



# Operating Instructions

---

**Fronius GEN24**  
**Modbus TCP & RTU**

**DE** | Bedienungsanleitung

**EN-US** | Operating instructions



42,0410,2649

027-02052022



# Inhaltsverzeichnis

Das Modbus Protokoll .....	5
Allgemeines .....	5
Aufbau von Modbus Nachrichten .....	5
Modbus TCP – MBAP Header .....	6
Unterstützte Funktionscodes .....	6
03 (0x03) Read Holding Registers .....	7
06 (0x06) Write Single Register .....	7
16 (0x10) Write Multiple Registers .....	8
Exception Codes .....	9
CRC Berechnung für Modbus RTU .....	9
CRC Prüfsumme berechnen .....	10
CRC Prüfsumme zur Nachricht hinzufügen .....	11
Allgemeines .....	12
Verwendete Abkürzungen .....	12
Register Maps .....	12
Antwortzeiten .....	13
Modbus Geräte-ID für Wechselrichter .....	13
Modbus Geräte-ID für Energiezähler .....	13
Registeradressen .....	13
Nicht vorhandene Datensätze .....	16
Zeitverhalten der unterstützten Betriebsarten .....	17
Vorzeichenkonvention für den Power Factor .....	18
Auf der Karte gespeicherte Werte .....	18
Skalierungsfaktoren .....	19
Nicht beschreibbare Register .....	19
Schreiben ungültiger Werte .....	19
Einstellungen - Modbus .....	20
Allgemeines .....	20
Einstellungen - Modbus öffnen .....	20
Modbus .....	20
Steuerung einschränken .....	21
Common & Inverter Model .....	22
Common Block Register .....	22
Inverter Model Register .....	22
SunSpec Operating Codes .....	22
Nameplate Model (120) .....	23
Allgemeines .....	23
Nameplate Register .....	23
Basic Settings Model (121) .....	24
Basic Settings Register .....	24
Referenzspannung .....	24
Abweichung zur Referenzspannung .....	24
Extended Measurements & Status Model (122) .....	25
Allgemeines .....	25
Extended Measurements & Status Register .....	25
Immediate Controls Model (123) .....	26
Allgemeines .....	26
Immediate Controls Register .....	26
Standby .....	26
Leistungsreduktion .....	26
Beispiel: Leistungsreduktion einstellen .....	27
Beispiel: Ändern der Rückkehrzeit bei aktiver Leistungsreduktion .....	27
Auswirkungen der Blindleistungs-Vorgaben auf die Wirkleistung .....	28
Konstanter Power Factor .....	29
Beispiel: Konstanten Power Factor vorgeben .....	29
Konstante relative Blindleistung .....	30
Beispiel: Konstante Blindleistung vorgeben .....	30
Multiple MPPT Inverter Extension Model (160) .....	31
Allgemeines .....	31
Multiple MPPT Inverter Extension Register .....	31

Basic Storage Control Model (124).....	32
Allgemeines .....	32
Bereitgestellte Informationen.....	32
Leistungsfenster-Vorgaben .....	32
Vorgabe des minimalen Ladestandes .....	34
Laden des Energiespeichers vom Netz.....	34
Basic Storage Controls Register .....	34
Register- und Batteriestatusänderungen in Fronius Solar.web.....	34
Meter Model.....	36
Meter Model Register .....	36
End Block.....	37
Allgemeines .....	37
End Block.....	37

## Allgemeines

Die Beschreibung des Protokolls entstammt zum größten Teil den Modbus Spezifikationen, die öffentlich auf [www.modbus.org/specs.php](http://www.modbus.org/specs.php) erhältlich sind.

Modbus ist ein einfaches, offenes Kommunikationsprotokoll, mit dem eine Master-Slave- oder Client-Server-Kommunikation zwischen den am Netzwerk angeschlossenen Geräten realisiert werden kann. Das Grundprinzip von Modbus ist: Ein Master sendet eine Anfrage und ein Slave antwortet darauf. Bei Modbus TCP wird der Master als Client, ein Slave als Server bezeichnet. Die Funktion ist dieselbe. In weiterer Folge werden für die Beschreibungen der Funktionen des Protokolls unabhängig von den Varianten RTU und TCP nur die gebräuchlicheren Namen Master und Slave verwendet. In Fällen, wo Unterschiede bei zwischen RTU und TCP auftreten, wird speziell darauf hingewiesen.

Am Wechselrichter kann Modbus auf 2 Arten benutzt werden:

- Modbus TCP  
Mittels TCP/IP über Ethernet (kabelgebunden oder über WLAN)
- Modbus RTU  
Mittels asynchroner serieller Übertragung über RS-485 (EIA/TIA-485-A)

Im Fall von Modbus RTU kann es immer nur einen Master im System geben. Grundsätzlich gilt, dass nur ein Master Anforderungen (Requests) initiieren darf. Ein Slave darf nur antworten (Response), wenn dieser vom Master angesprochen wurde; untereinander dürfen die Slaves nicht kommunizieren. Wird ein Broadcast Request (Anforderung an alle vorhandenen Slaves per Slave ID oder Unit ID 0) ausgesendet, darf keiner der Slaves antworten. Daher können Broadcasts nur für Schreibbefehle verwendet.

Wenn ein Master eine Anforderung an einen Slave sendet, dann erwartet dieser eine Antwort. Bei einer Anforderung eines Masters gibt es eine von fünf Möglichkeiten:

- Erhält der Slave die Anforderung ohne Kommunikationsfehler und kann diese die Anforderung fehlerfrei bearbeiten, dann wird eine normale Antwort mit den gewünschten Daten zurückgesendet.
- Erhält der Slave die Anforderung wegen eines Kommunikationsfehlers nicht, dann wird keine Antwort gesendet. Das führt zu einem Timeout am Master.
- Erhält der Slave die Anforderung, entdeckt aber einen Kommunikationsfehler (Parity, CRC, ...), wird keine Antwort gesendet. Das führt zu einem Timeout am Master.
- Erhält der Slave die Anforderung ohne Kommunikationsfehler, kann aber diese nicht fehlerfrei bearbeiten (z. B. wenn ein nicht vorhandenes Register ausgelesen werden soll), wird eine Fehlermeldung (Exception Response) mit dem Grund für den Fehler zurückgesendet.
- Erhält der Slave eine Broadcast Anforderung, die auch an alle anderen Geräte geht, so wird weder im Fehlerfall noch wenn die Anforderung erfolgreich bearbeitet wurde, eine Antwort gesendet. Daher sind Broadcast Anforderungen nur für Schreibbefehle geeignet.

Modbus Geräte stellen Daten in 16 Bit großen Datenblöcken (Register) zur Verfügung.

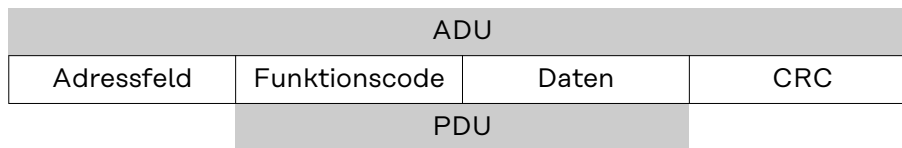
In bestimmten Fällen können einzelne Datenpunkte auch mehrere Datenblöcke umfassen (z. B. 2 Register = 32 Bit Wert).

## Aufbau von Modbus Nachrichten

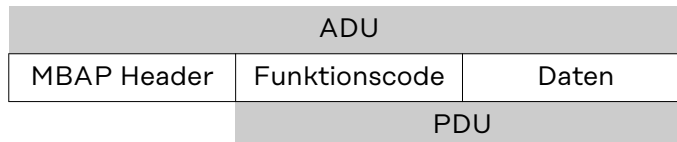
Eine Modbus Nachricht besteht grundsätzlich aus der Protokolldateneinheit (protocol data unit, PDU). Diese ist von darunter liegenden Kommunikations-

schichten unabhängig.

Abhängig von dem verwendeten Bus oder Netzwerk können noch weitere Felder hinzukommen. Diese Struktur wird dann Anwendungseinheit (application data unit, ADU) genannt.



Struktur einer Modbus Nachricht bei Modbus RTU



Struktur einer Modbus Nachricht bei Modbus TCP

Für Modbus TCP wird ein eigener Header verwendet, um die Anwendungseinheit zu identifizieren. Dieser Header heißt MBAP Header (MODBUS Application Protocol Header).

Die Größe der Protokolldateneinheit (PDU) ist limitiert, bedingt durch die ersten Modbus Implementierungen in einem seriellen Netzwerk (max. RS485 ADU = 256 Bytes). Dadurch ergibt sich für die Größe der Protokolldateneinheit PDU:  $PDU = 256 - \text{Slave ID (1 Byte)} - \text{CRC (2 Bytes)} = 253 \text{ Bytes}$

Damit ergibt sich:

- Modbus RTU ADU = 253 + Slave ID (1 Byte) + CRC (2 Bytes) = 256 Bytes
- Modbus TCP ADU = 253 Bytes + MBAP (7 Bytes) = 260 Bytes

---

### Modbus TCP – MBAP Header

Der MBAP Header umfasst 7 Bytes:

- **Transaction ID** (2 Bytes): Wird benutzt, um Anfrage und Antwort zu synchronisieren. Der Slave übernimmt die Transaction ID von der Anfrage in die Antwort.
- **Protocol ID** (2 Bytes): Ist immer 0 (Modbus Protokoll).
- **Länge** (2 Bytes): Das Längengeld enthält die Anzahl der Bytes der nachkommenden Felder, einschließlich Unit ID und Datenfelder.
- **Unit ID** (1 Byte): Dieses Feld wird zur Adressierung der am Wechselrichter angeschlossenen Geräte verwendet (Gateway-Funktion). Die Unit ID entspricht der Slave ID bei Modbus RTU. Der Wert wird vom Master vorgegeben und wird vom Slave unverändert mit der Antwort zurückgegeben.

Für Details über die Adressierung der Geräte siehe:

- [Modbus Geräte-ID für Wechselrichter](#) auf Seite **13**
- [Modbus Geräte-ID für Energiezähler](#) auf Seite **13**

**WICHTIG!** Die richtige Unit ID muss immer angegeben werden, auch wenn der Fronius Datamanager nur mit einem einzelnen Wechselrichter verbunden ist.

---

### Unterstützte Funktionscodes

Der Funktionscode bestimmt die am Slave auszuführende Aktion. Drei Funktionscodes für Lese- und Schreiboperationen werden unterstützt:

- 03 (0x03)<sup>1)</sup> Read Holding Registers
- 06 (0x06)<sup>1)</sup> Write Single Register
- 16 (0x10)<sup>1)</sup> Write Multiple Registers

Tritt am Slave bei der Bearbeitung einer Anforderung ein Fehler auf, so wird eine Fehlernachricht als Antwort (Exception Response) gesendet. Bei einer solchen Antwort wird beim Funktionscode das höchstwertige Bit auf 1 gesetzt (entspricht einer Addition des Funktionscodes mit 0x80)<sup>1)</sup> und ein Exception Code hinzugefügt, der den Grund des Fehlers angibt.

<sup>1)</sup> Das Prefix "0x" steht für hexadezimale Zahlen

### 03 (0x03) Read Holding Registers

Dieser Funktionscode wird dazu verwendet, den Inhalt eines oder mehrerer aufeinanderfolgenden Register eines Gerätes auszulesen. Die Anforderung enthält die Adresse des ersten auszulesenden Registers und die Anzahl der zu lesenden Register. In der Anforderung werden Register beginnend bei 0 adressiert. Das bedeutet, dass die Register 1 bis 16 über die Adressen 0 bis 15 angesprochen werden.

#### Anforderung

Funktionscode	1 Byte	0x03
Startadresse	2 Bytes	0x0000 bis 0xFFFF (0 bis 65535)
Anzahl der Register	2 Bytes	1 bis 125

#### Antwort

Funktionscode	1 Byte	0x03
Anzahl der Bytes	1 Byte	2 x N*
Registerwerte	N* x 2 Bytes	

\*N = Anzahl der Register

#### Fehler

Fehlercode	1 Byte	0x83
Exception Code	1 Byte	01 oder 02 oder 03 oder 04 oder 11

### 06 (0x06) Write Single Register

Dieser Funktionscode wird dazu verwendet, ein einzelnes Register zu beschreiben. Die Anforderung enthält nur die Adresse des zu beschreibenden Registers. Register werden beginnend bei 0 adressiert. Das bedeutet, dass das Register 1 über die Adresse 0 angesprochen. Die normale Antwort ist eine Kopie der Anforderung, die nach dem erfolgreichen Beschreiben des Registers gesendet wird.

#### Anforderung

Funktionscode	1 Byte	0x06
Registeradresse	2 Bytes	0x0000 bis 0xFFFF (0 bis 65535)
Registerwert	2 Bytes	

Antwort

Funktionscode	1 Byte	0x06
Registeradresse	2 Bytes	0x0000 bis 0xFFFF (0 bis 65535)
Registerwert	2 Bytes	

Fehler

Fehlercode	1 Byte	0x86
Exception Code	1 Byte	01 oder 02 oder 03 oder 04 oder 11

---

**16 (0x10) Write Multiple Registers**

Dieser Funktionscode wird dazu verwendet, einen Block von aufeinanderfolgenden Registern zu beschreiben. Die Anforderung enthält die Adresse des ersten zu beschreibenden Registers, die Anzahl der zu beschreibenden Register, die Anzahl der zu schreibenden Bytes und die zu schreibenden Werte (2 Bytes pro Register). Die normale Antwort enthält den Funktionscode, die Startadresse und die Anzahl der beschriebenen Register.

Anforderung

Funktionscode	1 Byte	0x10
Startadresse	2 Bytes	0x0000 bis 0xFFFF (0 bis 65535)
Anzahl der Register	2 Bytes	1 bis 123
Anzahl der Bytes	1 Byte	2 x N*
Registerwerte	N* x 2 Bytes	

\*N = Anzahl der Register

Antwort

Funktionscode	1 Byte	0x10
Startadresse	2 Bytes	0x0000 bis 0xFFFF (0 bis 65535)
Anzahl der Register	2 Bytes	1 bis 123

Fehler

Fehlercode	1 Byte	0x90
Exception Code	1 Byte	01 oder 02 oder 03 oder 04 oder 11



- Exception Codes** Eine Fehlnachricht (Exception Response) besitzt zwei Felder, die sie von einer normalen Antwort unterscheidet:
- **Feld Funktionscode**  
In einer normalen Antwort wird der Funktionscode der Anforderung in das Funktionscode Feld der Antwort übernommen. Bei allen Funktionscodes ist das höchstwertige Bit (MSB) 0 (die Werte der Funktionscodes sind alle kleiner als 0x80). In einer Fehlnachricht wird das MSB auf 1 gesetzt. Das bedeutet eine Addition des Wertes für den Funktionscode mit 0x80. Aufgrund des gesetzten MSB kann der Master die Antwort als Fehlnachricht identifizieren.
  - **Datenfeld**  
Eine normale Antwort enthält Daten oder Statistikwerte im Datenfeld. Bei einer Fehlnachricht wird ein Exception Code im Datenfeld zurückgeliefert. Dieser Exception Code zeigt den Grund für die Fehlnachricht an.

Modbus Exception Codes		
Code	Name	Bedeutung
01	ILLEGAL FUNCTION	Der Funktionscode in der Anforderung wird vom Slave nicht unterstützt.
02	ILLEGAL DATA ADDRESS	Es werden ungültige Registeradressen abgefragt.
03	ILLEGAL DATA VALUE	Ein Wert in der Anforderung ist außerhalb des gültigen Bereichs. Dies gilt sowohl für die Felder einer Anforderung (z. B. ungültige Anzahl an Registern) als auch für ungültige Einstellungswerte der SunSpec Inverter Control Models.
04	SLAVE DEVICE FAILURE	Während des Versuchs, ein oder mehrere Register zu beschreiben, ist ein Fehler aufgetreten.
11	GATEWAY TARGET DEVICE FAILED TO RESPOND	Das angesprochene Gerät ist ausgeschaltet und kann nicht gefunden werden.

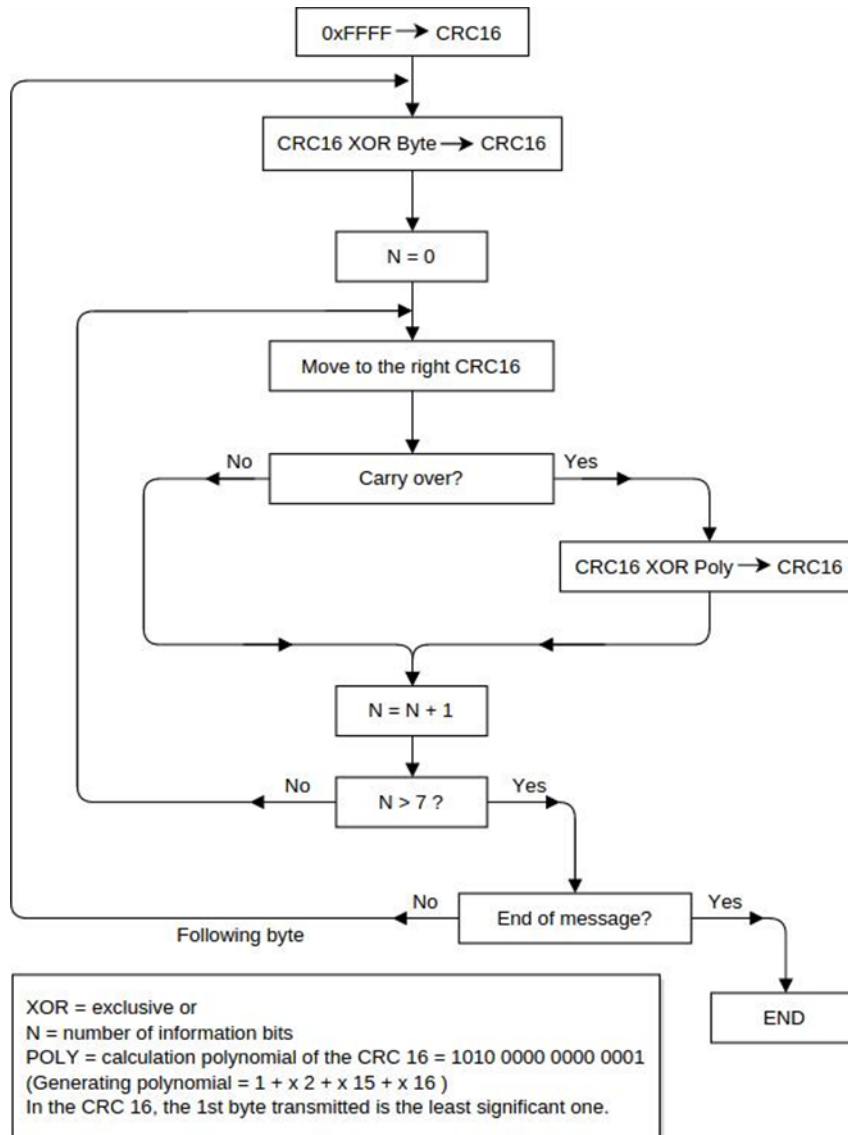
### CRC Berechnung für Modbus RTU

Jede Modbus RTU Nachricht wird mit einer Prüfsumme (CRC, Cyclic Redundancy Check) versehen, um Übertragungsfehler erkennen zu können. Die Prüfsumme ist 2 Bytes groß. Sie wird vom sendenden Gerät berechnet und an die zu sendende Nachricht angehängt. Der Empfänger berechnet seinerseits über alle Bytes der erhaltenen Nachricht (ohne CRC) die Prüfsumme und vergleicht diese mit der empfangenen Prüfsumme. Wenn diese beiden Prüfsummen unterschiedlich sind, ist ein Fehler aufgetreten.

Die Berechnung der Prüfsumme beginnt mit dem Setzen aller Bits eines 16 Bit Registers (CRC Register) auf 1 (0xFFFF). Danach werden alle Bytes der Nachricht einzeln mit dem CRC Register verarbeitet. Nur die Datenbytes einer Nachricht werden zur Berechnung herangezogen. Start-, Stopp- und Paritätsbits werden nicht berücksichtigt.

Während der Berechnung der CRC wird jedes Byte mit dem CRC Register XOR-verknüpft. Danach wird das Ergebnis in Richtung des niederwertigsten Bits (LSB) verschoben und das höchstwertige Bit (MSB) auf 0 gesetzt. Das LSB wird betrachtet. Wenn das LSB vorhin 1 war, wird das CRC Register mit einem fix vorgegebenen Wert XOR-verknüpft. War das LSB 0, dann ist nichts zu tun.

Dieser Prozess wird so oft wiederholt, bis das CRC Register 8 Mal verschoben wurde. Nach dem letzten (achten) Schiebepvorgang, wird das nächste Byte genommen und mit dem aktuellen CRC Register XOR-verknüpft. Danach beginnt der Schiebepprozess von vorne; wieder wird 8 Mal verschoben. Nach Abhandlung aller Bytes der Nachricht ist der Wert des CRC Registers die Prüfsumme.



Berechnungsalgorithmus der CRC16

### CRC Prüfsumme berechnen

- 1 Initialisierung eines 16 Bit Registers (2 Bytes) mit 0xFFFF. Dieses Register wird als CRC16 Register bezeichnet.
- 2 XOR-Verknüpfung des ersten Bytes der Nachricht mit dem niederwertigen Byte des CRC16 Registers. Das Ergebnis wird im CRC16 Register gespeichert.
- 3 Verschieben des CRC16 Registers um 1 Bit nach rechts (in Richtung LSB), MSB mit 0 auffüllen. LSB betrachten.
- 4 LSB Wert überprüfen
  - War das LSB 0: Gehe zu Schritt 3 (neuerlich verschieben).
  - War das LSB 1: XOR Verknüpfung des CRC16 Registers mit dem CRC Polynom 0xA001 (1010 0000 0000 0001).

- 5 Wiederholung der Schritte 3 und 4 bis 8 Schiebeoperationen durchgeführt worden sind. Wenn diese durchgeführt wurden, wurde ein komplettes Byte der Nachricht bearbeitet.
- 6 Wiederholung der Schritte 3 bis 5 für das nächste Byte der Nachricht. Das ganze wiederholen bis alle Bytes der Nachricht abgearbeitet wurden.
- 7 Nach dem letzten Byte enthält das CRC16 Register die Prüfsumme.
- 8 Wenn die Prüfsumme an die zu sendende Nachricht angehängt wird, dann müssen die beiden Bytes wie unten beschreiben vertauscht werden.

**CRC Prüfsumme zur Nachricht hinzufügen**

Wenn die 16 Bit (2 Bytes) CRC Prüfsumme mit einer Nachricht versendet wird, dann wird das niederwertige vor dem höherwertigen Byte übertragen.

Zum Beispiel, wenn die CRC Prüfsumme 0x1241 (0001 0010 0100 0001) ist:

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi
							0x41	0x12

# Allgemeines

## Verwendete Abkürzungen

AC	Wechselstrom
DC	Gleichstrom
FW	Firmware
PF	Power Factor (cos $\phi$ )
PV	Photovoltaik
RTC	Echtzeit-Uhr
SF	Skalierungsfaktor
SW	Software
V	Spannung (Volt)
VA	Scheinleistung
VAr	Blindleistung
VMax	Maximale Spannung
VMin	Minimale Spannung
VRef	Referenzspannung
W	Leistung (Watt)
WR	Wechselrichter

## Register Maps

### Wechselrichter

<b>SID</b> Identifizierung als SunSpec Gerät
<b>Common Block</b> Geräteinformationen
<b>Inverter Model</b> Wechselrichter-Daten
<b>Nameplate Model</b>
<b>Basic Settings Model</b>
<b>Ext. Measurement Model</b>
<b>Immediate Controls Model</b>
<b>Multi. MPPT Inv. Ext. Model</b>
<b>Basic Storage Control</b>
<b>End Block</b>

### Energiezähler

<b>SID</b> Identifizierung als SunSpec Gerät
<b>Common Block</b> Geräteinformationen
<b>Meter Model</b> Energiezähler-Daten
<b>End Block</b>

Die Registerlisten können von der Fronius Homepage heruntergeladen werden:

[https://www.fronius.com/de/downloads / Solar Energy / Modbus SunSpec Maps, State Codes und Events](https://www.fronius.com/de/downloads/Solar%20Energy/Modbus%20SunSpec%20Maps,%20State%20Codes%20und%20Events)

**Antwortzeiten****Empfehlung für Timeout-Werte**

Modbus-Abfragen sollten nur sequenziell und nicht parallel ausgeführt werden (maximal 2 Anfragen parallel). Die Abfragen mit einem Timeout von mindestens 1 Sekunden durchführen. Abfragen im Millisekunden-Takt können zu langen Antwortzeiten führen.

Multiple Registerabfragen in einer Nachricht sind schneller als mehrere Abfragen einzelner Register.

**Modbus Geräte-ID für Wechselrichter**

TCP: Die unit-id vom Wechselrichter ist immer 0x01. Die Identifikation ist durch die IP-Adresse möglich.

RTU: Die slave-id muss auf der Weboberfläche des GEN24 konfiguriert werden. Es können mehrere GEN24-Geräte zusammengeschaltet werden. Jedes Gerät muss eine eindeutige Nummer besitzen.

**Modbus Geräte-ID für Energiezähler**

Ist ein Energiezähler (z. B. Fronius Smart Meter 63A) per Modbus RTU angeschlossen, kann dieser per Modbus TCP über die einstellbare Modbus Geräte-ID ausgelesen werden.

*Fronius Smart Meter  
Adresse*

*Modbus Geräte-ID*

1	200 (default, Achtung - 240 bei SnapINverter)
2	201
3	202
4	203
5	204

**Registeradressen****WICHTIG!**

- Registeradressen bleiben nicht konstant.
- Die tatsächlichen Registeradressen sind abhängig von der Zusammensetzung der dynamischen Sunspec Registerliste.

Richtige Vorgehensweise:

- das Model per Abfrage suchen (Startadresse ermitteln)
- dann mit Offsets arbeiten

Um ein Register auszulesen muss in der Modbus-Anfrage die Startadresse des Registers angegeben werden.

SunSpec Basis Register: 40001

Register beginnen bei 1 und stellen keinen Funktionscode dar.

Register nicht mit dem Modicon Adress-Schema verwechseln:

Beim Modicon Adress-Schema wird 40001 als 4x40001 dargestellt.

Um Register 40001 auszulesen, die Adresse 40000 (0x9C40) verwenden.

Die ausgesendete Registeradresse ist also immer um 1 geringer als die eigentliche Registernummer.

## WICHTIG!

**Aufgrund der verwendeten Datentypen können sich die Längen von einzelnen Models verändern.**

Daher werden bei einigen Registertabellen für SunSpec Models Startadressen angegeben.

Diese Startadresse zusammen mit dem Offset aus der Tabelle ergibt dann den Wert der tatsächlichen Registernummer.

**Beispiel:** Tabelle **Nameplate Model (120)** auf Seite **23**:

Das Register *WRtg* des Nameplate Model hat einen Offset von 4. Die Startadresse ist bei der Einstellung „float“ mit 40131 angegeben.

Somit ist die korrekte Registernummer:  $40131 + 4 = 40135$ .

---

## Beispiele für Modbus RTU:

### 1. Abfrage von 4 Registern ab Register 40005 (Mn, Manufacturer)

Senden (Bytes in Hexadezimal)

01	03	9C 44	00 04	2A 4C
Geräte-ID	Function Code	Adresse 40004 (entspricht Register 40005)	Anzahl der auszulesenden Register	Checksumme Low Byte High Byte

Empfangen (Bytes in Hexadezimal)

01	03	08	46 72	6F 6E	69 75	73 00	8A 2A
Geräte-ID	Function Code	Anzahl der Bytes	Adresse 40005 "F" und "r"	Adresse 40006 "o" und "n"	Adresse 40007 "i" und "u"	Adresse 40008 "s" und 0	Checksumme Low Byte High Byte

**2. Schreiben von 1 Register ab Register 40242 (WmaxLimPct)**

01	10	9D 32	00 01	02	13 88	E3 DD
Gerät e-ID	Function Code	Adresse 40242	Anzahl der zu schreibenden Register	Anzahl Datenbytes, die noch folgen	zu schreibender Registerwert 0x1388 = 5000	Checksumme  Low Byte High Byte

01	10	9D 32	00 01	8F AA	
Gerät e-ID	Function Code	Adresse 40242	Anzahl der geschriebenen Register	Checksumme "i" und "u"	40008 "s" und o
				Low Byte High Byte	

**Beispiele für Modbus TCP:**

**1. Abfrage von 4 Registern ab Register 40005 (Mn, Manufacturer)**

Senden (Bytes in Hexadezimal)

MBAP Header	03	9C 44	00 04
Details siehe Beschreibung MBAP Header	Function Code	Adresse 40004 (entspricht Register 40005)	Anzahl der auszulesenden Register

Empfangen (Bytes in Hexadezimal)

MBAP Header	03	08 46 72	6F 6E	69 75	73 00	
Details siehe Beschreibung MBAP Header	Function Code	Anzahl der Bytes	Adresse 40005 "F" und "r"	Adresse 40006 "o" und "n"	Adresse 40007 "i" und "u"	Adresse 40008 "s" und o

## 2. Schreiben von 1 Register ab Register 40242 (WmaxLimPct)

MBAP Header	10	9D 32	00 01	02	13 88
Details siehe Beschreibung MBAP Header	Function Code	Adresse 40242	Anzahl der zu schreibenden Register	Anzahl Datenbytes, die noch folgen	zu schreibender Registerwert 0x1388 = 5000

MBAP Header	10	9D 32	00 01
Details siehe Beschreibung MBAP Header	Function Code	Adresse 40242	Anzahl der geschriebenen Register

### Nicht vorhandene Datensätze

Fronius Wechselrichter können nicht immer alle Daten, die in den SunSpec-Datenmodellen spezifiziert sind, zur Verfügung stellen. Diese Daten werden je nach Datentyp laut SunSpec Spezifikation durch folgende Werte dargestellt:

- int16 (-32767 bis 32767): 0x8000<sup>1)</sup>
- uint16 (0 bis 65534): 0xFFFF
- acc16 (0 bis 65535): 0
- enum16 (0 bis 65534): 0xFFFF
- bitfield16 (0 bis 0x7FFF): 0xFFFF
- pad (0x8000): immer 0x8000
- int32 (-2147483647 bis 2147483647) : 0x80000000
- uint32 (0 bis 4294967294): 0xFFFFFFFF
- acc32 (0 bis 4294967295) : 0
- enum32 (0 bis 4294967294): 0xFFFFFFFF
- bitfield32 (0 bis 0xFFFFFFFF): 0xFFFFFFFF
- int64 (-9223372036854775807 bis 9223372036854775807): 0x8000000000000000
- uint64 (0 bis 18446744073709551615): 0xFFFFFFFFFFFFFFFF
- acc64 (0 bis 18446744073709551615): FFF
- stringX: 0
- float32 (Bereich siehe IEEE 754): alle X Register mit 0x0000 gefüllt
- sunssf (Skalierungsfaktoren; -10 bis 10): 0x7FC00000 (NaN)  
0x8000

<sup>1)</sup> Das Prefix "0x" steht für hexadezimale Zahlen

### HINWEIS!

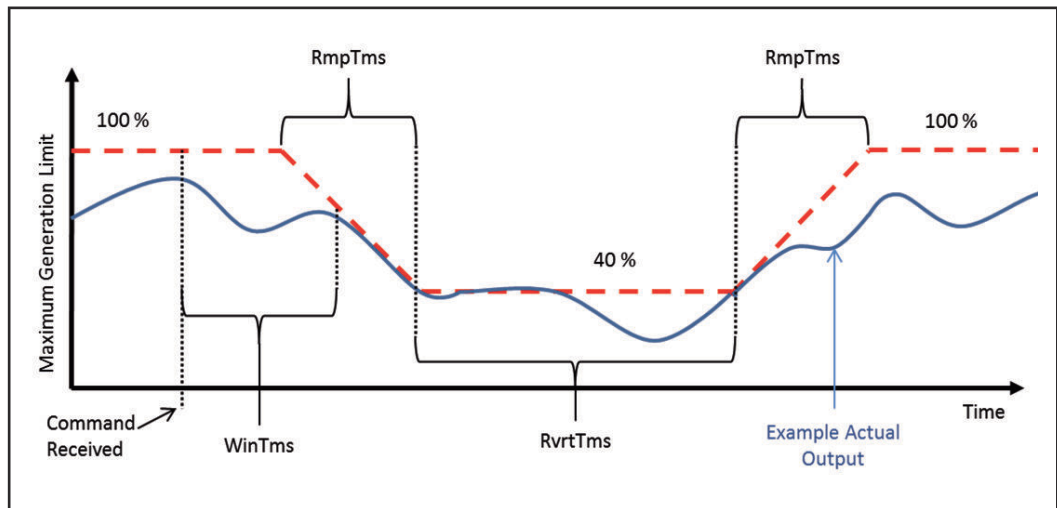
**Nicht unterstützte Datenpunkte sind in den Registertabellen in der Spalte „Range of values“ mit „Not supported“ gekennzeichnet.**

In diesem Fall erhält man beim Auslesen je nach Datentyp den entsprechenden Wert aus der obigen Liste.

In bestimmten Fällen kann es vorkommen, dass grundsätzlich als unterstützt angeführte Register ebenfalls einen solchen Wert zurückliefern. Der Grund dafür ist, dass einige Werte vom Gerätetyp abhängig sind, z.B. die Ströme AphB und AphC bei einem einphasigen Wechselrichter.



## Zeitverhalten der unterstützten Betriebsarten



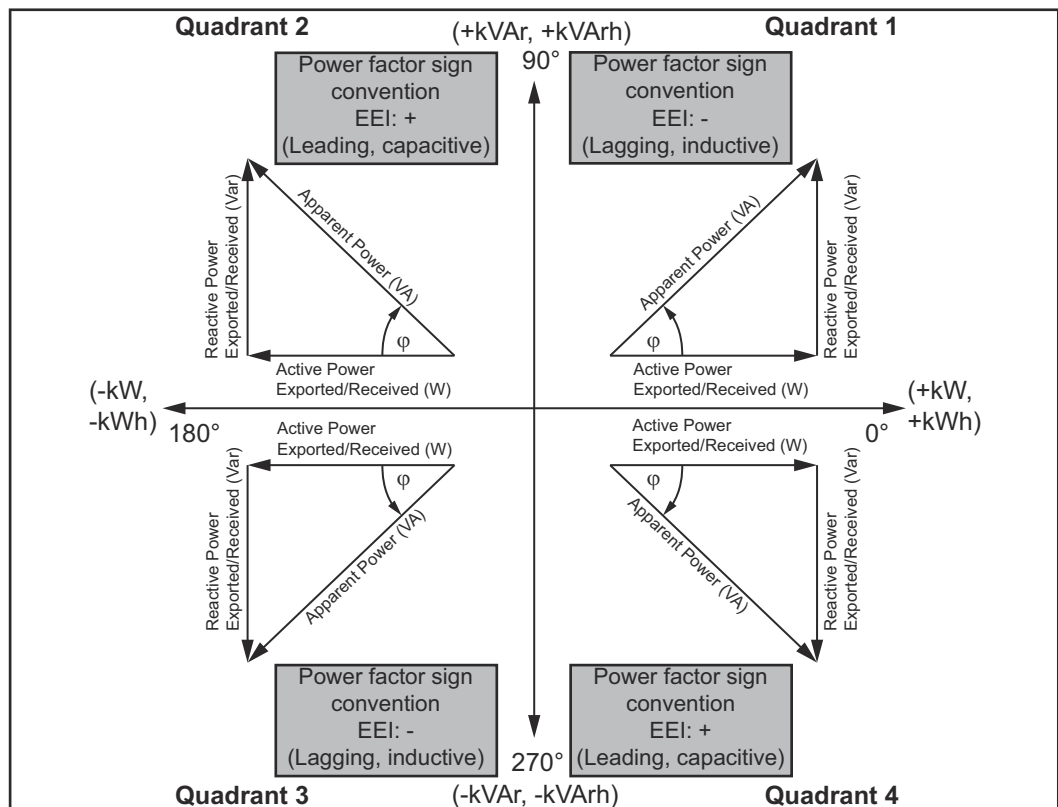
Zeitverhalten am Beispiel einer Leistungsreduktion

Das Zeitverhalten des Wechselrichters in einer Betriebsart kann durch mehrere Zeitwerte festgelegt werden.

In der Abbildung "Zeitverhalten am Beispiel einer Leistungsreduktion" sind die drei möglichen Zeitwerte dargestellt:

- **WinTms 0 - 300 [Sekunden]**  
gibt ein Zeitfenster an, in dem die Betriebsart zufällig gestartet wird. Das Zeitfenster beginnt mit dem Startbefehl der Betriebsart (z.B. *OutPF-Set\_Ena* = 1).  
Mit *WinTms* kann verhindert werden, dass alle Wechselrichter in der Anlage die Änderungen gleichzeitig übernehmen. Bei 0 (Standardwert) startet die Betriebsart sofort.
- **RvrtTms 0 - 28800 [Sekunden]**  
bestimmt, wie lange die Betriebsart aktiv sein soll. Mit jeder empfangenen Modbus Nachricht wird der Timer neu gestartet. Wenn während der Fallback-Zeit (= *RvrtTms*) keine neue Modbus Nachricht empfangen wurde, wird die Betriebsart automatisch beendet und auf die Betriebsart mit der nächsten Priorität zurückgeschaltet, beispielsweise auf dynamische Leistungsreduzierung. Ist *RvrtTms* = 0 (Standardwert) bleibt die Betriebsart so lange aktiv, bis diese manuell über das entsprechende Register wieder deaktiviert wird. Die Fallback Option steht in diesem Fall nicht zur Verfügung.
- **RmpTms**  
gibt vor, wie schnell die Änderungen durchgeführt werden sollen. Der entsprechende Wert wird in der angegebenen Zeit schrittweise vom alten zum neuen Wert hin verändert.  
Ist *RmpTms* = 0 (Standardwert) oder wird dieser Wert gar nicht unterstützt, wird sofort der neue Wert aktuell.

## Vorzeichenkonvention für den Power Factor



Die EEI-Vorzeichenkonvention<sup>1)</sup> für den Power Factor entspricht der SunSpec Spezifikation, und basiert auf den Angaben aus dem "Handbook for Electricity Metering" und der IEC 61557-12 (2007).

Der Power Factor ist:

- negativ bei positiver Blindleistung (übererregt, Quadrant 1)
- positiv bei negativer Blindleistung (untererregt, Quadrant 4)

<sup>1)</sup> EEI = Edison Electrical Institute

## Auf der Karte gespeicherte Werte

Nameplate Model (IC120):

- **WRtg**  
AC Nennleistung des Wechselrichters
- **VARtg**  
AC Nennscheinleistung des Wechselrichters  
Standardwert = WRtg
- **VArRtgQ1**  
Maximale AC Blindleistung im 1. Quadranten (übererregt).  
Standardwert wird anhand von verfügbarem cos Phi (0.85) und der Nennscheinleistung berechnet. Skalierungsfaktor VArRtg\_SF beachten
- **VArRtgQ4**  
Maximale AC Blindleistung im 4. Quadranten (untererregt).  
Standardwert wird anhand von verfügbarem cos Phi (0.85) und der Nennscheinleistung berechnet. Skalierungsfaktor VArRtg\_SF beachten
- **ARtg**  
AC Nennstrom des Wechselrichters

Basic Settings Model (IC121):

- **WMax**  
Maximale AC Leistung  
Standardwert = WRtg
- **VRef**  
Referenzspannung am Einspeisepunkt
- **VRefOfs**  
Abweichung zur Referenzspannung
- **VMax**  
Maximale AC Spannung
- **VMin**  
Minimale AC Spannung
- **VAMax**  
Maximale AC Scheinleistung  
Standardwert = VARTg

### Skalierungsfaktoren

**WICHTIG!** Skalierungsfaktoren (auch bei Auswahl von "Float" möglich!) sind nicht statisch, auch wenn diese als Fixwert in dieser BA angegeben werden. Skalierungsfaktoren können sich bei jeder Firmware-Änderung und auch zur Laufzeit (auto-scale) verändern (z.B.: Skalierungsfaktor für Leistungsvorgabe).

Skalierungsfaktoren mit unveränderlichen Werten sind in den Tabellen in der Spalte "Range of values" angeführt. Aktualdaten (Daten von Wechselrichtern und Energiezählern) können veränderliche Skalierungsfaktoren haben. Diese müssen aus den entsprechenden Registern ausgelesen werden.

Der Datentyp „sunssf“ ist ein signed integer mit 16bit.

Rechenbeispiel:

(Model 160):  $1\_DCW = 10000$ ,  $DCW\_SF = -1$  -> Leistung =  $10000 \times 10^{(-1)} = 1000$  W

### Nicht beschreibbare Register

Folgende Register können nicht beschrieben werden:

- Read-Only (R) Register
- aktuell nicht unterstützte Register

#### HINWEIS!

**Wird versucht solche Register zu beschreiben, gibt der Wechselrichter keinen Exception Code zurück!**

**Die in diese Register geschriebenen Werte werden ohne Fehlermeldung ignoriert.**

**Im Model 123 und 124 kommt eine Exception beim Schreibzugriff, wenn die Steuerungsmöglichkeit im lokalen Webinterface deaktiviert wurden.**

### Schreiben ungültiger Werte

Einige Register lassen nur bestimmte Werte zu. Die gültigen Werte sind der jeweiligen Register-Tabelle zu entnehmen.

Wird ein ungültiger Wert in ein Register geschrieben, so gibt der Fronius Datamanager den Exception Code 3 (Illegal Data Value) zurück. Der ungültige Wert wird ignoriert.

Werden mehrere Register auf einmal beschrieben, werden alle gültigen Werte bis zu dem Register mit dem ungültigen Wert geschrieben. Anschließend wird der Schreibvorgang abgebrochen.

# Einstellungen - Modbus

---

**Allgemeines** Über die Web-Schnittstelle des Wechselrichters können via Internet-Browser Einstellungen für die Modbus Anbindung vorgenommen werden, welche über das Modbus-Protokoll nicht ansprechbar sind.

---

- Einstellungen - Modbus öffnen**
- 1** Die Weboberfläche des Wechselrichters öffnen
  - 2** Den Bereich "Kommunikation" (1) auswählen
  - 3** Den Menüpunkt "Modbus" (2) öffnen
- 

## Modbus

### Modbus RTU-Schnittstelle 0 / 1

Wird eine der beiden Modbus RTU Schnittstellen auf Slave gestellt, stehen folgende Eingabefelder zur Verfügung:

---

#### „Baudrate“

Die Baudrate beeinflusst die Geschwindigkeit der Übertragung zwischen den einzelnen im System angeschlossenen Komponenten. Bei der Auswahl der Baudrate ist darauf zu achten, dass diese auf Sende- und Empfangsseite gleich sind.

---

#### „Parität“

Das Paritätsbit kann zur Paritätskontrolle genutzt werden. Diese dient der Erkennung von Übertragungsfehlern. Ein Paritätsbit kann dabei eine bestimmte Anzahl von Bits absichern. Der Wert (0 oder 1) des Paritätsbits muss beim Sender berechnet werden und wird beim Empfänger mithilfe der gleichen Berechnung überprüft. Die Berechnung des Paritätsbits kann für gerade oder ungerade Parität erfolgen..

---

#### „SunSpec Model Type“

Je nach Sunspec Model gibt es zwei verschiedene Einstellungen.

**float:** SunSpec Inverter Model 111, 112, 113 bzw. 211, 212, 213.

**int + SF:** SunSpec Inverter Model 101, 102, 103 bzw. 201, 202, 203.

---

#### „Zähleradresse“

Der eingegebene Wert ist die dem Zähler zugewiesene Identifikationsnummer (Unit ID). Zu finden auf der Webseite des Wechselrichters im Menü Kommunikation - Modbus.

Werkseinstellung: 200

---

#### „Wechselrichteradresse“

Der eingegebene Wert ist die dem Wechselrichter zugewiesene Identifikationsnummer (Unit ID). Zu finden auf der Webseite des Wechselrichters im Menü Kommunikation - Modbus.

Werkseinstellung: 1

---

### Slave als Modbus TCP

Wird die Funktion „Slave als Modbus TCP“ aktiviert, stehen folgende Eingabefelder zur Verfügung:

---

#### „Modbus-Port“

Nummer des TCP Ports, der für die Modbus-Kommunikation zu verwenden ist.

---

„SunSpec Model Type“

Je nach SunSpec Model gibt es zwei verschiedene Einstellungen.

**float:** SunSpec Inverter Model 111, 112, 113 bzw. 211, 212, 213.

**int + SF:** SunSpec Inverter Model 101, 102, 103 bzw. 201, 202, 203.

„Zähleradresse“

Der eingegebene Wert ist die dem Zähler zugewiesene Identifikationsnummer (Unit ID).

Zu finden auf der Webseite des Wechselrichters im Menü Kommunikation - Modbus.

Werkseinstellung: 200

„Wechselrichteradresse“

Der eingegebene Wert ist die dem Wechselrichter zugewiesene Identifikationsnummer (Unit ID). Zu finden auf der Webseite des Wechselrichters im Menü Kommunikation - Modbus.

Werkseinstellung: Dieser Wert ist unveränderbar mit 1 definiert.

### Wechselrichter-Steuerung über Modbus

Wenn diese Option aktiviert ist, erfolgt die Wechselrichter-Steuerung über Modbus.

Zur Wechselrichter-Steuerung gehören folgende Funktionen:

- Ein / Aus
- Leistungsreduktion
- Vorgabe eines konstanten Power Factors (cos Phi)
- Vorgabe einer konstanten Blindleistung
- Batteriesteuerungsvorgaben mit Batterie

### Steuerung einschränken

Die Option „Steuerung einschränken“ ist nur beim Übertragungsprotokollen tcp verfügbar.

Sie dient dazu Wechselrichter-Steuerungsbefehle durch Unbefugte zu verhindern, indem die Steuerung nur für bestimmte Geräte erlaubt wird.

#### Steuerung einschränken

Wenn diese Option aktiviert ist, dürfen nur bestimmte Geräte Steuerungsbefehle schicken.

#### IP-Adresse

Um die Wechselrichter-Steuerung auf ein oder mehrere Geräte zu beschränken, werden in diesem Feld die IP-Adressen jener Geräte eingetragen die Befehle an den Wechselrichter senden dürfen. Mehrere Einträge werden durch Beistriche getrennt.

Beispiele:

- eine IP-Adresse: **98.7.65.4**
  - Steuerung nur durch IP Adresse 98.7.65.4 zulässig
- mehrere IP-Adressen: **98.7.65.4,222.44.33.1**
  - Steuerung nur durch IP Adressen 98.7.65.4 und 222.44.33.1 zulässig
- IP-Adressbereich z.B. von 98.7.65.1 bis 98.7.65.254 (CIDR Notation): **98.7.65.0/24**
  - Steuerung nur durch IP Adressen 98.7.65.1 bis 98.7.65.254 zulässig

# Common & Inverter Model

## Common Block Register

Die Beschreibung des Common Block inklusive der SID Register (Register 40001-40002) zur Identifizierung als SunSpec Gerät gilt für jeden Gerätetyp (Wechselrichter, Energiezähler). Jedes Gerät besitzt einen eigenen Common Block, in dem Informationen über das Gerät (Modell, Seriennummer, SW Version, etc.) aufgeführt sind.

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

## Inverter Model Register

Für die Wechselrichter-Daten werden zwei verschiedene SunSpec Models unterstützt:

- das standardmäßig eingestellte Inverter Model mit Gleitkomma-Darstellung (Einstellung „float“; 111, 112 oder 113)
- das Inverter Model mit ganzen Zahlen und Skalierungsfaktoren (Einstellung „int+SF“; 101, 102 oder 103)

Die Registeranzahl der beiden Model-Typen ist unterschiedlich!

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

## SunSpec Operating Codes

Name	Wert	Beschreibung
I_STATUS_OFF	1	Wechselrichter ist aus
I_STATUS_SLEEPING	2	Auto-Shutdown
I_STATUS_STARTING	3	Wechselrichter startet
I_STATUS_MPPT	4	Wechselrichter arbeitet normal
I_STATUS_THROTTLED	5	Leistungsreduktion aktiv
I_STATUS_SHUTTING_DOWN	6	Wechselrichter schaltet ab
I_STATUS_FAULT	7	Ein oder mehr Fehler existieren, siehe St *oder Evt * Register
I_STATUS_STANDBY	8	Standby

\* Inverter Model Register

# Nameplate Model (120)

---

## Allgemeines

Dieses Modell entspricht einem Leistungsschild. Folgende Daten können ausgelesen werden:

- ***DERType (3)***  
Art des Geräts. Das Register liefert den Wert 4 zurück (PV-Gerät)
- ***WRtg (4)***  
Nennleistung des Wechselrichters
- ***VARtg (6)***  
Nenn-Scheinleistung des Wechselrichters
- ***VArRtgQ1 (8) - VArRtgQ4 (11)***  
Nenn-Blindleistungswerte für die vier Quadranten
- ***ARtg (13)***  
Nennstrom des Wechselrichters
- ***PFRtgQ1 (15) – PFRtgQ4 (18)***  
Minimale Werte für den Power Factor für die vier Quadranten

---

## Nameplate Register

Startadresse:

- bei Einstellung „float“: **40131**
- bei Einstellung „int+SF“: **40121**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

# Basic Settings Model (121)

## Basic Settings Register

Startadresse:

- bei Einstellung „float“: **40159**
- bei Einstellung „int+SF“: **40149**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

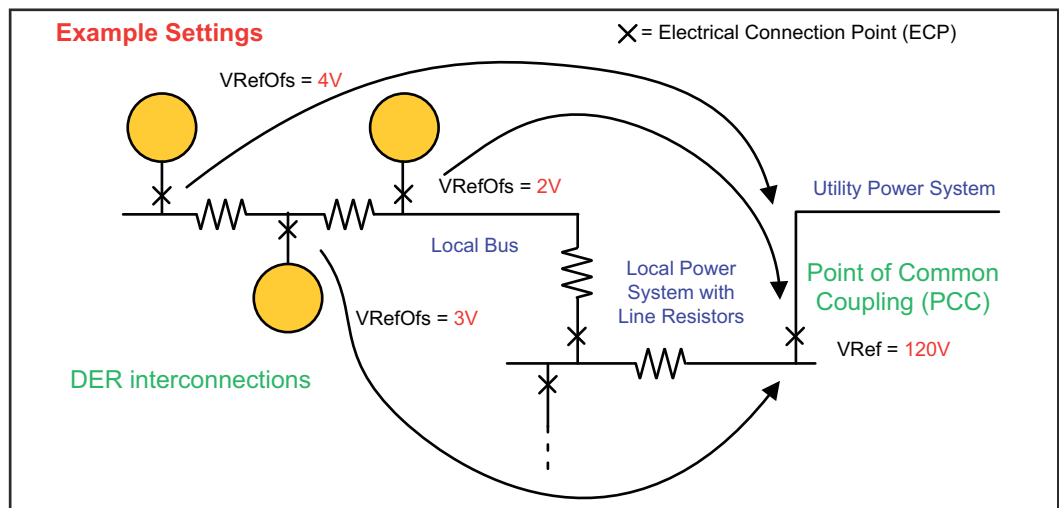
## Referenzspannung

### *VRef (4)*

Die Referenzspannung ist die Spannung an jenem gemeinsamen Anschlusspunkt, an welchem das lokale Netz mit dem öffentlichen Stromnetz verknüpft ist, und entspricht der Nennspannung des Wechselrichters.

=> siehe Abbildung "Gemeinsamer Anschlusspunkt"

Die Angabe erfolgt in Volt im Bereich von 0 (0x0000) bis 400 (0x0190).



Gemeinsamer Anschlusspunkt

## Abweichung zur Referenzspannung

### *VRefOfs (5)*

Je nach Verschaltung des lokalen Netzes kann es am Anschlusspunkt jedes einzelnen Wechselrichters an das lokale Netz zu einer Abweichung zur Referenzspannung kommen (siehe Abbildung "Gemeinsamer Anschlusspunkt").



# Extended Measurements & Status Model (122)

## Allgemeines

Dieses Modell liefert einige zusätzliche Mess- und Statuswerte, die das normale Inverter Model nicht abdeckt:

- **PVConn (3)**  
Dieses Bitfeld zeigt den Status des Wechselrichter an
  - Bit 0: Verbunden
  - Bit 1: Ansprechbar
  - Bit 2: Arbeitet (Wechselrichter speist ein)
  
- **ECPCConn (5)**  
Dieses Register zeigt den Verbindungsstatus zum Netz an
  - ECPCConn = 1: Wechselrichter speist gerade ein
  - ECPCConn = 0: Wechselrichter speist nicht ein
  
- **ActWH (6 - 9)**  
Wirkenergiezähler
  
- **StActCtl (36 - 37)**  
Bitfeld für zurzeit aktive Wechselrichter-Modi
  - Bit 0: Leistungsreduktion (FixedW; entspricht WMaxLimPct Vorgabe)
  - Bit 1: konstante Blindleistungs-Vorgabe (FixedVAR; entspricht VAR-MaxPct)
  - Bit 2: Vorgabe eines konstanten Power Factors (FixedPF; entspricht Out-PFSet)
  
- **TmSrc (38 - 41)**  
Quelle für die Zeitsynchronisation. Das Register liefert den String „RTC“ zurück.
  
- **Tms (42 - 43)**  
Aktuelle Uhrzeit und Datum der RTC  
Angegeben werden die Sekunden vom 1. Jänner 2000 00:00 (UTC) bis zur aktuellen Zeit
  
- **Ris**  
Iso Widerstand

## Extended Measurements & Status Register

Startadresse:

- bei Einstellung „float“: **40191**
- bei Einstellung „int+SF“: **40181**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

# Immediate Controls Model (123)

---

## Allgemeines

Mit den Immediate Controls können folgende Einstellungen am Wechselrichter vorgenommen werden:

- Unterbrechung des Einspeisebetriebs des Wechselrichters (Standby)
- Konstante Reduktion der Ausgangsleistung
- Vorgabe eines konstanten Power Factors
- Vorgabe einer konstanten relativen Blindleistung

Am Webinterface des Wechselrichters muss in den Einstellungen unter Modbus die Einstellung „Wechselrichter-Steuerung über Modbus“ aktiviert sein, um hier schreibend aktiv werden zu können. Je nach eingestellter Steuerungs-Priorität (IO-Steuerung, Dynamische Leistungsreduzierung oder Steuerung über Modbus) werden Modbus Kommandos eventuell nicht angenommen.

---

## Immediate Controls Register

Startadresse:

- bei Einstellung „float“: **40237**
- bei Einstellung „int+SF“: **40227**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

---

## Standby

### **Conn\_WinTms (3) bis Conn (5)**

Diese Register dienen zur Steuerung des Standby Modus (kein Einspeisebetrieb) des Wechselrichters.

### **Conn\_WinTms (3) und Conn\_RvrtTms (4)**

Mit diesen Registern kann das Verhalten des Wechselrichters zeitlich gesteuert werden. => siehe Abschnitt "Zeitverhalten der unterstützten Betriebsarten".

Als Standard ist für alle Register 0 vorgegeben.

### **Conn (5)**

Register *Conn* zeigt an, ob der Wechselrichter aktuell einspeist (0 = Standby, 1 = Einspeisebetrieb).

- Um den Wechselrichter in den Standby zu schalten schreibt man in dieses Register den Wert 0
- Um den Wechselrichter wieder zu aktivieren schreibt man in dieses Register den Wert 1

### **HINWEIS!**

**Ob der Wechselrichter einspeist oder nicht kann auch über das Register ECP-Conn aus dem Extended Measurements and Status Model ausgelesen werden.**

---

## Leistungsreduktion

### **WMaxLimPct (6) bis WMaxLim\_Ena (10)**

Über diese Register kann beim Wechselrichter eine Reduktion der Ausgangsleistung eingestellt werden.

### **WMaxLimPct (6)**

In Register *WMaxLimPct* können Werte zwischen 0% und 100% eingetragen wer-

den.

Die Werte beschränken die maximal mögliche Ausgangsleistung des Gerätes, und haben daher nicht unbedingt eine Auswirkung auf die aktuelle Leistung.

**WICHTIG!** Den Skalierungsfaktor für dieses Register beachten!

Weitere Informationen unter:

<http://sunspec.org/wp-content/uploads/2015/06/SunSpec-Information-Models-12041.pdf>

#### ***WMaxLimPct\_WinTms (7), WMaxLimPct\_RvrtTms (8)***

Mit diesen Registern kann das Verhalten des Wechselrichters für diese Betriebsart zeitlich gesteuert werden. => siehe Abschnitt "Zeitverhalten der unterstützten Betriebsarten".

Als Standard ist für alle Register 0 vorgegeben.

#### ***WMaxLim\_Ena (10)***

Zum Starten und Beenden diese Betriebsart

- Wert 1 in das Register *WMaxLim\_Ena* schreiben = Betriebsart starten
- Wert 0 in das Register *WMaxLim\_Ena* schreiben = Betriebsart beenden

### HINWEIS!

#### **Um bei einer aktiven Betriebsart Werte zu verändern (z.**

B. ein anderes Leistungslimit oder eine andere Rückkehrzeit einstellen), folgendermaßen vorgehen:

- ▶ neuen Wert in das entsprechende Register schreiben
- ▶ die Betriebsart über Register *WMaxLim\_Ena* durch Setzen einer 1 erneut starten

#### **Beispiel: Leistungsreduktion einstellen**

Wenn mit Funktionscode 0x10 (write multiple registers) gearbeitet wird, kann eine Performance-Verbesserung bei den Leistungsvorgaben erreicht werden. Es kann mit nur einem statt zwei Modbusbefehlen die Leistung und das Enable gleichzeitig vorgegeben werden. Es können alle 5 Register (*WMaxLimPct*, *WMaxLimPct\_WinTms*, *WMaxLimPct\_RvrtTms*, *WMaxLimPct\_RmpTms*, *WMaxLim\_Ena*) mit einem Befehl geschrieben werden. Das Schreiben auf das "Read Only"-Register *WMaxLimPct\_RmpTms* erfolgt ohne Rückgabe eines sonst üblichen Exception-(Fehler)-Codes.

Z.B. Registerwerte für 80% Vorgabe ohne Timingvorgaben: 8000, 0, 0, 0, 1

- 1** Wert für die Reduktion der Ausgangsleistung in Register *WMaxLimPct* schreiben  
(z. B. 3000 für 30%)
- 2** Optional die Start- und Rückkehrzeit über Register *WMaxLimPct\_WinTms* und *WMaxLimPct\_RvrtTms* einstellen
- 3** Durch Schreiben von 1 in Register *WMaxLim\_Ena* die Betriebsart starten

**WICHTIG!** Den Skalierungsfaktor für dieses Register beachten!

Weitere Informationen unter:

<http://sunspec.org/wp-content/uploads/2015/06/SunSpec-Information-Models-12041.pdf>

#### **Beispiel: Ändern der Rückkehrzeit bei aktiver Leistungsreduktion**

Leistungsreduktion ursprünglich mit *WMaxLimPct\_RvrtTms* = 0 gestartet, das heißt die Betriebsart muss manuell beendet werden.

- 1** *WMaxLimPct\_RvrtTms* auf z.B. 30 setzen

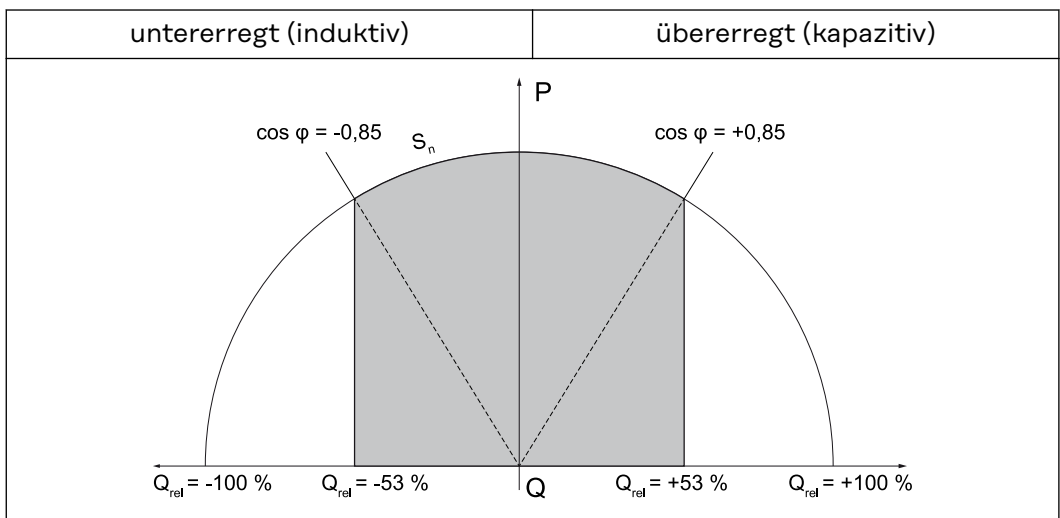
- 2] Durch Schreiben von 1 in Register *WMaxLim\_Ena* Änderung übernehmen
- Betriebsart wird nach 30 Sekunden selbständig beendet und auf die nächste Priorität zurückgestellt (z.B.: Dynamische Leistungsreduzierung)

**Auswirkungen der Blindleistungs-Vorgaben auf die Wirkleistung**

Der Blindleistungs-Betrieb wird grundsätzlich durch den maximalen Ausgangsstrom (die maximale Scheinleistung) sowie durch die operative Blindleistungsgrenze des Wechselrichters begrenzt:

Die folgende Abbildung zeigt den möglichen Arbeitsbereich des Wechselrichters. Alle durch Wirkleistung P und Blindleistung Q definierten gültigen Arbeitspunkte sind innerhalb des grauen Bereiches.

Die Maximalwerte müssen über die Register *VARrtgQ1* bis *VARrtgQ4* und *VARrtg\_SF* aus den Nameplate Model ausgelesen werden.



Blindleistung und Power Factor

**Legende:**

W	Leistung	VAR <sub>max</sub>	Nenn-Blindleistung
W <sub>max</sub>	Nennleistung	VAR <sub>rel</sub>	relative Blindleistung (VAR/VA <sub>max</sub> )
VAR	Blindleistung		

## Konstanter Power Factor

### *OutPFSet (11)* bis *OutPFSet\_Ena (15)*

Über diese Register kann beim Wechselrichter ein konstanter Power Factor vorgegeben werden.

#### *OutPFSet (11)*

- In Register *OutPFSet* können positive und negative Werte für den Power Factor eingegeben werden
- Die Werte sind mit Faktor im Register *OutPFSet\_SF* zu skalieren
- Die minimal möglichen Werte hängen vom Wechselrichter-Typ ab und können dem Nameplate Model entnommen werden

#### HINWEIS!

**Der Wert für den Power Factor muss mit dem korrekten Vorzeichen eingegeben werden, siehe Abschnitt "Vorzeichenkonvention für den Power Factor"**

- ▶ positiv für untererregt
- ▶ negativ für übererregt

### *OutPFSet\_WinTms (12)*, *OutPFSet\_RvrtTms (13)*

Mit diesen Registern kann das Verhalten des Wechselrichters für diese Betriebsart zeitlich gesteuert werden. => siehe Abschnitt "Zeitverhalten der unterstützten Betriebsarten".

Als Standard ist für alle Register 0 vorgegeben.

### *OutPFSet\_Ena (15)*

Zum Starten und Beenden dieser Betriebsart

- Wert 1 in das Register *OutPFSet\_Ena* schreiben = Betriebsart starten
- Wert 0 in das Register *OutPFSet\_Ena* schreiben = Betriebsart beenden

#### HINWEIS!

**Um bei einer aktiven Betriebsart Werte zu verändern (z.B. ein anderen Power Factor oder eine andere Rückkehrzeit einstellen), folgendermaßen vorgehen:**

- ▶ neuen Wert in das entsprechende Register schreiben
- ▶ die Betriebsart über Register *OutPFSet\_Ena* durch Setzen einer 1 erneut starten

### *OutPFSet\_RmpTMS*

-

## Beispiel: Konstanten Power Factor vorgeben

- 1 Wert für den Power Factor in Register *OutPFSet* schreiben (z. B. 950 für 0,95)
- 2 Optional die Start- und Rückkehrzeit über Register *OutPFSet\_WinTms* und *OutPFSet\_RvrtTms* einstellen
- 3 Durch Schreiben von 1 in Register *OutPFSet\_Ena* die Betriebsart starten

---

**Konstante relative Blindleistung**

***VARMaxPct (17) bis VARPct\_Ena (23)***

Über diese Register kann am Wechselrichter ein konstanter Wert für die Blindleistung eingestellt werden, die der Wechselrichter liefern soll.

***VARMaxPct (17)***

- zum Einstellen eines Wertes für die konstante Blindleistung
- Die minimal und maximal möglichen Werte hängen vom Wechselrichter-Typ ab

**HINWEIS!**

**Im praktischen Betrieb wird die tatsächlich verfügbare Blindleistung durch die Betriebsgrenzen des Wechselrichters vorgegeben.**

Deshalb kann die Blindleistungs-Vorgabe nur dann erreicht werden, wenn ausreichend Wirkleistung eingespeist wird.

Wird zu wenig Wirkleistung eingespeist, arbeitet der Wechselrichter an der Betriebsgrenze.

---

***VARPct\_WinTms (19), VARPct\_RvrtTms (20)***

Mit diesen Registern kann das Verhalten des Wechselrichters für diese Betriebsart zeitlich gesteuert werden. => siehe Abschnitt "Zeitverhalten der unterstützten Betriebsarten".

Als Standard ist für alle Register 0 vorgegeben.

***VARPct\_Mod (22)***

- dieses Register kann nicht verändert werden
- liefert die (derzeit) unterstützte Betriebsart zurück  
Blindleistung in Prozent der maximal möglichen Blindleistung

***VARPct\_Ena (23)***

Zum Starten und Beenden dieser Betriebsart

- Wert 1 in das Register *VARPct\_Ena* schreiben = Betriebsart starten
- Wert 0 in das Register *VARPct\_Ena* schreiben = Betriebsart beenden

**HINWEIS!**

**Um bei einer aktiven Betriebsart Werte zu verändern (z.**

B. ein andere Blindleistung oder eine andere Rückkehrzeit einstellen), folgendermaßen vorgehen:

- ▶ neuen Wert in das entsprechende Register schreiben
  - ▶ die Betriebsart über Register *VARPct\_Ena* durch Setzen einer 1 erneut starten
- 

***VARPct\_RmpTms (23)***

- der relative Annäherungswert in %/s
- 

**Beispiel:  
Konstante Blindleistung vorgeben**

- 1 Wert für die relative Blindleistung in Register *VARMaxPct* schreiben (z. B. 80 für 80%)
- 2 Optional die Start- und Rückkehrzeit über Register *VARPct\_WinTms* und *VARPct\_RvrtTms* einstellen
- 3 Durch Schreiben von 1 in Register *VARPct\_Ena* den Betriebsart starten

# Multiple MPPT Inverter Extension Model (160)

## Allgemeines

Das Multiple MPPT Inverter Extension Model beinhaltet die Werte der DC Eingänge des Wechselrichters.

Verfügt der Wechselrichter über mehrere DC Eingänge, so werden Strom, Spannung, Leistung, Energie und Statusmeldungen der einzelnen Eingänge hier aufgelistet. Im Inverter Model (101 -103 oder 111 - 113) wird in diesem Fall nur die gesamte DC Leistung beider Eingänge ausgegeben. DC Strom und DC Spannung werden als "not implemented" angezeigt.

Die Anzahl der Blöcke wird anhand der DC-Eingänge automatisch angepasst. Bei Geräten mit einer Speicherlösung gibt es zwei zusätzliche Blöcke (Laden (MPP3) und Entladen (MPP4)). Die Registeradressen verschieben sich bei den nachfolgenden Models (absolutbezogen auf die Registeradressen).

## Multiple MPPT Inverter Extension Register

Startadresse:

- bei Einstellung „float“: **40263**
- bei Einstellung „int+SF“: **40253**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

# Basic Storage Control Model (124)

---

## Allgemeines

Dieses Model ist nur für Wechselrichter mit einer Speicherlösung verfügbar.

Mit dem Basic Storage Control Model können folgende Einstellungen am Wechselrichter vorgenommen werden:

- Vorgabe eines Leistungsfensters, in dem sich die Lade-/Entladeleistung vom Energiespeicher bewegen soll.
- Vorgabe eines minimalen Ladestandes, den der Energiespeicher nicht unterschreiten soll
- Ladung des Energiespeichers vom Netz erlauben/verbieten

### HINWEIS!

**Alle Vorgaben verstehen sich als Empfehlungen!  
Der Wechselrichter kann von den Vorgaben abweichen, wenn dies aus Gründen der Betriebssicherheit erforderlich ist.**

---

## Bereitgestellte Informationen

Das Basic Storage Control Model stellt folgende Informationen lesend zu Verfügung:

### WChaMax

- Wenn ein Energiespeicher verfügbar ist liefert dieses Register den Bezugswert für die Register OutWRt und InWRt zurück.  
 $WChaMax := \max(\text{MaxChaRte}, \text{MaxDisChaRte})$
- Wenn kein Energiespeicher verfügbar ist liefert das Register den Wert 0 zurück.

### ChaState

- Ladestand des Energiespeicher in %:  
 $\text{Estimated\_Capacity\_Remaining [Wh]} / \text{Estimated\_Capacity\_Maximum [Wh]}$

### ChaSt

Betriebsstatus des Energiespeichers

- OFF: Energiespeicher ist nicht verfügbar
- EMPTY: Energiespeicher ist derzeit vollständig entladen
- DISCHARGING: Energiespeicher wird derzeit entladen
- CHARGING: Energiespeicher wird derzeit geladen
- FULL: Energiespeicher ist derzeit vollständig geladen
- HOLDING: Energiespeicher wird derzeit weder geladen noch entladen
- TESTING: wird während Kalibrations- oder Service-Ladung benutzt

## Leistungsfenster-Vorgaben

Am Webinterface des Wechselrichters muss in den Einstellungen unter Modbus die Einstellung „Wechselrichter-Steuerung über Modbus“ aktiviert sein, um hier schreibend aktiv werden zu können. Je nach eingestellter Steuerungs-Priorität (IO-Steuerung, Dynamische Leistungsreduzierung oder Steuerung über Modbus) werden Modbus Kommandos eventuell nicht angenommen.

Für die folgenden Beispiele wird  $WchaMax = 3300 \text{ W}$  angenommen.

Für resultierende Leistungsfenster gilt:

- negative Leistungswerte entsprechen einer Ladung des Energiespeichers
- positive Werte entsprechen einer Entladung des Energiespeichers



**Die Werte in den folgenden Beispielen müssen nach dem Lesen und vor dem Schreiben entsprechend ihren Skalierungsfaktoren in den angegebenen Skalierungsregistern skaliert werden.**

Die Manipulation der Register InWRte, OutWRte und StorCtl\_Mod führt zu Änderungen des Batteriestatus in Fronius Solar.web, zum Beispiel „Erzwungene Nachladung“ und „Energiesparmodus“, abhängig von den Benutzereinstellungen und dem aktuellen Status der Batterie.

---

#### **Beispiel 1: Nur Laden des Energiespeichers erlauben**

Dieses Verhalten kann durch Limitierung der maximalen Entladeleistung auf 0% erreicht werden => resultiert in Fenster [-3300 W, 0 W]

- OutWRte = 0% (setze Entladelimit auf 0% von WchaMax)
- StorCtl\_Mod = 2 (schaltet Entladegrenzwert aktiv, Bit-Muster: 10)
- InWRte ist in diesem Fall nicht relevant

#### **Beispiel 2: Nur Entladen des Energiespeichers erlauben**

Dieses Verhalten kann durch Limitierung der maximalen Ladeleistung auf 0% erreicht werden => resultiert in Fenster [0 W, 3300 W]

- InWRte = 0% (setze Ladelimit auf 0% von WchaMax)
- StorCtl\_Mod = 1 (Bit 1 schaltet Ladegrenzwert aktiv, Bit-Muster: 01)
- OutWRte ist in diesem Fall nicht relevant

#### **Beispiel 3: Weder Laden noch Entladen erlauben**

Dieses Verhalten kann durch Limitierung der maximalen Ladeleistung auf 0% und Limitierung der maximalen Entladeleistung auf 0% erreicht werden

=> resultiert in Fenster [0 W, 0 W]

- InWRte = 0% (setze Ladelimit auf 0% von WchaMax)
- OutWRte = 0% (setze Entladelimit auf 0% von WchaMax)
- StorCtl\_Mod = 3 (schalte beide Grenzwerte aktiv, Bit-Muster: 11)

#### **Beispiel 4: Laden und Entladen mit maximal 50% der nominalen Leistung**

Dieses Verhalten kann durch Limitierung der maximalen Ladeleistung auf 50% und Limitierung der maximalen Entladeleistung auf 50% erreicht werden

=> resultiert in Fenster [-1650 W, 1650 W]

- InWRte = 50% (setze Ladelimit auf 50% von WchaMax)
- OutWRte = 50% (setze Entladelimit auf 50% von WchaMax)
- StorCtl\_Mod = 3 (schalte beide Grenzwerte aktiv, Bit-Muster: 11)

#### **Beispiel 5: Laden im Bereich von 50% bis 75% der nominalen Leistung**

Dieses Verhalten kann durch Limitierung der maximalen Ladeleistung auf 75% und Limitierung der maximalen Entladeleistung auf -50% erreicht werden

=> resultiert in Fenster [1650 W, 2475 W]

- InWRte = 75% (setze Ladelimit auf 75% von WchaMax)
- OutWRte = -50% (setze Entladelimit auf -50% von WchaMax)
- StorCtl\_Mod = 3 (schalte beide Grenzwerte aktiv, Bit-Muster: 11)
- Der Batteriestatus in Fronius Solar.web wechselt zu „Erzwungene Nachladung“

#### **Beispiel 6: Entladen mit 50% der nominalen Leistung**

Dieses Verhalten kann durch Limitierung der maximalen Ladeleistung auf -50% und Limitierung der maximalen Entladeleistung auf 50% erreicht werden

=> resultiert in Fenster [-1650 W, -1650 W]

- InWRte = -50% (setze Ladelimit auf -50% von WchaMax)
- OutWRte = 50% (setze Entladelimit auf 50% von WchaMax)
- StorCtl\_Mod = 3 (schalte beide Grenzwerte aktiv, Bit-Muster: 11)

### Beispiel 7: Laden mit 50% bis 100% der nominalen Leistung

Dieses Verhalten kann durch Limitierung der maximalen Entladeleistung auf -50% erreicht werden => resultiert in Fenster [1650 W, 3300 W]

- OutWRte = -50% (setze Entladelimit auf -50% von WchaMax)
- StorCtl\_Mod = 2 (schaltet Entladegrenzwert aktiv, Bit-Muster: 10)
- InWRte ist in diesem Fall nicht relevant
- Der Batteriestatus in Fronius Solar.web wechselt zu „Erzwungene Nachladung“

#### Vorgabe des minimalen Ladezustandes

Durch Setzen von Register MinRsvPct kann ein minimal zu erhaltender Ladezustand des Speichers festgelegt werden. Beispielsweise kann durch Setzen von MinRsvPct=20% eine Reserve von 20% des Ladezustandes reserviert werden, die der Ladezustand nicht unterschreiten soll.

#### Laden des Energiespeichers vom Netz

Mit dem Register ChaGriSet kann es dem Wechselrichter erlaubt oder verboten werden, den Speicher vom Netz zu laden. Das Register CharGriSet und das Feld ‚Batterieladung aus EVU Netz erlauben‘ in den Einstellungen der Fronius Anlagenüberwachung sind UND-verknüpft (Gerätekonfiguration - Komponenten - Batterie). Soll das Verhalten über das Flag ChaGriSet gesteuert werden, muss das Häkchen bei ‚Batterieladung aus EVU Netz erlauben‘ gesetzt sein.

Die Batterie kann über das Modell IC124 aus dem Standby-Betrieb geweckt werden. Wird der SocMin unter den letzten bekannten SoC gesetzt während sich die Batterie im Standby befindet, wird diese aktiviert.

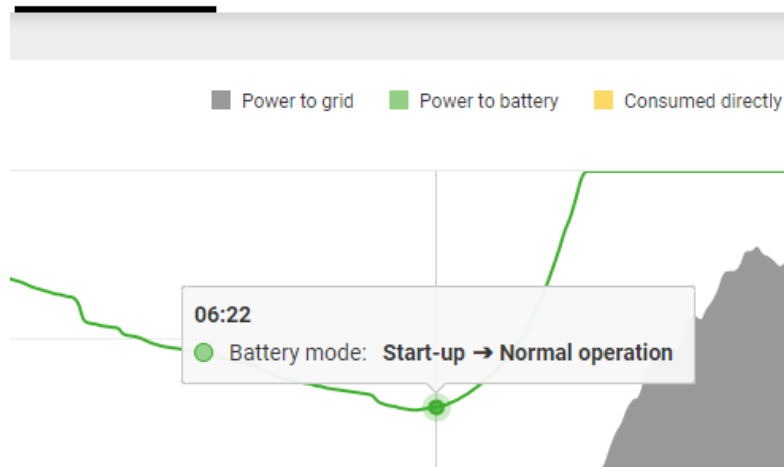
#### Basic Storage Controls Register

- Startadresse:
- bei Einstellung „float“: **40313**
  - bei Einstellung „int+SF“: **40303**

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link <http://www.fronius.com/QR-link/0024> abrufbar.

#### Register- und Batteriestatusänderungen in Fronius Solar.web

Mit Fronius Solar.web können Statusänderungen der Batterie visualisiert werden. Diese Änderungen können in Fronius Solar.web unter der Option Energiebilanz und dann Erzeugung oder Verbrauch eingesehen werden. Die Änderungen sind mit einem Blasenstatus gekennzeichnet, ein Klick auf eine Zustandsänderung zeigt den vorherigen Zustand gefolgt von einem Pfeil und dem neuen Zustand.



Änderung des Batteriezustandes von Inbetriebnahme zum Normalbetrieb

Batteriezustandsänderungen werden während des normalen Betriebs ausgelöst (wenn die Batterie betriebsbereit ist, aus Sicherheitsgründen...) oder durch Änderungen der Modbus-Register MinRsvPct, InWRte, OutWRte und StorCtl\_Mod.

Die Änderungen können wie folgt ausgelöst werden:

- Über das Register MinRsvPct wird ein minimaler Ladezustand eingestellt, der entsprechende Zustandswechsel ist der „Energiesparmodus“.
- Durch Setzen der Register InWRte, OutWRte, StorCtl\_Mod könnte der Batteriestatus auf „Erzwungene Nachladung“ wechseln.

# Meter Model

## Meter Model Register

Die Daten eines per Modbus RTU mit dem Fronius Datamanager verbundenen Energiezählers können per Modbus TCP über die entsprechenden SunSpec Models ausgelesen werden.

Ähnlich wie bei den Inverter Models gibt es auch hier zwei verschiedene SunSpec Models:

- das Meter Model mit Gleitkommadarstellung (Einstellung „float“; 211, 212 oder 213)
- das Meter Model mit ganzen Zahlen und Skalierungsfaktoren (Einstellung „int+SF“; 201, 202 oder 203)

Die Registeranzahl der beiden Model-Typen ist unterschiedlich!

Die Modbus Geräte-ID des Energiezählers ist konfigurierbar (default = 200).

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.

Es gibt 4 verschiedene Meter-Locations, diese werden durch die Locationnummer beschrieben (siehe Tabelle). Je nachdem, wo sich der Smart Meter befindet und ob der Wechselrichter produziert oder konsumiert, ändern sich die Vorzeichen der PowerReal-Werte und der Energy-Werte. Diese werden in der folgenden Tabelle aufgezeigt:

Meter_Location	0 (grid)	1 (load)	3 (ext. generator)	256-511 (subload)
PowerReal_P_Sum (+ positive)	Konsumieren vom Netz	Power produzieren	Generieren	Load produziert Power
PowerReal_P_Sum (- negative)	Einspeisung ins Netz	Normaler Konsum	Konsumieren	Normaler Konsum
Energy plus (absolute counter)	Importieren vom Netz = Energie konsumieren	Power produzieren* = Energie produzieren	Generieren = Energie produzieren	Power produzieren* = Energie produzieren
Energy minus (absolute counter)	Exportieren ins Netz = Energie produzieren	Konsumieren = Energie konsumieren	Konsumieren = Energie konsumieren	Konsumieren = Energie konsumieren

*\*Tritt nur bei Ausnahmen auf. Zum Beispiel bei einem versteckten Generator.*

# End Block

---

## Allgemeines

Zwei Register nach dem letzten Datenmodell zeigen an, dass keine weiteren Sun-Spec-Modelle mehr folgen.

Die Adressen dieser beiden Register sind je nach Gerätetyp (Wechselrichter, String Control, Energiezähler) und ausgewähltem Datentyp („float“ oder „int+SF“) verschieden.

- Wechselrichter:
  - - Startadresse für bei Einstellung „float“: 40313
  - - Startadresse bei Einstellung „int+SF“: 40303
- String Control:
  - - Startadresse: 40127
- Energiezähler:
  - - Startadresse für bei Einstellung „float“: 40195
  - - Startadresse bei Einstellung „int+SF“: 40176

---

## End Block

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link

<http://www.fronius.com/QR-link/0024> abrufbar.



# Table of contents

The Modbus Protocol.....	41
General.....	41
Structure of Modbus Messages.....	41
Modbus TCP – MBAP Header.....	42
Supported function codes.....	42
03 (0x03) Read Holding Registers.....	43
06 (0x06) Write Single Register.....	43
16 (0x10) Write Multiple Registers.....	44
Exception codes.....	44
CRC Calculation for Modbus RTU.....	45
Calculating CRC Checksum.....	46
Adding CRC Checksum to the Message.....	47
General.....	48
Abbreviations Used.....	48
Register maps.....	48
Response times.....	49
Modbus device ID for inverters.....	49
Modbus device ID for energy meters.....	49
Register addresses.....	49
Unavailable data records.....	52
Time response of the supported operating modes.....	52
Sign Convention for the Power Factor.....	53
Values saved on the card.....	54
Scale factors.....	54
Non-writable registers.....	55
Entering Invalid Values.....	55
Modbus Settings.....	56
General.....	56
Open the Modbus settings.....	56
Modbus.....	56
Limiting control.....	57
Common & Inverter Model.....	58
Common Block Register.....	58
Inverter Model Register.....	58
SunSpec Operating Codes.....	58
Nameplate Model (120).....	59
General.....	59
Nameplate Register.....	59
Basic Settings Model (121).....	60
Basic Settings Register.....	60
Reference Voltage.....	60
Deviation from reference voltage.....	60
Extended Measurements & Status Model (122).....	61
General.....	61
Extended Measurements & Status Register.....	61
Immediate Control Model (123).....	62
General.....	62
Immediate Controls Register.....	62
Standby.....	62
Power reduction.....	62
Example: setting a power reduction.....	63
Example: Changing the Return Time When Power Reduction Has Been Activated.....	63
Effects of reactive power specifications on effective power.....	64
Constant power factor.....	65
Example: Setting a Constant Power Factor.....	65
Constant relative reactive power.....	66
Example: Setting Constant Reactive Power.....	66
Multiple MPPT Inverter Extension Model (160).....	67
General.....	67
Multiple MPPT Inverter Extension Register.....	67

Basic Storage Control Model (124).....	68
General.....	68
Information Provided .....	68
Power Window Specifications .....	68
Setting the Minimum Charge Level.....	70
Charging the energy storage via the grid.....	70
Basic Storage Controls Register .....	70
Register manipulation and Battery status changes in Fronius Solar.web.....	70
Meter Model.....	72
Meter Model Register .....	72
End Block.....	73
General.....	73
End Block.....	73



# The Modbus Protocol

---

## General

The description of the protocol is largely taken from the Modbus specifications, which are publicly available at [www.modbus.org/specs.php](http://www.modbus.org/specs.php).

Modbus is a simple, open communication protocol, with which master-slave or client-server communication can be carried out between the devices connected to the network. The basic principle of Modbus is: a master sends a request and a slave responds to this. In Modbus TCP, the master is referred to as the client and a slave as a server. The function is the same. The descriptions of the protocol functions provided below will use the more common names master and slave, irrespective of the RTU and TCP variants. In cases where there are differences between RTU and TCP, this will be specifically indicated.

Modbus can be used in two ways on the inverter:

- Modbus TCP  
using TCP/IP via Ethernet (connected by cable or via WLAN)
- Modbus RTU  
using asynchronous serial transmission via RS-485 (EIA/TIA-485-A)

In the case of Modbus RTU, there can only ever be one master in the system. In principle, only one master may initiate requests. A slave may only give a response if it has been addressed by the master; the slaves cannot communicate with each other. If a broadcast request (request to all available slaves via slave ID or unit ID 0) is sent, none of the slaves can respond. Broadcasts can therefore only be used for write commands.

If a master sends a request to a slave, then it expects a response. In the event of a request from a master, there are five options:

- If the slave receives the request without communication errors and can process this request without errors, then a normal response will be sent with the required data.
- If the slave does not receive the request due to a communication error, then no response is sent. This leads to a timeout on the master.
- If the slave receives the request, but discovers a communication error (parity, CRC, etc.), then no response is sent. This leads to a timeout on the master.
- If the slave receives the request without communication errors, but cannot process it without errors (e.g., if a register that is not available needs to be read), then an error message (exception response) is returned with the reason for the error.
- If the slave receives a broadcast request, which also goes to all other devices, then no response will be sent either in the event of an error or if the request has been successfully processed. Broadcast requests are therefore only suitable for write commands.

Modbus devices provide data in 16 bit large data blocks (registers).

In certain cases, individual data points may also cover several data blocks (e.g., 2 registers = 32 bit value).

---

## Structure of Modbus Messages

In principle, a Modbus message is made up of the protocol data unit (PDU). This is independent of the underlying communication layers.

Depending on the bus or network that is used, additional fields can also be added. This structure is then referred to as the application data unit (ADU).

ADU

Address field	Function code	Data	CRC
---------------	---------------	------	-----

PDU

Structure of a Modbus message for Modbus RTU

ADU

MBAP header	Function code	Data
-------------	---------------	------

PDU

Structure of a Modbus message for Modbus TCP

Modbus TCP uses its own header to identify the application data unit. This header is called MBAP header (MODBUS application protocol header).

The size of the protocol data unit (PDU) is limited due to the first Modbus implementations in a serial network (max. RS-485 ADU = 256 bytes). This results in the following for the size of the protocol data unit PDU: PDU = 256 – slave ID (1 byte) – CRC (2 bytes) = 253 bytes

This results in:

- Modbus RTU ADU = 253 + slave ID (1 byte) + CRC (2 bytes) = 256 bytes
- Modbus TCP ADU = 253 bytes + MBAP (7 bytes) = 260 bytes

### Modbus TCP – MBAP Header

The MBAP header includes 7 bytes:

- **Transaction ID** (2 bytes): Is used in order to synchronize request and response. The slave adopts the transaction ID from the request into the response.
- **Protocol ID** (2 bytes): Is always 0 (Modbus protocol).
- **Length** (2 bytes): The length field includes the number of bytes of the subsequent fields, including unit ID and data fields.
- **Unit ID** (1 byte): This field is used for addressing devices connected to the inverter (gateway function). The unit ID corresponds to the slave ID in Modbus RTU. The value is specified by the master and is returned unchanged by the slave with the response.

For details about the addressing of the devices, see:

- [Modbus device ID for inverters](#) on page 49
- [Modbus device ID for energy meters](#) on page 49

**IMPORTANT!** The correct unit ID must always be specified, even if the Fronius Datamanager is only connected to one individual inverter.

### Supported function codes

The function code determines the action to be carried out on the slave. Three function codes for read and write operations are supported:

- 03 (0x03)<sup>1)</sup> read holding registers
- 06 (0x06)<sup>1)</sup> write single register
- 16 (0x10)<sup>1)</sup> write multiple registers

If an error occurs on the slave during the processing of a request, an error message is sent as the response (exception response). In the event of this kind of response, the most significant bit of the function code is set to 1 (corresponds to adding 0x80 to the function code)<sup>1)</sup> and an exception code is added, which indicates the reason for the error.

<sup>1)</sup> The prefix "0x" stands for hexadecimal numbers.

### 03 (0x03) Read Holding Registers

This function code is used to read the content of one or more successive registers of a device. The request contains the address of the first register to be read and the number of registers to be read. Registers are addressed in the request starting at 0. This means that registers 1 to 16 will be addressed via addresses 0 to 15.

#### Request

Function code	1 byte	0x03
Start address	2 bytes	0x0000 to 0xFFFF (0 to 65535)
Number of registers	2 bytes	1 to 125

#### Response

Function code	1 byte	0x03
Number of bytes	1 byte	2 x N*
Register values	N* x 2 bytes	

\*N = number of registers

#### Error

Error code	1 byte	0x83
Exception code	1 byte	01 or 02 or 03 or 04 or 11

### 06 (0x06) Write Single Register

This function code is used in order to write a single register. The request only contains the address of the register to be written. Registers are addressed starting at 0. This means that register 1 is addressed via address 0. The normal response is a copy of the request, which is sent after successfully writing the register.

#### Request

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF (0 to 65535)
Register value	2 bytes	

#### Response

Function code	1 byte	0x06
Register address	2 bytes	0x0000 to 0xFFFF (0 to 65535)
Register value	2 bytes	

#### Error

Error code	1 byte	0x86
Exception code	1 byte	01 or 02 or 03 or 04 or 11

## 16 (0x10) Write Multiple Registers

This function code is used in order to write a block of successive registers. The request contains the address of the first register to be written, the number of registers to be written, the number of bytes to be written, and the values to be written (2 bytes per register). The normal response contains the function code, the start address, and the number of registers written.

### Request

Function code	1 byte	0x10
Start address	2 bytes	0x0000 to 0xFFFF (0 to 65535)
Number of registers	2 bytes	1 to 123
Number of bytes	1 byte	2 x N*
Register values	N* x 2 bytes	

\*N = number of registers

### Response

Function code	1 byte	0x10
Start address	2 bytes	0x0000 to 0xFFFF (0 to 65535)
Number of registers	2 bytes	1 to 123

### Error

Error code	1 byte	0x90
Exception code	1 byte	01 or 02 or 03 or 04 or 11

## Exception codes

An error message (exception response) has two fields, which distinguishes it from a normal response:

- **Function code field**  
In a normal response, the function code of the request is adopted into the function code field of the response. In all function codes, the most significant bit (MSB) is 0 (the values of the function codes are all lower than 0x80). In an error message, the MSB is set to 1. This means that 0x80 is added to the value for the function code. The master can identify the response as an error message due to the set MSB.
- **Data field**  
A normal response contains data or statistical values in the data field. In an error message, an exception code is returned in the data field. This exception code indicates the reason for the error message.

Modbus exception codes		
Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code in the request is not supported by the slave.
02	ILLEGAL DATA ADDRESS	Invalid register addresses have been requested.

Modbus exception codes		
Code	Name	Meaning
03	ILLEGAL DATA VALUE	A value in the request is outside of the valid range. This applies both for the fields of a request (e.g., invalid number of registers) and for invalid setting values for the Sun-Spec inverter control models.
04	SLAVE DEVICE FAILURE	An error occurred during an attempt to write one or more registers.
11	GATEWAY TARGET DEVICE FAILED TO RESPOND	The addressed device is switched off and cannot be found.

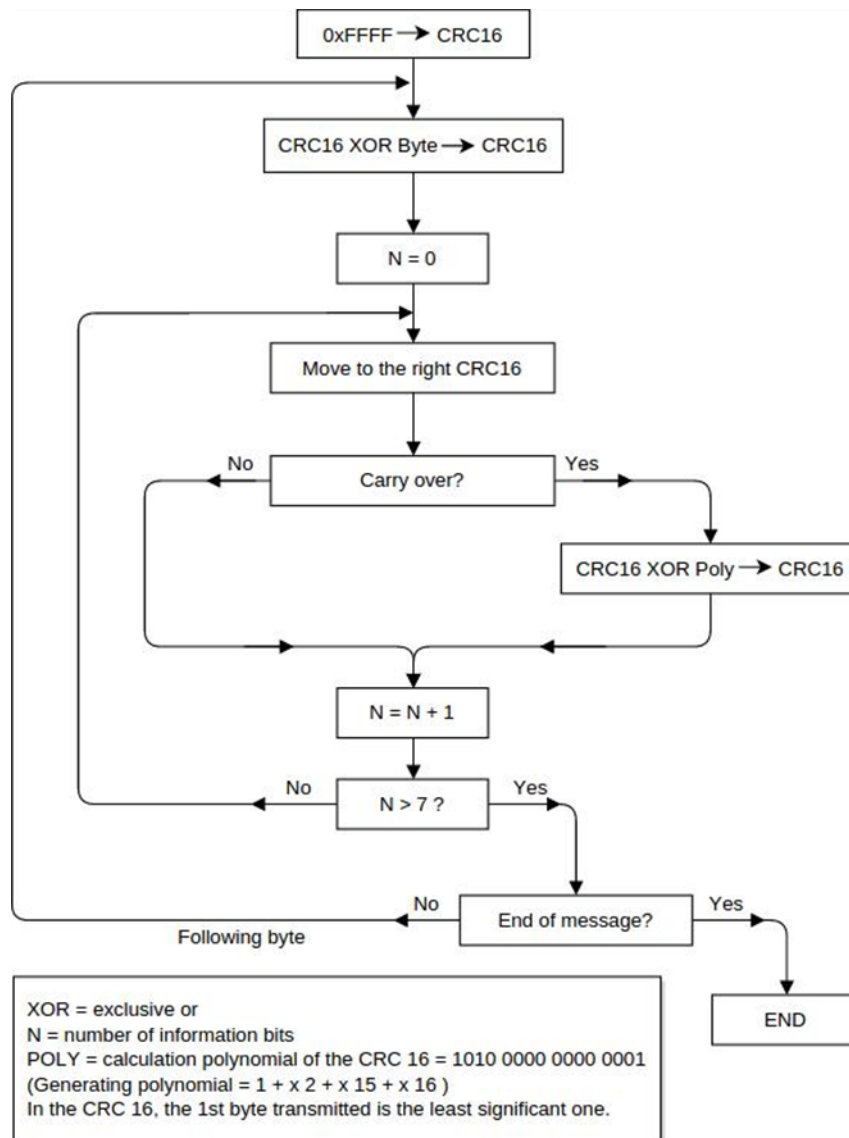
### CRC Calculation for Modbus RTU

Each Modbus RTU message is equipped with a checksum (CRC, Cyclic Redundancy Check) in order to be able to identify transmission errors. The size of the checksum is 2 bytes. It is calculated by the sending device and attached to the message to be sent. For its part, the receiver calculates the checksum from all bytes of the received message (without CRC) and compares this with the received checksum. If these two checksums are different, then an error has occurred.

The calculation of the checksum starts with setting all bits of a 16 bit register (CRC register) to 1 (0xFFFF). All bytes of the message are then individually processed with the CRC register. Only the data bytes of one message are used for the calculation. Start, stop, and parity bits are not considered.

During the calculation of the CRC, each byte is XOR-linked with the CRC register. The result is then moved in the direction of the least significant bit (LSB) and the most significant bit (MSB) is set to 0. The LSB is considered. If the LSB was previously 1, then the CRC register is XOR-linked with a fixed assigned value. If the LSB was 0, then nothing needs to be done.

This process is repeated until the CRC register has been moved eight times. After the last (eighth) movement, the next byte is taken and XOR-linked to the current CRC register. The write process then starts from the beginning; it is again moved eight times. After dealing with all bytes of the message, the value of the CRC register is the checksum.



Calculation algorithm of the CRC16

## Calculating CRC Checksum

- 1 Initialize a 16 bit register (2 bytes) with 0xFFFF. This register is referred to as the CRC16 register.
- 2 XOR-link the first byte of the message with the less significant byte of the CRC16 register. The result is saved in the CRC16 register.
- 3 Move the CRC16 register 1 bit to the right (in the direction of the LSB), fill MSB with 0. Look at LSB.
- 4 Check LSB value
  - If the LSB was 0: Go to step 3 (move again).
  - If the LSB was 1: XOR-link the CRC16 register with the CRC polynomial 0xA001 (1010 0000 0000 0001).
- 5 Repeat steps 3 and 4 until eight movement operations have been carried out. When these have been carried out, a complete byte of the message will have been processed.
- 6 Repeat steps 3 to 5 for the next byte of the message. Repeat everything until all bytes of the message have been processed.
- 7 After the last byte, the CRC16 register contains the checksum.
- 8 When the checksum is added to the message to be sent, then the two bytes must be inverted as described below.

---

**Adding CRC  
Checksum to the  
Message**

If the 16 bit (2 bytes) CRC checksum is sent with a message, then the less significant byte is transferred before the more significant one.

For example, if the CRC checksum is 0x1241 (0001 0010 0100 0001):

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi
							0x41	0x12

# General

## Abbreviations Used

AC	Alternating current
DC	Direct current
FW	Firmware
PF	Power factor (cos j)
PV	Photovoltaics
RTC	Real-time clock
SF	Scale factor
SW	Software
V	Voltage (volts)
VA	Apparent power
VAr	Reactive power
VMax	Maximum voltage
VMin	Minimum voltage
VRef	Reference voltage
W	Power (watts)
IN	Inverter

## Register maps

### Inverter

<b>SID</b> Identification as a SunSpec device
<b>Common Block</b> Device information
<b>Inverter Model</b> Inverter data
<b>Nameplate Model</b>
<b>Basic Settings Model</b>
<b>Ext. Measurement Model</b>
<b>Immediate Controls Model</b>
<b>Multi. MPPT Inv. Ext. Model</b>
<b>Basic Storage Control</b>
<b>End Block</b>

### Energy meter

<b>SID</b> Identification as a SunSpec device
<b>Common Block</b> Device information
<b>Meter Model</b> Energy meter data
<b>End Block</b>

The register lists can be downloaded from the Fronius website:

[https://www.fronius.com/de/downloads / Solar Energy / Modbus Sunspec Maps, State Codes and Events](https://www.fronius.com/de/downloads/Solar_Energy/Modbus_Sunspec_Maps_State_Codes_and_Events)



**Response times****Recommendation for timeout values**

Modbus requests should only be executed sequentially and not in parallel (maximum 2 queries in parallel). Perform the requests with a timeout of at least 1 second. Requests in millisecond intervals can lead to long response times. Multiple register requests in one message are faster than multiple requests of individual registers.

**Modbus device ID for inverters**

TCP: The unit-id of the inverter is always 0x01. Identification is possible by the IP address.

RTU: The slave-id must be configured on the web interface of the GEN24. Several GEN24 devices can be connected together. Each device must have a unique number.

**Modbus device ID for energy meters**

If an energy meter (e.g., Fronius Smart Meter 63A) is connected via Modbus RTU, it can be read out via the settable Modbus device ID using Modbus TCP.

<i>Fronius Smart Meter address</i>	<i>Modbus device ID</i>
1	200 (default, note: 240 on SnapINverter)
2	201
3	202
4	203
5	204

**Register addresses****IMPORTANT!**

- Register addresses do not remain constant.
- The actual register addresses depend on the composition of the dynamic SunSpec register list.

Correct procedure:

- Search for the model by making a request (determine start address)
- Then work with offsets

To read a register, the register's start address must be specified in the Modbus request.

SunSpec Basic Register: 40001

Registers begin at 1 and do not represent a function code.

Do not confuse the registers with the Modicon address scheme: in the Modicon address scheme, 40001 is displayed as 4x40001. To read register 40001, use address 40000 (0x9C40).

The register address that is output therefore always has 1 number less than the actual register number.

**IMPORTANT!**

**The lengths of individual models may vary due to the data types used.**

Start addresses are therefore specified for SunSpec models in the case of some register tables.

This start address, together with the offset from the table, then produces the value of the actual register number.

**Example:** Table **Nameplate Model (120)** on page 59:

the register *WRtg* of the Nameplate Model has an offset of 4. The start address is specified as 40131 with the setting "float".

Therefore, the correct register number is:  $40131 + 4 = 40135$ .

**Examples for Modbus RTU:**

**1. Request for 4 registers starting from register 40005 (Mn, Manufacturer)**

Send (bytes in hexadecimal)

01	03	9C 44	00 04	2A 4C
De-vice ID	Func-tion code	Address 40004 (cor-responds to register 40005)	Number of registers to be read	Checksum Low byte High byte

Receive (bytes in hexadecimal)

01	03	08 46 72	6F 6E	69 75	73 00	8A 2A
De-vice ID	Func-tion code	Num-ber of bytes Address 40005 "F" and "r"	Address 40006 "o" and "n"	Address 40007 "i" and "u"	Address 40008 "s" and 0	Checksum Low byte High byte

**2. Enter one register starting from register 40242 (WmaxLimPct)**

01	10	9D 32	00 01	02	13 88	E3 DD
De-vice ID	Func-tion code	Address 40242	Number of registers to be entered	Number of data bytes still to follow	Register value to be entered 0x1388 = 5000	Checks-um Low byte High byte

01	10	9D 32	00 01	8F AA
De-vice ID	Func-tion code	Address 40242	Number of registers entered	Checksums "i" and "u"  Low byte High byte

400  
08  
"s"  
and  
0

**Examples for Modbus TCP:**

**1. Request for 4 registers starting from register 40005 (Mn, Manufacturer)**

Send (bytes in hexadecimal)

MBAP header	03	9C 44	00 04
For details, see description of MBAP header	Function code	Address 40004 (corresponds to register 40005)	Number of registers to be read

Receive (bytes in hexadecimal)

MBAP header	03	08	46 72	6F 6E	69 75	73 00
For details, see description of MBAP header	Function code	Number of bytes	Address 40005 "F" and "r"	Address 40006 "o" and "n"	Address 40007 "i" and "u"	Address 40008 "s" and "o"

**2. Enter one register starting from register 40242 (WmaxLimPct)**

MBAP header	10	9D 32	00 01	02	13 88
For details, see description of MBAP header	Function code	Address 40242	Number of registers to be entered	Number of data bytes still to follow	Register value to be entered 0x1388 = 5000

MBAP header	10	9D 32	00 01
For details, see description of MBAP header	Function code	Address 40242	Number of registers entered

**Unavailable data records**

Fronius inverters cannot always provide all the data specified in the SunSpec data models. Depending on the data type, this data is represented by the following values in accordance with the SunSpec specification:

- int16 (-32767 to 32767): 0x8000<sup>1)</sup>
- uint16 (0 to 65534): 0xFFFF
- acc16 (0 to 65535): 0
- enum16 (0 to 65534): 0xFFFF
- bitfield16 (0 to 0x7FFF): 0xFFFF
- pad (0x8000): always 0x8000
- int32 (-2147483647 to 2147483647): 0x80000000
- uint32 (0 to 4294967294): 0xFFFFFFFF
- acc32 (0 to 4294967295): 0
- enum32 (0 to 4294967294): 0xFFFFFFFF
- bitfield32 (0 to 0x7FFFFFFF): 0xFFFFFFFF
- int64 (-9223372036854775807 to 9223372036854775807): 0x8000000000000000
- uint64 (0 to 18446744073709551615): 0xFFFFFFFFFFFFFFFF
- acc64 (0 to 18446744073709551615): FFF
- stringX: 0
- float32 (range see IEEE 754): all X registers filled with 0x0000
- sunssf (scale factors; -10 to 10): 0x7FC00000 (NaN)
- 0x8000

<sup>1)</sup> The prefix "0x" stands for hexadecimal numbers.

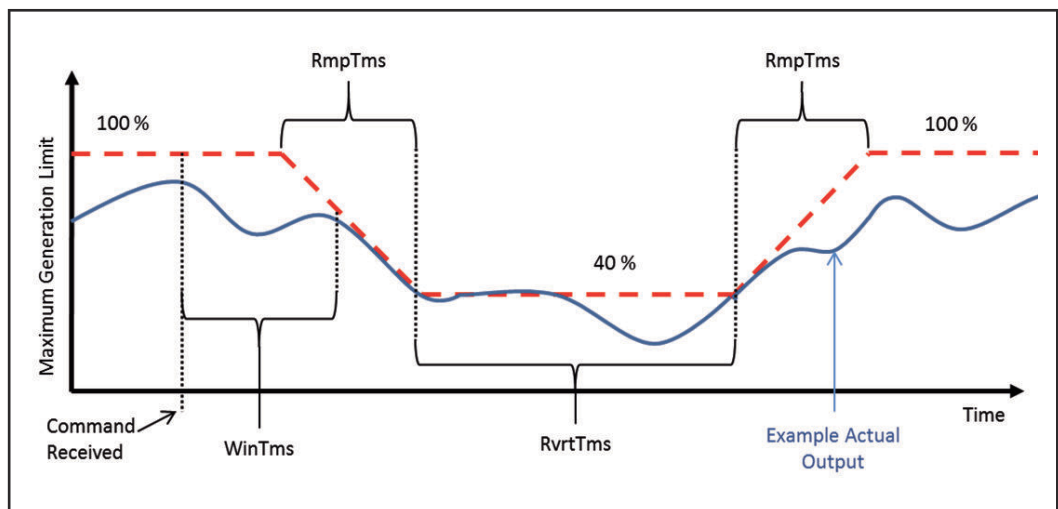
**NOTE!**

**Data points that are not supported are marked with "Not supported" in the "Range of values" column in the register tables.**

In this case, during reading, the corresponding value from the list above is obtained depending on the data type.

In certain instances, registers which are basically listed as supported may also return this value. This is because some values depend on the device type, e.g., currents AphB and AphC in the case of a single-phase inverter.

**Time response of the supported operating modes**



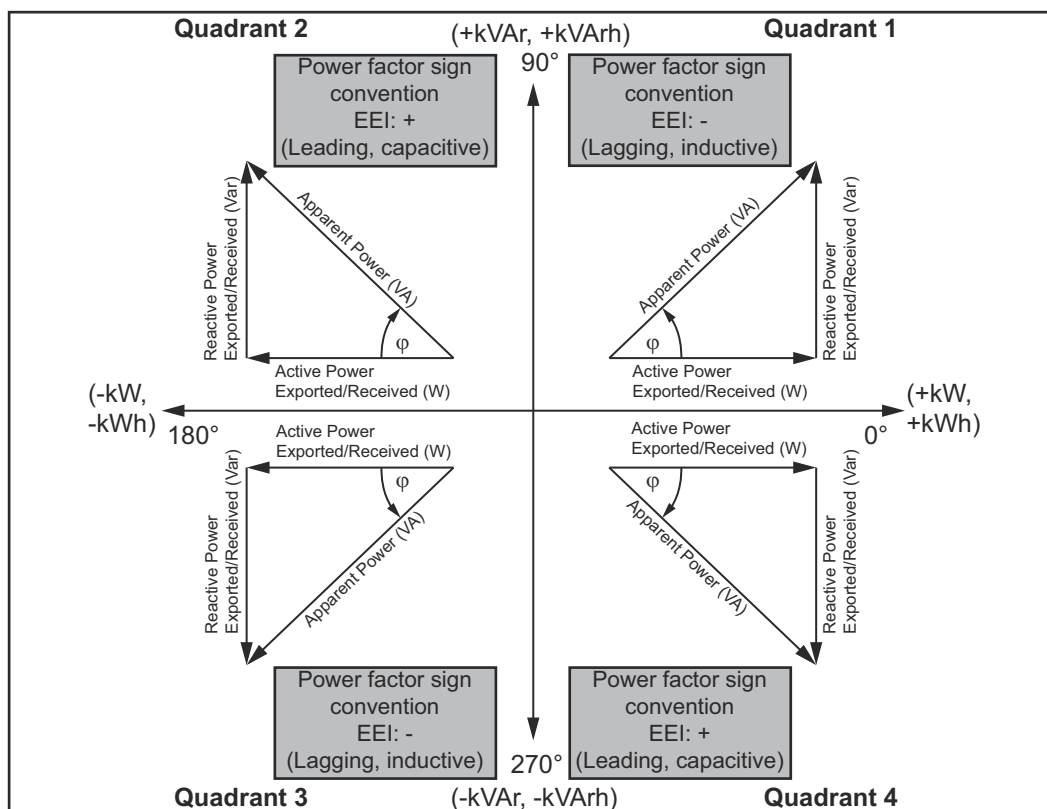
Time response illustrated by power reduction

The inverter's time response in an operating mode can be defined by several time values.

Three possible time values are shown in the figure "Time response illustrated by power reduction":

- **WinTms 0–300 [seconds]**  
 Specifies a time window in which the operating mode is randomly started. The time window starts when the start command for the operating mode is issued (e.g., *OutPFSet\_Ena* = 1). *WinTms* can be used to prevent all the inverters in the system from applying the changes at the same time. If the time window is set to 0 (the default value), the operating mode will start immediately.
- **RvrtTms 0–28800 [seconds]**  
 Determines how long the operating mode will remain active. The timer is restarted with every Modbus message received. If no new Modbus message was received during the fallback time (= *RvrtTms*), the operating mode is automatically ended and the operating mode with the next highest priority becomes active, e.g., dynamic power reduction. If *RvrtTms* is 0 (the default value), the operating mode remains active until it is manually deactivated via the corresponding register. In this instance the fallback option is not available.
- **RmpTms**  
 Specifies how quickly the changes are to be made. The corresponding value gradually changes during the specified time period from the old to the new value. If *RmpTms* is 0 (the default value) or if this value is not supported, the new value will be valid immediately.

**Sign Convention for the Power Factor**



The EEI sign convention<sup>1)</sup> for the power factor is in line with the SunSpec specification and is based on the information contained in the "Handbook for Electricity Metering" and IEC 61557-12 (2007).

The power factor is:

- negative if the reactive power is positive (over-excited, quadrant 1)
- positive if the reactive power is negative (under-excited, quadrant 4)

1) EEI = Edison Electrical Institute

---

**Values saved on the card**

Nameplate Model (IC120):

- **WRtg**  
AC nominal output of inverter
- **VARtg**  
AC nominal apparent output of inverter.  
Default value = WRtg
- **VArRtgQ1**  
Maximum AC reactive power in the first quadrant (over-excited).  
Default value is calculated based on the available cos Phi (0.85) and the nominal apparent power. Note the scale factor VArRtg\_SF
- **VArRtgQ4**  
Maximum AC reactive power in the fourth quadrant (under-excited).  
Default value is calculated based on the available cos Phi (0.85) and the nominal apparent power. Note the scale factor VArRtg\_SF
- **ARtg**  
AC nominal current of inverter

Basic Settings Model (IC121):

- **WMax**  
Maximum AC power  
Default value = WRtg
- **VRef**  
Reference voltage at the feed-in point
- **VRefOfs**  
Deviation from reference voltage
- **VMax**  
Maximum AC voltage
- **VMin**  
Minimum AC voltage
- **VAMax**  
Maximum AC apparent power  
Default value = VARtg

---

**Scale factors**

**IMPORTANT!** Scale factors (also possible when selecting "Float!") are not static, even if they are entered as a fixed value in these Operating Instructions. Scale factors can change every time the firmware is changed and also change with the runtime (auto-scale) (e.g., scale factor for power specification).

Scale factors with constant values are listed in the tables in the column "Range of values".

Current data (data of inverters and energy meters) may have variable scale factors. These must be read from the corresponding registers.

The data type "sunssf" is a signed integer with 16 bits.

Example calculation:

(Model 160): 1\_DCW = 10000, DCW\_SF = -1 -> Power = 10000 x 10<sup>(-1)</sup> = 1000 W

---

**Non-writable registers**

- The following registers cannot be written:
- Read-only (R) registers
  - Registers which are currently not supported

**NOTE!**

**If an attempt is made to write to such registers, the inverter does not return an exception code!**

**The values written to these registers are ignored without an error message.**

**In Model 123 and 124, an exception occurs during write access if the control option in the local web interface has been deactivated.**

---

**Entering Invalid Values**

Some registers only permit certain values. The valid values can be found in the relevant register table.

If an invalid value is entered into a register, the Fronius Datamanager will return exception code 3 (illegal data value). The invalid value is ignored.

If several registers are written at the same time, all the valid values will be entered up to the register containing the invalid value. The write operation will then be canceled.

# Modbus Settings

---

## General

From your web browser, you can use the inverter web interface to apply the Modbus connection settings which cannot be accessed via the Modbus protocol.

---

## Open the Modbus settings

- 1 Open the web interface of the inverter
  - 2 Select the "Communication" section (1)
  - 3 Open the "Modbus" menu item (2)
- 

## Modbus

### Modbus RTU interface 0 / 1

If one of the two Modbus RTU interfaces is set to slave, the following input fields are available:

---

#### "Baud rate"

The baud rate influences the speed of the transmission between the individual components connected in the system. When selecting the baud rate, it should be ensured that this is the same on the transmit and receive side.

---

#### "Parity"

The parity bit can be used for parity checks. This is used to identify transmission errors. In this case, a parity bit can ensure a specified number of bits. The value (0 or 1) of the parity bit must be calculated at the transmitter, and is checked at the receiver using the same calculation. The calculation of the parity bit can be carried out for even or odd parity.

---

#### "SunSpec Model Type"

There are two different settings, depending on the SunSpec model.

**float:** SunSpec Inverter Model 111, 112, 113 or 211, 212, 213.

**int + SF:** SunSpec Inverter Model 101, 102, 103 or 201, 202, 203.

---

#### "Meter address"

The value entered is the identification number (Unit ID) assigned to the meter. Can be found on the inverter's website in the Communication - Modbus menu. Factory setting: 200

---

#### "Inverter address"

The value entered is the identification number (Unit ID) assigned to the inverter. Can be found on the inverter's website in the Communication - Modbus menu. Factory setting: 1

---

### Slave as Modbus TCP

If the function "Slave as Modbus TCP" is activated, the following input fields are available:

---

#### "Modbus-Port"

Number of the TCP port to be used for Modbus communication.

---

#### "SunSpec Model Type"

There are two different settings, depending on the SunSpec model.

**float:** SunSpec Inverter Model 111, 112, 113 or 211, 212, 213.

**int + SF:** SunSpec Inverter Model 101, 102, 103 or 201, 202, 203.

---



**"Meter address"**

The value entered is the identification number (Unit ID) assigned to the meter. Can be found on the inverter's website in the Communication - Modbus menu. Factory setting: 200

---

**"Inverter address"**

The value entered is the identification number (Unit ID) assigned to the inverter. Can be found on the inverter's website in the Communication - Modbus menu. Factory setting: This value is invariably defined as 1.

**Inverter control via Modbus**

If this option is activated, the inverter is controlled via Modbus.

Inverter control includes the following functions:

- On/off
  - Power reduction
  - Setting a constant power factor (cos phi)
  - Setting a constant reactive power
  - Battery control specifications with battery
- 

**Limiting control**

The "Limit Control" option is only available for the TCP transmission protocols. It is used to block inverter control commands from unauthorized users by only permitting control for specific devices.

**Limit Control**

If this option is activated, only certain devices will be able to send control commands.

**IP address**

To limit inverter control to one or more devices, enter the IP addresses of the devices which are permitted to send commands to the inverter in this field. Multiple entries are separated by commas.

Examples:

- One IP address: **98.7.65.4**
  - Control only permitted by IP address 98.7.65.4
- Several IP addresses: **98.7.65.4,222.44.33.1**
  - Control only permitted by IP addresses 98.7.65.4 and 222.44.33.1
- IP address range, e.g., from 98.7.65.1 to 98.7.65.254 (CIDR notation): **98.7.65.0/24**
  - Control only permitted by IP addresses 98.7.65.1 to 98.7.65.254

# Common & Inverter Model

---

## Common Block Register

The description of the Common Block including the SID register (register 40001–40002) for identification as a SunSpec device applies for each device type (inverter, energy meter). Each device has its own Common Block, which lists information about the device (model, serial number, SW version, etc.).

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

---

## Inverter Model Register

Two different SunSpec models are supported for the inverter data:

- the default set inverter model with floating point display (setting "float"; 111, 112 or 113)
- the inverter model with integers and scale factors (setting "int+SF"; 101, 102 or 103)

The register number of the two model types is different!

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

---

## SunSpec Operating Codes

Name	Value	Description
I_STATUS_OFF	1	Inverter is off
I_STATUS_SLEEPING	2	Auto shutdown
I_STATUS_STARTING	3	Inverter starting
I_STATUS_MPPT	4	Inverter working normally
I_STATUS_THROTTLED	5	Power reduction active
I_STATUS_SHUTTING_DOWN	6	Inverter shutting down
I_STATUS_FAULT	7	One or more faults present, see St*or Evt* register
I_STATUS_STANDBY	8	Standby

\* Inverter model register

# Nameplate Model (120)

---

## General

This model corresponds to a rating plate. The following data can be read:

- ***DERType (3)***  
Type of device. The register returns the value 4 (PV device).
- ***WRtg (4)***  
Nominal power of inverter.
- ***VARtg (6)***  
Nominal apparent power of inverter.
- ***VArRtgQ1 (8) – VArRtgQ4 (11)***  
Nominal reactive power values for the four quadrants.
- ***ARtg (13)***  
Nominal current of inverter.
- ***PFRtgQ1 (15) – PFRtgQ4 (18)***  
Minimal power factor values for the four quadrants.

---

## Nameplate Register

Start address:

- for "float" setting: ***40131***
- for "int+SF" setting: ***40121***

The register tables can be found on the Fronius website or opened using the link:  
<http://www.fronius.com/QR-link/0024> .

# Basic Settings Model (121)

## Basic Settings Register

- Start address:
- for "float" setting: **40159**
  - for "int+SF" setting: **40149**

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

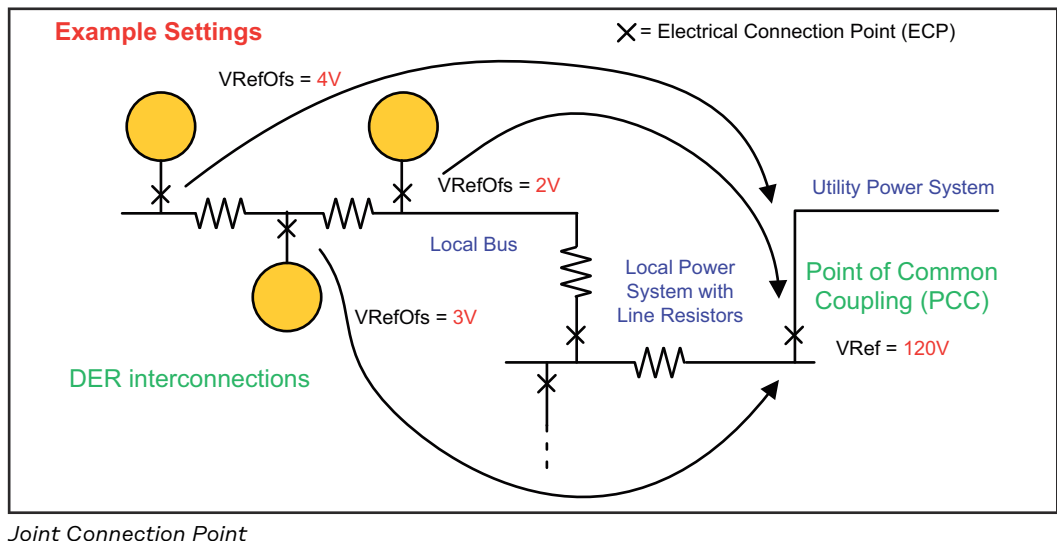
## Reference Voltage

### *VRef (4)*

The reference voltage is the voltage at the joint connection point where the local grid is connected to the public grid. The reference voltage is the same as the inverter's nominal voltage.

=> See figure "Joint Connection Point."

The value is given in volts in the range of 0 (0x0000) to 400 (0x0190).



## Deviation from reference voltage

### *VRefOfs (5)*

Depending on the wiring of the local grid, there may be a deviation from the reference voltage at the point where each individual inverter is connected to the local grid (see "Joint connection point" diagram).

# Extended Measurements & Status Model (122)

## General

This model provides some additional measurement and status values which the normal inverter model does not cover:

- **PVConn (3)**  
This bit field displays the inverter's status
  - Bit 0: Connected
  - Bit 1: Responsive
  - Bit 2: Operating (inverter feeds energy in)
- **ECPConn (5)**  
This register displays the status of connection to the grid
  - ECPConn = 1: inverter is currently feeding power into the grid
  - ECPConn = 0: inverter is not feeding power into the grid
- **ActWH (6–9)**  
Active energy meter
- **StActCtl (36–37)**  
Bit field for currently active inverter modes
  - Bit 0: power reduction (FixedW; corresponds to WMaxLimPct specification)
  - Bit 1: constant reactive power specification (FixedVAR; corresponds to VArMaxPct)
  - Bit 2: specification of a constant power factor (FixedPF; corresponds to OutPFSet)
- **TmSrc (38–41)**  
Source for the time synchronization, the register returns the string "RTC"
- **Tms (42–43)**  
Current time and date of the RTC  
The seconds are specified from January 1, 2000 00:00 (UTC) to the current time.
- **Ris**  
Iso Resistance

## Extended Measurements & Status Register

Start address:

- for "float" setting: **40191**
- for "int+SF" setting: **40181**

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

# Immediate Control Model (123)

---

- General**            The immediate controls can be used to make the following settings on the inverter:
- deactivation of inverter's grid power feed operation (standby)
  - constant reduction of output power
  - specification of a constant power factor
  - specification of a constant relative reactive power

In the settings on the inverter's web interface, the setting "Inverter control via Modbus" must be enabled under Modbus for write functions to be possible. Depending on the control priority that has been set (IO control, dynamic power reduction, or control via Modbus), Modbus commands may not be accepted.

---

- Immediate Controls Register**    Start address:
- for "float" setting: **40237**
  - for "int+SF" setting: **40227**
- The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .
- 

- Standby**            **Conn\_WinTms (3) to Conn (5)**  
These registers are used to control the standby mode (no grid power feed operation) of the inverter.

**Conn\_WinTms (3) and Conn\_RvrtTms (4)**  
These registers can be used to control the inverter's time response. => See section "Time Response of the Supported Operating Modes".  
0 is set as the default for all registers.

**Conn (5)**  
Register *Conn* indicates whether or not the inverter is currently feeding power into the grid (0 = standby, 1 = grid power feed operation).

- In order to switch the inverter to standby, enter the value 0 into this register.
- In order to reactivate the inverter, enter the value 1 into this register.

**NOTE!**  
**To find out whether or not the inverter is feeding power into the grid, you can also use the ECPCnn register and check the extended measurements and status model.**

---

- Power reduction**    **WMaxLimPct (6) to WMaxLim\_Ena (10)**  
These registers can be used to set an output power reduction in the inverter.
- WMaxLimPct (6)**  
Values between 0% and 100% can be entered in register *WMaxLimPct*.  
The values limit the maximum possible output power of the device, and therefore do not necessarily have an effect on the current power.

**IMPORTANT!** Observe the scale factor for this register.

Further information can be found at:

<http://sunspec.org/wp-content/uploads/2015/06/SunSpec-Information-Models-12041.pdf>

#### ***WMaxLimPct\_WinTms (7), WMaxLimPct\_RvrtTms (8)***

These registers can be used to control the inverter's time response for this operating mode. => See section "Time response of the supported operating modes". 0 is set as the default for all registers.

#### ***WMaxLim\_Ena (10)***

Used to start and end this operating mode

- Enter value 1 into register *WMaxLim\_Ena* = start operating mode
- Enter value 0 into register *WMaxLim\_Ena* = end operating mode

#### **NOTE!**

#### **To change values in an active operating mode (e. g.**

to set a different power limit or a different return time), proceed as follows:

- ▶ Enter the new value into the relevant register
- ▶ Restart the operating mode using register *WMaxLim\_Ena* by setting a 1

#### **Example: setting a power reduction**

If you are working with function code 0x10 (write multiple registers), performance specifications can be used to achieve a higher level of performance. Instead of using two Modbus commands, it is now possible to preset both the power and enable at the same time with just one command. All 5 registers (*WMaxLimPct*, *WMaxLimPct\_WinTms*, *WMaxLimPct\_RvrtTms*, *WMaxLimPct\_RmpTms*, *WMaxLim\_Ena*) can be written with one command. Writing to the "Read Only" register *WMaxLimPct\_RmpTms* takes place without returning an otherwise usual exception (error) code.

For example, register values for 80% specification without timing specification: 8000, 0, 0, 0, 1

- 1** Enter the value for the output power reduction in register *WMaxLimPct* (e.g., 3000 for 30%).
- 2** As an option, you can set the start and return time using registers *WMaxLimPct\_WinTms* and *WMaxLimPct\_RvrtTms*.
- 3** Start the operating mode by entering 1 in register *WMaxLim\_Ena*.

**IMPORTANT!** Observe the scale factor for this register.

Further information can be found at:

<http://sunspec.org/wp-content/uploads/2015/06/SunSpec-Information-Models-12041.pdf>

#### **Example: Changing the Return Time When Power Re- duction Has Be- en Activated**

If the power reduction was originally started using *WMaxLimPct\_RvrtTms* = 0, the operating mode must be manually deactivated.

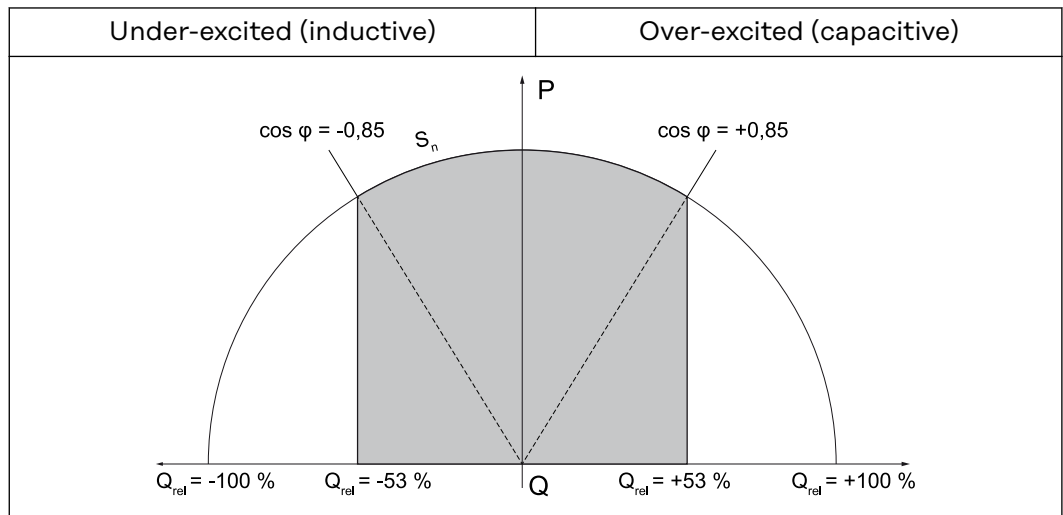
- 1** Set *WMaxLimPct\_RvrtTms* to 30, for example
- 2** Apply the change by entering 1 in register *WMaxLim\_Ena*
  - The operating mode is automatically deactivated after 30 seconds and the mode with the next highest priority becomes active (e.g., dynamic power reduction)

**Effects of reactive power specifications on effective power**

In principle, reactive power operation is limited by the maximum output current (the maximum apparent power) and by the operative reactive power limit of the inverter:

the following diagram shows the possible working range of the inverter. All valid operating points defined by effective power P and reactive power Q are within the gray area.

The maximum values must be read out from the Nameplate Model via registers VArRtgQ1 to VArRtgQ4 and VArRtg\_SF.



Reactive power and power factor

**Key:**

W	Power	VAr <sub>max</sub>	Nominal reactive power
W <sub>max</sub>	Nominal power	VAr <sub>rel</sub>	Relative reactive power (VAr/VAr <sub>max</sub> )
VAr	Reactive power		



**Constant power factor*****OutPFSet (11)* to *OutPFSet\_Ena (15)***

These registers can be used to set a constant power factor in the inverter.

***OutPFSet (11)***

- In register *OutPFSet* it is possible to enter both positive and negative values for the power factor.
- The values must be scaled up by the factor in register *OutPFSet\_SF*.
- The lowest possible values depend on the inverter type and can be found in the Nameplate Model.

**NOTE!**

**The power factor value must be entered with the correct sign, see section "Sign convention for the power factor"**

- ▶ positive for under-excited
- ▶ negative for over-excited.

***OutPFSet\_WinTms (12)*, *OutPFSet\_RvrtTms (13)***

These registers can be used to control the inverter's time response for this operating mode. => See section "Time response of the supported operating modes". 0 is set as the default for all registers.

***OutPFSet\_Ena (15)***

Used to start and end this operating mode

- Enter value 1 into register *OutPFSet\_Ena* = start operating mode
- Enter value 0 into register *OutPFSet\_Ena* = end operating mode.

**NOTE!**

**Proceed as follows to change values when an operating mode is active (e.g., when setting a different power factor or return time):**

- ▶ Enter the new value into the relevant register
- ▶ Restart the operating mode using register *OutPFSet\_Ena* by setting a 1.

***OutPFSet\_RmpTMS***

-

**Example:  
Setting a Constant Power Factor**

- 1 Enter the power factor value in register *OutPFSet* (e.g., 950 for 0.95).
- 2 As an option, you can set the start and return time using registers *OutPFSet\_WinTms* and *OutPFSet\_RvrtTms*.
- 3 Start the operating mode by entering 1 in register *OutPFSet\_Ena*.

---

**Constant relative reactive power**

***VArMaxPct (17) to VArPct\_Ena (23)***

These registers can be used to set on the inverter a constant value for the reactive power to be produced by the inverter.

***VArMaxPct (17)***

- Used to set a value for constant reactive power.
- The minimum and maximum limits depend on the type of inverter.

**NOTE!**

**In practical operation, the reactive power that is actually available is specified by the inverter's operating limits.**

For this reason, the reactive power specification can only be reached if enough effective power is fed into the grid.

If too little effective power is fed into the grid, the inverter will operate at its operating limit.

---

***VArPct\_WinTms (19), VArPct\_RvrtTms (20)***

These registers can be used to control the inverter's time response for this operating mode. => See section "Time response of the supported operating modes". 0 is set as the default for all registers.

***VArPct\_Mod (22)***

- This register cannot be changed.
- It returns the (currently) supported operating mode.  
Reactive power as a percentage of the maximum possible reactive power.

***VArPct\_Ena (23)***

Used to start and end this operating mode

- Enter value 1 into register *VArPct\_Ena* = start operating mode
- Enter value 0 into register *VArPct\_Ena* = end operating mode.

**NOTE!**

**To change values in an active operating mode (e. g.**

to set a different reactive power or a different return time), proceed as follows:

- ▶ Enter the new value into the relevant register
  - ▶ Restart the operating mode using register *VArPct\_Ena* by setting a 1.
- 

***VArPct\_RmpTms (23)***

- the relative approximation value in %/s
- 

**Example:  
Setting Constant  
Reactive Power**

- 1** Enter the relative reactive power value in register *VArMaxPct* (e.g., 80 for 80%).
- 2** As an option, you can set the start and return time using registers *VArPct\_WinTms* and *VArPct\_RvrtTms*.
- 3** Start the operating mode by entering 1 in register *VArPct\_Ena*.

# Multiple MPPT Inverter Extension Model (160)

---

## General

The Multiple MPPT Inverter Extension Model contains the values of the DC inverter inputs.

If the inverter has several DC inputs, then this is where the current, voltage, power, energy, and status codes for the individual inputs are listed. In the inverter model (101–103 or 111–113), only the full DC power of both inputs is output in this case. DC current and DC voltage are displayed as "not implemented".

The number of blocks is automatically adjusted based on the DC inputs. For devices with a storage solution, there are two additional blocks (charging (MPP3) and discharging (MPP4)). The register addresses are shifted in the following models (absolutely related to the register addresses).

---

## Multiple MPPT Inverter Extension Register

Start address:

- for "float" setting: **40263**
- for "int+SF" setting: **40253**

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

# Basic Storage Control Model (124)

---

## General

This model is only available for inverters with a storage solution.

The Basic Storage Control Model can be used to make the following settings on the inverter:

- Setting a power window within which the charge/discharge capacity of the energy storage may fluctuate.
- Setting a minimum charge level that the energy storage must not fall below.
- Permitting/preventing grid charging of the energy storage.

### NOTE!

**All specifications are to be considered recommendations. The inverter may deviate from the specifications if this is necessary for operational safety reasons.**

---

## Information Provided

The Basic Storage Control Model provides the following read-only information:

WChaMax

- If energy storage is available, this register feeds back the baseline value for the registers OutWRte and InWRt.  
 $WChaMax := \max(\text{MaxChaRte}, \text{MaxDisChaRte})$
- If energy storage is not available, the register feeds back a value of 0.

ChaState

- Energy storage charge level in %:  
 $\text{Estimated\_Capacity\_Remaining [Wh]} / \text{Estimated\_Capacity\_Maximum [Wh]}$

ChaSt

Energy storage operating status

- OFF: Energy storage is not available
- EMPTY: Energy storage is currently fully discharged
- DISCHARGING: Energy storage is in the process of being discharged
- CHARGING: Energy storage is in the process of being charged
- FULL: Energy storage is currently fully charged
- HOLDING: Energy storage is currently neither charged nor discharged
- TESTING: used during calibration or service charge

## Power Window Specifications

In the settings on the inverter's web interface, the setting "Inverter control via Modbus" must be enabled under Modbus for write functions to be possible. Depending on the control priority that has been set (IO control, dynamic power reduction, or control via Modbus), Modbus commands may not be accepted.

The following examples assume that WchaMax = 3300 W.

The following applies for the resulting power windows:

- Negative power values indicate that the energy storage is charging
- Positive values indicate that the energy storage is discharging

## NOTE!

**The values in the following examples must be scaled according to their scale factors in the specified scale registers after reading and before writing.**

Manipulating the registers InWRte, OutWRte and StorCtl\_Mod will generate changes in the battery status in Fronius Solar.web, ex: Forced Recharge and Energy saving mode, depending on user settings and current status of the battery.

---

### Example 1: Only permit energy storage charging

This behavior can be achieved by limiting the maximum discharge capacity to 0% => results in window [-3300 W, 0 W]

- OutWRte = 0% (set discharge limit of WchaMax to 0%)
- StorCtl\_Mod = 2 (activates discharge limit, bit pattern: 10)
- InWRte is not relevant in this case

### Example 2: Only permit energy storage discharging

This behavior can be achieved by limiting the maximum charge capacity to 0% => results in window [0 W, 3300 W]

- InWRte = 0% (set charge limit of WchaMax to 0%)
- StorCtl\_Mod = 1 (bit 1 activates charge limit, bit pattern: 01)
- OutWRte is not relevant in this case

### Example 3: Do not permit charging or discharging

This behavior can be achieved by limiting the maximum charge capacity to 0% and the maximum discharge capacity to 0%

=> results in window [0 W, 0 W]

- InWRte = 0% (set charge limit of WchaMax to 0%)
- OutWRte = 0% (set discharge limit of WchaMax to 0%)
- StorCtl\_Mod = 3 (activate both limit values, bit pattern: 11)

### Example 4: Charging and discharging with maximum 50% of the nominal power

This behavior can be achieved by limiting the maximum charge capacity to 50% and the maximum discharge capacity to 50%

=> results in window [-1650 W, 1650 W]

- InWRte = 50% (set charge limit of WchaMax to 50%)
- OutWRte = 50% (set discharge limit of WchaMax to 50%)
- StorCtl\_Mod = 3 (activate both limit values, bit pattern: 11)

### Example 5: Charging in the range of 50% to 75% of the nominal power

This behavior can be achieved by limiting the maximum charge capacity to 75% and the maximum discharge capacity to -50%

=> results in window [1650 W, 2475 W]

- InWRte = 75% (set charge limit of WchaMax to 75%)
- OutWRte = -50% (set discharge limit of WchaMax to -50%)
- StorCtl\_Mod = 3 (activate both limit values, bit pattern: 11)
- Battery status in Fronius Solar.web will change to Forced Recharge

### Example 6: Discharging with 50% of the nominal power

This behavior can be achieved by limiting the maximum charge capacity to -50% and the maximum discharge capacity to 50%

=> results in window [-1650 W, -1650 W]

- InWRte = -50% (set charge limit of WchaMax to -50%)
- OutWRte = 50% (set discharge limit of WchaMax to 50%)
- StorCtl\_Mod = 3 (activate both limit values, bit pattern: 11)

### Example 7: Charging with 50% to 100% of the nominal power

This behavior can be achieved by limiting the maximum discharge capacity to -50% => results in window [1650 W, 3300 W]

- OutWRte = -50% (set discharge limit of WchaMax to -50%)
- StorCtl\_Mod = 2 (activates discharge limit, bit pattern: 10)
- InWRte is not relevant in this case
- Battery status in Fronius Solar.web will change to Forced Recharge

---

#### Setting the Minimum Charge Level

By setting register MinRsvPct, a minimum state of charge of the energy storage can be set.

For example, by setting MinRsvPct to 20%, a reserve of 20% of the state of charge can be reserved that the state of charge should not fall below.

---

#### Charging the energy storage via the grid

The ChaGriSet register can be used to allow or prevent inverter storage charging via the grid. The register ChaGriSet and the field "battery charging from DNO grid" in the Fronius system monitoring settings are AND-linked (Device configuration - Components - Battery). If the behavior is to be controlled by the ChaGriSet flag, "battery charging from DNO grid" must be checked.

The battery can be woken from standby mode via the IC124 model. If the *SoCMin* under the last known SoC is set while the battery is in standby mode, this will be enabled.

---

#### Basic Storage Controls Register

Start address:

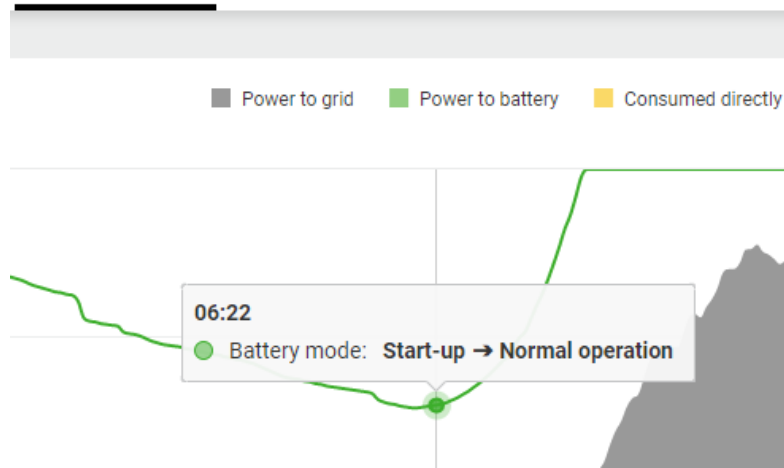
- for "float" setting: **40313**
- for "int+SF" setting: **40303**

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

---

#### Register manipulation and Battery status changes in Fronius Solar.web

Fronius Solar.web allow users to visualize status changes from the battery. These changes can be seen in Fronius Solar.web under the option Energy balance then Production or Consumption. The changes are marked with a bubble status, clicking on a state change will show the previous state followed by an arrow and the new state.



Battery state change from Start-up to Normal Operation.

Battery status changes are triggered during normal operation (when the battery is ready to enter in operation, security reasons, etc) or by manipulating the modbus registers MinRsvPct, InWRte, OutWRte and StorCtl\_Mod.

The changes could be triggered as follows:

- A minimum state of charge is set using the register MinRsvPct, the corresponding state change is "Energy-saving mode".
- Setting the registers InWRte, OutWRte, StorCtl\_Mod the battery status could change to "Forced Recharge".

# Meter Model

## Meter Model Register

The data of an energy meter connected with the Fronius Datamanager via Modbus RTU can be read by the relevant SunSpec models via Modbus TCP. In a similar way to the inverter models, there are also two different SunSpec models in this case:

- the meter model with floating point display (setting "float"; 211, 212 or 213)
- the meter model with integers and scale factors (setting "int+SF"; 201, 202 or 203)

The register number of the two model types is different!

The Modbus device ID of the energy meter is configurable (default = 200).

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .

There are 4 different meter locations, which are described by the location number (see table). Depending on where the Smart Meter is located and whether the inverter is producing or consuming, the signs of the PowerReal values and the Energy values change. These are shown in the following table:

Meter_Location	0 (grid)	1 (load)	3 (ext. generator)	256-511 (subload)
PowerReal_P_Sum (+ positive)	consuming from grid	producing power	generation	load is producing power
PowerReal_P_Sum (- negative)	feeding in to grid	normal consumption	consumption	normal consumption
energy plus (absolute counter)	import from grid = energy consumed	producing power* = energy produced	generation = energy produced	producing power* = energy produced
energy minus (absolute counter)	export to grid = energy produced	consumption = energy consumed	consumption = energy consumed	consumption = energy consumed

*\*is not typically. May occur when other power generation is located in load path and producing more power than load can consume.*



# End Block

---

## General

Two registers according to the last data model indicate that no further SunSpec models will follow.

The addresses of these two registers are different depending on the device type (inverter, String Control, energy meter) and selected data type ("float" or "int+SF").

- Inverter:
  - - Start address for setting "float": 40313
  - - Start address for setting "int+SF": 40303
- Fronius String Control:
  - - Start address: 40127
- Energy meter:
  - - Start address for setting "float": 40195
  - - Start address for setting "int+SF": 40176

---

## End Block

The register tables can be found on the Fronius website or opened using the link: <http://www.fronius.com/QR-link/0024> .







[fronius.com/en/solar-energy/installers-partners/products-solutions/monitoring-digital-tools](https://fronius.com/en/solar-energy/installers-partners/products-solutions/monitoring-digital-tools)

MONITORING &  
DIGITAL TOOLS

**Fronius International GmbH**

Froniusstraße 1  
4643 Pettenbach  
Austria  
[contact@fronius.com](mailto:contact@fronius.com)  
[www.fronius.com](http://www.fronius.com)

Under [www.fronius.com/contact](http://www.fronius.com/contact) you will find the addresses of all Fronius Sales & Service Partners and locations.