



Dokumentation
der TwinCAT Library
„Lynus Energy ModbusTCP“

Inhalt

Change Log	3
Systemvoraussetzungen.....	4
Einführung	4
Funktionsweise.....	5
Beispiel	5
Verknüpfungen im Dashboard	6
Installation der Bibliothek	7
Verwendete Datentypen.....	8
Funktionsblöcke	11
FB_ECS_Alfen_NG9xx_Platform.....	11
FB_ECS_Alpitronic_Hypercharger	13
FB_Gen24_Hybrid_Inverter.....	15
FB_Symo_PV_Inverter_FLOAT_1_DM_1_Inverter	18
FB_Symo_PV_Inverter_FLOAT_1_DM_30_Inverter	19
FB_Symo_PV_Inverter_INT_1_DM_1_Inverter	21
FB_Symo_PV_Inverter_INT_1_DM_30_Inverter	21
FB_HP_Soltop_M_Tec_Modular	23
FB_HP_Viessmann_Vitogate300_1Stage	25
FB_Xelectrix_Powerbox.....	27
FB_HP_CTA_Optiplus3_Modular	29

Change Log

Version	Datum	Änderungen
1.0	16.09.2022	Initialversion EfficientIO

Systemvoraussetzungen

Die EfficientIO Bibliothek funktioniert auf allen Beckhoff Steuerungen welche TwinCat 3.1 4022.0 oder höher installiert haben. Für diese Bibliothek werden nur Beckhoff Funktionen verwendet, welche standardmäßig bei der Installation von TwinCat 3.1 dabei sind. Zusätzlich dazu braucht es noch die TC3 TF6250 Function (Modbus TCP).

Diese Bibliothek funktioniert nur in Verbindung mit der „EfficientIO Communicator“ Bibliothek und einer aktiven Verbindung zu einem erzeugten Projekt im EfficientIO Dashboard. Gibt es keine aktive Verbindung zu einem Cloud Projekt, stoppen die Funktionalitäten dieser Bibliothek automatisch nach 11 Tagen betrieb.

Diese Bibliothek verlangt die Installation der „Lynus Standard“ Bibliothek.

Einführung

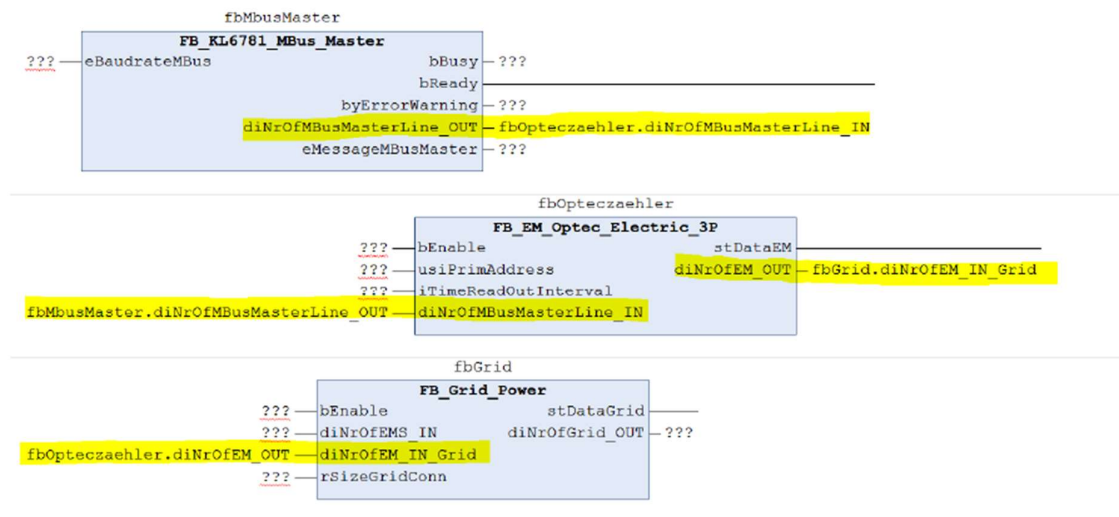
In der Lynus Energy Modbus TCP Bibliothek befinden sich Funktionsblöcke und Datentypen, welche vor allem mit dem Thema Energie zu tun haben. Man findet hier ausschließlich Funktionsblöcke, die die spezifische Schnittstelle ModbusTCP verwenden. Unter anderem findet man hier Funktionsblöcke zu Wärmepumpen, Ladestationen, Energiemessungen, Wechselrichtern usw.

Funktionsweise

Die meisten Funktionsblöcke sind so aufgebaut, dass Sie die Daten wie oben genannt über verschiedene Schnittstellen aufnehmen und dann intern auf eine eigene Schnittstelle legen. Diese Daten werden dann intern weitergegeben und stehen somit anderen Funktionsblöcken zur Verfügung. Somit hat man z.B. die Möglichkeit die Energiedaten einer klassischen M-Bus Messung an einen spezifischen «Netzanschluss» oder «PV-Anlage» Funktionsblock weiterzugeben und diese dann auch als solche zu spezifizieren. Somit ist man extrem flexibel, was den Datenaustausch unter den Funktionsblöcken betrifft, hat aber gleichzeitig schon fix fertige Funktionsblöcke die Plug&Play verwendet werden können. Diese Übergabe der Daten kann über eine einfache Nummer, welche das System selber berechnet, erledigt werden.

Beispiel

Hier wird das ganze anhand einer Energiemessung eines M-Bus Elektrozählers erklärt, welcher im realen Umfeld die Leistung am Netzanschluss misst. Die Bezeichnungen unterscheiden sich voneinander im Text, beinhalten aber immer einmal die Bezeichnung _IN oder _OUT. _IN bedeutet, dass hier die Nummer übergeben kann, von welchem Funktionsbaustein dieser hier die Daten empfangen soll. _OUT bedeutet, dass dieser Funktionsbaustein die Daten versendet und anderen am _IN Eingang bereitstellt.



Hier sieht man recht gut, wie der M-Bus Masterbaustein, welcher die Kommunikation zur Beckhoff M-Bus Klemme übernimmt über den Ausgang ,diNrOfMbusMasterLine_OUT' die Daten weitergibt an den M-Bus Zähler von Optec über den Eingang ,diNrOfMbusMasterLine_IN'. Dieser Baustein gibt die Daten dann über den Ausgang ,diNrOfEM_OUT' an den Eingang ,diNrOfEM_IN_Grid' des Netzanschluss Funktionsblockes. Diese Ein und Ausgänge müssen bei allen Funktionsblöcken richtig zugewiesen werden, um eine störungsfreie Funktion zu gewährleisten.

Verknüpfungen im Dashboard

Bestimmte Geräte im Dashboard benötigen bestimmte Verknüpfungen zu Variablen damit diese auch ordnungsgemäß funktionieren. Um es dem Kunden einfacher zu machen haben bereits alle Funktionsblöcke aus dieser Beschreibung, die zu einem spezifischen Device im Dashboard gemappt werden können, alle nötigen Variablen dazu. Diese sind durch das klassische Lynus Kommentar im Funktionsbaustein markiert. Diese werden dann automatisch in das Projekt hochgeschickt sobald die SPS über den Communicator mit dem Backend verbunden ist. Dazu muss allerdings mit dem TMC File des jeweiligen Projektes gearbeitet werden, um die Variablen direkt aus den Funktionsblöcken verwenden zu können. Siehe dazu mehr in der Beschreibung für den „EfficientIO Communicator“.

Nötiges Variablen-Mapping im Dashboard

Create ML Model

CHARTS (7)ML MODEL (3)

Model Information

Mapping

Settings

GENERALPV SYSTEMSGRIDEV CHARGING STATIONS

Input

Output

START DATE

Heartbeat

Required
32 Bit Value

State Enable EMS (Switch)

Error/Warning (Show Event)

Required
0-1

Required
0= OK 1 = Warning 2= Error

State Allowed charging from Grid (Swit...

Energy optimization Ready (Show Event)

Required
0-1

Required
0-1

State Main Fuse (Switch)

Operation Mode EMS (DropDown)

Required
0-1

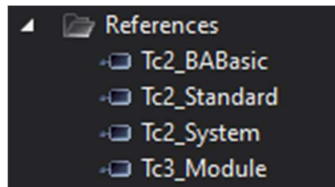
Required
0 = All off; 1 = Self Consumption; 2 = Peak Load capping; 3 = Load management;

Variablen im Funktionsblock, markiert über das Kommentar

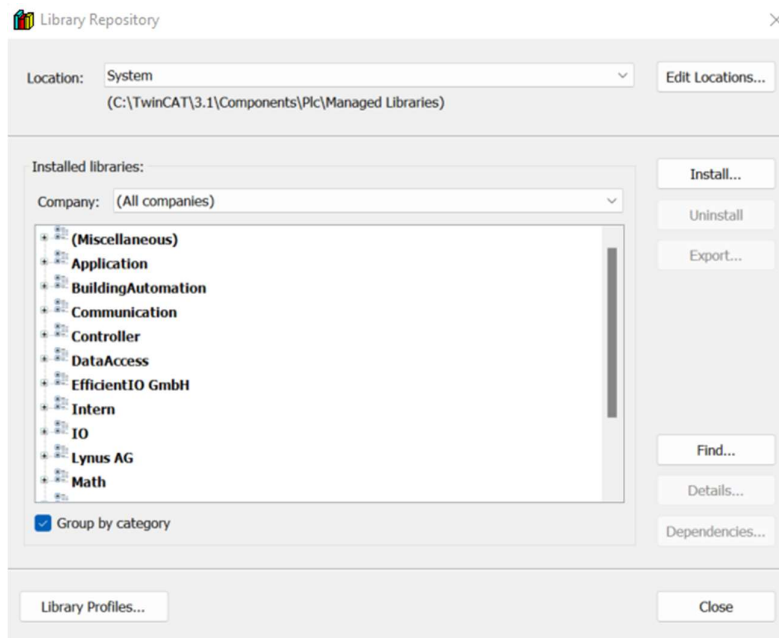
FUNCTION_BLOCK FB_EMS						
Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar	
bEnable	BOOL				{#lynus.ag#} () //Enable and dissable this function	
dIntOfEM_IN_ECS	DINT				Number of the electric meter for incoming power data. (Its for the complete consumption of all charging stations)	
stSetupEMS	ST_Setup_EMS			STRUCT (uiUpdateTime := 15)	{#lynus.ag#} () //Setup EMS	
eOperationMode	E_OperatingMode_EMS				{#lynus.ag#} () //Operation Mode EMS	
eSpeedModeBackendController	E_SpeedModeBackendController_EMS				Speed for the Controller what discharge the Battery when the commands come from the Backend	
bActiveStop	BOOL				{#lynus.ag#}	

Installation der Bibliothek

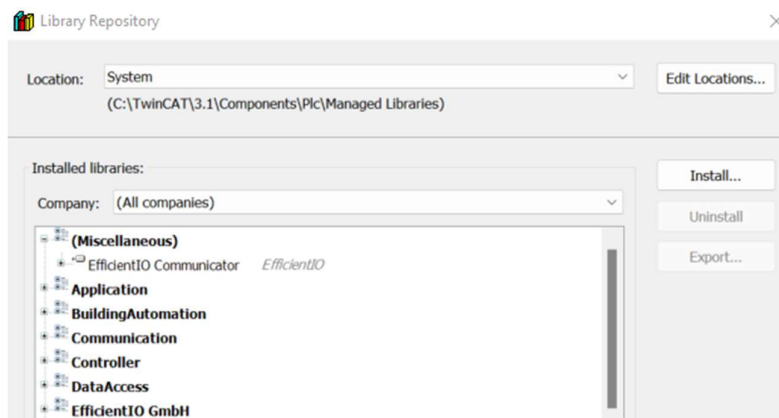
Im SPS Projekt rechtsklick auf References und dann klick auf Bibliotheksrepository:



Danach auf installieren klicken:



Die EfficientIO Bibliothek im abgespeicherten Pfad auswählen und dann auf Öffnen klicken. Nach erfolgreicher Installation erscheint die Bibliothek im Ordner „(Sonstige)“ oder „EfficientIO GmbH“ bzw. „Lynus AG“.



Verwendete Datentypen

Anbei findet man die Beschreibung der Datentypen, welche ausschließlich in dieser Bibliothek verwendet werden. Alle anderen Datentypen findet man in der „Lynus Standards“ Bibliothek.

Achtung: Es ist zu beachten, dass je nach verwendetem Funktionsblock oder dessen Konfiguration, nicht immer alle Variablen der Datentypen mit Werten belegt sind.

ST_Gen24_Out	
<i>Diese Struktur enthält gängige Ausgabewerte eines Fronius Gen24 Hybrid Wechselrichters.</i>	
Name	Beschreibung
bError	Ist True wenn der Inverter einen Fehler hat
bFunctionIsActive	Zeigt an wenn die internen Steuerfunktionen aktiv sind.
rSetpointMaxCh_Disch	Zeigt die maximale Lade und Entladeleistung des Inverters in kW an.
eOperatingStateInverter	Zeigt den aktuellen Status des Inverters an.
eChargeStateInverter	Zeigt den Ladestatus der Batterie an.
eErrorState	Zeigt den Fehlerstatus des Inverters an.

ST_Symo_PV_OutData	
<i>Diese Struktur enthält gängige Ausgabewerte eines Fronius Symo PV Wechselrichters.</i>	
Name	Beschreibung
bError	Ist True wenn der Inverter einen Fehler hat
bFunctionIsActive	Zeigt an wenn die internen Steuerfunktionen aktiv sind.
IrPowerAcInverter	Gibt die aktuelle AC Leistung des Inverters in KW aus.
eOperatingStateInverter	Zeigt den aktuellen Status des Inverters an.
eErrorState	Zeigt den Fehlerstatus des Inverters an.

ST_Xelextrix_Powerbox_OutData	
<i>Diese Struktur enthält gängige Ausgabewerte eines Xelextrix Batteriespeichersystemes.</i>	
Name	Beschreibung
bError	Ist True wenn der Inverter einen Fehler hat
bFunctionIsActive	Zeigt an wenn die internen Steuerfunktionen aktiv sind.
eGridState	Zeigt den aktuellen Status der Netzmessung an.
eOperatingState	Zeigt den aktuellen Status des Inverters an.
eErrorState	Zeigt den Fehlerstatus des Inverters an.

E_Gen24_ChargeState	
<i>Enum welches den aktuellen Ladezustand eines Fronius Gen24 Hybrid Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Fronius Dokumentation.
eEmpty	Siehe Fronius Dokumentation.
eDischarge	Siehe Fronius Dokumentation.
eCharging	Siehe Fronius Dokumentation.
eFull	Siehe Fronius Dokumentation.

eHolding	Siehe Fronius Dokumentation.
eTesting	Siehe Fronius Dokumentation.

E_Gen24_Error	
<i>Enum welches den aktuellen Fehlerzustand eines Fronius Gen24 Hybrid Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoError	Momentan liegen kein Fehler vor.
eModbusTCPError	Fehler beim lesen oder schreiben von Modbus TCP Daten von oder an den Inverter. Es handelt sich um eine Kommunikationsstörung.
eGen24Error	Der Inverter selber hat eine Störung.
eGen24TooBigValueReceived	Keine plausiblen und gültigen Werte vom Inverter empfangen.

E_Gen24_OperatingState	
<i>Enum welches den aktuellen Modus eines Fronius Gen24 Hybrid Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Fronius Dokumentation.
eSleeping	Siehe Fronius Dokumentation.
eStartingUp	Siehe Fronius Dokumentation.
eTrackingPowerPoint	Siehe Fronius Dokumentation.
eForcedPowerReduction	Siehe Fronius Dokumentation.
eShuttingDown	Siehe Fronius Dokumentation.
eFaults	Siehe Fronius Dokumentation.
eStandby	Siehe Fronius Dokumentation.

E_Symo_PV_Error	
<i>Enum welches den aktuellen Fehlerzustand eines Fronius Symo PV Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoError	Momentan liegen kein Fehler vor.
eModbusTCPError	Fehler beim lesen oder schreiben von Modbus TCP Daten von oder an den Inverter. Es handelt sich um eine Kommunikationsstörung.
eSymo10Error	Der Inverter selber hat eine Störung.
eSymo10TooBigValueReceived	Keine plausiblen und gültigen Werte vom Inverter empfangen.

E_Symo_PV_OperatingState	
<i>Enum welches den aktuellen Modus eines Fronius Gen24 Hybrid Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Fronius Dokumentation.
eSleeping	Siehe Fronius Dokumentation.
eStartingUp	Siehe Fronius Dokumentation.
eTrackingPowerPoint	Siehe Fronius Dokumentation.
eForcedPowerReduction	Siehe Fronius Dokumentation.

eShuttingDown	Siehe Fronius Dokumentation.
eFaults	Siehe Fronius Dokumentation.
eStandby	Siehe Fronius Dokumentation.

E_Xelectrix_BatteryState	
<i>Enum welches den aktuellen Batterie Status eines Xelectrix Batteriesystemes anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Xelectrix Dokumentation.
eStartUp	Siehe Xelectrix Dokumentation.
eBalancing	Siehe Xelectrix Dokumentation.
eReady	Siehe Xelectrix Dokumentation.
eOperating	Siehe Xelectrix Dokumentation.
eError	Siehe Xelectrix Dokumentation.

E_Xelectrix_Error	
<i>Enum welches den aktuellen Fehlerzustand eines Xelectrix Batteriesystemes anzeigt.</i>	
Name	Beschreibung
eNoError	Momentan liegen kein Fehler vor.
eModbusTCPError	Fehler beim lesen oder schreiben von Modbus TCP Daten von oder an den Inverter. Es handelt sich um eine Kommunikationsstörung.
eXelectrixError	Das Xelectrix System selber hat eine Störung.

E_Xelectrix_GridState	
<i>Enum welches den aktuellen Zustand am Netzanschluss eines Xelectrix Batteriesystemes anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Xelectrix Dokumentation.
eIslandMode	Siehe Xelectrix Dokumentation.
eOnline	Siehe Xelectrix Dokumentation.
eError	Siehe Xelectrix Dokumentation.

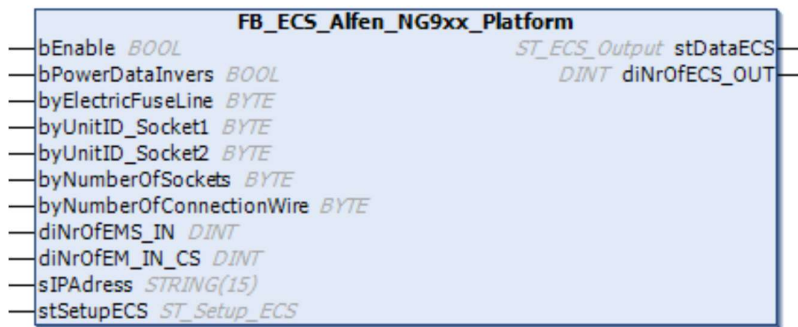
E_Xelectrix_OperatingState	
<i>Enum welches den aktuellen Modus eines Fronius Gen24 Hybrid Wechselrichters anzeigt.</i>	
Name	Beschreibung
eNoDataReceived	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Xelectrix Dokumentation.
eDischarge	Siehe Xelectrix Dokumentation.
eCharge	Siehe Xelectrix Dokumentation.
eIdle	Siehe Xelectrix Dokumentation.

Funktionsblöcke

Anbei findet man eine Beschreibung zu allen Funktionsblöcken welche sich in der „Lynus Energy ModbusTCP“ Bibliothek befinden. Hier findet man Funktionen, welche über die Modbus TCP Schnittstelle kommunizieren. Dies können Geräte sein wie Wechselrichter, Energiemessgeräte oder auch E-Ladestationen.

FB_ECS_Alfen_NG9xx_Platform

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einer Ladestation der Firma Alfen aufzubauen. Konkret handelt es sich um die Ladestationen welche bei Alfen auf der NG9xx Plattform basieren. Es können Ladestationen mit nur einer Anschlussleitung und einem Stecker für das Fahrzeug geregelt werden, es funktioniert aber auch mit Ladestationen welche über eine oder 2 Anschlussleitungen verfügen und 1 oder 2 Stecker für das Fahrzeug besitzen. Sollte die Ladestation 2 Zuleitungen haben und 2 Stecker, ist es wichtig in der Software bei der Maximalen Leistung der Ladestation in kW die eigentliche Leistung der Ladestation * 2 anzugeben. Die Version mit welcher die Ladestation in Lynus implementiert wurde lautet: 2.3.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
bPowerDataInvers	BOOL	Leistungsdaten vom verknüpften Energiezähler werden invertiert.
byElectricFuseLine	BYTE	Hier kann die Absicherungsline der Ladestation angegeben werden bei welcher diese angeschlossen ist und welche das EMS zusätzlich überwachen soll. Es sind bis zu separate 30 Absicherungslinien pro EMS möglich.
byUnitID_Socket1	BYTE	Identifizierungsnummer im des 1 Steckers im TCP Netzwerk für das Laden am Fahrzeug.
byUnitID_Socket2	BYTE	Identifizierungsnummer des 2 Steckers im TCP Netzwerk für das

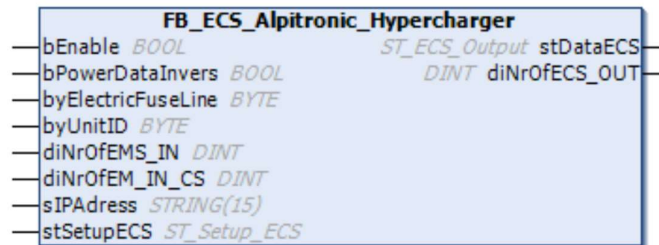
		Laden am Fahrzeug. Nur nötig wenn die Ladestation über 2 Stecker zum Laden von Fahrzeugen verfügt.
byNumberOfSockets	BYTE	Angabe über wie viele Stecker die Ladestation verfügt zum laden von Fahrzeugen. Erlaubte Werte zwischen 1 und 2.
byNumberOfConnectionWire	BYTE	Angabe über wie viele Anschluss, bzw. Versorgungsleitungen die Ladestation verfügt. Erlaubte Werte zwischen 1 und 2. Bei 2 nicht vergessen die eigentliche Leistung in kW der Ladestation pro Ladestecker mit 2 zu multiplizieren.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Ladestation regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.
diNrOfEM_IN_CS	DINT	Angabe der Nummer der Elektromessung, welche die Leistung der Ladestation misst. Siehe jeweiligen Ausgang der Elektromessung.
siPAdress	STRING(15)	IP Adresse der Ladestation
stSetupECS	ST_Setup_ECS	Struktur mit allen relevanten Einstellungen der Ladestation.

Ausgänge

Name	Typ	Beschreibung
stDataECS	ST_ECS_Output	Struktur mit allen relevanten Ausgangssignalen der Ladestation.
diNrOfECS_Out	DINT	Berechnete Nummer der Ladestation, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten der Ladestation an diese weiterzugeben.
lrTotalPowerECSRealTime	LREAL	Aktuelle Leistungsaufnahme der Ladestation in Echtzeit in kW. Kann an andere Funktionsblöcke übergeben werden.

FB_ECS_Alpitronic_Hypercharger

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einer Ladestation der Firma Alpitronic aufzubauen. Konkret handelt es sich um die DC Hypercharger welche bei Alpitronic über Modbus TCP kommunizieren. Die Ladestationen teilen über das Protokoll mit, über wie viele Ladepunkte sie verfügen. Pro Ladestation werden maximal 4 Ladepunkte unterstützt. Die Leistungen zu den einzelnen Ladepunkten verteilt die Ladestation selbstständig. Die Version mit welcher die Ladestation in Lynus implementiert wurde lautet : 1.3A vom Oktober 2020.



Eingänge

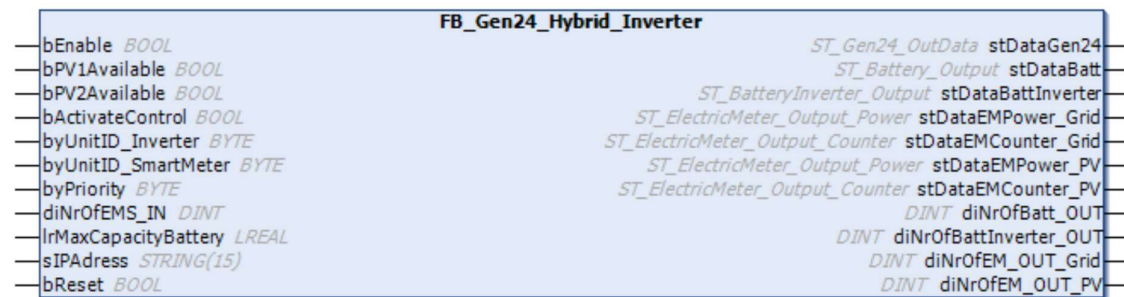
Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
bPowerDataInvers	BOOL	Leistungsdaten vom verknüpften Energiezähler werden invertiert.
byElectricFuseLine	BYTE	Hier kann die Absicherungsline der Ladestation angegeben werden bei welcher diese angeschlossen ist und welche das EMS zusätzlich überwachen soll. Es sind bis zu separate 30 Absicherungslinien pro EMS möglich.
byUnitID	BYTE	Identifizierungsnummer der Ladestation im TCP Netzwerk.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Ladestation regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.
diNrOfEM_IN_CS	DINT	Angabe der Nummer der Elektromessung, welche die Leistung der Ladestation misst. Siehe jeweiligen Ausgang der Elektromessung.
sIPAdress	STRING(15)	IP Adresse der Ladestation
stSetupECS	ST_Setup_ECS	Struktur mit allen relevanten Einstellungen der Ladestation.

Ausgänge

Name	Typ	Beschreibung
stDataECS	ST_ECS_Output	Struktur mit allen relevanten Ausgangssignalen der Ladestation.
diNrOfECS_Out	DINT	Berechnete Nummer der Ladestation, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten der Ladestation an diese weiterzugeben.
IrTotalPowerECSRealTime	LREAL	Aktuelle Leistungsaufnahme der Ladestation in Echtzeit in kW. Kann an andere Funktionsblöcke übergeben werden.

FB_Gen24_Hybrid_Inverter

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem Fronius Gen24 Hybrid Wechselrichter aufzubauen. Somit kann die Leistung welche in oder aus der Batterie fließt durch Lynus geregelt werden. Der Fronius Smart Meter, welcher für die Netzmessung zuständig ist, muss am selben Fronius Datenmanager angeschlossen sein, wie der Hybrid Inverter selber. Wenn auch Leistung von AC gekoppelten PV Anlagen in die Batterie fließen soll, muss das Laden vom Netz in den Einstellungen vom Inverter erlaubt werden. Ansonsten fließt nur die PV Leistung in die Batterie, welche an den Strings des Hybrid Wechselrichter selber angeschlossen sind. Wenn dieser Funktionsblock «Disabled» wird, ist der Inverter nicht aus, sondern schaltet um auf die Fronius interne Regellogik. Ab da fährt Fronius sein eigenes Energie Programm und ist nicht mehr über Lynus fremdgesteuert. Auch eine Abschaltung des Wechselrichter im Notstrombetrieb beim Unterschreiten einer bestimmten Batteriekapazität ist mit dieser Funktion nicht möglich, da dies die Kommunikationsschnittstelle zu Fronius nicht zulässt. Dies erledigt der Inverter eigenständig und kann somit nicht beeinflusst werden. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 1.1.2-0.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschaltet sondern fährt seine eigene interne Regellogik.
bPV1Available	BOOL	Auf True zu setzen wenn an PV String 1 des Inverters eine PV Anlage angeschlossen ist.
bPV2Available	BOOL	Auf True zu setzen wenn an PV String 2 des Inverters eine PV Anlage angeschlossen ist.
bActivateControl	BOOL	Wenn dieser Eingang zusammen mit bEnable gesetzt ist, dann regelt Lynus den Inverter. Wenn dieser Eingang auf False gesetzt wird, regelt der Inverter nach seiner eigenen internen Logik und ist nicht mehr Fremdgesteuert.

byUnitID_Inverter	BOOL	Identifizierungsnummer des Inverters im TCP Netzwerk.
byUnitID_SmartMeter	BYTE	Identifizierungsnummer des Smart Meters für die Netzmessung im TCP Netzwerk.
byPriority	BYTE	Gibt die aktuelle Priorität wieder, mit welcher der Batterie Wechselrichter vom EMS Funktionsbaustein abgearbeitet wird.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Ladestation regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.
IrMaxCapacityBattery	LREAL	Angabe der Batterie für die Maximalen Batteriekapazität in kWh.
slPAdress	STRING(15)	IP Adresse des Hybridwechselrichters
bReset	BOOL	Eingang um bei einem Fehler des Inverters die internen Warnung oder Fehler Ausgänge zurückzusetzen.

Ausgänge

Name	Typ	Beschreibung
stDataGen24	ST_Gen24_OutData	Struktur mit allen relevanten Ausgangssignalen des Fronius Gen24 Hybrid Inverters.
stDataBatt	ST_Battery_Out	Struktur mit allen relevanten Ausgangssignalen der Batterie.
stDataBattInverter	ST_BatteryInverter_Output	Struktur mit allen relevanten Ausgangssignalen des Batterie Inverters.
stDataEMPower_Grid	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung für den Netzanschluss.
stDataEMCounter_Grid	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung für den Netzanschluss.
stDataEMPower_PV	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung für die internen PV Strings des Hybridwechselrichters.
stDataEMCounter_PV	ST_ElectricMeter_Output-Counter	Struktur mit allen relevanten Zählerdaten der Elektromessung für die internen PV Strings des Hybridwechselrichters.
diNrOfBatt_Out	DINT	Berechnete Nummer des Batterie Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des

		Batterie Bausteines an diese weiterzugeben.
diNrOfBattInverter_Out	DINT	Berechnete Nummer des Batterie Inverter Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Batterie Inverter Bausteines an diese weiterzugeben.
diNrOfEM_Out_Grid	DINT	Berechnete Nummer des Elektro Messungs- Bausteines für den Netzanschluss, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs-Bausteines an diese weiterzugeben.
diNrOfEM_Out_PV	DINT	Berechnete Nummer des Elektro Messungs- Bausteines für die internen PV Strings, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs-Bausteines an diese weiterzugeben.

FB_Symo_PV_Inverter_FLOAT_1_DM_1_Inverter

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem Fronius Symo 10 PV Wechselrichter aufzubauen. Somit kann z.B. die Leistung welche gerade durch die PV Anlage erzeugt wird ermittelt werden. Dieser Funktionsblock kann die Daten von einem Fronius Datenmanager und maximal einem Inverter empfangen. Als Übertragungsformat muss am Inverter selbst «FLOAT» ausgewählt werden. Wenn dieser Funktionsblock «Dissabled» wird, ist der Inverter nicht aus, sondern es werden keine Daten mehr vom Wechselrichter abgefragt. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 1.1.5-0.



Eingänge

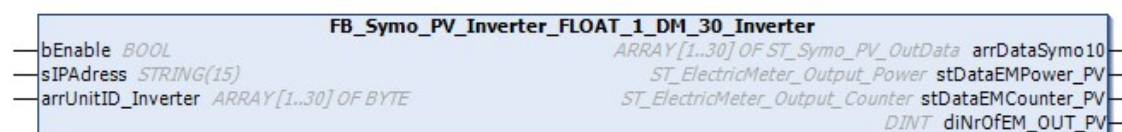
Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschaltet sondern fährt seine eigene interne Regellogik.
sIPAdress	STRING(15)	IP Adresse des PV Wechselrichters.
byUnitID_Inverter	BYTE	Identifizierungsnummer des PV Inverters im TCP Netzwerk.

Ausgänge

Name	Typ	Beschreibung
stDataSymo10	ST_Symo_PV_OutData	Struktur mit allen relevanten Ausgangssignalen des Fronius Symo 10 PV Inverters.
stDataEMPower_PV	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung.
stDataEMCounter_PV	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung.
diNrOfEM_Out_PV	DINT	Berechnete Nummer des Elektro Messungs- Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs- Bausteines an diese weiterzugeben.

FB_Symo_PV_Inverter_FLOAT_1_DM_30_Inverter

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem oder mehreren Fronius Symo 10 PV Wechselrichtern aufzubauen. Somit kann z.B. die Gesamt Leistung welche gerade durch die PV Anlagen erzeugt wird ermittelt werden. Dieser Funktionsblock kann die Daten von einem Fronius Datenmanager und maximal 30 Invertern empfangen. Die Inverter müssen nach dem Fronius Master-Slave Prinzip betrieben werden. Ein Inverter ist der Master, alle anderen sind die Slaves. Über den Master können die Daten aller anderen Wechselrichter in dem Verbund abgefragt werden. Als Übertragungsformat muss an den Invertern selbst «FLOAT» ausgewählt werden. Wenn dieser Funktionsblock «Disabled» wird, sind die Inverter nicht aus, sondern es werden lediglich keine Daten mehr von den Wechselrichtern abgefragt. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 1.1.5-0.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschalten.
sIPAdress	STRING(15)	IP Adresse des PV Wechselrichters.
arrUnitID_Inverter	ARRAY[1..30] OF BYTE	Array mit allen Identifizierungsnummern der maximal 30 PV Inverter im TCP Netzwerk.

Ausgänge

Name	Typ	Beschreibung
arrDataSymo10	ARRAY[1..30] OF ST_Symo_PV_OutData	Array mit der Struktur aller relevanten Ausgangssignalen der maximal 30 Fronius Symo 10 PV Inverter.
stDataEMPower_PV	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung.
stDataEMCounter_PV	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung.
diNrOfEM_Out_PV	DINT	Berechnete Nummer des Elektro Messungs- Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs-

		Bausteines an diese weiterzugeben.
--	--	------------------------------------

FB_Symo_PV_Inverter_INT_1_DM_1_Inverter

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem Fronius Symo 10 PV Wechselrichter aufzubauen. Somit kann z.B. die Leistung welche gerade durch die PV Anlage erzeugt wird ermittelt werden. Dieser Funktionsblock kann die Daten von einem Fronius Datenmanager und maximal einem Inverter empfangen. Als Übertragungsformat muss am Inverter selbst «INT» ausgewählt werden. Wenn dieser Funktionsblock «Dissabled» wird, ist der Inverter nicht aus, sondern es werden keine Daten mehr vom Wechselrichter abgefragt. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 1.1.5-0.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschalten.
sIPAdress	STRING(15)	IP Adresse des PV Wechselrichters.
byUnitID_Inverter	BYTE	Identifizierungsnummer des PV Inverters im TCP Netzwerk.

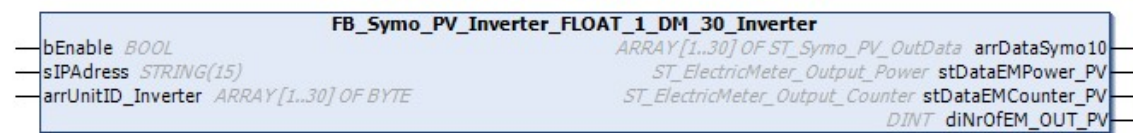
Ausgänge

Name	Typ	Beschreibung
stDataSymo10	ST_Symo_PV_OutData	Struktur mit allen relevanten Ausgangssignalen des Fronius Symo 10 PV Inverters.
stDataEMPower_PV	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung.
stDataEMCounter_PV	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung.
diNrOfEM_Out_PV	DINT	Berechnete Nummer des Elektro Messungs- Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs- Bausteines an diese weiterzugeben.

FB_Symo_PV_Inverter_INT_1_DM_30_Inverter

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem oder mehreren Fronius Symo 10 PV Wechselrichtern aufzubauen. Somit kann z.B. die Gesamt Leistung welche gerade durch die PV Anlagen erzeugt wird ermittelt werden. Dieser Funktionsblock kann die Daten von einem Fronius Datenmanager

und maximal 30 Invertern empfangen. Die Inverter müssen nach dem Fronius Master-Slave Prinzip betrieben werden. Ein Inverter ist der Master, alle anderen sind die Slaves. Über den Master können die Daten aller anderen Wechselrichter in dem Verbund abgefragt werden. Als Übertragungsformat muss an den Invertern selbst «INT» ausgewählt werden. Wenn dieser Funktionsblock «Dissabled» wird, sind die Inverter nicht aus, sondern es werden lediglich keine Daten mehr von den Wechselrichtern abgefragt. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 1.1.5-0.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschalten.
sIPAdress	STRING(15)	IP Adresse des PV Wechselrichters.
arrUnitID_Inverter	ARRAY[1..30] OF BYTE	Array mit allen Identifizierungsnummern der maximal 30 PV Inverter im TCP Netzwerk.

Ausgänge

Name	Typ	Beschreibung
arrDataSymo10	ARRAY[1..30] OF ST_Symo_PV_OutData	Array mit der Struktur aller relevanten Ausgangssignalen der maximal 30 Fronius Symo 10 PV Inverter.
stDataEMPower_PV	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung.
stDataEMCounter_PV	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung.
diNrOfEM_Out_PV	DINT	Berechnete Nummer des Elektro Messungs- Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs- Bausteines an diese weiterzugeben.

FB_HP_Soltop_M_Tec_Modular

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einer Soltop/M-Tec Wärmepumpe aufzubauen. Kommuniziert wird dafür mit dem M-Tech Modbus Server. Diese Funktion funktioniert nur mit Wärmepumpen von Soltop/M-Tech, welche sich linear in Ihrer Leistung regeln lassen. Sie müssen somit Modulierende Wärmepumpen sein. Zudem funktioniert dieser Funktionsblock nur mit den Wärmepumpen, welche über die Modbus Register in der M-Tech Dokumentation in der Abteilung «heatpump[0]» gesteuert werden können. Zusätzlich ermittelt dieser Funktionsblock auch die obere Temperatur im Brauchwasserspeicher, der in der M-Tech Dokumentation unter «hotWaterTank[0]» zu finden ist. Die Überschüssige Leistung die über diesen Funktionsblock an die Wärmepumpe übermittelt wird, teilt diese dann selbstständig dort ein wo diese gebraucht werden, z.B. bei der Warmwasseraufbereitung. Die Version mit welcher die Wärmepumpe in Lynus implementiert wurde lautet : 20210531.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
bPowerDataInvers	BOOL	Leistungsdaten vom verknüpften Energiezähler werden invertiert.
byUnitID	BYTE	Identifizierungsnummer der Wärmepumpe im TCP Netzwerk.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Wärmepumpe regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.
diNrOfEM_IN_HP	DINT	Angabe der Nummer der Elektromessung, welche die Leistung der Wärmepumpe misst. Siehe jeweiligen Ausgang der Elektromessung.
sIPAdress	STRING(15)	IP Adresse der Wärmepumpe.
stSetupHP	ST_Setup_HP	Struktur mit allen relevanten Einstellungen der Wärmepumpe.
fbDIExternalLock	FB_XL1XXX_DI	Digitaler Hardware Eingang um eine externe Sperre der Wärmepumpe zu aktivieren.

Ausgänge

Name	Typ	Beschreibung
stDataHP	ST_HP_Output	Struktur mit allen relevanten Ausgangssignalen der Wärmepumpe.
stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfHP_Out	DINT	Berechnete Nummer der Wärmepumpe, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten der Wärmepumpe an diese weiterzugeben.
diNrOfTempS_OUT_ServiceWaterT op	DINT	Berechnete Nummer des Temperaturmessungs-Bausteines welcher die obere Temperatur im Brauchwasser Speicher misst, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Temperaturmessungs-Bausteines an diese weiterzugeben.

FB_HP_Viessmann_Vitogate300_1Stage

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einer Viessmann Wärmepumpe aufzubauen. Kommuniziert wird dafür mit dem Viessmann Vitogate 300, welches an der Wärmepumpe verbaut sein muss. Diese Funktion funktioniert nur mit Wärmepumpen von Viessmann, welche 1 Stufig sind. Um den Überschuss an Leitung mit der Wärmepumpe zu verarbeiten wird eine erhöhte Warmwassertemperatur im Brauchwasserspeicher angefahren. Diese kann an einem separaten Eingang der Funktion angegeben werden. Damit die Kommunikation zum Vitogate 300 funktioniert und die relevanten Daten für diesen Funktionsblock zur Verfügung stehen, müssen folgende Register, bzw. Datenpunkte im Vitogate 300 freigegeben sein : IN-103, IN-110, IN-101, IN-99, IN- 97, IN-112, IN-113, ST-201, ST-25, HO-30. Die Version mit welcher die Wärmepumpe in Lynus implementiert wurde lautet : 2.1.3.0.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
bPowerDataInvers	BOOL	Leistungsdaten vom verknüpften Energiezähler werden invertiert.
byUnitID	BYTE	Identifizierungsnummer der Wärmepumpe im TCP Netzwerk.
byPlantNr	BYTE	Viessmann Anlagennummer zu der die Wärmepumpe im Vitogate 300 gehört.
byParticipianNr	BYTE	Viessmann Teilnehmernummer der Wärmepumpe im Vitogate 300.
siCorrectionMBReg	SINT	Korrektur der Modbus Register Adressen für die Kommunikation mit der Wärmepumpe. Im Normalfall beträgt dieser Wert Default immer 0.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Wärmepumpe regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.

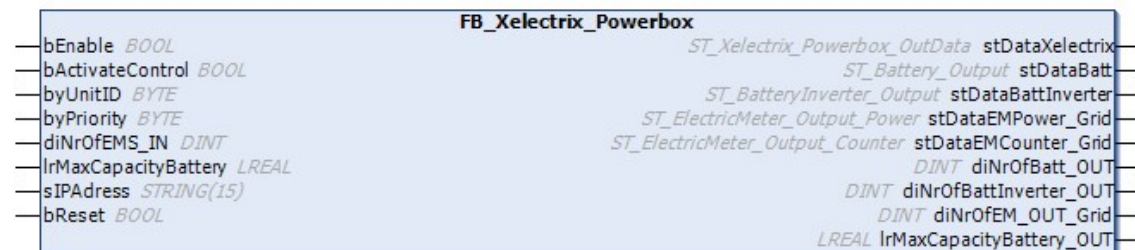
diNrOfEM_IN_HP	DINT	Angabe der Nummer der Elektromessung, welche die Leistung der Wärmepumpe misst. Siehe jeweiligen Ausgang der Elektromessung.
rSetTempBoilerServiceWaterTopMax	REAL	Maximale Warmwasser Temperatur wenn die Wärmepumpe mit PV Überschussleistung betrieben wird.
rSetTempBoilerServiceWaterTopMin	REAL	Minimale Warmwasser Temperatur wenn der Wärmepumpe keine Überschussleistung der PV zu Verfügung steht.
sIPAdress	STRING(15)	IP Adresse der Wärmepumpe.
stSetupHP	ST_Setup_HP	Struktur mit allen relevanten Einstellungen der Wärmepumpe.
fbDIExternalLock	FB_XL1XXX_DI	Digitaler Hardware Eingang um eine externe Sperre der Wärmepumpe zu aktivieren.

Ausgänge

Name	Typ	Beschreibung
stDataHP	ST_HP_Output	Struktur mit allen relevanten Ausgangssignalen der Wärmepumpe.
stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfHP_Out	DINT	Berechnete Nummer der Wärmepumpe, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten der Wärmepumpe an diese weiterzugeben.
diNrOfTempS_OUT_ServiceWaterTop	DINT	Berechnete Nummer des Temperaturmessungs-Bausteines welcher die obere Temperatur im Brauchwasser Speicher misst, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Temperaturmessungs-Bausteines an diese weiterzugeben.

FB_Xelectrix_Powerbox

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem Xelectrix Batteriespeicher System aufzubauen. Das Gerät dazu nennt sich Powerbox. Somit kann die Leistung welche in oder aus der Batterie fließt durch Lynus geregelt werden. Wenn dieser Funktionsblock «Dissabled» wird, ist der Inverter zwar nicht aus, aber es fließt dann weder Leistung in die Batterie noch aus der Batterie. Auch eine Abschaltung des Wechselrichter im Notstrombetrieb beim Unterschreiten einer bestimmten Batteriekapazität ist mit dieser Funktion nicht möglich, da dies die Kommunikationsschnittstelle zu Xelectrix nicht zulässt. Dies erledigt der Inverter eigenständig und kann somit nicht beeinflusst werden. Wenn keine Leistungsvorgabe an den Inverter geschickt wird, also weder Leistung in die Batterie noch aus der Batterie soll, dann regelt diese Funktion auch den Standby Verbrauch des Systems aus um ein unerwünschtes Entladen der Batterie zu verhindern. Die Version mit welcher der Inverter in Lynus implementiert wurde lautet : 106.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt. Achtung, der Inverter ist somit nicht ausgeschalten sondern fährt seine eigene interne Regellogik.
bActivateControl	BOOL	Wenn dieser Eingang zusammen mit bEnable gesetzt ist, dann regelt Lynus den Inverter. Wenn dieser Eingang auf False gesetzt wird, regelt der Inverter gar nicht.
byUnitID	BOOL	Identifizierungsnummer des Inverters im TCP Netzwerk.
byPriority	BYTE	Gibt die aktuelle Priorität wieder, mit welcher der Batterie Wechselrichter vom EMS Funktionsbaustein abgearbeitet wird.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Ladestation regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.

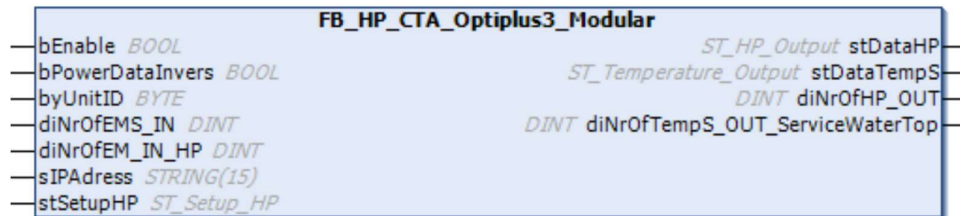
lrMaxCapacityBattery	LREAL	Angabe der Batterie für die Maximalen Batteriekapazität in kWh.
siPAdress	STRING(15)	IP Adresse des Hybridwechselrichters
bReset	BOOL	Eingang um bei einem Fehler des Inverters die internen Warnung oder Fehler Ausgänge zurückzusetzen.

Ausgänge

Name	Typ	Beschreibung
stDataGenXelectrix	ST_Xelectrix_Powerbox_Out Data	Struktur mit allen relevanten Ausgangssignalen des Xelectrix Batteriesystemes.
stDataBatt	ST_Battery_Out	Struktur mit allen relevanten Ausgangssignalen der Batterie.
stDataBattInverter	ST_BatteryInverter_Output	Struktur mit allen relevanten Ausgangssignalen des Batterie Inverters.
stDataEMPower_Grid	ST_ElectricMeter_Output_Power	Struktur mit allen relevanten Leistungsdaten der Elektromessung für den Netzanschluss.
stDataEMCounter_Grid	ST_ElectricMeter_Output-Counter	Struktur mit allen Relevanten Zählerdaten der Elektromessung für den Netzanschluss.
diNrOfBatt_Out	DINT	Berechnete Nummer des Batterie Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Batterie Bausteines an diese weiterzugeben.
diNrOfBattInverter_Out	DINT	Berechnete Nummer des Batterie Inverter Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Batterie Inverter Bausteines an diese weiterzugeben.
diNrOfEM_Out_Grid	DINT	Berechnete Nummer des Elektro Messungs- Bausteines für den Netzanschluss, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Elektro Messungs- Bausteines an diese weiterzugeben.
lrMaxCapacityBattery_Out	LREAL	Maximale Batteriekapazität in kWh die über die Schnittstelle vom Speicher abgefragt werden können.

FB_HP_CTA_Optiplus3_Modular

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einer CTA Wärmepumpe aufzubauen. Dieser Funktionsblock funktioniert mit den CTA Wärmepumpen, welche über die Optiplus 3 Schnittstelle verfügen. Dabei wird der Wärmepumpe über ihr internes PV Management eine Überschuss Leistung in Watt übermittelt, welche die Wärmepumpe dann selbstständig dort um- und einsetzt, wo es gebraucht, bzw. verbraucht werden kann. Die Temperatur im Brauchwasser Speicher wird dadurch erhöht. Die Version mit welcher die Wärmepumpe in Lynus implementiert wurde lautet : V1.10.1.



Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
bPowerDataInvers	BOOL	Leistungsdaten vom verknüpften Energiezähler werden invertiert.
byUnitID	BYTE	Identifizierungsnummer der Wärmepumpe im TCP Netzwerk.
diNrOfEMS_IN	DINT	Angabe der EMS Nummer, welche die Wärmepumpe regeln soll. Siehe jeweiligen Ausgang vom EMS Funktionsblock.
diNrOfEM_IN_HP	DINT	Angabe der Nummer der Elektromessung, welche die Leistung der Wärmepumpe misst. Siehe jeweiligen Ausgang der Elektromessung.
sIPAdress	STRING(15)	IP Adresse der Wärmepumpe.
stSetupHP	ST_Setup_HP	Struktur mit allen relevanten Einstellungen der Wärmepumpe.
fbDIExternalLock	FB_XL1XXX_DI	Digitaler Hardware Eingang um eine externe Sperre der Wärmepumpe zu aktivieren.

Ausgänge

Name	Typ	Beschreibung
stDataHP	ST_HP_Output	Struktur mit allen relevanten Ausgangssignalen der Wärmepumpe.

stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfHP_Out	DINT	Berechnete Nummer der Wärmepumpe, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten der Wärmepumpe an diese weiterzugeben.
diNrOfTempS_OUT_ServiceWaterTop	DINT	Berechnete Nummer des Temperaturmessung s- Bausteines welcher die obere Temperatur im Brauchwasser Speicher misst, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Temperaturmessung s- Bausteines an diese weiterzugeben.