Fronius Datamanager Modbus TCP & RTU

DE

Bedienungsanleitung

Anlagenüberwachung

EN-US

Operating Instructions

System monitoring

42,0410,2049 018-13062019

Contents

Das Modbus Protokoll

Allgemeines

Die Beschreibung des Protokolls entstammt zum größten Teil den Modbus Spezifikationen, die öffentlich auf 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 Fronius Datamanager 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), nur bei Fronius Datamanager 2.0

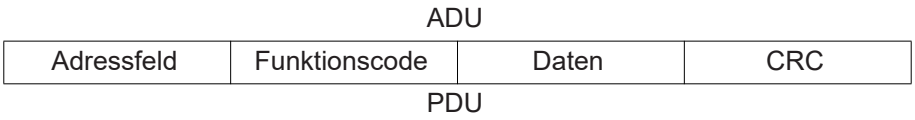
Im Fall von Modbus RTU kann es immer nur einen Master im System geben. Grundsätzlich gilt, dass nur ein Master Anforderungen (Requests) initiieren darf. Ein Slave darf nur antworten (Response), wenn dieser vom Master angesprochen wurde; untereinander dürfen die Slaves nicht kommunizieren. Wird ein Broadcast 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 dieser 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 Fehlernachricht (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.

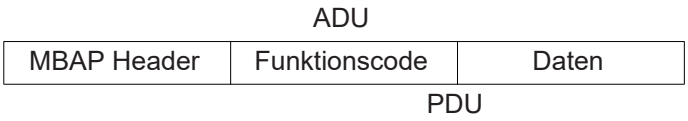
Modbus Geräte stellen Daten in 16 Bit großen Datenblöcken (Register) zur Verfügung. In bestimmten Fällen können einzelne Datenpunkte auch mehrere Datenblöcke umfassen (z. B. 2 Register = 32 Bit Wert).

Aufbau von Modbus Nachrichten

Eine Modbus Nachricht besteht grundsätzlich aus der Protokolldateneinheit (protocol data unit, PDU). Diese ist von darunter liegenden Kommunikationsschichten unabhängig. Abhängig von dem verwendeten Bus oder Netzwerk können noch weitere Felder hinzukommen. Diese Struktur wird dann Anwendungsdateneinheit (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 Anwendungsdateneinheit 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 – Slave ID (1 Byte) – CRC (2 Bytes) = 253 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 Längenfeld enthält die Anzahl der Bytes der nachkommenden Felder, einschließlich Unit ID und Datenfelder.
- **Unit ID** (1 Byte): Dieses Feld wird zur Adressierung der an den Fronius Datamanager angeschlossenen Geräte verwendet (Gateway-Funktion des Fronius Datamanagers). 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 **14**
  - **Modbus Geräte-ID für Fronius String Controls** auf Seite **14**
  - **Modbus Geräte-ID für Energiezähler** auf Seite **15**

  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. Der Fronius Datamanager unterstützt drei Funktionscodes für Lese- und Schreiboperationen:
- 03 (0x03) [1] Read Holding Registers
- 06 (0x06) [1] Write Single Register
- 16 (0x10) [1] 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) [1] und ein Exception Code hinzugefügt, der den Grund des Fehlers angibt.

[1] 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 Fehlernachricht (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 Fehlernachricht 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 Fehlernachricht identifizieren.
- **Datenfeld**
  Eine normale Antwort enthält Daten oder Statistikwerte im Datenfeld. Bei einer Fehlernachricht wird ein Exception Code im Datenfeld zurückgeliefert. Dieser Exception Code zeigt den Grund für die Fehlernachricht 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	Nur bei Modbus TCP. Das angesprochene Gerät kann nicht gefunden werden: a) das Gerät befindet sich nicht im SolarNet Ring oder b) das Gerät ist ausgeschaltet oder c) der SolarNet Ring ist offen

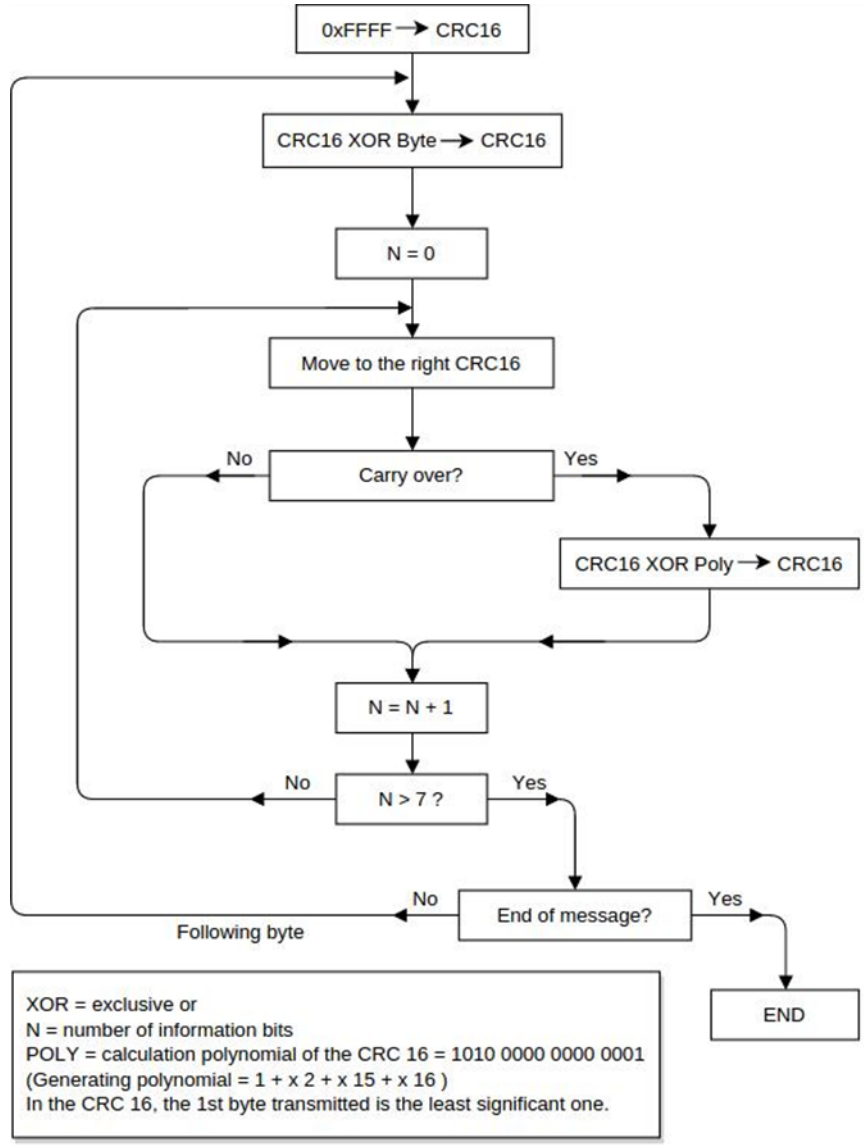
CRC Berechnung für Modbus RTU

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) Schiebevorgang, wird das nächste Byte genommen und mit dem aktuellen CRC Register XOR-verknüpft. Danach beginnt der Schiebeprozess 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
MBC	Fronius Modbus Card
PF	Power Factor (cos φ)
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

Kommunikation mit dem Modbus Master

Die Kommunikation des Fronius Datamanager mit dem Modbus-Master erfolgt über Registeradressen entsprechend der Spezifikationen der SunSpec Alliance. (http://www.sunspec.org/)

HINWEIS!

Der Fronius Datamanager unterstützt auch die Anbindung von Fronius String Controls über Fronius Solar Net.

Fronius String Controls werden durch einen eigenen Common Block und das darauffolgende String Combiner Model dargestellt.
Zusätzlich bietet der Fronius Datamanager die Möglichkeit, die Daten eines über Modbus RTU angeschlossenen Energiezählers via Modbus TCP zur Verfügung zu stellen. Der Zähler wird durch einen eigenen Common Block und das darauffolgende Meter Model dargestellt.

Die Zuordnung der Registeradressen zur entsprechenden Funktion ist folgenden Tabellen zu entnehmen:

- Für alle Geräte:
  - Common Block (1)
- Für Wechselrichter:
  - Fronius Register
  - Inverter Model (101, 102, 103, 111, 112 oder 113)
  - Inverter Controls:
    - Nameplate (120)
    - Basic Settings (121)

- Extended Measurements & Status (122)
- Immediate Controls (123)
- Multiple MPPT Inverter Extension (160)
- Basic Storage Control (124)
  nur bei Fronius Hybrid Wechselrichtern verfügbar

- Für String Controls:
  - String Combiner Model (403)
- Für Energiezähler:
  - Meter Model (201, 202, 203, 211, 212 oder 213)

HINWEIS!

gilt nur für Modbus RTU und nur wenn kein Energiezähler angeschlossen ist: Wenn kein Datenaustausch am RS-485 Bus stattfindet, können Rauschen und Störungen die Leitungen beeinflussen.
Damit ein Empfänger in einem definierten Zustand bleibt wenn keine Datensignale anliegen, sollten Vorspannungswiderstände verwendet werden, um einen definierten Ruhezustand auf den Datenleitungen zu erhalten.
Der Fronius Datamanager verfügt über keine Vorspannungswiderstände. Detaillierte Informationen über die Verwendung solcher Widerstände finden sich im Dokument „MODBUS over serial line specification and implementation guide V1.02" (http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf).

Register Maps

Wechselrichter
SID Identifizierung als SunSpec Gerät
Common Block Geräteinformationen
Inverter Model Wechselrichter-Daten
Nameplate Model
Basic Settings Model
Ext. Measurement Model
Immediate Controls Model
Multi. MPPT Inv. Ext. Model
Basic Storage Control (nur bei Fronius Hybrid Wechselrichter)
End Block

String Control
SID Identifizierung als SunSpec Gerät
Common Block Geräteinformationen
String Combiner Model String Control Daten
End Block

Energiezähler
SID Identifizierung als SunSpec Gerät
Common Block Geräteinformationen
Meter Model Energiezähler-Daten
End Block

Die Registerlisten können von der Fronius Homepage heruntergeladen werden:

https://www.fronius.com/de/downloads / Solar Energy / Modbus Sunspec Maps, State Codes und Events

Antwortzeiten

Die Antwortzeiten hängen unter anderem von der Anzahl der Geräte im Fronius Solar Net Ring ab. Je mehr Geräte verwendet werden, desto größer muss das Timeout für Antworten sein.

HINWEIS!

Bei mehreren Geräten im Fronius Solar Net Ring sollte für Abfragen von Wechselrichterdaten ein Timeout von mindestens 1 Sekunde verwendet werden.

Empfehlung für Timeout-Werte
Da bei Fronius String Controls eine einzige Modbus-Abfrage zwei Abfragen über Fronius Solar Net bewirken kann, sind etwas längere Antwortzeiten als bei Wechselrichteranfragen möglich. Wenn Fronius String Controls vorhanden sind, sollte daher ein größerer Timeout-Wert für Antworten verwendet werden.

Bei der ersten Abfrage der Common Block Daten nach einem Neustart des Fronius Datamanagers müssen die Informationen über die Fronius String Control einmalig über Fronius Solar Net abgefragt werden. Daher benötigt diese erste Abfrage ein wenig mehr Zeit als die darauffolgenden.

Bei einer größeren Anzahl von Geräten in einem Fronius Solar Net Ring, wird empfohlen diese auf mehrere Fronius Solar Net Ringe mit jeweils einem eigenen Fronius Datamanager aufzuteilen, um noch vertretbare Antwortzeiten zu erhalten. Fronius empfiehlt maximal 6 Wechselrichter mit einem Datamanager zu steuern.

Modbus Geräte-ID für Wechselrichter

Die Modbus Geräte-ID des Wechselrichters entspricht seiner Wechselrichter-Nummer, welche über das Bedienpanel des Wechselrichters eingestellt werden kann.
(siehe Bedienungsanleitung des Wechselrichters)

HINWEIS!

Hierbei gibt es nur eine einzige Ausnahme:
Die Wechselrichter-Nummer 00 wird auf Geräte-ID 100 umgelegt, da bei Modbus die Geräte-ID 0 für Broadcast Nachrichten reserviert ist.

Beispiel:

Wechselrichter-Nummer	Modbus Geräte-ID
00	100
01	001
02	002
03	003
99	099

Modbus Geräte-ID für Fronius String Controls

Die Modbus Geräte-ID einer Fronius String Control ergibt sich aus
- ihrer Adresse im Fronius Solar Net
- einem String Control Offset-Wert

Der Standardwert für den String Control Offset ist 101 da für die Wechselrichter der Bereich bis Modbus Geräte-ID 100 reserviert ist.
Der Offset-Wert kann jedoch über die Webseite des Fronius Datamanager verändert werden.
=> siehe Abschnitt “Datenausgabe über Modbus“

Beispiel 1: String Control Offset = 101 (Standardwert)

Fronius String Control Adresse	*Modbus Geräte-ID*
0	101
1	102
2	103
99	200

Ein Fronius Solar Net Ring erlaubt bis zu 100 Wechselrichter und bis zu 200 Fronius String Controls. Die verfügbaren Modbus Geräte-IDs sind ab 240 für andere Funktionen reserviert (z. B. für Energiezähler).
Mit dem Standard String Control Offset von 101 wären also Fronius String Control Adressen ab 139 (entspricht Modbus ID 240) nicht möglich.

Daher kann der String Control Offset über die Website des Fronius Datamanager verändert werden, wenn weniger als 100 Wechselrichter zum Einsatz kommen.

Beispiel 2: 30 Wechselrichter, 200 Fronius String Controls, String Control Offset = 40

Fronius String Control Adresse	*Modbus Geräte-ID*
0	40
1	41
2	42
199	239

Modbus Geräte-ID für Energiezähler

Ist ein Energiezähler (z. B. Fronius Smart Meter 63A) per Modbus RTU an den Fronius Datamanager angeschlossen, kann dieser per Modbus TCP über die fixe Modbus Geräte-ID 240 ausgelesen werden.

Event Flags

Zustandsänderungen und Fehler der Wechselrichter und Fronius String Controls werden als Event Flags dargestellt.

Detaillierte Informationen und Listen in verschiedenen Dateiformaten (xlsx, csv, json) können von der Fronius Homepage heruntergeladen werden:

https://www.fronius.com/de/downloads / Solar Energy / Modbus Sunspec Maps, State Codes und Events

HINWEIS!

Es können auch mehrere State Codes zu einem Ereignis zusammengefasst sein.

Für Wechselrichter gilt:
Eine genaue Beschreibung der State Codes ist in der Bedienungsanleitung des betreffenden Wechselrichters zu finden.
Wenn der Wechselrichter einen State Code erzeugt, wird im Fronius Datamanager das entsprechende Event Flag gesetzt.

HINWEIS!

Zusätzlich wird der entsprechende State Code auch in Register F_Active_State_-Code (214) angezeigt.

Event Flag und State Code bleiben so lange aktiv, wie auch der State Code am Wechselrichter anliegt. Tritt ein weiterer State Code auf, wird dieser ebenfalls in den Event Flags dargestellt. In diesem Fall kann es passieren, dass das vorherige Event Flag nicht gelöscht wird.
Daher ist es möglich, die Event Flags und den State Code manuell zu löschen:
durch Schreiben von 0xFFFF in Register F_Reset_All_Event_Flags (215)

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.

Fronius Basis Register: 212
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 **32**:
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
Geräte-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
Geräte-ID	Function Code	Adresse 40242	Anzahl der geschriebenen Register	Checksumme Low Byte / High Byte

Beispiele für Modbus TCP:

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

Senden (Bytes in Hexadezimal)

MBAP Header	01	03	9C 44	00 04
Details siehe Beschreibung MPAB Header	Geräte-ID	Function Code	Adresse 40004 (entspricht Register 40005)	Anzahl der auszulesenden Register

Empfangen (Bytes in Hexadezimal)

MBAP Header	01	03	08	46 72	6F 6E	69 75	73 00
Details siehe Beschreibung MPAB Header	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

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

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

MBAP Header	01	10	9D 32	00 01
Details siehe Beschreibung MPAB Header	Geräte-ID	Function Code	Adresse 40242	Anzahl der geschiebenen 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[1)]
- 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 0x7FFFFFFF): 0xFFFFFFFF
- int64 (-9223372036854775807 bis 9223372036854775807): 0x8000000000000000
- acc64 (0 bis 18446744073709551615): 0
- stringX: alle X Register mit 0x0000 gefüllt
- float32 (Bereich siehe IEEE 754): 0x7FC00000 (NaN)
- sunssf (Skalierungsfaktoren; -10 bis 10): 0x8000

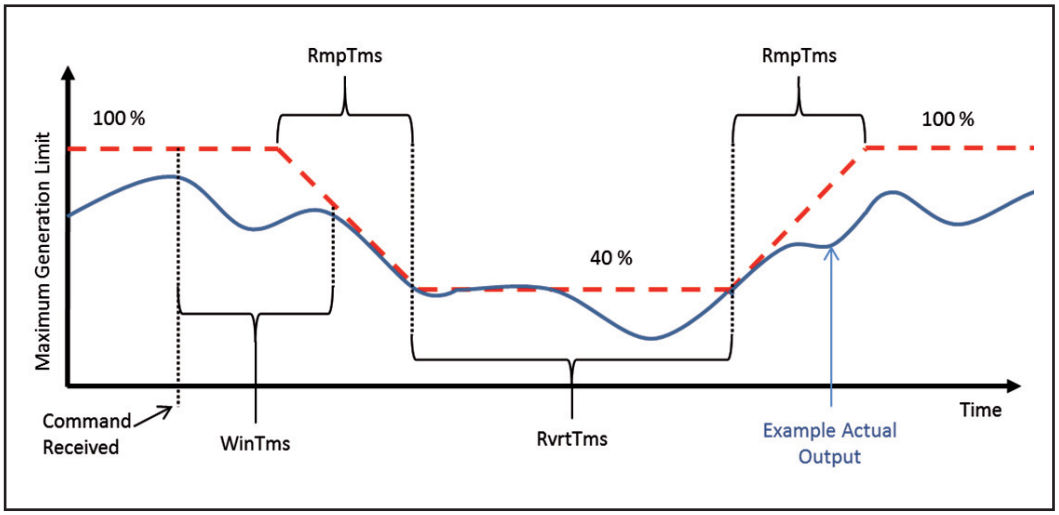
[1)] Das Prefix "0x" steht für hexadezimale Zahlen

HINWEIS!

Vom Datamanager 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

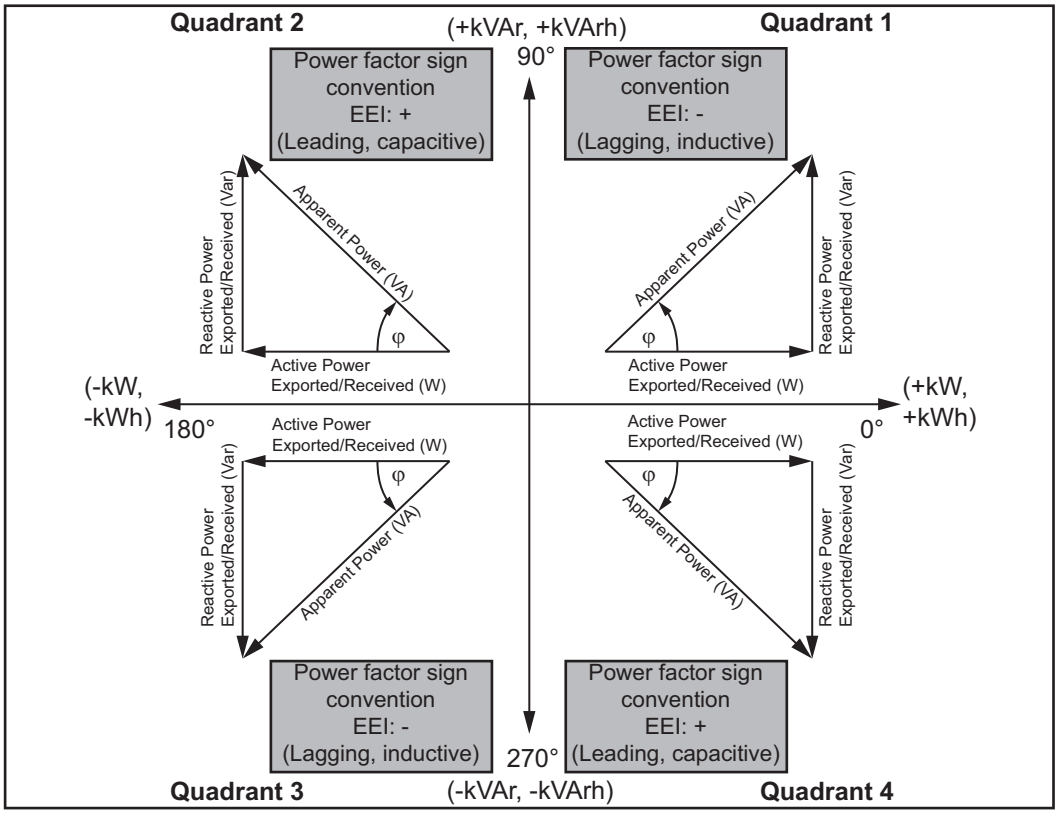


Zeitverhalten am Beispiel einer Leistungsreduktion

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. *OutPFSet_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 (Datamanager Webinterface - Einstellungen - EVU Editor) 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*** (derzeit nicht vom Datamanager unterstützt)
  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[1)] 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)

[1)] 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

Werte speichern
Bei nicht vorhandenen oder falsch angezeigten Daten können die oben angeführten Werte angepasst und am Datamanager gespeichert werden.
Änderungen haben derzeit keinen Einfluss auf die Funktionsweise des Datamanagers oder der Wechselrichter und dienen ausschließlich zur Anzeige von gerätespezifischen Informationen.
Um die Werte zu speichern, muss das Register *F_Store_Data (213)* eines beliebigen Wechselrichters mit 0xFFFF beschrieben werden. Anschließend sind die Werte für alle Wechselrichter permanent gespeichert und auch nach einem AC Reset des Datamanagers verfügbar.

Werte löschen
Es können nur die Werte für einen einzelnen Wechselrichter gelöscht werden. Dazu ist Register *F_Delete_Data (212)* des Wechselrichters mit 0xFFFF zu beschreiben.

Skalierungsfaktoren

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

Skalierungsfaktoren mit unveränderlichen Werten sind in den Tabellen in der Spalte "Range of values" angeführt.
Aktuelldaten (Daten von Wechselrichtern, String Controls 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 x 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 Fronius Datamanager keinen Exception Code zurück!
Die in diese Register geschriebenen Werte werden ohne Fehlermeldung vom Fronius Datamanager ignoriert.

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 Fronius Datamanager können via Internet-Browser Einstellungen für die Modbus Anbindung vorgenommen werden, welche über das Modbus-Protokoll nicht ansprechbar sind.

HINWEIS!

Bei Datenübertragung über Modbus RTU ist die Verwendung der Web-Schnittstelle normalerweise nicht erforderlich, da Modbus RTU werkseitig aktiviert ist.

Einstellungen - Modbus öffnen

1 Fronius Datamanager installieren
=> siehe Bedienungsanleitung Fronius Datamanager

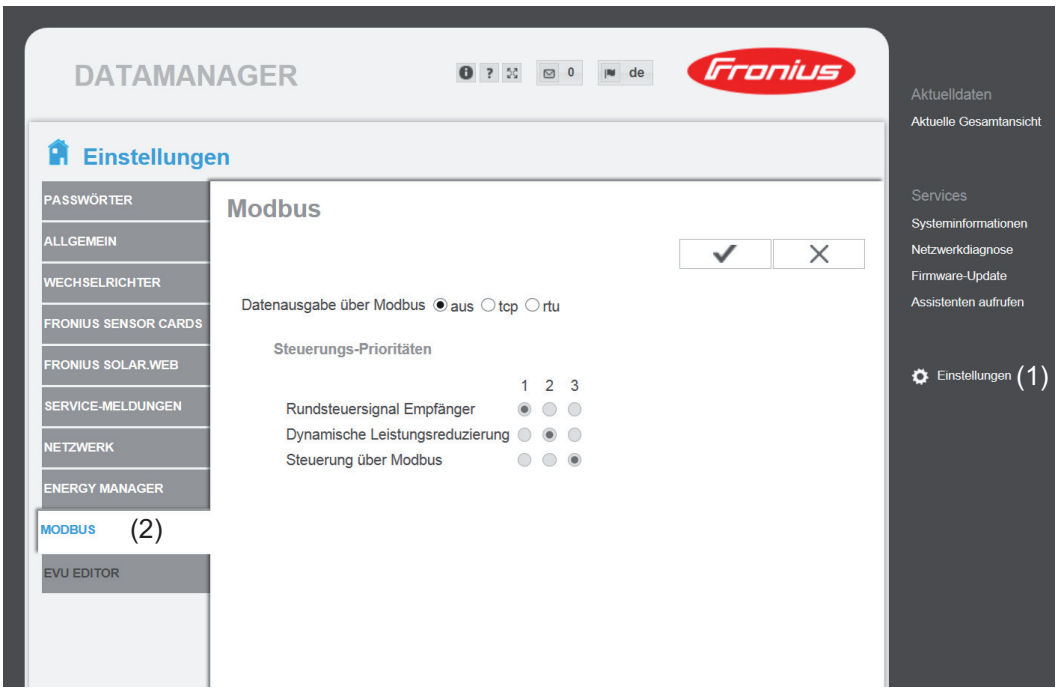
2 Internet-Browser öffnen

3 Im Adressfeld des Internet-Browsers eingeben:
- die IP Adresse des Fronius Datamanager (unter *Systeminformationen* abrufbar)
- oder Hostnamen und Domainnamen des Fronius Datamanager

Die Startseite der Web-Schnittstelle wird angezeigt

4 Den Bereich “Einstellungen“ (1) auswählen

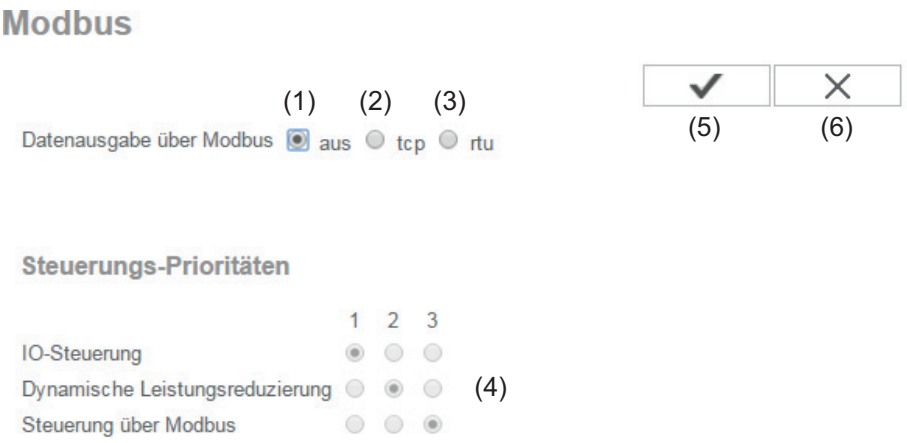
5 Den Menüpunkt “Modbus“ (2) öffnen



HINWEIS!

Beim Fronius Datamanager 2.
0 ist die ‘Datenausgabe über Modbus‘ werkseitig auf rtu eingestellt.
Die Auswahlmöglichkeit rtu ist beim Datamanager nicht vorhanden.

Datenausgabe über Modbus



Datenausgabe über Modbus
Aktivierung des Modbus Dienstes und Auswahl des Übertragungs-Protokolles.
Wird der Modbus Dienst aktiviert, stehen weitere Eingabefelder zur Verfügung.

Das Übertragungs-Protokoll Modbus rtu ist nur beim Fronius Datamanager 2.0 verfügbar.

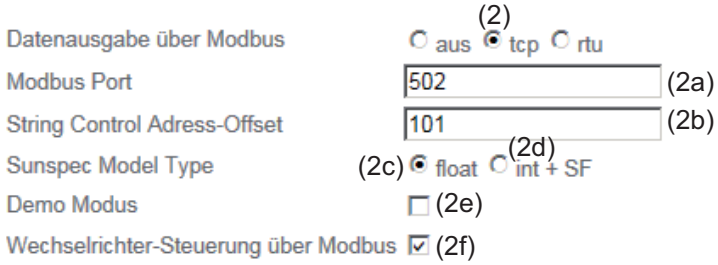
Hinweis! Befindet sich ein unter Einstellungen / Zähler konfigurierter Modbus Energiezähler (z. B. Fronius Smart Meter) im System, kann die Einstellung „rtu“ nicht verwendet werden. Bei Auswahl von „rtu“ wird in diesem Fall die Datenausgabe per Modbus automatisch deaktiviert. Diese Änderung ist erst nach einem erneuten Laden der Datamanager Web-Seite sichtbar.

Ein über RS485 an den Datamanager angeschlossener Energiezähler kann auch per Modbus TCP über die entsprechenden SunSpec Modelle ausgelesen werden. Die Modbus ID für den Zähler ist 240.

(1) **aus**
keine Datenausgabe über Modbus

Ist die Datenausgabe über Modbus deaktiviert, werden über Modbus an die Wechselrichter übertragene Steuerungsbefehle zurückgesetzt, z.B. keine Leistungsreduktion oder keine Blindleistungs-Vorgabe.

(2) **tcp**
Datenausgabe über Modbus tcp



(2a) **Modbus Port**
Nummer des TCP Ports, der für die Modbus-Kommunikation zu verwenden ist.

Voreinstellung: 502
Port 80 kann hierfür nicht verwendet werden.

(2b) **String Control Adress-Offset**
Offset-Wert für die Adressierung von Fronius String Controls per Modbus.
Für weitere Details siehe Abschnitt "Modbus Geräte-ID für Fronius String Controls".

Sunspec Model Type
zum Auswählen des Datentyps von Datenmodellen für Wechselrichter und von Datenmodellen für Energiezähler

(2c) **float**
Darstellung als Gleitkommazahlen
SunSpec Inverter Model 111, 112 oder 113
SunSpec Meter Model 211, 212 oder 213

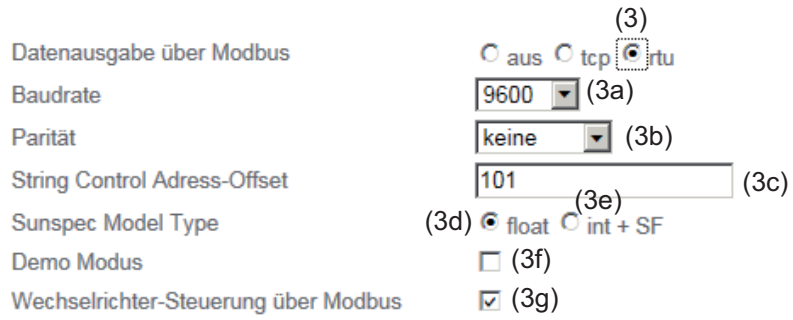
(2d) **int+SF**
Darstellung als ganze Zahlen mit Skalierungsfaktoren
SunSpec Inverter Model 101, 102 oder 103
SunSpec Meter Model 201, 202 oder 203

WICHTIG! Da die verschiedenen Modelle über unterschiedliche Anzahlen an Registern verfügen, ändern sich durch den Wechsel des Datentyps auch die Registeradressen aller nachfolgenden Modelle.

(2e) **Demo Modus**
Der Demo Modus dient zur Implementierung oder Validierung eines Modbus Masters. Er ermöglicht es, Wechselrichter-, Energiezähler- und String Control Daten auszulesen, ohne dass ein Gerät wirklich angeschlossen oder aktiv ist. Es werden für alle Register immer dieselben Daten zurückgeliefert.

(2f) **Wechselrichter-Steuerung über Modbus**
Wenn diese Option aktiviert ist, können die Wechselrichter über Modbus gesteuert werden.
Das Auswahlfeld Steuerung einschränken wird angezeigt.
Zur Wechselrichter-Steuerung gehören folgende Funktionen:
- Ein / Aus
- Leistungsreduktion
- Vorgabe eines konstanten Leistungs-Faktors cos Phi
- Vorgabe einer konstanten Blindleistung

(3) **rtu**
Datenausgabe über Modbus rtu



Hinweis: bei Anbindung eines Fronius Smart Meters wird Modbus RTU automatisch deaktiviert.

(3a) **Baudrate**
zum Eingeben der Baudrate,

(3b) **Parität**
Auswahlfeld zum Eingeben der Parität

(3c) **String Control Adress-Offset**
Offset-Wert für die Adressierung von Fronius String Controls per Modbus.
Für weitere Details siehe Abschnitt "Modbus Geräte-ID für Fronius String Controls".

Sunspec Model Type
zum Auswählen des Datentyps von Datenmodellen für Wechselrichter

(3d) **float**
Darstellung als Gleitkommazahlen
SunSpec Inverter Model 111, 112 oder 113

(3e) **int+SF**
Darstellung als ganze Zahlen mit Skalierungsfaktoren
SunSpec Inverter Model 101, 102 oder 103

WICHTIG! Da die verschiedenen Modelle über unterschiedliche Anzahlen an Registern verfügen, ändern sich durch den Wechsel des Datentyps auch die Registeradressen aller nachfolgenden Modelle.

(3f) **Demo Modus**
Der Demo Modus dient zur Implementierung und Validierung eines Modbus Masters. Er ermöglicht es, Wechselrichter-, Energiezähler- und String Control Daten auszulesen, ohne dass ein Gerät wirklich angeschlossen oder aktiv ist. Es werden für alle Register immer dieselben Daten zurückgeliefert.

(3g) **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

Symo Hybrid: Batteriesteuerungsvorgaben nicht möglich

(4) **Steuerungs-Prioritäten**
Die Steuerungs-Prioritäten legen fest, welcher Dienst bei der Wechselrichtersteuerung priorisiert wird.

1 = höchste Priorität, 3 = niedrigste Priorität

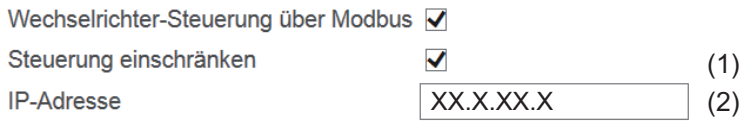
Die Steuerungs-Prioritäten können nur im Menüpunkt **EVU EDITOR** geändert werden.

(5) **Schaltfläche Übernehmen / Speichern**

(6) **Schaltfläche Abbrechen / Eingaben verwerfen**

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.



(1) **Steuerung einschränken**
Wenn diese Option aktiviert ist, dürfen nur bestimmte Geräte Steuerungsbefehle schicken.

(2) **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 Fronius Datamanager 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 98.7.65.25*4 zulässig

---

Änderungen speichern oder verwerfen

Speichert die Einstellungen und zeigt eine Meldung an, dass die Speicherung erfolgreich war.
Wird der Menüpunkt “Modbus“ verlassen ohne zu speichern, so werden alle vorgenommenen Änderungen verworfen.

Stellt eine Sicherheitsabfrage ob die vorgenommenen Änderungen tatsächlich verworfen werden sollen, und stellt dann die zuletzt gespeicherten Werte wieder her.

Fronius Register

Fronius Register

Diese Register gelten nur für Wechselrichter. Für Fronius String Controls und Energiezähler sind diese Register nicht relevant.

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link http://www.fronius.com/QR-link/0006 abrufbar.

Status-Code des Wechselrichters

Das Register ***F_Active_State_Code*** **(214)** zeigt den Status-Code des Wechselrichter an der gerade aufgetreten ist. Dieser wird eventuell auch am Display des Wechselrichter angezeigt. Dieser Code wird auch als Event Flag im Inverter Modell dargestellt. Der angezeigte Code bleibt so lange aktiv bis der entsprechende Status nicht mehr am Wechselrichter anliegt. Alternativ kann der Status auch per Register *F_Reset_All_Event_Flags* gelöscht werden.

Löschen der Event Flags und des Status-Codes

Die Event Flags in den Inverter Models (101, 102, 103 und 111, 112, 113) bleiben so lange aktiv bis der entsprechende Status nicht mehr am Wechselrichter anliegt. Es gibt einige wenige Ausnahmen, wo die Event Flags nicht mehr gelöscht werden. Daher können die Event Flags und der angezeigte Status-Code per Modbus-Befehl zurückgesetzt werden.

1 0xFFFF in das Register ***F_Reset_All_Event_Flags*** **(215)** schreiben

Der Inhalt folgender Register wird gelöscht:
- *F_Active_State_Code* (214)
- *Evt1*
- *Evt2*
- *EvtVnd1* bis *EvtVnd4*

Daten speichern und löschen

Schreibt man in das Register ***F_Store_Data*** **(213)** den Wert 0xFFFF werden alle Nennwerte (Ratings) für alle Wechselrichter am Fronius Datamanager gespeichert. Diese Werte können in den entsprechenden Registern des Nameplate Models und des Basic Settings Models verändert werden. Dies kann nützlich sein, wenn z. B. für ein Gerät keine Nennwerte automatisch ermittelt werden konnten und man die Werte manuell eintragen will.

Will man die gespeicherten Werte für einen bestimmten Wechselrichter löschen, muss man in das Register ***F_Delete_Data*** **(212)** den Wert 0xFFFF schreiben. Dann werden die Werte nur für diesen Wechselrichter gelöscht. Das Löschen kann immer nur auf den Wechselrichter angewendet werden, mit dem gerade kommuniziert wird.

Datentyp ändern

Über das Register ***F_ModelType*** **(216)** kann der Datentyp für die Datenmodelle für Wechselrichter und Energiezähler ausgewählt werden. Entweder kann die Darstellung als Gleitkommazahlen (float, Standard) oder als ganze Zahlen mit Skalierungsfaktoren (int+SF) ausgewählt werden.

HINWEIS!

Diese Einstellung betrifft nur das Inverter Model (Wechselrichter) und das Meter Model (Energiezähler).
Alle anderen Models verwenden weiterhin ganze Zahlen und Skalierungsfaktoren.
Diese Einstellung funktioniert gleich wie die über das Webinterface Modbus Einstellungen - SunSpec Model Type.

Einstellmöglichkeiten:
- Float = 1 (Standard): Inverter Model 111, 112 oder 113; Meter Model 211, 212 oder 213
- int+SF = 2: Inverter Model 101, 102 oder 103; Meter Model 201, 202 oder 203

HINWEIS!

Da die verschiedenen Models über eine unterschiedliche Anzahl an Registern verfügen, ändern sich durch den Wechsel des Datentyps auch die Registeradressen aller nachfolgenden Models.

HINWEIS!

Um irrtümlichen Ändern vorzubeugen, muss eine Änderung der Einstellung in Register F_ModelType unmittelbar nach Schreiben des Registers durch erneutes Schreiben des Wertes 0x06 bestätigt werden.
Erfolgt dies nicht wird die Änderung nach einigen Sekunden zurückgesetzt.

Anlagensummen

Über die folgenden Register können Leistungs- und Energiedaten von allen per Solar Net mit diesem Fronius Datamanager verbundenen Wechselrichtern, abgefragt werden.
Die Werte werden in Watt (W) bzw. Wattstunden (Wh) abgebildet und benötigen keine Skalierungsfaktoren.
- ***F_Site_Power*** **(500-501)**: Leistung
- ***F_Site_Energy_Day*** **(502-505)**: Tagesenergie
- ***F_Site_Energy_Year*** **(506-509)**: Jahresenergie
- ***F_Site_Energy_Total*** **(510-513)**: Gesamtenergie der gesamten Anlage

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, String Control, 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/0006 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/0006 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

Fronius 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

Name	Wert	Beschreibung
I_STATUS_NO_BUSINIT	9	Keine SolarNet Kommunikation
I_STATUS_NO_COMM_INV	10	Keine Kommunikation mit Wechselrichter möglich
I_STATUS_SN_OVERCURRENT	11	Überstrom an SolarNet Stecker erkannt
I_STATUS_BOOTLOAD	12	Wechselrichter wird gerade upgedatet
I_STATUS_AFCI	13	AFCI Event (Arc-Erkennung)

* 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/0006 abrufbar.

Basic Settings Model (121)

Basic Settings Register

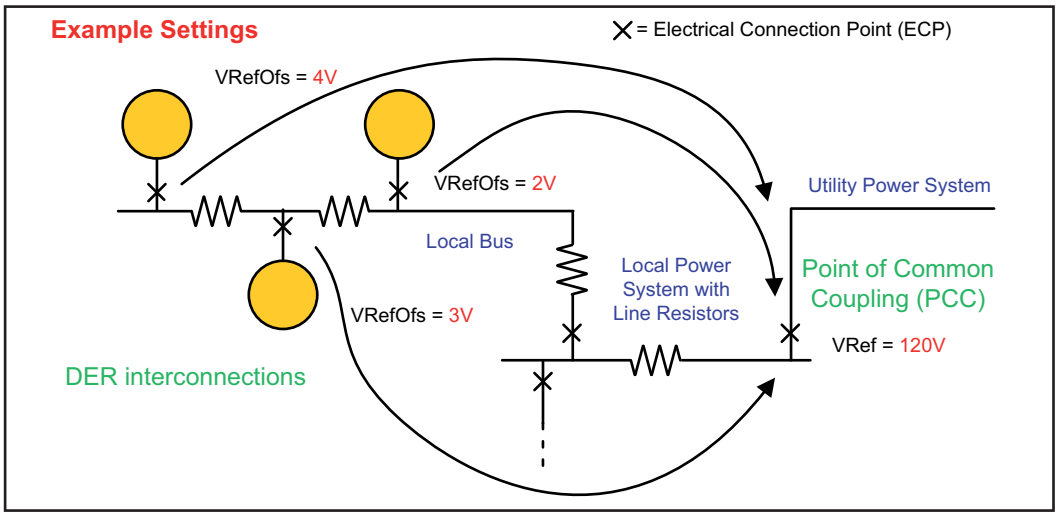
Startadresse:
- bei Einstellung „float“: ***40159***
- bei Einstellung „int+SF“: ***40149***

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link http://www.fronius.com/QR-link/0006 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“).

Die Angabe erfolgt in Volt im Bereich -20 (0xFFEC) bis 20 (0x0014).

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)

- ***ECPConn (5)***
  Dieses Register zeigt den Verbindungsstatus zum Netz an
  - *ECPConn* = 1: Wechselrichter speist gerade ein
  - *ECPConn* = 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 VArMaxPct)
  - Bit 2: Vorgabe eines konstanten Power Factors (FixedPF; entspricht OutPFSet)

- ***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

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/0006 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/0006 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 ECPConn 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 werden. Abhängig von der Software-Version des Wechselrichters können Werte kleiner als 10 zu einem erzwungenen Standby des Wechselrichters führen (kein Einspeisebetrieb).
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* 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 nicht-unterstützte "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. 30 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

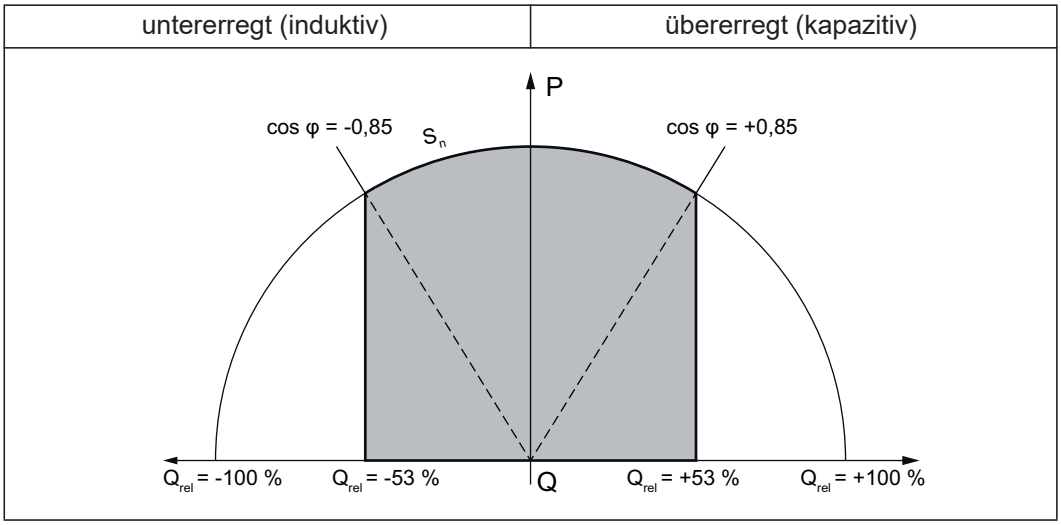
Der Blindleistungs-Betrieb wird grundsätzlich durch den maximalen Ausgangsstrom (die maximale Scheinleistung) sowie durch die operative Blindleistungs-Grenze des Wechselrichters begrenzt:

- Fronius IG Plus, CL cos phi = 0,85, VArrel = 100 %
- Fronius Galvo cos phi = 0,85, VArrel = 53 %
- Fronius Symo cos phi = 0,7, VArrel = 71 %

HINWEIS!

Aufgrund der aktuellen technischen Gegebenheiten kann per Modbus nur ein cos phi bis zu maximal ±0,80 vorgegeben werden.
VAr_{rel} Vorgaben können unter Umständen aber einen niedrigeren Wert erzwingen.

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



Blindleistung und Power Factor

Legende:

W	Leistung	VAr_{max}	Nenn-Blindleistung
W_{max}	Nennleistung	VAr_{rel}	relative Blindleistung (VAr/VArmax)
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* erneut starten

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* erneut starten

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 von bis zu zwei DC Eingängen des Wechselrichters.

Verfügt der Wechselrichter über zwei 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.

Sollte der Wechselrichter nur über einen DC Eingang verfügen, werden alle Werte des zweiten Strings auf "not implemented" gesetzt (ab Register 2_DCA). Die Bezeichnung des zweiten Eingangs (Register 2_IDStr) lautet in diesem Fall "Not supported". Die Werte des ersten (und einzigen) Eingangs werden normal angezeigt.

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/0006 abrufbar.

Basic Storage Control Model (124)

Allgemeines

Dieses Model ist nur für Fronius Hybrid Wechselrichter 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 OutWRte und InWRt zurück.
  WChaMax := max(MaxChaRte, MaxDisChaRte)
- Wenn kein Energiespeicher verfügbar ist liefert das Register den Wert 0 zurück.

ChaState
- Ladestand des Energiespeicher in %:
  Estimated_Capacity_Remaining [Wh] / 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

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 W angenommen.

Für resultierende Leistungsfenster gilt:
- negative Leistungswerte entsprechen einer Ladung des Energiespeichers
- positive Werte entsprechen einer Entladung des Energiespeichers

HINWEIS!

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

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)

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

Vorgabe des minmalen Ladestandes

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 Speicher 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 (Fronius Anlagenüberwachung - Einstellungen - EVU-Editor - Batterie Ladung). 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 Fronius Solar Battery 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/0006 abrufbar.

String Combiner Model (403)

String Combiner Register	Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link http://www.fronius.com/QR-link/0006 abrufbar.

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 240.

Die Register Tabellen sind auf der Fronius Homepage zu finden oder direkt über den Link http://www.fronius.com/QR-link/0006 abrufbar.

End Block

Allgemeines

Zwei Register nach dem letzten Datenmodell zeigen an, dass keine weiteren SunSpec-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/0006 abrufbar.

String Combiner Event Flags

String Combiner Event Flags

Name	Event Flags
LOW_VOLTAGE	0x00000001
LOW_POWER	0x00000002
LOW_EFFICIENCY	0x00000004
CURRENT	0x00000008
VOLTAGE	0x00000010
POWER	0x00000020
PR	0x00000040
DISCONNECTED	0x00000080
FUSE_FAULT	0x00000100
COMBINER_FUSE_FAULT	0x00000200
COMBINER_CABINET_OPEN	0x00000400
TEMP	0x00000800
GROUNDFAULT	0x00001000
REVERSED_POLARITY	0x00002000
INCOMPATIBLE	0x00004000
COMM_ERROR	0x00008000
INTERNAL_ERROR	0x00010000
THEFT	0x00020000
ARC_DETECTED	0x00040000

Contents

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.

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 Fronius Datamanager:
- 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), only for Fronius Datamanager 2.0.

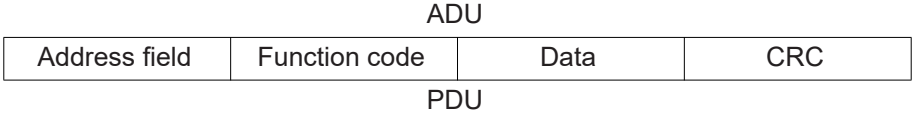
In the case of Modbus RTU, there can only ever be one master in the system. In principle, only one master may initiate requests. A slave may only give a response if it has been addressed by the master; the slaves cannot communicate with each other. If a broadcast request (request to all available slaves 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.

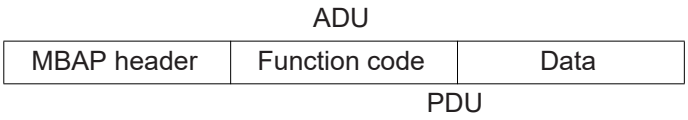
Modbus devices provide data in 16 bit large data blocks (registers).
In certain cases, individual data points may also cover several data blocks (e.g., 2 registers = 32 bit value).

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).



Structure of a Modbus message for Modbus RTU



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 Fronius Datamanager (gateway function of the Fronius Datamanager). 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 **60**
  - **Modbus Device ID for Fronius String Controls** on page **60**
  - **Modbus Device ID for Energy Meters** on page **61**

  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. The Fronius Datamanager supports three function codes for read and write operations:
- 03 (0x03) [1] read holding registers
- 06 (0x06) [1] write single register
- 16 (0x10) [1] 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) [1] and an exception code is added, which indicates the reason for the error.

[1] 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.
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 SunSpec 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	Only for Modbus TCP. The addressed device cannot be found: a) the device is not in the SolarNet Ring or b) the device is switched off or c) the SolarNet Ring is open.

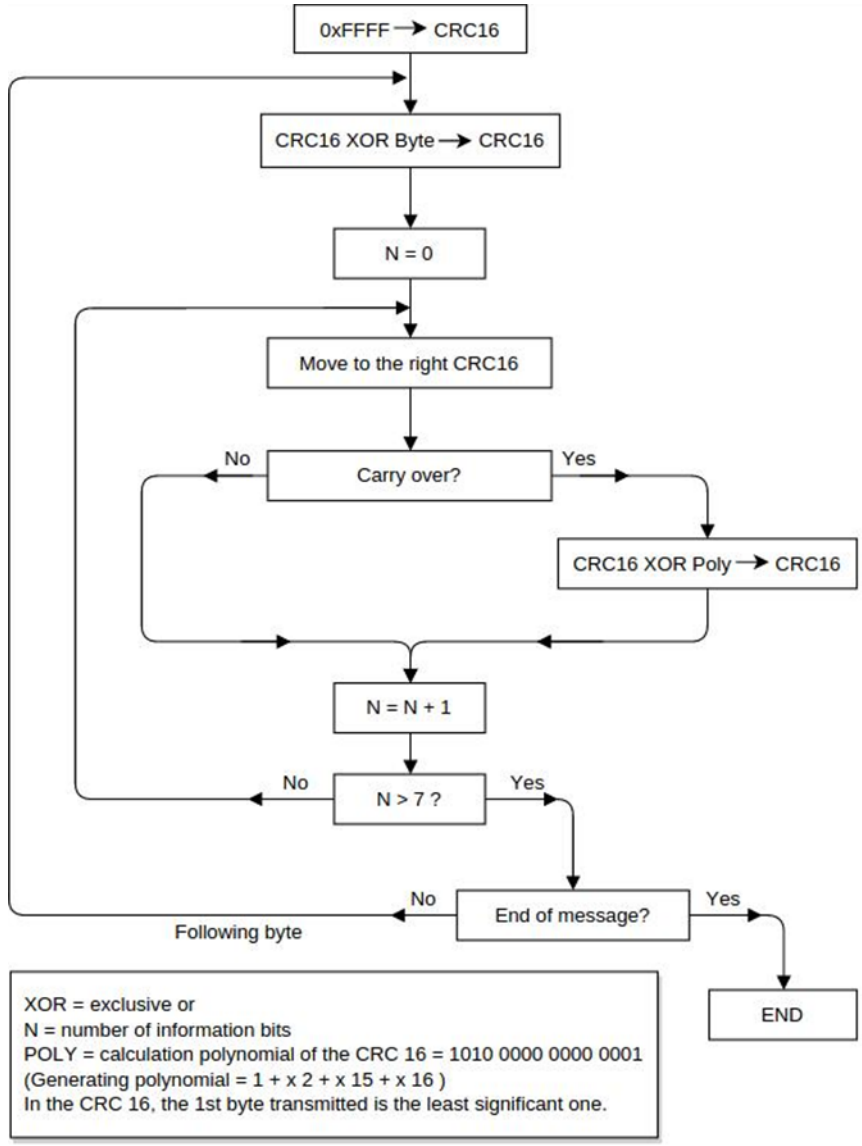
CRC Calculation for Modbus RTU

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 byes 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
MBC	Fronius Modbus Card
PF	Power factor (cos φ)
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

Communication with the Modbus Master

The Fronius Datamanager communicates with the Modbus master using register addresses in accordance with the SunSpec Alliance specifications.
(http://www.sunspec.org/)

NOTE!

The Fronius Datamanager also supports the integration of Fronius String Controls via Fronius Solar Net.

Fronius String Controls are displayed by an integrated Common Block and the subsequent String Combiner Model.
In addition, the Fronius Datamanager offers the option of providing via Modbus TCP data of an energy meter connected via Modbus RTU. The meter is displayed via an integrated Common Block and the subsequent Meter Model.

The allocation of register addresses to the corresponding function can be found in the following tables:

- For all devices:
  - Common Block (1)
- For inverters:
  - Fronius Register
  - Inverter model (101, 102, 103, 111, 112, or 113)
  - Inverter Controls:
    - Nameplate (120)
    - Basic Settings (121)

- Extended Measurements & Status (122)
- Immediate Controls (123)
- Multiple MPPT Inverter Extension (160)
- Basic Storage Control (124)
  only available with Fronius Hybrid inverters

- For Fronius String Controls:
  - String Combiner Model (403)
- For energy meters:
  - Meter Model (201, 202, 203, 211, 212, or 213).

NOTE!

Only applies for Modbus RTU and only if no energy meter is connected:
If no data exchange takes place on the RS-485 bus, noise and interference may affect the lines.
In order for a receiver to remain in a defined status when there are no data signals, bias resistors should be used in order to maintain a defined idle state on the data lines.
The Fronius Datamanager does not have any bias resistors. Detailed information about the use of these resistors can be found in the document "MODBUS over serial line specification and implementation guide V1.02" (http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf).

Maps Register

Inverter
SID Identification as a SunSpec device
Common Block Device information
Inverter Model Inverter data
Nameplate Model
Basic Settings Model
Ext. Measurement Model
Immediate Controls Model
Multi. MPPT Inv. Ext. Model
Basic Storage Control (only in Fronius Hybrid inverter)
End Block

Fronius String Control
SID Identification as a SunSpec device
Common Block Device information
String Combiner Model Fronius String Control data
End Block

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

The register lists can be downloaded from the Fronius homepage:

https://www.fronius.com/de/downloads / Solar Energy / Modbus Sunspec Maps, State Codes und Events

Response Times

The response times depend on factors such as the number of devices in the Fronius Solar Net ring. The higher the number of devices used, the longer the timeout for responses needs to be.

NOTE!

If there are several devices in the Fronius Solar Net ring, a timeout of at least 1 second should be used when querying inverter data.

Recommendation for Timeout Values
When using Fronius String Controls, a single Modbus request might result in two requests being sent via Fronius Solar Net; this can lead to longer response times than when using inverter requests. If Fronius String Controls are present, you should therefore use a higher timeout value for responses.

When first requesting common block data after restarting the Fronius Datamanager, the information about the Fronius String Control must first be requested using Fronius Solar Net. For this reason, this first request will take a little more time than subsequent requests.

If there are a larger number of devices in a Fronius Solar Net ring, it is advisable to split these between several Fronius Solar Net rings, which each have their own Fronius Datamanager, in order to speed up responses further. Fronius recommends operating a maximum of 6 inverters with a Datamanager.

Modbus Device ID for Inverters

The inverter's Modbus device ID is the same as its inverter number, which can be set using the control panel on the inverter.
(See the inverter operating instructions.)

NOTE!

There is only one exception to this rule:
The inverter number 00 converts to device ID 100 because Modbus reserves device ID 0 for broadcast messages.

Example:

Inverter number	**Modbus device ID**
00	100
01	001
02	002
03	003
99	099

Modbus Device ID for Fronius String Controls

The Modbus device ID of a Fronius String Control is derived from
- its address in Fronius Solar Net
- a String Control offset value.

The default value for the String Control offset is 101 because the range reserved for inverters goes up to Modbus device ID 100.
The offset value can, however, be adjusted via the Fronius Datamanager web page.
=> see section "Data Output via Modbus"

Example 1: String Control offset = 101 (standard value)

Fronius String Control address	*Modbus device ID*
0	101
1	102
2	103
99	200

A Fronius Solar Net Ring allows up to 100 inverters and up to 200 Fronius String Controls. The available Modbus device IDs are reserved for other functions (e.g., for energy meters) from 240.
With the standard String Control offset of 101, it would therefore not be possible to have Fronius String Control addresses from 139 (which corresponds to Modbus ID 240) upwards.

For this reason, it is possible to adjust the String Control offset on the Fronius Datamanager website if fewer than 100 inverters are being used.

Example 2: 30 inverters, 200 Fronius String Controls, String Control offset = 40

Fronius String Control address	*Modbus device ID*
0	40
1	41
2	42
199	239

Modbus Device ID for Energy Meters

If an energy meter (e.g., Fronius Smart Meter 63A) is connected to the Fronius Datamanager via Modbus RTU, it can be read out via the fixed Modbus device ID 240 using Modbus TCP.

Event Flags

Status changes and faults in the inverters and Fronius String Controls are shown as event flags.

Detailed information and lists can be downloaded in various formats (xlsx, csv, json) from the Fronius website:

https://www.fronius.com/de/downloads / Solar Energy / Modbus Sunspec Maps, State Codes und Events

NOTE!

It is also possible to combine several state codes for one event.

For inverters:
An accurate description of the state codes can be found in the operating instructions of the relevant inverter.
If the inverter generates a state code, the relevant event flag is set in the Fronius Datamanager.

NOTE!

In addition, the relevant state code is also displayed in register F_Active_State_Code (214).

The event flag and state code will remain active for as long as the state code is displayed on the inverter. If another state code is generated, it will also be displayed in the event flags. In this case, there is a chance that the previous event flag will not be deleted.
It is therefore possible to manually delete the event flags and the state code by entering 0xFFFF in register F_Reset_All_Event_Flags (215).

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.

Fronius Basic Register: 212
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 **78**:
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 four registers starting from register 40005 (Mn, Manufacturer)

Send (bytes in hexadecimal)

01	03	9C 44	00 04	2A 4C
Device ID	Function code	Address 40004 (corresponds 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
Device ID	Function code	Number 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
Device ID	Function code	Address 40242	Number of registers to be entered	Number of data bytes still to follow	Register value to be entered 0x1388 = 5000	Checksum Low byte / High byte

01	10	9D 32	00 01	8F AA
Device ID	Function code	Address 40242	Number of registers entered	Checksums Low byte / High byte

Examples for Modbus TCP:

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

Send (bytes in hexadecimal)

MBAP header	01	03	9C 44	00 04
For details, see description of MBAP header	Device ID	Function code	Address 40004 (corresponds to register 40005)	Number of registers to be read

Receive (bytes in hexadecimal)

MBAP header	01	03	08	46 72	6F 6E	69 75	73 00
For details, see description of MBAP header	Device ID	Function code	Number of bytes	Address 40005 "F" and "r"	Address 40006 "o" and "n"	Address 40007 "i" and "u"	Address 40008 "s" and 0

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

MBAP header	01	10	9D 32	00 01	02	13 88
For details, see description of MBAP header	Device ID	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	01	10	9D 32	00 01
For details, see description of MBAP header	Device ID	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[1)]
- 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
- acc64 (0 to 18446744073709551615): 0
- stringX: all X registers filled with 0x0000
- float32 (range see IEEE 754): 0x7FC00000 (NaN)
- sunssf (scaling factors; -10 to 10): 0x8000

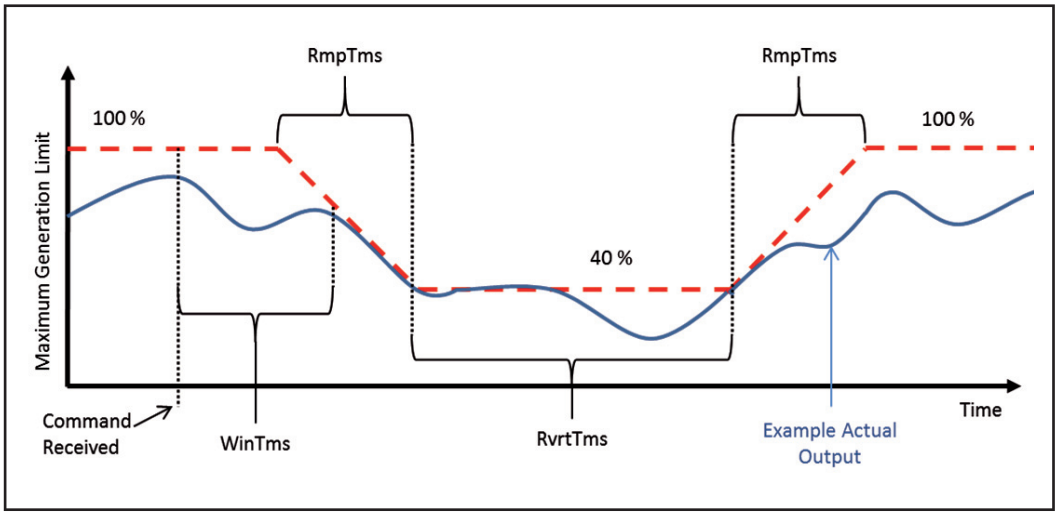
[1)] The prefix “0x” stands for hexadecimal numbers.

NOTE!

Data points not supported by the data manager 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

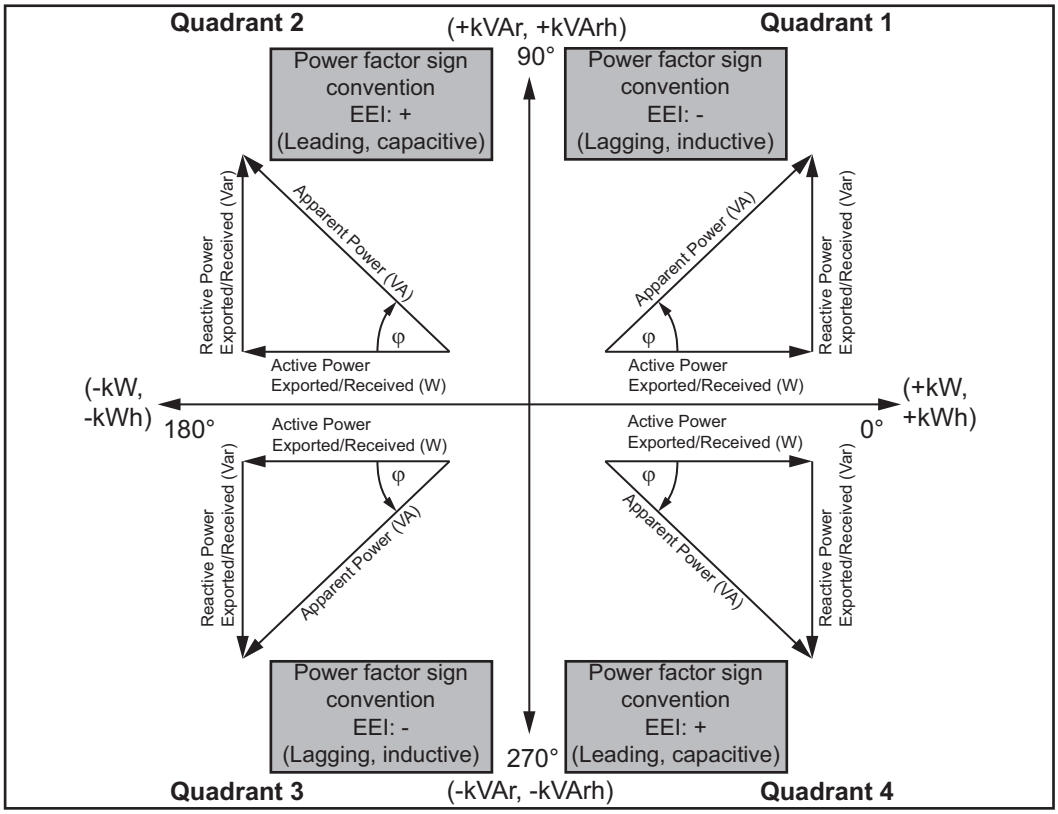


Time Response Illustrated by Power Reduction

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 (Datamanager web interface - Settings - UC Editor) 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*** (currently not supported by the Datamanager)
  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[1)] 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 scaling 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 scaling 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

Saving Values
If data is not available or is incorrectly displayed, the values listed above can be adjusted and saved on the Datamanager.
Changes currently have no influence on the way the Datamanager or the inverters function and are merely used to display device-specific information.
In order to save the values, the register *F_Store_Data (213)* of any inverter must be written with 0xFFFF. The values for all inverters are then permanently saved and are also available after an AC reset of the Datamanager.

Deleting Values
It is only possible to delete values for an individual inverter. To do this, enter 0xFFFF into the register *F_Delete_Data (212)* of the relevant inverter.

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 (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, Fronius String Controls, 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^(-1) = 1000 W

Non-Writable Registers

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

NOTE!

If you try to write these registers, the Fronius Datamanager will not return an exception code!
The values entered into these registers will be ignored without any error message being issued by the Fronius Datamanager.

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 Fronius Datamanager web interface to apply the Modbus connection settings which cannot be accessed via the Modbus protocol.

NOTE!

It is not usually necessary to use a web interface when transferring data via Modbus RTU since Modbus RTU is enabled at the factory.

Opening the Modbus Settings

1 Install Fronius Datamanager
=> see the Fronius Datamanager operating instructions.

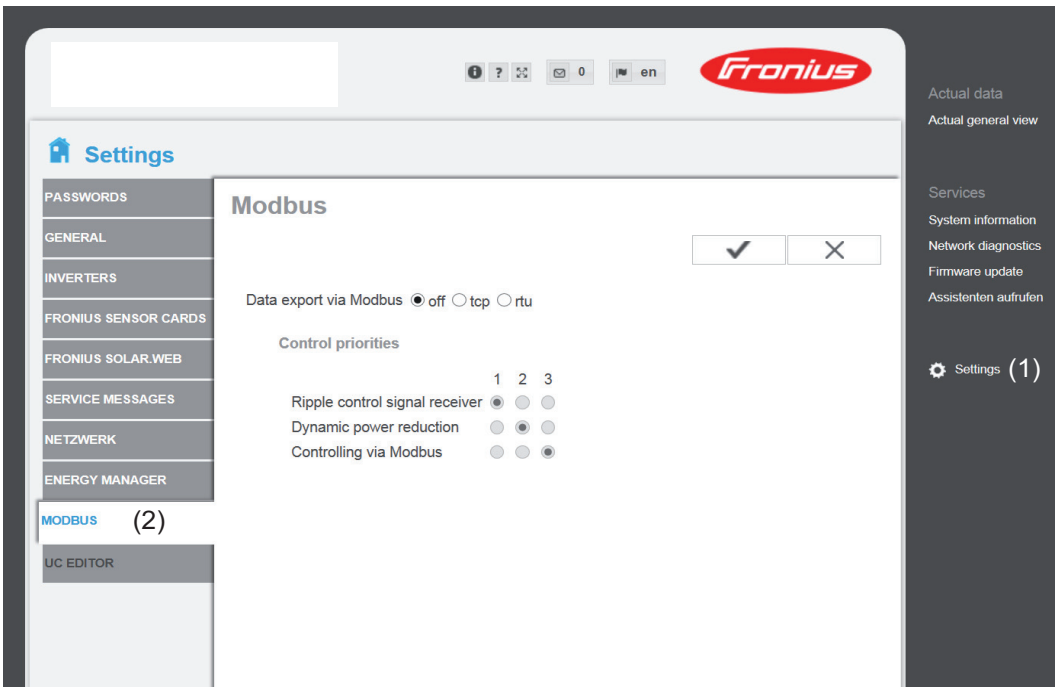
2 Open Internet browser

3 Enter the following in the address field of the Internet browser:
- the IP address of the Fronius Datamanager (can be accessed via *System Information*)
- or host name and domain name of the Fronius Datamanager.

The web interface's start page is displayed.

4 Select the "Settings" section (1).

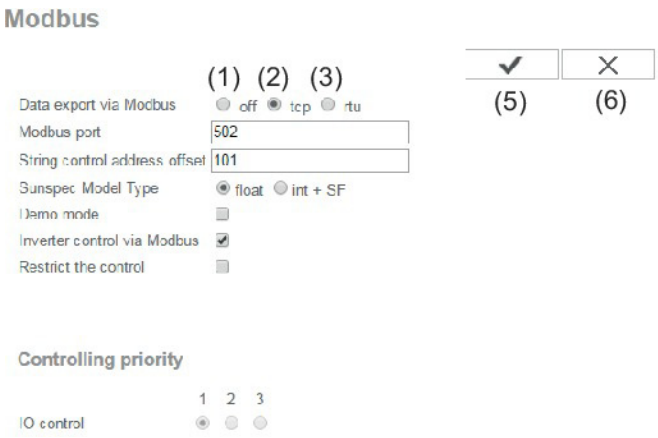
5 Open the "Modbus" section (2).



NOTE!

In the case of Fronius Datamanager 2.
0, the "Data output via Modbus" is set to rtu in the factory.
The rtu option is not available for the Datamanager.

Data Output via Modbus



Data Output via Modbus
Activation of the Modbus service and selection of the transmission protocol.
If the Modbus service is activated, additional entry fields are available.

The Modbus rtu transmission protocol is only available for Fronius Datamanager 2.0.

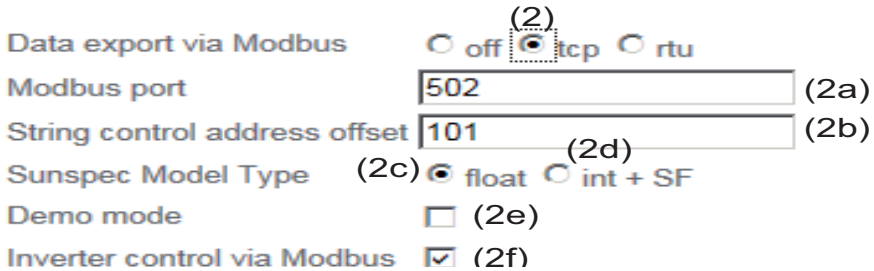
Note! If there is a Modbus energy meter (e.g., Fronius Smart Meter) configured under Settings/Meter on the system, it will not be possible to use the "rtu" setting. In this case, data output via Modbus will be deactivated automatically upon selection of "rtu." This change will only be visible once the Datamanager website has been reloaded.

Any energy meter connected via RS485 can also be read by Modbus TCP via the corresponding SunSpec models. The Modbus ID for the meter is 240.

(1) **off**
No data output via Modbus.

If the data output via Modbus is deactivated, control commands sent to the inverter via Modbus are reset, e.g., no power reduction or no reactive power specification.

(2) **tcp**
Data output via Modbus TCP.



(2a) **Modbus port**
Number of the TCP port to be used for Modbus communication.

Presetting: 502
Port 80 cannot be used for this purpose.

(2b) **String Control address offset**
Offset value used to assign addresses to Fronius String Controls via Modbus.
For further details, see the section entitled "Modbus Device ID for Fronius String Controls."

SunSpec Model Type
Used to select the data type of data models for inverters and energy meters.

(2c) **float**
Display as floating-point numbers.
SunSpec inverter model 111, 112 or 113
SunSpec meter model 211, 212 or 213

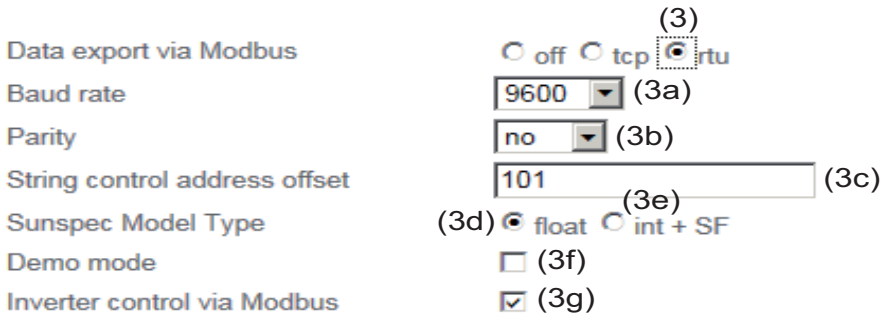
(2d) **int+SF**
Display as integers with scaling factors.
SunSpec inverter model 101, 102 or 103
SunSpec meter model 201, 202 or 203

IMPORTANT! Since the different models have different numbers of registers, the register addresses of all the subsequent models also change when the data type is changed.

(2e) **Demo mode**
The demo mode is used to implement or validate a Modbus master. It enables you to read inverter, energy meter, and Fronius String Control data without actually having to connect or activate a device. The same data are always sent back for all the registers.

(2f) **Inverter control via Modbus**
If this option is activated, the inverter can be controlled via Modbus.
The "Restrict the control" selection field is displayed.
Inverter control includes the following functions:
- On/off
- Power reduction
- Setting a constant power factor (cos phi)
- Setting a constant reactive power

(3) **rtu**
Data output via Modbus rtu.



Notification: when connecting a Fronius Smart Meter, Modbus RTU is automatically disabled

(3a) **Baud rate**
Used to enter the baud rate.

(3b) **Parity**
Selection field for entering the parity.

(3c) **String Control address offset**
Offset value used to assign addresses to Fronius String Controls via Modbus.
For further details, see the section entitled "Modbus Device ID for Fronius String Controls."

SunSpec model type
Used to select the data type of data models for inverters.

(3d) **float**
Display as floating-point numbers.
SunSpec inverter model 111, 112 or 113

(3e) **int+SF**
Display as integers with scaling factors.
SunSpec inverter model 101, 102 or 103

IMPORTANT! Since the different models have different numbers of registers, the register addresses of all the subsequent models also change when the data type is changed.

(3f) **Demo mode**
The demo mode is used to implement and validate a Modbus master. It enables you to read inverter, energy meter, and Fronius String Control data without actually having to connect or activate a device. The same data are always sent back for all the registers.

(3g) **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

Symo Hybrid: Not possible to specify battery control settings

(4) **Controlling priority**
Used to specify which service is given priority by the inverter control unit.

1 = highest priority, 3 = lowest priority.

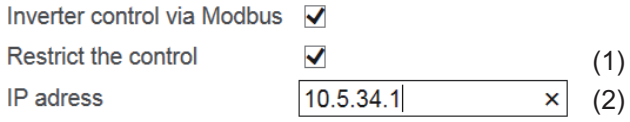
The control priorities can only be changed in the **UC EDITOR** menu item.

(5) **"Apply/Save" button**

(6) **"Cancel/Discard entries" button**

Limit 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.



(1) **Limit Control**
If this option is activated, only certain devices will be able to send control commands.

(2) **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 Fronius Datamanager 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 through IP addresses *98.7.65.1 to 98.7.65.254*

Save or Reject Changes

Saves the changes and displays a message confirming this.
If you exit the "Modbus" section without saving your changes, all the changes you have made will be rejected.

Prompts you to confirm whether or not you wish to reject the changes you have made and then reinstates the most recently saved values.

Fronius Registers

Fronius Register

These registers only apply to inverters. These registers are not relevant to Fronius String Controls and energy meters.

The Register tables can be found on the Fronius homepage or opened using the link: http://www.fronius.com/QR-link/0006

Inverter Status Code

Register ***F_Active_State_Code* (214)** displays the inverter status code which has just been generated. This may also be displayed on the inverter's display. This code is also displayed as an event flag in the inverter model. The displayed code remains active for as long the inverter has the corresponding status. Alternatively, the status can also be deleted by using register *F_Reset_All_Event_Flags*.

Deleting Event Flags and Status Codes

The event flags in the inverter models (101, 102, 103 and 111, 112, 113) remain active until the corresponding status is no longer present on the inverter. There are a few exceptional cases in which the event flags are not deleted. For this reason, it is possible to reset the event flags and the displayed status code by issuing the Modbus command.

1 Enter 0xFFFF in register ***F_Reset_All_Event_Flags* (215)**

The content of the following registers is deleted:
- *F_Active_State_Code* (214)
- *Evt1*
- *Evt2*
- *EvtVnd1* to *EvtVnd4*

Saving and Deleting Data

If the value 0xFFFF is written in the register ***F_Store_Data* (213)**, then all nominal values (ratings) for all inverters are saved on the Fronius Datamanager. These values can be changed in the corresponding registers of the Nameplate Model and the Basic Settings Model. This can be useful if, for example, no nominal values could be automatically determined for a device and you want to enter the values manually.

If you want to delete the saved values for a particular inverter, you must write the value 0xFFFF in the ***F_Delete_Data* (212)** register. The values are then only deleted for this inverter. The deletion can only ever be applied to the inverter with which there is currently communication.

Changing the Data Type

The data type for the data models for inverters and energy meters can be selected via the ***F_ModelType* (216)** register. It is possible to select either display as floating point numbers (float, standard) or as integers with scale factors (int+SF).

NOTE!

This setting only relates to the inverter model (inverter) and the meter model (energy meter).
All other models continue to use integers and scale factors.
This setting functions in the same way as the web interface Modbus settings – SunSpec model type.

Setting options:
- Float = 1 (standard): Inverter model 111, 112, or 113; meter model 211, 212, or 213
- int+SF = 2: Inverter model 101, 102, or 103; meter model 201, 202, or 203.

NOTE!

Since the different models have different numbers of registers, the register addresses of all the subsequent models also change when the data type is changed.

NOTE!

To avoid accidental changes, writing a value to setting F_ModelType must be confirmed by writing value 0x06 to the same register immediately after writing the type.
If the confirmation is omitted, changes will be reset after a few seconds.

System Totals

The following registers can be used to query power and energy data from all inverters connected to this Fronius Datamanager via Fronius Solar Net.
These values are displayed in Watt (W) or Watt hours (Wh) and do not require scale factors.
- ***F_Site_Power*** **(500–501)**: Power
- ***F_Site_Energy_Day*** **(502–505)**: Daily Energy
- ***F_Site_Energy_Year*** **(506–509)**: Yearly Energy
- ***F_Site_Energy_Total*** **(510–513)**: Total energy of the entire system.

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, Fronius String Control, 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 homepage or opened using the link: http://www.fronius.com/QR-link/0006

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 scaling 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 homepage or opened using the link: http://www.fronius.com/QR-link/0006

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

Fronius 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

Name	Value	Description
I_STATUS_NO_BUSINIT	9	No SolarNet communication
I_STATUS_NO_COMM_INV	10	No communication with inverter possible
I_STATUS_SN_OVERCURRENT	11	Overcurrent detected on SolarNet plug
I_STATUS_BOOTLOAD	12	Inverter is currently being updated
I_STATUS_AFCI	13	AFCI event (arc detection)

* 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 homepage or opened using the link:
http://www.fronius.com/QR-link/0006

Basic Settings Model (121)

Basic Settings Register

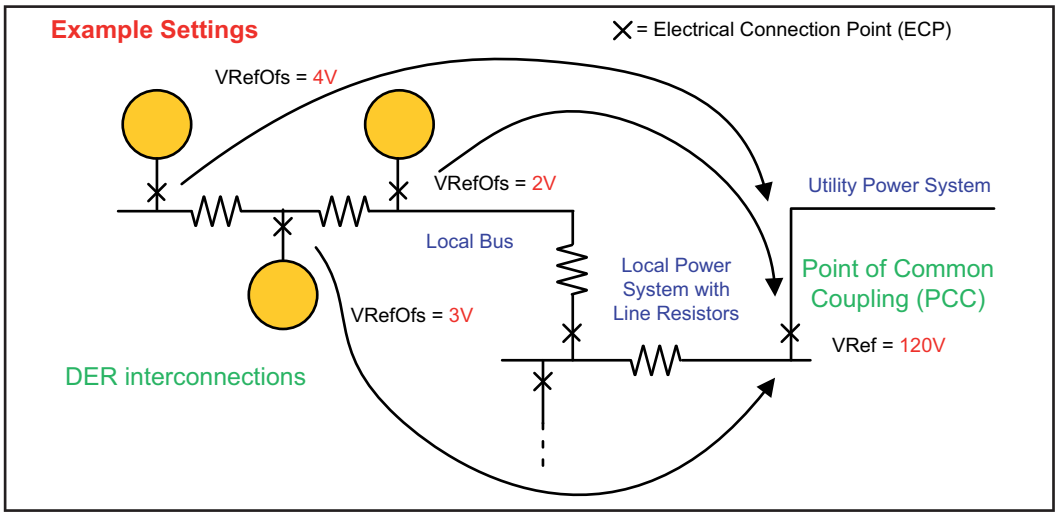
Start address:
- for "float" setting: ***40159***
- for "int+SF" setting: ***40149***

The Register tables can be found on the Fronius homepage or opened using the link: http://www.fronius.com/QR-link/0006

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).



Joint Connection Point

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).

The value is given in volts in the range of -20 (0xFFEC) to 20 (0x0014).

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: Setting 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.

Extended Measurements & Status Register

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

The Register tables can be found on the Fronius homepage or opened using the link:
http://www.fronius.com/QR-link/0006

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 homepage or opened using the link:
http://www.fronius.com/QR-link/0006

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 ECPConn 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)
In register *WMaxLimPct* you can enter values between 0% and 100%. Depending on the inverter's software version, values below 10 may force the inverter into standby (no grid power feed operation).
The values limit the device's maximum possible output power and therefore may not necessarily affect the real-time 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!

Proceed as follows to change values when an operating mode is active (e. g., when setting a different power limit or return time):
► Enter the new value into the relevant register
► Restart the operating mode using register *WMaxLim_Ena*

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 non-supported "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., 30 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 Reduction Has Been 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

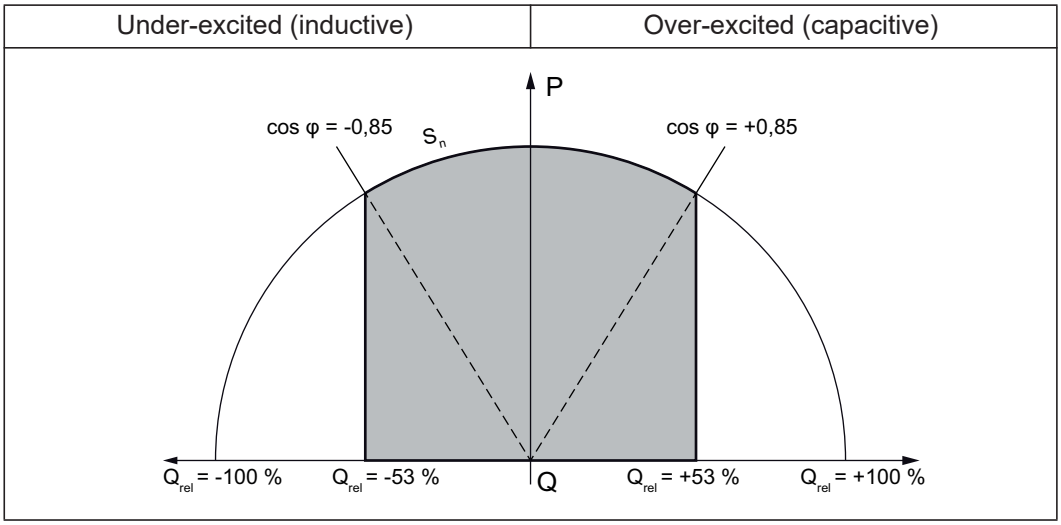
In principle, reactive power operation is limited by the maximum output current (the maximum apparent power) and by the operative reactive power limit of the inverter:

- Fronius IG Plus, CL cos phi = 0.85, VArrel = 100%
- Fronius Galvo cos phi = 0.85, VArrel = 53%
- Fronius Symo cos phi = 0.7, VArrel = 71%.

NOTE!

Due to the current technical conditions, only a cos phi up to a maximum of ±0. 80 can be specified per Modbus. In some circumstances, however, VAr_{rel} specifications may demand a lower value.

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



Reactive Power and Power Factor

Legend:

W	Power	VAr_{max}	Nominal reactive power
W_{max}	Nominal power	VAr_{rel}	Relative reactive power (VAr/VArmax)
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*.

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!

Proceed as follows to change values when an operating mode is active (e.
g., when setting a different reactive power value or return time):
► Enter the new value into the relevant register.
► Restart the operating mode using register *VArPct_Ena*.

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 up to two DC inverter inputs.

If the inverter has two 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”.

If the inverter only has one DC input, all values for the second string are set to “not implemented” (from register 2_DCA). The description of the second input (register 2_IDStr) appears as “not supported” in this case. The values for the first (and only) input are displayed normally.

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 homepage or opened using the link: http://www.fronius.com/QR-link/0006

Basic Storage Control Model (124)

General

This model is only available for Fronius Hybrid inverters.

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(MaxChaRte, MaxDisChaRte)
- If energy storage is not available, the register feeds back a value of 0.

ChaState
- Energy storage charge level in %:
  Estimated_Capacity_Remaining [Wh] / 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

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.

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)

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

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 memory 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 (Fronius system monitoring - Settings - DNO Editor - Battery charge). 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 Fronius Solar 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 homepage or opened using the link: http://www.fronius.com/QR-link/0006

String Combiner Model (403)

String Combiner Register	The Register tables can be found on the Fronius homepage or opened using the link: http://www.fronius.com/QR-link/0006

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 scaling 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 240.

The Register tables can be found on the Fronius homepage or opened using the link:
http://www.fronius.com/QR-link/0006

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 homepage or opened using the link:
http://www.fronius.com/QR-link/0006

String Combiner Event Flags

String Combiner Event Flags

Name	Event Flags
LOW_VOLTAGE	0x00000001
LOW_POWER	0x00000002
LOW_EFFICIENCY	0x00000004
CURRENT	0x00000008
VOLTAGE	0x00000010
POWER	0x00000020
PR	0x00000040
DISCONNECTED	0x00000080
FUSE_FAULT	0x00000100
COMBINER_FUSE_FAULT	0x00000200
COMBINER_CABINET_OPEN	0x00000400
TEMP	0x00000800
GROUNDFAULT	0x00001000
REVERSED_POLARITY	0x00002000
INCOMPATIBLE	0x00004000
COMM_ERROR	0x00008000
INTERNAL_ERROR	0x00010000
THEFT	0x00020000
ARC_DETECTED	0x00040000

FRONIUS INTERNATIONAL GMBH
Froniusstraße 1, A-4643 Pettenbach, Austria
E-Mail: sales@fronius.com
www.fronius.com

Under **www.fronius.com/contact** you will find the addresses
of all Fronius Sales & Service Partners and locations