Dokumentation zur Lynus Sensor Basic Bibliothek für die Beckhoff Steuerungen

Inhalt

Version	Datum	Bearbeiter/Ersteller	Äderung/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 nur Beckhoff Funktionen verwendet welche standartmäßig bei der Installation von TwinCat 3.1 dabei sind. Es sind keine sonstigen kostenpflichtigen Functions von Beckhoff nötig.

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 Sensor Basic Bibliothek befinden sich Funktionsblöcke und Datentypen welche vor allem mit dem Thema Sensorik und Meldungen zu tun haben. Man findet hier vor allem Funktionsblöcke die spezifisch zu einem Gerätetyp passen, oder generell ein bestimmtes Ausgangs und Eingangssignal haben und somit zu einer bestimmen Beckhoff Klemme gehören. Die Schnittstellen reichen von klassischen Digitalen Ein und Ausgangssignalen, bis hin zu Analogen Signalen oder spezielle Bussysteme wie KNX. Unter anderem findet man hier Funktionsblöcke zu Temperaturmessungen oder Statusmeldungen von verschiedenen Bussystemen.

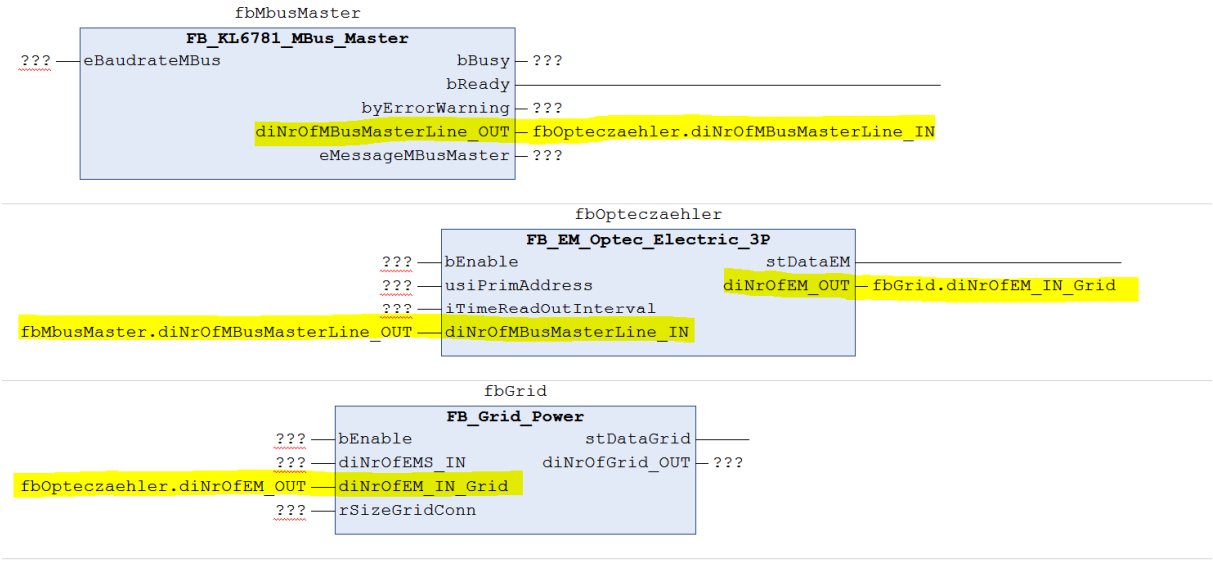
3. Funktionsweise

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 unterschieden 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.

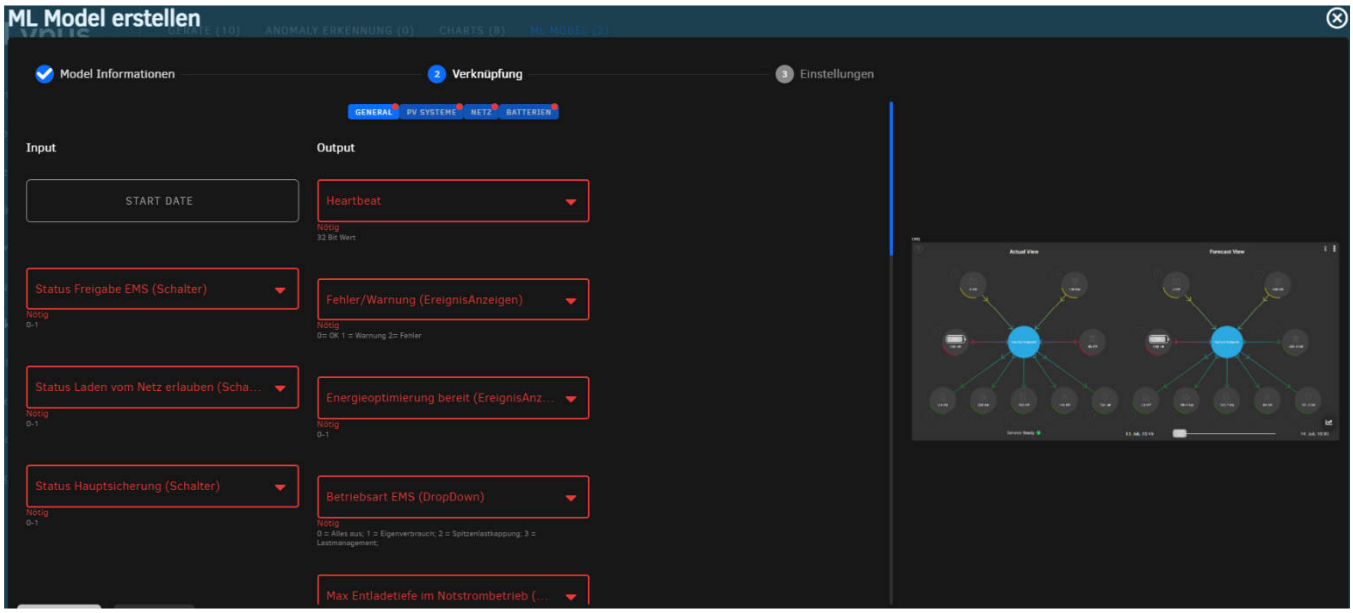


Hier sieht man recht gut, wie der M-Bus Masterbaustein, welcher die Kommunikation zur Beckhoff M-Bus Klemme übernimmt über den Ausgang diNrOfMBusMasterLine_OUT die Daten weitergibt an den M-Bus Zähler von Optec über den Eingang diNrOfMBusMasterLine_IN. Dieser Baustein gibt die Daten dann über den Ausgang 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



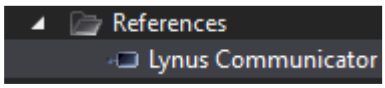
Variablen im Funktionsblock markiert über das Kommentar

FUNCTION_BLOCK **FB_EMS**

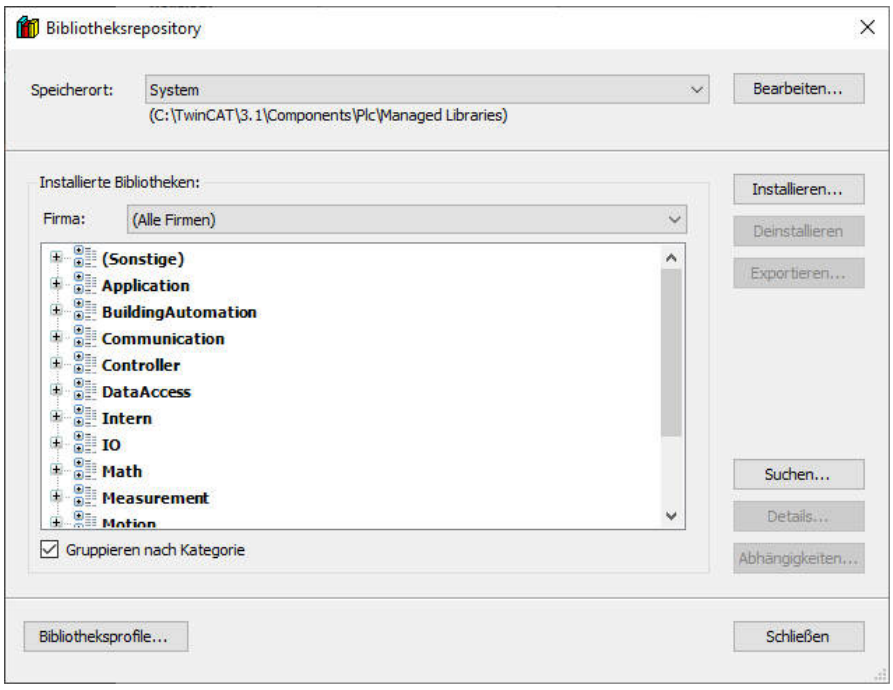
Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
bEnable	BOOL				{#lynus.ag#()} //Enable and dissable this function
diNrOfEM_IN_ECS	DINT				Number of the electric meter for incoming power data. (Its for the complete consumption of all charging stations)
stSetupEMS	ST_Setup_EMS			STRUCT (uiUpdateTime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

6. Funktionsblöcke

Anbei findet man eine Beschreibung zu allen Funktionsblöcken welche sich in der Lynus Sensor Basic Bibliothek befinden.

6.1 FB_Read_1BitValue_KNX

Mit diesem Funktionsblock lässt sich ein 1Bit Telegram das über das KNX System an eine Gruppenadresse verschickt wird empfangen und auswerten. Dieser Baustein wir in der Logik als eine «Art» Sensor angesehen. Dieser Baustein benötigt dazu zusätzlich den KNX Master Baustein damit er funktioniert.

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.
diNrOfKNXMasterLine_IN	DINT	Angabe der KNX Master Linien Nummer, welche die Daten diese Bausteines empfangen und senden soll. Siehe jeweiligen Ausgang vom KNX Master Funktionsblock.
stGroupAddr	EIB_GROUP_ADDR	Gruppenadresse an welches das KNX Telegramm gesendet werden soll. Link

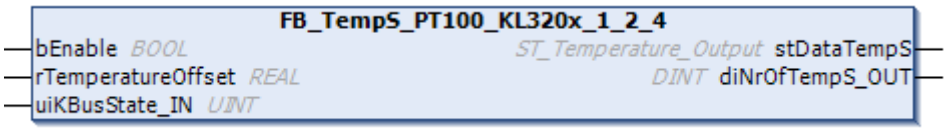
Ausgänge

Name	Typ	Beschreibung
bData	BOOL	Zeigt den Wert des erhaltenen 1Bit KNX Telegrames an.
byErrorWarning	BYTE	Dieser Ausgang zeigt an ob die Funktionen eine

		Störung oder eine Warnung hat. OK = Wert 0; Warnung = Wert 1; Fehler = Wert 2.
diNrOfKNX_S_OUT	DINT	Berechnete Nummer des KNX Sensor Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des KNX Sensors an diese weiterzugeben.

6.2 FB_TempS_PT100_KL320x_1_2_4

Mit diesem Funktionsblock lässt sich eine Temperaturmessung mit einem PT100 Fühler an einer Beckhoff Klemme realisieren. Unterstütze Klemmen sind KL3201, KL3202 und KL3204. Die KL3208 wird nicht unterstützt. Die Konfiguration der Klemme, bzw. dessen Kanal für den PT 100 findet automatisch mit diesem Funktionsblock statt.



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.
rTemperatureOffset	REAL	An diesem Eingang kann eine Temperaturoffset in °C für die reale gemessene Temperatur vorgenommen werden.
uiKBusState_IN	UINT	K-Bus State welcher hier von der Funktion FB_K_Bus_State übergeben werden kann.

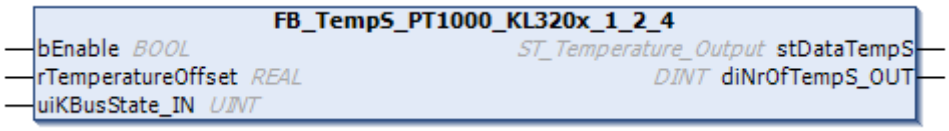
stDataIn	ST_InputData_KL320x	Eingangsstruktur welche mit der Beckhoff Hardware verknüpft werden muss.

Ausgänge

Name	Typ	Beschreibung
stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfTempS_OUT	DINT	Berechnete Nummer des Temperaturmessungs-Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Temperaturmessungs-Bausteines an diese weiterzugeben.
stDataOut	ST_OutputData_KL320x	Ausgangsstruktur welche mit der Beckhoff Hardware verknüpft werden muss.

6.3 FB_TempS_PT1000_KL320x_1_2_4

Mit diesem Funktionsblock lässt sich eine Temperaturmessung mit einem PT1000 Fühler an einer Beckhoff Klemme realisieren. Unterstütze Klemmen sind KL3201, KL3202 und KL3204. Die KL3208 wird nicht unterstützt. Die Konfiguration der Klemme, bzw. dessen Kanal für den PT 1000 findet automatisch mit diesem Funktionsblock statt.



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.
rTemperatureOffset	REAL	An diesem Eingang kann eine Temperaturoffset in °C für die reale gemessene Temperatur vorgenommen werden.
uiKBusState_IN	UINT	K-Bus State welcher hier von der Funktion FB_K_Bus_State übergeben werden kann.
stDataIn	ST_InputData_KL320x	Eingangsstruktur welche mit der Beckhoff Hardware verknüpft werden muss.

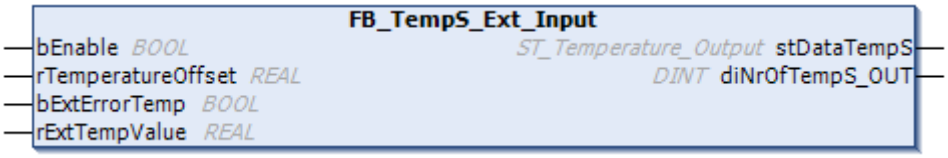
Ausgänge

Name	Typ	Beschreibung
stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfTempS_OUT	DINT	Berechnete Nummer des Temperaturmessungs-Bausteines, welche dann an andere Funktionsblöcke

		übergeben werden kann um die Daten des Temperaturmessungs-Bausteines an diese weiterzugeben.
stDataOut	ST_OutputData_KL320x	Ausgangsstruktur welche mit der Beckhoff Hardware verknüpft werden muss.

6.4 FB_TempS_Ext_Input

Dieser Funktionsbaustein dient dazu, einen Temperaturwert einer externen Temperaturmessung einzulesen und zu verarbeiten. Somit ist dieser FB ideal als «Dummy» zu gebrauchen, um z.B. von einem noch nicht integriertem Temperaturfühler im Lynus System die Daten trotzdem verwenden zu können und an diesen FB weiterzugeben. Die Temperatur Funktion kann ihre Daten dann an weitere Funktionen übergeben.



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.
rTemperatureOffset	REAL	An diesem Eingang kann eine Temperaturoffset in °C für die reale gemessene Temperatur vorgenommen werden.
bExtErrorTemp	BOOL	Störmeldung der Externen Temperaturmessung die an diesen Eingang weitergegeben werden kann.
rExtTempValue	REAL	Temperaturwert in °C der Externen

		Temperaturmessung die an diesen Eingang weitergegeben werden kann.

Ausgänge

Name	Typ	Beschreibung
stDataTempS	ST_Temperature_Output	Struktur mit allen relevanten Daten der Temperaturmessung.
diNrOfTempS_OUT	DINT	Berechnete Nummer des Temperaturmessungs-Bausteines, welche dann an andere Funktionsblöcke übergeben werden kann um die Daten des Temperaturmessungs-Bausteines an diese weiterzugeben.