

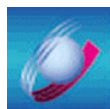
Lenze

Handbuch

IEC 61131-3

inside

***Global Drive
PLC Developer Studio***



Global Drive

Funktionsbibliothek

LenzeCanDSxDrv.lib

Die Funktionsbibliothek **LenzeCanDSxDrv.lib** kann für folgende Lenze PLC-Geräte verwendet werden:

	Typenbezeichnung	ab Hardwarestand	ab Softwarestand
9300 Servo PLC	EVS93XX-xI	6A	6.0
9300 Servo PLC	EVS93XX-xT	6A	6.0
Drive PLC	EPL10200	1A	6.0
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.

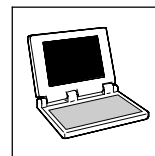
Wir übernehmen keine Verantwortung für direkt oder indirekt entstandene Schäden, z. B. Gewinnverluste, Auftragsverluste oder geschäftliche Beeinträchtigungen jeglicher Art.

© 2006 Lenze Drive Systems GmbH

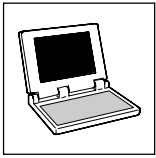
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.

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.

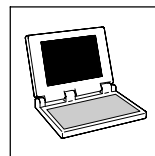


1 Vorwort und Allgemeines	1-1
1.1 Über dieses Handbuch	1-1
1.1.1 Konventionen in diesem Handbuch	1-1
1.1.2 Aufbau der Beschreibungen	1-2
1.1.3 Piktogramme in diesem Handbuch	1-2
1.1.4 Verwendete Begriffe	1-2
1.2 Versionskennung der Funktionsbibliothek	1-3
2 Einführung	2-1
2.1 Allgemeines	2-1
2.2 CanDSx-Treiber	2-1
2.2.1 Funktionsprinzip anhand eines Beispiels	2-2
2.3 Überwachungsmechanismen	2-3
2.3.1 "Heartbeat"	2-3
2.3.2 "Node Guarding"	2-4
3 Funktionen/Funktionsblöcke	3-1
3.1 Übersicht	3-1
3.2 L_CanDSxInitIndexCode - Index-Mapping konfigurieren	3-2
3.3 L_CanDSxOpen - CanDSx-Treiber initialisieren	3-4
3.4 L_CanDSxClose - Index-Mapping deaktivieren	3-5
3.5 L_CanDSxOpenHeartBeat - "Heartbeat" initialisieren	3-6
3.6 L_CanDSxHeartBeat - "Heartbeat" ausführen	3-7
3.7 L_CanDSxCloseHeartBeat - "Heartbeat" deaktivieren	3-9
3.8 L_CanDSxOpenNodeGuarding - "Node Guarding" initialisieren	3-10
3.9 L_CanDSxNodeGuarding - "Node Guarding" ausführen	3-11
3.10 L_CanDSxCloseNodeGuarding - "Node Guarding" deaktivieren	3-14



Funktionsbibliothek LenzeCanDSxDrv.lib

Inhalt



1 Vorwort und Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch enthält Informationen zur Funktionsbibliothek **LenzeCanDSxDrv.lib** für das **Drive PLC Developer Studio**.

- In der Funktionsbibliothek **LenzeCanDSxDrv.lib** sind Funktionen enthalten, mit denen über die Systembus-Schnittstelle empfangene CAN-Indizes innerhalb der PLC auf andere als die automatisch zugeordneten Codestellen "gemappt" werden können.
- Desweiteren werden Funktionen/FBs zur Verfügung gestellt, mit denen die Überwachungsmechanismen "Heartbeat" und "Node Guarding" zur Sicherstellung der Funktionalität von Systembus-Teilnehmern realisiert werden können.

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 *bEnable...*"



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 IEC61131-3 Programmierung".

Lenze Funktionen/Funktionsblöcke

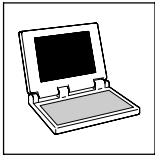
erkennen Sie am Namen. Sie beginnen immer mit einem "L_":

- "Mit dem FB **L_XYZ...**"

Programmlistings

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

- "**IF** (nValue1 < 0) **THEN...**"



Funktionsbibliothek LenzeCanDSxDrv.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:

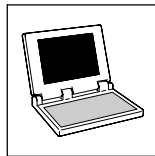
	Funktion	Funktionsblock (FB)/ Systembaustein (SB)
①	Überschrift mit Funktion und Bezeichner	
②	Deklaration der Funktion: • Datentyp des Rückgabewertes • Funktionsbezeichner • Liste der Übergabeparameter	-
③	Kurzbeschreibung mit den wichtigsten Eigenschaften	
④	Grafische Darstellung der Funktion mit den 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.

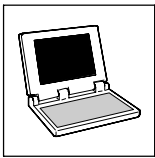
Versionskennung ab PLC Software-Stand 7.x:

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

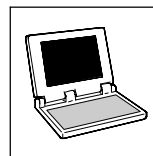
Der Wert dieser Konstanten ist hexadezimal-kodiert.

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



Funktionsbibliothek LenzeCanDSxDrv.lib

Vorwort und Allgemeines



2 Einführung

2.1 Allgemeines

Für die CAN-Kommunikation existieren diverse Kommunikationsprofile, die das kommunikationsmäßige Verhalten von CAN-Busteilnehmern definieren, z. B.:

- **DS301** - Standard-Kommunikation zwischen Busteilnehmern
- **DS402** - Kommunikation und Steuerung von Positioniersystemen

In diesen Kommunikationsprofilen sind u. a. folgende Punkte beschrieben:

- Objektverzeichnis (Zusammenhang zwischen Indexnummern und Gerätefunktionen)
- Netzwerkmanagement (Start, Reset, Zustände von Busteilnehmern)
- Bus-Boot-up-Nachrichten
- Busüberwachung (Funktionen "Node Guarding", "Heartbeat")
- Servicedatenkommunikation (Schreiben/Lesen von Indexen/Parametern)
- Prozeßdatenkommunikation (ereignisgesteuerter/zyklischer Datenverkehr)
- Blockdatentransfer



Tip!

Die Funktionsbibliotheken **LenzeCanDrv.lib** und **LenzeCanDSxDrv.lib** stellen Funktionen zur Verfügung, mit denen sich in IEC61131 Kommunikationsmechanismen programmieren lassen, die den wesentlichen Anforderungen der oben genannten Kommunikationsprofile entsprechen.

2.2 CanDSx-Treiber

Das Betriebssystem (ab V6.0) der Lenze PLC's enthält einen speziellen CanDSx-Treiber, der mit Hilfe der Funktionen der Funktionsbibliothek **LenzeCanDSxDrv.lib** aktiviert werden kann.

Mit Hilfe dieses Treibers lassen sich Indizes innerhalb der PLC einer anderen als der automatisch zugeordneten Codestelle zuweisen.



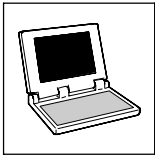
Hinweise!

- Jede Lenze-Codestelle ist über folgende Formel fest einem Index zugeordnet:

$$\text{Index} = 5FFF_{\text{hex}} - \text{Codestelle}$$

$$\text{Index} = 24575_{\text{dec}} - \text{Codestelle}$$

- Die Funktion des CanDSx-Treibers beschränkt sich nur auf den Systembus (CAN). Für den bei der Drive PLC zur Verfügung stehenden zweiten FIF-CAN-Kanal kann der CanDSx-Treiber nicht verwendet werden!



Funktionsbibliothek *LenzeCanDSxDrv.lib*

Einführung

CanDSx-Treiber

2.2.1 Funktionsprinzip anhand eines Beispiels

Aufgabe

In der PLC wurde von Anwenderseite eine Funktionalität realisiert, die über die User-Codestelle C3200/5 parametrieren werden kann. Der Codestelle C3200 ist automatisch der Index 21375_{dec} zugeordnet:

$$\text{Index} = 24575_{\text{dec}} - \text{Codestelle} = 24575_{\text{dec}} - 3200 = 21375_{\text{dec}}$$

Aufgrund des verwendeten Kommunikationsprofils soll diese Funktionalität stattdessen aber über den Index 4101_{dec}/Subindex 2 parametrieren werden können.

Lösung

Mit Hilfe der Funktionen der Funktionsbibliothek **LenzeCanDSxDrv.lib** wird der Index 4101_{dec}/Subindex 2 innerhalb der PLC einfach auf die Codestelle C3200/5 umgeleitet, so daß das Kommunikationsprofil unverändert verwendet werden kann.

Funktionsprinzip

Das Betriebssystem (ab V6.0) der Lenze PLC's enthält eine sogenannte "Mappingtabelle", anhand der bis zu 256 Indizes innerhalb der PLC auf andere als die automatisch zugeordneten Codestellen "gemappt" werden können.

Läuft ein CAN-Telegramm ein und liegt der Index im gültigen Bereich, so wird überprüft, ob dieser Index in der Mappingtabelle aufgeführt ist.

- Ist der Index in der Mappingtabelle aufgeführt, so wird auf die Codestelle zugegriffen, die diesem Index in der Mappingtabelle neu zugeordnet wurde. ①
- Ist der Index **nicht** in der Mappingtabelle aufgeführt, so wird auf die automatisch zugeordnete Codestelle zugegriffen, die sich aus der oben aufgeführten Formel ergibt. ②

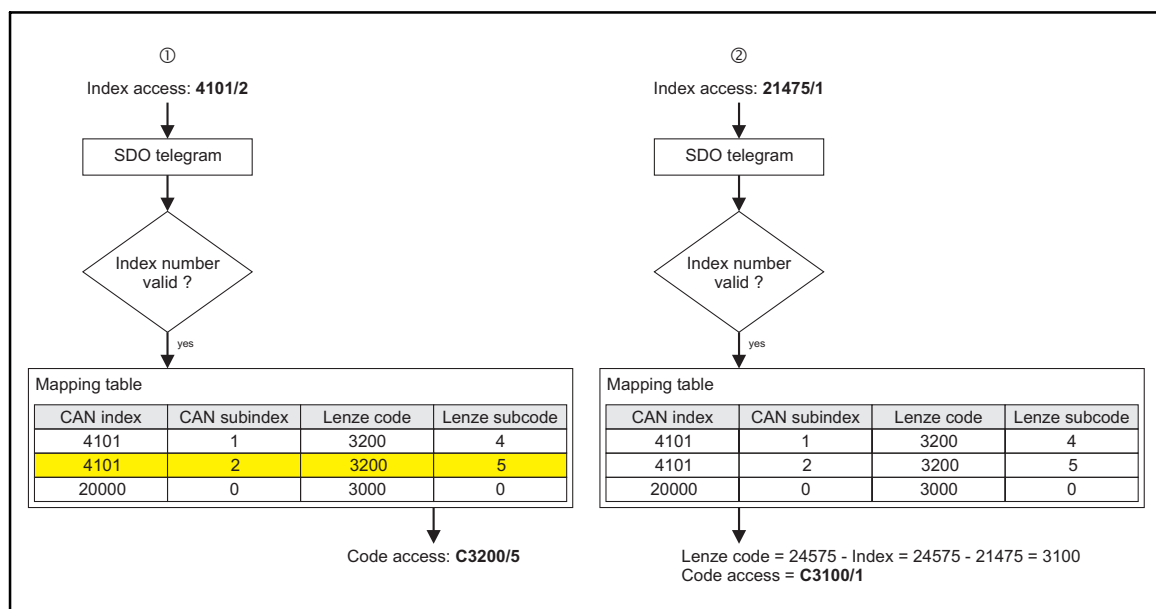
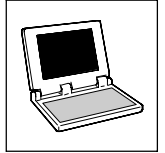


Abb. 2-1 Ablauf der Umleitung von Indizes auf Codestellen

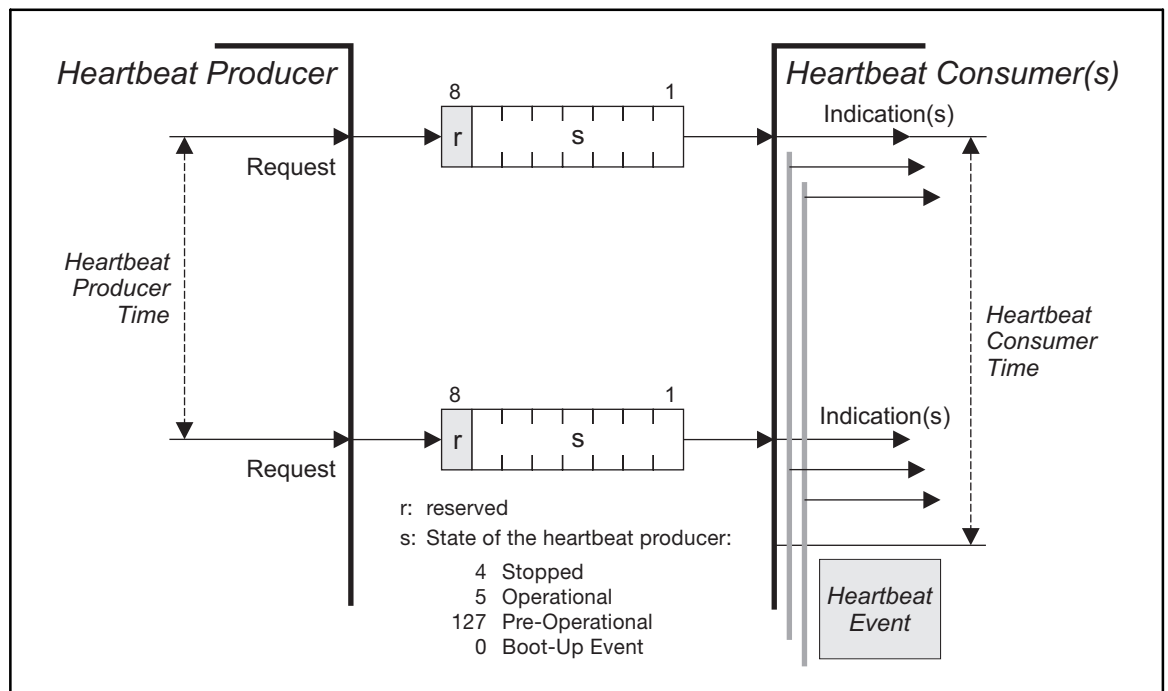


2.3 Überwachungsmechanismen

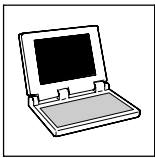
Im Kommunikationsprofil CANopen (CiA DS301, Version 4.01) sind zwei optionale Überwachungsmechanismen zur Sicherstellung der Funktionalität von Systembus-Teilnehmern spezifiziert, "Heartbeat" und "Node Guarding".

2.3.1 "Heartbeat"

Der "Heartbeat"-Überwachungsmechanismus ist eine *Producer-Consumer* orientierte Methode, bei der keine Abfragenachricht erforderlich ist und jeder Busteilnehmer den Zustand aller anderen Busteilnehmer überwachen kann.



- Ein Busteilnehmer (*Producer*) signalisiert seinen Kommunikationsstatus durch zyklisches Senden einer sogenannten "Heartbeat"-Nachricht.
- Diese "Heartbeat"-Nachricht können ein, mehrere oder alle anderen Busteilnehmer (*Consumer*) empfangen und damit den entsprechenden Busteilnehmer überwachen.
- Empfängt der verantwortliche überwachende Busteilnehmer (*Consumer*) nicht innerhalb der eingestellten Überwachungszeit (*HeartBeatConsumerTime*) die "Heartbeat"-Nachricht vom zu überwachenden Busteilnehmer (*Producer*), so wird in seiner Anwendung ein "Heartbeat"-Event angezeigt.



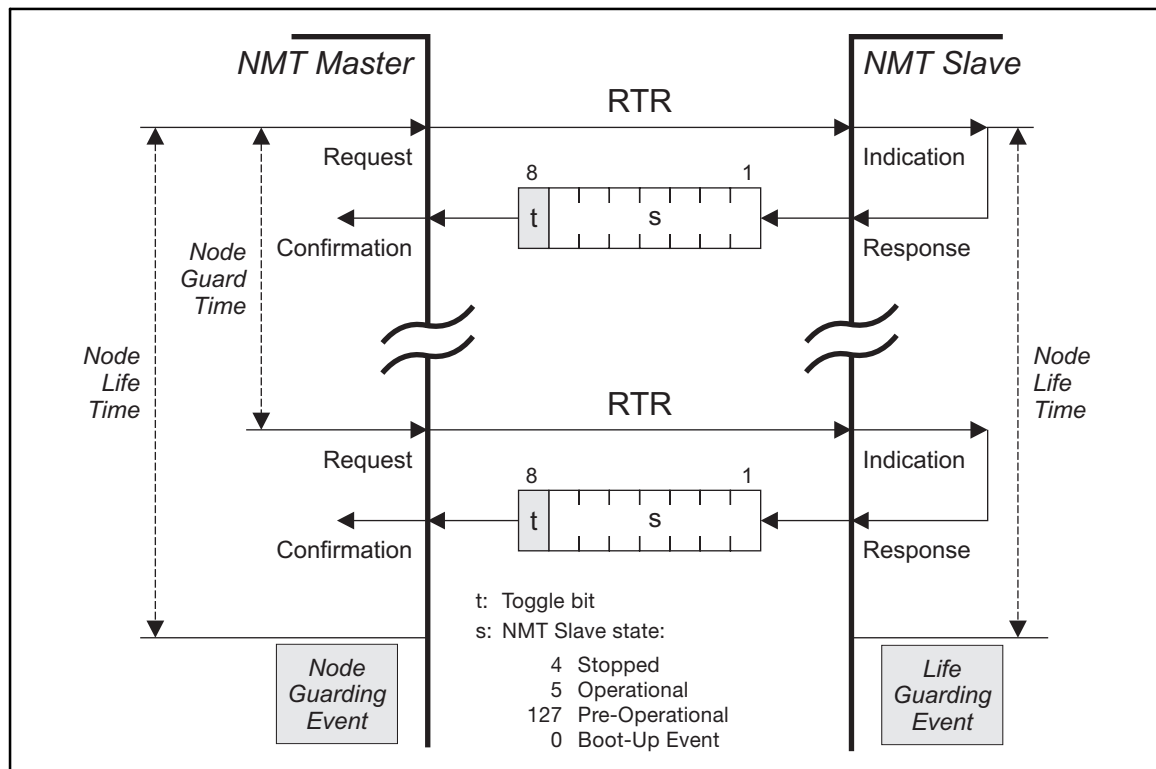
Funktionsbibliothek LenzeCanDSxDrv.lib

Einführung

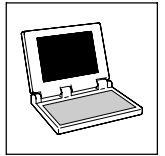
Überwachungsmechanismen

2.3.2 "Node Guarding"

Im Gegensatz zum "Heartbeat"-Überwachungsmechanismus ist beim "Node Guarding" eine Abfragenachricht vom überwachenden Busteilnehmer (*NMT Master*) erforderlich.



- Der überwachende Busteilnehmer (*NMT Master*) pollt jeden zu überwachenden Busteilnehmer (*NMT Slave*) zyklisch mit einem knotenspezifischen "Remote-Transmission-Request"-Telegramm.
- Der zu überwachende Busteilnehmer (*NMT Slave*) sendet als Antwort auf diese Anfrage seinen Kommunikationsstatus zurück.
- Empfängt der überwachende Busteilnehmer (*NMT Master*) nicht innerhalb der eingestellten Überwachungszeit (*NodeLifeTime*) die Nachricht vom zu überwachenden Busteilnehmer (*NMT Slave*), so wird in seiner Anwendung ein "Node Guarding"-Event angezeigt.
- Beim zu überwachenden Busteilnehmer (*NMT Slave*) wird dagegen ein "Life Guarding"-Event ausgelöst, wenn sein Status vom überwachenden Busteilnehmer (*NMT Master*) länger als seine "Node Life Time" nicht mehr abgefragt wurde.



3 Funktionen/Funktionsblöcke

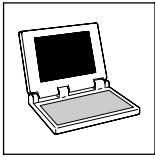
3.1 Übersicht

Funktion/FB	Info	
Mapping von Indizes auf Codestellen		
Die Funktionen beschränken sich auf den Parameterzugriff über die in der PLC integrierte Systembus-Schnittstelle.		
L_CanDSxInitIndexCode	Index-Mapping konfigurieren	3-2
L_CanDSxOpen	Index-Mapping aktivieren	3-4
L_CanDSxClose	Index-Mapping wieder deaktivieren	3-5
Überwachungsmechanismus "Heartbeat"		
L_CanDSxOpenHeartBeat	"Heartbeat" initialisieren	3-6
L_CanDSxHeartBeat	"Heartbeat" ausführen	3-7
L_CanDSxCloseHeartBeat	"Heartbeat" wieder deaktivieren	3-9
Überwachungsmechanismus "Node Guarding"		
L_CanDSxOpenNodeGuarding	"Node Guarding" initialisieren	3-10
L_CanDSxNodeGuarding	"Node Guarding" ausführen	3-11
L_CanDSxCloseNodeGuarding	"Node Guarding" wieder deaktivieren	3-14



Tip!

Für ein Mapping von Codestellen, auf die über die AIF-Schnittstelle zugegriffen wird, steht Ihnen die Funktionsbibliothek **LenzeAifParMapDrv.lib** mit entsprechenden Funktionen zur Verfügung.



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxInitIndexCode - Index-Mapping konfigurieren

3.2 L_CanDSxInitIndexCode - Index-Mapping konfigurieren

Funktion

Mit dieser Funktion wird die Mappingtabelle und somit die Umleitung von Indizes auf andere als die automatisch zugeordneten Codestellen konfiguriert.

- Mit jedem Funktionsaufruf kann ein Index und dessen zugehörige Lenze-Codestelle in die Mappingtabelle eingetragen werden.

Deklaration	
INT	L_CanDSxInitIndexCode (byTabIndex, wCANIndex, byCANSubIndex, wLenzeCodeNumber, byLenzeSubCodeNumber);

Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten	
byTabIndex	Byte	0 ... 255	Nummer des Konfigurationseintrags in der Mappingtabelle.
Umzuleitender Index:			
wCANIndex	Word	1000 _{hex} ... 8FFF _{hex} (4096 _{dec} ... 36863 _{dec})	CAN-Index
byCANSubIndex	Byte	0 ... 255	CAN-SubIndex
Umleitungsziel (Lenze-Codestelle):			
wLenzeCodeNumber	Word	1 ... 7999	Codenummer
byLenzeSubCodeNumber	Byte	0 ... 255	Subcodenummer

Rückgabewert	Datentyp	Wert/Bedeutung	
	Integer	Status	
		0	Der Eintrag in die Mappingtabelle war erfolgreich.
		-20	Fehler: Übergabeparameter <i>wCANIndex</i> ist ungültig.
		-30	Fehler: Übergabeparameter <i>wLenzeCodeNumber</i> ist ungültig.

- Die Mappingtabelle kann maximal 256 Einträge aufnehmen:

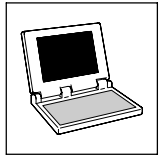
byTabIndex	Umzuleitende Codestelle		Umleitungsziel	
	wCANIndex	byCANSubIndex	wLenzeCodeNumber	byLenzeSubCodeNumber
0				
1	4104	2	3200	5
2				
...				
255				



Hinweis!

Wenn die Funktion **L_CanDSxInitIndexCode** aufgerufen wird, während noch Schreib- oder Leseaufträge für Codestellen aktiv sind, so werden diese evtl. gestört!

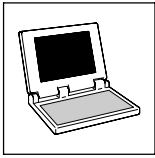
Daher sollten alle Aktionen mit Codestellenzugriff vor dem Aufruf dieser Funktion abgeschlossen sein.



Beispiel

Aufruf der Funktion in ST:

```
nReturnInitIndexCode := L_CanDSxInitIndexCode (byTabIndex:=1,  
                                                wCANIndex:=4101,  
                                                byCANSubIndex:=2,  
                                                wLenzeCodeNumber:=3200,  
                                                byLenzeSubCodeNumber:=5);
```



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxOpen - CanDSx-Treiber initialisieren

3.3

L_CanDSxOpen - CanDSx-Treiber initialisieren

Funktion

Mit dieser Funktion wird der CanDSx-Treiber im Betriebssystem der PLC initialisiert.

- Der Übergabeparameter *bOpen* muß für die Initialisierung auf TRUE gesetzt sein.
- Nach Ausführung dieser Funktion werden Indexzugriffe über die Systembus-Schnittstelle entsprechend der mit Hilfe der Funktion **L_CanDSxIndexInitCode** konfigurierten Mappingtabelle auf andere als die automatisch zugeordneten Codestellen umgeleitet.

Deklaration	
DWORD	L_CanDSxOpen (bOpen) ;

Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bOpen	Bool	CanDSx-Treiber im Betriebssystem initialisieren.
		TRUE Der CanDSx-Treiber im Betriebssystem wird initialisiert.

Rückgabewert	Datentyp	Wert/Bedeutung	
	Double Word	Status	
		Bit	Wert
		0	0 Treiber ist initialisiert.
			1 Treiber ist nicht initialisiert.
			• Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bOpen</i> = TRUE.
		1-15	Reserviert für zukünftige Erweiterungen (Bits sind auf 0 gesetzt).
			• Ungültig bei Bit 0 = 1
		16-31	Version der Funktionsbibliothek LenzeCanDSxDrv.lib
			• Format: Hauptversion/Unterversion (z. B. 0103hex = Version 1.03)
			• Ungültig bei Bit 0 = 1



Hinweis!

Wenn die Funktion **L_CanDSxOpen** aufgerufen wird, während noch Schreib- oder Leseaufträge für Codestellen aktiv sind, so werden diese evtl. gestört!

Daher sollten alle Aktionen mit Codestellenzugriff vor dem Aufruf dieser Funktion abgeschlossen sein.

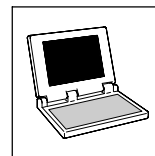
Beispiel

Aufruf der Funktion in ST:

```

IF bOpenCanDSxDriver AND NOT bOpen THEN
    bOpen := TRUE;
    dwReturnOpen := L_CanDSxOpen (bOpen:=TRUE) ;
END_IF

```



3.4 L_CanDSxClose - Index-Mapping deaktivieren

Funktion

Mit dieser Funktion wird die Mappingtabelle und somit die Umleitung von Indizes wieder deaktiviert.

- Der Übergabeparameter *bClose* muß für die Deaktivierung auf TRUE gesetzt sein.
- Nach Ausführung dieser Funktion werden Indexzugriffe über die Systembus-Schnittstelle **nicht mehr** entsprechend der Mappingtabelle auf andere Codestellen umgeleitet.

Deklaration	
BOOL	L_CanDSxClose (bClose);

Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bClose	Bool	Indexumleitungen gemäß Mappingtabelle deaktivieren.
		TRUE Der CanDSx-Treiber im Betriebssystem wird deaktiviert.

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Der CanDSx-Treiber im Betriebssystem wurde deaktiviert.
		FALSE Der CanDSx-Treiber im Betriebssystem wurde nicht deaktiviert. • Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bClose</i> = TRUE.



Hinweis!

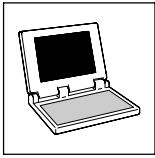
Wenn die Funktion **L_CanDSxClose** aufgerufen wird, während noch Schreib- oder Leseaufträge für Codestellen aktiv sind, so werden diese evtl. gestört!

Daher sollten alle Aktionen mit Codestellenzugriff vor dem Aufruf dieser Funktion abgeschlossen sein.

Beispiel

Aufruf der Funktion in ST:

```
IF bCloseCanDSxDriver AND NOT bClose THEN
  bClose := TRUE;
  dwReturnClose := L_CanDSxClose (bClose:=TRUE);
END_IF
```



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxOpenHeartBeat - "Heartbeat" initialisieren

3.5

L_CanDSxOpenHeartBeat - "Heartbeat" initialisieren

Funktion

Im Kommunikationsprofil CANopen (CiA DS301, Version 4.01) sind zwei optionale Überwachungsmechanismen zur Sicherstellung der Funktionalität von Systembus-Teilnehmern spezifiziert, "Heartbeat" und "Node Guarding".

Mit dieser Funktion wird der Überwachungsmechanismus "Heartbeat" des CanDSx-Treibers initialisiert.

- Der Übergabeparameter *bOpen* muß für die Initialisierung auf TRUE gesetzt sein.
- Die eigentliche Überwachung erfolgt mit Hilfe des FBs **L_CanDSxHeartBeat**. (▢ 3-7)
- Mit der Funktion **L_CanDSxCloseHeartBeat** können Sie den Überwachungsmechanismus "Heartbeat" wieder deaktivieren. (▢ 3-9)



Hinweis!

Die gleichzeitige Anwendung beider Überwachungsmechanismen ist nicht erlaubt!

Ist beim zu überwachenden Busteilnehmer eine Sendezykluszeit für die "Heartbeat"-Nachricht ungleich Null konfiguriert, so wird der "Heartbeat"-Mechanismus vorrangig vor dem "Node Guarding"-Mechanismus angewendet.

Deklaration	
BOOL	L_CanDSxOpenHeartBeat (<i>bOpen</i>);

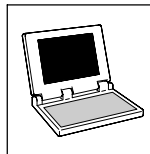
Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bOpen	Bool	Überwachungsmechanismus "Heartbeat" initialisieren.
		TRUE Der Überwachungsmechanismus "Heartbeat" des CanDSx-Treibers wird initialisiert.

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Überwachungsmechanismus "Heartbeat" wurde initialisiert.
		FALSE A Überwachungsmechanismus "Heartbeat" wurde nicht initialisiert. – Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bOpen</i> = TRUE. oder B Es wurde zuvor die Funktion L_CanDSxOpenNodeGuarding aufgerufen ("Node Guarding" ist aktiviert). – Abhilfe: Funktionsaufruf L_CanDSxCloseNodeGuarding mit Übergabeparameter <i>bClose</i> = TRUE ("Node Guarding" deaktivieren).

Beispiel

Aufruf der Funktion in ST:

```
bReturnOpenHeartBeat := L_CanDSxOpenHeartBeat (bOpen:=TRUE);
```

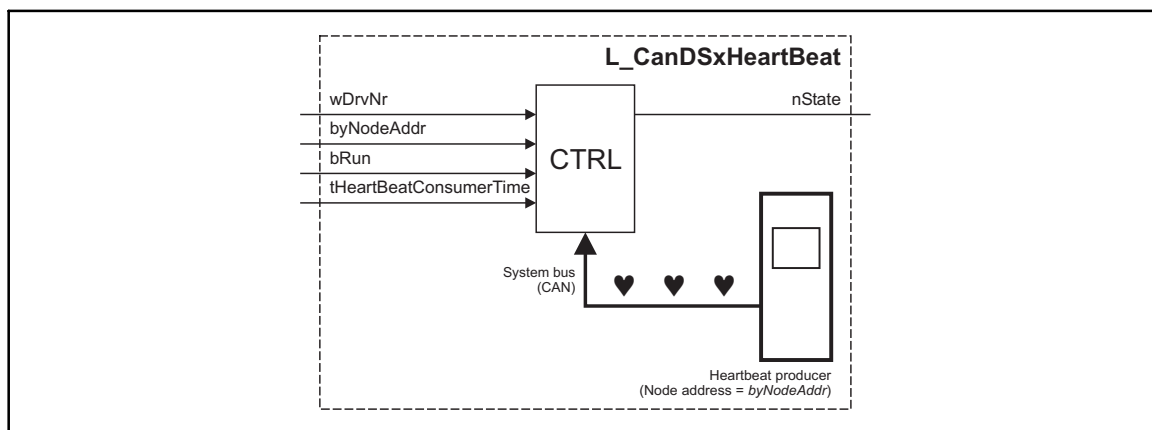


3.6 L_CanDSxHeartBeat - "Heartbeat" ausführen

Funktionsblock

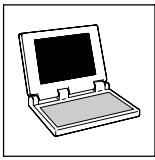
Mit diesem FB können Sie die CAN-Verbindung zwischen der PLC und anderen Systembus-Teilnehmern mit Hilfe des sogenannten "Heartbeat"-Mechanismus zyklisch überwachen.

- Der FB übernimmt hierbei die Funktion des "Heartbeat Consumer" und muß daher auf der überwachenden PLC aufgerufen werden.
- Der "Heartbeat"-Überwachungsmechanismus ist zuvor mit Hilfe der Funktion **L_CanDSxOpenHeartBeat** im CanDSx-Treiber zu initialisieren. (☐ 3-6)



FB-Aufruf in:	☐ Zyklischer Task (PLC_PRG)	☒ Zeitgesteuerter Task (INTERVAL)	☐ Ereignisgesteuerter Task (EVENT)	☐ Interrupt-Task
----------------------	--	---	---	---

Eingänge	Datentyp	Info/Einstellmöglichkeiten
wDrv	Word	Treiber Nummer für die zu verwendene CAN-Schnittstelle der PLC 10 On board-Systembus (CAN)
byNodeAddr	Byte	Knotenadresse des Busteilnehmers, der überwacht werden soll. 1 ... 128 Knotenadresse
bRun	Bool	"Heartbeat"-Überwachung aktivieren. TRUE Überwachung ist aktiviert. Der FB überwacht nun fortlaufend, ob der Busteilnehmer mit der Knotenadresse <i>byNodeAddr</i> seine "Heartbeat"-Nachricht innerhalb der eingestellten Überwachungszeit <i>tHeartBeatConsumerTime</i> über den Systembus sendet und gibt einen entsprechenden Status am Ausgang <i>nState</i> aus.
tHeartBeat ConsumerTime	Time	Überwachungszeit, innerhalb der die "Heartbeat"-Nachricht vom zu überwachenden Busteilnehmer eintreffen muß, damit kein "Heartbeat"-Event (<i>nState</i> = -10) ausgelöst wird. - Passen Sie diese Zeit entsprechend der "Heartbeat Producer Time" an, mit der der zu überwachende Busteilnehmer die "Heartbeat"-Nachricht versendet. - Empfohlene Einstellung: 200 ... 2000 ms - Maximale Einstellung: 65535 ms



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxHeartBeat - "Heartbeat" ausführen

Ausgänge	Datentyp	Info/Einstellmöglichkeiten
nState	Integer	Status
		300 FB ist deaktiviert (<i>bRun</i> = FALSE).
		127 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Pre-Operational</i> .
		5 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Operational</i> .
		4 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Stopped</i> .
		0 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Boot-Up</i> oder der FB wird nicht aufgerufen.
		-5 Die Überwachungszeit <i>tHeartBeatConsumerTime</i> ist auf den Wert "0" eingestellt.
		-10 "Heartbeat"-Event: Innerhalb der Überwachungszeit <i>tHeartBeatConsumerTime</i> wurde keine "Heartbeat"-Nachricht vom zu überwachenden Busteilnehmer empfangen.
		-12 Die eingestellte Knotenadresse (<i>byNodeAddr</i>) ist ungültig.
		-120 Der Überwachungsmechanismus wurde nicht im CanDSx-Treiber initialisiert. • Initialisieren Sie den Überwachungsmechanismus mit Hilfe der Funktion L_CanDSxOpenHeartBeat .
		-121 Die eingestellte Treibernummer (<i>wDrvNr</i>) ist ungültig.

Erforderliche Einstellungen bei der zu überwachenden PLC

(Gültig für 9300 Servo PLC/Drive PLC ab V6.2)

Bei der zu überwachenden PLC sind folgende Einstellungen erforderlich, damit die PLC die Funktion des "Heartbeat Producer" übernimmt:

1. Setzen Sie in der zu überwachenden PLC die Codestelle C0352 auf den Wert "3", um die entsprechende PLC als "Slave und Heartbeat Producer" zu konfigurieren:

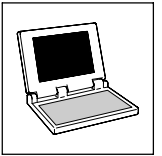
Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0352	CAN mst	0		Systembus: Master/Slave-Konfiguration der PLC
			0	
			1	
			2	
			3	
			4	

2. Stellen Sie in der zu überwachenden PLC über C0381 den Zeitintervall für das Senden der "Heartbeat"-Nachricht ein, je nach Buslast empfehlen wir eine Einstellung im Bereich von 200 ... 2000 ms:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0381	HeartProTime	0		Systembus: Heartbeat (Slave): HeartbeatProducerTime
			0 {1 ms} 65535	

3. Setzen Sie in der zu überwachenden PLC die Codestelle C0003 auf den Wert "1", um die vorgenommenen Änderungen netzausfallsicher zu speichern.
4. Setzen Sie in der zu überwachenden PLC die Codestelle C0358 auf den Wert "1", um einen CAN-Reset-Node durchzuführen.
– Alternativ kann der CAN-Reset-Node auch durch Netzschalten durchgeführt werden.

Nach der Ausführung des CAN-Reset-Node sendet die zu überwachende PLC als "Heartbeat Producer" kontinuierlich das "Heartbeat"-Telegramm im unter C0381 eingestellten Zeitintervall über den Systembus. In der überwachenden PLC kann daraufhin mit Hilfe dieses FBs die "Heartbeat"-Überwachung aktiviert werden.



3.7 L_CanDSxCloseHeartBeat - "Heartbeat" deaktivieren

Funktion

Mit dieser Funktion wird der Überwachungsmechanismus "Heartbeat" des CanDSx-Treibers wieder deaktiviert.

- Der Übergabeparameter *bClose* muß für die Deaktivierung auf TRUE gesetzt sein.

Deklaration	
BOOL	L_CanDSxCloseHeartBeat (bClose);

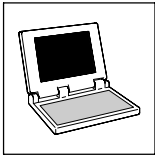
Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bClose	Bool	Überwachungsmechanismus "Heartbeat" deaktivieren.
		TRUE Der Überwachungsmechanismus "Heartbeat" des CanDSx-Treiber im Betriebssystem wird deaktiviert.

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Der Überwachungsmechanismus "Heartbeat" des CanDSx-Treibers wurde deaktiviert.
		FALSE Der Überwachungsmechanismus "Heartbeat" des CanDSx-Treibers wurde nicht deaktiviert.
		Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bClose</i> = TRUE.

Beispiel

Aufruf der Funktion in ST:

```
bReturnCloseHeartBeat := L_CanDSxCloseHeartBeat (bClose:=TRUE);
```



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxOpenNodeGuarding - "Node Guarding" initialisieren

3.8 L_CanDSxOpenNodeGuarding - "Node Guarding" initialisieren

Funktion

Im Kommunikationsprofil CANopen (CiA DS301, Version 4.01) sind zwei optionale Überwachungsmechanismen zur Sicherstellung der Funktionalität von Systembus-Teilnehmern spezifiziert, "Heartbeat" und "NodeGuarding".

Mit dieser Funktion wird der Überwachungsmechanismus "Node Guarding" des CanDSx-Treibers initialisiert.

- Der Übergabeparameter *bOpen* muß für die Initialisierung auf TRUE gesetzt sein.
- Die eigentliche Überwachung erfolgt mit Hilfe des FBs **L_CanDSxNodeGuarding**. (3-11)
- Mit der Funktion **L_CanDSxCloseNodeGuarding** können Sie den Überwachungsmechanismus "Node Guarding" wieder deaktivieren. (3-14)



Hinweis!

Die gleichzeitige Anwendung beider Überwachungsmechanismen ist nicht erlaubt!

Ist beim zu überwachenden Busteilnehmer eine Sendezykluszeit für die "Heartbeat"-Nachricht ungleich Null konfiguriert, so wird der "Heartbeat"-Mechanismus vorrangig vor dem "Node Guarding"-Mechanismus angewendet.

Deklaration

```
BOOL L_CanDSxOpenNodeGuarding (bOpen);
```

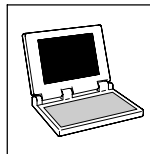
Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bOpen	Bool	Überwachungsmechanismus "Node Guarding" initialisieren.
		TRUE Der Überwachungsmechanismus "Node Guarding" des CanDSx-Treibers wird initialisiert.

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Überwachungsmechanismus "Node Guarding" wurde initialisiert.
		FALSE A Überwachungsmechanismus "Node Guarding" wurde nicht initialisiert. – Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bOpen</i> = TRUE. oder B Die PLC ist nicht als "Master mit Node Guarding" konfiguriert. – Abhilfe: Codestelle C352 auf den Wert "2" einstellen, um die PLC als "Master mit Node Guarding" zu konfigurieren.

Beispiel

Aufruf der Funktion in ST:

```
bReturnOpenNodeGuarding := L_CanDSxOpenNodeGuarding (bOpen:=TRUE);
```

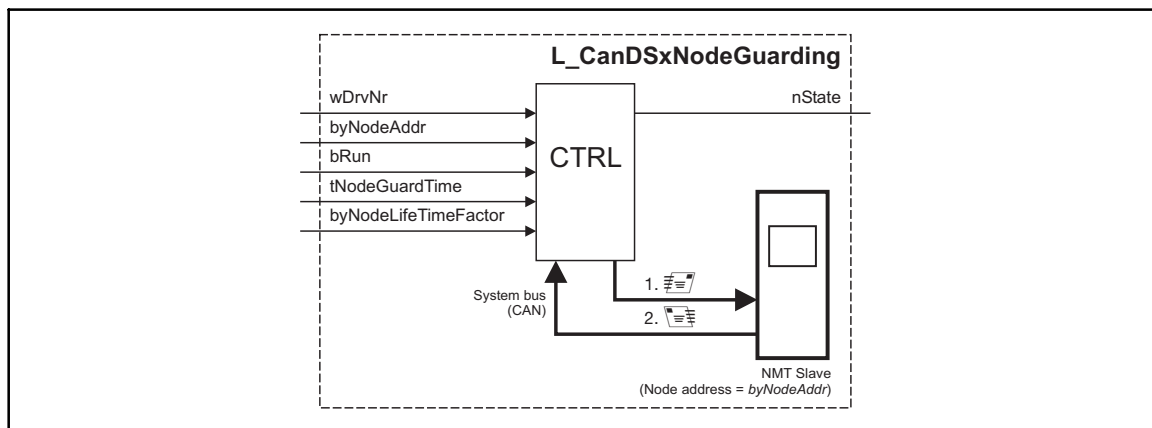


3.9 L_CanDSxNodeGuarding - "Node Guarding" ausführen

Funktionsblock

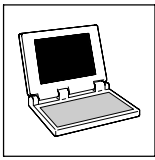
Mit diesem FB können Sie die CAN-Verbindung zwischen der PLC und anderen Systembus-Teilnehmern mit Hilfe des sogenannten "Node Guarding"-Mechanismus zyklisch überwachen.

- Dieser Überwachungsmechanismus ist zuvor mit Hilfe der Funktion **L_CanDSxOpenNodeGuarding** im CanDSx-Treiber zu initialisieren. (3-10)



FB-Aufruf in:	☐ Zyklischer Task (PLC_PRG)	☒ Zeitgesteuerter Task (INTERVAL)	☐ Ereignisgesteuerter Task (EVENT)	☐ Interrupt-Task
----------------------	--	---	---	---

Eingänge	Datentyp	Info/Einstellmöglichkeiten
wDrv	Word	Treiber Nummer für die zu verwendene CAN-Schnittstelle der PLC 10 On board-Systembus (CAN)
byNodeAddr	Byte	Knotenadresse des Busteilnehmers, der überwacht werden soll. 1 ... 128 Knotenadresse
bRun	Bool	"Node Guarding" aktivieren. TRUE Überwachung ist aktiviert. Der FB sendet nun im über <i>tNodeGuardTime</i> eingestellten Sendezyklus ein "Remote-Transmission-Request"-Telegramm an den Busteilnehmer mit der Knotenadresse <i>byNodeAddr</i> und wartet auf eine entsprechende Antwort. Erfolgt diese nicht innerhalb der Überwachungszeit "NodeLifeTime", so gibt der FB einen entsprechenden Status am Ausgang <i>nState</i> aus.
tNodeGuardTime	Time	Sendezyklus des Remote-Transmission-Request-Telegramm. - Zeitabstand, in dem die PLC an den zu überwachenden Busteilnehmer eine Status-Anfrage sendet (Zyklisches Polling). - Diese Einstellung muß mit der in der zu überwachenden PLC eingestellten "GuardTime" (C0382) korrespondieren.
byNodeLifeTimeFactor	Byte	Faktor für die sogenannte "NodeLifeTime". $\text{"NodeLifeTime"} = \text{byNodeLifeTimeFactor} \cdot \text{tNodeGuardTime}$ - Antwortet der zu überwachende Busteilnehmer auf die Status-Abfrage der PLC nicht innerhalb der "NodeLifeTime", so wird ein "Node Guarding"-Event ausgelöst (<i>nState</i> = -10). - Diese Einstellung muß mit dem in der zu überwachenden PLC eingestellten "NodeLifeTimeFactor" (C383) korrespondieren.



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxNodeGuarding - "Node Guarding" ausführen

Ausgänge	Datentyp	Info/Einstellmöglichkeiten
nState	Integer	Status
		300 FB ist deaktiviert (<i>bRun</i> = FALSE).
		127 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Pre-Operational</i> .
		5 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Operational</i> .
		4 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Stopped</i> .
		0 Zu überwachender Busteilnehmer befindet sich im CAN-Zustand <i>Boot-Up</i> oder der FB wird nicht aufgerufen.
		-5 Die Überwachungszeit <i>tNodeGuardTime</i> oder der Faktor <i>byNodeLifeTimeFactor</i> ist auf den Wert "0" eingestellt.
		-9 Antwort vom zu überwachenden Busteilnehmer ist ungültig.
		-10 "Node Guarding"-Event: Innerhalb der Überwachungszeit "NodeLifeTime" wurde keine Statusantwort vom zu überwachenden Busteilnehmer empfangen.
		-12 Die eingestellte Knotenadresse (<i>byNodeAddr</i>) ist ungültig.
		-120 Der Überwachungsmechanismus wurde nicht im CanDSx-Treiber initialisiert. • Initialisieren Sie den Überwachungsmechanismus mit Hilfe der Funktion L_CanDSxOpenNodeGuarding .
		-121 Die eingestellte Treibernummer (<i>wDrvNr</i>) ist ungültig.

Erforderliche Einstellungen bei der zu überwachenden PLC

(Gültig für 9300 Servo PLC/Drive PLC ab V6.2 sowie ECSxA ab V7.0)

Bei der zu überwachenden PLC sind folgende Einstellungen erforderlich, damit die PLC die Funktion des "Node Guarding Slave" übernimmt:

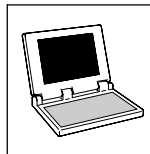
1. Setzen Sie in der zu überwachenden PLC die Codestelle C0352 auf den Wert "4", um die entsprechende PLC als "Slave mit Node Guarding" zu konfigurieren:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0352	CAN mst	0		Systembus: Master/Slave-Konfiguration der PLC
			0	
			1	
			2	
			3	
			4	
			Slave (Boot-Up nicht aktiv)	
			Master (Boot-Up aktiv)	
			Master mit Node Guarding (kein SyncReceived mehr möglich)	
			Slave und Heartbeat Producer	
			Slave mit Node Guarding	

2. Stellen Sie in der zu überwachenden PLC über C0382 den Zeitintervall der Statusanfrage vom Master ein. Dieser Wert muß mit der Einstellung am FB-Eingang *tNodeGuardTime* in der überwachenden PLC korrespondieren:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0382	GuardTime	0		Systembus: Node Guarding (Slave): NodeGuardTime
			0 {1 ms} 65535	

3. Stellen Sie an der überwachenden PLC den Node-Guarding-Master C0352/0 = 2 ein.



4. Stellen Sie in der zu überwachenden PLC über C0383 den Faktor für die Überwachungszeit "NodeLifeTime" ein. Dieser Wert muß mit der Einstellung am FB-Eingang *byNodeLifeTimeFactor* in der überwachenden PLC korrespondieren:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0383	LifeTimeFact.	0		Systembus: Node Guarding (Slave): NodeLifeTimeFactor
			0 {1} 255	



Tip!

Aus den Einstellungen für die "NodeGuardTime" (C0382) und dem "NodeLifeTimeFactor" (C0383) in der zu überwachenden PLC ergibt sich die sogenannte "NodeLifeTime":

$$\text{NodeLifeTime} = \text{NodeGuardTime (C0382)} \cdot \text{NodeLifeTimeFactor (C0383)}$$

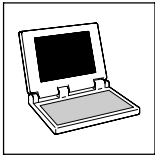
- Empfängt die zu überwachende PLC nicht innerhalb dieser "NodeLifeTime" eine Statusanfrage von der überwachenden PLC, so wird in der zu überwachenden PLC ein sogenanntes "Life Guarding"-Event ausgelöst.

5. Stellen Sie in der zu überwachenden PLC über C0384 die gewünschte Reaktion ein, mit der die zu überwachende PLC auf ein "Life Guarding"-Event reagieren soll:

Code	LCD	Einstellmöglichkeiten		Info
		Lenze	Auswahl	
C0384	Err NodeGuard	3		Systembus: Node Guarding (Slave): Reaktion beim Auftreten eines "Life Guard"-Event.
			0 TRIP 1 Meldung 2 Warnung 3 Aus 4 Fail-QSP	

6. Setzen Sie in der zu überwachenden PLC die Codestelle C0003 auf den Wert "1", um die vorgenommenen Änderungen netzausfallsicher zu speichern.
7. Setzen Sie in der zu überwachenden PLC die Codestelle C0358 auf den Wert "1", um einen CAN-Reset-Node durchzuführen.
- Alternativ kann der CAN-Reset-Node auch durch Netzschalten durchgeführt werden.

Nach der Ausführung des CAN-Reset-Node ist die zu überwachende PLC als "Node Guarding Slave" konfiguriert und im "Node Guarding Master" kann die Überwachung mit Hilfe dieses FBs aktiviert werden.



Funktionsbibliothek LenzeCanDSxDrv.lib

Funktionen/Funktionsblöcke

L_CanDSxCloseNodeGuarding - "Node Guarding" deaktivieren

3.10 L_CanDSxCloseNodeGuarding - "Node Guarding" deaktivieren

Funktion

Mit dieser Funktion wird der Überwachungsmechanismus "Node Guarding" des CanDSx-Treibers wieder deaktiviert.

- Der Übergabeparameter *bClose* muß für die Deaktivierung auf TRUE gesetzt sein.

Deklaration	
BOOL	L_CanDSxCloseNodeGuarding (bClose);

Übergabeparameter	Datentyp	Info/Einstellmöglichkeiten
bClose	Bool	Überwachungsmechanismus "Node Guarding" deaktivieren.
		TRUE Der Überwachungsmechanismus "Node Guarding" des CanDSx-Treiber wird deaktiviert.

Rückgabewert	Datentyp	Wert/Bedeutung
	Bool	Status
		TRUE Überwachungsmechanismus "Node Guarding" des CanDSx-Treibers wurde deaktiviert.
		FALSE Überwachungsmechanismus "Node Guarding" des CanDSx-Treibers wurde nicht deaktiviert. • Abhilfe: Funktionsaufruf mit Übergabeparameter <i>bClose</i> = TRUE.

Beispiel

Aufruf der Funktion in ST:

```
bReturnCloseNodeGuarding := L_CanDSxCloseNodeGuarding (bClose:=TRUE);
```