

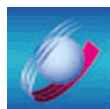
Lenze

Handbuch

IEC 61131-3

inside

***Global Drive
PLC Developer Studio***



Global Drive

Funktionsbibliothek

LenzeIOSystem.lib

Die Funktionsbibliotheken **LenzeIOSystem.lib** und **LenzeIOSystemVxxxx.lib** können für folgende Lenze PLC-Geräte verwendet werden:

	Typenbezeichnung	ab Hardwarestand	ab Softwarestand
Drive PLC	EPL10200_xl	VC	2.x
Drive PLC	EPL10200_xT	VC	2.x
9300 Servo PLC	EVS93xx_xl	2K	2.x
9300 Servo PLC	EVS93xx_xT	2K	2.x
ECSxA	ECSxAxxx	1C	7.0

Wichtiger Hinweis:

Die Software wird dem Benutzer in der vorliegenden Form zur Verfügung gestellt. Alle Risiken hinsichtlich der Qualität und der durch ihren Einsatz ermittelten Ergebnisse verbleiben beim Benutzer. Entsprechende Sicherheitsvorkehrungen gegen eventuelle Fehlbedienungen sind vom Benutzer vorzusehen.

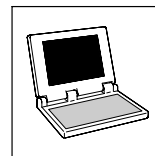
Wir übernehmen keine Verantwortung für direkt oder indirekt entstandene Schäden, z. B. Gewinnverluste, Auftragsverluste oder geschäftliche Beeinträchtigungen jeglicher Art.

© 2006 Lenze Drive Systems GmbH

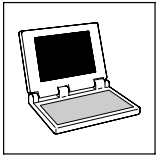
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.

Wir haben alle Angaben in dieser Dokumentation mit größter Sorgfalt zusammengestellt und auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Trotzdem können wir Abweichungen nicht ganz ausschließen. Wir übernehmen keine juristische Verantwortung oder Haftung für Schäden, die dadurch eventuell entstehen. Notwendige Korrekturen werden wir in die nachfolgenden Auflagen einarbeiten.

Alle in dieser Dokumentation aufgeführten Markennamen sind Warenzeichen ihrer jeweiligen Besitzer.



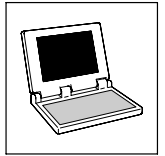
1 Vorwort und Allgemeines	1-1
1.1 Über dieses Handbuch	1-1
1.1.1 Konventionen in diesem Handbuch	1-1
1.1.2 Aufbau der Beschreibungen	1-2
1.1.3 Piktogramme in diesem Handbuch	1-2
1.1.4 Verwendete Begriffe	1-2
1.2 Versionskennung der Funktionsbibliothek	1-3
2 Einführung	2-1
2.1 Einleitung	2-1
2.2 Projektierungskonzept	2-1
2.3 Minimalkonfiguration	2-2
3 Anwendungsbeispiele	3-1
3.1 Modulares, dezentrales I/O-System	3-1
3.2 Kompaktes, dezentrales I/O-System	3-5
4 Funktionsblöcke zur Parametrierung	4-1
4.1 L_IOParComGuarding - Überwachungsfunktion	4-2
4.2 L_IOParAlnModule - Parametrierfunktion	4-4
4.3 L_IOParAOutModule - Parametrierfunktion	4-7
4.4 L_IOParAIAOModule - Parametrierfunktion	4-9
4.5 L_IOParSSIModule - Parametrierfunktion	4-12
4.6 L_IOParCounterModule - Zählerfunktion	4-14
4.7 L_IOParCounterDIModule - Zählerfunktion	4-16
4.8 L_IOParPDO15 - Parametrierfunktion	4-18
4.9 L_IOParPDO610 - Parametrierfunktion	4-22
4.10 L_IOParCompactModule - Parametrierfunktion	4-26
5 Funktionsblöcke zur Prozessdatenverarbeitung	5-1
5.1 L_IOData15 - Daten koordinieren	5-2
5.2 L_IOData610 - Daten koordinieren	5-5
5.3 L_IOCompactModule - Daten koordinieren	5-8
5.4 L_IOCounterDataToIO - Zählerfunktion	5-11
5.5 L_IOCounterDataFromIO - Zählerfunktion	5-13
5.6 L_IOCounterDIModuleDataToIO - Zählerfunktion	5-14
5.7 L_IOCounterDIModuleDataFromIO - Zählerfunktion	5-15
5.8 L_IOSSIDataToIO - Zählerfunktion	5-16
5.9 L_IOSSIDataFromIO - Zählerfunktion	5-17
5.10 L_IOConvByteToByteArray - Konvertierungsfunktion	5-18
5.11 L_IOConvByteArrayToByte - Konvertierungsfunktion	5-19
5.12 L_IODInModule - Bitkonvertierung	5-20
5.13 L_IODOOutModule - Bitkonvertierung	5-21



Funktionsbibliothek LenzeIOSystem.lib

Inhalt

5.14 L_IOInModule - Signalkonvertierung	5-22
5.15 L_IOAOutModule - Signalkonvertierung	5-23
6 Anhang	6-1
6.1 Signalfunktionen für analoge Eingänge EPM-T310	6-1
6.2 Signalfunktionen für analoge Ausgänge EPM-T320	6-4
6.3 Signalfunktionen für analoge Ein- und Ausgänge EPM-T330	6-6
6.4 Normierung der Messwerte	6-9
6.5 Auswahl der Zähler-Modi EPM-T410	6-10
7 Index	7-1



1 Vorwort und Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch enthält Informationen zu den Funktionsbibliotheken **LenzeIOSystem.lib** und **LenzeIOSystemVxxxx** für das **Drive PLC Developer Studio**.

1.1.1 Konventionen in diesem Handbuch

Dieses Handbuch verwendet folgende Konventionen zur Unterscheidung von verschiedenen Arten von Informationen:

Variablenbezeichner

sind im erklärenden Text in Kursivschrift dargestellt:

- "Mit *bReset...*"



Tip!

Informationen zu den Konventionen, die für die Variablenbezeichner von Lenze Systembausteinen, Funktionsblöcken sowie Funktionen verwendet werden, um eine einheitliche und durchgängige Benennung zu gewährleisten und dadurch die Lesbarkeit von SPS-Programmen zu fördern, finden Sie im Anhang der DDS-Online-Dokumentation "Einführung in die IEC 61131-3 Programmierung".

Lenze Funktionen/Funktionsblöcke

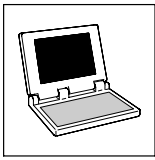
erkennen Sie am Namen. Sie beginnen immer mit einem "L_":

- "Mit dem FB **L_IOPar...**"

Programmlistings

sind in der Schriftart "Courier" aufgeführt, wobei Schlüsselwörter fett ausgezeichnet sind:

- "**IF** (ReturnValue < 0) **THEN...**"



Funktionsbibliothek LenzeIOSystem.lib

Vorwort und Allgemeines Über dieses Handbuch

1.1.2 Aufbau der Beschreibungen

Die Beschreibungen der einzelnen Funktionen/Funktionsblöcke sowie Systembausteine in diesem Handbuch sind einheitlich nach folgendem Schema aufgebaut:

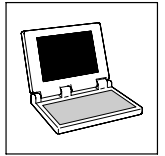
	Funktion	Funktionsblock (FB)/ Systembaustein (SB)
①	Überschrift mit Funktion und Bezeichner	
②	Deklaration der Funktion: • Datentyp des Rückgabewertes • Funktionsbezeichner • Liste der Übergabeparameter	-
③	Kurzbeschreibung mit den wichtigsten Eigenschaften	
④	Grafische Darstellung der Funktion mit den zugehörigen Variablen • Übergabeparameter • Rückgabewert	Grafische Darstellung des FB/SB mit den zugehörigen Variablen • Eingangsvariablen • Ausgangsvariablen
⑤	Tabelle mit Informationen zu den Übergabeparametern: • Bezeichner • Datentyp • Einstellmöglichkeiten • Info	Tabelle mit Informationen zu den Ein- und Ausgangsvariablen: • Bezeichner • Datentyp • Variablentyp • Einstellmöglichkeiten • Info
⑥	Tabelle mit Informationen zum Rückgabewert: • Datentyp des Rückgabewertes • Mögliche Rückgabewerte und ihre Bedeutung	-
⑦	Weitere Informationen (Anmerkungen, Tips, Anwendungsbeispiele, etc.)	

1.1.3 Piktogramme in diesem Handbuch

	verwendete Piktogramme	Signalwörter
Warnung vor Sachschäden		Stop! Warnt vor möglichen Sachschäden . Mögliche Folgen bei Missachtung: Beschädigung des Antriebsreglers/Antriebssystems oder seiner Umgebung.
Sonstige Hinweise		Tip! Hinweis! Kennzeichnet einen Tip bzw. Hinweis.

1.1.4 Verwendete Begriffe

Begriff	Im folgenden Text verwendet für
DDS	Drive PLC Developer Studio
FB	Funktionsblock
GDC	Global Drive Control (Parametrier-Programm von Lenze)
Parameter-Codestellen	Codestellen zum Einstellen der Funktionalität eines Funktionsblocks
PLC	9300 Servo PLC - Drive PLC - Achsmodul ECSxA "Application"
SB	Systembaustein



1.2 Versionskennung der Funktionsbibliothek

Die Version der Funktionsbibliothek können Sie über die globale Konstante `C_w[Funktionsbibliotheksname]Version` abfragen.

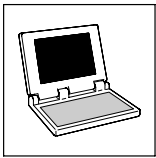
Versionskennung ab PLC Software-Stand 7.x:

Konstante	Bedeutung	Beispiel Wert
<code>C_w[Funktionsbibliotheksname]VersionER</code>	Externes Release	01
<code>C_w[Funktionsbibliotheksname]VersionEL</code>	Externes Level	05
<code>C_w[Funktionsbibliotheksname]VersionIR</code>	Internes Release	00
<code>C_w[Funktionsbibliotheksname]VersionBN</code>	Build-Nummer	00

Version: 01 05 00 00

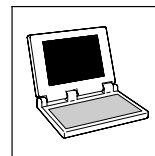
Der Wert dieser Konstanten ist hexadezimal-kodiert.

- Im Beispiel bedeutet der Wert "01050000" Version "1.05".



Funktionsbibliothek LenzeIOSystem.lib

Vorwort und Allgemeines



2 Einführung

2.1 Einleitung

Zur Unterstützung des dezentralen I/O-Systems durch Lenze PLC Produkte werden dem Anwender entsprechende Funktionsblöcke in Funktionsbibliotheken zur Verfügung gestellt.

Diese Funktionsblöcke werden in einem IEC 61131 Programm eingebunden und dienen sowohl zur Parametrierung des I/O Systems als auch zur Prozessdatenverarbeitung. Sie stellen digitale und analoge Ein- und Ausgangsinformationen des I/O Systems zur Verfügung.

LenzeIOSystem.lib

Mit dieser Bibliothek werden alle Einstellmöglichkeiten zur Verfügung gestellt. Konfigurationen vom Anwender über Usercodestellen sind nicht notwendig.

2.2 Projektierungskonzept

Funktionsblöcke treten in den Funktionsbibliotheken in zwei Typen auf.

- Parameterdatenbausteine
- Prozessdatenbausteine

Parameterdatenbausteine

Diese Funktionsblöcke ermöglichen das Verändern von Indexinformationen im dezentralen I/O-System und das Übertragen von festgelegten Daten. Während der Initialisierungsphase werden diese Daten nur einmal zum I/O-System übertragen.

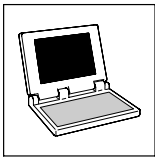
- Parametrierung der Kommunikation
 - Durch diese Bausteine werden die Kommunikationsbeziehungen (Identifizierung für PDO-Kommunikation, Übertragungsart, Übertragungszeit usw.) zwischen PLC und I/O-System parametriert.
 - Diese Einstellungen sind für den Datenaustausch zwingend erforderlich.
- Parametrierung der Module und Überwachungsfunktionen
 - Die Funktionsblöcke sind notwendig bei individueller Parametrierung von analogen Modulen, individueller Parametrierung von Zähler-Modulen, Kommunikationsüberwachung

Prozessdatenbausteine

Mit Hilfe dieser Funktionsblöcke können die aktuellen Prozessdateninformationen von den Eingängen des I/O-Systems in dem PLC-Programm ausgewertet und weiter verarbeitet werden.

Informationen und Daten können über die Ausgangsmodule an den Prozess übergeben werden.

- Prozessdatenbausteine zum Senden und Empfangen von E/A-Daten zwischen PLC und I/O-System.
- Prozessdatenbausteine zum Konvertieren der E/A-Signale.



Funktionsbibliothek LenzeIOSystem.lib

Einführung

Minimalkonfiguration

2.3 Minimalkonfiguration

Zum Einbinden des I/O-Systems muss eine Minimalkonfiguration vorgenommen werden. Sie besteht aus einem Parameterdatenbaustein **L_IOParPDO15** zur Parametrierung der Kommunikationsbeziehungen und einem Prozessdatenbaustein **L_IOData15** zur Auswertung der Eingangs-/Ausgangsinformation.

Die Informationen des Parameterbausteins werden zum I/O-System und zum Funktionsblock für den Prozessdatenaustausch zwischen PLC und I/O-Systemen übertragen.

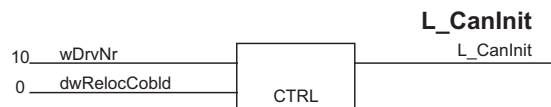
Zwischen dem Ausgang des Parameterbausteins und dem Eingang des Prozessdatenbausteins ist ein Datenaustausch mit einer globalen Variablen möglich.



Hinweis!

Die Funktionsbibliotheken für das dezentrale I/O-System basieren auf freie CAN-Objekte. Deshalb ist eine Initialisierung des CAN-Treibers über den Baustein **L_CANInit** notwendig.

Den Baustein **L_CANInit** finden Sie in der Lenze Bibliothek **LenzeCanDrv.lib**.



10 = Systembus

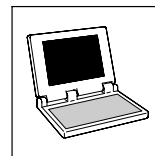
0 = ID-Bereich 832..895 (nur ServoPLC)



Hinweis!

Bei der 9300 ServoPLC ist der Identifizierungsbereich eingeschränkt.

Bei der Verwendung der Funktionsblöcke **L_IOParPDO15** bzw. **L_IOParPDO610** müssen die Cob-Id's ab PDO4 angepasst werden.



3 Anwendungsbeispiele

Die Beispiele wurden mit dem DDS Version 2.0 erstellt. Sie finden die Beispiele zum I/O-System im Verzeichnis: C:\Programme\Lenze\DDS_P_2_20\Projects\IOSystem\Samples

3.1 Modulares, dezentrales I/O-System

Das modulare, dezentrale I/O-System mit acht digitalen Ein-/ und Ausgängen wird, unter Verwendung der LenzeIOSystem.lib, in ein PLC-Programm eingebunden.

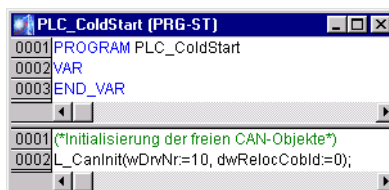
Während des Vorgangs müssen folgende Schritte in der in der angegebenen Reihenfolge durchgeführt werden.

1. CAN-Treiber initialisieren
2. Kommunikationsbedingungen parametrieren
3. Prozessdatenaustausch zwischen PLC und I/O-System programmieren
4. Codestellen initialisieren

CAN-Treiber initialisieren

Die Kommunikation zwischen PLC und I/O-System basiert auf den "Freien CAN-Objekten" (Control, Alarm, Diagnose...). Bei dem DDS ab Version 2.x sind diese Objekte verfügbar.

Zum Initialisieren der freien CAN-Objekte muss die Funktion **L_CANInit** einmalig in der System-POE **PLC_ColdStart** aufgerufen werden.

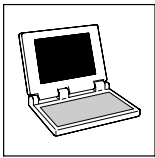


Kommunikationsbedingungen parametrieren

Der Funktionsbaustein **L_IOParPDO15** zur Parametrierung der Kommunikationsbedingungen von PDO1-PDO5 wird in den AS-Editor eingebunden und ist in jedem Projekt zwangsläufig zu verwenden. Dieser Funktionsblock legt die Kommunikationsparameter im I/O-System fest.

Die Werte werden als Struktur (STRUCT) an den Prozessdatenbaustein **L_IOData15** übergeben. Die Kommunikationsparameter der Geräte (Antriebsregler und I/O-System) sind damit aufeinander abgestimmt.

Bei Ausführung des Programms wird dieser Schritt nur einmal am Anfang ausgeführt.

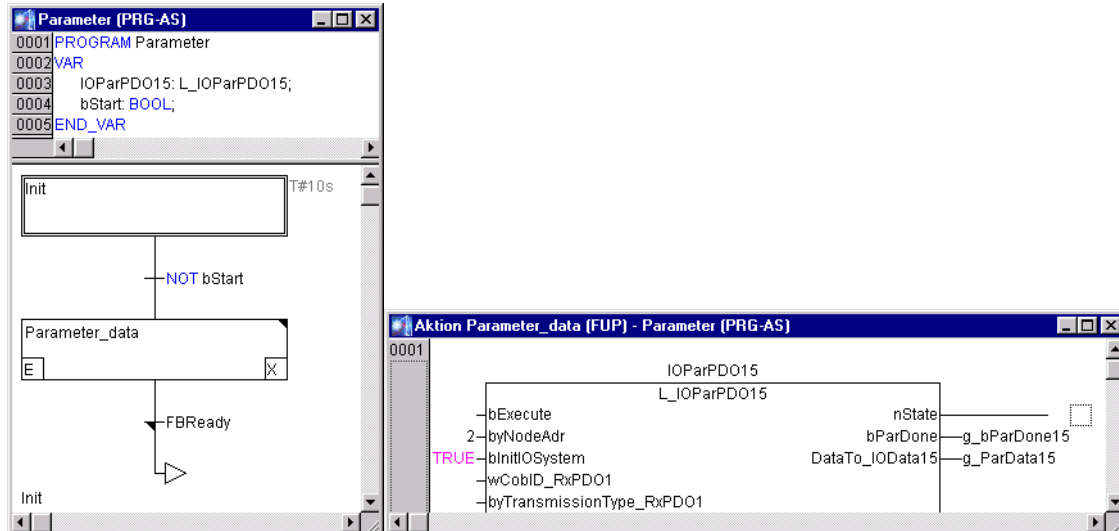


Funktionsbibliothek LenzeIOSystem.lib

Anwendungsbeispiel

Modulares, dezentrales I/O-System

Es folgt das Programm **Parameter**, das im AS-Editor erzeugt werden kann. Mit dem Aufruf der Funktionsblock Instanz hat der Schritt **Parameter_data** die im FUP-Editor erzeugte Aktion mit dem Aufruf der Funktionsblock Instanz.

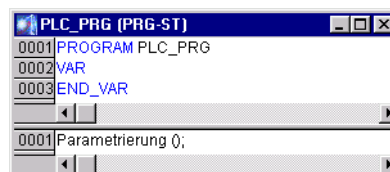


Hinweis!

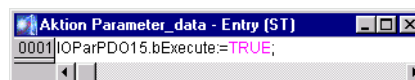
- Geben Sie über Menübefehl **Extras→Schritt Attribute...** eine minimale Zeit von zehn Sekunden für den Schritt **Init** ein. Korrekte Eingabe: T#10s

Es ist sicherzustellen, dass der Schritt **Parameter_data** nicht abgearbeitet wird bevor die Initialisierungsphase des I/O-Systems abgeschlossen ist. Das wird mit einer minimalen Zeit von zehn Sekunden erreicht.

Die Abarbeitung dieser POE muss zyklisch erfolgen, deshalb wird das Programm im PLC_PRG (Hauptprogramm) aufgerufen. Die Ablaufsprache stellt ein zyklische Task sicher.



Über die Eingangsaktion im Schritt **Parameter_data** wird der Parametriervorgang gestartet.

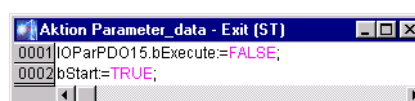


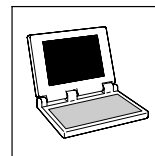
Nach abgeschlossener Parametrierung wird in der Ausgangsaktion des Schritts **Parameter_data** der Eingang **bExecute** auf FALSE gesetzt und die Variable **bStart** auf TRUE.

Die Variable **bStart** stellt sicher, dass der Schritt **Parameter_data**

- beim Einschalten
- beim Reset und anschließendem Starten durchgeführt wird.

Nach der Parametrierung bleibt das PLC_PRG (Hauptprogramm) im Schritt **Init** stehen. Es belastet das Zielsystem im weiteren Ablauf nicht.

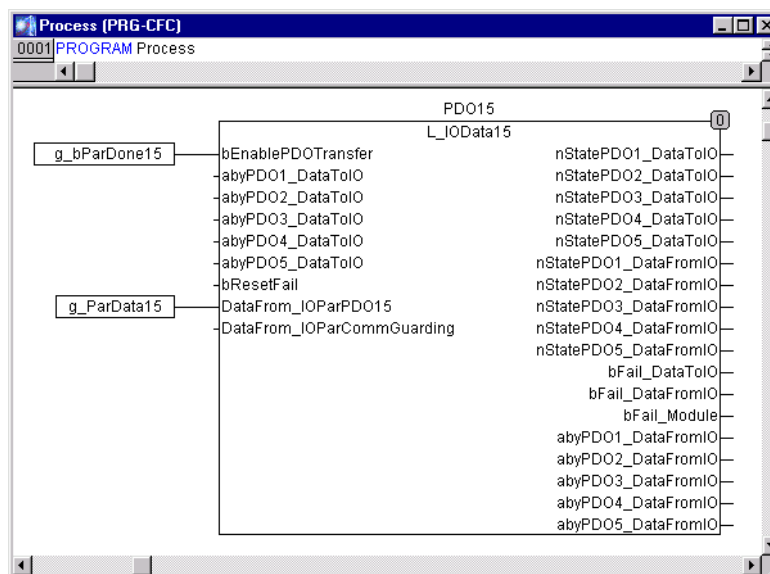




Prozessdatenaustausch zwischen PLC und I/O-System programmieren

Mit dem CFC-Editor wird ein Programm mit dem Namen **Process** erzeugt.

- Deklarieren Sie in **Process** eine Instanz des Funktionsbausteins **L_IOData15**.
- Erzeugen Sie zwei Eingänge im CFC-Editor, vergeben Sie die Namen *g_bParDone15* und *g_ParData15* und verbinden Sie diese, wie im Bild gezeigt, mit den entsprechenden Eingängen des Funktionsbausteins **L_IOData15**.



Das Auswerten der Eingangsdaten und das Übergeben der Ausgangsdaten kann mit einigen zur Verfügung stehenden Konvertierungsbausteinen durchgeführt werden.

Der Funktionsbaustein **L_IOData15** überträgt Ausgangsdaten an das I/O-System und empfängt Eingangsdaten vom I/O-System.

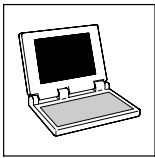
Am Eingang *DataFrom_IOParPDO15* werden die CAN Kommunikationsbeziehungen zwischen PLC und I/O-System von dem Parametrierblock **L_IOParPDO15** an den Prozessdatenblock **L_IOData15** übertragen.

Der Prozessdatenverkehr wird nach der Parametrierung durch ein High-Signal am Eingang *bEnablePDOTransfer* gestartet.

Konvertierung der Daten zur Auswertung der einzelnen Kanäle

Das Programm **Process** ist folgendermaßen zu erweitern.

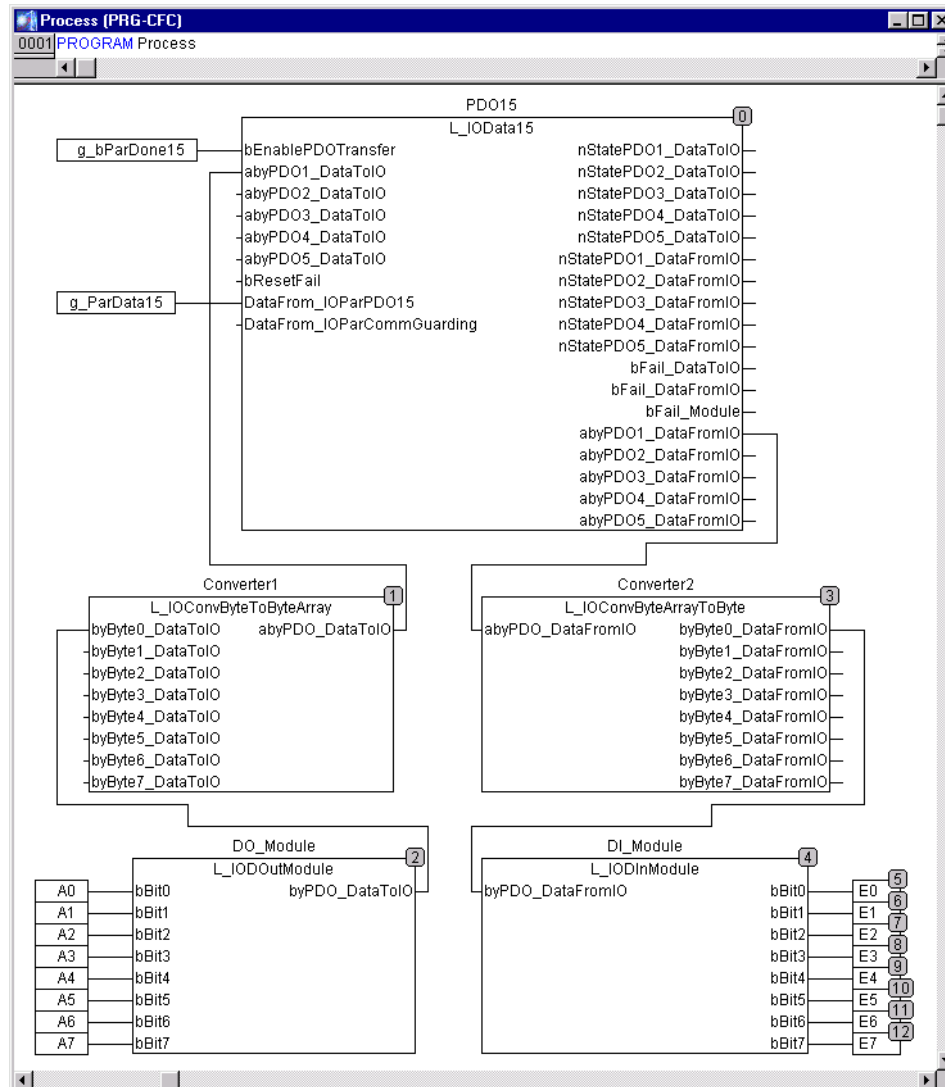
- Aktivieren Sie mit dem Menübefehl **Extras→Reihenfolge**, damit die Reihenfolge angezeigt wird.
- Verschieben Sie die einzelnen Bausteine in die Position, damit Sie den Programmablauf nachvollziehen können.



Funktionsbibliothek LenzeIOSystem.lib

Anwendungsbeispiel

Modulares, dezentrales I/O-System



Die Sende-/ und Empfangsdaten werden vom Funktionsbaustein **L_IODData15** als 8 Byte Information über das CAN Telegramm zur Verfügung gestellt.

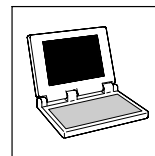
Über die zwei Konverter werden die Informationen zu einzelnen Bytes konvertiert.

Über ein digitales Eingangs-/ und Ausgangsmodul werden die einzelnen Bits abgebildet.

Codestellen initialisieren

Folgende Codestellen sind in der PLC zu initialisieren

- Code C0352
Subcode 0 = 1
PLC als CAN-Master definieren, damit die Konfiguration in den Zustand Operational versetzt wird.
- Code C0356
Subcode 1 = 10000
Erhöhen der Boot-Up Zeit auf 10s
- Code C2104
Subcode 0 = 1
Automatischer Start des PLC-Programms nach Netz-Einschalten



3.2 Kompaktes, dezentrales I/O-System

Das Kompaktmodul EPM - T832 mit 24 digitalen Eingängen und acht digitalen Ausgängen wird, unter Verwendung der LenzeIOSystem.lib, in einem PLC-Programm eingebunden.

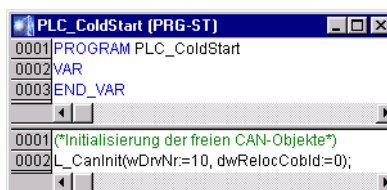
Während des Vorgangs müssen folgende Schritte in der in der angegebenen Reihenfolge durchgeführt werden.

1. CAN-Treiber initialisieren
2. Kommunikationsbedingungen parametrieren
3. Prozessdatenaustausch zwischen PLC und I/O-System programmieren
4. Codestellen initialisieren

CAN-Treiber initialisieren

Die Kommunikation zwischen PLC und I/O-System basiert auf den "Freien CAN-Objekten" (Control, Alarm, Diagnose...). Bei dem DDS ab Version 2.x sind diese Objekte verfügbar.

Zum Initialisieren der freien CAN-Objekte muss die Funktion **L_CANInit** einmalig in der System-POE **PLC_ColdStart** aufgerufen werden.



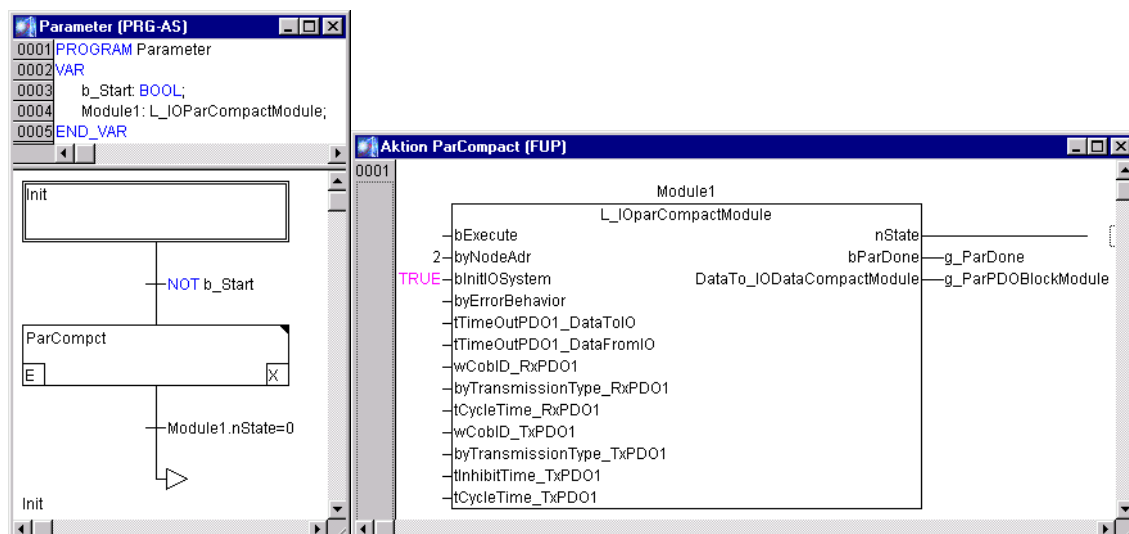
Kommunikationsbedingungen parametrieren

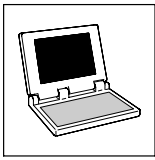
Der Funktionsbaustein **L_IOParCompactModule** zur Parametrierung der Kommunikationsbedingungen wird in den AS-Editor eingebunden und ist in jedem Projekt zwangsläufig zu verwenden. Dieser Funktionsblock legt die Kommunikationsparameter im I/O-System fest.

Die Werte werden als Struktur (STRUCT) an den Prozessdatenbaustein **L_IOCompactModule** übergeben. Die Kommunikationsparameter der Geräte (Antriebsregler und I/O-System) sind damit aufeinander abgestimmt.

Bei Ausführung des Programms wird dieser Schritt nur einmal am Anfang ausgeführt.

Es folgt das Programm **Parameter**, das im AS-Editor erzeugt werden kann. Mit dem Aufruf der Funktionsblock Instanz hat der Schritt **ParCompact** die im FUP-Editor erzeugte Aktion mit dem Aufruf der Funktionsblock Instanz.





Funktionsbibliothek LenzeIOSystem.lib

Anwendungsbeispiel

Kompaktes, dezentrales I/O-System

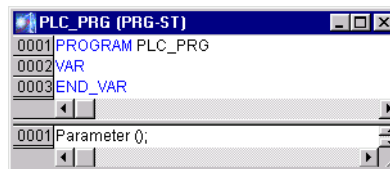


Hinweis!

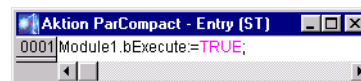
- Geben Sie über Menübefehl **Extras→Schritt Attribute...** eine minimale Zeit von zehn Sekunden für den Schritt **Init** ein. Korrekte Eingabe: T#10s

Es ist sicherzustellen, dass der Schritt **Parameter_data** nicht abgearbeitet wird bevor die Initialisierungsphase des I/O-Systems abgeschlossen ist. Das wird mit einer minimalen Zeit von zehn Sekunden erreicht.

Die Abarbeitung dieser POE muss zyklisch erfolgen, deshalb wird das Programm im PLC_PRG (Hauptprogramm) aufgerufen. Die Ablaufsprache stellt ein zyklische Task sicher.



Über die Eingangsaktion im Schritt **ParCompact** wird der Parametriervorgang gestartet.

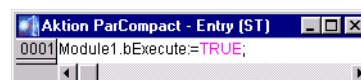


Nach abgeschlossener Parametrierung wird in der Ausgangsaktion des Schritts **ParCompact** der Eingang *bExecute* auf FALSE gesetzt und die Variable *bStart* auf TRUE.

Die Variable *bStart* stellt sicher, dass der Schritt **ParCompact**

- beim Einschalten
- beim Reset und anschließendem Starten durchgeführt wird.

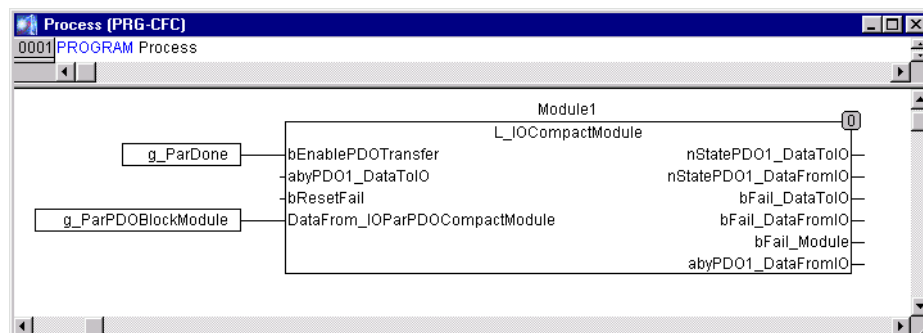
Nach der Parametrierung bleibt das PLC_PRG (Hauptprogramm) im Schritt **Init** stehen. Es belastet das Zielsystem im weiteren Ablauf nicht.



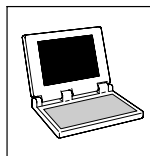
Prozessdatenaustausch zwischen PLC und I/O-System programmieren

Mit dem CFC-Editor wird ein Programm mit dem Namen **Process** erzeugt.

- Deklarieren Sie in **Process** eine Instanz des Funktionsbausteins **L_IOCompactModule**.
- Erzeugen Sie zwei Eingänge im CFC-Editor, vergeben Sie die Namen *g_bParDone15* und *g_ParData15* und verbinden Sie diese, wie im Bild gezeigt, an die Eingänge vom Funktionsbaustein **L_IOCompactModule**.



Das Auswerten der Eingangsdaten und das Übergeben der Ausgangsdaten kann mit einigen zur Verfügung stehenden Konvertierungsbausteinen durchgeführt werden.



Der Funktionsbaustein **L_IOCompactModule** überträgt Ausgangsdaten an das I/O-System und empfängt Eingangsdaten vom I/O-System.

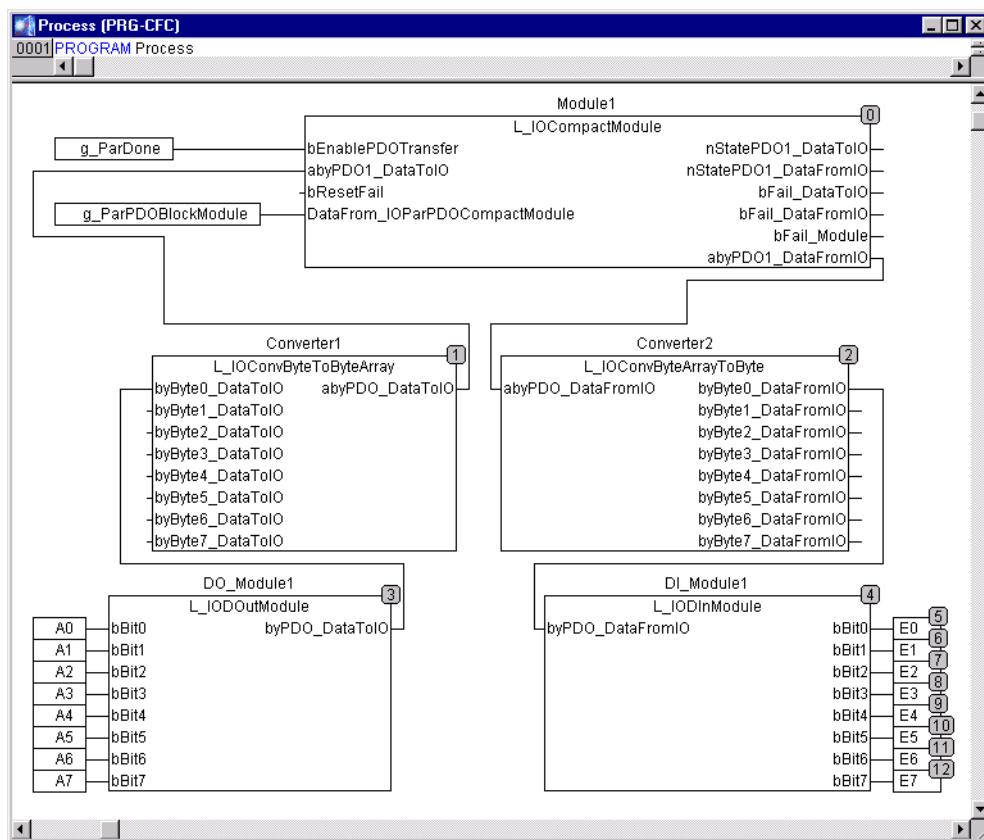
Am Eingang *DataFrom_IOParPDO15* werden die CAN Kommunikationsbeziehungen zwischen PLC und I/O-System von dem Parametrierblock **L_IOParCompactModule** an den Prozessdatenblock **L_IOCompactModule** übertragen.

Der Prozessdatenverkehr wird nach der Parametrierung durch ein High-Signal am Eingang *bEnablePDOTransfer* gestartet.

Konvertierung der Daten zur Auswertung der einzelnen Kanäle

Das Programm **Process** ist folgendermaßen zu erweitern.

- Aktivieren Sie mit dem Menübefehl **Extras→Reihenfolge**, damit die Reihenfolge angezeigt wird.
- Verschieben Sie die einzelnen Bausteine in die Position, damit Sie den Programmablauf nachvollziehen können.

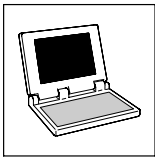


Die Sende-/ und Empfangsdaten werden vom Funktionsbaustein **L_IOCompactModule** als 8*8 Byte Information des CAN Telegramms zur Verfügung gestellt.

Über die zwei Konverter werden die Informationen zu einzelnen Bits konvertiert.

Über ein digitales Eingangs-/ und Ausgangsmodul können die einzelnen Bits abgebildet werden.

Soll das Beispiel von acht Ein-/ Ausgänge auf 24 Eingänge erweitert werden, müssen zwei weitere Funktionsblöcke **L_IODInModule** erzeugt und mit dem Konverter **L_IOConvByteArrayToByte** verbunden werden.



Funktionsbibliothek LenzelOSystem.lib

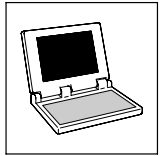
Anwendungsbeispiel

Kompaktes, dezentrales I/O-System

Codestellen initialisieren

Folgende Codestellen sind in der PLC zu initialisieren

- Code C0352
Subcode 0 = 1
PLC als CAN-Master definieren, damit die Konfiguration in den Zustand Operational versetzt wird.
- Code C0356
Subcode 1 = 10000
Erhöhen der Boot-Up Zeit auf 10s
- Code C2104
Subcode 0 = 1
Automatischer Start des PLC-Programms nach Netz-Einschalten

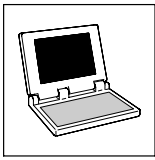


4 Funktionsblöcke zur Parametrierung

In diesem Kapitel erhalten Sie Informationen über Funktionsblöcke zum Parametrieren.

Folgende Funktionen und Module werden parametriert.

- Überwachungsfunktion
- Analoge Eingangsmodule
- Analoge Ausgangsmodule
- Zählermodule
- Verbindung zwischen PLC-Steuerung und I/O-System
- Verbindung zwischen PLC-Steuerung und Kompakt Modul



Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOParComGuarding - Überwachungsfunktion

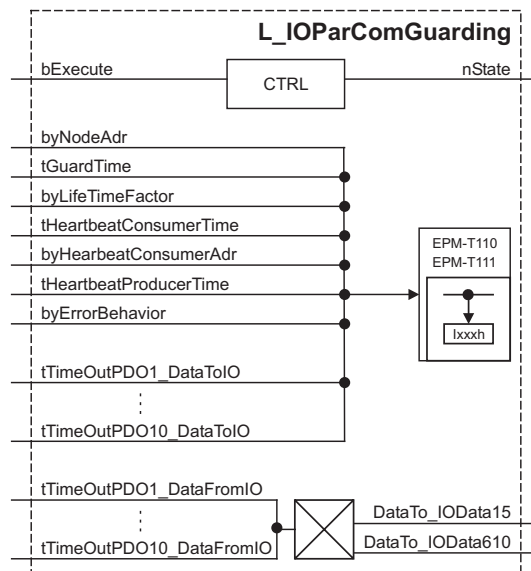
4.1 L_IOParComGuarding - Überwachungsfunktion

Dieser Funktionsblock parametriert über die Eingangsbeschaltung die Überwachungsfunktionen, die das I/O-System zur Verfügung stellt.

Dazu gehört

- die NodeGuarding-, Heartbeat- und eine lenzespezifische Überwachungsfunktion.
- das Einstellen des Schaltverhaltens der Ausgangskanäle im Fehlerfall über den Eingang *byErrorBehavior*.

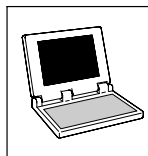
Am Ausgang des Funktionsblocks wird der Übertragungsstatus angezeigt.



Hinweis!

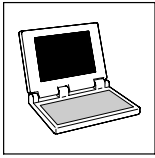
Zur Auswertung der Funktionen NodeGuarding und Heartbeat sind die FB's der LenzeCanDSxDrv.lib zu verwenden.

⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE ↗ TRUE Schreibauftrag wird ausgeführt.
byNodeAdr	Byte	Geräteadresse: 1 - 63
tGuardTime	Time	Nodeguarding: Zeitintervall. Innerhalb der Guard Time muss ein NMT-Master das I/O-System abfragen. default t = 0 ms t = 0 - 65535 ms
byLifeTimeFactor	Byte	Lifeguarding: Life Time Factor x Guard Time = Life Time, nach Ablauf dieser Zeit wird das I/O-System in einen vordefinierten Zustand geschaltet.
tHeartbeatConsumerTime	Time	Heartbeat: Überwachungsintervall, wird innerhalb des Intervalls kein Telegramm vom Heartbeat Producer empfangen, wird das I/O-System in einen vordefinierten Zustand geschaltet. t = 0 - 65535 ms
byHeartbeatConsumerAdr	Byte	NodelD des zu überwachenden Gerätes. (NodelD vom HeartbeatProducer)



⇒ Eingänge (Variablentyp: VAR_INPUT)										
Variablenname	Datentyp	Info/Einstellmöglichkeiten								
tHeartbeatProducerTime	Time	Einstellung der Übertragungszeit für ein Heartbeat-Telegramm t = 0 - 65535ms								
byErrorBehavior	Byte	Einstellung des Buszustands / Fehlerverhalten bei einer kritischen Gerätestörung <table><tr><td>0</td><td>Pre-Operational (default)</td></tr><tr><td>1</td><td>keine Statusänderung</td></tr><tr><td>2</td><td>Stopp</td></tr><tr><td>3</td><td>Reset nach 2 Sekunden</td></tr></table>	0	Pre-Operational (default)	1	keine Statusänderung	2	Stopp	3	Reset nach 2 Sekunden
0	Pre-Operational (default)									
1	keine Statusänderung									
2	Stopp									
3	Reset nach 2 Sekunden									
tTimeOutPDO1_DataToIO ... tTimeOutPDO10_DataToIO	Time	Überwachungszeit für Prozesseingangsdaten (lenzespezifisch). Werden innerhalb der Überwachungszeit auf dem ausgewählten PDO keine Daten empfangen nimmt das I/O-System den unter <i>byErrorBehavior</i> eingestellte Zustand an. <table><tr><td>default</td><td>t = 0 ms t = 0 - 65535 ms</td></tr></table>	default	t = 0 ms t = 0 - 65535 ms						
default	t = 0 ms t = 0 - 65535 ms									
tTimeOutPDO1_DataFromIO ... tTimeOutPDO10_DataFromIO	Time	Überwachungszeit für Prozessausgangsdaten. Werden innerhalb der Überwachungszeit auf dem ausgewählten PDO keine Daten vom I/O-System an die PLC gesendet. Wird im Funktionsblock L_IODataXXX ein Zeitüberwachungsfehler angezeigt. <table><tr><td>default</td><td>t = 0 ms t = 0 - 65535 ms</td></tr></table>	default	t = 0 ms t = 0 - 65535 ms						
default	t = 0 ms t = 0 - 65535 ms									

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)			
Bezeichner	Datentyp	Wert/Bedeutung	
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.	
		0	Der Sendeauftrag ist ausgeführt
		1	Sendeauftrag abgesetzt - Bearbeitung läuft
		2001	Fehler beim Bus-Transport aufgetreten
		2002	Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004	Zugriff verweigert - falsches Zielsystem: Subindex existiert nicht
		2005	Zugriff verweigert - falsches Zielsystem: Index existiert nicht
		2013	Zugriff verweigert - falsches Zielsystem: ungültiger Wertebereich
		2111	Zugriff verweigert - falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117	Kommunikationskanal existiert nicht - s.a. Einstellung byCom-Channel
		2120	Anzahl der angeschlossenen Module ist zu groß - der Parametertransfer wurde abgebrochen
-10xx	Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten		
DataTo_IOData15	Struct	Struktur zum Prozessdatenblock L_IOData15 . Übergabe der Zeiten zur Überwachung der Prozessausgangsdaten an den Prozessdatenblock L_IOData15 . Globale Variable vom Typ Struct	
DataTo_IOData610	Struct	Struktur zum Prozessdatenblock L_IOData610 Übergabe der Zeiten zur Überwachung der Prozessausgangsdaten an den Prozessdatenblock L_IOData610 Globale Variable vom Typ Struct	



Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOParAlnModule - Parametrierfunktion

4.2 L_IOParAlnModule - Parametrierfunktion

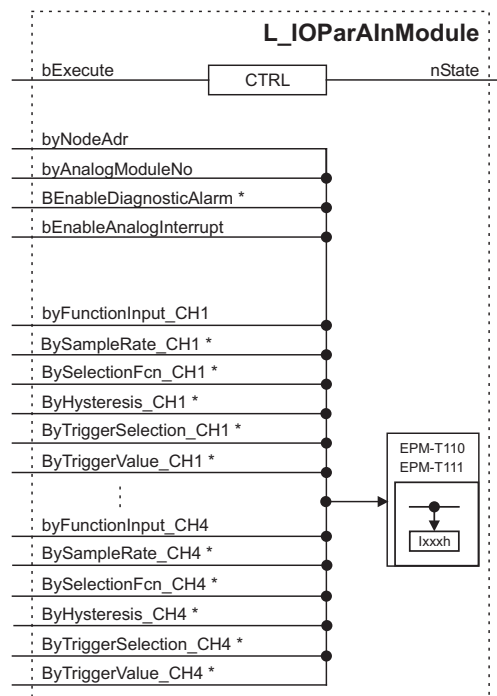
Mit dem Funktionsblock **L_IOParAlnModule** werden die analogen Eingangsmodule (EPM-T310 / T311 / T312) parametrierung. Durch eine High-Flanke am Eingang *bExecute* wird die Parametrierung gestartet.

Jedes Modul hat vier Kanäle wobei jeder Kanal individuell eingestellt werden kann.

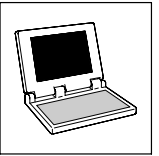
Zur Auswahl stehen

- Spannungsmessung
- Strommessung
- Widerstandsmessung
- Temperaturmessung

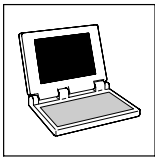
Am Ausgang des Funktionsblocks wird der Übertragungsstatus und die Informationen über die angeschlossenen Module angezeigt.



* Eingang/Funktion wird ausschließlich vom Typ EPM-T310 unterstützt.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE ↗ TRUE Schreibauftrag wird ausgeführt
byNodeAdr	Byte	Geräteadresse: 1 - 63
byAnalogModuleNo	Byte	Steckplatznummer • Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1. • Maximal können 16 analoge Module gesteckt werden.
bEnableDiagnosticAlarm	Bool	Diagnosealarm freischalten TRUE Diagnosealarm ist freigeschaltet
bEnableAnalogInterrupt	Bool	Analoge Datenübertragung freischalten TRUE Datenübertragung ist freigeschaltet
byFunctionInput_CH1 ... byFunctionInput_CH4	Byte	Funktion parametrieren default ±10V Funktionscode = 3B _H 6-1
bySampleRate_CH1 ... bySampleRate_CH4	Byte	Anzahl der Wandlungen pro Sekunde auswählen 0 15 Wandlungen 1 30 Wandlungen 2 60 Wandlungen 3 123 Wandlungen 4 168 Wandlungen 5 202 Wandlungen 6 3.7 Wandlungen 7 7.5 Wandlungen
bySelectionFcn_CH1 ... bySelectionFcn_CH4	Byte	Selektion 0 Selektion deaktiviert 1 2 aus 3 Werte verwenden 2 4 aus 6 Werte verwenden
byHysteresis_CH1 ... byHysteresis_CH4	Byte	Hysteresis 0 Hysteresis deaktiviert 1 Hysteresis ±8 2 Hysteresis ±16
byTriggerSelection_CH1 ... byTriggerSelection_CH4	Byte	Übertragungsergebnis 0 Voreinstellungen verwenden 1 oberer Grenzwert überschritten 2 unterer Grenzwert unterschritten 3 Eingangswert hat sich um einen definierten Wert geändert.
byTriggerValue_CH1 ... byTriggerValue_CH4	DWord	Wertvorgabe für ein Übertragungsereignis 2 default

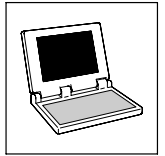


Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOParAlnModule - Parametrierfunktion

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametrierbar.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft, XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



4.3 L_IOParAOutModule - Parametrierfunktion

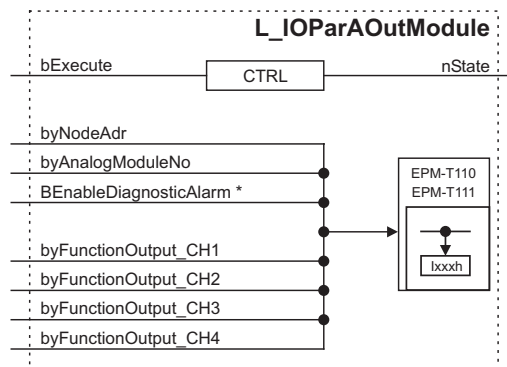
Mit dem Funktionsblock **L_IOParAOutModule** werden die analogen Ausgangsmodule (EPM-T320 / T321 / T322) parametrieren. Durch eine High-Flanke am Eingang *bExecute* wird die Parametrierung gestartet.

Jedes Modul hat vier Kanäle wobei jeder Kanal individuell eingestellt werden kann.

Zur Auswahl stehen

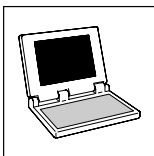
- Spannungsmessung
- Strommessung

Am Ausgang des Funktionsblocks wird der Übertragungsstatus angezeigt.



* Eingang/Funktion wird ausschließlich vom Typ EPM-T320 unterstützt.

⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE ↗ TRUE Schreibauftrag wird ausgeführt
byNodeAdr	Byte	Geräteadresse: 1 - 63
byAnalogModuleNo	Byte	Steckplatznummer • Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1. • Maximal können 16 analoge Module gesteckt werden.
bEnableDiagnosticAlarm	Bool	Diagnosealarm freischalten TRUE Diagnosealarm ist freigeschaltet
byFunctionOutput_CH1 ... byFunctionOutput_CH4	Byte	Funktion parametrieren default ±10V Funktionscode = 01 _h 6-4

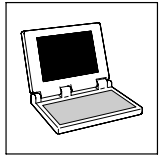


Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOParAOutModule - Parametrierfunktion

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametrierbar.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



4.4 L_IOParAIAOModule - Parametrierfunktion

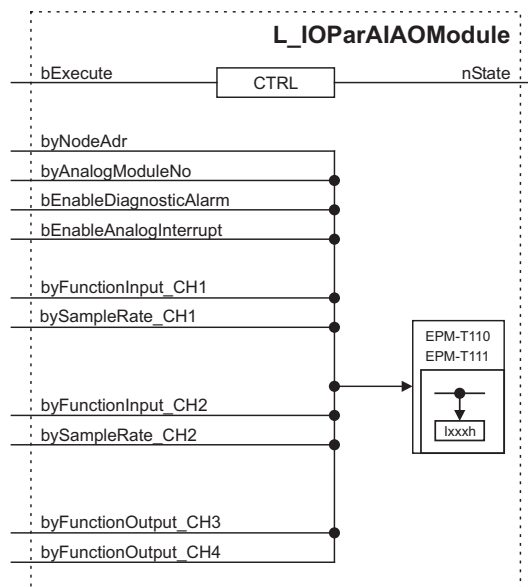
Mit dem Funktionsblock **L_IOParAIAOModule** werden die analogen Eingangs- und Ausgangsmodule EPM-T330 parametriert. Durch eine High-Flanke am Eingang *bExecute* wird die Parametrierung gestartet.

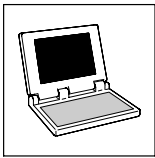
Jedes Modul hat vier Kanäle wobei jeder Kanal individuell eingestellt werden kann. Zur Auswahl stehen zwei analoge Eingänge und zwei analoge Ausgänge.

Folgende Funktionen können eingestellt werden.

- Spannungsmessung
- Strommessung
- Spannungsausgabe
- Stromausgabe

Am Ausgang des Funktionsblocks wird der Übertragungsstatus und die Informationen über die angeschlossenen Module angezeigt.



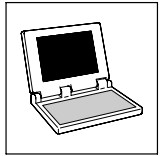


Funktionsbibliothek LenzeIOSystem.lib

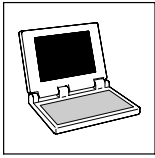
Funktionsblöcke

L_IOParAIAOModule - Parametrierfunktion

⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE → TRUE Schreibauftrag wird ausgeführt
byNodeAdr	Byte	Geräteadresse: 1 - 63
byAnalogModuleNo	Byte	Steckplatznummer • Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1. • Maximal können 16 analoge Module gesteckt werden.
bEnableDiagnosticAlarm	Bool	Diagnosealarm freischalten TRUE Diagnosealarm ist freigeschaltet
bEnableAnalogInterrupt	Bool	Analoge Datenübertragung freischalten TRUE Datenübertragung ist freigeschaltet
byFunctionInput_CH1 ... byFunctionInput_CH2	Byte	Funktion parametrieren default ±10V Funktionsparameter = 01 _h 6-6
bySampleRate_CH1 ... bySampleRate_CH2	Byte	Anzahl der Wandlungen pro Sekunde auswählen 0 15 Wandlungen 1 30 Wandlungen 2 60 Wandlungen 3 123 Wandlungen 4 168 Wandlungen 5 202 Wandlungen 6 3.7 Wandlungen 7 7.5 Wandlungen
byFunctionOutput_CH3 ... byFunctionOutput_CH4	Byte	Funktion parametrieren default ±10V Funktionscode = 01 _h 6-6



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft., XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



Funktionsbibliothek LenzeIOSystem.lib

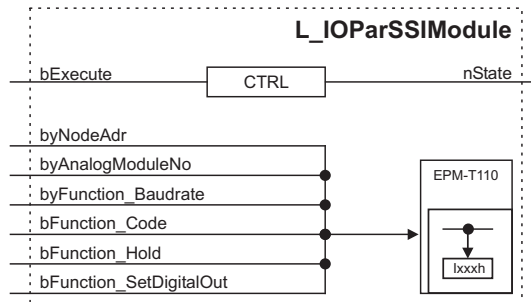
Funktionsblöcke

L_IOParSSIModule - Parametrierfunktion

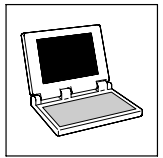
4.5 L_IOParSSIModule - Parametrierfunktion

Die SSI Schnittstelle (SSI = Synchron Serielles Interface) ist eine Synchron getaktete Schnittstelle.

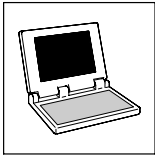
Das Funktionsmodul **L_IOParSSIModule** wandelt die serielle Information der Messwerte in parallele Information um. Die Daten werden anschließend der Steuerung zur Verfügung gestellt.



⇒ Eingänge (Variablentyp: VAR_INPUT)												
Variablenname	Datentyp	Info/Einstellmöglichkeiten										
bExecute	Bool	Schreibauftrag ausführen <table><tr><td>FALSE ↗ TRUE</td><td>Schreibauftrag wird ausgeführt</td></tr></table>	FALSE ↗ TRUE	Schreibauftrag wird ausgeführt								
FALSE ↗ TRUE	Schreibauftrag wird ausgeführt											
byNodeAdr	Byte	Geräteadresse: 1 - 63										
byAnalogModuleNo	Byte	Steckplatznummer - Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1. - Maximal können 9 Module gesteckt werden.										
byFunction_Baudrate	Byte	Baudrateneinstellung <table><tr><td>0</td><td>300 kBit/s</td></tr><tr><td>1</td><td>100 kBit/s</td></tr><tr><td>2</td><td>300 kBit/s</td></tr><tr><td>3</td><td>600 kBit/s</td></tr><tr><td>4</td><td>300 kBit/s</td></tr></table>	0	300 kBit/s	1	100 kBit/s	2	300 kBit/s	3	600 kBit/s	4	300 kBit/s
0	300 kBit/s											
1	100 kBit/s											
2	300 kBit/s											
3	600 kBit/s											
4	300 kBit/s											
bFunction_Code	Bool	Codierung <table><tr><td>default 0</td><td>Binär-Code</td></tr><tr><td>1</td><td>Gray-Code</td></tr></table>	default 0	Binär-Code	1	Gray-Code						
default 0	Binär-Code											
1	Gray-Code											
bFunction_Hold	Bool	Hold-Funktion <table><tr><td>0</td><td>nicht aktiv</td></tr><tr><td>1</td><td>aktiv</td></tr></table>	0	nicht aktiv	1	aktiv						
0	nicht aktiv											
1	aktiv											
bFunction_SetDigitalOut	Bool	Digitale Ausgänge <table><tr><td>0</td><td>nicht aktiv</td></tr><tr><td>1</td><td>aktiv</td></tr></table>	0	nicht aktiv	1	aktiv						
0	nicht aktiv											
1	aktiv											



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



Funktionsbibliothek LenzeIOSystem.lib

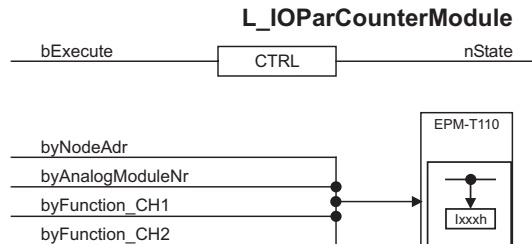
Funktionsblöcke

L_IOParCounterModule - Zählerfunktion

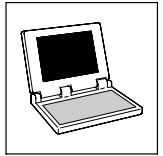
4.6 L_IOParCounterModule - Zählerfunktion

Die Parametrierung der Zählermodule erfolgt über den Funktionsblock **L_IOParCounterModule**.

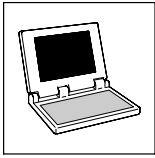
- Die Funktion des Zählers wird über die Eingänge *byFunction_CH1* und *byFunction_CH2* vorgegeben.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE → TRUE Schreibauftrag wird ausgeführt
byNodeAdr	Byte	Geräteadresse: 1 - 63
byAnalogModuleNo	Byte	Steckplatznummer • Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1. • Maximal können 16 analoge Module gesteckt werden.
byFunction_CH1	Byte	Funktion parametrieren default (2 * 32 Bit Zähler) Funktionscode 00 _h 6-10
byFunction_CH2	Byte	Funktion parametrieren default (2 * 32 Bit Zähler) Funktionscode 00 _h 6-10



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



Funktionsbibliothek LenzeIOSystem.lib

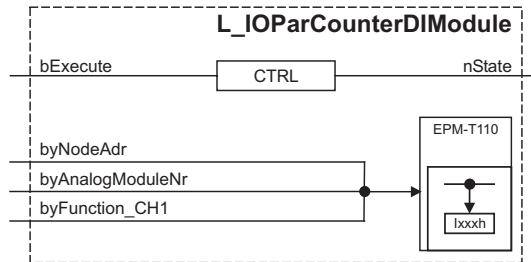
Funktionsblöcke

L_IOParCounterDIModule - Zählerfunktion

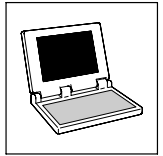
4.7 L_IOParCounterDIModule - Zählerfunktion

Die Parametrierung der Zählerfunktion des Moduls EPM-T430 erfolgt über den Funktionsblock **L_IOParCounterDIModule**.

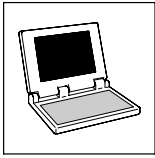
- Die Funktion des Zählers wird über die Eingänge *byFunction_CH1* vorgegeben.



⇒ Eingänge (Variablentyp: VAR_INPUT)												
Variablenname	Datentyp	Info/Einstellmöglichkeiten										
bExecute	Bool	Schreibauftrag ausführen FALSE ↗ TRUE Schreibauftrag wird ausgeführt										
byNodeAdr	Byte	Geräteadresse: 1 - 63										
byAnalogModuleNo	Byte	Steckplatznummer - Das erste eingebaute Analogmodul bekommt die Steckplatz-Nr. 1 - Maximal können 16 analoge Module gesteckt werden.										
byFunction_CH1	Byte	Zählfunktion <table><tr><td>0</td><td>Vierfach Impulsauswertung</td></tr><tr><td>1</td><td>Impuls- und Richtungsauswertung</td></tr><tr><td>2</td><td>Zähler aufwärts / abwärts</td></tr><tr><td>3</td><td>Frequenzmessung</td></tr><tr><td>4</td><td>Periodenmessung</td></tr></table>	0	Vierfach Impulsauswertung	1	Impuls- und Richtungsauswertung	2	Zähler aufwärts / abwärts	3	Frequenzmessung	4	Periodenmessung
0	Vierfach Impulsauswertung											
1	Impuls- und Richtungsauswertung											
2	Zähler aufwärts / abwärts											
3	Frequenzmessung											
4	Periodenmessung											



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametrierungsfähig.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten



Funktionsbibliothek LenzeIOSystem.lib

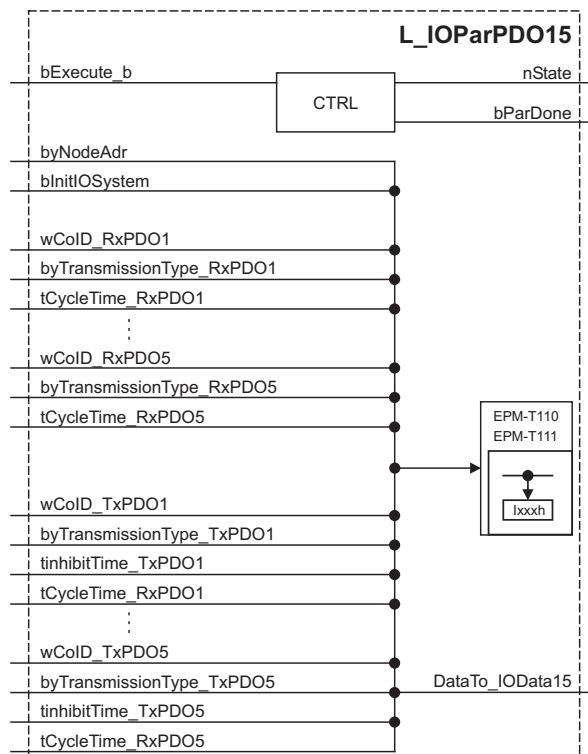
Funktionsblöcke

L_IOParPDO15 - Parametrierfunktion

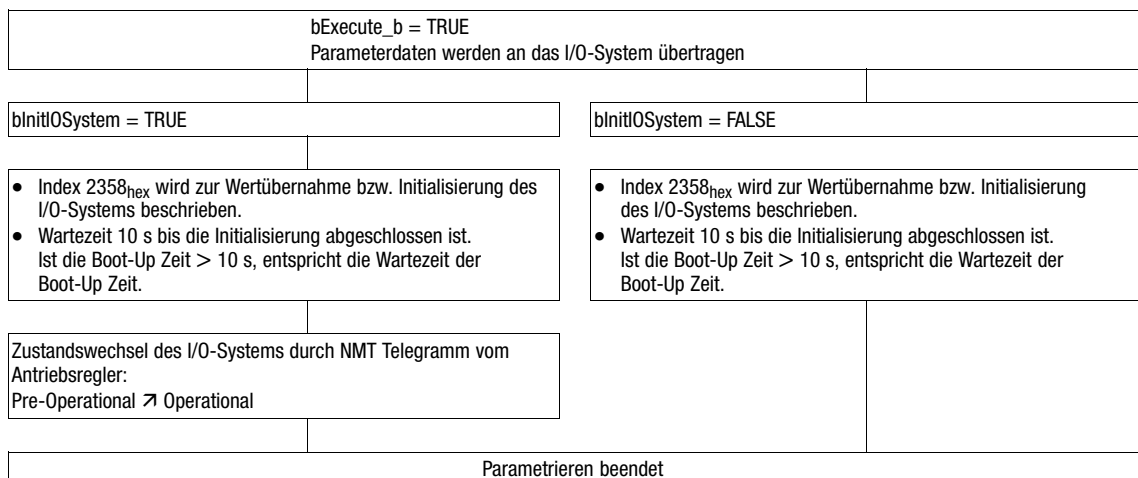
4.8 L_IOParPDO15 - Parametrierfunktion

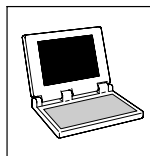
Dieser Funktionsblock dient zur Einstellung kommunikationsrelevanter Parameter zwischen PLC Steuerung und dem I/O-System für PDO 1 - 5.

- Über 3 bzw. 4 Eingangswerte wird ein Sende-/Empfangsobjekt parametrier.
- Der Ausgang *nState* zeigt den Status der Parametrierung an.
- Über den Ausgang *DataTo_IOData15* werden Informationen in einer festen Struktur an den Prozessdatenbaustein **L_IOData15** übergeben.



Interne Funktion





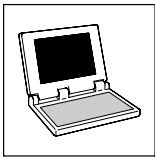
Hinweis!

Eigenschaften und Verwendung des FB **L_IOParPDO15**.

- Verwenden Sie den FB **L_IOParPDO15** als letztes in einer Schrittkette, die aus mehreren FB's zur Parametrierung besteht. Durch den FB **L_IOParPDO15** wird die Parametrierung des I/O-Systems durchgeführt.
- Zum Senden des NMT-Telegramms muss sich der Regler im CAN-Status *Operational* befinden, deshalb den Regler als Master parametrieren. Nach Ablauf der Boot-Up Zeit erfolgt der Zustandswechsel von *Pre-Operational* nach *Operational* automatisch.
- Werden mehrere IO-Systeme (Stationen) am Systembus über den FB **L_IOParPDO15** mit Standardwerten parametriert, kann eine Identifikationskollision zwischen den einzelnen PDOs dieser Stationen auftreten. Überprüfen Sie deshalb die verwendeten Identifier.

Nach dem Wechsel in den CAN-Status *Operational* wird das aktuelle Prozessabbild des I/O-Systems gesendet.

⇒ Eingänge		(Variablentyp: VAR_INPUT)			
Variablenname	Datentyp	Info/Einstellmöglichkeiten			
bExecute	Bool	Schreibauftrag ausführen			
		FALSE ↗ TRUE	Schreibauftrag wird ausgeführt		
byNodeAdr	Byte	Geräteadresse: 1 - 63			
blnInitIOSystem	Bool	I/O-System nach Parametrierung initialisieren			
		TRUE	• Wertübernahme durch Reset Node • I/O-System wird neu initialisiert, Zustandswechsel nach Operational		
		FALSE	• keine Wertübernahme • I/O-System wird nicht initialisiert		
wCobID_RxPDO1 ... wCobID_RxPDO5	Word	Identifier einstellen Default-Werte: Eingangsbeschaltung = 0 RxPDO = Empfangsidentifier vom I/O-System Identifier können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifier wie folgt zusammen. Hinweis! Die Basis-Identifier liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden.			
		RxPDOX	Identifier =	Basis-ID +	Knotenadresse
		RxPDO1	837	835	2
		RxPDO2		840	
		RxPDO3		845	
		RxPDO4		850	
		RxPDO5		855	
byTransmissionType_RxPDO1 ... byTransmissionType_RxPDO5	Byte	Übertragungsart			
		0	ereignisgesteuert		
		1	zeitgesteuert		
		2	ereignisgesteuert mit überlagerter Zykluszeit		
tCycleTime_RxPDO1 ... tCycleTime_RxPDO5	Time	Zykluszeit: t = 0 - 65535 ms			

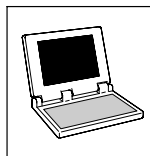


Funktionsbibliothek LenzeIOSystem.lib

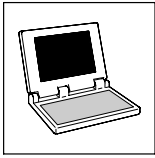
Funktionsblöcke

L_IOParPDO15 - Parametrierfunktion

⇒ Eingänge (Variablentyp: VAR_INPUT)																											
Variablenname	Datentyp	Info/Einstellmöglichkeiten																									
wCobID_TxPDO1 ... wCobID_TxPDO5	Word	Identifizier einstellen Default-Werte: Eingangsbeschaltung = 0 TxPDO = Empfangsidentifizier vom I/O-System Identifizier können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifizier wie folgt zusammen. Hinweis! Die Basis-Identifizier liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden. <table><tr><th>TxPDOX</th><th>Identifizier =</th><th>Basis-ID +</th><th>Knotenadresse</th></tr><tr><td>TxPDO1</td><td>838</td><td>836</td><td>2</td></tr><tr><td>TxPDO2</td><td></td><td>841</td><td></td></tr><tr><td>TxPDO3</td><td></td><td>846</td><td></td></tr><tr><td>TxPDO4</td><td></td><td>851</td><td></td></tr><tr><td>TxPDO5</td><td></td><td>856</td><td></td></tr></table>		TxPDOX	Identifizier =	Basis-ID +	Knotenadresse	TxPDO1	838	836	2	TxPDO2		841		TxPDO3		846		TxPDO4		851		TxPDO5		856	
TxPDOX	Identifizier =	Basis-ID +	Knotenadresse																								
TxPDO1	838	836	2																								
TxPDO2		841																									
TxPDO3		846																									
TxPDO4		851																									
TxPDO5		856																									
byTransmissionType_TxPDO1 ... byTransmissionType_TxPDO5	Byte	Übertragungsart <table><tr><td>0</td><td>ereignisgesteuert</td></tr><tr><td>1</td><td>zeitgesteuert</td></tr><tr><td>2</td><td>ereignisgesteuert mit überlagerter Zykluszeit</td></tr></table>		0	ereignisgesteuert	1	zeitgesteuert	2	ereignisgesteuert mit überlagerter Zykluszeit																		
0	ereignisgesteuert																										
1	zeitgesteuert																										
2	ereignisgesteuert mit überlagerter Zykluszeit																										
tinhibitTime_TxPDO1 ... tinhibitTime_TxPDO5	Time	Sperrzeit: t = 0 - 65535 ms																									
tCycleTime_TxPDO1 ... tCycleTime_TxPDO5	Time	Zykluszeit: t = 0 - 65535 ms																									



Ausgänge ⇒		(Variablentyp: VAR_OUTPUT)
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		-150 Can-Bus befindet sich nicht im Zustand Operational.
		-121 Falsche Treibernummer
		-120 CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.
		-119 Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.
		-118 Kein freier Can-Kanal zur Verfügung.
		-12 Message-Identifizier ist außerhalb des zulässigen Bereiches.
		-11 Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.
		0 Sende- bzw. Empfangsauftrag wurde ausgeführt.
		1 Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.
		10 Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.
200 Daten im Empfangsspeicher wurden überschrieben.		
bParDone	Bool	Parametrieren TRUE Parametrieren abgeschlossen
DataTO_IOData15	Struct	Struktur zum Prozessdatenblock L_IOData15 Globale Variable vom Typ Struct



Funktionsbibliothek LenzeIOSystem.lib

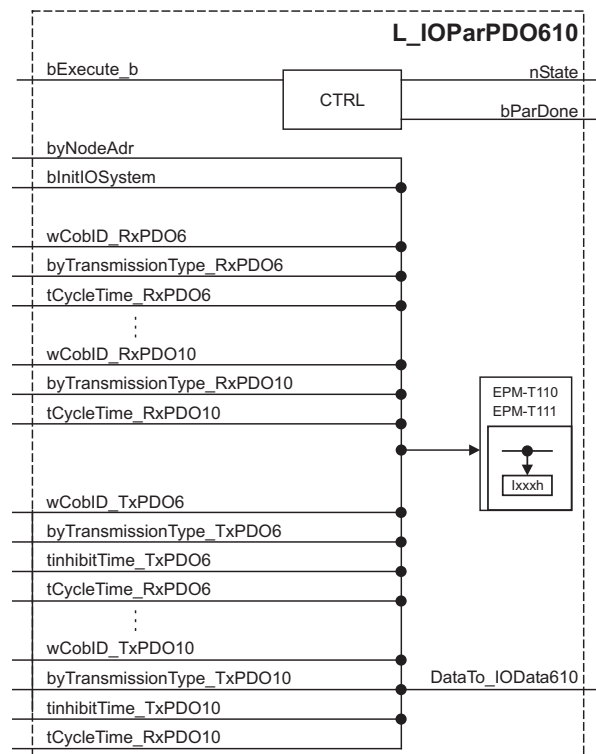
Funktionsblöcke

L_IOParPDO610 - Parametrierfunktion

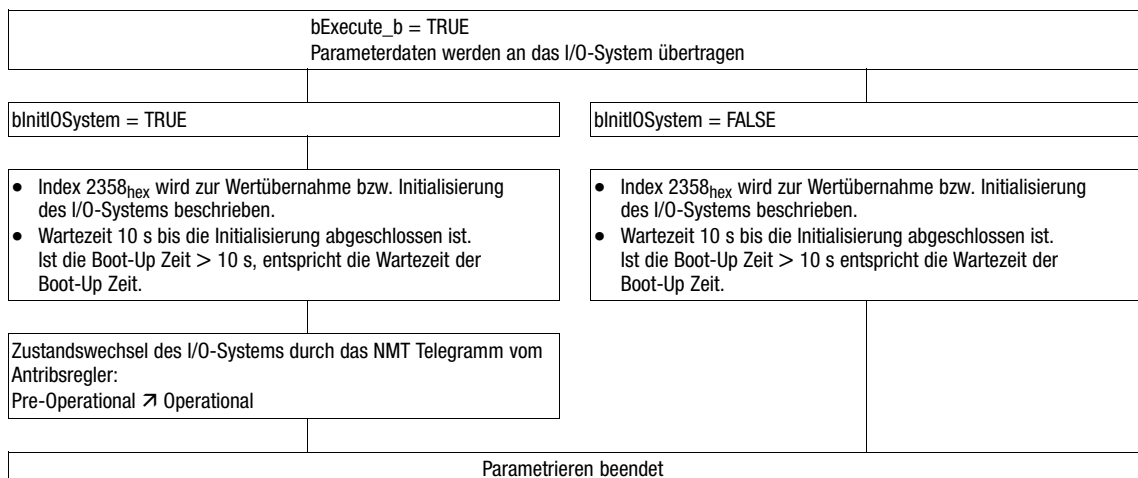
4.9 L_IOParPDO610 - Parametrierfunktion

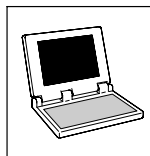
Dieser Funktionsblock dient der Einstellung der kommunikationsrelevanten Parameter zwischen PLC Steuerung und dem I/O-System für PDO 6 - 10.

- Über 3 bzw. 4 Eingangswerte wird ein Sende-/Empfangsobjekt parametriert.
- Der Ausgang *nState* zeigt den Status der Parametrierung an.
- Über den Ausgang *DataTo_IOData610* werden Informationen in einer festen Struktur an den Prozessdatenbaustein **L_IOData610** übergeben.



Interne Funktion





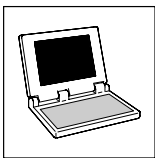
Hinweis!

Eigenschaften und Verwendung des FB **L_IOParPDO610**.

- Verwenden Sie den FB **L_IOParPDO610** als letztes in einer Schrittkette, die aus mehreren FB's zur Parametrierung besteht. Durch den FB **L_IOParPDO610** wird die Parametrierung des I/O-Systems durchgeführt.
- Parametrieren Sie den Regler als Master.
Nach Ablauf der Boot-Up Zeit erfolgt beim Master der Wechsel vom CAN-Status *Pre-Operational* nach *Operational* automatisch.
- Zum Senden des NMT-Telegramms muss sich der Regler im CAN-Status *Operational* befinden.
- Werden mehrere IO-Systeme (Stationen) am Systembus über den FB **L_IOParPDO610** mit Standardwerten parametrierung, kann eine Identifikerkollision zwischen den einzelnen PDOs dieser Stationen auftreten. Überprüfen Sie deshalb die verwendeten Identifier.

Nach dem Wechsel in den CAN-Status *Operational* wird das aktuelle Prozessabbild des I/O-Systems gesendet.

⇒ Eingänge		(Variablentyp: VAR_INPUT)			
Variablenname	Datentyp	Info/Einstellmöglichkeiten			
bExecute	Bool	Schreibauftrag ausführen			
		FALSE ➞ TRUE		Schreibauftrag wird ausgeführt	
byNodeAdr	Byte	Geräteadresse: 1 - 63			
bInitIOSystem	Bool	I/O-System nach Parametrierung initialisieren			
		TRUE		• Wertübernahme durch Reset Node	
		FALSE		• keine Wertübernahme • I/O-System wird nicht initialisiert	
wCobID_RxPDO6 ... wCobID_RxPDO10	Word	Identifizier einstellen Default-Werte: Eingangsbeschaltung = 0 RxPDO = Empfangsidentifizier vom I/O-System Identifizier können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifizier wie folgt zusammen. Hinweis! Die Basis-Identifizier liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden.			
		RxPDOX	Identifizier =	Basis-ID +	Knotenadresse
		RxPDO6	862	860	2
		RxPDO7		865	
		RxPDO8		870	
		RxPDO9		875	
		RxPDO10		880	
byTransmissionType_RxPDO6 ... byTransmissionType_RxPDO10	Byte	Übertragungsart			
		0	ereignisgesteuert		
		1	zeitgesteuert		
		2	ereignisgesteuert mit überlagerter Zykluszeit		
tCyceTime_RxPDO6 ... tCyceTime_RxPDO10	Time	Zykluszeit t = 0 - 65535 ms			

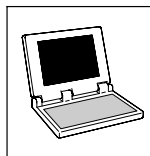


Funktionsbibliothek LenzeIOSystem.lib

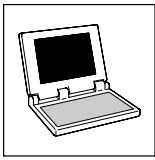
Funktionsblöcke

L_IOParPDO610 - Parametrierfunktion

⇒ Eingänge		(Variablentyp: VAR_INPUT)																			
Variablenname	Datentyp	Info/Einstellmöglichkeiten																			
wCobID_TxPDO6 ... wCobID_TxPDO10	Word	Identifizier einstellen Default-Werte: Eingangsbeschaltung = 0 TxPDO = Empfangsidentifizier vom I/O-System Identifizier können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifizier wie folgt zusammen. Hinweis! Die Basis-Identifizier liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden. <table><tr><th>TxPDOX</th><th>Identifizier =</th><th>Basis-ID +</th><th>Knotenadresse</th></tr><tr><td>TxPDO6</td><td rowspan="5">863</td><td>861</td><td rowspan="5">2</td></tr><tr><td>TxPDO7</td><td>866</td></tr><tr><td>TxPDO8</td><td>871</td></tr><tr><td>TxPDO9</td><td>876</td></tr><tr><td>TxPDO10</td><td>881</td></tr></table>				TxPDOX	Identifizier =	Basis-ID +	Knotenadresse	TxPDO6	863	861	2	TxPDO7	866	TxPDO8	871	TxPDO9	876	TxPDO10	881
TxPDOX	Identifizier =	Basis-ID +	Knotenadresse																		
TxPDO6	863	861	2																		
TxPDO7		866																			
TxPDO8		871																			
TxPDO9		876																			
TxPDO10		881																			
byTransmissionType_TxPDO6 ... byTransmissionType_TxPDO10	Byte	Übertragungsart <table><tr><td>0</td><td>ereignisgesteuert</td></tr><tr><td>1</td><td>zeitgesteuert</td></tr><tr><td>2</td><td>ereignisgesteuert mit überlagerter Zykluszeit</td></tr></table>				0	ereignisgesteuert	1	zeitgesteuert	2	ereignisgesteuert mit überlagerter Zykluszeit										
0	ereignisgesteuert																				
1	zeitgesteuert																				
2	ereignisgesteuert mit überlagerter Zykluszeit																				
tinhibitTime_TxPDO6 ... tinhibitTime_TxPDO10	Time	Sperrzeit t = 0 - 65535 ms																			
tCyceTime_TxPDO6 ... tCyceTime_TxPDO10	Time	Zykluszeit t = 0 - 65535 ms																			



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)				
Bezeichner	Datentyp	Wert/Bedeutung		
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.		
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.		
		1 Sendeauftrag abgesetzt Bearbeitung läuft		
		2001 Fehler beim Bus-Transport aufgetreten		
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten		
		2004 Zugriff verweigert Falsches Zielsystem: Index existiert nicht		
		2005 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht		
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich		
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format		
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel		
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten		
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.		
		-150 Can-Bus befindet sich nicht im Zustand Operational.		
		-121 Falsche Treibernummer		
		-120 CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.		
		-119 Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.		
		-118 Kein freier Can-Kanal zur Verfügung.		
		-12 Message-Identifizier ist außerhalb des zulässigen Bereiches.		
		-11 Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.		
		0 Sende- bzw. Empfangsauftrag wurde ausgeführt.		
		1 Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.		
		10 Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.		
200 Daten im Empfangsspeicher wurden überschrieben.				
bParDone	Bool	Parametrieren <table><tr><td>TRUE</td><td>Parametrieren abgeschlossen</td></tr></table>	TRUE	Parametrieren abgeschlossen
TRUE	Parametrieren abgeschlossen			
DataTO_IOData610	Struct	Struktur zum Prozessdatenblock L_IOData610 Globale Variable vom Typ Struct		



Funktionsbibliothek LenzeIOSystem.lib

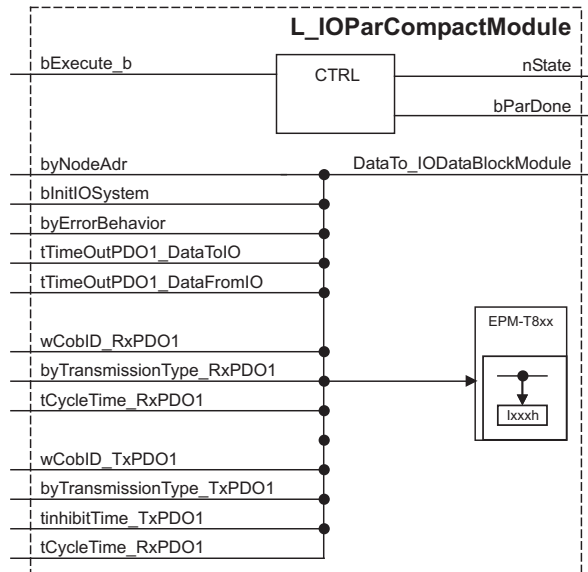
Funktionsblöcke

L_IOParCompactModule - Parametrierfunktion

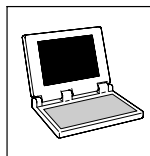
4.10 L_IOParCompactModule - Parametrierfunktion

Dieser Funktionsblock dient zur Einstellung der kommunikationsrelevanten Parameter zwischen PLC Steuerung und einem Kompaktmodul.

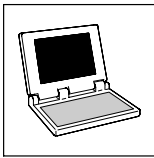
- Parametrieren einer Funktion zur zeitlichen Überwachung des Prozessdatenaustausches.
- Der Ausgang *nState* zeigt den Status der Parametrierung an.
- Über den Ausgang *DataTo_IODataBlockModule* werden Informationen in einer festen Struktur an den Prozessdatenbaustein **L_IOCompactModule** übergeben.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bExecute	Bool	Schreibauftrag ausführen FALSE ↗ TRUE Schreibauftrag wird ausgeführt
byNodeAdr	Byte	Geräteadresse: 1 - 63
bInitIOSystem	Bool	I/O-System nach Parametrierung initialisieren TRUE • Wertübernahme durch Reset Node • I/O-System wird neu initialisiert, Zustandswechsel nach OPERATIONAL FALSE • keine Wertübernahme • I/O-System wird nicht initialisiert
byErrorBehavior	Byte	Einstellen der Buszustände Fehlerverhalten während einer kritischen Gerätestörung 0 Pre-Operational 1 keine Statusänderung 2 Stopp 3 Reset nach 2 Sekunden
tTimeOutPDO1_DataToIO	Time	Überwachungszeit für Prozesseingangsdaten (lenzespezifisch). Werden innerhalb der Überwachungszeit auf dem ausgewählten PDO keine Daten empfangen, nimmt das I/O-System den unter <i>byErrorBehavior</i> eingestellten Zustand an. default t = 0 ms (Überwachung abgeschaltet) t = 0 - 65535 ms
tTimeOutPDO1_DataFromIO	Time	Überwachungszeit für Prozessausgangsdaten. Werden innerhalb der Überwachungszeit auf dem ausgewählten PDO keine Daten vom I/O-System an die PLC gesendet Wird im Funktionsblock L_IODataXXX ein Zeitüberwachungsfehler angezeigt. default t = 0 ms (Überwachung abgeschaltet) t = 0 - 65535 ms



⇒ Eingänge (Variablentyp: VAR_INPUT)												
Variablenname	Datentyp	Info/Einstellmöglichkeiten										
wCobID_RxPDO1	Word	Identifizieren einstellen Default-Werte: Eingangsbeschaltung = 0 RxPDO = Empfangsidentifizieren vom I/O-System Identifizieren können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifizieren wie folgt zusammen. Hinweis! Die Basis-Identifizieren liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden. <table><tr><td>RxPDOX</td><td>Identifizieren =</td><td>Basis-ID +</td><td>Knotenadresse</td></tr><tr><td>RxPDO1</td><td>837</td><td>835</td><td>2</td></tr></table>			RxPDOX	Identifizieren =	Basis-ID +	Knotenadresse	RxPDO1	837	835	2
RxPDOX	Identifizieren =	Basis-ID +	Knotenadresse									
RxPDO1	837	835	2									
byTransmissionType_RxPDO1	Byte	Übertragungsart <table><tr><td>0</td><td>ereignisgesteuert</td></tr><tr><td>1</td><td>zeitgesteuert</td></tr><tr><td>2</td><td>ereignisgesteuert mit überlagerter Zykluszeit</td></tr></table>			0	ereignisgesteuert	1	zeitgesteuert	2	ereignisgesteuert mit überlagerter Zykluszeit		
0	ereignisgesteuert											
1	zeitgesteuert											
2	ereignisgesteuert mit überlagerter Zykluszeit											
tCyceTime_TxPDO6 ... tCyceTime_TxPDO10	Time	Zykluszeit t = 0 - 65535 ms										
wCobID_TxPDO1	Word	Identifizieren einstellen Default-Werte: Eingangsbeschaltung = 0 TxPDO = Empfangsidentifizieren vom I/O-System Identifizieren können durch Wertvorgabe beliebig eingestellt werden. Ist die Eingangsbeschaltung = 0 setzen sich die Identifizieren wie folgt zusammen. Die Basis-Identifizieren liegen nicht im vorbelegten Bereich des Standard-Systembusses. ID-Bereich: 384-831 Dadurch werden Überschneidungen der ID mit den freien CAN-Objekten auf denen die IO-lib basiert, vermieden. <table><tr><td>TxPDOX</td><td>Identifizieren =</td><td>Basis-ID +</td><td>Knotenadresse</td></tr><tr><td>TxPDO1</td><td>838</td><td>836</td><td>2</td></tr></table>			TxPDOX	Identifizieren =	Basis-ID +	Knotenadresse	TxPDO1	838	836	2
TxPDOX	Identifizieren =	Basis-ID +	Knotenadresse									
TxPDO1	838	836	2									
byTransmissionType_TxPDO1	Byte	Übertragungsart <table><tr><td>0</td><td>ereignisgesteuert</td></tr><tr><td>1</td><td>zeitgesteuert</td></tr><tr><td>2</td><td>ereignisgesteuert mit überlagerter Zykluszeit</td></tr></table>			0	ereignisgesteuert	1	zeitgesteuert	2	ereignisgesteuert mit überlagerter Zykluszeit		
0	ereignisgesteuert											
1	zeitgesteuert											
2	ereignisgesteuert mit überlagerter Zykluszeit											
tInhibitTime_TxPDO6 ... tInhibitTime_TxPDO10	Time	Sperrzeit t = 0 - 65535 ms										
tCyceTime_TxPDO6 ... tCyceTime_TxPDO10	Time	Zykluszeit t = 0 - 65535 ms										

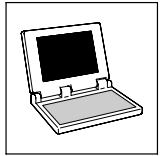


Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOParCompactModule - Parametrierfunktion

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		0 Der Sendeauftrag ist ausgeführt. Das Modul ist parametriert.
		1 Sendeauftrag abgesetzt Bearbeitung läuft
		2001 Fehler beim Bus-Transport aufgetreten
		2002 Beim Sendeauftrag ist ein Time-Out aufgetreten
		2004 Zugriff verweigert Falsches Zielsystem: Index existiert nicht
		2005 Zugriff verweigert Falsches Zielsystem: Subindex existiert nicht
		2013 Zugriff verweigert Falsches Zielsystem: ungültiger Wertebereich
		2111 Zugriff verweigert Falsches Zielsystem: mehr als 4 Nachkommastellen angegeben oder kein Index mit Fixkomma-Format
		2117 Kommunikationskanal existiert nicht Einstellung byCom-Channel
		-10xx Wert am Eingang des Funktionsblocks ist fehlerhaft. XX := Nummer des fehlerhaften Eingangs (Zählrichtung von oben), keine Parametrierung möglich. Abhilfe: Eingangswert ändern, Datenübertragung erneut starten
nState	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. Fehler werden durch Flankenwechsel am Eingang <i>bExecute</i> oder bei einem Programm-Neustart zurückgesetzt.
		-150 Can-Bus befindet sich nicht im Zustand Operational.
		-121 Falsche Treibernummer
		-120 CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.
		-119 Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.
		-118 Kein freier Can-Kanal zur Verfügung.
		-12 Message-Identifizier ist außerhalb des zulässigen Bereiches.
		-11 Zeiger <i>piOAddress</i> zeigt nicht auf SPS-RAM.
		0 Sende- bzw. Empfangsauftrag wurde ausgeführt.
		1 Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.
		10 Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.
		200 Daten im Empfangsspeicher wurden überschrieben.
bParDone	Byte	Status Parametrierung
		TRUE Parametrierung ausgeführt
DataTO_IODataCompactmodule	Struct	Struktur zum Prozessdatenblock L_IOCompactModule globale Variable vom Typ Struct



5 Funktionsblöcke zur Prozessdatenverarbeitung

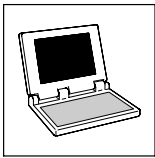
In diesem Kapitel erhalten Sie Informationen über Funktionsblöcke zum Koordinieren, Unterstützen und Konvertieren.

Folgende Funktionen und Module werden koordiniert.

- Digitale und analoge Ein- und Ausgangsdaten

Folgende Funktionen und Module werden unterstützt.

- Analoge Eingangsmodule
- Analoge Ausgangsmodule
- Zählermodule
- Verbindung zwischen PLC-Steuerung und I/O-System
- Verbindung zwischen PLC-Steuerung und Kompakt Modul



Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

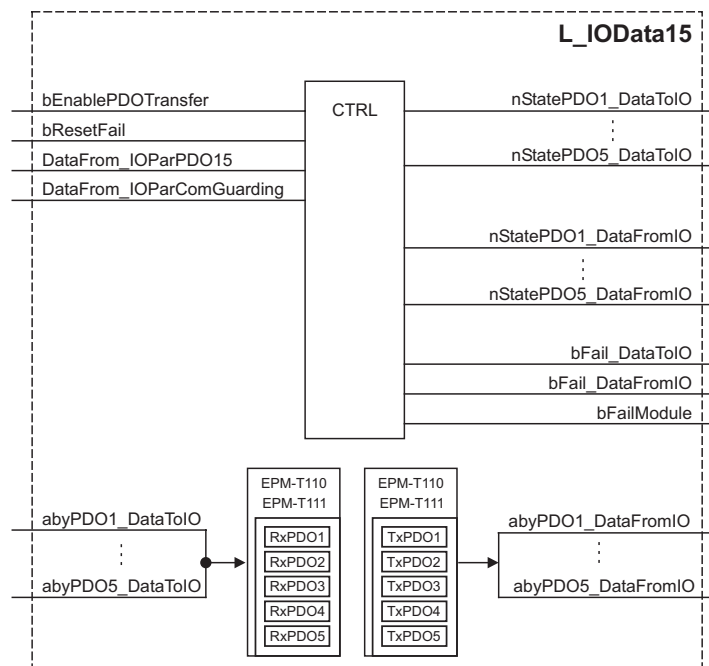
L_IOData15 - Daten koordinieren

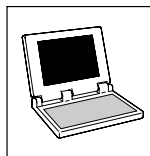
5.1 L_IOData15 - Daten koordinieren

Der Funktionsblock **L_IOData15** koordiniert die digitalen und analogen Ein- bzw. Ausgangsdaten.

Der Baustein ist die Schnittstelle zwischen PLC und I/O-System.

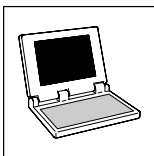
- Daten von der Steuerung werden über die Eingänge *abyPDO_DataToIO* an das I/O-System (digitale/analogue Ausgangskanäle) übermittelt.
- Die Zustände der digitalen und analogen Eingangskanäle werden über die Ausgänge *abyPDO_DataFromIO* an die Steuerung übergeben.
- Der Funktionsblock setzt sich aus den freien CAN-Objekten zusammen. Deshalb ist die Statusanzeige an den Ausgängen *nState...* äquivalent zu den Sende- und Empfangsblöcken der CAN Library.





⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bEnablePDOTransfer	Bool	Datentransfer TRUE PDO Datentransfer aktiv
abyPDO1_DataToIO ... abyPDO5_DataToIO	Array of Byte	Empfangsobjekte des I/O-Systems • Übergabe der Prozessdaten von der PLC an das I/O-System. (digitale/analoge Ausgänge)
bResetFail	Bool	Reset der Ausgangsdaten <i>bFail_DataToIO</i> , <i>bFail_DataFromIO</i> und <i>bFail_Module</i> TRUE Reset der Ausgangsdaten ausgeführt
DataFrom_IOParPDO15	Struct	Strukturdaten vom Parameterblock L_IOParPDO15 werden über diesen Eingang an den Prozessdatenblock L_IOData15 übergeben. Globale Variable vom Typ Struct
DataFrom_IOParComGuarding	Struct	Übergabe der Überwachungszeiten <i>tTimeout_DataFromIO</i> für PDO 1-5. Strukturdaten vom Parameterblock L_IOParComGuarding . Globale Variable vom Typ Struct

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																												
Bezeichner	Datentyp	Wert/Bedeutung																										
nStatePDO1_DataFromIO ... nStatePDO5_DataFromIO	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. <table><tr><td>-150</td><td>Can-Bus befindet sich nicht im Zustand Operational.</td></tr><tr><td>-121</td><td>Falsche Treibernummer</td></tr><tr><td>-120</td><td>CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.</td></tr><tr><td>-119</td><td>Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.</td></tr><tr><td>-118</td><td>Kein freier Can-Kanal zur Verfügung.</td></tr><tr><td>-12</td><td>Message-Identifizier ist außerhalb des zulässigen Bereiches.</td></tr><tr><td>-11</td><td>Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.</td></tr><tr><td>0</td><td>Sende- bzw. Empfangsauftrag wurde ausgeführt.</td></tr><tr><td>1</td><td>Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.</td></tr><tr><td>2</td><td>Datenübertragung ist inaktiv • bEnablePDOTransfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen </td></tr><tr><td>10</td><td>Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.</td></tr><tr><td>200</td><td>Daten im Empfangsspeicher wurden überschrieben.</td></tr><tr><td>-30xx</td><td>TimeOut Überwachung (xx ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.</td></tr></table>	-150	Can-Bus befindet sich nicht im Zustand Operational.	-121	Falsche Treibernummer	-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.	-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.	-118	Kein freier Can-Kanal zur Verfügung.	-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.	-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.	0	Sende- bzw. Empfangsauftrag wurde ausgeführt.	1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.	2	Datenübertragung ist inaktiv • bEnablePDOTransfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen	10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.	200	Daten im Empfangsspeicher wurden überschrieben.	-30xx	TimeOut Überwachung (xx ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.
-150	Can-Bus befindet sich nicht im Zustand Operational.																											
-121	Falsche Treibernummer																											
-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.																											
-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.																											
-118	Kein freier Can-Kanal zur Verfügung.																											
-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.																											
-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.																											
0	Sende- bzw. Empfangsauftrag wurde ausgeführt.																											
1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.																											
2	Datenübertragung ist inaktiv • bEnablePDOTransfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen																											
10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.																											
200	Daten im Empfangsspeicher wurden überschrieben.																											
-30xx	TimeOut Überwachung (xx ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.																											

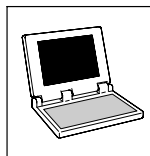


Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOData15 - Daten koordinieren

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																										
Bezeichner	Datentyp	Wert/Bedeutung																								
nStatePD01_DataToIO nStatePD05_DataToIO	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. <table><tr><td>-150</td><td>Can-Bus befindet sich nicht im Zustand Operational.</td></tr><tr><td>-121</td><td>Falsche Treibernummer</td></tr><tr><td>-120</td><td>CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.</td></tr><tr><td>-119</td><td>Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.</td></tr><tr><td>-118</td><td>Kein freier Can-Kanal zur Verfügung.</td></tr><tr><td>-12</td><td>Message-Identifizier ist außerhalb des zulässigen Bereiches.</td></tr><tr><td>-11</td><td>Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.</td></tr><tr><td>0</td><td>Sende- bzw. Empfangsauftrag wurde ausgeführt.</td></tr><tr><td>1</td><td>Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.</td></tr><tr><td>2</td><td>Datenübertragung ist inaktiv • bEnablePD0Transfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen </td></tr><tr><td>10</td><td>Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.</td></tr><tr><td>200</td><td>Daten im Empfangsspeicher wurden überschrieben.</td></tr></table>	-150	Can-Bus befindet sich nicht im Zustand Operational.	-121	Falsche Treibernummer	-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.	-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.	-118	Kein freier Can-Kanal zur Verfügung.	-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.	-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.	0	Sende- bzw. Empfangsauftrag wurde ausgeführt.	1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.	2	Datenübertragung ist inaktiv • bEnablePD0Transfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen	10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.	200	Daten im Empfangsspeicher wurden überschrieben.
-150	Can-Bus befindet sich nicht im Zustand Operational.																									
-121	Falsche Treibernummer																									
-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.																									
-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.																									
-118	Kein freier Can-Kanal zur Verfügung.																									
-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.																									
-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.																									
0	Sende- bzw. Empfangsauftrag wurde ausgeführt.																									
1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.																									
2	Datenübertragung ist inaktiv • bEnablePD0Transfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen																									
10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.																									
200	Daten im Empfangsspeicher wurden überschrieben.																									
bFail_DataToIO	Bool	Sendeauftrag zum I/O-System <table><tr><td>TRUE</td><td>Sendeauftrag zum I/O-System ist nicht korrekt</td></tr></table>	TRUE	Sendeauftrag zum I/O-System ist nicht korrekt																						
TRUE	Sendeauftrag zum I/O-System ist nicht korrekt																									
bFail_DataFromIO	Bool	Empfangsauftrag vom I/O-System <table><tr><td>TRUE</td><td>Empfangsauftrag vom I/O-System ist nicht korrekt</td></tr></table>	TRUE	Empfangsauftrag vom I/O-System ist nicht korrekt																						
TRUE	Empfangsauftrag vom I/O-System ist nicht korrekt																									
bFail_Module	Bool	Modulfehler <table><tr><td>TRUE</td><td> • Moduldiagnose aktiv • Emergency Objekt zeigt Fehler an </td></tr></table>	TRUE	• Moduldiagnose aktiv • Emergency Objekt zeigt Fehler an																						
TRUE	• Moduldiagnose aktiv • Emergency Objekt zeigt Fehler an																									
abyPD01_DataFromIO ... abyPD05_DataFromIO	Array of Byte	Sendeobjekte des I/O-Systems Übergabe der Prozessdaten vom I/O-System an die PLC (digitale/analoge Eingänge)																								

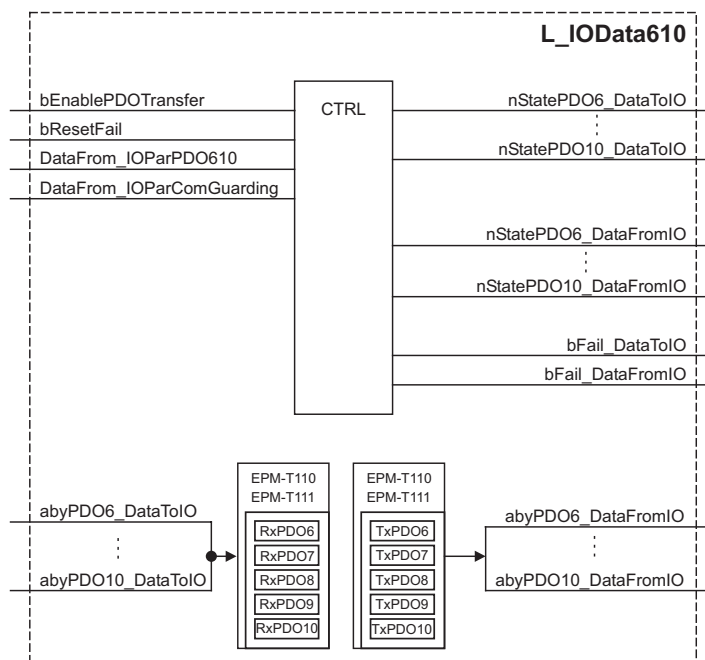


5.2 L_IOData610 - Daten koordinieren

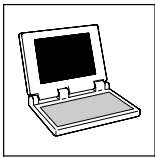
Der Funktionsblock **L_IOData610** koordiniert die digitalen und analogen Ein- bzw. Ausgangsdaten für die Prozessdatenobjekte 6 - 10.

Der Baustein ist die Schnittstelle zwischen PLC und I/O-System.

- Daten von der Steuerung werden über die Eingänge *abyPDO_DataToIO* an das I/O-System (digitale/analogue Ausgangskanäle) übermittelt.
- Die Zustände der digitalen und analogen Eingangskanäle werden über die Ausgänge *abyPDO_DataFromIO* an die Steuerung übergeben.
- Der Funktionsblock setzt sich aus den freien CAN-Objekten zusammen. Daher ist die Statusanzeige an den Ausgängen *nState...* genauso wie bei den Send- und Empfangsblöcken der CAN Library.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bEnablePDOTransfer	Bool	Datentransfer TRUE PDO Datentransfer aktiv
abyPDO6_DataToIO ... abyPDO10_DataToIO	Array of Byte	Empfangsobjekte des I/O-Systems • Übergabe der Prozessdaten von der PLC an das I/O-System. (digitale/analogue Ausgänge)
bResetFail	Bool	Reset der Ausgangsdaten <i>bFail_DataToIO</i> und <i>bFail_DataFromIO</i> TRUE Reset der Ausgangsdaten ausgeführt
DataFrom_IOParPDO610	Struct	Strukturdaten vom Parameterblock L_IOParPDO610 werden über diesen Eingang an den Prozessdatenblock L_IOData610 übergeben. Globale Variable vom Typ Struct
DataFrom_IOParComGuarding	Struct	Übergabe der Überwachungszeiten <i>tTimeout_DataFromIO</i> für PDO 6-10. Strukturdaten vom Parameterblock L_IOParComGuarding . Globale Variable vom Typ Struct

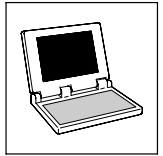


Funktionsbibliothek LenzeIOSystem.lib

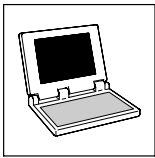
Funktionsblöcke

L_IOData610 - Daten koordinieren

Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																										
Bezeichner	Datentyp	Wert/Bedeutung																								
nStatePDO6_DataToIO ... nStatePDO10_DataToIO	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. <table><tr><td>-150</td><td>Can-Bus befindet sich nicht im Zustand Operational.</td></tr><tr><td>-121</td><td>Falsche Treibernummer</td></tr><tr><td>-120</td><td>CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.</td></tr><tr><td>-119</td><td>Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.</td></tr><tr><td>-118</td><td>Kein freier Can-Kanal zur Verfügung.</td></tr><tr><td>-12</td><td>Message-Identifizier ist außerhalb des zulässigen Bereiches.</td></tr><tr><td>-11</td><td>Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.</td></tr><tr><td>0</td><td>Sende- bzw. Empfangsauftrag wurde ausgeführt.</td></tr><tr><td>1</td><td>Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.</td></tr><tr><td>2</td><td>Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen </td></tr><tr><td>10</td><td>Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.</td></tr><tr><td>200</td><td>Daten im Empfangsspeicher wurden überschrieben.</td></tr></table>	-150	Can-Bus befindet sich nicht im Zustand Operational.	-121	Falsche Treibernummer	-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.	-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.	-118	Kein freier Can-Kanal zur Verfügung.	-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.	-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.	0	Sende- bzw. Empfangsauftrag wurde ausgeführt.	1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.	2	Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen	10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.	200	Daten im Empfangsspeicher wurden überschrieben.
-150	Can-Bus befindet sich nicht im Zustand Operational.																									
-121	Falsche Treibernummer																									
-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.																									
-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.																									
-118	Kein freier Can-Kanal zur Verfügung.																									
-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.																									
-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.																									
0	Sende- bzw. Empfangsauftrag wurde ausgeführt.																									
1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.																									
2	Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen																									
10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.																									
200	Daten im Empfangsspeicher wurden überschrieben.																									



Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)																												
Bezeichner	Datentyp	Wert/Bedeutung																										
nStatePDO6_DataFromIO ... nStatePDO10_DataFromIO	Integer	Statuszustände Tritt während der Parametrierung ein Übertragungsfehler auf oder sind die Eingangsdaten fehlerhaft, wird die Parametrierung abgebrochen und eine Fehlernummer wird am Ausgang <i>nState</i> angezeigt. <table><tr><td>-150</td><td>Can-Bus befindet sich nicht im Zustand Operational.</td></tr><tr><td>-121</td><td>Falsche Treibernummer</td></tr><tr><td>-120</td><td>CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.</td></tr><tr><td>-119</td><td>Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.</td></tr><tr><td>-118</td><td>Kein freier Can-Kanal zur Verfügung.</td></tr><tr><td>-12</td><td>Message-Identifizier ist außerhalb des zulässigen Bereiches.</td></tr><tr><td>-11</td><td>Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.</td></tr><tr><td>0</td><td>Sende- bzw. Empfangsauftrag wurde ausgeführt.</td></tr><tr><td>1</td><td>Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.</td></tr><tr><td>2</td><td>Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen </td></tr><tr><td>10</td><td>Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.</td></tr><tr><td>200</td><td>Daten im Empfangsspeicher wurden überschrieben.</td></tr><tr><td>-30xx</td><td>TimeOut Überwachung (<i>xx</i> ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.</td></tr></table>	-150	Can-Bus befindet sich nicht im Zustand Operational.	-121	Falsche Treibernummer	-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.	-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.	-118	Kein freier Can-Kanal zur Verfügung.	-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.	-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.	0	Sende- bzw. Empfangsauftrag wurde ausgeführt.	1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.	2	Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen	10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.	200	Daten im Empfangsspeicher wurden überschrieben.	-30xx	TimeOut Überwachung (<i>xx</i> ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.
-150	Can-Bus befindet sich nicht im Zustand Operational.																											
-121	Falsche Treibernummer																											
-120	CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.																											
-119	Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.																											
-118	Kein freier Can-Kanal zur Verfügung.																											
-12	Message-Identifizier ist außerhalb des zulässigen Bereiches.																											
-11	Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.																											
0	Sende- bzw. Empfangsauftrag wurde ausgeführt.																											
1	Auftrag ist noch nicht abgeschlossen und steht noch im Auftragspeicher.																											
2	Datenübertragung ist inaktiv • <i>bEnablePDOTransfer</i> = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen																											
10	Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.																											
200	Daten im Empfangsspeicher wurden überschrieben.																											
-30xx	TimeOut Überwachung (<i>xx</i> ≙ PDOxx) Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.																											
bFail_DataToIO	Bool	Sendeauftrag zum I/O-System <table><tr><td>TRUE</td><td>Sendeauftrag zum I/O-System ist nicht korrekt</td></tr></table>	TRUE	Sendeauftrag zum I/O-System ist nicht korrekt																								
TRUE	Sendeauftrag zum I/O-System ist nicht korrekt																											
bFail_DataFromIO	Bool	Empfangsauftrag vom I/O-System <table><tr><td>TRUE</td><td>Empfangsauftrag vom I/O-System ist nicht korrekt</td></tr></table>	TRUE	Empfangsauftrag vom I/O-System ist nicht korrekt																								
TRUE	Empfangsauftrag vom I/O-System ist nicht korrekt																											
abyPDO6_DataFromIO ... abyPDO10_DataFromIO	Array of Byte	Sendeobjekte des I/O-Systems Übergabe der Prozessdaten vom I/O-System an die PLC (digitale/analoge Eingänge)																										



Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOCompactModule - Daten koordinieren

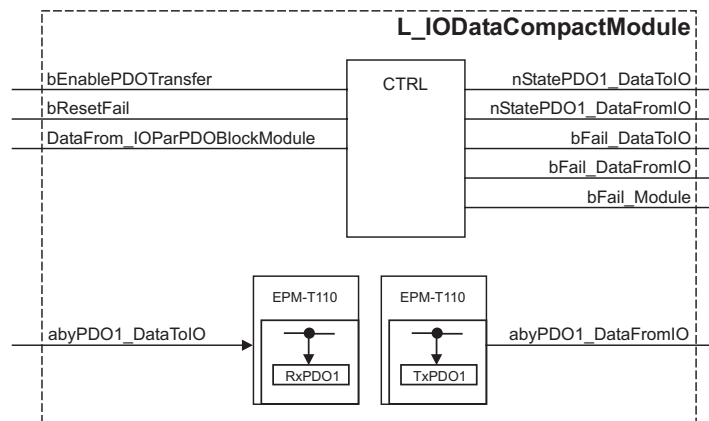
5.3

L_IOCompactModule - Daten koordinieren

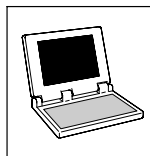
Der Funktionsblock **L_IOCompactModule** stellt die digitalen Ein- bzw. Ausgangsdaten des I/O Systems zur Verfügung. Dieser Funktionsblock wird für die Kompakt Module EPM-T83X verwendet.

Der Baustein ist die Schnittstelle zwischen PLC und I/O-System.

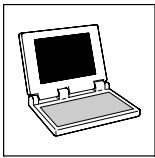
- Daten von der Steuerung werden über die Eingänge *abyPDO_DataToIO* an das I/O-System (digitale/analoge Ausgangskanäle) übermittelt.
- Die Zustände der digitalen und analogen Eingangskanäle werden über die Ausgänge *abyPDO_DataFromIO* an die Steuerung übergeben.
- Der Funktionsblock setzt sich aus den freien CAN-Objekten zusammen. Daher ist die Statusanzeige an den Ausgängen *nState...* genauso wie bei den Send- und Empfangsblöcken der CAN Library.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bEnablePDOTransfer	Bool	Datentransfer TRUE PDO Datentransfer aktiv
abyPDO1_DataToIO	Array of Byte	Empfangsobjekt des I/O-Systems • Übergabe der Prozessdaten von der PLC an das I/O-System. (digitale Ausgänge)
bResetFail	Bool	Reset der Ausgangsdaten <i>bFail_DataToIO</i> , <i>bFail_DataFromIO</i> und <i>bFail_Module</i> TRUE Reset der Ausgangsdaten ausgeführt
DataFrom_IOParPDOblockModule	Struct	Strukturdaten vom Parameterblock L_IOParPDOblockModule . Globale Variable vom Typ Struct.



Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nStatePD01_DataToIO	Integer	Statuszustände Zeigt den aktuellen Status der Sendeobjekte an.
		-150 Can-Bus befindet sich nicht im Zustand Operational.
		-121 Falsche Treibernummer
		-120 CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.
		-119 Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.
		-118 Kein freier Can-Kanal zur Verfügung.
		-12 Message-Identifizier ist außerhalb des zulässigen Bereiches.
		-11 Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.
		0 Sende- bzw. Empfangsauftrag wurde ausgeführt.
		1 Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.
		2 Datenübertragung ist inaktiv • bEnablePD0Transfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen
		10 Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.
		200 Daten im Empfangsspeicher wurden überschrieben.
nStatePD01_DataFromIO	Integer	Statuszustände Zeigt den aktuellen Status der Empfangsobjekte an.
		-150 Can-Bus befindet sich nicht im Zustand Operational.
		-121 Falsche Treibernummer
		-120 CAN-Treiber für freie CAN-Objekte ist momentan nicht initialisiert.
		-119 Der Sendeauftragsspeicher ist voll. Der Sendeauftrag konnte nicht mehr eingetragen werden.
		-118 Kein freier Can-Kanal zur Verfügung.
		-12 Message-Identifizier ist außerhalb des zulässigen Bereiches.
		-11 Zeiger <i>pIOAddress</i> zeigt nicht auf SPS-RAM.
		0 Sende- bzw. Empfangsauftrag wurde ausgeführt.
		1 Auftrag ist noch nicht abgeschlossen und steht noch im Auftragsspeicher.
		2 Datenübertragung ist inaktiv • bEnablePD0Transfer = False • Parametrierung des I/O-Systems ist noch nicht abgeschlossen
		10 Telegrammlänge ist größer als 8 Byte. Telegrammlänge wird auf 8 Byte begrenzt.
		200 Daten im Empfangsspeicher wurden überschrieben.
		-30xx TimeOut Überwachung Die Datenübertragung vom I/O-System zur PLC ist ausgefallen. Die Daten sind verspätet oder nicht eingetroffen.

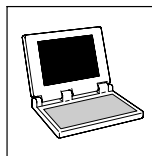


Funktionsbibliothek LenzelOSystem.lib

Funktionsblöcke

L_IOCompactModule - Daten koordinieren

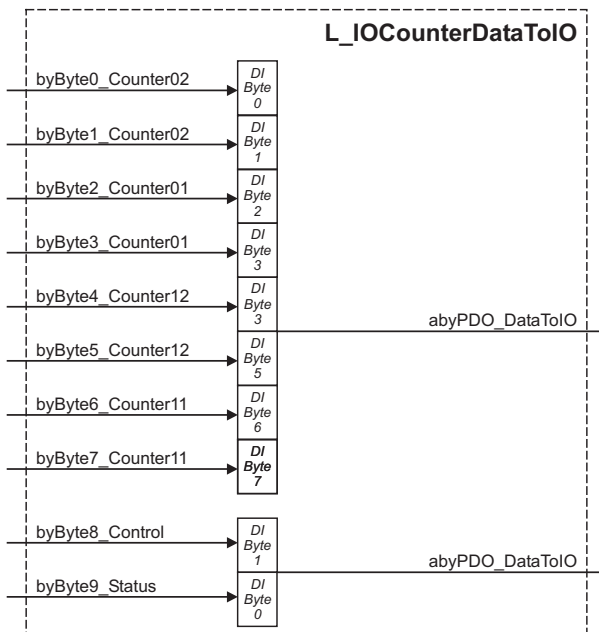
Ausgänge ⇨ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
bFail_DataToIO	Bool	Sendeauftrag zum I/O-System
		TRUE Sendeauftrag zum I/O-System ist nicht korrekt
bFail_DataFromIO	Bool	Empfangsauftrag vom I/O-System
		TRUE Empfangsauftrag vom I/O-System ist nicht korrekt
bFail_Module	Bool	Modulfehler
		TRUE • Moduldiagnose aktiv • Emergency Objekt zeigt Fehler an
abyPDO1_DataFromIO	Array of Byte	Sendeobjekt des I/O-Systems Übergabe der Prozessdaten vom I/O-System an die PLC (digitale Eingänge)



5.4 L_IOCounterDataToIO - Zählerfunktion

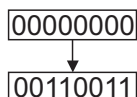
Die Funktion des Zählermoduls wird über den Parametrierdatenbaustein

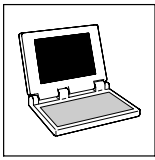
L_IOParCounterDataToIO eingestellt. Abhängig von dieser Parametrierung kann der Zähler mit Zähl- oder Vergleichswerten vorbelegt werden. Für diese Vorbelegung wird der Funktionsblock **L_IOCounterDataToIO** benutzt.



Die Belegung der Bytes 0...7 ist Abhängig vom ausgewählten Modus. Der Default-Wert im Zählermodus ist null. Daraus ergibt sich folgende Belegung.

- Für die Vorbelegung des Zählers 0 mit Zählwerten dienen die Eingänge
 - byByte0_Counter02 LowByte Bit16...23
 - byByte1_Counter02 HigByte Bit24...32
 - byByte2_Counter01 LowByte Bit0...7
 - byByte3_Counter01 HighByte Bit8...15
- Das Einstellen entsprechender Werte für den Zähler 1 erfolgt über die Eingänge
 - byByte4_Counter12 LowByte Bit16...23
 - byByte5_Counter12 HigByte Bit24...32
 - byByte6_Counter11 LowByte Bit0...7
 - byByte7_Counter11 HighByte Bit8...15
- Nach einem Flankenwechsel am Eingang byByte8_Control von Low nach High werden die Wertevorgaben an das Zählermodul übertragen. Vorgabe im Control Byte





Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IOCounterDataToIO - Zählerfunktion

byByte9_Status

Mit diesem Eingang kann der letzte Zählwert im Modul gespeichert werden oder der Zählwert soll nach Wiedereinschalten mit 0 beginnen.

- 1 = Zählwert wird gespeichert
- 2 = Zählwert wird bei Wiederanlauf gelöscht (default)
- Die beschriebenen Daten werden durch folgende Ausgänge zur Verfügung gestellt.
 - abyPDO_Data1ToIO
 - abyPDO_Data2ToIO

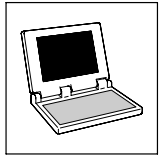


Hinweis!

Vom Zählermodul werden zwei Rx und zwei Tx PDO's belegt.

⇒ Eingänge (Variablentyp: VAR_INPUT)						
Variablenname	Datentyp	Info/Einstellmöglichkeiten				
byByte0_Counter02 ... byByte7_Counter11	Byte	Vorgabe von Start oder Vergleichswerten an das Zählermodul. Beispiel: Das Modul wurde als 2 mal 32 Bit Zähler parametrisiert. Soll der Zähler mit dem Zählwert 77 beginnen muss dieser Wert am Eingang <i>byByte2_Counter01</i> zur Verfügung stehen.				
byByte8_Control	Byte	Übernahme der Wertevorgaben an folgenden Eingängen nach Flankenwechsel von Low nach High. byByte0_Counter02 byByte7_Counter11 ( 6-10)				
byByte9_Status	Byte	Zählwert nach Wiedereinschalten <table><tr><td>1</td><td>Letzter Zählwert ist gespeichert</td></tr><tr><td>2</td><td>Wiederanlauf mit Zählwert 0 (default)</td></tr></table>	1	Letzter Zählwert ist gespeichert	2	Wiederanlauf mit Zählwert 0 (default)
1	Letzter Zählwert ist gespeichert					
2	Wiederanlauf mit Zählwert 0 (default)					

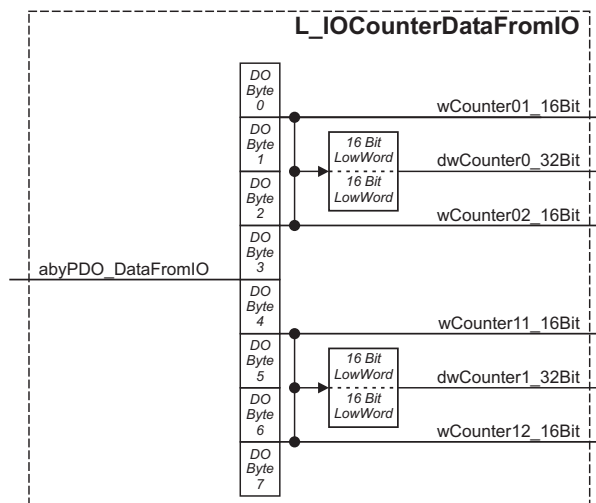
Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
abyPDOData1ToIO	Array of Byte	Schnittstelle zu den Eingängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe der Information an das Zählermodul.
abyPDOData2ToIO	Array of Byte	Schnittstelle zu den Eingängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe der Information an das Zählermodul.



5.5 L_IOCounterDataFromIO - Zählerfunktion

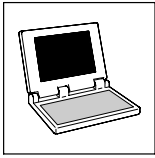
Über den Funktionsblock **L_IOCounterDataFromIO** werden die Zählinformationen des Zählermoduls zur Verfügung gestellt.

- Zählerinformation von 4 Zählern mit einer Datenbreite von 16 Bit werden von folgenden Ausgängen zur Verfügung gestellt.
 - wCounter01_16Bit
 - wCounter02_16Bit
 - wCounter11_16Bit
 - wCounter12_16Bit
- Zählerinformation von 2 Zählern mit einer Datenbreite von 32 Bit werden von folgenden Ausgängen zur Verfügung gestellt.
 - dwCounter0_32Bit
 - dwCounter1_32Bit



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
abyPDO_DataFromIO	Array of Byte	Schnittstelle zu den Ausgängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe von Informationen vom Zählermodul.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
wCounter01_16Bit wCounter02_16Bit wCounter11_16Bit wCounter12_16Bit	Word	Zählerdaten 16 Bit Breite
dwCounter0_32Bit dwCounter1_32Bit	Double Word	Zählerdaten 32 Bit Breite



Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

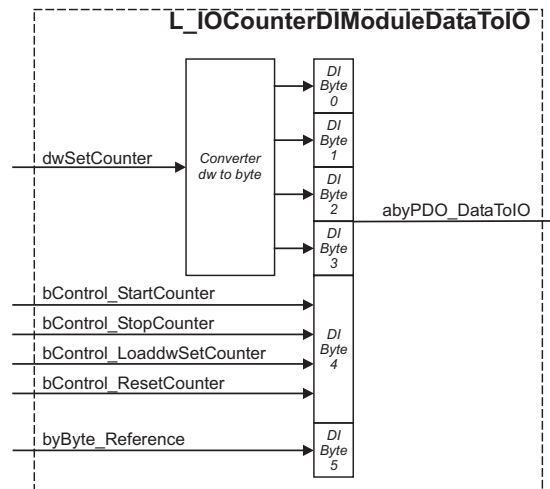
L_IOCounterDIModuleDataToIO - Zählerfunktion

5.6

L_IOCounterDIModuleDataToIO - Zählerfunktion

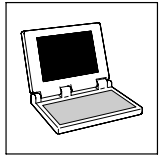
Die Zählerfunktion des Moduls EPM-T430 wird über den Prozessdatenbaustein **L_IOCounterDIModuleDataToIO** eingestellt. Abhängig von den Funktionseinstellungen der Eingänge kann der Zähler mit Zählwerten vorbelegt werden.

Der Funktionsblock **L_IOCounterDIModuleDataToIO** wird außerdem für Start, Stop, Set, Reset u.a. benutzt.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
dwSetCounter	Double Word	Wertvorgabe für das Zählmodul
bControl_StartCounter	Bool	Start Zähler
bControl_StopCounter	Bool	Stop Zähler
bControl_LoaddwSetCounter	Bool	Wertvorgabe wird vom Zählwert übernommen
bControl_ResetCounter	Bool	Zähler wird zurückgesetzt
byByte_Reference	Byte	Referenzfrequenz
		0 16 MHz
		1 8 MHz
		2 4 MHz
		3 1 MHz
		4 100 kHz
		5 10 kHz
		6 1000 Hz
		7 100 Hz

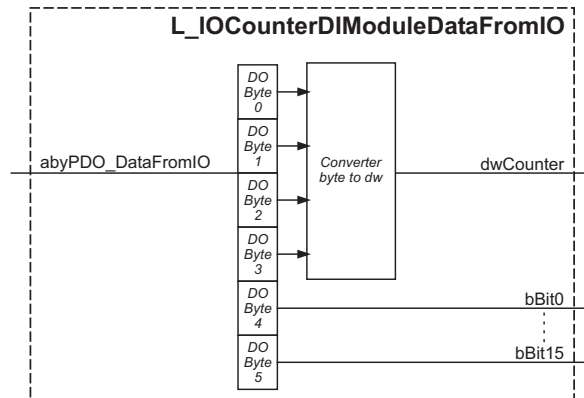
Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
abyPDOData1ToIO	Array of Byte	Schnittstelle zu den Eingängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe der Information an das Zählermodul.



5.7 L_IOCounterDIModuleDataFromIO - Zählerfunktion

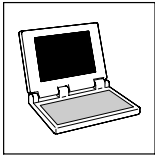
Über den Funktionsblock **L_IOCounterDIModuleDataFromIO** werden die Zählinformationen des Zählermoduls zur Verfügung gestellt.

Der 32 Bit Wert des Zählers wird am Ausgang *dwCounter* zur Verfügung gestellt.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
abyPDO_DataFromIO	Array of Byte	Schnittstelle zu den Ausgängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe von Zählwerten vom Zählermodul.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
dwCounter	Double Word	Ausgabe Zählwert
bBit0 - bBit15	Bool	Ausgabe Status digitale Eingänge 0...15



Funktionsbibliothek LenzeIOSystem.lib

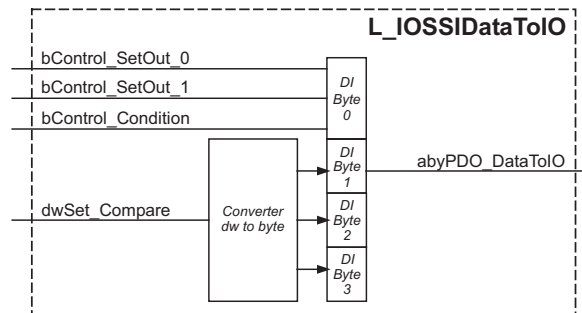
Funktionsblöcke

L_IOSSIDataToIO - Zählerfunktion

5.8 L_IOSSIDataToIO - Zählerfunktion

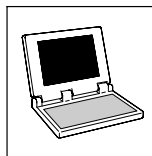
Über den Funktionsblock **L_IOSSIDataToIO** können die zwei Ausgänge des Moduls in Abhängigkeit von einem Vergleichswert angesteuert werden.

Den Ausgängen können unterschiedliche Vergleichsfunktionen bzw. Vergleichswerte zugeordnet werden, die nacheinander an das SSI Modul übertragen werden müssen.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bControl_Set_0	Bool	Ausgang 0 Überwachung Vergleichswert
		0 Aus
		1 Ein
bControl_Set_1	Bool	Ausgang 1 Überwachung Vergleichswert
		0 Aus
		1 Ein
bControl_Condition	Bool	Bedingung der Ausgänge setzen
		0 Ist-Wert > Vergleichswert
		1 Ist-Wert < Vergleichswert
dwSet_Compare	Double Word	Eingabe Vergleichswert

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
abyPDO_DataToIO	Array of Byte	Schnittstelle zu den Eingängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe von Informationen vom Zählermodul.

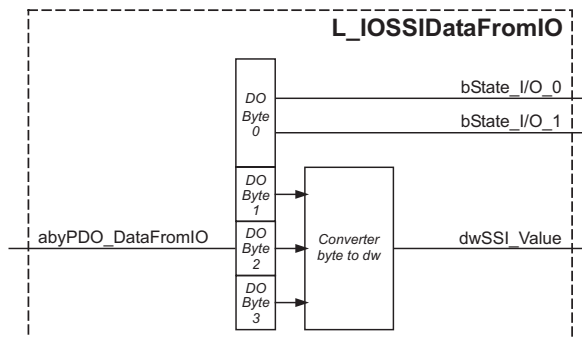


5.9 L_IOSSIDataFromIO - Zählerfunktion

Durch den Funktionsblock **L_IOSSIDataFromIO** wird der Zählwert des SSI Moduls am Ausgang *dwSSI_Value* zur Verfügung gestellt.

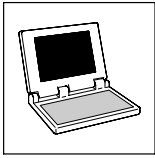
- Ausgang *bStatusIO_0* → Status Kanal E/A 0
- Ausgang *bStatusIO_1* → Status Kanal E/A 1

Die Konfiguration der Ausgänge erfolgt über den Funktionsblock **L_IOSSIDataToIO**.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
abyPDO_DataToIO	Array of Byte	Schnittstelle zu den Ausgängen der Funktionsblöcke L_IOPD015 oder L_IOPD0610 für die Übergabe der Zählwerte vom SSI Modul.

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
bState_I/O_0	Bool	Status Kanal E/A 0
bState_I/O_1	Bool	Status Kanal E/A 1
dwSSI_Value	Double Word	Zählerwert SSI Modul



Funktionsbibliothek LenzelOSsystem.lib

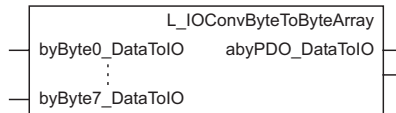
Funktionsblöcke

L_IOConvByteToByteArray - Konvertierungsfunktion

5.10 L_IOConvByteToByteArray - Konvertierungsfunktion

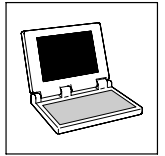
Der Funktionsblock **L_IOConvByteToByteArray** konvertiert 8 Byte Ausgangsdaten in ein Array of Byte mit 8 Feldern.

Damit entspricht die Ausgangsdatenbreite genau einem Sendeobjekt *PDO_DataToIO* zum I/O-System.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
byByte0_DataToIO	Byte	8 Byte digitale Ausgangsdaten zum I/O-System
...		
byByte7_DataToIO		

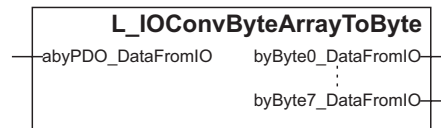
Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
abyPDO_DataToIO	Array of Byte	Array of 8 Byte: Abbildung eines Sendeobjekts <i>PDO_DataToIO</i>



5.11 L_IOConvByteArrayToByte - Konvertierungsfunktion

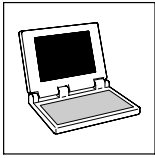
Der Funktionsblock **L_IOConvByteArrayToByte** konvertiert 8 Byte eines Empfangsobjektes in 8 Byte Eingangsdaten.

Damit entspricht die Eingangsdatenbreite einem Empfangsobjekt *PDO_DataFromIO* vom I/O-System.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
abyPDO_DataFromIO	Array of Byte	Array of 8 Byte: Abbildung eines Empfangsobjekts <i>PDO_DataFromIO</i>

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
byByte0_DataFromIO ... byByte7_DataFromIO	Byte	Empfangsobjekt: 8 x 1Byte digitale/analoge Eingangsdaten vom I/O-System)



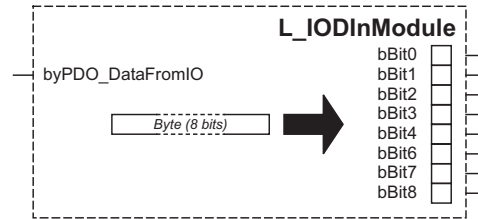
Funktionsbibliothek LenzeIOSystem.lib

Funktionsblöcke

L_IODInModule - Bitkonvertierung

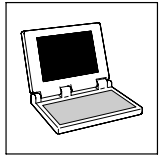
5.12 L_IODInModule - Bitkonvertierung

Der Funktionsblock **L_IODInModule** entspricht einer digitalen Eingangsscheibe (EPM-T210) und konvertiert ein Datenbyte in die einzelnen Bits 0 - 7.



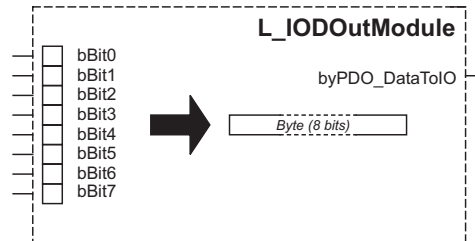
⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
byPDO_DataFromIO	Byte	Digitale Eingänge: 1 Byte Daten = 1 DI Modul

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
bBit0	Bool	digitale Eingangsinformation von Bit0.
...		...
bBit7		digitale Eingangsinformation von Bit7.



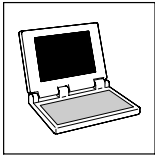
5.13 L_IODOutModule - Bitkonvertierung

Der Funktionsblock **L_IODOutModule** entspricht einer digitalen Ausgangsscheibe (EPM-T22X) und konvertiert die einzelnen Bits 0 - 7 in ein Datenbyte.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
bBit0	Bool	digitale Ausgangsinformation von Bit0
...		...
bBit7		digitale Ausgangsinformation von Bit7

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
byPDO_DataFromIO	Byte	Digitale Ausgänge: 1 Byte Daten = 1 DO Modul



Funktionsbibliothek LenzeIOSystem.lib

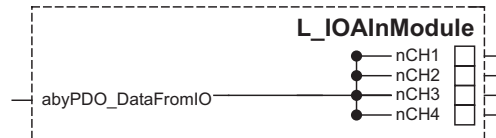
Funktionsblöcke

L_IOAIInModule - Signalkonvertierung

5.14 L_IOAIInModule - Signalkonvertierung

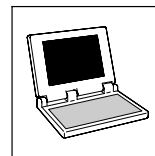
Der Funktionsblock **L_IOAIInModule** entspricht einer analogen Eingangsscheibe (EPM-T310 / T311 / T312) und teilt die vom I/O System gesendeten Daten in die Kanalinformation 1 - 4 auf.

Die Eingangsdaten werden im Integer-Format abgebildet.



⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
abyPDO_DataFromIO	Array of Byte	analoge Eingangsinformation von Kanal 1-4

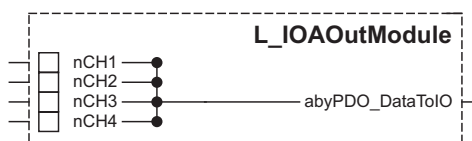
Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
nCH1 ... nCH4	Integer	analoge Eingänge: 4 x 2 Byte Daten = 1 AI Modul
		default ±10V Funktionscode 2B _h 6-1



5.15 L_IOAOutModule - Signalkonvertierung

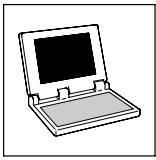
Der Funktionsblock **L_IOAOutModule** entspricht einer analogen Ausgangsscheibe (EPM-T320 / T321 / T322) und bündelt die Information von Kanal 1 - 4 in ein Sendeobjekt, das zum I/O System gesendet wird.

Die Ausgangsdaten werden im Integer-Format abgebildet.



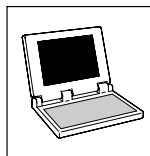
⇒ Eingänge (Variablentyp: VAR_INPUT)		
Variablenname	Datentyp	Info/Einstellmöglichkeiten
nCH1	Byte	analoge Ausgänge: 4 x 2 Byte Daten = 1 AO Modul
...		
nCH4		
		default ±10 = Funktionscode 01 _h 6-4

Ausgänge ⇒ (Variablentyp: VAR_OUTPUT)		
Bezeichner	Datentyp	Wert/Bedeutung
abyPDO_DataFromIO	Array of Byte	analoge Ausgangsinformation von Kanal 1-4



Funktionsbibliothek LenzeIOSystem.lib

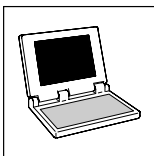
Funktionsblöcke



6 Anhang

6.1 Signalfunktionen für analoge Eingänge EPM-T310

Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
00 _h	Parameterdaten im Modul werden nicht überschrieben			
01 _h	Temperaturmessung mit Zweileiter-Anschluss (Zweierkomplement)	mit PT100	-200.0 {0.1 °C} +850.0	±1 °C ²⁾
02 _h		mit PT1000	-200.0 {0.1 °C} +500.0	±1 °C ²⁾
03 _h		mit NI100	-50.0 {0.1 °C} +250.0	±1 °C ²⁾
04 _h		mit NI1000	-50.0 {0.1 °C} +250.0	±1 °C ²⁾
05 _h	Widerstandsmessung mit Zweileiter-Anschluss	mit 60 Ω	0 {0.01 Ω} +60.00 0 {1 dec} 32767	±0.2 % vom Endwert ²⁾
05 _h		mit 600 Ω	0 {0.01 Ω} +600.00 0 {1 dec} 32767	±0.2 % vom Endwert ²⁾
07 _h		mit 3000 Ω	0 {0.01 Ω} +3000.00 0 {1 dec} 32767	±0.2 % vom Endwert ²⁾
08 _h		mit 6000 Ω	0 {0.01 Ω} +6000.00 0 {1 dec} 32767	±0.2 % vom Endwert ²⁾
09 _h	Temperaturmessung mit Vierleiter-Anschluss (Zweierkomplement)	mit PT100	-200.0 {0.1 °C} +850.0	±0.5 °C
0A _h		mit PT1000	-200.0 {0.1 °C} +500.0	±0.5 °C
0B _h		mit NI100	-50.0 {0.1 °C} +250.0	±0.5 °C
0C _h		mit NI1000	-50.0 {0.1 °C} +250.0	±0.5 °C
0D _h	Widerstandsmessung mit Zweileiter-Anschluss	mit 60 Ω	0 {0.01 Ω} +60.00	±0.1 % vom Endwert
0E _h		mit 600 Ω	0 {0.01 Ω} +600.00	±0.05 % vom Endwert
0F _h		mit 3000 Ω	0 {0.01 Ω} +3000.00	±0.05 % vom Endwert
10 _h	Temperaturmessung mit Thermoelement und externer Kompensation ³⁾ (Zweierkomplement)	mit Typ J	-210.0 {0.1 °C} +850.0	±1 °C
11 _h		mit Typ K	-270.0 {0.1 °C} +1200.0	±1.5 °C
12 _h		mit Typ N	-200.0 {0.1 °C} +1300.0	±1.5 °C
13 _h		mit Typ R	-50.0 {0.1 °C} +1760.0	±4 °C
14 _h		mit Typ T	-270.0 {0.1 °C} +400.0	±1.5 °C
15 _h		mit Typ S	-50.0 {0.1 °C} +1760.0	±5 °C
18 _h	Temperaturmessung mit Thermoelement und interner Kompensation ⁴⁾ (Zweierkomplement)	mit Typ J	-210.0 {0.1 °C} +850.0	±1.5 °C
19 _h		mit Typ K	-270.0 {0.1 °C} +1200.0	±2 °C
1A _h		mit Typ N	-200.0 {0.1 °C} +1300.0	±2 °C
1B _h		mit Typ R	-50.0 {0.1 °C} +1760.0	±5 °C
1C _h		mit Typ T	-270.0 {0.1 °C} +400.0	±2 °C
1D _h		mit Typ S	-50.0 {0.1 °C} +1760.0	±5 °C
27 _h	Spannungsmessung (Zweierkomplement)	0 ... 50 mV	0 {0.01 mV} +50.00 0 {1 dec} 27648 Untersteuergrenze: – Übersteuergrenze: +59.25 mV 32767 dec	±0.1 % vom Endwert
29 _h	Spannungsmessung (Zweierkomplement)	±4 V	-4.00 {0.01 V} +4.00 V -27648 {1 dec} 27648 dec Untersteuergrenze: -4.74 V -32767 dec Übersteuergrenze: +4.74 V 32767 dec	±0.05 % vom Endwert
2A _h	Spannungsmessung (Zweierkomplement)	±400 mV	-400 {1 mV} +400 -27648 {1 dec} 27648 Untersteuergrenze: -474 mV -32767 dec Übersteuergrenze: +474 mV 32767 dec	±0.1 % vom Endwert

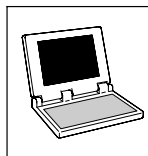


Funktionsbibliothek LenzeIOSystem.lib

Anhang

Signalfunktionen für analoge Eingänge EPM-T310

Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
3B _h	Spannungsmessung (Zweierkomplement)	±10 V -10.00 {0.01 V} +10.00 -16384 {1 dec} 16384 Untersteuerungsgrenze: -11.85 V -20480 dec Übersteuerungsgrenze: +11.85 V 20480 dec	B	±0.2 % vom Endwert
2C _h	Strommessung (Zweierkomplement)	±20 mA -20.00 {0.01 mA} +20.00 -27648 {1 dec} 27648 Untersteuerungsgrenze: -23.70 mA -32767 dec Übersteuerungsgrenze: +23.70 mA +32767 dec	A	±0.05 % vom Endwert
2D _h	Strommessung (Zweierkomplement)	4 ... 20 mA 4.00 {0.01 mA} 20.00 0 {1 dec} 27648 Untersteuerungsgrenze: 1.185 mA -4864 dec Übersteuerungsgrenze: +22.96 mA +32767 dec	A	±0.05 % vom Endwert
2E _h	Strommessung	4 ... 20 mA 4.00 {0.01 mA} 20.00 0 {1 dec} 16384 Untersteuerungsgrenze: 1.185 mA -2882 dec Übersteuerungsgrenze: +22.96 mA +20480 dec	B	±0.2 % vom Endwert
32 _h	Widerstandsmessung mit Vierleiter-Anschluss	mit 6000 Ω 0 {0.01 Ω} +6000.00 0 {1 dec} 32767 dec	A	±0.05 % vom Endwert
33 _h		mit 6000 Ω 0 {0.01 Ω} +6000.00 0 {1 dec} 6000 dec		±0.05 % vom Endwert
35 _h		mit 60 Ω 0 {0.01 Ω} +6000.00 0 6000 dec		±0.2 % vom Endwert ²⁾
36 _h		mit 600 Ω 0 {0.01 Ω} +600.00 0 {1 dec} 6000 dec		±0.1 % vom Endwert ²⁾
37 _h		mit 3000 Ω 0 {0.01 Ω} +3000.00 0 {1 dec} 30000 dec		±0.1 % vom Endwert ²⁾
38 _h		mit 6000 Ω 0 {0.01 Ω} +6000.00 0 {1 dec} 6000 dec		±0.1 % vom Endwert ²⁾



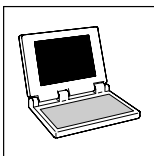
Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
3A _h	Strommessung (Zweierkomplement)	±20 mA -20.00 {0.01 mA} +20.00 -16384 {1 dec} 16384 Untersteuerungsgrenze: -23.70 mA -19456 dec Übersteuerungsgrenze: +23.70 mA +19456 dec	A	±0.05 % vom Endwert
3B _h	Spannungsmessung (Zweierkomplement)	±10 V -10.00 {0.01 V} +10.00 -16384 {1 dec} 16384 Untersteuerungsgrenze: -12.5 V -20480 dec Übersteuerungsgrenze: +12.5 V 20480 dec	B	±0.2 % vom Endwert
3D5 _h	Widerstandsmessung mit Vierleiter-Anschluss	mit 60 Ω 0 {0.01 Ω} +60.00 0 {1 dec} 6000 dec	A	±0.1 % vom Endwert
3E _h		mit 600 Ω 0 {0.01 Ω} +600.00 0 {1 dec} 6000 dec		±0.05 % vom Endwert
3F _h		mit 3000 Ω 0 {0.01 Ω} +3000.00 0 {1 dec} 30000 dec		±0.05 % vom Endwert
57 _h	Spannungsmessung	0 ... 50 mV 0 {0.01 mV} +50.00 0 {1 dec} 5000 Übersteuerungsgrenze: +59.25 V 5925 dec	A	±0.1 % vom Endwert
58 _h	Spannungsmessung (Zweierkomplement)	±10 V -10.00 {0.01 V} +10.00 -10000 {1 dec} 10000 Untersteuerungsgrenze: -11.85 V -11850 dec Übersteuerungsgrenze: +11.85 V 11850 dec	A	±0.05 % vom Endwert
59 _h	Spannungsmessung (Zweierkomplement)	±4 V -4.00 {0.01 V} +4.00 V -40000 {1 dec} 40000 Untersteuerungsgrenze: -4.74 V -47400 dec Übersteuerungsgrenze: +4.74 V 47400 dec	A	±0.05 % vom Endwert
5A _h	Spannungsmessung (Zweierkomplement)	±400 mV -400 {1 mV} +400 -40000 {1 dec} 40000 Untersteuerungsgrenze: -474 mV -47400 dec Übersteuerungsgrenze: +474 mV 47400 dec	A	±0.1 % vom Endwert
5C _h	Strommessung (Zweierkomplement)	±20 mA -20.00 {0.01 mA} +20.00 -20000 {1 dec} 20000 Untersteuerungsgrenze: -23.70 mA -23700 dec Übersteuerungsgrenze: +23.70 mA +23700 dec	A	±0.05 % vom Endwert
5D _h	Strommessung (Zweierkomplement)	4 ... 20 mA 4.00 {0.01 mA} 20.00 0 {1 dec} 16000 Untersteuerungsgrenze: 1.185 mA -2815 dec Übersteuerungsgrenze: +22.96 mA +22960 dec	A	±0.05 % vom Endwert
FF _h	Analoger Eingang abgeschaltet			

1) Die Toleranzwerte wurden ermittelt bei 25 °C Umgebungstemperatur und bei 15 Wandlungen/s. Ungenauigkeiten von Messwertgebern wurden nicht berücksichtigt.

2) Übergangswiderstände an Kontakten sowie Leitungswiderstände wurden nicht berücksichtigt.

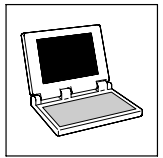
3) Die Kompensation der Kaltstelle muss extern durchgeführt werden.

4) Die Kompensation der Kaltstelle erfolgt intern. Es wird die Temperatur des Anschlussklemme berücksichtigt. Schließen Sie die Leiter der Thermoelemente unmittelbar an die Anschlussklemme an. Ggf. müssen Sie mit Thermoelement-Verlängerungskabel verlängern.



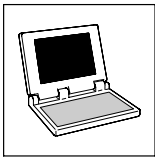
6.2 Signalfunktionen für analoge Ausgänge EPM-T320

Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Signalauf-lösung	Toleranz ¹⁾
00 _h	Keine Signalausgabe am Ausgang			
01 _h	Ausgabe Spannungssignal	±10 V -10.00 {0.01 V} +10.00 -16384 {1 dec} 16384 Untersteuerungsgrenze: -11.85 V -20480 dec Übersteuerungsgrenze: +11.85 V 20480 dec	B	±0.2 %
02 _h	Ausgabe Spannungssignal	+1 ... +5 V 1.0 {0.1 V} +5.0 0 {1 dec} 16384 Untersteuerungsgrenze: 0 V 20480 dec Übersteuerungsgrenze: +6.0 V 20480 dec	B	±0.05 %
05 _h	Ausgabe Spannungssignal	0 ... +10 V 0 {0.1 V} +10.0 0 {1 dec} 16384 Untersteuerungsgrenze: – Übersteuerungsgrenze: +12.5 V 20480 dec	B	±0.2 %
09 _h	Ausgabe Spannungssignal (Zweierkomplement)	±10 V -10.00 {0.01 V} +10.00 V -27648 {1 dec} 27648 dec Untersteuerungsgrenze: -11.85 V -32767 dec Übersteuerungsgrenze: +11.84 V 32767 dec	A	±0.05 %
0A _h	Ausgabe Spannungssignal (Zweierkomplement)	+1 ... +5 V 1.00 {0.01 V} +5.00 0 {1 dec} 27648 Untersteuerungsgrenze: 0 V -6912 dec Übersteuerungsgrenze: +5.75 V 32767 dec	A	±0.05 %
0D _h	Ausgabe Spannungssignal (Zweierkomplement)	0 ... +10 V 0 {0.1 V} +10.0 0 {1 dec} 27648 Untersteuerungsgrenze: – Übersteuerungsgrenze: +11.5 V 32767 dec	A	±0.2 %
03 _h	Ausgabe Stromsignal	±20 mA -20.00 {0.01 mA} +20.00 -16384 {1 dec} 16384 Untersteuerungsgrenze: -23.70 mA -20480 dec Übersteuerungsgrenze: +23.70 mA +20480 dec	B	±0.2 %
04 _h	Ausgabe Stromsignal	4 ... 20 mA 4.00 {0.01 mA} 20.00 0 {1 dec} 16384 Untersteuerungsgrenze: 0 mA -4096 dec Übersteuerungsgrenze: +23.70 mA +20480 dec	B	±0.2 %
06 _h	Ausgabe Stromsignal	0 ... 20 mA 0 {0.01 mA} +20.00 0 {1 dec} 16384 Untersteuerungsgrenze: – Übersteuerungsgrenze: +23.70 mA +20480 dec	B	±0.2 %



Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Signalauf-lösung	Toleranz ¹⁾
0B _h	Ausgabe Stromsignal (Zweierkomplement)	±20 mA -20.00 {0.01 mA} +20.00 -27648 {1 dec} 27648 Untersteuerungsgrenze: -23.70 mA -32767 dec Übersteuerungsgrenze: +23.70 mA +32767 dec	A	±0.05 %
0C _h	Ausgabe Stromsignal (Zweierkomplement)	4 ... 20 mA 4.00 {0.01 mA} 20.00 0 {1 dec} 27648 Untersteuerungsgrenze: 0 mA -5530 dec Übersteuerungsgrenze: +22.96 mA +32767 dec	A	±0.05 %
0E _h	Ausgabe Stromsignal (Zweierkomplement)	0 ... 20 mA 0 {0.01 mA} 20.00 0 {1 dec} 27648 Untersteuerungsgrenze: – Übersteuerungsgrenze: +22.96 mA +32767 dec	A	±0.2 %

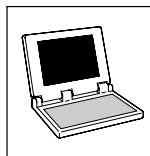
¹⁾ Die Toleranzwerte wurden ermittelt bei 25 °C Umgebungstemperatur und bei 15 Wandlungen/s. Die Werte beziehen sich jeweils auf den Endwert.



6.3 Signalfunktionen für analoge Ein- und Ausgänge EPM-T330

Eingabebereich (Kanal 0, 1)

Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
00 _h	Parameterdaten im Modul werden nicht überschrieben			
3B _h	Spannungsmessung (Zweierkomplement)	±10 V -10.00 {0.01 V} -16384 {1 dec} Untersteuergrenze: -12.50 V -20480 _{dec} Übersteuergrenze: +12.50 V +20480 _{dec}	B	±0.2 % vom Endwert
75 _h	Spannungsmessung	0...10 V 0.00 {0.01 V} 0 {1 dec} Untersteuergrenze: - Übersteuergrenze: +12.50 V +20480 _{dec}	B	±0.4 % vom Endwert
28 _h	Spannungsmessung (Zweierkomplement)	±10 V -10.00 {0.01 V} -27648 {1 dec} Untersteuergrenze: -11.76 V -32512 _{dec} Übersteuergrenze: +11.76 V +32511 _{dec}	A	±0.2 % vom Endwert
7A _h	Spannungsmessung	1...5 V +1.00 {0.01 V} 0 {1 dec} Untersteuergrenze: 0.00 V -6912 _{dec} Übersteuergrenze: +5.704 V +32511 _{dec}	A	±0.6 % vom Endwert
7D _h	Spannungsmessung	0...10 V 0.00 {0.01 V} 0 {1 dec} Untersteuergrenze: - Übersteuergrenze: +11.76 V +32511 _{dec}	A	±0.4 % vom Endwert
3A _h	Strommessung (Zweierkomplement)	±20 mA -20.00 {0.01 mA} -16384 {1 dec} Untersteuergrenze: -25.00 mA -20480 _{dec} Übersteuergrenze: +25.00 mA +20480 _{dec}	B	±0.3 % vom Endwert
76 _h	Strommessung	0...20 mA 0.00 {0.01 mA} 0 {1 dec} Untersteuergrenze: - Übersteuergrenze: +25.00 mA +20480 _{dec}	B	±0.6 % vom Endwert
2C _h	Strommessung (Zweierkomplement)	±20 mA -20.00 {0.01 mA} -27648 {1 dec} Untersteuergrenze: -23.51 mA -32512 _{dec} Übersteuergrenze: +23.51 mA +32511 _{dec}	A	±0.3 % vom Endwert
2D _h	Strommessung (Zweierkomplement)	4...20 mA +4.00 {0.01 mA} 0 {1 dec} Untersteuergrenze: +1.18 mA -4864 _{dec} Übersteuergrenze: +22.81 mA +32511 _{dec}	A	±0.8 % vom Endwert

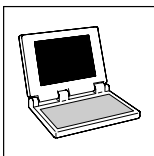


Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
7E _h	Strommessung	0...20 mA 0.00 {0.01 mA} +20.00 0 {1 dec} +27648 Untersteuerungsgrenze - Übersteuerungsgrenze: +23.52 mA +32511 _{dec}	A	±0.6 % vom Endwert
FF _h	Analoger Eingang abgeschaltet			

¹⁾ Die Toleranzwerte wurden ermittelt bei 25 °C Umgebungstemperatur und bei 15 Wandlungen/s. Ungenauigkeiten von Messwertgebern wurden nicht berücksichtigt.

Ausgabebereich (Kanal 2, 3)

Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
00 _h	Parameterdaten im Modul werden nicht überschrieben			
01 _h	Ausgabe Spannungssignal	±10 V -10.00 {0.01 V} +10.00 -16384 {1 dec} +16384 Untersteuerungsgrenze: -12.50 V -20480 _{dec} Übersteuerungsgrenze: +12.50 V +20480 _{dec}	B	±0.2 % vom Endwert
02 _h	Ausgabe Spannungssignal	1...5 V +1.00 {0.01 V} +5.00 0 {1 dec} +16384 Untersteuerungsgrenze: 0.00 V -4096 _{dec} Übersteuerungsgrenze: +6.00 V +20480 _{dec}	B	±0.6 % vom Endwert
05 _h	Ausgabe Spannungssignal	0...10 V 0.00 {0.01 V} +10.00 0 {1 dec} +16384 Untersteuerungsgrenze: - Übersteuerungsgrenze: +12.50 V +20480 _{dec}	B	±0.4 % vom Endwert
09 _h	Ausgabe Spannungssignal (Zweierkomplement)	±10 V -10.00 {0.01 V} +10.00 -27648 {1 dec} +27648 Untersteuerungsgrenze: -11.76 V -32512 _{dec} Übersteuerungsgrenze: +11.76 V +32511 _{dec}	A	±0.2 % vom Endwert
0A _h	Ausgabe Spannungssignal (Zweierkomplement)	1...5 V +1.00 {0.01 V} +5.00 0 {1 dec} +27648 Untersteuerungsgrenze: 0.00 V -6912 _{dec} Übersteuerungsgrenze: +5.704 V +32511 _{dec}	A	±0.6 % vom Endwert
0D _h	Ausgabe Spannungssignal	0...10 V 0.00 {0.01 V} +10.00 0 {1 dec} +27648 Untersteuerungsgrenze: - Übersteuerungsgrenze: +11.76 V +32511 _{dec}	A	±0.4 % vom Endwert
03 _h	Ausgabe Stromsignal	±20 mA -20.00 {0.01 mA} +20.00 -16384 {1 dec} +16384 Untersteuerungsgrenze: -25.00 mA -20480 _{dec} Übersteuerungsgrenze: +25.00 mA +20480 _{dec}	B	±0.3 % vom Endwert



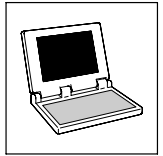
Funktionsbibliothek LenzeIOSystem.lib

Anhang

Signalfunktionen für analoge Ein- und Ausgänge EPM-T330

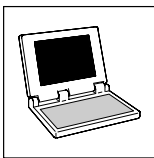
Parameter-byte 2/3/4/5	Signalfunktion	Signalbereich	Format	Toleranz ¹⁾
04 _h	Ausgabe Stromsignal	4...20 mA +4.00 {0.01 mA} +20.00 0 {1 dec} +16384 Untersteuerungsgrenze +0.00 mA -4096 _{dec} Übersteuerungsgrenze: +24.00 mA +20480 _{dec}	B	±0.8 % vom Endwert
06 _h	Ausgabe Stromsignal	0...20 mA 0.00 {0.01 mA} +20.00 0 {1 dec} +16384 Untersteuerungsgrenze - Übersteuerungsgrenze: +25.00 mA +20480 _{dec}	B	±0.6 % vom Endwert
0B _h	Ausgabe Stromsignal (Zweierkomplement)	±20 mA -20.00 {0.01 mA} +20.00 -27648 {1 dec} +27648 Untersteuerungsgrenze -23.52 mA -32512 _{dec} Übersteuerungsgrenze: +23.52 mA +32511 _{dec}	A	±0.3 % vom Endwert
0C _h	Ausgabe Stromsignal (Zweierkomplement)	4...20 mA +4.00 {0.01 mA} +20.00 0 {1 dec} +27648 Untersteuerungsgrenze +0.00 mA -6912 _{dec} Übersteuerungsgrenze: +22.81 mA +32511 _{dec}	A	±0.8 % vom Endwert
0E _h	Ausgabe Stromsignal	0...20 mA 0.00 {0.01 mA} +20.00 0 {1 dec} +27648 Untersteuerungsgrenze - Übersteuerungsgrenze: +23.52 mA +32511 _{dec}	A	±0.6 % vom Endwert
FF _h	Analoger Ausgang abgeschaltet			

¹⁾ Die Toleranzwerte wurden ermittelt bei 25 °C Umgebungstemperatur und bei 15 Wandlungen/s. Ungenauigkeiten von Messwertgebern wurden nicht berücksichtigt.



6.4 Normierung der Messwerte

Signalbereich	Signal [U] / [I]	15 Bit Signalauflösung			12 Bit Signalauflösung		
		Dezimalwert [dec]	Hexadezimalwert [hex]	Formeln zur Berechnung	Dezimalwert [dec]	Hexadezimalwert [hex]	Formeln zur Berechnung
±10 V	-10 V	-27648	9400	$\text{dec} = 27648 \cdot \frac{U}{10}$ $U = \text{dec} \cdot \frac{10}{27648}$	-16384	C000	$\text{dec} = 16384 \cdot \frac{U}{10}$ $U = \text{dec} \cdot \frac{10}{16384}$
	-5 V	-13824	CA00		-8192	E000	
	0 V	0	0000		0	0000	
	+5 V	+13824	3600		+8192	2000	
	+10 V	+27648	6C00		+16384	4000	
0 ... 10 V	0 V	0	0000	$\text{dec} = 16384 \cdot \frac{U}{10}$ $U = \text{dec} \cdot \frac{10}{16384}$	0	0000	$\text{dec} = 16384 \cdot \frac{U}{10}$ $U = \text{dec} \cdot \frac{10}{16384}$
	+5 V	+8192	2000		+8192	2000	
	+10 V	+16384	4000		+16384	4000	
1 ... 5 V	+1 V	—	—		0	0000	$\text{dec} = 27648 \cdot \frac{U - 1}{4}$ $U = \text{dec} \cdot \frac{4}{16384} + 1$
	+3 V				+8192	2000	
	+5 V				+16384	4000	
±4 V	-4 V	-27648	9400	$\text{dec} = 27648 \cdot \frac{U}{4}$ $U = \text{dec} \cdot \frac{4}{27648}$	—	—	
	0 V	0	0000				
	+4 V	+27648	6C00				
±400 mV	-400 mV	-27648	9400	$\text{dec} = 27648 \cdot \frac{U}{400}$ $U = \text{dec} \cdot \frac{400}{27648}$	—	—	
	0 V	0	0000				
	+400 mV	+27648	6C00				
4 ... 20 mA	+4 mA	0	0000	$\text{dec} = 27648 \cdot \frac{I - 4}{16}$ $U = \text{dec} \cdot \frac{16}{27648} + 1$	0	0000	$\text{dec} = 16384 \cdot \frac{I - 4}{16}$ $U = \text{dec} \cdot \frac{16}{16384} + 1$
	+12 mA	+13824	3600		+8192	2000	
	+20 mA	+27648	6C00		+16384	4000	
±20 mA	-20 mA	-27648	9400	$\text{dec} = 27648 \cdot \frac{I}{20}$ $U = \text{dec} \cdot \frac{20}{27648}$	-16384	C000	$\text{dec} = 16384 \cdot \frac{I}{20}$ $U = \text{dec} \cdot \frac{20}{16384}$
	-10 mA	-13824	CA00		-8192	E000	
	0 mA	0	0000		0	0000	
	+10 mA	+13824	3600		+8192	2000	
	+20 mA	+27648	6C00		+16384	4000	



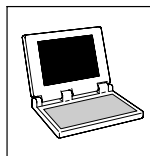
Funktionsbibliothek LenzeIOSystem.lib

Anhang

Auswahl der Zähler-Modi EPM-T410

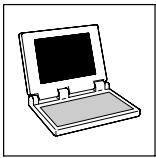
6.5 Auswahl der Zähler-Modi EPM-T410

Modus		Funktion	IN1	IN2	IN3	IN4	IN5	IN6	OUT0	OUT1	Auto Reload	Compare Load
[hex]	[dec]											
2 Zähler			0			1						
00 _h	0	32 Bit-Zähler	RES	CLK	DIR	RES	CLK	DIR	•	•	–	–
01 _h	1	Encoder 1 Flanke	RES	A	B	RES	A	B	•	•	–	–
03 _h	3	Encoder 2 Flanken	RES	A	B	RES	A	B	•	•	–	–
05 _h	5	Encoder 4 Flanken	RES	A	B	RES	A	B	•	•	–	–
4 Zähler				0.1	0.2		1.1	1.2				
08 _h	8	2 × 16 Bit-Zähler (Zählrichtung aufwärts/aufwärts)	–	CLK	CLK	–	CLK	CLK	–	–	–	–
09 _h	9	2 × 16 Bit-Zähler (Zählrichtung abwärts/aufwärts)	–	CLK	CLK	–	CLK	CLK	–	–	–	–
0A _h	10	2 × 16 Bit-Zähler (Zählrichtung abwärts/aufwärts)	–	CLK	CLK	–	CLK	CLK	–	–	–	–
0B _h	11	2 × 16 Bit-Zähler (Zählrichtung abwärts/aufwärts)	–	CLK	CLK	–	CLK	CLK	–	–	–	–
2 Zähler			0			1						
0C _h	12	2 × 32 Bit-Zähler (Zählrichtung aufwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	–	✓
0D _h	13	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	–	✓
0E _h	14	2 × 32 Bit-Zähler (Zählrichtung aufwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	✓	✓
0F _h	15	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	✓	✓
1 Zähler			0/1									
10 _h	16	Frequenzmessung	RES	CLK	START	STOP	–	–	•	•	–	✓
11 _h	17	Messung der Periodendauer	RES	CLK	START	STOP	–	–	•	•	–	✓
12 _h	18	Frequenzmessung (Ausgabe Zähler läuft/stoppt)	RES	CLK	START	STOP	–	–	•	•	–	✓
13 _h	19	Messung der Periodendauer (Ausgabe Zähler läuft/stoppt)	RES	CLK	START	STOP	–	–	•	•	–	✓
2 Zähler			0			1						
06 _h	6	Pulsbreite messen (f _{ref} 50 kHz, Zählrichtung wählbar)	RES	PULSE	DIR	RES	PULSE	DIR	–	–	–	–
14 _h	20	Pulsbreite messen (f _{ref} programmierbar, Zählrichtung wählbar)	RES	PULSE	DIR	RES	PULSE	DIR	–	–	–	–
15 _h	21	Pulsbreite messen (f _{ref} programmierbar, Zählrichtung aufwärts)	RES	PULSE	DIR	RES	PULSE	GATE	–	–	–	–
16 _h	22	Pulsbreite messen (f _{ref} programmierbar, Zählrichtung abwärts)	RES	PULSE	DIR	RES	PULSE	GATE	–	–	–	–
2 Zähler			0			1						
17 _h	23	2 × 32 Bit-Zähler (Zählrichtung aufwärts, Set-Funktion)	RES	CLK	GATE	RES	CLK	GATE	–	–	–	✓
18 _h	24	2 × 32 Bit-Zähler (Zählrichtung aufwärts, Set-Funktion)	RES	CLK	GATE	RES	CLK	GATE	–	–	–	✓
19 _h	25	2 × 32 Bit-Zähler (Zählrichtung aufwärts, Reset-Funktion)	RES	CLK	GATE	RES	CLK	GATE	–	–	–	✓
1A _h	26	2 × 32 Bit-Zähler (Zählrichtung aufwärts, Reset-Funktion)	RES	CLK	GATE	RES	CLK	GATE	–	–	–	✓
2 Zähler			0			1						
1B _h	27	32 Bit-Zähler	G/RES	CLK	DIR	G/RES	CLK	DIR	•	•	–	–
1C _h	28	Encoder 1 Flanke	G/RES	A	B	G/RES	A	B	•	•	–	–
1D _h	29	Encoder 2 Flanken	G/RES	A	B	G/RES	A	B	•	•	–	–
1E _h	30	Encoder 4 Flanken	G/RES	A	B	G/RES	A	B	•	•	–	–
2 Zähler			0			1						
1F _h	31	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	–	✓
20 _h	32	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES	CLK	GATE	RES	CLK	GATE	•	•	–	✓



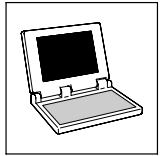
Modus		Funktion	IN1	IN2	IN3	IN4	IN5	IN6	OUT0	OUT1	Auto Reload	Compare Load
[hex]	[dec]											
21 _h	33	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES _↓	CLK	GATE	RES _↓	CLK	GATE	•	•	✓	✓
22 _h	34	2 × 32 Bit-Zähler (Zählrichtung abwärts)	RES _↓	CLK	GATE	RES _↓	CLK	GATE	•	•	✓	✓

- Digitaler Ausgang kann ein Ereignis melden
- ✓ Funktion steht zur Verfügung
- Keine Funktion / Funktion steht nicht zur Verfügung
- A Encoder-Signal A
- Auto Reload Mit Auto Reload übernimmt der Zähler einen vorgegebenen Wert, sobald der Zählerstand mit dem Inhalt des Compare-Registers übereinstimmt.
- B Encoder-Signal B
- Compare Load Mit Compare Load können sie einen Zählerendwert angeben, bei dessen Erreichen ein Ausgang ansteuert wird oder über Auto Reload die Zähler neu gestartet werden.
- CLK Taktsignal eines angeschlossenen Gebers
HIGH: Startet bzw. stoppt den Zählvorgang
- DIR Gibt je nach Signalpegel die Zählrichtung an
LOW: Aufwärtszähler
HIGH: Abwärtszähler
- GATE Gate-Signal pegelgetriggert
HIGH: Impulse werden gemessen
- G/RES_↓ Gate-Signal pegelgetriggert und Reset-Signal flankengetriggert
HIGH: Impulse werden gemessen
LOW-HIGH-Flanke: Löscht einen oder beide Zähler
- PULSE Die Pulsbreite des eingespeisten Signals wird mit einer internen Zeitbasis gemessen
- RES Reset-Signal pegelgetriggert
HIGH: Löscht einen oder beide Zähler
- RES_↓ Reset-Signal flankengetriggert
LOW-HIGH-Flanke: Löscht einen oder beide Zähler
- START Startsignal flankengetriggert
- STOP Stoppsignal flankengetriggert



Funktionsbibliothek LenzeIOSystem.lib

Anhang



7 Index

A

analoge Ausgänge EPM-T320, 6-4
analoge Ein- und Ausgänge EPM-T330, 6-6
analoge Eingänge EPM-T310, 6-1

B

Begriffsdefinitionen, 1-2
Bitkonvertierung, 5-20 , 5-21

C

CAN-Treiber initialisieren, CAN-Treiber, initialisieren, 3-1

D

Daten koordinieren, 5-2 , 5-5 , 5-8

I

Initialisieren, 3-5

K

Konvertierungsfunktion, 5-18 , 5-19

L

L_IOAInModule, 5-22
L_IOAOutModule, 5-23
L_IOConvByteArrayToByte, 5-19
L_IOConvByteToByteArray, 5-18
L_IOCounterDataFromIO, 5-13
L_IOCounterDataToIO, 5-11
L_IOCounterDIModuleDataFromIO, 5-15
L_IOCounterDIModuleDataToIO, 5-14
L_IODData15, 5-2
L_IODData610, 5-5
L_IODDataCompactModule, 5-8
L_IODInModule, 5-20
L_IODOutModule, 5-21
L_IOParAIAOModule, 4-9
L_IOParAInModule, 4-4
L_IOParAOutModule, 4-7

L_IOParComGuarding, 4-2
L_IOParCompactModule, 4-26
L_IOParCounterDIModule, 4-16
L_IOParCounterModule, 4-14
L_IOParPDO15, 4-18
L_IOParPDO610, 4-22
L_IOParSSIModule, 4-12
L_IOSSIDataFromIO, 5-17
L_IOSSIDataToIO, 5-16

M

Minimalkonfiguration, 2-2

N

Normierung, 6-9

P

Parameterdatenbausteine, 2-1
Parametrierfunktion, 4-4 , 4-7 , 4-9 , 4-12 , 4-18 , 4-22
 , 4-26
Prozessdatenbausteine, 2-1

S

Sicherheitshinweise, Gestaltung
Sonstige Hinweise, 1-2
Warnung vor Sachschäden, 1-2
Signalfunktionen, 6-1 , 6-4 , 6-6
Signalkonvertierung, 5-22 , 5-23

U

Überwachungsfunktion, 4-2

V

Versionskennung der Funktionsbibliothek, 1-3

Z

Zähler-Modi, 6-10
Zählerfunktion, 4-14 , 4-16 , 5-11 , 5-13 , 5-14 , 5-15 ,
5-16 , 5-17