

Fronius Modbus Steuerung

Speichersteuerung eines BDY Speichers an einem Fronius Gen24 Inverter über Modbus

Meine Anlage: Fronius Symo Gen24 10kW (SW 1.11.6-0) mit Smart Meter und BYD HVS 7.7 Speicher

Steuerung über ioBroker Modbus Adapter

Ziel: neben der Darstellung der Ist-Werte (welches ich bisher mit Erfolg über den Fronius Adapter V. 1.1.1 gemacht habe) will ich die Ladung des Speichers beeinflussen, um a) netzdienlich und b) unter Beachtung der 70% Einspeisegrenze zu steuern. Dies ist bei Fronius derzeit wohl nur über Modbus möglich.

Verwendung des Modbus Adapters in Version 3.3.6

Protokolle	2 von 2 Adapter aus dieser Gruppe installiert
ModBus	ModBus Verbindung Slave oder Master

Als Erstes ist der Modbus in der Konfiguration des Fronius Gen24 freizugeben. Dies erfolgt auf der Techniker-Ebene der Einstellungen. Das Passwort ist notwendig oder ihr müsst euch die Einstellungen vom Solateur machen lassen. Neben der Freigabe der Schnittstelle ist auch die Freigabe der Steuerung notwendig. Ich habe mich für das Sunspec Model int + SF entschieden.

Die Scaling Factoren (SF) werden vom Modbus Adapter bei entsprechender Einrichtung automatisch beachtet (s.u.)

Die Modbus RTU Schnittstellen werden nicht verändert Wir verwenden nur TCP

Modbus RTU-Schnittstelle 1

Master Slave Deaktiviert

Slave als Modbus TCP

Slave als Modbus TCP

Modbus-Port * 502
SunSpec Model Type * **int + SF**

Zähleradresse * 200

Wechselrichter-Steuerung über Modbus

Steuerung einschränken

Nun zur Konfiguration der Instanz des ioBroker Modbus Adapters unter Reiter Allgemein

Modbus Adapter Default Einstellungen:

Verbindungsparameter:	
TCP/Serial RTU:	TCP
Partner IP Adresse:	127.0.0.1
Port:	502
Geräte ID:	1
Mehrere Geräte-IDs:	<input type="checkbox"/>
Typ:	Master

Allgemein	
Alias benutzen:	<input checked="" type="checkbox"/>
Direkte Adressen benutzen (bei Alias):	<input type="checkbox"/>
Die Adressen nicht auf 16 Bits ausrichten:	<input type="checkbox"/>
Nicht Mehrere Register schreiben verwenden:	<input type="checkbox"/>
Aufrunden Real auf:	2
Poll delay:	1000 ms
Reconnect-Zeit:	60000 ms
Read timeout:	5000 ms
Pulsetime:	1000 ms
Wartezeit:	50 ms
Max Leseanforderungslänge (float)	100 Register
Max Leseanforderungslänge (booleans)	128 Register
Schreibintervall:	ms
Unveränderte Zustände aktualisieren:	<input type="checkbox"/>
Adresse nicht in ID aufnehmen:	<input type="checkbox"/>
Punkte in ID beibehalten:	<input type="checkbox"/>

meine Einstellungen

Verbindungsparameter:	
TCP/Serial RTU:	TCP
Partner IP Adresse:	127.0.0.1
Port:	502
Geräte ID:	1
Mehrere Geräte-IDs:	<input checked="" type="checkbox"/>
Typ:	Master

Allgemein	
Alias benutzen:	<input type="checkbox"/>
Direkte Adressen benutzen (bei Alias):	<input type="checkbox"/>
Die Adressen nicht auf 16 Bits ausrichten:	<input type="checkbox"/>
Nicht Mehrere Register schreiben verwenden:	<input type="checkbox"/>
Aufrunden Real auf:	2
Poll delay:	1000 ms
Reconnect-Zeit:	60000 ms
Read timeout:	5000 ms
Pulsetime:	1000 ms
Wartezeit:	50 ms
Max Leseanforderungslänge (float)	100 Register
Max Leseanforderungslänge (booleans)	128 Register
Schreibintervall:	ms
Unveränderte Zustände aktualisieren:	<input type="checkbox"/>
Adresse nicht in ID aufnehmen:	<input type="checkbox"/>
Punkte in ID beibehalten:	<input type="checkbox"/>

Eigene IP des Fronius einsetzen

Anm.: Die Benutzung des Alias habe ich vermieden, da bei der Verwendung der Register mit den Skalierungsfaktoren diese dennoch mit ihrer ursprünglichen Registeradresse zu verwenden sind
 Mehrere Geräte-IDs werden benötigt, um auch auf das Smart-Meter zugreifen zu können. Dies hat die ID 200.

Nun ist die Fronius Dokumentation der Modbus Register hilfreich:

.. Geht auf: <https://www.fronius.com/de/sol...lateure-partner/downloads> und gebt in das Suchfeld "gen 24 Modbus" ein - die ZIP Datei, die euch dann angezeigt wird, enthält die Registerdoku.

Evtl. funktioniert auch diese Links hier direkt: <https://www.fronius.com/~/-/down...bus-api-external-docs.zip>
 (aus Post #32,Autor: soly3141592 – Danke)

Wer mehr über die Fronius Implementierung des Modbus erfahren will liest hier nach:

Fronius Dokument mit dem Namen: Fronius Gen24 Modbus TCP RTU

<https://www.fronius.com/~/-/downloads/Solar%20Energy/Operating%20Instructions/42%2C0410%2C2649.pdf>

Wie üblich sind die Registernummern bei der Eingabe im Adapter um 1 zu erniedrigen (vermutlich beginnt die Registerbezeichnung mit Register 1, die Registeradresse aber als unsInt mit 0)

Start	End	Size	R/W	Functioncode	Name	Description	Type	Units	Scale Factor	Range of val
40003	40003	1	R	0x03	ID	Well-known value. Uniquely identifies this as a sunspec model common (1)	uint16			1
40004	40004	1	R	0x03	L	Length of sunspec model common (1)	uint16	Registers		65
40005	40020	16	R	0x03	Mn	Manufacturer	string			Fronius
40021	40036	16	R	0x03	Md	Device	string			e.g. 'PRIMO GEN24 6.00 kW', 'SYMO GEN24 10.00 kW', ...
40037	40044	8	R	0x03	Opt	Options	string			not supported
40045	40052	8	R	0x03	Vr	SW version of inverter	string			e.g. 1.8.10-0

40084	40084	1	R	0x03	W	AC Power	int16	W	W_SF	
40085	40085	1	R	0x03	W_SF		sunssf			auto-scaled on W

Auszug aus dem Dokument "Gen24_Primo_Symo_Inverter_Register_Map_Int&SF_storage.xlsx"

Das Register 40021 (mit Adresse 40020) habe ich als Leseregister gewählt; hier ist als Stringdarstellung die korrekte Funktion des Adapters gut zu erkennen.

Nach den Einstellungen im ioBroker Adapter sind die Holding Register einzutragen. (In den anderen Reitern werden keine Eingaben benötigt.)

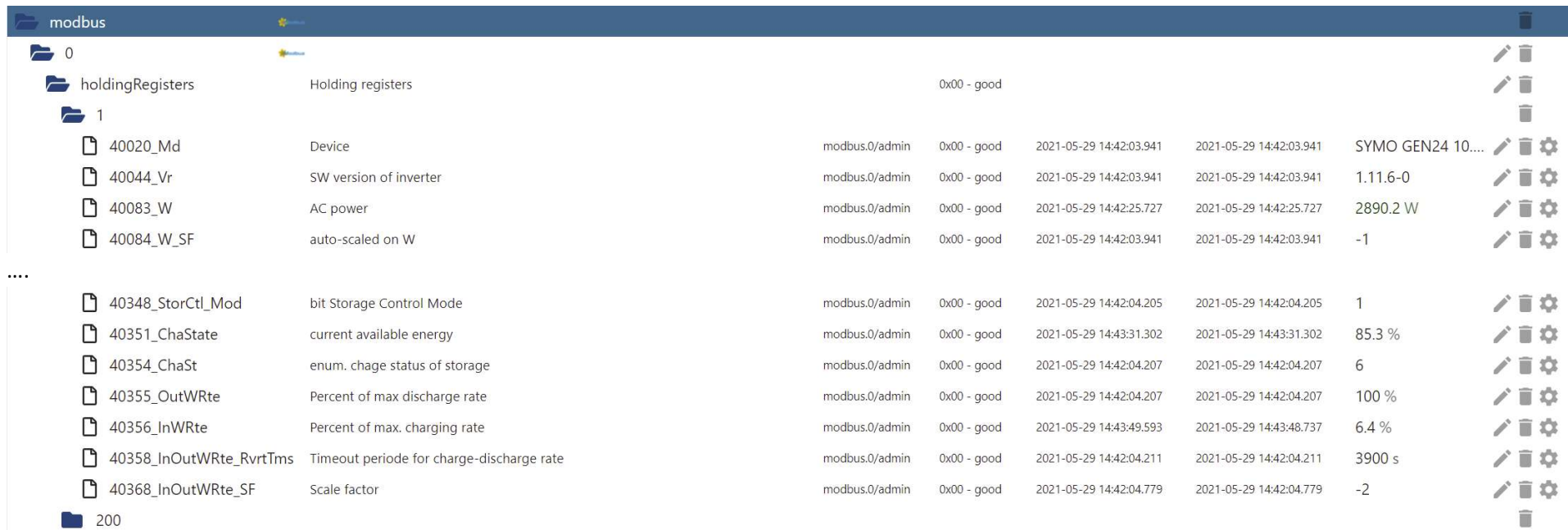
Die Eintragungen der von mir verwendeten Register könnt ihr aus der anhängenden Datei durch einfaches Kopieren in das durch + geöffnete Fenster in den Adapter laden. Das sollte dann so aussehen:

Instanzeinstellungen: modbus.0

Adresse	Slave-ID	Name	Beschreibung	Einheit	Typ	Länge	Faktor	Offset	formula	Rolle	Raum	Poll	WP	CW	SF
40020	1	Md	Device		String (Zero-end)	16				value		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40044	1	Vr	SW version of inverter		String (Zero-end)	8				value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40083	1	W	AC power	W	Signed 16 bit (Big Endian)	1	1	0	x * Math.pow (10, sf [40084]);	value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40084	1	W_SF	auto-scaled on W		Signed 16 bit (Big Endian)	1	1	0		value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
40093	1	WH	AC Energy	Wh	Unsigned 32 bit (Big Endian)	2	1	0	x * Math.pow (10, sf [40094]);	value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40095	1	WH_SF	auto-scaled on WH		Signed 16 bit (Big Endian)	1	1	0		value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
40100	1	DCW	DC Power	W	Unsigned 16 bit (Big Endian)	1	1	0	x * Math.pow (10, sf [40101]);	value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40101	1	DCW_SF	auto-scaled on DCW		Signed 16 bit (Big Endian)	1	1	0		value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
40102	1	TmpCab	Cabinet Temperature	°C	Unsigned 16 bit (Big Endian)	1	0.1	0		value		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- (1) Einfaches Register zum Testen der prinzipiellen Adapterfunktion
- (2) Register beinhaltet dem Scaling Factor für (4) und ist daher als SF aktiviert (3)
- (3) Register ist SF
- (4) Register mit AC Ausgangsleistung des Inverters mit variablem Scaling Factor
- (5) Mit der Formel wird über den SF automatisch vom Adapter die Umrechnung durchgeführt
(in der Formel wird auch bei Verwendung von Alias die "original" Registeradresse benötigt; daher meine Entscheidung kein Alias zu verwenden)

Nachdem nun alle Register eingerichtet sind werden beim Speichern der Instanz die Datenpunkte in ioBroker angelegt: (hier Auszug!)



Folder	Register Name	Description	Status	Value
0	holdingRegisters	Holding registers	0x00 - good	
1	40020_Md	Device	modbus.0/admin 0x00 - good	2021-05-29 14:42:03.941
	40044_Vr	SW version of inverter	modbus.0/admin 0x00 - good	2021-05-29 14:42:03.941 1.11.6-0
	40083_W	AC power	modbus.0/admin 0x00 - good	2021-05-29 14:42:25.727 2890.2 W
	40084_W_SF	auto-scaled on W	modbus.0/admin 0x00 - good	2021-05-29 14:42:03.941 -1
....				
	40348_StorCtl_Mod	bit Storage Control Mode	modbus.0/admin 0x00 - good	2021-05-29 14:42:04.205 1
	40351_ChaState	current available energy	modbus.0/admin 0x00 - good	2021-05-29 14:43:31.302 85.3 %
	40354_ChaSt	enum. chage status of storage	modbus.0/admin 0x00 - good	2021-05-29 14:42:04.207 6
	40355_OutWRte	Percent of max discharge rate	modbus.0/admin 0x00 - good	2021-05-29 14:42:04.207 100 %
	40356_InWRte	Percent of max. charging rate	modbus.0/admin 0x00 - good	2021-05-29 14:43:49.593 6.4 %
	40358_InOutWRte_RvrtTms	Timeout periode for charge-discharge rate	modbus.0/admin 0x00 - good	2021-05-29 14:42:04.211 3900 s
	40368_InOutWRte_SF	Scale factor	modbus.0/admin 0x00 - good	2021-05-29 14:42:04.779 -2
200				

Das eigentliche Steuer des Speichers:

Ich möchte die Ladeleistung begrenzen, so dass der Speicher nicht gleich morgens mit der ersten PV Energie vollgepumpt wird.
(Die Entladeleistung begrenze ich nicht)

Dazu muss über das Register StorCtl_Mod durch Setzen von Bit 0 (schreiben einer 1) diese Funktion freigegeben werden (bei Begrenzen auch der Entladeleistung Bit 0 + 1 setzen – schreibe 3).

Die Ladeleistung wird über Register InWRte in Prozent der maximale möglichen Lageleistung, die in Register WChaMax steht, vorgegeben. WChaMax ist bei mir 7800W auf Grund meiner Speichergröße. Geladen wird im Beispiel mit 6,4% also $7800W * 6,4\% = 499,2W$. Eine Vorgabe direkt in Watt ist nicht möglich.

Das Register InWRte hat auch wieder wie oben schon erwähnt einen Scale Factor in Register InOutWRte_SF. Er ist hier -2 was 10 hoch -2 also 1/100 entspricht. Für 100% steht hier also eigentlich 10000. Das ist durch die automatische Umrechnung nicht erkennbar. Er **muss** jedoch beim Schreiben eines Wertes beachtet werden, da hier keine automatische Umrechnung erfolgt. (Vielleicht mal noch ein Thema für die Programmierer des Adapters?) Ich schreibe hier im Beispiel

also 640 in das Register. Diese Zahl ist beim Schreiben in dem Datenpunkt auch kurz sichtbar, bevor sie dann durch die Bestätigung des Inverters überschrieben wird und hierbei durch die Umrechnung dann wieder 6.4 erscheint.

Das Register InOutWRte_RvrTms erlaubt die Verwendung eines Timeout Mechanismus. Das Register hat einen Default Wert von 0. Damit ist der Mechanismus deaktiviert. Ich habe es auf 3900 (Sekunden) gesetzt, was 65 Minuten entspricht. Damit muss mindestens ein Mal in diesem Zeitabstand auf das Register InWRte geschrieben werden, sonst wird die Fremdsteuerung der Ladeleistung deaktiviert und geht auf ihren Default Wert von 100% zurück. Ich finde diesem Sicherheitsmechanismus prima – wenn sich irgendetwas in meiner Steuerung aufhängt, sei es mein JavaScript oder der Modbus Adapter, so wird die Speicherladung in den Default Zustand versetzt.

29.05.2021