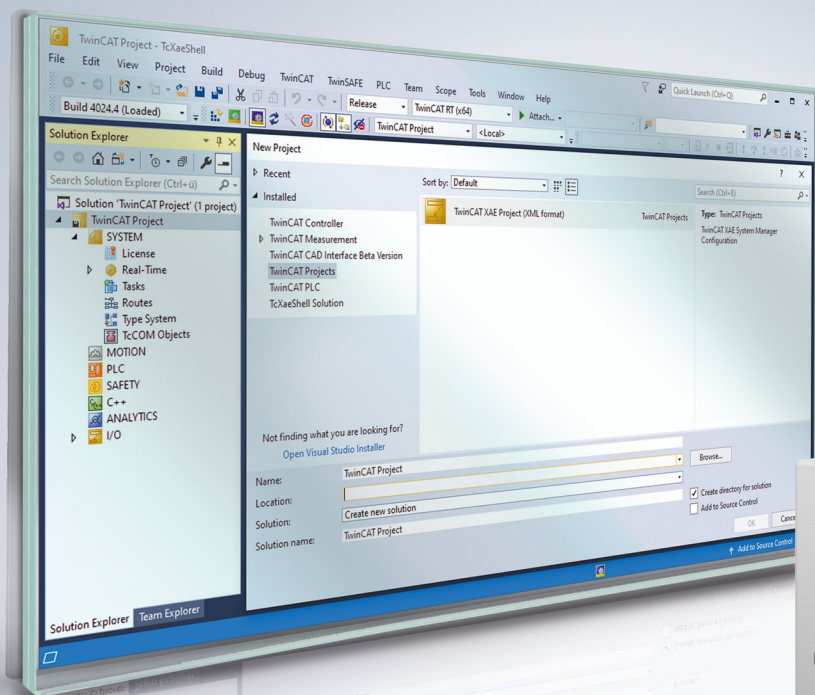


BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc3_PackML



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Einleitung.....	8
2.1	Übersicht.....	8
3	Packaging Machine State	9
3.1	Datentypen.....	9
3.1.1	E_PMLState	9
3.1.2	E_PMLUnitMode	10
3.2	Funktionsbausteine	11
3.2.1	PS_PackML_StateMachine_Auto	11
3.2.2	PS_PackML_StateMachine_Maintenance.....	13
3.2.3	PS_PackML_StateMachine_Manual.....	16
3.2.4	PS_PackML_StateMachine_SemiAuto.....	18
3.2.5	PS_UnitModeManager.....	21
4	Packaging Machine Tags	25
4.1	Einleitung	25
4.2	Tag-Arten	25
4.3	Tag-Details.....	26
4.4	Datentypen.....	30
4.4.1	Alarm.....	30
4.4.2	Allgemein	32
4.4.3	ST_PMLa	34
4.4.4	ST_PMLc	35
4.4.5	ST_PMLs	35
4.5	Globale Parameter	36
4.6	Globale Konstanten.....	37
5	Support und Service	38

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte 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 weiter entwickelt.

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.

Zuwendungen 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

2.1 Übersicht

OMAC Packaging Workgroup (OPW), eine Untergruppe der Arbeitsgruppe Open Modular Architecture Controls, hat einige Funktionen definiert, die Endbenutzern die Steuerung und Automatisierung von Verpackungsmaschinen und –systemen erleichtern.

Strategie der Bibliothek

- Oberste Priorität - Maschine, nachrangig - Prozess
- Benutzerfreundliche Schnittstelle für Verpackungsfunktionen
- Angeglihen an bestehende Verpackungsstandards
- Wiederverwendbare Elemente, verwendbar für viele Anwendungen.
- Anwendungsprogramm sollte auf jeder Plattform implementierbar sein
- Einheitliche Basis, Terminologie, Referenzen
- Die Anbindung dieser Funktionsbausteine an ein Anwendungsprogramm erfordert eine für Verpackungsanwendungen geeignete Umgebung. Diese Norm befasst sich mit einigen Anforderungen und Einschränkungen solcher Umgebungen.

Organisation des Verhaltens von Verpackungsmaschinen

[PS PackML StateMachine Auto \[► 11\]](#)

[PS PackML StateMachine Maintenance \[► 13\]](#)

[PS PackML StateMachine Manual \[► 16\]](#)

[PS PackML StateMachine SemiAuto \[► 18\]](#)

[PS UnitModeManager \[► 21\]](#)

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT 3.1 Build 4018	PC (i386)	Tc3_PackML

3 Packaging Machine State

Die Packaging Machine State Funktionsbausteine haben eine gemeinsame Schnittstelle zu den existierenden PackML Machine State Model Ausführungen. Es wird erwartet, dass anwendungsspezifische Logik, wie Zustands-Übergänge, in externen Funktionsbausteinen programmiert ist und der Packaging Machine State Funktionsbaustein die zentrale Logik der Zustandsmaschine und die Zustandsdarstellung übernimmt. Deswegen gibt es für diesen Funktionsbaustein eine Empfehlung, wie er mit anderer Logik kombiniert werden kann.

Der Zustandsübergang in einer Maschinenanwendung ist immer anwendungsspezifisch. Deswegen gestaltet man am besten mit PackML State Machine V3 verknüpfte „State“ Funktionsbausteine, um die Standardisierung zu vereinfachen. Die „State“ Funktionsbausteine erfassen anwendungsspezifische Signale und stellen die Übergangslogik zu angrenzenden Zuständen dar (siehe PackML Zustandsmodell). Die „State“ Funktionsbausteine geben Feedback an PackML State Machine V3, wodurch eine Standard-Zustandsmaschine und Zustandsmeldung möglich wird. Die „State“ Funktionsbausteine enthalten den Maschinenausführungscode und die anwendungsspezifische Übergangslogik.

„State“ Funktionsbausteine sind unten gelistet und werden vom Anwendungsprogrammierer so programmiert, dass Integrität und Funktionalität der PackML State Machine gewahrt bleibt.

Namen der PackML State Machine V3 Funktionsbausteine:

- PS_Starting
- PS_Completing
- PS_Resetting
- PS_Holding
- PS_Unholding
- PS_Suspending
- PS_Clearing
- PS_Stopping
- PS_Aborting
- PS_Execute
- PS_Complete
- PS_Idle
- PS_Held
- PS_Suspended
- PS_Stopped
- PS_Aborted

Anforderungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.1 Datentypen

3.1.1 E_PMLState

E_PMLState

```

TYPE E_PMLState : (
  (* states according to PackTags v3.0 *)
  ePMLState_UNDEFINED := 0,
  ePMLState_CLEARING  := 1,
  ePMLState_STOPPED   := 2,
  ePMLState_STARTING  := 3,
  ePMLState_IDLE      := 4,
  ePMLState_SUSPENDED := 5,

```

```

ePMLState_EXECUTE      := 6,
ePMLState_STOPPING    := 7,
ePMLState_ABORTING    := 8,
ePMLState_ABORTED     := 9,
ePMLState_HOLDING     := 10,
ePMLState_HELD        := 11,
ePMLState_UNHOLDING   := 12,
ePMLState_SUSPENDING  := 13,
ePMLState_UNSPENDING  := 14,
ePMLState_RESETTING   := 15,
ePMLState_COMPLETING  := 16,
ePMLState_COMPLETE    := 17
);
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.1.2 E_PMLUnitMode

E_PMLUnitMode

```

TYPE E_PMLUnitMode : (
  ePMLUnitMode_UNDEFINED      := 0,
  ePMLUnitMode_AUTOMATIC     := 1,
  ePMLUnitMode_MAINTENANCE   := 2,
  ePMLUnitMode_MANUAL        := 3,
  ePMLUnitMode_SEMIAUTOMATIC := 4,
  ePMLUnitMode_DRYRUN        := 5,
  ePMLUnitMode_USERMODE1     := 6,
  ePMLUnitMode_USERMODE2    := 7,
  ePMLUnitMode_IDLE          := 8,
  ePMLUnitMode_ESTOP         := 9
);
END_TYPE

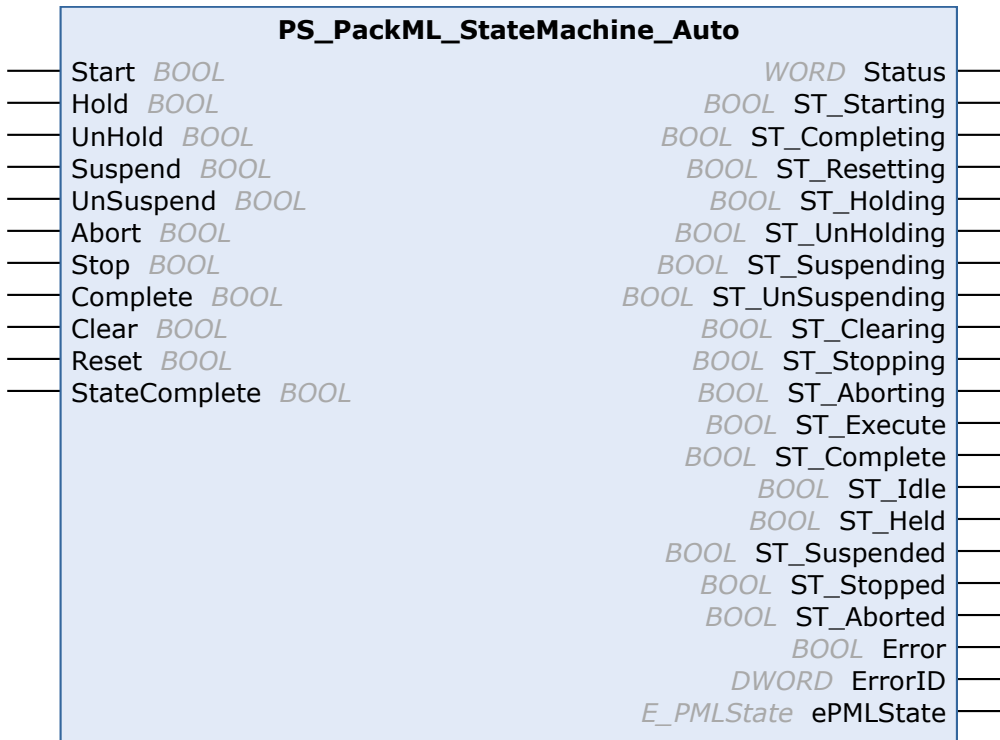
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

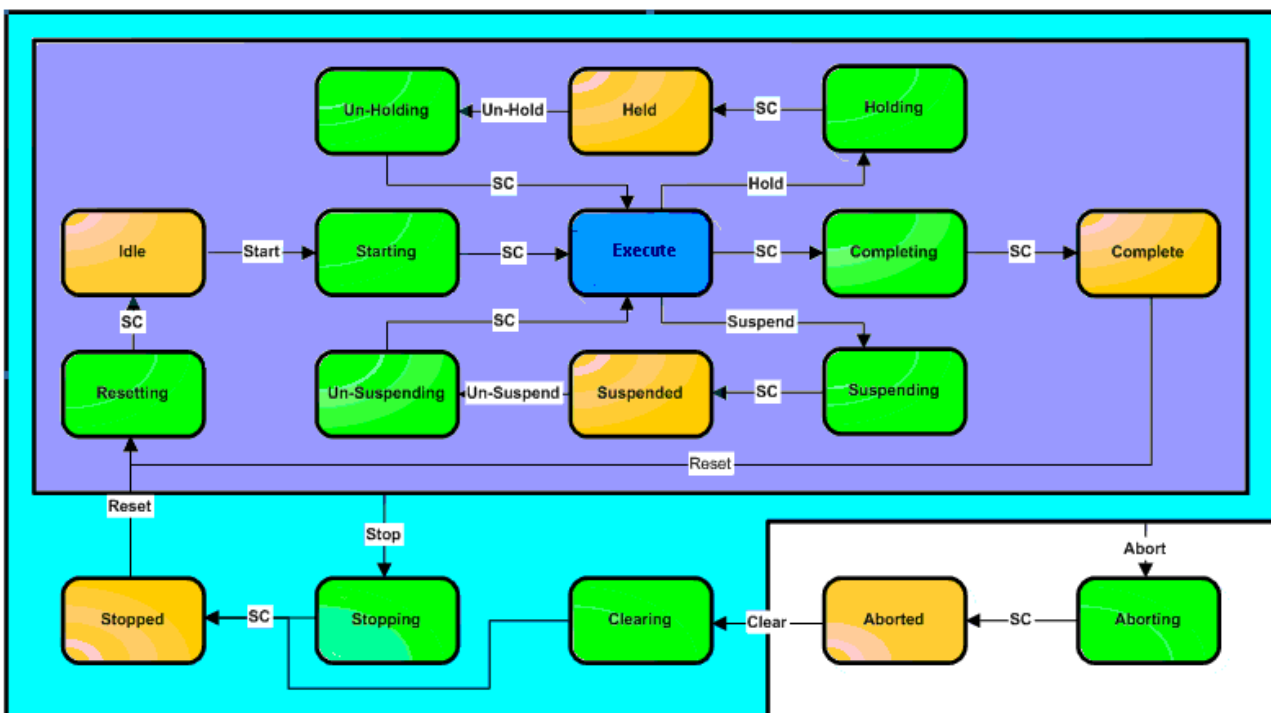
3.2 Funktionsbausteine

3.2.1 PS_PackML_StateMachine_Auto



Die Funktionsbausteine zum Packaging Machine State haben in der aktualisierten Form eine gemeinsame Schnittstelle zum PackML Machine State Model V3. Es wird erwartet, dass anwendungsspezifische Logik, wie Zustands-Übergänge, in externen Funktionsbausteinen programmiert ist und der Funktionsbaustein Pack_ML_State_Machine die zentrale Logik der Zustandsmaschine und die Zustandsdarstellung übernimmt. Deswegen gibt es für diesen Funktionsbaustein eine Empfehlung, wie er mit anderer Logik kombiniert werden kann.

Die Logik für Übergänge, insbesondere zwischen manuellem-, halbautomatischem- und automatischem Modus (siehe Abbildung), ist abhängig von der Anwendung.



Eingänge

```

VAR_INPUT
  Start          : BOOL;
  Hold           : BOOL;
  unHold         : BOOL;
  Suspend        : BOOL;
  unSuspend      : BOOL;
  Abort          : BOOL;
  Stop           : BOOL;
  Complete       : BOOL;
  Clear          : BOOL;
  Reset          : BOOL;
  StateComplete  : BOOL;
END_VAR

```

Tabelle/Beschreibung: Zustandsmaschine ab steigender Flanke ausführen...

Name	Typ	Beschreibung
Start	BOOL	...bis Starting.
Hold	BOOL	...bis Holding oder Held.
unHold	BOOL	...bis Unholding.
Suspend	BOOL	...bis Suspending oder Suspend.
unSuspend	BOOL	...bis Unsuspending.
Abort	BOOL	...bis Aborting.
Stop	BOOL	...bis Stopping.
Complete	BOOL	...bis Resetting.
Clear	BOOL	...bis Clearing.
Reset	BOOL	...bis Resetting.
StateComplete	BOOL	Transition

Ausgänge

```

VAR_OUTPUT
  Status          : WORD;
  ST_Starting     : BOOL;
  ST_Completing   : BOOL;
  ST_Resetting    : BOOL;
  ST_Holding      : BOOL;
  ST_UnHolding    : BOOL;
  ST_Suspending   : BOOL;
  ST_UnSuspending : BOOL;
  ST_Clearing     : BOOL;
  ST_Stopping     : BOOL;
  ST_Aborting     : BOOL;
  ST_Execute      : BOOL;
  ST_Complete     : BOOL;
  ST_Idle         : BOOL;
  ST_Held         : BOOL;
  ST_Suspended    : BOOL;
  ST_Stopped      : BOOL;
  ST_Aborted      : BOOL;
  Error           : BOOL;
  ErrorID         : UDINT;
  ePMLState       : E_PMLState;
END_VAR

```

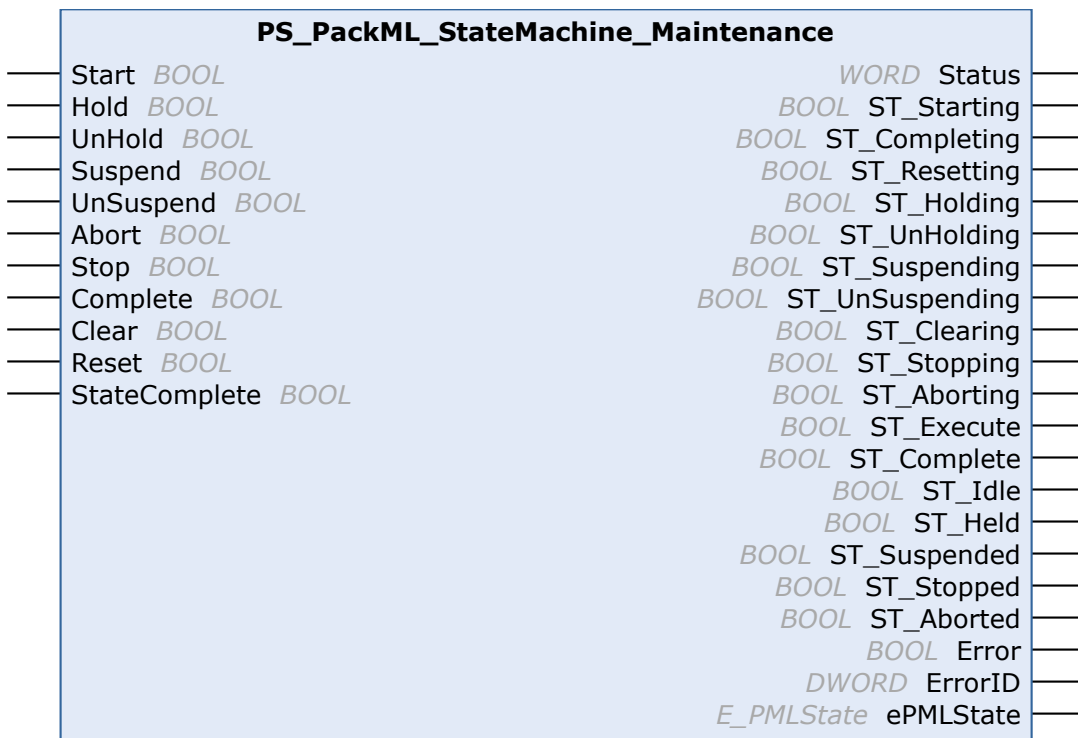
Name	Typ	Beschreibung
Status	WORD	Statuswort, das den Zustand der Zustandsmaschine repräsentiert.
ST_Starting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Starting.
ST_Completing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Completing.
ST_Resetting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Resetting.
ST_Holding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Holding.
ST_UnHolding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unholding.
ST_Suspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspending.

Name	Typ	Beschreibung
ST_UnSuspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unsuspending.
ST_Clearing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Clearing.
ST_Stopping	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopping.
ST_Aborting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborting.
ST_Execute	BOOL	Wahr, wenn Zustandsmaschine im Zustand Execute.
ST_Complete	BOOL	Wahr, wenn Zustandsmaschine im Zustand Complete.
ST_Idle	BOOL	Wahr, wenn Zustandsmaschine im Zustand Idle.
ST_Held	BOOL	Wahr, wenn Zustandsmaschine im Zustand Held.
ST_Suspended	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspended.
ST_Stopped	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopped.
ST_Aborted	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborted.
Error	BOOL	Wird TRUE, sobald ein Fehler auftritt.
ErrorID	UDINT	Wenn der Fehlerausgang gesetzt ist, liefert dieser Parameter die Fehlernummer.
ePMLState	E_PMLState	Aktueller PML-Zustand der automatischen Zustandsmaschine.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.2.2 PS_PackML_StateMachine_Maintenance



Die Funktionsbausteine zum Packaging Machine State haben in der aktualisierten Form eine gemeinsame Schnittstelle zum PackML Machine State Model V3. Es wird erwartet, dass anwendungsspezifische Logik, wie Zustands-Übergänge, in externen Funktionsbausteinen programmiert ist und der Funktionsbaustein Pack_ML_State_Machine die zentrale Logik der Zustandsmaschine und die Zustandsdarstellung übernimmt. Deswegen gibt es für diesen Funktionsbaustein eine Empfehlung, wie er mit anderer Logik kombiniert werden kann.

Die Logik für Übergänge, insbesondere zwischen manuellem-, halbautomatischem- und automatischem Modus (siehe Abbildung), ist abhängig von der Anwendung.

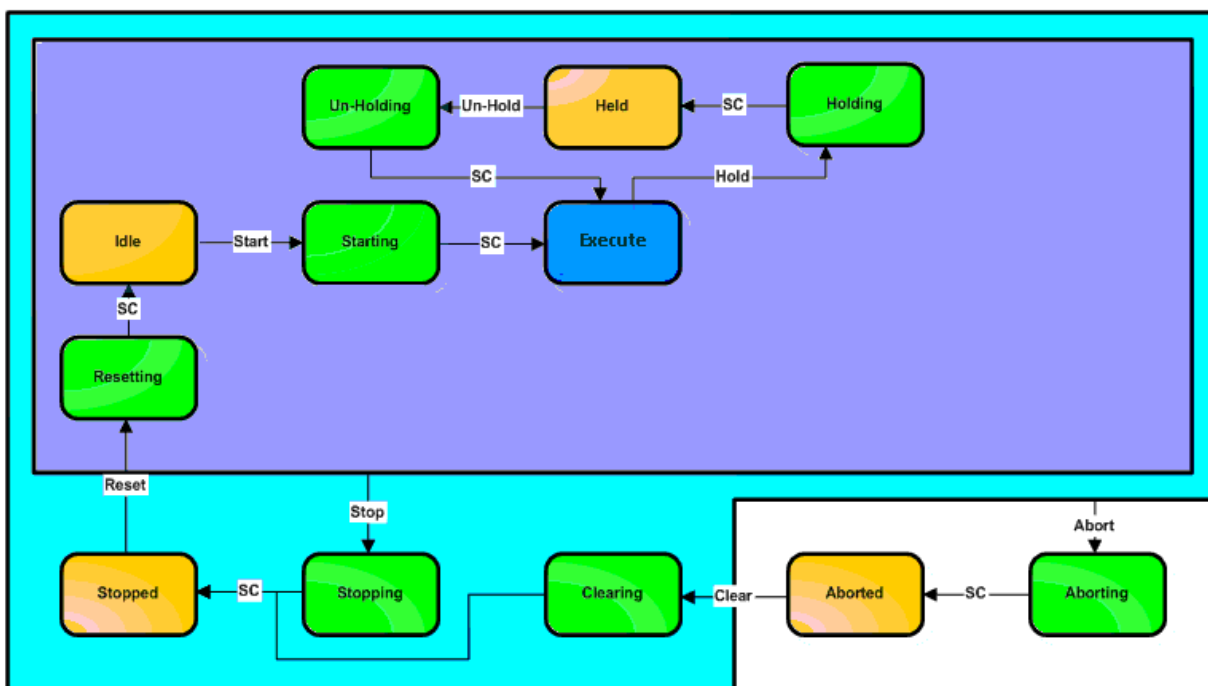


Figure : PS_Pack_ML_State_Model_Maintenance Mode

Eingänge

```

VAR_INPUT
  Start      : BOOL;
  Hold       : BOOL;
  unHold     : BOOL;
  Suspend    : BOOL;
  unSuspend  : BOOL;
  Abort      : BOOL;
  Stop       : BOOL;
  Complete   : BOOL;
  Clear      : BOOL;
  Reset      : BOOL;
  StateComplete : BOOL;
END_VAR
    
```

Tabelle/Beschreibung: Zustandsmaschine ab steigender Flanke ausführen...

Name	Typ	Beschreibung
Start	BOOL	...bis Starting.
Hold	BOOL	...bis Holding oder Held.
unHold	BOOL	...bis Unholding.
Suspend	BOOL	...bis Suspending oder Suspend.
unSuspend	BOOL	...bis Unsuspending.
Abort	BOOL	...bis Aborting.
Stop	BOOL	...bis Stopping.
Complete	BOOL	...bis Resetting.
Clear	BOOL	...bis Clearing.
Reset	BOOL	...bis Resetting.
StateComplete	BOOL	Transition

 **Ausgänge**

```

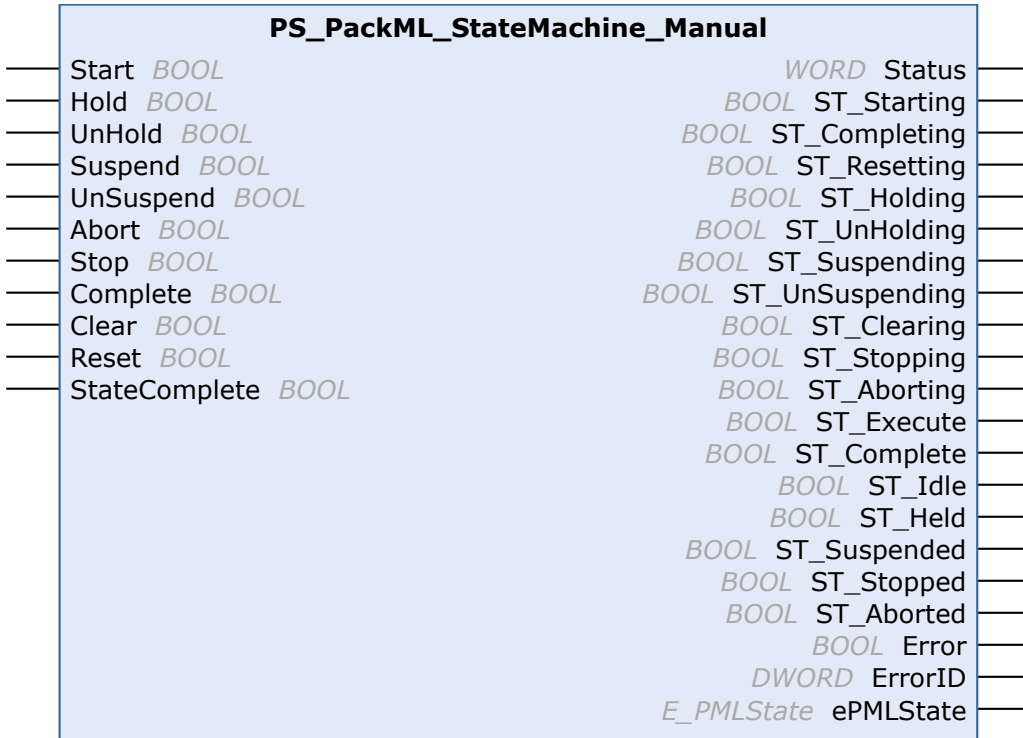
VAR_OUTPUT
    Status          : WORD;
    ST_Starting     : BOOL;
    ST_Completing   : BOOL;
    ST_Resetting    : BOOL;
    ST_Holding      : BOOL;
    ST_UnHolding    : BOOL;
    ST_Suspending   : BOOL;
    ST_UnSuspending : BOOL;
    ST_Clearing     : BOOL;
    ST_Stopping     : BOOL;
    ST_Aborting     : BOOL;
    ST_Execute      : BOOL;
    ST_Complete     : BOOL;
    ST_Idle         : BOOL;
    ST_Held         : BOOL;
    ST_Suspended    : BOOL;
    ST_Stopped     : BOOL;
    ST_Aborted      : BOOL;
    Error          : BOOL;
    ErrorID        : UDINT;
    ePMLState      : E_PMLState;
END_VAR
    
```

Name	Typ	Beschreibung
Status	WORD	Statuswort, das den Zustand der Zustandsmaschine repräsentiert.
ST_Starting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Starting.
ST_Completing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Completing.
ST_Resetting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Resetting.
ST_Holding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Holding.
ST_UnHolding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unholding.
ST_Suspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspending.
ST_UnSuspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unsuspending.
ST_Clearing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Clearing.
ST_Stopping	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopping.
ST_Aborting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborting.
ST_Execute	BOOL	Wahr, wenn Zustandsmaschine im Zustand Execute.
ST_Complete	BOOL	Wahr, wenn Zustandsmaschine im Zustand Complete.
ST_Idle	BOOL	Wahr, wenn Zustandsmaschine im Zustand Idle.
ST_Held	BOOL	Wahr, wenn Zustandsmaschine im Zustand Held.
ST_Suspended	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspended.
ST_Stopped	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopped.
ST_Aborted	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborted.
Error	BOOL	Wird TRUE, sobald ein Fehler auftritt.
ErrorID	UDINT	Wenn der Fehlerausgang gesetzt ist, liefert dieser Parameter die Fehlernummer.
ePMLState	E_PMLState	Aktueller PML-Zustand der automatischen Zustandsmaschine.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.2.3 PS_PackML_StateMachine_Manual



Die Funktionsbausteine Packaging Machine State haben in der aktualisierten Form eine gemeinsame Schnittstelle zum PackML Machine State Model V3. Es wird erwartet, dass anwendungsspezifische Logik, wie Zustands-Übergänge, in externen Funktionsbausteinen programmiert ist und der Funktionsbaustein Pack_ML_State_Machine die zentrale Logik der Zustandsmaschine und die Zustandsdarstellung übernimmt. Deswegen gibt es für diesen Funktionsbaustein eine Empfehlung, wie er mit anderer Logik kombiniert werden kann.

Die Logik für Übergänge, insbesondere zwischen manuellem-, halbautomatischem- und automatischem Modus (siehe Abbildung), ist abhängig von der Anwendung.

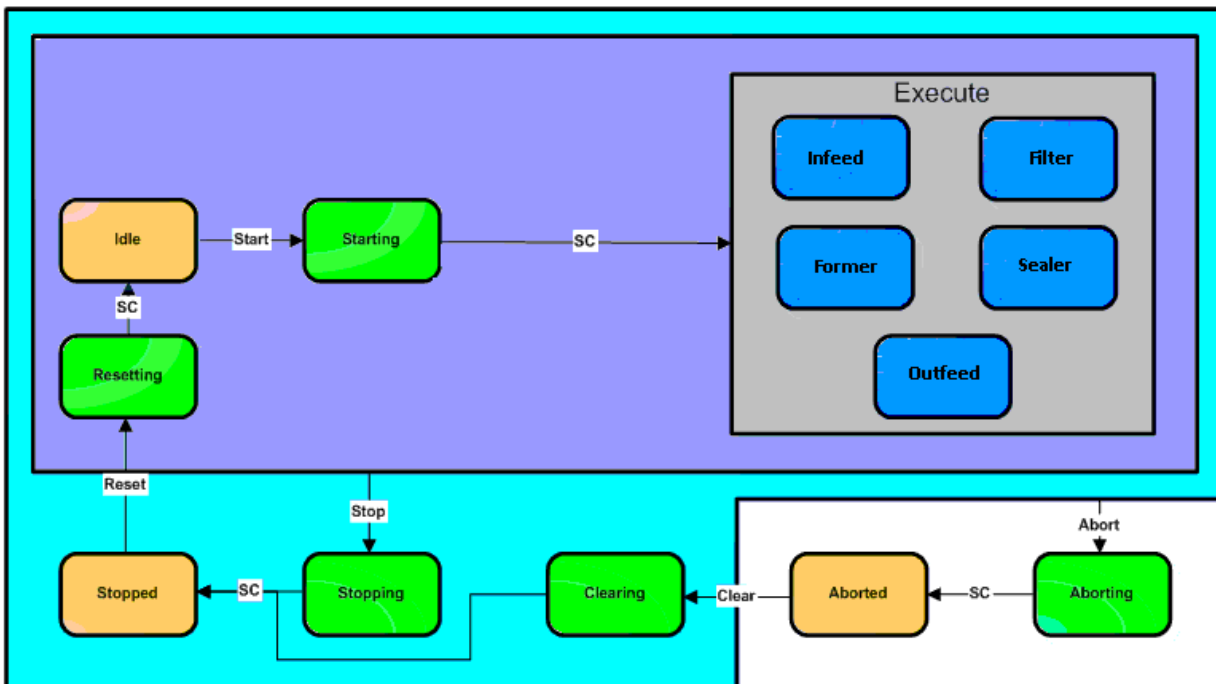


Figure : PS_Pack_ML_State_Model_Manual Mode

 **Eingänge**

```

VAR_INPUT
  Start          : BOOL;
  Hold           : BOOL;
  unHold         : BOOL;
  Suspend        : BOOL;
  unSuspend      : BOOL;
  Abort          : BOOL;
  Stop           : BOOL;
  Complete       : BOOL;
  Clear          : BOOL;
  Reset          : BOOL;
  StateComplete  : BOOL;
END_VAR
    
```

Tabelle/Beschreibung: Zustandsmaschine ab steigender Flanke ausführen...

Name	Typ	Beschreibung
Start	BOOL	...bis Starting.
Hold	BOOL	...bis Holding oder Held.
unHold	BOOL	...bis Unholding.
Suspend	BOOL	...bis Suspending oder Suspend.
unSuspend	BOOL	...bis Unsuspending.
Abort	BOOL	...bis Aborting.
Stop	BOOL	...bis Stopping.
Complete	BOOL	...bis Resetting.
Clear	BOOL	...bis Clearing.
Reset	BOOL	...bis Resetting.
StateComplete	BOOL	Transition

 **Ausgänge**

```

VAR_OUTPUT
  Status          : WORD;
  ST_Starting     : BOOL;
  ST_Completing   : BOOL;
  ST_Resetting    : BOOL;
  ST_Holding      : BOOL;
  ST_UnHolding    : BOOL;
  ST_Suspending   : BOOL;
  ST_UnSuspending : BOOL;
  ST_Clearing     : BOOL;
  ST_Stopping     : BOOL;
  ST_Aborting     : BOOL;
  ST_Execute      : BOOL;
  ST_Complete     : BOOL;
  ST_Idle         : BOOL;
  ST_Held         : BOOL;
  ST_Suspended    : BOOL;
  ST_Stopped      : BOOL;
  ST_Aborted      : BOOL;
  Error           : BOOL;
  ErrorID         : UDINT;
  ePMLState       : E_PMLState;
END_VAR
    
```

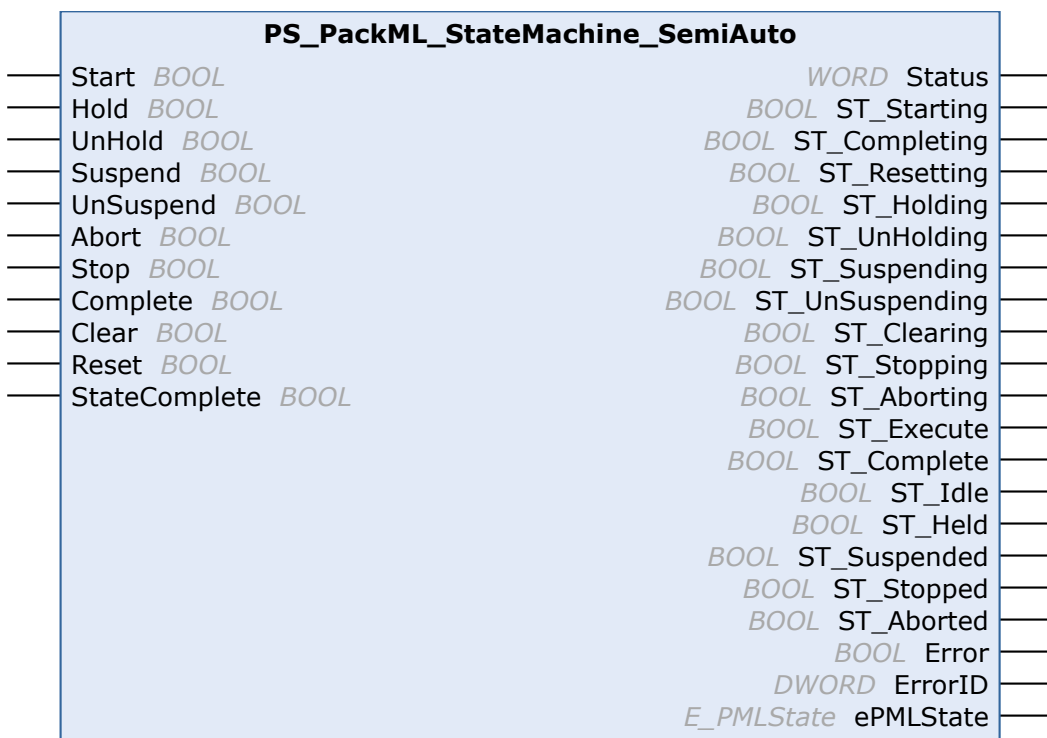
Name	Typ	Beschreibung
Status	WORD	Statuswort, das den Zustand der Zustandsmaschine repräsentiert.
ST_Starting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Starting.
ST_Completing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Completing.
ST_Resetting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Resetting.
ST_Holding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Holding.
ST_UnHolding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unholding.
ST_Suspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspending.

Name	Typ	Beschreibung
ST_UnSuspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unsuspending.
ST_Clearing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Clearing.
ST_Stopping	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopping.
ST_Aborting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborting.
ST_Execute	BOOL	Wahr, wenn Zustandsmaschine im Zustand Execute.
ST_Complete	BOOL	Wahr, wenn Zustandsmaschine im Zustand Complete.
ST_Idle	BOOL	Wahr, wenn Zustandsmaschine im Zustand Idle.
ST_Held	BOOL	Wahr, wenn Zustandsmaschine im Zustand Held.
ST_Suspended	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspended.
ST_Stopped	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopped.
ST_Aborted	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborted.
Error	BOOL	Wird TRUE, sobald ein Fehler auftritt.
ErrorID	UDINT	Wenn der Fehlerausgang gesetzt ist, liefert dieser Parameter die Fehlernummer.
ePMLState	E_PMLState	Aktueller PML-Zustand der automatischen Zustandsmaschine.

Voraussetzungen

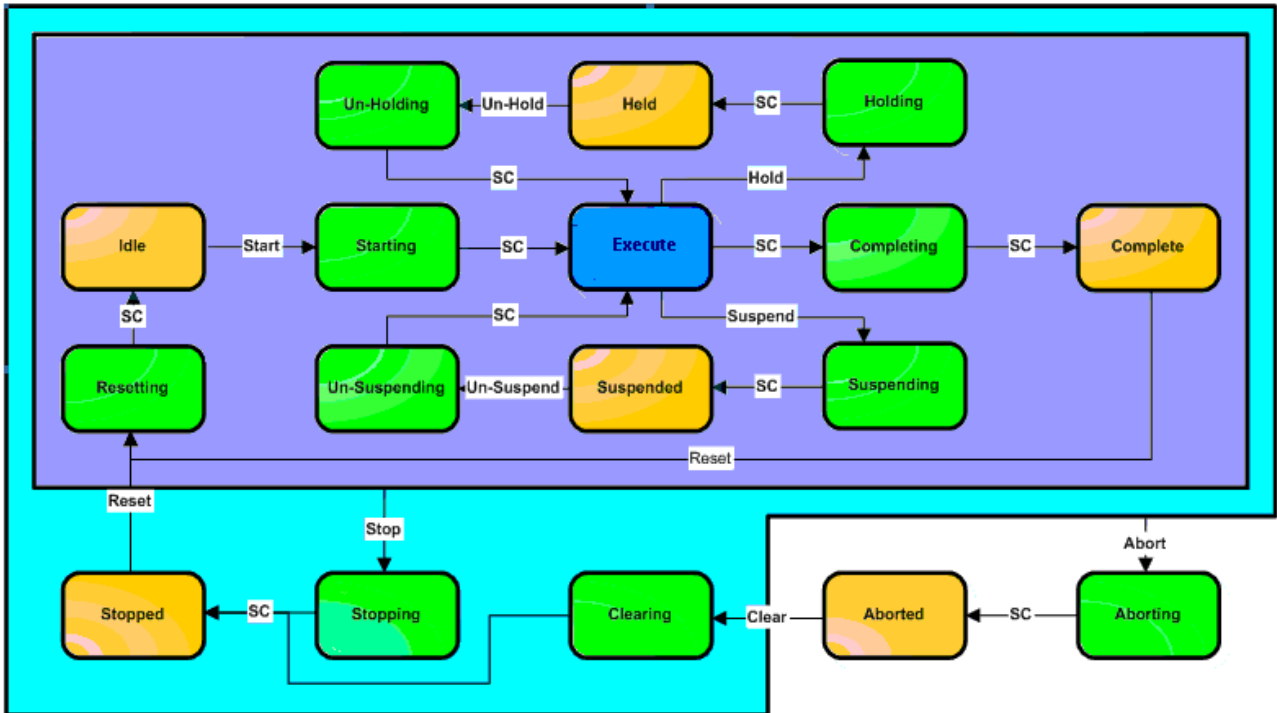
Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.2.4 PS_PackML_StateMachine_SemiAuto



Die Funktionsbausteine Packaging Machine State haben in der aktualisierten Form eine gemeinsame Schnittstelle zum PackML Machine State Model V3. Es wird erwartet, dass anwendungsspezifische Logik, wie Zustands-Übergänge, in externen Funktionsbausteinen programmiert ist und der Funktionsbaustein Pack_ML_State_Machine die zentrale Logik der Zustandsmaschine und die Zustandsdarstellung übernimmt. Deswegen gibt es für diesen Funktionsbaustein eine Empfehlung, wie er mit anderer Logik kombiniert werden kann.

Die Logik für Übergänge, insbesondere zwischen manuellem-, halbautomatischem- und automatischem Modus (siehe Abbildung), ist abhängig von der Anwendung.



Eingänge

```
VAR_INPUT
  Start      : BOOL;
  Hold      : BOOL;
  unHold    : BOOL;
  Suspend   : BOOL;
  unSuspend : BOOL;
  Abort     : BOOL;
  Stop      : BOOL;
  Complete  : BOOL;
  Clear     : BOOL;
  Reset     : BOOL;
  StateComplete : BOOL;
END_VAR
```

Tabelle/Beschreibung: Zustandsmaschine ab steigender Flanke ausführen...

Name	Typ	Beschreibung
Start	BOOL	...bis Starting.
Hold	BOOL	...bis Holding oder Held.
unHold	BOOL	...bis Unholding.
Suspend	BOOL	...bis Suspending oder Suspend.
unSuspend	BOOL	...bis Unsuspending.
Abort	BOOL	...bis Aborting.
Stop	BOOL	...bis Stopping.
Complete	BOOL	...bis Resetting.
Clear	BOOL	...bis Clearing.
Reset	BOOL	...bis Resetting.
StateComplete	BOOL	Transition

Ausgänge

```
VAR_OUTPUT
  Status      : WORD;
  ST_Starting : BOOL;
```

```

ST_Completing      : BOOL;
ST_Resetting       : BOOL;
ST_Holding         : BOOL;
ST_UnHolding       : BOOL;
ST_Suspending      : BOOL;
ST_UnSuspending    : BOOL;
ST_Clearing        : BOOL;
ST_Stopping        : BOOL;
ST_Aborting        : BOOL;
ST_Execute         : BOOL;
ST_Complete        : BOOL;
ST_Idle            : BOOL;
ST_Held            : BOOL;
ST_Suspended       : BOOL;
ST_Stopped         : BOOL;
ST_Aborted         : BOOL;
Error              : BOOL;
ErrorID            : UDINT;
ePMLState          : E_PMLState;

```

```
END_VAR
```

Name	Typ	Beschreibung
Status	WORD	Statuswort, das den Zustand der Zustandsmaschine repräsentiert.
ST_Starting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Starting.
ST_Completing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Completing.
ST_Resetting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Resetting.
ST_Holding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Holding.
ST_UnHolding	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unholding.
ST_Suspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspending.
ST_UnSuspending	BOOL	Wahr, wenn Zustandsmaschine im Zustand Unsuspending.
ST_Clearing	BOOL	Wahr, wenn Zustandsmaschine im Zustand Clearing.
ST_Stopping	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopping.
ST_Aborting	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborting.
ST_Execute	BOOL	Wahr, wenn Zustandsmaschine im Zustand Execute.
ST_Complete	BOOL	Wahr, wenn Zustandsmaschine im Zustand Complete.
ST_Idle	BOOL	Wahr, wenn Zustandsmaschine im Zustand Idle.
ST_Held	BOOL	Wahr, wenn Zustandsmaschine im Zustand Held.
ST_Suspended	BOOL	Wahr, wenn Zustandsmaschine im Zustand Suspended.
ST_Stopped	BOOL	Wahr, wenn Zustandsmaschine im Zustand Stopped.
ST_Aborted	BOOL	Wahr, wenn Zustandsmaschine im Zustand Aborted.
Error	BOOL	Wird TRUE, sobald ein Fehler auftritt.
ErrorID	UDINT	Wenn der Fehlerausgang gesetzt ist, liefert dieser Parameter die Fehlernummer.
ePMLState	E_PMLState	Aktueller PML-Zustand der automatischen Zustandsmaschine.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

3.2.5 PS_UnitModeManager



Verpackungsmaschinen haben verschiedene Anlagenmodi und nicht nur den Modus "automatisch". Jeder Anlagenmodus ist durch ein eigenes Zustandsmodell definiert. Für Übergänge zwischen den Modi muss ein „Mode Manager“ definiert werden. Der „Mode Manager“ legt fest, wie und in welchen Zustand eine Maschine Anlagenmodi ändern kann, d. h. eingebaute Sperren verhindern, dass die Maschine in ungeeignete Zustände wechselt.

Siehe Abbildung unten als Beispiel.

⚠️ WARNUNG

Ordnungsgemäße Modusänderungen einhalten

Die Logik für Übergänge zwischen den Modi ist abhängig von der Anwendung, insbesondere für Übergänge zwischen manuellem, halbautomatischem und automatischem Modus. Für solche Modusänderungen können darüber hinaus Hardware Sperren oder Sicherheitsausrüstung erforderlich sein. Die Verantwortung für ordnungsgemäße Moduswechsel liegt bei demjenigen, der sie implementiert.

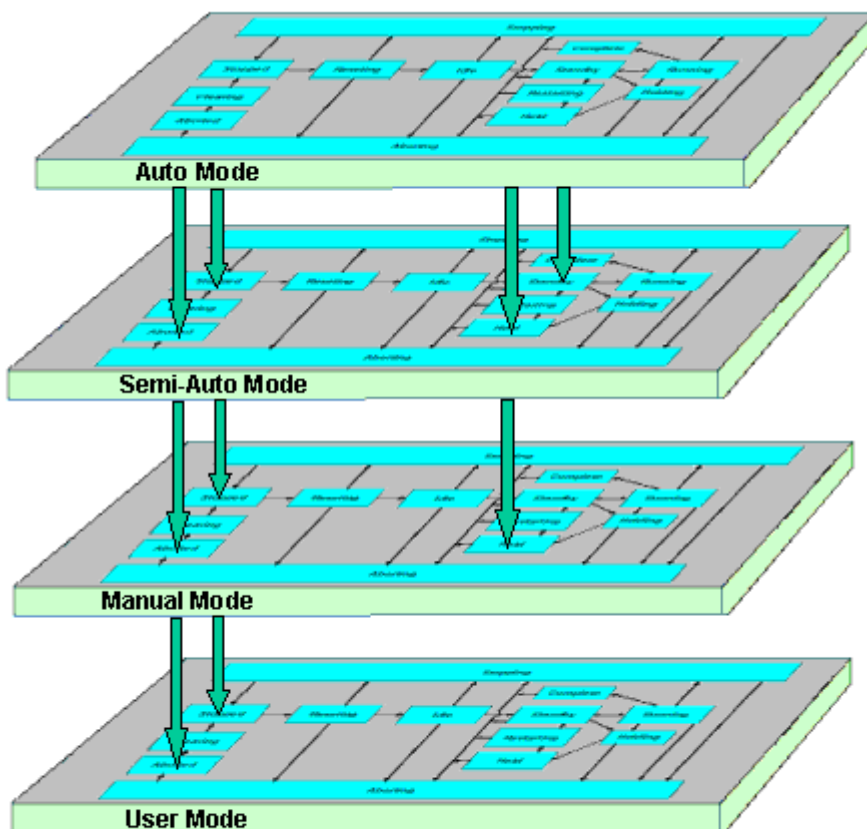


Figure : User Mode Operations State Model

Eingänge

```
VAR_INPUT
  Execute          : BOOL;
  eModeCommand    : E_PMLUnitMode;
  ePMLState       : E_PMLState;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Moduswechsel bei steigender Flanke.
eModeCommand	E_PMLUnitMode	Angeforderter Anlagenmodus (E_PMLUnitMode [► 10])
ePMLState	E_PMLState	Aktueller PML-Zustand des aktuellen Modus (E_PMLState [► 9])

Ein/Ausgänge

```
VAR_IN_OUT
  Machine_ID      : MACHINE_REF;    (* Identifies the axis which position shall be latched at the
  trigger event *)
END_VAR
```

Name	Typ	Beschreibung
Machine_ID	MACHINE_REF	Identifizierung der vom Zustandsmodell ausgeführten Maschine

Ausgänge

```
VAR_OUTPUT
  eModeStatus     : E_PMLUnitMode;
  Done            : BOOL;
  Error           : BOOL;
  ErrorID        : UDINT;
END_VAR
```

Name	Typ	Beschreibung
eModeStatus	E_PMLUnitMode	Aktueller Anlagenmodus (E_PMLUnitMode [► 10])
Done	BOOL	Wahr, wenn Moduswechsel erfolgreich war.
Error	BOOL	Signalisiert einen Fehler im Funktionsbaustein, z. B. "Moduswechsel unzulässig".
ErrorID	UDINT	Wenn der Fehlerausgang gesetzt ist, liefert dieser Parameter die Fehlernummer. 0 = kein Fehler; 1 = Moduswechsel unzulässig. ePMLState nicht idle, stopped, aborted, held, suspended oder completed oder der entsprechende Zustand existiert im angeforderten Modus nicht.

Realisierung

Der Moduswechsel ist auf bestimmte Modi begrenzt, siehe Implementierung unten.



Es sind noch nicht alle Modi implementiert.

```
rTrig(CLK:= Execute);
IF rTrig.Q THEN
  Done := FALSE;
  Error := FALSE;
  ErrorID := 0;
  CASE eModeStatus OF
    ePMLUnitMode_AUTOMATIC:
      IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
```

```

        Done := TRUE;
    ELSIF ((ePMLState = ePMLState_SUSPENDED) OR (ePMLState = ePMLState_HELD) OR (ePMLState = ePMLState_COMPLETE))
    AND (eModeCommand = ePMLUnitMode_SEMIAUTOMATIC) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSIF (ePMLState = ePMLState_HELD) AND (eModeCommand = ePMLUnitMode_MAINTENANCE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ePMLUnitMode_MAINTENANCE:
    IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSIF (ePMLState = ePMLState_HELD) AND ((eModeCommand = ePMLUnitMode_AUTOMATIC) OR (eModeCommand = ePMLUnitMode_SEMIAUTOMATIC)) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ePMLUnitMode_MANUAL:
    IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ePMLUnitMode_SEMIAUTOMATIC:
    IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSIF ((ePMLState = ePMLState_SUSPENDED) OR (ePMLState = ePMLState_HELD) OR (ePMLState = ePMLState_COMPLETE))
    AND (eModeCommand = ePMLUnitMode_AUTOMATIC) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSIF (ePMLState = ePMLState_HELD) AND (eModeCommand = ePMLUnitMode_MAINTENANCE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ePMLUnitMode_IDLE:
    IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ePMLUnitMode_ESTOP:
    IF (ePMLState = ePMLState_STOPPED) OR (ePMLState = ePMLState_ABORTED) OR (ePMLState = ePMLState_IDLE) THEN
        eModeStatus := eModeCommand;
        Done := TRUE;
    ELSE
        Error := TRUE;
        ErrorID := 1;
    END_IF
ELSE
    eModeStatus := eModeCommand;
    Done := TRUE;
END_CASE
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4 Packaging Machine Tags

4.1 Einleitung

PackTags stellen einen einheitlichen Satz Namenskonventionen für Datenelemente bereit, die in den prozeduralen Elementen des Basiszustandsmodells verwendet werden. Wie weiter oben im Dokument beschrieben stellt das Base State Model (Basiszustandsmodell) einen einheitlichen Satz Maschinentypen bereit, so dass alle automatisierten Maschinen gleich betrachtet werden können. PackTags sind mit Namen versehene Datenelemente für interoperablen Datenaustausch bei automatisierten Maschinen offener Architekturen. In diesem Dokument finden sich die wesentlichen Namen der Datenelemente, Datentyp, Werte, Bereiche und gegebenenfalls Datenstrukturen. PackTags werden für die Maschine-zu-Maschine-Kommunikation eingesetzt, z.B. zwischen einem Abfüller und einem Kappenaufsetzer. PackTags können auch für den Datenaustausch zwischen Maschine und übergeordnetem Informationssystem wie Manufacturing Operations Management und Enterprise Information Systemen eingesetzt werden.

Dieses Dokument definiert alle PackTags für die Navigation durch ein Zustandsmodell und für die Definition und Betätigung des Anlagensteuerungsmodus. Desweiteren definiert dieses Dokument auch eine Liste von PackTags, die eventuelle wichtige Informationen einer Maschine bereitstellen. Alle PackTags müssen genutzt werden, um den Prinzipien integrierter Konnektivität mit Systemen mit der gleichen Implementierungsmethode zu entsprechen.

Notwendig sind die Tags, die für die Funktion der automatisierten Maschine oder die Konnektivität zu Kontroll- oder Fernsystemen benötigt werden.

4.2 Tag-Arten

PackTags werden in drei Gruppen aufgegliedert: Command (Befehl), Status (Zustand) und Administration (Verwaltung). Befehls- und Zustands-Tags enthalten Daten für die Anbindung der Maschine an die Liniensteuerung zur Koordination oder zum Herunterladen von Rezepten/Parametern. Befehlstags werden als „Informationsempfänger“ an das Maschinenprogramm „übergeben“ und von ihm konsumiert, Zustands-Tags werden vom Maschinenprogramm erzeugt und ausgelesen. Verwaltungs-Tags enthalten Daten, die übergeordnete Systeme zur Analyse der Maschinenleistung oder zur Information der Bediener sammeln.

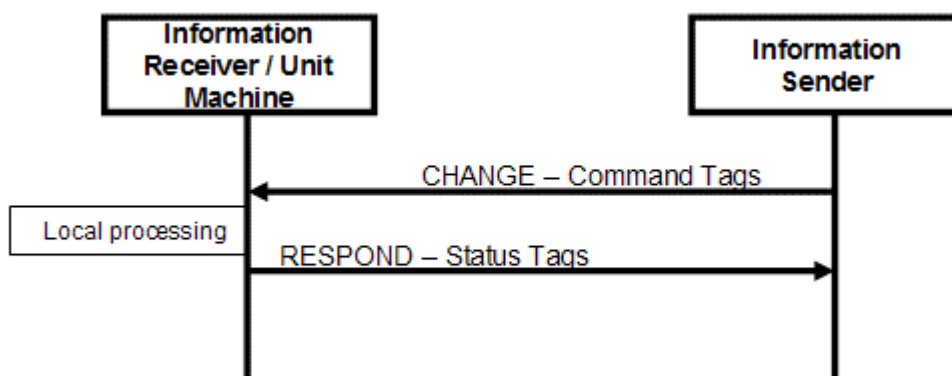
Die Gruppierung von Daten sollte in benachbarten Registern erfolgen, um die Kommunikation zu optimieren.

Normalerweise werden Informationsdaten via OPC in einem Ethernet-basierten Kommunikationsnetzwerk übertragen.

Präfix von Befehls-Tags ist „PMLc“.

Präfix von Zustands-Tags ist „PMLs“.

Präfix von Verwaltungs-Tags ist „PMLa“.



4.3 Tag-Details

Der folgende Abschnitt gibt einen Überblick über die Tags. In den nachfolgenden Tabellen sind Befehls-, Zustands- und Verwaltungs-PackTags aufgelistet.

Befehlsstruktur PMLc

		Tag Name	Data Type
PMLc		PMLc	ST_PMLc
	UnitMode	PMLc.UnitMode	DINT
	UnitModeChangeRequest	PMLc.UnitModeChangeRequest	BOOL
	ProcMode	PMLc.ProcMode	DINT
	ProcModeChangeRequest	PMLc.ProcModeChangeRequest	BOOL
	CurMachSpeed	PMLc.CurMachSpeed	DINT
	MatReady	PMLc.MatReady	ST_Material
	RawMaterial	PMLc.MatReady.RawMaterial	BOOL
	CO2	PMLc.MatReady.CO2	BOOL
	Container	PMLc.MatReady.Container	BOOL
	Lubrication	PMLc.MatReady.Lubrication	BOOL
	Water	PMLc.MatReady.Water	BOOL
	ContainerClosures	PMLc.MatReady.ContainerClosures	BOOL
	Unused0	PMLc.MatReady.Unused0	BOOL
	Unused1	PMLc.MatReady.Unused1	BOOL
	Unused2	PMLc.MatReady.Unused2	BOOL
	Unused3	PMLc.MatReady.Unused3	BOOL
	Unused4	PMLc.MatReady.Unused4	BOOL
	Unused5	PMLc.MatReady.Unused5	BOOL
	Unused6	PMLc.MatReady.Unused6	BOOL
	Unused7	PMLc.MatReady.Unused7	BOOL
	Unused8	PMLc.MatReady.Unused8	BOOL
	Unused9	PMLc.MatReady.Unused9	BOOL
	MatLow	PMLc.MatLow	ST_Material
	RawMaterial	PMLc.MatLow.RawMaterial	BOOL
	CO2	PMLc.MatLow.CO2	BOOL
	Container	PMLc.MatLow.Container	BOOL
	Lubrication	PMLc.MatLow.Lubrication	BOOL
	Water	PMLc.MatLow.Water	BOOL
	ContainerClosures	PMLc.MatLow.ContainerClosures	BOOL
	Unused0	PMLc.MatLow.Unused0	BOOL
	Unused1	PMLc.MatLow.Unused1	BOOL
	Unused2	PMLc.MatLow.Unused2	BOOL
	Unused3	PMLc.MatLow.Unused3	BOOL
	Unused4	PMLc.MatLow.Unused4	BOOL
	Unused5	PMLc.MatLow.Unused5	BOOL
	Unused6	PMLc.MatLow.Unused6	BOOL
	Unused7	PMLc.MatLow.Unused7	BOOL
	Unused8	PMLc.MatLow.Unused8	BOOL
	Unused9	PMLc.MatLow.Unused9	BOOL
	State	PMLc.State	DINT
	StateChangeRequest	PMLc.StateChangeRequest	BOOL
	CntrlCmd	PMLc.CntrlCmd	DINT
	Node[#]	PMLc.Node[#]	ST_Node
	Number	PMLc.Node[#].Number	DINT
	ControlCmdNumber	PMLc.Node[#].ControlCmdNumber	DINT
	CmdValue	PMLc.Node[#].CmdValue	DINT
	Parameter[#]	PMLc.Node[#].Parameter[#]	ST_Descriptor
	Id	PMLc.Node[#].Parameter[#].Id	DINT
	Name	PMLc.Node[#].Parameter[#].Name	STRING

	Unit	PMLc.Node[#].Parameter[#].Unit	STRING
	Value	PMLc.Node[#].Parameter[#].Value	REAL
ProcessVariables[#]		PMLc.ProcessVariables[#]	ST_Descriptor
	Id	PMLc.ProcessVariables[#].Id	DINT
	Name	PMLc.ProcessVariables[#].Name	STRING
	Unit	PMLc.ProcessVariables[#].Unit	STRING
	Value	PMLc.ProcessVariables[#].Values	REAL
Product[#]		PMLc.Product[#]	ST_Product
	ProductId	PMLc.Product[#].ProductId	DINT
	ProcessVariables[#]	PMLc.Product[#].ProcessVariables[#]	ST_Descriptor
	Id	PMLc.Product[#].ProcessVariables[#].Id	DINT
	Name	PMLc.Product[#].ProcessVariables[#].Name	STRING
	Unit	PMLc.Product[#].ProcessVariables[#].Unit	STRING
	Value	PMLc.Product[#].ProcessVariables[#].Value	REAL
	Ingredients[#]	PMLc.Product[#].Ingredients[#]	ST_Ingredient
	IngredientId	PMLc.Product[#].Ingredients[#].IngredientId	DINT
	Parameter[#]	PMLc.Product[#].Ingredients[#].Parameter[#]	ST_Descriptor
	Id	PMLc.Product[#].Ingredients[#].Parameter[#].Id	DINT
	Name	PMLc.Product[#].Ingredients[#].Parameter[#].Name	STRING
	Unit	PMLc.Product[#].Ingredients[#].Parameter[#].Unit	STRING
	Value	PMLc.Product[#].Ingredients[#].Parameter[#].Value	REAL
Limits[#]		PMLc.Limits[#]	ST_Descriptor
	Id	PMLc.Limits[#].Id	DINT
	Name	PMLc.Limits[#].Name	STRING
	Unit	PMLc.Limits[#].Unit	STRING
	Value	PMLc.Limits[#].Value	REAL
TargetDownstreamNodeID		PMLc.TargetDownstreamNodeID	DINT
TargetUpstreamNodeID		PMLc.TargetUpstreamNodeID	DINT
ChangeNodeServicedUpstream		PMLc.ChangeNodeServicedUpstream	DINT
ChangeNodeServicedDownstream		PMLc.ChangeNodeServicedDownstream	DINT

Zustandsstruktur PMLs

	Tag Name	Data Type
PMLs	PMLs	ST_PMLs
CommandRejected	PMLs.CommandRejected	BOOL
UnitModeCurrent	PMLs.UnitModeCurrent	DINT
UnitModeRequested	PMLs.UnitModerequested	DINT
UnitModeChangeInProcesses	PMLs.UnitModeChangeInProcess	BOOL
ProcModeCurrent	PMLs.ProcModeCurrent	DINT
ProcModeRequested	PMLs.ProcModeRequested	DINT
ProcModeChangeInProcesses	PMLs.ProcModeChangeInProcess	BOOL
StateCurrent	PMLs.StateCurrent	DINT
StateRequested	PMLs.StateRequested	DINT
StateChangeInProcess	PMLs.StateChangeInProcess	BOOL
StateChangeProgress	StatusStateChangeProgress	DINT
StateLastCompleted	PMLs.StateLastCompleted	DINT
SeqNumber	PMLs.SeqNumber	DINT
CurMachSpd	PMLs.CurMachSpd	DINT
MatReady	PMLs.MatReady	ST_Material
RawMaterial	PMLs.MatReady.RawMaterial	BOOL

	CO2	PMLs.MatReady.CO2	BOOL
	Container	PMLs.MatReady.Container	BOOL
	Lubrication	PMLs.MatReady.Lubrication	BOOL
	Water	PMLs.MatReady.Water	BOOL
	ContainerClosures	PMLs.MatReady.ContainerClosures	BOOL
	Unused0	PMLs.MatReady.Unused0	BOOL
	Unused1	PMLs.MatReady.Unused1	BOOL
	Unused2	PMLs.MatReady.Unused2	BOOL
	Unused3	PMLs.MatReady.Unused3	BOOL
	Unused4	PMLs.MatReady.Unused4	BOOL
	Unused5	PMLs.MatReady.Unused5	BOOL
	Unused6	PMLs.MatReady.Unused6	BOOL
	Unused7	PMLs.MatReady.Unused7	BOOL
	Unused8	PMLs.MatReady.Unused8	BOOL
	Unused9	PMLs.MatReady.Unused9	BOOL
MatLow		PMLs.MatLow	ST_Material
	RawMaterial	PMLs.MatLow.RawMaterial	BOOL
	CO2	PMLs.MatLow.CO2	BOOL
	Container	PMLs.MatLow.Container	BOOL
	Lubrication	PMLs.MatLow.Lubrication	BOOL
	Water	PMLs.MatLow.Water	BOOL
	ContainerClosures	PMLs.MatLow.ContainerClosures	BOOL
	Unused0	PMLs.MatLow.Unused0	BOOL
	Unused1	PMLs.MatLow.Unused1	BOOL
	Unused2	PMLs.MatLow.Unused2	BOOL
	Unused3	PMLs.MatLow.Unused3	BOOL
	Unused4	PMLs.MatLow.Unused4	BOOL
	Unused5	PMLs.MatLow.Unused5	BOOL
	Unused6	PMLs.MatLow.Unused6	BOOL
	Unused7	PMLs.MatLow.Unused7	BOOL
	Unused8	PMLs.MatLow.Unused8	BOOL
	Unused9	PMLs.MatLow.Unused9	BOOL
MachDesignSpeed		PMLs.MachDesignSpeed	REAL
MachCycle		PMLs.MachCycle	DINT
ProdRatio		PMLs.ProdRatio	DINT
Dirty		PMLs.Dirty	BOOL
Clean		PMLs.Clean	BOOL
TimeToDirty		PMLs.TimeToDirty	DINT
EquipmentAllocatedToUnitModelID		PMLs.EquipmentAllocatedToUnitModelID	DINT
MachineReusableForUnitModelID		PMLs.MachineReusableForUnitModelID	DINT
MachineReusableTimeLeft		PMLs.MachineReusableTimeLeft	DINT
MachineStoringProductID		PMLs.MachineStoringProductID	DINT
MachineTransferringProductID		PMLs.MachineTransferringProductID	DINT
Node[#]		PMLs.Node[#]	ST_Node
	Number	PMLs.Node[#].Number	DINT
	ControlCmdNumber	PMLs.Node[#].ControlCmdNumber	DINT
	CmdValue	PMLs.Node[#].CmdValue	DINT
	Parameter[#]	PMLs.Node[#].Parameter[#]	ST_Descriptor
	Id	PMLs.Node[#].Parameter[#].Id	DINT
	Name	PMLs.Node[#].Parameter[#].Name	STRING
	Unit	PMLs.Node[#].Parameter[#].Unit	STRING
	Value	PMLs.Node[#].Parameter[#].Value	REAL
ProcessVariables[#]		PMLs.ProcessVariables[#]	ST_Descriptor
	Id	PMLs.ProcessVariables[#].Id	DINT
	Name	PMLs.ProcessVariables[#].Name	STRING

Product[#]	Unit	PMLs.ProcessVariables[#].Unit	STRING
	Value	PMLs.ProcessVariables[#].Value	REAL
	ProductId	PMLs.Product[#]	ST_Product
	ProcessVariables[#]	PMLs.Product[#].ProductId	DINT
		PMLs.Product[#].ProcessVariables[#]	ST_Descriptor
	Id	PMLs.Product[#].ProcessVariables[#].Id	DINT
	Name	PMLs.Product[#].ProcessVariables[#].Name	STRING
	Unit	PMLs.Product[#].ProcessVariables[#].Unit	STRING
	Value	PMLs.Product[#].ProcessVariables[#].Value	REAL
	Ingredients[#]	PMLs.Product[#].Ingredients[#]	ST_Inredient
	IngredientId	PMLs.Product[#].Ingredients[#].IngredientId	DINT
	Parameter[#]	PMLs.Product[#].Ingredients[#].Parameter[#]	ST_Descriptor
	Id	PMLs.Product[#].Ingredients[#].Parameter[#].Id	DINT
	Name	PMLs.Product[#].Ingredients[#].Parameter[#].Name	STRING
Unit	PMLs.Product[#].Ingredients[#].Parameter[#].Unit	STRING	
Value	PMLs.Product[#].Ingredients[#].Parameter[#].Value	REAL	
Limits[#]		PMLs.Limits[#]	ST_Descriptor
	Id	PMLs.Limits[#].Id	DINT
	Name	PMLs.Limits[#].Name	STRING
	Einheit	PMLs.Limits[#].Unit	STRING
	Value	PMLs.Limits[#].Value	REAL

Verwaltungsstruktur PMLa

		Tag Name	Data Type
PMLa		Admin	ST_PMLa
Alarm[#]		PMLa.Alarm[#]	ST_Alarm
	Id	PMLa.Alarm[#].Id	DINT
	Value	PMLa.Alarm[#].Value	DINT
	Message	PMLa.Alarm[#].Message	STRING
	TimeEvent	PMLa.Alarm[#].TimeEvent	ST_TimeStamp
	Year	PMLa.Alarm[#].TimeEvent.Year	DINT
	Month	PMLa.Alarm[#].TimeEvent.Month	DINT
	Day	PMLa.Alarm[#].TimeEvent.Day	DINT
	Hour	PMLa.Alarm[#].TimeEvent.Hour	DINT
	Minute	PMLa.Alarm[#].TimeEvent.Minute	DINT
	Second	PMLa.Alarm[#].TimeEvent.Second	DINT
	mSec	PMLa.Alarm[#].TimeEvent.mSec	DINT
	TimeAck	PMLa.Alarm[#].TimeAck	ST_TimeStamp
	Year	PMLa.Alarm[#].TimeAck.Year	DINT
	Month	PMLa.Alarm[#].TimeAck.Month	DINT
	Day	PMLa.Alarm[#].TimeAck.Day	DINT
	Hour	PMLa.Alarm[#].TimeAck.Hour	DINT
	Minute	PMLa.Alarm[#].TimeAck.Minute	DINT
	Second	PMLa.Alarm[#].TimeAck.Second	DINT
	mSec	PMLa.Alarm[#].TimeAck.mSec	DINT
	ModeCurrentTime[#]	PMLa.Alarm[#].ModeCurrentTime[#]	DINT
	ModeCummulativeTime[#]	PMLa.Alarm[#].ModeCummulativeTime[#]	DINT
	StateCurrentTime[#, #]	PMLa.Alarm[#].StateCurrentTime[#, #]	DINT
	StateCummulativeTime[#, #]	PMLa.Alarm[#].StateCummulativeTime[#, #]	DINT
	ProdProcessed	PMLa.Alarm[#].ProdProcessed	DINT
	DefectProd	PMLa.Alarm[#].DefectProd	DINT
	ReworkProd	PMLa.Alarm[#].ReworkProd	DINT

ResetTimersCounters	PMLa.Alarm[#].ResetTimersCounters	DINT
UpstreamMessage	PMLa.Alarm[#].UpstreamMessage	DINT
DownstreamMessage	PMLa.Alarm[#].DownStreamMessage	DINT
CurrentDownstreamNodeID[#]	PMLa.Alarm[#].CurrentDownstreamNodeID[#]	DINT
CurrentUpstreamNodeID[#]	PMLa.Alarm[#].CurrentUpstreamNodeID[#]	DINT

4.4 Datentypen

4.4.1 Alarm

4.4.1.1 E_AlarmID

Der Datentyp E_AlarmID definiert die Fehlerursache.

```

TYPE E_AlarmID : (
  eAID_Undefined                := 0,

  (** 1..32 Machine internal reason - safeties - OMAC defined **)
  eAID_EStop_Pushed             := 1,
  eAID_PerimeterProtectionFault,
  eAID_MainsTurnedOff,
  eAID_SafetyGateOrGuardDoorOpen,
  (* 5..32 reserved for future OMAC safety codes *)

  (** 33..64 Machine internal reason - operator actions - OMAC defined *)
  eAID_CycleStopButtonPushed   := 33,
  eAID_StartButtonPushed,
  eAID_ResetButtonPushed,
  eAID_JogModeSelected,
  eAID_AutomaticModeSelected,
  eAID_ManualModeSelected,
  eAID_SemiAutomaticModeSelected,
  (* 40..64 reserved for future OMAC defined operator action codes *)

  (** 65..256 Machine internal reason - internal machine faults - product related - OMAC defined *)
  eAID_MaterialJam              := 65,
  (* 66..256 reserved for future OMAC defined internal material related codes *)

  (** 257..512 Machine internal reason - internal machine faults - machine related - OMAC defined *)
  eAID_MachineJam               := 257,
  eAID_ElectricalOverload,
  eAID_MechanicalOverload,
  eAID_DriveFault,
  eAID_DriveFailure,
  eAID_ServoAxisFault,
  eAID_ServoAxisFailure,
  eAID_CommunicationError,
  eAID_PlcErrorCode,
  eAID_Vacuum,
  eAID_AirPressure,
  eAID_Voltage,
  eAID_Temperature,
  eAID_HydraulicPressure,
  eAID_HydraulicLevel,
  eAID_HydraulicTemperature,
  (* 273..512 reserved for future OMAC defined internal machine related codes *)

  (** 513..999 Machine internal reason - general information - OMAC defined *)
  eAID_CounterPresetReached     := 513,
  eAID_ProductSelected,
  eAID_LocalSlowSpeedRequested,
  eAID_LocalMediumSpeedRequested,
  eAID_LocalHighSpeedRequested,
  eAID_LocalSurgeSpeedRequested,
  eAID_RemoteSpeedRequested,
  eAID_DriveWarning,
  eAID_ServoWarning,
  (* 522..998 reserved for future OMAC defined general information related codes *)

```

```
eAID_CatchAllUndefinedInternalReason      := 999,

(** 1000..1999 Machine internal reason - vendor defined *)
(* 1000..1999 vendor defined area for machine internal items *)

(** 2000..2499 Machine upstream process reason - OMAC defined *)
eAID_InfeedNotTurnedOn                   := 2000,
eAID_InfeedOverload,
eAID_LowPrimeMaterial,
eAID_HighPrimeMaterial,
(* 2004..2498 reserved for future OMAC defined upstream reason *)
eAID_CatchAllUndefinedUpstreamReason     := 2499,

(** 2500..2999 Machine upstream process reason - vendor defined *)
(* 2500..2999 vendor defined area for upstream items *)

(** 3000..3499 Machine downstream process reason - OMAC defined *)
eAID_DischargeNotTurnedOn                 := 3000,
eAID_DischargeOverload,
eAID_DischargeBlockedReason,
eAID_DischargeCycleStopReason,
eAID_DischargeImmediateStopReason,
(* 3004..3498 reserved for future OMAC defined downstream reason *)
eAID_CatchAllUndefinedDownstreamReason   := 3499,

(** 3500..3999 Machine upstream process reason - vendor defined *)
(* 3500..3999 vendor defined area for downstream items *)

(** 4000..4499 out of service - OMAC defined *)
eAID_LineNotScheduled                     := 4000,
eAID_PlannedMaintenance,
eAID_MealsAndRest,
eAID_Meetings,
eAID_Training,
eAID_NoPackagingMaterials,
eAID_RemoteStopRequested,
eAID_MachineNotSelected,
eAID_Changeover,
eAID_Lubrication,
eAID_ProductCountPresetReached,
eAID_SetupSelected,
eAID_NoIncommingProduct,
eAID_WaitingForElectrialService,
eAID_WaitingForMechanicalService
(* 4012..4499 reserved for future OMAC defined downstream reason *)

(** 4500..4999 out of service - vendor defined *)
(* 4500..4999 vendor defined area for out of service items *)
);
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.1.2 ST_Alarm

Sammlung der Tags für die Beschreibung von Alarmereignissen.

```
TYPE ST_Alarm :
STRUCT
  Id           : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Value       : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Message     : STRING; (* ~ (OPC : 1: enabled for OPC ) *)
  TimeEvent   : ST_TimeStamp; (* ~ (OPC : 1: enabled for OPC ) *)
  TimeAck     : ST_TimeStamp; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.1.3 ST_TimeStamp

Diese Struktur dient zur Speicherung von Datum und Uhrzeit eines Ereignisses oder zum Quittieren eines Ereignisses.

```

TYPE ST_TimeStamp :
STRUCT
  Year          : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Month         : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Day           : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Hour          : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Minute        : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Second        : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  mSec          : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2 Allgemein**4.4.2.1 E_CntrlCmd**

Aufzählung möglicher Befehle für den Maschinenzustand

```

TYPE E_CntrlCmd :
(
  eCntrlCmd_UNDEFINED,
  eCntrlCmd_RESET,
  eCntrlCmd_START,
  eCntrlCmd_STOP,
  eCntrlCmd_HOLD,
  eCntrlCmd_UNHOLD,
  eCntrlCmd_SUSPEND,
  eCntrlCmd_UNSPEND,
  eCntrlCmd_ABORT,
  eCntrlCmd_CLEAR,
  eCntrlCmd_COMPLETE,
);
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.2 E_CurMachSpd

Aufzählung, um verschiedene Maschinengeschwindigkeiten zu definieren

```

TYPE E_CurMachSpd :
(
  eCurMachSpd_UNDEFINED,
  eCurMachSpd_JOG,
  eCurMachSpd_PRIME,
  eCurMachSpd_PRELUBE,
  eCurMachSpd_MAINTENANCE,
  eCurMachSpd_SLOW,
);

```



```
eCurMachSpd_MEDIUM,
eCurMachSpd_HIGH,
eCurMachSpd_SURGE,
eCurMachSpd_TRACKING,
eCurMachSpd_ANALOG_CTRL_ONLY := 99
);
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.3 ST_Descriptor

Sammlung von Tags für die Beschreibung von Parametern in der Maschine.

```
TYPE ST_Descriptor :
STRUCT
  Id          : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Name       : STRING; (* ~ (OPC : 1: enabled for OPC ) *)
  Unit      : STRING; (* ~ (OPC : 1: enabled for OPC ) *)
  Value     : REAL; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.4 ST_Ingredient

Sammlung von Tags für die Beschreibung der benötigten Rohmaterialien für das Produkt.

```
TYPE ST_Ingredient :
STRUCT
  IngredientId : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Parameter   : ARRAY [1..IMAX_INGREDIENT_PARAMS] OF ST_Descriptor; (* ~ (OPC : 1: enabled
for OPC ) *)
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.5 ST_Material

Sammlung von Tags für die Beschreibung von Materialien in der Maschine.

```
TYPE ST_Materials :
STRUCT
  RawMaterial      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  CO2              : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Container        : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Lubrication     : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Water           : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  ContainerClosures : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)

  (* 10 more unused *)
  Unused0         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Unused1         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Unused2         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Unused3         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  Unused4         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
```

```

Unused5      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
Unused6      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
Unused7      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
Unused8      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
Unused9      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.6 ST_Node

Sammlung von Tags für die Beschreibung der Werte von Kommunikationsbefehlen zwischen Maschinen, die die PackTag Struktur nutzen.

```

TYPE ST_Node :
STRUCT
  Number      : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  ControlCmdNumber : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  CmdValue    : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Parameter   : ARRAY [1..iMAX_NODE_PARAMS] OF ST_Descriptor; (* ~ (OPC : 1: enabled for
OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.2.7 ST_Product

Sammlung von Tags für die Beschreibung des auf der Maschine gefertigten Produkts.

```

TYPE ST_Product :
STRUCT
  ProductId    : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  ProcessVariables : ARRAY [1..iMAX_PROD_PROCESS_VARS] OF ST_Descriptor; (* ~ (OPC : 1: enabled
for OPC ) *)
  Ingredients   : ARRAY [1.. iMAX_INGREDIENTS] OF ST_Ingredient; (* ~ (OPC : 1: enabled for
OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.3 ST_PMLa

Sammlung aller Verwaltungstags der PackTag Struktur.

```

TYPE ST_PMLa :
STRUCT
  Alarm      : ARRAY [1..iMAX_ALARMS] OF ST_Alarm; (* ~ (OPC : 1: enabled for OPC )
*)
  ModeCurrentTime : ARRAY [1..iMAX_CURRENT_MODE] OF DINT; (* ~ (OPC : 1: enabled for
OPC ) *)
  ModeCummulativeTime : ARRAY [1..iMAX_CUMMULATIVE_MODE] OF DINT; (* ~ (OPC : 1: enabled for
OPC ) *)
  StateCurrentTime : ARRAY [1..iMAX_CURRENT_MODE, 0..iMAX_CURRENT_STATE] OF DINT; (* ~
(OPC : 1: enabled for OPC ) *)

```

```

StateCummulativeTime      : ARRAY [1..iMAX_CURRENT_MODE, 0..iMAX_CURRENT_STATE] OF DINT; (* ~
(OPC : 1: enabled for OPC ) *)

ProdProcessed             : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
DefectProd                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
ReworkProd                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
ResetTimersCounters      : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
UpstreamMessage           : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
DownstreamMessage        : DINT; (* ~ (OPC : 1: enabled for OPC ) *)

CurrentDownstreamNodeID   : ARRAY [1..iMAX_CURR_NODE_IDS] OF DINT; (* ~ (OPC : 1: enabled for
OPC ) *)
CurrentUpstreamNodeID    : ARRAY [1..iMAX_CURR_NODE_IDS] OF DINT; (* ~ (OPC : 1: enabled for
OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.4 ST_PMLc

Sammlung aller Befehlstags der PackTag Struktur.

```

TYPE ST_PMLc :
STRUCT
  UnitMode                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  UnitModeChangeRequest  : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  ProcMode                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  ProcModeChangeRequest  : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  CurMachSpeed            : DINT; (* ~ (OPC : 1: enabled for OPC ) *)

  MatReady                : ST_Materials; (* ~ (OPC : 1: enabled for OPC ) *)
  MatLow                  : ST_Materials; (* ~ (OPC : 1: enabled for OPC ) *)

  State                   : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  StateChangeRequest     : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)

  CntrlCmd                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  Node                    : ARRAY [1..iMAX_NODES] OF ST_Node; (* ~ (OPC : 1: enabled for
OPC ) *)
  ProcessVariables        : ARRAY [1..iMAX_PROCESS_VARS] OF ST_Descriptor; (* ~ (OPC : 1:
enabled for OPC ) *)
  Product                 : ARRAY [1..iMAX_PRODUCTS] OF ST_Product; (* ~ (OPC : 1: enabled
for OPC ) *)
  Limits                  : ARRAY [1..iMAX_LIMITS] OF ST_Descriptor; (* ~ (OPC : 1: enabled
for OPC ) *)

  TargetDownstreamNodeID : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  TargetUpstreamNodeID   : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  ChangeNodeServicedUpstream : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  ChangeNodeServicedDownstream : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.4.5 ST_PMLs

Sammlung aller Zustands-Tags der PackTag Struktur.

```

TYPE ST_PMLs :
STRUCT
  CommandRejected        : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
  UnitModeCurrent        : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
  UnitModeRequested      : DINT; (* ~ (OPC : 1: enabled for OPC ) *)

```

```

UnitModeChangeInProgress      : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
ProcModeCurrent               : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
ProcModeRequested             : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
ProcModeChangeInProgress     : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
StateCurrent                  : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
StateRequested                : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
StateChangeInProgress        : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
StateChangeProgress           : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
StateLastCompleted           : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
SeqNumber                     : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
CurMachSpd                   : DINT; (* ~ (OPC : 1: enabled for OPC ) *)

MatReady                      : ST_Materials; (* ~ (OPC : 1: enabled for OPC ) *)
MatLow                        : ST_Materials; (* ~ (OPC : 1: enabled for OPC ) *)

MachDesignSpeed               : REAL; (* ~ (OPC : 1: enabled for OPC ) *)
MachCycle                     : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
ProdRatio                     : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
Dirty                         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
Clean                         : BOOL; (* ~ (OPC : 1: enabled for OPC ) *)
TimeToDirty                   : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
EquipmentAllocatedToUnitModelID : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
MachineReusableForUnitModelID : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
MachineReusableTimeLeft      : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
MachineStoringProductID      : DINT; (* ~ (OPC : 1: enabled for OPC ) *)
MachineTransferringProductID : DINT; (* ~ (OPC : 1: enabled for OPC ) *)

Node                          : ARRAY [1..iMAX_NODES] OF ST_Node; (* ~ (OPC : 1: enabled for
OPC ) *)
ProcessVariables              : ARRAY [1..iMAX_PROCESS_VARS] OF ST_Descriptor; (* ~ (OPC : 1:
enabled for OPC ) *)
Product                       : ARRAY [1..iMAX_PRODUCTS] OF ST_Product; (* ~ (OPC : 1: enabled
for OPC ) *)
Limits                        : ARRAY [1..iMAX_LIMITS] OF ST_Descriptor; (* ~ (OPC : 1: enabled
for OPC ) *)

END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.5 Globale Parameter

Parameter für den Aufbau der Packaging Maschine Tag Strukturen. Diese können beim Einfügen der Bibliothek für das aktuelle Projekt angepasst werden.

```

(* PMLc / PMLs *)
iMAX_NODE_PARAMS              : INT := 10
iMAX_NODES                    : INT := 10
iMAX_PROCESS_VARS             : INT := 10
iMAX_PROD_PROCESS_VARS       : INT := 10
iMAX_INGREDIENT_PARAMS       : INT := 10
iMAX_INGREDIENT               : INT := 10
iMAX_LIMITS                   : INT := 10
iMAX_PRODUCTS                 : INT := 10

(* PMLa *)
iMAX_ALARMS                   : INT := 10
iMAX_CURR_NODE_IDS            : INT := 10

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

4.6 Globale Konstanten

Konstanten für den Aufbau der Packaging Maschine Tag Strukturen. Diese können nicht verändert werden.

```
(* PMLa *)
iMAX_CURRENT_MODE           : INT := 10
iMAX_CUMMULATIVE_MODE      : INT := 10
iMAX_CURRENT_STATE         : INT := 17
iMAX_CUMMULATIVE_STATE     : INT := 17
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
Ab TwinCAT v3.1 Build 4018	PC (i386)	Tc3_PackML

5 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0

E-Mail: info@beckhoff.com

Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.de/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

