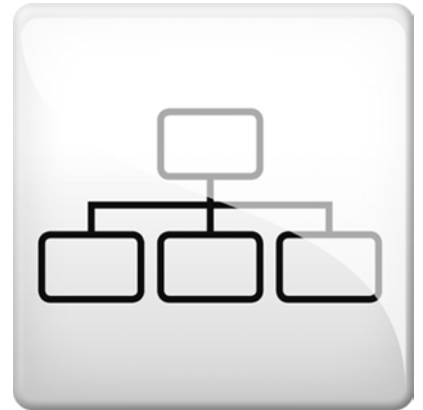


Application Template



Lenze-Standard

Software-Handbuch

DE



13558102

Lenze

1	Über diese Dokumentation	4
1.1	Dokumenthistorie	6
1.2	Verwendete Konventionen	7
1.3	Verwendete Begriffe	8
1.4	Verwendete Hinweise	9
2	Sicherheitshinweise	10
3	Systemvoraussetzungen	12
4	Strukturierte Programmierung mit dem Application Template	13
4.1	Vorgehensweise	14
4.2	Neues »PLC Designer«-Projekt erstellen – Application Template öffnen	15
4.3	Maschinenmodul anlegen	16
4.4	Maschinenmodul als Vorlage speichern	19
4.5	Maschinenmodul kopieren	20
4.6	Maschinenmodulinstantz erstellen	22
4.7	Maschinenmodul umbenennen	23
4.8	Maschinenmodul im Machine Module Tree (MMT) einbinden	24
4.9	Maschinenmodul-Referenzen löschen	27
4.10	Maschinenmodul löschen	28
4.11	Visualisierung einbinden und verbinden	29
4.12	FAST Technologiemodule einfügen	31
4.13	Achsen verbinden	34
4.14	Kommunikationskanal einrichten (ACD Slave Access)	36
4.15	Operation-Modes verwenden	37
4.16	Kommunikationskanal verwenden	39
4.17	Fehlermeldungen anlegen und bearbeiten	40
4.17.1	Fehlerliste exportieren (XML-Datei)	42
4.17.2	Fehlerliste importieren (XML-Datei)	43
4.17.3	Modulkopplung verwenden	45
5	Struktur des Application Template	47
5.1	A10_MachineModuleTree	48
5.2	A12_Configuration	50
5.3	A20_Visualisation	51
5.3.1	L_EATP_FAST_VisErrorList	52
5.3.2	L_EATP_FAST_VisModuleList	53
5.3.3	L_EATP_FAST_VisModuleDetail	54
5.4	A55_VarLists	55
5.5	A60_MotionObjects	55
5.6	A70_MachineModuleSources	56
5.6.1	Strukturen	57
5.6.2	User POU's	58
5.6.3	Visualisierungen	58
5.6.4	BF01_BasicFunction	61
5.6.4.1	L_EATP_FAST_OpModeControl	62
5.6.4.2	L_EATP_FAST_OpModeAccess	63
5.6.4.3	L_EATP_FAST_UserCoupling	66
5.6.4.4	L_EATP_FAST_ErrorAccess	67
5.6.5	BF02_SetErrors	69
5.6.6	I01_ReadInputs	71

5.6.7	MFB03_Processing	72
5.6.7.1	Mode-bezogene Aktionen	73
5.6.7.2	Fehler-Aktion	73
5.6.7.3	Zyklische Aktion	73
5.6.7.4	Ablauf von Aktionen	74
5.6.8	O01_WriteOutputs	75
5.6.9	MTC01_TaskMid / MTC02_TaskFree	76
Index		77
Ihre Meinung ist uns wichtig		79

1 Über diese Dokumentation

Diese Dokumentation beschreibt, wie mit Hilfe des Lenze FAST Application Template ein strukturiertes Programm von der Maschinenidee bis zum lauffähigen Programm im »PLC Designer« erstellt werden kann.


Es wird gezeigt wie einzelne Maschinenmodule erzeugt und miteinander im Machine Module Tree (MMT) verbunden werden. Zur Umsetzung der Maschinenapplikation werden FAST Technologiemodule eingebunden.

Diese Dokumentation ordnet sich in die Handbuchsammlung "Controller-based Automation" ein. Diese besteht aus folgenden Dokumentationen:


Dokumentationstyp	Thema
Produktkatalog	Controller-based Automation (Systemübersicht, Beispieltopologien) Lenze-Controller (Produktinformationen, Technische Daten)
Systemhandbücher	Visualisierung (Systemübersicht/Beispieltopologien)
Kommunikationshandbücher Online-Hilfen	Bussysteme <ul style="list-style-type: none">• Controller-based Automation EtherCAT®• Controller-based Automation CANopen®• Controller-based Automation PROFIBUS®• Controller-based Automation PROFINET®
Referenzhandbücher Online-Hilfen	Lenze-Controller: <ul style="list-style-type: none">• Controller 3200 C• Controller c300• Controller p300• Controller p500
Software-Handbücher Online-Hilfen	Lenze Engineering Tools: <ul style="list-style-type: none">• »PLC Designer« (Programmierung)• »Engineer« (Parametrierung, Konfigurierung, Diagnose)• »VisiWinNET® Smart« (Visualisierung)• »Backup & Restore« (Datensicherung, Wiederherstellung, Aktualisierung)

Weitere Technische Dokumentationen zu Lenze-Produkten

Weitere Informationen zu Lenze-Produkten, die in Verbindung mit der Controller-based Automation verwendbar sind, finden Sie in folgenden Dokumentationen:

Planung / Projektierung / Technische Daten	
<input type="checkbox"/>	Produktkataloge <ul style="list-style-type: none"> • Controller-based Automation • Controller • Inverter Drives/Servo Drives
Montage und Verdrahtung	
	Montageanleitungen <ul style="list-style-type: none"> • Controller • Kommunikationskarten (MC-xxx) • I/O-System 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Kommunikationsmodule
<input type="checkbox"/>	Gerätehandbücher <ul style="list-style-type: none"> • Inverter Drives/Servo Drives
Parametrierung / Konfigurierung / Inbetriebnahme	
<input type="checkbox"/>	Online-Hilfe / Referenzhandbücher <ul style="list-style-type: none"> • Controller • Inverter Drives/Servo Drives • I/O-System 1000 (EPM-Sxxx)
<input type="checkbox"/>	Online-Hilfe / Kommunikationshandbücher <ul style="list-style-type: none"> • Bussysteme • Kommunikationsmodule
Beispielapplikationen und Vorlagen	
<input type="checkbox"/>	Online-Hilfe / Software- und Referenzhandbücher <ul style="list-style-type: none"> • Application Sample i700 • Application Samples 8400/9400 • FAST Application Template • FAST Technologiemodule

Symbole:

-  Gedruckte Dokumentation
- ☐ PDF-Datei / Online-Hilfe im Lenze Engineering Tool



Tipp!

Aktuelle Dokumentationen und Software-Updates zu Lenze-Produkten finden Sie im Download-Bereich unter:

www.lenze.com

1 Über diese Dokumentation

1.1 Dokumenthistorie

Zielgruppe

Diese Dokumentation richtet sich an alle Personen, die ein Lenze-Automationssystem auf Basis der Application Software Lenze FAST programmieren und in Betrieb nehmen.

Screenshots/Anwendungsbeispiele

Alle Screenshots in dieser Dokumentation sind Anwendungsbeispiele. Je nach Firmware-Version der eingesetzten Lenze-Geräte und Software-Version der installierten Engineering Tools (z. B. »PLC Designer«) können die Screenshots in dieser Dokumentation von der Bildschirm-Darstellung abweichen.

Informationen zur Gültigkeit

Die Informationen in dieser Dokumentation sind gültig für folgende Lenze-Software:

Software	ab Software-Version
»PLC Designer« (Bibliothek L_EATP_ApplicationTemplate)	3.13



1.1 Dokumenthistorie

Version			Beschreibung
4.0	10/2018	TD29	Allgemeine Überarbeitung
3.0	06/2016	TD17	Aktualisiert zu »PLC Designer« V3.13 • Allgemeine Überarbeitung
2.0	12/2015	TD17	Aktualisiert zu »PLC Designer« V3.12 • Allgemeine Überarbeitung
1.6	05/2015	TD17	Aktualisiert zu »PLC Designer« V3.10
1.5	12/2014	TD11	Aktualisiert zu »PLC Designer« V3.9
1.4	10/2013	TD11	Aktualisiert zu »PLC Designer« V3.6 • Optimierungen aus Usability-Tests (User group) • Systemfehlermeldungen ergänzt. • Struktur L_EATP_MMD_Base ergänzt. • Befehl "Create MM Instance" ergänzt.
1.3	04/2013	TD11	Aktualisiert zu »PLC Designer« V3.5 • Softwareupdate von "Application Template Counter"/"Application Template". • Neu: Anwendungsbeispiel "Fliegende Säge".
1.2	11/2012	TD11	Aktualisiert zu »PLC Designer« V3.3.2 • Neu: Beispielprojekt "Application Template Counter" (Lenze-Standard)
1.1	07/2012	TD11	Aktualisiert • Allgemeine Korrektur • Anpassung an VISU-Layout gemäß Lenze Programmier-Styleguide für Funktionsbausteine.
1.0	04/2012	TD11	Erstausgabe

1 Über diese Dokumentation

1.2 Verwendete Konventionen

Diese Dokumentation verwendet folgende Konventionen zur Unterscheidung verschiedener Arten von Information:

Informationsart	Auszeichnung	Beispiele/Hinweise
Zahlenschreibweise		
Dezimaltrennzeichen	Punkt	Es wird generell der Dezimalpunkt verwendet. Zum Beispiel: 1234.56
Textauszeichnung		
Versionsinfo	Textfarbe blau	Alle Informationen, die nur für oder ab einem bestimmten Softwarestand des Antriebsreglers gelten, sind in dieser Dokumentation entsprechend gekennzeichnet. Beispiel: Diese Funktionserweiterung ist ab dem Softwarestand V3.0 verfügbar!
Programmname	» «	Lenze »PLC Designer«...
Fensterbereich	kursiv	Das <i>Meldungsfenster...</i> / Das Dialogfeld <i>Optionen...</i>
Variablenbezeichner		Durch Setzen von <i>bEnable</i> auf TRUE...
Steuerelement	fett	Die Schaltfläche OK... / Der Befehl Kopieren... / Die Registerkarte Eigenschaften... / Das Eingabefeld Name...
Folge von Menübefehlen		Sind zum Ausführen einer Funktion mehrere Befehle nacheinander erforderlich, sind die einzelnen Befehle durch einen Pfeil voneinander getrennt: Wählen Sie den Befehl Datei→Öffnen , um...
Tastaturbefehl	<fett>	Mit <F1> rufen Sie die Onlinehilfe auf.
		Ist für einen Befehl eine Tastenkombination erforderlich, ist zwischen den Tastenbezeichnern ein "+" gesetzt: Mit <Shift>+<ESC>...
Programmcode	Courier	IF var1 < var2 THEN a = a + 1 END IF
Schlüsselwort	Courier fett	
Hyperlink	<u>unterstrichen</u>	Optisch hervorgehobener Verweis auf ein anderes Thema. Wird in dieser Online-Dokumentation per Mausklick aktiviert.
Symbole		
Seitenverweis	( 7)	Verweis auf weiterführenden Informationen: Seitenzahl in PDF-Datei.
Schrittweise Anleitung		Schrittweise Anleitungen sind durch ein Piktogramm gekennzeichnet.

1.3

Verwendete Begriffe

Begriff	Bedeutung
Controller	Der Controller ist die zentrale Komponente des Lenze-Automationssystems, das mit Hilfe des Betriebssystems die Bewegungsabläufe steuert. Der Controller kommuniziert über den Feldbus mit den Feldgeräten (Inverter).
Engineering PC	Mit dem Engineering PC und den darauf installierten Engineering Tools konfigurieren und parametrieren Sie das System. Der Engineering PC kommuniziert über Ethernet mit dem Controller.
Inverter	Oberbegriff für Lenze-Frequenzumrichter, Servo-Umrichter
MFB	Maschinen-Funktionsbaustein Ein Maschinen-Funktionsbaustein enthält die Funktionen eines Maschinenmoduls (MM) im »PLC Designer«.
MM	Maschinenmodul Ein Maschinenmodul bildet eine Teilfunktionen der Maschine/Anlage im »PLC Designer« ab. Maschinenmodule werden über die entsprechenden Maschinen-Funktionsbausteine (MFB) miteinander verschaltet sind.
MMT	Maschine Module Tree Der Maschinenmodulbaum bildet die Struktur des Automationssystems ab. Die einzelnen Maschinenmodule (MM) werden hierarchisch in der Baumstruktur gegliedert.
PLC	Programmable Logic Controller (deutsche Bezeichnung: SPS - Speicherprogrammierbare Steuerung)

1.4

Verwendete Hinweise

Um auf Gefahren und wichtige Informationen hinzuweisen, werden in dieser Dokumentation folgende Signalwörter und Symbole verwendet:

Sicherheitshinweise

Aufbau der Sicherheitshinweise:

**Piktogramm und Signalwort!**

(kennzeichnen die Art und die Schwere der Gefahr)

Hinweistext

(beschreibt die Gefahr und gibt Hinweise, wie sie vermieden werden kann)

Piktogramm	Signalwort	Bedeutung
	Gefahr!	Gefahr von Personenschäden durch gefährliche elektrische Spannung Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Gefahr!	Gefahr von Personenschäden durch eine allgemeine Gefahrenquelle Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Stop!	Gefahr von Sachschäden Hinweis auf eine mögliche Gefahr, die Sachschäden zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.

Anwendungshinweise

Piktogramm	Signalwort	Bedeutung
	Hinweis!	Wichtiger Hinweis für die störungsfreie Funktion
	Tipp!	Nützlicher Tipp für die einfache Handhabung
		Verweis auf andere Dokumentation

2 Sicherheitshinweise

Beachten Sie die Sicherheitshinweise in dieser Dokumentation, wenn Sie ein Automationssystem oder eine Anlage mit einem Lenze-Controller in Betrieb nehmen möchten.



Die Gerätedokumentation enthält Sicherheitshinweise, die Sie beachten müssen!

Lesen Sie die mitgelieferten und zugehörigen Dokumentationen der jeweiligen Komponenten des Automationssystems sorgfältig durch, bevor Sie mit der Inbetriebnahme des Controllers und der angeschlossenen Geräte beginnen.



Gefahr!

Hohe elektrische Spannung

Personenschäden durch gefährliche elektrische Spannung

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

Die Spannungsversorgung ausschalten, bevor Arbeiten an den Komponenten des Automationssystems durchgeführt werden.

Nach dem Ausschalten der Spannungsversorgung spannungsführende Geräteteile und Leistungsanschlüsse nicht sofort berühren, weil Kondensatoren aufgeladen sein können.

Die entsprechenden Hinweisschilder auf dem Gerät beachten.



Gefahr!

Personenschäden

Verletzungsgefahr besteht durch ...

- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

- Anlagen mit eingebauten Invertern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).
- Während der Inbetriebnahme einen ausreichenden Sicherheitsabstand zum Motor oder den vom Motor angetriebenen Maschinenteilen einhalten.



Stop!

Beschädigung oder Zerstörung von Maschinenteilen

Beschädigung oder Zerstörung von Maschinenteilen besteht durch ...

- Kurzschluss oder statische Entladungen (ESD);
- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Schutzmaßnahmen

- Vor allen Arbeiten an den Komponenten des Automationssystems immer die Spannungsversorgung ausschalten.
- Elektronische Bauelemente und Kontakte nur berühren, wenn zuvor ESD-Maßnahmen getroffen wurden.
- Anlagen mit eingebauten Invertern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).

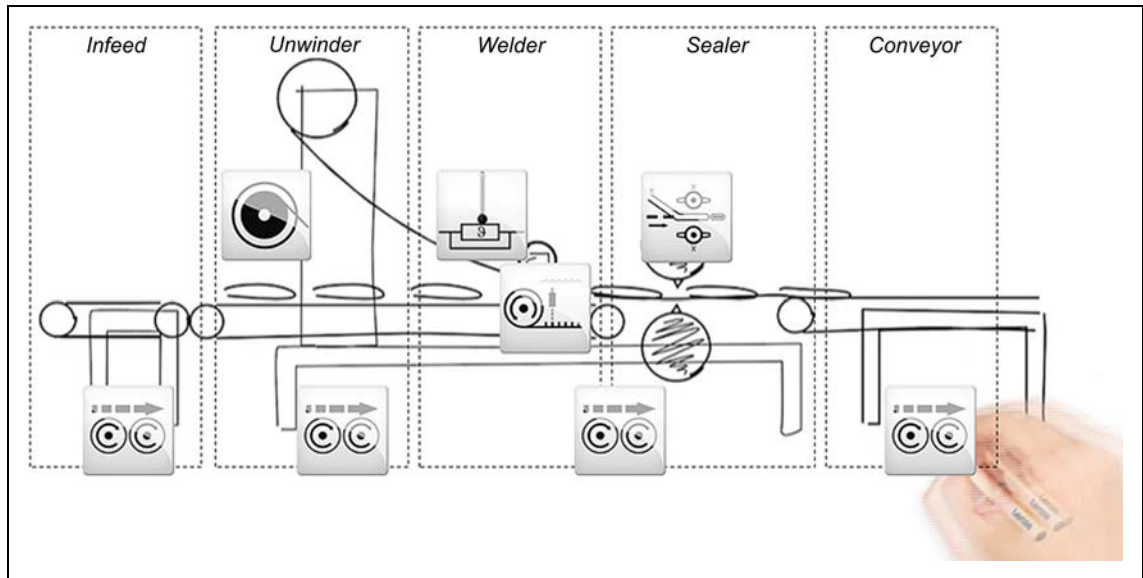
3 Systemvoraussetzungen

3 Systemvoraussetzungen

	Engineering PC	Lenze-Controller
Hardware	PC/Notebook	PLC (Logic) ab Firmware V3.13
Betriebssystem	<ul style="list-style-type: none">• Microsoft® Windows® XP Professional (32 Bit) ab SP3• Microsoft® Windows® 7 (32 und 64 Bit)	Microsoft® Windows® CE
Erforderliche Lenze-Software	»PLC Designer« ab V3.13 und installierte Bibliothek L_EATP_ApplicationTemplate (ab V3.13)	Runtime Software "Motion" Hierfür müssen die Projektinformationen aktualisiert werden: Befehl "Geräte aktualisieren".

4 Strukturierte Programmierung mit dem Application Template

Am Anfang steht eine Maschinenidee. Um aus dieser Idee eine Maschine zu erstellen, die mit dem Application Template umgesetzt werden soll, werden für die verschiedenen Antriebsaufgaben einzelne Maschinenmodule gebildet.



[4-1] Maschinenidee zerlegt in Maschinenmodule

Die Abbildung [\[4-1\]](#) zeigt eine Schlauchbeutelmaschine, die in mehrere Maschinenmodule zerlegt wurde.

Die Dokumentation bezieht sich im weiteren Verlauf nur auf eine beispielhafte Umsetzung der Module "Infeed" (Zuführen) und "Unwinder" (Abwickeln) im »PLC Designer«:

Infeed (Zuführen)		Unwinder (Abwickeln)	
	Das Maschinenmodul "Infeed" beinhaltet einen Antrieb der durch das FAST Technologiemodul "Electrical Shaft Position" abgebildet wird.		Das Maschinenmodul "Unwinder" beinhaltet einen Antrieb der ebenfalls durch das FAST Technologiemodul "Electrical Shaft Position" abgebildet wird. Das Technologiemodul "Winder Dancer-controlled" wird für dieses Beispiel nicht implementiert.

4.1 Vorgehensweise

Schritt	Tätigkeit
1.	Neues »PLC Designer«-Projekt erstellen – Application Template öffnen (15)
2.	Maschinenmodul anlegen (16) oder Maschinenmodul kopieren (20)
3.	Maschinenmodulinstanz erstellen (22)
4.	Maschinenmodul im Machine Module Tree (MMT) einbinden (24)
5.	Visualisierung einbinden und verbinden (29)
6.	FAST Technologiemodule einfügen (31)
7.	Achsen verbinden (34)
8.	Kommunikationskanal einrichten (ACD Slave Access) (36)
9.	Operation-Modes verwenden (37)
10.	Kommunikationskanal verwenden (39)
11.	Fehlermeldungen anlegen und bearbeiten (40)

Weitere Tätigkeiten

- ▶ [Maschinenmodul als Vorlage speichern \(19\)](#)
- ▶ [Maschinenmodul umbenennen \(23\)](#)
- ▶ [Maschinenmodul löschen \(28\)](#)
- ▶ [Maschinenmodul-Referenzen löschen \(27\)](#)

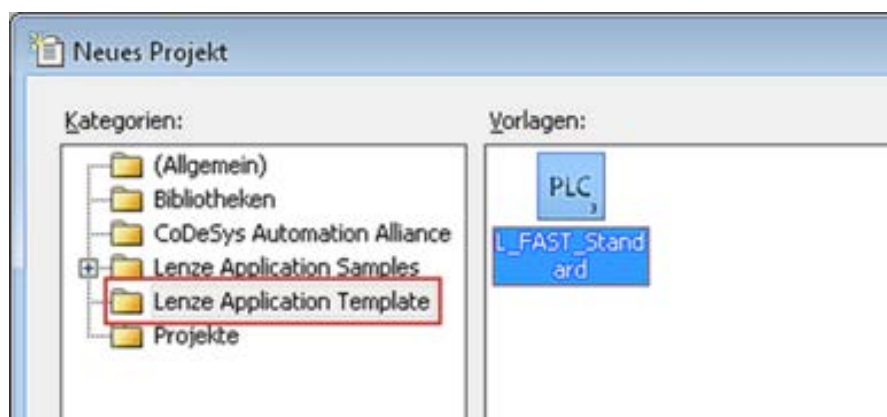
Die einzelnen Tätigkeiten sind in den folgenden Abschnitten ausführlich beschrieben.

4.2 Neues »PLC Designer«-Projekt erstellen – Application Template öffnen

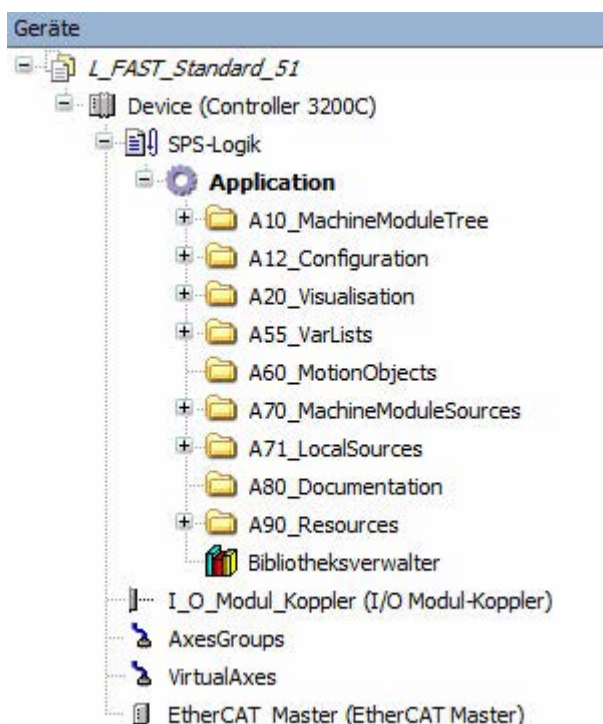


So gehen Sie vor:

1. Den »PLC Designer« starten.
2. Mit dem Menübefehl **Datei → Neues Projekt** ein neues Projekt erstellen.
Unter der Kategorie "Lenze Application Template" die Vorlage "L_FAST_Standard" auswählen:



3. Die Eingaben mit **OK** bestätigen.
Das Projekt wird mit dieser Gerätebaumstruktur geöffnet:



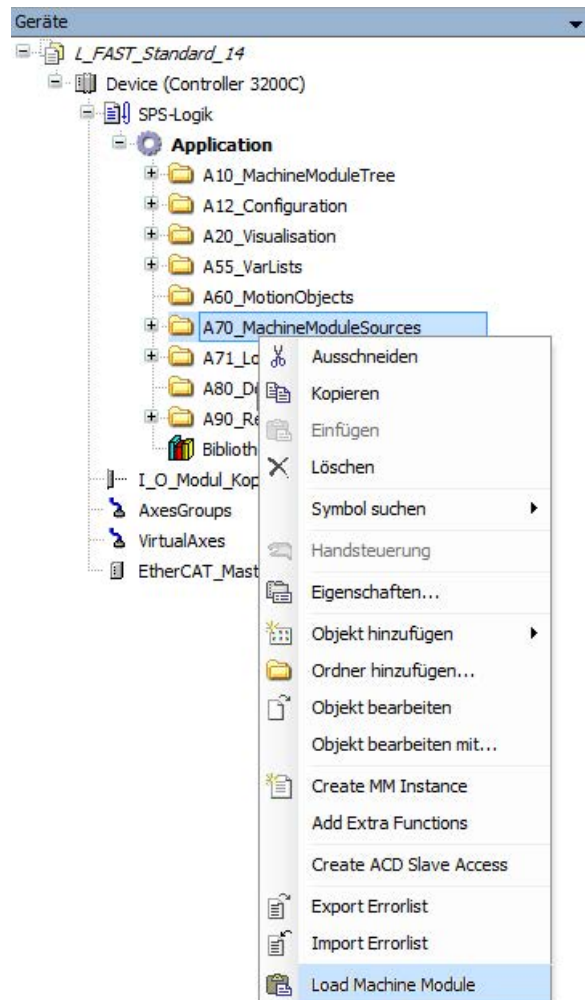
"Application" enthält die [Struktur des Application Template](#) (47).

4.3 Maschinenmodul anlegen

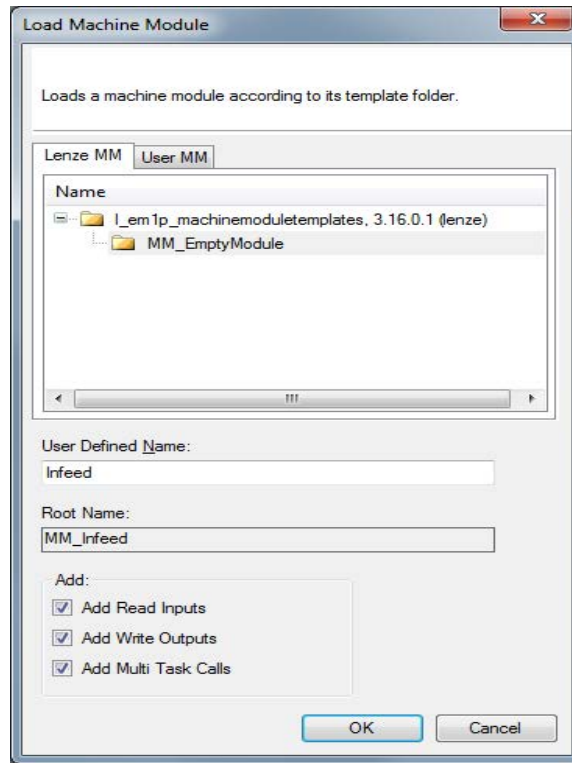


So gehen Sie vor:

1. Rechtsklick auf den Ordner **A70_MachineModuleSources** und den Menübefehl **Load Machine Module** wählen.



2. Im darauf eingeblendeten Dialogfenster "Load Machine Module" die Reiterkarte "Lenze MM" wählen, dort die Gruppe **I_em1p_machinemoduletemplates** öffnen und den Eintrag **MM_EmptyModule** markieren.



3. Im unteren Teil des Dialogs einen Namen für das Maschinenmodul vorgeben.
Der Modulname darf kein "MM_" oder Sonderzeichen enthalten. Erlaubt sind nur die Zeichen "A...Z", "a...z", "0...9".
4. Bei Bedarf weitere optionale Aktionen im Bereich "Add" durch Setzen von Kontrollfeldern zuweisen.
 - "Add Read Inputs" → [I01_ReadInputs](#) (71)
 - "Add Write Outputs" → [O01_WriteOutputs](#) (75)
 - "Add Multi Task Calls" → [MTC01_TaskMid / MTC02_TaskFree](#) (76)

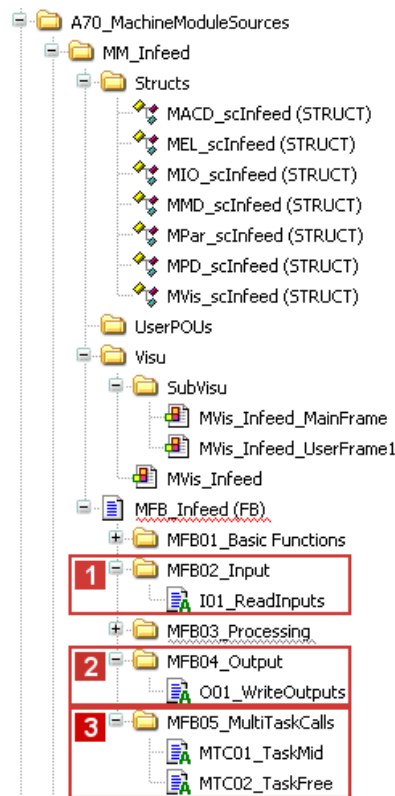
5. Die Eingaben mit **OK** bestätigen.

Unter **A70_MachineModuleSources** wird das Maschinenmodul mit dem Namen "MM_Infeed" eingefügt.

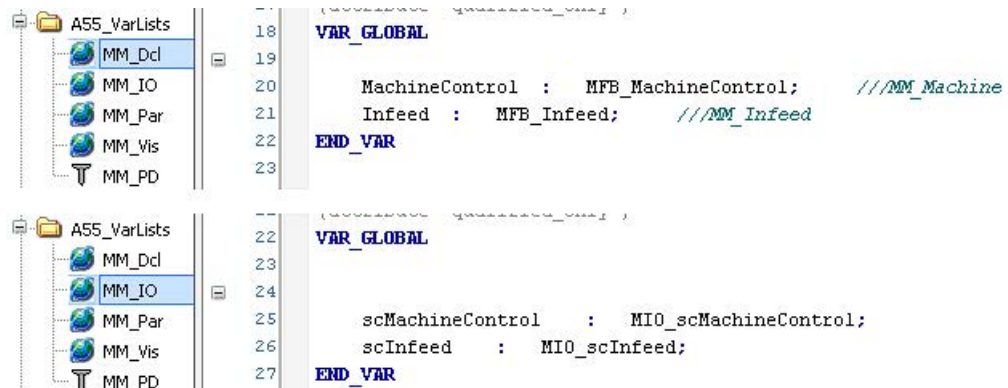
Die enthaltenen Strukturen, Visualisierungen und MFBs werden mit dem zuvor vergebenen Namen instanziiert.

Die gewählten (optionalen) Aktionen werden eingefügt.

- "Add Read Inputs" → **1** [I01_ReadInputs](#) (📄 71)
- "Add Write Outputs" → **2** [O01_WriteOutputs](#) (📄 75)
- "Add Multi Task Calls" → **3** [MTC01_TaskMid / MTC02_TaskFree](#) (📄 76)



In den vordefinierten globalen Variablenlisten **A55_VarLists** wird ebenfalls die entsprechende Instanz und die Variablenstruktur deklariert:

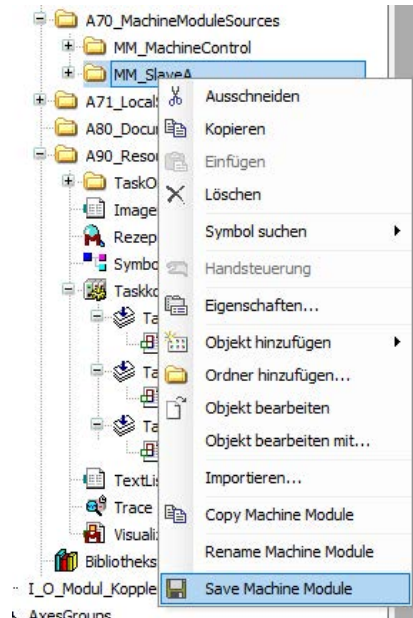


4.4 Maschinenmodul als Vorlage speichern

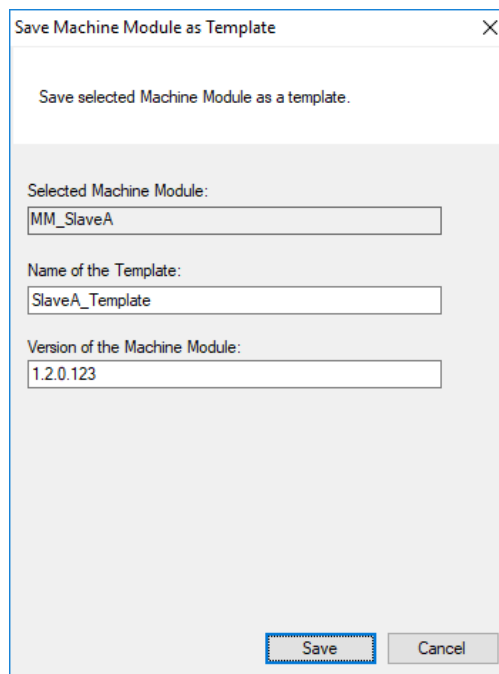


So gehen Sie vor:

1. Rechtsklick auf den Ordner des zu kopierenden Maschinenmoduls (im Beispiel "MM_SlaveA") und den Menübefehl **Save Machine Module** ausführen.



2. Im darauf erscheinenden Dialog "Save Machine Module as Template" einen Namen und eine Versionsnummer für die Maschinenmodul-Vorlage vergeben.



3. Abschließend den Dialog mit **OK** schließen.

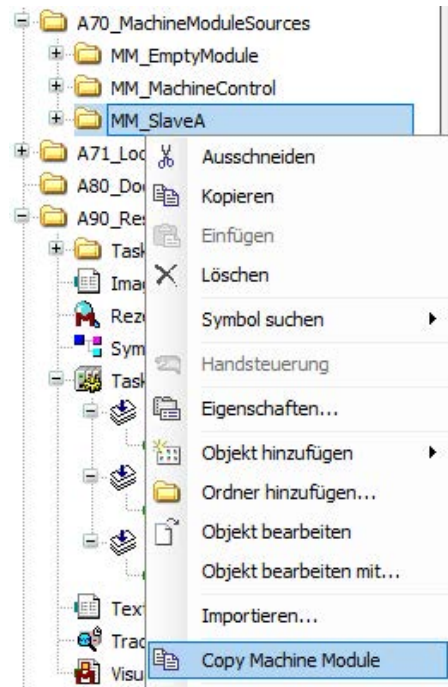
Das Maschinenmodul steht zukünftig beim Ausführen des Befehls **Maschinenmodul anlegen** auf der Reiterkarte "User MM" zur Verfügung. ▶ [Maschinenmodul anlegen](#) (16)

4.5 Maschinenmodul kopieren

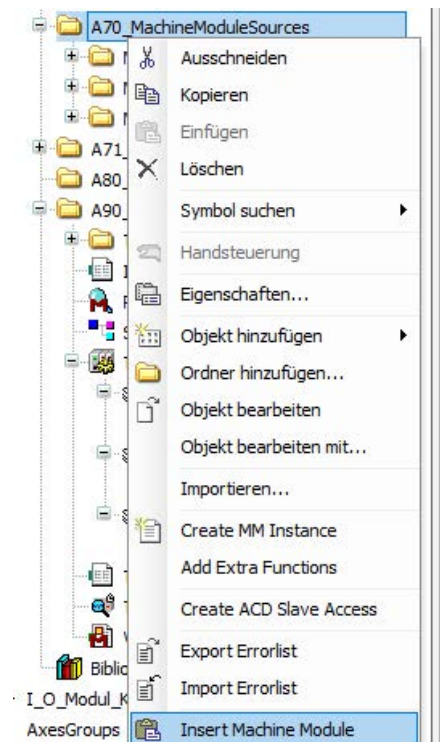


So gehen Sie vor:

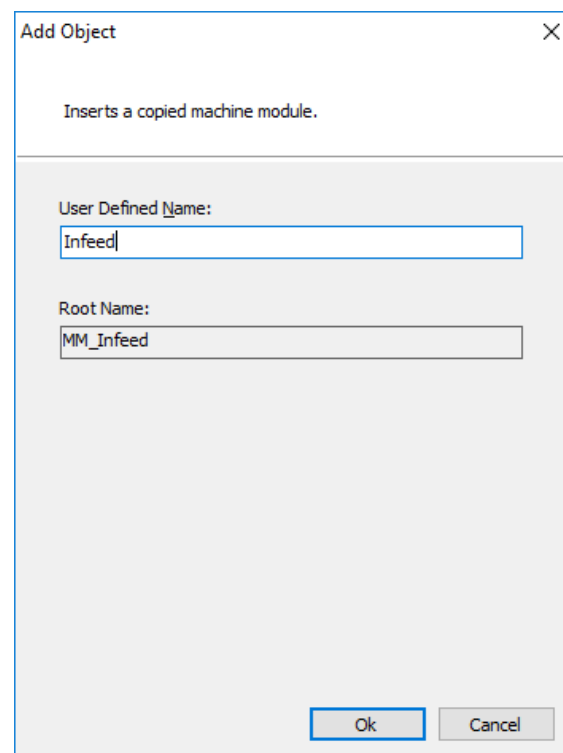
1. Rechtsklick auf den Ordner des zu kopierenden Maschinenmoduls (im Beispiel "MM_SlaveA") und mit dem Menübefehl **Copy Machine Module** das Modul als Vorlage in den Zwischenspeicher kopieren.



2. Rechtsklick auf den **Ordner A70_MachineModuleSources** und den Menübefehl **Insert Machine Module** ausführen.



3. Im darauf erscheinenden Dialog "Add Object" einen Namen für das neue Modul in das Feld **"User Defined Name"** eingeben, z. B. "Infeed".



4. Abschließend Dialog mit **OK** schließen.

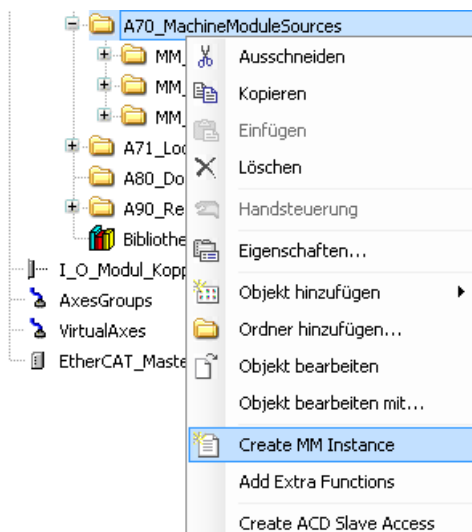
Das neue Maschinenmodul wird in den Ordner **A70_MachineModuleSources** eingefügt.

4.6 Maschinenmodulinstantz erstellen

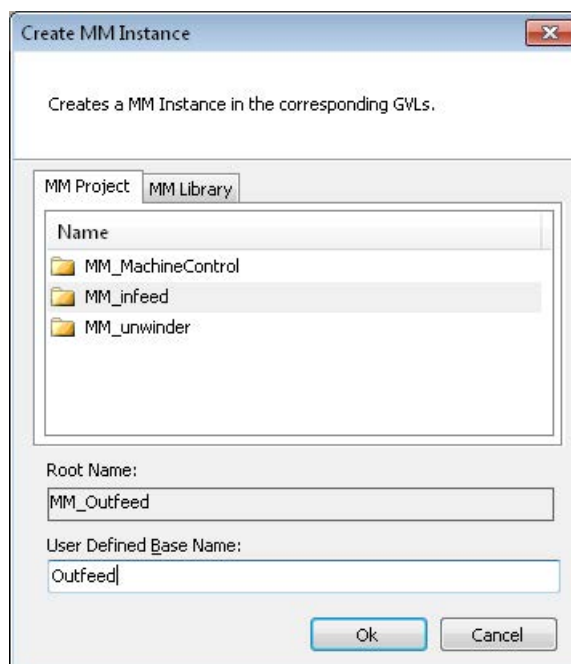


So gehen Sie vor:

1. Rechtsklick auf den Ordner **A70_MachineModuleSources** und den Menübefehl "**Create MM Instance**" ausführen.



2. Im erscheinenden Dialog das Maschinenmodul markieren, von dem eine Instanz erstellt werden soll.



3. Einen Instanznamen im Eingabefeld "User Defined Base Name" eingeben (im Beispiel "Outfeed").
4. Abschließend **OK** klicken.

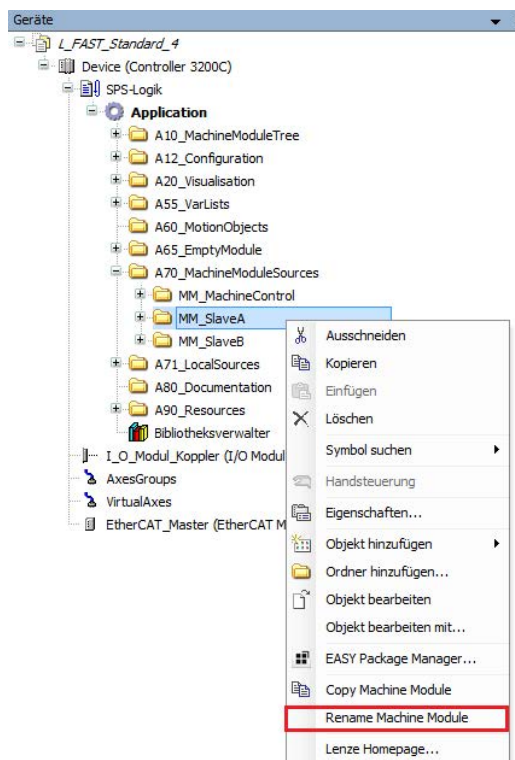
In allen globalen Variablenlisten im Ordner **A55_VarLists** wird eine Instanz des Maschinenmoduls und dessen Strukturen erstellt.

4.7 Maschinenmodul umbenennen

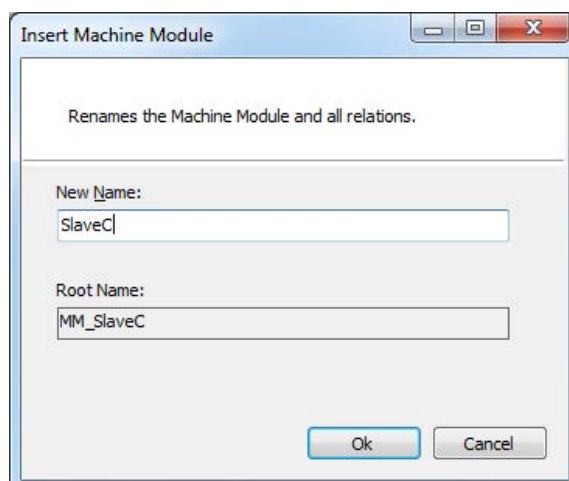


So gehen Sie vor:

1. Rechtsklick auf den Ordner des umzubennenden Maschinenmoduls und Menübefehl **Rename Machine Module** ausführen.



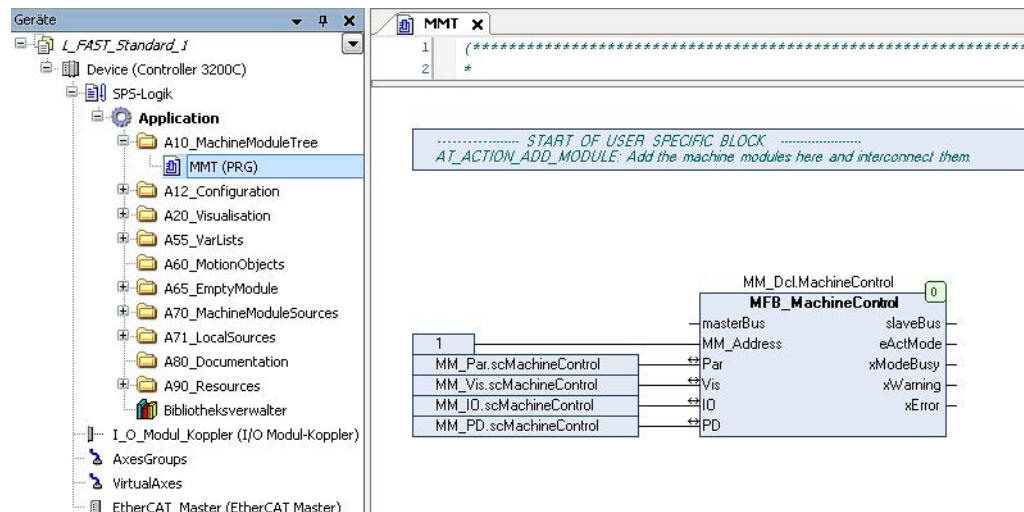
2. Im darauf erscheinenden Dialog "Insert Machine Module" einen neuen Namen für das neue Modul in das Feld **"New Name"** eingeben, z. B. "SlaveC".



3. Abschließend Dialog mit **OK** schließen.
Der Modulname wird umbenannt.

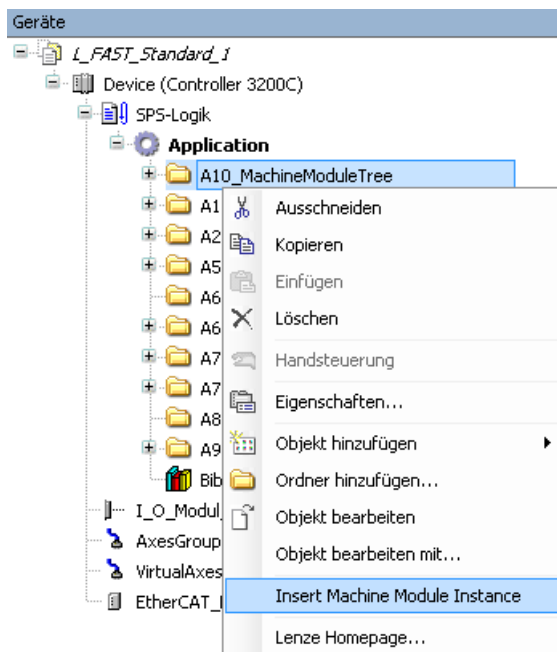
4.8 Maschinenmodul im Machine Module Tree (MMT) einbinden

Unter **A10_MachineModuleTree** im Programm **MMT (PRG)** werden die vorbereiteten Instanzen der Maschinenmodule aufgerufen. Die Vorlage beinhaltet bereits das Master-Modul "MachineControl", in dem übergeordnete Logikfunktionen programmiert werden.



So gehen Sie vor:

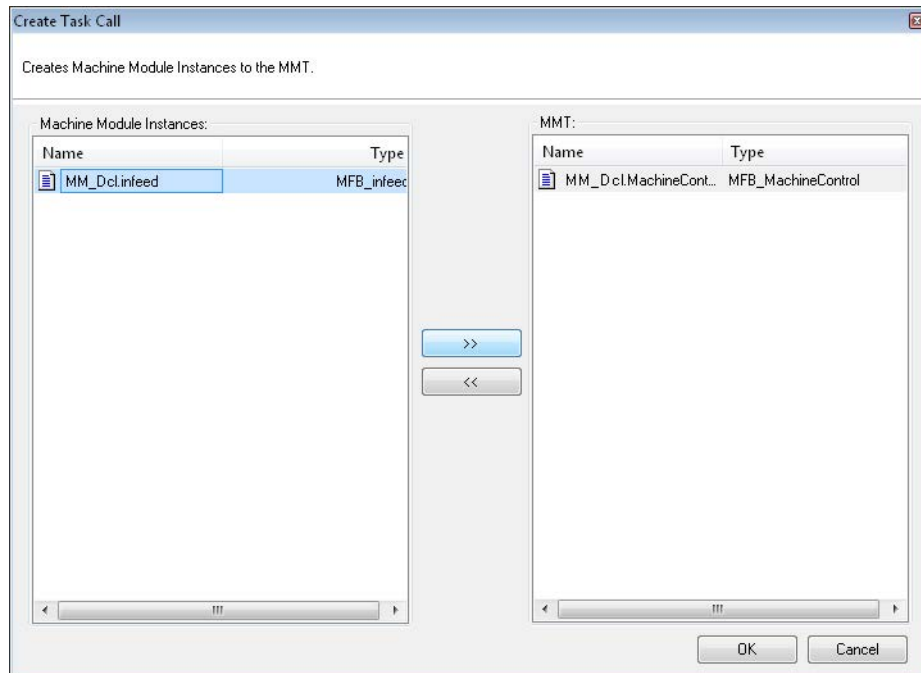
1. Unter **A10_MachineModuleTree** den Menübefehl "Insert Machine Module Instance" ausführen.



- Im erscheinenden Dialog die neue Instanz "MM_Dcl.Infeed" mit der Schaltfläche ">>" in die Liste "MMT" verschieben.

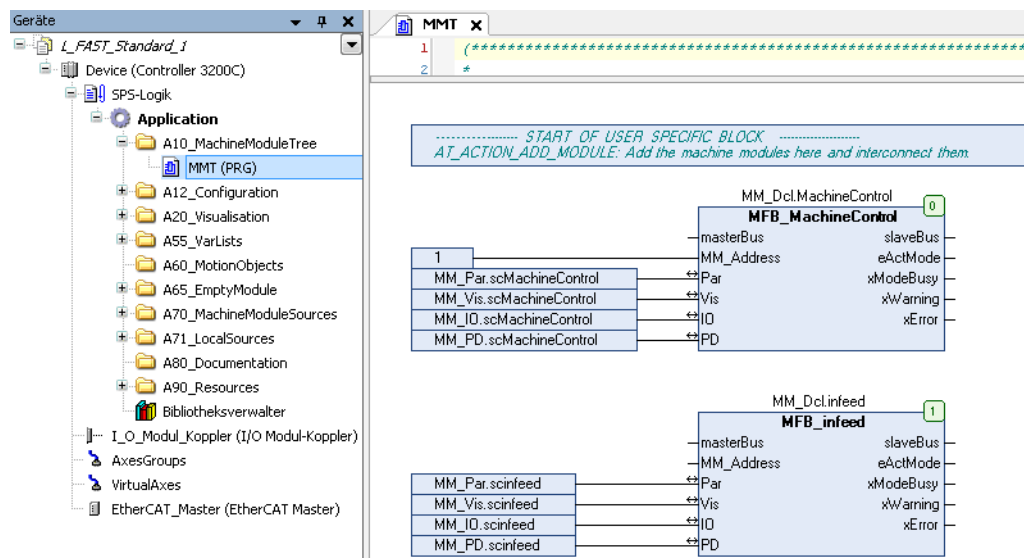
In der Liste "Machine Module Instances" (links) werden die Modulinstanzen angezeigt, die im Machine Module Tree eingebunden werden können.

In der Liste "MMT" (rechts) werden die Modulinstanzen angezeigt, die dem Machine Module Tree hinzugefügt werden sollen.



- Die Auswahl mit **OK** bestätigen.

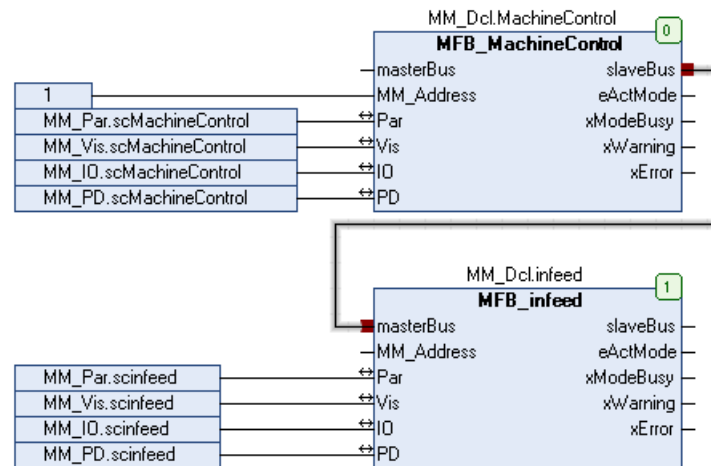
Die neue Instanz "Infeed" wird unter dem schon vorhandenen Master-Modul "MachineControl" platziert und kann per "Drag & Drop" an die gewünschte Position gezogen werden.



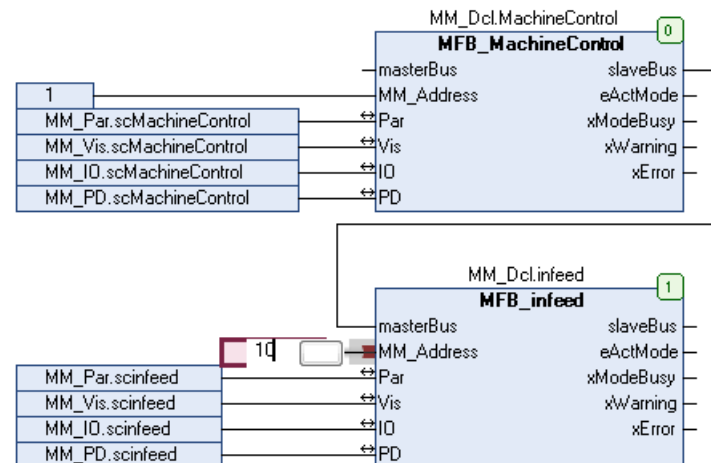
Die zugehörigen Strukturen (MACD, MEL, MIO, MMD, MPar, MPD, MVis) werden automatisch mit den Eingängen des Moduls verbunden.

Die Ausführungsreihenfolge wird an der oberen rechten Ecke des Moduls angezeigt. Mit einem Rechtsklick auf den Bildschirm des MMT und den Kontextmenübefehl **"Ausführungsreihenfolge"** kann die Reihenfolge geändert werden.

4. Eine Linie zwischen den Busanschlüssen *slaveBus* und *masterBus* ziehen, um die Kommunikation zwischen dem Master-Modul "MachineControl" und dem Maschinenmodul "Infeed" herzustellen.



5. Die Adresse des Slaves über den Eingang *MM_Address* vorgeben.
 - Die Linie am Eingang anklicken.
 - Die gewünschte Adresse eingeben.



In diesem Stadium reagiert das Maschinenmodul "Infeed" bereits auf Operation-Mode-Kommandos des Master-Moduls "MachineControl".

4.9 Maschinenmodul-Referenzen löschen



So gehen Sie vor:

1. Die globale Variablenliste **A55_VarLists->MM_Dcl** öffnen.
2. Rechtsklick auf die zu löschende Maschinenmodul-Instanz (hier "SlaveB") und Menübefehl **Delete Machine Module References** ausführen.



Hinweis!

Damit der Befehl **Delete Machine Module References** erscheint, muss der Cursor innerhalb des Instanznamens platziert sein, nicht davor oder dahinter.

```

1  {*****}
2  *
3  * (C) 2011 by Lenze SE
4  *
5  * Module :    MM_Dcl
6  *
7  * Summary :   In this list all machine module instances are declared
8  *
9  * History :
10 *
11 *   Date      Author      Changes
12 *   -----
13 *   yyyy-mm-dd John Q. Public  Initially created
14 *
15 *
16 (* AT_ACTION_ADD_MODULE *)
17 {attribute 'qualified_only'}
18 VAR_GLOBAL
19
20 MachineControl : MFB_MachineControl;    ///

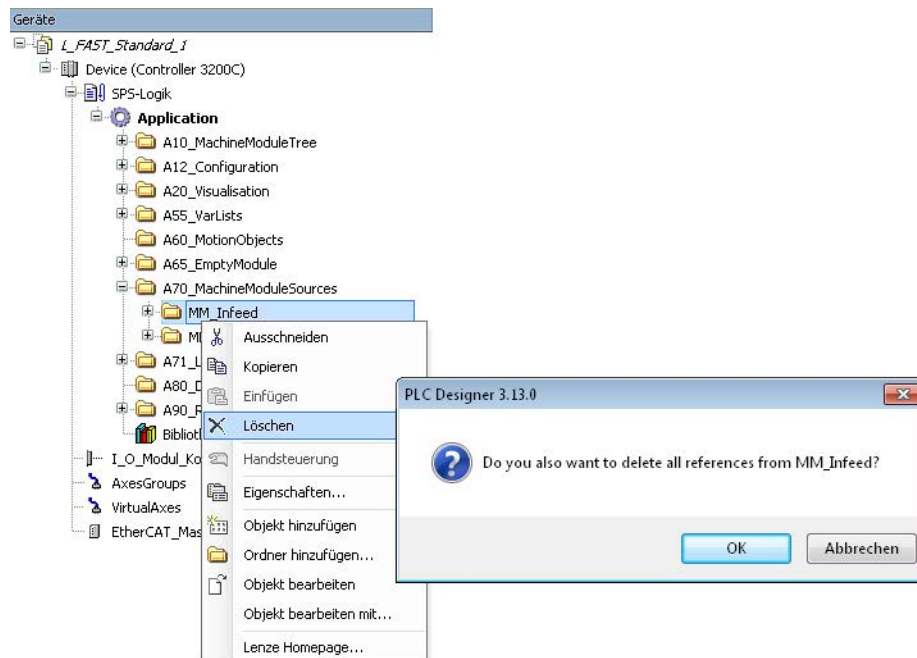
The screenshot shows a right-click context menu over the 'SlaveB' variable declaration. The menu options are: Ausschneiden, Kopieren, Einfügen, Löschen, Alles selektieren, Symbol suchen, Erweitert, Eingabehilfe..., Refactoring, Delete Machine Module References (highlighted), and Copy Machine Module.


```

Alle Referenzen des betreffenden Maschinenmoduls werden gelöscht.

4.10 Maschinenmodul löschen

Löscht man unter **A70_MachineModuleSources** einen Maschinenmodulordner werden nach einer Abfrage auch die automatisch generierten Deklarationen und auch der Aufruf im Machine Module Tree gelöscht.



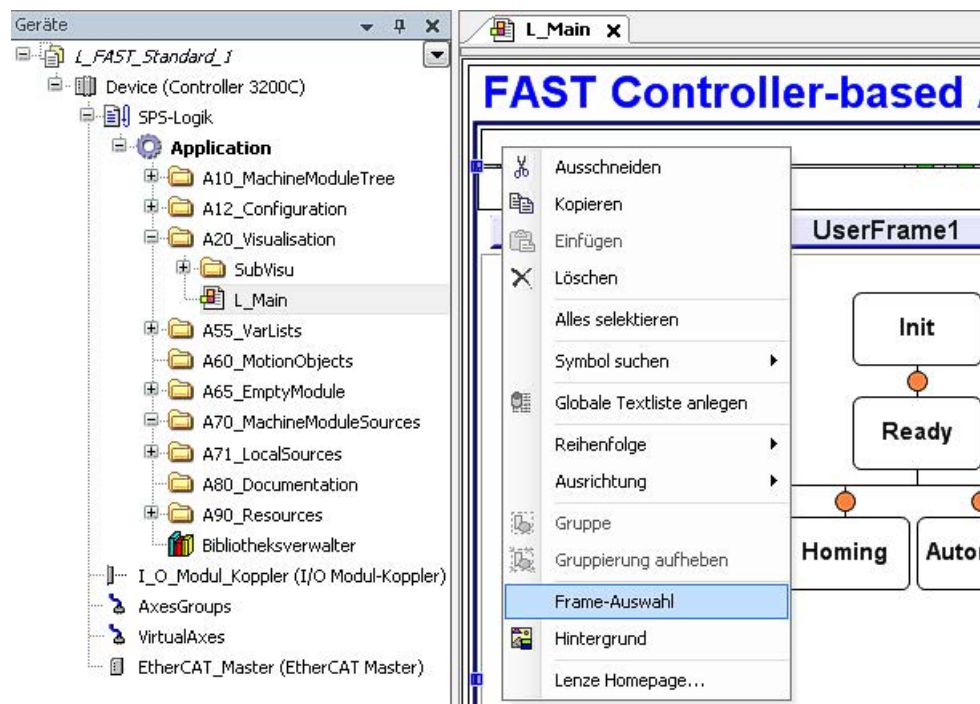
4.11 Visualisierung einbinden und verbinden

Im Application Template müssen die Visualisierungen der einzelnen Maschinenmodule in der Hauptvisualisierung **L_Main** hinzugefügt werden. Die Visualisierung für das Master-Modul "MachineControl" ist bereits enthalten.



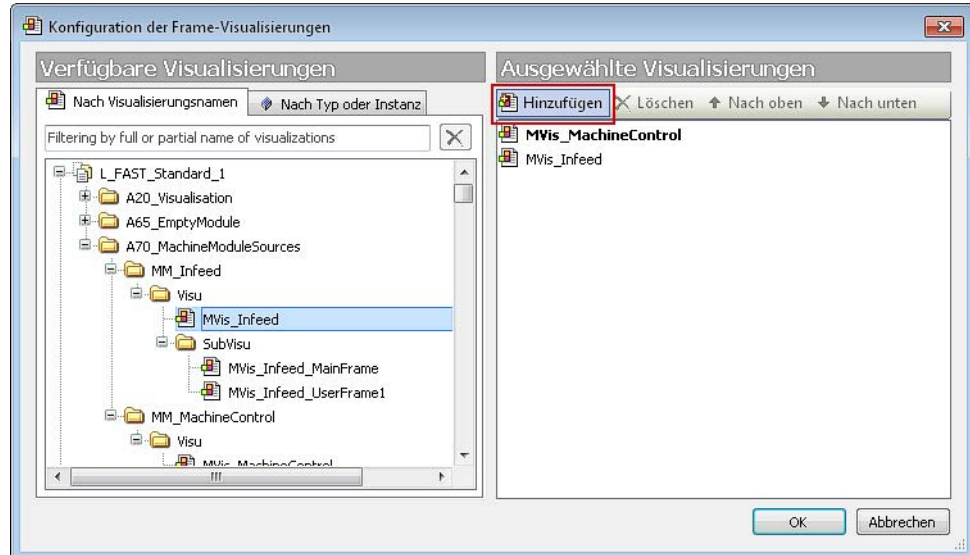
So fügen Sie die Visualisierung des Maschinenmoduls "MM_Infeed" hinzu:

1. Unter **A20_Visualisation** die Hauptvisualisierung **L_Main** öffnen.
2. Mit dem Menübefehl **Frame-Auswahl** den Dialog zur Konfiguration der Frame-Visualisierungen öffnen.



3. Unter "Verfügbare Visualisierungen" die Visualisierung **MVis_Infeed** im Ordner des Maschinenmoduls **A70_MachineModuleSources/MM_Infeed/Visu** auswählen und die Schaltfläche **Hinzufügen** klicken.

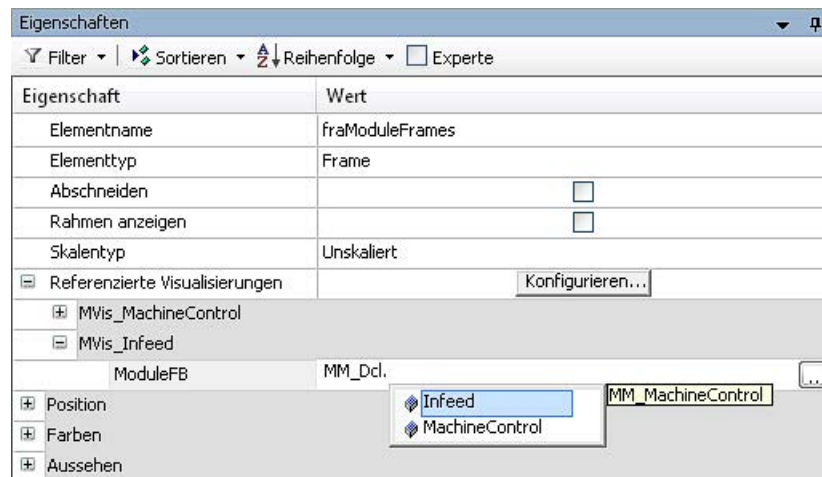
Die Visualisierung wird unter den "Ausgewählten Visualisierungen" eingefügt.



4. Die Auswahl mit **OK** bestätigen.

Mit der Maschinenmodulinstantz des Moduls **MM_Infeed**, welche in der globalen Variablenliste **MM_Dcl** deklariert ist, wird die Visualisierung mit Daten versorgt.

Durch die Intellisense-Funktion werden die verfügbaren Instanzen, welche in der Variablenliste **MM_Dcl** enthalten sind, nach Setzen des Punktes direkt angezeigt.



4.12

FAST Technologiemodule einfügen



So gehen Sie vor:

1. Das einzufügende FAST Technologiemodul "Electrical Shaft Position" in der Struktur **MMD_scInfeed** deklarieren.

Eingabe: InfeedConveyor : L_TT1P_ElectricalShaftPosBase;

Tipp: Durch einen Rechtsklick im Eingabebereich kann über das erscheinende Menü die Eingabehilfe geöffnet werden. Die FAST Technologiemodule finden Sie unter der Kategorie "Strukturierte Typen" → L_TT1P.

The screenshot displays the FAST development environment. On the left, the 'Geräte' (Devices) tree shows the project structure, with 'MMD_scInfeed (STRUCT)' selected under 'StrucTs'. The main editor shows the declaration of the 'InfeedConveyor' variable as 'L_TT1P_ElectricalShaftPosBase;'. A red arrow points from this declaration to the 'Eingabehilfe' (Input Help) dialog box at the bottom. The 'Eingabehilfe' dialog shows a search for 'L_TT1P' and lists various modules under the 'L_TT1P_TechnologyModules' category. The 'L_TT1P_ElectricalShaftPosBase' module is highlighted, showing its type as 'FUNCTION_BLOCK' and its origin as 'l_ttip_tech'.

```

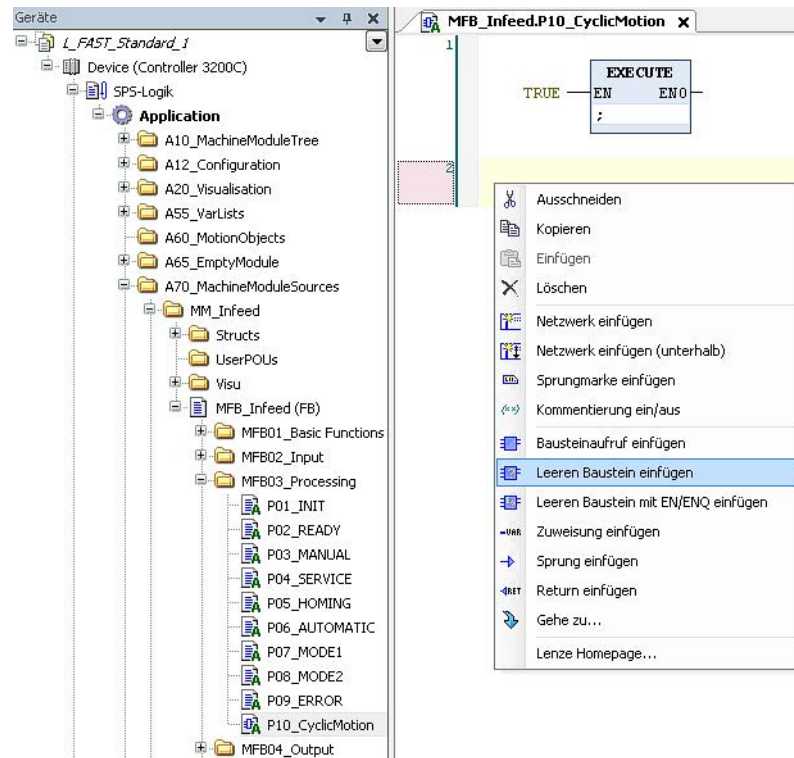
1  TYPE MMD_scInfeed EXTENDS L_EATP_MMD_Base :
2  STRUCT
3  // BasicFunctions vars
4  xDisableCoupling          : BOOL := FALSE;
5  xSendWarningIfDisabled    : BOOL := FALSE;
6  xSendErrorIfDisabled      : BOOL := FALSE;
7  xModeBusy                 : BOOL := FALSE;
8
9  xInit                     : BOOL := FALSE;
10 xReady                    : BOOL := FALSE;
11 xManual                   : BOOL := FALSE;
12 xService                  : BOOL := FALSE;
13 xHoming                   : BOOL := FALSE;
14 xAutomatic                : BOOL := FALSE;
15 xModel                    : BOOL := FALSE;
16 xMode2                    : BOOL := FALSE;
17
18 eOwnModeIfMasterManual    : L_EATP.L_EATP_FAST_OpModes
19 eOwnModeIfMasterService   : L_EATP.L_EATP_FAST_OpModes
20 eOwnModeIfMasterHoming    : L_EATP.L_EATP_FAST_OpModes
21 eOwnModeIfMasterAutomatic : L_EATP.L_EATP_FAST_OpModes
22 eOwnModeIfMasterModel     : L_EATP.L_EATP_FAST_OpModes
23 eOwnModeIfMasterMode2    : L_EATP.L_EATP_FAST_OpModes
24
25 xErrorQuit                : BOOL := FALSE;
26 xErrorAccessIncludeOwnModule : BOOL := TRUE;
27 xErrorAccessIncludeSlaveModules : BOOL := TRUE;
28
29 InfeedConveyor : L_TT1P_ElectricalShaftPosBase;
30 END_STRUCT
31 END_TYPE
32

```

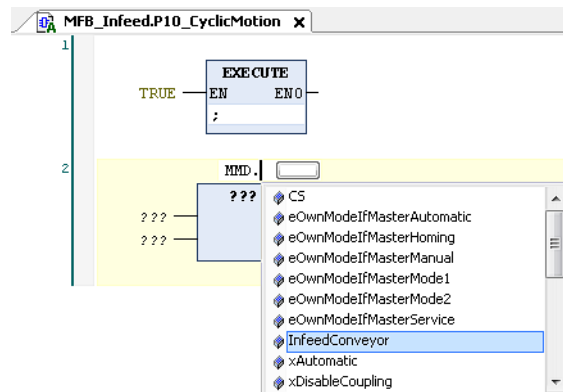
Name	Typ	Herkunft
L_MC1P	Bibliothek	l_mc1p_mo
L_TT1P	Bibliothek	l_ttip_tech
L_TT1P_TechnologyModules		
L_TT1P_BasicMotion		
L_TT1P_CrossCutter		
L_TT1P_ElectricalShaftPos		
L_TT1P_ElectricalShaftPosBase	FUNCTION_BLOCK	l_ttip_tech
L_TT1P_ElectricalShaftPosHigh	FUNCTION_BLOCK	l_ttip_tech
L_TT1P_ElectricalShaftPosState	FUNCTION_BLOCK	l_ttip_tech
Reserved		
structs		
L_TT1P_ElectricalShaftVel		
L_TT1P_ENUMS		
L_TT1P_FlexCam		
L_TT1P_FlyingSaw		

2. Unter **MFB03_Processing** → **P10_CyclicMotion** einen leeren Baustein einfügen.

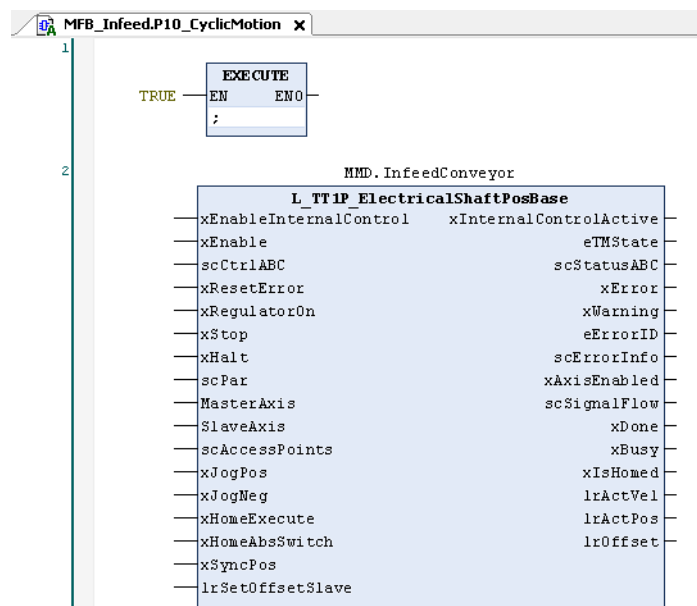
Rechtsklick im Eingabebereich und den Menübefehl "**Leeren Baustein einfügen**" ausführen.



3. An der Stelle "???" oberhalb des Bausteins "MMD." eingeben und in der erscheinenden Auswahlliste **InfeedConveyor** doppelt anklicken.



Der neue Baustein **InfeedConveyor** enthält nun das Technologiemodul "Electricsal Shaft Position".



Die Achsen des Technologiemoduls werden über die Schnittstelle des Maschinenmoduls nach außen geführt und erst dort, im Machine Module Tree, mit den realen Achsen verbunden.

4.13 Achsen verbinden

Im folgenden Beispiel sind für die Prozesse "Zuführen" und "Abwickeln" die FAST Technologiemodule "Electrical Shaft Position" und "Virtual Master" eingebunden.

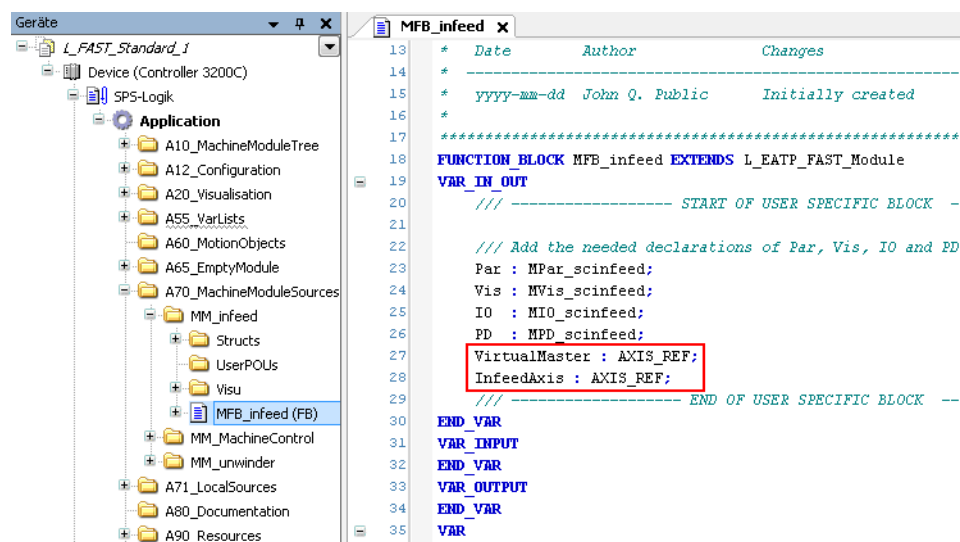
Der im Master-Modul "MachineControl" eingebundene "Virtual Master" gibt die Leitposition für die Maschinenmodule "Infeed" und "Unwinder" vor.

Die Maschinenmodule "Infeed" und "Unwinder" enthalten jeweils ein ElectricalShaftPosBase-Modul. Diese Module sollen nach Erreichen der Endgeschwindigkeit des "Virtual Master" auf die Leitposition einkuppeln. Dazu wird ein "Handshake" zwischen den Maschinenmodulen "Infeed" und "Unwinder" und dem Master-Modul "MachineControl" benötigt (siehe [Kommunikationskanal verwenden](#) (39)).



So gehen Sie vor:

1. Unter **A70_MachineModuleSources** in den eingefügten Maschinenmodulen **AXIS_REF**-Variablen anlegen.

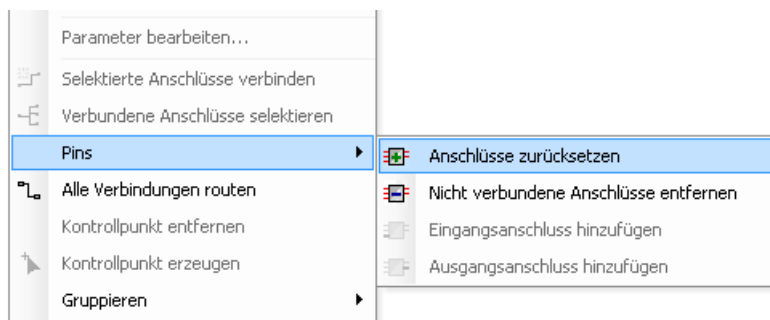


Eingaben

MFB_MachineControl	MFB_infeed	MFB_unwinder
VirtualMaster : AXIS_REF;	VirtualMaster : AXIS_REF; InfeedAxis : AXIS_REF;	VirtualMaster : AXIS_REF; UnwinderAxis : AXIS_REF;

2. Unter **A10_MachineModuleTree** das Programm **MMT (PRG)** öffnen.

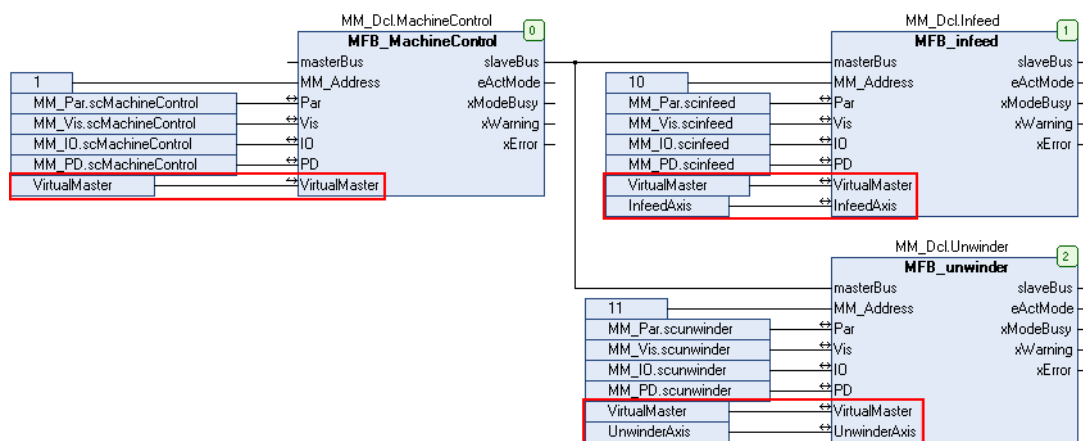
- Rechtsklick auf das Maschinenmodul "Infeed" und den Menübefehl "Pins → Anschlüsse zurücksetzen" ausführen.



Jetzt werden alle Technologiemodule gerechnet.

Im Programm **MMT (PRG)** werden die AXIS_REF-Variablen eingefügt.

- Die realen und virtuellen Achsen an die Bausteine anschließen.

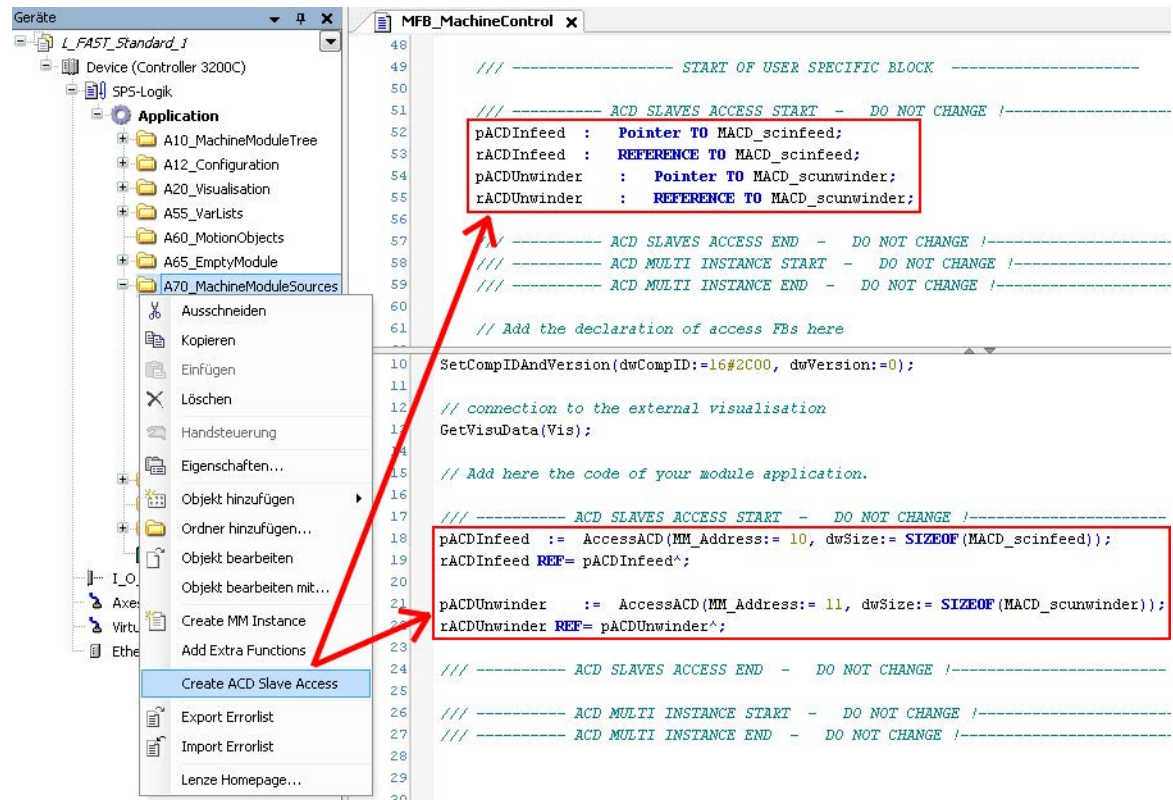


4.14

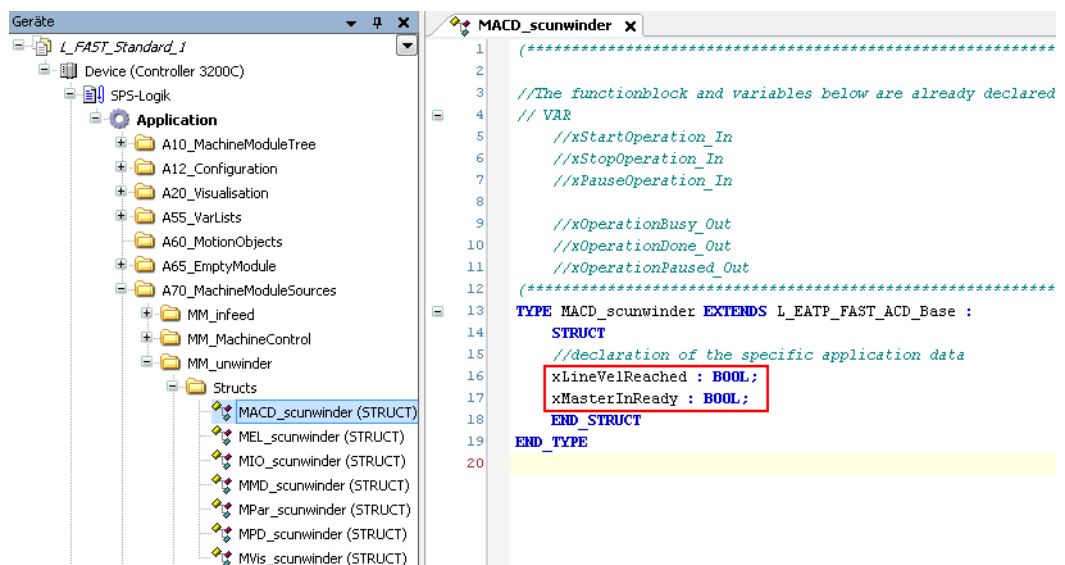
Kommunikationskanal einrichten (ACD Slave Access)

Mit einem Rechtsklick auf den Ordner **A70_MachineModuleSources** das Kontextmenü öffnen und den Menübefehl **"Create ACD Slave Access"** ausführen.

Der ACD-Kanal wird automatisch erstellt und kann sofort genutzt werden. Das Master-Modul "MachineControl" hat mit *rACDInfeed* und *rACDUnwinder* Zugriff auf die ACD-Strukturen von "Infeed" und "Unwinder".



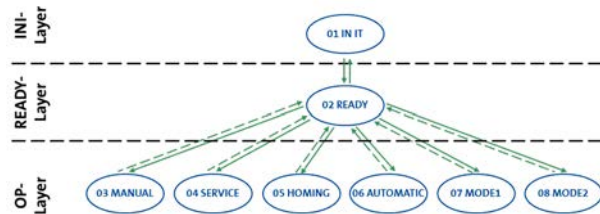
Zur Erweiterung der Schnittstelle können Variablen in der MACD-Struktur des jeweiligen Maschinenmduls eingetragen werden:



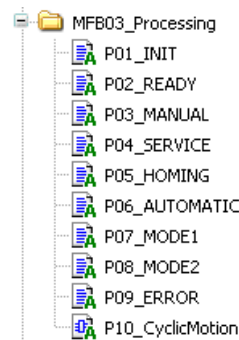
► [Kommunikationskanal verwenden](#) (39)

4.15 Operation-Modes verwenden

Die Operation-Modes eines Maschinenmoduls sind vordefiniert und nicht erweiterbar. Sie gliedern sich in den Initialisierungsmodus (01 INIT/INI-Layer), den Bereitschaftsmodus (02 READY/READY-Layer) und sechs namentliche Arbeitsmodi (03 MANUAL ... 08 MODE 2 / OP-Layer).



Für jeden der acht Operation-Modes ist im Ordner **MFB03_Processing** eine zugeordnete Aktion vordefiniert, in der die Programmlogik untergebracht werden kann. Das Mode-Modell zur Umschaltung der Operation-Modes ist in allen Maschinenmodulen gleich und wird in den meisten Applikationen zentral über den Master gesteuert.



In jedem Mode ist bereits vordefinierter Programmcode vorhanden. In dieses Gerüst kann der Anwender-Code geschrieben werden.

Es gibt drei Bereiche, in denen innerhalb eines Modes Programmcode eingefügt werden kann:

- 1** "ModeEntry" wird beim Eintritt in den Mode für einen Zyklus ausgeführt.
- 2** "Cyclic Area" wird solange ausgeführt, bis der Mode umgeschaltet wird.
- 3** "ModeExit" wird beim Verlassen des Modes ausgeführt.

```

1  (* AT_ACTION_CREATE_NEW_MODULE *)
2  // First pulse, last pulse and cyclic program
3  1 IF xModeEntry THEN
4      // Do init steps
5
6      // Set ModeBusy
7      MMD.xModeBusy := TRUE;
8
9  3 ELSEIF xModeExit THEN
10     // Do exit steps
11
12  2 ELSE
13     // Do cyclic things
14
15     // Reset own ModeBusy
16     MMD.xModeBusy := FALSE;
17
18     // Check if nobody is busy
19     IF NOT OpModeControl.xIsBusy THEN
20         // No module is busy
21         ;
22     END_IF
23 END_IF
24

```

Beispiel

Programmcode zum Einschalten des Technologiemoduls (ElectricalShaftPosBase) der Zuführachse (Infeed) im Mode "READY":

```

1  (* AT_ACTION_CREATE_NEW_MODULE *)
2  // First pulse, last pulse and cyclic program
3  IF xModeEntry THEN
4      // Do init steps
5
6      // Set ModeBusy
7      MMD.xModeBusy := TRUE;
8
9  ELSIF xModeExit THEN
10     // Do exit steps
11
12 ELSE
13     // Do cyclic things
14     MMD.InfeedConveyor.xEnable := TRUE;
15     MMD.InfeedConveyor.xRegulatorOn := TRUE;
16     // Reset own ModeBusy
17     IF MMD.InfeedConveyor.xAxisEnabled THEN
18         MMD.xModeBusy := FALSE;
19     END_IF
20
21     // Check if nobody is busy
22     IF NOT OpModeControl.xIsBusy THEN
23         // No module is busy
24         ;
25     END_IF
26 END_IF

```

Im ersten Schritt wird im Mode "READY" die Achse eingeschaltet. Dazu werden die Eingänge *xEnable* und *xRegulatorOn* auf TRUE gesetzt.

Solange der Ausgang *xAxisEnabled* auf FALSE gesetzt ist, wird das Bit *xModeBusy* nicht auf FALSE zurückgesetzt. Das bedeutet, solange *xModeBusy* auf TRUE gesetzt ist, ist die Umschaltung in einen anderen Mode blockiert. So kann auf einfache Art und Weise verhindert werden, dass z. B. der Mode von "READY" zu "AUTOMATIC" umgeschaltet wird, ohne dass alle Achsen betriebsbereit sind.

4.16 Kommunikationskanal verwenden

Der ACD-Kanal dient zum Austausch von Handshake-Signalen zwischen der Master-Achse (MachineControl/VirtualMaster) und den Slave-Achsen (Infeed, Unwinder).

Der ACD-Kanal besteht immer nur zwischen Master und Slave. Eine Querkommunikation ist nicht vorgesehen.

► [Kommunikationskanal einrichten \(ACD Slave Access\)](#) (36)

Im Beispielprojekt wird der ACD-Kanal verwendet, um die Synchronisierung auf die Master-Achse erst einzuleiten, wenn die Master-Achse ihre Soll Drehzahl erreicht hat. Dazu wurde im Modus "Automatic" im Cyclic-Bereich eine Abfrage programmiert, ob die Sollgeschwindigkeit *lrSetVel* gleich der Ausgabesollgeschwindigkeit *lrSetVelOut* ist:

```

MFB_MachineControl.P06_AUTOMATIC x
12 ELSE
13     // Do cyclic things
14     IF MMD.VirtualMaster.lrSetVel = MMD.VirtualMaster.lrSetVelOut THEN
15         rACDInfeed.LineVelReached := TRUE;
16         rACDUnwinder.LineVelReached := TRUE;
17     END_IF
18     // Reset own ModeBusy
19     IF OpModeAccess.xModeChangeRequested THEN

```

Ist die Soll Drehzahl erreicht, wird die Variable *LineVelReached* aus dem ACD- Kanal auf TRUE gesetzt.

Im Slave wird dieser Wert dann vom ACD-Kanal abgefragt und daraufhin die Synchronisierung gestartet.

Vom Master zum Slave wird der ACD-Kanal über die Referenz *rACD[Slave instance name]* genutzt. Soll das Bit *LineVelReached* für den Slave "Infeed" gesetzt werden, lautet die Programmierzeile:

```
rACDZufuehrung.LineVelReached := TRUE;
```

Um diesen Wert im Slave zu lesen wird auf die MACD-Struktur zugegriffen. Der Wert wird dann mit *MACD.LineVelReached* ausgelesen:

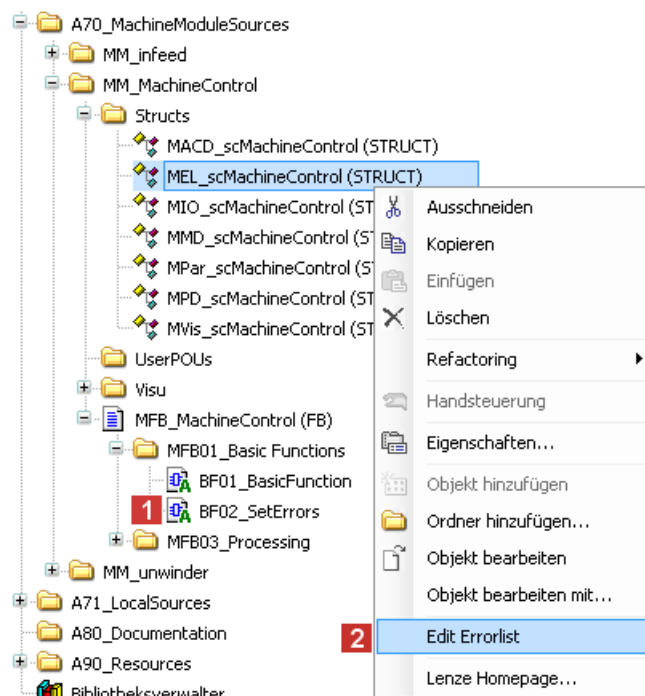
```

MFB_Infeed.P06_AUTOMATIC x
12 ELSE
13     // Do cyclic things
14     IF MACD.LineVelReached THEN
15         MMD.InfeedConveyor.xSyncPos := TRUE;
16     END_IF
17
18     IF MACD.MasterIsInReady := TRUE THEN

```

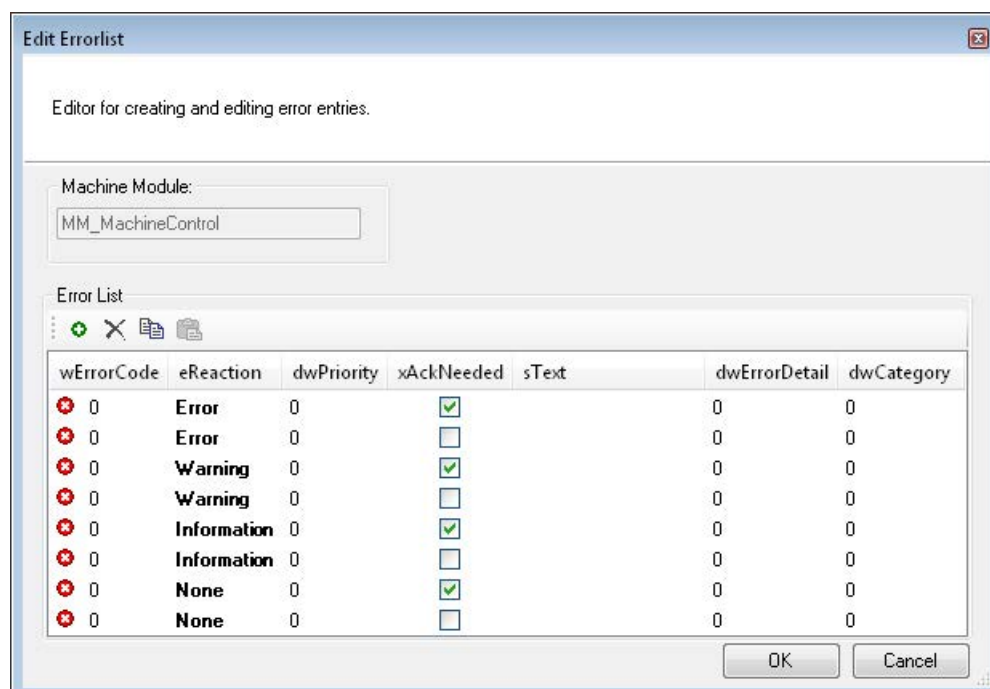
4.17 Fehlermeldungen anlegen und bearbeiten

Jedes Maschinenmodul verfügt über Aufrufbausteine zur Fehlerbehandlung und eine Struktur zur Definition von Fehlermeldungen, hier beispielhaft am Master-Modul "MachineControl" dargestellt:

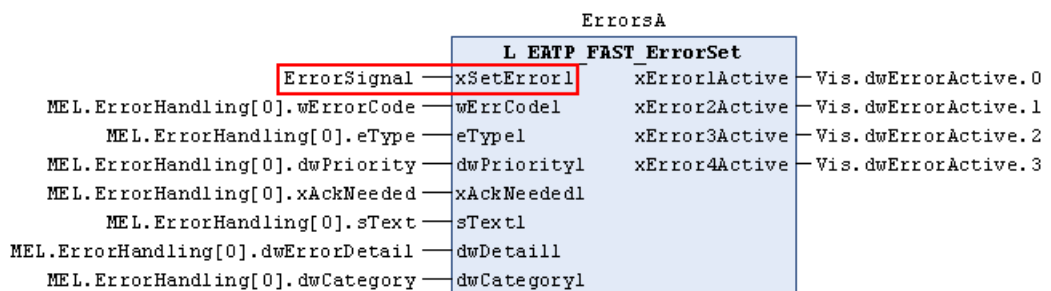


Im Ordner **MFB01_BasicFunctions** befindet sich die Aktion **1 BF02_SetErrors**, in der die Aufrufbausteine für die Fehlerbehandlung enthalten sind. In der Projektvorlage sind bereits zwei Aufrufe vorhanden, die die Auslösung von 8 Fehlermeldungen ermöglichen.

Mit einem Rechtsklick auf die MEL-Struktur und dem Menübefehl **2 "Edit Errorlist"** können Sie in einem Dialogfenster Fehlermeldungen hinzufügen, löschen und mit allen notwendigen Eigenschaften definieren.



Das auslösende Fehlersignal wird an den Baustein unter **BF02_SetErrors** an den Eingang *xSetError* angelegt.



Wird das Fehlersignal ausgelöst, so kommt es zur Reaktion "Error". Dabei wird die Aktion P09_Error parallel zum aktuellen Operation Mode ausgeführt.

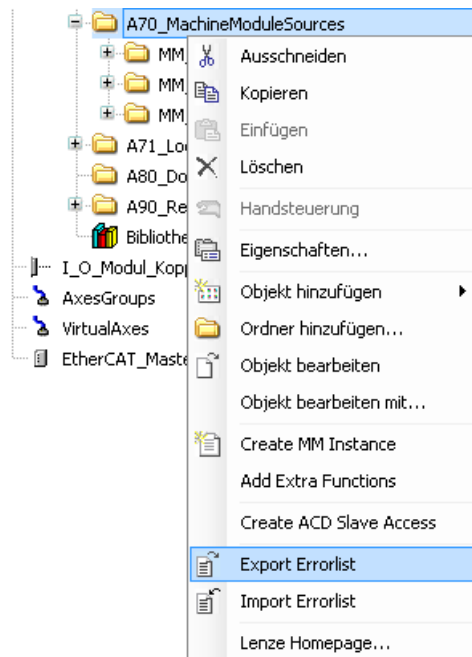
4.17.1 Fehlerliste exportieren (XML-Datei)

Eine Fehlerliste ist eine XML-Datei, die alle Fehler enthält, welche in den einzelnen Maschinenmodulen im Ordner **A70_MachineModuleSources** definiert sind. Die strukturierte Anordnung der Fehlerinformationen in der XML-Datei entspricht der Reihenfolge der Maschinenmodule im Projektordner.

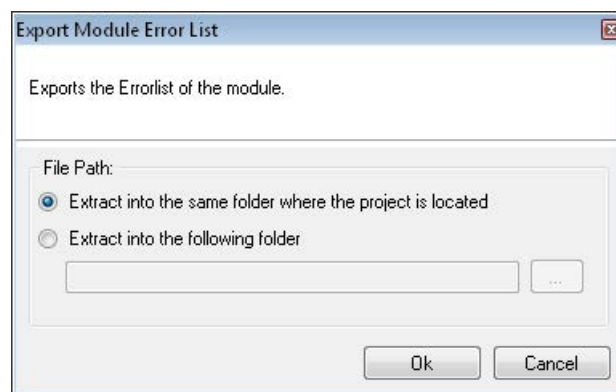


So gehen Sie vor:

1. Rechtsklick auf den Ordner **A70_MachineModuleSources** und den Menübefehl **Export Errorlist** ausführen.



2. Im folgenden Dialog das Zielverzeichnis für die XML-Datei vorgeben.



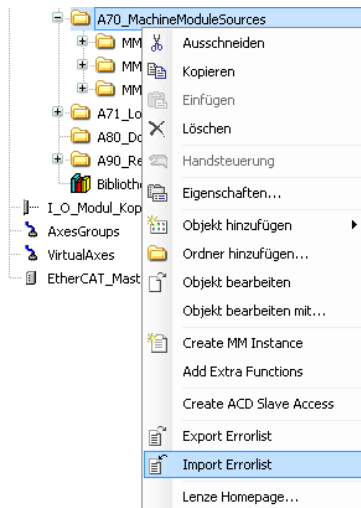
3. Abschließend mit **OK** den Export ausführen.

4.17.2 Fehlerliste importieren (XML-Datei)

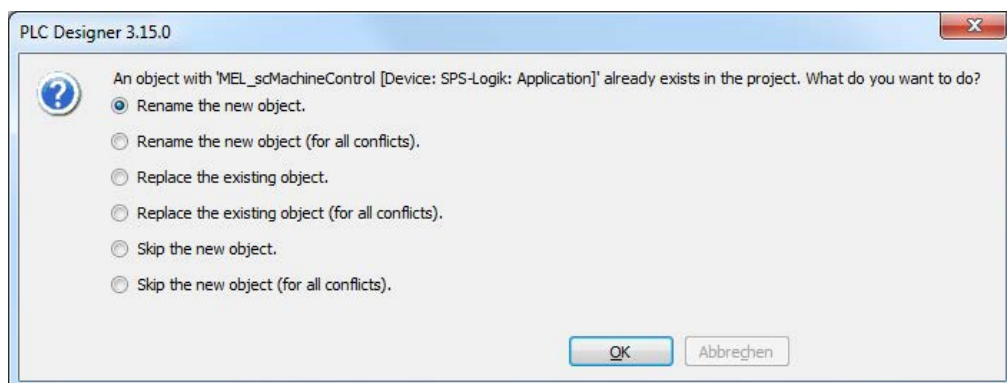


So gehen Sie vor:

1. Rechtsklick auf den Ordner **A70_MachineModuleSources** und den Menübefehl **Import Errorlist** ausführen.



2. Im folgenden Dialog das Quellverzeichnis und die zu importierende XML-Datei auswählen.
3. Fehlerlisten-Import mit **Öffnen** ausführen.
4. Aktionen für die Fehlerinformationen auswählen:



Für jede Fehlerliste eines Moduls kann einer der folgenden Aktionen gewählt werden:

Rename the new object: Die in der XML-Datei zu diesem Maschinenmodul enthaltenen Fehlerinformationen werden im Projekt unter einem neuen Namen gespeichert. Diese Option sollte nur in Ausnahmefällen verwendet werden.

Replace the existing object: Die im Projekt zu diesem Maschinenmodul enthaltenen Fehlerinformationen werden mit den Werten aus der XML-Datei überschrieben. Für die Mehrzahl der Anwendungsfälle, bei denen die Fehlerinformationen mit einem externen Tool bearbeitet werden, ist dies die Standardaktion.

Skip the new object: Die in der XML-Datei zu diesem Maschinenmodul enthaltenen Fehlerinformationen werden verworfen, die Fehlerinformationen im Projekt bleiben unverändert.

**Hinweis!**

In der Regel ist es nicht sinnvoll, eine Fehlerliste zu importieren, die mit einer anderen Anordnung und/oder Benennung von Maschinenmodulen im Ordner A70_MachineModuleSources erstellt wurde.

In der XML-Datei werden die Attribute für jeden Fehlereintrag einmal als einzelne XML-Tags abgelegt. Zusätzlich wird ein XML-Tag "InterfaceAsPlainText" ausgegeben, das die gesamte Fehlerinformation zu einem Maschinenmodul als Strukturierten Text (ST) enthält.

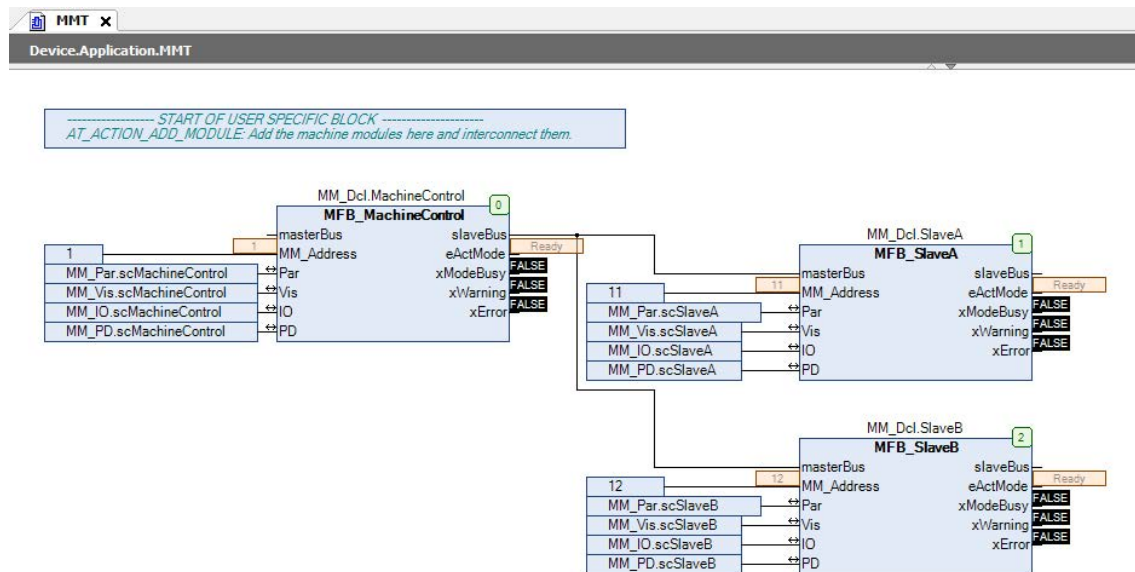
Sollen die Fehlerattribute mit einem externen Tool bearbeitet werden, so ist diese Bearbeitung innerhalb dieses XML-Tags vorzunehmen. Nur dieses XML-Tag wird beim (Re-)Importieren der Fehlerliste berücksichtigt.

4.17.3 Modulkopplung verwenden

Mit der Trennung der Kopplung eines Maschinenmoduls zu seinem übergeordneten Modul wird bewirkt, dass der Operation-Mode des abgekoppelten Moduls nicht mehr dem Master folgt, sondern unabhängig festgelegt werden kann.

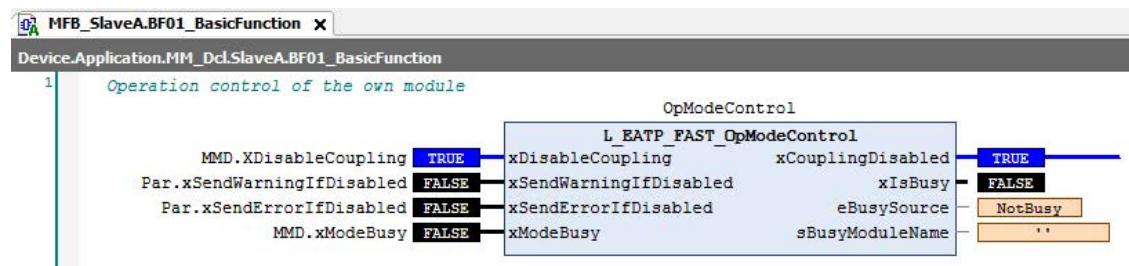
Beispiel

Die folgende Struktur besteht aus einem Maschinenmodul **MachineControl** und zwei unterlagerten Modulen **SlaveA** und **SlaveB**. Das Modul **SlaveA** soll abgekoppelt werden.



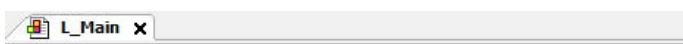
Das Signal *MMD.xDisableCoupling* des Moduls **SlaveA** wird auf TRUE gesetzt.

Dieses Signal liegt am Eingang *xDisableCoupling* des Bausteins **OpModeControl** in der Aktion *MFB_SlaveA.BF01_BasicFunction* an.



Der Ausgang *xCouplingDisabled* am gleichen Baustein zeigt den aktuellen Zustand der Modulkopplung für das Modul an.

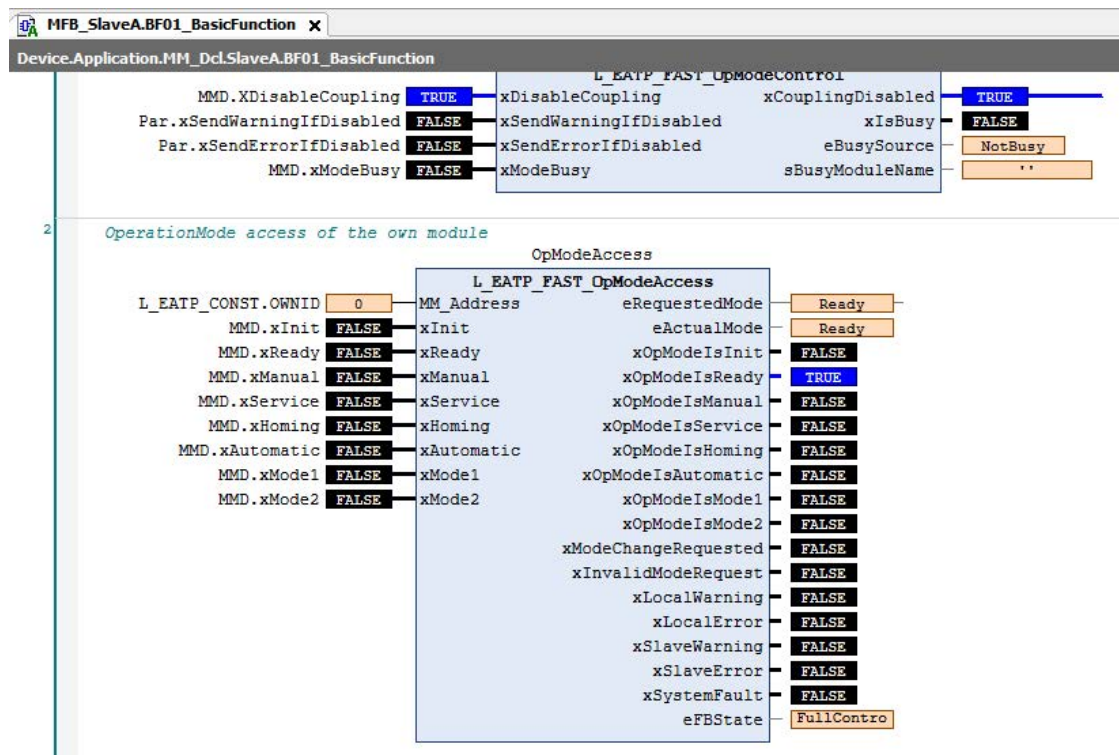
Die Visualisierung **L_Main** enthält in der Modulliste in der Spalte "CPL" die Kennung "DIS" für das abgekoppelte Modul **SlaveA**.



The screenshot shows a software window titled "L_Main" with a close button. Below the title bar is a table titled "FAST Controller-based Automation". The table has five columns: "MM Addr.", "MM Name", "ST", "BY", and "CPL". There are four rows of data. The first row has MM Addr. 1, MM Name MachineControl, ST green circle, BY green circle, and CPL DEF. The second row has MM Addr. 2, MM Name SlaveA, ST green circle, BY green circle, and CPL DIS. The third row has MM Addr. 3, MM Name SlaveB, ST green circle, BY green circle, and CPL DEF. The fourth row is empty. To the right of the table is a vertical sidebar with a scroll bar.

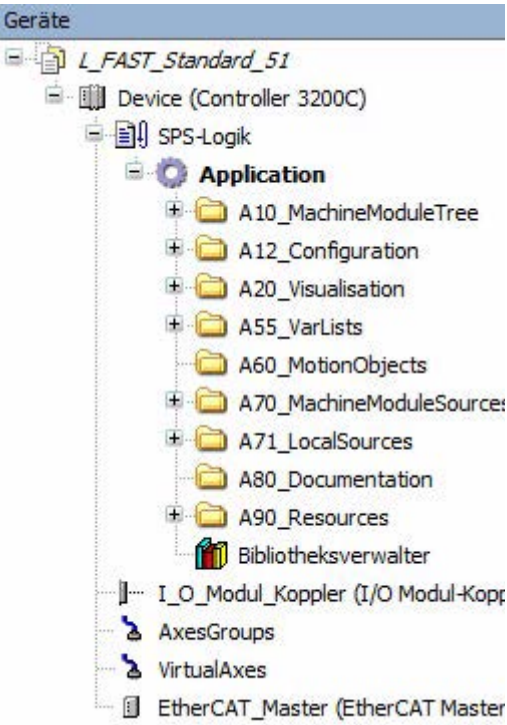
	MM Addr.	MM Name	ST	BY	CPL
1	1	MachineControl	●	●	DEF
2	1.11	SlaveA	●	●	DIS
3	1.12	SlaveB	●	●	DEF
4					

Der Operation-Mode des Moduls **SlaveA** kann jetzt über die Eingänge des Bausteins **OpModeAccess** (Aktion *MFB_SlaveA._BF01_BasicFunction*) unabhängig vom übergeordneten Modul **MachineControl** gesteuert werden.



Zur Wiederherstellung der Kopplung wird das Signal *MMD.xDisableCoupling* auf den Standardwert FALSE zurückgesetzt. Sobald die beiden beteiligten Maschinenmodule den gleichen Operation-Mode aufweisen, wird die Kopplung wiederhergestellt.

5 Struktur des Application Template

Struktur des Application Template	
	A10_MachineModuleTree (48) Verknüpfung der einzelnen Maschinenmodule
	A12_Configuration (50) Grundkonfiguration des Application Template
	A20_Visualisation (51) Hauptvisualisierung des Application Template zur Einbindung der einzelnen Maschinenmodul-Visualisierungen
	A55_VarLists (55) Deklaration der Maschinenmodulinstanzen und deren Strukturen
	A60_MotionObjects (55) Motion-relevante Daten (z. B. CAM-Profile)
	A70_MachineModuleSources (56) Maschinenmodule
	A71_LocalSources Maschinenunabhängige Enumerationen, Funktionsbausteine, Strukturen, Visualisierungen
	A80_Documentation Dokumentationsordner zur Ablage von Dokumenten zur Projekthistorie (z. B. Versionsinformationen, Änderungen) und Bedienungsanleitungen etc.
	A90_Resources Systeminformationen wie z. B. verwendete Bibliotheken und Task-Einstellungen

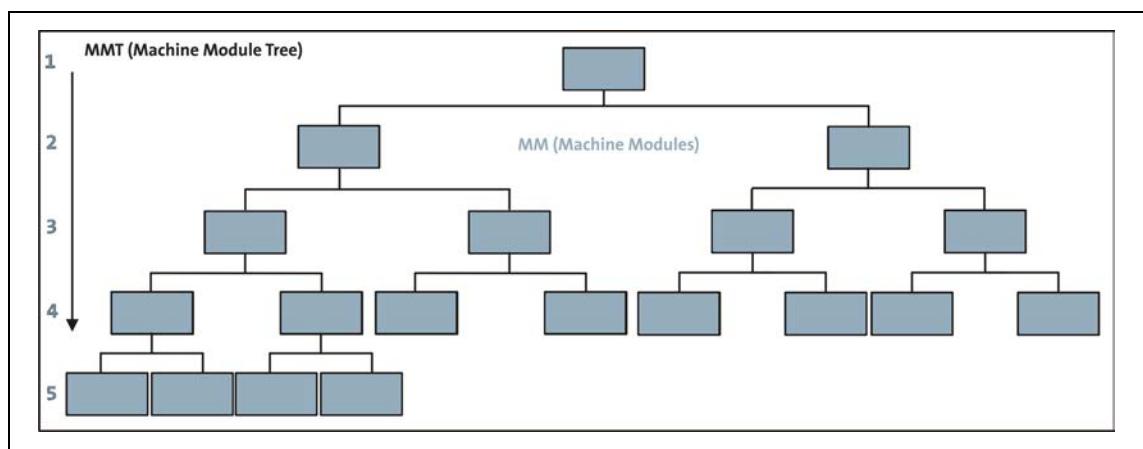
5.1 A10_MachineModuleTree

Um das gewünschte Automationssystem auf Basis des Application Template im »PLC Designer« abzubilden, müssen Sie zunächst die mechatronische Gesamtfunktionalität der Maschine in einzelne Teilfunktionen der Maschine aufteilen.

Aus den Teilfunktionen können Sie einzelne wiederverwendbare Maschinenmodule bilden, die sich in Form einer Baumstruktur – dem "Machine Module Tree" – abbilden lassen. Dabei ist das oberste Modul das Maschinensteuermodul. Diesem Master-Modul sind alle weiteren Maschinenmodule untergeordnet.

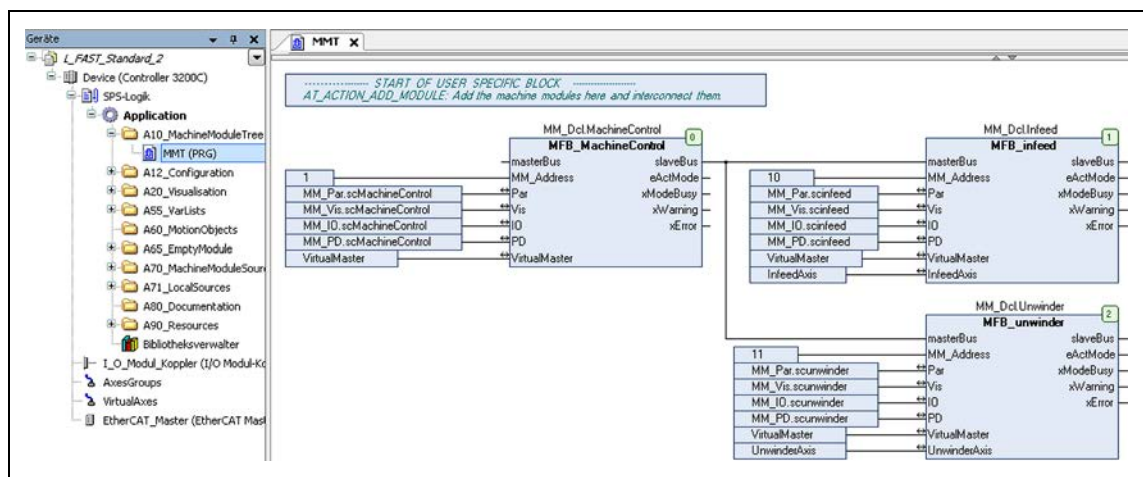
Das Application Template unterstützt ...

- zwei bis fünf Hierarchie-Ebenen von Maschinenmodulen;
- bis zu 40 Maschinenmodule.



[5-1] Machine Module Tree (MMT) mit fünf Hierarchie-Ebenen

Im »PLC Designer« unter dem Ordner **A10_MachineModuleTree** → **MMT (PRG)** wird die Baumstruktur von links nach rechts dargestellt:

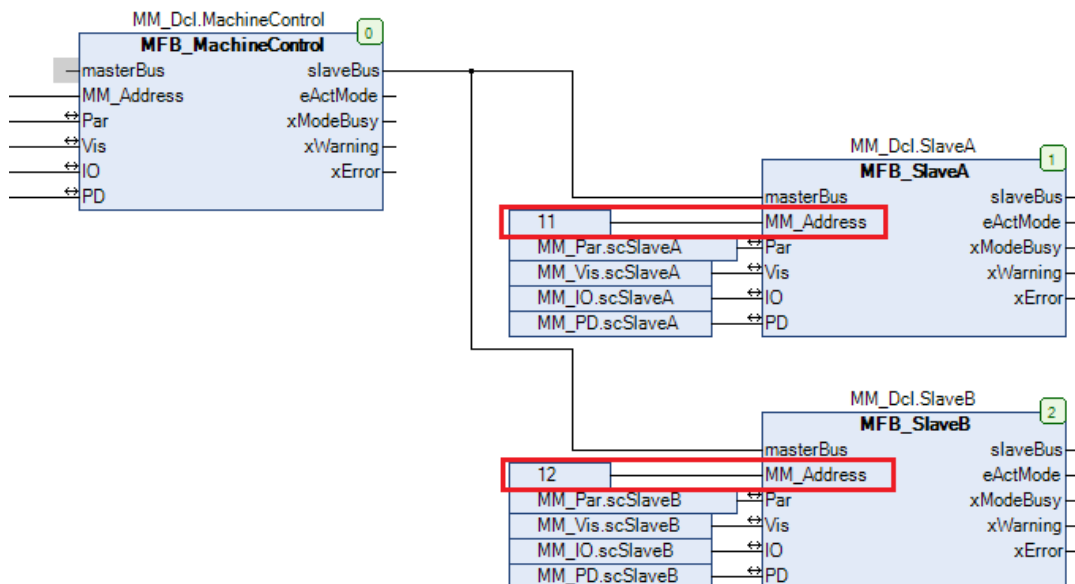


[5-2] Beispielprojekt: MMT (PRG)

- ▶ [Maschinenmodul anlegen](#) (16)
- ▶ [Maschinenmodul im Machine Module Tree \(MMT\) einbinden](#) (24)

MM_Address

Jedes Maschinenmodul hat einen Eingang *MM_Address*, der vom Anwender belegt werden muss. Mit dieser Adresse wird ein Maschinenmodul eindeutig gegenüber seinem übergeordneten Mastermodul identifiziert.



Für die Vergabe der Adressen gelten folgende Randbedingungen:

- Die Adresse ist eine positive ganze Zahl (>0).
- Alle Maschinenmodule, die gemeinsam an einem übergeordneten Mastermodul angeschlossen sind, müssen unterschiedliche Adressen aufweisen.
- Nicht erforderlich ist, die Adressen lückenlos oder bei 1 beginnend zu vergeben.

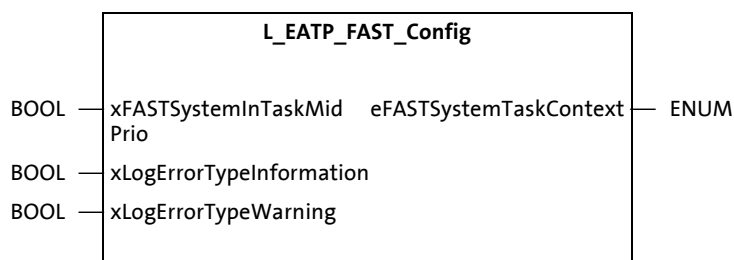
5.2 A12_Configuration

In diesem Ordner befindet sich der Funktionsbaustein **L_EATP_FAST_Config**, an dem eine Grundkonfiguration des Application Template vorgenommen werden kann.



Hinweis!

Einstellungen an diesem Funktionsbaustein können nur vor dem Start der PLC und nicht während der Laufzeit durchgeführt werden.



Eingänge

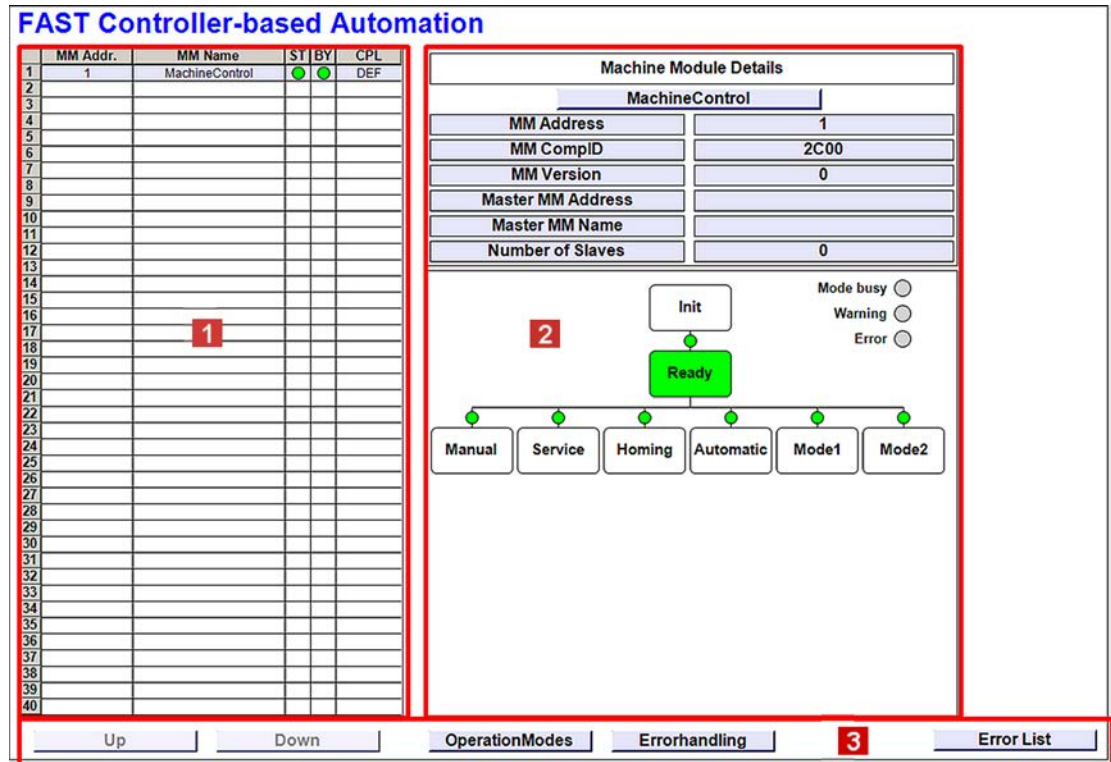
Bezeichner	Datentyp	Beschreibung	
xFASTSystemInTaskMidPrio	BOOL	TRUE	Die Logik- und Infrastrukturfunktionen des Application Template werden in der Mid-Task berechnet. Dadurch wird die High-Task entlastet.
xLogErrorTypeInfoInformation	BOOL	TRUE	Die Informationen, die über den ErrorSet-Baustein (BF02_SetErrors) erzeugt wurden, werden in das Logbuch eingetragen.
xLogErrorTypeWarning	BOOL	TRUE	Die Warnungen, die über den ErrorSet-Baustein (BF02_SetErrors) erzeugt wurden, werden in das Logbuch eingetragen.

Ausgänge

Bezeichner	Datentyp	Beschreibung
eFASTSystemTaskContext	ENUM	Anzeige der Task, in der die Logik- und Infrastrukturfunktionen des Application Template berechnet werden.

5.3 A20_Visualisation

In diesem Ordner ist die Hauptvisualisierung **L_Main** enthalten, in der alle Visualisierungen der einzelnen Maschinenmodule eingehängt werden.



Die Hauptvisualisierung besteht aus drei Bereichen:

1 Maschinenmodul-Liste

Anzeige aller Maschinenmodule, die im Machine Module Tree vorhanden sind.

MM Addr.: Maschinenmodul-Adresse

MM Name: Maschinenmodul-Name

ST: Maschinenmodul-Status (grün: fehlerfrei; rot: Fehler liegt an)

BY: Maschinenmodul-Zustand (grün: betriebsbereit; rot: nicht betriebsbereit)

CPL: Maschinenmodul-Kopplungsmodus

- "DEF": Modul-Kopplung ist eingeschaltet.
- "DIS": Modul-Kopplung ist ausgeschaltet. Das IC-Modul befindet sich im Zustand "Internal Control".

Die Reihenfolge in der Liste entspricht der Aufrufreihenfolge.

2 "Machine Module Details"

Anzeige von Detailinformationen des in der Maschinenmodul-Liste gewählten Moduls.

3 Schaltflächen

OperationModes: Detailansicht "Operation Modes" aktivieren.

Errorhandling: Detailansicht "Error handling" aktivieren.

ErrorList: Fehlerliste öffnen.

Up/Down: Cursor (Selektionsmarke) in der Liste verschieben.

Generische Visualisierungen

- ▶ [L_EATP_FAST_VisErrorList](#) (📖 52)
- ▶ [L_EATP_FAST_VisModuleList](#) (📖 53)
- ▶ [L_EATP_FAST_VisModuleDetail](#) (📖 54)

5.3.1 L_EATP_FAST_VisErrorList

Diese Visualisierung zeigt den Inhalt des Arrays *ErrorList*.







Die Schaltfläche **"Create CSV file"** erzeugt eine CSV-Datei, mit allen in den Maschinenmodulen definierten Fehlern.

[illegible]

5.3.2 L_EATP_FAST_VisModuleList

Diese Visualisierung zeigt alle im Machine Module Tree eingebundenen Maschinenmodule.

Über die Schaltflächen "**Up**" und "**Down**" kann die Selektionsmarke in der Liste um ein Element nach oben oder unten geschoben werden. Eine Selektion ist auch direkt per Mausklick auf ein Listenelement möglich.

	MM Addr.	MM Name	STBY	CPL
1	1	MachineControl	 	DEF
2	1.11	SlaveA	 	DEF
3	1.12	SlaveB	 	DEF
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				

Up

Down

5.3.3 L_EATP_FAST_VisModuleDetail

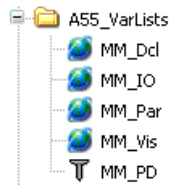
Diese Visualisierung zeigt die Detailinformationen eines Maschinenmoduls, das über die Visualisierung [L_EATP_FAST_VisModuleList](#) (53) ausgewählt wurde.

Detailansicht "Operation Modes"	Detailansicht "Error handling"																														
Machine Module Details																															
MM Address	1																														
MM Name	MachineControl																														
MM CompID	2C00																														
MM Version	0																														
Master MM Address																															
Master MM Name																															
Number of Slaves	2																														
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <p>Init</p> <p>Ready</p> <p>Manual Service Homing Automatic Mode1 Mode2</p> </div> <div> <p>State busy <input type="radio"/></p> <p>Warning <input type="radio"/></p> <p>Error <input checked="" type="radio"/></p> </div> </div>																															
<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p style="color: blue; font-weight: bold;">L_EATP_ErrorSet</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="5" style="text-align: left;">MachineControl.ErrorsA</th> </tr> <tr> <th style="width: 10%;">ErrCo.</th> <th style="width: 10%;">Type</th> <th style="width: 10%;">Cat.</th> <th style="width: 40%;">Error Text</th> <th style="width: 10%;">ErrDet.</th> </tr> </thead> <tbody> <tr> <td>Err1</td> <td>Error</td> <td>0</td> <td></td> <td>0</td> </tr> <tr> <td>Err2</td> <td>Error</td> <td>0</td> <td></td> <td>0</td> </tr> <tr> <td>Err3</td> <td>Warning</td> <td>0</td> <td></td> <td>0</td> </tr> <tr> <td>Err4</td> <td>Warning</td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table> </div> <div style="width: 35%; text-align: right;"> <p>Next</p> </div> </div>		MachineControl.ErrorsA					ErrCo.	Type	Cat.	Error Text	ErrDet.	Err1	Error	0		0	Err2	Error	0		0	Err3	Warning	0		0	Err4	Warning	0		0
MachineControl.ErrorsA																															
ErrCo.	Type	Cat.	Error Text	ErrDet.																											
Err1	Error	0		0																											
Err2	Error	0		0																											
Err3	Warning	0		0																											
Err4	Warning	0		0																											

Previous/Next (in Detailansicht "Errorhandling"): Die vorhergehende/nachfolgende Instanz des Funktionsbausteins L_EATP_FAST_ErrorSet wird anzeigen.

5.4 A55_VarLists

In den Variablenlisten werden die Instanzen der verwendeten Maschinenmodule und der zugehörigen Strukturen deklariert.

Variablenlisten	Beschreibung	
	MM_Dcl	Deklaration der Maschinenmodulinstanzen Die erste Instanz eines Maschinenmoduls wird mit dem Befehl "Insert Empty Module" gebildet. Weitere Instanzen von Maschinenmodulen können mit dem Befehl "Create MM Instance" gebildet werden. Beide Befehle können über das Kontextmenü ausgeführt werden, welches mit einem Rechtsklick auf den Ordner "A70_MachineModuleSources" erscheint.
	MM_IO	Deklaration der Instanzen der MIO-Strukturen der Maschinenmodule
	MM_Par	Deklaration der Instanzen der MPar-Strukturen der Maschinenmodule
	MM_Vis	Deklaration der Instanzen der MVis-Strukturen der Maschinenmodule
	MM_PD	Deklaration der Instanzen der MPD-Strukturen der Maschinenmodule

Die Strukturen (MIO, MPar, MVis, MPD) werden automatisch angelegt, sobald eine Instanz eines Maschinenmoduls mit den Befehlen "Insert Empty Module" oder "Create MM Instance" erzeugt wurde.

5.5 A60_MotionObjects

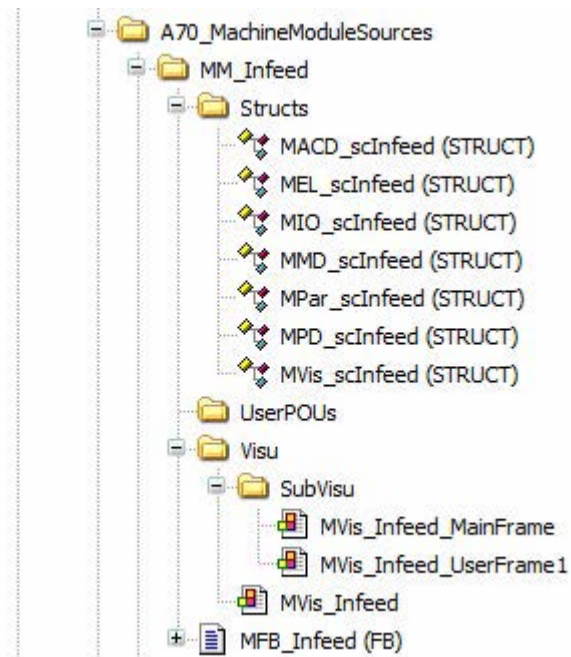
In diesem Ordner können eigene Funktionsbausteine, Funktionen etc. abgelegt werden, die in den Maschinenmodulen im Ordner **A70_MachineModuleSources** verwendet werden.

Wird der Ordner nicht benötigt, kann er gelöscht werden.

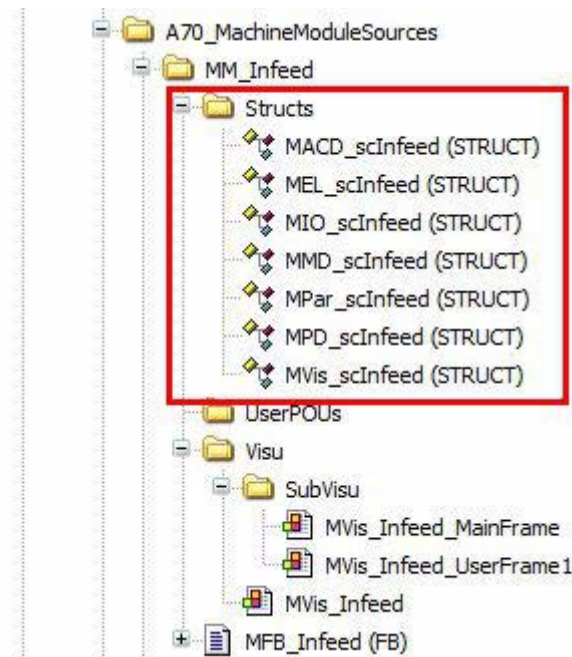
5.6 A70_MachineModuleSources

In diesem Ordner befinden sich bereits angelegte Maschinenmodule. Siehe [Maschinenmodul anlegen](#) (16).

Mit dem Befehl "**Load Machine Module**" werden Maschinenmodule in das Projekt geladen.



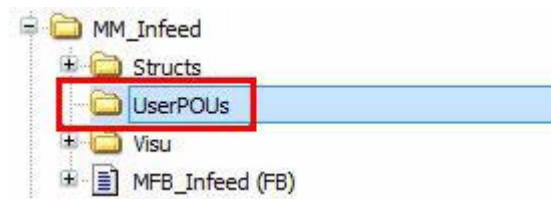
5.6.1 Strukturen



Der Ordner **Structs** enthält die Strukturen des Maschinenmoduls.

Strukturen	Beschreibung																					
MMD_scInfeed	Diese Struktur dient zur Instanziierung von Maschinenmoduldaten. Dazu gehören alle Funktionsbausteine und Variablen, die intern im Maschinenmodul für die Applikation verwendet werden. Auf diese Werte kann mit "MMD." zugegriffen werden.																					
MACD_scInfeed	Diese Struktur dient zum Austausch von Daten zwischen dem Master- und Slave-Maschinenmodul. In dieser Struktur sind folgende Variablen vordefiniert.																					
	<table><tr><th>Variable</th><th>Datentyp</th><th>Beschreibung</th></tr><tr><td>xStartOperation_In</td><td>BOOL</td><td>Steuersignal zum "Starten"</td></tr><tr><td>xStopOperation_In</td><td>BOOL</td><td>Steuersignal zum "Stoppen"</td></tr><tr><td>xPauseOperation_In</td><td>BOOL</td><td>Steuersignal "Pausieren"</td></tr><tr><td>xOperationBusy_Out</td><td>BOOL</td><td>Statussignal "Busy"</td></tr><tr><td>xOperationDone_Out</td><td>BOOL</td><td>Statussignal "Done"</td></tr><tr><td>xOperationPaused_Out</td><td>BOOL</td><td>Statussignal "Paused"</td></tr></table>	Variable	Datentyp	Beschreibung	xStartOperation_In	BOOL	Steuersignal zum "Starten"	xStopOperation_In	BOOL	Steuersignal zum "Stoppen"	xPauseOperation_In	BOOL	Steuersignal "Pausieren"	xOperationBusy_Out	BOOL	Statussignal "Busy"	xOperationDone_Out	BOOL	Statussignal "Done"	xOperationPaused_Out	BOOL	Statussignal "Paused"
	Variable	Datentyp	Beschreibung																			
	xStartOperation_In	BOOL	Steuersignal zum "Starten"																			
	xStopOperation_In	BOOL	Steuersignal zum "Stoppen"																			
	xPauseOperation_In	BOOL	Steuersignal "Pausieren"																			
	xOperationBusy_Out	BOOL	Statussignal "Busy"																			
	xOperationDone_Out	BOOL	Statussignal "Done"																			
	xOperationPaused_Out	BOOL	Statussignal "Paused"																			
Auf diese Werte kann vom Master-Modul mit "rACDModulName" und im Slave-Modul mit "MACD" zugegriffen werden. Mit der Eingabe "MACD." wird mit Hilfe der Intellisense-Funktion der Inhalt der Struktur angezeigt.																						
Kommunikationskanal einrichten (ACD Slave Access) (36)																						
Kommunikationskanal verwenden (39)																						
MEL_scInfeed	Diese Struktur enthält die Fehlereinträge, die an den Errorset-Baustein geschaltet werden. Diese Fehlereinträge können mit dem Befehl "Edit Errorlist" bearbeitet werden. Fehlermeldungen anlegen und bearbeiten (40)																					
MIO_scInfeed	Diese Struktur enthält IO-Variablen, die extern an das Maschinenmodul angeschlossen werden.																					
MPar_scInfeed	Diese Struktur enthält Parameter zur Parametrierung des Maschinenmoduls.																					
MPD_scInfeed	Diese Struktur enthält persistente Variablen/Parameter.																					
MVis_scInfeed	Diese Struktur enthält Visualisierungs-Variablen, die automatisch in die Symbolkonfiguration geschrieben werden.																					

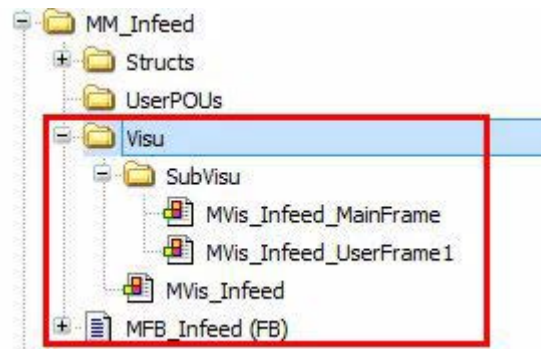
5.6.2 User POUs



In diesem Ordner können Funktionsbausteine, Funktionen etc. abgelegt werden, die speziell im Maschinenmodul verwendet werden

Wird der Ordner nicht benötigt, kann er gelöscht werden.

5.6.3 Visualisierungen



Der Ordner **Visu** enthält die Hauptvisualisierung **MVis_Infeed** sowie die Untervisualisierungen **MVis_Infeed_MainFrame** und **MVis_Infeed_UserFrame1**.

Die Visualisierung **MVis_Infeed_UserFrame1** kann verwendet werden, um eigene Inhalte (z. B. die Visualisierung eines Technologiemoduls) einzufügen.

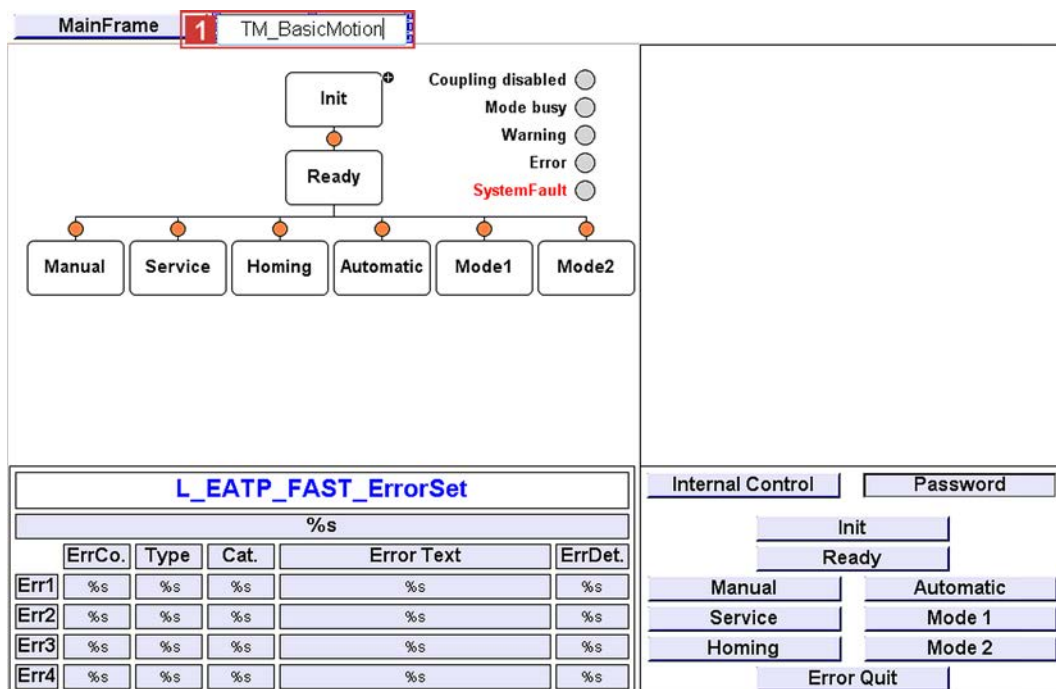
Die Hauptvisualisierung **MVis_Infeed** ist so vorbereitet, dass auf das "User Frame" umgeschaltet werden kann.



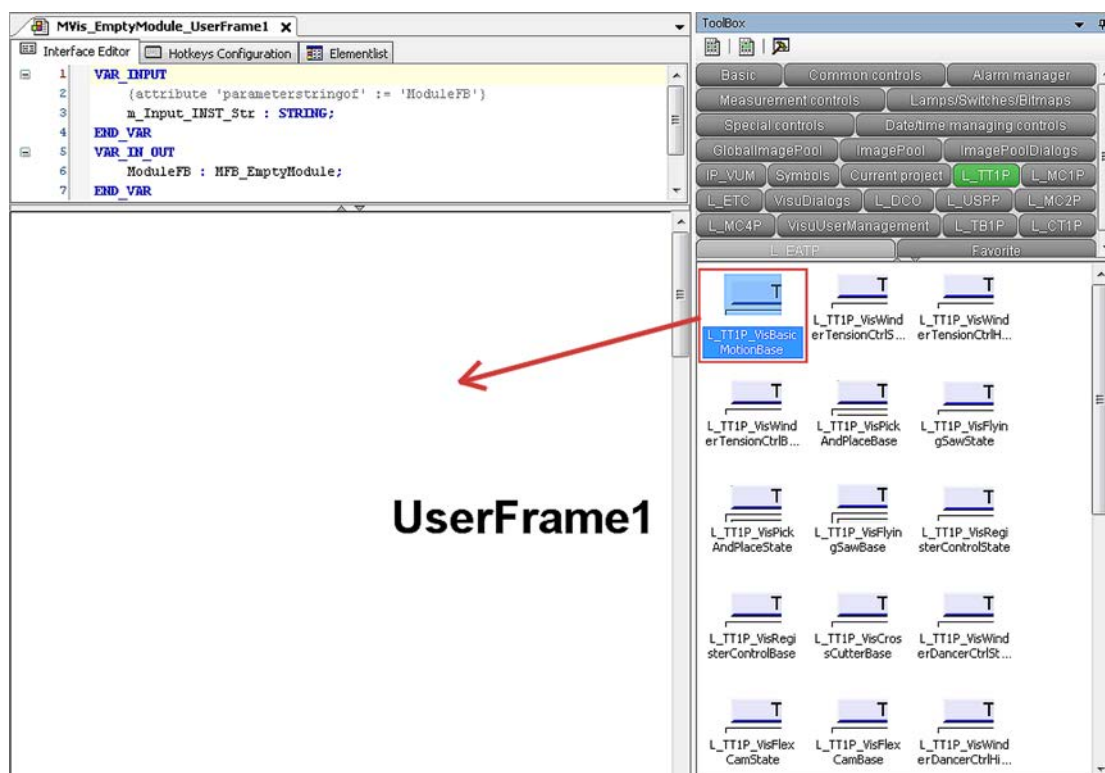
So fügen Sie die Visualisierung eines Technologiemo­duls ein:

1. Die Hauptvisualisierung **MVis_[ModuleName]** mit einem Doppelklick öffnen.
2. Den Namen der Schaltfläche **1** "UserFrame1" umbenennen.

Hier "TM_BasicMotion" für das Technologiemo­dul "Basic Motion":

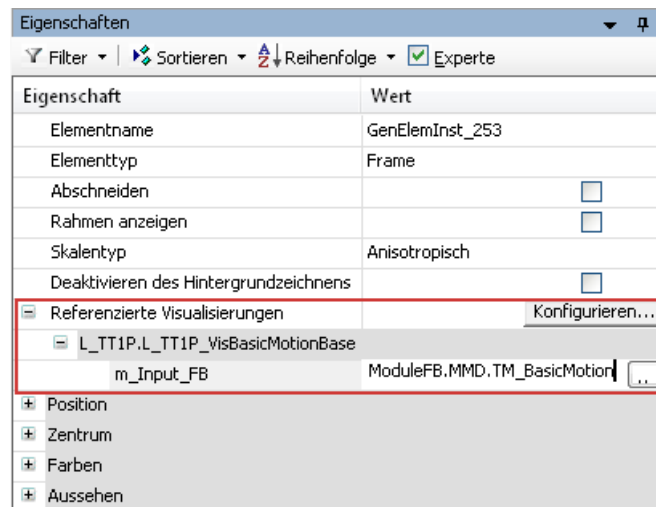


3. Die Visualisierung **MVis_[ModuleName]_UserFrame1** mit einem Doppelklick öffnen.
4. In den Werkzeugen unter **L_TT1P** die Visualisierung des Technologiemo­duls in den "UserFrame1" schieben.

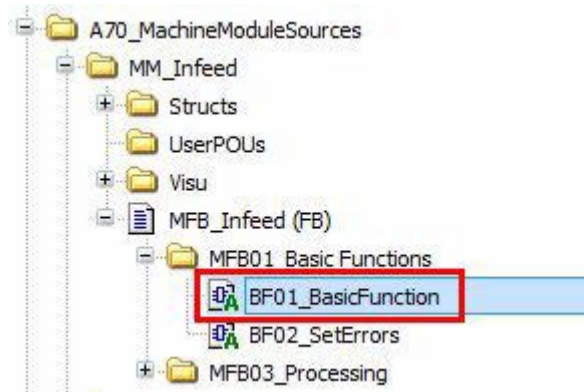


5. Abschließend unter den Eigenschaften der Visualisierung die Referenz der Instanz des Technologiemoduls angeben.

Hier "ModuleFB.MMD.TM_BasicMotion" für das Technologiemodul "Basic Motion":



5.6.4 BF01_BasicFunction



Die Aktion **BF01_BasicFunction** im Modulordner **MFB01_BasicFunctions** enthält die Schnittstelle des Applikationsprogramms in einem Maschinenmodul zu den Application-Template-Systemfunktionen.

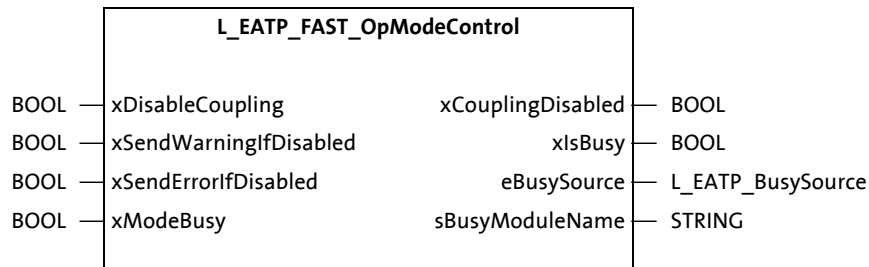


Hinweis!

Alle Eingänge der im Folgenden beschriebenen Schnittstellenbausteine sind mit namensgleichen Datenelementen in der MMD-Struktur beschaltet (vgl. [Strukturen \(57\)](#)). Diese Datenelemente dienen im Applikationsprogramm für die Ansteuerung von Systemfunktionen.

5.6.4.1 L_EATP_FAST_OpModeControl

Instanzen dieses Funktionsbausteins bilden die Schnittstelle zur Konfiguration des Operation-Mode-Modells für Mode-Wechsel ab.



Eingänge

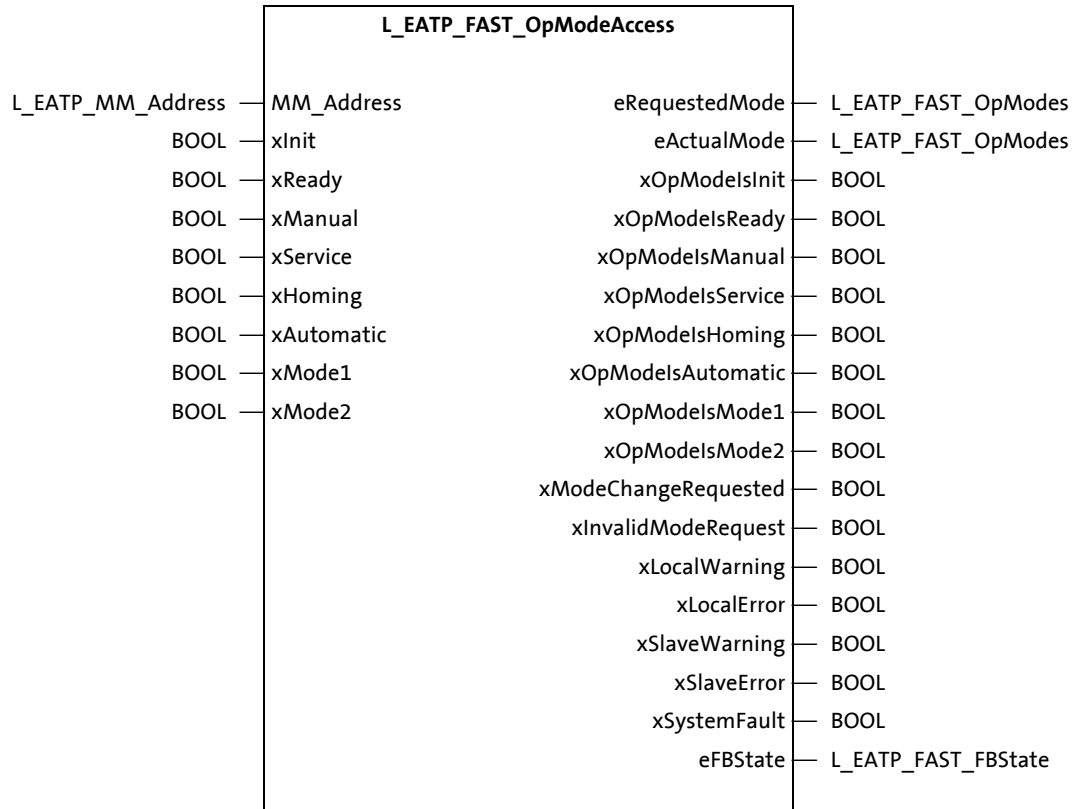
Bezeichner	Datentyp	Beschreibung	
xDisableCoupling	BOOL	FALSE	Die Standardkupplung des Maschinenmoduls zum Master-Modul wird deaktiviert.
		TRUE	Die Aktivierung der Standardkupplung des Maschinenmoduls zum Master-Modul wird freigegeben. Hinweis: Die Kupplung wird erst aktiviert, wenn sich beide Module im gleichen Operation Mode befinden!
xSendWarningIfDisabled	BOOL	TRUE	Wenn die Kupplung des Maschinenmoduls zum Master-Modul deaktiviert ist, werden Warnungen dennoch an das Master-Modul weitergeleitet.
		FALSE	Wenn die Kupplung des Maschinenmoduls zum Master-Modul deaktiviert ist, werden Warnungen nicht an das Master-Modul weitergeleitet.
xSendErrorIfDisabled	BOOL	TRUE	Wenn die Kupplung des Maschinenmoduls zum Master-Modul deaktiviert ist, werden Fehlermeldungen dennoch an das Master-Modul weitergeleitet.
		FALSE	Wenn die Kupplung des Maschinenmoduls zum Master-Modul deaktiviert ist, werden Fehlermeldungen nicht an das Master-Modul weitergeleitet.
xSendErrorIfDisabled	BOOL	TRUE	Das Maschinenmodul befindet sich im Zustand "ModeBusy", d. h. Mode-Wechsel sind gesperrt.
		FALSE	Das Maschinenmodul befindet sich nicht im Zustand "ModeBusy", d. h. Mode-Wechsel sind freigegeben.

Ausgänge

Bezeichner	Datentyp	Beschreibung	
xCouplingDisabled	BOOL	TRUE	Die Standardkupplung des Maschinenmoduls zum Master-Modul ist deaktiviert.
xIsBusy	BOOL	TRUE	Das Maschinenmodul selbst oder einer seiner unterlagerten und gekoppelten Slave-Module befindet sich im Zustand "ModeBusy".
eBusySource L_EATP_BusySource		Eine der folgenden Enumerationskonstanten: <ul style="list-style-type: none"> "NotBusy": Kein Maschinenmodul ist im Zustand "ModeBusy". "OwnModuleIsBusy": Das lokale Maschinenmodul ist im Zustand "ModeBusy". "SlavesBusy": Mindestens ein gekoppeltes Slave-Modul ist im Zustand "ModeBusy". 	
sBusyModuleName	STRING	Instanzname des Maschinenmoduls, das sich im Zustand "ModeBusy" befindet. Hinweis: Sind mehrere Slave-Module im Zustand "ModeBusy", so wird der Name des ersten gefundenen Slave-Moduls in der Aufrufhierarchie angegeben.	

5.6.4.2 L_EATP_FAST_OpModeAccess

Instanzen dieses Funktionsbausteins bilden die Schnittstelle zur Umschaltung der Operation Modes ab.



Eingänge



Hinweis!

Zu jedem Zeitpunkt darf jeweils nur max. ein Bausteineingang *xInit ... xMode2* mit TRUE beschaltet sein.

Bezeichner	Datentyp	Beschreibung	
MM_Address L_EATP_MM_Address		Moduladresse des Zielmoduls Entweder L_EATP_CONST.OWNID für das lokale Maschinenmodul oder eine gültige Adresse eines Slave-Moduls	
xInit	BOOL	TRUE	Ein Wechsel in den Mode "INIT" wird angefordert.
xReady	BOOL	TRUE	Ein Wechsel in den Mode "READY" wird angefordert.
xManual	BOOL	TRUE	Ein Wechsel in den Mode "MANUAL" wird angefordert.
xService	BOOL	TRUE	Ein Wechsel in den Mode "SERVICE" wird angefordert.
xHoming	BOOL	TRUE	Ein Wechsel in den Mode "HOMING" wird angefordert.

Bezeichner	Datentyp	Beschreibung	
xAutomatic	BOOL	TRUE	Ein Wechsel in den Mode "AUTOMATIC" wird angefordert.
xMode1	BOOL	TRUE	Ein Wechsel in den Mode "MODE1" wird angefordert.
xMode2	BOOL	TRUE	Ein Wechsel in den Mode "MODE2" wird angefordert.

Ausgänge

Bezeichner	Datentyp	Beschreibung	
eRequestedMode L_EATP_FAST_OpModes		Der beim adressierten Maschinenmodul angeforderte Mode	
eActualMode L_EATP_FAST_OpModes		Der beim adressierten Maschinenmodul aktive Mode	
xOpModelsInit	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "INIT".
xOpModelsReady	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "READY".
xOpModelsManual	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "MANUAL".
xOpModelsService	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "SERVICE".
xOpModelsHoming	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "HOMING".
xOpModelsAutomatic	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "AUTOMATIC".
xOpModelsMode1	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "MODE1".
xOpModelsMode2	BOOL	TRUE	Das adressierte Modul befindet sich im Mode "MODE2".
xModeChangeRequested	BOOL	TRUE	Ein Mode-Wechsel wurde angefordert.
xInvalidModeRequest	BOOL	TRUE	Eine ungültige Anforderung für einen Mode-Wechsel liegt vor. Mögliche Ursachen: <ul style="list-style-type: none"> • Mehr als ein Eingang von xInit ... xMode2 ist auf TRUE gesetzt. • Das adressierte Modul befindet sich in der Standardkupplung zu seinem Master-Modul. Die Anforderung eines Mode-Wechsels ist hierbei nur beim Master-Modul möglich. • Die FB-Instanz befindet sich im passiven Modus und es liegt eine Mode-Wechsel-Anforderung vor (siehe auch Ausgang eFBState).
xLocalWarning	BOOL	TRUE	Am adressierten Modul ist eine Warnung aktiv.
xLocalError	BOOL	TRUE	Am adressierten Modul ist eine Fehlermeldung aktiv.
xSlaveWarning	BOOL	TRUE	An einem unterlagerten Slave-Modul ist eine Warnung aktiv.
xSlaveError	BOOL	TRUE	An einem unterlagerten Slave-Modul ist eine Fehlermeldung aktiv.

Bezeichner	Datentyp	Beschreibung	
xSystemFault	BOOL	TRUE	Der Zustand "SystemFault" ist aktiv.
eFBState L_EATP_FAST_FBState		<p>Eine der folgenden Enumerationskonstanten:</p> <ul style="list-style-type: none"> • "FullControl": Die FB-Instanz hat den vollen Funktionsumfang. • "PassiveInternalControl": Die FB-Instanz ist im Passivmodus (die Bausteineingänge sind deaktiviert), weil beim Maschinenmodul "InternalControl" aktiv ist. • "PassiveCoupledToMaster": Die FB-Instanz ist im Passivmodus, weil das Maschinenmodul ein steuerndes Master-Modul hat. • "PassiveSecondInstance": Die FB-Instanz ist im Passivmodus, weil sie eine Zweit- oder Folgeinstanz für das adressierte Modul ist. • "PassiveMasterOverride": Die Kupplung des Moduls zu seinem Master-Modul ist deaktiviert. Eine Bausteininstanz ist aber im Master-Modul enthalten, die die Mode-Steuerung ausführt. 	

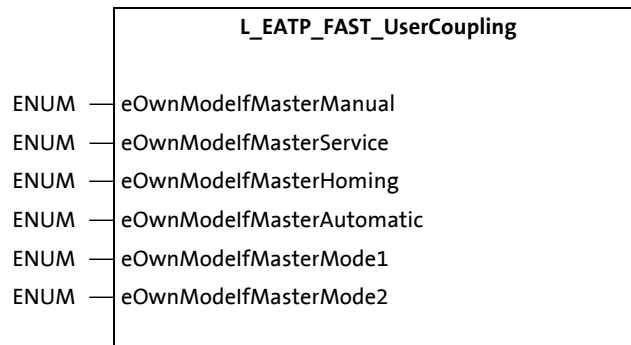
5.6.4.3 L_EATP_FAST_UserCoupling

Instanzen dieses Funktionsbausteins ermöglichen es, bei bestimmten Operation Modes (abhängig vom Mode des Master-Moduls) einen anderen Mode einzunehmen.



Hinweis!

- Die Vorbelegung der Datenelemente in der MMD-Struktur ist so gewählt, dass beim Slave-Modul immer der Operation Mode des Master-Moduls gewählt wird.
- Die Operation Modes "INIT" und "READY" des Master-Moduls werden immer übernommen und können nicht umgeschaltet werden.



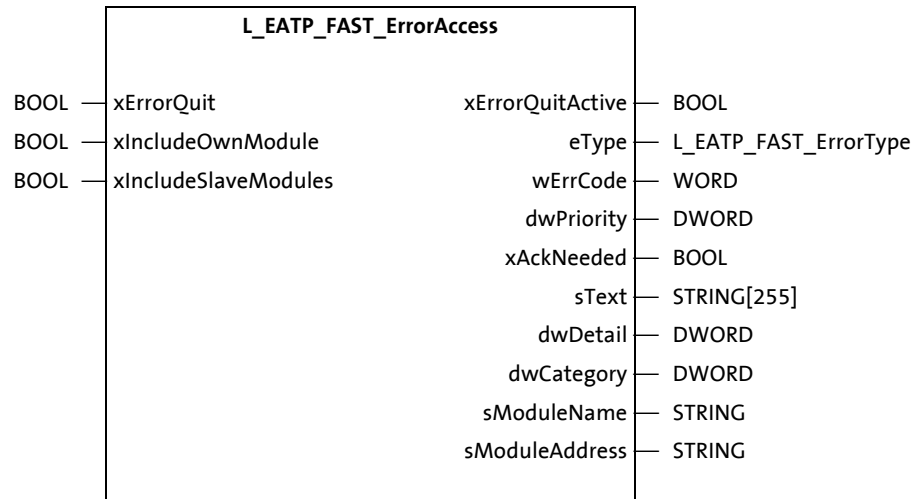
Eingänge

Bezeichner	Datentyp	Beschreibung
eOwnModelfMasterManual	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "MANUAL" befindet.
eOwnModelfMasterService	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "SERVICE" befindet.
eOwnModelfMasterHoming	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "HOMING" befindet.
eOwnModelfMasterAutomatic	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "AUTOMATIC" befindet.
eOwnModelfMasterMode1	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "MODE1" befindet.
eOwnModelfMasterMode2	ENUM	Festlegung des Operation Modes des lokalen Maschinenmoduls, wenn sich das Master-Modul im Mode "MODE2" befindet.

5.6.4.4 L_EATP_FAST_ErrorAccess

Instanzen dieses Funktionsbausteins bilden die folgenden Funktionen ab:

- Fehlerquittierung
- Ermittlung der Fehlerinformationen zum aktuell höchstpriorien Fehler des lokalen Maschinenmoduls und/oder seiner untergeordneten Slave-Module



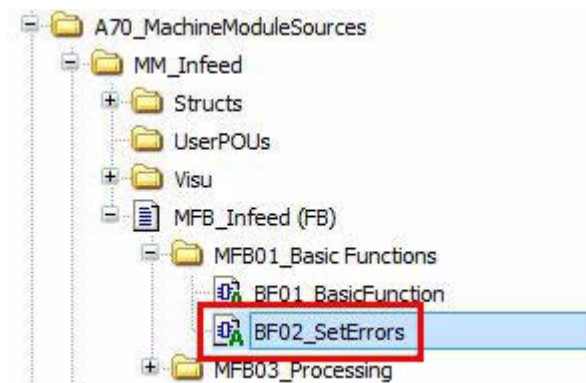
Eingänge

Bezeichner	Datentyp	Beschreibung	
xErrorQuit	BOOL	FALSE	Die Fehler am eigenen Maschinenmodul und an allen untergeordneten Slave-Modulen werden quittiert.
xIncludeOwnModule	BOOL	TRUE	Das lokale Maschinenmodul wird bei der Ermittlung des höchstpriorien Fehlers berücksichtigt.
		FALSE	Das lokale Maschinenmodul wird bei der Ermittlung des höchstpriorien Fehlers ignoriert.
xIncludeSlaveModules	BOOL	TRUE	Die untergeordneten Slave-Module werden bei der Ermittlung des höchstpriorien Fehlers berücksichtigt.
		FALSE	Die untergeordneten Slave-Module werden bei der Ermittlung des höchstpriorien Fehlers ignoriert.

Ausgänge

Bezeichner	Datentyp	Beschreibung	
xErrorQuitActive	BOOL	FALSE	Die Fehlerquittierung ist aktiv.
		TRUE	Hinweis: Dieses Signal sollte verwendet werden zur Fehlerquittierung lokaler FB-Instanzen.
eType		Fehlertyp des aktuell höchstpriorien Fehlers	
L_EATP_FAST_ErrorType			
wErrCode	WORD	Fehlernummer des aktuell höchstpriorien Fehlers	
dwPriority	DWORD	Fehlerpriorität des aktuell höchstpriorien Fehlers	
xAckNeeded	BOOL	Quittierpflichtigkeit des aktuell höchstpriorien Fehlers	
		TRUE	Quittierung erforderlich
		FALSE	Quittierung nicht erforderlich
sText	STRING[255]	Fehlertext des aktuell höchstpriorien Fehlers	
dwDetail	DWORD	Fehlerdetail des aktuell höchstpriorien Fehlers	
dwCategory	DWORD	Fehlerkategorie des aktuell höchstpriorien Fehlers	
sModuleName	STRING	Instanzname des Maschinenmoduls, bei dem der aktuell höchstpriorie Fehler aktiv ist.	
sModuleAddress	STRING	Globale Adresse des Maschinenmoduls, bei dem der aktuell höchstpriorie Fehler aktiv ist.	

5.6.5 BF02_SetErrors



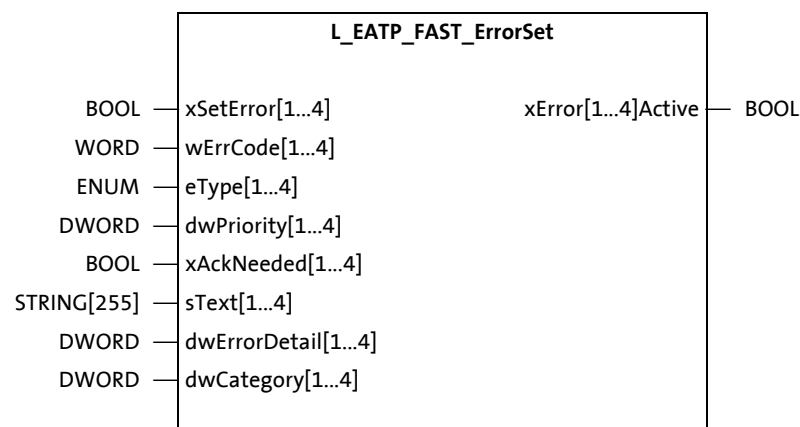
Die Aktion **BF02_SetErrors** im Modulordner **MFB01_BasicFunctions** enthält die vordefinierten Instanzen "ErrorsA" und "ErrorsB" des Funktionsbausteins **L_EATP_FAST_ErrorSet** für die Fehlerbehandlung und die Auslösung von 8 Fehlermeldungen.

Beim Auftreten des Fehlers wird die zugehörige Fehlermeldung in das Logbuch eingetragen.

► [A12_Configuration](#) (📖 50)

Eine direkte Programmierung oder Konfigurierung erfolgt i.d.R. nicht in dieser Aktion, sondern mit Hilfe des Befehls "Edit Errorlist".

► [Fehlermeldungen anlegen und bearbeiten](#) (📖 40)



Hinweis!

Die Eingänge sind mit namensgleichen Datenelementen in der MEL-Struktur beschaltet.

Die Ausgänge sind mit namensgleichen Datenelementen in der MVis-Struktur beschaltet.

► [Strukturen](#) (📖 57)

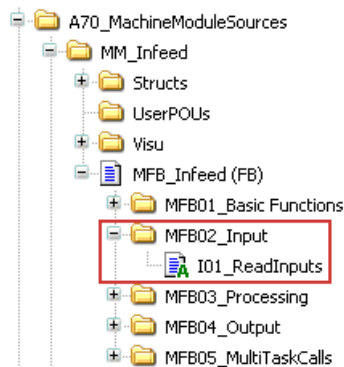
Eingänge

Bezeichner	Datentyp	Beschreibung	
xSetError[1...4]	BOOL	FALSE	Ein Fehler ist aktiv.
wErrCode[1...4]	WORD	Fehlernummer des aktuell höchstpriorien Fehlers	
eType[1...4] L_EATP_FAST_ErrorType		Fehlertyp des aktuell höchstpriorien Fehlers	
dwPriority[1...4]	DWORD	Fehlerpriorität des aktuell höchstpriorien Fehlers	
xAckNeeded[1...4]	BOOL	Quittierpflichtigkeit des aktuell höchstpriorien Fehlers	
		TRUE	Quittierung erforderlich
		FALSE	Quittierung nicht erforderlich
sText[1...4]	STRING[255]	Fehlertext des aktuell höchstpriorien Fehlers	
dwDetail[1...4]	DWORD	Fehlerdetail des aktuell höchstpriorien Fehlers	
dwCategory[1...4]	DWORD	Fehlerkategorie des aktuell höchstpriorien Fehlers	

Ausgänge

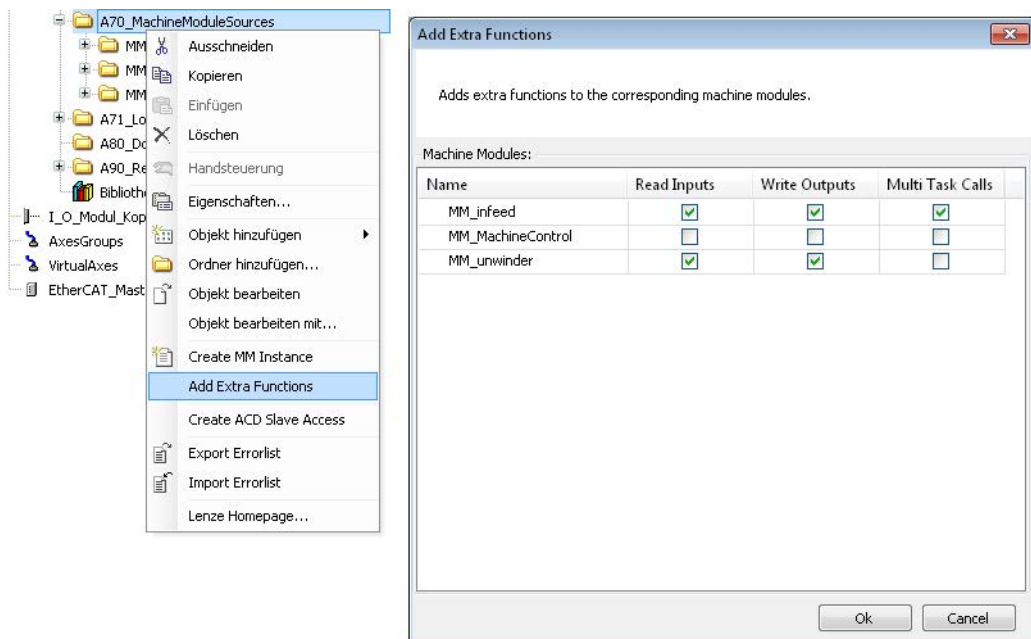
Bezeichner	Datentyp	Beschreibung	
xError[1...4]Active	BOOL	TRUE	Ausgabe an die MVis-Struktur (Variable <i>dwErrorActive</i>), dass ein Fehler aktiv ist.

5.6.6 I01_ReadInputs

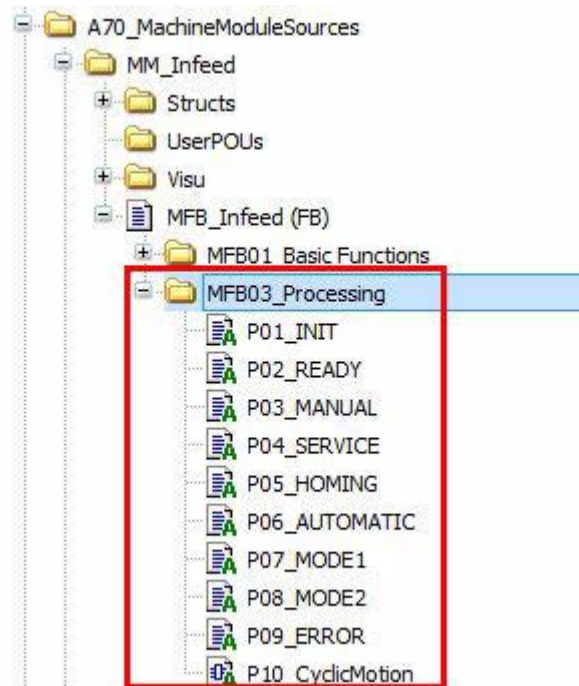


Die Aktion **I01_ReadInputs** im Modulordner **MFB02_Input** dient als Strukturierungshilfsmittel bei der Verwendung des sogenannten EVA-Prinzips (Eingabe – Verarbeitung – Ausgabe). Eine detaillierte Beschreibung des EVA-Prinzips ist im Abschnitt [MFB03 Processing](#) (72) enthalten.

Der Modulordner **MFB02_Input** ist zunächst im Projektbaum nicht sichtbar. Er kann bei Bedarf, wenn das EVA-Prinzip bei der Applikationsprogrammierung verwendet werden soll, mit dem Befehl "Add Extra Functions" unter **A70_MachineModuleSources** ein- und ausgeblendet werden. Dazu können im erscheinenden Dialog unter "Read Inputs" Häkchen gesetzt oder entfernt werden.



5.6.7 MFB03_Processing



Dieser Modulordner fasst die maßgeblichen Aktionen zusammen, die für die Festlegung des Applikationsprogramms vorgesehen sind. Die Aktionen gliedern sich in ...

- [Mode-bezogene Aktionen](#) (73) "P01_INIT" ... "P08_MODE2",
- [Fehler-Aktion](#) (73) "P09_ERROR",
- [Zyklische Aktion](#) (73) "P10_CyclicMotion".

5.6.7.1 Mode-bezogene Aktionen

In jedem Programmzyklus wird abhängig vom aktiven Mode eines Maschinenmoduls die jeweils zugehörige Mode-Aktion **P01_INIT** ... **P08_MODE2** durchlaufen.

In jeder Mode-Aktion ist bereits ein vordefiniertes Programmgerüst vorhanden, in dem der Programmcode ergänzt werden kann.

Es gibt drei Bereiche , in denen innerhalb eines Modes Programmcode eingefügt werden kann:

- 1** "ModeEntry" wird beim Eintritt in den Mode für einen Zyklus ausgeführt.
- 2** "Cyclic Area" wird solange ausgeführt, bis der Mode umgeschaltet wird.
- 3** "ModeExit" wird beim Verlassen des Modes ausgeführt.

```

1  (* AT_ACTION_CREATE_NEW_MODULE *)
2  // First pulse, last pulse and cyclic program
3  1 IF xModeEntry THEN
4      // Do init steps
5
6      // Set ModeBusy
7      MMD.xModeBusy := TRUE;
8
9  3 ELIF xModeExit THEN
10     // Do exit steps
11
12  2 ELSE
13     // Do cyclic things
14
15     // Reset own ModeBusy
16     MMD.xModeBusy := FALSE;
17
18     // Check if nobody is busy
19     IF NOT OpModeControl.xIsBusy THEN
20         // No module is busy
21         ;
22     END_IF
23 END_IF
24

```

► [Operation-Modes verwenden](#) (📖 37)

5.6.7.2 Fehler-Aktion

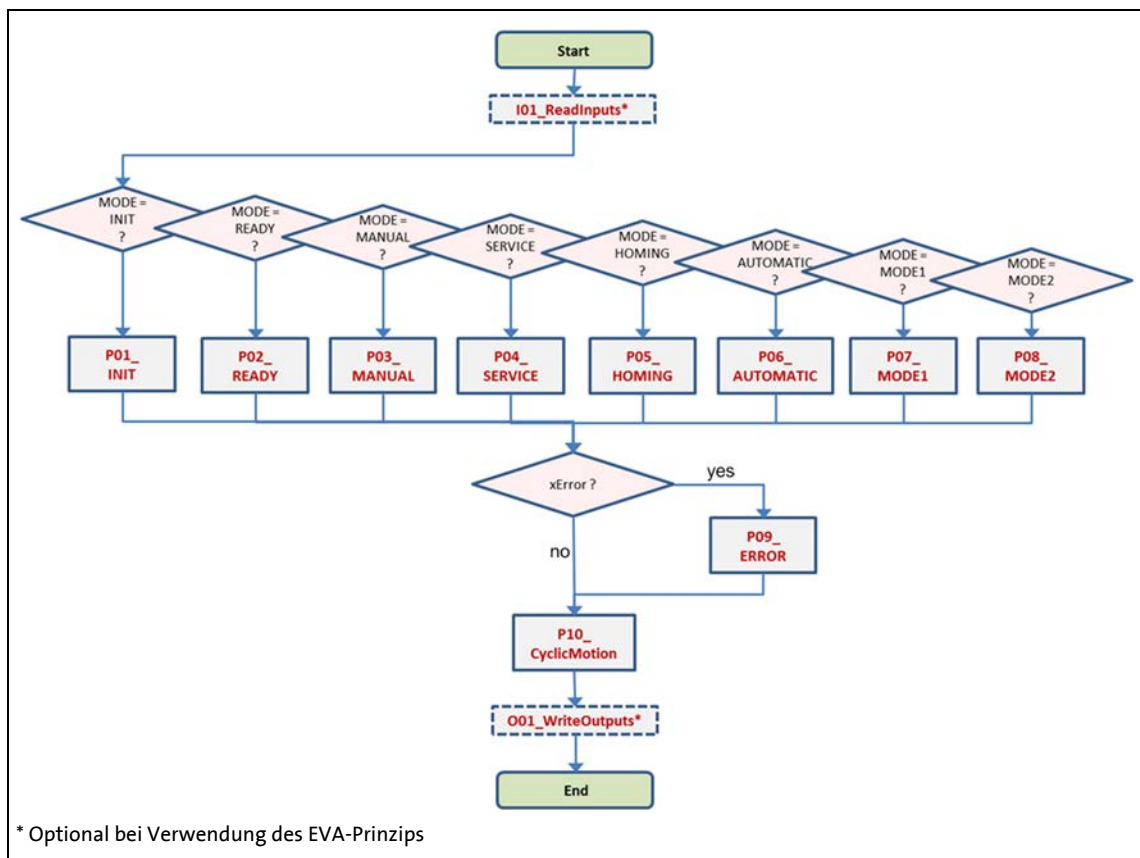
Solange im Maschinenmodul ein Fehler aktiv ist, erfolgt der Aufruf der Aktion **P09_ERROR** in jedem Programmzyklus.

Bei aktiven Warnungen wird diese Aktion nicht durchlaufen.

5.6.7.3 Zyklische Aktion

Die Aktion **P10_CyclicMotion** wird in jedem Programmzyklus der Task "HighPriority" genau einmal aufgerufen. Sie dient u. a. dazu, Aufrufe von Motion-Bausteinen und weiteren Applikationsteilen aufzunehmen.

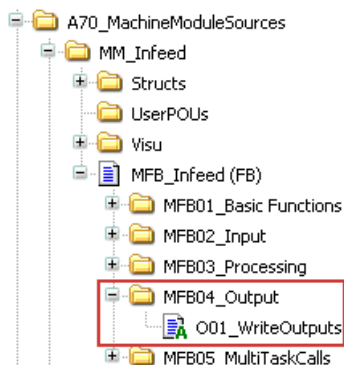
5.6.7.4 Ablauf von Aktionen



[5-3] Ablauf: Programmzyklus im Maschinenmodul

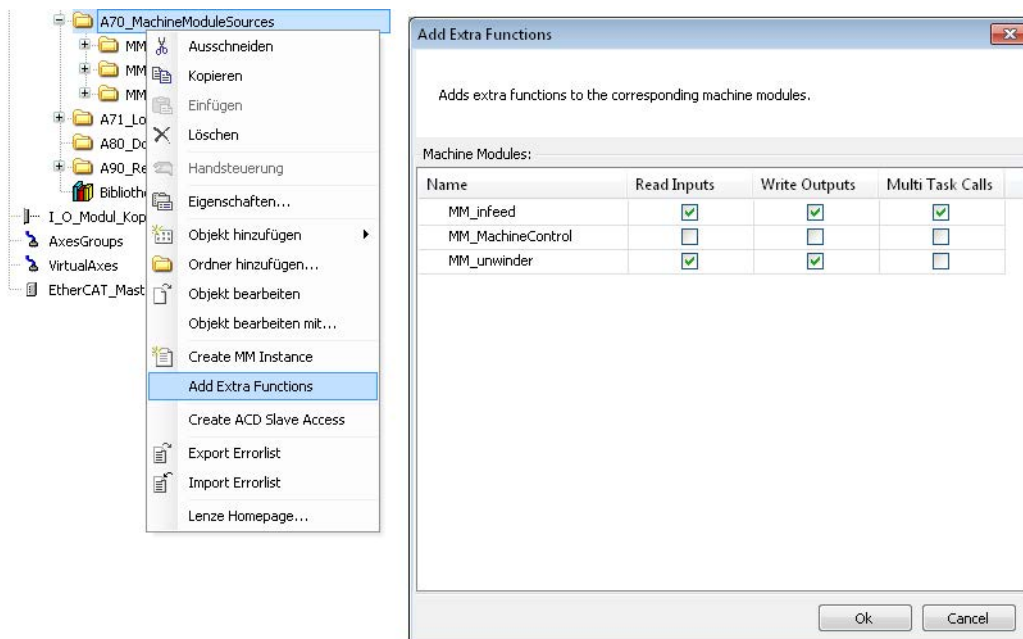
Die Abbildung [5-3] zeigt den gesamten Ablauf eines Programmzyklus der Task "HighPriority" mit den optionalen Aktionen [I01_ReadInputs](#) (71) und [O01_WriteOutputs](#) (75) für die Verwendung des EVA-Prinzips.

5.6.8 001_WriteOutputs

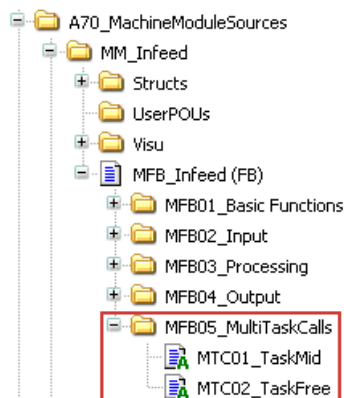


Die Aktion **001_WriteOutputs** im Modulordner **MFB04_Output** dient als Strukturierungshilfsmittel bei der Verwendung des sogenannten EVA-Prinzips (Eingabe – Verarbeitung – Ausgabe). Eine detaillierte Beschreibung des EVA-Prinzips ist im Abschnitt [MFB03 Processing](#) (72) enthalten.

Der Modulordner **MFB04_Output** ist zunächst im Projektbaum nicht sichtbar. Er kann bei Bedarf, wenn das EVA-Prinzip bei der Applikationsprogrammierung verwendet werden soll, mit dem Befehl "Add Extra Functions" unter **A70_MachineModuleSources** ein- und ausgeblendet werden. Dazu können im erscheinenden Dialog unter "Write Outputs" Häkchen gesetzt oder entfernt werden.



5.6.9 MTC01_TaskMid / MTC02_TaskFree



Im Normalfall kann das gesamte Applikationsprogramm als Single-Task-Lösung im Task-Kontext "HighPriority" in den vordefinierten Aktionen des Application Template programmiert werden.

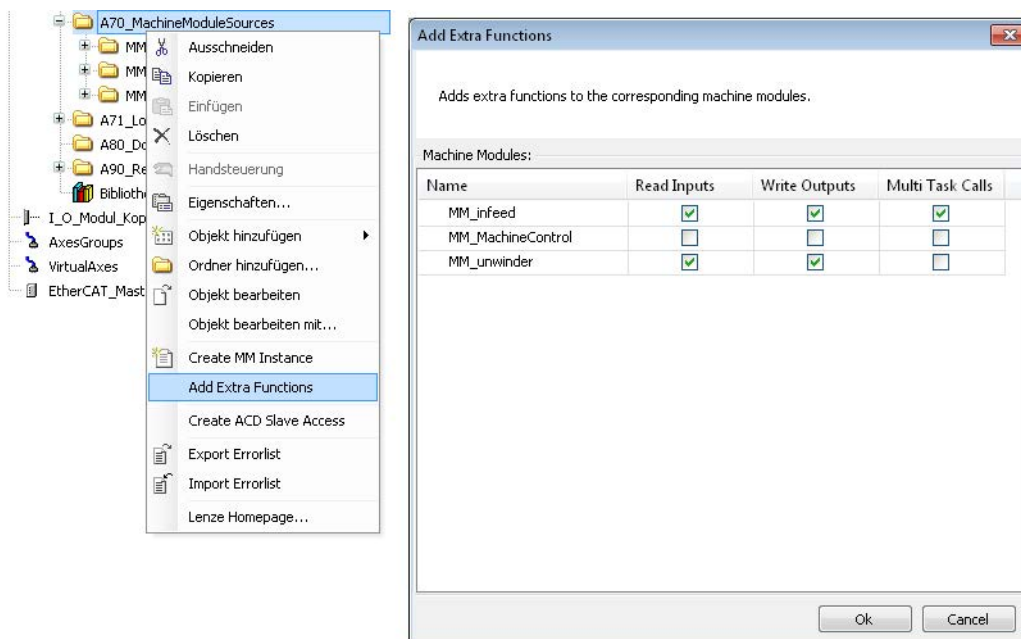
Für den Ausnahmefall, dass Teile der Modulapplikation in den Task-Kontexten "MidPriority" und/oder "Freewheeling" aufgerufen werden müssen, stehen im Modulordner **MFB05_MultiTaskCalls** die Aktionen **MTC01_TaskMid** und **MTC02_TaskFree** zur Verfügung.



Hinweis!

Enthält das Applikationsprogramm Task-übergreifende Datenzugriffe, so müssen entsprechende Maßnahmen zur Sicherstellung der Datenkonsistenz vorgesehen werden.

Der Modulordner **MFB05_MultiTaskCalls** ist zunächst im Projektbaum nicht sichtbar. Er kann bei Bedarf, wenn das EVA-Prinzip bei der Applikationsprogrammierung verwendet werden soll, mit dem Befehl **"Add Extra Functions"** unter **A70_MachineModuleSources** ein- und ausgeblendet werden. Dazu können im erscheinenden Dialog unter "Multi Task Calls" Häkchen gesetzt oder entfernt werden.



A

A10_MachineModuleTree [48](#)
A12_Configuration [50](#)
A20_Visualisation [51](#)
A55_VarLists [55](#)
A60_MotionObjects [55](#)
A70_MachineModuleSources [56](#)
Ablauf von Aktionen [74](#)
ACD Slave Access [36](#)
ACD-Kanal [39](#)
Achsen verbinden [34](#)
Add Extra Functions (Befehl) [71](#), [75](#), [76](#)
Aktionen (Ablauf) [74](#)
Anschlüsse zurücksetzen (Befehl) [35](#)
Anwendungsbeispiele [6](#)
Anwendungshinweise (Darstellung) [9](#)
Application Template öffnen [15](#)
Application Template Struktur [47](#)

B

Befehl
 Add Extra Functions [71](#), [75](#), [76](#)
 Anschlüsse zurücksetzen [35](#)
 Create ACD Slave Access [36](#)
 Create MM Instance [22](#)
 Edit Errorlist [40](#)
 Export Errorlist [42](#)
 Import Errorlist [43](#)
 Insert Machine Module Instance [24](#)
Begriffe [8](#)
BF01_BasicFunction [61](#)
BF02_SetErrors [69](#)

C

Copy Machine Module (Befehl) [20](#)
Create ACD Slave Access (Befehl) [36](#)
Create MM Instance (Befehl) [22](#)

D

Delete Machine Module References (Befehl) [27](#)
Dokumenthistorie [6](#)

E

Edit Errorlist (Befehl) [40](#)
E-Mail an Lenze [79](#)
Export Errorlist (Befehl) [42](#)

F

FAST Technologiemodule einfügen [31](#)
Feedback an Lenze [79](#)
Fehler-Aktion [73](#)
Fehlerliste exportieren [42](#)
Fehlerliste importieren [43](#)

Fehlermeldungen anlegen und bearbeiten [40](#)

G

Gültigkeit der Dokumentation [6](#)

I

I01_ReadInputs [71](#)
Import Errorlist (Befehl) [43](#)
Insert Machine Module (Befehl) [21](#)
Insert Machine Module Instance (Befehl) [24](#)

K

Kommunikationskanal (ACD) einrichten [36](#)
Kommunikationskanal (ACD) verwenden [39](#)
Konfiguration des Application Template [50](#)

L

L_EATP_FAST_Config [50](#)
L_EATP_FAST_ErrorAccess [67](#)
L_EATP_FAST_ErrorSet [69](#)
L_EATP_FAST_OpModeAccess [63](#)
L_EATP_FAST_OpModeControl [62](#)
L_EATP_FAST_UserCoupling [66](#)
L_EATP_FAST_VisErrorList [52](#)
L_EATP_FAST_VisModuleDetail [54](#)
L_EATP_FAST_VisModuleList [53](#)
L_Main [51](#)
Load Machine Module (Befehl) [16](#)

M

Machine Module Tree (MMT) [48](#)
Maschinenmodul als Vorlage speichern [19](#)
Maschinenmodul kopieren [20](#)
Maschinenmodul umbenennen [23](#)
Maschinenmodulbaum [48](#)
Maschinenmodule anlegen [16](#)
Maschinenmodule einbinden [24](#)
Maschinenmodule löschen [28](#)
Maschinenmodulinstanzen erstellen [22](#)
Maschinenmodul-Referenzen löschen [27](#)
MFB01_BasicFunctions [61](#), [69](#)
MFB02_Input [71](#)
MFB03_Processing [72](#)
MFB04_Output [75](#)
MFB05_MultiTaskCalls [76](#)
MM_EmptyModule (Befehl) [17](#)
Mode-bezogene Aktionen [73](#)
Modes verwenden [37](#)
Modulkopplung verwenden [45](#)
MTC01_TaskMid [76](#)
MTC02_TaskFree [76](#)

Index

O

O01_WriteOutputs [75](#)

Operation Modes verwenden [37](#)

P

P01_INIT [72](#)

P02_READY [72](#)

P03_MANUAL [72](#)

P04_Service [72](#)

P05_HOMING [72](#)

P06_AUTOMATIC [72](#)

P07_MODE1 [72](#)

P08_MODE2 [72](#)

P09_ERROR [72](#)

P10_CyclicMotion [72](#)

Programmierung mit dem Application Template [13](#)

Programmzyklus im Maschinenmodul [74](#)

R

Rename Machine Module (Befehl) [23](#)

S

Save Machine Module (Befehl) [19](#)

Screenshots [6](#)

Sicherheitshinweise [10](#)

Sicherheitshinweise (Darstellung) [9](#)

Slave Access (ACD) [36](#)

Struktur des Application Template [47](#)

Strukturen [57](#)

Strukturierte Programmierung [13](#)

Systemvoraussetzungen [12](#)

U

User POU's [58](#)

V

Variablenlisten [55](#)

Verwendete Hinweise [9](#)

Verwendete Konventionen [7](#)

Verwendung des Kommunikationskanals (ACD) [39](#)

Visualisierung einbinden und verbinden [29](#)

Visualisierung L_Main [51](#)

Visualisierungen [58](#)

Z

Zielgruppe [6](#)

Zyklische Aktion [73](#)



Ihre Meinung ist uns wichtig

Wir erstellen diese Anleitung nach bestem Wissen mit dem Ziel, Sie bestmöglich beim Umgang mit unserem Produkt zu unterstützen.

Vielleicht ist uns das nicht überall gelungen. Wenn Sie das feststellen sollten, senden Sie uns Ihre Anregungen und Ihre Kritik in einer kurzen E-Mail an:

feedback-docu@lenze.com

Vielen Dank für Ihre Unterstützung.

Ihr Lenze-Dokumentationsteam

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
☎ +49 5154 82-0
📠 +49 5154 82-2800
✉ sales.de@lenze.com
🌐 www.lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
GERMANY
☎ 008000 24 46877 (24 h helpline)
📠 +49 5154 82-1112
✉ service.de@lenze.com