

Dokumentation zur Lynus Energy Modbus RTU Bibliothek für die Beckhoff Steuerungen

Inhalt

1.	Systemvoraussetzungen	4
2.	Einführung.....	4
3.	Funktionsweise.....	4
3.1	Beispiel.....	4
3.2	Verknüpfungen im Dashboard	5
4.	Installation der Bibliothek	7
5.	Verwendete Datentypen.....	8
5.1	ST_DeltaPvInverter_Data	8
5.2	E_DeltaEvent.....	8
5.3	E_DeltaOperatingState	9
5.4	E_Schrack_ECS_Error	9
6.	Funktionsblöcke	10
6.1	FB_Delta_PV_Inverter_Sunspec	10
6.2	FB_KL6xxx_ModbusRTU_22B_Master.....	12
6.3	FB_ECS_Schrack_EM_CionHome	14

Version	Datum	Bearbeiter/Ersteller	Änderung/Ergänzung	Firma
1.0.0.0	19.01.2022	Kai Ebensperger	Dokument erstellt	Lynus AG

1. Systemvoraussetzungen

Die Lynus Bibliothek funktioniert auf allen Beckhoff Steuerungen welche TwinCat 3.1 4022.0 oder höher installiert haben. Für diese Bibliothek werden Beckhoff Funktionen verwendet welche standartmäßig bei der Installation von TwinCat 3.1 dabei sind. Zusätzlich dazu braucht es noch die TC3 TF6255 Function (RTU) und die TC3 TF6340 Function (Serial).

Diese Bibliothek funktioniert nur in Verbindung mit der Lynus Communicator Bibliothek und einer aktiven Verbindung zu einem erzeugten Projekt im Lynus Dashboard. Gibt es keine aktive Verbindung zu einem Cloud Projekt, stoppen die Funktionalitäten dieser Bibliothek automatisch nach 7 Tagen betrieb.

Diese Bibliothek verlangt die Installation der Lynus Standard Bibliothek.

2. Einführung

In der Lynus Energy Modbus RTU 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 Modbus RTU verwenden. Unter anderem findet man hier Funktionsblöcke zu Wärmepumpen, Ladestationen, Energiemessungen, Wechselrichtern usw...

3. 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.

3.1 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.

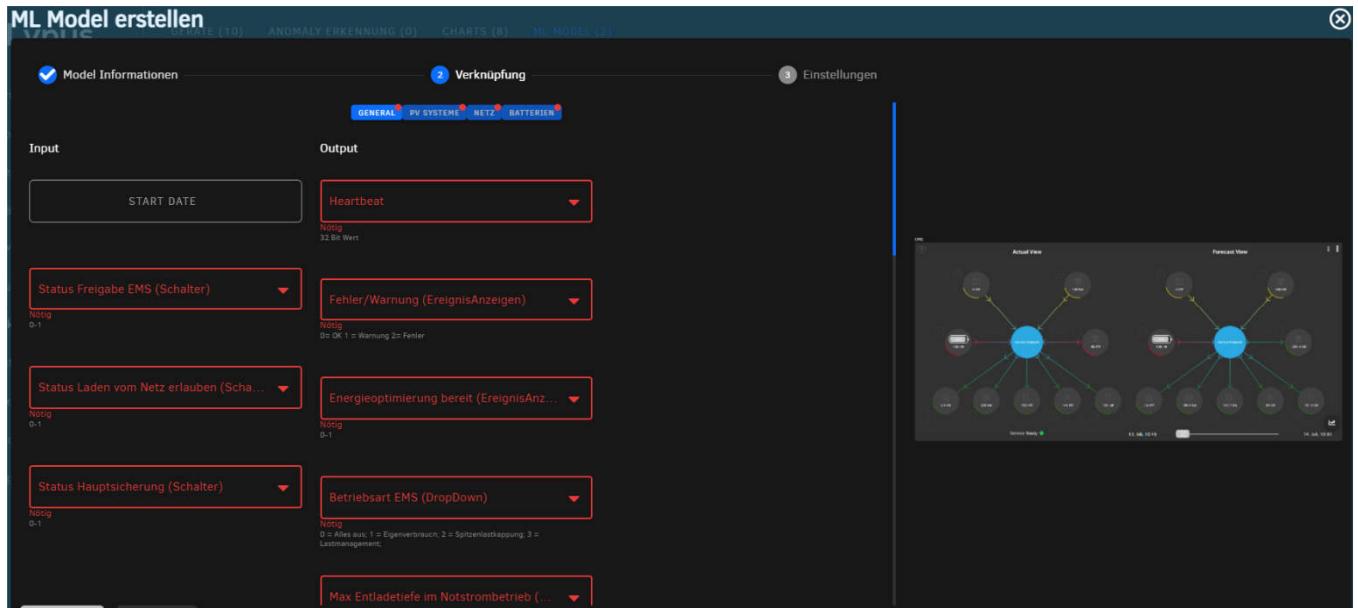


Hier sieht man recht gut, wie der M-Bus Masterbaustein, welcher die Kommunikation zur Beckhoff M-Bus Klemme übernimmt über den Ausgang **diNrOfMBusMasterLine_OUT** die Daten weitergibt an den M-Bus Zähler von Optec über den Eingang **diNrOfMBusMasterLine_IN**. Dieser Baustein gibt die Daten dann über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

3.2 Verknüpfungen im Dashboard

Bestimmte Geräte im Lynus 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 markiert im Funktionsbaustein. 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 Lynus Communicator.

Nötiges Variablenmapping im Dashboard



Variablen im Funktionsblock markiert über das Kommentar

FUNCTION_BLOCK FB_EMS

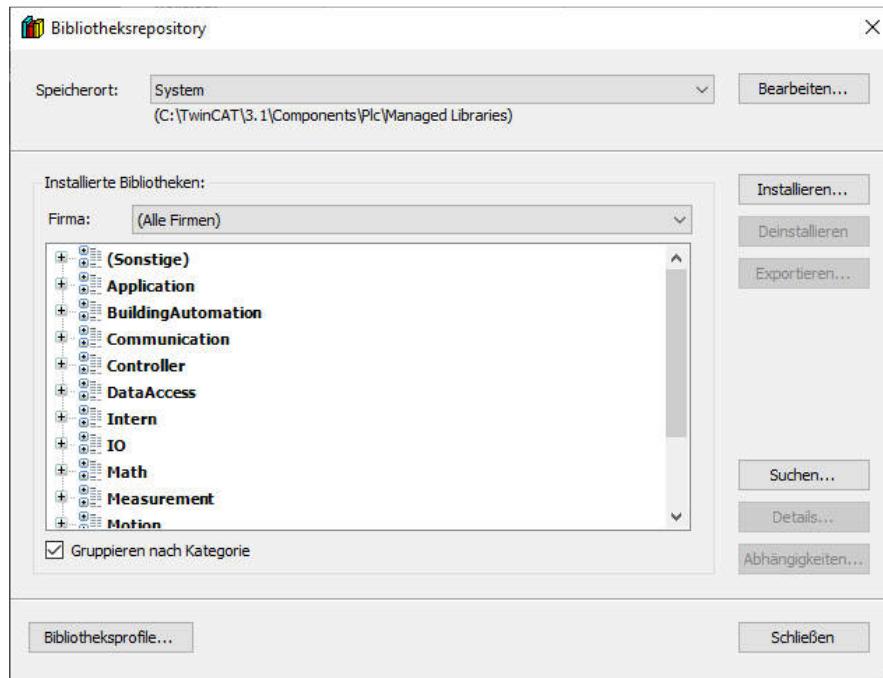
Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
bEnable	BOOL				{#lynus.ag#() //Enable and dissable this function
dINrOfEM_IN_ECS	DINT				Number of the electric meter for incoming power data. (Its for the complete consumption of all charging stations)
stSetupEMS	ST_Setup_EMS	STRUCT (uiUpdateTime := 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
bActivteUPD	BOOL				{#lynus.ag#}

4. Installation der Bibliothek

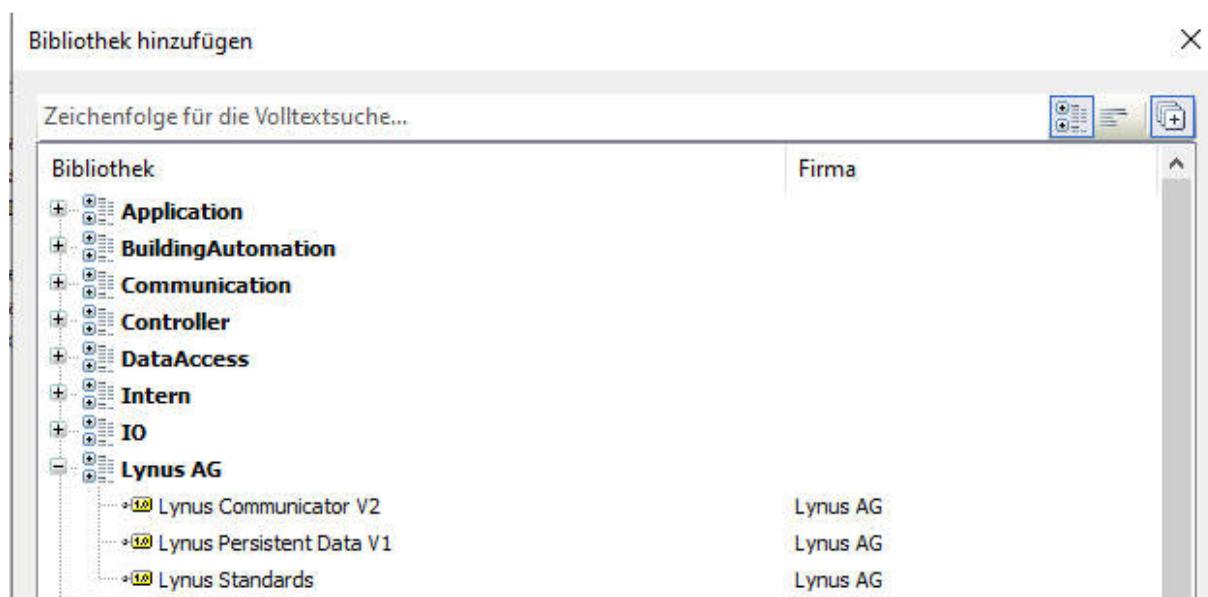
Nachdem die Bibliothek über den erstellten Account heruntergeladen wurde, im SPS Projekt rechtsklick auf References und dann klick auf Bibliotheksrepository =>



Danach auf installieren klicken =>



Die Lynus Bibliothek im abgespeicherten Pfad auswählen und dann auf Öffnen klicken. Nach erfolgreicher Installation erscheint die Bibliothek im Ordner „Lynus AG“ =>



5. 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.

5.1 ST_DeltaPvInverter_Data	
Diese Struktur enthält gängige Ausgabewerte eines Delta PV Wechselrichters.	
Name	Beschreibung
eOperatingState	Gibt den aktuellen Status des Wechselrichters aus.
eEvent1	Zeigt das aktuelle Event an welches Gerade am Wechselrichter ansteht.
rTotalMPPTPower	Zeigt die totale MPPT Leistung des Inverters in kW an.
rCabinetTemp	Gibt die Temperatur im Gehäuse in °C aus.
rHeatSinkTemp	Gibt die Kühlkörpertemperatur in °C aus.
arrCurrentFault[1..8]	Array mit den unterschiedlichen Fehlerzuständen. Siehe Bitmap in der Dokumentation zu den Delta Invertern.
arrCurrentError[1..5]	Array mit den unterschiedlichen Störungszuständen. Siehe Bitmap in der Dokumentation zu den Delta Invertern.
arrCurrentWarning[1..3]	Array mit den unterschiedlichen Warnungszuständen. Siehe Bitmap in der Dokumentation zu den Delta Invertern.
arrMPPTPower [1..6]	Maximale MPPT Leistung in kW. Maximal 6 MPPT pro Inverter sind möglich.

5.2 E_DeltaEvent	
Enum welche die Events eines Delta Wechselrichters enthält.	
Name	Beschreibung
eNolInfo	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eGroundFault	Siehe Delta Dokumentation.
eDCOverVolt	. Siehe Delta Dokumentation.
eACDisconnect	Siehe Delta Dokumentation.
eDCDisconnect	Siehe Delta Dokumentation.
eGridDisconnect	Siehe Delta Dokumentation.
eCabinetOpen	Siehe Delta Dokumentation.
eManualShutdown	Siehe Delta Dokumentation.

eOverTemp	Siehe Delta Dokumentation.
eOverFrequency	Siehe Delta Dokumentation.
eUnderFrequency	Siehe Delta Dokumentation.
eACOverVolt	Siehe Delta Dokumentation.
eACUnderVolt	Siehe Delta Dokumentation.
eBlownStringFuse	Siehe Delta Dokumentation.
eUnderTemp	Siehe Delta Dokumentation.
eMemoryLoss	Siehe Delta Dokumentation.
eHWTestFailure	Siehe Delta Dokumentation.
eOtherAlarm	Siehe Delta Dokumentation.
eOtherWarning	Siehe Delta Dokumentation.

5.3 E_DeltaOperatingState	
Enum welches den aktuellen Zustand eines Delta Wechselrichters enthält.	
Name	Beschreibung
eNolInfo	Momentan liegen keine Informationen vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eOff	Siehe Delta Dokumentation.
eSleeping	Siehe Delta Dokumentation.
eStarting	Siehe Delta Dokumentation.
eMppt	Siehe Delta Dokumentation.
eThrottled	Siehe Delta Dokumentation.
eShuttingDown	Siehe Delta Dokumentation.
eFault	Siehe Delta Dokumentation.
eStandby	Siehe Delta Dokumentation.

5.4 E_Schrack_ECS_Error	
Enum welches die Fehlerzustände einer Schrack Ladestation enthält.	
Name	Beschreibung
eNoError	Momentan liegen kein Fehler vor. Dies kann auch angezeigt werden wenn der Baustein gerade neu gestartet wurde.
eGroupError	Siehe Schrack Dokumentation.
eFILSError	Siehe Schrack Dokumentation.
eChargingCableError	Siehe Schrack Dokumentation.
eChargingCableRejectedError	Siehe Schrack Dokumentation.
eLoadContactError	Siehe Schrack Dokumentation.
eCommunicationErrorToTheCar	Siehe Schrack Dokumentation.
eConfigurationErrorOnLP	Siehe Schrack Dokumentation.
eRCMUError	Siehe Schrack Dokumentation.
eRCMUErrorCurrentDetected	Siehe Schrack Dokumentation.

eLockingError	Siehe Schrack Dokumentation.
eUndervoltageError	Siehe Schrack Dokumentation.
eOvervoltageError	Siehe Schrack Dokumentation.
eControllerError	Siehe Schrack Dokumentation.

6. Funktionsblöcke

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

6.1 FB_Delta_PV_Inverter_Sunspec

Mit diesem Funktionsblock ist es möglich eine Kommunikation mit einem PV Wechselrichter der Firma Delta aufzubauen. Als Protokoll muss der Wechselrichter das Sunspec Protokoll unterstützen. Die Version mit welcher das Protokoll von Delta implementiert wurde lautet : 1.16.



Eingänge

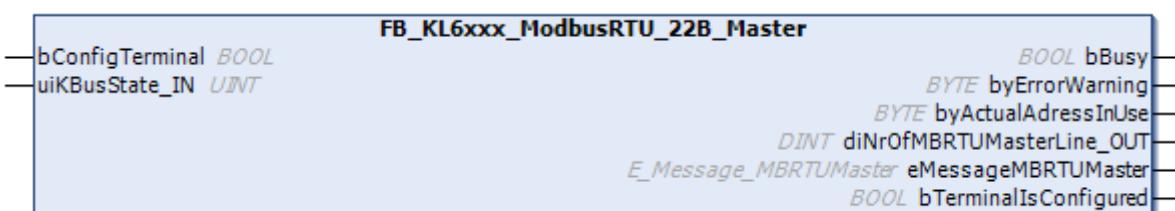
Name	Typ	Beschreibung
bEnable	BOOL	Mit diesem Eingang kann der Funktionsblock aktiviert oder deaktiviert werden. Wenn die Funktion deaktiviert ist, werden alle Ausgänge zurückgesetzt.
byMBAdress	BYTE	Modbus Stationsadresse. Gültig zwischen 1 und 247.
diNrOfMBRTUMasterLine_IN	DINT	Angabe der Modbus RTU Master Nummer, welche Daten von dieser Funktion erhalten oder senden soll. Siehe jeweiligen Ausgang vom Modbus RTU Master Funktionsblock.

Ausgänge

Name	Typ	Beschreibung
stDeltaInverterData	ST_DeltaPvInverter_Data	Struktur mit allen relevanten Daten des Delta PV Wechselrichters.
stDataEMPower_PV	ST_ElectricMeter_OutputPower	Struktur mit allen relevanten Leistungsdaten der Elektromessung.
stDataEMCounter_PV	ST_ElectricMeter_OutputCounter	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.

6.2 FB_KL6xxx_ModbusRTU_22B_Master

Mit diesem Funktionsbaustein kann ein Modbus RTU Master anhand einer Seriellen 22Byte Beckhoff Klemme realisiert werden. Dieser Baustein empfängt anfragen von anderen Funktionen im Code, führt diese aus und leitet Sie an die Serielle Klemme weiter. Gleichzeitig übernimmt dieser Baustein die Konfiguration der Klemme passend zu den Geräten welche im Feld abgefragt werden sollen. Es dürfen immer nur Geräte/Funktionen vom selben Typ an diesem Baustein betrieben werden. Unterschiedliche Geräte/Funktionen hätten ein ständiges umkonfigurieren der Klemme zur Folge und ist daher nicht ratsam. Dieser Baustein sollte in einer Task mit einer Zykluszeit von < = 5MS betrieben werden. Wenn mehrere Funktionen mit unterschiedlichen Modbus Adressen über diesen Master kommunizieren, so werden diese nacheinander abgearbeitet.



Eingänge

Name	Typ	Beschreibung
bConfigTerminal	BOOL	Mit diesem Eingang kann die Klemme manuell Konfiguriert werden, sollte der Ausgang bTerminalIsConfigured nicht automatisch nach dem Start des TwinCat Systems gesetzt worden sein.
uiKBusState_IN	UINT	K-Bus State welcher hier von der Funktion FB_K_Bus_State übergeben werden kann.
fbmodbusmaster	ModbusRtuMaster_KL6x22B	Beckhoff RTU Masterbaustein. Ein und Ausgangsdaten müssen mit der Seriellen Beckhoff Klemme verknüpft werden. Link

Ausgänge

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein wurde aktiviert.
byErrorWarning	BYTE	Dieser Ausgang zeigt an ob die Funktionen eine Störung oder eine Warnung hat. OK = Wert 0; Warnung = Wert 1; Fehler = Wert 2.
byActualAdressInUse	BYTE	Zeigt an welche Modbus RTU Stationsadresse der externen Funktionen sich gerade in Abarbeitung befindet.
diNrOfMBRTUMasterLine_Out	DINT	Berechnete Nummer des Modbus RTU Master Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Masterbausteines an diese weiterzugeben.
eMessageMBRTUMaster	E_Message_MBRTUMaster	Gibt die aktuelle Nachricht des Modbus RTU Masters aus.
bTerminalIsConfigured	BOOL	Zeigt an dass die serielle erfolgreich Konfiguriert worden ist und bereit für den Betrieb ist.

6.3 FB_ECS_Schrack_EM_CionHome

Dieser Funktionsblock kann dazu genutzt werden, um eine E-Ladestation der Firma Schrack über Modbus RTU in Ihrer Leistung zu regeln. Unterstützt werden folgende Typen von Schrack : EMCION11C; EMCION22C; EMCION22P; EMCION11CR; EMCION22CR; EMCION22PR. Die Versionsnummer mit welcher die Spezifikation von Schrack implementiert wurde lautet : Datum 13.04.2021.



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.
byMBAdress	BYTE	Modbus Stationsadresse. Gültig zwischen 1 und 247.
diNrOfMBRTUMasterLine_IN	DINT	Angabe der Modbus RTU Master Nummer, welche Daten von dieser Funktion erhalten oder senden soll. Siehe jeweiligen Ausgang

		vom Modbus RTU Master Funktionsblock.
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.
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.
eSchrackErrors	E_Schrack_ECS_Error	Errorliste der Firma Schrack.