

**Honeywell**

THE POWER OF **CONNECTED**

Flow Computer  
Geräteserie enCore  
ZM1, BM1, MC1, FC1

---

Handbuch  
AFB Modbus

---

## Kontakt

Elster GmbH (Hersteller)

Steinern Straße 19-21

55252 Mainz-Kastel/Germany

Telefon: +49 6134 605-0

E-Mail: [info@elster.com](mailto:info@elster.com)

Website: [www.elster-instromet.com](http://www.elster-instromet.com)

Technischer Support Flow Computer und Gasqualitätsmessung

Telefon: +49 231 937110-88

E-Mail: [ElsterSupport@Honeywell.com](mailto:ElsterSupport@Honeywell.com)

Website: [www.elster-instromet.com](http://www.elster-instromet.com)

# Inhalt

1	Über diese Anleitung	5
2	Funktionale Beschreibung	7
2.1	Die Modbus-Protokollvarianten im Überblick	9
2.1.1	Modbus RTU/Modbus ASCII (seriell)	10
2.1.2	Modbus TCP	10
2.2	Beispielszenarien	11
2.2.1	Datenübertragung via Modbus RTU/Modbus ASCII	11
2.2.2	Datenübertragung via Modbus TCP	12
3	Besonderheit des AFB: Register- und Archivbereiche	13
3.1	Register-/Archivbereiche und Register – eine Gegenüberstellung	14
3.2	Register- oder Archivbereiche definieren	16
3.3	Benutzerdefinierte Register bereitstellen (Registerbereich)	23
3.4	Archive über Modbus auslesen (Archivbereich)	25
3.5	Exportregister am Gerät prüfen	28
4	Anzeige und Bedienung	30
4.1	Anzeigen in der Übersicht	31
4.2	Anzeigen im Detail	32
5	FAQs	34
5.1	Wie unterscheiden sich im AFB Modbus der Normal- und der Expertenmodus?	34
5.2	Wie parametriere ich im Normalmodus in wenigen Schritten einen FC1 als Modbus-Slave an COM2?	35
5.3	Wie synchronisiere ich die Systemzeit über Modbus? (Normalmodus)	38
5.4	Wie kann ich einen Protokollfehler (Meldung) beheben?	40

---

6	Anhang	43
6.1	Nomenklatur	43
6.2	Unterstützte Modbus-Telegrammtypen	43
6.3	Datenbeschreibung der Registerformate	44
6.3.1	Byte-, Word- und DWord-Reihenfolge	44
6.3.2	Unterstützte Datentypen	45
7	Literaturverzeichnis	48
8	Index	49

# 1 Über diese Anleitung

Das enCore FC-Handbuch ist modular aufgebaut. Einen Überblick über das enCore/enSuite-Konzept und den Aufbau des Handbuchs, Sicherheitsinformationen sowie die Textkennzeichnung erhalten Sie in der „Betriebsanleitung“ des enCore FC.

Der vorliegende Band beschreibt die grundlegende Funktionalität und Bedienung des AFB Modbus.



## Der AFB Modbus in der Geräteserie enCore FC

Der AFB Modbus steht für alle Geräte der Geräteserie enCore Flow Computer (kurz: enCore FC) zur Verfügung. Welche Funktionen ein Gerät im Einzelnen unterstützt, hängt von seinem Gerätetyp ab und ist im Detail in der Online-Hilfe beschrieben.

Die Bedeutung der einzelnen Parameter ist ausführlich in der Online-Hilfe von enSuite dokumentiert, deshalb wird auf die Parametrierung in diesem Dokument nur beispielhaft eingegangen. Parameter, die für Spezialanwendungen vorgesehen sind, werden ebenfalls in der Online-Hilfe beschrieben.



## Aufruf der Online-Hilfe

In enSuite rufen Sie die allgemeine Hilfe über den Menüeintrag **Hilfe** –  **Online-Hilfe anzeigen** auf. Die kontextsensitive Hilfe öffnen Sie direkt im Parametrierfenster aus dem gewünschten Zweig mit **[F1]**.

Dieser Teil der Dokumentation wendet sich an Fachpersonal, das nach erfolgter Montage des Geräts und Installation der aktuellen enSuite-Version auf dem PC für Servicetätigkeiten folgender Aufgaben verantwortlich ist:

- Anpassung der Geräteparametrierung an die Messaufgabe
- Test aller Datenpunkte und Inbetriebnahme
- weitere Servicemaßnahmen

Die Abbildungen in dieser Anleitung dienen der Darstellung der erläuterten Sachverhalte, daher können sie je nach Konfiguration des Geräts und enSuite abweichen.

**NOT UP-TO-DATE**  
[www.docuthek.com](http://www.docuthek.com)

## 2 Funktionale Beschreibung

Dieses Handbuch beschreibt die Funktion und Arbeitsweise des AFB Modbus bei enCore-Geräten. Der AFB basiert auf dem Modbus-Protokoll, das den Datenaustausch mit externen Geräten ermöglicht.<sup>1</sup> Damit kann der AFB vorteilhaft verwendet werden, um applikationsspezifisch den Datenaustausch mit rückwirkungsfreien Zusatzeinrichtungen wie Stationsleittechnik und Fernwirkgeräten zu ermöglichen.

Der AFB Modbus verwaltet benutzerdefinierte Register, die Sie bei der Parametrierung flexibel mit enCore-internen Werten verknüpfen können. Dabei können Sie Modbus-Adressen, Datenformate, Bytereihenfolge usw. frei parametrieren. Des Weiteren können Sie Archive über Modbus auslesen. Die Kommunikation erfolgt über ein TCP/IP-Netzwerk oder eine serielle Verbindung (Punkt-zu-Punkt oder Bus).



### Exkurs: Anbindung von intelligenten Messgeräten über Modbus-Scripte

Das Modbus-Protokoll wird ebenfalls häufig verwendet, um Messgeräte wie Ultraschallzähler und Prozessgas-Chromatographen („intelligente Messgeräte“) an enCore-Geräte anzuschließen. Hier geben die Hersteller für einzelne Gerätetypen die spezifischen Details für die Modbus-Kommunikation und deren Auswertung wie z. B. Registerbelegung fest vor. Elster setzt diese Herstellervorgaben und typischen Registerbelegungen in Scripten um. Die Scripte für die gebräuchlichsten intelligenten Messgeräte sind bereits im Auslieferungszustand von enCore-Geräten enthalten.

---

<sup>1</sup> Die Implementierung des Modbus-Protokolls basiert auf den jeweiligen Modbus Spezifikationen. (⇒ Anhang [8 Literaturverzeichnis](#), S. 22)

Für die Anbindung eines PGCs oder USMs wählen Sie bei der Parametrierung für die entsprechende serielle oder LAN-Schnittstelle das erforderliche Script aus, eine weitere Modbus-Parametrierung entfällt. Nur in Ausnahmefällen wird für intelligente Messgeräte ein `AFB Modbus` benötigt, und zwar nur dann, wenn ein Script nicht alle benötigten Register bereitstellt.

Die Anbindung von intelligenten Messgeräten wie Gasbeschaffenheitsmessgeräten oder Ultraschall-Gaszählern über gerätespezifische Modbus-Scripte ist nicht Bestandteil dieses Handbuchs.

⇒ Band „Grundsystem mit SFBs“ des FC-Handbuchs

Das Modbus-Protokoll ist ein Kommunikationsprotokoll, mit dem eine Master/Slave- bzw. eine Client/Server-Kommunikation zwischen Geräten realisiert werden kann, die in einem Netzwerk verbunden sind. Es handelt sich um ein offenes Protokoll, das 1979 von Gould-Modicon zunächst für die serielle Kommunikation zwischen speicherprogrammierbaren Steuerungen entwickelt wurde. Seit 2007 ist die Version Modbus TCP Teil der Norm IEC 61158.

enCore-Geräte unterstützen die drei Varianten des Modbus-Protokolls:

- Modbus ASCII
- Modbus RTU (engl. **R**emote **T**erminal **U**nit)
- Modbus TCP

Generell gilt, pro `AFB Modbus` können Sie...

... bei serieller Kommunikation jeweils

- 1 Modbus-Slave ( $\hat{=}$  1 Gerät) mit 1 Slave-ID abbilden.
- 1 Modbus-Master ( $\hat{=}$  1 Gerät) abbilden, der bis zu 10 Slaves über ihre Slave-ID adressieren kann.

... über TCP/IP jeweils

- 1 Modbus-Server ( $\hat{=}$  1 Gerät) mit IP-Adresse und Port abbilden, auf den bis zu 10 Clients gleichzeitig zugreifen können.

- 1 Modbus-Client ( $\cong$  1 Gerät) abbilden, der mehrere Server über ihre IP-Adresse und Port adressieren kann.

## 2.1 Die Modbus-Protokollvarianten im Überblick

Das Modbus-Protokoll basiert auf der Master/Slave- bzw. Client/Server-Architektur. Es ist ein sog. Single-Master-Protokoll, d.h. der Master bzw. Client steuert die gesamte Übertragung und überwacht z.B. evtl. auftretende Timeouts. Angeschlossene Geräte dürfen nur nach Anforderung durch den Master bzw. Client Telegramme versenden. Als fehlerhaft erkannte Telegramme werden grundsätzlich verworfen und nach einer Wartezeit neu angefordert.

Bei allen Protokollvarianten gehören zu den Modbus-Daten der Funktionscode, Startadresse oder Anzahl der Register. Die Daten liegen in einem oder mehreren Registern mit je 16-Bit Länge; die Belegung der Register mit Anwendungsdaten parametrieren Sie bei Inbetriebnahme.

Die allgemeinen Einstellungen der seriellen bzw. der LAN-Schnittstelle des enCore-Geräts werden im Normalmodus unter **Start (E/A-Konfiguration)** und im Expertenmodus im **Grundsystem – E/A** für die gewünschte Karte parametriert. Die Kommunikationseinstellungen für Master bzw. Client und Slave oder Server parametrieren Sie im **AFB Modbus**.

⇒ Auf die Parametrierung der Kommunikationseinstellungen wird im Handbuch nicht im Detail eingegangen, da Sie ausführlich in der Online-Hilfe dokumentiert ist.



### Slave bzw. Server im Normalmodus parametrieren

Beachten Sie, dass Sie im Normalmodus bei serieller Kommunikation nur einen (lokalen) Slave und bei TCP/IP nur einen (lokalen) Modbus-Server parametrieren können. Um einen Modbus-Master bzw. Modbus-Client zu parametrieren wechseln Sie in den Expertenmodus.

⇒ [FAQ 5.1 Wie unterscheiden sich im AFB Modbus der Normal- und der Expertenmodus?](#) (S. 34)

### 2.1.1 Modbus RTU/Modbus ASCII (seriell)

Die Varianten Modbus RTU und Modbus ASCII unterscheiden sich nur unwesentlich im Framing. Während Modbus ASCII bei der Übertragung den ASCII-Zeichensatz verwendet, ist die Übertragung bei Modbus RTU binär. Beide verwenden eine serielle asynchrone Punkt-zu-Punkt Datenverbindung über RS232C oder eine Mehrpunktverbindung über RS422 oder RS485.

Jeder Slave benötigt eine eindeutige ID (**Slave-ID**), durch die der Master ihn am Bus identifizieren kann – die ID Null (0) ist reserviert.

Die seriellen Kommunikationseinstellungen für Modbus RTU und Modbus ASCII müssen auf Sender- und Empfängerseite gleich definiert sein. Dazu gehören Treiber-Modus, Baudrate, Anzahl der Datenbits (8 bei Modbus RTU, 7 bei Modbus ASCII), Parität und Stoppbits.

### 2.1.2 Modbus TCP

Modbus TCP benutzt eine LAN-Schnittstelle der CPU oder einer ESER4 und verwendet das Übertragungsprotokoll TCP/IP. Die Modbus-Daten werden als TCP/IP-Paket übertragen. Standardmäßig ist der Port 502 für den Modbus TCP-Server reserviert.

Jeder Datenaustausch eines Telegramms erfolgt Punkt-zu-Punkt, wobei einer der beiden Teilnehmer an der Modbus-Kommunikation ein Client und der andere ein Server ist. Da die IP-Verbindung nach jeder Übertragung wieder abgebaut wird, kann danach eine Übertragung mit einem anderen Teilnehmer am Netzwerk stattfinden.

Die Adressierung der Modbus-Geräte erfolgt über die IP-Adresse eines Gerät, eine weitere ID wie die Slave-ID bei Modbus RTU/Modbus ASCII wird eigentlich nicht benötigt, ist aber optional parametrierbar.

## 2.2 Beispielszenarien

### 2.2.1 Datenübertragung via Modbus RTU/Modbus ASCII

Folgendes Beispiel zeigt die Adressierung an einem seriellen Bus über Modbus RTU oder Modbus ASCII. Voraussetzung für den Datenaustausch ist, dass jeder Slave eine eindeutige Slave-ID besitzt, die jeweils im Master hinterlegt ist:

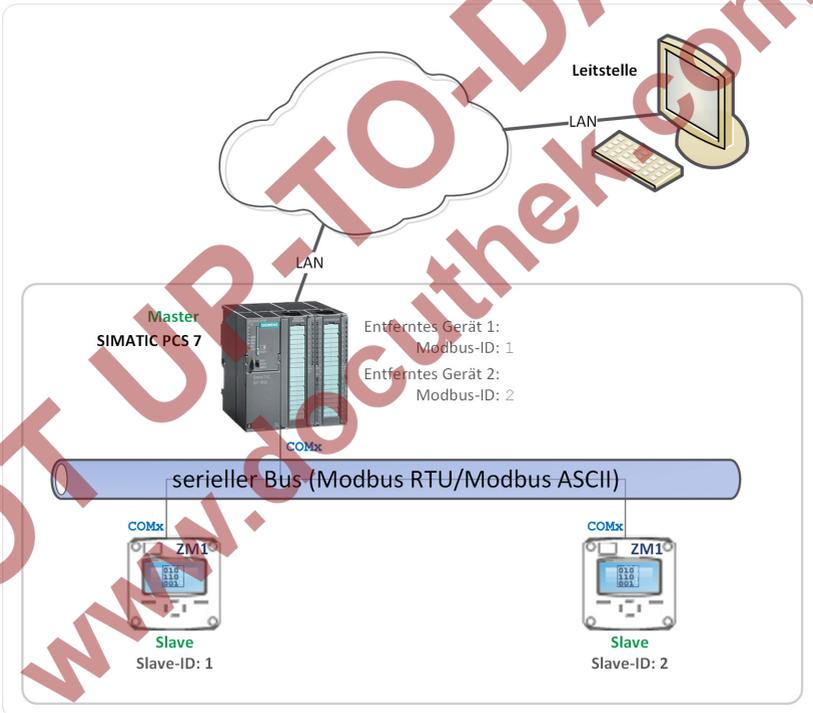


Abb. 2-1: Datenübertragung via Modbus RTU/Modbus ASCII – Beispiel

## 2.2.2 Datenübertragung via Modbus TCP

Folgendes Beispiel zeigt die Adressierung in einem TCP-Netzwerk über Modbus TCP. Ein Client adressiert einen Server über dessen (eindeutige) IP-Adresse und Portnummer. Voraussetzung für den Datenaustausch ist, dass diese Angaben sowohl beim Server als auch Client hinterlegt sind:

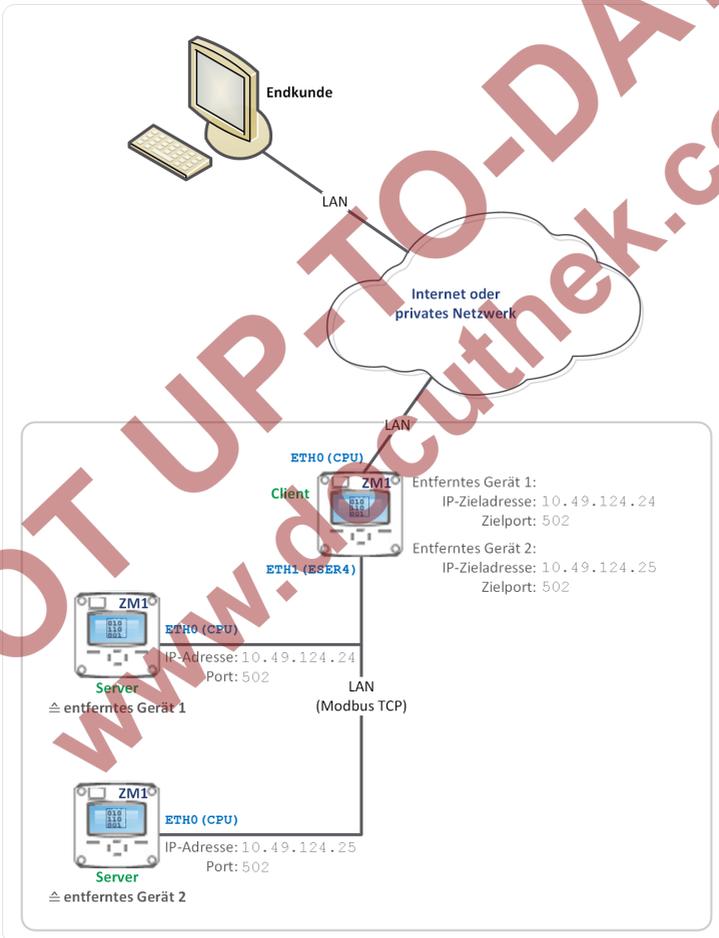


Abb. 2-2: Datenübertragung via Modbus TCP – Beispiel

### 3 Besonderheit des AFB: Register- und Archivbereiche

Die Modbus-Spezifikation definiert den Datenframe für die Datenübertragung, beschreibt jedoch keine Datenformate. Deshalb kommen in der Praxis viele verschiedene Formate zum Einsatz.

Um bei der Datenbeschreibung flexibel zu sein und gleichzeitig die Parametrierung so einfach wie möglich zu halten, führt der AFB Modbus neben den einzelnen Registern zusätzlich Register- und Archivbereiche ein. Jedes Register ist dabei genau einem Bereich zugeordnet. Das hat den Vorteil, dass alle Register eines Bereichs die allgemeineren Eigenschaften dieses Bereichs (z. B. die Datenformate) erben.

Die Parametrierung im AFB Modbus erfolgt immer zweistufig:

(1) Register- oder Archivbereich definieren.

Zunächst legen Sie einen Bereich an und definieren Eigenschaften, die für alle Register dieses Bereiches gültig sind.

(2) Register parametrieren.

Anschließend definieren Sie für diesen Bereich die einzelnen Register und legen die speziellen Eigenschaften der einzelnen Register fest.



#### Register-Startadresse gemäß Modbus-Spezifikation mit einem Offset von Minus 1

Modbus verwendet die Startadresse eines Registers, um es auszulesen oder zu schreiben. Beachten Sie, dass Registernummern immer bei 1 beginnen, die Startadresse bei Modbus aber um 1 kleiner ist als die Registernummer. Das bedeutet, dass im Modbustelegramm als Startadresse die „Registernummer – 1“ steht. Damit verhält sich der AFB Modbus gemäß Modbus-Standard.

**Zum Beispiel:**

Die momentane Systemzeit hat die Registernummer 4200 und entsprechend als Modbus-Startadresse 4199 gesendet.

Dieses Kapitel ist in folgende Abschnitte unterteilt:

- ⇒ [3.1 Register-/Archivbereiche und Register – eine Gegenüberstellung](#) (S. 14)
- ⇒ [3.2 Register- oder Archivbereiche definieren](#) (S. 16)
- ⇒ [3.3 Benutzerdefinierte Register bereitstellen \(Registerbereich\)](#) (S. 23)
- ⇒ [3.4 Archive über Modbus auslesen \(Archivbereich\)](#) (S. 25)

### 3.1 Register-/Archivbereiche und Register – eine Gegenüberstellung

In diesem Abschnitt werden die Unterschiede zwischen Register- und Archivbereichen und Registern herausgestellt.

**Begriffsklärung: Datenimport und Datenexport**

Um Missverständnisse zu vermeiden, werden in diesem Handbuch die Begriffe „Schreiben“ und „Lesen“ von Registern vermieden, da die Bedeutung je nach Kommunikationsmodus (Master/Client bzw. Slave/Server) variiert. Um die Kommunikationsrichtung anzuzeigen, verwenden wir stattdessen die Begriffe Datenimport und Datenexport:

-  Datenimport  
Hier werden die Modbus-Daten vom externen Gerät zum enCore-Gerät übertragen.
-  Datenexport  
Hier werden die Modbus-Daten vom enCore-Gerät zum externen Gerät übertragen.

Register-/Archivbereiche	Register
<p><b>Anzahl</b></p> <ul style="list-style-type: none"> <li>(Master/Client) bis zu 10 Archiv- bzw. Registerbereiche pro entferntem Gerät</li> <li>(Slave/Server) bis zu 30 Archiv- bzw. Registerbereiche pro lokalem Gerät</li> </ul>	<p><b>Anzahl</b></p> <ul style="list-style-type: none"> <li>insgesamt max. 2.000 Register (über alle Register- und Archivbereiche hinweg)</li> </ul>
<p><b>2 Arten von Bereichen</b></p> <ul style="list-style-type: none"> <li><b>Registerbereich</b> <ul style="list-style-type: none"> <li>generell für benutzerdefinierte Register</li> <li>sowohl für lokales Gerät (Slave/Server) als auch entfernte Geräte (Master/Client)</li> <li>unterstützt  Importregister und  Exportregister</li> </ul> </li> <li><b>Archivbereich</b> <ul style="list-style-type: none"> <li>speziell für das Auslesen von Archivgruppen des AFB Benutzerarchive</li> <li>nur für lokales Gerät (Slave/Server)</li> <li>unterstützt nur  Exportregister</li> </ul> </li> </ul>	<p><b>2 Arten von Registern</b></p> <ul style="list-style-type: none"> <li> <b>Importregister</b> Ein Wert wird über Modbus in das Register importiert und zur Weiterverarbeitung im enCore-Gerät abgebildet.</li> <li> <b>Exportregister</b> Das enCore-Gerät bildet einen Wert im Exportregister ab und stellt ihn über Modbus zur Verfügung.</li> </ul>
<p><b>Jeder Bereich</b></p> <ul style="list-style-type: none"> <li>... kann eine variable Anzahl an Registern verwalten – einzige Einschränkung: Anzahl von 2.000 Registern darf nicht überschritten werden.</li> </ul>	<p><b>Jedes Register</b></p> <ul style="list-style-type: none"> <li>... ist immer einem Register- oder einem Archivbereich zugeordnet und erbt die allgemeinen Eigenschaften seines Bereichs und Datentyps.</li> </ul>
<p><b>Pro Bereich</b></p> <ul style="list-style-type: none"> <li>... definieren Sie jeweils eine allgemeine Datenbeschreibung, die alle Register dieses Bereichs erben.</li> </ul>	<p><b>Pro Register</b></p> <ul style="list-style-type: none"> <li>... definieren Sie die speziellen Eigenschaften des Registers <b>Name</b> und <b>Registernummer</b>.</li> </ul>

Register-/Archivbereiche	Register
<ul style="list-style-type: none"> <li>• ... legen Sie allgemeine Datenbeschreibung mit folgenden Eigenschaften fest:               <ul style="list-style-type: none"> <li>– <b>Registerlänge:</b> 16-Bit, 32-Bit, 64-Bit</li> <li>– <b>Byte-/Word-/DWord-Lage:</b> HI/LO (Big-Endian), LO/HI (Little-Endian)</li> <li>– <b>Datenformate</b> für: Messwerte, Zähler und Eventzähler, Bitleisten und Status sowie Zeit und Datum</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• ... legen Sie die datentyp-spezifischen Eigenschaften fest, die für Import- und Exportregister variieren können.</li> </ul> <p style="margin-left: 20px;">Am Beispiel eines <b>Import Messwerts:</b></p> <ul style="list-style-type: none"> <li>– <b>Physikalische Größe</b></li> <li>– <b>Skalierungsfaktor und Skalierungsoffset</b></li> <li>– <b>Untere Begrenzung und obere Begrenzung</b></li> </ul>
<p>Pro Archivbereich ...</p> <ul style="list-style-type: none"> <li>• .. können Sie genau 1 Archivgruppe des  AFB Benutzerarchive abbilden.</li> <li>• ... legen Sie fest, ob die Archivgruppe adress- oder registerbasiert auf Modbus-Register abgebildet wird.</li> </ul>	

Tabelle 3-1: Register- und Archivbereiche vs. einzelne Register – eine Gegenüberstellung

## 3.2 Register- oder Archivbereiche definieren

Für jeden Registerbereich und Archivbereich legen Sie im ersten Schritt die allgemeinen Eigenschaften und Datenformate fest, die für alle Register dieses Bereich gültig sind.

Archivbereiche haben zusätzlich noch weitere archivspezifische Eigenschaften. Diese werden im ⇒ Kapitel [3.4 Archive über Modbus auslesen \(Archivbereich\)](#) (S. 25) beschrieben.

## Vorgehen in enSuite

### Voraussetzungen

- Die Kommunikationseinstellungen in  **E/A-Konfiguration** und  **Modbus <x>** sind parametrierbar.
- *Nur für Archivbereiche*  
Die gewünschten Archivgruppen sind im  **AFB Benutzerarchive** bereits parametrierbar.
- Die Kommunikationseinstellungen in  **E/A-Konfiguration** und  **Modbus <x>** sind parametrierbar.
- *für lokales Gerät (Slave oder Server)*  
In der Parametrierung  ist der Ordner des AFB **Modbus** geöffnet:  **<Gerät>** – [ **<Gruppe>** –]  **Modbus [<x>]** – **Kommunikationsmodus: Slave oder Server** –  **Lokales Gerät** –  **Registerbereiche** geöffnet  
ODER  
*für entfernte Geräte (Master oder Client)*  
In der Parametrierung  ist der Ordner des AFB **Modbus** geöffnet:  **<Gerät>** – [ **<Gruppe>** –]  **Modbus [<x>]** – **Kommunikationsmodus: Master oder Client** –  **Entfernte Geräte** –  **Entferntes Gerät** geöffnet.

Um einen neuen Bereich hinzuzufügen, ...

- ▶ ... wechseln Sie auf die Registerkarte **Parameter**.
- ▶ Im Bereich **Registerbereiche** fügen Sie mit dem  Pluszeichen einen neuen Bereich hinzu.
- ✓ Im Parametrierfenster wurde ein neuer Ordner  **Bereich <x>** **[: Registerbereich]** eingefügt.
- ▶ Öffnen Sie im Ordner  **Bereich <x>[: [Registerbereich]** die Registerkarte **Parameter**.
- ✓ Hier definieren Sie die allgemeinen Eigenschaften und Datenformate für diesen Bereich.

- ▶ *Nur für lokales Gerät (Slave oder Server)*  
Legen Sie mit der Auswahlliste **Bereich <x>** fest, ob Sie einen **Registerbereich** oder **Archivbereich** definieren.
- ✓ Nur für den Archivbereich werden zusätzliche Parameter eingeblendet.
- ▶ Legen Sie mit der Auswahlliste **Registerlänge** fest, ob die Daten in Form von **16-Bit**-, **32-Bit**- oder **64-Bit**-Registern übertragen werden. *Standardeinstellung ist 16-Bit.*

Um für die Dateneinheiten Byte (8-Bit), Word (16-Bit) und DWord (32-Bit) die Position des höher- und niedrigwertigen Bytes festzulegen, ...

- ▶ ... wählen Sie in den Auswahllisten **Byte-Lage**, **Word-Lage** und **DWord-Lage** den Eintrag ...
  - ... **HI/LO (Big-Endian)**  
wenn das höchstwertige Byte an der Anfangsadresse liegt.
  - ... **LO/HI (Little-Endian)** (*Standardeinstellung*)  
wenn das kleinstwertige Byte auf der Anfangsadresse liegt.



### So parametrieren Sie bestimmte Byte-Reihenfolgen eines Register- oder Archivbereichs

**Verwendete Reihenfolge**  
(z.B. von einem GC)

**Parametrierung in enSuite**

Verwendete Reihenfolge (z.B. von einem GC)	Byte-Lage	Word-Lage	Dword-Lage
1234	HI/LO (Big-Endian)	HI/LO (Big-Endian)	HI/LO (Big-Endian)
4321	LO/HI (Little-Endian)	LO/HI (Little-Endian)	LO/HI (Little-Endian)
2143	LO/HI (Little-Endian)	HI/LO (Big-Endian)	HI/LO (Big-Endian)
3412	HI/LO (Big-Endian)	LO/HI (Little-Endian)	LO/HI (Little-Endian)

- Legen Sie in der Auswahlliste **Registerformat (Messwerte)** den Datentyp fest, mit dem Messwerte in Registern abgebildet werden:
- **Word**  
als ganze Zahl mit Wort-Länge 16 Bit und Wertebereich  $-2^{15} \dots 2^{15} - 1$
  - **DWord**  
als ganze Zahl mit Wort-Länge 32 Bit und Wertebereich  $-2^{31} \dots 2^{31} - 1$
  - **QWord**  
als ganze Zahl mit Wort-Länge 64 Bit und Wertebereich  $-2^{63} \dots 2^{63} - 1$
  - **Float S5**  
als Gleitkommazahl 32 Bit (Siemens Datentyp)
  - **Float** (*Standardeinstellung*)  
als normalisierte Gleitkommazahl 32 Bit (single precision) nach IEEE 754
  - **Double**  
als normalisierte Gleitkommazahl mit 64 Bit (double precision) nach IEEE 754
- Legen Sie in der Auswahlliste **Registerformat (Zähler/Eventzähler)** den Datentyp fest, mit dem Messwerte in Registern abgebildet werden:
- **Word**  
als ganze Zahl mit Wort-Länge 16 Bit und Wertebereich  $-2^{15} \dots 2^{15} - 1$
  - **DWord** (*Standardeinstellung*)  
als ganze Zahl mit Wort-Länge 32 Bit und Wertebereich  $-2^{31} \dots 2^{31} - 1$
  - **QWord**  
als ganze Zahl mit Wort-Länge 64 Bit und Wertebereich  $-2^{63} \dots 2^{63} - 1$
  - **Float S5**  
als Gleitkommazahl 32 Bit (Siemens Datentyp)

- **Float**  
als normalisierte Gleitkommazahl 32 Bit (single precision) nach IEEE 754
- **Double**  
als normalisierte Gleitkommazahl mit 64 Bit (double precision) nach IEEE 754
- **BCD8**  
als binär codierte Dezimalzahl 8 Stellen, Wortlänge 32 Bit und Wertebereich 0 ...  $10^9 - 1$
- **BCD16**  
als binär codierte Dezimalzahl 16 Stellen, Wortlänge 64 Bit und Wertebereich 0 ...  $10^{17} - 1$

► (ab AFB-Version 03-12)

Legen Sie in der Auswahlliste **Registerformat für Zähler**

**(Nachkommateil)** den Datentyp fest, mit dem Nachkommateile von Zählwerten (solche Werte sind also immer  $< 1$ ) in Registern abgebildet werden:

- **Float**  
als normalisierte Gleitkommazahl 32 Bit (single precision) nach IEEE 754
- **Double**  
als normalisierte Gleitkommazahl mit 64 Bit (double precision) nach IEEE 754

► Legen Sie in der Auswahlliste **Registerformat (Bitleisten/Status)** den Datentyp fest, mit dem Messwerte in Registern abgebildet werden:

- **Word** (StandardEinstellung)  
als ganze Zahl mit Wort-Länge 16 Bit und Wertebereich  $-2^{15} \dots 2^{15} - 1$
- **DWord**  
als ganze Zahl mit Wort-Länge 64 Bit und Wertebereich  $-2^{63} \dots 2^{63} - 1$
- **QWord**  
als ganze Zahl mit Wort-Länge 64 Bit und Wertebereich  $-2^{63} \dots 2^{63} - 1$

- Legen Sie in der Auswahlliste **Registerformat (Zeit/Datum)** den Datentyp fest, mit dem Messwerte in Registern abgebildet werden:



### Bedeutung der Abkürzungen für Datum und Uhrzeit

- `ss`  $\triangleq$  Sekunden (0 bis 59)
- `MM`  $\triangleq$  Minuten (0 bis 59)
- `hh`  $\triangleq$  Stunden (0 bis 23, wobei: 0 = Mitternacht)
- `WD`  $\triangleq$  Wochentag (1 bis 7, wobei: 1 = Sonntag)
- `dd`  $\triangleq$  Tag (1 bis 31)
- `mm`  $\triangleq$  Monat (1 bis 12)
- `yy`  $\triangleq$  Jahr ausgehend von 20yy (2000 bis 2099)
- `xx`  $\triangleq$  (reserviert)
- `.`  $\triangleq$  (reserviert)
- `0`  $\triangleq$  Füllzeichen Null (0)

- **QWord**  
als `ssMMhhWDddmmyyxx` mit einem Register von 64 Bit Wortlänge
- **Word (mm,dd,yy,hh,MM)**  
als `mm, dd, yy, hh, MM` mit fünf Registern von je 16 Bit Wortlänge
- **Word (mm,dd,yy,hh,MM,ss)**  
als `mm, dd, yy, hh, MM, ss` mit sechs Registern von je 16 Bit Wortlänge
- **Word (yy,mm,dd,hh,MM,ss)**  
als `yy, mm, dd, hh, MM, ss` mit sechs Registern von je 16 Bit Wortlänge
- **Word (dd,mm,yy,hh,MM,ss)**  
als `dd, mm, yy, hh, MM, ss` mit sechs Registern von je 16 Bit Wortlänge
- **Float (mmddy.0,hhMMss.0)**  
als `mmddy.0, hhMMss.0` mit zwei Registern von je 32 Bit Wortlänge

- **Float (yy,mm,dd,hh,MM,ss)**

als yy, mm, dd, hh, MM, ss mit sechs Registern von je 32 Bit Wortlänge

- **Unix**

Anzahl Sekunden seit dem 1.1.1970 00:00:00 mit einem Register von 32 Bit Wortlänge – Bedeutung der Zeit (lokal oder UTC) anwendungsabhängig

- ▶ *(Nur für Registerbereiche ab AFB-Version 03-12 – Betriebsart „Master“)*  
Mit dem Parameter **Import-Registertyp** legen Sie fest, mit welchem Kommando der Master Daten aus dem Slave ausliest. Der AFB unterstützt folgende Kommandos:

- **Holding-Register (0x03)**

Der Master liest die Register des Slaves mit Funktionscode 0x03 aus dem Adressbereich 40001 bis 49999 aus.

- **Input-Register (0x04)**

Der Master liest die Register des Slaves mit Funktionscode 0x04 aus dem Adressbereich 30001 bis 39999 aus.

- ▶ *(Nur bei Messwerten in einem Registerbereich)*

In der Auswahlliste **Wenn Bereichsgrenze verletzt** legen Sie das Verhalten für den Fall fest, dass die parametrisierte **Untere Begrenzung** oder **Obere Begrenzung** von einem Messwert verletzt wird – dieser Parameter wird nur dann ausgewertet, wenn eine dieser Bereichsgrenzen parametrisiert ist.

Sie haben folgende Optionen:

- **Wert auf Grenzwert setzen (Standardeinstellung)**

Je nachdem, ob die obere oder untere Bereichsgrenze verletzt ist, wird der entsprechende Grenzwert in das Register geschrieben.

- **Wert mit Fehler zurückweisen**

*(Nur bei Import-Registern)* Der Wert, der die obere oder untere Begrenzung verletzt, wird zurückgewiesen. Stattdessen wird der Fehlercode 0x03 (unzulässiger Wert) über Modbus gesendet.



### Registerbelegung

Bei enCore-Geräten legen Sie die Registerbelegung mit den Parametern **Registerlänge** und **Registerformat <Datentyp>** fest. Je nach Datenformat des Registers werden Anwendungsdaten in einem oder mehreren aufeinanderfolgenden Registern mit der parametrisierten **Registerlänge** übertragen.

Zum Beispiel erfolgt bei einer Registerlänge von 16-Bit die Übertragung von 64-Bit-Daten in 4 aufeinanderfolgenden 16-Bit-Registern.

Das weitere Vorgehen:

für Registerbereiche

⇒ [3.3 Benutzerdefinierte Register bereitstellen \(Registerbereich\)](#) (S. 23)

für Archivbereiche

⇒ [3.4 Archive über Modbus auslesen \(Archivbereich\)](#) (S. 25)

## 3.3 Benutzerdefinierte Register bereitstellen (Registerbereich)

Mit Ausnahme von Archivdaten verwalten Sie benutzerdefinierte Register komfortabel nach Bedarf in ein oder mehreren Registerbereichen. Für jeden Registerbereich definieren Sie in einem zweiten Schritt die erforderlichen Import- und Exportregister.



### Vorbelegung der ersten Modbus-Liste bei einer Standardparametrierung

Wenn Sie mit dem Wizard eine Standardanwendung mit (mindestens) einer Modbus-Liste erstellen, dann enthält die erste Liste bereits einen Registerbereich, der mit Exportregistern für typische Werte der Mengenumwertung vorbelegt ist. Hierbei handelt es sich um Vorschläge von Elster. Passen Sie diese Register an Ihre Anforderungen an.

Wenn Sie mithilfe des Wizards zwei Listen erstellt haben, so enthält der zweite AFB Modbus keine vordefinierten Register.

## Vorgehen in enSuite

### Voraussetzungen

- Der Registerbereich ist bereits definiert.  
⇒ [3.2 Register- oder Archivbereiche definieren](#) (S. 16)
- *für lokales Gerät (Slave oder Server)*  
In der Parametrierung  ist der Ordner des AFB Modbus geöffnet:  <Gerät> – [ <Gruppe> –]  Modbus [<x>] – **Kommunikationsmodus: Slave oder Server** –  **Lokales Gerät**  
ODER  
*für entfernte Geräte (Master oder Client)*  
In der Parametrierung  ist der Ordner des AFB Modbus geöffnet:  <Gerät> – [ <Gruppe> –]  Modbus [<x>] – **Kommunikationsmodus: Master oder Client** –  **Entfernte Geräte** –  **Entferntes Gerät <x>** geöffnet.

Um eine Registerliste zu bearbeiten, ...

- ▶ ... wechseln Sie auf die Registerkarte **Registerliste**.

Auf dieser Registerkarte verwalten Sie komfortabel Register für die einzelnen Registerbereiche. Hier können Sie neue Import- und Exportregister hinzufügen, vorhandene Register verschieben, ändern oder löschen.

- ▶ Für Exportregister vom Typ Messwert oder Zähler stellt enSuite auf der Registerkarte Intervall eine komfortable Möglichkeit bereit, zyklisch Verbräuche oder Zählerstände per Modbus übertragen.

⇒ Welche Registertypen und Registerdetails der AFB unterstützt, ist detailliert in der Online-Hilfe beschrieben.

### 3.4 Archive über Modbus auslesen (Archivbereich)

Der Archivbereich ist für die Verarbeitung von Archivgruppen des AFB Benutzerarchive optimiert. Einen Archivbereich können Sie mit einem oder mehreren Archivgruppen des AFB Benutzerarchive verknüpfen.

Wahlweise können Sie einen Archivkanal adressbasiert oder gezielt registerbasiert auslesen:

- Bei adressbasiertem Zugriff bildet der AFB Modbus einen Archivkanal (oder dessen jüngsten Teil) 1:1 auf Modbus-Register ab. Dabei können Sie die maximale Archivtiefe parametrieren, mit der Archivkanäle per Modbus übertragen werden. Bei diesem Verfahren berechnet der AFB, wie viele Modbus-Register ein Archiveintrag benötigt.
- Bei registerbasiertem Zugriff ist es (anders als beim adressbasierten Modus) möglich, gezielt einen Archiveintrag des Archivkanals über einen Index auszulesen (vs. einer fest parametrierten Anzahl der jüngsten Archiveinträge). Mit aufeinanderfolgenden Anfragen (jeweils mit geändertem Index) ist es auf diese Weise möglich, schrittweise die gesamte Archivtiefe eines Archivkanals auszulesen, ohne für die gesamte Archivtiefe Modbus-Register bereitzustellen. Hierbei muss jedoch sichergestellt sein, dass der Archivkanal während des gesamten Lesevorgangs nicht aktualisiert wird. Zum Beispiel, indem Sie den Lesevorgang eines Stundenwertarchivs immer nach der vollen Stunde starten.

## Vorgehen in enSuite

Um den Archivbereich mit einer Archivgruppe des  AFB Benutzerarchive zu verknüpfen, ...

- ▶ ... wählen Sie in der Auswahlliste **Zugehörige Archivgruppe** die gewünschte Archivgruppe aus.

Der AFB unterstützt 2 unterschiedliche Arten, die ausgewählte Archivgruppe auf Modbus-Register abzubilden:

- ▶ Wählen Sie in der der Auswahlliste **Zugriffsmodus**, den Eintrag...
  - ... **Adressbasiert**  
wenn Sie eine fest parametrisierte Anzahl der jüngsten Archiveinträge über Modbus auslesen.  
⇒ [A. adressbasierten Zugriff parametrieren](#) (S. 26)
  - ... **Registerbasiert**  
wenn Sie gezielt Archiveinträge des Archivkanals über einen Index auslesen.  
⇒ [B. registerbasierten Zugriff parametrieren](#) (S. 27)

### A. adressbasierten Zugriff parametrieren

- ▶ Geben Sie die Archivtiefe im Parameter **Anzahl Archiveinträge an – Standardwert ist 100**.

 Gemäß der gewählten Archivtiefe verwendet der AFB für diesen Archivkanal einen zusammenhängenden Registerbereich mit fortlaufenden Registernummern.

Hierbei berücksichtigt er die Registerlänge und die Datenformate, die Sie für diesen Archivbereich parametrisiert haben.

- ▶ Legen Sie die Reihenfolge im Parameter **Indexverfahren** fest:
  - **Push up**  
Bei Push-up-Verfahren wird der aktuellste Archiveintrag in dem reservierten Archivbereich immer in der niedrigen Registernummer übertragen und der älteste in der höchsten (belegten).
  - **Push down**  
Beim Push-down-Verfahren wird der aktuellste Archiveintrag in dem

reservierten Archivbereich immer in der höchsten Registernummer übertragen und der älteste in der jüngsten (belegten).

#### B. registerbasierten Zugriff parametrieren

- ▶ Geben Sie im Parameter **Index-Registernummer** Nummer des Index-Registers an – *Standardwert* ist **1**.
- ▶ Wählen Sie im Parameter **Format Index-Register** das Format dieses Registers aus:
  - **Word/short** mit 16 Bit Wortlänge
  - **DWord/int** mit 32 Bit Wortlänge
- ▶ Legen Sie die Reihenfolge im Parameter **Indexverfahren** fest:
  - **Push up**  
Beim Push-up-Verfahren zeigt ein Index von Null (0) auf den ältesten Archiveintrag des Archivkanals, ein Index von 1 auf den nächst jüngeren etc.
  - **Push down**  
Beim Push-down-Verfahren ist es umgekehrt, hier zeigt ein Index von Null (0) auf den aktuellen Archiveintrag, ein Index von 1 auf den nächst älteren etc.
  - **Ordnungsnummer**  
Beim diesem Indexverfahren fragen Sie einen Archiveintrag gezielt über seine Ordnungsnummer ab. Da die Ordnungsnummer in einem Archivkanal eindeutig ist, liefert dieser Lesevorgang auch bei wiederholtem Lesen immer den gleichen Archiveintrag (solange dieser im Archivkanal vorhanden ist).
- ✓ Die allgemeinen Eigenschaften und Datentypen des Archivbereichs sind definiert.

Im zweiten Schritt definieren Sie die erforderlichen Exportregister:

Um eine Registerliste zu bearbeiten, ...

- ▶ ... wechseln Sie auf die Registerkarte **Registerliste**.
- ✓ Auf dieser Registerkarte verwalten Sie komfortabel Register für die einzelnen Registerbereiche. Hier können Sie neue Exportregister hinzufügen, vorhandene Register verschieben, ändern oder löschen.

⇒ Welche Registertypen und Registerdetails der AFB unterstützt, ist detailliert in der Online-Hilfe beschrieben.

### 3.5 Exportregister am Gerät prüfen

Mit dem AFB ist es möglich, dass Sie direkt in der Geräteanzeige einzelne Exportregister überprüfen – ausgenommen hiervon sind nur die Exportregister vom Typ **Export Datum und Zeit**.

Sobald Sie sich als Benutzer mit dem Benutzerrecht **Allgemeine Systemeinstellungen ändern**<sup>2</sup> am Gerät anmelden, können Sie die Testfunktion aktivieren bzw. deaktivieren.

#### Vorgehen in enSuite

##### Voraussetzung

- Sie sind als Benutzer mit dem Benutzerrecht **Allgemeine Systemeinstellungen ändern** am Gerät angemeldet.

Um den Test eines Exportregisters zu aktivieren, ...

- ▶ ... wechseln Sie in die Detailanzeige des gewünschten Exportregisters über **Modbus – Hauptanzeige**.
- ☒ In der Anzeige **Modbusregister** wird die Zeile **Test** als Auswahlliste angezeigt.
- ▶ Wählen Sie in der Auswahlliste **Test** den Eintrag **an**.
- ☒ Der Testmodus ist aktiv und der Wert für das **Datenobjekt** ist editierbar. Der Text **Datenobjekt** wird während des Testmodus in roter Schriftfarbe dargestellt.

---

<sup>2</sup> Die Einstellung für die Berechtigung finden Sie in der Parametrierung unter **<Gerät> – Grundsystem –  Benutzer – Allgemeine Systemeinstellungen ändern**.

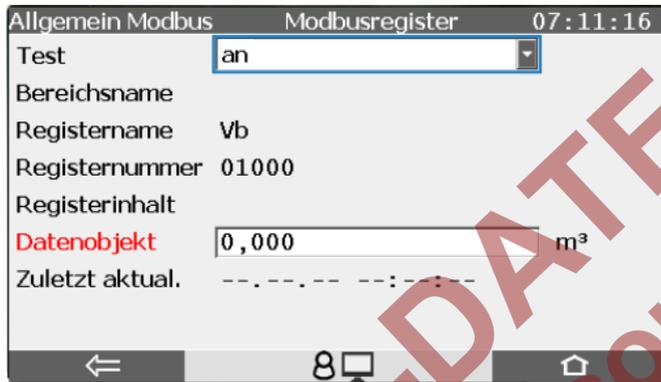


Abb. 3-1: Testfunktion für ausgewähltes Exportregister aktiv

- ▶ Geben Sie in das Feld **Datenobjekt** den gewünschten Wert ein, den Sie simulieren möchten und bestätigen Sie Ihre Eingabe mit der Schaltfläche ✓.  
Die Wertebereiche variieren je nach Registertyp und zugeordnetem Format.
- ☒ Der angegebene Wert wird über Modbus bereitgestellt, bis Sie einen neuen Wert eintragen oder die Testfunktion beenden.



#### Testfunktion beenden

Die Testfunktion endet in folgenden Fällen:

- Sobald Sie in der Anzeige auf eine andere Seite navigieren, z.B. um den Wert eines anderen Registers zu simulieren.
- Abmeldung des Benutzers z.B. aufgrund des Inaktivitätstimeouts.
- Nach einem Neustart des enCore Geräts.

## 4 Anzeige und Bedienung

In den Anzeigen des AFB Modbus werden alle Register- und Archivbereiche mit den zugehörigen Modbus-Registern angezeigt.



### Anzeige und Navigation bei enCore FC-Geräten

Der generelle Aufbau der Anzeigen bei enCore FC-Geräten und die grundlegenden Navigationsmöglichkeiten sind detailliert in der „Betriebsanleitung“ des enCore FC im Kapitel zu Anzeige und Navigation dokumentiert.

Generell werden bei der Bedienung von enCore FC-Geräten Hyperlinks und Aktionen unterschieden. Während Sie mit Hyperlinks durch die Anzeigen des Geräts navigieren, führen Sie mit Aktionen eine bestimmte Funktionalität aus. Hyperlinks und Aktionen werden im Gerät und im Handbuch blau unterstrichen dargestellt

Eine Liste der im Folgenden verwendeten Symbole und Bezeichnungen finden Sie im Anhang (⇒ Anhang [6.1 Nomenklatur](#), S. 43).

## 4.1 Anzeigen in der Übersicht

Die folgende Abbildung skizziert die hierarchische Anordnung und die Navigation durch die Anzeigen des AFB Modbus :

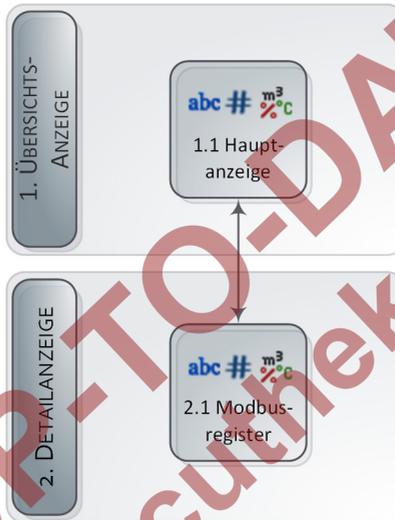


Abb. 4-1: Anzeige – hierarchische Struktur

## 4.2 Anzeigen im Detail

Die **Hauptanzeige** des AFB Modbus gibt Ihnen eine Übersicht über alle (parametrierten) Modbus-Bereiche des enCore-Geräts und listet aufsteigend alle zugehörigen Register zum ausgewählten Bereich auf. Optional können Sie zu jedem Register in eine Detailanzeige wechseln.

Modbus 3	Hauptanzeige	TC:	225
04806	900.000000	m <sup>3</sup> /h	/
04808	000000001212.000	m <sup>3</sup>	/
04810	09.02.18 14:50:34		/
04814	3.716000	mo1%	/
04816	2.500000	bar	/
04818	23.799999	°C	/
04820	75895.898438	MJ/h	/
04822	1698.856445	kg/h	/

Abb. 4-2: Modbus **Hauptanzeige** – Beispiel

Im Beispiel werden die ersten Modbus-Register des Registerbereichs **Q.Sonic plus** mit ihrer Registernummer und dem Registerinhalt (dezimal) angezeigt. Bislang wurden 84384 Telegramme übertragen.

Zu jedem Import- oder Exportregister signalisiert ein Zeichen – im Beispiel | (senkrechter Strich) –, dass die Daten dieses Registers erfolgreich übertragen wurden. Mithilfe der Zeichen – (Gedankenstrich), \ (Backslash), | (senkrechter Strich) und / (Slash) wird eine Art Drehkreuz simuliert, dass sich jedes Mal um einen Schritt ändert, wenn das individuelle Register erfolgreich übermittelt wird.

## Anzeige Übersicht im Detail

[1.1 Hauptanzeige]			
<b>&lt;Auswahlliste&gt;</b> enthält alle parametrisierten Modbus- bereiche – für Slave oder Server ist nur 1 Einheit vorhanden		<b>TC: &lt;Zähler&gt;</b> (Abk. für Telegram Counter) Telegrammzähler des ausgewählten Bereichs	
<b>&lt;Reg.-Nr.&gt;</b> Registernummer des Import- oder Exportregisters  [⇒ 2.1 <b>Modbusregister]</b>		<b>&lt;Datenobjekt&gt;</b> übermittelter Wert formatiert	→ \ → - ⇒ / Änderung des Zeichens signalisiert, dass das Register erfolgreich übermittelt wurde

[2.1 Modbusregister]	
<b>Test</b> <b>an/aus</b>	Testfunktion des ausgewählten Exportregisters aktiviert/deaktiviert ⇒ 3.5 Exportregister am Gerät prüfen (S. 28)
<b>Bereichsname</b>	Name des Registerbereichs (Parameter <b>Name</b> )
<b>Registername</b>	Name des Registers (Parameter <b>Name</b> )
<b>Registernummer</b>	Registernummer des Import- oder Exportregisters
<b>Registerinhalt</b>	übertragener Wert (hexadezimal)
<b>Datenobjekt</b>	übertragener Wert formatiert mit Einheit  Sobald die Testfunktion aktiviert ist, können Sie einen Wert manuell eingeben, um ihn für Testzwecke zu simulieren.
<b>Zuletzt aktual.</b>	<Datum> <Uhrzeit>

## 5 FAQs

Dieses Kapitel enthält die wichtigsten Einstellungen und Fragestellungen, um Sie bei regelmäßig anfallenden Aufgaben zu unterstützen:

- ⇒ [5.1 Wie unterscheiden sich im AFB Modbus der Normal- und der Expertenmodus? \(S. 34\)](#)
- ⇒ [5.2 Wie parametrierte ich im Normalmodus in wenigen Schritten einen FC1 als Modbus-Slave an COM2? \(S. 35\)](#)
- ⇒ [5.3 Wie synchronisiere ich die Systemzeit über Modbus? \(S. 38\)](#)
- ⇒ [5.4 Wie kann ich einen Protokollfehler \(Meldung\) beheben? \(S. 40\)](#)

### 5.1 Wie unterscheiden sich im AFB Modbus der Normal- und der Expertenmodus?

#### Hintergrund

Der Normalmodus dient zum komfortablen Parametrierung von Standardanwendungen. Darum sind im Normalmodus nicht alle Parameter und Exportwerte sichtbar und auch nicht alle Bearbeitungsmöglichkeiten verfügbar. Sie können bei Bedarf aber jederzeit über den Menüpunkt **Extras – Expertenmodus** in den Expertenmodus umschalten, der *alle* verfügbaren Parameter und Funktionen anbietet. Beachten Sie, dass Sie bei Änderungen ggf. nicht mehr in den Normalmodus zurückschalten können.

#### Unterschiede

Der wesentliche Unterschied bei Modbus zwischen Normal- und Expertenmodus ist, dass Sie im Normalmodus ausschließlich enCore-Gerät parametrieren können, die die Rolle des Slaves bzw. Servers übernehmen. Im Expertenmodus hingegen legen Sie pro Instanz eines AFB Modbus fest,

ob das enCore-Gerät bei der Modbus-Kommunikation entweder der Master bzw. Client ist, oder als Slave bzw. Server kommuniziert.

Der Parameter **Kommunikationsmodus** steht nur im Expertenmodus zur Verfügung.

Ein weiterer Unterschied ist die übersichtliche Parametrierung von Modbus-Listen im Normalmodus anhand der drei Registerkarten **Einstellungen**, **Liste der Bereiche** und **Registerliste**. Die **Registerliste** steht Ihnen auch im Expertenmodus zur Verfügung.

Im Expertenmodus parametrieren Sie wie gewohnt anhand von Verzweigungen des Parameterbaums. Da die Parametrierung einer Modbus-Liste im Expertenmodus sehr verzweigt und komplex werden kann, empfehlen wir, nur zum Parametrieren eines Masters bzw. Clients in den Expertenmodus zu wechseln.

## 5.2 Wie parametriere ich im Normalmodus in wenigen Schritten einen FC1 als Modbus-Slave an COM2?

### Hintergrund

Im Normalmodus können Sie ein enCore-Gerät in wenigen Schritten mithilfe des AFB `Modbus` als Modbus-Slave parametrieren. Wenn Sie eine neue Parametrierung für eine Standardanwendung erstellen, legen Sie bereits im Wizard fest, ob im Normalmodus keine, eine oder zwei Modbus-Listen zur Verfügung stehen. Pro Liste legt der Wizard in der Parametrierung eine Instanz des AFB `Modbus` (**Modbus 1/Modbus 2**) an.

Wenn Sie Modbus verwenden, ist in der Modbus-Liste **Modbus 1** bereits ein Registerbereich mit Exportregistern für typische Werte für eine Mengenumwertung vordefiniert. Bei zwei Modbus-Listen enthält die zweite Liste **Modbus 2** noch keine vordefinierten Register.

Das folgende Szenario veranschaulicht beispielhaft die Parametrierung im Normalmodus:

enCore-Gerät:	FC1 mit Standardparametrierung
Protokolltyp:	Modbus RTU
Schnittstelle:	COM 2 der CPU
Übertragungsrate:	9600 Baud
Übertragungseinstellungen:	8 Datenbits, gerade Parität, 1 Stoppbit
Slave-ID:	2
Beginn des Registerbereichs:	3000
Exportregister:	Normvolumen $V_{n,}$ Normvolumendurchfluss $Q_{n,}$ Meldung Alarm Umwertung
Registerlänge	16-bit

### Vorgehen in wenigen Schritten

Im Folgenden sind nur die Modbus-relevanten Einstellungen beschrieben.

Im ersten Schritt legen Sie die Kommunikationseinstellungen fest:

- ▶ Erstellen Sie mit dem Wizard für den FC1 eine **Standard-Mengen-umwertung für Gas** mit 1 Modbus-Liste.
- ▶ Markieren Sie unter **Start (E/A-Konfiguration)** auf der Registerkarte **E/A-Konfiguration** den COM-Port **CH2** der **CPU3**.
- ▶ Wählen Sie im unteren Bereich in der Auswahlliste **CH2** den Eintrag **COM-Port (einfach)**.
- ✓ Die **Baudrate** ist bereits korrekt auf **9600** Baud voreingestellt.
- ▶ Setzen Sie die **Parität** auf den Wert **gerade**.
- ▶ Öffnen Sie im Parameterfenster den Ordner  **Modbus 1** und wechseln Sie auf die Registerkarte **Einstellungen**.
- ▶ Wählen Sie als **Schnittstelle** den Eintrag **seriell**.
- ▶ Wählen Sie als **Port** den Eintrag **CH2.COM-Port**.
- ✓ **Modbus RTU** ist bereits korrekt als **Protokolltyp** voreingestellt.
- ▶ Tragen Sie als **Slave-ID** den Wert **2** ein.

Definieren Sie im zweiten Schritt die erforderlichen Exportregister:

### Vorgehen 1: vorbelegte Register bearbeiten

Standardmäßig ist auf der Registerkarte **Liste der Bereiche** die Liste bereits als **Registerbereich** mit einer Registerlänge **16-Bit** vorbelegt.

- ▶ Wechseln Sie auf die Registerkarte **Registerliste**.
- ▶ Verschieben Sie die folgenden Exportregister mit dem  Pfeil-nach-oben-Symbol wie folgt:
  - **V<sub>n</sub>** an Position 1
  - **Q<sub>n</sub>** an Position 2
  - **Alarm Volumenmessung** an Position 3

Um alle anderen vorbelegten Register zu löschen, ...

- ▶ ... markieren Sie das Register an Position 4, halten Sie die [Shift]-Taste gedrückt und markieren Sie das letzte Register in der Liste.
- ▶ Mit einem Klick auf das  Entfernen-Symbol werden die markierten Register ohne Rückfrage gelöscht.

Um aufeinanderfolgende Registernummer zu vergeben, ...

- ▶ ... markieren Sie die 3 Exportregister **V<sub>n</sub>**, **Q<sub>n</sub>** und **Alarm Volumenmessung**.
- ▶ Klicken Sie auf das  Nummerieren-Symbol und geben Sie als Startnummer **3000** ein.
- ✓ Das Szenario ist parametrisiert.

### Vorgehen 2: neue Register definieren

Um alle von Elster vorbelegten Register zu löschen, verwenden wir einen kleinen Trick:

- ▶ Wechseln Sie auf die Registerkarte **Liste der Registerbereiche**.
- ▶ Wählen Sie für den vordefinierten **Bereich 1** den Wert **Archivbereich** aus und anschließend (wieder) den Wert **Registerbereich**.
- ✓ Die **Registerlänge** beträgt standardmäßig **16-Bit**.

Um die neuen Register zu definieren, ...

- ▶ ... wechseln Sie auf die Registerkarte **Registerliste**.
- ▶ Öffnen Sie das Exportwertefenster im Ordner  **Schiene1**.
- ▶ Ziehen Sie die folgenden Exportwerte per Drag-and-drop in die Registerliste:
  -  **Berechnungen.V<sub>n</sub>** an Position 1
  -  **Berechnungen.Q<sub>n</sub>** an Position 2
  -  **Alarm Volumenmessung** an Position 3

Um aufeinanderfolgenden Registernummer zu vergeben, ...

- ▶ ... markieren Sie die 3 Exportregister **V<sub>n</sub>**, **Q<sub>n</sub>** und **Alarm Volumenmessung**.
- ▶ Klicken Sie auf das  Nummerieren-Symbol und geben Sie als Startnummer den Wert **3000** ein.
- ✓ Das Szenario ist parametrierbar.

### 5.3 Wie synchronisiere ich die Systemzeit über Modbus? (Normalmodus)

#### Hintergrund

Der `Zeitservice`<sup>3</sup> ist für das Synchronisieren oder Stellen der geräte-internen Systemzeit zuständig. Voraussetzung ist, dass im `Zeitservice` die gewünschte(n) Zeitquelle(n) als Importwerte in den Parametern **Prim. externe Zeitquelle** und (optional) **Sek. externe Zeitquelle** hinterlegt sind.

Um die Systemzeit eines Geräts abzugleichen, können Sie als verlässliche Zeitquellen NTP, Modbus oder DSfG-Zeit verwenden.

Damit ein enCore-Gerät seine Systemzeit mit der Modbus-Zeit abgleichen kann, parametrieren Sie im ersten Schritt das zugehörige Importregister im `AFB Modbus` und importieren die Modbus-Zeit im zweiten Schritt im `Zeitservice` als externe Zeitquelle.

<sup>3</sup> Die Funktionsweise des `Zeitservice` ist detailliert im FC-Handbuch „Grundsystem mit SFBs“ beschrieben.



Wir empfehlen den Zeitabgleich der Systemzeit über Modbus nur für den Fall, dass keine externe NTP-Quelle zur Verfügung steht.

## Vorgehen in enSuite in Kürze

Die Parametrierung zeigen wir beispielhaft anhand folgenden Szenarios: Als primäre Zeitquelle dient ein Zeitserver, der seine aktuelle Zeit über Modbus bereitstellt.

### Voraussetzung

- Kommunikationseinstellungen und Bereich der Modbus-Liste sind bereits parametriert.

Um im ersten Schritt das Importregister im Modbus AFB zu definieren, ...

- ▶ Öffnen Sie die Parametrierung im Normalmodus.
- ▶ Wechseln im Modbus auf die Registerkarte **Registerliste**.
- ▶ Markieren Sie die Zeile, vor die das Importregister eingefügt werden soll.  
Sie können die Position auch nachträglich mit nach oben und mit unten verschieben.
- ▶ Legen Sie mit ein neues Importregister an.
- ▶ In der Detailanzeige wählen Sie in der Auswahlliste **Register <x>** den Eintrag **Import Datum und Uhrzeit** aus.
- ▶ Geben Sie mit dem Parameter **Zeitmodus** an, ob Datum und Uhrzeit des Importregisters in **UTC (Universal Time Coordinated)** oder in **Ortszeit** vorliegen.
- Wenn die Daten in **Ortszeit** übertragen werden, wandelt das enCore-Gerät Datum und Uhrzeit zunächst in UTC um, bevor es Datum und Uhrzeit im Exportwert [**<Reg.-Nr.>**]: **Import Datum und Zeit.Time** bereitstellt.

Standardmäßig aktualisiert das Gerät den Exportwert [**<Reg.-Nr.>**]: **<Name>: Import Datum und Zeit.Time**, sobald sich mindestens ein Wert im Importregister Datum und Zeit ändert.

Um alternativ die Aktualisierung dieses Exportwerts gezielt über ein anderes Register zu steuern, ...

- ▶ ... aktivieren Sie das Kontrollkästchen Trigger-Register und geben Sie die gültige Nummer des Importregisters an.
- ☒ Nur wenn der Wert des angegebenen Trigger-Registers sich von Null (0) auf einen anderen Wert ändert, wird der Exportwert  [**<Reg.-Nr.>**] **<Name>: Import Datum und Zeit.Time** aktualisiert.

Um im zweiten Schritt die Modbus-Zeit im `Zeitservice` zuzuweisen, ...

- ▶ ... wechseln Sie im Ordner **Allg. Einstellungen** auf die Registerkarte **Zeitservice**.
- ▶ Im Bereich **Externe Zeitquelle** wählen Sie aus der Auswahlliste **Prim. externe Zeitquelle** das parametrisierte Modbus-Register aus.
- ☒ Der `Zeitservice` synchronisiert die Systemzeit mit der Modbus-Zeit, wenn eine Übernahme nach den `Zeitservice` hinterlegten Regeln zulässig ist.

## 5.4 Wie kann ich einen **Protokollfehler** (Meldung) beheben?

### Hintergrund

Master (seriell) oder Client (TCP/IP) überwachen die Kommunikation über Modbus. Sie stellen dann einen Übertragungsfehler fest, wenn entweder die Modbus-Kommunikation unterbrochen ist, oder der Datenimport bzw. Datenexport von mindestens einem Register nicht gelingt. In diesem Fall generiert der AFB die gleichnamige Meldung  **Protokollfehler**. Die Meldung steht so lange an, bis alle Register dieser Modbus-Einheit wieder erfolgreich importiert bzw. exportiert werden können.

Wichtige Parameter für die Fehlererkennung sind der **Transaktionstimeout** (Master) und **Fehlerfilter** (Slave/Server). Beim **Transaktionstimeout** legen Sie die maximale Zeitspanne fest, die ein Master auf die Antwort eines Slaves wartet, bevor er einen Protokollfehler generiert. Beim Fehlerfilter geben Sie pro Slave bzw. Server an, wie oft der AFB versucht, ein Register

zu im- bzw. exportieren (Round Robin-Verfahren), bevor Master bzw. Client die Meldung generiert.

### Mögliche Ursachen und Abhilfen

- Das Gerät ist eingeschaltet, ist jedoch aufgrund fehlerhafter Parametrierung der Kommunikationsparameter nicht erreichbar.
- ▶ Stellen Sie sicher, dass die Kommunikationsparameter im **Grundsystem** und AFB Modbus zwischen Master und Slave bzw. Client und Server aufeinander abgestimmt sind.

ODER

[nur Slave/Server]

- Die Fehlertoleranz beim Datenimport oder Datenexport eines Registers ist zu niedrig.
- ▶ Erhöhen Sie testweise den Parameter **Fehlerfilter** unter:  
**Modbus – Kommunikationsmodus: Master oder Client – Entfernte Geräte – <Entferntes Gerät x>, Registerkarte Parameter, Bereich Allgemein.**

ODER

[Master und Slave]

- Mindestens ein Register ist auf einer der beiden Seiten falsch oder gar nicht parametrierung.
- ▶ Schauen Sie am Bedienfeld des enCore-Geräts in die zum AFB Modbus zugehörige Anzeige und beobachten Sie, welche Register nicht regelmäßig aufgefrischt werden. Suchen Sie auf beiden Seiten bezüglich dieser Registernummern nach Fehlern in der jeweiligen Parametrierung.

ODER

[Master und Slave]

- Die Verkabelung ist defekt oder ein Kabelbruch liegt vor.
- ▶ Prüfen Sie die Verkabelung und tauschen Sie beschädigte Kabel aus.

## ODER

[nur Master]

-  Das entfernte Gerät (Slave bzw. Server) verarbeitet Telegramme langsamer als erwartet.
- ▶ Erhöhen Sie testweise den **Transaktionstimeout** des Masters unter:  
 **Modbus** –  **Kommunikationsmodus: Master oder Client**,  
Registerkarte **Parameter**, Bereich **Schnittstelle**

## 6 Anhang

### 6.1 Nomenklatur

Folgende Symbole und Bezeichnungen werden im enCore FC-Gerät und in enSuite für Messdaten und berechnete Werte im Kontext des AFB

Modbus verwendet:

Symbol	Kurzform	Beschreibung
abc	<Text>	parametrierter Register- oder Bereichsname
#	<Ziffer>	Platzhalter für eine Ziffer, z.B. für eine Registernummer oder einen Messwert
	<Einheit>	beliebige Einheit z.B. eines Messwerts

Tabelle 6-1: Nomenklatur

### 6.2 Unterstützte Modbus-Telegrammtypen

Das Modbus-Protokoll kennt insgesamt 18 verschiedene Telegrammtypen, von denen enCore-Geräte die Folgenden unterstützen – alle anderen Telegrammtypen werden entweder abgewiesen oder nicht ausgewertet.

Typ	Beschreibung
0x03	(„Read Holding Registers“) Lesen eines oder mehrerer zusammenhängender Register eines entfernten Geräts
0x10	(„Write Multiple Registers“) Ändern eines oder mehrerer zusammenhängender Register in einem entfernten Gerät
0x08	(„Diagnostics“) – nur seriell liefert Diagnosewerte zur Kommunikation zwischen einem Master und einem Slave
zusätzlich für Slave oder Server:	
0x04	(„Read Input Registers“) Lesen eines oder mehrerer zusammenhängender Register eines entfernten Geräts
0x06	(„Write Single Register“) Ändern eines Registers in einem entfernten Gerät

Tabelle 6-2: Unterstützte Telegrammtypen

## 6.3 Datenbeschreibung der Registerformate

Die Modbus-Spezifikation definiert den allgemeinen Datenframe für die Datenübertragung, jedoch nicht, wie die Daten in den Modbus-Registern abgebildet werden. Deshalb kommen in der Praxis viele verschiedene Datenformate zum Einsatz. Der AFB `Modbus` beschreibt verschiedene Formate mithilfe der Registerlänge, Byte-Reihenfolge und dem Datentyp.

### 6.3.1 Byte-, Word- und DWord-Reihenfolge

Die Byte-Reihenfolge gibt die Position des höher- und niedrigwertigen Bytes an. Die Byte-Reihenfolge legen Sie im AFB für jeden Registerbereich für die Dateneinheiten Byte (8-Bit), Word (16-Bit) sowie DWord (32-Bit) an.

Byte-Reihenfolge	Beschreibung
LO/HI	Low-Byte/High-Byte („Little-Endian“) ( <i>Standardeinstellung</i> ) Das höherwertige Byte liegt auf der höheren Adresse.
HI/LO	High-Byte/Low-Byte („Big-Endian“) Das höherwertige Byte liegt auf der niedrigeren Adresse.

### 6.3.2 Unterstützte Datentypen

#### für Messwerte, Zähler/Eventzähler, Bitleisten/Status

Name	Datentyp	Größe
Word (short)	Dual- und Hexadezimalzahlen	16-Bit
DWord/int	Dual- und Hexadezimalzahlen	32-Bit
QWord (64-Bit)	Dual- und Hexadezimalzahlen	64-Bit
Float S5	Gleitkommazahl (Siemens Datentyp)	32-Bit
Float	normalisierte Gleitkommazahl single precision nach IEEE 754	32-Bit
Double	normalisierte Gleitkommazahl double precision nach IEEE 754	64-Bit
BCD8	binär codierte Dezimalzahl	8-Bit
BCD16	binär codierte Dezimalzahl	16-Bit

**für Datum und Uhrzeit:****Bedeutung der Abkürzungen für Datum und Uhrzeit**

- ss  $\triangleq$  Sekunden (0 bis 59)
- MM  $\triangleq$  Minuten (0 bis 59)
- hh  $\triangleq$  Stunden (0 bis 23, wobei: 0 = Mitternacht)
- WD  $\triangleq$  Wochentag (1 bis 7, wobei: 1 = Sonntag)
- dd  $\triangleq$  Tag (1 bis 31)
- mm  $\triangleq$  Monat (1 bis 12)
- yy  $\triangleq$  Jahr ausgehend von 20yy (2000 bis 2099)
- xx  $\triangleq$  Validierungscode (nur aus Kompatibilitätsgründen mit dem FC2000) wird bei Importregistern ignoriert, und hat bei Exportregistern immer den Wert 1)
- .  $\triangleq$  (reserviert)
- 0  $\triangleq$  Füllzeichen Null (0)

Name	Datentyp	Wortlänge
<b>QWord</b>	ssMMhhWDddmmyyxx	1 Register von 64-Bit
<b>Word (mm,dd,yy,hh,MM)</b>	mm, dd, yy, hh, MM	5 Register von je 16-Bit
<b>Word (mm,dd,yy,hh,MM,ss)</b>	mm, dd, yy, hh, MM, ss	6 Register von je 16-Bit
<b>Word (yy,mm,dd,hh,MM,ss)</b>	yy, mm, dd, hh, MM, ss	6 Register von je 16-Bit
<b>Word (dd,mm,yy,hh,MM,ss)</b>	dd, mm, yy, hh, MM, ss	6 Register von je 16-Bit
<b>Float (mmdyy.0,hhMMss.0)</b>	mmdyy.0, hhMMss.0	2 Register von je 32-Bit
<b>Float (yy,mm,dd,hh,MM,ss)</b>	yy, mm, dd, hh, MM, ss	6 Register von je 32-Bit
<b>Unix</b>	Anzahl Sekunden seit dem 1.1.1970 00:00:00 – ohne Angabe der UTC	1 Register von 32-Bit

## 7 Literaturverzeichnis

- MODBUS Application Protocol Specification V1.1b3 (Apr. 2012)
- MODBUS Over Serial Line Specification and Implementation Guide V1.02 (Dez. 2006)
- MODBUS Messaging on TCP/IP Implementation Guide V1.0b (Okt. 2006)

Herausgeber: Modbus Organization

Verfügbar unter: [modbus.org/specs.php](http://modbus.org/specs.php)

## 8 Index

### A

Anhang 43  
Anzeige und Bedienung 30  
Anzeigenübersicht 31

### D

Datenbeschreibung 44  
Datenexport 14  
Datenimport 14  
Datentyp  
  Bitleisten/Status 45  
  Datum und Uhrzeit 46  
  Messwert 45  
  Zähler/Eventzähler 45

### F

FAQs 34

### N

Navigation (enCore-Gerät) 30  
Nomenklatur 43

### O

Online-Hilfe aufrufen 5

### P

Protokollfehler beheben 40  
Protokollfehler Ursachen 41

### R

Registerformate 44

### S

Slave  
  Beispielparametrierung 35  
Systemzeit  
  über Modbus aktualisieren 38

### T

Telegrammtyp  
  0x03 44  
  0x04 44  
  0x06 44  
  0x08 44  
  0x10 44  
Telegrammtypen 43

### Z

Zeitabgleich 38