

# Dokumentation zur Lynus Persistent Data Bibliothek für die Beckhoff Steuerungen

## Inhalt

1. Systemvoraussetzungen .....	4
2. Einführung.....	4
3. Installation der Bibliothek.....	5
4. Funktionsblöcke.....	6
4.1 FB_WritePersistentData_CX51x0.....	6
4.2 FB_WritePersistentData_CX81xx.....	7
4.3 FB_WritePersistentData_CX9020_U900.....	9
4.4 FB_WritePersistentData_General.....	11
5 Verwendete Datentypen .....	13
5.1 eUSVSaveMode.....	13
6 Beispiele .....	13
6.1 Abspeichern der Daten im Mode Eventbasiert.....	13
6.2 Abspeichern der Daten im Mode Spannungsausfall .....	14
6.3 Abspeichern der Daten wenn beide Modi aktiv sind .....	14

Version	Datum	Bearbeiter/Ersteller	Äderung/Ergänzung	Firma
1.0.0.0	16.08.2021	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 standardmäßig 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.

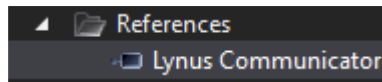
**Diese Bibliothek verlangt die Installation der Lynus Standard Bibliothek.**

## 2. Einführung

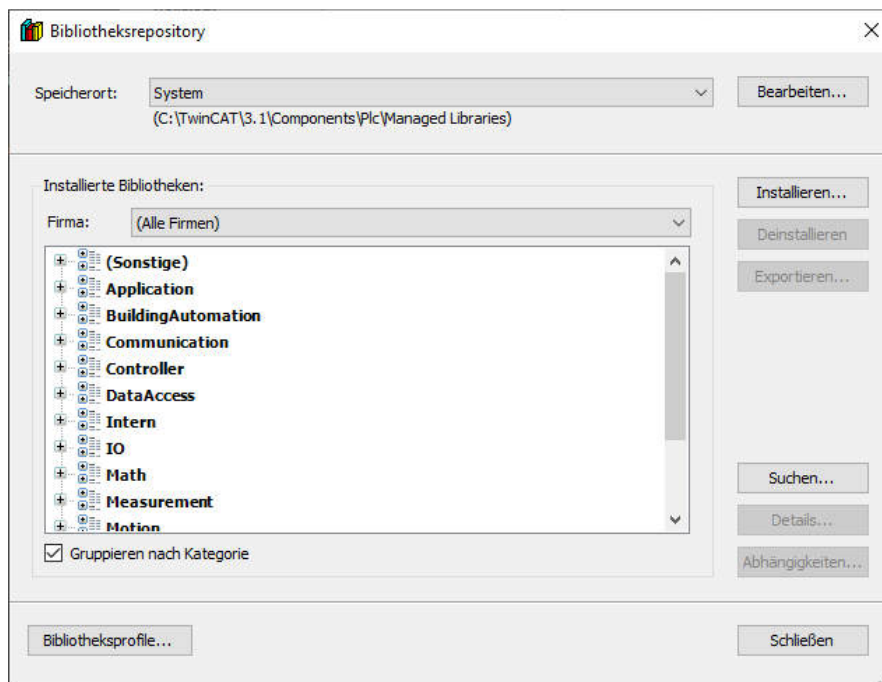
Die Lynus Persistent Data Bibliothek wurde entwickelt um eine einfache Möglichkeit auf der Beckhoff SPS zu bieten um Daten persistent abzuspeichern. Somit haben die Variablen nach dem Ausfall der Spannungsversorgung und dem darauf folgendem Neustart wieder den selber Wert wie davor. Bei den Funktionsbausteinen kann ausgewählt werden um die Daten bei Spannungsausfall (Nur bei PLC's mit 1 Sekunden USV möglich), Eventbasiert oder bei beidem abgespeichert werden. Diese Library funktioniert nur in Verbindung mit einer weiteren Lynus Bibliothek und kann nicht alleine genutzt werden. Die Funktionsblöcke dieser Bibliothek können dann in jedem Programmaufruf des SPS Programmes instanziiert und aufgerufen werden. **WICHTIG** : Ohne den Aufruf eines solchen Funktionsblocks werden **keine** Daten aller Lynus Bibliotheken persistent abgespeichert.

### 3. 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“ =>



## 4. Funktionsblöcke

In dieser Bibliothek findet man verschieden Funktionsblöcke um auf unterschiedlichen Beckhoff Steuerungen die Variablen persistent abspeichern zu können. Dies kann den Grund haben, da manche Steuerungen von Beckhoff eine sogenannte «1 Sekunden» USV besitzen und manchen nicht. Die Funktionsblöcke werden von anderen Funktionsblöcken intern angesteuert bei Spannungsausfall oder einem Event und lösen dann das Speichern aus.

### 4.1 FB\_WritePersistentData\_CX51x0

Dieser FB kann auf allen Beckhoff CX51x0 Systemen mit einer 1 Sekunden USV verwendet werden. Der Funktionsblock darf auf keinem anderen Beckhoff System verwendet werden da dies zu einem Absturz des Systems führen kann. Empfohlene Zykluszeit < = 10MS.



#### Eingänge

Name	Typ	Beschreibung
eSaveModeData	eUSVSaveMode	An diesem Eingang kann über ein ENUM angegeben werden in welchem Modus die Daten abgespeichert werden sollen. Zur Auswahl stehen 3 Modi. (Bei Spannungsausfall; bei einem Event; bei beidem)

#### Ausgänge

Name	Typ	Beschreibung
bLoadPersistentDataOK	BOOL	Dieser Ausgang gibt an ob beim Neustart der Steuerung die Persistenten Daten erfolgreich geladen werden konnten.
bPowerFailDetect	BOOL	Wird True wenn eine Spannungsausfall auf der Steuerung detektiert wird.

bError	BOOL	Dieser Ausgang wird True wenn es beim Laden oder speichern der Daten zu Problemen kommt oder kommen kann.
bySaveProgress	BYTE	Gibt beim Eventbasierten speichern den Fortschritt an, bis die Daten erfolgreich gespeichert wurden.
dwStorageLast24h	DWORD	Dieser Zählerstand gibt an wie oft in den letzten 24 Stunden Daten Eventbasiert persistent abgespeichert wurden. Der Zähler wird alle 24 Stunden resettet.
udiErrorCode	UDINT	Dieser Ausgang gibt bei einem internen Beckhoff Fehler den Fehlercode an.

#### 4.2 FB\_WritePersistentData\_CX81xx

Dieser FB kann auf allen Beckhoff CX81xx Systemen mit einer 1 Sekunden USV verwendet werden. Der Funktionsblock darf auf keinem anderen Beckhoff System verwendet werden da dies zu einem Absturz des Systems führen kann. Empfohlene Zykluszeit < = 10MS.



#### Eingänge

Name	Typ	Beschreibung
eSaveModeData	eUSVSaveMode	An diesem Eingang kann über ein ENUM angegeben werden in welchem Modus die Daten abgespeichert werden sollen. Zur Auswahl stehen 3 Modi. (Bei Spannungsausfall; bei einem Event; bei beidem)

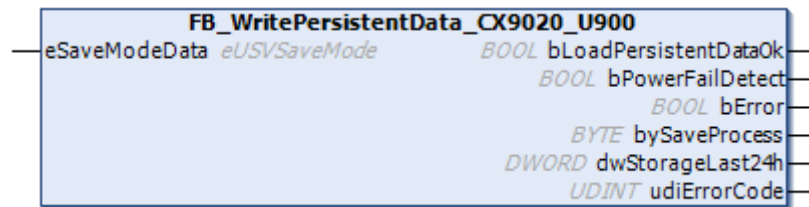
## Ausgänge

Name	Typ	Beschreibung
bLoadPersistentDataOK	BOOL	Dieser Ausgang gibt an ob beim Neustart der Steuerung die Persistenten Daten erfolgreich geladen werden konnten.
bPowerFailDetect	BOOL	Wird True wenn eine Spannungsausfall auf der Steuerung detektiert wird.
bError	BOOL	Dieser Ausgang wird True wenn es beim Laden oder speichern der Daten zu Problemen kommt oder kommen kann.
bySaveProgress	BYTE	Gibt beim Eventbasierten speichern den Fortschritt an, bis die Daten erfolgreich gespeichert wurden.
dwStorageLast24h	DWORD	Dieser Zählerstand gibt an wie oft in den letzten 24 Stunden Daten Eventbasiert persistent abgespeichert wurden. Der Zähler wird alle 24 Stunden resettet.
udiErrorCode	UDINT	Dieser Ausgang gibt bei einem internen Beckhoff Fehler den Fehlercode an.



### 4.3 FB\_WritePersistentData\_CX9020\_U900

Dieser FB kann auf allen Beckhoff CX9020\_U900 Systemen mit einer 1 Sekunden USV verwendet werden. Der Funktionsblock darf auf keinem anderen Beckhoff System verwendet werden da dies zu einem Absturz des Systems führen kann. Empfohlene Zykluszeit < = 10MS.



#### Eingänge

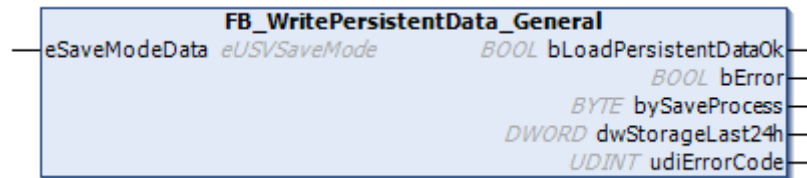
Name	Typ	Beschreibung
eSaveModeData	eUSVSaveMode	An diesem Eingang kann über ein ENUM angegeben werden in welchem Modus die Daten abgespeichert werden sollen. Zur Auswahl stehen 3 Modi. (Bei Spannungsausfall; bei einem Event; bei beidem)

**Ausgänge**

<b>Name</b>	<b>Typ</b>	<b>Beschreibung</b>
bLoadPersistentDataOK	BOOL	Dieser Ausgang gibt an ob beim Neustart der Steuerung die Persistenten Daten erfolgreich geladen werden konnten.
bPowerFailDetect	BOOL	Wird True wenn eine Spannungsausfall auf der Steuerung detektiert wird.
bError	BOOL	Dieser Ausgang wird True wenn es beim Laden oder speichern der Daten zu Problemen kommt oder kommen kann.
bySaveProgress	BYTE	Gibt beim Eventbasierten speichern den Fortschritt an, bis die Daten erfolgreich gespeichert wurden.
dwStorageLast24h	DWORD	Dieser Zählerstand gibt an wie oft in den letzten 24 Stunden Daten Eventbasiert persistent abgespeichert wurden. Der Zähler wird alle 24 Stunden resettet.
udiErrorCode	UDINT	Dieser Ausgang gibt bei einem internen Beckhoff Fehler den Fehlercode an.

#### 4.4 FB\_WritePersistentData\_General

Dieser FB kann auf allen Beckhoff Systemen ohne 1 Sekunden USV verwendet werden. Somit kann dieser FB auf so gut wie allen Beckhoff PC's verwendet werden. Bei diesem Funktionsbaustein kann das Abspeichern der Daten bei Spannungsausfall nicht verwendet werden. Empfohlene Zykluszeit < = 10MS.



##### Eingänge

Name	Typ	Beschreibung
eSaveModeData	eUSVSaveMode	An diesem Eingang kann über ein ENUM angegeben werden in welchem Modus die Daten abgespeichert werden sollen. Zur Auswahl stehen 3 Modi. (Bei Spannungsausfall; bei einem Event; bei beidem)

**Ausgänge**

<b>Name</b>	<b>Typ</b>	<b>Beschreibung</b>
bLoadPersistentDataOK	BOOL	Dieser Ausgang gibt an ob beim Neustart der Steuerung die Persistenten Daten erfolgreich geladen werden konnten.
bError	BOOL	Dieser Ausgang wird True wenn es beim Laden oder speichern der Daten zu Problemen kommt oder kommen kann.
bySaveProgress	BYTE	Gibt beim Eventbasierten speichern den Fortschritt an, bis die Daten erfolgreich gespeichert wurden.
dwStorageLast24h	DWORD	Dieser Zählerstand gibt an wie oft in den letzten 24 Stunden Daten Eventbasiert persistent abgespeichert wurden. Der Zähler wird alle 24 Stunden resettet.
udiErrorCode	UDINT	Dieser Ausgang gibt bei einem internen Beckhoff Fehler den Fehlercode an.

## 5 Verwendete Datentypen

Anbei findet man die Beschreibung der verwendeten Datentypen der Bibliothek.

5.1 eUSVSaveMode	
Name	Beschreibung
eSaveWhenPowerfail	Die Persistenten Daten werden bei einem Spannungsausfall über die 1 Sekunden USV abgespeichert.
eSaveEventBased	Die Persistenten Daten werden bei einem erfolgtem Event abgespeichert. Dies kann z.B. eine Wertänderung der Variable sein.
eSaveOnBoth	Die Persistenten Daten werden bei Spannungsausfall wie auch Eventbasiert abgespeichert.

## 6 Beispiele

Folgend finden Sie einige Beispiele wie Sie die Bibliothek in einem SPS Programm anwenden können.

### 6.1 Abspeichern der Daten im Mode Eventbasiert

```

PROGRAM MAIN
VAR
    fbPD          : FB_WritePersistentData_CX51x0;
END_VAR

```

```

fbPD(
    eSaveModeData:= eUSVSaveMode.eSaveEventBased,
    bLoadPersistentDataOk=> ,
    bPowerFailDetect=> ,
    bError=> ,
    bySaveProcess=> ,
    dwStorageLast24h=> ,
    udiErrorCode=> );

```

## 6.2 Abspeichern der Daten im Mode Spannungsausfall

```
PROGRAM MAIN
VAR
    fbPD          : FB_WritePersistentData_CX51x0;
END_VAR
```

---

```
fbPD(
    eSaveModeData:= eUSVSaveMode.eSaveWhenPowerfail,
    bLoadPersistentDataOk=> ,
    bPowerFailDetect=> ,
    bError=> ,
    bySaveProcess=> ,
    dwStorageLast24h=> ,
    udiErrorCode=> );
```

## 6.3 Abspeichern der Daten wenn beide Modi aktiv sind

---

```
PROGRAM MAIN
VAR
    fbPD          : FB_WritePersistentData_CX51x0;
END_VAR
```

---

```
fbPD(
    eSaveModeData:= eUSVSaveMode.eSaveOnBoth,
    bLoadPersistentDataOk=> ,
    bPowerFailDetect=> ,
    bError=> ,
    bySaveProcess=> ,
    dwStorageLast24h=> ,
    udiErrorCode=> );
```