.1 Modbus configurator.....	8
3.2 Reference tables.....	9
3.2.1 Alarm list monitoring.....	9
3.3 Data type.....	9
3.4 Data format.....	9
3.5 Data scaling.....	10
<b>4. Parameter setting</b>	
4.1 Introduction.....	11
4.2 Address areas.....	11
4.2.1 Read coil (Function code 01).....	11
4.2.2 Read discrete inputs (Function code 02).....	11
4.2.3 Read holding registers (Function code 03).....	11
4.2.4 Read input registers (Function code 04).....	12
4.2.5 Write single/multiple coils (Function code 05/15).....	12
4.2.6 Write single/multiple holding registers (Function code 06/16).....	12
4.3 Modbus addresses and examples for parameters.....	13
4.3.1 Examples for commissioning.....	15
4.4 Modbus addresses and examples for IOs.....	16
4.5 Modbus addresses and example for CIO modules.....	16
4.5.1 Modbus alarm offset addresses for CIO 116.....	17
4.5.2 Modbus alarm offset addresses for CIO 208.....	18
4.5.3 Modbus alarm offset addresses for CIO 308.....	19
<b>5. DEIF Open protocol</b>	
5.1 Using the DEIF Open protocol.....	21

# 1. General information

## 1.1 Modbus server

The iE 150 Battery/Solar 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 **iE 150 Battery Solar Modbus server tables** (an Excel spreadsheet) for all the Modbus server addresses.

**NOTE** The iE 150 Modbus client is described in the **iE 150 Battery Solar Modbus client User manual**.

## 1.2 Software version

This document is based on iE 150 software version 1.37.

## 1.3 Disclaimer

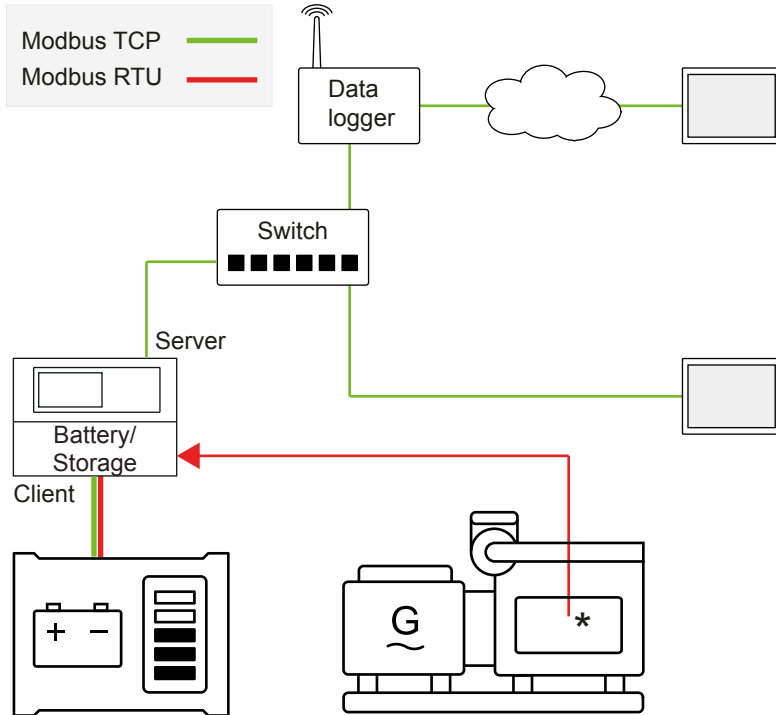
DEIF A/S reserves the right to change any of the contents of this document without prior notice.

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 Battery/Storage controller communication

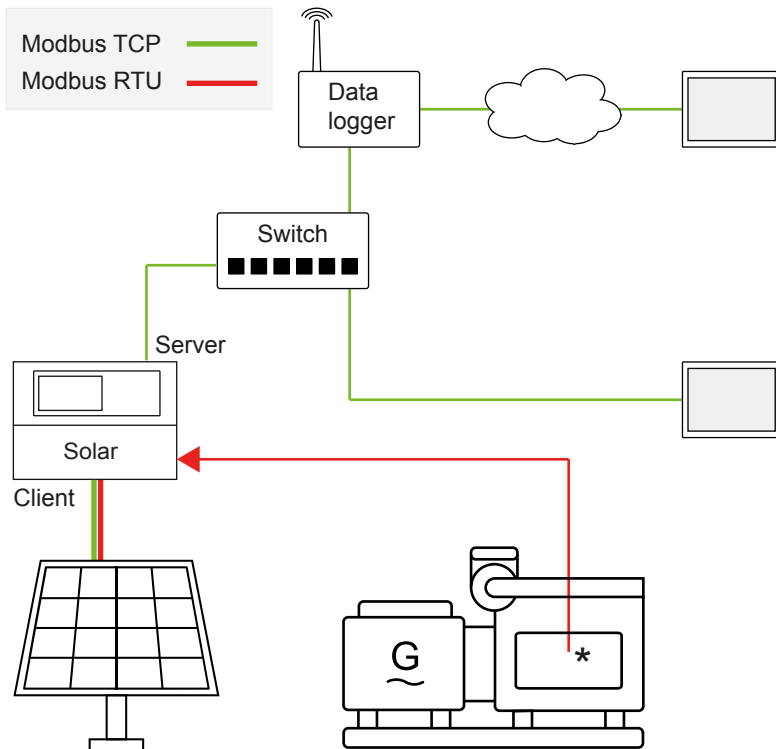
### Example of communication for a single controller application



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

## 2.2 Solar controller communication

### Example of Solar controller communication for a single controller application



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

## 2.3 Connections for Modbus server

The external controllers and/or SCADA system are the Modbus clients, and the iE 150 is the Modbus server.

## 2.4 Wiring



### More information

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

## 2.5 Configuring Modbus connections

### 2.5.1 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 controller Modbus server. The external Modbus client uses this ID to communicate with the controller.
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	

### 2.5.2 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*.

Network parameters	Remote Display	Power Management	NTP parameters
IP address			192.168.18.12
Net mask			255.255.255.0
Gateway			192.168.12.1
DNS Primary IP			8.8.8.8
DNS Secondary IP			8.8.4.4

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

## 3. Data tables

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

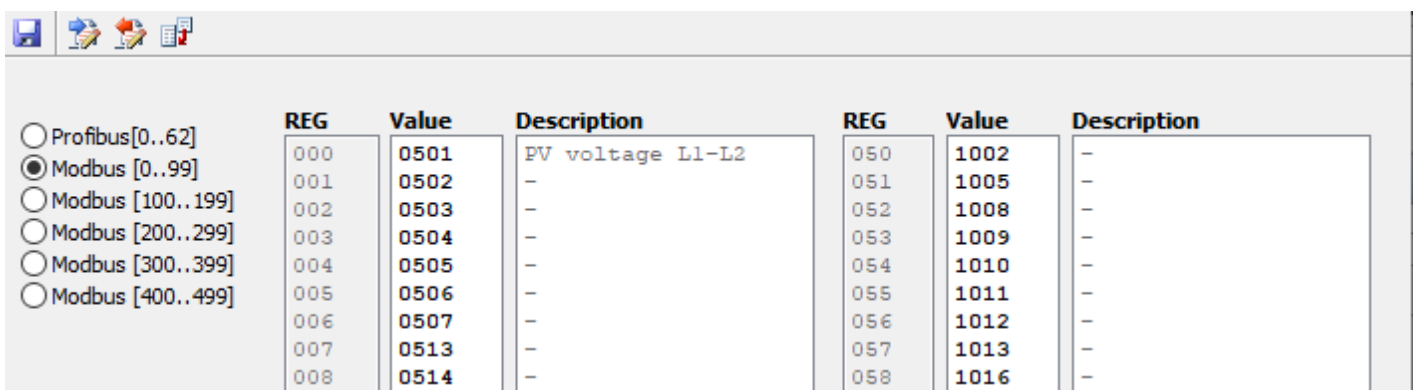
#### 3.1.1 Modbus configurator

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.



	REG	Value	Description	REG	Value	Description
<input type="radio"/> Profibus[0..62]	000	0501	PV voltage L1-L2	050	1002	-
<input checked="" type="radio"/> Modbus [0..99]	001	0502	-	051	1005	-
<input type="radio"/> Modbus [100..199]	002	0503	-	052	1008	-
<input type="radio"/> Modbus [200..299]	003	0504	-	053	1009	-
<input type="radio"/> Modbus [300..399]	004	0505	-	054	1010	-
<input type="radio"/> 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 (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 on the documentation page under **Communication**. The Modbus server tables are stored in an .xlsx file that includes:

- Descriptions
- Discrete output coil (01; 05; 15)
- Digital input contact (02)
- Holding register (03; 06; 16)
- Input register (04)

The number in brackets refers to the Modbus function code (as a decimal value) for the information.

### 3.2.1 Alarm list monitoring

Alarm list monitoring is in function code 04 and Modbus addresses 28000 to 28099. You can use these Modbus addresses to read the **Address** for each active alarm in the controller. Use the **Parameters** tab in the utility software to look up which alarm corresponds to each **Address**.



#### Example of alarm list monitoring

Reading function code 04 and Modbus addresses 28000 to 28099:

- Address 28000: Value **236**
- Addresses 28001 to 28099: Value **0**

Looking at the utility software, address 236 corresponds to channel 3490 Emergency STOP.

iii	Category	Channel	△ Text	Address	Value	Unit	Timer
	Digital In	3490	Emergency STOP	236	N/A		0
	Digital In	3570	M-Logic alarm 01	1722	N/A		10
	Digital In	3580	M-Logic alarm 02	1723	N/A		10

Therefore, the only active alarm in the controller is **Emergency STOP**.

## 3.3 Data type

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

- BOOL: Boolean data
- INT16s: Signed 16-bit integers
- INT16u: Unsigned 16-bit integers
- INT32s: Signed 32-bit integers
- INT32u: Unsigned 32-bit integers
- string: Text data

## 3.4 Data format

The controller 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.

## 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:

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

The table below lists parameter types and the scaling values that are normally used for them. Parameter types that are not scaled, are not listed in the table.

Scaling values normally used by specific parameter types:

Parameter type	Scaling
Phase angle	1
DC supply voltage	1
Frequency	2
cos phi	2
Power factor	2



### Scaling example

The Modbus address for battery frequency (holding register 0x03, 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 to 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 to 5999
Alarm acknowledge	6000 to 7999
Timer output	8000 to 9999
Timer running	10000 to 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 to 3999
Values used	4000 to 5999
Values minimum	6000 to 7999
Values maximum	8000 to 9999

Data to request	Address area
Output A	10000 to 11999
Output B	12000 to 13999
Fail class used	14000 to 15999
Enable	16000 to 17999
Inhibit	18000 to 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 to 3999
Timers maximum	4000 to 5999
Output A minimum	6000 to 7999
Output A maximum	8000 to 9999
Output B minimum	10000 to 11999
Output B maximum	12000 to 13999
Fail class minimum	14000 to 15999
Fail class maximum	16000 to 17999
Timers elapsed time	20000 to 21999
Actual values	22000 to 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 to 3999
Acknowledge alarm	6000 to 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 to 3999
Values used	4000 to 4999
Output A	10000 to 11999
Output B	12000 to 13999

Data to request	Address area
Fail class used	14000 to 15999
Enable	16000 to 17999
Inhibit	18000 to 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.

View mode:  Tree  List

All groups  Battery  Communication  Mains  Ext. Ctrl.  Power Man  Synchronisation  Regulat

Drag a column header here to group by that column

Category	Channel	Text	Address	Value	Unit
Protection	1150	ES U> 1	12	103	%
Protection	1160	ES U> 2	13	105	%
Protection	1170	ES U< 1	14	97	%
Protection	1180	ES U< 2	15	95	%
Protection	1190	ES U< 3	16	95	%
Protection	1201	ES U detect. type	17	0	
Protection	1202	BB U detect. type	31	0	

- 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).

View mode:  Tree  List

All groups  Battery  Communication  Mains  Ext. Ctrl.  Power Man  Synchronisation  Regular

Drag a column header here to group by that column

Category	Channel	Text	Address	Value	Unit
General	6001	Nom. f 1	407	50	Hz
General	6002	Nom. P 1	408	480	kW
General	6003	Nom. I 1	409	867	A
General	6004	Nom. U 1	410	400	V
General	6005	Nom. Q 1	596	480	kvar
General	6006	Nom. S 1	742	480	kVA
General	6007	Enable nom. set	412	0	
General	6008	Nom. I E/N/M 1	1460	867	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

Parameter U< aux. term. 1" (Channel 4960)

Set point : 24.5 V

8 32

Timer : 50 sec

0 999

Fail class : Warning

Output A : Not used

Output B : Not used

Password level : service

Enable

High Alarm

Inverse proportional

Auto acknowledge

Inhibits... "Shutdown

Commissioning

Actual value : 24.2 V

Time elapsed : 6.6 sec (13.2 %)

0 sec 50 sec

Write OK Cancel



#### 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.

## 4.4 Modbus addresses and examples for IOs

The Modbus address offsets for the IOs can be seen on the *I/O setup* pages in the USW.

### Examples



#### Configure the multi input 102 2nd alarm

1. From the USW, the offset for *Multi input 102 2nd alarm* is **269**.
2. To check whether the alarm is enabled (address area 2000 to 3999), use function code **01** and Modbus address  $2000 + 269 = \mathbf{2269}$ .
  - If the controller returns **0**, the alarm is disabled.
  - If the controller returns **1**, the alarm is enabled.
3. To check the fail class (address area 14000 to 15999), use function code **03** and Modbus address  $14000 + 269 = \mathbf{14269}$ .
  - The controller returns an integer. For example, if the fail class is *Warning*, the controller returns **1**.
4. To change the fail class to *Shutdown*, use function code **06** and Modbus address **14269** to send the value **4**.
  - To verify that the fail class has changed, use the USW *I/O setup* page, read the I/O setup from the controller.
5. To enable the alarm, use function code **05** and Modbus address **2269** to send the value **1**.
  - To verify that the alarm is enabled, use the USW *I/O setup* page, read the I/O setup from the controller.



#### Check whether Multi input 108 has a wire failure

1. From the USW, the offset for *Multi input 108 wire fail* is **276**.
2. To check whether the alarm is enabled, use function code **01** and Modbus address **2276**.
3. To see whether the alarm is activated (address area 4000 to 5999), use function code **02** and Modbus address **4276**.
  - To verify the alarm status, use the USW *Logs* page and read the alarm log from the controller.



#### Configure the digital input 51 alarm timer (option M12)

1. From the USW, the offset for *Digital input 51* is **205**.
2. To check the configured alarm timer (address area 2000 to 3999), use function code **03** and Modbus address  $2000 + 205 = \mathbf{2205}$ .
  - If the controller returns **100**, the delay setting is 10 s (since the scaling is 1).
3. To change the delay setting to 5 s, use function code **06** and Modbus address **2205** to send the value **50**.



#### Configure the relay output 59 alarm function (option M12)

1. From the USW, the offset for *Relay output 59* is **331**.
2. To check the alarm function (address area 4000 to 5999), use function code **03** and Modbus address  $4000 + 331 = \mathbf{4331}$ .
  - If the controller returns **0**, the alarm function is *Alarm relay ND*.
3. To change the alarm function to *M-Logic / Limit relay*, use function code **06** and Modbus address **4431** to send the value **4**.

## 4.5 Modbus addresses and example for CIO modules

The tables after the examples show the CIO modules' Modbus offset addresses (they are not shown in the USW or the Modbus tables). For CIO modules, the Modbus reading/writing is the same as for other IOs.

**NOTE** DEIF recommends using the USW to configure the CIO modules.

## Example



### Monitor CIO 116 no. 1 input 14

1. From the table, the offset for *CIO no. 1 input 14* is **1051**.
2. To check whether the alarm is enabled (address area 2000 to 3999), use function code **01** and Modbus address  $2000 + 1051 = \mathbf{3051}$ .
  - If the controller returns **0**, the alarm is disabled.
  - If the controller returns **1**, the alarm is enabled.

### 4.5.1 Modbus alarm offset addresses for CIO 116

Description	Address offset	Description	Address offset
CIO 116 no. 1. Input 10	1047	CIO 116 no. 2. Input 21	1260
CIO 116 no. 1. Input 11	1048	CIO 116 no. 2. Input 22	1261
CIO 116 no. 1. Input 12	1049	CIO 116 no. 2. Input 23	1264
CIO 116 no. 1. Input 13	1050	CIO 116 no. 2. Input 24	1266
CIO 116 no. 1. Input 14	1051	CIO 116 no. 2. Input 25	1267
CIO 116 no. 1. Input 15	1052	CIO 116 no. 2. Input 26	1272
CIO 116 no. 1. Input 16	1053	CIO 116 no. 3. Input 10	1296
CIO 116 no. 1. Input 17	1054	CIO 116 no. 3. Input 11	1316
CIO 116 no. 1. Input 19	1055	CIO 116 no. 3. Input 12	1323
CIO 116 no. 1. Input 20	1056	CIO 116 no. 3. Input 13	1324
CIO 116 no. 1. Input 21	1057	CIO 116 no. 3. Input 14	1325
CIO 116 no. 1. Input 22	1058	CIO 116 no. 3. Input 15	1327
CIO 116 no. 1. Input 23	1059	CIO 116 no. 3. Input 16	1328
CIO 116 no. 1. Input 24	1060	CIO 116 no. 3. Input 17	1329
CIO 116 no. 1. Input 25	1061	CIO 116 no. 3. Input 19	1330
CIO 116 no. 1. Input 26	1062	CIO 116 no. 3. Input 20	1331
CIO 116 no. 2. Input 10	1194	CIO 116 no. 3. Input 21	1332
CIO 116 no. 2. Input 11	1195	CIO 116 no. 3. Input 22	1333
CIO 116 no. 2. Input 12	1196	CIO 116 no. 3. Input 23	1334
CIO 116 no. 2. Input 13	1197	CIO 116 no. 3. Input 24	1362
CIO 116 no. 2. Input 14	1198	CIO 116 no. 3. Input 25	1363
CIO 116 no. 2. Input 15	1199	CIO 116 no. 3. Input 26	1364
CIO 116 no. 2. Input 16	1200	CIO 116 no. 1 missing	957*
CIO 116 no. 2. Input 17	1237	CIO 116 no. 2 missing	960*
CIO 116 no. 2. Input 19	1238	CIO 116 no. 3 missing	963*
CIO 116 no. 2. Input 20	1257		

**NOTE** \* Address offset 957, 960 and 963 are read-only.

## 4.5.2 Modbus alarm offset addresses for CIO 208

Description	Address offset
CIO 208 no. 1 missing	958*
CIO 208 no. 2 missing	961*
CIO 208 no. 3 missing	964*

**NOTE** \* Address offset 958, 961 and 964 are read-only.

### 4.5.3 Modbus alarm offset addresses for CIO 308

Description	Address offset	Description	Address offset
CIO 308 no. 1. Input 8 alarm 1	999	CIO 308 no. 3. Input 17 alarm 1	1567
CIO 308 no. 1. Input 8 alarm 2	1000	CIO 308 no. 3. Input 17 alarm 2	1568
CIO 308 no. 1. Input 11 alarm 1	1001	CIO 308 no. 3. Input 20 alarm 1	1569
CIO 308 no. 1. Input 11 alarm 2	1002	CIO 308 no. 3. Input 20 alarm 2	1570
CIO 308 no. 1. Input 14 alarm 1	1003	CIO 308 no. 3. Input 23 alarm 1	1571
CIO 308 no. 1. Input 14 alarm 2	1004	CIO 308 no. 3. Input 23 alarm 2	1572
CIO 308 no. 1. Input 17 alarm 1	1005	CIO 308 no. 3. Input 26 alarm 1	1573
CIO 308 no. 1. Input 17 alarm 2	1006	CIO 308 no. 3. Input 26 alarm 2	1574
CIO 308 no. 1. Input 20 alarm 1	1007	CIO 308 no. 3. Input 29 alarm 1	1575
CIO 308 no. 1. Input 20 alarm 2	1008	CIO 308 no. 3. Input 29 alarm 2	1576
CIO 308 no. 1. Input 23 alarm 1	1009	CIO 308 no. 1 missing	959*
CIO 308 no. 1. Input 23 alarm 2	1010	CIO 308 no. 1 missing	962*
CIO 308 no. 1. Input 26 alarm 1	1011	CIO 308 no. 1 missing	965*
CIO 308 no. 1. Input 26 alarm 2	1012	CIO 308 no. 1. Input 8 wire fail	1031
CIO 308 no. 1. Input 29 alarm 1	1013	CIO 308 no. 1. Input 11 wire fail	1032
CIO 308 no. 1. Input 29 alarm 2	1014	CIO 308 no. 1. Input 14 wire fail	1033
CIO 308 no. 2. Input 8 alarm 1	1015	CIO 308 no. 1. Input 17 wire fail	1034
CIO 308 no. 2. Input 8 alarm 2	1016	CIO 308 no. 1. Input 20 wire fail	1035
CIO 308 no. 2. Input 11 alarm 1	1017	CIO 308 no. 1. Input 23 wire fail	1036
CIO 308 no. 2. Input 11 alarm 2	1018	CIO 308 no. 1. Input 26 wire fail	1037
CIO 308 no. 2. Input 14 alarm 1	1019	CIO 308 no. 1. Input 29 wire fail	1038
CIO 308 no. 2. Input 14 alarm 2	1020	CIO 308 no. 2. Input 8 wire fail	1039
CIO 308 no. 2. Input 17 alarm 1	1021	CIO 308 no. 2. Input 11 wire fail	1040
CIO 308 no. 2. Input 17 alarm 2	1022	CIO 308 no. 2. Input 14 wire fail	1041
CIO 308 no. 2. Input 20 alarm 1	1023	CIO 308 no. 2. Input 17 wire fail	1042
CIO 308 no. 2. Input 20 alarm 2	1024	CIO 308 no. 2. Input 20 wire fail	1043
CIO 308 no. 2. Input 23 alarm 1	1025	CIO 308 no. 2. Input 23 wire fail	1044
CIO 308 no. 2. Input 23 alarm 2	1026	CIO 308 no. 2. Input 26 wire fail	1045
CIO 308 no. 2. Input 26 alarm 1	1027	CIO 308 no. 2. Input 29 wire fail	1046
CIO 308 no. 2. Input 26 alarm 2	1028	CIO 308 no. 3. Input 8 wire fail	1083
CIO 308 no. 2. Input 29 alarm 1	1029	CIO 308 no. 3. Input 11 wire fail	1085
CIO 308 no. 2. Input 29 alarm 2	1030	CIO 308 no. 3. Input 14 wire fail	1086
CIO 308 no. 3. Input 8 alarm 1	1531	CIO 308 no. 3. Input 17 wire fail	1110
CIO 308 no. 3. Input 8 alarm 2	1532	CIO 308 no. 3. Input 20 wire fail	1391
CIO 308 no. 3. Input 11 alarm 1	1560	CIO 308 no. 3. Input 23 wire fail	1392
CIO 308 no. 3. Input 11 alarm 2	1563	CIO 308 no. 3. Input 26 wire fail	1427
CIO 308 no. 3. Input 14 alarm 1	1565	CIO 308 no. 3. Input 29 wire fail	1529
CIO 308 no. 3. Input 14 alarm 2	1566		

**NOTE** \* Address offset 959, 962 and 965 are read-only.

## 5. DEIF Open protocol

### 5.1 Using the DEIF Open protocol

The controller can use its Modbus server interface to control the PV/ESS. The controller 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 controller. Modbus addresses 46000 to 46999 (function code 0x03) are used for this.

The DEIF open protocol includes the following function groups:

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

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

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