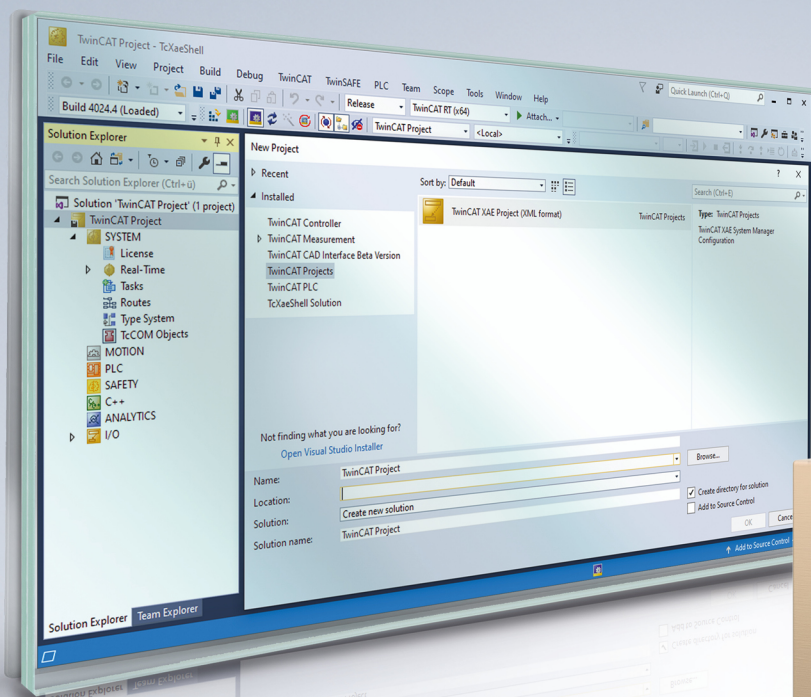


# BECKHOFF New Automation Technology

Handbuch | DE

## TF85xx

TwinCAT 3 | Plastic Application





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort.....</b>	<b>7</b>
1.1	Hinweise zur Dokumentation .....	7
1.2	Zu Ihrer Sicherheit.....	8
1.3	Hinweise zur Informationssicherheit .....	9
<b>2</b>	<b>Einleitung.....</b>	<b>10</b>
<b>3</b>	<b>Version der dokumentierten Software .....</b>	<b>12</b>
<b>4</b>	<b>PLC Konzept und Strategie.....</b>	<b>13</b>
4.1	HMI-Schnittstelle .....	13
4.2	Änderung und Erweiterung durch Vererbung.....	14
4.3	Nutzung von Schnittstellen.....	15
4.4	Initialisierung und Zyklusmethoden.....	15
4.5	Datenhandling .....	17
4.6	Projektaufbau .....	18
<b>5</b>	<b>PLC-API (Klassenübersicht) .....</b>	<b>24</b>
5.1	Base - Basisklassen .....	24
5.1.1	FB_Base .....	24
5.1.2	FB_BaseHmi .....	25
5.1.3	FB_BaseMd .....	26
5.1.4	FB_BaseMdHmi .....	27
5.2	Runtime - Handling von Initialisierung und Zyklusaufrufen .....	27
5.2.1	FB_BaseRuntime .....	28
5.2.2	I_RuntimeInterface.....	30
5.3	Axis - Allgemeine Achstypen.....	30
5.3.1	FB_Axis.....	30
5.3.2	FB_PtpMotion .....	37
5.3.3	FB_PtpMotionHmi .....	39
5.3.4	FB_BaseCammingHmi.....	42
5.3.5	FB_Extruder .....	43
5.4	Clamp, Carriage - Standard PTP-Achsen .....	45
5.4.1	FB_Clamp .....	45
5.4.2	FB_Carriage.....	46
5.5	ManualFunction - Handfunktionen .....	47
5.5.1	FB_ManualFunctionHmi.....	47
5.5.2	FB_ManualPower.....	49
5.5.3	FB_ManualPtp .....	49
5.5.4	FB_ManualTurnrate .....	50
5.6	Temperature - TF8540 Temperature Interface .....	51
5.6.1	FB_Temperature .....	51
5.6.2	FB_TemperatureHmi.....	57
5.6.3	FB_TemperatureGroup .....	59
5.6.4	FB_TemperatureGroupHmi.....	63
5.6.5	FB_TempChannelBase.....	66
5.6.6	FB_TempChannel .....	67

5.6.7	FB_TempChannelHmi.....	69
5.6.8	FB_TempRecipe .....	69
5.6.9	FB_TempSupply .....	70
5.6.10	FB_TempSupplyLine.....	71
5.6.11	FB_TempSchedule .....	71
5.6.12	FB_TempScheduleHmi.....	72
5.6.13	FB_TimerTempHmi.....	73
5.7	Recipe - Rezeptverwaltung .....	74
5.7.1	FB_PlcStateToHmi.....	74
5.7.2	I_RecipeState.....	76
5.7.3	FB_Recipe .....	76
5.8	MachineData - Maschinendaten .....	77
5.8.1	FB_MdBaseContainer.....	79
5.8.2	FB_MdBaseComponent.....	81
5.8.3	FB_MdCollection.....	82
5.9	OperationData - Produktionsdaten und Statistiken .....	83
5.9.1	FB_ProductionCounter.....	83
5.9.2	FB_ProductionCounterComponent.....	83
5.10	EventLogger - Logging von Ereignissen und Fehlern .....	84
5.10.1	FB_AlarmHandler.....	84
5.10.2	FB_AppMessage.....	85
5.11	Analog - Analogwertskalierung .....	86
5.11.1	FB_ScaleAnalogHmi .....	86
5.11.2	FB_Monitoring.....	87
5.11.3	FB_Setpoints.....	89
5.12	Timer - Prozess timing .....	90
5.12.1	FB_TimerHmi.....	90
5.12.2	FB_TimerTon .....	90
5.12.3	FB_TimerTof .....	91
5.12.4	FB_TimerTp .....	92
5.12.5	FB_TimerWeekdayHmi.....	93
5.12.6	FB_TimerWeekdayMaster .....	94
5.13	BlowMolding - Blasformspezifische Klassen.....	95
5.13.1	FB_Blowing.....	95
5.13.2	FB_IntervalBlowing .....	97
5.13.3	FB_Blowpin.....	98
5.13.4	FB_BlowMoldingExtruder.....	99
5.13.5	FB_Wtc .....	100
5.13.6	FB_WtcTimeMaster .....	102
5.13.7	FB_WtcAccuMaster .....	102
5.14	Utilities - Hilfsmittel.....	103
5.14.1	FB_AdaptableSequence .....	103
5.14.2	FB_FlexValue.....	109
5.14.3	FB_Parameter.....	109
5.14.4	FB_TableGeneratorAsciiFile.....	110
5.14.5	FB_TrendHmi.....	111

5.14.6	FB_Trigger .....	116
5.14.7	FB_LatchTrigger .....	116
5.14.8	FB_LibVersion.....	117
5.14.9	FB_LibVersionBeta .....	118
5.14.10	F_SecondsToTime() .....	119
5.14.11	F_TimeToSeconds() .....	119
5.14.12	F_GetCycleTime().....	120
5.14.13	F_TryDivide() .....	120
5.14.14	F_GetLocalSystemtime .....	121
5.14.15	F_GetLocalSystemtimeEx.....	121
5.14.16	PlasticStatusHmi .....	122
5.15	Einstellparameter – Tc3_PlasticBaseAppStaticParams.....	122
<b>6</b>	<b>PLC Beispiele / Anleitungen .....</b>	<b>124</b>
6.1	Allgemein .....	124
6.1.1	Neues TwinCAT Projekt aufsetzen .....	124
6.1.2	Leeres Projekt aufsetzen / bestehendes Projekt erweitern.....	125
6.1.3	Plastic Base Application nachträglich im Projekt aktualisieren .....	126
6.2	Objektorientierung.....	128
6.2.1	Eine Variable einer Klasse (FB) hinzufügen .....	128
6.2.2	Eine Eigenschaft oder Methode einer Klasse (FB) hinzufügen.....	128
6.2.3	Innere Prozeduren einer Klasse (FB) anpassen .....	132
6.2.4	Erweitern der HMI-Parallelklasse (FB).....	133
6.3	Achsen .....	135
6.3.1	NC-Achse anlegen und initialisieren .....	135
6.3.2	NC-Transformationsachse anlegen und initialisieren.....	137
6.3.3	Handfunktion in eine Achse integrieren .....	140
6.4	Datenverwaltung .....	141
6.4.1	Maschinendaten erzeugen.....	141
6.4.2	Rezeptfreigabe integrieren.....	141
6.5	Temperaturregelung.....	141
6.5.1	Temperaturregelung instanzieren und initiieren .....	141
6.5.2	Mapping und Konfiguration von Temperaturzonen .....	142
6.5.3	Temperaturregelung in Betrieb nehmen .....	143
<b>7</b>	<b>HMI Projektaufbau.....</b>	<b>147</b>
7.1	References .....	147
7.2	Themes .....	148
7.3	Contents .....	149
7.3.1	Startseite .....	155
7.3.2	Navigation .....	158
7.3.3	Info .....	159
7.3.4	Handfunktionen .....	160
7.3.5	Achsen .....	161
7.3.6	Extruder.....	166
7.3.7	Parameter .....	170
7.3.8	Prozess .....	175

7.3.9	System .....	180
7.3.10	Temperatur.....	183
7.3.11	WDS.....	194
7.4	Localization .....	198
7.5	View .....	198
<b>8</b>	<b>Anhang .....</b>	<b>202</b>
8.1	Temperaturregelung in Betrieb nehmen .....	202
8.2	Erzeugung und Nutzung des ZonelmaageLayoutConfig-Serversymbols .....	206
8.3	PLC-API (obsolet) .....	207
8.3.1	F_TryDevide() .....	208
8.3.2	FB_TrafoTableGenerator .....	208
8.3.3	FB_MonitoringZone.....	215
8.3.4	FB_TempCtrl.....	216
8.3.5	FB_TempCtrlHmi .....	223
8.3.6	FB_TempGroup .....	225
8.3.7	FB_TempGroupHmi .....	227
8.3.8	FB_TempGroupOpModeHmi .....	228
8.3.9	FB_TempZone .....	229
8.3.10	FB_TempZoneHmi.....	231

# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.  
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

### Warnungen vor Personenschäden

#### **GEFAHR**

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

#### **WARNUNG**

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

#### **VORSICHT**

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

### Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:  
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.



## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Einleitung

Die TwinCAT 3 Plastic Application repräsentiert die dritte Implementierungsebene der Branchenlösung für Kunststoffmaschinen von Beckhoff Automation. In dieser Ebene werden die technologiespezifischen Funktionen des TwinCAT 3 Plastic Frameworks (zweite Ebene) in fertigungsspezifischer Ausführung implementiert.



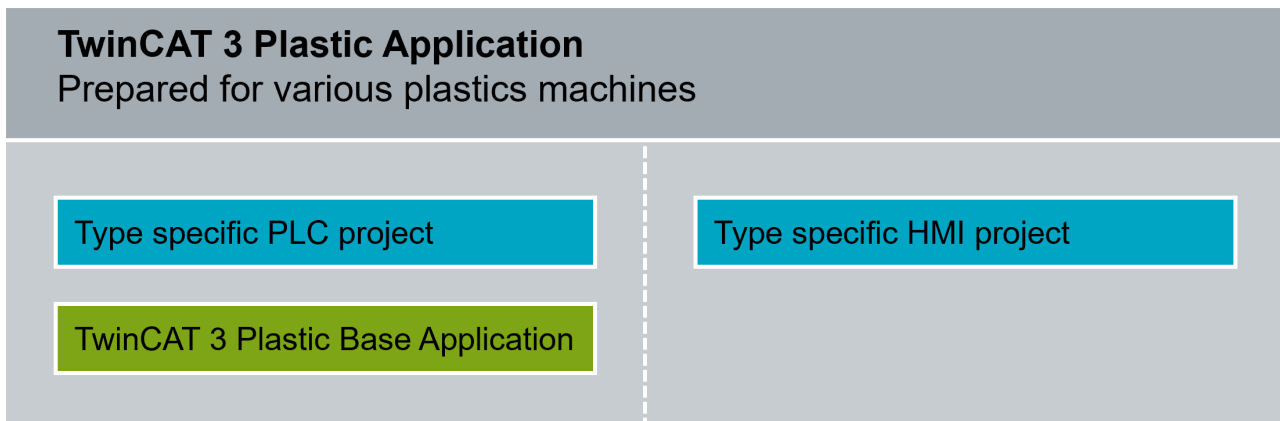
Als Beispiel für fertigungsspezifische Funktionen können verschiedene Fertigungsteilprozesse aufgeführt werden:

- Bläßformanwendung: Wanddickenregelung (WTC)
- Sptitzgießanwendung: Einspritzvorgang (Injection Unit)
- Extrusionsanwendung: Konstant-Extrusion (Extruder)

In der TwinCAT 3 Plastic Application sind genau diese Fertigungsteilprozesse implementiert und beispielhaft vorkonfiguriert. Der Source-Code für die Plastic Application wird kostenfrei zur Verfügung gestellt, wodurch das Editieren und Erweitern der Lösung in jeglicher Hinsicht möglich ist. Daher ist die Base Application ideal als Grundlage zum Starten eines neuen Projektes geeignet.

### Konkretisierte Struktur

Das Projekt ist neben der vertikalen Aufteilung in Sample Code und Plastic Base Application auch horizontal zu differenzieren: **PLC** und **HMI**.



Für die horizontale Aufteilung ergeben sich auf Seite der PLC zwei Projekte, welche aufeinander abgestimmt die Funktionen der Plastic Application implementieren.

In der vertikalen Unterscheidung werden beide Projekte in einen allgemeinen Anteil (Plastic Base Application) und einen maschinenspezifischen Anteil (Type specific PLC project) aufgeteilt. Durch diese Aufteilung kann gewährleistet werden, dass auch nach dem Beginn eines Projektes von der Weiterentwicklung der Plastic Application profitiert werden kann.

#### **● Editieren der Plastic Base Application nicht empfohlen**

**i** Das Editieren der Plastic Base Application ist durchaus möglich. Jedoch verfällt damit die Option der Unterstützung durch Beckhoff Automation in puncto Anpassungen und Erweiterungen. Im Normalfall editieren Sie ausschließlich den maschinenspezifischen Anteil (Sample Code) und nutzen die Plastic Base Application als Bibliothek.

Derzeit unterstützt die Plastic Application folgende Maschinentypen:

- Blasformmaschinen
- Extrusionsmaschinen

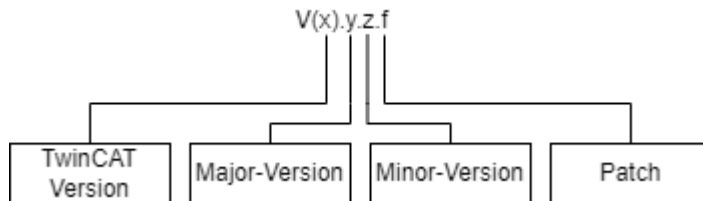
Weitere Maschinentypen sind für die zukünftige Entwicklung geplant.

### 3 Version der dokumentierten Software

#### Projekt:

<b>TwinCAT 3 Plastic Application</b>
--------------------------------------

V12.6.0
---------



#### Abhängigkeiten:

Name	Lizenznummer	Version
TwinCAT 3	/	3.1.4024.42 (oder höher)
TwinCAT 3 HMI	TF2000	12.760.44
TC3 Plastic TC HMI Controls	TF8550	12.8.1
TC3 Plastic Technology Functions	TF8560	3.12.4.32

#### **i** TwinCAT HMI Version

Nutzen Sie ausschließlich die hier angegebene Minor-Version der TwinCAT 3 HMI. Erfahrungsgemäß kann es bei Minor-Updates zu Inkompatibilitäten mit dem TwinCAT Plastic Application HMI Projekt kommen.

# 4 PLC Konzept und Strategie

Die PLC der Plastic Application implementiert die Steuerungsfunktionen des Plastic Frameworks (TF8540 & TF8560) in einer fertigungsorientierten Betrachtungsweise. Das Framework (insbesondere TF8560) selbst ist losgelöst von dieser Betrachtungsweise und implementiert technologische Kernfunktionen (CoreFunctions) für die Kunststofftechnik. So werden zum Beispiel eine allgemeine Punkt zu Punkt Bewegung oder Kurvenscheiben zur Verfügung gestellt. Darüber hinaus bietet das allgemeine Achs-Interface `I_AxisBase` die Möglichkeit ohne Festlegung der Antriebstechnologie die fertigungsorientierten Applikationsalgorithmen zu implementieren. Daraus ergeben sich die Klassen (FBs) der Plastic Base Application Bibliothek.

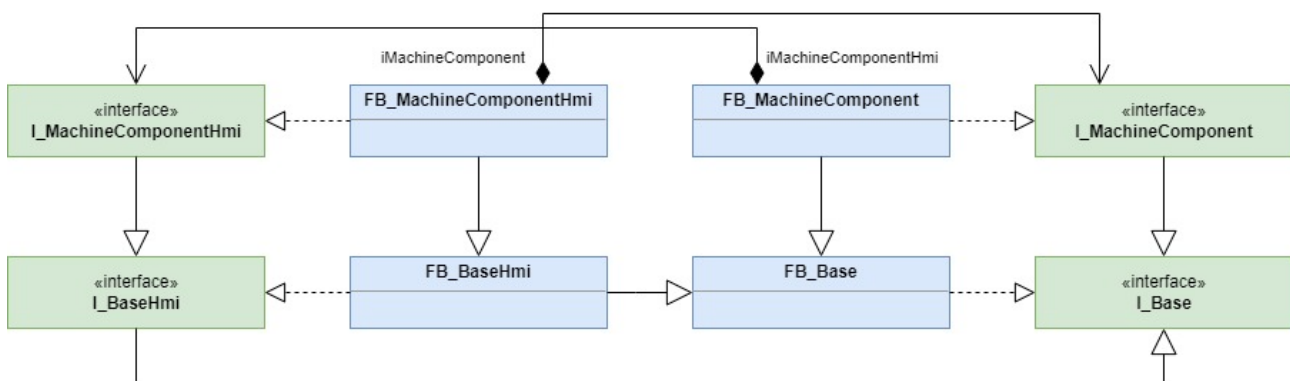
Als Beispiel für das Konzept kann hier die Wanddickensteuerung (WTC) betrachtet werden: Diese Komponente einer Blasformmaschine basiert fundamental auf der durch TF8560 zur Verfügung gestellten Kurvenscheiben-Kernfunktion (`Camming`). Für den Anwender ist dies häufig von sekundärer Bedeutung, da er die darauf aufsetzenden Funktionen wie **Start Wanddickenprofil** oder **Fahre auf Testposition** in seine Prozedur integrieren möchte. Dieser Schritt wird durch die Plastic Base Application vorimplementiert, sodass der Umfang an zu bewältigenden Aufgaben zur fertigen Maschine möglichst minimiert wird.

Auf weitere Konzeptpunkte wird in den folgenden Unterkapiteln näher eingegangen:

Thematik	Beschreibung
<a href="#">PLC und HMI [► 13]</a>	Kommunikationsweg zwischen PLC und HMI
<a href="#">Vererbung [► 14]</a>	Nutzung von Vererbung zur effizienteren Nutzung von redundantem Code und einfacheren Funktionserweiterung
<a href="#">Schnittstellen [► 15]</a>	Struktur von Schnittstelle in der Plastic Base Application und Programmierung mit Schnittstellen zur Erweiterung der Steuerungsflexibilität
<a href="#">Initialisierung &amp; Zyklusmethoden [► 15]</a>	Initialisierung von Referenzen, Parametern und Daten
<a href="#">Datenhandling [► 17]</a>	Handhabung von Daten auf Maschinen- und Produktebene
<a href="#">Projektaufbau [► 18]</a>	Erläuterung des TwinCAT Projektes

## 4.1 HMI-Schnittstelle

Die Kommunikation zwischen PLC und HMI ist in der Plastic Application konzeptionell berücksichtigt. Hierdurch wird gewährleistet, dass ein geregelter Zugriff des HMI auf die PLC möglich ist. Darüber hinaus wird auch in der PLC-Programmierung schnell ersichtlich, dass es sich bei den jeweiligen Abschnitten um Eingabe- und/oder Ausgabewerte des HMI handelt.

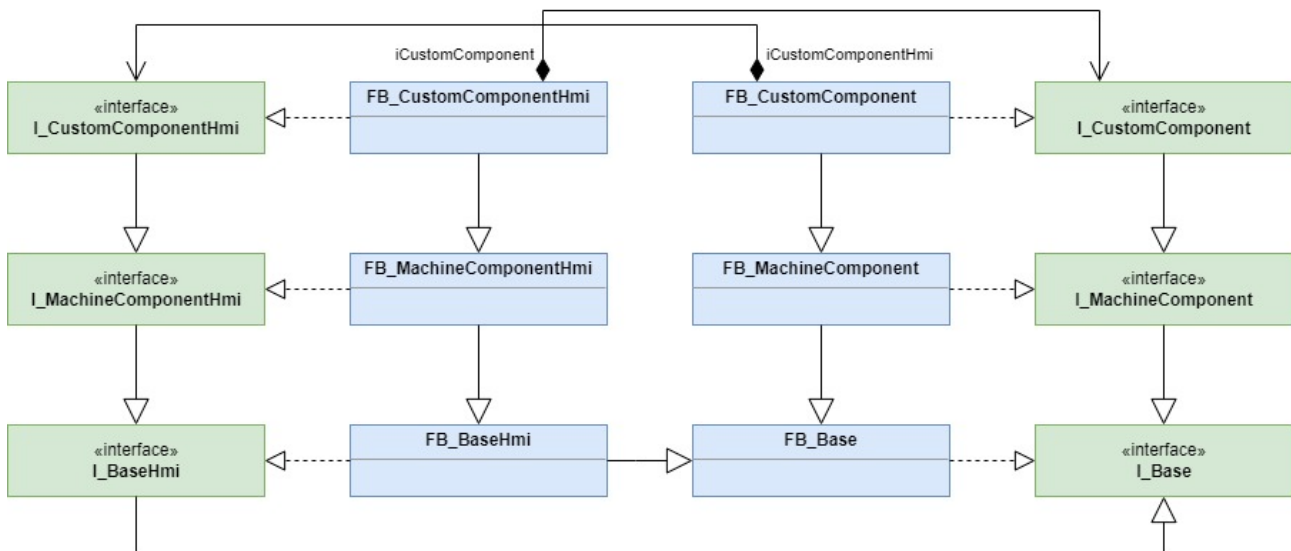


Das Konzept sieht vor, dass jede Klasse mit Informationsaustausch eine zweite Klasse übergeben bekommt, welche genau diese Funktion übernimmt. Daraus ergibt sich das Parallel-Klassen Modell. Neben der Hauptklasse der Betrachtung (`FB_Xyz`) existiert für jede hinzugefügte Klasse eine parallele Klasse vom Typ `FB_XyzHmi`. Die Parallelklasse wird über ein Interface mit der Hauptklasse verknüpft. In der Instanziierung muss hierfür die Instanz der Parallelklasse an die Instanz der Hauptklasse übergeben werden. Je nach Anwendungsfall kann in diesem Schritt auch die Hauptklasse der Parallelklasse übergeben werden.

Diese Anordnung ergibt eine Aufteilung von Steuerungsdaten und Benutzereingaben/Anzeigewerten. Darüber hinaus werden Controls der T8550 Plastic TC HMI Controls implizit unterstützt, wodurch die Integrität des Projektes stark verbessert wird. Genauso kann die Rezeptverwaltung optimal mit der PLC verknüpft und genutzt werden.

## 4.2 Änderung und Erweiterung durch Vererbung

In der vertikalen Aufteilung der Projektstruktur in Plastic Base Application und Sample Code wird die Erweiterung mittels der IEC 61131-3 Vererbung realisiert. Dadurch ist der maschinenspezifische Teil (Sample Code) frei anpassbar, kann jedoch auf den implementierten Algorithmen der Plastic Base Application aufbauen.



Die konsequente Nutzung von Vererbung für Änderungen an der Maschine-Steuerung ermöglicht darüber hinaus das direkte Aufspielen von Updates der Plastic Base Application durch Beckhoff Automation.

### ● Editieren der Plastic Base Application nicht empfohlen

**i** Das Editieren der Plastic Base Application ist durchaus möglich. Jedoch verfällt damit die Option der Unterstützung durch Beckhoff Automation in puncto Anpassungen und Erweiterungen. Im Normalfall editieren Sie ausschließlich den maschinenspezifischen Anteil (Sample Code) und nutzen die Plastic Base Application als Bibliothek.

### ● Konzept der objektorientierten Programmierung

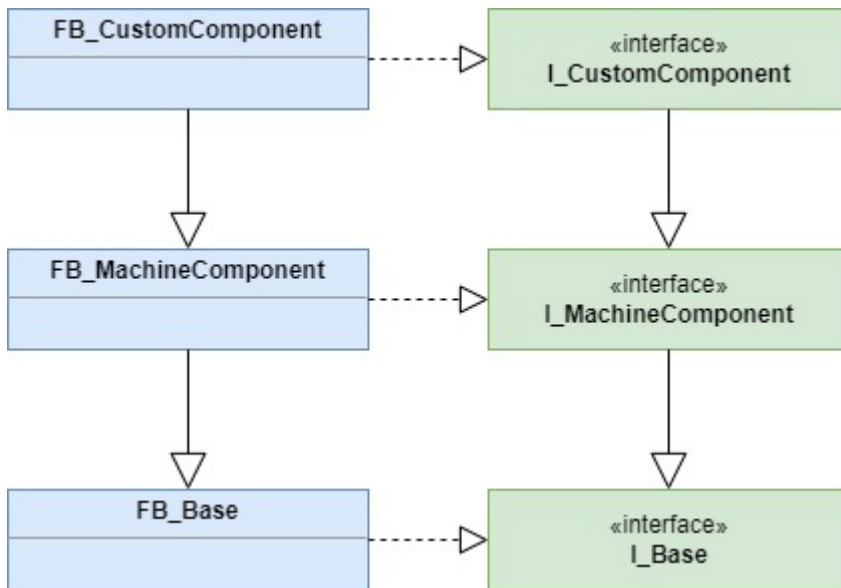
**i** Es ist zu empfehlen, sich im Detail mit dem Konzept der objektorientierten Programmierung nach IEC 61131-3 auseinanderzusetzen, um die Vorteile effektiv nutzen zu können. Weitere Informationen zur objektorientierten Programmierung finden Sie in der Dokumentation zur [TwinCAT 3 PLC-Programmierung](#)

In der Plastic Application gibt es insgesamt vier Fälle, in denen eine erbende Klasse (EXTENDS) die geerbte Klasse erweitern, ändern oder entfernen kann:

- Laufzeitspeicher in der Klassen-Variablendeklaration
  - Erweitern mit EXTENDS möglich
- Eigenschaften (Properties) oder Methoden hinzufügen, überschreiben und erweitern
  - Erweitern mit SUPER^. Aufruf möglich
- Innere Prozeduren erweitern, ändern und deaktivieren
  - Erweitern mit FB\_AdaptableSequence [► 103] möglich
- HMI-Parallelklasse erweitern und ändern
  - Erweitern der Schnittstelle und der Parallelklasse mit EXTENDS möglich

### 4.3 Nutzung von Schnittstellen

Die Plastic Base Application setzt, wie auch schon die TF8560 Plastic Technology Functions, auf das Arbeiten mit Schnittstellen (Interface). Dafür existiert seitens der Plastic Base Application für nahezu jede Klasse eine Standard-Schnittstelle, die Zugriff auf die relevanten Eigenschaften und Methoden gibt.



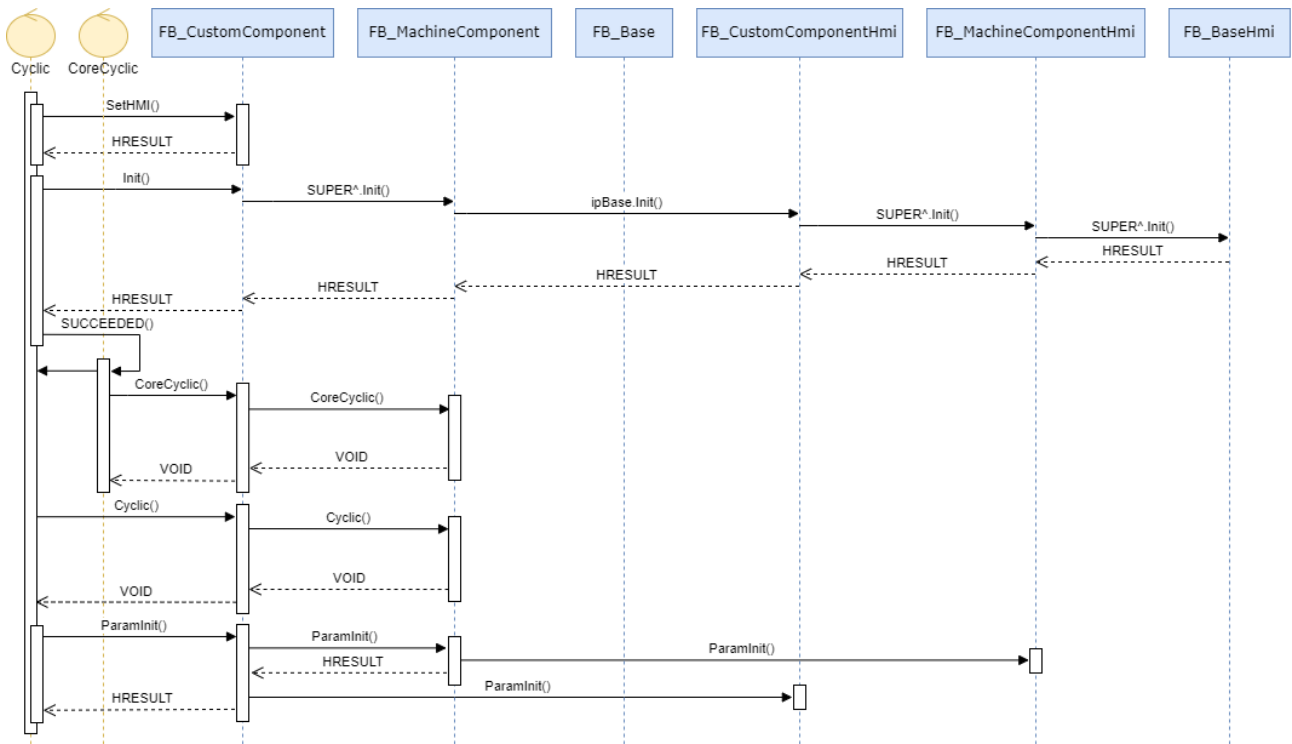
Für viele praktische Anwendungsfälle in der Programmierung gestaltet sich der Zugriff über eine Schnittstelle als deutlich übersichtlicher. Beispielsweise sind Methoden zur internen Kommunikation zwischen zwei Klassen als auch weitere Eigenschaften nicht sichtbar.

Beim Erweitern einer Klasse mittels Vererbung (siehe [Abschnitt Vererbung \[► 14\]](#)) ist es ebenfalls möglich, die entsprechende Standard-Schnittstelle mitzuerweitern. Dies kann vergleichbar mit dem Erben von einer Klasse mit dem Keyword `EXTENDS` bewerkstelligt werden.

### 4.4 Initialisierung und Zyklusmethoden

Da im Konzept der Plastic Base Application gehäuft mit Referenzwerten wie Schnittstellen gearbeitet wird, muss gewährleistet sein, dass genutzte Referenzwerte eine gültige Zuweisung besitzen. Um die Laufzeit nicht durch die ständige Validierung dieser Zuweisungen zusätzlich zu belasten, implementiert die Plastic Base Application ein Konzept zur Initialisierung eines Objektes.

Der Zugriff von ungültigen Referenzwerten betrifft im Wesentlichen den Aufruf von Zyklusmethoden eines Plastic Base Application Objekts. Aus diesem Grund muss ein Objekt erfolgreich initialisiert worden sein, bevor die Zyklusmethode ausgeführt werden darf.



**● Handling mit der FB\_BaseRuntime Klasse empfohlen**

**i** Alle im Folgenden beschriebenen Initialisierungsschritte und Zyklusaufrufe können mithilfe der `FB_BaseRuntime` umgesetzt werden. Dadurch müssen Objekte lediglich der Runtime übergeben werden. Beim Fehlschlagen der Initialisierungsschritte werden entsprechende Informationen über den `TcEventLogger` ausgegeben.

**SetHMI() Methode**

Mittels der `SetHMI()` Methode kann eine HMI-Parallelklasse an die Hauptklasse übergeben werden. Die Methode muss immer vor der `Init()` Methode ausgeführt werden, sofern die Klasse nicht auch ohne HMI-Parallelklasse funktioniert.

Fehlercode	Kürzel	Beschreibung
0x701	SrvNotSupported	Die Klasse unterstützt keine HMI-Parallelklasse.
0x71B	InvalidInterface	Die übergebene Schnittstelle ist ungültig.

**Init() Methode**

Die `Init()` Methode dient neben dem Setzen und Konfigurieren von Referenzwertzuweisungen der Überprüfung des Initialisierungs-Zustands. Dementsprechend muss beim Initialisieren eines Objektes der `HRESULT` Rückgabewert der Methode überprüft werden. Ist die Ausführung der `Init()` Methode nicht erfolgreich, existiert ein Fehler im Aufbau der Programm-Codes, dieser muss auf Code-Ebene gelöst werden.

**● Nutzen des BaseStates**

**i** Redundant zum Rückgabewert der `Init()` Methode kann auch der `BaseState` der Basisklasse `FB_Base` auf einen Wert größer als `E_BaseState.eInit` überprüft werden.

**Cyclic() Methode**

Die `Cyclic()` Zyklusmethode fungiert als objektorientierte Version eines Klassen-Instanzaufrufs. Das heißt, die Methode wird in der Regel einmal pro Zyklus aufgerufen um Prozeduren und/oder Aktualisierungen zu bearbeiten. Bei den in der `Cyclic()`-Methode implementierten Routinen handelt es sich meist um



Mechanismen mit mittlerer Priorität. Daher eignet sich diese Methode für die Implementierung einer vordefinierten Kommandoabfolge oder der Verarbeitung von Steuerungsbefehlen aus dem HMI (z. B. die Umsetzung von Handfunktionen).

**ParamInit() Methode**

Die ParamInit() Methode dient zur Initialisierung von Parametern. Heißt, in dieser Methode werden hartcodierte Standardwerte definiert, mit denen eine Inbetriebnahme ohne vorkonfigurierte Maschinendaten gestartet wird. Da in dieser Methode auch asynchrone Kommunikation benötigt werden darf, kann die Ausführung über mehrere Zyklen andauern bis der HRESULT Rückgabewert den Erfolg der Operation signalisiert. Um einen ausbleibenden Erfolg der Ausführung zu erkennen, empfiehlt es sich einen Timeout hinzuzufügen.

**● Laden von Maschinendaten**

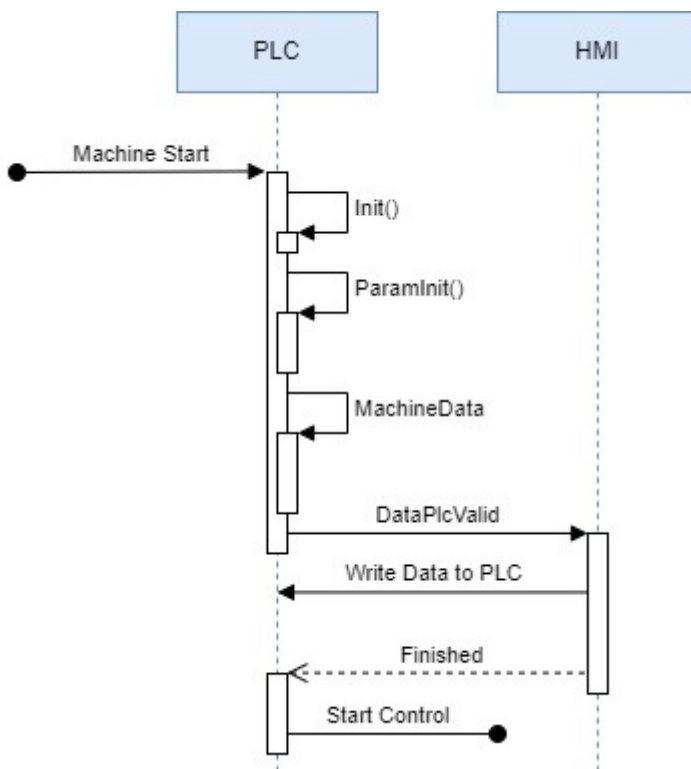
**i** Ist das Laden von Maschinendaten aus einer Datei beim Starten der Maschine erwünscht, ist dies in der ParamInit() Methode nicht möglich. Implementieren Sie daher eine separate Prozedur, die im Anschluss an die erfolgreiche Ausführung der ParamInit() startet. Im Plastic Application Projekt wird dies durch die FB\_BaseRuntime Klasse umgesetzt.

**CoreCyclic() Methode**

Die spezielle Abwandlung einer Cyclic() Methode als CoreCyclic() findet sich bei einzelnen Klassen wie beispielsweise FB\_Axis wieder. Diese Methode bearbeitet besonders echtzeitkritische Aufgaben wie beispielsweise den Aufruf der Zyklusmethode einer TF8560 Achse. Da eine solche Achse auch Regelungsalgorithmen enthält, empfiehlt es sich die Zykluszeit möglichst klein zu wählen (Standard 2 ms).

**4.5 Datenhandling**

Das Konzept für das Datenhandling der Plastic Application teilt sich auf mehrere Ebenen auf.



Im Wesentlichen implementiert die Plastic Base Application drei Ebenen zum Laden von Parametern und weiteren Daten. Diese Ebenen werden während der Hochlaufphase nacheinander durchlaufen und können teils auch nach dem Start erneut ausgeführt werden.

### Standard-Parametrierung (in Code)– ParamInit ()

Nach erfolgreicher Ausführung der Init() Methode werden die im Code definierten Standard-Parameter geladen. Daher gilt diese Ebene als Grundlage, kann aber von jeder folgenden Instanz überschrieben werden.

### Konstruktions- und Inbetriebnahme abhängige Daten in einer Binärdatei – MachineData (Maschinendaten)

Unter den Maschinendaten werden solche Daten zusammengefasst, die nur einmalig während der Konstruktion, Inbetriebnahme oder Änderung der Maschine eingestellt werden. Da diese Daten Einfluss auf das Maschinenverhalten haben können, wird die Datei im Binärformat abgespeichert und durch eine CRC-Checksumme gegen externe Manipulierung grob abgesichert. Die Verwaltung der Maschinendaten wird vollständig über die SPS gelöst, kann aber durch die HMI eingestellt und/oder geladen/gespeichert werden.

Das implementierte Maschinendaten-Handling basiert auf den TF8560 Utilities. Dabei teilt sich der SPS-Code-Teil auf zwei Klassen auf: Container und Komponenten.

Die Container sind in viele Objekte der Plastic Base Application integriert und dienen als Sammelobjekt für die Daten einer Datei. Eine Klasse, die Maschinendaten enthält, hat entweder einen eigenen Container oder verfügt über eine Komponente, welche einem Container hinzugefügt werden kann. Die Klassen mit einem eigenen Container erben von der Klasse FB\_BaseMd und stellen ein Interface auf diesen zur Verfügung. In der Vererbungsstruktur werden dem Container mehrere Komponenten (pro Vererbungsstufe) hinzugefügt. Dadurch kann auch in der Endapplikation der Umfang der Maschinendaten weiter erweitert werden.

Da die Indices der Parameter hierfür nicht willkürlich gewählt werden dürfen, werden in der Plastic Base Application die Parameter nach einem definierten Schema gespeichert. Dadurch kann ebenfalls gewährleistet werden, dass die Parameter in einem in Zukunft bereitgestellten Programm außerhalb der Steuerung vorbereitet werden können.

### Produktdaten / Maschinendaten – HMI-Recipe

Die für die Variation des Fertigungsprozesses regelmäßig neu einzustellenden Parameter fassen sich unter den Rezeptdaten zusammen. Hierzu zählen beispielsweise die Solltemperaturen der Temperaturzonen, Timings, Fertigungsgeschwindigkeiten, etc... Die Rezeptdaten werden gesammelt durch das HMI verwaltet und können im Engineering beliebig erweitert werden. Nähere Erläuterungen zur Rezeptverwaltung finden Sie im Kapitel [Rezeptverwaltung \[▶ 182\]](#).

## 4.6 Projektaufbau

Auf die einzelnen Bestandteile des Projektbaums wird im Folgenden genauer eingegangen.

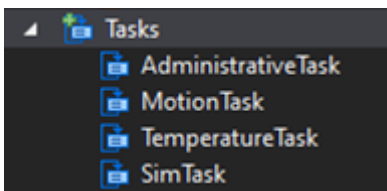
### System > License

Order No	License	Instances	License TAN	Current Status
TC1200	TC3 PLC	cpu license		unknown
TF2000	TC3 HMI Server	cpu license		unknown
TF5000	TC3 NC PTP	cpu license		unknown
TF5050	TC3 NC Camming	cpu license		unknown
TF5810	TC3 Hydraulic Positioning	cpu license		unknown
TF8540	Plastic Processing TwinCAT Framework	cpu license		unknown
TF8550	TC3 Plastic TC HMI Framework	cpu license		unknown
TF8560	TC3 Plastic Technology Functions	cpu license		unknown

Unter folgenden Bedingungen können Lizenzen aus dem Projekt ausgeschlossen werden:

Nummer	Kontext	Ausschließoption
TC1200	PLC Projekt	Nein
TF2000	TwinCAT HMI	Manuell aus der Lizenzliste entfernen
TF5000	Elektrische NC-Achsen	NC-Task unter MOTION entfernen
TF5050	Elektrische NC-Trafo-Achsen	NC-Trafo Achsen aus der PLC entfernen
TF5810	Hydraulische Achsen	Tc3_PlasticHydraulics Referenz und Achsen aus der PLC entfernen
TF8540	Temperaturregelung	Nein
TF8550	HMI Plastic Controls	NuGet-Packages und Controls im HMI entfernen
TF8560	Kunststoff Technologiefunktionen	Nein (konsequent in der Plastic Application verwendet)

**System > Tasks**



Die vier angelegten Tasks verfolgen folgende Strategien:

- **AdministrativeTask** - Allgemeine Steuerungsaufgaben mittlerer Priorität
  - Standard-Zykluszeit: 10 ms
  - Aufruf der HMI <> PLC Rezeptkommunikation
  - Speicher- und Laderoutinen
  - Kommunikation Inbetriebnahme-Tool von Hydraulic-Achsen
  - Zyklische Bearbeitung von Applikationsroutinen wie: Handfunktionen, Prozessablauf, etc...
- **MotionTask** – Echtzeitkritische Routinen
  - Standard-Zykluszeit: 2 ms (= NC Task)
  - Zyklusmethoden der Bewegungsachsen
  - Innere Regelungsmechanismen
- **TemperaturTask** – Träge Temperaturreglung
  - Standard-Zykluszeit: 25 ms
  - Aufruf der Temperaturreglung
- **SimTask** –Simulation
  - Standard-Zykluszeit: 2 ms
  - Aufruf der Zyklusmethoden der Simulation

**HINWEIS**

**Richtige Zykluszeit für die Temperaturreglung**

Die Zykluszeit der Temperaturreglung sollte asynchron zur Frequenz der AC-Versorgungsspannung sein. Eine synchrone Zykluszeit fördert eine Instabilität der Regelung!

**System > Real-Time**

Available Cores

Shared / Isolated:

Core	RT-Core	Base ...	Core Limit	Latency Warning
0	<input checked="" type="checkbox"/>	1 ...	80 %	(none)
1	<input checked="" type="checkbox"/> Default	1 ...	80 %	(none)

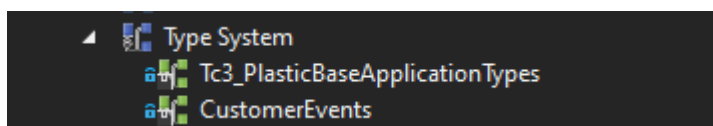
  

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
MotionTask	Default (1)	1 ms	2 ms	2	4
NC-Task 1 SAF	Default (1)	1 ms	2 ms	2	6
SimTask	Core 0	1 ms	2 ms	2	8
I/O Idle Task	Default (1)	1 ms	1 ms	1	10
AdministrativeT...	Default (1)	1 ms	10 ms	10	14
TemperatureTask	Default (1)	1 ms	25 ms	25	16
PlcAuxTask	Default (1)	1 ms	(none)	0	50

Je nach Kernanzahl und Singelcore-Performance der CPU können die PLC-Tasks aufgeteilt werden. Für Entwicklungssysteme kann die Projektkonfiguration übernommen werden, für Endgeräte sind folgende Konfigurationen in Praxistests erprobt worden:

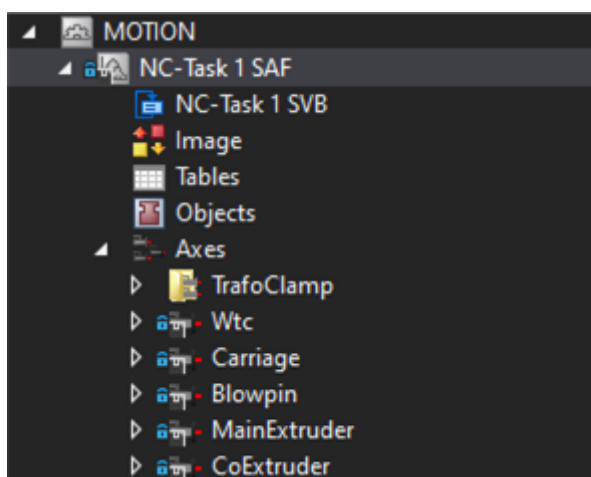
IPC / CX	Anpassungen
CX2033	Keine
CX2043	Keine
C6030 (Basis)	Keine

**System > Type System**



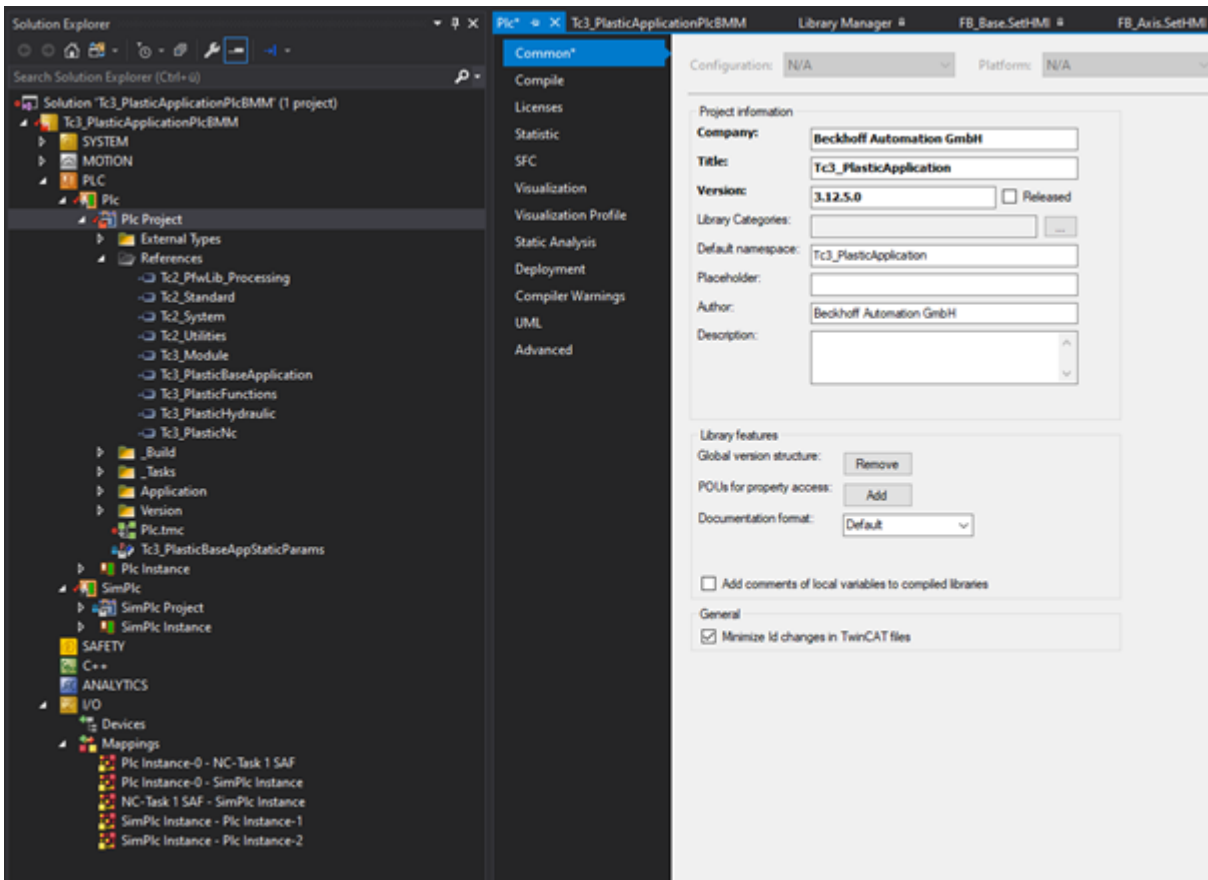
Das Type System beinhaltet die genutzten EventClasses. Diese EventClasses beinhalten die jeweiligen Events die in der Plastic Application verwendet werden.

**Motion**



In der NC-Task sind alle exemplarisch implementierten NC-Achsen (elektrisch) angelegt. Die Bezeichnungen der Achsen orientieren sich an der fertigungstechnischen Bedeutung.

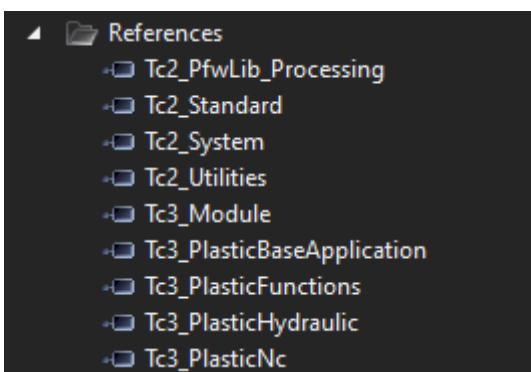
**PLC > Version**



Die aktuell verwendete Version des Projektes kann auf zwei Wegen ermittelt werden:

- PLC > PLC > PLC Project > Properties (Rechtsklick) > Common > Version:
- PLC > PLC > PLC Project > Version > Global\_Version

**PLC > Bibliotheksreferenzen**



Neben einiger allgemeiner Bibliotheken aus dem allgemeinen TwinCAT Pool sind in dem Projekt folgende Bibliotheken installiert:

- Tc2\_PfwLib\_Processing
  - Algorithmen der Temperaturregelung
- Tc3\_PlasticBaseApplication
  - Sammlung an applikationsorientierten Klassen, erläutert in dieser Dokumentation
- Tc3\_PlasticFunctions

- Virtuelle Achsschnittstelle und Technologiefunktionen
- Tc3\_PlasticHydraulic
  - Hydraulische Achsen kompatibel zur virtuellen Achsschnittstelle
- Tc3\_PlasticNc
  - Elektrische Achsen kompatibel zur virtuellen Achsschnittstelle

### ● Fehler trotz vorhandener Bibliothek

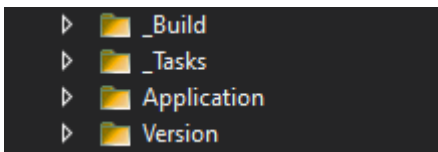
**i** Aus Kompatibilitäts- und Wartungszwecken sind die Versionen der enthaltenen Bibliotheken festgelegt. Wenn Sie bereits in vergangenen Versionen mit den genannten Bibliotheken gearbeitet haben, müssen Sie die aktualisierten Versionen nachinstallieren. Diese finden sich im gelieferten Projektordner unter **Dependencies**.

### HINWEIS

#### Anpassen auf eine ältere Version unzulässig

Ändern Sie nicht die im Projekt festgelegten Bibliotheksversionen auf eine ältere Version. Es kann zu Inkompatibilitäten und unvorhergesehenem Verhalten der Software kommen!

#### PLC > Code



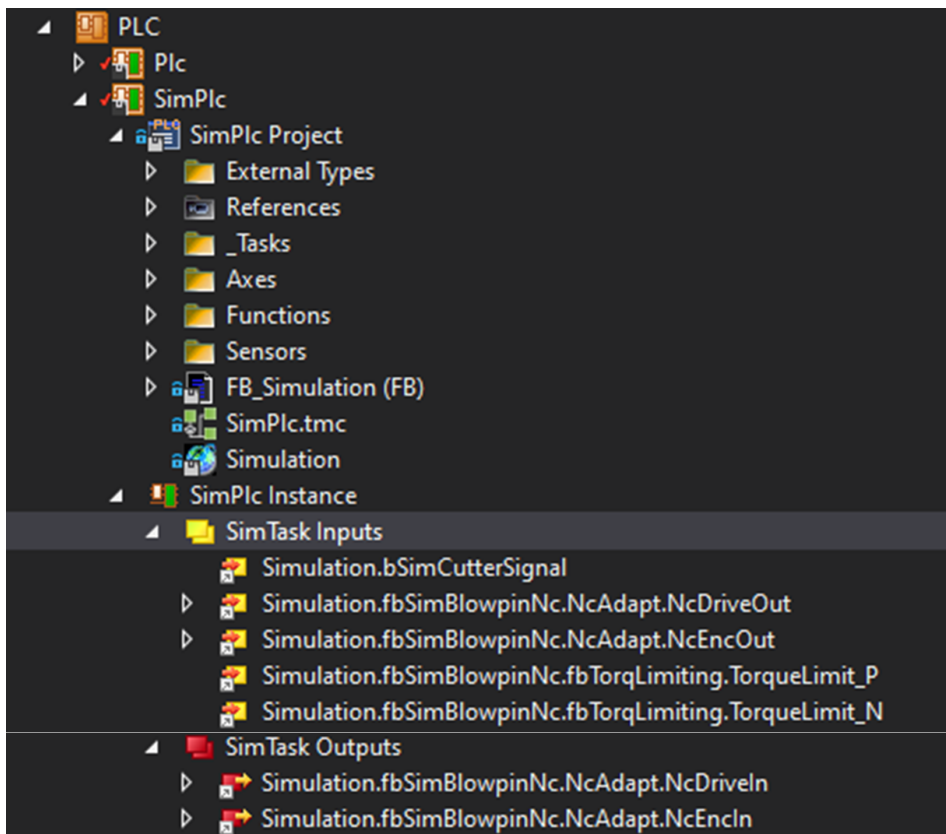
Der Steuerungs-Code ist auf vier Ordner aufgeteilt.

- `_Build`: Konfiguration des Maschinentyps
- `_Tasks`: Instanziierung der Runtime und Festlegung der Taskaufrufe
- `Application`: Applikationsprogramm
- `Version`: Automatisch erzeugte GVL mit der genutzten Projektversion

### ● Weiterführende Informationen zum PLC-Code

- i**
- [Klassenübersicht \[▶ 24\]](#)
  - [Erweiterung durch Vererbung \[▶ 14\]](#)

## PLC &gt; SimPlc



In der zweiten PLC (SimPlc) ist eine Maschinensimulation implementiert. Diese kann über das Mapping mit der Hauptsteuerung verbunden werden (vergleichbar mit dem E/A einer Maschine). Dadurch kann die PLC implementiert werden, ohne Simulationselemente zu enthalten. Darüber hinaus kann auch die Simulation mit den Elementen der PLC-Programmierung erweitert werden. Das bietet den Vorteil, dass die Steuerungs-PLC aus der Projektvorbereitung nicht an den Betrieb mit einer Realmaschine angepasst werden muss. Lediglich das Mapping auf E/A-Komponenten ist erforderlich.

## 5 PLC-API (Klassenübersicht)

### 5.1 Base - Basisklassen

#### 5.1.1 FB\_Base



FB\_Base ist die Basisklasse für die meisten verfügbaren Klassen der Plastic Base Application.

##### Interne Funktionen:

- Standard Error und Reset
- Erzeugung eines Instanz-Standardnamens
- Bereitstellung der Ereignis-Schnittstelle

##### Syntax:

```
FUNCTION_BLOCK FB_Base
```



##### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
BaseState	E_BaseState [▶ 25]	Get	eNotHandled	Initialisierungszustand der Klasseninstanz.
Error	BOOL	Get	FALSE	Klasse befindet sich in einem Fehlerzustand.
Name	STRING	Get, Set	-	Gibt den Namen der Klasseninstanz an.
ResultMessage	I_TcMessage	Get	-	Zugriff auf den Klasseninternen EventLogger.
_Name	STRING	Get	<SourceName >	[PROTECTED] .Name ohne führendes ‚fb‘

##### Methoden

Name	Beschreibung
Reset()	Setzt den Fehlerzustand der Klasse zurück.
SetHMI()	Allgemeine Zuweisungsmethode für eine FB_BaseHmi erweiternde Klasse.



##### Schnittstellen

Typ	Beschreibung
I_Base	Standardschnittstelle auf FB_Base.
I_BaseEmpty	Zur Erweiterung ohne Standardschnittstelle.
I_BaseDev	Deckt alle Methoden und Properties ab



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

**5.1.1.1 E\_BaseState**

Basis-State für den Zustand eines Plastic Base Application Objektes.

**Syntax:**

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_BaseState :
(
    eFailed := -9999,      // Init routine failed
    eReinit := -100,      // Object has to be reinitialized

    eNotHandled := 0,    // It starts Here

    eInit,                // FB_Init is succeeded
    eReady,               // Init is succeeded
    eIdle,                // ParamInit is succeeded
    eBusy,                // implementation usable busy flag

    eError := 1000       // implementation usable error flag
);
END_TYPE
```

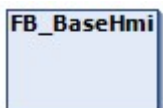
**Werte**

Name	Beschreibung
eFailed	Init-Prozedur ist fehlgeschlagen.
eReinit	Objekt muss re-initialisiert werden.
eNotHandled	Objekt wurde nicht behandelt.
eInit	FB_init wurde erfolgreich ausgeführt.
eReady	Init wurde erfolgreich ausgeführt.
eIdle	Paramlnit wurde erfolgreich ausgeführt.
eBusy	Implementierungs-nutzbares Busy-Flag.
eError	Implementierungs-nutzbares Error-Flag.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.1.2 FB\_BaseHmi**



FB\_BaseHmi ist die Basisklasse aller HMI-Parallelklassen. Durch die zur Verfügung gestellte Init(ipBase) Methode wird vordefiniert, dass das Parallelobjekt über das Interface I\_Base übergeben wird.

**Syntax:**

```
FUNCTION_BLOCK FB_BaseHmi EXTENDS FB_Base
```

 **Methoden**

Name	Beschreibung
Init()	Standardinitialisierungsmethode zum Übergeben der Basisklasse.

 **Schnittstellen**

Typ	Beschreibung
I_BaseHmi	Standardschnittstelle auf FB_BaseHmi.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.1.3 FB\_BaseMd



Erweitert die Basisklasse FB\_Base um die Funktionen des Maschinendaten-Handlings. Der Aufbau der Maschinendaten wird im [Kapitel Maschinendaten \[▶ 77\]](#) näher erläutert.

**Syntax:**

```
FUNCTION_BLOCK FB_BaseMd EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
MachineData	I_MdBaseContainer	Get	-	Interface auf das Maschinendatenhandling.

 **Methoden**

Name	Beschreibung
AddMdComp()	[PROTECTED] Fügt dem Maschinendaten-Container eine Komponente vom Typ I_MdComponent hinzu.
Cyclic()	Zyklusmethode – Aufruf einmal pro PLC-Zyklus.

 **Schnittstellen**

Typ	Beschreibung
I_BaseMd	Standardschnittstelle auf FB_BaseMd.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.1.4 FB\_BaseMdHmi



HMI-Klasse parallel zur Basis Maschinendaten-Klasse FB\_BaseMd. Ist notwendig zum Laden und Speichern der Maschinendaten über das HMI.

### Syntax:

```
FUNCTION_BLOCK FB_BaseMdHmi EXTENDS FB_BaseHmi
```

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
ParamHandle	REFERENCE TO FB_MdHandleHmi	Get	-	Schnittstelle zum Handling der Maschinendaten über das HMI.

### Methoden

Name	Beschreibung
Init()	Standardinitialisierungsmethode zum Übergeben der Basisklasse.

### Schnittstellen

Typ	Beschreibung
I_BaseMdHmi	Standardschnittstelle auf FB_BaseMdHmi.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.2 Runtime - Handling von Initialisierung und Zyklusaufrufen

### Vorteil / Nutzen

Die `FB_BaseRuntime` [▶ 28]-Klasse der Plastic Base Application dient zur Vereinfachung von Zyklusaufrufen und automatisierten Initialisierung von Steuerungsobjekten. Dadurch werden mehrere Schritte wie beispielsweise das Anlegen, Initialisieren, Rückgabewerte prüfen, Fehlermeldungen erzeugen, zyklische Aufrufen usw. stark vereinfacht.

### Voraussetzungen

Damit ein Objekt der Runtime angefügt werden kann, muss eine der folgenden Schnittstellen implementiert werden.

 **Schnittstellen**

Name	Beschreibung
I_RuntimeInterface	Allgemeine Schnittstelle für ein anhängbares Steuerungsobjekt.
I_OneTaskInterface	Schnittstelle mit einer ausführbaren Methode.
I_TwoTaskInterface	Schnittstelle mit zwei ausführbaren Methoden.
I_TempTaskInterface	Schnittstelle mit einer langsamen ausführbaren Methode.

## 5.2.1 FB\_BaseRuntime



Die Klasse automatisiert das Initialisieren und zyklische Aufrufen von instanziierten Steuerungsobjekten wie z.B. Achsen, Sequenzalgorithmen, Temperaturregler, etc...

**Interne Funktionen:**

- Initialisierung (Init(), ParamInit() und MdInit())
- Zyklusaufrufe (3-Tasks)
- PlcMcManager-Support Funktion (bei Verwendung hydraulischer Achsen)
- Softwareversion-Überprüfung

**Syntax:**

```
FUNCTION_BLOCK FB_BaseRuntime EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DisableMdFillCollections	BOOL	Get, Set	FALSE	Sperrt das Füllen von angehängten <a href="#">FB_MdCollection</a> [ <a href="#">▶ 82</a> ] Instanzen.
DisableMdInit	BOOL	Get, Set	FALSE	Sperrt das automatische Laden der Maschinendaten durch die Runtime.
DisableRecipe	BOOL	Get, Set	FALSE	Sperrt das Recipehandling von Objekten die das Interface <a href="#">I_Recipe</a> [ <a href="#">▶ 76</a> ] unterstützen.
MdInitExecuted	BOOL	Get	FALSE	Das Laden der Maschinendaten wurde für alle Objekte ausgeführt.
NumRuntimeObjects	INT	Get	0	Anzahl an angehängten Runtime-Objekten.

 **Methoden**

Name	Beschreibung
<a href="#">Append()</a> [ <a href="#">▶ 29</a> ]	Hängt ein Steuerungsobjekt der Runtime an.
Clear()	Löscht die Liste an angehängten Steuerungsobjekten.
MdSaveAll()	Startet den Speichervorgang aller angehängten Steuerungsobjekte mit Maschinendatencontainer.

**Zyklusmethoden**

Name	Beschreibung	Zykluszeit (empfohlen)
CoreCyclic()	Schnelle Zyklusmethode für Steuerungsobjekte mit I_TwoTaskInterface.	2 ms
Cyclic()	Normale Zyklusmethode für Steuerungsobjekte mit I_OneTaskInterface oder I_TwoTaskInterface.	10 ms
TemperatureCyclic()	Langsame Zyklusmethode für Steuerungsobjekte mit I_TempTaskInterface.	25 ms

Die Zyklusmethoden müssen mit verschiedenen Tasks zugewiesenen Programmen (PRG) aufgerufen werden, sodass die Runtime die angehängten Objekte den einzelnen Tasks zuordnen kann.

**Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
1xx	Rezept	Message
10xx	Initialisierung	Alarm + Message
11xx	Parametrierung	Alarm + Message
12xx	Versionsinkompatibilität	Alarm
20xx	Maschinendaten	Alarm + Message

**Schnittstellen**

Typ	Beschreibung
I_BaseRuntime	Standardschnittstelle auf FB_BaseRuntime.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.2.1.1 Append()**



Hängt ein Steuerungsobjekt der Runtime an.

**Syntax:**

```

METHOD Append : VOID
VAR_INPUT
    iObj: I_RuntimeInterface;
    iObjHmi: I_BaseHmi;
END_VAR
  
```

**Eingänge**

Name	Typ	Beschreibung
iObj	I_RuntimeInterface	Objekt das der Runtime angehängt werden soll.
iObjHmi	I_BaseHmi	Zugehöriges HMI-Objekt, welches mit dem Objekt verknüpft werden soll.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.2.2 I\_RuntimeInterface

Stellt die allgemeine Schnittstelle für ein Runtime-kompatibles Steuerungsobjekt dar.

**Syntax:**

```
INTERFACE I_RuntimeInterface
```

 **Methoden**

Name	Beschreibung
Init()	Methode zur Überprüfung der Initialisierung eines Objektes
ParamInit()	Methode zur Standardparametrierung eines Objektes
SetHMI(ipBaseHmi)	Methode zur Übergabe eines parallelen HMI-Objektes

Alle Methoden werden von der FB\_BaseRuntime Klasse auf den HRESULT Rückgabewert überprüft.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.3 Axis - Allgemeine Achstypen

### 5.3.1 FB\_Axis



Stellt die Standardklasse für alle Achs-Objekte aufbauend auf TF8560 Achsen dar.

**Syntax:**

```
FUNCTION_BLOCK FB_Axis EXTENDS FB_BaseMd
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
BaseAxisInterfaces	I_AxisBase	Get	NULL	Interface auf die zugewiesene TF8560-Achse.
ErrorAlarmTL	TcEventSeverity	Get, Set	.Error (3)	Severity des Alarms, den die Achse im Fehlerzustand auslöst.
Homing [ <a href="#">▶_35</a> ]	I_Homing	Get	-	Homing-Funktionen
Specific [ <a href="#">▶_31</a> ]	I_AxisSpecific	Get	-	Achstechnologie-spezifische Funktionen/Werte

 **Methoden**

Name	Beschreibung
CheckAxisState() [ <a href="#">▶ 36</a> ]	Überprüft den aktuellen Status der Achse.
FeedEnable()	Freigabe einer Bewegungsrichtung.
Power()	Einschalten der Antriebsregelung.
Reset()	Zurücksetzen von Achsfehlern.
SetAxisRef()	Zuweisungsmethode für die TF8560-Achse.
SetPosition()	Setzen der Achsposition.

 **Zyklusmethoden**

Name	Beschreibung
CoreCyclic()	Zyklusmethode mit kurzer Zykluszeit (default: 2 ms)

 **Schnittstellen**

Typ	Beschreibung
I_Axis	Standardschnittstelle auf FB_Axis
I_TwoTaskInterface	Runtime Schnittstelle für zwei PLC-Tasks

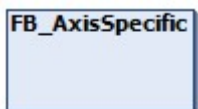
 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
10	Achsfehler	Alarm
20	Achskommando abgelehnt	Alarm

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.5)

**5.3.1.1 FB\_AxisSpecific**



Die Klasse implementiert technologiespezifische Funktionen einer TF8560 Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_AxisSpecific EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Hydraulic [ <a href="#">▶ 32</a> ]	I_HydraulicFunctions	Get	-	Hydraulikspezifische Funktionen
Nc [ <a href="#">▶ 32</a> ]	I_NcFunctions	Get	-	NC-spezifische (servoelektrische) Funktionen
Trafo [ <a href="#">▶ 33</a> ]	I_TrafoFunctions	Get	-	Transformationsspezifische Funktionen

 **Schnittstellen**

Typ	Beschreibung
I_AxisSpecific	Standardschnittstelle auf FB_AxisSpecific

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.1.1.1 FB\_HydraulicFunctions**



Implementiert hydraulik-spezifische Funktionen einer TF8560 Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_HydraulicFunctions EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AutoldentFinished	BOOL	Get	FALSE	Die ausgeführte Kennlinienvermessung wurde erfolgreich abgeschlossen.
IsHydraulic	BOOL	Get	FALSE	Hydraulikfunktionen werden vom zugewiesenen Achstyp unterstützt.

 **Methoden**

Name	Beschreibung
Autoldent()	Kennlinienvermessung einer hydraulischen Achse.

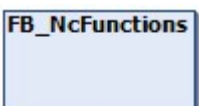
 **Schnittstellen**

Typ	Beschreibung
I_HydraulicFunctions	Standardschnittstelle auf FB_HydraulicFunctions

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.1.1.2 FB\_NcFunctions**



Implementiert spezifische Funktionen für eine NC basierte TF8560 Achse.



**Syntax:**

```
FUNCTION_BLOCK FB_NcFunctions EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Interface auf die Maschinendatenkomponente
IsNc	BOOL	Get	FALSE	NC-Funktionen werden vom zugewiesenen Achstyp unterstützt.

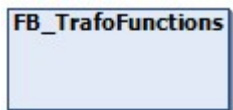
 **Schnittstellen**

Typ	Beschreibung
I_NcFunctions	Standardschnittstelle auf FB_NcFunctions
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.1.1.3 FB\_TrafoFunctions**



Implementiert spezifische Funktionen für eine transformierende TF8560-Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_TrafoFunctions EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Interface auf die Maschinendatenkomponente
IsTrafo	BOOL	Get	FALSE	Trafo-Funktionen werden vom zugewiesenen Achstyp unterstützt.
LowerLimit	LREAL	Get, Set	0.0	Untere Limitierung der realen Transformation
ParamList	I_Parameter	Get	-	Interne Liste zur Speicherung der TableGenerator-Parameter in den Maschinendaten
Table	I_CammingLookUp	Get	-	Interface auf die Instanz der Tabelle
TableGenerator	I_TrafoTableGenerator	Get, Set	NULL	Schnittstelle auf eine zuzuweisende TableGenerator-Klasse
UpperLimit	LREAL	Get, Set	0.0	Obere Limitierung der realen Transformation

 **Methoden**

Name	Beschreibung
AssignTableToAxis()	Weist die interne Tabelle der Achsinstanz zu (bedeutet das Aktivieren der Transformationskurve)
ConvDriveEndsToLoadEnds()	Berechnet die resultierenden Softwareendlagen der Lastseite anhand der Transformationstabelle
CopyTableDriveEnds()	Kopiert die Start- und Endposition der TableGenerator Klasse in die Softwareendlagen der Antriebsachse.
FillDebugTable()	Kopiert den Inhalt der internen Tabelle in ein zwei-dimensionales Array
ParamListToTableGenerator()	Kopiert die interne Parameterliste auf die Parameter des TableGenerators
TableGeneratorToParamList()	Kopiert die Parameter des TableGenerators in die interne Parameterliste

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
201	Aufrufen der TableGenerator Eigenschaft ohne vorherige Zuweisung	Message

 **Schnittstellen**

Typ	Beschreibung
I_TrafoFunctions	Standardschnittstelle auf FB_TrafoFunctions
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.5)

### 5.3.1.2 FB\_Homing



Die Homing (Referenzfahrt) Klasse implementiert Standard-Prozeduren zum Referenzieren einer Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_Homing EXTENDS FB_Base
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AbsoluteSwitch	BOOL	Set	FALSE	Eingangswert eines Absolutlage-Schalters
AttachableMdInterface	I_MdComponent	Get	-	Interface auf die Maschinendatenkomponente
ExternalStates	I_AdaptableSeqExt	Get, Set	NULL	Interface auf ein Objekt mit zusätzlichen Homing-Prozedurschritten

**Methoden**

Name	Beschreibung
DoAbort()	Bricht das aktuell aktive Referenzieren ab.
DoAbsSwitch()	Startet das Referenzieren der Achse gegen einen Absolutlage-Schalter.
DoAbsSwitchSequence()	Startet eine Sequenz aus DoAbsSwitch() und DoFinish().
DoBlock()	Startet das Referenzieren gegen einen Endanschlag.
DoBlockDetect()	Startet das Position-aufnehmen an einen Endanschlag.
DoBlockSequence()	Startet eine Sequenz aus DoBlock(), DoBlockDetect() und DoFinish().
DoFinish()	Startet das Beenden der Referenzfahrt.
DoSetZero()	Setzt die aktuelle Position der Achse auf 0

**Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Name	Beschreibung
HomingStates() [▶ 36]	Zustandsautomat der Homing-Prozeduren

**Ereignisgesteuerte Methoden (Callback-Methoden)**

Name	Beschreibung
ExtAdaptSeq()	Dummy-Methode() für die Nutzung ohne „ExternalStates“

**Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
4000	Homing nicht erlaubt	Alarm
4001	Homing fehlgeschlagen	Alarm
4002	Homing speichern war erfolgreich	Message
4003	Homing speichern war nicht erfolgreich	Message

**Schnittstellen**

Typ	Beschreibung
I_Homing	Standardschnittstelle auf FB_Homing
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten
I_AdaptableSeqExt	Schnittstelle zur Bereitstellung externer Homingprozeduren

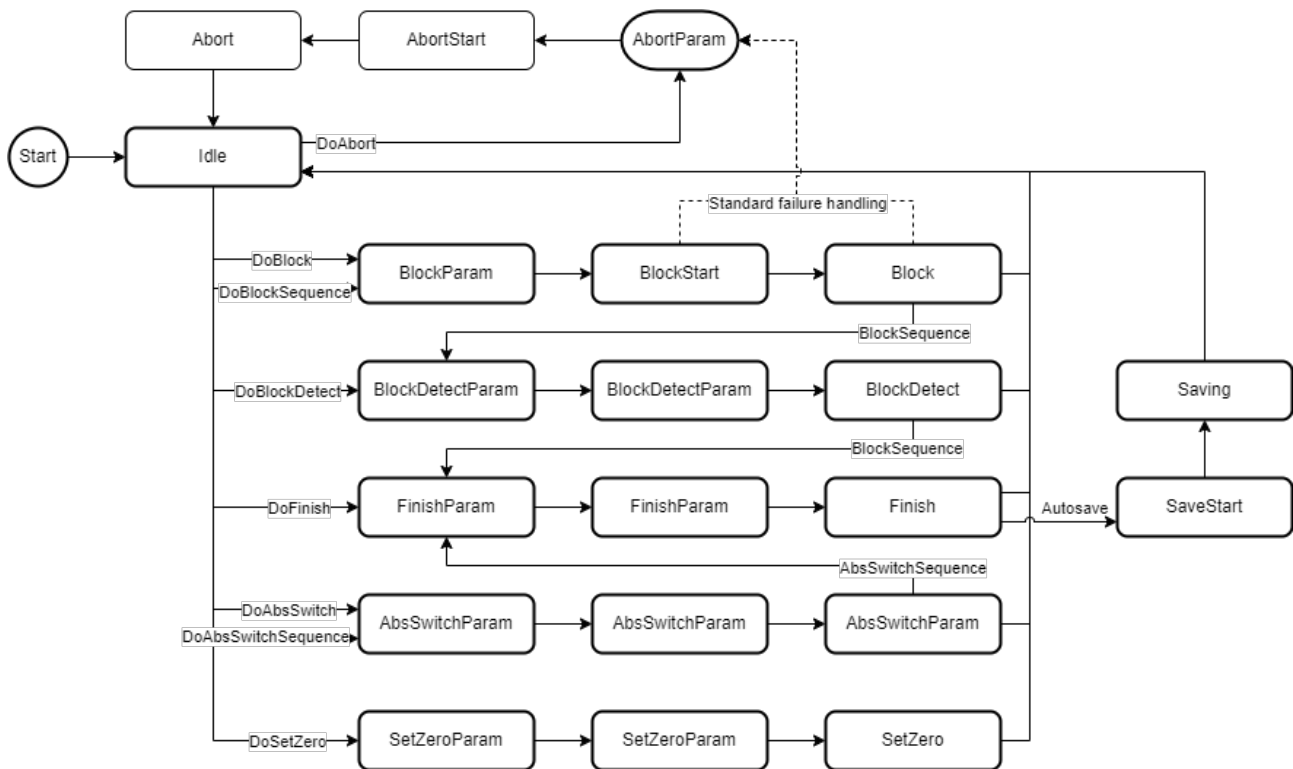
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.3.1.2.1 HomingStates()**

Intanztyp	Instanzname
Master	fbHomingStates
Slaves	aBaseSeqMembers[E_HomingState.eLength]

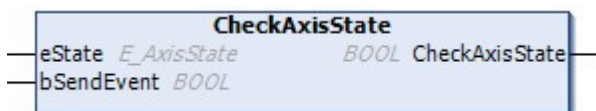
**State-Diagramm:**



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

**5.3.1.3 CheckAxisState()**



Überprüft den aktuellen Status der TF8560-Achse.

**Syntax:**

```
METHOD CheckAxisState : BOOL
VAR_INPUT
    eState:      BOOL;
    bSendEvent:  BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
eState	E_AxisState	Achsenzustand, der geprüft werden soll
bSendEvent	BOOL	Loggen eines Events, wenn der überprüfte Zustand nicht übereinstimmt.

 **Ausgänge**

Name	Typ	Beschreibung
CheckAxisState	BOOL	Ist TRUE, wenn der Achsenzustand übereinstimmt.

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
1000	Axis is not in state '{0}'	Message

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.2 FB\_PtpMotion**



Erbt von der FB\_Axis-Klasse und implementiert weitere Funktionen für PTP-basierte Bewegungen. Darüber hinaus wird über die FB\_PtpMotionHmi-Klasse das TF8550 Control Arrow-Motion-Graph unterstützt.

**Syntax:**

```
FUNCTION_BLOCK FB_PtpMotion EXTENDS FB_Axis
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ClampOnlyLastSeg	BOOL	Get, Set	TRUE	Clamping ist ausschließlich im letzten Segment möglich.
FixedMoveDirection <sup>1</sup>	BOOL	Get, Set	TRUE	Fixiert die Bewegungsrichtung für den Cluster 1 (negativ) und 2 (positiv).
PtpMoveFinished	BOOL	Get	FALSE	Die PTP-Bewegung wurde erfolgreich ausgeführt.
PtpMoveStopDone	BOOL	Get	FALSE	Die PTP-Bewegung wurde erfolgreich angehalten.
StackSegmentCount	BOOL	Get, Set	TRUE	Abgebrochene Segmente werden bei dem Folgebefehl berücksichtigt.
UseManualSpeed	BOOL	Get, Set	FALSE	Kommandierte PTP-Bewegungen sollen mit Handgeschwindigkeit ausgeführt werden.

<sup>1</sup> Ist obsolete (Alternativ: `FB_PtpMotionHmi.AxisMove.Moves[].Direction`)

 **Methoden**

Name	Beschreibung
CmpSegPos()	[PROTECTED] Bildet die Differenz zweier Segmentpositionen.
JogNegative()	Startet/Stoppt eine Jog-Bewegung in negativer Richtung.
JogPositive()	Startet/Stoppt eine Jog-Bewegung in positiver Richtung.
MovePtp()	Startet/Stoppt eine PTP-Bewegung mit der Parametrierung aus dem HMI.

 **Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103](#))**

Name	Beschreibung
PtpSeq() <a href="#">▶ 39</a>	Prozedur zum Laden einer PTP-Bewegung

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
2000	PTP Befehlsfehler	Message

 **Schnittstellen**

Typ	Beschreibung
I_PtpMotion	Standardschnittstelle auf FB_PtpMotion

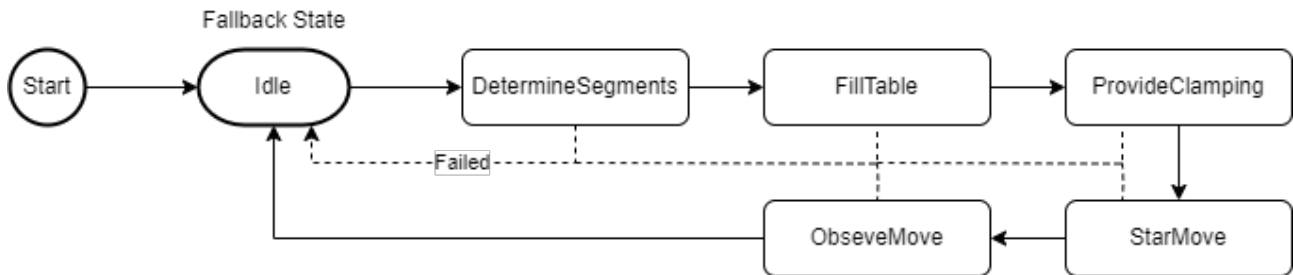
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.5)

### 5.3.2.1 PtpSeq()

Intanztyp	Instanzname
Master	fbPtpSeq
Slaves	aBaseSeqMembers [E_PtpBaseSeq.eLength]

State-Diagramm:



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

### 5.3.3 FB\_PtpMotionHmi



HMI-Parallelklasse der FB\_PtpMotion Klasse

Syntax:

```
FUNCTION_BLOCK FB_PtpMotionHmi EXTENDS FB_AxisHmi
```

#### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
ActiveMove	INT	Get, Set	0	Index des aktiven MoveClusters
ActiveSegment	INT	Get, Set	0	Index des aktiven Segments
AxisMove [ <a href="#">▶ 40</a> ]	REFERENCE TO FB_AxisMoveHmi	Get	-	Objekt der Bewegungskonfiguration
ParamPtpMotion	REFERENCE TO FB_MdPtpMotionHmi	Get	-	HMI-Zugriff auf die PTP-spezifischen Maschinendaten

#### Schnittstellen

Typ	Beschreibung
I_PtpMotionHmi	Standardschnittstelle auf FB_PtpMotionHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.3.1 FB\_AxisMoveHmi**

Fasst alle Move Cluster zusammen

**Syntax:**

```
FUNCTION_BLOCK FB_AxisMoveHmi EXTENDS FB_BaseHmi
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Moves [ <a href="#">▶ 40</a> ]	REFERENCE TO ARRAY[] OF FB_MoveCfg Hmi	Get	-	Move Cluster

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.3.1.1 FB\_MoveCfgHmi**

Fasst die Informationen eines Move Cluster zusammen

**Syntax:**

```
FUNCTION_BLOCK FB_MoveCfgHmi EXTENDS FB_BaseHmi
```



 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
<a href="#">Cams [▶ 41]</a>	REFERENCE TO ARRAY[] OF FB_CamCfgHmi	Get	-	Nocken des Clusters
ClampingDistance	LREAL	Get, Set	0.0	Position ab der das Clamping beginnt
ClampingVelocity	LREAL	Get, Set	0.0	Vorgesteuerte Geschwindigkeit, mit der das Clamping durchgeführt wird
Direction	INT	Get, Set	0	Vorgesehene Richtung des Clusters <ul style="list-style-type: none"> <li>• &gt; 0 – Positive Richtung</li> <li>• = 0 – undefinierte Richtung</li> <li>• &lt; 0 – Negative Richtung</li> </ul>
EndFunction	INT	Get, Set	0	ID der Move Cluster Startfunktion
InUse	BOOL	Get, Set	FALSE	Cluster wird verwendet
ManualVelocity	LREAL	Get, Set	0.0	Handgeschwindigkeit des Clusters
<a href="#">Segments [▶ 42]</a>	REFERENCE TO ARRAY[] OF FB_SegCfgHmi	Get	-	Segmente des Clusters
StartFunction	INT	Get, Set	0	ID der Move Cluster Endfunktion

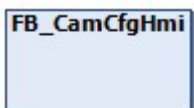
 **Methoden**

Name	Beschreibung
GetLastSegInUse()	[PROTECTED] Gibt das letzte Segment zurück, dessen .InUse Eigenschaft = TRUE ist

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

**5.3.3.1.1 FB\_CamCfgHmi**



Fasst die Informationen einer Nocke zusammen

**Syntax:**

```
FUNCTION_BLOCK FB_CamCfgHmi EXTENDS FB_BaseHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
CamActive	BOOL	Get, Set	FALSE	Nocke ist aktiv
Hysteresis	LREAL	Get, Set	0.0	Breite des Schwellwertes
Threshold	LREAL	Get, Set	0.0	Schwellwert

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.3.1.1.2 FB\_SegCfgHmi**



Fasst die Informationen eines Segmentes zusammen

**Syntax:**

```
FUNCTION_BLOCK FB_SegCfgHmi EXTENDS FB_BaseHmi
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Acceleration	LREAL	Get, Set	0.0	Beschleunigung des Segmentes
Decelration	LREAL	Get, Set	0.0	Verzögerung des Segmentes
IgnoreOnce	BOOL	Get, Set	FALSE	Ignoriert das Segment einmalig (beim nächsten Kommando)
InUse	BOOL	Get, Set	FALSE	Segment wird verwendet
Jerk	LREAL	Get, Set	0.0	Ruck des Segments
Limiting	LREAL	Get, Set	0.0	Druck-/Drehmomentlimitierung des Segmentes
Position	LREAL	Get, Set	0.0	Zielposition des Segmentes
SegFunction	INT	Get, Set	0	ID der Segmentfunktion
Velocity	LREAL	Get, Set	0.0	Geschwindigkeit des Segmentes

**Methoden**

Name	Beschreibung
GetPtpLookup()	Gibt das Segment als Variable vom Typ ST_LookUpPtpPoint zurück

**Schnittstellen**

Name	Beschreibung
I_SegCfgHmi	Standardschnittstelle auf FB_SegCfgHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

**5.3.4 FB\_BaseCammingHmi**



Erweitert die FB\_AxisHmi-Klasse für Camming-basierte Achsen um die notwendige Schnittstelle zum TF8550 CurveEditor.

**Syntax:**

```
FUNCTION_BLOCK FB_BaseCammingHmi EXTENDS FB_AxisHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ActivateCurve	BOOL	Get, Set	FALSE	Befehl des HMI zum Übernehmen der Kurvenscheibe auf Steuerungsebene
CurrentIndex	UDINT	Get	0	Bezeichnet den aktuellen Index der Kurvenscheibe, an der sich die Antriebsposition befindet.

 **Methoden**

Name	Beschreibung
GetActPoint()	Gibt einen Istwert der angezeigten Kurve zurück.
GetSetPoint()	Gibt einen Sollwert der angezeigten Kurve zurück.
SetActPoint()	Setzt einen Istwert der angezeigten Kurve.
SetSetPoint()	Setzt einen Sollwert der angezeigten Kurve.

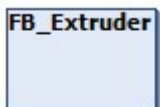
 **Schnittstellen**

Typ	Beschreibung
I_BaseCammingHmi	Standardschnittstelle auf FB_BaseCammingHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.3.5 FB\_Extruder**



Erbt von der FB\_Axis-Klasse und implementiert weitere Funktionen für eine kontinuierliche Bewegung eines Extruders.

**Syntax:**

```
FUNCTION_BLOCK FB_Extruder EXTENDS FB_Axis
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
GuidingValue	LREAL	Get	0.0	Enthält die aktuelle Drehzahl für einen Master/Slave-Verbund von zwei FB_Extruder-Objekten.
MasterExtruder	I_Extruder	Get, Set	NULL	Dient zur Zuweisung eines Master-Extruders.

 **Methoden**

Name	Beschreibung
DoBasicRpm()	Startet eine Drehbewegung mit der in der HMI eingestellten Basisgeschwindigkeit.
DoNominalRpm()	Startet eine Drehbewegung mit der in der HMI eingestellten Produktionsgeschwindigkeit.
DoTurnrateDown()	Verringert die Drehzahl.
DoTurnrateUp()	Erhöht die Drehzahl.

 **Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103](#))**

Name	Beschreibung
PowerStates() <a href="#">▶ 44</a>	Prozedur zum Ausführen der kontinuierlichen Drehbewegung

 **Schnittstellen**

Typ	Beschreibung
I_Extruder	Standardschnittstelle auf FB_Extruder

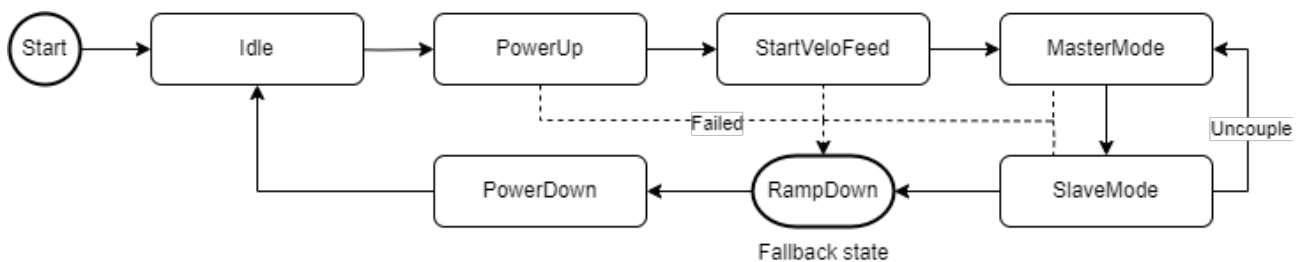
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

**5.3.5.1 PowerStates()**

Intanztyp	Instanzname
Master	fbPowerStates
Slaves	aSeqBaseMembers[E_ExtruderPowerStates.eLength]

**State-Diagram:**

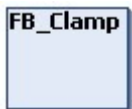


Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.4 Clamp, Carriage - Standard PTP-Achsen

### 5.4.1 FB\_Clamp



Implementiert spezifische Funktionen für eine Schließeinheit.

- Fügt der PTP-Bewegung ein Schließsegment hinzu, welches fest in den Maschinendaten parametrier wird.

**Syntax:**

```
FUNCTION_BLOCK FB_Clamp EXTENDS FB_PtpMotion
```



**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DisableLockSegment	BOOL	Get, Set	FALSE	Deaktiviert die Nutzung des Verschlusspunkt-Segments bei NC-Clamp Achsen.



**Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Name	Beschreibung
PtpSeq() [▶ 45]	Prozedur zum Laden einer PTP-Bewegung (erweitert)



**Schnittstellen**

Typ	Beschreibung
I_Clamp	Standardschnittstelle auf FB_Clamp

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

#### 5.4.1.1 PtpSeq()

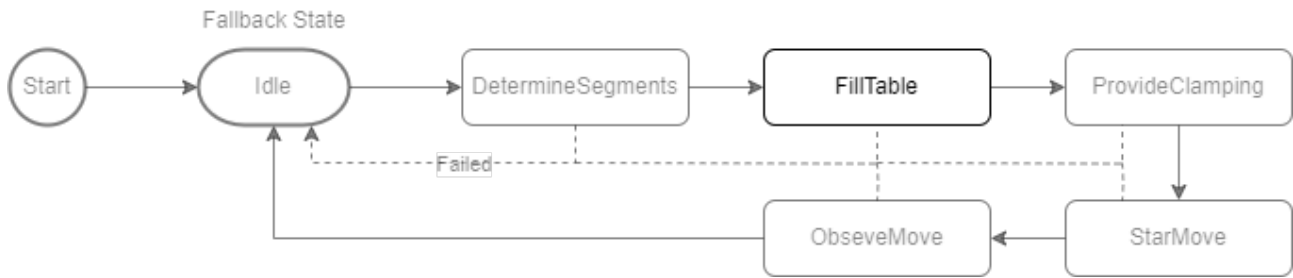


**Clamp-PTP-Variante nur bei NC-Trafo-Achsen aktiv**

Der modifizierte State `eFillTable` wird nur bei der Verwendung einer NC-Trafo-Achse verwendet!

Intanztyp	Instanzname
Master	fbPtpSeq
Slaves	aBaseSeqMembers[E_PtpBaseSeq.eFillTable]

**State-Diagram:**



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.4.2 FB\_Carriage**



Implementiert Zusatzfunktionen für den Betrieb einer Wagenbewegung (z. B. mithilfe eines Kurbeltrieb).

**Syntax:**

```
FUNCTION_BLOCK FB_Carriage EXTENDS FB_PtpMotion
```

**Methoden**

Name	Beschreibung
DynamicScaler()	Skaliert die Achsdynamik anhand der Geschwindigkeit im Verhältnis zur Maximalgeschwindigkeit. Die Methoden sind als PRIVATE implementiert. Der Aufruf in der Zyklusmethode wird über die Eigenschaften des FB_CarriageHmi aktiviert.
SmoothStart()	Ermöglicht ein Anfahren mit verringerter Dynamik im Bereich außerhalb der Endlagen. Die Methoden sind als PRIVATE implementiert. Der Aufruf in der Zyklusmethode wird über die Eigenschaften des FB_CarriageHmi aktiviert.

**Schnittstellen**

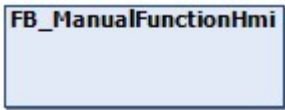
Typ	Beschreibung
I_Carriage	Standardschnittstelle auf FB_Carriage

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.5 ManualFunction - Handfunktionen

### 5.5.1 FB\_ManualFunctionHmi



Die Klasse dient zur Anbindung an ein TF8550 ManualOperation Control.

**Syntax:**

```
FUNCTION_BLOCK FB_ManualFunctionHmi EXTENDS FB_BaseHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Cmd [ <a href="#">▶</a> <a href="#">47</a> ]	I_ManualFunctionCmdHmi	Get	-	Kommandos aus dem HMI
State [ <a href="#">▶</a> <a href="#">48</a> ]	I_ManualFunctionStateHmi	Get	-	Feedback-Signale an das HMI

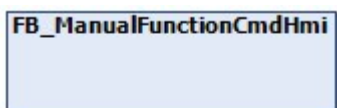
 **Schnittstellen**

Typ	Beschreibung
I_ManualFunctionHmi	Standardschnittstelle auf FB_ManualFunctionHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

#### 5.5.1.1 FB\_ManualFunctionCmdHmi



Fasst die Kommandos des HMI im FB\_ManualFunctionHmi zusammen.

**Syntax:**

```
FUNCTION_BLOCK FB_ManualFunctionCmdHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ToBasePos	BOOL	Get, Set	FALSE	Kommando zum Ansteuern der Grundposition
ToWorkPos	BOOL	Get, Set	FALSE	Kommando zum Ansteuern der Arbeitsposition

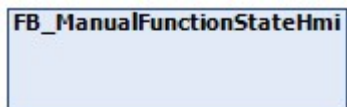
 **Schnittstellen**

Typ	Beschreibung
I_ManualFunctionCmdHmi	Standardschnittstelle auf FB_ManualFunctionCmdHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.5.1.2 FB\_ManualFunctionStateHmi**



Fasst die Feedback-Signale des HMI im FB\_ManualFunctionHmi zusammen.

**Interne Funktionen:**

- Die `MovingToXy` Eigenschaften setzen sich automatisch, wenn das jeweilige Signal des `FB_ManualFunctionCmd` [► 47] gesetzt wird

**Syntax:**

```
FUNCTION_BLOCK FB_ManualFunctionCmdHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
EnableBasePos	BOOL	Get, Set	FALSE	Gibt den Button der Grundposition frei.
EnableWorkPos	BOOL	Get, Set	FALSE	Gibt den Button der Arbeitsposition frei.
FaultBasePos	BOOL	Get, Set	FALSE	Signalisiert einen Fehler in der Grundposition.
FaultWorkPos	BOOL	Get, Set	FALSE	Signalisiert einen Fehler in der Arbeitsposition.
InBasePos	BOOL	Get, Set	FALSE	Signalisiert das Erreichen der Grundposition.
InWorkPos	BOOL	Get, Set	FALSE	Signalisiert das Erreichen der Arbeitsposition.
MovingToBasePos	BOOL	Get, Set	FALSE	Signalisiert die Ausführung zur Grundposition.
MovingToWorkPos	BOOL	Get, Set	FALSE	Signalisiert die Ausführung zur Arbeitsposition.

 **Schnittstellen**

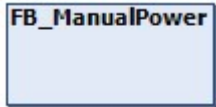
Typ	Beschreibung
I_ManualFunctionStateHmi	Standardschnittstelle auf FB_ManualFunctionStateHmi



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.5.2 FB\_ManualPower



Implementiert eine vorgefertigte Version einer Handfunktion zum Einschalten der Achsregelung.

**Syntax:**

```
FUNCTION_BLOCK FB_ManualPower EXTENDS FB_Base
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Enable	BOOL	Get, Set	FALSE	Gibt die Handfunktion frei.

**Methoden**

Name	Beschreibung
Cyclic()	Zyklusmethode
SetFeedEnableUse()	Konfiguriert die Richtungsfreigabe der Achsregelung.

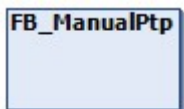
**Schnittstellen**

Typ	Beschreibung
I_ManualFunction	Universelle Schnittstelle auf eine Handfunktion

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.5.3 FB\_ManualPtp



Implementiert eine vorgefertigte Version einer Handfunktion für das Ansteuern einer PTP-Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_ManualPtp EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Enable	BOOL	Get, Set	FALSE	Gibt die Handfunktion frei.
TrigCmdBasePos	I_Trigger	Get	-	Trigger auf das Kommando der Grundposition
TrigCmdWorkPos	I_Trigger	Get	-	Trigger auf das Kommando der Arbeitsposition

 **Methoden**

Name	Beschreibung
Cyclic()	Zyklusmethode
SetWorkBasePos()	Setzt die Basis- und Arbeitsposition zur Indikation auf dem HMI

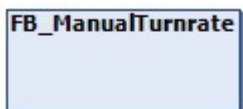
 **Schnittstellen**

Typ	Beschreibung
I_ManualFunction	Universelle Schnittstelle auf eine Handfunktion

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.5.4 FB\_ManualTurnrate



Implementiert eine Standardlimitierung und Anzeige für tastende Handfunktionen zum Anpassen der Drehzahl einer Achse.

**Syntax:**

```
FUNCTION_BLOCK FB_ManualTurnrate EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Enable	BOOL	Get, Set	FALSE	Gibt die Handfunktion frei.
CmdFaster	BOOL	Get	FALSE	Gibt den Befehl des gedrückten Buttons in der Arbeitsposition weiter.
CmdSlower	BOOL	Get	FALSE	Gibt den Befehl des gedrückten Buttons in der Basisposition weiter.

 **Methoden**

Name	Beschreibung
Cyclic()	Zyklusmethode
SetTurnrateLimits()	Konfiguriert die Minimal- und Maximalumdrehungszahl sowie die Nutzung der Limitierungsoptionen.

 **Schnittstellen**

Typ	Beschreibung
I_ManualFunction	Universelle Schnittstelle auf eine Handfunktion

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.6 Temperature - TF8540 Temperature Interface

### 5.6.1 FB\_Temperature



Hauptklasse der Temperaturregelung zur Verwaltung aller Temperaturkanäle und -gruppen.

**Interne Funktionen:**

- Handling der TcPfw-Temperaturregelung (TF8540)
- Integriertes Datenhandling der Gruppierungseinstellungen
- Verwaltung einzelner Subelemente wie:
  - Temperaturgruppen
  - Versorgungskanäle
  - Zeitplaner

**Syntax:**

```
FUNCTION_BLOCK FB_Temperature EXTENDS FB_BaseMd
```

## Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
DisableAlarms	BOOL	Get, Set	FALSE	Unterdrückt Alarmer, die durch Fehler in eines Temperaturkanals ausgelöst werden.
DisableMessages	BOOL	Get, Set	FALSE	Unterdrückt Debug-Meldungen der TF8540 Bibliothek.
EnableCallback	BOOL	Get, Set	TRUE	Aktiviert die Kommunikation mit den E/A-Komponenten.
EnableConfig	BOOL	Get, Set	TRUE	Gibt die Konfiguration aller Temperaturkanäle frei.
EnableLooptest	BOOL	Get, Set	FALSE	Aktiviert die Stromüberwachung aller Zonen.
LibScopeVars	REFERENCE TO FB_Scope_TempCtrlV ariables	Get	-	Zugriff auf eine Übersicht an TF8540 Live Daten.
Timer	I_TempSchedule	Get	NULL	Zugriff auf den verbundenen Zeitplaner.

## Methoden

Name	Beschreibung
Channels(x)	Gibt ein Interface zum x-ten Temperaturkanal von TF8540 zurück
CreateDefaultParams() [ <a href="#">▶ 53</a> ]	Erzeugt eine Standardparametrierung für alle Temperaturkanäle
EnableAll() [ <a href="#">▶ 54</a> ]	Gibt alle Temperaturkanäle PLC-seitig frei.
Groups(x)	Gibt ein Interface zur x-ten Temperaturgruppe zurück
LinkGroup() [ <a href="#">▶ 54</a> ]	Weist eine lineare Anordnung an Temperaturkanälen einer Gruppe zu.
LinkSupply() [ <a href="#">▶ 55</a> ]	Weist eine Gruppe einem Versorgungskanal zu.
LinkZone() [ <a href="#">▶ 56</a> ]	Weist einen Temperaturkanal einer Gruppe zu.
SetScheduler()	Weist der Temperaturregelung eine Zeitplanung zu.
StandbyAll() [ <a href="#">▶ 57</a> ]	Stellt alle Temperaturkanäle auf Standby.
Supply(x)	Gibt ein Interface zur x-ten Versorgungseinheit zurück
SupplyLines(x)	Gibt ein Interface zum x-ten Versorgungskanal zurück
UnlinkGroup()	Entfernt alle Temperaturkanäle aus einer Gruppe

### Vermeidung von Exceptions

**I** Die Listen-Zugriffsfunktionen (e.g. Channels(x), Groups(x), etc...) geben bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

## Schnittstellen

Typ	Beschreibung
I_Temperature	Standardschnittstelle auf FB_Temperature
I_TempTaskInterface	Runtime Schnittstelle für eine langsame PLC Task

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.1 CreateDefaultParams()**

```

CreateDefaultParams
HRESULT CreateDefaultParams(
    bAllInUse BOOL,
    eSensor E_TcPfw_TempSensType,
    eTerminal E_TcPfw_TerminalType,
    nChPerTerm INT,
    eOutHeating E_TcPfw_TctrlOutSelect,
    eOutCooling E_TcPfw_TctrlOutSelect,
    fSetpoint LREAL,
    fStandbySetp LREAL,
    fPwmCycleTime LREAL
)
    
```

Erzeugt eine Standardparametrierung für alle Temperaturkanäle

**Syntax:**

```

METHOD CreateDefaultParams : HRESULT
VAR_INPUT
    bAllInUse:          BOOL;
    eSensor:            E_TcPfw_TempSensType;
    eTerminal:          E_TcPfw_TerminalType;
    nChPerTerm:         INT;
    eOutHeating:        E_TcPfw_TctrlOutSelect;
    eOutCooling:        E_TcPfw_TctrlOutSelect;
    fSetpoint:          LREAL;
    fStandbySetp:       LREAL;
    fPwmCycleTime:     LREAL;
END_VAR
    
```

 **Eingänge**

Name	Typ	Beschreibung	Empfohlener Standard
bAllInUse	BOOL	Alle Kanäle werden als „InUse“ initialisiert.	TRUE
eSensor	E_TcPfw_TempSensType	Sensortyp – NoSensor im Simulationsmodus	eTcPfwTempSensT_NoSensor
eTerminal	E_TcPfwTerminalType	Klemmentyp – NoTerminal im Simulationsmodus	eTcPfwTermT_NoTerminal
nChPerTerm	INT	Anzahl der Kanäle pro Klemme	8
eOutHeating	E_TcPfw_TctrlOutSelect	Ausgabeart des Heizausgangs – NoSignal, um die Heizfunktion zu deaktivieren (bspw. bei Messzonen)	eTcPfwTcOut_PWM
eOutCooling	E_TcPfw_TctrlOutSelect	Ausgabeart des Kühlausgangs	eTcPfwTcOut_NoSignal
fSetpoint	LREAL	Temperatursollwert für alle Kanäle	180.0
fStandbySetp	LREAL	Temperatursollwert für die Standby-Temperatur aller Kanäle	18.0
fPwmCycleTime	LREAL	PWM-Zykluszeit für alle Ausgänge (dutyCycle = fPwmCycleTime * 0.1)	1.0

 **Ausgänge**

Name	Typ	Beschreibung
CreateDefaultParams	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.2 EnableAll()**



Gibt alle Temperaturkanäle PLC-seitig frei.

**Syntax:**

```

METHOD EnableAll
VAR_INPUT
    bCmd:      BOOL;
    bGroups:   BOOL;
END_VAR
    
```

 **Eingänge**

Name	Typ	Beschreibung
bCmd	BOOL	TRUE um die Freigabe zu erteilen, FALSE um die Freigabe zurückzunehmen.
bGroups	BOOL	Die Freigabe berücksichtigt nur Kanäle, die einer Gruppe zugewiesen sind.

 **Ausgänge**

Name	Typ	Beschreibung
EnableAll	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.3 LinkGroup()**



Weist eine Reihe an Temperaturkanälen einer Gruppe zu.

**Syntax:**

```
METHOD LinkGroup : HRESULT
VAR_INPUT
    nStartIdx:      INT;
    nEndIdx:        INT;
    nGroupIdx:      INT;
    bOverwrite:     BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nStartIdx	INT	Index des ersten Kanals der zugewiesen werden soll
nEndIdx	INT	Index des letzten Kanals der zugewiesen werden soll
nGroupIdx	INT	Index der Gruppe der die Kanäle zugewiesen werden sollen
bOverwrite <sup>1</sup>	BOOL	Zonen werden zugewiesen, auch wenn die Gruppe bereits zugewiesene Zonen enthält.

<sup>1</sup> Obsolete und wird ignoriert

 **Ausgänge**

Name	Typ	Beschreibung
LinkGroup	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.4 LinkSupply()**



Weist einer Gruppe eine Versorgungseinheit zu.

**Syntax:**

```
METHOD LinkSupply : HRESULT
VAR_INPUT
    nGroupIdx:      INT;
    nSupplyIdx:     INT;
    nLine:          INT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nGroupIdx	INT	Index der Gruppe der eine Versorgungseinheit zugewiesen werden soll
nSupplyIdx	INT	Index der Versorgungseinheit die der Gruppe zugewiesen werden soll
nLine	INT	Versorgungskanal an dem die Gruppenteilnehmer angeschlossen sind <ul style="list-style-type: none"> <li>• 1; 2; 3 – Phase L1, L2 oder L3</li> <li>• 4 – Zwischen Phasen ohne Verbindung zu N</li> </ul>

 **Ausgänge**

Name	Typ	Beschreibung
LinkSupply	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.5 LinkZone()**



Weist einen einzelnen Temperaturkanal (Zone) einer Gruppe hinzu.

**Syntax:**

```
METHOD LinkZone : HRESULT
VAR_INPUT
    nLinearIdx:      INT;
    nGroupIdx:       INT;
    nGroupMemberIdx: INT;
    bOverwrite:      BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nLinearIdx	INT	Index des Kanals der zugewiesen werden soll
nGroupIdx	INT	Index der Gruppe, der der Kanal zugewiesen werden soll
nGroupMemberIdx <sup>1</sup>	INT	<i>Index in der Zielgruppe</i>
bOverwrite <sup>1</sup>	BOOL	Zone wird zugewiesen, auch wenn der Index in der Zielgruppe bereits belegt ist.

<sup>1</sup> Obsolet, wird ignoriert



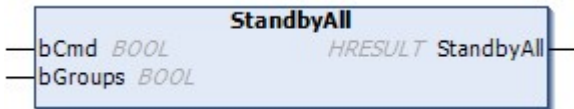
**Ausgänge**

Name	Typ	Beschreibung
LinkZone	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.1.6 StandbyAll()**



Stellt alle Temperaturkanäle auf Standby-Temperatur.

**Syntax:**

```

METHOD StandbyAll : HRESULT
VAR_INPUT
    bCmd:      BOOL;
    bGroups:   BOOL;
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
bCmd	BOOL	TRUE um die Standby-Temperatur zu aktivieren, FALSE zum deaktivieren.
bGroups	BOOL	Die Funktion berücksichtigt nur Kanäle, die einer Gruppe zugewiesen sind.

**Ausgänge**

Name	Typ	Beschreibung
StandbyAll	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.2 FB\_TemperatureHmi**



HMI-Parallelklasse zum FB\_Temperature.

**Syntax:**

```

FUNCTION_BLOCK FB_TemperatureHmi EXTENDS FB_BaseMdHmi
    
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Ein Kanal (InUse = TRUE) hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Ein Kanal (InUse = TRUE) hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Mindestens ein Kanal mit aktiver Regelung hat die HighHigh-Toleranz überschritten.
AlarmHigh	BOOL	Get	FALSE	Mindestens ein Kanal mit aktiver Regelung hat die High-Toleranz überschritten.
AlarmLow	BOOL	Get	FALSE	Mindestens ein Kanal mit aktiver Regelung hat die Low-Toleranz unterschritten.
AlarmLowLow	BOOL	Get	FALSE	Mindestens ein Kanal mit aktiver Regelung hat die LowLow-Toleranz unterschritten.
CountPfwChannels	LREAL	Get	20.0	Anzahl an verfügbaren TF8540 Temperaturkanälen
ParamTempSupply	REFERENCE TO ARRAY[] OF FB_MdTempSupplyHmi	Get	-	Parameterschnittstelle für die Parametrierung über das HMI
ParamTempZone	REFERENCE TO ARRAY[] OF FB_MdTempZoneHmi	Get	-	Parameterschnittstelle für die Parametrierung über das HMI
TempAmbient	LREAL	Get, Set	18.0	Standard-Umgebungstemperatur (für Simulation)

 **Methoden**

Name	Beschreibung
Groups(x)	Gibt ein Interface zur x-ten Temperaturgruppe (HMI-Klasse) zurück

**● Vermeidung von Exceptions**

**i** Die Listen-Zugriffsfunktionen (e.g. Channels(x), Groups(x), etc...) geben bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

 **Schnittstellen**

Typ	Beschreibung
I_TemperatureHmi	Standardschnittstelle auf FB_TemperatureHmi
I_Recipe	Schnittstelle zur Verwaltung strukturierter Rezeptwerte

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.3 FB\_TemperatureGroup**

Klasse zur Gruppensteuerung der Temperaturkanäle.

**Syntax:**

```
FUNCTION_BLOCK FB_TemperatureGroup EXTENDS FB_BaseMd
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die HighHigh-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die High-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die Low-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLowLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die LowLow-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmNoResponse	BOOL	Get	FALSE	Temperaturwert der Gruppe (mindestens ein Kanal) reagiert nicht auf die Ansteuerung.
AttachableMdInterface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
ConfigHash	T_SHA256	Get	0x0	Hash-Wert der aktuellen Gruppenkonfiguration
ConfigID	UINT	Get	0	ID der Gruppenkonfiguration (inkremental)
EnableLimitAlarms	BOOL	Get, Set	FALSE	Das Überschreiten eines Toleranzwertes/-limits löst einen Alarm aus.
Fault	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat einen Fehler.
Index	INT	Get	0	Index (ID) der Gruppe
IsActive	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) ist aktiv geregelt.
IsEnabled	BOOL	Get	FALSE	Alle Kanäle der Gruppe sind freigegeben.

Name	Typ	Zugriff	Initialwert	Beschreibung
IsStandby	BOOL	Get	FALSE	Alle Kanäle der Gruppe sind im StandbyBetrieb.
LoadHash	T_SHA256	Get	0x0	Hash-Wert der zuletzt aus einer Datei geladenen Konfiguration
ZonesCount	INT	Get	0	Anzahl an Zonen in der Gruppe

 **Methoden**

Name	Beschreibung
EnablePLC()	Gibt alle Zonen der Gruppe PLC-seitig frei.
Force()	Zwingt alle Zonen der Gruppe auf Heiz-/Kühlleistung 100%.
UpdateGroup()	Aktualisiert die interne Auflistung der zugeteilten Gruppen
Zones(x)	Gibt ein Interface zur x-ten Zone der Gruppe zurück

**Vermeidung von Exceptions**

**i** Die Listen-Zugriffsfunktionen (e.g. Channels(x), Groups(x), etc...) geben bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

 **Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Typ	Beschreibung
GroupStates() [▶ 62]	Prozedur zum Laden, Ausführen und Speichern des Tunings

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
401x	Temperaturbetrieb Überwachung	Alarm
402x	Tuningparameter Ladevorgang	Alarm
403x	Tuningparameter Speichervorgang	Message

 **Schnittstellen**

Typ	Beschreibung
I_TemperatureGroup	Standardschnittstelle auf FB_TemperatureGroup
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

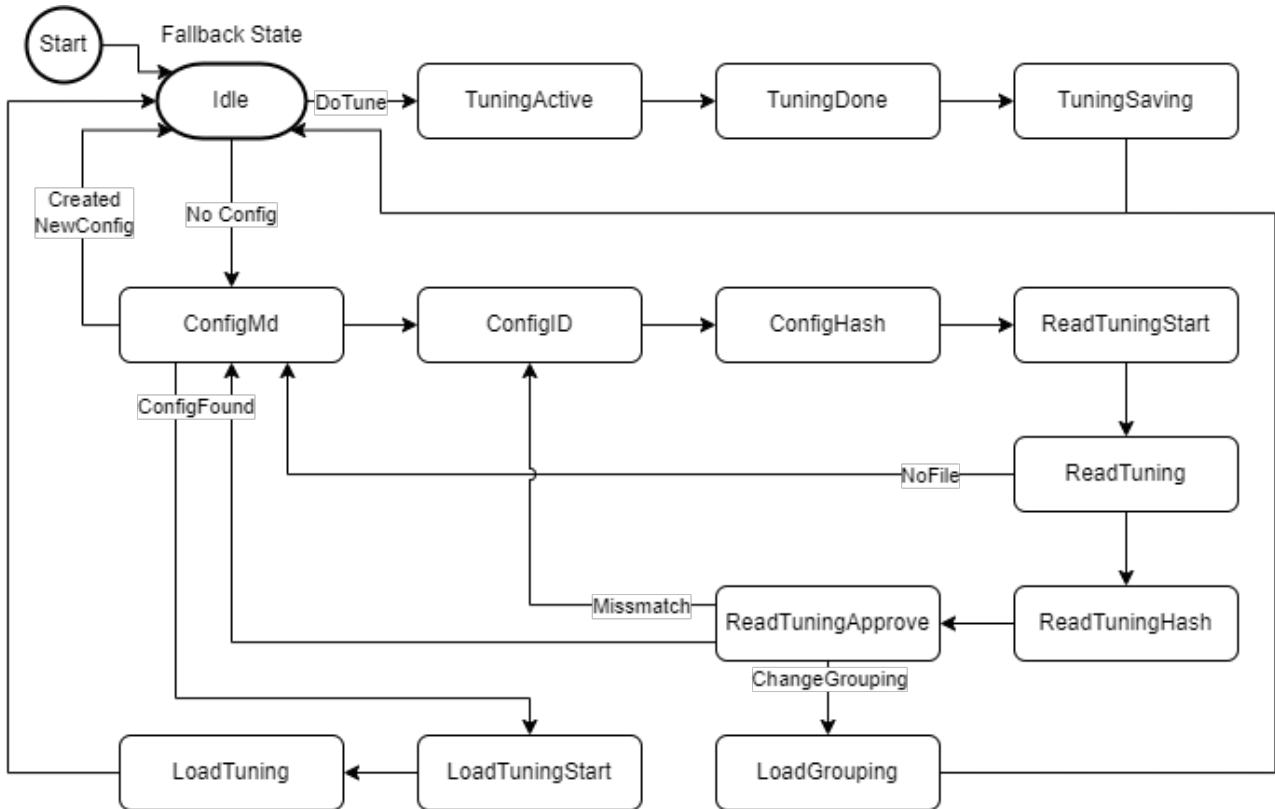
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.3.1 GroupStates()**

Intanztyp	Instanzname
Master	fbGroupStates
Slaves	aBaseSeqMembers[E_GroupStates.eLength]

**State-Diagram:**



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.4 FB\_TemperatureGroupHmi**



HMI-Parallelklasse zur FB\_TemperatureGroup Klasse.

**Syntax:**

```
FUNCTION_BLOCK FB_TemperatureGroupHmi EXTENDS FB_BaseMdHmi
```

 **Eigenschaften**



Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die HighHigh-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmHigh	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die High-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die Low-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLowLow	BOOL	Get	FALSE	Gruppe (mindestens ein Kanal) hat die LowLow-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmState	PlasticStatusHmi	Get	0	Alarm Status zur Darstellung mit einem TchMI-StateIndicator
BootAsScheduled	BOOL	Get, Set	FALSE	Die Temperaturgruppe soll im zeitgeplanten Betriebsmodus starten
ConfigHash	T_SHA256	Get	0x0	Hash-Wert der geladenen Gruppen-Konfiguration
ConfigNote	STRING(127)	Get, Set	“	Notiz zur geladenen Gruppen-Konfiguration
DoTune	BOOL	Get, Set	FALSE	Startet das Tuning aller aktiven Zonen der Gruppe.
GroupName	STRING	Get, Set	“	Name der Temperaturgruppe
Index	INT	Get	0	Index (ID) der Gruppe
IsOff	BOOL	Get	FALSE	Gruppe ist ausgeschaltet
IsOn	BOOL	Get	FALSE	Gruppe ist eingeschaltet
IsScheduled	BOOL	Get	FALSE	Gruppe ist im zeitgeplanten Betriebsmodus
IsStandby	BOOL	Get	FALSE	Gruppe ist im Standby-Betrieb

Name	Typ	Zugriff	Initialwert	Beschreibung
SetOff	BOOL	Get, Set	FALSE	Gruppe ausschalten
SetOn	BOOL	Get, Set	FALSE	Gruppe einschalten
SetScheduled	BOOL	Get, Set	FALSE	Gruppe in den zeitgeplanten Modus versetzen
SetStandby	BOOL	Get, Set	FALSE	Gruppe in den Standby-Betrieb versetzen
TuningActive	BOOL	Get	FALSE	Das Tuning der Gruppe ist aktiv
TuningDone	BOOL	Get	FALSE	Das Tuning der Gruppe ist abgeschlossen.
TuningFailed	BOOL	Get	FALSE	Das Tuning der Gruppe ist fehlgeschlagen
ZonesCount	INT	Get	0	Anzahl an Zonen in der Gruppe

 **Methoden**

Name	Beschreibung
CollectRemainingSave()	[INTERNAL] Fragt ab, ob eine Eigenschaft, welche Persistent gespeichert werden soll, gesetzt (.Set) wurde
UpdateState()	[INTERNAL] Setzt den aktiven Betriebsmodus

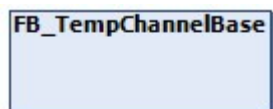
 **Schnittstellen**

Typ	Beschreibung
I_TemperatureGroupHmi	Standardschnittstelle auf FB_TemperatureGroupHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.5 FB\_TempChannelBase



Basisklasse für einen einzelnen Temperaturkanal

**Syntax:**

```
FUNCTION_BLOCK FB_TempChannelBase EXTENDS FB_BaseHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Index	INT	Get, Set	0	Referenziert einen PlasticBaseApplication Kanal zu einer TF8540 Zone.

**i Klasse enthält deutlich mehr Eigenschaften als aufgelistet**

Die Eigenschaften der FB\_TempChannelBase Klasse überschneiden sich mit den TF8540 Globalen Variablen aaaPfwTempToHmi, aaaPfwTempMparamFromHmi, aaaPfwTempPparamFromHmi und stPfwTempCtrl. Für weitere Informationen zu den überschneidenden Eigenschaften wird empfohlen, die TF8540-Dokumentation zu nutzen.

**Schnittstellen**

Name	Beschreibung
I_TempZoneHmi	Mit FB_TempZoneHMI kompatible Schnittstelle.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

### 5.6.6 FB\_TempChannel



Klasse zur Einzelsteuerung eines Temperaturkanals.

**Syntax:**

```
FUNCTION_BLOCK FB_TempChannel EXTENDS FB_TempChannelBase
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmNoResponse	BOOL	Get	FALSE	Temperaturwert des Kanals reagiert nicht auf die Ansteuerung.
AttachableMdInterface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
ConfigID	UINT	Get, Set	0	ID der aktiven Gruppen-Konfiguration
EnableLimitAlarms	BOOL	Get, Set	FALSE	Das Überschreiten eines Toleranzwertes/-limits löst einen Alarm aus.
Error	BOOL	Get	FALSE	Alias zu 'Fault'
ExtruderID	INT	Get, Set	0	ID des verwendeten Extruders
Fault	BOOL	Get	FALSE	Es ist ein Fehler in diesem Temperaturkanal aufgetreten
GroupID	INT	Get, Set	0	ID der zugewiesenen Gruppe
IsActive	BOOL	Get	FALSE	Kanal ist aktiv geregelt
IsEnabled	BOOL	Get	FALSE	Kanal ist freigegeben
IsLinked	BOOL	Get	FALSE	Kanal ist einer Gruppe zugewiesen
IsStandby	BOOL	Get	FALSE	Kanal ist im Standby-Betrieb
SupplyID	INT	Get	0	ID des verwendeten Versorgungskanals
TuningLastExecution	DATE_AND_TIME	Get, Set	DT#1900-01-01T00:00:00Z	Datum der letzten erfolgreichen Durchführung eines Tunings
TuningRequired	BOOL	Get, Set	TRUE	Die aktuelle Konfiguration erfordert ein Autotuning

 **Methoden**

Typ	Beschreibung
EnablePLC()	Gibt den Temperaturkanal PLC-seitig frei
Force()	Zwingt den Temperaturkanal auf Heiz-/Kühlleistung 100%.
Standby()	Stellt den Temperaturkanal auf Standby

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
400x	Temperaturbetrieb Überwachung	Alarm

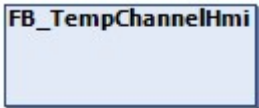
 **Schnittstellen**

Typ	Beschreibung
I_TempChannel	Standardschnittstelle auf FB_TempChannel
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten
I_TempZone	Kompatible Schnittstelle mit FB_TempZone

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.7 FB\_TempChannelHmi



HMI-Parallelklasse zur FB\_TempChannel Klasse

**Syntax:**

```
FUNCTION_BLOCK FB_TempChannelHmi EXTENDS FB_TempChannelBase
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
GroupID	INT	Get, Set	0	ID der zugewiesenen Gruppe

**Schnittstellen**

Typ	Beschreibung
I_TempChannelHmi	Standardschnittstelle auf FB_TempChannelHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.8 FB\_TempRecipe



Rezeptverwaltungsklasse der Temperaturregelung.

**Syntax:**

```
FUNCTION_BLOCK FB_TempRecipe EXTENDS FB_Recipe
```

**Lokale HMI-Variablen**

Name	Datentyp	Beschreibung
Setpoint	LREAL	Temperatur Sollwert
Standby	LREAL	Temperatur-Standby Sollwert
ThresholdM	LREAL	Negative innere Toleranz
ThresholdMM	LREAL	Negative äußere Toleranz
ThresholdP	LREAL	Positive innere Toleranz
ThresholdPP	LREAL	Positive äußere Toleranz

 **Methoden**

Name	Beschreibung
Init(ipChannel)	Initialisierung zur Verknüpfung mit einem Temperaturkanal

 **Schnittstellen**

Typ	Beschreibung
I_Recipe	Standardschnittstelle auf FB_Recipe.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.9 FB\_TempSupply



Implementiert eine Versorgungseinheit der Temperaturregelung

**Syntax:**

```
FUNCTION_BLOCK FB_TempSupply EXTENDS FB_BaseHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Index	INT	Get, Set	0	Index der Versorgungseinheit

 **Methoden**

Typ	Beschreibung
Line(x)	Gibt den x-ten Versorgungskanal der Versorgungsgruppe zurück

**Vermeidung von Exceptions**

**i** Die Listen-Zugriffsfunktionen (e.g. Channels(x), Groups(x), etc...) geben bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

 **Schnittstellen**

Typ	Beschreibung
I_TempSupply	Standardschnittstelle auf FB_TempSupply

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.10 FB\_TempSupplyLine



Implementiert eine Versorgungseinheit der Temperaturregelung

### Syntax:

FUNCTION\_BLOCK FB\_TempSupplyLine EXTENDS FB\_BaseHmi

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
ActSupplyCurrent	LREAL	Get	0.0	Ist-Strom des Kanals
ActSupplyLoad	LREAL	Get	0.0	Ist-Leistung des Kanals
AttachableMdInterface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
CalcErrorCurrent	LREAL	Get	0.0	Berechneter Fehlerstrom des Kanals
CalcSupplyLoad	LREAL	Get	0.0	Berechnete Leistung des Kanals
CalcSupplyMatch	LREAL	Get	0.0	Abweichung der berechneten und aktuellen Leistung des Kanals
Frequency	LREAL	Get	0.0	Frequenz des Kanals
PwmCycleTime	LREAL	Get, Set	0.0	Zykluszeit des PWM-Signals
PwmFactorC	INT	Get, Set	0	Faktor PWM-Zykluszeit von Kühlung zur Heizung
PwmMaxOnTime	LREAL	Get, Set	0.1	Maximale PWM-Einschaltzeit
PwmMaxOnTimeC	LREAL	Get, Set	0.1	Maximale PWM-Einschaltzeit der Kühlung
PwmMaxRampLoad	LREAL	Get, Set	0.0	Reserved
PwmMinOnTime	LREAL	Get, Set	0.0	Minimale PWM Einschaltzeit
SupplyID	INT	Get, Set	0	ID des Kanals

### Schnittstellen

Typ	Beschreibung
I_TempSupplyLine	Standardschnittstelle auf FB_TempSupplyLine
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.11 FB\_TempSchedule



Klasse für die Verwendung von Wochentag-Timers im Zusammenhang mit der Temperaturregelung

**Syntax:**

```
FUNCTION_BLOCK FB_TempSchedule EXTENDS FB_TimerWeekdayMaster
```

 **Methoden**

Typ	Beschreibung
MemberSchedule(x)	Gibt ein Interface zum ersten Timer zurück, der der Gruppe (Member) x zugewiesen ist
MemberScheduledActive(x)	Gibt ein Interface zum ersten Timer zurück, der der Gruppe (Member) x zugewiesen und zurzeit aktiv ist

 **Vermeidung von Exceptions**

**i** Die Member-Zugriffsfunktionen geben bei fehlender Überschneidung mit dem Member x eine Dummy-Instanz zurück.

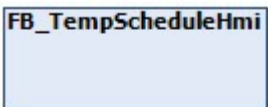
 **Schnittstellen**

Typ	Beschreibung
I_TempSchedule	Standardschnittstelle auf FB_TempSchedule
I_OneTaskInterface	Runtime Schnittstelle für eine PLC-Task

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.6.12 FB\_TempScheduleHmi



HMI-Parallelklasse zur FB\_TempSchedule Klasse.

**Syntax:**

```
FUNCTION_BLOCK FB_TempScheduleHmi EXTENDS FB_BaseMdHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Timer <sup>1</sup>	REFERENCE TO ARRAY[] OF FB_TimerWeekdayHmi	Get	-	Schnittstelle auf die einzelnen Termine der Zeitplanung

<sup>1</sup> Obsolete

 **Methoden**

Typ	Beschreibung
Timers(x)	Gibt ein Interface zum x-ten Timer zurück



**Vermeidung von Exceptions**

Die Listen-Zugriffsfunktionen (e.g. Channels(x), Groups(x), etc...) geben bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.6.13 FB\_TimerTempHmi**



Erweiterung des FB\_TimerWeekdayHmi um zusätzliche Temperatur-relevante Attribute.

**Syntax:**

```
FUNCTION_BLOCK FB_TimerTempHmi EXTENDS FB_TimerWeekdayHmi
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Interface auf die Maschinendatenkomponente
MembersCount	INT	Get	0	Anzahl an zugewiesenen Teilnehmern (Gruppen)
Standby	BOOL	Get, Set	FALSE	Timer soll als Standby-Betrieb gewertet werden

**Methoden**

Typ	Beschreibung
Clear()	Leert die Liste an Teilnehmern (Gruppen)
Exists(x)	Überprüft ob Teilnehmer (Gruppe) x dem Timer zugewiesen ist
Members() [▶ 74]	Gibt eine Schnittstelle auf den Teilnehmer zurück
Subscribe(x)	Lässt Gruppe x zum Teilnehmer des Timers werden

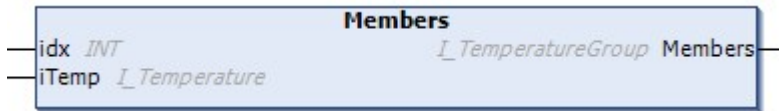
**Schnittstellen**

Typ	Beschreibung
I_TimerTempHmi	Standardschnittstelle auf FB_TimerTempHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

### 5.6.13.1 Members()



Gibt eine Schnittstelle auf den idx-ten Teilnehmer (Gruppe) zurück

**Syntax:**

```
METHOD Members : I_TemperatureGroup
VAR_INPUT
    idx: INT;
    iTemp: I_Temperature;
END_VAR
```

**Eingänge**

Name	Typ	Beschreibung
Idx	INT	Index des Teilnehmers in der Liste alle Teilnehmer
iTemp	I_Temperature	Referenz der Temperaturregelung zur Ermittlung der Temperaturgruppen-Instanz

**Ausgänge**

Name	Typ	Beschreibung
Members	I_TemperatureGroup	Angefragte Temperaturgruppe

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.7 Recipe - Rezeptverwaltung

Die Rezeptverwaltung des TF8550 Recipe Helpers baut auf der Implementation der [FB\\_PlcStateToHmi](#) [▶ 74] Klasse und der implementierten Handshake-Prozedur auf. Hierfür ist die Klasse bereits in der GVL HmiCommunication instanziiert.

```
{attribute 'qualified_only'}
VAR_GLOBAL
// =====
// communication PLC <> HMI
    fbPlcStateToHmi: FB_PlcStateToHmi;
// =====
END_VAR
```

### 5.7.1 FB\_PlcStateToHmi



Implementiert die Handshake-Prozedur für die TF8560 Rezeptverwaltung auf Seite der PLC.

**Syntax:**

FUNCTION\_BLOCK FB\_PlcStateToHmi EXTENDS FB\_BaseHmi

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung	PLC / HMI <sup>1</sup>
AxesInitialised	BOOL	Get, Set	FALSE	Achsen wurden initialisiert, das Rezept darf initialisiert werden.	PLC
ClientID	UINT	Get, Set	0	ID des aktuell verbundenen HMI-Clients	HMI
DataReqFailed	BOOL	Get, Set	FALSE	Fehler in der Kommunikation. Die PLC wartet auf ein Reset.	PLC
DataRequestPlc	BOOL	Get, Set	FALSE	Anforderung von Daten während der Hochlaufphase oder eines Produktwechsels.	PLC
DataRequestPlcActiveHmi	BOOL	Get, Set	FALSE	Schreiben von Rezeptdaten in die PLC aktiv.	HMI
DataRequestPlcQuitHmi	BOOL	Get, Set	FALSE	Schreiben von Rezeptdaten in die PLC abgeschlossen.	HMI
DataValidPlc	BOOL	Get, Set	FALSE	Die Rezeptdaten wurden vollständig geladen.	PLC
LiveSignHMI	UINT	Get, Set	0	Wechselnder Wert des HMI-Clients zur Signalisierung einer aktiven Verbindung	HMI
PlcInitialized	BOOL	Get, Set	FALSE	Alias zu AxesInitialized	PLC
ProductChangeConfirmPlc	BOOL	Get, Set	FALSE	Die Anfrage zum Produktwechsel wird akzeptiert.	PLC
ProductChangeEnable	BOOL	Get, Set	FALSE	Signalisiert dem HMI die Möglichkeit zum Wechsel des Rezeptes.	PLC
ProductRequestHmi	BOOL	Get, Set	FALSE	Anfragen eines Produktwechsels	HMI
Reset	BOOL	Get, Set	FALSE	Anfrage zur Reinitialisierung des Rezeptes	HMI
SaveDataQuitPlc	BOOL	Get, Set	FALSE	Bestätigt die Bearbeitung einer Rezeptspeicherung.	HMI
SaveDataRequestPlc	BOOL	Get, Set	FALSE	Anfrage zum Speichern des Rezeptes	PLC
VersionBaseApplication	STRING	Get	'v0.0.0.0'	Version der Plastic Base Application	PLC
VersionPalsticFunctions	STRING	Get	'v0.0.0.0'	Version der Plastic Technology Functions (TF8560)	PLC

<sup>1</sup>Die Spalte PLC/HMI beschreibt die Zuordnung des Schreibzugriffs.

 **Methoden**

Name	Beschreibung
DeclareBeta()	Erweitert die Anzeigeversion der Plastic Base Application um eine beta-Etikette.

## Schnittstellen

Typ	Beschreibung
I_RecipeState	Schnittstelle für die in der Hauptprozedur zu verarbeitende Handshake-Variablen

### Voraussetzungen

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.7.2 I\_RecipeState

Schnittstelle auf die PLC-Variablen der TF8550 Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper Klasse  
FB\_PlcStateToHmi

### Syntax:

```
INTERFACE I_RecipeState EXTENDS I_BaseEmpty
```

## Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
DataRequestPlc	BOOL	Get	FALSE	Anforderung von Daten während der Hochlaufphase oder eines Produktwechsels.
DataValidPlc	BOOL	Get	FALSE	Die Rezeptdaten wurden vollständig geladen.
PlcInitialized	BOOL	Get, Set	FALSE	Daten wurden initialisiert, das Rezept darf initialisiert werden.
ProductChangeEnable	BOOL	Get, Set	FALSE	Signalisiert dem HMI die Möglichkeit zum Wechsel des Rezeptes.
Reset	BOOL	Get, Set	FALSE	Anfrage zur Reinitialisierung des Rezeptes

## Methoden

Name	Beschreibung
DeclareBeta()	Erweitert die Anzeigeversion der Plastic Base Application um eine beta-Etikette.

### Voraussetzungen

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.7.3 FB\_Recipe



Rezeptverwaltungsklasse zur Zusammenfassung von strukturierten Rezeptwerten. Diese werden in integrierten TcHMI Controls verwendet.



**Klasse ist abstrakt**

Da die Klasse als `ABSTRACT` definiert ist, kann die Klasse nicht instanziiert werden und muss mittels Vererbung implementiert werden.

**Syntax:**

```
FUNCTION_BLOCK ABSTRACT FB_Recipe EXTENDS FB_Base
```

**Methoden**

Name	Beschreibung
Lock()	Sperrt die Rezeptvariablen <ul style="list-style-type: none"> <li>Die nächste Ausführung von <code>.Update()</code> schreibt die Rezeptwerte in die PLC</li> <li>Kann aus der HMI aufgerufen/ausgeführt werden</li> </ul>
Reset()	Setzt den Sperrzustand zurück
Update()	[ABSTRACT] Aktualisiert die Rezeptvariablen auf den aktuellen PLC-Wert <ul style="list-style-type: none"> <li>Kann aus der HMI aufgerufen/ausgeführt werden</li> </ul>

**Schnittstellen**

Typ	Beschreibung
I_Recipe	Standardschnittstelle auf <code>FB_Recipe</code> .

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.8 MachineData - Maschinendaten

**Base**

Im Unterordner `\MachineData\Base\` sind die Basis-Klassen des Maschinendaten-Handlings enthalten.

**Klassen**

Name	Beschreibung
<a href="#">FB_MdBaseContainer</a> [ <a href="#">79</a> ]	Stellt den Basiscontainer für Maschinendaten dar.
<a href="#">FB_MdBaseComponent</a> [ <a href="#">81</a> ]	Stellt die Basisklasse für Maschinenkomponenten dar.

**Components**

Im Unterordner `\MachineData\Components\` sind alle bereits implementierten Maschinendaten-Komponenten enthalten. Alle enthaltenen Komponenten erben von der `FB_MdBaseComponent`-Klasse und verfügen über eine Enumeration, welche den Index (ohne Offset) der einzelnen Parameter definiert.

 **Klassen**

Name	Verwendung	Beschreibung
FB_MdFileInfo	FB_MdBaseContainer	Enthält die Informationen einer gespeicherten Datei.
FB_MdAnalogValue	FB_Monitoring FB_Setpoints FB_Blowing	Enthält Skalierungsparameter für Analogwerte .
FB_MdBlowpin	FB_Blowpin	Enthält Blowpin-spezifische Parameter.
FB_MdClamp	FB_Clamp	Enthält Clamp-spezifische Parameter.
FB_MdWtc	FB_Wtc	Enthält Wtc-spezifische Parameter.
FB_MdAxis	FB_Axis	Enthält allgemeine Achsparameter.
FB_MdContinuousMotion	FB_Extruder	Enthält Parameter für Achsen mit durchgehender Drehbewegung.
FB_MdHoming	FB_Homing	Enthält Einstellungsparameter für Homing-Prozeduren.
FB_MdNc	FB_NcFunctions	Enthält NC-spezifische Parameter.
FB_MdPtpMotion	FB_PtpMotion	Enthält Parameter für PTP-basierte Bewegungen.
FB_MdTrafo	FB_TrafoFunctions	Enthält Parameter für transformierende Achsen.
FB_MdTempSupply	FB_TempSupply	Enthält Parameter für eine Temperatur-Versorgungseinheit.
FB_MdTempZone	FB_TempZone	Enthält Parameter für eine Temperaturzone.
FB_MdWeekdayTiming	FB_TimerWeekdayHmi	Enthält Daten über den TimeSchedule des Weekday Timers.

**Erstellen eigener Maschinendaten-Komponenten**

**i** Für das Erstellen eigener Maschinendaten-Komponenten gilt der Standard der TF8560-Maschinendaten. Eine neue Komponente muss dabei die abstrakten Methoden MdNextParameter() und MdSetParameter() implementieren sowie einige interne Variablen setzen. Hierfür wird empfohlen die bestehenden Klassen als Vorlage zu nutzen und den Anweisungen der TF8560-Dokumentation zu folgen.

**Hmi**

Im Unterordner \MachineData\Hmi\ sind Klassen definiert, die die pro Komponente gespeicherten Daten für das HMI zusammenfassen. Hierbei werden ausschließlich Referenz-Zugriffe (Interfaces) genutzt und keinerlei Daten lokal in die Klasse kopiert. Verwendet werden diese unter anderem im HMI auf den Konfigurations-Seiten.

**Subitems**

Im Unterordner \MachineData\Subitems\ sind mehrere Klassen, Schnittstellen und Enumerationen enthalten, die zur Funktionalität der Maschinendaten-Klassen beitragen.

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.8.1 FB\_MdBaseContainer



Die Klasse erweitert den TF8560 FB\_MdContainer um das Maschinendaten-Encoding der TwinCAT 3 Plastic Application. Darüber hinaus werden auftretende Fehler durch boolesche Werte auswertbar und die Dateiablage vereinfacht.

### Syntax:

```
FUNCTION_BLOCK FB_MdBaseContainer EXTENDS FB_MdContainer
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AllowFolderCreation	BOOL	Get, Set	TRUE	Erlaubt dem Container das Erzeugen der Ordnerstruktur zum Ablegen der Maschinendaten.
CrcInvalid	BOOL	Get	FALSE	Der aufgetretene Fehler wurde durch eine falsche CRC-Checksumme ausgelöst.
CreationDate	DATE_AND_TIME	Get	DT#1970-1-1-0:0:0	Datum der ersten Erzeugung einer Datei
Details	I_MdBaseContainerDetails	Get	THIS^	Fasst die Eigenschaften der Datei-Details zusammen.
ErrorDetails	I_MdBaseContainerErrors	Get	THIS^	Fasst die Eigenschaften der Fehlerinformationen zusammen.
ErrorId	UDINT	Get	0	Fehler ID des zuletzt aufgetretenen Fehlers
FileAccessDenied	BOOL	Get	FALSE	Der aufgetretene Fehler wurde durch fehlende Dateizugriffsrechte ausgelöst.
FileNotFound	BOOL	Get	FALSE	Der aufgetretene Fehler wurde durch das Fehlen der Datei ausgelöst.
IgnoreMissmatches	BOOL	Get, Set	FALSE	Zwingt den Container eine Datei trotz Versionskollision zu laden.
MissmatchDetected	BOOL	Get	FALSE	Der aufgetretene Fehler wurde durch einen Versionskonflikt ausgelöst.
MissmatchBaseApp	BOOL	Get	FALSE	Der aufgetretene Versionskonflikt besteht in den Versionen der TwinCAT Base Application Bibliothek.
MissmatchIdxFormat	BOOL	Get	FALSE	Der aufgetretene Versionskonflikt besteht in der Version der Index-Kodierung.
StoreCount	UDINT	Get	0	Zählt die Iterationen an Dateischreibvorgängen seit Erstellung der Datei.
StoreDate	DATE_AND_TIME	Get	DT#1970-1-1-0:0:0	Datum des letzten Speichervorgangs
UnknownParameter	BOOL	Get	FALSE	Der aufgetretene Fehler wurde durch einen unbekanntem Parameter in der Datei ausgelöst.
VersionBaseApp	I_LibVersion	Get	-	Version der Plastic Base Application Bibliothek
VersionIdxFormat	I_IdxFormatVersion	Get	-	Version der Index-Kodierung



 **Methoden**

Name	Beschreibung
AddComponent()	Fügt dem Container eine weitere Komponente hinzu.
CompareFileVersion()	[INTERNAL] Vergleicht die übergebenen Versionen mit der definierten Version im Source-Code.
CreateFilepath()	Erzeugt einen neuen Dateipfad basierend auf dem Container-Namen
OverwriteFromFile()	[INTERNAL] Wird von der FB_MdFileInfo Klasse verwendet, um die geladene Version im Container zu aktualisieren.

 **Schnittstellen**

Typ	Beschreibung
I_MdBaseContainer	Standardschnittstelle auf FB_MdBaseContainer
I_MdBaseContainerDev	Erweiterte Schnittstelle mit Zugriff auf die als „[INTERNAL]“ gekennzeichneten Methoden
I_MdBaseContainerDetails	Schnittstelle auf die Eigenschaften mit dateibezogenen Informationen (z. B. CreationDate)
I_MdBaseContainerErrors	Schnittstelle auf die Eigenschaften mit Fehlerinformationen

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.5)

## 5.8.2 FB\_MdBaseComponent



Die Klasse erweitert die Standardkomponente FB\_MdComponent der TF8560-Utilities um Funktionen der Index-Kodierung der Plastic Base Application.

**Syntax:**

```
FUNCTION_BLOCK FB_MdBaseComponent EXTENDS FB_MdComponent
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
LinkedContainer	I_MdBaseContainer	Get	NULL	Verweist auf den Container, dem die Komponente angehängt wurde.

 **Methoden**

Name	Beschreibung
ConfigCompType()	Dient bei erbenden Klassen zur Initialisierung des Kodierungs-Offsets nach E_StandardCompType.
ConfigCompTypeEx()	Dient bei der erbenden Klasse zur Initialisierung des Kodierungs-Offsets.

Beide Methoden sind als **PROTECTED** deklariert und können nur innerhalb der Klasse genutzt werden.

 **Schnittstellen**

Typ	Beschreibung
I_MdBaseComponent	Standardschnittstelle auf FB_MdBaseComponent

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.8.3 FB\_MdCollection



Die Klasse kann eine Sammlung von Komponenten eines Klassen-Typs speichern. Beim Anhängen der Klasse an die FB\_BaseRuntime können Komponenten von Objekten, die die Schnittstelle I\_AttachableMdInterface implementieren, gesammelt und der Kollektion angehängt werden.

**Syntax:**

```
FUNCTION_BLOCK FB_MdCollection EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
MachineData	I_MdBaseContainer	Get	-	Schnittstelle zum internen Maschinendaten-Container

 **Methoden**

Name	Beschreibung
AddComponent()	Fügt eine Komponente der Kollektion hinzu: <ul style="list-style-type: none"> <li>• Zählt den .ComponentIndex der angehängten Komponenten automatisch hoch</li> <li>• Verweigert Komponenten die nicht mit dem Prototyp übereinstimmen</li> </ul>
CheckType()	[PROTECTED] Gibt eine unter den unterstützten Typen verwendete ID zurück
SetPrototype()	Fixiert den Typ der Komponente die von der Runtime gesammelt werden soll

 **Schnittstellen**

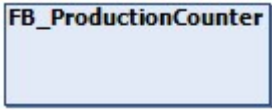
Typ	Beschreibung
I_MdCollection	Standardschnittstelle auf FB_MdCollection
I_OneTaskInterface	Runtime Schnittstelle für eine PLC-Task

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.9 OperationData - Produktionsdaten und Statistiken

### 5.9.1 FB\_ProductionCounter



Implementiert einen allgemeinen Zähler für fertigungsabhängige Daten wie z.B. einen Stück- oder ein Produktionszeitzähler.

**Syntax:**

```
FUNCTION_BLOCK FB_ProductionCounter EXTENDS FB_BaseMd
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
SavingInterval	LREAL	Get, Set	10.0	Intervall für das Abspeicherns des Zählerstands in einer Maschinendatendatei
SinceLastReset	I_ProductionDataComponent	Get	-	Zählerstand seit dem letzten Ausführen der Reset() Methode
SinceProducing	I_ProductionDataComponent	Get	-	Zählerstand seit Start des aktuellen Zählvorgangs
SinceStart	I_ProductionDataComponent	Get	-	Zählerstand seit Maschinenstart
Total	I_ProductionDataComponent	Get	-	Zählerstand seit Beginn der Maschinenproduktionszeit

**Methoden**

Name	Beschreibung
NewPart()	Loggt ein neues Teil für den Stückzähler.
Producing()	Zeigt an, dass die Produktion aktiv läuft.

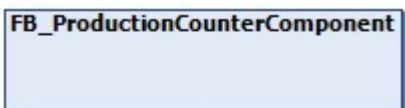
**Schnittstellen**

Typ	Beschreibung
I_ProductionCounter	Standardschnittstelle auf FB_ProductionCounter

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.9.2 FB\_ProductionCounterComponent



Enthält die Zähler-Daten für einen definierten Zeitraum.

**Syntax:**

```
FUNCTION_BLOCK FB_ProductionDataComponent EXTENDS FB_MdBaseComponent
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
OperationTime	LREAL	Get, Set	0.0	Produktionszeit
Parts	ULINT	Get, Set	0	Stückzahl
PartsPerHour	LREAL	Get, Set	0	Teile pro Stunde
TimePerCycle	LREAL	Get, Set	0.0	Produktions-Zykluszeit
ComponentType	USINT	Get	0	Typ der Komponente als Identifikator
Connected	BOOL	Get	FALSE	Komponente ist mit seinen Zugriffsabhängigkeiten verbunden.

 **Methoden**

Name	Beschreibung
Connect()	Verbindet Referenzen mit der Klasse.

 **Ereignisgesteuerte Methoden (Callback-Methoden)**

Name	Beschreibung
MdNextParameter()	Container-Abruf zum Speichern der Parameter in eine Datei
MdSetParameter()	Container-Abruf zum Laden der Parameter in die Laufzeit

 **Schnittstellen**

Typ	Beschreibung
I_ProductionCounterComponent	Standardschnittstelle auf FB_ProductionCounterComponent

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.10 EventLogger - Logging von Ereignissen und Fehlern

### 5.10.1 FB\_AlarmHandler



Die Klasse implementiert ein Handling von anliegenden Alarmen zur Einflussnahme auf den Prozessablauf. Dadurch kann beispielsweise auf das Anliegen eines Alarms mit der Severity „Error“ mit einem direkten Stopp der Maschine reagiert werden.

**Syntax:**

```
FUNCTION_BLOCK FB_AlarmHandler EXTENDS FB_ListenerBase2
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmCritical	BOOL	Get	FALSE	Es liegt ein kritischer Alarm an.
AlarmError	BOOL	Get	FALSE	Es liegt ein Fehler-Alarm an.
AlarmWarning	BOOL	Get	FALSE	Es liegt ein Warnungs-Alarm an.
AlarmInfo	BOOL	Get	FALSE	Es liegt ein Informations-Alarm an.
Error	BOOL	Get	FALSE	Der Handler ist in einem Fehlerzustand.

 **Methoden**

Name	Beschreibung
ClearAlarms()	Bestätigt und bereinigt alle anliegenden Alarme.

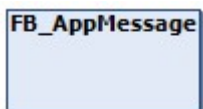
 **Schnittstellen**

Typ	Beschreibung
I_AlarmHandler	Standardschnittstelle auf FB_AlarmHandler
I_OneTaskInterface	Runtime Schnittstelle für eine PLC-Task

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.10.2 FB\_AppMessage



Diese Klasse ist Bestandteil der FB\_Base Klasse und bietet eine Schnittstelle zum EventLogger für nahezu jedes TC3 Plastic Base Application Objekt.

**Syntax:**

```
FUNCTION_BLOCK EXTENDS FB_Message
```

 **Methoden**

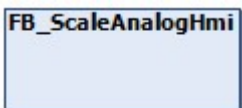
Name	Beschreibung
ClearAlarm()	Bereinigt einen spezifischen Alarm.
ConfirmAlarm()	Quittiert einen spezifischen Alarm.
Reset()	Setzt alle von dieser Instanz erzeugten Alarme zurück.
SendAlarm()	Löst einen Alarm aus.
SendEqualMessage()	Loggt eine Nachricht im EventLogger.
SendHresult()	Loggt eine Nachricht im EventLogger mit der Hexa-Dezimaldarstellung der Fehlercodes.
SendMessage()	Loggt eine Nachricht im EventLogger, außer die Nachricht wurde bereits zuvor gesendet.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.11 Analog - Analogwertskalierung

### 5.11.1 FB\_ScaleAnalogHmi



Implementiert Funktionen zum Skalieren von Analogwerten mit Parametrierung über das HMI.

**Syntax:**

```
FUNCTION_BLOCK FB_ScaleAnalogHmi EXTENDS FB_BaseHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ParamAnalogValue	REFERENCE TO FB_MdAnalogValueHmi	Get	-	Schnittstelle zur Parametrierung der Analogwertskalierung
ScalingElectricalMax	LREAL	Get, Set	10.0	Maximum des elektrischen Spannungseingangs
ScalingElectricalMin	LREAL	Get, Set	0.0	Minimum des elektrischen Spannungseingangs
ScalingRawMax	LREAL	Get, Set	32767	Maximum des rohen Analogmesswertes (bit)
ScalingRawMin	LREAL	Get, Set	0	Minimum des rohen Analogmesswertes (bit)
ScalingValueMax	LREAL	Get, Set	100.0	Maximum der skalierten Endeinheit
ScalingValueMin	LREAL	Get, Set	0.0	Minimum der skalierten Endeinheit

 **Methoden**

Name	Beschreibung
ElectricalToRaw()	Skaliert die elektrische Spannung auf den rohen Analogmesswert (bit).
ElectricalToValue()	Skaliert die elektrische Spannung auf die Einheit der Endskalierung.
RawToElectrical()	Skaliert den rohen Analogmesswert (bit) auf die elektrische Spannung.
RawToValue()	Skaliert den rohen Analogmesswert (bit) auf die Einheit der Endskalierung.
ValueToElectrical()	Skaliert die Einheit der Endskalierung auf die elektrische Spannung.
ValueToRaw()	Skaliert die Einheit der Endskalierung auf den rohen Analogmesswert (bit).

 **Schnittstellen**

Typ	Beschreibung
I_ScaleAnalogHmi	Standardschnittstelle auf FB_ScaleAnalogHmi
I_ScaleAnalogScalings	Schnittstelle auf die Skalierungsminima und -maxima

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.11.2 FB\_Monitoring



Implementiert eine Überwachungsfunktion für Analogsignale. Dabei können die Schwellwerte für die Überwachung variabel auf der HMI eingestellt werden.

**Syntax:**

```
FUNCTION_BLOCK FB_Monitoring EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
AutoClearAlarms	BOOL	Get, Set	FALSE	Ausgelöste Alarmer werden nach beheben der Ursache automatisch bereinigt.
EnableAlarms	BOOL	Get, Set	FALSE	Das Unter-/Überschreiten löst vorimplementierte Alarmer aus.
Input	I_InputBase	Get, Set	NULL	Schnittstelle des einzulesenden Analogsignals
Value	LREAL	Get, Set	0	Gibt den skalierten Analogwert zurück, kann gesetzt werden, wenn kein Input zugewiesen wurde.

 **Mögliche Ereignisse (Events)**

ID	Beschreibung	Alarm/Message
300x	Über-/Unterschreitung der eingestellten Toleranzen	Alarm

 **Schnittstellen**

Typ	Beschreibung
I_Monitoring	Standardschnittstelle auf FB_Monitoring
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten
I_OneTaskInterface	Runtime Schnittstelle für eine PLC-Task

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.11.2.1 FB\_MonitoringTemp**



Implementiert die Überwachung eines Temperaturkanals mit direkter Verbindung zu TF8540. Die einstellbaren Limitierungen werden mit der Temperaturzone synchronisiert.

**Syntax:**

```
FUNCTION_BLOCK FB_MonitoringTemp EXTENDS FB_Monitoring
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
TempChannel	I_TempChannel	Get, Set	NULL	Zugewiesener Temperaturkanal

 **Schnittstellen**

Typ	Beschreibung
I_MonitoringTemp	Standardschnittstelle auf FB_MonitoringTemp

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.11.2.2 FB\_MonitoringIPC**



Implementiert die Überwachung der CPU Temperatur eines Beckhoff IPCs.



**Syntax:**

```
FUNCTION_BLOCK FB_MonitoringIPC EXTENDS FB_Monitoring
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
RefreshRate	LREAL	Get, Set	5.0	Rate [s] der asynchronen Abfrage des IPC Wertes

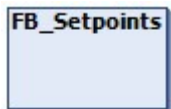
 **Schnittstellen**

Typ	Beschreibung
I_MonitoringIPC	Standardschnittstelle auf FB_MonitoringIPC

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.11.3 FB\_Setpoints



Implementiert eine Skalierung von Analogsollwerten, die über das HMI eingestellt werden können.

**Syntax:**

```
FUNCTION_BLOCK FB_Setpoints EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
Output	I_OutputBase	Get, Set	NULL	Schnittstelle des zuzuweisenden Analogausgang

 **Schnittstellen**

Typ	Beschreibung
I_Setpoints	Standardschnittstelle auf FB_Setpoints
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten
I_OneTaskInterface	Runtime Schnittstelle für eine PLC-Task

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.12 Timer - Prozesstiming

### 5.12.1 FB\_TimerHmi



HMI-Schnittstelle für IEC 61131-3 Timer zum Einstellen von Prozesstimings.

#### Syntax:

```
FUNCTION_BLOCK FB_TimerHmi EXTENDS FB_BaseHmi
```

#### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
ActualValue	LREAL	Get, Set	0.0	Aktueller Zeitwert in Sekunden
LatchedValue	LREAL	Get, Set	0.0	Gespeicherter Zeitwert der letzten Ausführung in Sekunden
Out	BOOL	Get, Set	FALSE	Timer-Ausgang (abhängig von TON, TOF, TP)
SetValue	LREAL	Get, Set	0.0	Vorgegebener Zeitsollwert in Sekunden

#### Schnittstellen

Typ	Beschreibung
I_TimerHmi	Standardschnittstelle auf FB_TimerHmi

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.12.2 FB\_TimerTon



Implementiert die Funktion des IEC 61131-3 Timers mit integrierter Schnittstelle für PLC und HMI.

#### Syntax:

```
FUNCTION_BLOCK FB_TimerTon EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Elapsed	LREAL	Get	0.0	Aktueller Zeitwert in Sekunden
Latched	LREAL	Get	0.0	Gespeicherter Zeitwert der letzten Ausführung in Sekunden
Preset	LREAL	Get, Set	0.0	Vorgegebener Zeitwert in Sekunden
Et	TIME	Get	T#0ms	Aktueller Zeitwert in Millisekunden
In	BOOL	Get, Set	FALSE	Aktivierungseingang
L	TIME	Get	T#0ms	Gespeicherter Zeitwert der letzten Ausführung in Millisekunden
Pt	TIME	Get, Set	T#0ms	Vorgegebener Zeitwert in Millisekunden
Q	BOOL	Get	FALSE	Timer-Ausgang

 **Schnittstellen**

Typ	Beschreibung
I_Timer	Allgemeine Schnittstelle für IEC 61131-3 Timer

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.12.3 FB\_TimerTof



Implementiert die Funktion des IEC 61131-3 Timers mit integrierter Schnittstelle für PLC und HMI.

**Syntax:**

```
FUNCTION_BLOCK FB_TimerTof EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Elapsed	LREAL	Get	0.0	Aktueller Zeitwert in Sekunden
Latched	LREAL	Get	0.0	Gespeicherter Zeitwert der letzten Ausführung in Sekunden
Preset	LREAL	Get, Set	0.0	Vorgegebener Zeitwert in Sekunden
Et	TIME	Get	T#0ms	Aktueller Zeitwert in Millisekunden
In	BOOL	Get, Set	FALSE	Aktivierungseingang
L	TIME	Get	T#0ms	Gespeicherter Zeitwert der letzten Ausführung in Millisekunden
Pt	TIME	Get, Set	T#0ms	Vorgegebener Zeitwert in Millisekunden
Q	BOOL	Get	FALSE	Timer-Ausgang

 **Schnittstellen**

Typ	Beschreibung
I_Timer	Allgemeine Schnittstelle für IEC 61131-3 Timer

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.12.4 FB\_TimerTp



Implementiert die Funktion des IEC 61131-3 Timers mit integrierter Schnittstelle für PLC und HMI.

**Syntax:**

```
FUNCTION_BLOCK FB_TimerTp EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Elapsed	LREAL	Get	0.0	Aktueller Zeitwert in Sekunden
Latched	LREAL	Get	0.0	Gespeicherter Zeitwert der letzten Ausführung in Sekunden
Preset	LREAL	Get, Set	0.0	Vorgegebener Zeitwert in Sekunden
Et	TIME	Get	T#0ms	Aktueller Zeitwert in Millisekunden
In	BOOL	Get, Set	FALSE	Aktivierungseingang
L	TIME	Get	T#0ms	Gespeicherter Zeitwert der letzten Ausführung in Millisekunden
Pt	TIME	Get, Set	T#0ms	Vorgegebener Zeitwert in Millisekunden
Q	BOOL	Get	FALSE	Timer-Ausgang

 **Schnittstellen**

Typ	Beschreibung
I_Timer	Allgemeine Schnittstelle für IEC 61131-3 Timer

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.12.5 FB\_TimerWeekdayHmi



HMI-Schnittstelle für einen Wochentimer zur Nutzung mit dem TF8550 Control TimeScheduler.

**Syntax:**

```
FUNCTION_BLOCK FB_TimerWeekdayHmi EXTENDS FB_TimerHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInterface	I_MdComponent	Get	-	Interface auf die Maschinendatenkomponente
Duration	LREAL	Get	0.0	Zeitdifferenz zwischen Start- und Endzeitpunkt
EndTime	LREAL	Get, Set	0.0	Endzeitpunkt in Sekunden
StartTime	LREAL	Get, Set	0.0	Startzeitpunkt in Sekunden
Weekday	E_Weekday [ <a href="#">▶ 94</a> ]	Get, Set	eNone	Wochentag des Startzeitpunkts

## Schnittstellen

Typ	Beschreibung
I_TimerWeekday	Standardschnittstelle auf FB_TimerWeekdayMaster
I_TimerWeekdayHmiQuery	Schnittstelle für interne Anreihung mehrerer Wochen-Timer
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.12.5.1 E\_Weekday

Aufzählung der Wochentage an denen ein Zeitplan starten soll.

#### Syntax:

```

TYPE E_Weekday :
(
  // invalid - inactive
  eNone          := 2#0000_0000,

  // One day
  eMonday        := 2#0000_0001,
  eTuesday       := 2#0000_0010,
  eWednesday     := 2#0000_0100,
  eThursday      := 2#0000_1000,
  eFriday        := 2#0001_0000,
  eSaturday      := 2#0010_0000,
  eSunday        := 2#0100_0000
)

```

#### Werte

Name	Beschreibung
eNone	Inaktiv bzw. kein Wochentag
eMonday	Montag
eTuesday	Dienstag
eWednesday	Mittwoch
eThursday	Donnerstag
eFriday	Freitag
eSaturday	Samstag
eSunday	Sonntag

### 5.12.6 FB\_TimerWeekdayMaster

**FB\_TimerWeekdayMaster**

Implementiert einen Wochentimer, kompatibel zu IEC 61131-3 Timern.

#### Syntax:

```

FUNCTION_BLOCK FB_TimerWeekdayMaster EXTENDS FB_BaseMd

```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Elapsed	LREAL	Get	0.0	Aktueller Zeitwert in Sekunden
Latched	LREAL	Get	0.0	Gespeicherter Zeitwert der letzten Ausführung in Sekunden
Preset	LREAL	Get, Set	0.0	Vorgegebener Zeitwert in Sekunden
Et	TIME	Get	T#0ms	Aktueller Zeitwert in Millisekunden
In	BOOL	Get, Set	FALSE	Aktivierungseingang
L	TIME	Get	T#0ms	Gespeicherter Zeitwert der letzten Ausführung in Millisekunden
Pt	TIME	Get, Set	T#0ms	Vorgegebener Zeitwert in Millisekunden
Q	BOOL	Get	FALSE	Timer-Ausgang

 **Schnittstellen**

Typ	Beschreibung
I_TimerWeekday	Standardschnittstelle auf FB_TimerWeekdayMaster

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.13 BlowMolding - Blasformspezifische Klassen

### 5.13.1 FB\_Blowing



Implementiert einen typischen Blasablauf in zwei Blasphasen mit einstellbarem Druck.

**Syntax:**

```
FUNCTION_BLOCK FB_Blowing EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AttachableMdInteface	I_MdComponent	Get	-	Schnittstelle auf die Maschinendatenkomponente
Done	BOOL	Get	FALSE	Der Blasablauf wurde erfolgreich durchgeführt.
EnableOutput	BOOL	Get, Set	FALSE	Gibt die Ausgabe des Blasdrucks frei.
Output	I_OutputBase	Get, Set	NULL	Stellt die Schnittstelle zur Analogausgabe dar.

 **Methoden**

Name	Beschreibung
Start()	Startet den Blasablauf.

 **Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Name	Beschreibung
BlowSeq() [▶ 96]	Prozedur zur Blasdruckausgabe

 **Schnittstellen**

Typ	Beschreibung
I_Blowing	Standardschnittstelle auf FB_Blowing
I_AttachableMdInterface	Schnittstelle für containerlose Maschinendaten-Komponenten

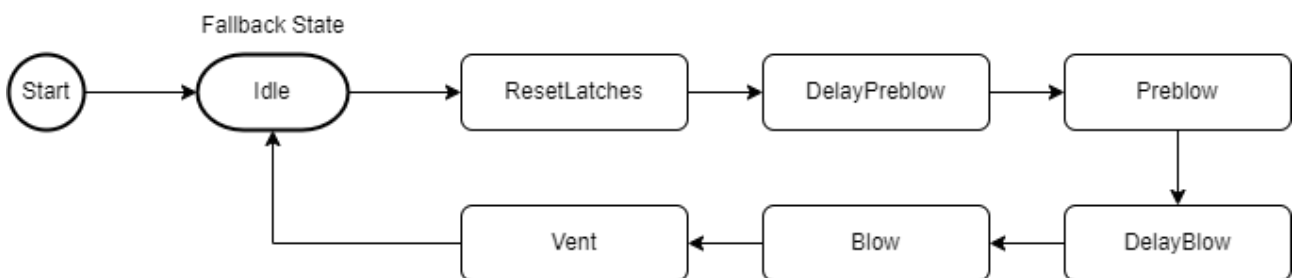
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.13.1.1 BlowSeq()**

Intanztyp	Instanzname
Master	fbBlowSeq
Slaves	aBaseSeqMembers[E_BlowingSequence.eLength]

**State-Diagram:**





**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.13.2 FB\_IntervalBlowing



Erweitert die FB\_Blowing Klasse um ein wechselndes Blasintervall während der Hauptblasphase.

**Syntax:**

```
FUNCTION_BLOCK FB_IntervalBlowing EXTENDS FB_Blowing
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
EnableInterval	BOOL	Get, Set	-	Schaltet die Intervall-Erweiterung ein.

**Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Name	Beschreibung
BlowSeq() [▶ 97]	Prozedur zur Blasdruckausgabe (erweitert)

**Schnittstellen**

Typ	Beschreibung
I_IntervalBlowing	Standardschnittstelle auf FB_IntervalBlowing

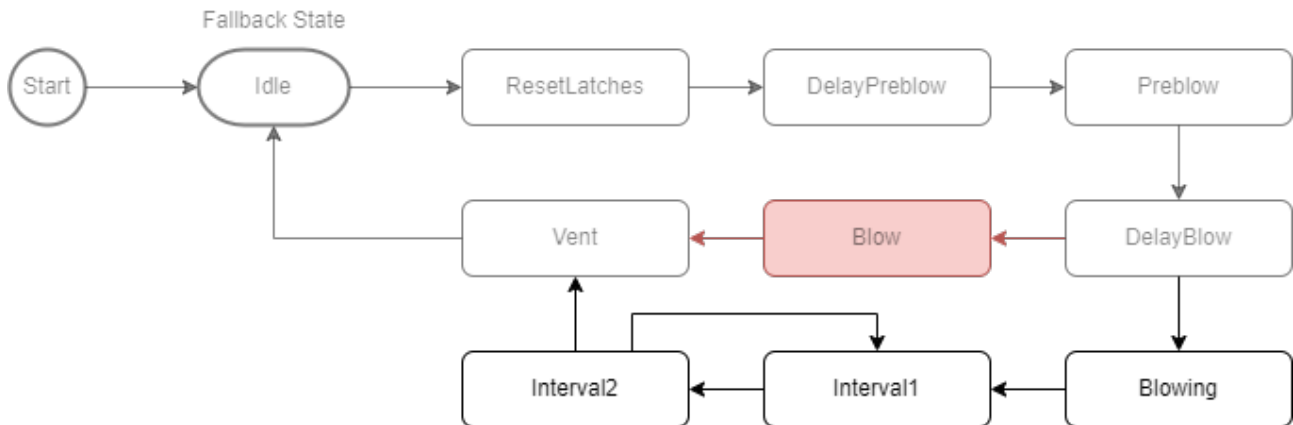
**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.13.2.1 BlowSeq()

Intanztyp	Instanzname
Master	fbBlowSeq
Slaves	fbStateBlowing fbStateIntervall1 fbStateInverval2

**State-Diagram:**



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.13.3 FB\_Blowpin



Implementiert Blowpin-spezifische Funktionen.

**Syntax:**

```
FUNCTION_BLOCK FB_Blowpin EXTENDS FB_PtpMotion
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DisableHoldingTorque	BOOL	Get, Set	FALSE	Sperrt das Haltedrehmoment bei NC-Blowpin Achsen.

**Prozedursteuernde Methoden (FB AdaptableSequence [▶ 103])**

Name	Beschreibung
PtpSeq() [▶ 99]	Prozedur zum Laden einer PTP-Bewegung (erweitert)

**Schnittstellen**

Typ	Beschreibung
I_Blowpin	Standardschnittstelle auf FB_Blowpin

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

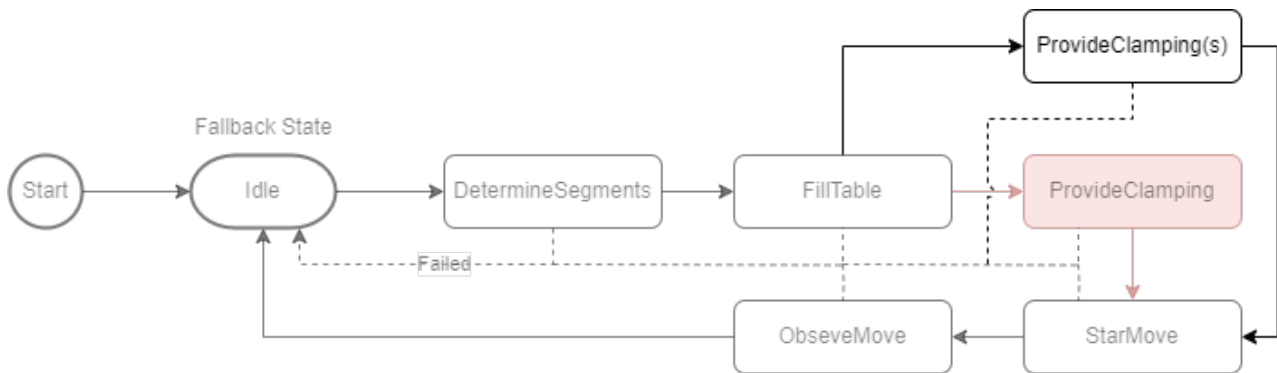
### 5.13.3.1 PtpSeq()

**i** Blowpin-PTP-Variante nur bei NC-Achsen aktiv

Der fbSeqProvideClamping wird nur bei der Verwendung einer NC-Achse eingefügt. Für hydraulische Achsen wird das normale Clamping der FB\_PtpMotion Klasse verwendet.

Intanztyp	Instanzname
Master	fbPtpSeq
Slaves	fbSeqProvideClamping

**State-Diagram:**



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

### 5.13.4 FB\_BlowMoldingExtruder



Implementiert spezifische Funktionen für Extruder im Blasformprozess.

**Syntax:**

```
FUNCTION_BLOCK FB_BlowMoldingExtruder EXTENDS FB_Extruder
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ParisonLengthControl [▶ 100]	I_ParisonLengthContro I	Get	-	Schnittstelle auf die integrierte Schlauchlängenregelung

**Schnittstellen**

Typ	Beschreibung
I_BlowMoldingExtruder	Standardschnittstelle auf FB_BlowMoldingExtruder

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.13.4.1 FB\_ParisonLengthControl**



Implementiert die Schlauchlängenregelung im typischen Anwendungsfall des Blasformprozesses.

**Syntax:**

```
FUNCTION_BLOCK FB_ParisonLengthControl EXTENDS FB_Base
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Photocell	BOOL	Set	FALSE	Eingang für das Signal der Photozelle
TurnrateDown	BOOL	Get	FALSE	Steuerausgang zum Absenken der Drehzahl
TurnrateUp	BOOL	Get	FALSE	Steuerausgang zum Anheben der Drehzahl
WtcStart	BOOL	Get, Set	FALSE	Eingangssignal für den Start eines neuen Zyklus

**Methoden**

Name	Beschreibung
Activate()	Aktiviert die Schlauchlängenregelung.

**Schnittstellen**

Typ	Beschreibung
I_ParisonLengthControl	Standardschnittstelle auf FB_ParisonLengthControl

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.13.5 FB\_Wtc**



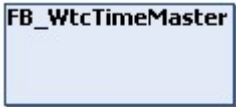
Implementiert die Wanddickensteuerung für den extrudierten Schlauch in Blasformanwendungen.

**Syntax:**

```
FUNCTION_BLOCK FB_Wtc EXTENDS FB_Axis
```



### 5.13.6 FB\_WtcTimeMaster



Implementiert einen Zeitgeber (Timer) als Master für das WTC-Camming.

**Syntax:**

```
FUNCTION_BLOCK FB_WtcTimeMaster EXTENDS FB_Axis
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ActualTime	LREAL	Get	0.0	Abgelaufene Zeit seit Start des WTC-Zyklus in Sekunden
ActualTimeLatched	LREAL	Get	0.0	Gespeicherte Zeit des letzten WTC-Zyklus in Sekunden
FirstStart	BOOL	Get	FALSE	Die WTC befindet sich in ihrem ersten Zyklus seit dem letzten Start.
GuidingValue	LREAL	Get	0.0	Zeitwert als resultierenden Camming Master-Wert
ProfileStarted	BOOL	Get	FALSE	TRUE, wenn der Timer gestartet wurde.

**Methoden**

Name	Beschreibung
ProfileStartAck()	Setzt das Feedback-Signal „ProfileStarted“ zurück.
Start()	Startet den Timer.

**Schnittstellen**

Typ	Beschreibung
I_WtcTimeMaster	Standardschnittstelle auf FB_WtcTimeMaster
I_WtcMaster	Allgemeine Schnittstelle einer WTC-Masterachse

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.13.7 FB\_WtcAccuMaster



Implementiert einen Akkumulator als Master für das WTC-Camming.

**Syntax:**

```
FUNCTION_BLOCK FB_WtcAccuMaster EXTENDS FB_Axis
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
FillingDone	BOOL	Get	FALSE	Der Akkumulator hat das Füllvolumen erreicht.
GuidingValue	LREAL	Get	0.0	Position als resultierenden Camming Master-Wert
PushoutDone	BOOL	Get	FALSE	Der Ausstoß ist abgeschlossen (das verbleibende Füllvolumen ist unter dem Puffervolumen).

 **Methoden**

Name	Beschreibung
Filling()	Startet den Füllprozess des Akkumulators.
Pushout()	Startet das Ausdrücken des Füllvolumens.

 **Schnittstellen**

Typ	Beschreibung
I_WtcAccuMaster	Standardschnittstelle auf FB_WtcAccuMaster
I_WtcMaster	Allgemeine Schnittstelle einer WTC-Masterachse

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.14 Utilities - Hilfsmittel

### 5.14.1 FB\_AdaptableSequence



Ermöglicht die variable Erweiterung von implementierten Prozeduren einer Klasse. Die Klasse selbst dient zur Verwaltung aller Prozedurschritte und zur Indikation des Zustands eines einzelnen Schrittes mit Referenz zur Verwaltung (Master).

**Syntax:**

```
FUNCTION_BLOCK FB_AdaptableSequence
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
ActiveSeq	I_AdaptableSeqItf	Get	THIS^	Schnittstelle auf den aktuellen Schritt der Prozedur
Done	BOOL	Get, Set	FALSE	Zeigt die erfolgreiche Abarbeitung eines Prozedur-Schrittes an.
Failed	BOOL	Get, Set	FALSE	Zeigt die gescheiterte Abarbeitung eines Prozedur-Schrittes an.
Index	BOOL	Get	INT	Index der Klasse in Bezug auf die Gesamtprozedur
IsActive	BOOL	Get	FALSE	Die Klasse ist als aktueller Prozedurschritt aktiv.
IsMaster	BOOL	Get	TRUE	Die Klasse ist das Verwaltungsobjekt einer Prozedur.
Length	INT	Get	0	Länge der Liste an angehängten Prozedurschritten
Next	I_AdaptableSeqItf	Get	NULL	Schnittstelle auf den nächsten Schritt der Prozedur
Tag	I_FlexValue	Get	-	Beliebiger Wert zum Speichern von benutzerdefinierten Informationen

 **Methoden**

Name	Beschreibung
<a href="#">Check() [▶ 105]</a>	Überprüft, ob der aktuelle Schritt der Prozedur abgearbeitet wurde.
<a href="#">Clear() [▶ 105]</a>	Löscht alle Prozedurschritte aus dem Master.
<a href="#">Exists() [▶ 106]</a>	Überprüft, ob ein Prozedurschritt im Master enthalten ist.
<a href="#">Idx() [▶ 106]</a>	Gibt den x-ten Schritt aus der Prozedur zurück.
<a href="#">Insert() [▶ 107]</a>	Fügt einen weiteren Schritt der Prozedur hinzu.
<a href="#">Jump() [▶ 108]</a>	Fordert das Springen in einen nicht folgenden Prozedurschritt an.
Reset()	Slave: Setzt den Zustand des Prozedurschritts zurück. Master: Setzt den Fortschritt der Prozedur zurück.
SetMaster()	[INTERNAL] Weist dem Prozedurschritt einen Master zu. Wird von der Insert() Methode genutzt und muss nicht separat aufgerufen werden.

 **Schnittstellen**

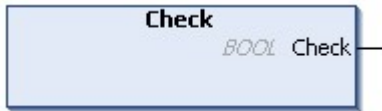
Typ	Beschreibung
I_AdaptableSeqItf	Schnittstelle zur Implementation als Slave
I_AdaptableSeqState	Schnittstelle zum Auslesen des Prozedurschritt-Zustands
I_AdaptableSeqQuery	Erweiterte Schnittstelle für die Bearbeitung durch den Master
I_AdaptableSeqMaster	Schnittstelle zur Implementation als Master



**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.14.1.1 Check()**



Überprüft, ob der aktuelle Schritt der Prozedur abgearbeitet wurde. Wenn dies der Fall ist, bewirkt die Methode ebenfalls das Umschalten auf den nächsten Prozedurschritt.

**● Verwendung ist bereits integriert**

**i** Diese Methode wird bei bestehenden Instanzen in der TwinCAT 3 Plastic Base Application bereits implementiert. Es wird nicht empfohlen dieser Methode zur Überprüfung innerhalb eines Prozedurschrittes zu verwenden. Ein Aufruf der Funktion ist ausschließlich bei Neuaufsetzen einer Prozedur notwendig.

**Syntax:**

```
METHOD Check : BOOL
```

**🚪 Ausgänge**

Name	Typ	Beschreibung
Check	BOOL	TRUE, wenn der aktuelle Schritt abgeschlossen wurde und zum nächsten Schritt übergegangen wird.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.1.2 Clear()**



Löscht alle Prozedurschritte aus dem Master.

**Syntax:**

```
METHOD Clear : HRESULT
```

**🚪 Ausgänge**

Name	Typ	Beschreibung
Clear	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.1.3 Exists()**



Überprüft, ob ein Prozessschritt im Master enthalten ist.

**Syntax:**

```
METHOD Exists : HRESULT
VAR_INPUT
    iSeq:          I_AdaptableSeqItf;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
iSeq	I_AdaptableSeqItf	Zu überprüfender Prozessschritt

 **Ausgänge**

Name	Typ	Beschreibung
Exists	HRESULT	Rückgabewert mit Feedback zum Erfolg der Überprüfung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.1.4 Idx()**



Gibt den x-ten Schritt aus der Prozedur zurück.

**Syntax:**

```
METHOD Idx : I_AdaptableSeqItf
VAR_INPUT
    nIdx:          INT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nIdx	INT	Der angeforderte Index

 **Ausgänge**

Name	Typ	Beschreibung
Idx	I_AdaptableSeqItf	Gefundener Schritt der Prozedur

**Ungültige Indizes**

**I** Zur Vermeidung von Exceptions, verursacht durch einen ungültigen Wert auf dem Eingang nIdx, gibt die Funktion im Fehlerfall den Master zurück. Daher sollte die Methode vergleichbar einem Array-Index behandelt werden, indem die Gesamtlänge der Prozedurschrittliste berücksichtigt wird.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.1.5 Insert()**



Fügt einen weiteren Schritt der Prozedur hinzu.

**Syntax:**

```
METHOD Insert : HRESULT
VAR_INPUT
    iCurrent:      I_AdaptableSeqItf;
    iNew:          I_AdaptableSeqItf;
    bOverwrite:    BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
iCurrent	I_AdaptableSeqItf	Der aktuelle Teilnehmer, der hinter den neuen Teilnehmer verschoben werden soll.
iNew	I_AdaptableSeqItf	Neuer hinzuzufügender Teilnehmer
bOverwrite	BOOL	TRUE, wenn der aktuelle Teilnehmer nicht verschoben, sondern ersetzt werden soll.

 **Ausgänge**

Name	Typ	Beschreibung
Insert	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.14.1.6 Jump()



Fordert das Springen in einen nicht folgenden Prozedurschritt an.

**● Anfrage führt nicht direkt zur Ausführung**

**i** Wenn die Methode ausgeführt wird, wird lediglich die Anfrage im Master hinterlegt. Zum Ausführen des Sprungs muss die Eigenschaft Done oder Failed des aktiven Prozessschrittes gesetzt werden und die Methode Check() aufgerufen werden.

Siehe die [Check\(\) \[▶\\_105\]](#) Methode für weitere wichtige Hinweise.

**Syntax:**

```
METHOD Jump : HRESULT
VAR_INPUT
    iTarget:      I_AdaptableSeqItf;
END_VAR
```

**👉 Eingänge**

Name	Typ	Beschreibung
iTarget	I_AdaptableSeqItf	Prozedurschritt zu dem gesprungen werden soll

**👉 Ausgänge**

Name	Typ	Beschreibung
Jump	HRESULT	Rückgabewert mit Feedback zum Erfolg der Anfrage

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.14.1.7 I\_AdaptableSeqExt

Ermöglicht die Implementation von Prozedurschritten außerhalb der prozedurimplementierenden Klasse

**Syntax:**

```
INTERFACE I_AdaptableSeqExt EXTENDS I_BaseEmpty
```

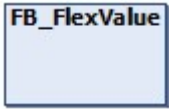
**⚡ Ereignisgesteuerte Methoden (Callback-Methoden)**

Name	Beschreibung
ExtAdaptSeq()	Wird von der implementierten Prozedur aufgerufen, um die Prozedurschritte außerhalb zu bearbeiten

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

## 5.14.2 FB\_FlexValue



Klasse (FB) vergleichbar mit dem strukturierten Datentyp ST\_FlexValue aus TF8560. Stellt eine Variable ohne festgelegten Datentyp dar.

### Syntax:

```
FUNCTION_BLOCK FB_FlexValue
```

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
Value	U_FlexValue	Get	0	Wert als Union (alle Typen)
ValueType	E_FlexValue	Get	eBOOL	Zuletzt zugewiesener Wert-Datentyp (außer wenn über Union zugewiesen)
_BOOL	BOOL	Get, Set	FALSE	Wert als BOOL
_INT	INT	Get, Set	0	Wert als INT
_LREAL	LREAL	Get, Set	0.0	Wert als LREAL

### Schnittstellen

Typ	Beschreibung
I_FlexValue	Standardschnittstelle auf FB_FlexValue

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.14.3 FB\_Parameter



Ermöglicht die Verkettung von mehreren 64bit Floating Point (LREAL) Werten mit zuweisbarem Namen.

### Syntax:

```
FUNCTION_BLOCK FB_Parameter EXTENDS FB_Base
```

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
Name	STRING	Get, Set	"	Name des Parameters
Next	I_Parameter	Get, Set	NULL	Nächstes Element in der Liste
Root	I_Parameter	Get, Set	THIS^	Erstes Element der Liste
Value	LREAL	Get, Set	0.0	Wert des Parameters

 **Methoden**

Name	Beschreibung
Idx()	Gibt das Element der x-ten Stelle der Liste zurück.

**● Vermeidung von Exceptions**

**i** Die Idx() Funktion gibt bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

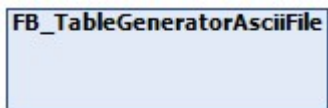
 **Schnittstellen**

Typ	Beschreibung
I_Parameter	Standardschnittstelle auf FB_Parameter
I_ParameterQuery	Erweiterung der I_Parameter Schnittstelle mit Set-Zugriff auf Next- und Root-Eigenschaften

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

### 5.14.4 FB\_TableGeneratorAsciiFile



Implementiert einen Datei-Lesemechanismus für eine benutzerdefinierte Transformationstabelle.

**Formatierung der Datei:**

Antrieb            Last

```
0.0            100.0
1.0            110.0
2.0            121.0
3.0            131.0
4.0            142.0
...
```

**● Antriebspunkte müssen äquidistant sein**

**i** Die Punkte der Antriebsseite müssen äquidistant sein! Andernfalls verschieben sich alle Punkte zwischen Start und Ende auf äquidistante Abstände, was unerwünschte Ungenauigkeit zur Folge hat.

**Syntax:**

```
FUNCTION_BLOCK FB_TableGeneratorAsciiFile EXTENDS Tc3_PlasticFunctions.FB_TrafoTableGenerator
```

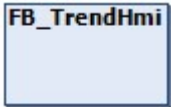
 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
FilePath	STRING	Get, Set	“	Dateipfad auf dem Zielsystem zur abgelegten Beschreibungsdatei
LoadHighEnd	LREAL	Get	0	Gelesener höchster Punkt der Lastseite
LoadLowEnd	LREAL	Get	0	Gelesener niedrigster Punkt der Lastseite

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.14.5 FB\_TrendHmi**



Unterstützungsklasse für den TwinCAT HMI SQLiteTrend mit vorimplementierten Ansichten. Zur Verwendung ist die TF8550-Funktion `TcHmi.Functions.Plastic.UpdateTrend()` erforderlich. Die ausgewählte Ansicht wird zusätzlich über die Eigenschaft `SelectedView` bestimmt. Folgende Objekte mit entsprechender Schnittstelle können dem Trend angehängt werden:

Typ	Anzeigewert
I_ExtruderHmi	Ist-Drehzahl des Extruders [RPM]
I_MonitoringHmi	Überwachungswert [ <i>any</i> ]
I_TempChannel	Ist-Temperatur eines Temperaturkanals [°C]
I_Temperature	Ist-Temperatur jedes gruppierten Temperaturkanals [°C]
I_TempCtrlHmi <sup>1</sup>	Temperatur jeder Zone [°C]

<sup>1</sup> Ist obsolete

**Interne Funktionen:**

- Die ersten x Ansichten werden zur Darstellung der Temperaturgruppen vorreserviert
  - Default x := 5
  - Automatisch anpassend zur eingestellten Anzahl an Gruppen [[▶ 122](#)]
  - Bei Änderungen der FB Temperature [[▶ 51](#)]-Gruppenkonfiguration passen sich die Ansichten automatisch mit an
- Alle per `Append()` angehängten Werte sind zur Konfiguration durch die `ConfigXYZ()` Methoden verfügbar

**Syntax:**

```
FUNCTION_BLOCK FB_TrendHmi EXTENDS FB_Base
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
SelectedView	INT	Get, Set	0	Ausgewählte Ansicht

**Methoden**

Name	Beschreibung
<a href="#">Append() [▶ 112]</a>	Anhängen eines Wertes, der im Trend aufgezeichnet und in Ansichten ausgewählt werden soll.
<a href="#">AppendTempChannels() [▶ 113]</a>	Anhängen der Temperaturzonen einer FB_Temperature Instanz
<a href="#">CheckSupport()</a>	[PROTECTED] Überprüft die Unterstützung des angehängten Objektes
<a href="#">ConfigDisplayName() [▶ 114]</a>	Überschreibt den Instanznamen eines Anzeigewertes.
<a href="#">ConfigDisplayLocalisation() [▶ 114]</a>	Überschreibt den Instanznamen mit einem Lokalisierungsschlüssel.
<a href="#">ConfigDisplayLocalisationNuguet()</a>	Überschreibt den Instanznamen mit einem Lokalisierungsschlüssel aus dem TF8550.Localisation Paket.
<a href="#">ConfigView() [▶ 115]</a>	Konfiguriert einen Wert in einer auswählbaren Ansicht.
<a href="#">Idx()</a>	Gibt das Objekt an der x-ten Stelle zurück.
<a href="#">AppendTempZones() <sup>1</sup></a>	Anhängen der Temperaturzonen einer FB_TempCtrl Instanz.

<sup>1</sup> Ist obsolete

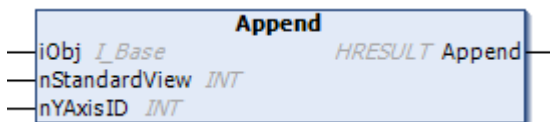
**Vermeidung von Exceptions**

**I** Die `Idx()` Funktion gibt bei ungültig angefordertem Index das erste Element (Root) der Liste zurück.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.14.5.1 Append()**



Hängt einen Wert dem Trend an.

**Syntax:**

```

METHOD Append : HRESULT
VAR_INPUT
    iObj:          I_Base;
    nStandardView: INT;
    nYAxisID:      INT;
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
<code>iObj</code>	<code>I_Base</code>	Objekt das in den Trend mit aufgenommen werden soll
<code>nStandardView</code>	<code>INT</code>	Standard-Ansicht in der der Wert angezeigt wird
<code>nYAxisID</code>	<code>INT</code>	Y-Achse in der HMI, auf der der Wert angezeigt wird

Die `YAxisID` ist für folgende Einheiten vordefiniert:



ID	Einheit	Beschreibung
1	°C	Temperatur
2	A	Strom
3	Bar	Druck
4	RPM	Drehzahl

 **Ausgänge**

Name	Typ	Beschreibung
Append	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.5.2 AppendTempChannels()**



Hängt die Temperaturkonfiguration dem Trend an. Falls sich die Gruppen-Konfiguration aller Zonen ändert, wird dies automatisch vom FB\_TrendHmi übernommen.

**Syntax:**

```
METHOD AppendTempChannels : HRESULT
VAR_INPUT
    ipTemp:          I_Temperature;
    nYAxisID:        INT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
ipTemp	I_Temperature	Instanz der Temperaturregelungs-Klasse (FB)
nYAxisID	INT	Y-Achse in der HMI, auf der der Wert angezeigt wird

Die YAxisID ist für folgende Einheiten vordefiniert:

ID	Einheit	Beschreibung
1	°C	Temperatur
2	A	Strom
3	Bar	Druck
4	RPM	Drehzahl

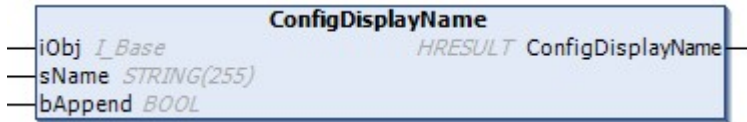
 **Ausgänge**

Name	Typ	Beschreibung
AppendTempChannels	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

**5.14.5.3 ConfigDisplayName()**



Passt den Anzeigenamen für einen Trendwert an.

**Syntax:**

```

METHOD ConfigDisplayName : HRESULT
VAR_INPUT
  iObj:          I_Base;
  sName:         STRING(255);
  bAppend:       BOOL;
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
iObj	I_Base	Objekt dessen Name angepasst werden soll
sName	STRING(255)	Anzuzeigende Zeichenfolge
bAppend	BOOL	Übergebene Zeichenfolge soll an den bestehenden Anzeigenamen angehängt werden.

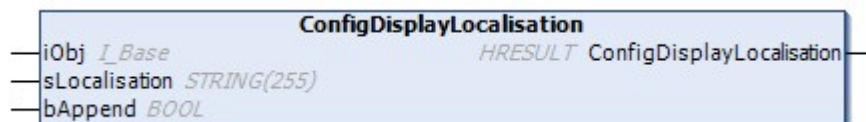
**Ausgänge**

Name	Typ	Beschreibung
ConfigDisplayName	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.5.4 ConfigDisplayLocalisation()**



Passt den Anzeigenamen für einen Trendwert an.

**Syntax:**

```

METHOD ConfigDisplayLocalisation : HRESULT
VAR_INPUT
  iObj:          I_Base;
  sLocalisation: STRING(255);
  bAppend:       BOOL;
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
iObj	I_Base	Objekt dessen Name angepasst werden soll
sLocalisation	STRING(255)	Anzuzeigender Lokalisierungsschlüssel
bAppend	BOOL	Übergebener Lokalisierungsschlüssel soll an den bestehenden Anzeigenamen angehängt werden.

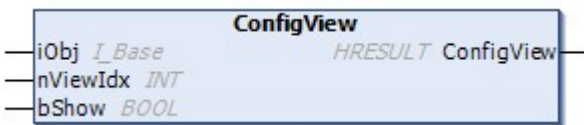
**Ausgänge**

Name	Typ	Beschreibung
ConfigDisplayLocalisation	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.5.5 ConfigView()**



Ändert die Zusammenstellung einer auswählbaren Ansicht.

**Syntax:**

```

METHOD ConfigView : HRESULT
VAR_INPUT
  iObj:          I_Base;
  nViewIdx:     INT;
  bShow:        BOOL;
END_VAR
  
```

**Eingänge**

Name	Typ	Beschreibung
iObj	I_Base	Objekt das in einer Ansicht angepasst werden soll
nViewIdx	INT	Ansicht (ID) die angepasst werden soll
bShow	BOOL	Objekt wird angezeigt (TRUE).

**Ausgänge**

Name	Typ	Beschreibung
ConfigView	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.14.6 FB\_Trigger



Kombiniert Trigger vom Typ R\_TRIG und F\_TRIG mit objektorientierter Schnittstelle.

### Syntax:

```
FUNCTION_BLOCK FB_Trigger
```

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
CLK	BOOL	Get, Set	FALSE	Abgetastetes Eingangssignal
FQ	BOOL	Get	FALSE	Eingangssignal hat eine fallende Flanke.
Q	BOOL	Get	FALSE	Eingangssignal hat eine steigende oder fallende Flanke.
RQ	BOOL	Get	FALSE	Eingangssignal hat eine steigende Flanke.

### Methoden

Name	Beschreibung
Cyclic()	Zyklusmethode, durch die das Signal abgetastet wird

### Schnittstellen

Typ	Beschreibung
I_Trigger	Standardschnittstelle auf FB_Trigger

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.14.7 FB\_LatchTrigger



Erweitert die Klasse FB\_Trigger um eine Speicherfunktion der Flanken.

### Syntax:

```
FUNCTION_BLOCK FB_LatchTrigger EXTENDS FB_Trigger
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
LF	BOOL	Get	FALSE	Eine fallende Flanke am Eingangssignal lag an.
LR	BOOL	Get	FALSE	Eine steigende Flanke am Eingangssignal lag an.

 **Methoden**

Name	Beschreibung
Reset()	Setzt LF und LR zurück.

 **Schnittstellen**

Typ	Beschreibung
I_LatchTrigger	Standardschnittstelle auf FB_LatchTrigger

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.14.8 FB\_LibVersion



Definiert den Aufbau der Versionsnummer einer Bibliothek

**Syntax:**

```
FUNCTION_BLOCK FB_LibVersion
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Build	UDINT	Get	0	Dritte Stelle der Versionsnummer
Major	UDINT	Get	0	Erste Stelle der Versionsnummer
Minor	UDINT	Get	0	Zweite Stelle der Versionsnummer
Released	BOOL	Get	FALSE	Version ist als ‚Released‘ gekennzeichnet
Revision	UDINT	Get	0	Vierte Stelle der Versionsnummer
Version	STRING	Get	'v0.0.0.0'	Versionsnummer als Ascii-String
Version3	STRING	Get	'v0.0.0'	.Version ohne die erste Stelle der Versionsnummer

 **Methoden**

Name	Beschreibung
IsEqualTo()	Vergleicht eine andere Version auf Übereinstimmung
IsNewerThan()	Überprüft, ob diese Version neuer als die übergebene Vergleichsversion ist
IsOlderThan()	Überprüft, ob diese Version älter als die übergebene Vergleichsversion ist
SetVersion()	Setzt die Versionsnummer
SetVersionStruct()	Setzt die Versionsnummer anhand einer Version vom Typ ST_LibVersion

 **Schnittstellen**

Typ	Beschreibung
I_LibVersion	Standardschnittstelle auf FB_LibVersion

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

## 5.14.9 FB\_LibVersionBeta



Definiert den Aufbau der Versionsnummer einer Bibliothek, zzgl. eines Beta-Tags. Die Eigenschaft `FB_LibVersionBeta.Version` präsentiert sich dadurch beispielsweise als 'v5.3.32.9-beta554'

**Syntax:**

```
FUNCTION_BLOCK FB_LibVersion
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Betalteration	UDINT	Get, Set	0	Bestimmt die Beta-Iteration der Version

 **Schnittstellen**

Typ	Beschreibung
I_LibVersionBeta	Standardschnittstelle auf FB_LibVersionBeta

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.6.0)

### 5.14.10 F\_SecondsToTime()



Konvertiert eine Zeit in Sekunden vom Typ LREAL in Millisekunden vom Typ TIME.

**Syntax:**

```
FUNCTION F_SecondsToTime : TIME
VAR_INPUT
    fTime:      LREAL;
END_VAR
```

**Eingänge**

Name	Typ	Beschreibung
fTime	LREAL	Zeitwert in Sekunden als Fließkommazahl

**Ausgänge**

Name	Typ	Beschreibung
F_SecondsToTime	TIME	Zeitwert in Millisekunden

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 5.14.11 F\_TimeToSeconds()



Konvertiert eine Zeit in Millisekunden vom Typ TIME in Sekunden vom Typ LREAL.

**Syntax:**

```
FUNCTION F_SecondsToTime : LREAL
VAR_INPUT
    tTime:      TIME;
END_VAR
```

**Eingänge**

Name	Typ	Beschreibung
tTime	TIME	Zeitwert in Millisekunden

**Ausgänge**

Name	Typ	Beschreibung
F_TimeToSeconds	LREAL	Zeitwert in Sekunden als Fließkommazahl

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.12 F\_GetCycleTime()**



Gibt die Zykluszeit der aufrufenden Task als LREAL Fließkommawert in Sekunden zurück.

**Syntax:**

```
FUNCTION F_GetCycleTime : LREAL
```

**Ausgänge**

Name	Typ	Beschreibung
F_GetCycleTime	LREAL	Zykluszeit in Sekunden als Fließkommawert

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**5.14.13 F\_TryDivide()**



Dividiert zwei Werte, ohne eine Exception auszulösen.

**● Rückgabewert mathematisch ungültig**

**i** Die Funktion definiert den mathematisch ungültigen Fall  $x / 0$  als 0. Dies ist ein mathematisch ungültiges Ergebnis, ist jedoch für viele Anwendungsfälle hinreichend. Prüfen Sie für ihren Anwendungsfall, ob diese Definition nicht zu unvorhergesehenem Fehlverhalten führt.

**Syntax:**

```
FUNCTION F_TryDivide : HRESULT
VAR_INPUT
    fNominator:    LREAL;
    fDenominator:  LREAL;
    refResult:     REFERENCE TO LREAL;
END_VAR
```

**Eingänge**

Name	Typ	Beschreibung
fNominator	LREAL	Wert der geteilt werden soll
fDenominator	LREAL	Wert durch den geteilt werden soll
refResult	REFERENCE TO LREAL	Ergebnis der Division



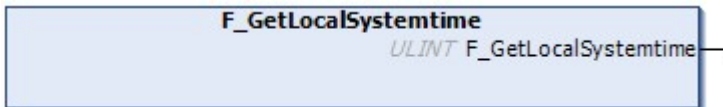
 **Ausgänge**

Name	Typ	Beschreibung
F_TryDivide	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

### 5.14.14 F\_GetLocalSystemtime



Gibt die lokale Systemzeit zurück, mit Berücksichtigung der Zeitzone. Ist geeignet für die Erstellung von systemzeitbezogenen Zeitstempeln.

**Syntax:**

```
FUNCTION F_GetLocalSystemtime : ULINT
```

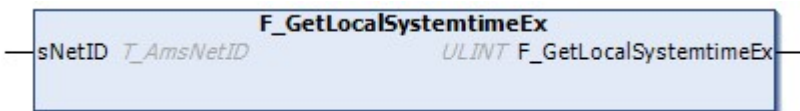
 **Ausgänge**

Name	Typ	Beschreibung
F_GetLocalSystemtime	ULINT	Systemuhrzeit, basierend auf der Definition von <u>T_FILETIME</u>

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

### 5.14.15 F\_GetLocalSystemtimeEx



Gibt die Systemzeit eines Gerätes mit AMS-NetID zurück, mit Berücksichtigung der Zeitzone. Ist geeignet für die Erstellung von systemzeitbezogenen Zeitstempeln.

**Syntax:**

```
FUNCTION F_GetLocalSystemtimeEx : ULINT
```

 **Eingänge**

Name	Typ	Beschreibung
sNetID	T_AmsNetID	Net-ID des auszulesenden Systems

## Ausgänge

Name	Typ	Beschreibung
F_GetLocalSystemtimeEx	ULINT	Systemuhrzeit, basierend auf der Definition von <code>T_FILETIME</code>

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.4)

## 5.14.16 PlasticStatusHmi

Status-Variable zur Verknüpfung mit einem HMI-StatusIndicator. Diese Variable wird Bit-weise interpretiert.

### Syntax:

```
TYPE PlasticStatusHmi : BYTE; END_TYPE
```

### Werte

bit	Beschreibung
0	Erfolgreich (Grün)
1	Warnung (Orange)
2	Fehler (Rot)
3	Reserviert
4	Reserviert
5	Reserviert
6	Reserviert
7	Zustand ist ungültig

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 5.15 Einstellparameter – Tc3\_PlasticBaseAppStaticParams

Parameterliste für die Nutzung der Tc3\_PlasticBaseApplication.

### Syntax:

```
{attribute 'qualified_only'}
VAR_GLOBAL CONSTANT
  // Static motion paramters
  {attribute 'hide'}
  cnNoOfTrafoPoints          :          INT := 181;
  {attribute 'TcHmiSymbol.Hide'}
  cnNoOfCammmingPoints     :          INT := 400;
  {attribute 'TcHmiSymbol.Hide'}
  cnMaxTrafoPoints         :          INT := 181;
  {attribute 'TcHmiSymbol.Hide'}
  cnMaxPtpPoints           :          INT := 6;
  {attribute 'TcHmiSymbol.Hide'}
  cnMaxMoveCluster         :          INT := 2;
  {attribute 'TcHmiSymbol.Hide'}
  cnMaxPtpCams             :          INT := 5;
```

```
// Static temperature parameters
{attribute 'TcHmiSymbol.Hide'}
cnTempGroups           :           INT := 10;
{attribute 'TcHmiSymbol.Hide'}
cnTempZonesPerGroup   :           INT := 20;
{attribute 'TcHmiSymbol.Hide'}
cnTempTimers          :           INT := 30;

// Trend
{attribute 'TcHmiSymbol.Hide'}
cnTrendSize           :           INT := 50;

// Runtime handling
{attribute 'TcHmiSymbol.Hide'}
cnRuntimeObjects      :           INT := 200;

// Machine Data
{attribute 'TcHmiSymbol.Hide'}
csHardDisk            :           STRING := 'C:';
{attribute 'TcHmiSymbol.Hide'}
csDataFolderName     :           STRING := 'Data\Machine';
{attribute 'TcHmiSymbol.Hide'}
csMachineName        :           STRING := 'Beckhoff';           // left empty for using object
name only in machine-data-filename
END_VAR
```

**Werte**

Name	Beschreibung	Default
cnNoOfCammingPoints	Standardpunkteanzahl einer Kurvenscheibe (z. B. für Nutzung des TF8550 CurveEditors)	400
cnMaxPtpPoints	Maximale Anzahl an PTP-Segmenten	6
cnMaxTrafoPoints	Maximale Anzahl an Transformationspunkten für den Buffer zum Laden einer Tabelle aus einer Datei	181
cnMaxMoveCluster	Maximale Anzahl an Clustern (Gruppierte PTP-Segmente)	2 (positiv/negativ)
cnMaxPtpCams	Maximale Anzahl an PTP-Nocken pro Cluster	5
cnTempGroups	Anzahl an verfügbaren Temperaturgruppen	10
cnTempZonesPerGroup	Anzahl an verfügbaren Zonen pro Temperaturgruppe	20
cnTempTimers	Anzahl an Wochentimings für die Zeitplanung der Temperaturregelung	30
cnTrendSize	Maximale Anzahl an historisierbaren Trendwerten	50
cnRuntimeObjects	Anzahl der an die Runtime anhängbaren Steuerungsobjekte	200
csHardDisk	Maschinendaten: Ziellaufwerk	C:
csDataFolderName	Maschinendaten: Zielordner auf dem Ziellaufwerk	Data\Machine
csMachineName	Maschinendaten: Kürzel zur Identifikation einer Maschinendatendatei	Beckhoff

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.42	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.5)

## 6 PLC Beispiele / Anleitungen

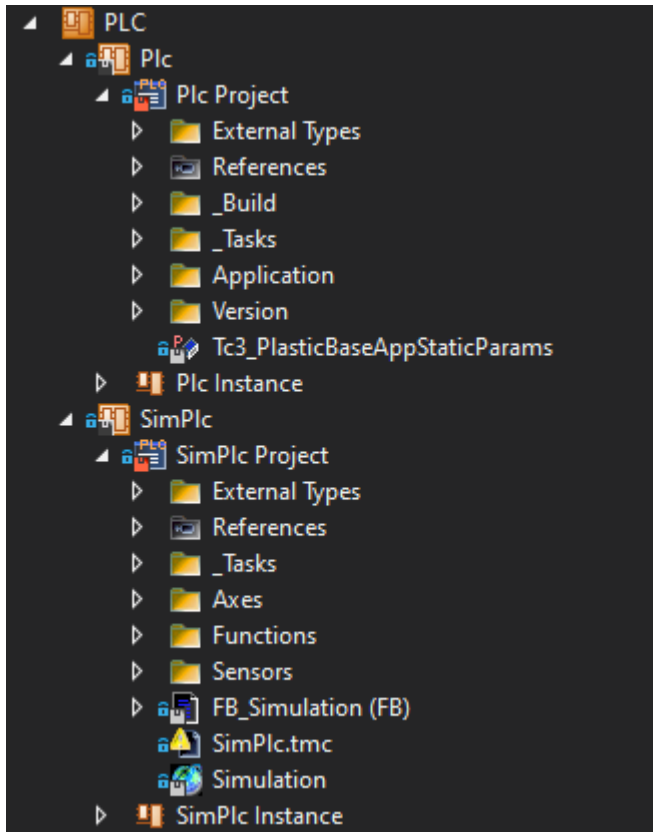
### 6.1 Allgemein

#### 6.1.1 Neues TwinCAT Projekt aufsetzen

Wenn Sie ein neues TwinCAT Projekt mit der Plastic Application beginnen möchten, haben Sie mehrere Optionen das Projekt aufzusetzen. In diesem Beispiel werden die Einrichtungsschritte für ein neues Projekt gezeigt, welches auf das mitgelieferte ApplicationSample aufsetzt.

1. Legen Sie die übermittelte Projektdatei *Tc3\_PlasticApplication\_V12.5.0.zip* in einen Ordner mit möglichst kurzem Projektpfad ab.  
In diesem Beispiel wird der Pfad *C:\Projects\* gewählt.  
**Vermeiden Sie lange Dateipfade - ein zu langer Dateipfad kann zu Fehlern beim Exportieren der ZIP-Datei oder beim Projekt öffnen führen. Vermeiden Sie daher die Ablage des Projektes unter langen Dateipfaden wie beispielsweise den Benutzerordnern (C:\Users\{UserName}\Documents\TcXaeShell).**
  2. Entpacken Sie die ZIP-Datei.
  3. Öffnen Sie die *Tc3\_PlasticApplicationPlc\*.tsproj* Datei in einem der Unterverzeichnisse mittels der TcXaeShell.  
Blasformmaschine:  
*Tc3\_PlasticApplicationPlc\BlowMolding\Tc3\_PlasticBaseApplication\Tc3\_PlasticApplicationPlcBMM.tsproj*  
Extruder:  
*Tc3\_PlasticApplicationPlc\Extruder\Tc3\_PlasticBaseApplication\Tc3\_PlasticApplicationPlcEXTR.tsproj*  
**Die beim Öffnen des Projektes erscheinende Warnmeldung, dass eine tmc-Datei nicht gefunden wurde, kann ignoriert werden. Die fehlende Datei wird beim Erstellen des Projektes automatisch erzeugt.**
- ⇒ Das Projekt ist startbereit.  
Sie sehen zwei PLC-Projekte:  
Die Applikation und eine Simulations-PLC. Die Simulation fungiert in seiner Form als alleinstehendes

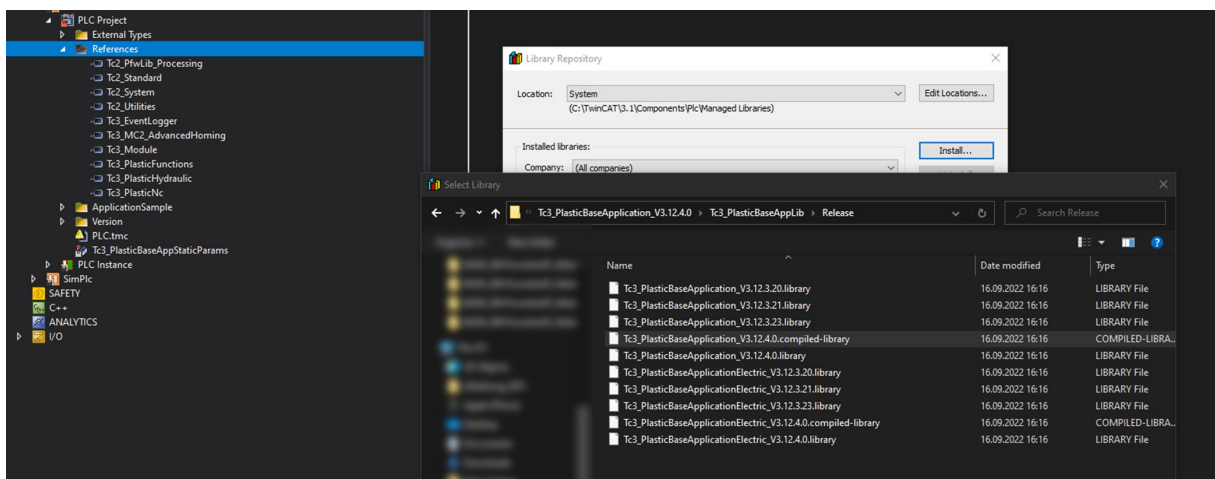
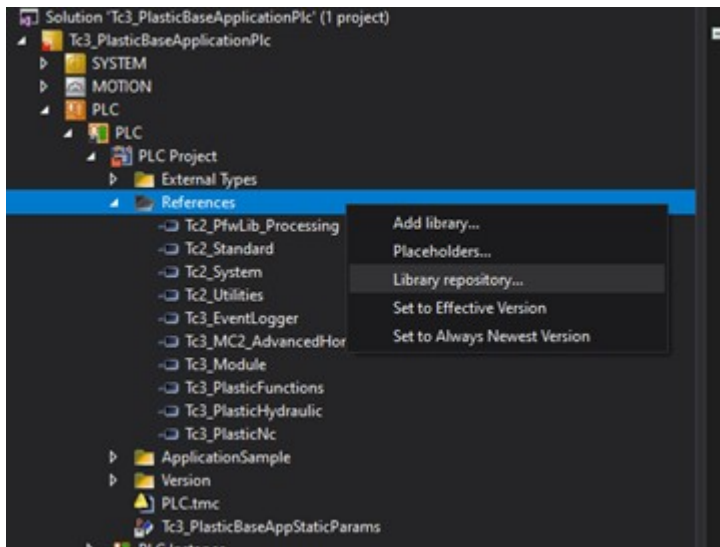
Projekt wie das E/A einer Maschine. Entsprechend kann die Simulation wie eine Maschine auf die Steuerungs-PLC gemappt werden.



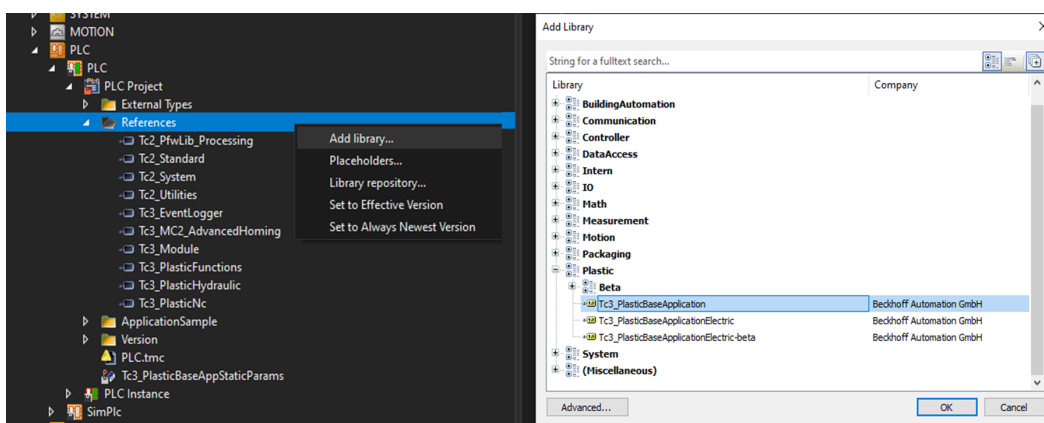
## 6.1.2 Leeres Projekt aufsetzen / bestehendes Projekt erweitern

Neben der Möglichkeit ein Projekt auf das mitgelieferte Projekt aufzusetzen kann die Plastic Base Application in ein bestehendes oder leeres Projekt integriert werden. Dies wird ermöglicht, indem die Plastic Base Application als Bibliothek einem bestehenden oder neuen Projekt hinzugefügt wird.

1. Installieren Sie die Bibliothek im TwinCAT Library-Manager. Die Datei befindet sich in dem Unterverzeichnis `\Dependencies\Tc3_PlasticBaseApplication_V3.12.5.0.compiled-library`



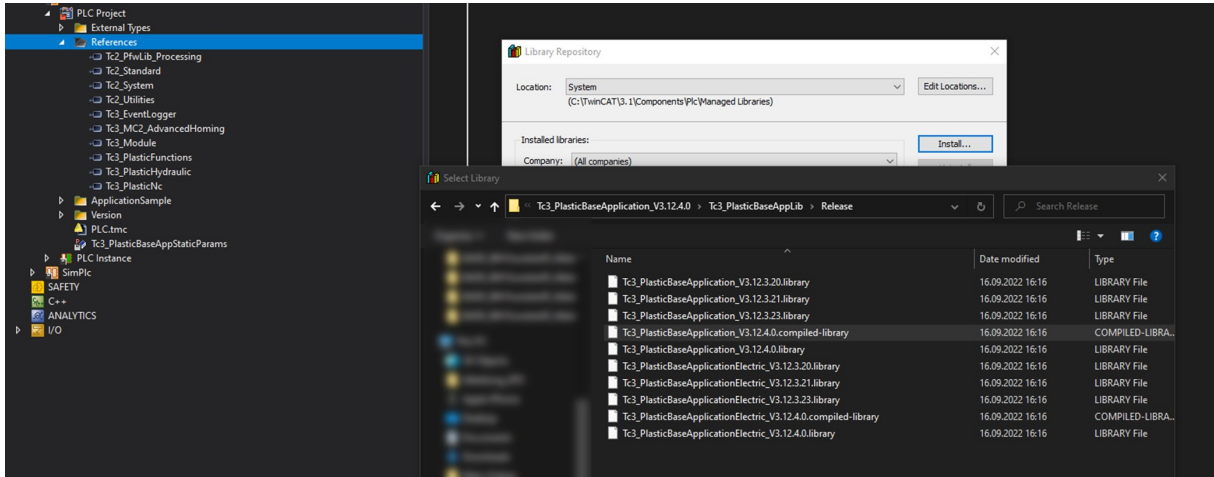
2. Fügen Sie dem Projekt die Bibliothek hinzu



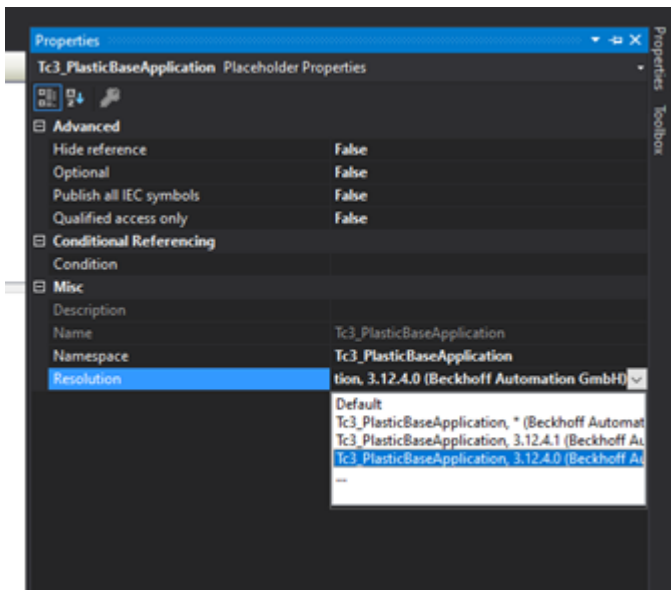
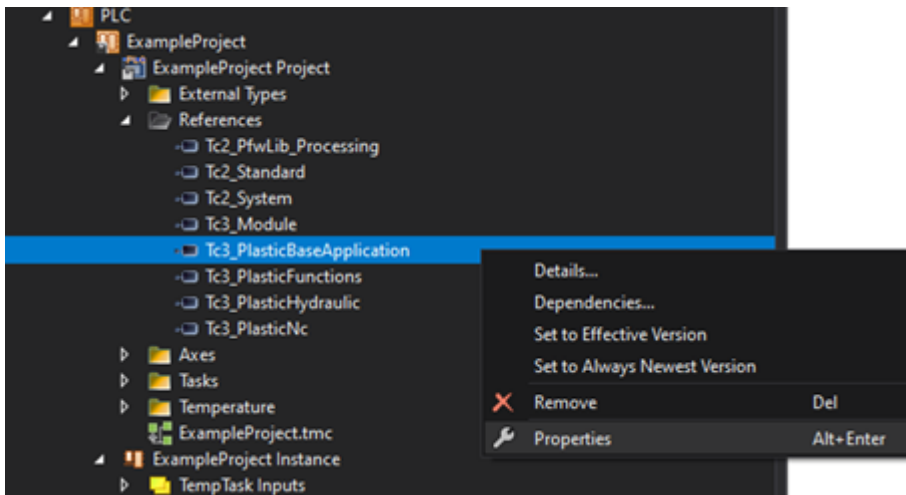
### 6.1.3 Plastic Base Application nachträglich im Projekt aktualisieren

Wenn Sie neue Features folgender Plastic Base Application Versionen in ihr bestehendes Projekt integrieren möchten, bietet die TwinCAT 3 Plastic Base Application die Möglichkeit eines Updates.

1. Installieren Sie die neuere Version der Plastic Base Application Version in ihrem Bibliotheks-Manager.



2. Ändern Sie die festgesetzte Version der Bibliothek auf die neu installierte Version.



3. Überprüfen Sie mit dem Erstellen des Projektes, ob die neue Bibliothek einwandfrei übernommen wurde.

## 6.2 Objektorientierung

### 6.2.1 Eine Variable einer Klasse (FB) hinzufügen

Soll der Umfang der internen Variablen einer Klasse (FB) erweitert werden, müssen folgende Schritte vorgenommen werden. Im folgenden Beispiel wird dem Standard-Achstyp `FB_Axis` eine Variable hinzugefügt.

1. Legen Sie eine neue Klasse (FB) an und entfernen Sie `VAR_INPUT` und `VAR_OUTPUT`.

```
FUNCTION_BLOCK FB_CustomAxis
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
```

2. Fügen Sie der Klassendefinition über das Schlüsselwort `EXTENDS` die zu erbende Klasse hinzu.

```
FUNCTION_BLOCK FB_CustomAxis EXTENDS FB_Axis
VAR
END_VAR
```

3. Legen Sie die neuen Variablen im Bereich `VAR` an.

```
FUNCTION_BLOCK FB_CustomAxis EXTENDS FB_Axis
VAR
    bNewVariable1:      BOOL;
    nNewVariable2:      INT;
    fNewVariable3:      LREAL;
END_VAR
```

4. Kompilieren Sie das Projekt, um die Implementierung auf korrekte Syntax zu überprüfen.

⇒ Sie haben erfolgreich einer Klasse neue Variablen hinzugefügt.

Beispielergebnis in der eingeloggten PLC:

fbCustomAxis	FB_CustomAxis	
eBaseState	E_BASESTATE	eInit
hrLastInternalError	HRESULT	16#00000000
sObjectName	STRING	'fbCustomAxis'
fbMessage	FB_AppMessage	
bError	BOOL	FALSE
sNamespace	STRING	'PRG_AxisApplica...
fbMachineData	FB_MdBaseContainer	
MachineData	I_MdBaseContainer	16#FFFFDE87FC...
iAxisBase	I_AxisBase	16#0000000000...
iAxisHmi	I_AxisHmi	16#0000000000...
eErrorAlarmTL	TCEVENTSEVERITY	Verbose
fbAxisHoming	FB_Homing	
fbAxisSpecific	FB_AxisSpecific	
fbAxisData	FB_MdAxis	
bNewVariable1	BOOL	FALSE
nNewVariable2	INT	0
fNewVariable3	LREAL	0

### 6.2.2 Eine Eigenschaft oder Methode einer Klasse (FB) hinzufügen

In vielen Fällen soll der Umfang an Methoden und Eigenschaften einer Klasse geändert werden. Hierzu zählt neben dem Hinzufügen von neuen Elementen auch das Ändern oder Entfernen bestehender Elemente. In den folgenden Beispielen werden diese drei Vorhaben am Beispiel des Standard-Achstyps `FB_Axis` erläutert.

Für alle drei Vorhaben sind folgende Schritte vorab durchzuführen:

1. Legen Sie eine neue Klasse (FB) an und entfernen Sie `VAR_INPUT` und `VAR_OUTPUT`.

```
FUNCTION_BLOCK FB_CustomAxis
VAR_INPUT
END_VAR
```



```
VAR_OUTPUT
END_VAR
VAR
END_VAR
```

2. Fügen Sie der Klassendefinition über das Schlüsselwort `EXTENDS` die zu erbende Klasse hinzu.

```
FUNCTION_BLOCK FB_CustomAxis EXTENDS FB_Axis
VAR
END_VAR
```

3. Damit Sie die Klasse und die hinzugefügten Elemente auch über eine Schnittstelle ansprechen können, legen Sie eine gleichnamige Schnittstelle an.

```
INTERFACE I_CustomAxis
```

4. Lassen Sie die Schnittstelle von der Schnittstelle der geerbten Klasse erben.

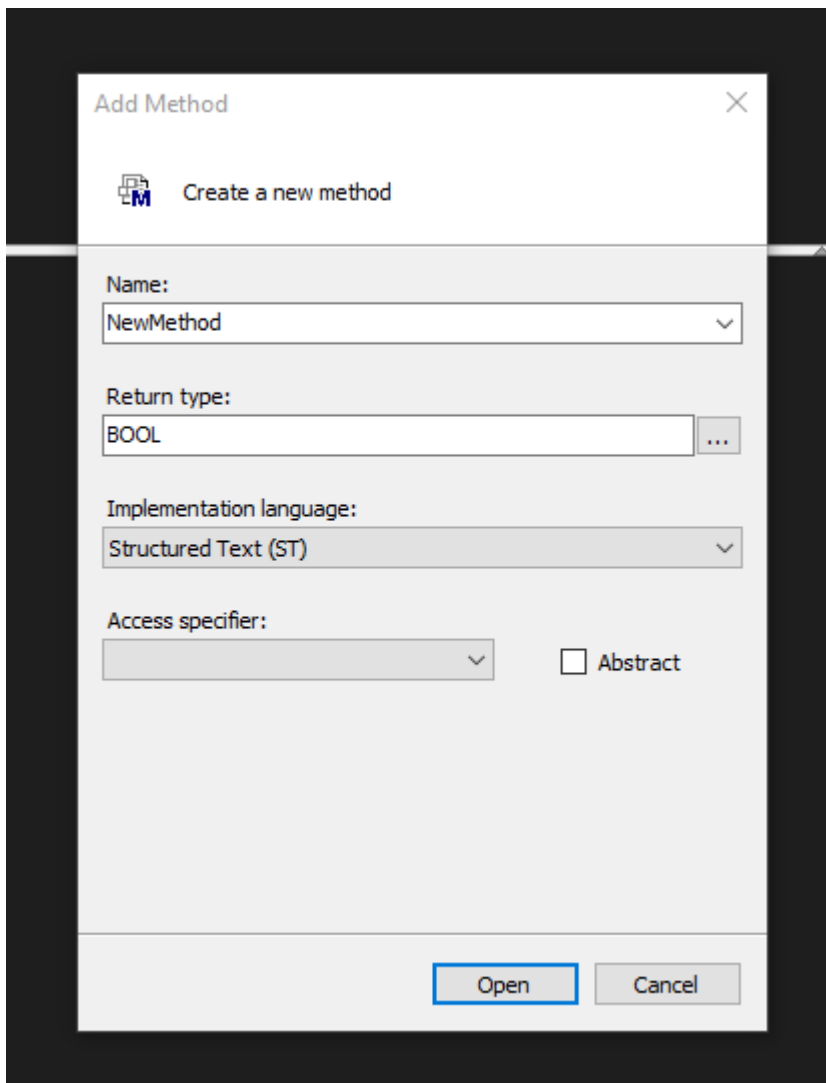
```
INTERFACE I_CustomAxis EXTENDS I_Axis
```

5. Implementieren Sie die Schnittstelle in der zuvor erstellten Klasse (FB).

```
FUNCTION_BLOCK FB_CustomAxis EXTENDS FB_Axis IMPLEMENTS I_CustomAxis
VAR
END_VAR
```

**Eine neue Methode/Eigenschaft hinzufügen**

1. Fügen Sie der Klasse eine neue Methode/Eigenschaft hinzu.



```
METHOD NewMethod : HRESULT
VAR
END_VAR
```

- Kopieren Sie die Methode in die erstellte Schnittstelle, sofern die Methode von außen zugänglich sein soll.  
⇒ Sie haben erfolgreich eine neue Methode hinzugefügt und können mit der Implementation beginnen.

### Eine Methode/Eigenschaft von der geerbten Klasse erweitern oder überschreiben

- Fügen Sie der Klasse eine Methode/Eigenschaft mit dem Namen der zu erweiternden Methode/Eigenschaft hinzu.

```
METHOD Power : HRESULT
VAR_INPUT
    bCommand:      BOOL;
END_VAR
VAR
    nValue:        INT;
END_VAR
```

- Falls Sie die Methode/Eigenschaft nicht überschreiben, sondern erweitern möchten, müssen Sie die Basis-Implementation an geeigneter Stelle aufrufen.

```
METHOD Power : HRESULT
VAR_INPUT
    bCommand:      BOOL;
END_VAR
VAR
    nValue:        INT;
END_VAR

IF bCommand THEN
    nValue := 10;
END_IF

SUPER^.Power(bCommand);
```

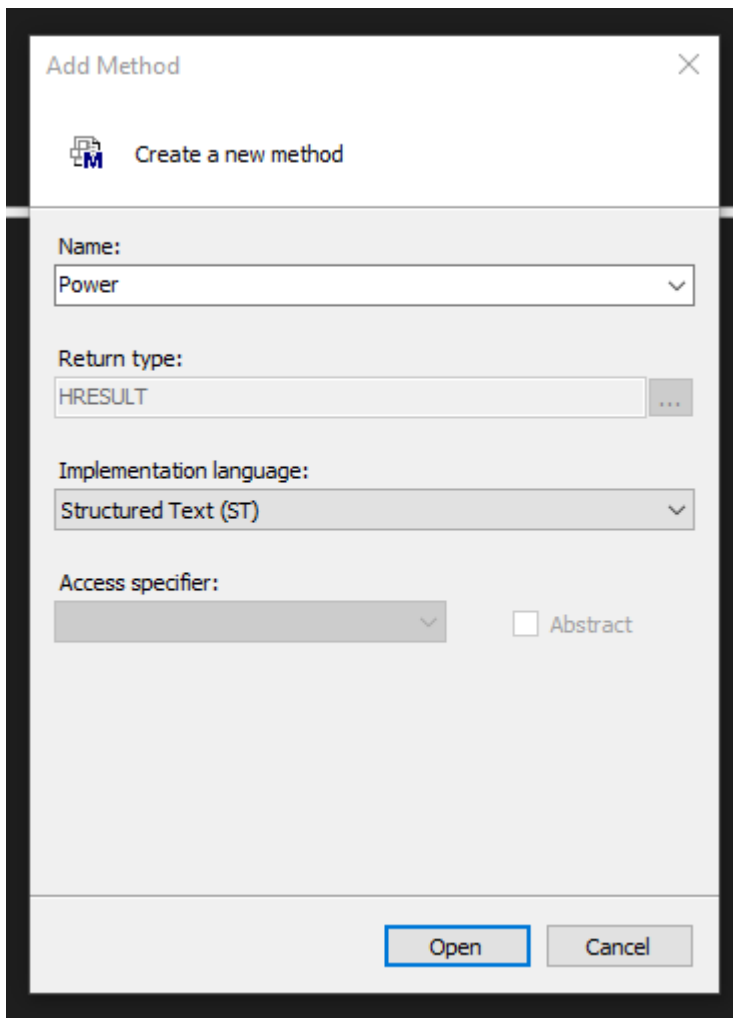
3. Prüfen Sie den Rückgabewert und die `INPUT` Variablen auf Übereinstimmung. Die Basis-Implementation können Sie durch das Auswählen des `SUPER^.MethodName()` und anschließendes Drücken der F12-Taste einsehen.
4. Implementieren Sie ihre eigenen Code-Zeilen in der Methode.
  - ⇒ Sie haben erfolgreich eine Methode/Eigenschaft erweitert.

**Eine Methode/Eigenschaft von der geerbten Klasse entfernen**

**Ein Entfernen einer Methode/Eigenschaft ist nur indirekt möglich**

**I** Beachten Sie, dass Sie eine Methode/Eigenschaft nicht komplett entfernen können! Die im Folgenden beschriebenen Schritte führen lediglich dazu, dass der Aufruf der „entfernten“ Methode/Eigenschaft keine Reaktion hervorruft.

1. Fügen Sie der Klasse eine Methode/Eigenschaft mit dem Namen der zu entfernenden Methode/Eigenschaft hinzu.



2. Lassen Sie den Inhalt der Methode/Eigenschaft leer und rufen Sie **nicht** die `SUPER^.Method()` auf.

```
METHOD Power : HRESULT
VAR_INPUT
    bCommand:      BOOL;
END_VAR
VAR
END_VAR
```

3. Optional: Fügen Sie der Methode/Eigenschaft das `{attribute 'hide'}` hinzu, um die Methode/Eigenschaft in der Entwicklungsumgebung zu verbergen.
  - ⇒ Sie haben erfolgreich eine Methode/Eigenschaft deaktiviert.

## 6.2.3 Innere Prozeduren einer Klasse (FB) anpassen

Manche Klassen enthalten innere Abläufe/Prozeduren, welche durch Vererbungsstufen sich erweitern/ändern oder durch die Applikation angepasst werden sollen. Dies wird mit der `FB_AdaptableSequence` [► 103] Klasse realisiert.

1. Legen Sie eine neue Klasse (FB) an und lassen Sie diese von einer Klasse mit innerer Prozedur erben.

```
FUNCTION_BLOCK FB_AdaptableClass EXTENDS FB_Extruder
VAR
END_VAR
```

2. Überschreiben Sie die interne Callback Methode mit der integrierten Prozedur.  
Im Falle der `FB_Extruder` Klasse ist die interne Prozedur in der `PowerStates()` Methode implementiert.

```
METHOD PROTECTED PowerStates
VAR_INPUT
END_VAR
```

3. Fügen Sie der Callback-Methode eine Abfrage für ihren Prozessschritt hinzu, ob Sie einen bestehenden Prozedurschritt erweitern oder einen neuen hinzufügen und ob Sie bei einer bestehenden Implementation diese ausführen oder überspringen möchten.
4. Evaluieren Sie, ob der Aufruf der bestehenden Implementation vor oder nach ihrer neuen Implementation sinnvoll ist. Der Aufruf der `SUPER^` Methode ist nur erforderlich, wenn Sie die bestehenden Prozedurschritte weiterhin nutzen.

```
// React on existing sequence state
IF aSeqBaseMembers[E_ExtruderPowerStates.eStartVeloFeed].IsActive THEN
;
// Call return to replace existing implementation
RETURN;
END_IF

// Define additional/replacing sequence state
IF fbSetProdTurnrate.IsActive THEN
;
RETURN;
END_IF
Platzieren Sie den SUPER^ Aufruf der Callback Methode.
```

5. Definieren Sie die Bedingung, zu der der Prozedurschritt fertiggestellt ist.

```
// React on existing sequence state
IF aSeqBaseMembers[E_ExtruderPowerStates.eStartVeloFeed].IsActive THEN
// Set FB_AdaptableSequence interface locally
iSeq := aSeqBaseMembers[E_ExtruderPowerStates.eStartVeloFeed];

IF bAdditionalAction THEN
nSaveValueToThis := 10;
// command a jump to a state that is not the default "next" element
iSeq.Jump(fbSetProdTurnrate);
// feedback on finishing the sequence state
iSeq.Done := TRUE;
END_IF

// Call return to replace existing implementation
RETURN;
END_IF

// Define additional/replacing sequence state
IF fbSetProdTurnrate.IsActive THEN
// Set FB_AdaptableSequence interface locally
iSeq := fbSetProdTurnrate;

IF bAdditionalAction THEN
nSaveValueToThis := 10;
iSeq.Done := TRUE;
END_IF

RETURN;
END_IF

// Call implementation of other sequence steps
SUPER^.PowerStates();
```

6. [Nur beim Hinzufügen]: Instanzieren Sie in der Klasse eine Instanz vom Typ `FB_AdaptableSequence` mit der Bezeichnung des Prozedurschrittes.

```
FUNCTION_BLOCK FB_AdaptableClass EXTENDS FB_Extruder
VAR
    fbSetProdTurnrate:      FB_AdaptableSequence;
END_VAR
```

7. Fügen Sie den Prozedurschritt in der Initialisierung an der gewünschten Stelle ein.

```
IF NOT F_SucceededHr(SUPER^.Init(), Init) THEN
    RETURN;
END_IF

fbPowerStates.Insert [▶_107] (
    iCurrent      := aSeqBaseMembers[E_ExtruderPowerStates.eMasterMode],
    iNew          := fbSetProdTurnrate,
    bOverwrite    := FALSE);
```

8. Spielen Sie die Änderungen auf ihr Zielsystem und starten Sie die PLC neu.

⇒ Sie haben erfolgreich eine innere Prozedur einer Klasse erweitert.

## 6.2.4 Erweitern der HMI-Parallelklasse (FB)

In vielen Anwendungsfällen soll die Anzahl an dargestellten Werten auf der Oberfläche erweitert werden. Hierfür kann die HMI-Parallelklasse erweitert werden, sodass die neuen Werte in der HMI-Umgebung an geeigneter Stelle erreichbar sind.

✓ Für die vollständige Implementation ist die Erweiterung der Basisklasse (FB) notwendig.

1. Führen Sie Schritt 1 bis 5 der Erweiterung der Basisklasse (FB) für die Basisklasse durch.
2. Legen Sie eine neue Klasse (FB) an und entfernen Sie VAR\_INPUT und VAR\_OUTPUT.

```
FUNCTION_BLOCK FB_CustomAxisHmi
VAR
END_VAR
```

3. Fügen Sie der Klassendefinition über das Schlüsselwort EXTENDS die zu erbende Klasse hinzu.

```
FUNCTION_BLOCK FB_CustomAxisHmi EXTENDS FB_AxisHmi
VAR
END_VAR
```

4. Damit Sie die Klasse und die hinzugefügten Elemente auch über eine Schnittstelle ansprechen können, legen Sie eine gleichnamige Schnittstelle an.

```
INTERFACE I_CustomAxisHmi
```

5. Lassen Sie die Schnittstelle von der Schnittstelle der geerbten Klasse erben.

```
INTERFACE I_CustomAxisHmi EXTENDS I_AxisHmi
```

6. Implementieren Sie die Schnittstelle in der zuvor erstellten Klasse (FB).

```
FUNCTION_BLOCK FB_CustomAxisHmi EXTENDS FB_AxisHmi IMPLEMENTS I_CustomAxisHmi
VAR
END_VAR
```

7. Instanzieren Sie die Schnittstelle in der Basisklasse (FB).

```
FUNCTION_BLOCK FB_CustomAxis EXTENDS FB_Axis IMPLEMENTS I_CustomAxis
VAR
    iCustomAxisHmi:      I_CustomAxisHmi;
END_VAR
```

8. Überschreiben Sie die SetHMI() Methode der Basisklasse.

```
// Setter method for HMI-Class
METHOD SetHMI : HRESULT
VAR_INPUT
    ipBaseHmi:      I_BaseHmi;      // interface on hmi object
END_VAR

IF NOT __QUERYINTERFACE(ipBaseHmi, iCustomAxisHmi) THEN
    SetHMI := F_HresultFailure(E_AdsErr.DEVICE_INVALIDINTERFACE);
    RETURN;
END_IF

SetHmi := S_OK;
```

9. Erweitern Sie die Init() Methode und fügen Sie eine \_\_QUERYINTERFACE() Operation für das neue Interface hinzu.

```
METHOD Init : HRESULT

IF NOT __QUERYINTERFACE(iCustomAxisHmi, iAxisHmi) THEN
    RETURN;
ELSIF NOT F_SucceededHr(SUPER^.Init(), Init) THEN
    RETURN;
END_IF
```

## Einen neuen Einstell- oder Kommandowert für das HMI hinzufügen

10. Fügen Sie eine neue Eigenschaft der HMI-Klasse hinzu.

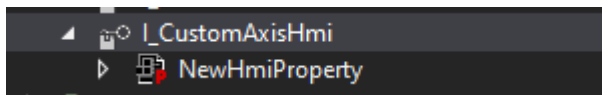
```
PROPERTY NewHmiProperty : LREAL
```

11. Fügen Sie der Deklaration das Attribut Monitoring hinzu, um die Eigenschaft für das HMI sichtbar zu machen.

Dadurch wird die Eigenschaft auch im eingeloggten Zustand der PLC sichtbar.

```
{attribute 'monitoring' := 'call'}
PROPERTY NewHmiProperty : LREAL
```

12. Kopieren Sie die Eigenschaft in die gleichnamige Schnittstelle.



13. Legen Sie in der HMI-Klasse eine ähnlich benannte Variable an, in der der Wert zwischengespeichert werden kann.

```
FUNCTION_BLOCK FB_CustomAxisHmi EXTENDS FB_AxisHmi IMPLEMENTS I_CustomAxisHmi
VAR
    fNewHmiProperty:          LREAL;
END_VAR
```

14. Fügen Sie der Deklaration das Attribut 'TcHmiSymbol.Hide' hinzu, sodass die Variable vor dem HMI verdeckt wird.

Dadurch wird gewährleistet, dass die Variable nicht fälschlicherweise anstatt des Properties vom HMI benutzt wird. Wenn Sie im HMI das ADS-Mapping auf Use whitelisting stellen, können Sie grundsätzlich alle Variablen verstecken. Um die HMI-Klassen weiterhin anzuzeigen, müssen Sie bei der Deklaration das {attribute 'TcHmiSymbol.ShowRecursively'} hinzufügen.

```
FUNCTION_BLOCK FB_CustomAxisHmi EXTENDS FB_AxisHmi IMPLEMENTS I_CustomAxisHmi
VAR
    {attribute 'TcHmiSymbol.Hide'}
    fNewHmiProperty:          LREAL;
END_VAR
```

15. Implementieren Sie den Schreibe- und Lesevorgang der Eigenschaft in der Get und Set Methode.

⇒ Sie haben erfolgreich eine Eigenschaft hinzugefügt.

## Eine vorhandene Eigenschaft der Basis-Klasse für die HMI zugänglich machen

1. Instanzieren sie in der HMI-Parallelklasse eine Schnittstelle vom Typ der Basisklasse.

```
FUNCTION_BLOCK FB_CustomAxisHmi EXTENDS FB_AxisHmi IMPLEMENTS I_CustomAxisHmi
VAR
    {attribute 'TcHmiSymbol.Hide'}
    iCustomAxis:              I_CustomAxis;
    {attribute 'TcHmiSymbol.Hide'}
    fNewHmiProperty:          LREAL;
END_VAR
```

2. Fügen Sie die Init() Methode der HMI Klasse hinzu.

```
// Init method for linking to a control class (FB)
METHOD Init : HRESULT
VAR_INPUT
    ipBase:    I_Base;    // Base interface on linked control class (FB)
END_VAR
```

3. Führen Sie in der Init()-Methode eine `__QUERYINTERFACE ()` Operation von der übergebenen Basisschnittstelle auf die zuletzt instanziierte Schnittstelle durch.

```
Init := F_HresultFailure(E_AdsErr.DEVICE_NOTINIT);

RETURN (NOT __QUERYINTERFACE(ipBase, iCustomAxis));
RETURN (NOT F_SucceededHr(SUPER^.Init(ipBase), Init));

Init := F_HresultSuccess(NOERR);
```

4. [Falls erbdend von `FB_BaseHmi`] Implementieren Sie den Aufruf der HMI-Init() in der Init()-Methode der Basis-Klasse.

```
Init:=F_HresultFailure(E_AdsErr.DEVICE_NOTINIT);

IF iCustomAxisHmi = 0 THEN
    RETURN;
ELSIF NOT F_SucceededHr(iCustomAxisHmi.Init(THIS^), Init) THEN
    RETURN;
ELSIF NOT F_SucceededHr(SUPER^.Init(), Init) THEN
    RETURN;
END_IF
```

5. Fügen Sie eine neue Eigenschaft der HMI-Klasse hinzu.

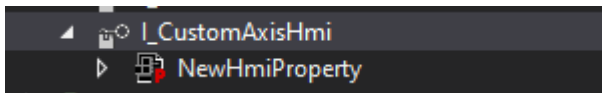
```
PROPERTY SecondNewHmiProperty : LREAL
```

6. Fügen Sie der Deklaration das Attribut `Monitoring` hinzu, um die Eigenschaft für das HMI sichtbar zu machen.

Dadurch wird die Eigenschaft auch im eingeloggten Zustand der PLC sichtbar.

```
{attribute 'monitoring' := 'call'}
PROPERTY NewHmiProperty : LREAL
```

7. Kopieren Sie die Eigenschaft in die gleichnamige Schnittstelle.



8. Programmieren Sie in die `Get` (und ggf. `Set`) Methode der Eigenschaft den Zugriff auf die Eigenschaft der Schnittstelle.

```
IF iCustomAxis <> 0 THEN
    SecondNewHmiProperty := iCustomAxis.ExistingValue;
END_IF
```

9. Für weitere Eigenschaften mit Referenzzugriff entfällt entsprechend Schritt 1 bis 4.

⇒ Sie haben erfolgreich eine Eigenschaft mit Referenzzugriff auf die Hauptklasse hinzugefügt.

## 6.3 Achsen

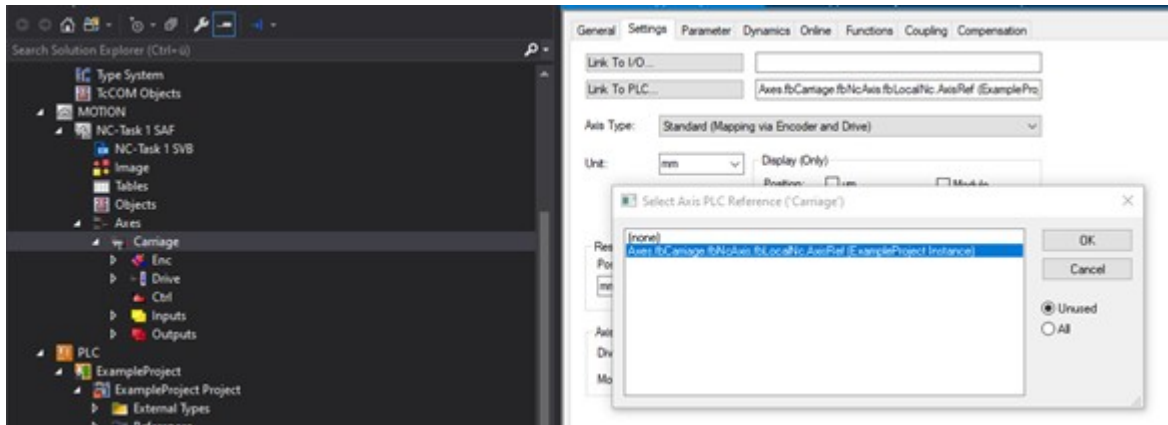
### 6.3.1 NC-Achse anlegen und initialisieren

Zum Anlegen einer NC-Achse gibt es verschiedene Möglichkeiten. Im folgenden Beispiel wird ein exemplarisches Vorgehen dargestellt.

- Legen Sie die drei Instanzen folgender Objekte an:  
`fbCarriageAxis (FB_AxisNcBase)`: TF8560 Achse (hier NC)  
`fbCarriage (FB_Carriage)`: Spezifisch implementierter Achstyp (hier Wagenachse)  
`fbCarriageHmi (FB_CarriageHmi)`: Parallele HMI-Schnittstelle des Achstyps (hier Wagenachse)

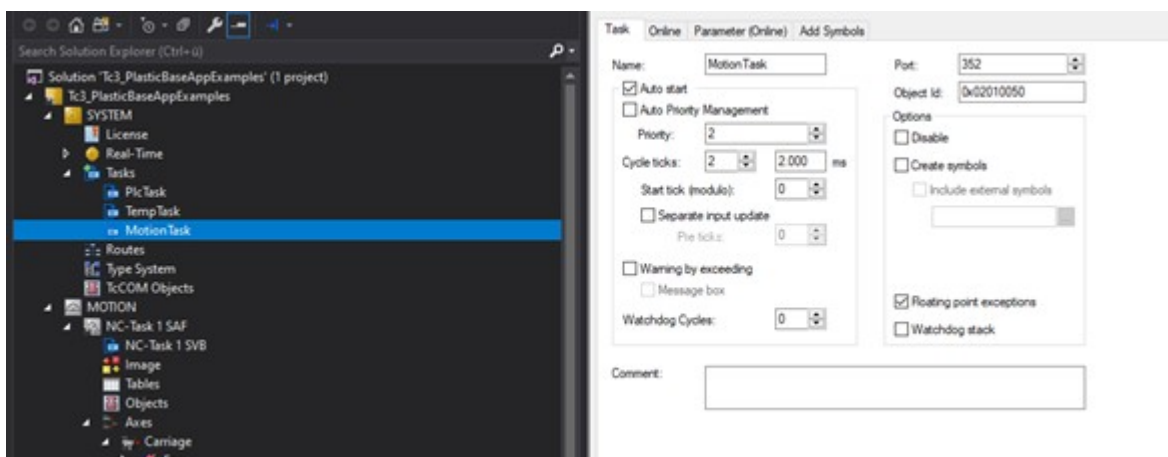
```
{attribute 'qualified_only'}
VAR_GLOBAL
    {attribute 'TcHmiSymbol.Hide'}
    {attribute 'TcContextName':='MotionTask'}
    fbCarriageAxis:      FB_AxisNcBase('', Tc3_PlasticBaseAppStaticParams.cnMaxPtpPoints, 0, 0, 0);
    {attribute 'TcHmiSymbol.Hide'}
    fbCarriage:          FB_Carriage;
    fbCarriageHmi:      FB_CarriageHmi;
END_VAR
```

- Erstellen/Verknüpfen Sie eine NC-Achse im Projekt auf die angelegte Instanz der fbCarriage Achse.



Wenn die TF8560 Achsen aus der PLC nicht in dem Auswahldialog erscheinen, wurde das Projekt nicht erstellt. Erst wenn das PLC-Projekt nach Anlegen der Instanz einmal erfolgreich erstellt wurde, werden die Instanzen im Mapping sichtbar.

- Es ist zu empfehlen, dass Sie für den Antriebssteuerungs-Anteil eine PLC-Task mit der gleichen Zykluszeit der NC (default 2 ms) anlegen (falls nicht schon eine Achse existiert).



- Weisen Sie in der PLC der spezifischen Achse eine TF8560 Achse zu.

```
Axes.fbCarriage.SetAxisRef(Axes.fbCarriageAxis);
```

- Fügen Sie der Runtime die Achse hinzu.

```
Tasks.fbRuntime.Append(Axes.fbCarriage, Axes.fbCarriageHmi);
```

Falls Sie nicht mit der FB\_BaseRuntime arbeiten möchten, sind die folgenden weiteren Schritte notwendig:

- Weisen Sie der spezifischen Achse die HMI-Schnittstelle zu.

```
Axes.fbCarriage.SetHMI(Axes.fbCarriageHmi);
```

- Initialisieren Sie die Achse mit einem einmaligen Aufruf der Init()-Methode und überprüfen Sie die Rückgabewerte.

```
IF NOT bInit THEN
    bInit := TRUE;

    Axes.fbCarriage.SetAxisRef(Axes.fbCarriageAxis);
    Axes.fbCarriage.SetHMI(Axes.fbCarriageHmi);

    hr := Axes.fbCarriage.Init();
    bInitFailed := FAILED(hr);
END_IF
```

- Rufen Sie nach erfolgreicher Initialisierung die CoreCyclic() Methode der Achse mit einer schnellen Task auf.

```
IF NOT PRG_AxisApplication.bInitFailed THEN
    Axes.fbCarriage.CoreCyclic();
END_IF
```



9. Rufen Sie parallel zur `CoreCyclic()` Methode die `Cyclic()` Methode in einer langsameren Task auf und initialisieren Sie die Standardparametrierung mittels der `ParamInit()` Methode.

```

VAR
    bInit:          BOOL;
    bInitFailed:    BOOL;
    bParamInit:     BOOL;
    hr:             HRESULT;
END_VAR

IF NOT bInit THEN
    bInit := TRUE;

    Axes.fbCarriage.SetAxisRef(Axes.fbCarriageAxis);
    Axes.fbCarriage.SetHMI(Axes.fbCarriageHmi);

    hr := Axes.fbCarriage.Init();
    bInitFailed := FAILED(hr);

ELSIF NOT bInitFailed THEN
    IF NOT bParamInit THEN
        hr := Axes.fbCarriage.ParamInit();
        bParamInit := SUCCEEDED(hr);
    END_IF

    Axes.fbCarriage.Cyclic();
END_IF
    
```

### 6.3.2 NC-Transformationsachse anlegen und initialisieren

Zum Anlegen einer neuen NC-Transformationsachse sind mehrere Schritte notwendig. Im folgenden Beispiel wird ein exemplarisches Vorgehen dargestellt.

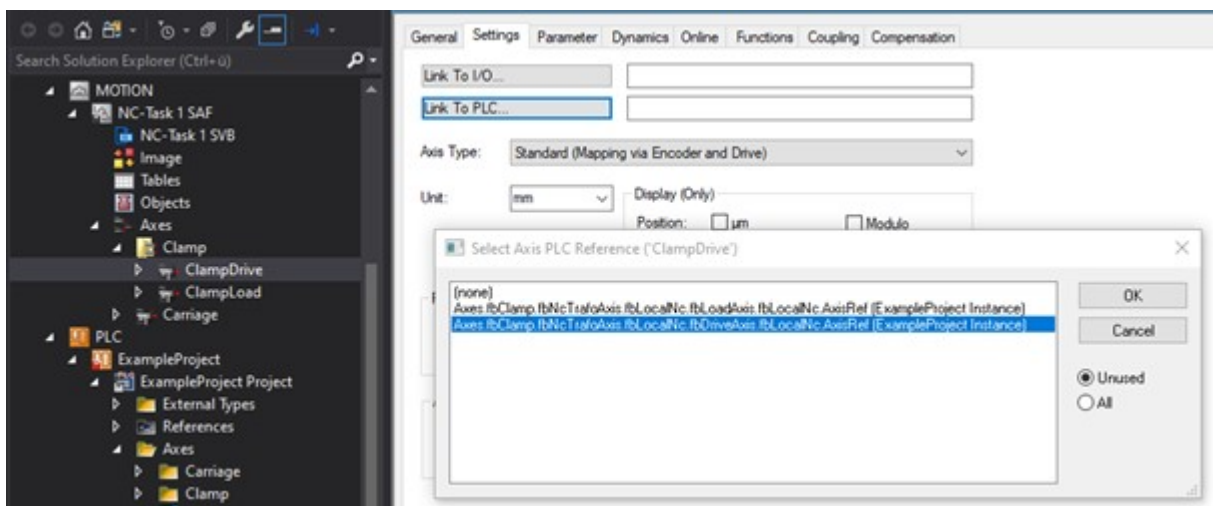
- Legen Sie die drei Instanzen folgender Objekte an:  
 fbClampAxis (FB\_AxisNcTrafoBase): TF8560 Achse (hier Transformation-NC)  
 fbClamp (FB\_Clamp): Spezifisch implementierter Achstyp (hier Schließeinheit)  
 fbClampHmi (FB\_ClampHmi): Parallele HMI-Schnittstelle des Achstyps (hier Schließeinheit).

```

{attribute 'qualified_only'}
VAR_GLOBAL
    {attribute 'TcHmiSymbol.Hide'}
    {attribute 'TcContextName' := 'MotionTask'}
    fbClampAxis:      FB_AxisNcTrafoBase('', Tc3_PlasticBaseAppStaticParams.cnMaxPtpPoints, Tc
3_PlasticBaseAppStaticParams.cnNoOfTrafoPoints, 0, 0, 0);
    {attribute 'TcHmiSymbol.Hide'}
    fbClamp:         FB_Clamp;

    fbClampHmi:     FB_ClampHmi;
END_VAR
    
```

- Erstellen/Verknüpfen Sie zwei NC-Achsen im Projekt für die Antriebs- und Lastseite der angelegten Instanz der fbClamp Achse.



**Wenn die TF8560 Achsen aus der PLC nicht in dem Auswahldialog erscheinen, wurde das Projekt nicht erstellt. Erst wenn das PLC-Projekt nach Anlegen der Instanz einmal erfolgreich erstellt wurde, werden die Instanzen im Mapping sichtbar.**

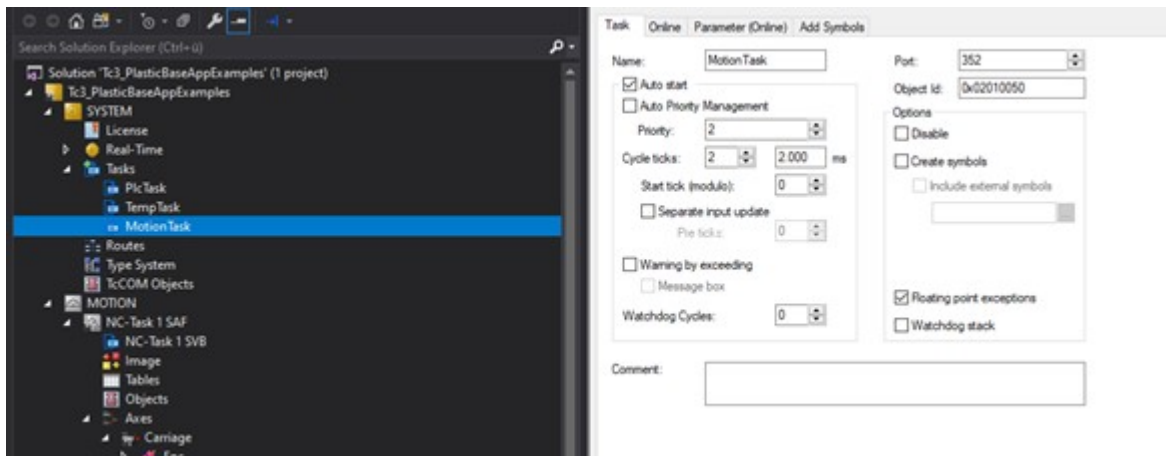
3. Stellen Sie auf der NC-Achse der Lastseite folgende Parametrierung ein:

- Axes > ClampLoad > Settings > Axis Type: = Standard
- Axes > ClampLoad > Enc > NC-Encoder > Type: = Encoder SSI
- Axes > ClampLoad > Enc > Parameter > Scaling Factor Numerator = 0.0001
- Axes > ClampLoad > Enc > Parameter > Scaling Factor Denominator = 1.0
- Axes > ClampLoad > Enc > Parameter > Position Bias = -1000.0
- Axes > ClampLoad > Enc > Parameter > Encoder Mode = POSVELOACC

Und mappen Sie folgende Variablen zusätzlich zwischen NC und PLC:

- Axes > ClampLoad > Enc.Inputs.In.nDataIn1 <-->  
GVL\_Xyz.fbNcTrafoAxis.fbLocalNc.fbActuals.nDataIn1
- Axes > ClampLoad > Enc.Inputs.In.nState4 <-->  
GVL\_Xyz.fbNcTrafoAxis.fbLocalNc.fbActuals.nState4

4. Es ist zu empfehlen, dass Sie für den Antriebssteuerungs-Anteil eine PLC-Task mit der gleichen Zykluszeit der NC (default 2 ms) anlegen (falls nicht schon eine Achse existiert).



5. Instanzieren Sie zusätzlich in ihrer Applikation einen Table Generator zum Erzeugen einer Transformationstabelle.

```
VAR
    fbTableGen:          FB_TableGeneratorClampStandard_1;
END_VAR
```

6. Weisen Sie in der PLC der spezifischen Achse eine TF8560 Achse zu.

```
Axes.fbClamp.SetAxisRef(Axes.fbClampAxis);
```

7. Weisen Sie in der PLC der spezifischen Achse den Table Generator zu.

```
Axes.fbClamp.Specific.Trafo.TableGenerator := fbTableGen;
```

8. Definieren Sie zur Nutzung der Transformationsfunktion die Geometrie der Mechanik. Im Falle der Schließeinheit wird in diesem Beispiel ein Standard-Klemmmechanismus parametrier.

```
Method DefineTable : HRESULT

// Assign geometries
fbTableGen.BaseDistance      := 672.0; // [mm]
fbTableGen.DriveArm          := 228.0; // [mm]
fbTableGen.LoadArm           := 325.2; // [mm]
fbTableGen.ToolArm_1         := 602.52; // [mm]
fbTableGen.ToolArm_2         := 455.4; // [mm]
fbTableGen.ToolArm_3         := 114.0; // [mm]
fbTableGen.ToolArm_Angle     := 216.0; // [mm]
fbTableGen.ToolOffset        := 288.0; // [mm]
fbTableGen.DriveLowEnd       := 0.0; // [°]
fbTableGen.DriveHighEnd     := 180.0; // [°]
```

```
// calculate resulting point table
IF NOT fbTableGen.DefineTable() THEN
    DefineTable:= F_HresultFailure(E_AdsErr.DEVICE_INVALIDPARM);
    RETURN;
END_IF

// copy parameter to machine data file
fbTableGen.WriteToParamList();

// activates table in TF8560 axis
F_SucceededHr(Axes.fbClamp.Specific.Trafo.AssignTableToAxis(FALSE), DefineTable);
// copies table drive ends to nc-Softends
F_SucceededHr(Axes.fbClamp.Specific.Trafo.CopyTableDriveEnds(FALSE, FALSE), DefineTable);
```

### 9. Fügen Sie der Runtime die Achse hinzu.

```
Tasks.fbRuntime.Append(Axes.fbClamp, Axes.fbClampHmi);
```

**Falls Sie nicht mit der FB\_BaseRuntime arbeiten möchten, sind die folgenden weiteren Schritte notwendig:**

### 10. Weisen Sie der spezifischen Achse die HMI Schnittstelle zu.

```
Axes.fbClamp.SetHMI(Axes.fbClampHmi);
```

### 11. Initialisieren Sie die Achse mit einem einmaligen Aufruf der Init()-Methode und überprüfen Sie die Rückgabewert.

```
IF NOT bInit THEN
    bInit := TRUE;

    Axes.fbClamp.SetAxisRef(Axes.fbClampAxis);
    Axes.fbClamp.SetHMI(Axes.fbClampHmi);

    hr := Axes.fbClamp.Init();
    bInitFailed := FAILED(hr);
END_IF
```

### 12. Rufen Sie nach erfolgreicher Initialisierung die CoreCyclic() Methode der Achse mit einer schnellen Task auf.

```
IF NOT PRG_AxisApplication.bInitFailed THEN
    Axes.fbClamp.CoreCyclic();
END_IF
```

### 13. Rufen Sie parallel zur CoreCyclic() Methode die Cyclic() Methode in einer langsameren Task auf und initialisieren Sie die Standardparametrierung mittels der ParamInit() Methode.

```
VAR
    bInit:          BOOL;
    bInitFailed:   BOOL;
    bParamInit:    BOOL;
    hr:            HRESULT;

    fbTableGen:    FB_TableGeneratorClampStandard_1;
END_VAR

IF NOT bInit THEN
    bInit := TRUE;

    Axes.fbClamp.SetAxisRef(Axes.fbClampAxis);
    Axes.fbClamp.Specific.Trafo.TableGenerator := fbTableGen;
    Axes.fbClamp.SetHMI(Axes.fbClampHmi);

    hr := Axes.fbClamp.Init();
    bInitFailed := FAILED(hr);

ELSIF NOT bInitFailed THEN
    IF NOT bParamInit THEN

        hr := DefineTable();
        bParamInit := SUCCEEDED(hr);
        hr := Axes.fbClamp.ParamInit();
        bParamInit := bParamInit AND SUCCEEDED(hr);
    END_IF

    Axes.fbClamp.Cyclic();
END_IF
```

### 6.3.3 Handfunktion in eine Achse integrieren

Eine Handfunktion kann an mehreren Stellen in der Applikation angelegt werden. Im folgenden Beispiel wird erläutert, wie eine Handfunktion zum Verfahren und Einschalten der Achse in diese integriert werden kann.

1. Erweitern Sie eine bestehende Klasse durch Vererbung. In diesem Beispiel wird hierfür die Klasse `FB_Carriage` verwendet.

```
FUNCTION_BLOCK FB_CustomCarriage EXTENDS FB_Carriage
```

2. Instanzieren Sie die Handfunktion `FB_ManualXYZ` in der Klasse der Achse. Für die Handfunktion der Vor-/Zurückbewegung ist dies nicht notwendig, da diese von der Klasse `FB_Carriage` bereits instanziiert wird.

```
FUNCTION_BLOCK FB_CustomCarriage EXTENDS FB_Carriage
VAR
    fbManualPower:          FB_ManualPower;
END_VAR
```

3. Die hier verwendete Klasse `FB_CarriageHmi` stellt für beide Handfunktionen bereits ein HMI-Interface zur Verfügung. Für weitere Handfunktionen müssen Sie die Klasse um jeweilige Instanzen der Klasse `FB_ManualFunctionHmi` erweitern.
4. Anschließend muss die neu instanziierte Handfunktion initialisiert werden. Zur Vereinfachung können Sie die Funktion `F_SucceededHr()` verwenden, um den Rückgabewert der Initialisierungsfunktion auf Erfolg zu überprüfen und auf eine lokale Variable zu kopieren.

```
METHOD Init : HRESULT

IF NOT F_SucceededHr(fbManualPower.Init(THIS^, iCarriageHmi), Init) THEN
    RETURN;
END_IF

Init := SUPER^.Init();
```

5. Zur Verarbeitung der inneren Algorithmen der Handfunktionen müssen Sie die jeweiligen Zyklusmethoden in der `Cyclic()` aufrufen.

```
METHOD Cyclic

fbManualPower.Cyclic();
fbManualForBack.Cyclic();
```

6. In derselben Methode können Sie zusätzlich definieren, wann die Handfunktion für die Benutzung aktiv geschaltet werden sollen und/oder welche Methode durch die Handfunktion aufgerufen wird.

```
fbManualPower.Enable := bManualMode OR bSetupMode OR bAutomaticMode;
fbManualPower.Cyclic();

fbManualForBack.Enable := bManualMode OR bSetupMode;
fbManualForBack.Cyclic();

IF bSetupMode THEN
    IF fbManualForBack.TrigCmdWorkPos.RQ THEN
        JogNegative(TRUE);
    ELSIF fbManualForBack.TrigCmdWorkPos.FQ THEN
        JogNegative(FALSE);
    ELSIF fbManualForBack.TrigCmdBasePos.Q THEN
        JogPositive(fbManualForBack.TrigCmdBasePos.Q);
    END_IF
ELSIF bManualMode THEN
    IF fbManualForBack.TrigCmdWorkPos.Q THEN
        MovePtp(1, fbManualForBack.TrigCmdWorkPos.Q, 8);
    ELSIF fbManualForBack.TrigCmdBasePos.Q THEN
        MovePtp(2, fbManualForBack.TrigCmdBasePos.Q, 8);
    END_IF
END_IF
```

## 6.4 Datenverwaltung

### 6.4.1 Maschinendaten erzeugen

Zum Startzeitpunkt eines Projektes muss ein erster Satz an Maschinendaten erstellt werden, sodass die während der Inbetriebnahme ermittelten Parameter persistent gespeichert werden können.

#### ● Fehlermeldungen im Eventlogger

**I** Alle an die `FB_BaseRuntime` angehängten Runtime-Objekte lösen zwangsläufig eine Fehlermeldung unter folgenden Bedingungen aus:

- Alle `Init()` Methoden sind erfolgreich ausgeführt worden
- Es existiert bisher keine Maschinendaten-Datei
- `FB_BaseRuntime.DisableMdInit` ist nicht gesetzt

✓ Für die im Folgenden beschriebene Schrittfolge wird die Nutzung der `FB_BaseRuntime` vorausgesetzt.

1. Starten Sie die PLC-Runtime.
2. Rufen Sie die Methode `FB_BaseRuntime.MdSaveAll()` **einmalig** auf.
3. Warten Sie auf die Fertigstellung des Speichervorgangs.  
**Je nach Umfang der zu speichernden Daten kann die Bearbeitungszeit variieren. Im PLC-Online Modus kann die Variable `bMdSaveAll` überwacht werden, um den Abschluss des Speichervorgangs zu überprüfen.**

⇒ Sie haben erfolgreich einen ersten Satz an Maschinendaten erzeugt

### 6.4.2 Rezeptfreigabe integrieren

Da in bestimmten Betriebsszenarien (z.B. im Automatikbetrieb) keine Rezepte in die PLC geladen werden sollen, muss die Freigabe der Rezeptverwaltung durch die PLC erteilt werden.

✓ Die Rezeptfreigabe setzt eine Erstinitialisierung nach Maschinenstart voraus. Hierfür muss die Eigenschaft `FB_PlcStateToHmi.PlcInitialized` gesetzt werden.

1. Schreiben Sie die boolesche Bedingung auf folgende Eigenschaft:  
`FB_PlcStateToHmi.ProductChangeEnable`

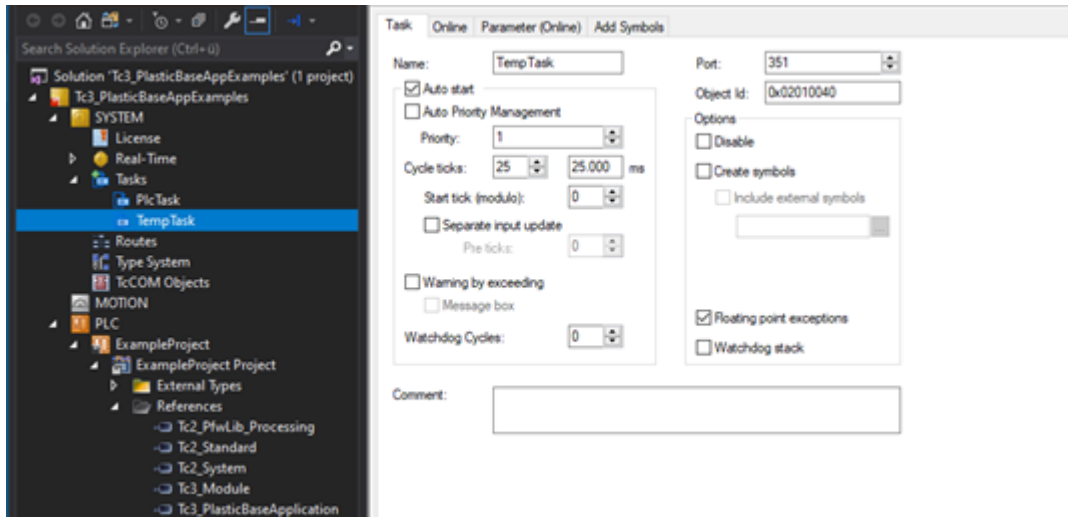
⇒ Wenn die boolesche Bedingung wahr ist, können Rezepte in die PLC geladen werden.

## 6.5 Temperaturregelung

### 6.5.1 Temperaturregelung instanziiieren und initiieren

Für bestimmte Applikationen soll aus dem Umfang der TC3 Plastic Application nur die Temperaturregelung verwendet werden. In diesem, aber auch im normalen Anwendungsfall muss hierfür der Temperaturregler einzeln instanziiert und initiiert werden können. Im folgenden Beispiel wird dieser Prozess Schritt für Schritt durchlaufen.

1. Legen Sie eine neue Task mit einer Zykluszeit von 25 ms an. Dies ist nicht zwingend notwendig, wird aber - basierend auf Erfahrungswerten - empfohlen.



2. Erzeugen Sie eine Instanz des FB\_TempCtrl und des FB\_TempCtrlHmi und ordnen Sie das Mapping der neu angelegten Task zu.

```
{attribute 'qualified_only'}
VAR_GLOBAL
  {attribute 'TcHmiSymbol.Hide'}
  {attribute 'TcContextName':='TempTask'}
  fbTempCtrl:          FB_TempCtrl;

  fbTempCtrlHmi:      FB_TempCtrlHmi;
END_VAR
```

3. Rufen Sie die SetHMI() und Init() Methode des FB\_TempCtrl auf und überprüfen Sie den Rückgabewert auf erfolgreiche Ausführung. Wenn dies der Fall ist, können Sie die ParamInit() Methode zur Parameterinitialisierung und die Zyklusmethode aufrufen.

```
VAR
  bInit:          BOOL;
  bInitFailed:   BOOL;
  bParamInit:    BOOL;
  hr:            HRESULT;
END_VAR

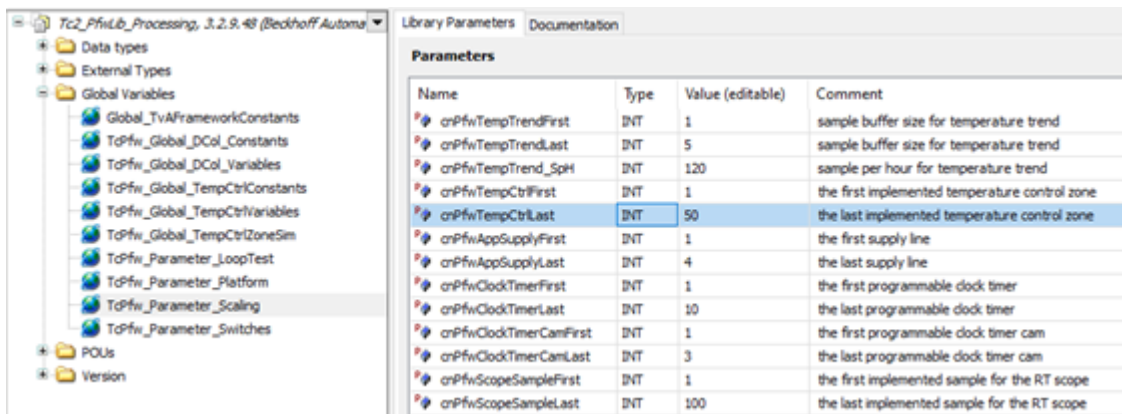
IF NOT bInit AND NOT bInitFailed THEN
  hr := fbTempCtrl.SetHMI(fbTempCtrlHmi);
  hr := fbTempCtrl.Init();
  IF SUCCEEDED(hr) THEN
    bInit := TRUE;
  ELSE
    bInitFailed := TRUE;
  END_IF
ELSIF bInit THEN
  IF NOT bParamInit THEN
    hr := fbTempCtrl.ParamInit();
    IF SUCCEEDED(hr) THEN
      bParamInit := TRUE;
    END_IF
  END_IF

  fbTempCtrl.Cyclic();
END_IF
```

## 6.5.2 Mapping und Konfiguration von Temperaturzonen

Zur Inbetriebnahme der Temperaturregelung müssen einzelne wenige Schritte im Engineering vorgenommen werden. Danach ist ein Einrichten und Konfigurieren rein auf der Benutzeroberfläche möglich. Die folgende Schritt-für-Schritt-Anleitung zeigt Ihnen die Einstellnotwendigkeiten im TwinCAT Engineering.

1. Konfigurieren Sie die nutzbare Anzahl an Zonen in der Tc2\_PfwLib\_Processing.  
References > Tc2\_PfwLib\_Processing > Global Variables > TcPfw\_Parameter\_Scaling > cnPfwTempCtrlLast



Die in der Tc3\_PlasticBaseApplication enthaltene Bibliothek kann nicht angepasst werden! Es ist notwendig, dass Sie die Bibliothek Tc2\_PfwLib\_Processing ihrem Projekt hinzufügen, um die Anzahl an Temperaturzonen anzupassen.

2. Verknüpfen Sie ihre Analogeingänge der Temperatursensoren auf das Array `.in_pfwTempCtrlInput[]`.

```

TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL3318)
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_AdsAddr → VarB: InfoData^AdsAddr
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsData → VarB: TC Channel 1^Value
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsError → VarB: TC Channel 2^Status^Error Size: 1
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsOverrun → VarB: TC Channel 2^Status^Overrange Size: 1
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsState → VarB: TC Channel 1^Status
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsUnderrun → VarB: TC Channel 2^Status^Underrange Size: 1
  VarA: TempTask Inputs^.in_pfwTempCtrlInput[]^EL_SnsWcState → VarB: WcState^WcState Size: 1
    
```

Nach dem Einstellen der Zonenanzahl muss das Projekt erfolgreich erstellt werden, sodass das Mapping aktualisiert wird, ansonsten hat das Mapping nicht die eingestellte Anzahl an Zonen.

3. Verknüpfen Sie ihre Digitalausgänge der Heizungs- und Kühlungsrelais auf das Array `.out_pfwTempCtrlOutput[]`.

```

TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 3 (EL2008)
  VarA: TempTask Outputs^.out_pfwTempCtrlOutput[]^SelOutPos → VarB: Channel 1^Output Size: 1
    
```

4. Gruppieren Sie die lineare Anordnung der Zonen passend zu ihrer Anwendung. Hierfür haben Sie folgende Möglichkeiten:
  - Ein zusammenhängender Abschnitt der Anordnung ist einer Gruppe an Temperaturzonen zuzuordnen. In diesem Beispiel wird Zone 1 bis 10 der Gruppe 1 zugewiesen.

```
hr := fbTempCtrl.LinkGroup(nStartIdx := 1, nEndIdx := 10, nGroupIdx := 1, bOverwrite := FALSE);
```

- Die Teilnehmer einer Temperaturzonengruppe sind über die Anordnung verstreut und müssen einzeln der Gruppe zugeordnet werden. In diesem Beispiel wird Zone 12 und Zone 20 der Gruppe 2 zugewiesen.

```
hr := fbTempCtrl.LinkZone(nLinearIdx := 12, nGroupIdx := 2, nGroupMemberIdx := 1, bOverwrite := FALSE);
hr := fbTempCtrl.LinkZone(nLinearIdx := 20, nGroupIdx := 2, nGroupMemberIdx := 2, bOverwrite := FALSE);
```

### 6.5.3 Temperaturregelung in Betrieb nehmen

Die Inbetriebnahme der Temperaturregelung umfasst sowohl einen TwinCAT Engineering als auch einen zur Laufzeit durchzuführenden Teil. In diesem Beispiel werden die einzelnen Schritte, die zur Laufzeit durchgeführt werden, zusammengefasst.

#### **i** Inbetriebnahme über HMI empfohlen

Dieses Beispiel beschreibt das Vorgehen alleinig über den Weg der PLC. Nutzen Sie den Abschnitt [Temperaturregelung in Betrieb nehmen \[► 202\]](#) für eine Inbetriebnahme mittels des HMI.

Vor den folgenden Schritten ist die Vorbereitung des TwinCAT Projektes nach dem Vorbild aus Abschnitt Mapping und Konfiguration von Temperaturzonen [▶ 142] durchzuführen.

**Stellen Sie für alle ihre Temperaturzonen die passenden Ein- und Ausgangssignalarten und -geräte ein**

**i Hardware-Parametrierung ohne Gruppierung**

Dieser Schritt bezieht sich auf das lineare Mapping der TF8540 Bibliothek. Bereits konfigurierte Gruppierungen werden in diesem Schritt ignoriert.

1. Legen Sie einen Programmabschnitt zur einmaligen Zuweisung der Parameter an.
2. Weisen Sie folgende Parameter entsprechend ihrer Hardware-Konfiguration zu.

```
// set cooling output type to 'no cooling'
fbTempCtrlHmi.ParamTempZone[1].OutputSel_C := E_TcPfw_TctrlOutSelect.eTcPfwTcOut_NoSignal;
// set heating output type to 'pwm'
fbTempCtrlHmi.ParamTempZone[1].OutputSel_H := E_TcPfw_TctrlOutSelect.eTcPfwTcOut_PWM;
// set sensor terminal type to 'EL3314'
fbTempCtrlHmi.ParamTempZone[1].TempSensTerm := E_TcPfw_TerminalType.eTcPfwTermT_EL331x;
// set sensor type to 'ThermoCouple Typ K'
fbTempCtrlHmi.ParamTempZone[1].SensorType := E_TcPfw_TempSensType.eTcPfwTempSensT_TC_K;
// set channel of the sensor on the linked terminal to 'Channel 1'
fbTempCtrlHmi.ParamTempZone[1].TermChannel := 1;
```

3. Stellen Sie das 'InUse' flag um die Verwendung der Zone zu validieren.

```
fbTempCtrlHmi.ParamTempZone[1].InUse := TRUE
```

4. Wiederholen Sie Schritt 2 und 3 für alle verwendeten Zonen
5. Führen Sie das erstellte Code-Segment einmalig aus
6. Loggen Sie sich in die Steuerung ein
7. Sichern Sie die Parameter über fbTempCtrl.MachineData.Save

fbTempCtrl	FB_TempCtrl		
eBaseState	E_BASESTATE	eIdle	
hrlLastInternalError	HRESULT	16#00000000	
sObjectName	STRING	'fbTempCtrl'	
fbMessage	FB_AppMessage		
bError	BOOL	FALSE	
sNamespace	STRING	'PRG_Temperatu...	
fbMachineData	FB_MdBaseContainer		
MachineData	I_MdBaseContainer	16#FFFFDE87FC...	
Tc3_PlasticBaseAppExamples.ExampleProject.PRG_Temperature.fbTempCtrl.fbMachineData	FB_MdBaseContainer		
AllowFolderCreation	BOOL	TRUE	
Busy	BOOL	FALSE	
Details	I_MdBaseContainer...	16#FFFFDE87FC...	
Done	BOOL	FALSE	
Error	BOOL	FALSE	
IgnoreMissmatches	BOOL	FALSE	
Load	BOOL	FALSE	
MismatchDetected	BOOL	FALSE	
Save	BOOL	FALSE	TRUE

**Überprüfen Sie die Reaktion der Hardwareeingänge an der Maschine**

1. Loggen Sie sich in die Steuerung ein
2. Gehen Sie in der Baumstruktur fbTempCtrlHmi.fbGroups[...] .fbZones[...] in die erste verwendete Zone (nIndex <> 0)
3. Erwärmen Sie den Sensor der Zone über eine externe Wärmequelle



- Beobachten Sie über den Wert `ActualTemperature` ob die Temperaturänderung in der zu erwartenden Zone auftritt  
Wiederholen Sie Schritt 2 bis 4 für jede Zone

Expression	type	value
fbTempCtrlHmi	FB_TempCtrlHmi	
fbGroups	ARRAY [1..Tc3_Plas...	
fbGroups[1]	FB_TempGroupHmi	
fbZones	ARRAY [1..Tc3_Plas...	
fbZones[1]	FB_TempZoneHmi	
eBaseState	E_BASESTATE	eReady
hrLastInternalError	HRESULT	16#00000000
sObjectName	STRING	'fbZones[1]'
fbMessage	FB_AppMessage	
bError	BOOL	FALSE
iBase	I_Base	16#FFFFE87FC...
nIndex	INT	1
AbsoluteHigh	LREAL	300
AbsoluteLow	LREAL	100 <b>≠ 0</b>
ActTempGain	LREAL	1
ActTempOffset	LREAL	0
ActualCurrent	LREAL	0
ActualTemperature	LREAL	180

**Überprüfen Sie die Reaktion der Hardwareausgänge an der Maschine**

**i Einschalten einer Zone erzeugt kein Powerlevel**

Stellen Sie sicher das zu dem Zeitpunkt dieses Schrittes die Temperaturregelung von der PLC freigegeben wurde!

- Zur Freigabe aller Zonen führen Sie die Methode `FB_TempCtrl.EnableAll(...)` [▶ 54] aus
- Zur Freigabe genau einer Gruppe führen Sie die Methode `FB_TempCtrl.Groups[...].Enable(...)` [▶ 59] aus

- Schalten Sie eine einzelne Temperaturzone über `fbTempCtrlHmi.fbGroups[...].fbZones[...].Enable` ein
- Überprüfen Sie ob in derselben Zone der Wert der Variable `Heating = TRUE` ist und der Wert der Variable `ActualTemperature` (3) sich ändert
- Schalten Sie die Zone möglichst zeitnah wieder aus, um den Temperaturanstieg minimal zu halten
- Wiederholen Sie die Schritte 1 bis 3 für jede Zone

Expression	type	value
fbTempCtrlHmi	FB_TempCtrlHmi	
fbGroups	ARRAY [1..Tc3_Plas...	
fbGroups[1]	FB_TempGroupHmi	
fbZones	ARRAY [1..Tc3_Plas...	
fbZones[1]	FB_TempZoneHmi	
eBaseState	E_BASESTATE	eReady
hrLastInternalError	HRESULT	16#00000000
sObjectName	STRING	'fbZones[1]'
fbMessage	FB_AppMessage	
ActualCurrent	LREAL	0
ActualTemperature	LREAL	180
ExtruderCompEnable	BOOL	FALSE
FcEnable	BOOL	FALSE
FcOffTime	LREAL	600
FcOnTime	LREAL	10
HeaterSwapIdx	INT	0
Heating	BOOL	TRUE
IdleLoadActive	BOOL	FALSE

**Starten Sie das automatische Tuning der Regelparameter**

- Schalten Sie eine Temperaturgruppe über `fbTempCtrlHmi.fbGroups[...].fbOpModeActive.On` ein
- Aktivieren Sie im direkten Anschluss das Tuning einer Gruppe über `fbTempCtrlHmi.fbGroups[...].DoTune = TRUE` ein

3. Führen Sie Schritt 1 und 2 für alle in Betrieb zu nehmenden Gruppen aus

Expression	type	value
fbTempCtrlHmi	FB_TempCtrlHmi	
fbGroups	ARRAY [1..Tc3_Plas...	
fbGroups[1]	FB_TempGroupHmi	
bDoTune	BOOL	FALSE
bEnable	BOOL	FALSE
nGroupIndex	INT	1
sGroupName	STRING	'MainExtruder'
nCountLinkedZones	INT	7
aZoneData	ARRAY [1..Tc3_Plas...	
fbOpModeActive	FB_TempGroupOpM...	
bOn	BOOL	TRUE
bOff	BOOL	FALSE
bTimed	BOOL	FALSE

**Überwachen Sie das automatische Tuning, bis es erfolgreich abgeschlossen ist**

Sobald der Wert der Variable fbTempCtrlHmi.fbGroups[...].TuningActive zurückgesetzt wird ist das Tuning der Gruppe beendet

Der Wert der Variable fbTempCtrlHmi.fbGroups[...].TuningDone zeigt an ob das Tuning erfolgreich war

Expression	type	value
fbTempCtrlHmi	FB_TempCtrlHmi	
fbGroups	ARRAY [1..Tc3_Plas...	
fbGroups[1]	FB_TempGroupHmi	
OpModeActive	REFERENCE TO FB_...	
OpModeStandby	REFERENCE TO FB_...	
TuningActive	BOOL	FALSE
TuningDone	BOOL	TRUE
Zones	REFERENCE TO ARR...	

Sie haben ihre Temperaturregelung erfolgreich in Betrieb genommen

## 7 HMI Projektaufbau

Das Plastic Application HMI-Projekt ist ein in TwinCAT HMI entwickeltes Projekt mit Basisfunktionalitäten zur Erstellung von Bedienoberflächen für Kunststoffverarbeitungsmaschinen. Der Projektaufbau basiert auf dem Standard eines TwinCAT HMI-Projekts.

### Plastic Application HMI nutzen und weiterentwickeln

Da das HMI-Projekt der Plastic Application keine vollumfängliche Bedienoberfläche für jede Kunststoffverarbeitungsmaschine bietet, muss das Projekt maschinenspezifisch angepasst werden. Dazu kann das Projekt beliebig erweitert und modifiziert werden.



#### Updatefähigkeit der Plastic Application HMI

Das Editieren der Plastic Application HMI kann zu einer Einschränkung der Updatefähigkeit führen. Dadurch kann die Unterstützung durch Beckhoff Automation beeinträchtigt werden.

## 7.1 References

Über das Paketverwaltungssystem NuGet können dem Projekt unter dem Ordner References NuGet-Pakete hinzugefügt werden.

### TF8550 | Plastic HMI Framework

Speziell für Kunststoffverarbeitungsmaschinen können die folgenden NuGet-Pakete aus dem Produkt verwendet werden:

NuGet-Paket	Beschreibung
Beckhoff.TwinCAT.HMI.Plastic.Controls	Ergänzung zum NuGet-Paket Beckhoff.TwinCAT.HMI.Controls zur Erweiterung der Toolbox mit spezifischen Controls für Kunststoffverarbeitungsmaschinen (z. B. ArrowMotionGraph, CamControl, PfwSingleTempControl etc.).
Beckhoff.TwinCAT.HMI.Plastic.Functions	Ergänzende Funktionen für das Projekt.
Beckhoff.TwinCAT.HMI.Plastic.Images	Enthält die im Plastic Application HMI-Projekt verwendeten Icons und Grafiken sowie weitere Ressourcen für Kunststoffverarbeitungsmaschinen.
Beckhoff.TwinCAT.HMI.Plastic.Localizations	Enthält die im Plastic Application HMI-Projekt verwendeten Sprachschlüssel mit Übersetzungen in Englisch und Deutsch speziell für Kunststoffverarbeitungsmaschinen.
Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper	Ergänzung zum NuGet-Paket Beckhoff.TwinCAT.HMI.RecipeManagement zur Erweiterung der Recipe Management-Extension.
Beckhoff.TwinCAT.HMI.Plastic.Temperature	Ergänzung zum NuGet-Paket Beckhoff.TwinCAT.HMI.Plastic.Controls zur Erweiterung der Toolbox mit Controls und Funktionen im Bereich Temperatur.
Beckhoff.TwinCAT.HMI.Plastic.Themes	Ergänzt die Themes um ein Plastic-Theme und enthält spezielle Designanpassungen für das Plastic Application HMI-Projekt.

### TwinCAT HMI mit den TF8550 NuGet-Paketen nutzen

Die NuGet-Pakete des TF8550 Produktes können in jedes beliebige TwinCAT HMI Projekt eingebunden werden. Dazu wird in der PLC die entsprechende Lizenz für TF8550 benötigt. Das Paket liefert die im Kapitel beschriebenen NuGet-Pakete. Die NuGet-Pakete können über das Paketverwaltungssystem NuGet

eingebunden werden. Dazu muss die Paketquelle eingefügt werden, unter der sich die NuGet-Pakete befinden. Danach lassen sich die NuGet-Pakete über die Suche finden und installieren. Die installierten NuGet-Pakete sind nach einem Neustart des Projektes im Projekt verfügbar.

Das **Configurator-Control** ist Teil des `Beckhoff.TwinCAT.HMI.Plastic.Controls` NuGet-Paketes und kann über die Toolbox auf eine View oder einen Content gezogen werden. Das Control dient zur Einstellung von spezifischen Funktionen und dem Aussehen einiger weiterer Controls, die sich im gleichen NuGet-Paket befinden. Das Control muss lediglich mit den gewünschten Parametern existieren, hat jedoch keine weitere Funktionalität und muss somit nicht sichtbar sein.

### Voraussetzungen für TC3 Plastic Application

Version	TE2000   TwinCAT 3 HMI Engineering	TF8550   Plastic HMI Framework
v12.6.0	v12.760.44 <ul style="list-style-type: none"> <li>• Beckhoff.TwinCAT.HMI.Controls (12.760.44)</li> <li>• Beckhoff.TwinCAT.HMI.EventLogger (19.0.102)</li> <li>• Beckhoff.TwinCAT.HMI.Framework (12.760.44)</li> <li>• Beckhoff.TwinCAT.HMI.Functions (12.760.44)</li> <li>• Beckhoff.TwinCAT.HMI.RecipeManagement (19.0.102)</li> <li>• Beckhoff.TwinCAT.HMI.ResponsiveNavigation (12.760.44)</li> <li>• Beckhoff.TwinCAT.HMI.SqliteHistorize (19.0.102)</li> </ul>	v12.8.0 <ul style="list-style-type: none"> <li>• Beckhoff.TwinCAT.HMI.Plastic.Controls (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Functions (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Images (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Localizations (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Temperature (12.8.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Themes (12.8.0)</li> </ul>
v12.5.1	v12.758.8 <ul style="list-style-type: none"> <li>• Beckhoff.TwinCAT.HMI.Controls (12.758.8)</li> <li>• Beckhoff.TwinCAT.HMI.EventLogger (19.0.0)</li> <li>• Beckhoff.TwinCAT.HMI.Framework (12.758.8)</li> <li>• Beckhoff.TwinCAT.HMI.Functions (12.758.8)</li> <li>• Beckhoff.TwinCAT.HMI.RecipeManagement (19.0.0)</li> <li>• Beckhoff.TwinCAT.HMI.ResponsiveNavigation (12.758.8)</li> <li>• Beckhoff.TwinCAT.HMI.SqliteHistorize (19.0.0)</li> </ul>	v12.6.0 <ul style="list-style-type: none"> <li>• Beckhoff.TwinCAT.HMI.Plastic.Controls (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Functions (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Images (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Localizations (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Temperature (12.6.0)</li> <li>• Beckhoff.TwinCAT.HMI.Plastic.Themes (12.6.0)</li> </ul>

## 7.2 Themes

Im Ordner *Themes* befindet sich das Plastic-Theme. Das Plastic-Theme referenziert auf das `Beckhoff.TwinCAT.HMI.Plastic.Themes` NuGet-Paket und beinhaltet Designanpassungen bestehender Controls und Klassen, die im HMI-Projekt der Plastic Application verwendet werden. Die Datei kann nach Belieben angepasst werden, um das Design zu verändern, sofern sie in das Projekt geladen wurde.

### **i** Updatefähigkeit der Plastic Application HMI

Das Editieren der Plastic Application HMI kann zu einer Einschränkung der Updatefähigkeit führen. Dadurch kann die Unterstützung durch Beckhoff Automation beeinträchtigt werden.







Verfügbar ab Version 12.5.1

---

## 7.3 Contents

Der Ordner *Contents* beinhaltet in weiteren Unterordnern alle vorhandenen Seiten bzw. `.content`-Dateien des Projektes. Die Contents sind nach gewissen Strukturen aufgebaut. Dabei handelt es sich lediglich um Empfehlungen.

- ▲  Contents
  - ▶  Navigation
  - ▶  Slider
  - ▶  Home.content

Der Ordner *Contents/Navigation* enthält alle Contents kategorisiert in Ordnern, die in der Navigation des HMI-Projektes der Plastic Application zu finden sind.

- ▲ Contents
  - ▲ Navigation
    - ▲ Axes
      - ▲ Blowpin
        - 📄 Blowpin.content
        - 📄 Blowpin\_Homing.content
      - ▲ Carriage
        - 📄 Carriage.content
        - 📄 Carriage\_Homing.content
      - ▲ Clamp
        - 📄 Clamp.content
        - 📄 Clamp\_Homing.content
      - 📄 Scope.content
    - ▲ Extruder
      - 📄 CoExtruder.content
      - 📄 MainExtruder.content
      - 📄 Trend.content
    - ▲ Parameters
      - 📄 Parameters\_Blowpin.content
      - 📄 Parameters\_Carriage.content
      - 📄 Parameters\_Clamp.content
      - 📄 Parameters\_CoExtruder.content
      - 📄 Parameters\_MainExtruder.content
      - 📄 Parameters\_Monitoring.content
      - 📄 Parameters\_Setpoints.content
      - 📄 Parameters\_Temperature.content
      - 📄 Parameters\_Wtc.content
    - ▲ Process
      - 📄 Blowing.content
      - 📄 Monitoring.content
      - 📄 Setpoints.content
      - 📄 Timer.content
    - ▲ System
      - 📄 Administration.content
      - 📄 Alarms.content
      - 📄 RecipeManagement.content
    - ▲ Temperature
      - 📄 Temperature\_Configuration.content
      - 📄 Temperature\_Grouping.content
      - 📄 Temperature\_Layout.content
      - 📄 Temperature\_Overview.content
      - 📄 Temperature\_TimeScheduling.content
    - ▲ Wtc
      - 📄 Wtc.content
      - 📄 Wtc\_Homing.content
  - ▶ Slider
    - 📄 Home.content

Der Ordner *Contents/Slider* enthält alle Contents kategorisiert in Ordnern, die im Sliderbereich des Plastic Application HMI-Projektes angewendet werden.

- ▲ Contents
  - ▷ Navigation
  - ▲ Slider
    - ▲ Axes
      - ▲ Blowpin
        - 🔒 Blowpin\_Homing\_Settings.content
        - 🔒 Blowpin\_Settings.content
      - ▲ Carriage
        - 🔒 Carriage\_Homing\_Settings.content
        - 🔒 Carriage\_Settings.content
      - ▲ Clamp
        - 🔒 Clamp\_Homing\_Settings.content
        - 🔒 Clamp\_Settings.content
    - ▲ Extruder
      - 🔒 Trend\_Settings.content
    - ▲ Parameters
      - 🔒 Parameters\_Settings.content
    - Process
    - ▲ System
      - 🔒 RecipeManagement\_Settings.content
    - ▲ Temperature
      - 🔒 Temperature\_Configuration\_Settings.content
      - 🔒 Temperature\_Layout\_Settings.content
      - 🔒 Temperature\_Overview\_Settings.content
      - 🔒 Temperature\_TimeScheduling\_Settings.content
    - ▲ Wtc
      - 🔒 Wtc\_Homing\_Settings.content
      - 🔒 Wtc\_Settings.content
  - 🔒 Info.content
  - 🔒 ManualFunctions.content
  - 🔒 Navigation.content
  - 🔒 Home.content

Die Unterordner der beiden Ordner *Contents/Navigation* und *Contents/Slider* sind gleich aufgebaut, um die Zugehörigkeit der Contents aus dem Sliderbereich und den Contents des Hauptbereichs (in der Navigation enthalten) zu erkennen.

Nach Belieben können weitere Dateien vom Typ `.content` angelegt werden bzw. die bestehenden Contents gelöscht oder bearbeitet werden.

## **i** Updatefähigkeit der Plastic Application HMI

Das Editieren der Plastic Application HMI kann zu einer Einschränkung der Updatefähigkeit führen. Dadurch kann die Unterstützung durch Beckhoff Automation beeinträchtigt werden.

### Benennung von Controls

Die Benennung (*Identifier*) der Controls innerhalb eines Contents erfolgt nach einem speziellen Schema, um doppelte Benennungen zu vermeiden, da diese zu Fehlern führen.

Die Benennungen im Plastic Application HMI-Projekt richten sich an das folgende Schema:  
`ContentName_XYZ_ControlType`

#### 1. ContentName

Der Name, also die ID des Contents (der Identifier des Controls vom Typ `TcHmi.Controls.System.TcHmiContent`).

- Beispiel: *Temperature\_Parameter\_Settings*

## 2. XYZ

Beliebige Beschreibung des Controls möglich, dabei kann die hierarchische Anordnung innerhalb des Contents beachtet werden.

- Beispiel: *Group1\_Active\_On*

## 3. ControlType

Der Typ des Controls. Dabei wird eine Abkürzung verwendet.

- Beispiel: Für das Control vom Typ `TcHmi.Controls.Beckhoff.TcHmiButton` würde die Abkürzung *Button* genutzt werden.

**Ergebnis:** „Temperature\_Parameters\_Settings\_Group1\_Active\_On\_Button“ ist die ID des beschriebenen Buttons auf dem Content `Temperature_Parameter_Settings.content` mit dem zuvor beschriebenen Schema.



**Übersicht der Controls mit möglichen Abkürzungen**

Control Typ	Abkürzung
TcHmi.Controls.Beckhoff.TcHmiButton	Button
TcHmi.Controls.Beckhoff.TcHmiCheckbox	Checkbox
TcHmi.Controls.Beckhoff.TcHmiCombobox	Combobox
TcHmi.Controls.Beckhoff.TcHmiDatagrid	Datagrid
TcHmi.Controls.Beckhoff.TcHmiEllipse	Ellipse
TcHmi.Controls.Beckhoff.TcHmiEventGrid	EventGrid
TcHmi.Controls.Beckhoff.TcHmiEventLine	EventLine
TcHmi.Controls.Beckhoff.TcHmiLine	Line
TcHmi.Controls.Beckhoff.TcHmiLinearGauge	LinearGauge
TcHmi.Controls.Beckhoff.TcHmiLocalizationSelect	LocalizationSelect
TcHmi.Controls.Beckhoff.TcHmiRadialGauge	RadialGauge
TcHmi.Controls.Beckhoff.TcHmiRadioButton	RadioButton
TcHmi.Controls.Beckhoff.TcHmiRectangle	Rectangle
TcHmi.Controls.Beckhoff.TcHmiStateImage	StateImage
TcHmi.Controls.Beckhoff.TcHmiTextblock	Textblock
TcHmi.Controls.Beckhoff.TcHmiTextbox	Textbox
TcHmi.Controls.Beckhoff.TcHmiToggleButton	ToggleButton
TcHmi.Controls.Beckhoff.TcHmiToggleSwitch	ToggleSwitch
TcHmi.Controls.Beckhoff.TcHmiTrendLineChart	TrendLineChart
TcHmi.Controls.Beckhoff.TcHmiUserManagement	UserManagement
TcHmi.Controls.FavoriteBarControl.FavoriteBarControl	FavoriteBar / FavoriteBarControl
TcHmi.Controls.Plastic.ArrowMotionGraphControl	ArrowMotionGraph / ArrowMotionGraphControl
TcHmi.Controls.Plastic.BlowPressureChart	BlowPressureChart
TcHmi.Controls.Plastic.CamControl	Cam / CamControl
TcHmi.Controls.Plastic.Configurator	Configurator
TcHmi.Controls.Plastic.CurveEditorControl	CurveEditor / CurveEditorControl
TcHmi.Controls.Plastic.InputBox	InputBox
TcHmi.Controls.Plastic.ManualOperation	ManualOperation
TcHmi.Controls.Plastic.MeasurementUnitSelector	MeasurementUnitSelector
TcHmi.Controls.Plastic.MonitoringControl	Monitoring / MonitoringControl
TcHmi.Controls.Plastic.PfwSingleTempControl	PfwSingleTempControl
TcHmi.Controls.Plastic.PfwTempParameters	PfwTempParameters
TcHmi.Controls.Plastic.ProcessScheduler	ProcessScheduler
TcHmi.Controls.Plastic.StateIndicator	StateIndicator
TcHmi.Controls.Plastic.Table	Table
TcHmi.Controls.Plastic.TimerControl	Timer / TimerControl
TcHmi.Controls.Plastic.ZoneConfiguration	ZoneConfiguration
TcHmi.Controls.Plastic.ZoneGrouping	ZoneGrouping
TcHmi.Controls.Plastic.ZoneImageLayout	ZoneImageLayout
TcHmi.Controls.ResponsiveNavigation.TcHmiBreadcrumb	Breadcrumb
TcHmi.Controls.ResponsiveNavigation.TcHmiNavigationBar	NavigationBar
TcHmi.Controls.ResponsiveNavigation.TcHmiNavigationContent	NavigationContent
TcHmi.Controls.System.TcHmiContainer	Container
TcHmi.Controls.System.TcHmiContent	Content
TcHmi.Controls.System.TcHmiGrid	Grid
TcHmi.Controls.System.TcHmiRegion	Region

Control Typ	Abkürzung
TcHmi.Controls.System.TcHmiView	View

### Standardaufbau von Contents

Die meisten Contents sind mithilfe von Grids aufgebaut. Durch die Grids können die weiteren Controls, die sich in diesem Grid befinden, leichter ausgerichtet werden, um das im gesamten Projekt durchgängige Kachel-Design zu ermöglichen. Die dem Grid hinzugefügten Inhalte bzw. Controls sind ebenfalls in kleineren Grids eingeordnet, um innerhalb einer Kachel die enthaltenen Controls einfacher ausrichten zu können. Dem Grid, das eine Kachel darstellen soll, kann über das `ClassNames`-Property der Kategorie `Common` die CSS-Klasse aus dem `Beckhoff.TwinCAT.HMI.Plastic.Themes` NuGet-Paket mit dem Namen `Tiles` zugeordnet werden, um die Hintergrundfarbe weiß zu färben, die Ecken abzurunden und einen leichten Schatten hinzuzufügen. Mithilfe der CSS-Klasse `TilesHeadline` aus dem gleichen NuGet-Paket kann das Aussehen des Textblocks für die Überschrift der Kachel schnell angepasst werden. Die Klasse färbt den Hintergrund des Textblocks in einem entsprechenden Grauton, die Textfarbe wird weiß und die Ecken werden entsprechen der Kachel angepasst. Für den Textblock der Überschrift wird üblicherweise eine Höhe von 45 px verwendet.

### Verwendung von Grids

Bei der Verwendung von Grids in Grids ist auf den Z-Index der Controls zu achten. Wenn sich ein Control (z. B. ein `TcHmiGrid`) innerhalb eines `TcHmiGrid`-Controls befindet und mit seiner Größe den vorgegebenen Bereich der entsprechenden Stelle im Grid überschreitet, führt dies unter Umständen dazu, dass sich einige enthaltene Controls (wie z. B. `TcHmiButton`-Controls) nicht mehr bedienen lassen. Somit sollte das `Zindex`-Property der Kategorie `Layout` des Controls, dass sich innerhalb eines Grids befindet, einen höheren Wert erhalten. Bei dem äußeren Grid kann das `Zindex`-Property zum Beispiel auf 0 gesetzt werden.

### Sonderfall: Erzeugung mehrerer Seiten in einem Content

Die Contents, die für den Sliderbereich gedacht sind, können nach Belieben größer als der eigentlich zur Verfügung stehende Platz sein. Somit kann zum Beispiel die Breite des Contents doppelt so breit sein, als der eigentlich zur Verfügung stehende Platz des `TcHmiRegion`-Controls, wenn eine zweite Seite erzeugt werden soll. Durch die Änderung des `Left`-Properties aus der Kategorie `Layout` wird das enthaltene Grid, das den gesamten Content füllt, entsprechend verschoben, um die bestimmten Inhalte anzuzeigen, die gewünscht sind. Dies kann, wie in dem HMI-Projekt der Plastic Application, über Controls vom Typ `TcHmiRadioButton` erfolgen.

**Beispiel:** [Handfunktionen](#) [► 160]



Verfügbar ab Version 12.6.0

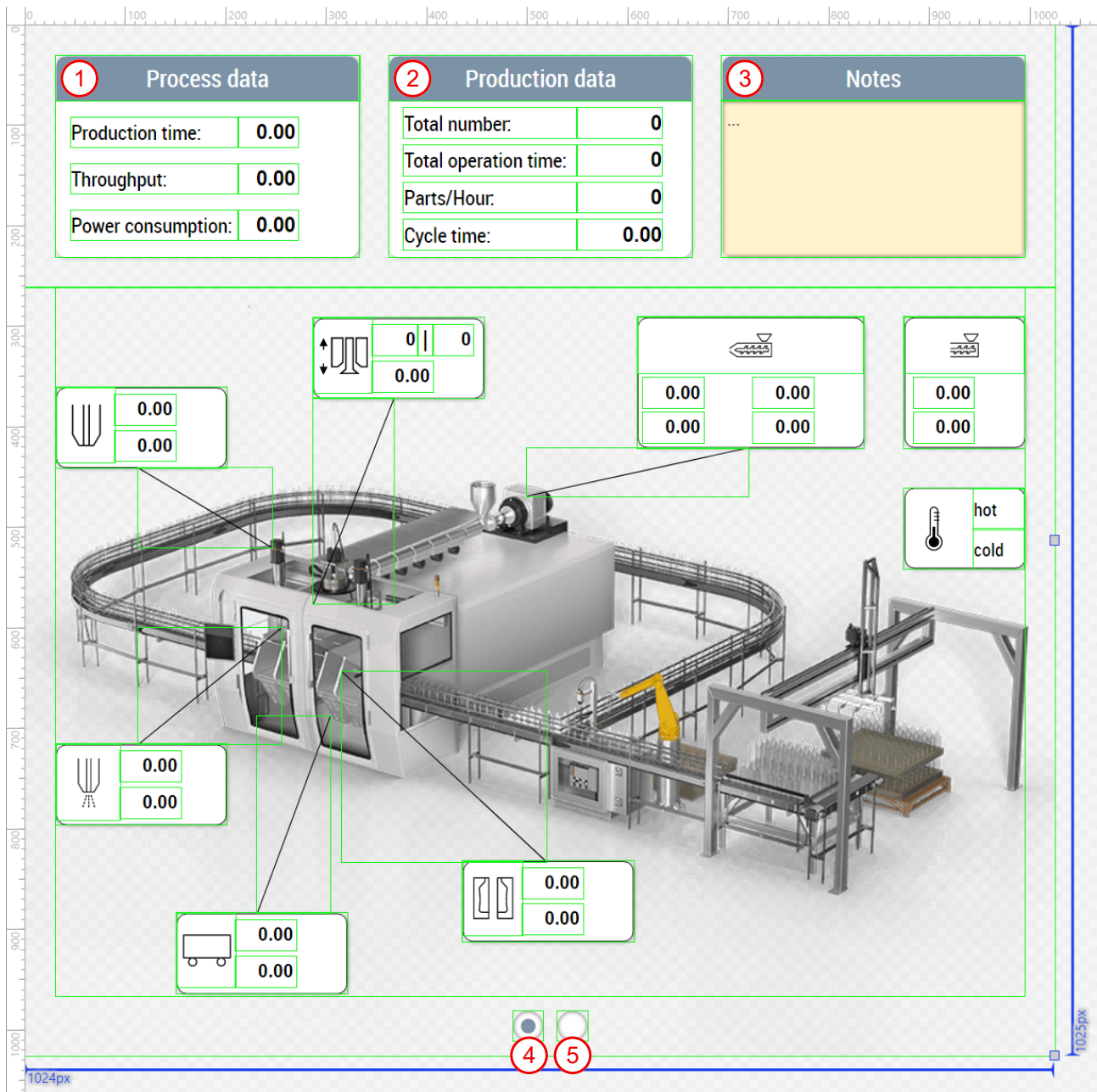
---

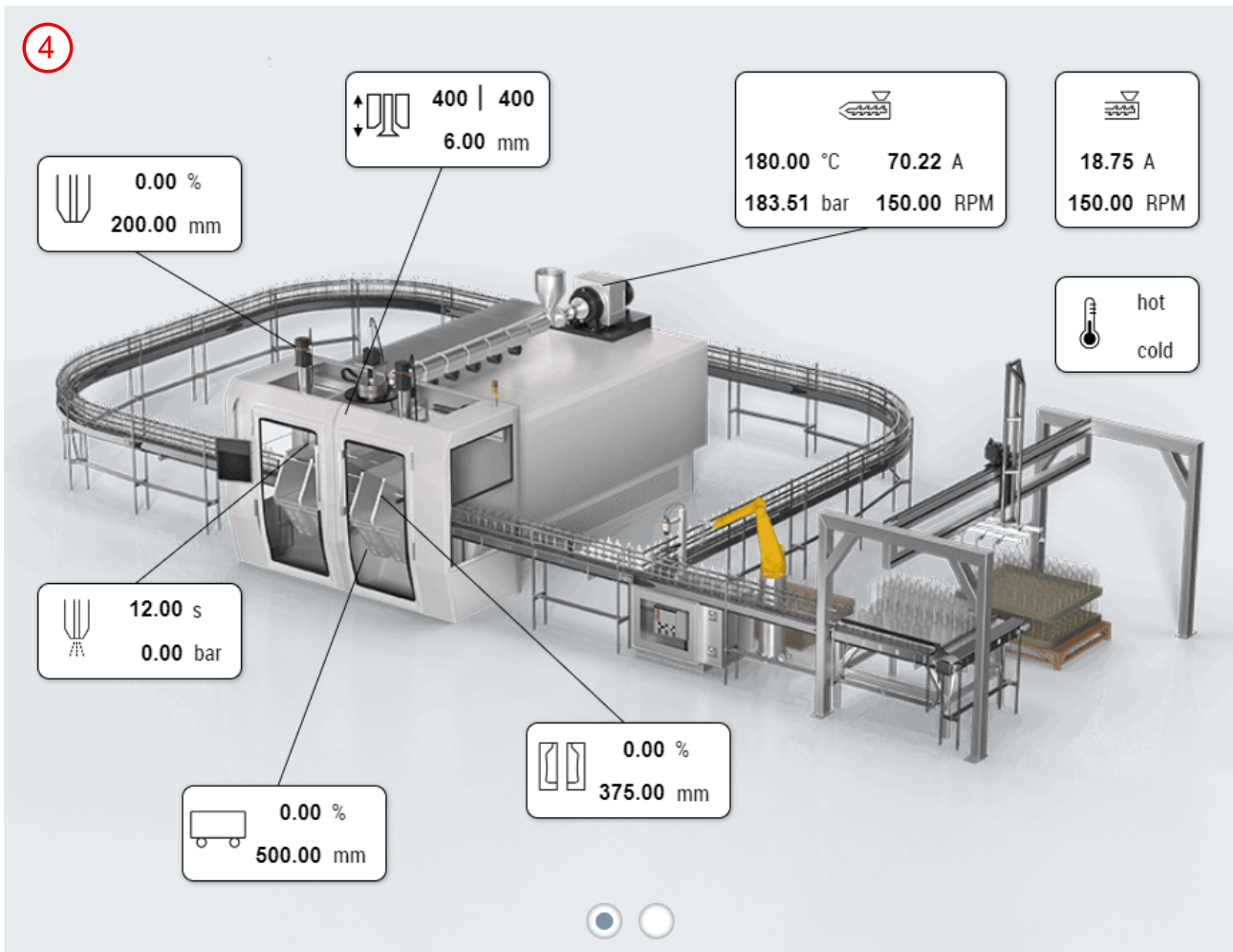
## 7.3.1 Startseite

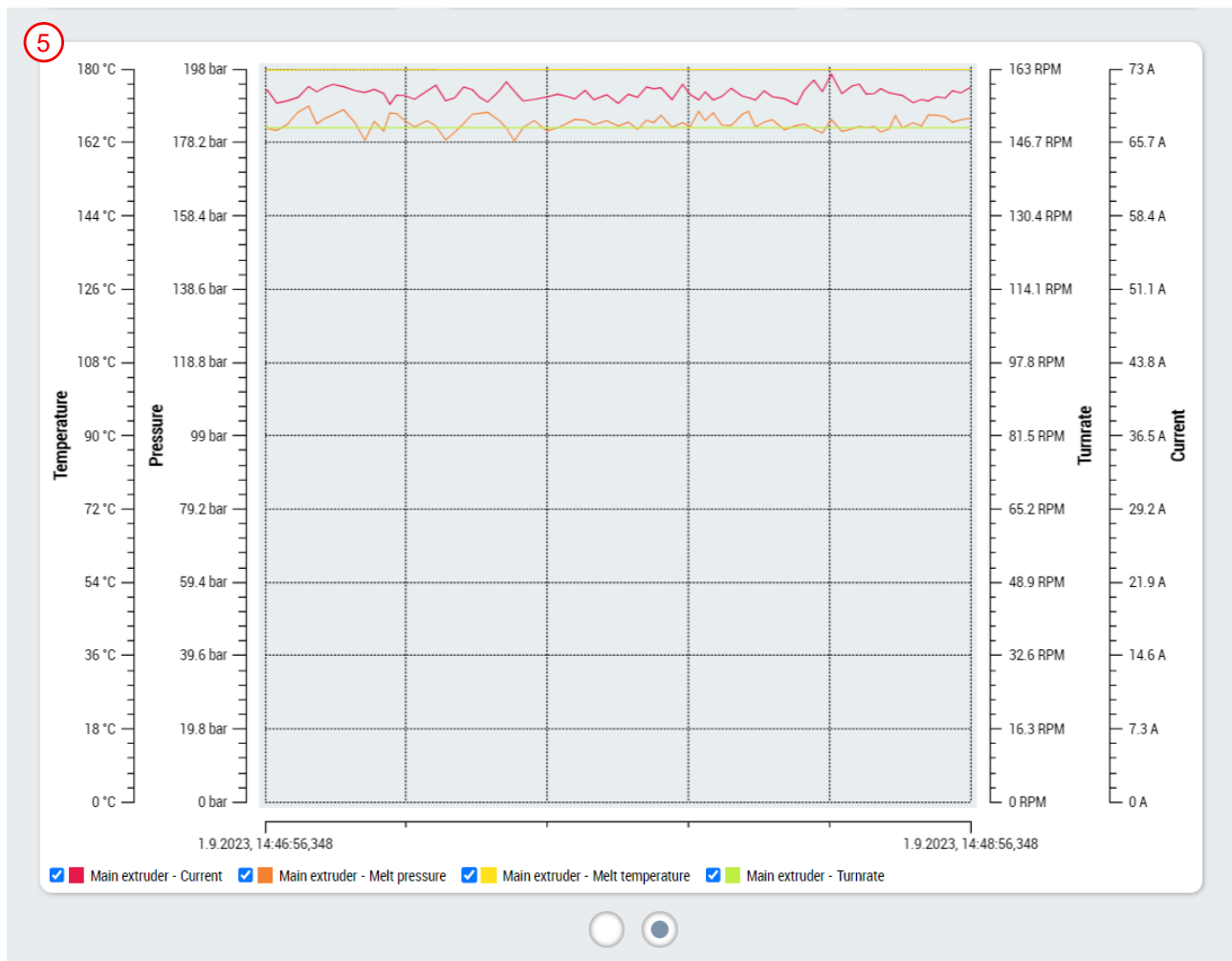
### Home.content

Die Startseite der Bedienoberfläche besteht aus einer Ansammlung von wichtigen Daten und Informationen. Die Seite enthält die folgenden Bestandteile:

1. Anzeige der wichtigsten Prozessdaten
2. Anzeige der wichtigsten Produktionsdaten
3. Feld zur Eingabe von Notizen
4. Umschaltung mithilfe des Radio Buttons zur Anzeige der Maschinengraphik mit den wichtigsten Istwerten
5. Umschaltung mithilfe des Radio Buttons zur Darstellung der Trendwerte des Hauptextruders







Verfügbar ab Version 12.6.0

## 7.3.2 Navigation

### Navigation.content

Die Navigationsseite wird im Sliderbereich über den 1. Tab aufgerufen. Sie besteht aus Controls des NuGet-Pakets `Beckhoff.TwinCAT.HMI.ResponsiveNavigation`. Das `TcHmiNavigationBar`-Control dient zur Definition der in der Navigation vorkommenden Seiten sowie zur Anzeige der Oberkategorien. Außerdem müssen die folgenden Controls festgelegt werden:

- **TcHmiNavigationContent:** Anzeige der zur Verfügung stehenden Seiten in der Navigation je nach ausgewählter Oberkategorie im `TcHmiNavigationBar`-Control (befindet sich auch auf dem `Navigation.content`)
- **TcHmiRegion:** Bereich zur Anzeige des ausgewählten Contents (Hauptcontent der [View \[► 198\]](#))
- **TcHmiBreadcrumb:** Anzeige des Breadcrumbs (befindet sich direkt auf der [View \[► 198\]](#))

Über das `MenuData`-Property der Kategorie `Common` des `TcHmiNavigationBar`-Controls wird der Navigationsinhalt bestimmt. Dieses Property kann mit dem `NavBarMenuData`-Property des `Configurator`-Controls verbunden werden. Dadurch können alle Einstellungen, die einen Bezug auf die Navigation haben, in dem `NavigationConfig`-Property des `Configurator`-Controls erledigt werden und müssen nicht mehrfach eingestellt werden.



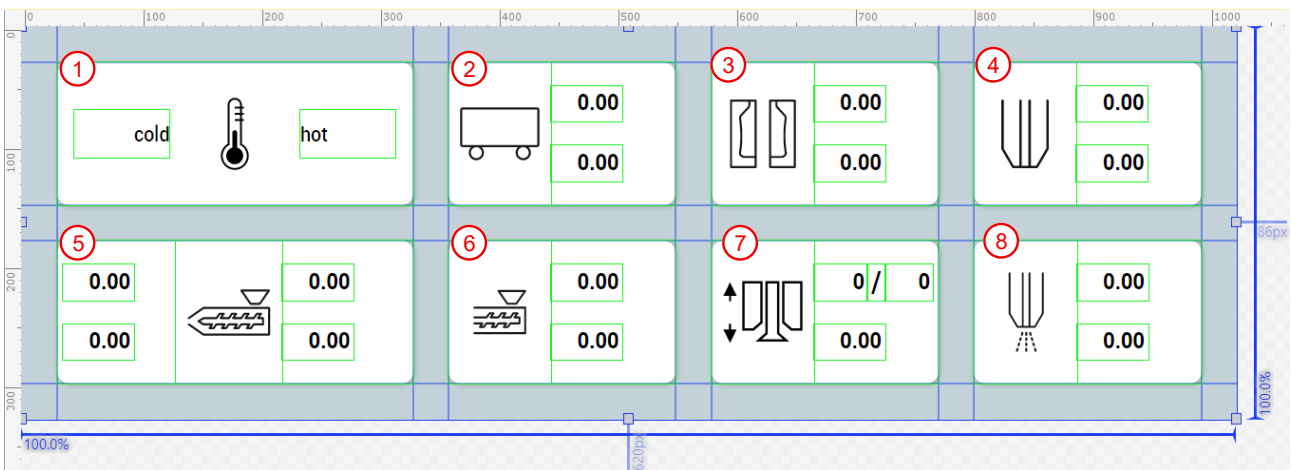
Verfügbar ab Version 12.6.0

### 7.3.3 Info

#### Info.content

Die Infoseite enthält die wichtigsten Maschinendaten. Diese Seite wird im Sliderbereich mit dem Info-Tab (2. Tab) eingeblendet. Die Seite muss an die spezielle Maschine angepasst werden. Die Variablen werden mithilfe von Icons den zugehörigen Maschinenbestandteilen zugeordnet. Die Seite enthält die folgenden Bestandteile:

1. Eine der Temperaturen befindet sich außerhalb der eingestellten Toleranz
2. Drehmoment und Position des Wagens
3. Drehmoment und Position der Schließeinheit
4. Drehmoment und Position des Blasdorns
5. Schmelztemperatur, Schmelzedruck, Strom und Geschwindigkeit des Hauptextruders
6. Strom und Geschwindigkeit des Koextruders
7. Index und Position der WDS
8. Blaszeit und Blasdruck des Blasvorgangs



Verfügbar ab Version 12.5.1

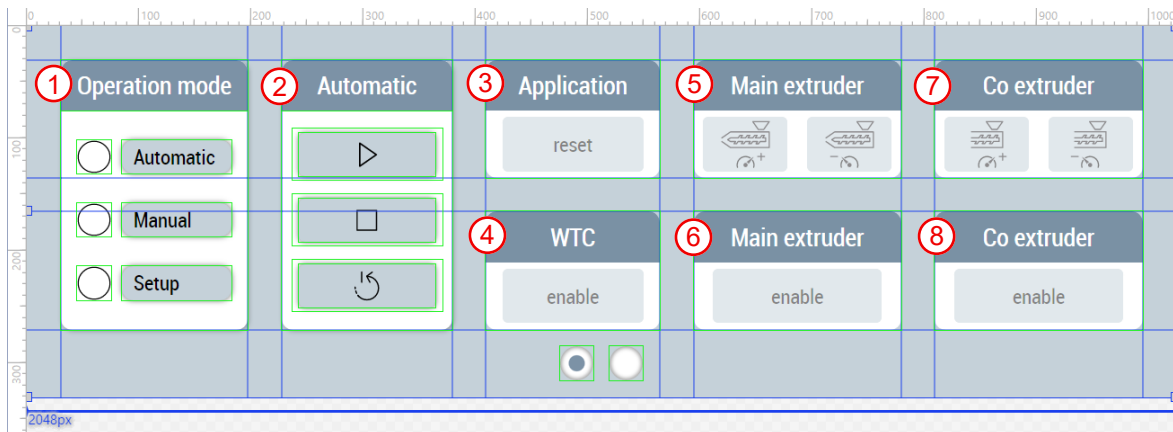
## 7.3.4 Handfunktionen

### ManualFunctions.content

Die Handfunktionsseite enthält alle Handfunktionen und muss maschinenspezifisch angepasst werden. Diese Seite wird im Sliderbereich mit dem Hand-Tab (3. Tab) eingeblendet. Mithilfe der ManualOperation-Controls und den Instanzen vom Typ `FB_ManualFunctionHmi` [► 47] können Bedienelemente für Handfunktionen leicht erzeugt werden. Der Content ist in zwei Seiten aufgeteilt.

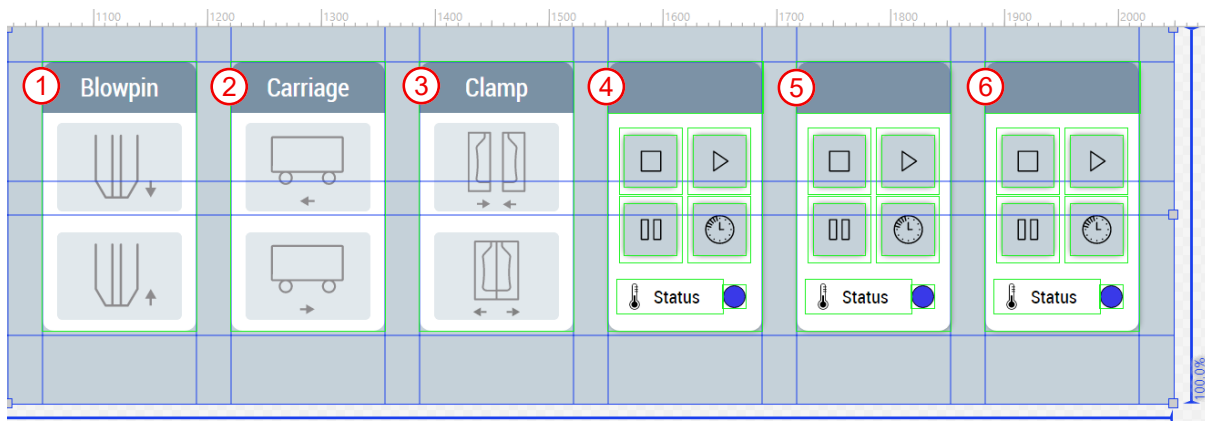
#### Die erste Seite enthält die folgenden Bestandteile:

1. Wechsel der Betriebsart
2. Ein- und ausschalten des Automatikmodus und Achsen in Basisposition fahren
3. Resetten der gesamten Applikation
4. Einschalten der WDS
5. Erhöhen und verringern der Drehzahl des Hauptextruders
6. Einschalten des Hauptextruders
7. Erhöhen und verringern der Drehzahl des Koextruders
8. Einschalten des Koextruders



#### Die zweite Seite enthält die folgenden Bestandteile:

1. Verfahren des Blasdorns
2. Verfahren des Wagens
3. Verfahren der Schließeinheit
4. Temperaturmodus der Gruppe 1 auswählen
5. Temperaturmodus der Gruppe 2 auswählen
6. Temperaturmodus der Gruppe 3 auswählen



Verfügbar ab Version 12.6.0



### 7.3.5 Achsen

Die Ordner mit dem Namen *Axes* (*Contents/Navigation/Axes* und *Contents/Slider/Axes*) beinhalten alle Contents, die für die Achsen der Maschine benötigt werden. Dazu zählen je ein Unterordner für die drei im Projekt enthaltenen Achsen ([Blasdorn \[▶ 161\]](#), [Wagen \[▶ 165\]](#) und [Schließenheit \[▶ 165\]](#)) sowie eine gemeinsame [Oszilloskopseite \[▶ 166\]](#) für alle Achsen.

#### 7.3.5.1 Blasdorn

Die Ordner mit dem Namen *Blowpin* (*Contents/Navigation/Axes/Blowpin* und *Contents/Slider/Axes/Blowpin*) beinhalten alle Contents, die für den Blasdorn benötigt werden.

Content	Beschreibung
<a href="#">Blowpin.content [▶ 161]</a>	Die Hauptseite des Blasdorns.
<a href="#">Blowpin_Settings.content [▶ 161]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <code>Blowpin.content</code> im Hauptbereich befindet.
<a href="#">Blowpin_Homing.content [▶ 163]</a>	Die Referenzfahrtseite des Blasdorns.
<a href="#">Blowpin_Homing_Settings.content [▶ 163]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <code>Blowpin_Homing.content</code> im Hauptbereich befindet.

##### 7.3.5.1.1 Bewegung

###### Blowpin.content

Die Hauptseite des Blasdorns besteht zum einen aus der Verwendung des ArrowMotionGraph-Controls und dient zur Einstellung und Darstellung der Blasdornbewegung in Segmenten. Das Control erlaubt die Einstellung der Geschwindigkeit und Position, Beschleunigung und Verzögerung sowie in beide Bewegungsrichtungen mit je bis zu 5 Segmenten.

Zum anderen enthält die Blasdorn-Hauptseite zweimal das Cam-Control für je eine Bewegungsrichtung der Achse. Die beiden Controls dienen zur Einstellung und Darstellung der Nocken mithilfe der Eingabe der Länge und Position pro Segment. Pro Control können bis zu 5 Segmente hinzugefügt werden.

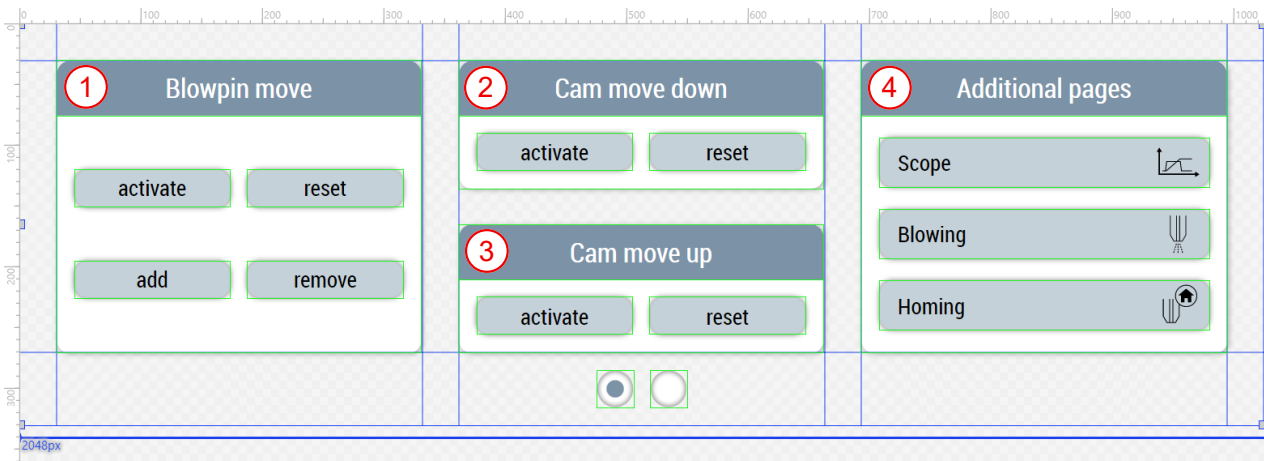


### Blowpin\_Settings.content

Zu der Hauptseite des Blasdorns gibt es noch eine ergänzende Seite für den Sliderbereich. Der Content ist in zwei Seiten aufgeteilt.

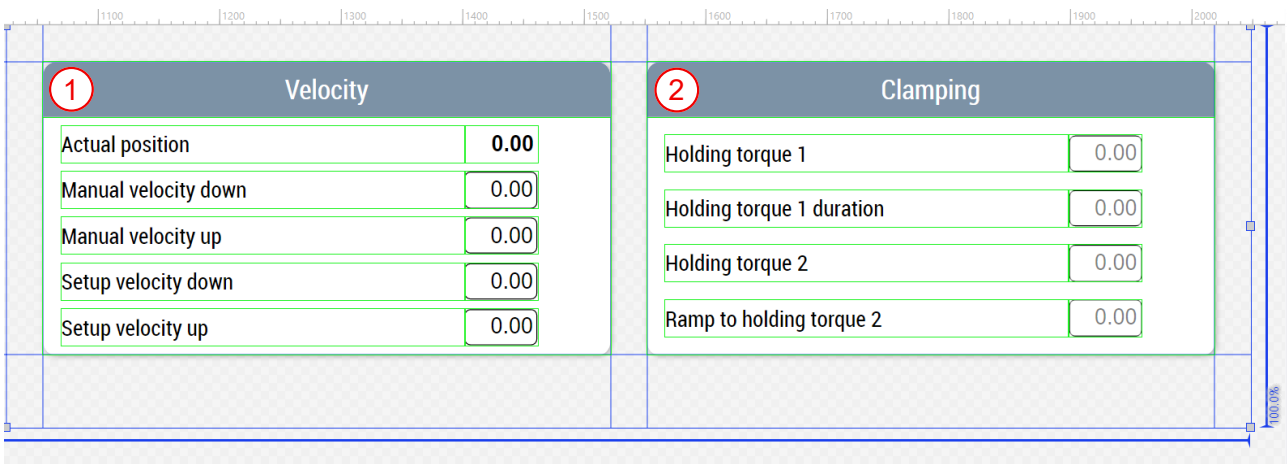
#### Die erste Seite enthält die folgenden Bestandteile:

1. Aktivieren und resetten der Einstellungen sowie hinzufügen und entfernen von Segmenten der Blasdornbewegung
2. Aktivieren und resetten der Einstellungen der Nocken in der Abwärtsbewegung des Blasdorns
3. Aktivieren und resetten der Einstellungen der Nocken in der Aufwärtsbewegung des Blasdorns
4. Bereich zur Navigation auf weitere zugehörige Seiten



Die zweite Seite enthält die folgenden Bestandteile:

1. Einstellung von Geschwindigkeitswerten
2. Einstellung von Drehmomenten fürs Spannen



Verfügbar ab Version 12.6.0

### 7.3.5.1.2 Referenzfahrt

#### Blowpin\_Homing.content

Die Referenzfahrtseite für den Blasdorn besteht aus den folgenden Bestandteilen:

1. Referenzfahrtblock
2. Referenzfahrblockerkennung
3. Referenzfahrtfinish
4. Encoder Nullpunktverschiebung

Mithilfe der Funktion `UpdateAxesHomingUnits` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket werden bestimmte Einheiten der Werte in Winkeleinheitengruppen umgewandelt, wenn es sich um eine transformierte Achse handelt.

The screenshot displays the Beckhoff HMI Homing Settings interface, organized into four main panels:

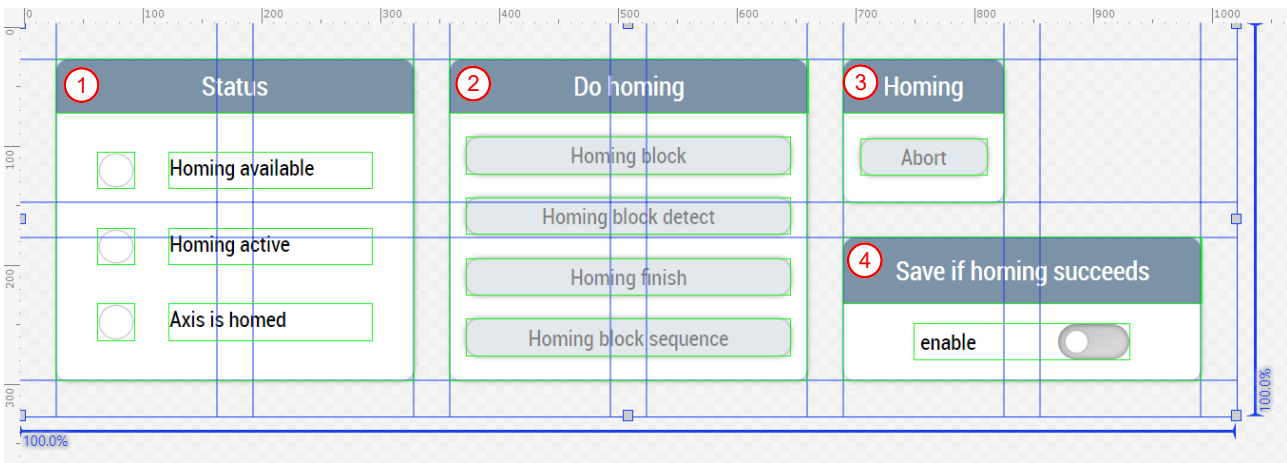
- Panel 1: Homing block**
  - Instant lag reduction:
  - Torque polarity inverted:
  - Disable drive access:
  - Direction:  ▾
  - Velocity:
  - Acceleration:
  - Deceleration:
  - Jerk:
  - Time limit:
  - Distance limit:
  - Torque limit:
  - Detection velocity limit:
  - Detection velocity time:
  - Torque tolerance:
  - Set position:
- Panel 2: Homing block detect**
  - Instant lag reduction:
  - Torque polarity inverted:
  - Disable drive access:
  - Set software end:
  - Direction:  ▾
  - Velocity:
  - Acceleration:
  - Deceleration:
  - Jerk:
  - Time limit:
  - Distance limit:
  - Torque limit:
  - Detection velocity limit:
  - Detection velocity time:
  - Torque tolerance:
  - Recorded position:
- Panel 3: Homing finish**
  - Disable drive access:
  - Distance:
  - Velocity:
  - Acceleration:
  - Deceleration:
- Panel 4: Encoder zero shift**
  - Actual position:
  - Encoder zero shift:
  - Set position to zero:

The interface includes a coordinate grid (0 to 1000 on the x-axis, 0 to 1000 on the y-axis) and a 100.0% zoom indicator at the bottom left.

### Blowpin\_Homing\_Settings.content

Zu der Referenzfahrseite des Blasdorns gibt es noch eine ergänzende Seite für den Sliderbereich. Diese Seite enthält die folgenden Bestandteile:

1. Referenzfahrt-Status-Anzeige
2. Referenzfahrt-Funktionen
3. Abbruch der Referenzfahrt
4. Aktivierung der Speichermöglichkeit



Verfügbar ab Version 12.6.0

### 7.3.5.2 Wagen

Die Ordner mit dem Namen *Carriage* (*Contents/Navigation/Axes/Carriage* und *Contents/Slider/Axes/Carriage*) beinhalten alle Contents, die für den Wagen benötigt werden.

Content	Beschreibung
Carriage.content	Die Hauptseite des Wagens.
Carriage_Settings.content	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich Carriage.content im Hauptbereich befindet.
Carriage_Homing.content	Die Referenzfahrtseite des Wagens.
Carriage_Homing_Settings.content	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich Carriage_Homing.content im Hauptbereich befindet.



Die Contents sind nahezu gleich aufgebaut, wie die Contents der [Blasdorn \[► 161\]-Achse](#).

### 7.3.5.3 Schließeinheit

Die Ordner mit dem Namen *Clamp* (*Contents/Navigation/Axes/Clamp* und *Contents/Slider/Axes/Clamp*) beinhalten alle Contents, die für die Schließeinheit benötigt werden.

Content	Beschreibung
Clamp.content	Die Hauptseite der Schließeinheit.
Clamp_Settings.content	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich Clamp.content im Hauptbereich befindet.
Clamp_Homing.content	Die Referenzfahrtseite der Schließeinheit.
Clamp_Homing_Settings.content	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich Clamp_Homing.content im Hauptbereich befindet.



Die Contents sind nahezu gleich aufgebaut, wie die Contents der [Blasdorn \[► 161\]-Achse](#).

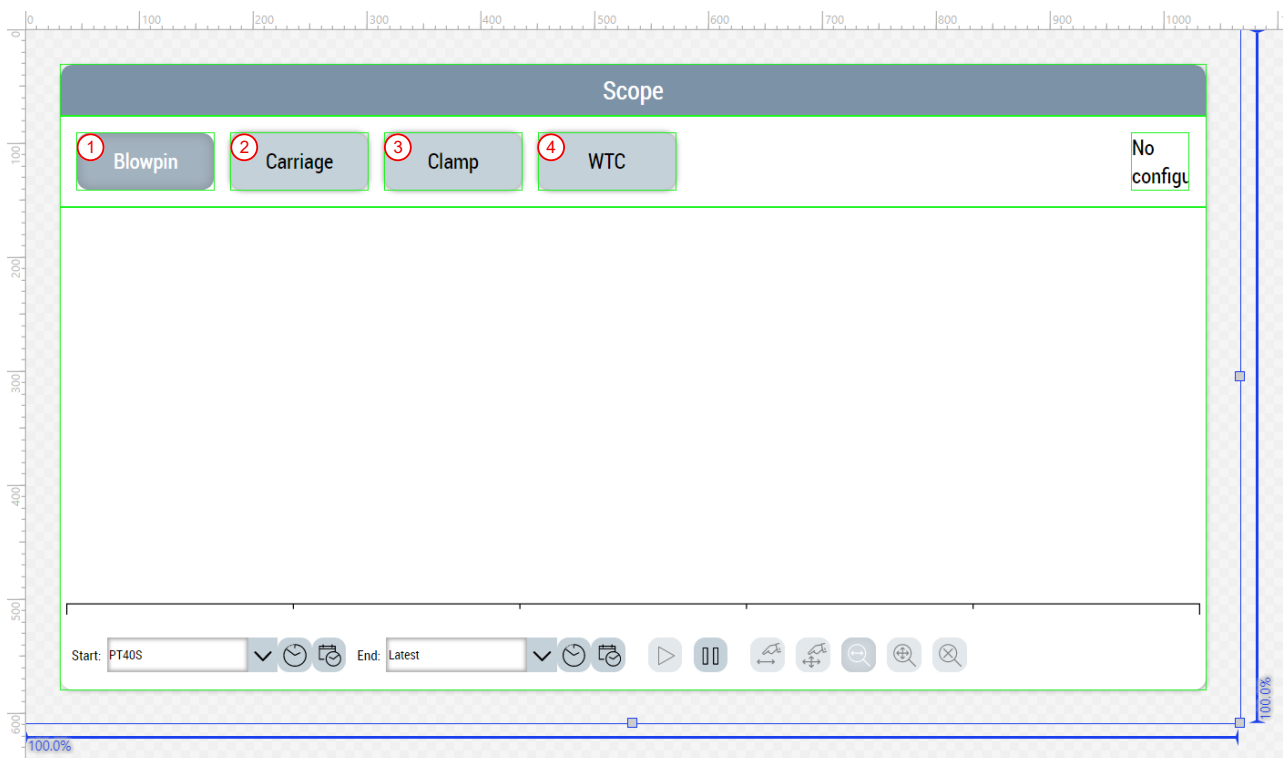
### 7.3.5.4 Oszilloskop

#### Scope.content

Die Oszilloskopseite dient zur Anzeige der folgenden Historized-Symbol-Werte pro Achse:

- Istwert der Position
- Istwert der Geschwindigkeit
- Istwert des Drehmoments

Mithilfe der `TcHmiToggleButton-Controls` wird die Funktion `UpdateScope` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket angetriggert. Die Funktion sorgt dafür, dass sich das `TcHmiTrendLineChart-Control` an die ausgewählte Achse anpasst: Blasdorn (1), Wagen (2), Schließeinheit (3), WDS (4). Zusätzlich kann über das `TcHmiStatImage-Control` ein Icon für die jeweilige Achse eingeblendet werden.



Verfügbar ab Version 12.6.0

### 7.3.6 Extruder

Die Ordner mit dem Namen *Extruder* (*Contents/Navigation/Extruder* und *Contents/Slider/Extruder*) beinhalten alle Contents, die für die Extruder benötigt werden.

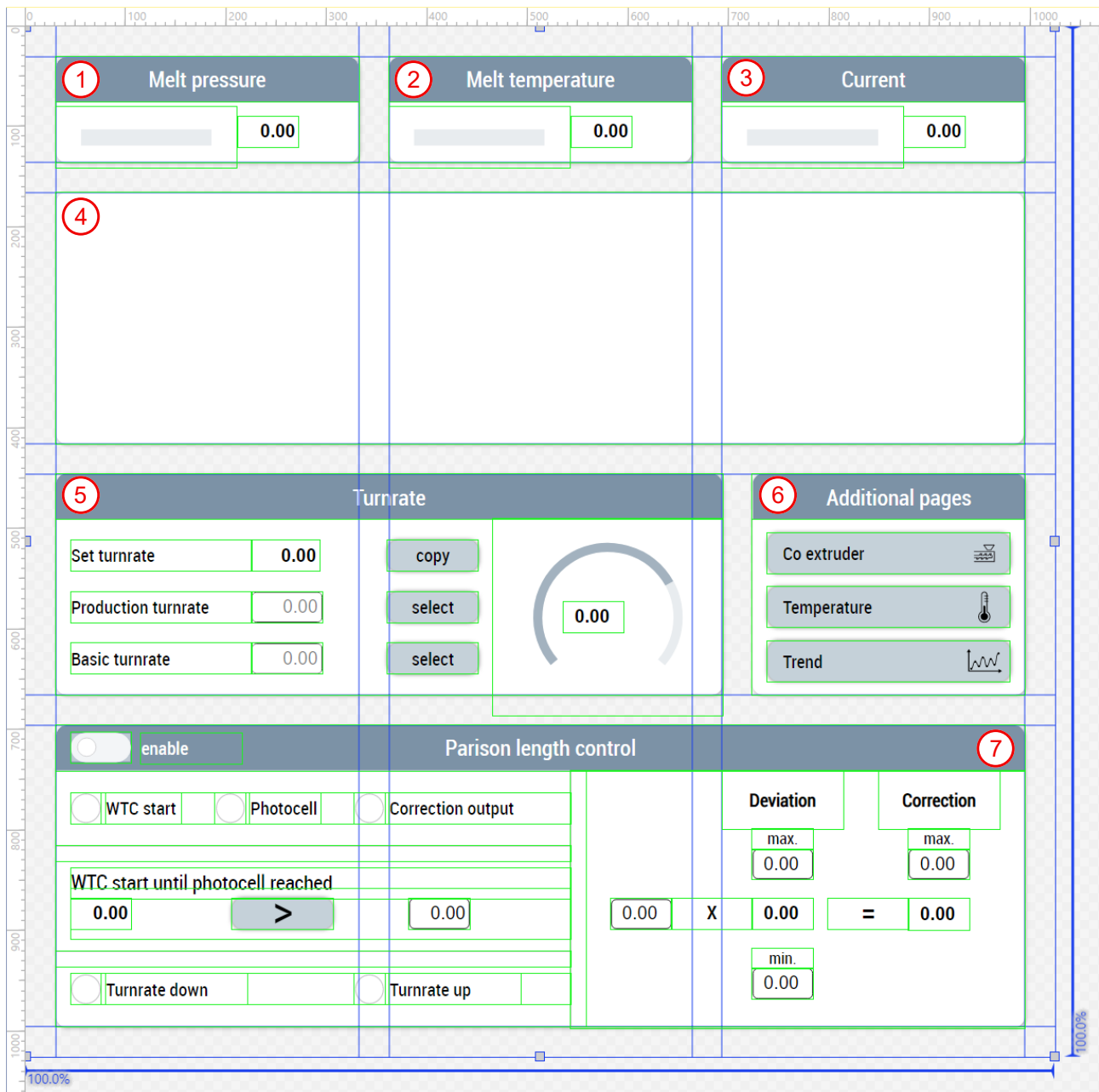
Content	Beschreibung
<a href="#">MainExtruder.content</a> [► 167]	Die Seite für den Hauptextruder.
<a href="#">CoExtruder.content</a> [► 168]	Die Seite für den Koextruder.
<a href="#">Trend.content</a> [► 169]	Die Trend-Seite für Extruder und Temperaturzonen.
<a href="#">Trend_Settings.content</a> [► 169]	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <code>Trend.content</code> im Hauptbereich befindet.

### 7.3.6.1 Hauptextruder

#### MainExtruder.content

Die Hauptextruderseite enthält die folgenden Bestandteile:

1. Anzeige des Schmelzedrucks als Wert und im Graphen
2. Anzeige der Schmelztemperatur als Wert und im Graphen
3. Anzeige des Stroms als Wert und im Graphen
4. Bildliche Darstellung des Hauptextruders mit den zugehörigen Temperaturzonen zur Anzeige der Istwerte und Einstellmöglichkeit der Sollwerte (wird über die [Layout \[► 191\]](#)-Seite der Temperaturen konfiguriert)
5. Bereich zur Anzeige und Einstellung der Drehzahl
  - Anzeige der Solldrehzahl
  - Button zum Kopieren der aktuell eingestellten Drehzahl in die Produktionsdrehzahl
  - Anzeige und Einstellung der Produktionsdrehzahl
  - Button zum Auswählen der Produktionsdrehzahl
  - Anzeige und Einstellung der Basisdrehzahl
  - Button zum Auswählen der Basisdrehzahl
  - Anzeige des Istwerts als numerischer Wert und im Graphen mit Referenz zur Basis- und Produktionsdrehzahl
6. Bereich zur Navigation auf weitere zugehörige Seiten
7. Bereich zur Einstellung der Schlauchlängenregelung



Verfügbar ab Version 12.6.0

### 7.3.6.2 Koextruder

#### CoExtruder.content

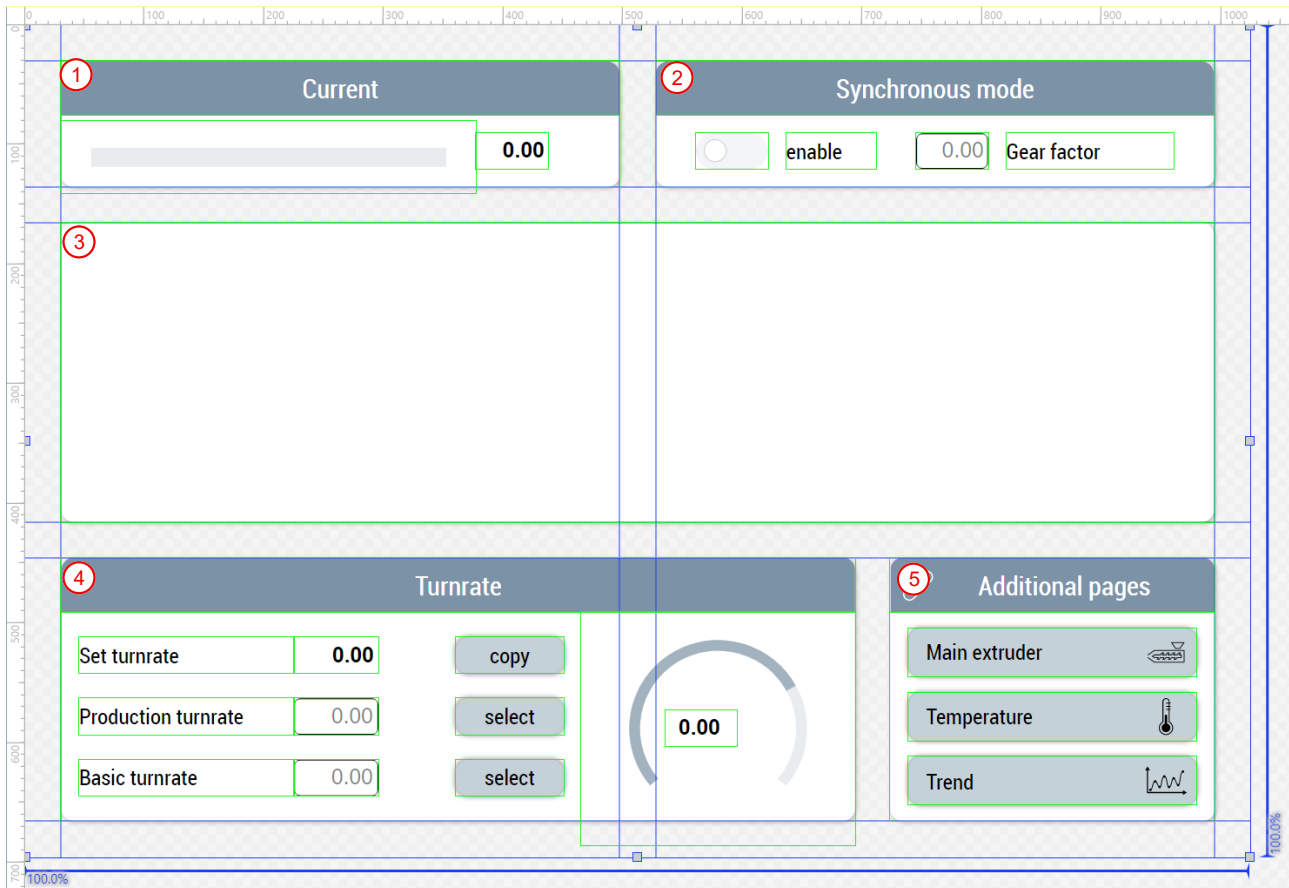
Die Koextruderseite enthält die folgenden Bestandteile:

1. Anzeige des Stroms als Wert und im Graphen
2. Bereich zum Einschalten des Synchronmodus und Einstellung des Übersetzungsfaktors
3. Bildliche Darstellung des Koextruders mit den zugehörigen Temperaturzonen zur Anzeige der Istwerte und Einstellmöglichkeit der Sollwerte (wird über die [Layout \[► 191\]](#)-Seite der Temperaturen konfiguriert)
4. Bereich zur Anzeige und Einstellung der Drehzahl
  - Anzeige der Solldrehzahl



- Button zum Kopieren der aktuell eingestellten Drehzahl in die Produktionsdrehzahl
- Anzeige und Einstellung der Produktionsdrehzahl
- Button zum Auswählen der Produktionsdrehzahl
- Anzeige und Einstellung der Basisdrehzahl
- Button zum Auswählen der Basisdrehzahl
- Anzeige des Istwerts als numerischer Wert und im Graphen mit Referenz zur Basis- und Produktionsdrehzahl

5. Bereich zur Navigation auf weitere zugehörige Seiten

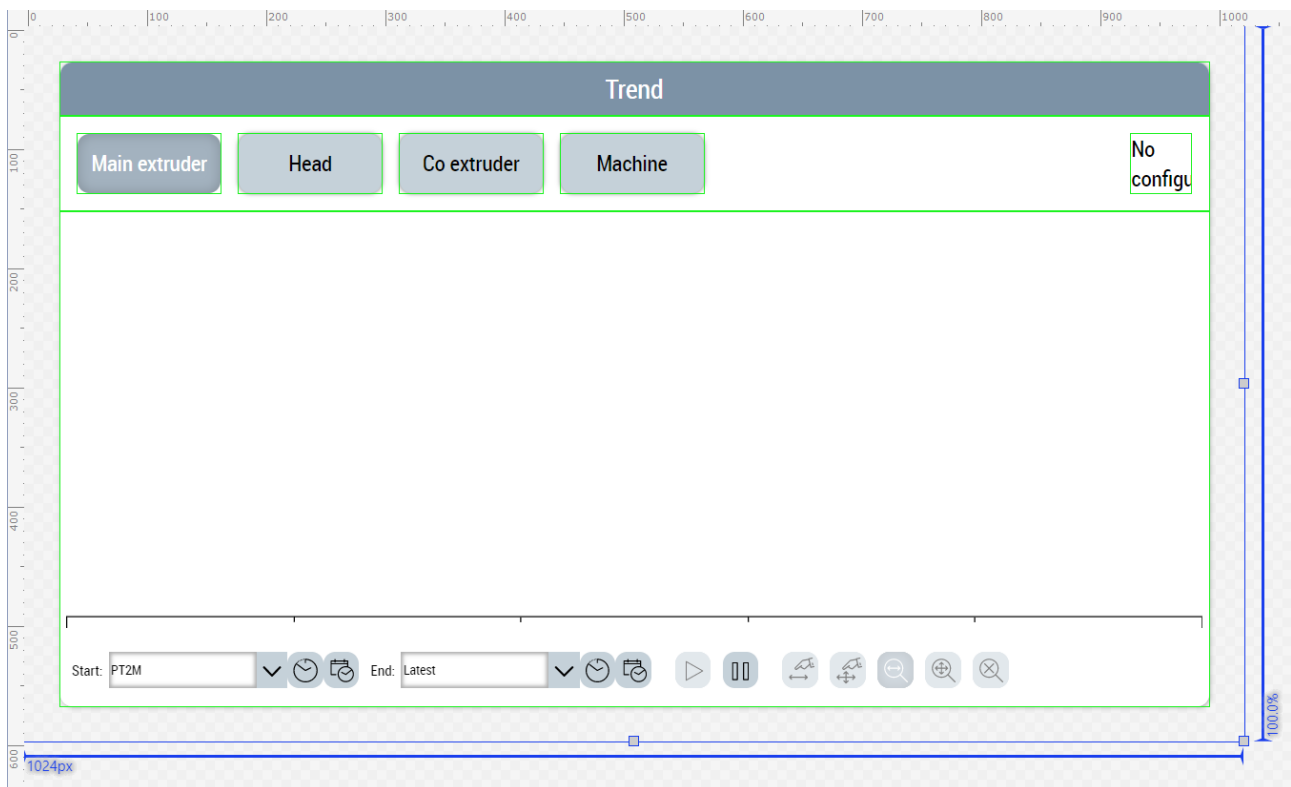


Verfügbar ab Version 12.6.0

### 7.3.6.3 Trend

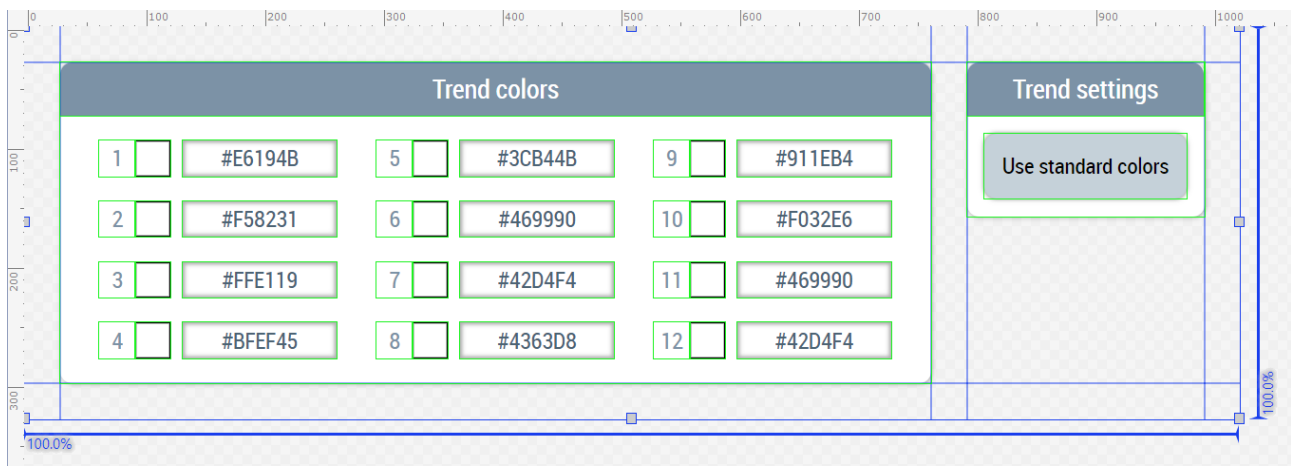
#### Trend.content

Die Trendseite dient zur Darstellung von Verläufen der Extruder- und Temperaturwerte in einem Trendgraphen (Historized Symbols). Mithilfe der Funktion `UpdateTrend` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket und dem `FB_TrendHmi` [► 111] wird das `TcHmiTrendLineChart`-Control je nach angeklicktem `TcHmiToggleButton`-Control angepasst. Zusätzlich kann über das `TcHmiStateImage`-Control ein Icon für die Auswahl eingeblendet werden.



### Trend\_Settings.content

Über diesen Content können die Farben der Kurvenverläufe angepasst werden. Die Eingabe des Hexadezimal-Farbcodes erfolgt über TchmiInput-Controls. Das TchmiButton-Control triggert die Funktion `ResetTrendAxisColors` aus dem Beckhoff.TwinCAT.HMI.Plastic.Functions NuGet-Paket an, um die Änderungen wieder mit Standardfarben überschreiben zu können.



Verfügbar ab Version 12.5.1

## 7.3.7 Parameter

Die Ordner mit dem Namen *Parameters* (*Contents/Navigation/Parameters* und *Contents/Slider/Parameters*) beinhalten alle Contents, die für die Konfiguration der Maschinendaten benötigt werden.

Content	Beschreibung
Parameters_Blowpin.content	Die Konfigurationsseite der Maschinendaten für den Blasdorn.
Parameters_Carriage.content	Die Konfigurationsseite der Maschinendaten für den Wagen.
Parameters_Clamp.content	Die Konfigurationsseite der Maschinendaten für die Schließeinheit.
Parameters_CoExtruder.content	Die Konfigurationsseite der Maschinendaten für den Koextruder.
Parameters_MainExtruder.content	Die Konfigurationsseite der Maschinendaten für den Hauptextruder.
Parameters_Monitoring.content	Die Konfigurationsseite der Maschinendaten für die Überwachungswerte.
Parameters_Setpoints.content	Die Konfigurationsseite der Maschinendaten für die Sollwerte.
Parameters_Temperature.content	Die Konfigurationsseite der Maschinendaten für die Temperaturzonen und die Temperaturversorgung.
Parameters_Wtc.content	Die Konfigurationsseite der Maschinendaten für die WDS.
Parameters_Settings.content	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich eine der Konfigurationsseiten im Hauptbereich befindet.

Eine Konfigurationsseite wird mithilfe des Table-Controls aufgebaut. Das Control bietet die Möglichkeit, diverse Variablen in unterschiedlichen Datentypen mit Einheiten anzulegen und auf verschiedene Unterseiten aufzuteilen. Mithilfe der Funktion `FillParamTable` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket kann das Table-Control automatisch aufgrund der übergebenen Parameter mit den entsprechenden Werten gefüllt werden. Änderungen in der PLC werden automatisch in der Tabelle übernommen. Weitere Zeilen lassen sich über das anpassen des `FirstTable`-Attributs des Table-Controls ergänzen.

**Parameters\_Blowpin.content**

Die Konfigurationsseite der Maschinendaten für den Blasdorn besteht aus drei Unterseiten. Die Variablen sind durch Zwischenüberschriften kategorisiert. Durch die `FillParamTable` Funktion wird die Tabelle automatisch gefüllt und aktualisiert, wenn es Änderungen in der PLC gibt.

Row			Unit
<b>Axis</b>			
2	Drive reversed	<input checked="" type="checkbox"/>	
3	Encoder interpolation	1048576.00	
4	Encoder reversed	<input checked="" type="checkbox"/>	
5	Encoder weighting	5.00	
6	Encoder zero shift	0.00	
7	Lag Kp	1.00	
8	Lag filter	0.02	
9	Lag limit	5.00	mm
10	Lag monitored	<input checked="" type="checkbox"/>	
11	Max acceleration	276.00	mm/s <sup>2</sup>
12	Max deceleration	276.00	mm/s <sup>2</sup>
13	Max jerk	830.00	mm/s <sup>3</sup>
14	Max application velocity	184.00	mm/s
15	Velo Max System	203.00	mm/s
<input type="button" value="←"/> <input type="button" value="→"/> Page 1 3			

Row			Unit
16	Software end max	200.00	mm
17	Enable software end max	<input checked="" type="checkbox"/>	
18	Software end min	0.00	mm
19	Enable software end min	<input checked="" type="checkbox"/>	
<b>Blowpin</b>			
21	Clamping distance	0.50	mm
22	Max settable clamping duration	30.00	s
23	Max settable clamping ramp	5.00	s
24	Clamping velocity	50.00	mm/s
<b>Homing</b>			
26	Detection velocity max settable time	10.00	s
27	Max settable distance	800.00	mm
28	Max settable Position	500.00	mm
29	Min settable Position	0.00	mm
30	Max settable time limit	30.00	s
<input type="button" value="←"/> <input type="button" value="→"/> Page 2 3			

Row		Unit
<b>Linear motion</b>		
32	Jog velocity negative	10.00 mm/s
33	Jog velocity positive	10.00 mm/s
34	Tool adaption inverted	<input type="checkbox"/>
35	Tool adaption offset	0.00 mm
<b>NC</b>		
37	Torque capacity	10.00
38	Torque limiting idle value	30.00 %
39	Torque limiting max value	100000.00 %
40	Torque limiting reference value	100.00 %

< > Page 3/3

### Parameters\_Temperature.content

Die Konfigurationsseite der Maschinendaten für die Temperatur besteht aus zwei Table-Controls, die mithilfe des TcHmiToggleSwitch-Controls umgeschaltet werden:

1. Parametrierung der Temperaturkanäle
2. Parametrierung der Temperaturversorgung

Durch die Funktionen `FillParamTable` und `UpdateTemperatureParametersTableNumberEx` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket, werden die Tabellen automatisch gefüllt und die Anzahl an Temperaturzonen aus der PLC ermittelt, um das Table-Control dementsprechend anzupassen.

Row	Last tuning: 2023-09-01T12:38:11Z	① Channels	<input checked="" type="checkbox"/> Temperature supply	Unit
<b>Temperature channel</b>				
2	Absolute high		<input type="text" value="300.00"/>	%
3	Absolute low		<input type="text" value="100.00"/>	%
4	Actual current		<b>0.00</b>	A
5	Actual temperature gain		<input type="text" value="1.00"/>	
6	Actual temperature offset		<input type="text" value="0.00"/>	%
7	Actual temperature		<b>18.00</b>	%
8	Cold junction compensation mode		<input type="text" value="Disabled"/>	▼
9	Cold junction compensation zone		<input type="text" value="0"/>	
10	Cooler swap index		<input type="text" value="0"/>	
11	Disable auto step		<input type="checkbox"/>	
12	Disable terminal communication		<input type="checkbox"/>	
13	Enable		<input type="checkbox"/>	
14	Enable error heating		<input type="checkbox"/>	
15	Error current tolerance		<input type="text" value="0.00"/>	

< > Page 1|6 Zone 1|20 < >

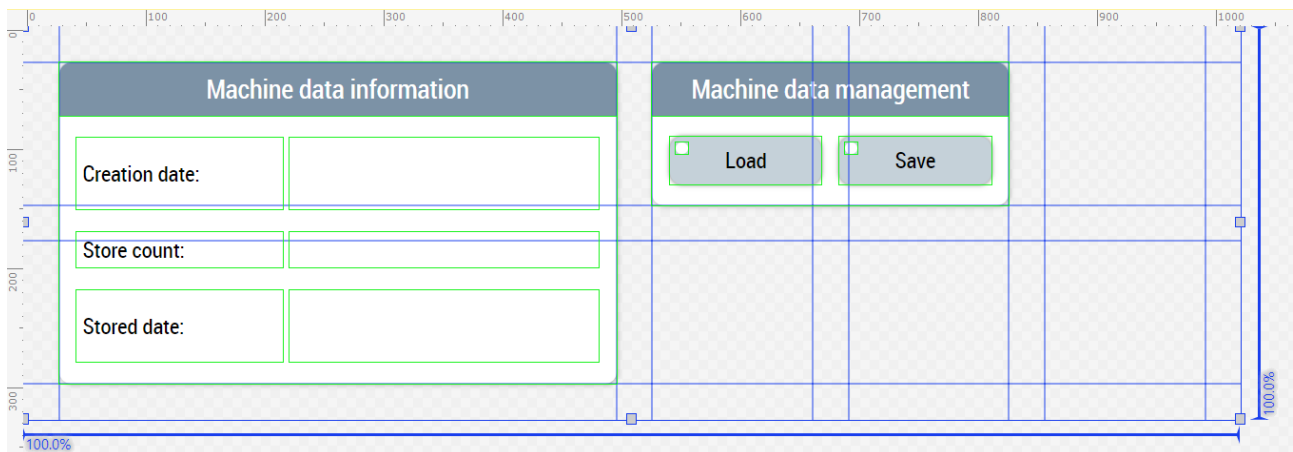
Row	Phase 1	Channels	<input checked="" type="checkbox"/> Temperature supply	② Unit
<b>Temperature supply</b>				
2	PWM cycle time		<input type="text" value="1.00"/>	
3	PWM factor cooling		<input type="text" value="10"/>	
4	PWM max on C		<input type="text" value="0.00"/>	
5	PWM max on time		<input type="text" value="1.00"/>	s
6	PWM max ramp load		<input type="text" value="1.00"/>	
7	PWM min on time		<input type="text" value="0.10"/>	s
8	Use supply PWM parameters		<input type="checkbox"/>	

Temperature supply 1|4 < >

**Parameters\_Settings.content**

Als Ergänzung dient der speziell für den Sliderbereich angefertigte Content. Er kann über den 4. Tab des Sliderbereichs angewählt werden, wenn der Hauptbereich eine der Konfigurationsseiten anzeigt. Enthalten sind Informationen über die abgespeicherte Datei sowie Funktionen zum Laden und Speichern dieser Daten.

Die angezeigten Texte und Funktionen der Buttons werden je nach angezeigter Konfigurationsseite im Hauptbereich der Bedienoberfläche mithilfe der Funktion `UpdateParametersSettingsControls` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket automatisch angepasst.



Verfügbar ab Version 12.6.0

### 7.3.8 Prozess

Die Ordner mit dem Namen *Process* (*Contents/Navigation/Process* und *Contents/Slider/Process*) beinhalten alle Contents, die für den Prozess der Maschinen benötigt werden.

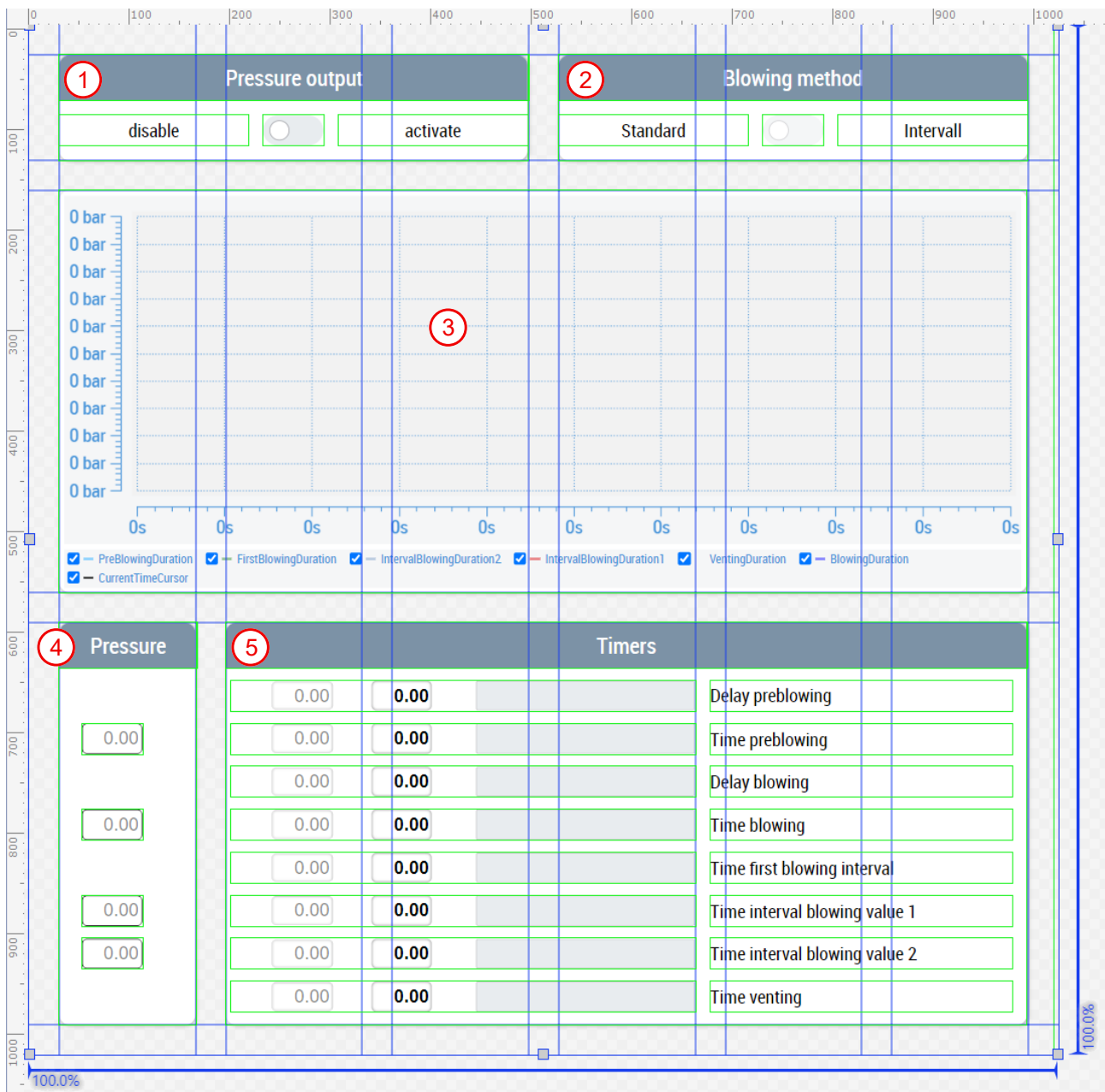
Content	Beschreibung
<a href="#">Blowing.content</a> [► 175]	Die Seite für den Blasprozess.
<a href="#">Monitoring.content</a> [► 176]	Die Seite für die Überwachung von Werten.
<a href="#">Setpoints.content</a> [► 178]	Die Seite für Sollwerte.
<a href="#">Timer.content</a> [► 179]	Die Seite für Zeiten.

#### 7.3.8.1 Blasen

##### Blowing.content

Der Blasseite besteht aus den folgenden Bestandteilen:

1. Aktivieren oder deaktivieren des Druckausgangs
2. Wahl der Blasmethode zwischen Standard- und Intervall-Blasen
3. Blasverlauf abgebildet im `BlowPressureChart-Control`
4. Druckeinstellungen zu den bestimmten Zeiten
5. Einstellung der Verzögerungen und Zeiten mithilfe von `Timer-Controls`



Verfügbar ab Version 12.5.1

### 7.3.8.2 Überwachung

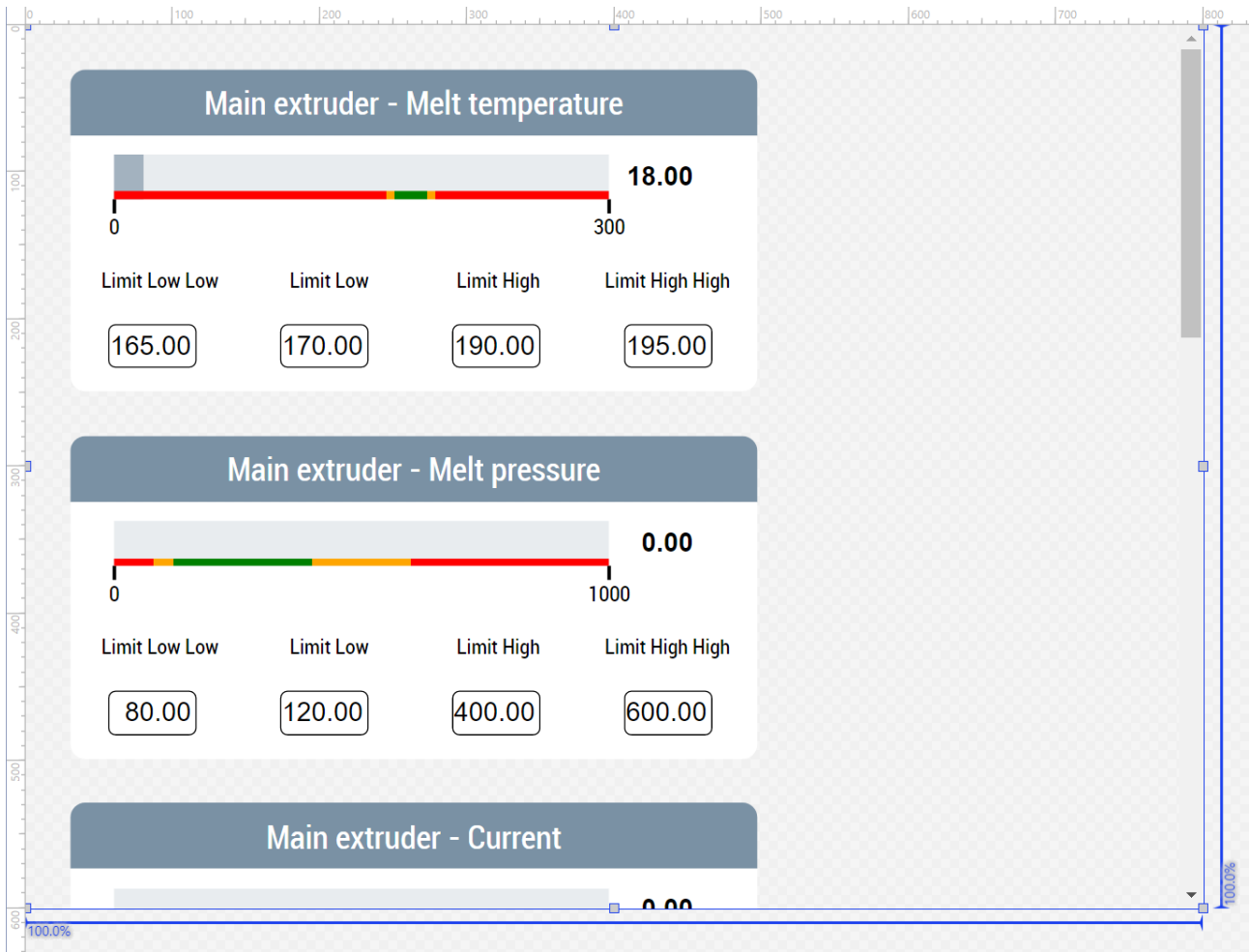
#### Monitoring.content

Die Überwachungsseite wird mithilfe des Monitoring-Controls aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket und Instanzen vom Typ `FB_MonitoringHmi` aufgebaut. Sie beinhaltet pro Wert je einen Bereich zur Anzeige des aktuellen Wertes und zur Einstellung der Schwellwerte. Zusätzlich werden die eingestellten und gemessenen Werte in einem Graphen veranschaulicht. Die aktuell enthaltenen Werte sind die Folgenden:

1. Schmelztemperatur des Hauptextruders
2. Schmelzedruck des Hauptextruders
3. Strom des Hauptextruders
4. Strom des Koextruders



- 5. Schaltschranktemperatur
- 6. CPU Temperatur
- 7. Hydrauliksystemdruck



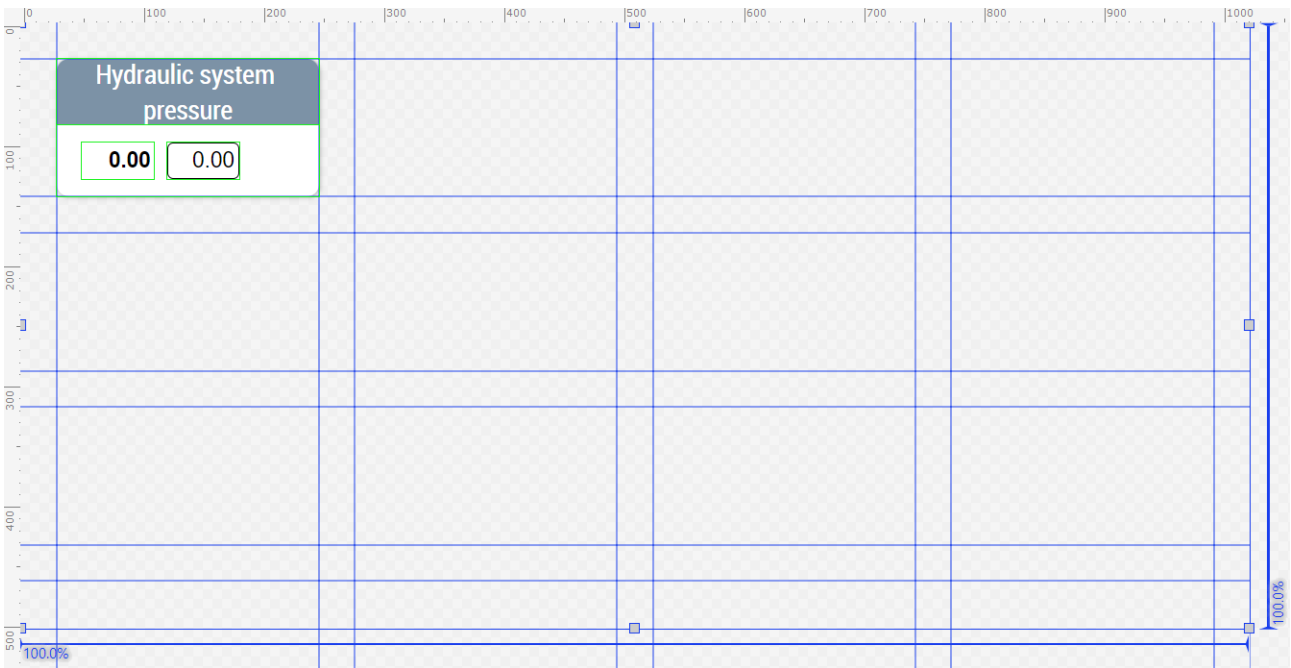


Verfügbar ab Version 12.6.0

### 7.3.8.3 Sollwerte

#### Setpoints.content

Die Sollwertseite bietet die Möglichkeit der Einstellung von Sollwerten. Aktuell ist der Sollwert des Hydrauliksystemdrucks enthalten.

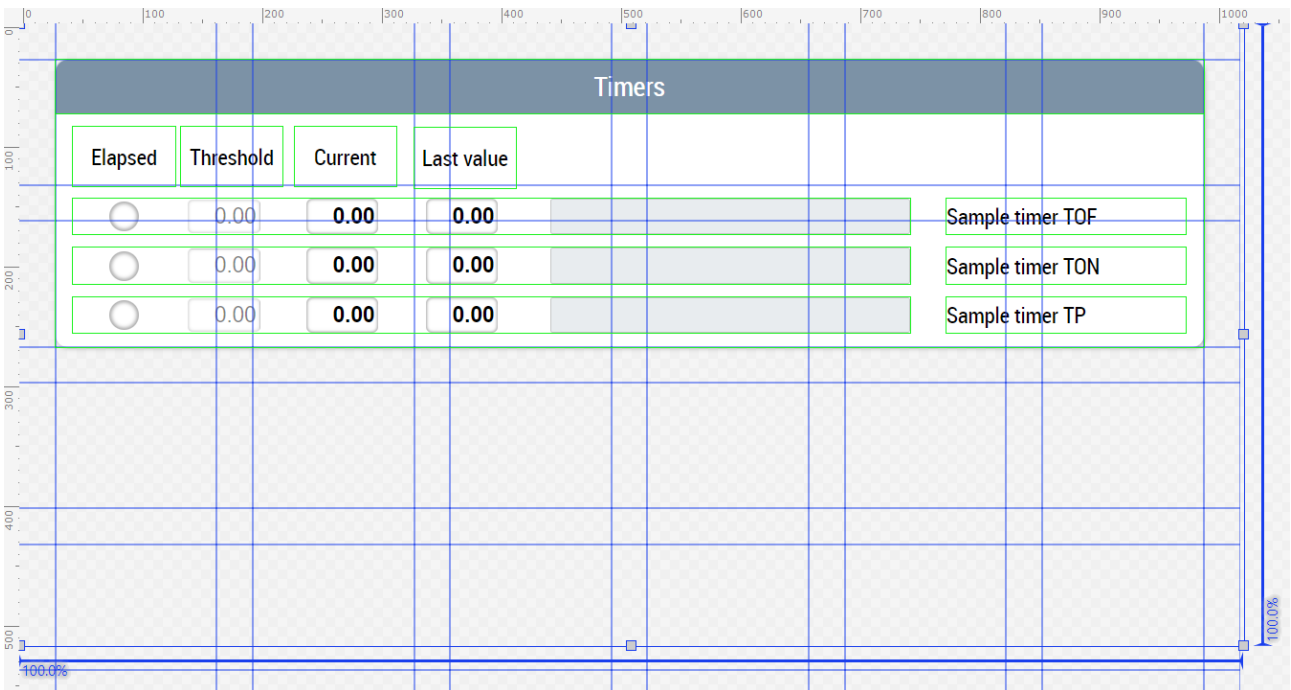


Verfügbar ab Version 12.5.1

### 7.3.8.4 Zeiten

#### Timer.content

Die Zeitsseite dient zur Einstellung und Überwachung von Zeiten. Mithilfe von Timer-Controls und dem FB TimerHmi [► 90] können Bedienelemente für Zeiten erzeugt werden.



Verfügbar ab Version 12.5.1

## 7.3.9 System

Die Ordner mit dem Namen *System* (*Contents/Navigation/System* und *Contents/Slider/System*) beinhalten alle systemrelevanten Contents der Maschine.

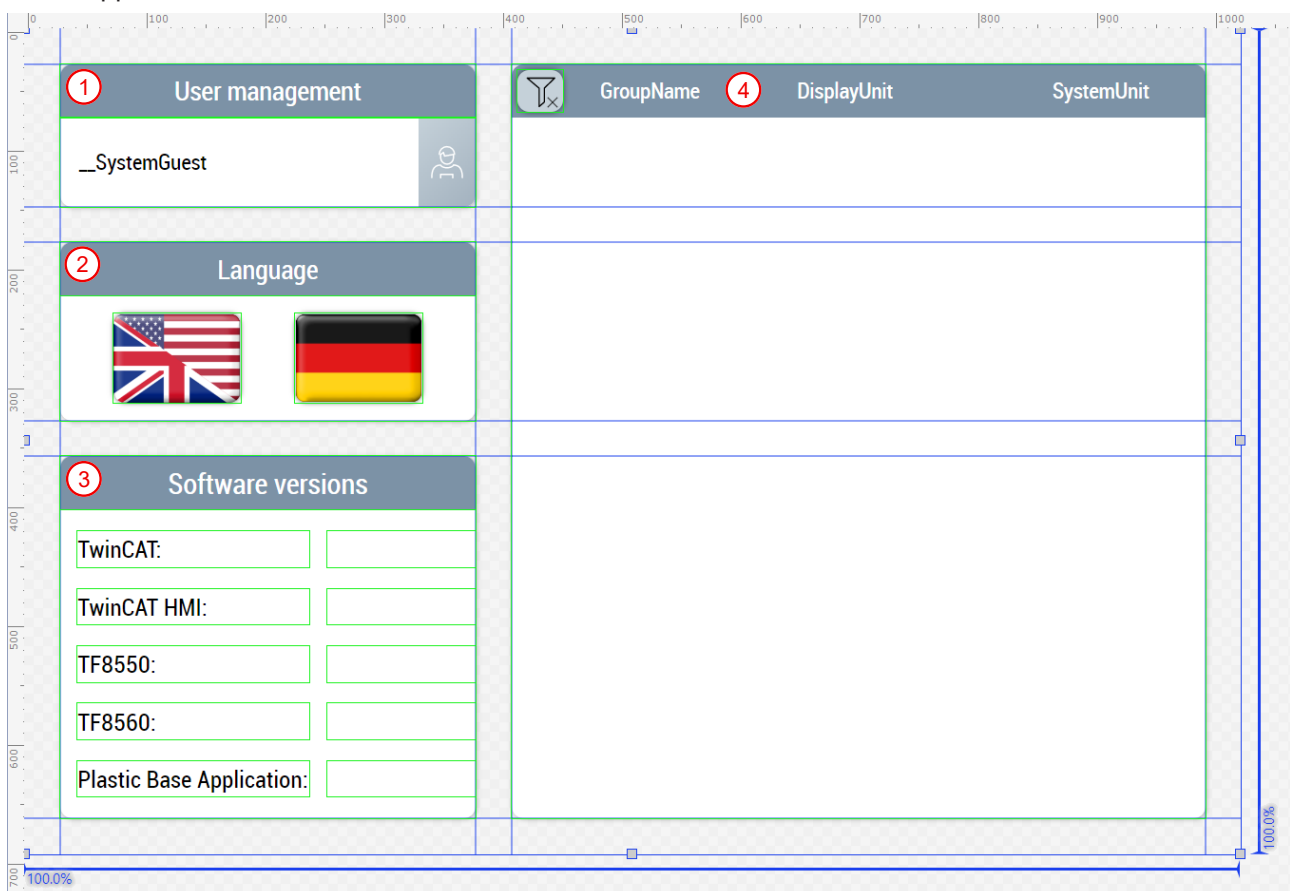
Content	Beschreibung
<a href="#">Administration.content</a> [▶ 180]	Die Seite für die Administration.
<a href="#">Alarms.content</a> [▶ 181]	Die Seite für Events und Alarme.
<a href="#">RecipeManagement.content</a> [▶ 182]	Die Seite für die Rezeptverwaltung.
<a href="#">RecipeManagement.Settings.content</a> [▶ 182]	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <code>RecipeManagement.content</code> im Hauptbereich befindet.

### 7.3.9.1 Administration

#### Administration.content

Die Administrationsseite dient zur Verwaltung und Einstellung der Bedienoberfläche und enthält die folgenden Bestandteile:

1. TcHmiUserManagement-Control zum Logout, Benutzer wechseln, Benutzereigenschaften bearbeiten und Benutzer verwalten
2. Bereich zum Wechseln der Sprache mit je einem TcHmiToggleButton-Control pro Sprache bzw. Flagge
3. Bereich zur Anzeige der Softwareversionen
4. MeasurementUnitSelector-Control zur Umschaltung der Einheiten im HMI-Projekt der Plastic Application





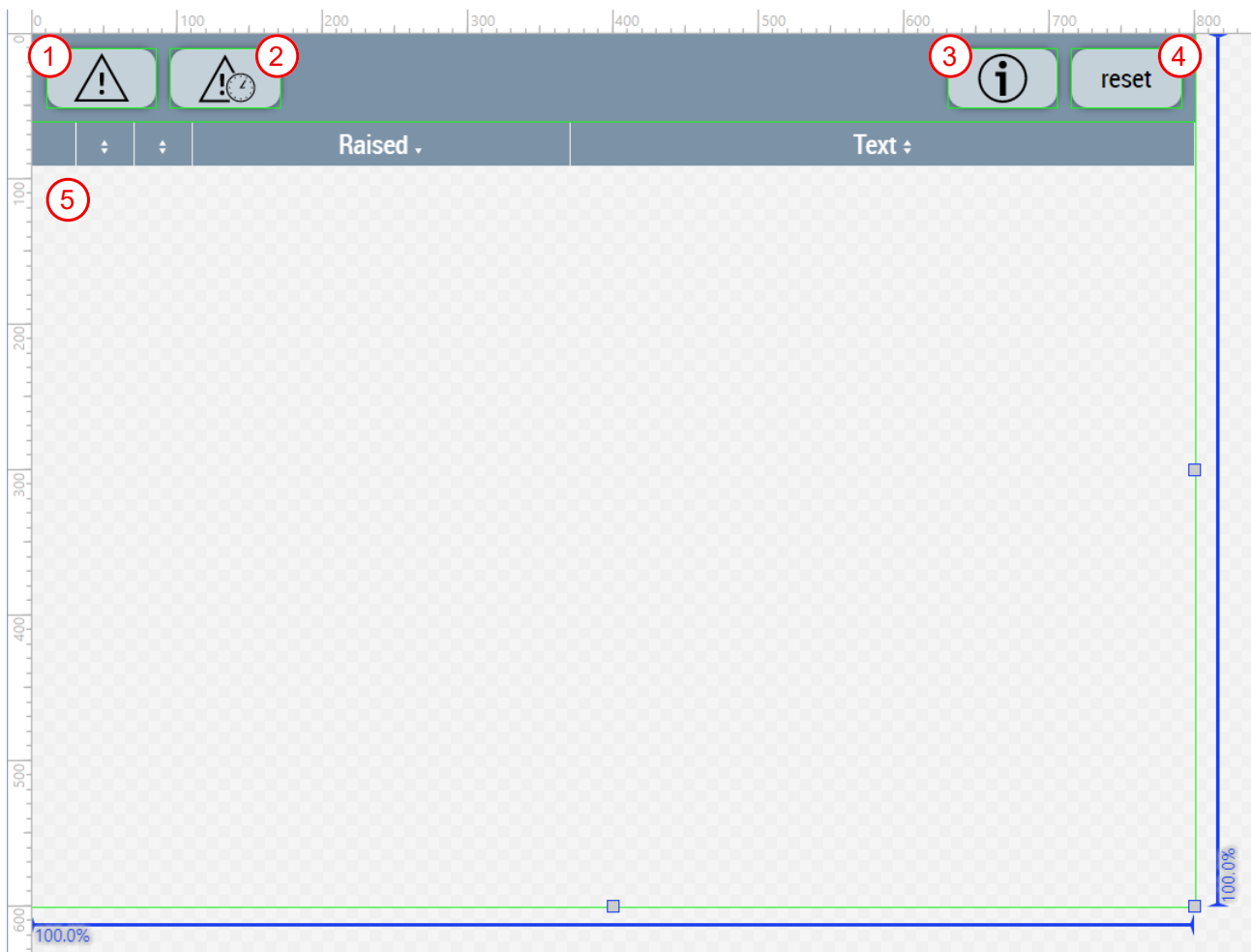
Verfügbar ab Version 12.5.1

### 7.3.9.2 Events / Alarme

#### Alarms.content

Die Alarm- und Event-Seite enthält ein TcHmiEventGrid-Control, das durch eine entsprechende Konfiguration alle Events und Alarme anzeigt, die von der PLC übertragen werden. Das Control bietet weitere Filtereinstellungen, um die Anzeige der Events zu beschränken. Die Events können genauer angeguckt und quittiert werden. Zusätzlich implementierte TcHmiButton-Controls vereinfachen das TcHmiEventGrid-Control mit vordefinierten Filtern. Die Seite enthält die folgenden Bestandteile:

1. Button zur Anzeige der anstehenden Alarme
2. Button zur Anzeige aller bislang geworfenen Alarme
3. Button zur Anzeige der Meldungen
4. Button zum resetten der anstehenden Alarme
5. Bereich zur Auflistung der Alarme und Meldungen mithilfe des TcHmiEventGrid-Controls



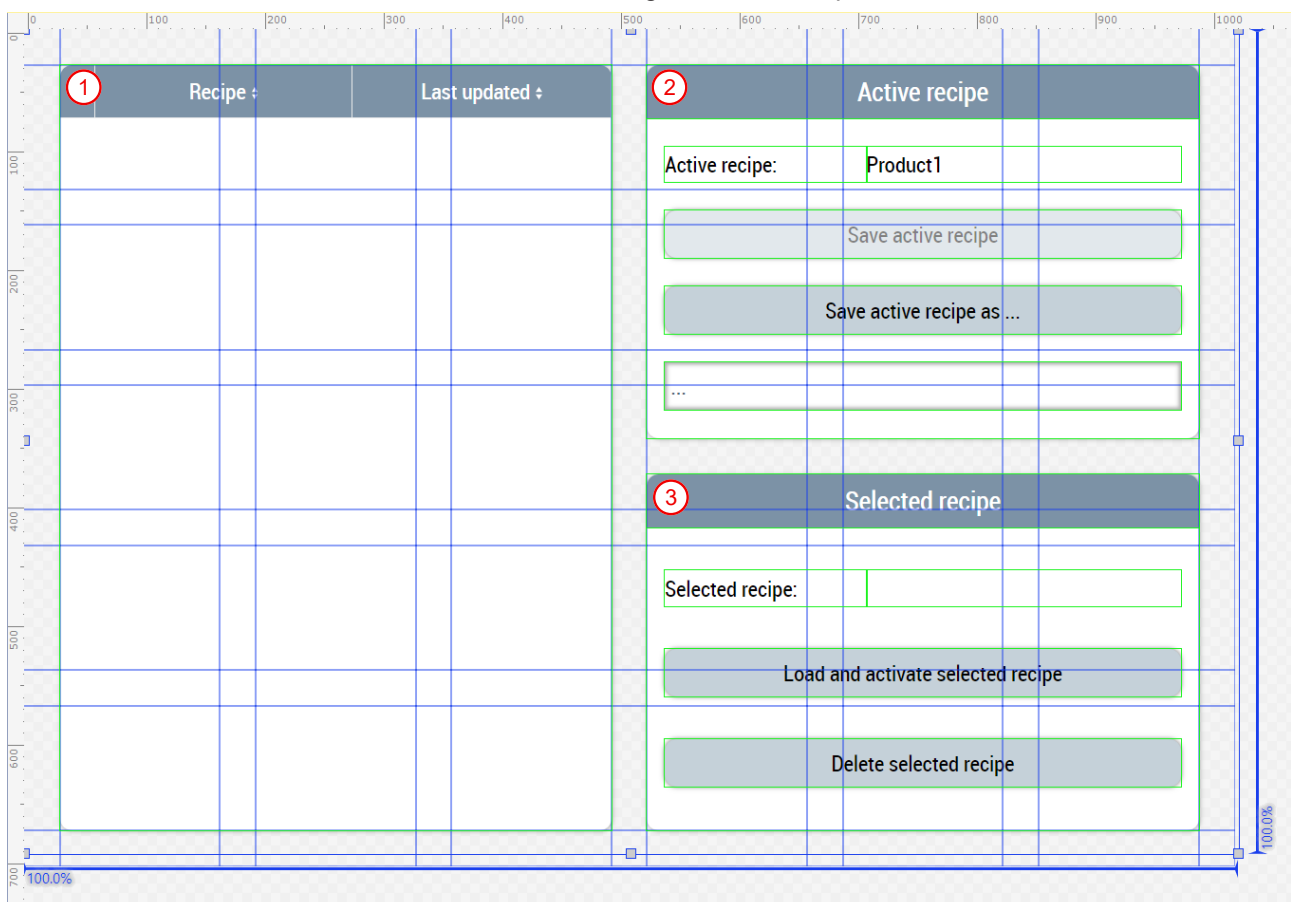
Verfügbar ab Version 12.6.0

### 7.3.9.3 Rezeptverwaltung

#### RecipeManagement.content

Die Rezeptverwaltungsseite besteht aus den folgenden Elementen:

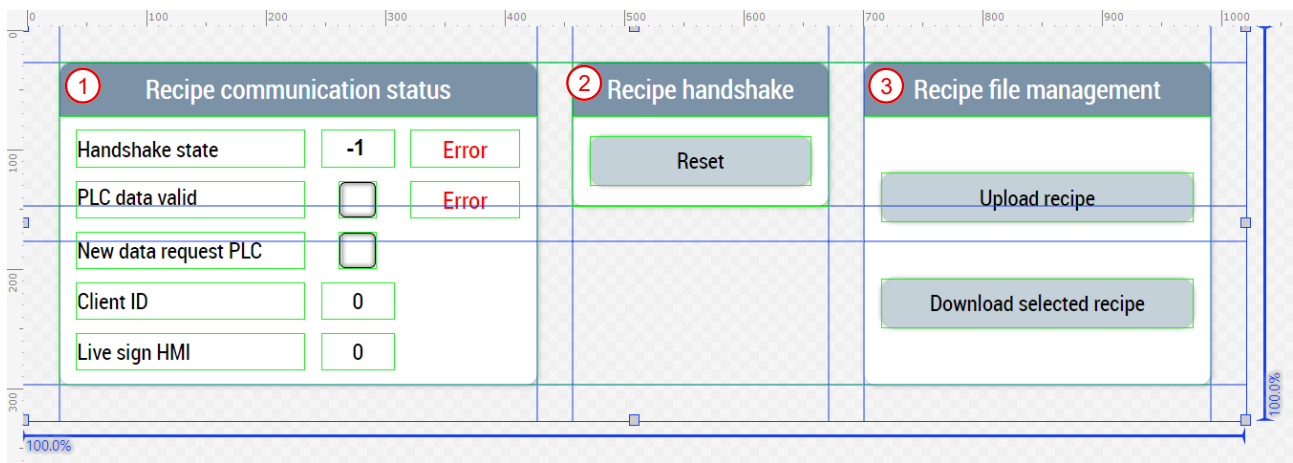
1. TcHmiDatagrid-Control als Tabelle zur Anzeige aller Rezepte mit dem Datum des letzten Updates
2. Bereich für das aktive Rezept:
  - Anzeige des Namens des aktiven Rezepts
  - TcHmiButton-Control zum Speichern der Änderungen in das aktive Rezept
  - TcHmiButton-Control zum Speichern der aktuellen Werte in ein neues Rezept
  - TcHmiTextbox-Control zum Eintragen des neuen Rezeptnamens
3. Bereich für das ausgewählte Rezept:
  - Anzeige des Namens des ausgewählten Rezeptes in der Tabelle
  - TcHmiButton-Control zum Laden und Aktivieren des ausgewählten Rezepts
  - TcHmiButton-Control zum Löschen des ausgewählten Rezepts



#### RecipeManagement\_Settings.content

Dieser Content wird im Sliderbereich angezeigt und dient als Ergänzung. Er besteht aus den folgenden Elementen:

1. Anzeige des Rezeptkommunikationsstatus
2. TcHmiButton-Control zum Resetten der Rezeptkommunikation
3. Bereich für die Rezeptdateiverwaltung:
  - TcHmiButton-Control zum Hochladen von Rezepten
  - TcHmiButton-Control zum Runterladen des ausgewählten Rezepts



Verfügbar ab Version 12.5.1

### 7.3.10 Temperatur

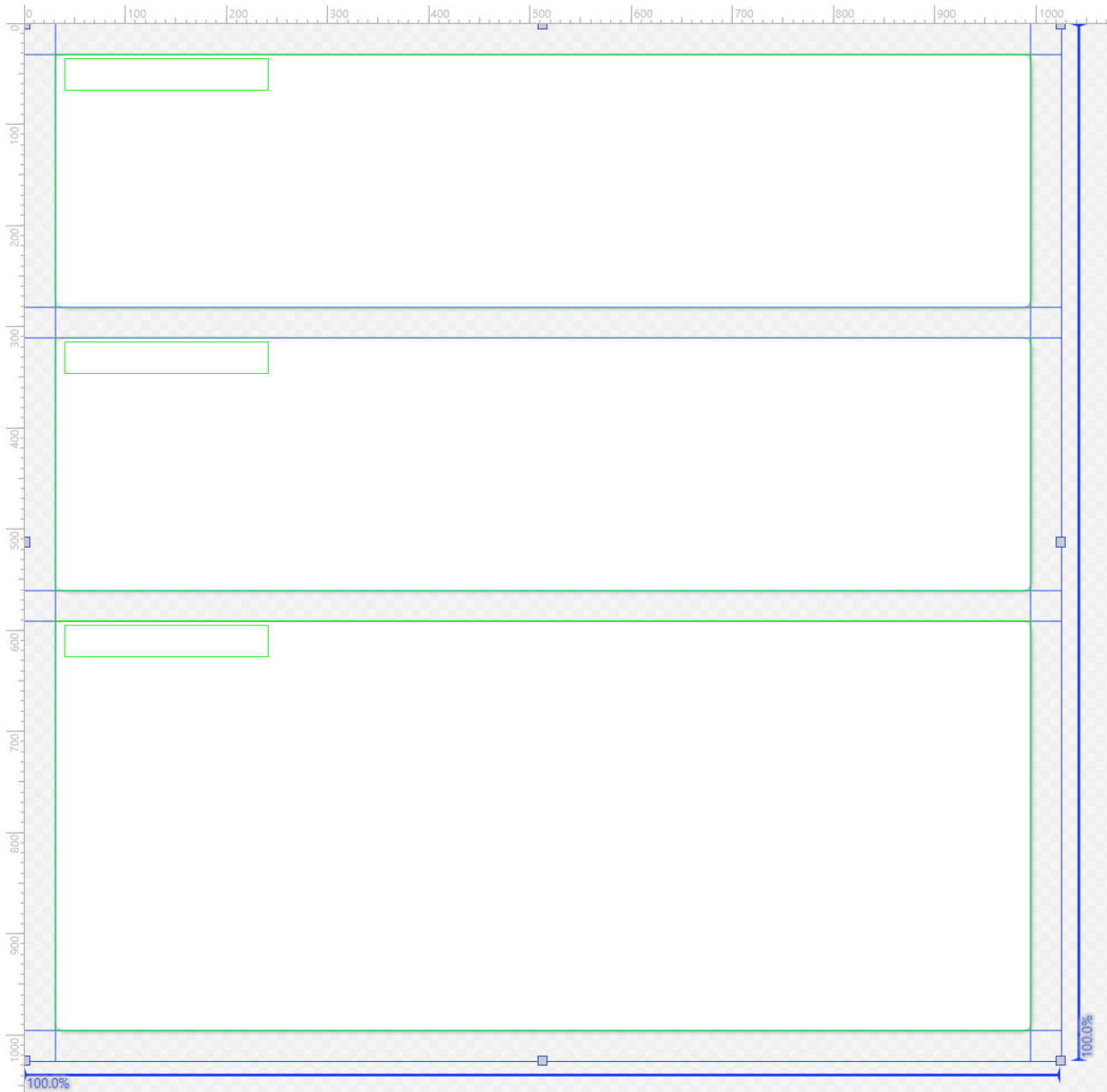
Die Ordner mit dem Namen *Temperature* (*Contents/Navigation/Temperature* und *Contents/Slider/Temperature*) beinhalten alle Contents, die für die Temperaturregelung an der Maschine benötigt werden.

Content	Beschreibung
<a href="#">Temperature_Overview.content [► 184]</a>	Die Seite zur Einzelzonenüberwachung der Temperaturen.
<a href="#">Temperature_Overview_Settings.content [► 184]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Temperature_Overview.content</i> im Hauptbereich befindet.
<a href="#">Temperature_Configuration.content [► 186]</a>	Die Seite für die Einstellung der Temperaturzonenparameter.
<a href="#">Temperature_Configuration_Settings.content [► 186]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Temperature_Configuration.content</i> im Hauptbereich befindet.
<a href="#">Temperature_TimeScheduling.content [► 188]</a>	Die Seite zur Zeitsteuerung der Temperaturzonen und Temperaturgruppen.
<a href="#">Temperature_TimeScheduling_Settings.content [► 188]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Temperature_TimeScheduling.content</i> im Hauptbereich befindet.
<a href="#">Temperature_Grouping.content [► 189]</a>	Die Seite zur Gruppierung der Temperaturkanäle in Temperaturgruppen.
<a href="#">Temperature_Layout.content [► 191]</a>	Die Seite zur Erzeugung von Layouts mit Temperaturzonen.
<a href="#">Temperature_Layout_Settings.content [► 191]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Temperature_Layout.content</i> im Hauptbereich befindet.

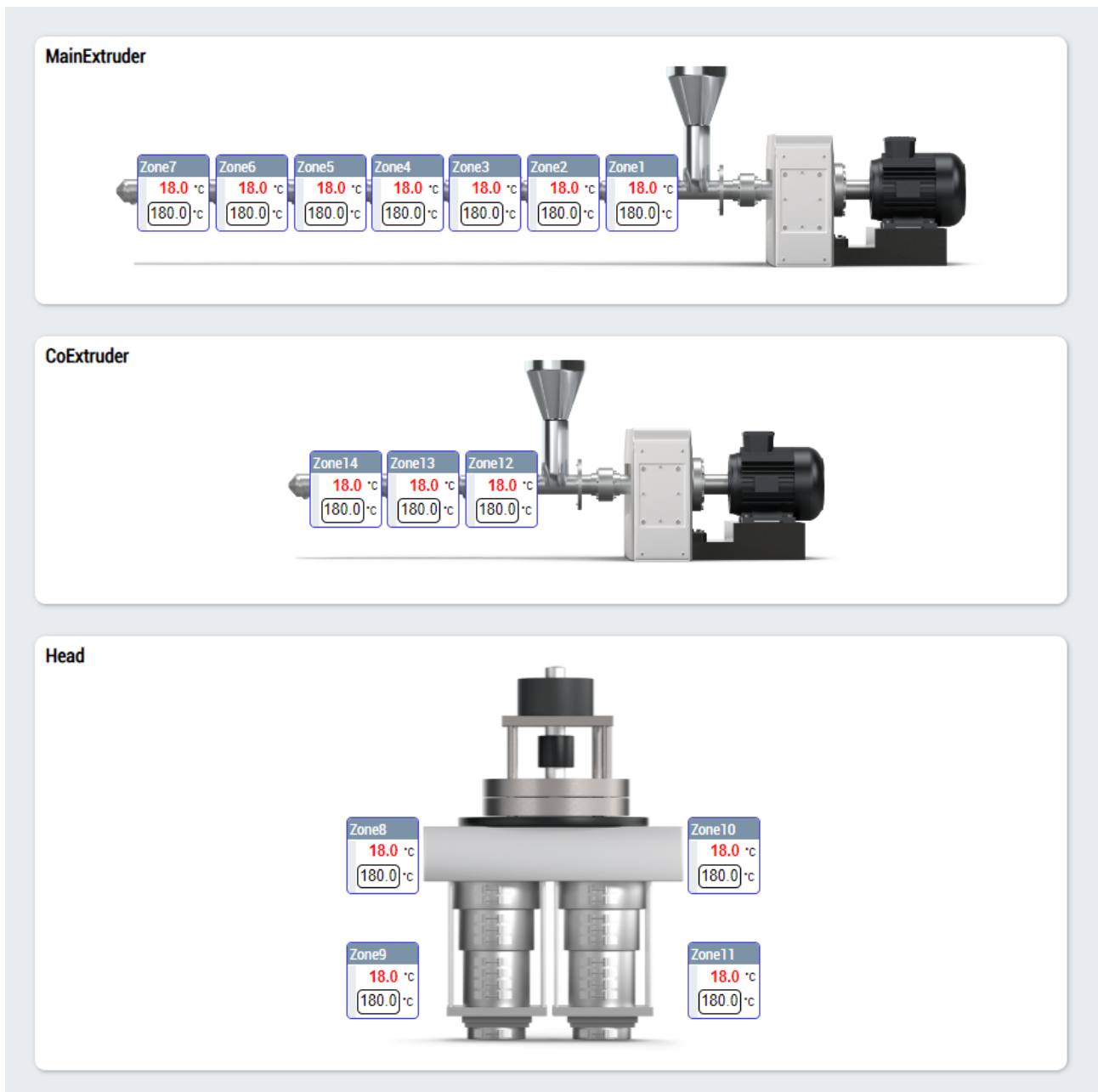
### 7.3.10.1 Übersicht

#### Temperature\_Overview.content

Diese Seite dient zur Temperaturzonenübersicht an der Maschine. Mithilfe des ZonedImageLayout-Controls im Anzeigemodus können die Konfigurationen von der [Layout \[▶ 191\]](#)-Seite auf dieser Seite angezeigt werden sowie die Temperaturen eingestellt werden. Durch einen Klick auf den Temperaturgruppennamen oben links in der Ecke der beiden Extruder ([MainExtruder \[▶ 167\]](#) und [CoExtruder \[▶ 168\]](#)) wird zu der zugehörigen Übersichtsseite navigiert.

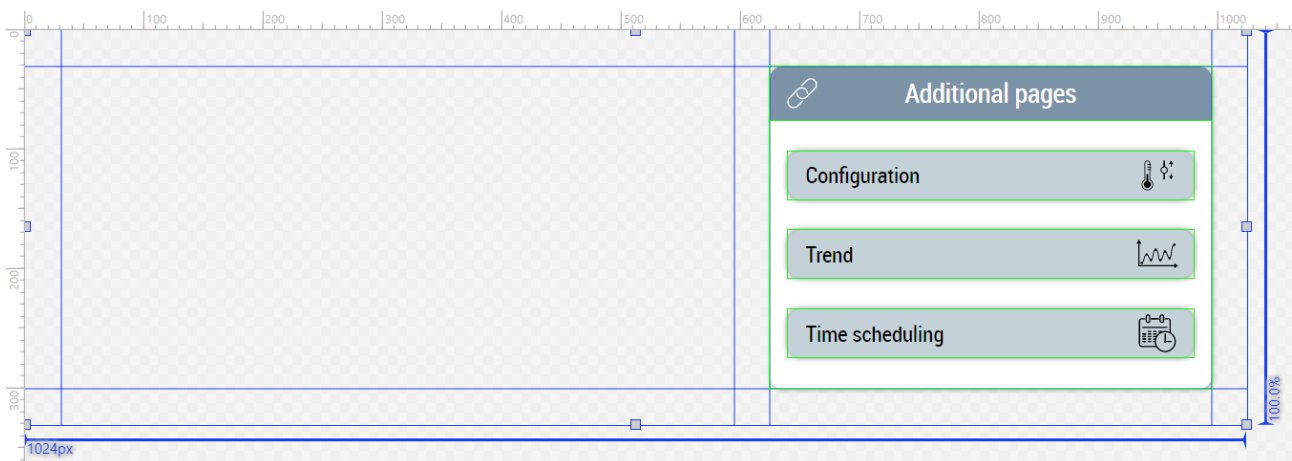






Temperature\_Overview\_Settings.content

Zu der Temperaturübersichtsseite gibt es noch eine ergänzende Seite im Sliderbereich, die zur Navigation auf weitere zugehörige Seiten dient.



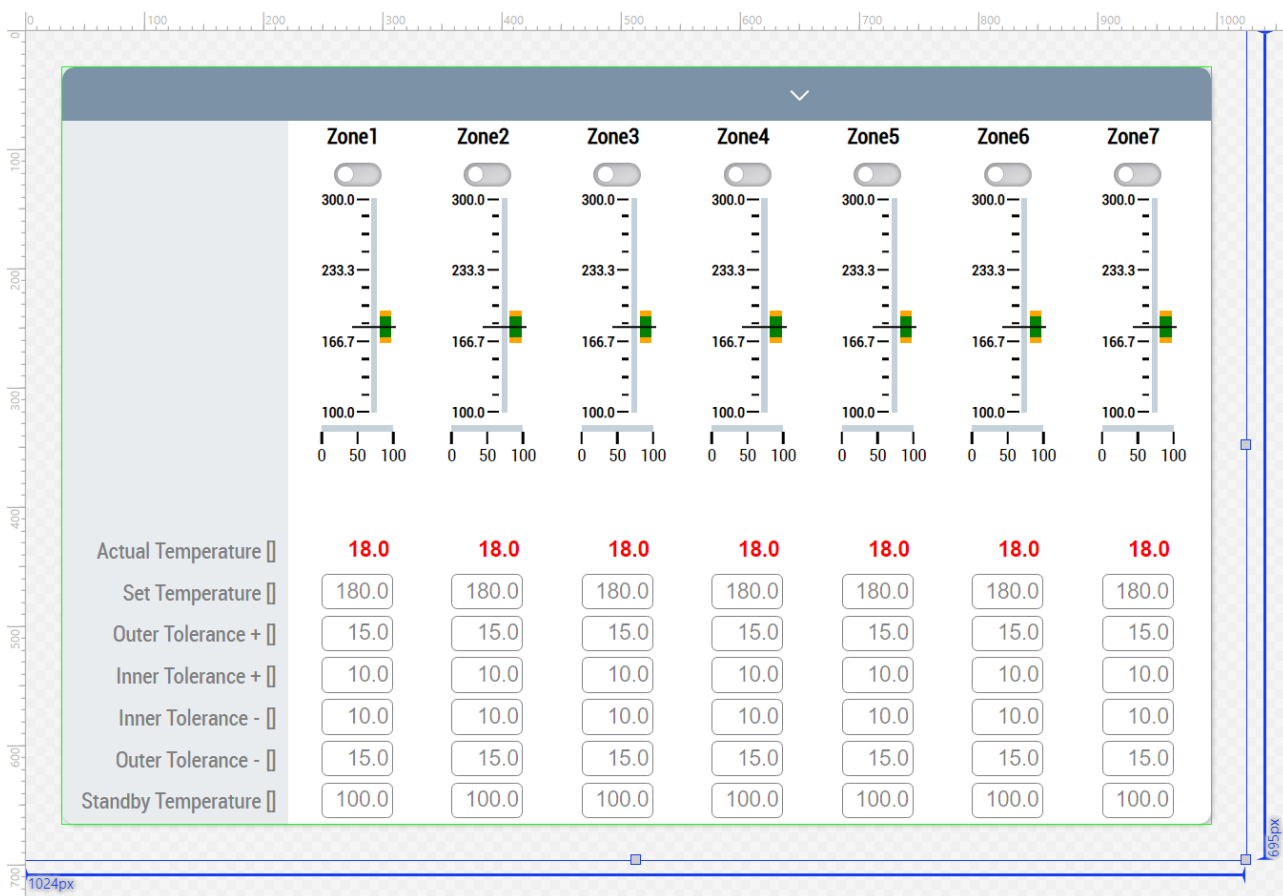


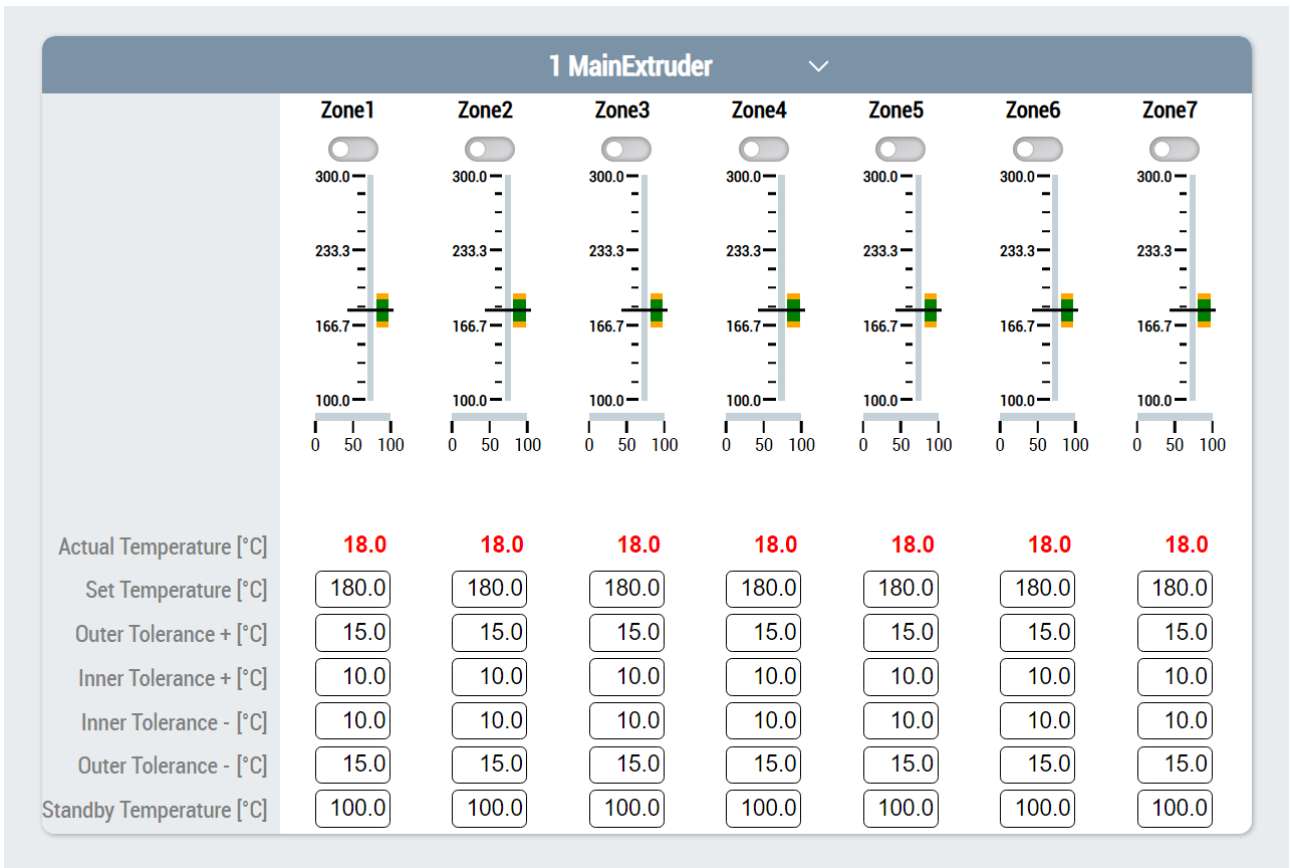
Verfügbar ab Version 12.6.0

### 7.3.10.2 Konfiguration

#### Temperature\_Configuration.content

Die Konfigurationsseite der Temperaturen enthält das ZoneConfiguration-Control und dient zur gruppierten Anzeige und Einstellung der Temperaturzonen. Für die Verwendung des Controls wird eine Instanz des [FB TemperatureHmi](#) [► 57] benötigt.

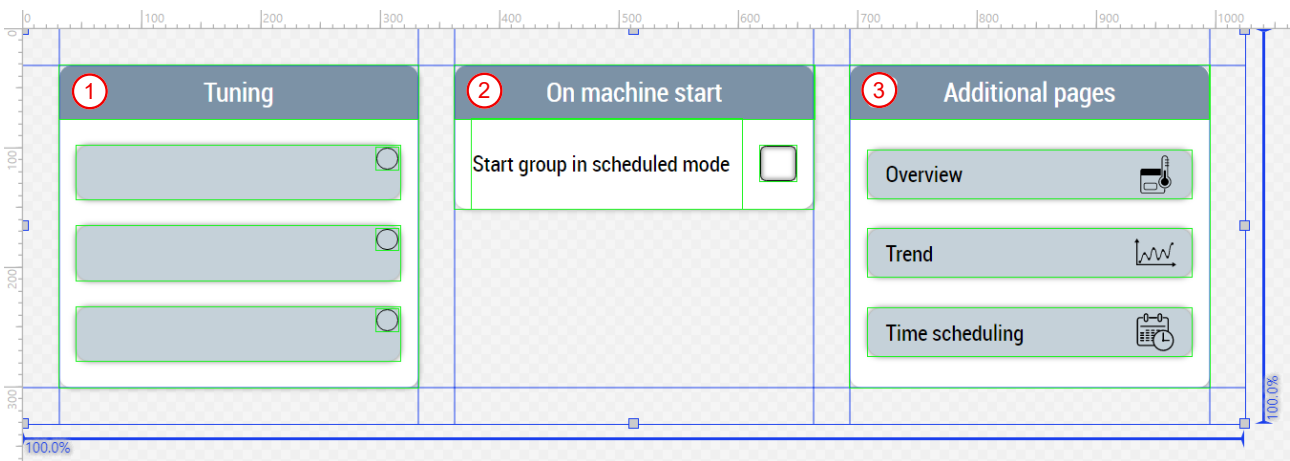


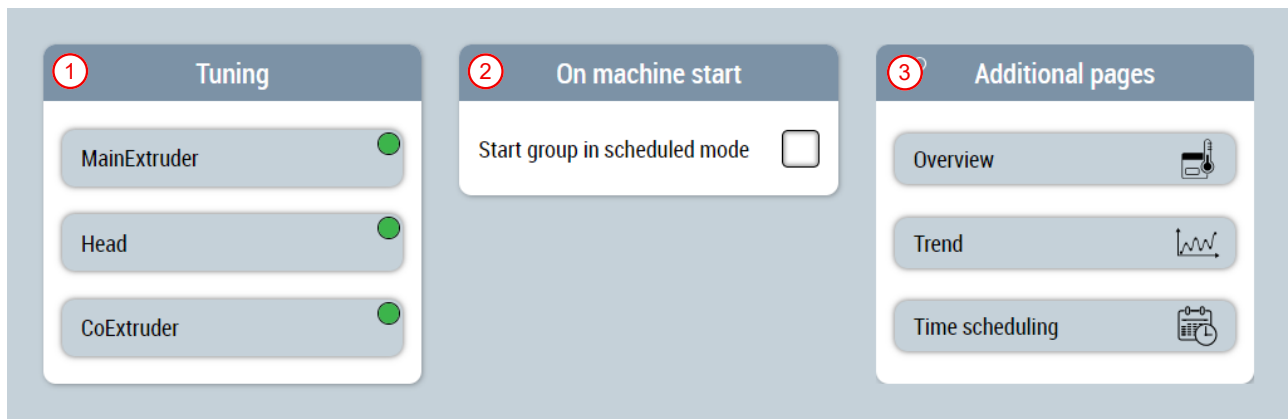


**Temperature\_Configuration\_Settings.content**

Diese Seite dient zur Ergänzung im Sliderbereich und enthält die folgenden Bestandteile:

1. Tuning der Temperaturzonengruppe und Zustandsanzeige des Tunings
2. Einstellungen die beim Maschinenstart wirken sollen
3. Bereich zur Navigation auf weitere zugehörige Seiten



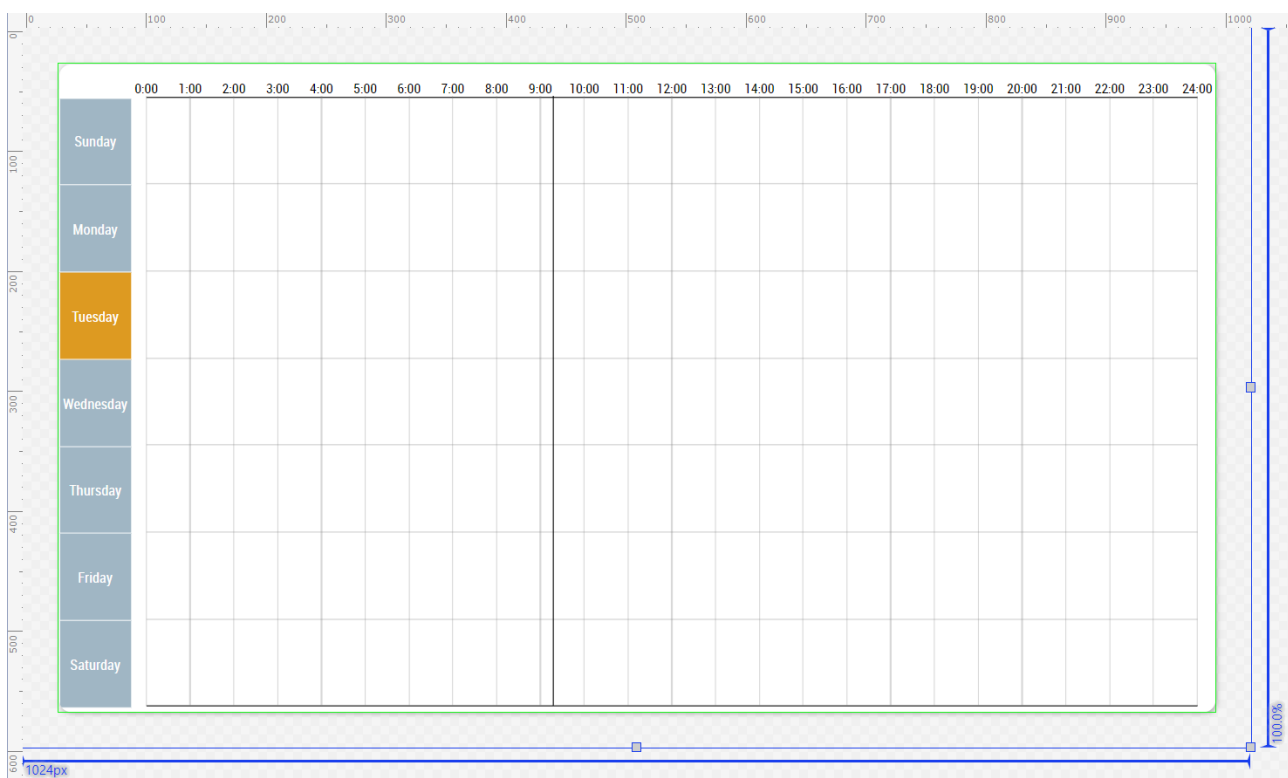


Verfügbar ab Version 12.6.0

### 7.3.10.3 Zeitsteuerung

#### Temperature\_TimeScheduling.content

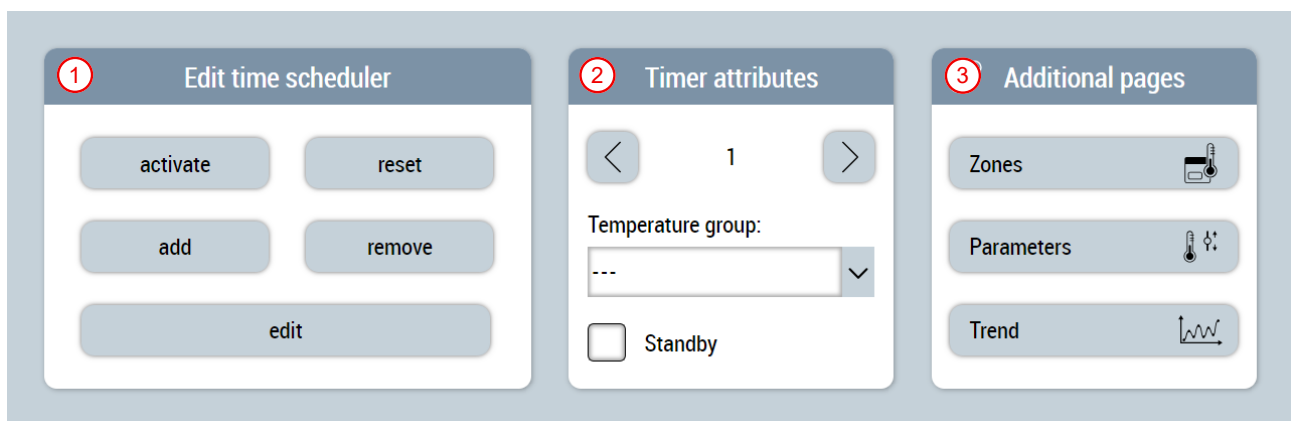
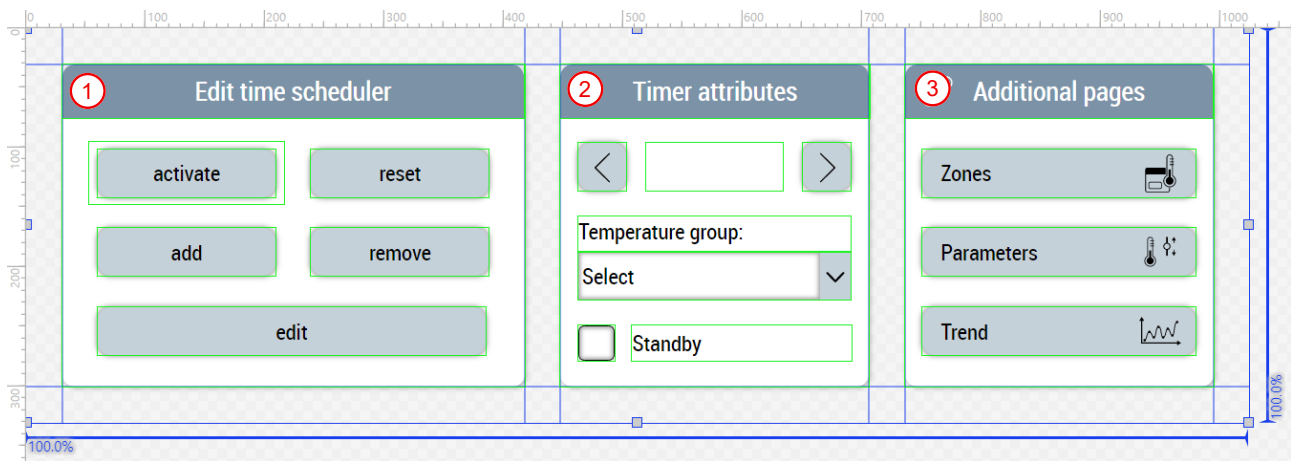
Die Seite zur Zeitsteuerung der Temperaturzonen und Temperaturgruppen enthält ein ProcessScheduler-Control. Das Control benötigt eine Instanz des [FB\\_TempScheduleHmi \[► 72\]](#) für das `TimerList`-Property der Kategorie `Scheduler`.



#### Temperature\_TimeScheduling\_Settings.content

Die Seite dient zur Ergänzung im Sliderbereich und beinhaltet die folgenden Funktionalitäten:

1. Aktivieren und resetten der Änderungen, hinzufügen von neuen Elementen, entfernen bzw. editieren des ausgewählten Elements
2. Auswahl des Timers über Pfeiltasten, Einstellung der Temperaturgruppe über die Combobox und Auswahl des Standby-Modus über die Checkbox
3. Bereich zur Navigation auf weitere zugehörige Seiten



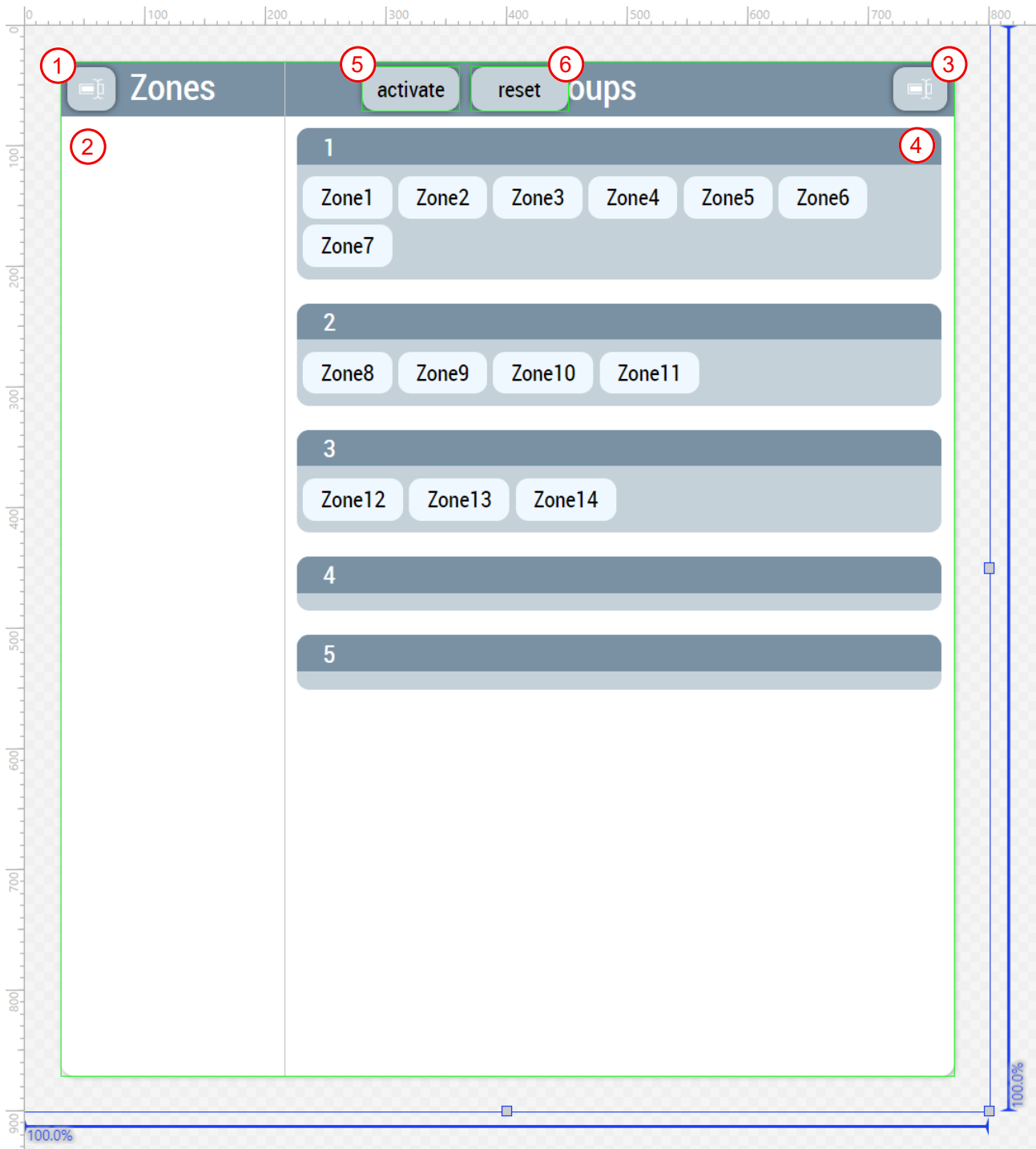
Verfügbar ab Version 12.6.0

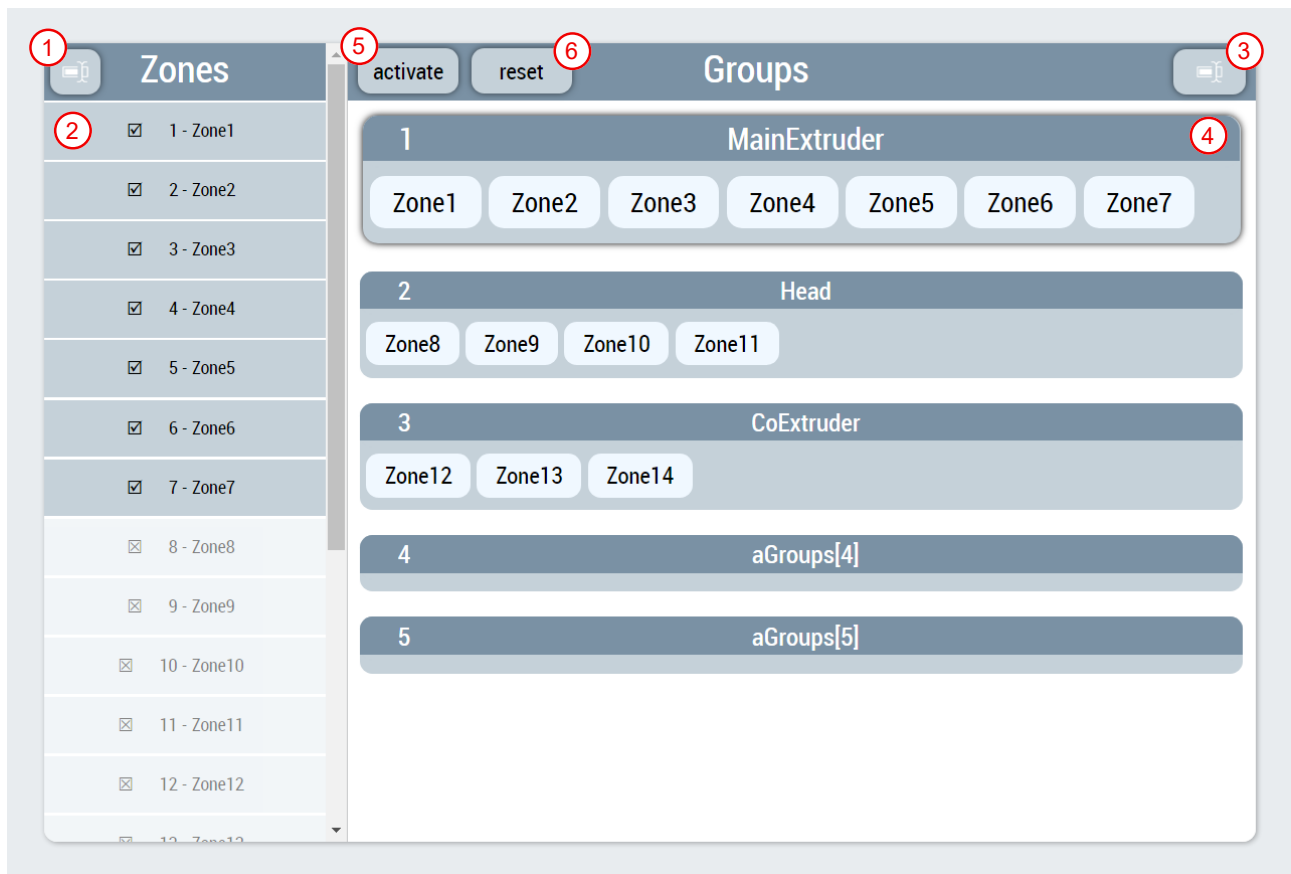
### 7.3.10.4 Grouping

#### Temperature\_Grouping.content

Diese Seite dient zur Gruppierung der Temperaturkanäle. Es werden nur die Temperaturkanäle in der Liste zur Verfügung gestellt, die über die [Temperaturparameterseite](#) [► 170] freigegeben sind. Die Seite besteht aus den folgenden Funktionen:

1. Anpassung der Temperaturzonennamen
2. Liste der zur Verfügung stehenden Temperaturkanäle und Anzeige der gruppierten Temperaturkanäle der ausgewählten Gruppe (4)
3. Anpassung der Temperaturgruppennamen
4. Anzeige der Temperaturgruppen mit den gruppierten Temperaturzonen und Auswahl der Temperaturgruppe durch Anklicken zur Anpassung über die Temperaturzonenliste (2)
5. Aktivierung der Änderung
6. Resetten der Änderungen





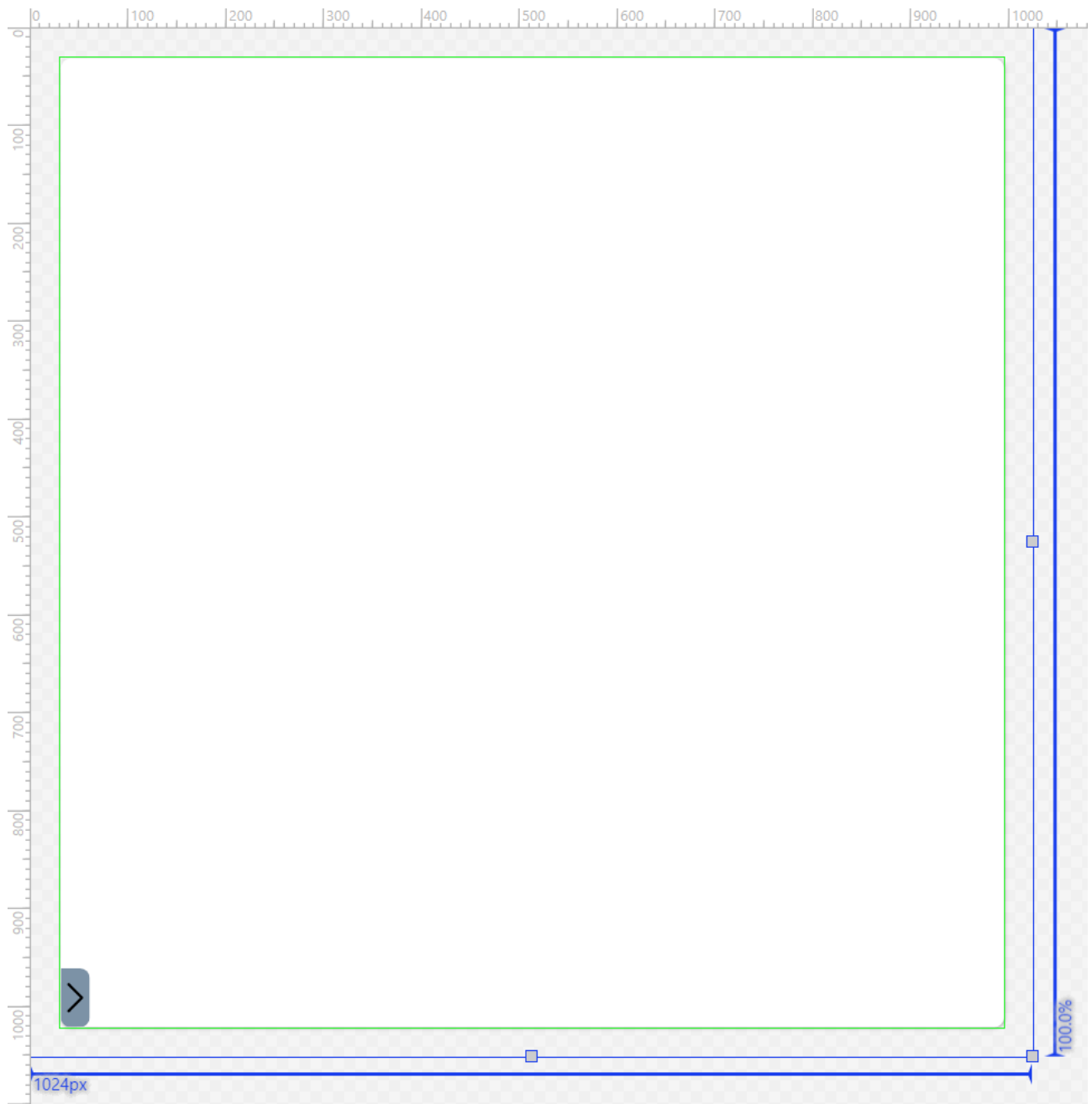
Verfügbar ab Version 12.6.0

### 7.3.10.5 Layout

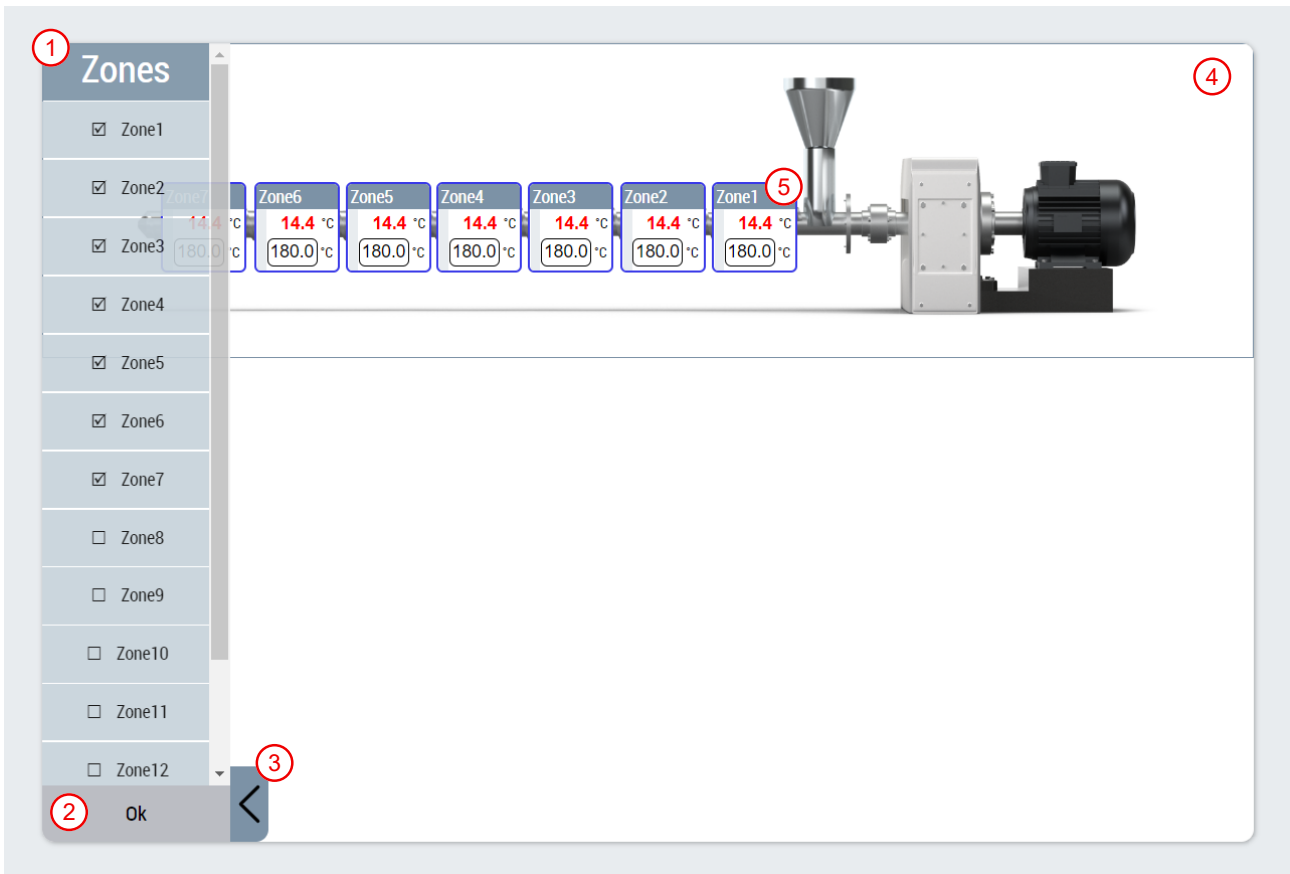
#### Temperature\_Layout.content

Diese Seite dient zur Erzeugung von Layouts, die im gesamten HMI wiederverwendet werden können. Dabei kann pro hinzugefügtes Layout ein Hintergrundbild gewählt und angepasst werden sowie Temperaturzonen beliebig ausgewählt und auf dem Bild positioniert werden. Es werden nur die Temperaturzonen in der Liste zur Verfügung gestellt, die über die [Temperaturgruppierungsseite](#) [► 189] einer Gruppe zugeordnet sind. Die Seite besteht aus den folgenden Funktionen:

1. Liste der zur Verfügung stehenden gruppierten Temperaturzonen
2. Aktivierung der Änderungen in der Liste mit dem Button
3. Schließen der Temperaturzonenliste durch die Pfeiltaste
4. Anzeige des über die Combobox im Sliderbereich ausgewählten Layouts
5. Positionierte Darstellung ausgewählter Temperaturzonen



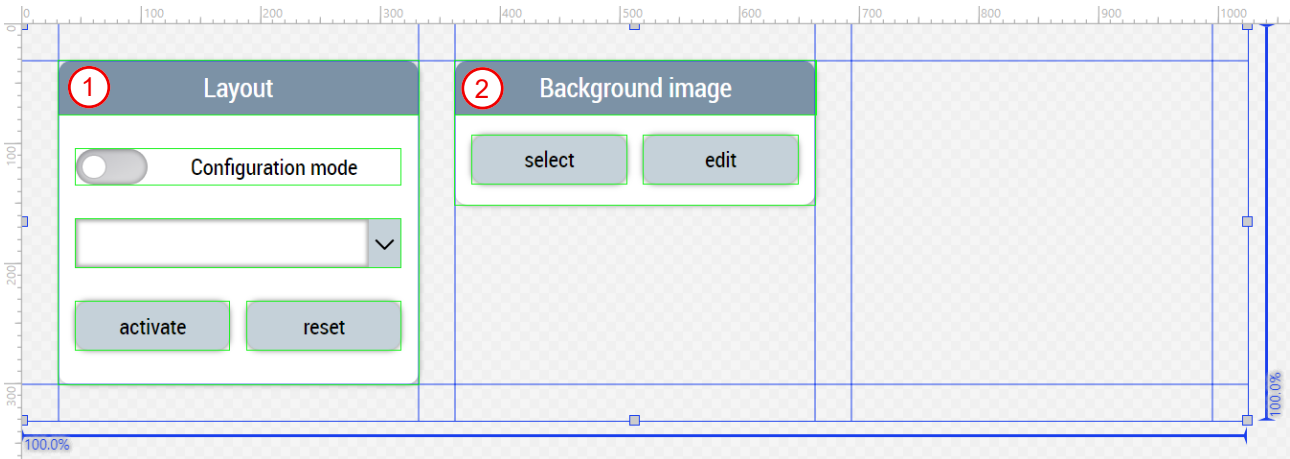


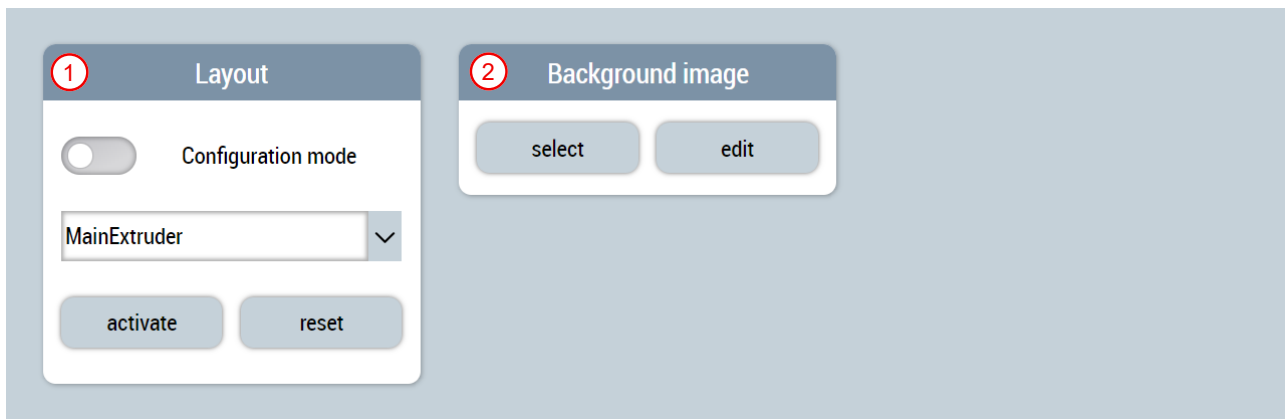


**Temperature\_Layout\_Settings.content**

Die Seite dient zur Ergänzung im Sliderbereich und beinhaltet die folgenden Funktionalitäten:

1. Einschalten des Konfigurationsmodus, Auswahl des Layouts über eine Combobox, aktivieren und resetten der Änderungen
2. Auswahl des Hintergrundbildes und editieren von Hintergrundbild-Eigenschaften über separate Pop-up-Fenster





Verfügbar ab Version 12.6.0

## 7.3.11 WDS

Die Ordner mit dem Namen *Wtc* (*Contents/Navigation/Wtc* und *Contents/Slider/Wtc*) beinhalten alle Contents, die für die Wanddickensteuerung (WDS) der Maschine benötigt werden.

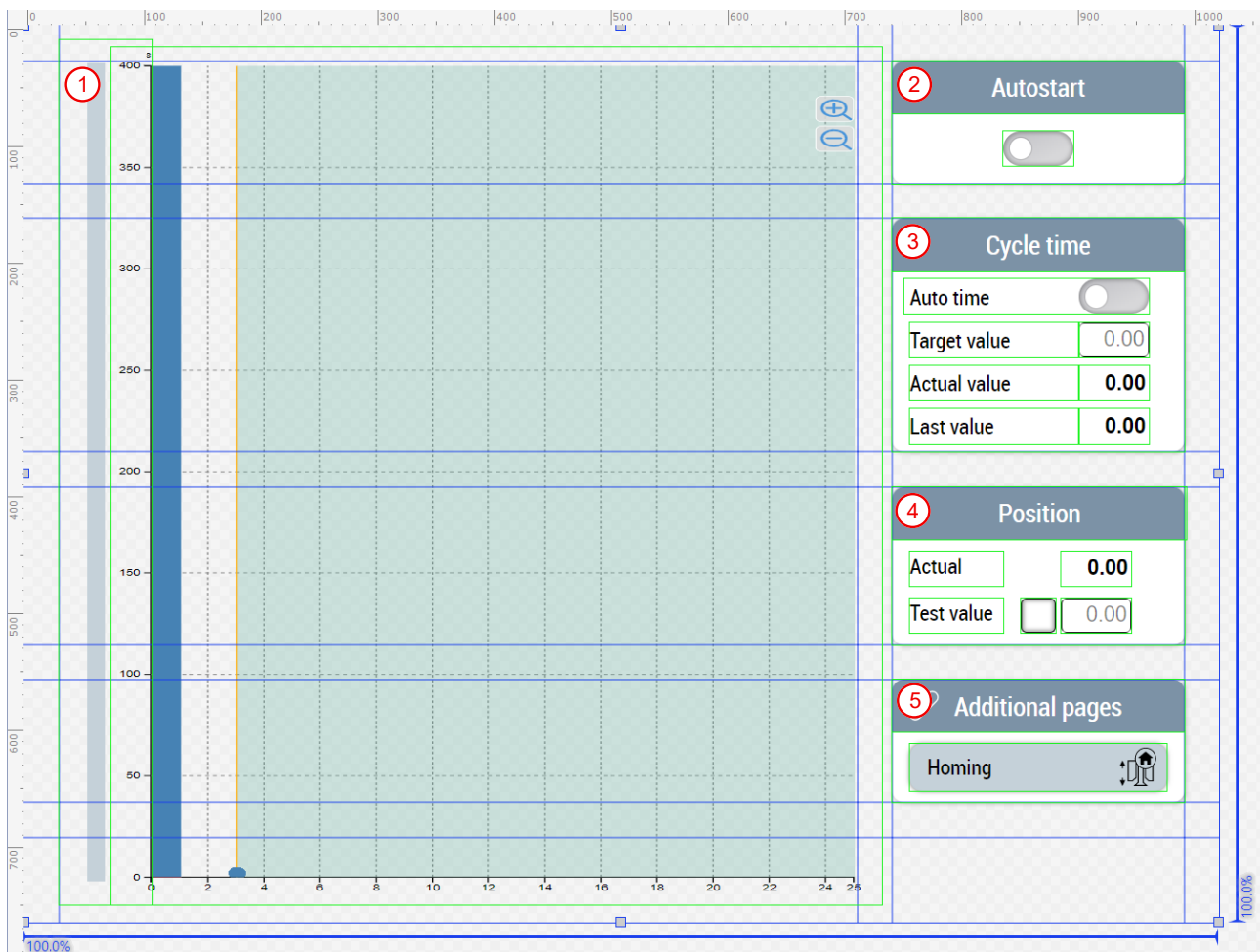
Content	Beschreibung
<a href="#">Wtc.content [► 194]</a>	Die Hauptseite der WDS.
<a href="#">Wtc_Settings.content [► 194]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Wtc.content</i> im Hauptbereich befindet.
<a href="#">Wtc_Homing.content [► 197]</a>	Die Referenzfahrtseite der WDS.
<a href="#">Wtc_Homing_Settings.content [► 197]</a>	Wird im Sliderbereich unter dem 4. Tab angezeigt, wenn sich <i>Wtc_Homing.content</i> im Hauptbereich befindet.

### 7.3.11.1 Kurve

#### Wtc.content

Der Hauptcontent der WDS besteht aus den folgenden Bestandteilen:

1. WDS-Kurve mithilfe des CurveEditor-Controls
2. Funktion für Autostart
3. Bereich für Einstellungen und Anzeigen in Bezug auf die Zykluszeit
4. Bereich zur Anzeige der aktuellen Position und Einstellung einer Testposition
5. Bereich zur Navigation auf eine zugehörige Seite

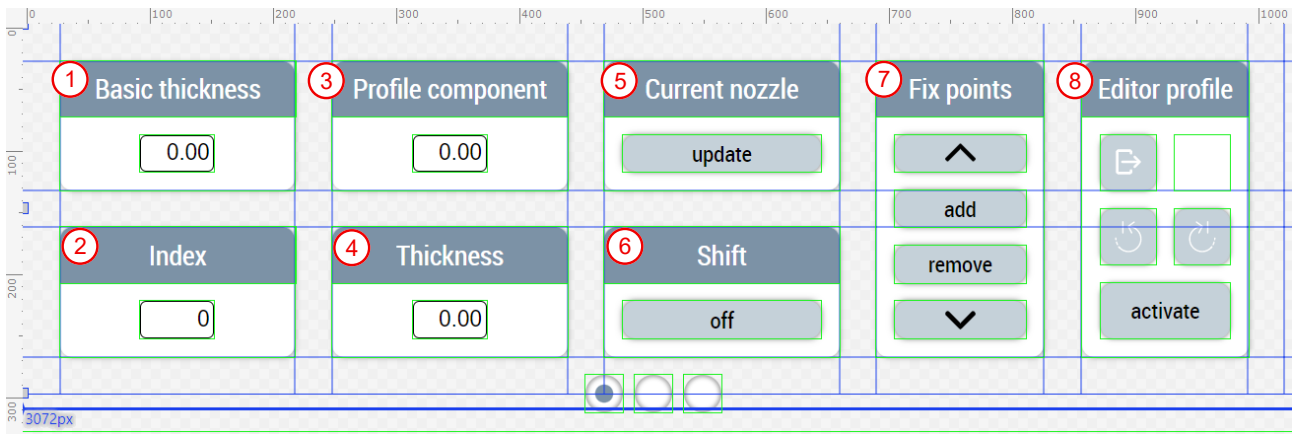


**Wtc\_Settings.content**

Zu der Hauptseite der WDS gibt es noch eine ergänzende Seite für den Sliderbereich. Dieser Content ist in drei Seiten aufgeteilt.

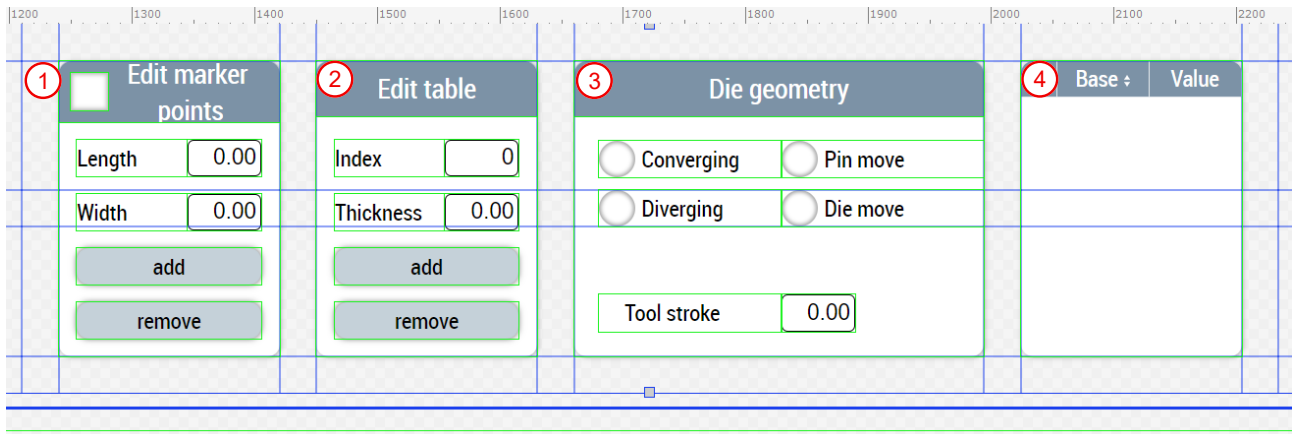
**Die erste Seite enthält die folgenden Bestandteile:**

1. Einstellung bzw. Anzeige der Basisdicke des gesamten Profils
2. Einstellung bzw. Anzeige des Index des ausgewählten Punkts
3. Einstellung bzw. Anzeige der Profilkomponente
4. Einstellung bzw. Anzeige der Dicke des ausgewählten Punkts
5. Updatemöglichkeit der Einstellung entsprechend der aktuell ausgewählten Düse
6. Verschiebefunktion ein- und ausschalten
7. Navigieren zwischen den existierenden Fixpunkten sowie Hinzufügen und Entfernen von Fixpunkten
8. Änderungen rückgängig machen, navigieren zwischen aktivierten Profilen und Aktivierung des angezeigten Profils



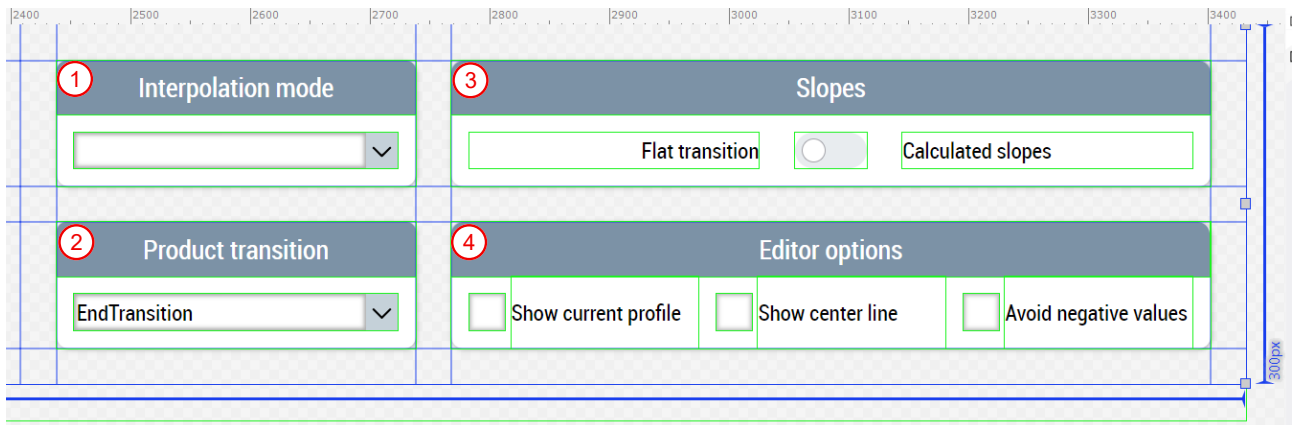
**Die zweite Seite enthält die folgenden Bestandteile:**

1. Hinzufügen, Entfernen und Bearbeiten von Markerpunkten sowie Auswahl der Verwendung
2. Hinzufügen, Entfernen und Bearbeiten von Fixpunkten
3. Anpassung der Düsengeometrie durch Umschalten zwischen Konvergent und Divergent, Umschalten zwischen Pinbewegung und Düsenbewegung und Einstellung des Werkzeughubs
4. Tabelle zur Anzeige der Fixpunkte



**Die dritte Seite enthält die folgenden Bestandteile:**

1. Einstellung bzw. Anzeige des Interpolationsmodus
2. Einstellung bzw. Anzeige des Produktübergangs
3. Anpassung der Steigungen
4. Anpassung der Editoroptionen



Verfügbar ab Version 12.5.1

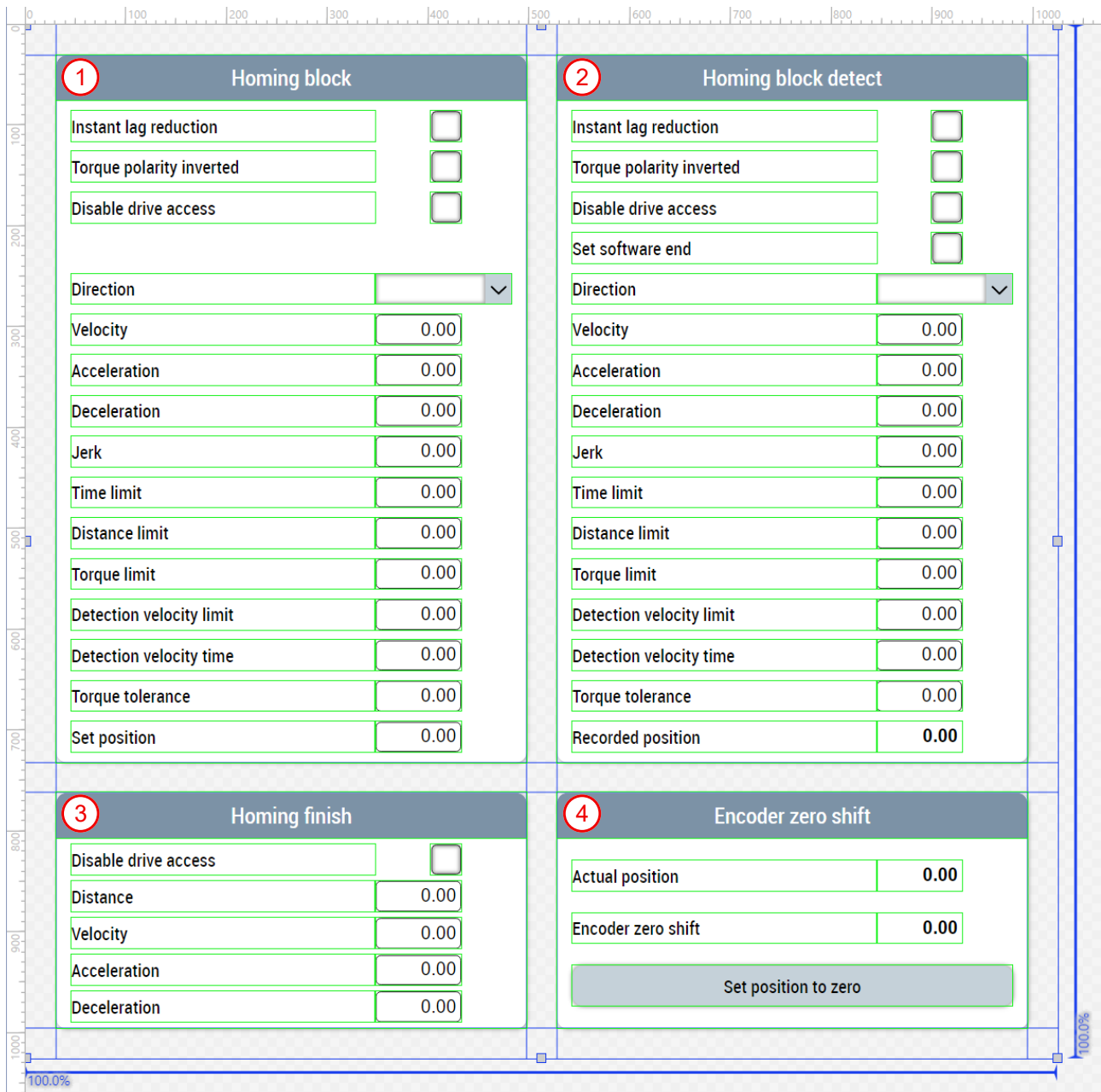
### 7.3.11.2 Referenzfahrt

#### Wtc\_Homing.content

Die Referenzfahrtseite für die WDS besteht aus den folgenden Bestandteilen:

1. Referenzfahrtblock
2. Referenzfahrblockerkennung
3. Referenzfahrtfinish
4. Encoder Nullpunktverschiebung

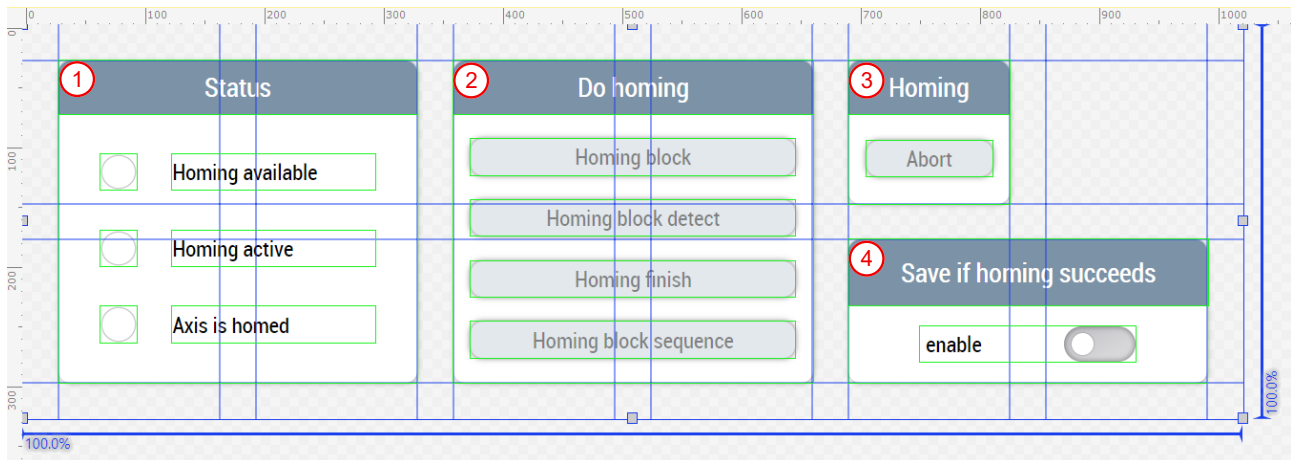
Mithilfe der Funktion `UpdateAxesHomingUnits` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket werden bestimmte Einheiten der Werte in Winkeleinheitengruppen umgewandelt, wenn es sich um eine transformierte Achse handelt.



#### Wtc\_Homing\_Settings.content

Zu der Referenzfahrseite der WDS gibt es noch eine ergänzende Seite für den Sliderbereich. Diese Seite enthält die folgenden Bestandteile:

1. Referenzfahrt-Status-Anzeige
2. Referenzfahrt-Funktionen
3. Abbruch der Referenzfahrt
4. Aktivierung der Speichermöglichkeit



Verfügbar ab Version 12.6.0

## 7.4 Localization

Im Ordner *Localization* werden die verschiedenen Sprachen verwaltet, die im Projekt zur Verfügung stehen sollen. Es können neue Sprachen hinzugefügt werden. Außerdem können neue *Localized Symbols* angelegt werden, für die dann in allen benötigten Sprachen die Übersetzung eingepflegt werden kann.

Durch die Installation des `Beckhoff.TwinCAT.HMI.Plastic.Localizations` NuGet-Pakets stehen dem Plastic Application HMI-Projekt Sprachschlüssel zur Verfügung, die im Projekt verwendet werden können.



### Updatefähigkeit der Plastic Application HMI

Das Editieren der Plastic Application HMI kann zu einer Einschränkung der Updatefähigkeit führen. Dadurch kann die Unterstützung durch Beckhoff Automation beeinträchtigt werden.



Verfügbar ab Version 12.5.1

## 7.5 View

Eine View ist eine Datei mit der Endung `.view` und beinhaltet den Gesamtaufbau der Bedienoberfläche. Ein Projekt kann mehrere Views beinhalten. Zu Beginn muss im Engineering festgelegt werden, welche View als Start View verwendet werden soll, um das HMI aufzubauen. Der Aufbau der Bedienoberfläche wird über die als Start View ausgewählte View definiert.



### Updatefähigkeit der Plastic Application HMI

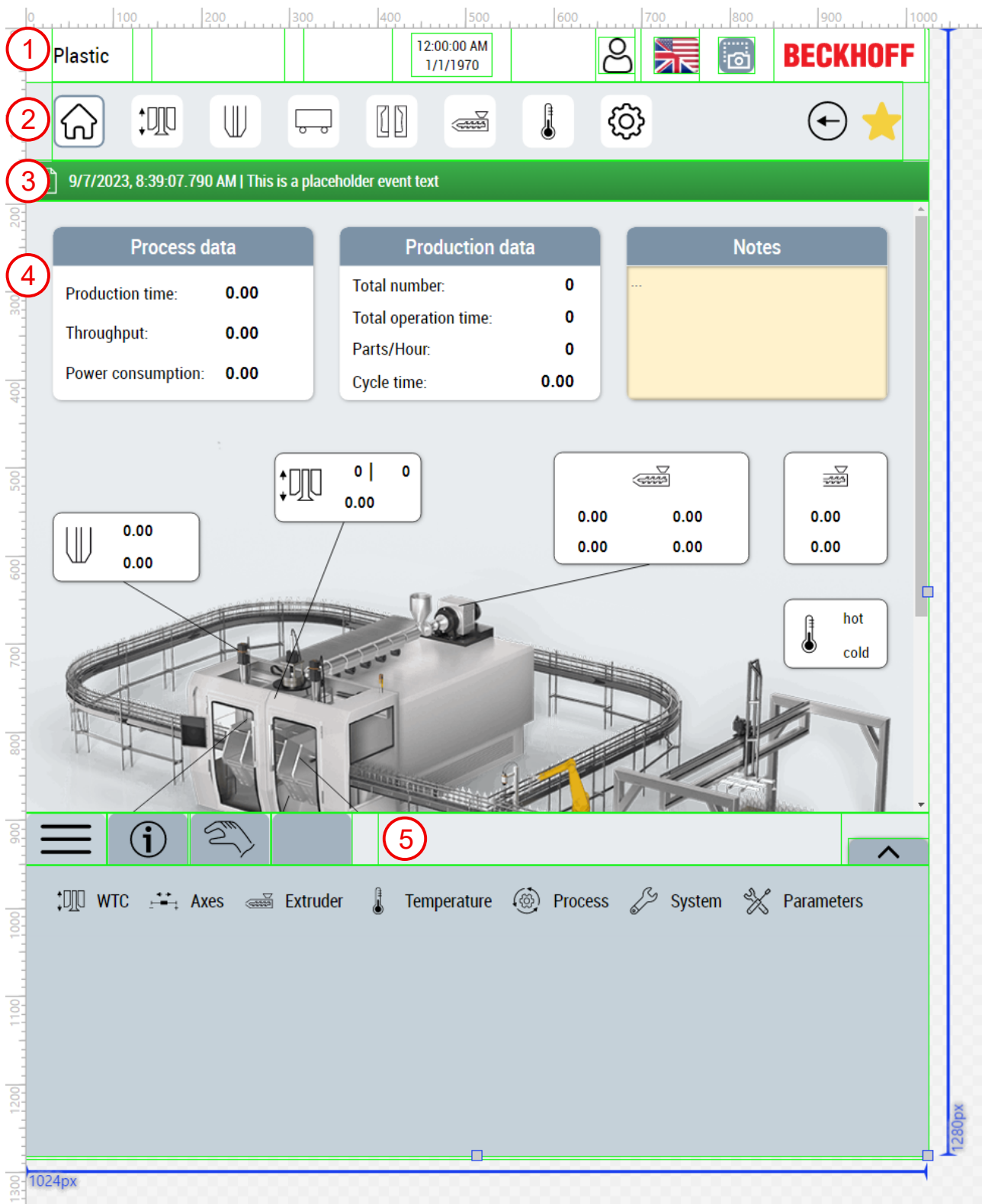
Das Editieren der Plastic Application HMI kann zu einer Einschränkung der Updatefähigkeit führen. Dadurch kann die Unterstützung durch Beckhoff Automation beeinträchtigt werden.

#### Portrait.view

Sofern die Datei `Portrait.view` nicht modifiziert ist, handelt es sich dabei um eine Bedienoberfläche im Hochformat (4:5) mit einer Breite von 1024 px und einer Höhe von 1280 px. Sie besteht aus den folgenden Bestandteilen:

**1. Kopfzeile:**

- Applikationsname (Draufklicken zeigt im Hauptbereich den `Home.content` an)
  - Aktuell geladenes Rezept (Draufklicken zeigt im Hauptbereich den `RecipeManagement.content` an)
  - Aktuelle Betriebsart
  - Uhrzeit und Datum
  - Benutzername (Draufklicken öffnet Untermenü)
  - Flagge zur Anzeige der ausgewählten Sprache (Draufklicken öffnet Untermenü)
  - Screenshot-Funktion (wird mithilfe der Funktion `TakeScreenShot` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket ermöglicht)
  - Logo
2. **Favoritenbereich:** Beinhaltet benutzerspezifisch die gewünschten Favoriten-Buttons zur schnellen Navigation für häufig genutzte Seiten.
3. **Event-Zeile:** Anzeige des neuesten anstehenden Alarms und ein Draufklicken zeigt im Hauptbereich den `Alarms.content` an.
4. **Hauptbereich:** Anzeige des ausgewählten Contents.
5. **Sliderbereich:** Mithilfe der Funktionen `ToggleSliderArea`, `ShowSliderArea` und `HideSliderArea` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket kann der Sliderbereich über die Pfeiltasten ein- und ausgeblendet werden, über einen Klick auf einen Tab eingeblendet oder bei gewünschten Contents im Hauptbereich (z. B. `Scope.content`) ausgeblendet werden. Durch einen Klick auf einen der folgenden Tabs wird der entsprechende Content im Sliderbereich angezeigt:
- **1. Tab:** `Navigation.content` beinhaltet im Vergleich zum Favoritenbereich alle zur Verfügung stehenden Contents des Projekts.
  - **2. Tab:** `Info.content` zeigt die wichtigsten Maschinendaten an.
  - **3. Tab:** `ManualFunctions.content` beinhaltet die Handfunktionen der Maschine.
  - **4. Tab:** Mithilfe der Funktion `UpdateSliderContentRegion` aus dem `Beckhoff.TwinCAT.HMI.Plastic.Functions` NuGet-Paket kann der 4.Tab optional eingeblendet werden und das Icon sowie der angezeigte Content im Sliderbereich werden je nach angezeigtem Content im Hauptbereich aktualisiert. Mögliche Contents lassen sich anhand der Namensendung `_Settings.content` erkennen.



### Configurator-Control

Das Configurator-Control dient als allgemeines Control zur Einstellung von Control-Übergreifenden Konfigurationen. Die Instanz des Controls muss lediglich existieren und benötigt keine Sichtbarkeit. Die folgenden Attribute werden zur weiteren Einstellung des HMIs benötigt:

- **UnitConfig:** Pfad zur JSON-Datei mit dem entsprechenden Schema für die Einheitenumschaltung. Die Datei wird standardmäßig mit dem Beckhoff.TwinCAT.HMI.Plastic.Controls NuGet-Paket mitgeliefert, sie kann jedoch auch kopiert und modifiziert werden, so dass ein anderer Pfad eingestellt werden muss.



- **NavigationConfig:** Einstellung des Navigationsaufbaus mit zusätzlichen Information wie Icons, Slider-Content etc.
- 



Verfügbar ab Version 12.6.0

---

## 8 Anhang

### 8.1 Temperaturregelung in Betrieb nehmen

Die Inbetriebnahme der Temperaturregelung umfasst sowohl einen TwinCAT Engineering als auch einen zur Laufzeit durchzuführenden Teil. In diesem Beispiel werden die einzelnen Schritte, die zur Laufzeit durchgeführt werden, zusammengefasst.

Vor den folgenden Schritten ist die Vorbereitung des TwinCAT Projektes nach dem Vorbild aus Abschnitt [Mapping und Konfiguration von Temperaturzonen \[► 142\]](#) durchzuführen.

**Stellen Sie für alle ihre Temperaturzonen die passenden Ein- und Ausgangssignalarten und -geräte ein**

---

#### ● Hardware-Parametrierung ohne Gruppierung

**i** Dieser Schritt bezieht sich auf das lineare Mapping der TF8540 Bibliothek. Bereits konfigurierte Gruppierungen werden in diesem Schritt ignoriert.

---

1. Loggen Sie sich mit Administrator-Rechten ein (Default User = 4. Administrator, Passwort = 4).
2. Navigieren Sie über die Navigation (≡-Symbol) zu *Parameter > Temperatur*.
3. Stellen Sie bei den mit (1) markierten Parameter ihre Hardware-Konfiguration für die erste gemappte Zone ein.
4. Setzen Sie den Haken des Parameters **In Use**, um die Verwendung der Zone zu validieren.
5. Wiederholen Sie Schritt 3 und 4 für alle zu verwendenden Zonen über die mit (2) markierte Schaltfläche.
6. Speichern Sie ihre Einstellungen über die mit (3) markierte Schaltfläche.

Row	Last tuning: 2023-01-23T18:31:38Z	Zones <input checked="" type="checkbox"/>	Temperature supply	Unit
1	Save zone name		<input type="checkbox"/>	
2	Module ID		<input type="text" value="1"/>	
3	Zone ID		<input type="text" value="1"/>	
4	Supply ID		<input type="text" value="1"/>	
5	Extruder ID		<input type="text" value="1"/>	
6	Select cooling output		NoSignal	▼
7	Select heating output		PWM	▼
8	Temperature sensor terminal		NoTerminal	▼
9	Sensor type		KL_RangeHigh	^
10	Terminal channel		EL331x	
11	In use		EL316x	
12	enable		EL312x	
13	Use cooling		EL331x_0010	▼
14	Tune cooling		<input type="checkbox"/>	
15	Forced cooling enabled		<input type="checkbox"/>	

Page 1|6

Zone 1|20 < >

Parameter > Temperatur

**Machine data information**

Creation date: 2022-10-13T08:08:31Z

Store count: 3

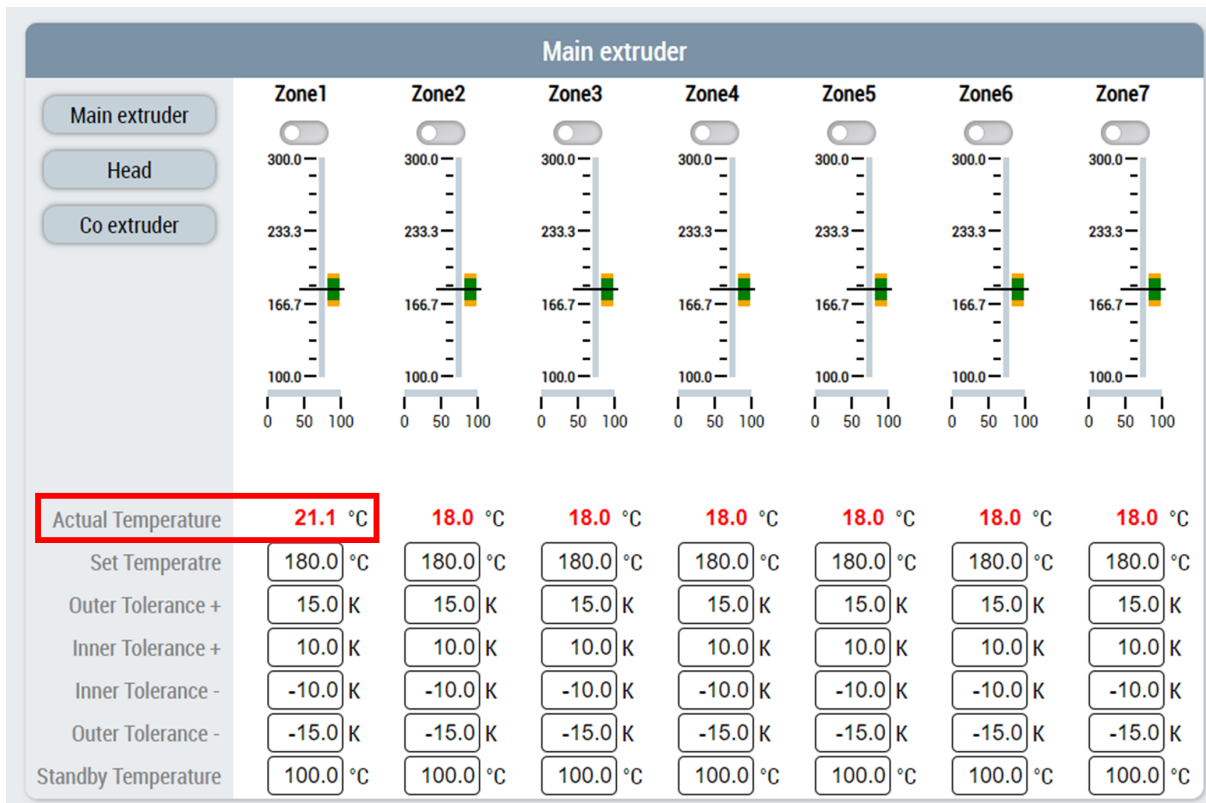
Stored date: 2023-02-01T13:23:27Z

**Machine data management**

Load Save

**Überprüfen Sie die Reaktion der Hardwareeingänge an der Maschine**

1. Navigieren Sie über die Navigation zu *Temperatur > Parameter*.
2. Erwärmen Sie den Sensor einer Zone über eine externe Wärmequelle.
3. Beobachten Sie ob die Temperaturänderung in der zu erwartenden Zone auftritt.
4. Wiederholen Sie Schritt 2 und 3 für jede Zone.



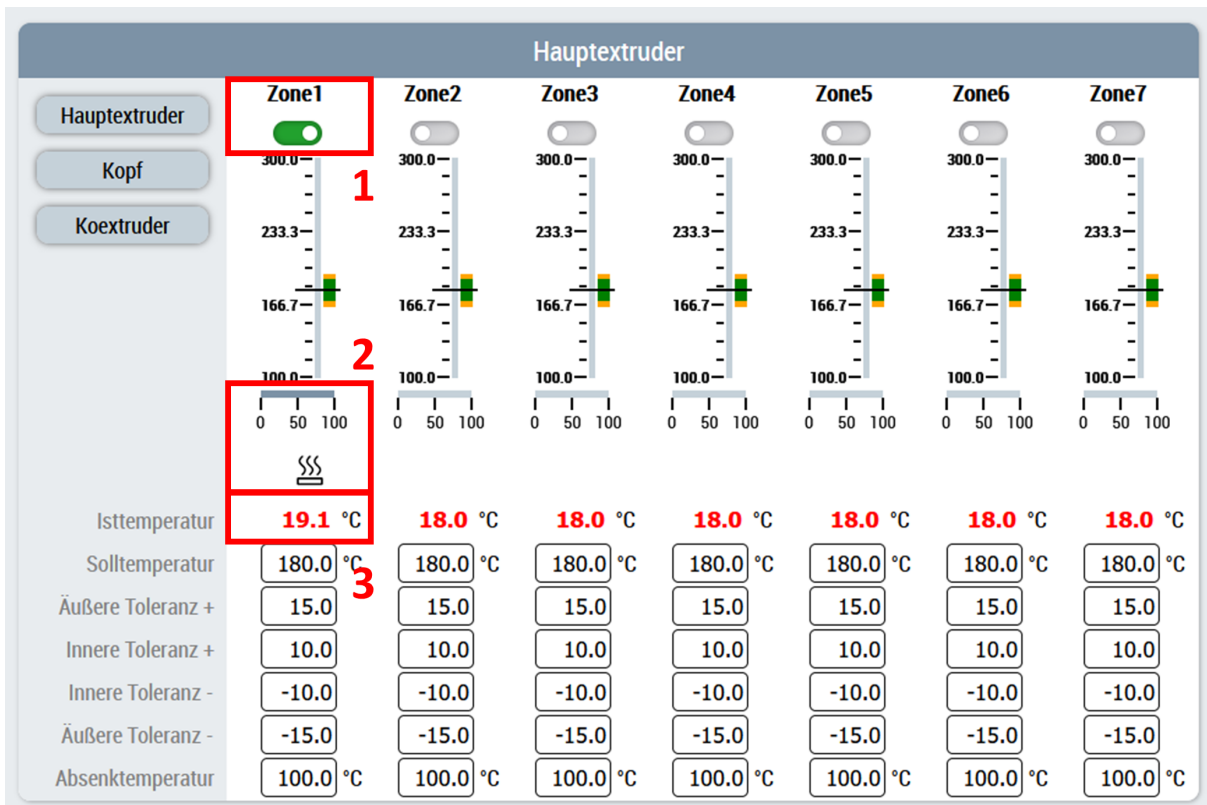
### Überprüfen Sie die Reaktion der Hardwareausgänge an der Maschine

#### ● Einschalten einer Zone erzeugt kein Powerlevel

**i** Stellen Sie sicher das zu dem Zeitpunkt dieses Schrittes die Temperaturregelung von der PLC freigegeben wurde!

- Zur Freigabe aller Zonen führen Sie die Methode `FB_TempCtrl.EnableAll(...)` [► 54] aus
- Zur Freigabe genau einer Gruppe führen Sie die Methode `FB_TempCtrl.Groups[...].Enable(...)` [► 59] aus

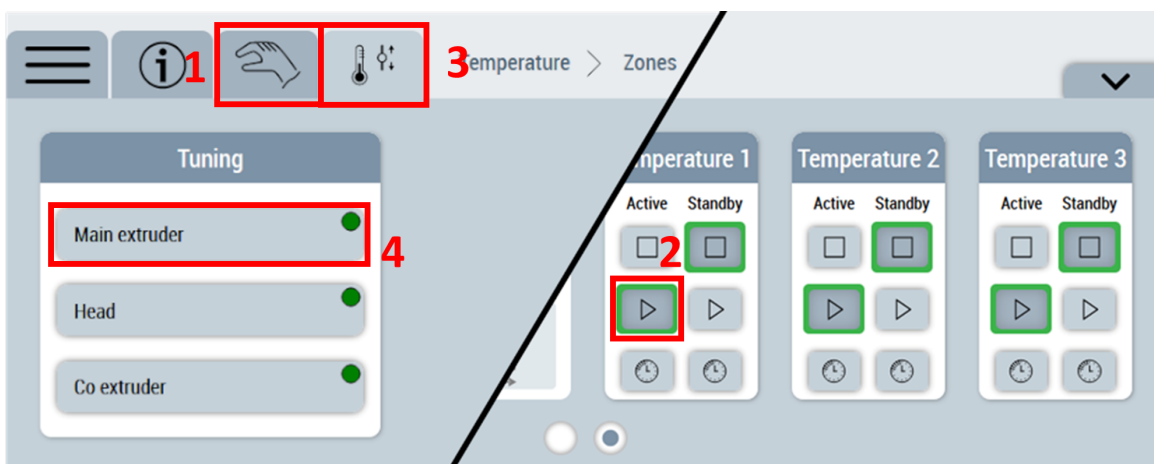
1. Schalten Sie eine einzelne Temperaturzone über den Slider (1) ein.
2. Überprüfen Sie ob das Powerlevel (2) ansteigt und der Istwert (3) der eingeschalteten Zone sich ändert.
3. Schalten Sie die Zone möglichst zeitnah wieder aus, um den Temperaturanstieg minimal zu halten.
4. Wiederholen Sie die Schritte 1 bis 3 für jede Zone.



**Starten Sie das automatische Tuning der Regelparameter**

Sie sollten sich weiterhin auf der *Temperatur > Parameter* Seite befinden

1. Wählen Sie im Slider Bereich den dritten Tab (1) aus.
2. Starten Sie die Temperaturregelung einer Gruppe (2) die getuned werden soll.
3. Wechseln Sie auf den vierten Tab (3) und starten Sie das Tuning über die entsprechende Schaltfläche (4) der Gruppe.
4. Führen Sie Schritt 1 bis 3 für alle in Betrieb zu nehmenden Gruppen aus.



**Überwachen Sie das automatische Tuning, bis es erfolgreich abgeschlossen ist**

1. Sobald das Anzeigeelement auf der Schaltfläche nicht mehr gelb ist, ist das Tuning beendet.
2. Wenn das Anzeigeelement auf der Schaltfläche grün ist, wurde das Tuning erfolgreich abgeschlossen.

Sie haben Ihre Temperaturregelung erfolgreich in Betrieb genommen.

## 8.2 Erzeugung und Nutzung des ZoneImageLayoutConfig-Serversymbols

Zur Speicherung und Wiederverwendung [▶ 184] von Layouts, die auf der Layout [▶ 191]-Seite der Temperaturen erzeugt werden, wird pro Layout eine Instanz des ZoneImageLayoutConfig-Serversymbols benötigt.

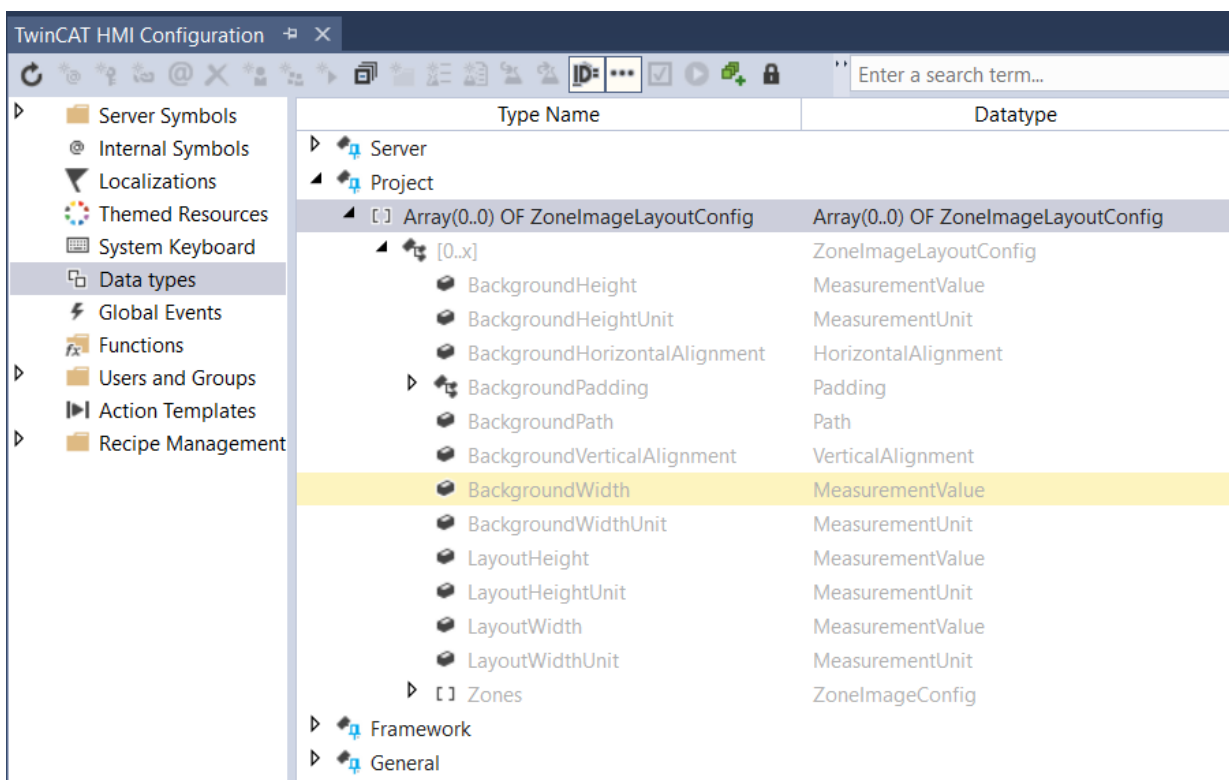
### Erzeugung des Serversymbols



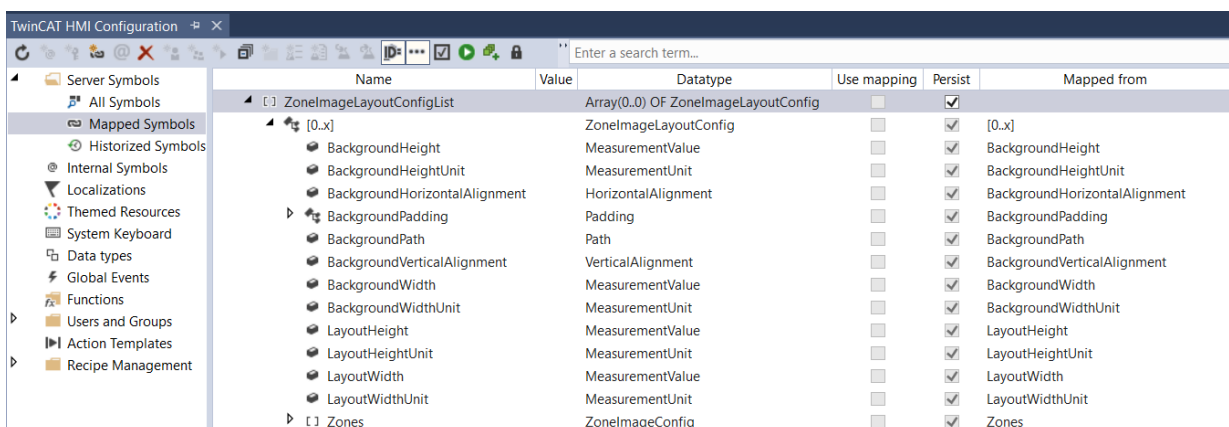
Je benötigtes Layout wird eine einzelne Stelle im ZoneImageLayoutConfigList-Array benötigt

Dafür empfiehlt es sich, ein dynamisches Array von diesem Typ anzulegen. Dies erfolgt im TwinCAT HMI Konfigurationsfenster:

#### 1. Erzeugung des Datentyps



#### 2. Anlegen eines Serversymbols unter der Kategorie TcHmiSrv vom entsprechenden Datentyp

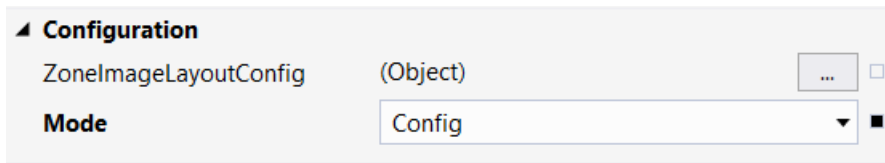


#### 3. Einstellen des Serversymbols als persistent über die Checkbox (Persist).

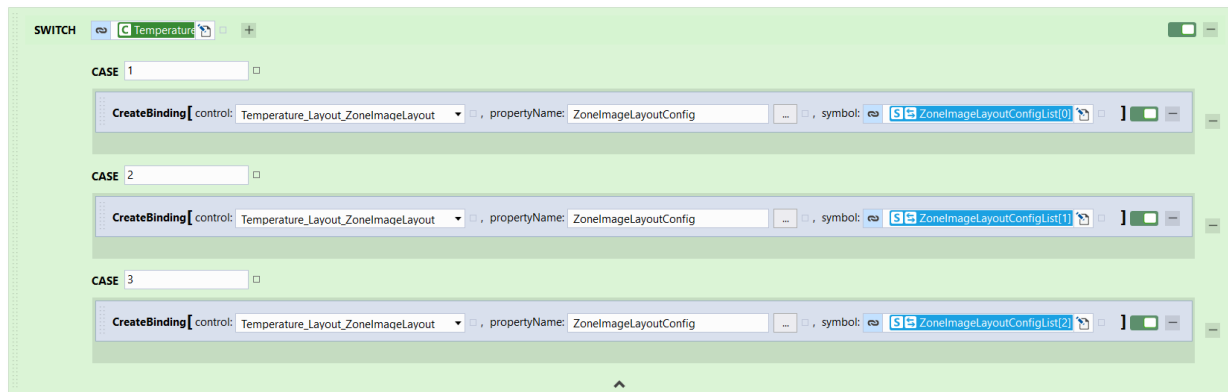
⇒ Serversymbol ist erzeugt.

**Nutzung des Serversymbols im Konfigurationsmodus des ZoneImageLayout-Controls**

1. ZoneImageLayout-Control in der Toolbox auswählen und auf einen Content ziehen.
2. Mode-Parameter unter der Kategorie Configuration auf Config stellen.



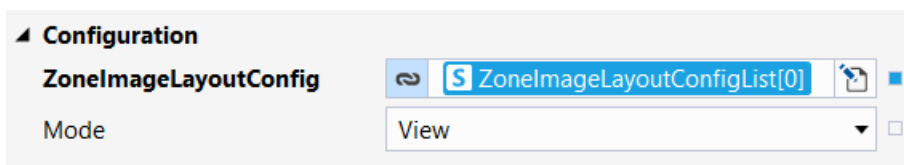
3. Zum Beispiel mithilfe eines TcHmiCombobox-Controls die im LiveView zur Verfügung stehende Anzahl an Layouts über das .onSelectionChanged Event konfigurieren.



4. Dazu kann ein neuer CASE mit dem Wert 4 für den SelectedId-Parameter der Combobox angelegt werden.
  5. Die CreateBinding-Funktion kopieren und in den neuen CASE einfügen und beim symbol-Übergabeparameter einen anderen Arrayplatz wie z. B. ZoneImageLayoutConfigList[3] auswählen.
- ⇒ Serversymbol ist im Konfigurationsmodus korrekt genutzt.

**Verwendung des Serversymbols im Anzeigemodus**

1. ZoneImageLayout-Control in der Toolbox auswählen und auf einen Content ziehen.
2. Unter der Kategorie Configuration den ZoneImageLayoutConfig-Parameter mit einem der Arrayplätze des ZoneImageLayoutConfigList-Serversymbols verknüpfen und den Mode-Parameter auf View stellen.



3. Die eingestellte Breite und Höhe des ZoneImageLayout-Controls muss in dem Serversymbol eingestellt werden (LayoutHeight, LayoutHeightUnit, LayoutWidth und LayoutWidthUnit). Dazu mit einem Rechtsklick auf das Serversymbol im TwinCAT HMI Configuration Fenster klicken und den Defaultwert anpassen.
- ⇒ Serversymbol ist im Anzeigemodus korrekt genutzt.

**8.3 PLC-API (obsolet)**

In diesem Kapitel werden die veralteten PLC-Elemente aufgeführt. Diese sind zu Kompatibilitätszwecken weiterhin verfügbar, ein Umstieg auf die ablösenden Elemente wird dringend empfohlen!

### 8.3.1 F\_TryDivide()

#### ● Funktion ist ‚obsolete‘

**i** Diese Funktion ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: [F\\_TryDivide](#) [► 120]

#### Originalfassung:

```

F_TryDivide
fNominator LREAL HRESULT F_TryDivide
fDenominator LREAL
refResult REFERENCE TO LREAL

```

Dividiert zwei Werte, ohne eine Exception auszulösen.

#### ● Rückgabewert mathematisch ungültig

**i** Die Funktion definiert den mathematisch ungültigen Fall  $x / 0$  als 0. Dies ist ein mathematisch ungültiges Ergebnis, ist jedoch für viele Anwendungsfälle hinreichend. Prüfen Sie für ihren Anwendungsfall, ob diese Definition nicht zu unvorhergesehenem Fehlverhalten führt.

#### Syntax:

```

FUNCTION F_TryDivide : HRESULT
VAR_INPUT
    fNominator:    LREAL;
    fDenominator:  LREAL;
    refResult:     REFERENCE TO LREAL;
END_VAR

```

#### 📦 Eingänge

Name	Typ	Beschreibung
fNominator	LREAL	Wert der geteilt werden soll
fDenominator	LREAL	Wert durch den geteilt werden soll
refResult	REFERENCE TO LREAL	Ergebnis der Division

#### 📦 Ausgänge

Name	Typ	Beschreibung
F_TryDivide	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

#### Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.2 FB\_TrafoTableGenerator

#### ● Klasse ist ‚obsolete‘

**i** Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: TF8560 – Tc3\_PlasticFunctions.FB\_TrafoTableGenerator



**Originalfassung:**



Stellt die Basisklasse für Erzeugungsalgorithmen von Transformations-Kurvenscheiben dar.

**● Klasse ist abstrakt**

**i** Da die Klasse als `ABSTRACT` definiert ist, kann die Klasse nicht instanziiert werden und muss mittels Vererbung implementiert werden.

Folgende vorimplementierte Geometrien sind inklusive:

1. [FB\\_TableGeneratorClampStandard \[▶ 211\]](#) – Geometrie einer typischen Schließeinheit
2. [FB\\_TableGeneratorCrankStandard \[▶ 213\]](#) – Geometrie eines Kurbeltriebs
3. [FB\\_TableGeneratorScotchYoke \[▶ 214\]](#) – Geometrie eines Scotch Yokes

**Syntax:**

```
FUNCTION_BLOCK ABSTRACT FB_TrafoTableGenerator
```

**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DriveHighEnd	LREAL	Get, Set	0.0	Oberes Limit der Antriebsposition
DriveLowEnd	LREAL	Get, Set	0.0	Unteres Limit der Antriebsposition
LookUp	I_CammingLookUp	Get, Set	NULL	Schnittstelle auf die zuzuweisende TF8560 FB_CammingLookUp Tabelle
ParameterList	I_Parameter	Get	-	Liste der geometriespezifischen Parameter
ParamValid	BOOL	Get	FALSE	Die eingestellten Parameter haben gültige Werte.
<a href="#">Scope [▶ 210]</a>	I_TrafoScope	Get	-	Diagnose-Werte aus der Berechnung der Transformationstabelle

**Methoden**

Name	Beschreibung
CalculateScope()	Berechnet die Diagnose-Werte der Eigenschaft Scope.
DefineTable()	Startet die Berechnung der Transformationstabelle.
ReadFromParamList()	Liest Parameter aus der Liste spezieller Parameter.
WriteToParamList()	Schreibt lokale Variablen in die Liste der speziellen Parameter.

**● Methoden sind abstrakt**

**i** Die Methoden sind als `ABSTRACT` definiert und müssen in erbdenden Klassen implementiert werden.

## Schnittstellen

Typ	Beschreibung
I_TrafoTableGenerator	Standardschnittstelle auf FB_TrafoTableGenerator

### Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.2.1 FB\_TrafoScope

#### Klasse ist ,obsolete‘

**i** Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: - (vollständig entfernt)

#### Originalfassung:



Implementiert Diagnose-Werte für die Berechnung der Transformationstabelle

#### Syntax:

```
FUNCTION_BLOCK FB_TrafoScope
```

#### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
HighLimitDefined	BOOL	Get, Set	FALSE	Ein oberes Limit ist definiert.
HighLimitDrive	LREAL	Get, Set	0.0	Oberes Limit des Antriebs
HighLimitLoad	LREAL	Get, Set	0.0	Oberes Limit der Lastseite
HighSideBlockpoint	BOOL	Get, Set	FALSE	Eine obere Blockierung der Mechanik existiert.
HighSideTurnpoint	BOOL	Get, Set	FALSE	Ein oberer Umkehrpunkt der Mechanik existiert.
LowLimitDefined	BOOL	Get, Set	FALSE	Ein unteres Limit ist definiert.
LowLimitDrive	LREAL	Get, Set	0.0	Unteres Limit des Antriebs
LowLimitLoad	LREAL	Get, Set	0.0	Unteres Limit der Lastseite
LowSideBlockpoint	BOOL	Get, Set	FALSE	Eine untere Blockierung der Mechanik existiert.
LowSideTurnpoint	BOOL	Get, Set	FALSE	Ein unterer Umkehrpunkt der Mechanik existiert.

## Schnittstellen

Typ	Beschreibung
I_TrafoScope	Standardschnittstelle auf FB_TrafoScope

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.2.2 FB\_TableGeneratorClampStandard\_1

- **Klasse ist ,obsolete‘**



Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: TF8560 – Tc3\_PlasticFunctions.FB\_ClampTableGenerator

#### Originalfassung:

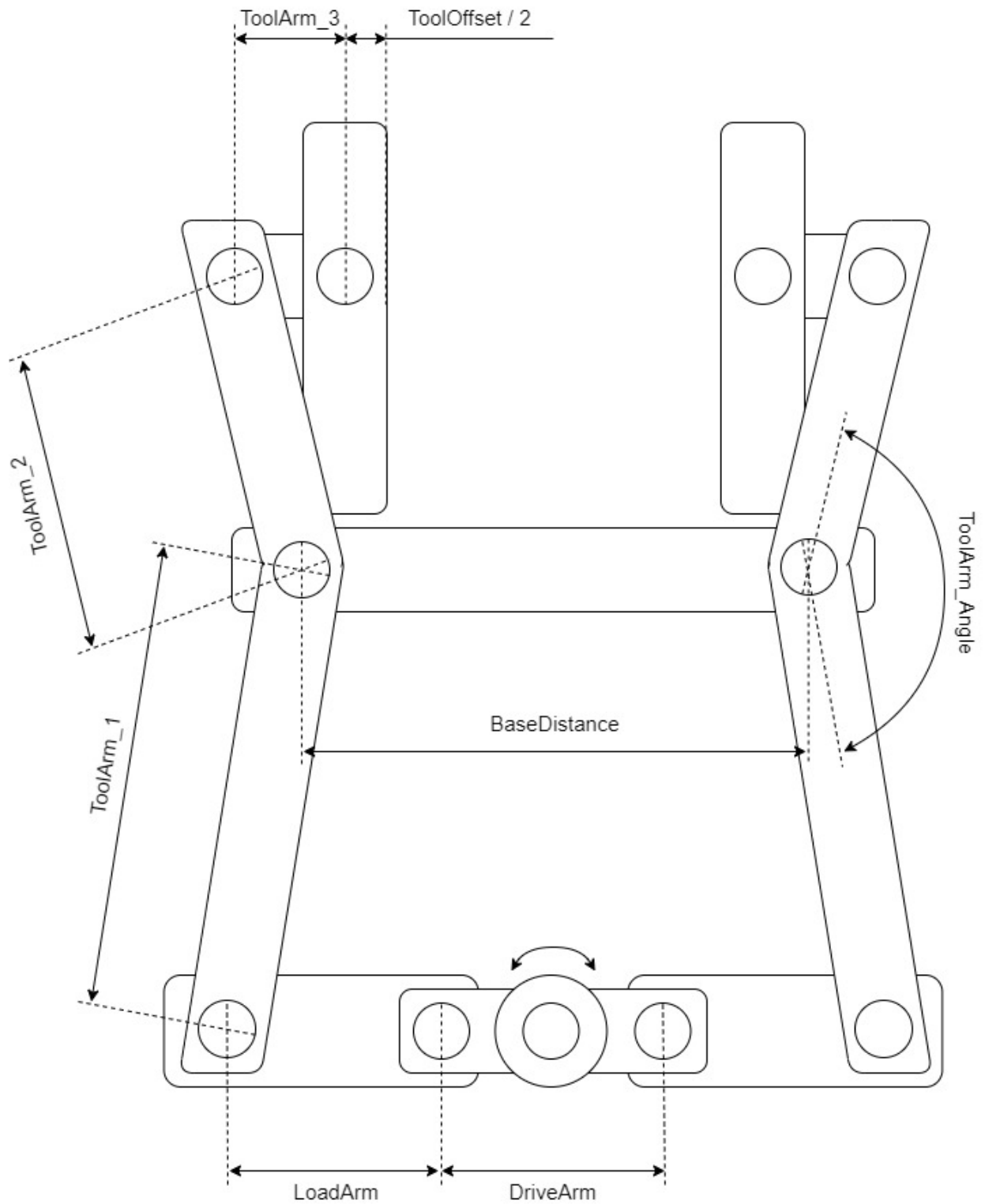
**FB\_TableGeneratorClampStandard\_1**

Implementiert einen Berechnungsalgorithmus für einen Standard-Schließenheit Mechanismus.

- **Namen der Bemaßungsbezeichnungen in der Skizze**



Die Bemaßungsbezeichnungen gleichen den Namen der zu konfigurierenden Eigenschaften der Klasse.

**Syntax:**

```
FUNCTION_BLOCK FB_TableGeneratorClampStandard_1 EXTENDS FB_TrafoTableGenerator
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
BaseDistance	LREAL	Get, Set	0.0	Abstand zwischen den zwei ortsfesten Lagerpunkten in der Mitte des Mechanismus
DriveArm	LREAL	Get, Set	0.0	Am Antrieb befestigter Hebelarm
DriveStep	LREAL	Get,	0.0	Resultierende Auflösung der Antriebsposition in der Transformationstabelle
LoadArm	LREAL	Get, Set	0.0	Übertragungshebel auf die Werkzeughebel
ToolArm_1	LREAL	Get, Set	0.0	Unterer Teil des seitlichen Werkzeugarms
ToolArm_2	LREAL	Get, Set	0.0	Oberer Teil des seitlichen Werkzeugarms
ToolArm_3	LREAL	Get, Set	0.0	Horizontaler Werkzeugarm
ToolArm_Angle	LREAL	Get, Set	0.0	Winkel zwischen den zwei Teilen des seitlichen Werkzeugarms
ToolOffset	LREAL	Get, Set	0.0	Gesamtversatz zwischen der Lagerung und der Werkzeugaufspannfläche  In der empfohlenen Auslegung des ToolOffset ergibt die Transformation den Abstand der Werkzeugaufspannflächen. Um die Distanz der Werkzeugöffnung zu nutzen, wird empfohlen die ToolAdaption zu verwenden. Diese ist in den Maschinendaten jeder FB_PtpMotion basierten Achse konfigurierbar.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.2.3 FB\_TableGeneratorCrankStandard**

**● Klasse ist ‚obsolete‘**



Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: TF8560 – Tc3\_PlasticFunctions.FB\_CrankTableGenerator

**Originalfassung:**



Implementiert einen Berechnungsalgorithmus für einen Kurbeltriebmechanismus.

**● Konstruktive Annahme**

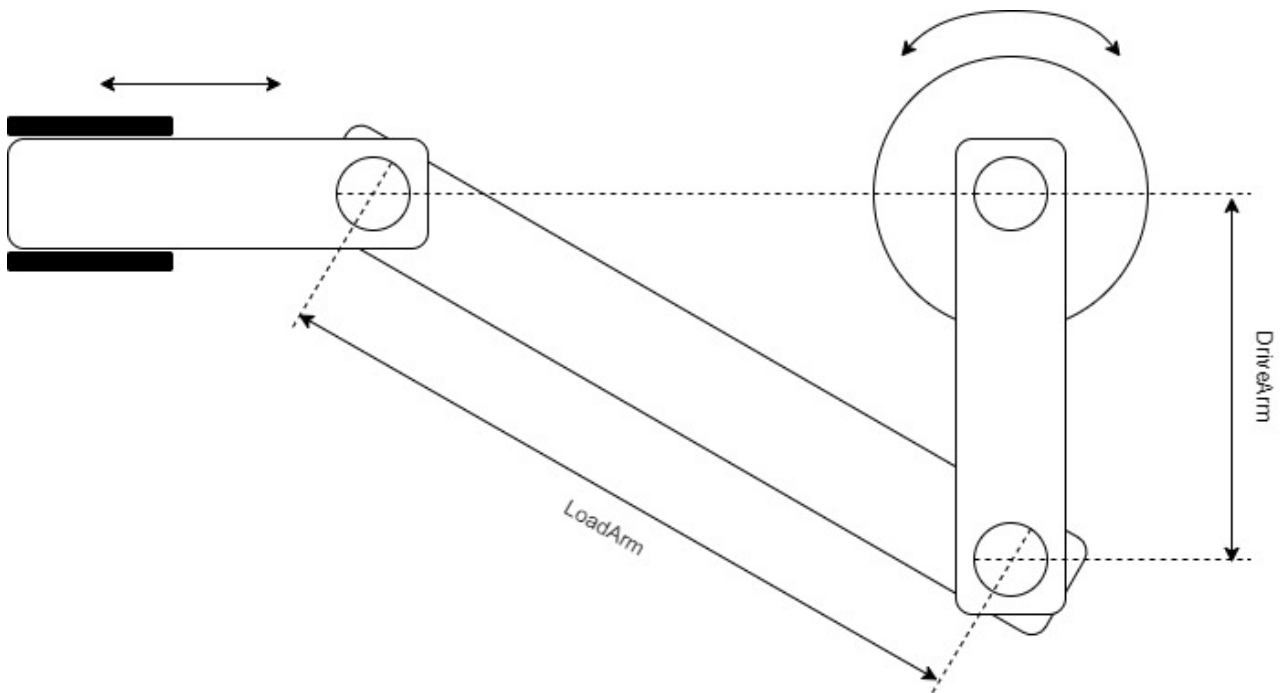


Die Berechnung setzt voraus, dass die Bewegungsachse der Lastseite in Flucht mit dem Drehpunkt der Antriebsseite ist. Dies ist in der Skizze durch die horizontale gestrichelte Linie angedeutet!

**● Namen der Bemaßungsbezeichnungen in der Skizze**



Die Bemaßungsbezeichnungen gleichen den Namen der zu konfigurierenden Eigenschaften der Klasse.

**Syntax:**

```
FUNCTION_BLOCK FB_TableGeneratorCrankStandard EXTENDS FB_TrafoTableGenerator
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DriveArm	LREAL	Get, Set	0.0	Am Antrieb befestigter Hebelarm
DriveStep	LREAL	Get	0.0	Resultierende Auflösung der Antriebsposition in der Transformationstabelle
LoadArm	LREAL	Get, Set	0.0	Übertragungshebel auf die geführte Lastseite

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.2.4 FB\_TableGeneratorScotchYoke****Klasse ist ,obsolete‘**

Diese Klasse ist als *obsolete* markiert und sollte nicht weiterhin verwendet werden!

Alternative: TF8560 – Tc3\_PlasticFunctions.FB\_ScotchYokeTableGenerator

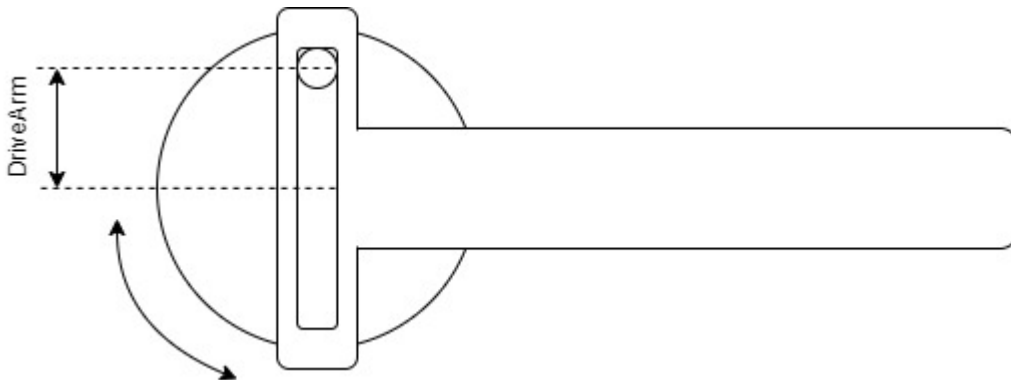
**Originalfassung:**

**FB\_TableGeneratorScotchYoke**

Implementiert einen Berechnungsalgorithmus für einen Scotch-Yoke-Mechanismus.

**● Namen der Bemaßungsbezeichnungen in der Skizze**

**i** Die Bemaßungsbezeichnungen gleichen den Namen der zu konfigurierenden Eigenschaften der Klasse.



**Syntax:**

```
FUNCTION_BLOCK FB_TableGeneratorScotchYoke EXTENDS FB_TrafoTableGenerator
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DriveArm	LREAL	Get, Set	0.0	Radius des exzentrischen Drehpunktes
DriveStep	LREAL	Get	0.0	Resultierende Auflösung der Antriebsposition in der Transformationstabelle

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.3 FB\_MonitoringZone**

**● Klasse ist ,obsolete‘**

**i** Diese Klasse ist als *obsolete* markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_MonitoringTemp](#) [▶ 88]

**Originalfassung:**



Implementiert die Überwachung einer Temperaturzone mit direkter Verbindung zu TF8540. Die einstellbaren Limitierungen werden mit der Temperaturzone synchronisiert.

**Syntax:**

```
FUNCTION_BLOCK FB_MonitoringZone EXTENDS FB_Monitoring
```

### Eigenschaften

Name	Typ	Zugriff	Initialwert	Beschreibung
TempZoneHmi	I_TempZoneHmi	Get	NULL	[INTERNAL] Referenzierung auf die Werte der zugewiesenen Zone

### Methoden

Name	Beschreibung
SetTempZone()	Setzt die zu überwachende Temperaturzone

### Schnittstellen

Typ	Beschreibung
I_MonitoringZone	Standardschnittstelle auf FB_MonitoringZone

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 8.3.4 FB\_TempCtrl

### Klasse ist ,obsolete‘

**i** Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_Temperature](#) [► 51]

### Originalfassung:



Hauptklasse der Temperaturregelung zur Verwaltung aller Temperaturzonen und -gruppen.

### Syntax:

```
FUNCTION_BLOCK FB_TempCtrl EXTENDS FB_BaseMd
```



 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
DisableAlarms	BOOL	Get, Set	FALSE	Unterdrückt Alarme, die durch Fehler in einer Temperaturzone ausgelöst werden.
DisableMessages	BOOL	Get, Set	FALSE	Unterdrückt Debug-Meldungen der TF8540 Bibliothek.
EnableCallback	BOOL	Get, Set	TRUE	Aktiviert die Kommunikation mit den E/A-Komponenten.
EnableConfig	BOOL	Get, Set	TRUE	Gibt die Konfiguration aller Zonen frei.
EnableLooptest	BOOL	Get, Set	FALSE	Aktiviert die Stromüberwachung aller Zonen.
Groups	REFERENCE TO ARRAY[] OF FB_TempGroup	Get	-	Steuerung der einzelnen Gruppen.
LibScopeVars	REFERENCE TO FB_Scope_TempCtrlV variables	Get	-	Zugriff auf eine Übersicht an TF8540 Live Daten.
RefMdTempSupply	REFERENCE TO ARRAY [] OF FB_MdTempSupply	Get	-	Zugriff auf das Array an Maschinendaten-Containern aller Versorgungseinheiten.
RefMdTempZone	REFERENCE TO ARRAY[] OF FB_MdTempZone	Get	-	Zugriff auf das Array an Maschinendaten-Containern aller Temperaturzonen.
Timer	I_TempSchedule	Get	NULL	Zugriff auf den verbundenen Zeitplaner.

 **Methoden**

Name	Beschreibung
<a href="#">CreateDefaultParams() [▶ 218]</a>	Erzeugt eine Standardparametrierung für alle Temperaturzonen.
<a href="#">EnableAll() [▶ 219]</a>	Gibt alle Temperaturzonen PLC-seitig frei.
<a href="#">LinkGroup() [▶ 219]</a>	Weist eine lineare Anordnung an Zonen einer Gruppe zu.
<a href="#">LinkSupply() [▶ 220]</a>	Weist eine Gruppe einer Versorgungseinheit zu.
<a href="#">LinkZone() [▶ 221]</a>	Weist eine Zone einer Gruppe zu.
<a href="#">SetOpMode() [▶ 221]</a>	Konfiguriert den aktuellen Betriebsmodus.
<a href="#">SetScheduler()</a>	Weist der Temperaturregelung eine Zeitplanung zu.
<a href="#">StandbyAll() [▶ 222]</a>	Stellt alle Zonen auf Standby.
<a href="#">UnlinkGroup()</a>	Entfernt alle Links zu einer Gruppe.

 **Schnittstellen**

Typ	Beschreibung
I_TempCtrl	Standardschnittstelle auf FB_TempCtrl
I_TempCtrlMdRef	Schnittstelle zur Übergabe der Zonen-Maschinendaten
I_TempTaskInterface	Runtime Schnittstelle für eine langsame PLC Task

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.4.1 CreateDefaultParams()

CreateDefaultParams	
bAllInUse	BOOL
eSensor	E_TcPfw_TempSensType
eTerminal	E_TcPfw_TerminalType
nChPerTerm	INT
eOutHeating	E_TcPfw_TctrlOutSelect
eOutCooling	E_TcPfw_TctrlOutSelect
fSetpoint	LREAL
fStandbySetp	LREAL
fPwmCycleTime	LREAL

Erzeugt eine Standardparametrierung für alle Temperaturzonen.

#### Syntax:

```
METHOD CreateDefaultParams : HRESULT
VAR_INPUT
    bAllInUse:          BOOL;
    eSensor:            E_TcPfw_TempSensType;
    eTerminal:          E_TcPfw_TerminalType;
    nChPerTerm:         INT;
    eOutHeating:        E_TcPfw_TctrlOutSelect;
    eOutCooling:        E_TcPfw_TctrlOutSelect;
    fSetpoint:          LREAL;
    fStandbySetp:       LREAL;
    fPwmCycleTime:     LREAL;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung	Empfohlener Standard
bAllInUse	BOOL	Alle Zonen werden als „InUse“ initialisiert.	TRUE
eSensor	E_TcPfw_TempSensType	Sensortyp – NoSensor im Simulationsmodus	eTcPfwTempSensT_NoSensor
eTerminal	E_TcPfwTerminalType	Klemmentyp – NoTerminal im Simulationsmodus	eTcPfwTermT_NoTerminal
nChPerTerm	INT	Anzahl der Kanäle pro Klemme	8
eOutHeating	E_TcPfw_TctrlOutSelect	Ausgabeart des Heizausgangs – NoSignal, um die Heizfunktion zu deaktivieren (bspw. bei Messzonen)	eTcPfwTcOut_PWM
eOutCooling	E_TcPfw_TctrlOutSelect	Ausgabeart des Kühlausgangs	eTcPfwTcOut_NoSignal
fSetpoint	LREAL	Temperatursollwert für alle Zonen	180.0
fStandbySetp	LREAL	Temperatursollwert für die Standby-Temperatur aller Zonen	18.0
fPwmCycleTime	LREAL	PWM-Zykluszeit für alle Ausgänge (dutyCycle = fPwmCycleTime * 0.1)	1.0

 **Ausgänge**

Name	Typ	Beschreibung
CreateDefaultParams	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.4.2 EnableAll()**



Gibt alle Temperaturzonen PLC-seitig frei.

**Syntax:**

```

METHOD EnableAll
VAR_INPUT
    bCmd:      BOOL;
    bGroups:   BOOL;
END_VAR
    
```

 **Eingänge**

Name	Typ	Beschreibung
bCmd	BOOL	TRUE um die Freigabe zu erteilen, FALSE um die Freigabe zurückzunehmen.
bGroups	BOOL	Die Freigabe berücksichtigt nur Zonen, die einer Gruppe zugewiesen sind.

 **Ausgänge**

Name	Typ	Beschreibung
EnableAll	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.4.3 LinkGroup()**



Weist eine Reihe an Zonen einer Gruppe zu.

**Syntax:**

```

METHOD LinkGroup : HRESULT
VAR_INPUT
    nStartIdx:      INT;
    nEndIdx:        INT;
    nGroupIdx:      INT;
    bOverwrite:     BOOL;
END_VAR

```

**👉 Eingänge**

Name	Typ	Beschreibung
nStartIdx	INT	Index der ersten Zone die aus der linearen TF8540-Library Anordnung zugewiesen werden soll
nEndIdx	INT	Index der letzten Zone die aus der linearen TF8540-Library Anordnung zugewiesen werden soll
nGroupIdx	INT	Index der Gruppe der die Zonen zugewiesen werden sollen
bOverwrite	BOOL	Zonen werden zugewiesen, auch wenn die Gruppe bereits zugewiesene Zonen enthält.

**👈 Ausgänge**

Name	Typ	Beschreibung
LinkGroup	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwincAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.4.4 LinkSupply()**

Weist einer Gruppe eine Versorgungseinheit zu.

**Syntax:**

```

METHOD LinkSupply : HRESULT
VAR_INPUT
    nGroupIdx:      INT;
    nSupplyIdx:     INT;
END_VAR

```

**👉 Eingänge**

Name	Typ	Beschreibung
nGroupIdx	INT	Index der Gruppe der eine Versorgungseinheit zugewiesen werden soll
nSupplyIdx	INT	Index der Versorgungseinheit die der Gruppe zugewiesen werden soll

**Ausgänge**

Name	Typ	Beschreibung
LinkSupply	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.4.5 LinkZone()**



Weist eine einzelne Zone einer Gruppe hinzu.

**Syntax:**

```
METHOD LinkZone : HRESULT
VAR_INPUT
    nLinearIdx:      INT;
    nGroupIdx:       INT;
    nGroupMemberIdx: INT;
    bOverwrite:      BOOL;
END_VAR
```

**Eingänge**

Name	Typ	Beschreibung
nLinearIdx	INT	Index der Zone aus der linearen TF8540-Library Anordnung, die zugewiesen werden soll
nGroupIdx	INT	Index der Gruppe, der die Zone zugewiesen werden soll
nGroupMemberIdx	INT	Index in der Zielgruppe
bOverwrite	BOOL	Zone wird zugewiesen, auch wenn der Index in der Zielgruppe bereits belegt ist.

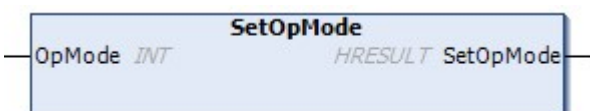
**Ausgänge**

Name	Typ	Beschreibung
LinkZone	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.4.6 SetOpMode()**



Definiert den aktuellen OpMode der Temperaturregelung.

- OpMode None (0)
  - Temperaturregelung nimmt keine Kommandos an
- OpMode Simple (1)
  - Zonen können ein- und ausgeschaltet werden über TempCtrlHmi.Group[].Enable
- OpMode Standard (2)
  - Zonen werden kontrolliert über TempCtrlHmi.Groups[].OpModeActive
  - Zonen unterstützen Standby über TempCtrlHmi.Groups[].OpModeStandby

### Syntax:

```
METHOD SetOpMode : HRESULT
VAR_INPUT
    OpMode:          INT;
END_VAR
```

### 🔌 Eingänge

Name	Typ	Beschreibung
OpMode	INT	Auswahlparameter: 0 – None, 1 – Simple, 2 - Standard

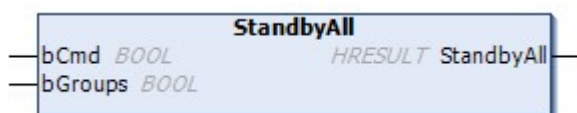
### 🔌 Ausgänge

Name	Typ	Beschreibung
SetOpMode	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

## 8.3.4.7 StandbyAll()



Stellt alle Zonen auf Standby-Temperatur.

### Syntax:

```
METHOD StandbyAll : HRESULT
VAR_INPUT
    bCmd:          BOOL;
    bGroups:       BOOL;
END_VAR
```

### 🔌 Eingänge

Name	Typ	Beschreibung
bCmd	BOOL	TRUE um die Standby-Temperatur zu aktivieren, FALSE zum deaktivieren.
bGroups	BOOL	Die Funktion berücksichtigt nur Zonen, die einer Gruppe zugewiesen sind.

 **Ausgänge**

Name	Typ	Beschreibung
StandbyAll	HRESULT	Rückgabewert mit Feedback zum Erfolg der Ausführung

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.5 FB\_TempCtrlHmi

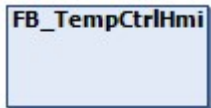
**● Klasse ist ,obsolete‘**



Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_TemperatureHmi](#) [[▶ 57](#)]

**Originalfassung:**



HMI-Parallelklasse zum `FB_TempCtrl`.

**Syntax:**

```
FUNCTION_BLOCK FB_TempCtrlHmi EXTENDS FB_BaseMdHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Eine Zone (InUse = TRUE) hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Eine Zone (InUse = TRUE) hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Mindestens eine Zone mit aktiver Regelung hat die HighHigh-Toleranz überschritten.
AlarmHigh	BOOL	Get	FALSE	Mindestens eine Zone mit aktiver Regelung hat die High-Toleranz überschritten.
AlarmLow	BOOL	Get	FALSE	Mindestens eine Zone mit aktiver Regelung hat die Low-Toleranz unterschritten.
AlarmLowLow	BOOL	Get	FALSE	Mindestens eine Zone mit aktiver Regelung hat die LowLow-Toleranz unterschritten.
CountPfwZones	LREAL	Get	20.0	Anzahl an verfügbaren TF8540 Temperaturzonen
Groups	REFERENCE TO ARRAY[] OF FB_TempGroupHmi	Get	-	Zugriff auf Gruppenbasierte Informationen
ParamTempSupply	REFERENCE TO ARRAY[] OF FB_MdTempSupplyHmi	Get	-	Parameterschnittstelle für die Parametrierung über das HMI
ParamTempZone	REFERENCE TO ARRAY[] OF FB_MdTempZoneHmi	Get	-	Parameterschnittstelle für die Parametrierung über das HMI
TempAmbient	LREAL	Get, Set	18.0	Standard-Umgebungstemperatur (für Simulation)

 **Schnittstellen**

Typ	Beschreibung
I_TempCtrlHmi	Standardschnittstelle auf FB_TempCtrlHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)



### 8.3.6 FB\_TempGroup

---

● **Klasse ist ,obsolete‘**



Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

---

Alternative: [FB\\_TemperatureGroup](#) [[▶ 59](#)]

**Originalfassung:**



Klasse zur Gruppensteuerung der Temperaturregelung.

**Syntax:**

```
FUNCTION_BLOCK FB_TempGroup EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat die HighHigh-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmHigh	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat die High-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLow	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat die Low-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmLowLow	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat die LowLow-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zonen aktiv geregelt werden.
AlarmNoResponse	BOOL	Get	FALSE	Temperaturwert der Gruppe (mindestens eine Zone) reagiert nicht auf die Ansteuerung.
EnableLimitAlarms	BOOL	Get, Set	FALSE	Das Überschreiten eines Toleranzwertes/-limits löst einen Alarm aus.
Fault	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) hat einen Fehler.
IsActive	BOOL	Get	FALSE	Gruppe (mindestens eine Zone) ist aktiv geregelt.
IsEnabled	BOOL	Get	FALSE	Alle Zonen der Gruppe sind freigegeben.
IsStandby	BOOL	Get	FALSE	Alle Zonen der Gruppe sind im StandbyBetrieb.
Zones	REFERENCE TO ARRAY[] OF FB_TempZone	Get	-	Steuerung der einzelnen Zonen

**Methoden**

Name	Beschreibung
Enable()	Gibt alle Zonen der Gruppe PLC-seitig frei.
Force()	Zwingt alle Zonen der Gruppe auf Heiz-/Kühlleistung 100%.

**Schnittstellen**

Typ	Beschreibung
I_TempGroup	Standardschnittstelle auf FB_TempGroup

**Voraussetzungen**

Entwicklungsumgebung	Zielpattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.7 FB\_TempGroupHmi



**Klasse ist ,obsolete‘**

Diese Klasse ist als *obsolete* markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_TemperatureGroupHmi](#) [▶ 63]

**Originalfassung:**



HMI-Parallelklasse zur FB\_TempGroup Klasse.

**Syntax:**

```
FUNCTION_BLOCK FB_TempGroupHmi EXTENDS FB_BaseMdHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
CountLinkedZones	INT	Get, Set	0	Anzahl an zugewiesenen Zonen über die ‚Link()‘-Methoden des FB_TempCtrl
DoTune	BOOL	Get, Set	FALSE	Startet das Tuning aller aktiven Zonen der Gruppe.
Enable <sup>1</sup>	BOOL	Get, Set	FALSE	Gibt die Zonen durch das HMI frei.
GroupIndex	INT	Get	0	Index der Gruppe im Array des FB_TempCtrlHmi
GroupName	STRING	Get	“	Name der Temperaturgruppe
OpModeActive <sup>2</sup>	REFERENCE TO FB_TempGroupOpModeHmi	Get	-	Interface zum Aktivschalten einer Temperaturgruppe
OpModeStandby <sup>2</sup>	REFERENCE TO FB_TempGroupOpModeHmi	Get	-	Interface für die Standbyschaltung der Temperaturgruppe
TuningActive	BOOL	Get	FALSE	Das Tuning von mindestens einer Zone ist aktiv.
TuningDone	BOOL	Get	FALSE	Das Tuning der Gruppe ist abgeschlossen.
Zones	REFERENCE TO ARRAY[] OF FB_TempZoneHmi	Get	-	Interface auf die einzelnen Zonen einer Gruppe

<sup>1</sup> Nur im OpMode ‚Simple‘

<sup>2</sup> Nur im OpMode ‚Standard‘

 **Methoden**

Name	Beschreibung
SetZoneData()	[INTERNAL] Verbindet die Temperaturzonen einer Gruppe mit den Maschinendaten.

 **Schnittstellen**

Typ	Beschreibung
I_TempGroupHmi	Standardschnittstelle auf FB_TempGroupHmi

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

**8.3.8 FB\_TempGroupOpModeHmi**
 **Klasse ist ‚obsolete‘**

Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: - (integriert in [FB\\_TemperatureGroupHmi](#) |▶ 63|)

**Originalfassung:**



Klasse zur Steuerung des Temperaturbetriebsmodus (im FB\_TempCtrl OpMode „Standard“).

**Syntax:**

```
FUNCTION_BLOCK FB_TempGroupOpModeHmi
```



**Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Off	BOOL	Get, Set	TRUE	Schaltet die Temperaturgruppe aus.
On	BOOL	Get, Set	FALSE	Schaltet die Temperaturgruppe an.
Timed	BOOL	Get, Set	FALSE	Schaltete die Temperaturgruppe in eine Zeitbasierte Einschaltung.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.9 FB\_TempZone



**Klasse ist ‚obsolete‘**

Diese Klasse ist als *obsolete* markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_TempChannel](#) [► 67]

**Originalfassung:**



Klasse zur Einzelsteuerung einer Temperaturzone.

**Syntax:**

```
FUNCTION_BLOCK FB_TempZone EXTENDS FB_Base
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
AlarmAbsoluteHigh	BOOL	Get	FALSE	Zone hat das absolute Temperaturmaximum überschritten.
AlarmAbsoluteLow	BOOL	Get	FALSE	Zone hat das absolute Temperaturminimum unterschritten.
AlarmHighHigh	BOOL	Get	FALSE	Zone hat die HighHigh-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zone aktiv geregelt wird.
AlarmHigh	BOOL	Get	FALSE	Zone hat die High-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zone aktiv geregelt wird.
AlarmLow	BOOL	Get	FALSE	Zone hat die Low-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zone aktiv geregelt wird.
AlarmLowLow	BOOL	Get	FALSE	Zone hat die LowLow-Toleranz überschritten. Alarmer zu den Toleranzwerten sind nur aktiv, wenn die Zone aktiv geregelt wird.
AlarmNoResponse	BOOL	Get	FALSE	Temperaturwert der Zone reagiert nicht auf die Ansteuerung.
EnableLimitAlarms	BOOL	Get, Set	FALSE	Das Überschreiten eines Toleranzwertes/-limits löst einen Alarm aus.
IsLinked	BOOL	Get	FALSE	Zone ist mit einer Gruppe verbunden.

 **Schnittstellen**

Typ	Beschreibung
I_TempZone	Standardschnittstelle auf FB_TempZone

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwincAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)

### 8.3.10 FB\_TempZoneHmi

**● Klasse ist ,obsolete‘**

**i** Diese Klasse ist als `obsolete` markiert und sollte nicht weiterhin verwendet werden!

Alternative: [FB\\_TempChannelBase](#) [▶ 66]

**Originalfassung:**



Zugriffsklasse für eine einzelne Temperaturzone über das HMI.

**Syntax:**

```
FUNCTION_BLOCK FB_TempZoneHmi
```

 **Eigenschaften**

Name	Typ	Zugriff	Initialwert	Beschreibung
Index	INT	Get, Set	0	Referenziert eine PlasticApplication Zone zu einer TF8540 Zone.

**● Klasse enthält deutlich mehr Eigenschaften als aufgelistet**

**i** Die Eigenschaften der `FB_TempZoneHmi` Klasse überschneiden sich mit den TF8540 Globalen Variablen `aaaPfwTempToHmi`, `aaaPfwTempMparamFromHmi` und `aaaPfwTempPparamFromHmi`. Für weitere Informationen zu den überschneidenden Eigenschaften wird empfohlen, die TF8540-Dokumentation zu nutzen.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.35	PC oder CX (x64, x86)	Tc3_PlasticBaseApplication (>= v3.12.5.0)





Mehr Informationen:

[www.beckhoff.com/de-de/produkte/automation/twincat/tfxxxx-twincat-3-functions/tf8xxx-industry-specific](http://www.beckhoff.com/de-de/produkte/automation/twincat/tfxxxx-twincat-3-functions/tf8xxx-industry-specific)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
info@beckhoff.com  
www.beckhoff.com

