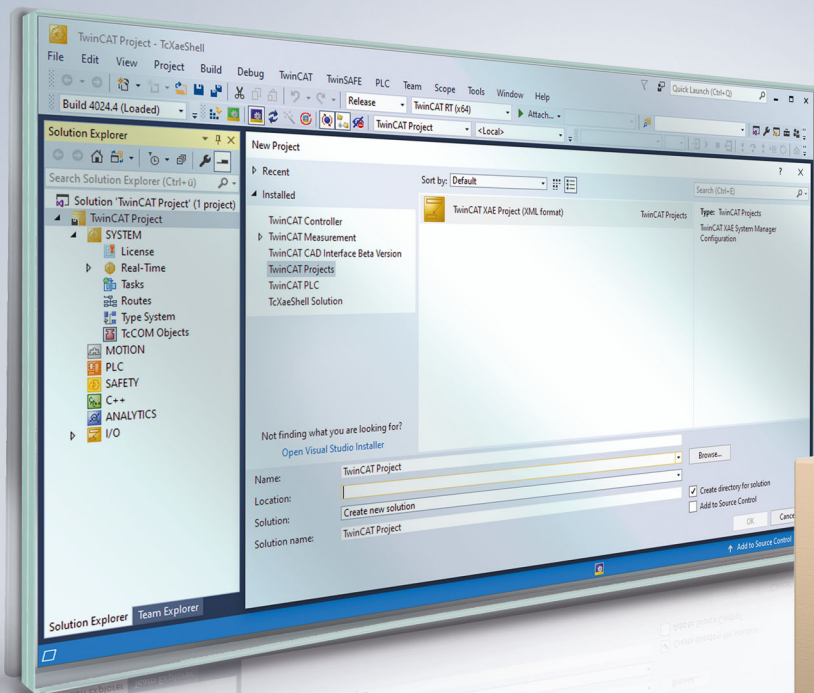


# BECKHOFF New Automation Technology

Handbuch | DE

# TF8400

TwinCAT 3 | MTP Runtime





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort.....</b>	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit .....	7
<b>2</b>	<b>Übersicht.....</b>	<b>8</b>
<b>3</b>	<b>Installation .....</b>	<b>9</b>
3.1	Lizenzierung .....	9
<b>4</b>	<b>Technische Einführung .....</b>	<b>12</b>
<b>5</b>	<b>SPS API .....</b>	<b>13</b>
5.1	Betriebsmodi .....	13
5.1.1	Operation Mode .....	13
5.1.2	Source Mode.....	15
5.1.3	Service Mode .....	17
5.2	Eigenschaften .....	18
5.2.1	Name des Objekts.....	18
5.2.2	Objektbeschreibung .....	18
5.2.3	OSLevel .....	18
5.2.4	WQC .....	19
5.2.5	Wertskalierung .....	19
5.2.6	Einheit .....	19
5.2.7	Wertbegrenzung.....	20
5.2.8	Rückmeldung .....	20
5.2.9	Rücklesen .....	20
5.2.10	Handshake-Verfahren .....	20
5.2.11	Verriegelung.....	20
5.2.12	Rückmeldeüberwachung.....	21
5.2.13	Grenzwertüberwachung.....	22
5.2.14	Reset.....	23
5.3	Funktionsbausteine .....	24
5.3.1	ActiveElements .....	24
5.3.2	DiagnosticElements .....	59
5.3.3	IndicatorElements .....	69
5.3.4	InputElements .....	76
5.3.5	OutputElements .....	80
5.3.6	OperationElements .....	81
5.3.7	ParameterElements .....	88
5.3.8	PeaElements.....	101
5.3.9	ServiceElements .....	103
5.3.10	Report Values .....	115
5.4	Datentypen .....	119
5.4.1	ST_MTP_InputElementConfig .....	119
<b>6</b>	<b>Support und Service .....</b>	<b>120</b>



# 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®

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.

Zuwiderhandlungen 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 Übersicht

Bei der TF8400 TwinCAT 3 MTP Runtime handelt es sich um eine IEC-61131-Bibliothek zur Implementierung der MTP-Schnittstellentypen. Die MTP-Schnittstellentypen orientieren sich an den Anforderungen der MTP-Richtlinie.

---



Die folgende Dokumentation orientiert sich an der Richtlinie VDI/VDE/NAMUR 2658.

---



## 3 Installation

### Systemvoraussetzungen

Die folgenden Systemvoraussetzungen müssen für eine ordnungsgemäße Funktion der TF8400 TwinCAT 3 MTP Runtime erfüllt sein.

Technische Daten	Voraussetzung
Zielformat	IPC oder CX, (x86, x64, Arm®)
Minimale TwinCAT-Version	3.1.4026
Erforderliches TwinCAT-Setup-Level	TC1200 TwinCAT 3 PLC

### TwinCAT Package Manager: Installation (TwinCAT 3.1 Build 4026)

Eine ausführliche Anleitung zur Installation von Produkten finden Sie im Kapitel [Workloads installieren](#) in der Installationsanleitung TwinCAT 3.1 Build 4026.

Installieren Sie das folgende Package in der Kommandozeile, um das Produkt nutzen zu können:

- TwinCAT.XAE.PLC.Lib.Tc3\_MTP

## 3.1 Lizenzierung

Die TwinCAT 3 Function ist als Vollversion oder als 7-Tage-Testversion freischaltbar. Beide Lizenztypen sind über die TwinCAT-3-Entwicklungsumgebung (XAE) aktivierbar.

### Lizenzierung der Vollversion einer TwinCAT 3 Function

Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

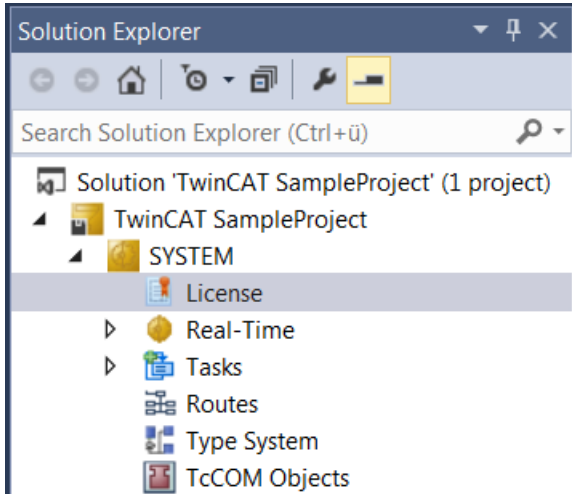
### Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



Eine 7-Tage-Testversion kann nicht für einen [TwinCAT-3-Lizenz-Dongle](#) freigeschaltet werden.

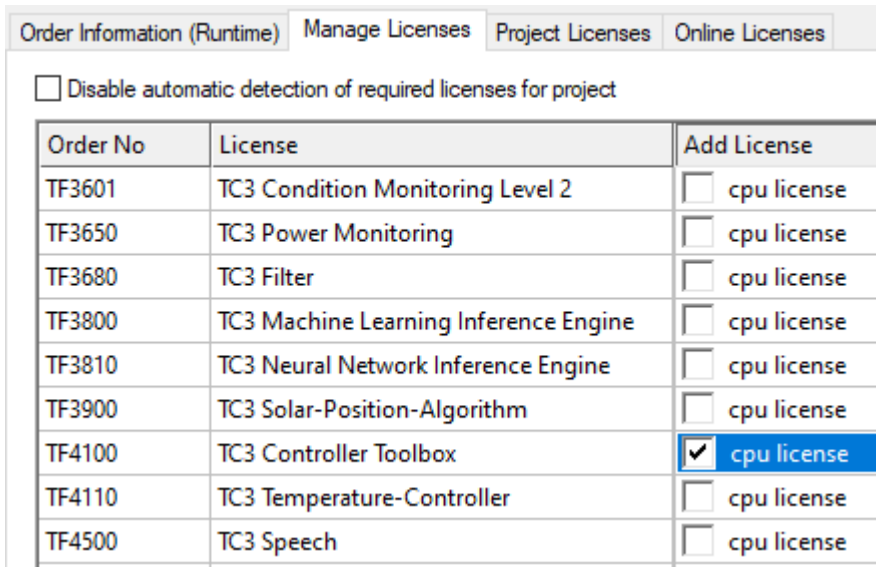
1. Starten Sie die TwinCAT-3-Entwicklungsumgebung (XAE).
2. Öffnen Sie ein bestehendes TwinCAT-3-Projekt oder legen Sie ein neues Projekt an.
3. Wenn Sie die Lizenz für ein Remote-Gerät aktivieren wollen, stellen Sie das gewünschte Zielsystem ein. Wählen Sie dazu in der Symbolleiste in der Drop-down-Liste **Choose Target System** das Zielsystem aus.
  - ⇒ Die Lizenzierungseinstellungen beziehen sich immer auf das eingestellte Zielsystem. Mit der Aktivierung des Projekts auf dem Zielsystem werden automatisch auch die zugehörigen TwinCAT-3-Lizenzen auf dieses System kopiert.

4. Klicken Sie im **Solution Explorer** im Teilbaum **SYSTEM** doppelt auf **License**.



⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

5. Öffnen Sie die Registerkarte **Manage Licenses**. Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).



6. Öffnen Sie die Registerkarte **Order Information (Runtime)**.

⇒ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt.

7. Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.

The screenshot shows a software interface with several sections:

- Order Information (Runtime)**: Includes tabs for 'Manage Licenses', 'Project Licenses', and 'Online Licenses'. Below are fields for 'License Device' (set to 'Target (Hardware Id)'), 'System Id' (containing '2DB25408-B4CD-81DF-5488-6A3D9B49EF19'), and 'Platform' (set to 'other (91)').
- License Request**: Includes a 'Provider' dropdown set to 'Beckhoff Automation', a 'Generate File...' button, and input fields for 'License Id', 'Customer Id', and 'Comment'.
- License Activation**: This section is highlighted with a red box and contains two buttons: '7 Days Trial License...' and 'License Response File...'.

⇒ Es öffnet sich ein Dialog, der Sie auffordert, den im Dialog angezeigten Sicherheitscode einzugeben.

The dialog box is titled 'Enter Security Code' and contains the following elements:

- A prompt: 'Please type the following 5 characters:'
- A text box containing the code 'Kg8T4'.
- An input field with a red border, currently empty.
- 'OK' and 'Cancel' buttons.

8. Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.

9. Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.

## 4 Technische Einführung

Die SPS-Bibliothek TF8400 TwinCAT 3 MTP Runtime stellt Objekte (Funktionsbausteine) zur Steuerung und Überwachung von Aktoren und Sensoren der Prozesstechnik dar. Sie ist primär für das TF8401 TwinCAT 3 MTP Engineering ausgelegt, kann aber auch eigenständig verwendet werden.

Die Objekte der SPS-Bibliothek sind in folgende Bereiche aufgeteilt:

- ActiveElements (Antriebe, Ventile und PID-Regler)
- DiagnosticElements (Verriegelungsanzeigen)
- IndicatorElements (Sensorik und Anzeigen)
- OperationElements (Wertvorgaben)
- InputElements (Eingehende Prozesswertsignale)
- OutputElements (Ausgehende Prozesswertsignale)
- ServiceElements (Dienste)
- ParameterElements (Parameter für Dienste)
- PeaElements (Bereitstellen von Herstellerinformation)

Der Zugriff auf die Variablen dieser Objekte wird über die [Betriebsmodi](#) [► 13] gesteuert und erfolgt über:

- Eingänge des Funktionsbausteins (für die interne Logik)
- Interne Variablen (Zugriff z. B. über OPC-UA)

## 5 SPS API

### 5.1 Betriebsmodi

In diesem Kapitel werden die Zustandsautomaten der Datenobjekte und Dienste beschrieben.

Diese werden nicht alleinstehend verwendet, sondern innerhalb der Datenobjekte und Dienste aufgerufen.

#### 5.1.1 Operation Mode

In dieser Bibliothek wird zwischen `Operator` und `Automatic` als Quellen für Schaltanfragen unterschieden. Der `Operation Mode` ist ein Zustandsautomat zur Verwaltung von Schaltanfragen von diesen Quellen.

##### Zugangswege

`Automatic` Schaltanfragen werden von der internen SPS-Logik geschrieben. Die Variablennamen werden dabei um das Suffix `*Aut` erweitert.

`Operator` Schaltanfragen erfolgen durch eine manuelle Bedienung (z.B. über OPC-UA). Sie sind als interne Variablen deklariert und können nicht über die interne SPS-Logik geschrieben werden. `Operator` Schaltanfragen erweitern den Variablennamen um das Suffix `*Op`.

##### Zustände

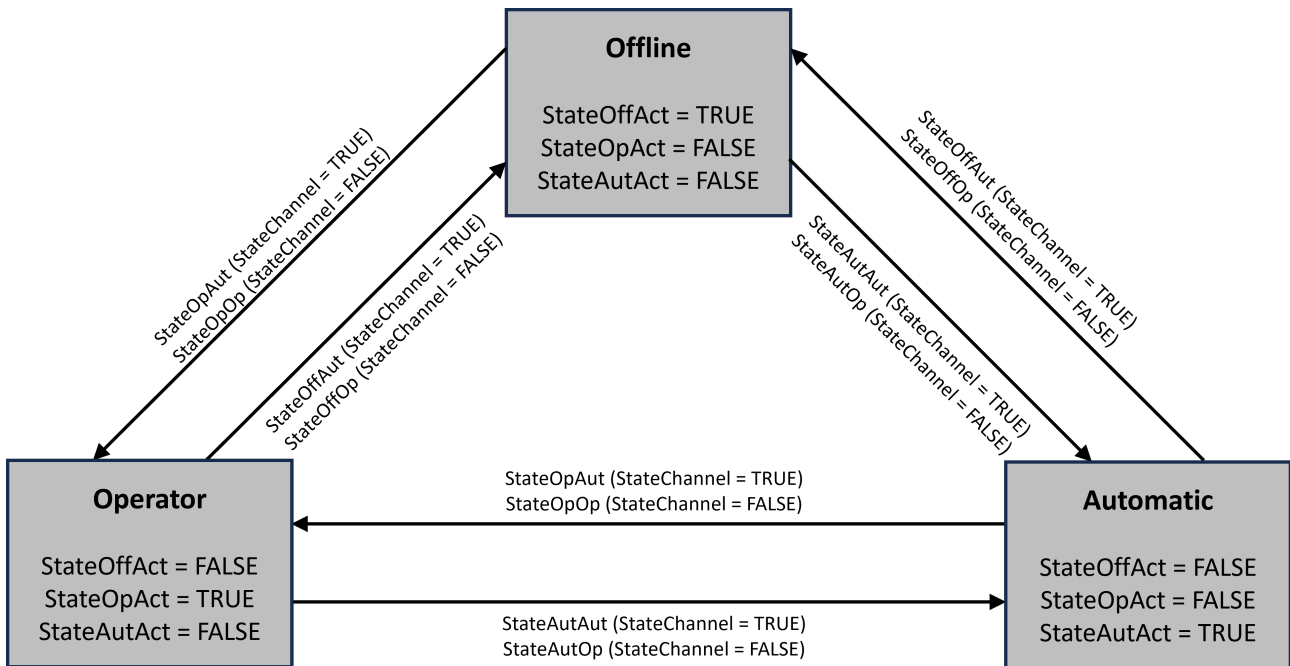
Der Zustandsautomat hat drei Zustände: `Offline`, `Operator` und `Automatic`. Im Zustand `Offline` werden keine neuen Schaltanfragen bearbeitet. Im Zustand `Operator` werden `*Op` Schaltanfragen bearbeitet bzw. im Zustand `Automatic` werden `*Aut` Schaltanfragen bearbeitet.

Der aktuelle Zustand wird über die Ausgänge `StateOffAct`, `StateOpAct` und `StateAutAct` angezeigt.

##### Zustandswechsel

Ein Zustandswechsel erfolgt über Schaltanfragen der internen SPS-Logik oder über einen Bediener (nach dem [Handshake-Verfahren](#) [▶ 20]). Der Eingang `StateChannel` wird über die interne SPS-Logik gesetzt und gibt an, welche Quelle zurzeit einen Zustandswechsel auslösen kann.

Zustände sind folgendermaßen priorisiert: `Offline` > `Operator` > `Automatic`. Daraus folgt, dass bei gleichzeitiger Anforderung aller drei Zustände der Zustand `Offline` eingenommen wird. Bei gleichzeitiger Anfrage von `Operator` und `Automatic` wird der Zustand `Operator` eingenommen.



Ein direkter Übergang vom Zustand *Offline* in den Zustand *Automatic* ist laut MTP-Richtlinie nicht vorgesehen, wird vom Zustandsautomaten jedoch unterstützt.



Der *Operation Mode* ist ein Bestandteil von Objekten dieser Bibliothek und kann nicht alleinstehend verwendet werden.

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
StateChannel	BOOL	Auswahl der Schaltanfragen für den Zustandsautomaten: 1: <i>Automatic</i> Schaltanfragen werden berücksichtigt. 0: <i>Operator</i> Schaltanfragen werden berücksichtigt.	Lesen
StateOffAut	BOOL	<i>Automatic</i> Schaltanfrage, um den <i>Operation Mode</i> in den Zustand <i>Offline</i> zu überführen.	Lesen
StateOpAut	BOOL	<i>Automatic</i> Schaltanfrage, um den <i>Operation Mode</i> in den Zustand <i>Operator</i> zu überführen.	Lesen
StateAutAut	BOOL	<i>Automatic</i> Schaltanfrage, um den <i>Operation Mode</i> in den Zustand <i>Automatic</i> zu überführen	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
StateOffAct	BOOL	1: Aktueller Zustand ist <i>Offline</i> .	Lesen
StateOpAct	BOOL	1: Aktueller Zustand ist <i>Operator</i> .	Lesen
StateAutAct	BOOL	1: Aktueller Zustand ist <i>Automatic</i> .	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
StateOffOp	BOOL	<i>Operator</i> Schaltanfrage, um den <i>Operation Mode</i> in den Zustand <i>Offline</i> zu überführen.	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
		0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	
StateOpOp	BOOL	Operator Schaltanfrage, um den Operation Mode in den Zustand Operator zu überführen. 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben
StateAutOp	BOOL	Operator Schaltanfrage, um den Operation Mode in den Zustand Automatic zu überführen. 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

 Methoden

Name	Beschreibung
SetOffline	Versetzt den Zustandsautomaten <code>OperationMode</code> der Schnittstelle in den Zustand <code>Offline</code> .
SetOperator	Versetzt den Zustandsautomaten <code>OperationMode</code> der Schnittstelle in den Zustand <code>Operator</code> .
SetAutomatic	Versetzt den Zustandsautomaten <code>OperationMode</code> der Schnittstelle in den Zustand <code>Automatic</code> .

### 5.1.2 Source Mode

In dieser Bibliothek wird zwischen `Internal` und `Manual` als Quellen für Wertvorgaben unterschieden. Der `Source Mode` ist ein Zustandsautomat zur Verwaltung von Wertvorgaben von diesen Quellen.

#### Zugangswege

`Internal` Wertvorgaben werden von der internen SPS-Logik geschrieben. Die Variablennamen werden dabei um das Suffix `*Int` erweitert.

`Manual` Wertvorgaben werden von einer manuellen Bedienung (z.B. über OPC-UA) geschrieben. Sie sind als interne Variablen deklariert und können nicht über die interne SPS-Logik geschrieben werden. `Manual` Wertvorgaben erweitern den Variablennamen um das Suffix `*Man`.

#### Zustände

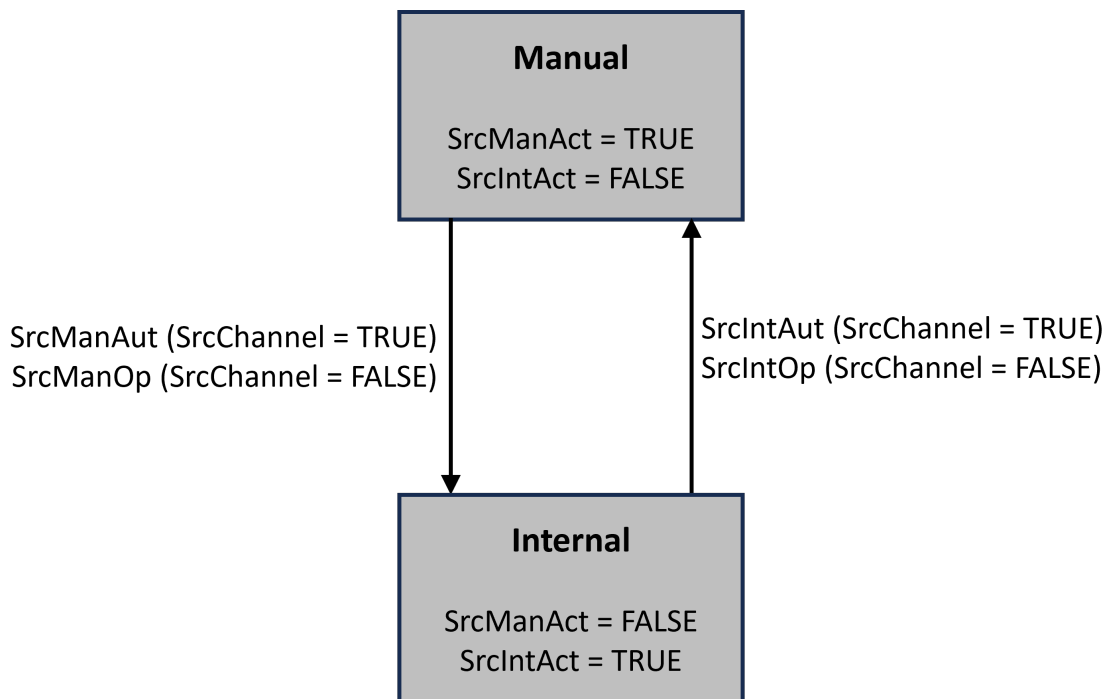
Der Zustandsautomat hat zwei Zustände: `Internal` und `Manual`. Im Zustand `Internal` werden `*Int` Wertvorgaben bearbeitet bzw. im Zustand `Manual` werden `*Man` Wertvorgaben bearbeitet.

Der aktuelle Zustand wird über die Ausgänge `SrcManAct` und `SrcIntAct` angezeigt.

#### Zustandswechsel

Ein Zustandswechsel erfolgt über Schaltanfragen der internen SPS-Logik `*Aut` oder über einen Bediener `*Op` (nach dem Handshake-Verfahren [▶ 20]). Der Eingang `SrcChannel` wird über die interne SPS-Logik gesetzt und gibt an, welche Quelle zurzeit einen Zustandswechsel auslösen kann.

`Internal` ist höher priorisiert als `Manual`. Daher wird bei gleichzeitiger Anfrage beider Zustände der Zustand `Internal` eingenommen.



Der `Source Mode` ist ein Bestandteil von Objekten und kann nicht alleinstehend verwendet werden.

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
SrcChannel	BOOL	Auswahl der Schaltanfragen für den Zustandsautomaten: 1: <code>Automatic</code> Schaltanfragen werden berücksichtigt. 0: <code>Operator</code> Schaltanfragen werden berücksichtigt.	Lesen
SrcIntAut	BOOL	Automatic Schaltanfrage, um den <code>SourceMode</code> in den Zustand <code>Internal</code> zu überführen	Lesen
SrcManAut	BOOL	Automatic Schaltanfrage, um den <code>SourceMode</code> in den Zustand <code>Manual</code> zu überführen	Lesen

#### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
SrcIntAct	BOOL	<code>SourceMode</code> ist im Zustand <code>Internal</code> .	Lesen
SrcManAct	BOOL	<code>SourceMode</code> ist im Zustand <code>Manual</code> .	Lesen

#### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
SrcIntOp	BOOL	Operator Schaltanfrage, um den <code>SourceMode</code> in den Zustand <code>Internal</code> zu überführen. 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben
SrcManOp	BOOL	Operator Schaltanfrage, um den <code>SourceMode</code> in den Zustand <code>Manual</code> zu überführen. 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben



☰ Methoden

Name	Beschreibung
SetInternal	Versetzt den Zustandsautomaten <code>SourceMode</code> der Schnittstelle in den Zustand <code>Internal</code> .
SetManual	Versetzt den Zustandsautomaten <code>SourceMode</code> der Schnittstelle in den Zustand <code>Manual</code> .

### 5.1.3 Service Mode

Für die Dienste stehen drei Quellen für Schaltanfragen und Wertvorgaben zur Verfügung:

- Manuelle Bedienung (z.B. über OPC-UA): `Operator`
- Interne SPS-Logik: `Automatic-Internal`
- Leitsystem (z. B. über OPC-UA): `Automatic-External`

#### Betriebsartensteuerung

Die Betriebsartensteuerung ermöglicht das Umschalten zwischen den Zuständen `Offline`, `Operator` und `Automatic`.

Im Zustand `Offline` werden keine neuen Schaltanfragen und Wertvorgaben berücksichtigt. Im Zustand `Operator` werden `*OP` Schaltanfragen bearbeitet bzw. im Zustand `Automatic` werden `*Aut` Schaltanfragen bearbeitet.

Der aktuelle Zustand wird über die Ausgänge `StateOffAct`, `StateOpAct` und `StateAutAct` angezeigt.

Ein Zustandswechsel erfolgt über die interne SPS-Logik oder über die Bediener. Der Eingang `StateChannel` wird über die interne SPS-Logik gesetzt und gibt an, welche Quelle bei der Bearbeitung berücksichtigt wird.

Die Zustände sind folgendermaßen priorisiert: `Offline` > `Operator` > `Automatic`.

Daraus folgt, dass bei gleichzeitiger Anforderung aller drei Zustände der Zustand `Offline` eingenommen wird. Bei gleichzeitiger Anfrage von `Operator` und `Automatic` wird der Zustand `Operator` eingenommen.

#### Quellenauswahl (Automatic)

Die Quellenauswahl wird in der Betriebsart `Automatic` verwendet und ermöglicht das Umschalten zwischen `Automatic-Internal` und `Automatic-External`.

`Automatic-Internal` Wertvorgaben werden von der internen SPS-Logik geschrieben. Die Variablennamen werden dabei um das Suffix `*Int` erweitert.

`Automatic-External` Wertvorgaben werden durch eine manuelle Bedienung (z. B. über OPC-UA) geschrieben.

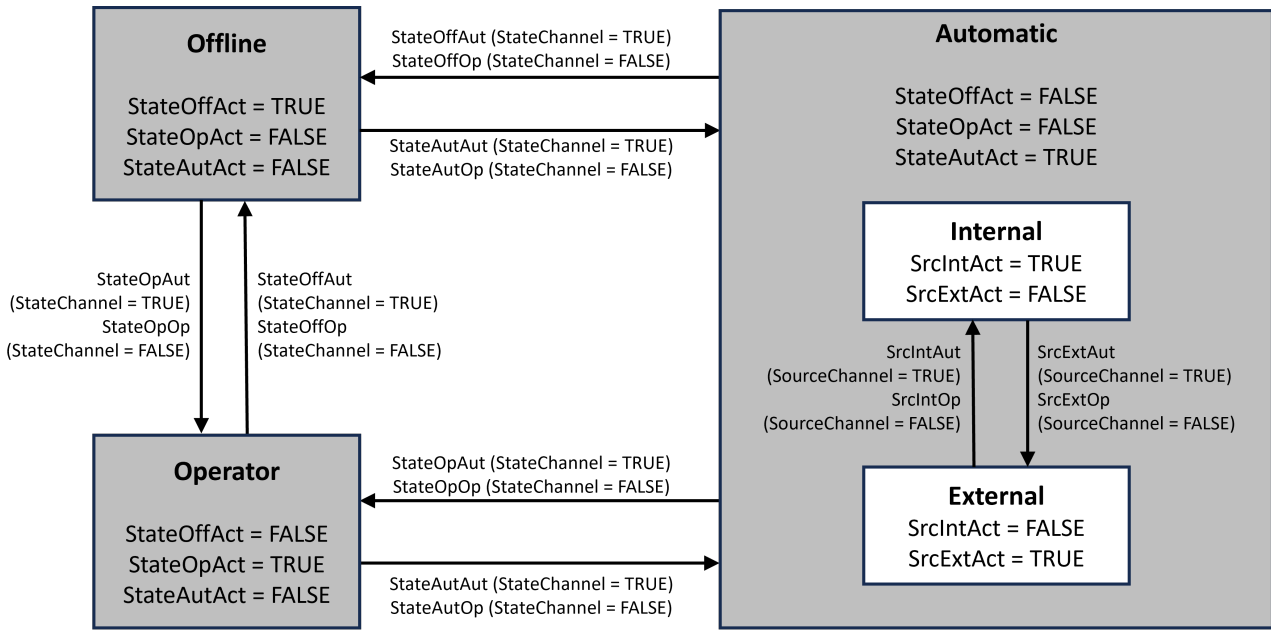
`Operator` Schaltanfragen erfolgen durch eine manuelle Bedienung (z. B. über OPC-UA). Sie sind als interne Variablen deklariert und können nicht über die interne SPS-Logik geschrieben werden. `Operator` Schaltanfragen erweitern den Variablennamen um das Suffix `*Op`.



`Service Mode` ist ein Bestandteil von Schnittstellen und kann nicht alleinstehend verwendet werden!

#### Schema der Zustandsautomaten:

Die Zustandsautomaten arbeiten nach dem folgenden Schema:



## 5.2 Eigenschaften

In diesem Kapitel werden Eigenschaften der Schnittstellen beschrieben.

Diese werden nicht alleinstehend, sondern nach Verwendungszweck der Schnittstellen verwendet.

### 5.2.1 Name des Objekts

Der Name des Objekts wird durch die Variable `TagName` beschrieben. Es ist die eindeutige Kennung des Objekts. Die Länge ist auf 64 Zeichen begrenzt.



Das Ändern des Werts während der Laufzeit ist möglich, laut MTP-Richtlinie jedoch nicht erlaubt!

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-

### 5.2.2 Objektbeschreibung

Die Beschreibung des Objekts erfolgt durch die Variable `TagDescription`.



Das Ändern des Werts während der Laufzeit ist möglich, laut MTP-Richtlinie aber nicht erlaubt!

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagDescription	STRING	Beschreibung der Schnittstelle	-

### 5.2.3 OSLevel

Das `OSLevel` dient der Überwachung von Bedienererebenen. Die Verwendung beschreibt folgende Abhängigkeiten:

OSLevel = 0: Bedienung erfolgt ausschließlich von lokaler Ebene (lokales HMI).

OSLevel = 1 - 255: Bedienung erfolgt von der Prozessleitebene (z.B. über OPC-UA).



Soll die Schnittstelle konform zur MTP-Richtlinie verwendet werden, darf das OSLevel nur von der Leitebene aus gesetzt werden. Nur bei Verlust der Verbindung zur Leitebene ist es der internen Logik erlaubt, diesen Wert manuell auf 0 zu setzen. Die Umsetzung der Bedienerebenen erfolgt durch den Ersteller der Leitebene und ist nicht Teil dieser Dokumentation.

Name	Typ	Beschreibung	OPC-UA-Zugriff
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben

### 5.2.4 WQC

Der WQC (Worst Quality Code) beschreibt, unter welchen Bedingungen der Wert der Schnittstelle erfasst wurde und wie vertrauenswürdig dieser Wert ist. Er wird zumeist von den Sensoren oder Baugruppen gebildet.



Die Werte sind in der aktuellen Version der MTP-Richtlinie nicht näher definiert. Sie können jedoch zum Beispiel aus der NE 184 abgeleitet werden.

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen

### 5.2.5 Wertskalierung

Die Darstellung von Werten erfolgt durch Wertskalierungen. Hierfür stehen Variablen mit den Suffixen \*Sc1Min und \*Sc1Max zur Verfügung.



Eine Änderung dieser Grenzwerte zur Laufzeit ist laut MTP-Richtlinie nicht vorgesehen, wird jedoch vom Objekt unterstützt.

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
*Sc1Min	REAL/ DINT	Unterer Grenzwert der Wertdarstellung	Lesen
*Sc1Max	REAL/ DINT	Oberer Grenzwert der Wertdarstellung	Lesen

### 5.2.6 Einheit

Die Einheit eines Werts wird durch die Unit-Variable mit dem Suffix \*Unit beschrieben. Im Dateityp E\_MTP\_Unit sind die möglichen Einheiten aufgelistet.

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
*Unit	INT	Aufzählungswert der Einheitenliste	Lesen

## 5.2.7 Wertbegrenzung

Wertvorgaben für Objekte dieser Bibliothek werden durch einen oberen und unteren Grenzwert begrenzt. Die Grenzwerte für Wertvorgaben werden durch Variablen mit den Suffixen `*Min` und `*Max` beschrieben. Wertvorgaben außerhalb der Grenzen werden entsprechend durch den unteren oder oberen Grenzwert ersetzt.

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
<code>*Min</code>	REAL/ DINT	Unterer Grenzwert der Wertvorgabe	Lesen
<code>*Max</code>	REAL/ DINT	Oberer Grenzwert der Wertvorgabe	Lesen

## 5.2.8 Rückmeldung

Rückmeldungen für Schaltanfrage oder Wertvorgaben können über die Eingangsvariablen mit dem Suffix `*Fbk` erfolgen. Diese Rückmeldungen können von Feldgeräten kommen oder simuliert werden.

Beispiele:

1. Rückmeldekontakt eines Schützes
2. Aktuelle Drehzahl eines Motors vom Encoder

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
<code>*Fbk</code>	REAL/ DINT/ BOOL	Rückgabewert	Lesen

## 5.2.9 Rücklesen

Das Rücklesen der Wertvorgaben erfolgt über die Variablen mit dem Suffix `*Rbk`. Diese Variablen geben die aktuelle, unmodifizierte Wertvorgabe wieder.

## 5.2.10 Handshake-Verfahren

Das Handshake-Verfahren wird bei einigen Variablen für den Transport von Informationen verwendet. Dabei wird von der manuellen Bedienung ein Wert, z.B. logisch „1“ (Schaltanforderung), auf eine Variable geschrieben. Dieser Wert wird nach Abarbeitung durch den Funktionsbaustein wieder zurück auf eine logische „0“ gesetzt:

0 → 1: Schaltanforderung durch manuelle Bedienung

1 → 0: Schaltanforderung wurde eingelesen

Das Zurücksetzen beschreibt nur, dass der Wert eingelesen wurde. Es heißt nicht, dass dieser Wert auch Einfluss genommen hat. Ob ein Wert Einfluss auf einen Ausgang des Funktionsbausteins genommen hat, wird mithilfe der zustandsbeschreibenden Variable dargestellt.

## 5.2.11 Verriegelung

Die Verriegelung besteht aus einem dreistufigen System, das die Zustände `Permit`, `Interlock` und `Protect` beinhaltet.

Permit ist eine Schaltfreigabe für Schaltanfragen. Ist `Permit = FALSE`, dann kann das Objekt noch bedient werden. Wenn das Objekt in einen Zustand überführt wird, welcher der Sicherheitsposition entspricht, so werden weitere Schaltanfragen nicht mehr berücksichtigt – die Schaltfreigabe ist entzogen.

Interlock ist eine Verriegelung, welche die Schaltfreigabe entzieht und das Objekt in die Sicherheitsposition versetzt.

Protect besitzt die gleiche Funktionalität wie Interlock. Protect setzt sich nicht automatisch zurück, sondern muss durch einen Reset [► 23] quitiert werden.

Die jeweiligen Verriegelungsmöglichkeiten können durch Enable-Variablen `*En` aktiviert werden.

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
PermEn	BOOL	Permit aktivieren: 1: Aktiviert 0: Deaktiviert	Lesen
Permit	BOOL	Permit Signal 1: Permit inaktiv 0: Permit aktiv	Lesen
IntlEn	BOOL	Interlock aktivieren: 1: Aktiviert 0: Deaktiviert	Lesen
Interlock	BOOL	Interlock Signal 1: Interlock inaktiv 0: Interlock aktiv	Lesen
ProtEn	BOOL	Protect aktivieren: 1: Aktiviert 0: Deaktiviert	Lesen
Protect	BOOL	Protect Signal 1: Protect inaktiv 0: Protect aktiv, Reset erforderlich	Lesen

## 5.2.12 Rückmeldeüberwachung

Rückmeldeüberwachung vergleicht die Zustände der Schaltanforderung `*Ctrl` mit den jeweiligen Rückmeldekontakten `*Fbk`. Es wird bei der Überwachung in statische und dynamische Fehler unterschieden. Die Überwachungsfunktion wird durch die Variable `MonEn` von einem Bediener (z.B. über OPC-UA) oder bereits initial im Modul aktiviert.

Ein statischer Fehler `MonStatErr` wird durch eine Änderung des Rückmeldekontakts bei unveränderter Schaltanforderung nach Ablauf der Überwachungszeit `MonStatTi` ausgelöst. Als Beispiel kann hier ein Motor betrachtet werden, bei dem die Versorgungsspannung ausfällt.

Ein dynamischer Fehler `MonDynErr` wird durch eine Änderung der Schaltanforderung ohne resultierende Änderung des Rückmeldekontakts nach Ablauf der Überwachungszeit `MonDynTi` ausgelöst. Dies kann z.B. durch ein Schütz, welches „kleben“ bleibt, ausgelöst werden.

Die Überwachungszeiten, in denen die jeweiligen Fehler ausgelöst werden, können von der internen SPS-Logik eingestellt werden. Über die Variable `MonSafePos` wird das Verhalten der Schnittstelle bei Auftreten eines Fehlers beschrieben.

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
MonStatTi	REAL	Überwachungszeit für statische Fehler [s]	Lesen
MonDynTi	REAL	Überwachungszeit für dynamische Fehler [s]	Lesen
MonSafePos	BOOL	Verhalten der Schnittstelle nach Auftreten eines Fehlers: 1: Sichere Position soll angefahren werden. 0: Aktueller Zustand wird beibehalten.	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
MonStatErr	REAL	Statischer Fehler: 1: aktiv 0: inaktiv	Lesen
MonDynErr	REAL	Dynamischer Fehler: 1: aktiv 0: inaktiv	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
MonEn	BOOL	FeedbackMonitoring aktivieren: 1: FeedbackMonitoring aktiv 0: FeedbackMonitoring inaktiv	Lesen/schreiben

## 5.2.13 Grenzwertüberwachung

Die Grenzwertüberwachung erweitert eine Schnittstelle um bis zu sechs überwachbare Grenzwerte. Der zu überwachende Grenzwert wird über die *\*Lim* Variable durch einen Bediener (z. B. über OPC-UA) eingestellt. Die Überwachung wird über die jeweilige *\*En* Variable aktiviert. Bei Überschreiten der oberen Grenzwerte bzw. unterschreiten der unteren Grenzwerte wird der jeweilige Ausgang *\*Act* auf TRUE gesetzt.

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
*AHEn	BOOL	Grenzwert Alarm High überwachen: 1: Überwachung aktiv 0: keine Überwachung	Lesen
*ALEn	BOOL	Grenzwert Alarm Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
*WHEn	BOOL	Grenzwert Warning High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
*WLEn	BOOL	Grenzwert Warning Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
*THEn	BOOL	Grenzwert Tolerance High überwachen:	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Überwachung aktiv 0: Keine Überwachung	
*TLEn	BOOL	Grenzwert <i>Tolerance Low</i> überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
*AHAct	BOOL	Alarm High Grenzwert überschritten.	Lesen
*ALAct	BOOL	Alarm Low Grenzwert unterschritten.	Lesen
*WHAct	BOOL	Warning High Grenzwert überschritten.	Lesen
*WLAct	BOOL	Warning Low Grenzwert unterschritten.	Lesen
*THAct	BOOL	Tolerance High Grenzwert überschritten.	Lesen
*TLAct	BOOL	Tolerance Low Grenzwert unterschritten.	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
*AHLim	REAL / DINT	Alarm High Grenzwert	Lesen/schreiben
*ALLim	REAL / DINT	Alarm Low Grenzwert	Lesen/schreiben
*WHLim	REAL / DINT	Warning High Grenzwert	Lesen/schreiben
*WLLim	REAL / DINT	Warning Low Grenzwert	Lesen/schreiben
*THLim	REAL / DINT	Tolerance High Grenzwert	Lesen/schreiben
*TLLim	REAL / DINT	Tolerance Low Grenzwert	Lesen/schreiben

## 5.2.14 Reset

Reset setzt anstehende \*Err Meldungen und die Protect-Verriegelung zurück. Reset erfolgt über zwei gleich priorisierte Eingänge: ResetOp und ResetAut. Der Reset durch die manuelle Bedienung (z. B. über OPC-UA) erfolgt über ein [Handshake-Verfahren](#) [► 20].

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
ResetAut	BOOL	Automatic Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
ResetOp	BOOL	Operator Reset Schaltanforderung:	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
		0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	

## 5.3 Funktionsbausteine

### 5.3.1 ActiveElements

`ActiveElements` sind Objekte, die es ermöglichen, auf Aktoren von unterschiedlichen Quellen zuzugreifen. Die Quellen können die interne SPS-Logik oder eine manuelle Bedienung sein. Zu den Aktoren gehören Ventile, elektrische Antriebe oder PID-Regler. Der Zugriff wird über Zustandsautomaten geregelt.

#### 5.3.1.1 FB\_MTP\_BinDrv

FB_MTP_BinDrv	
WQC <i>BYTE</i>	<i>BOOL</i> StateOpAct
OSLevel <i>BYTE</i>	<i>BOOL</i> StateAutAct
StateChannel <i>BOOL</i>	<i>BOOL</i> StateOffAct
StateOffAut <i>BOOL</i>	<i>BOOL</i> FwdCtrl
StateOpAut <i>BOOL</i>	<i>BOOL</i> RevCtrl
StateAutAut <i>BOOL</i>	<i>BOOL</i> SafePosAct
FwdEn <i>BOOL</i>	
RevEn <i>BOOL</i>	
StopAut <i>BOOL</i>	
FwdAut <i>BOOL</i>	
RevAut <i>BOOL</i>	
RevFbk <i>BOOL</i>	
RevFbkCalc <i>BOOL</i>	
FwdFbk <i>BOOL</i>	
FwdFbkCalc <i>BOOL</i>	
SafePos <i>BOOL</i>	
Trip <i>BOOL</i>	
PermEn <i>BOOL</i>	
Permit <i>BOOL</i>	
IntlEn <i>BOOL</i>	
Interlock <i>BOOL</i>	
ProtEn <i>BOOL</i>	
Protect <i>BOOL</i>	
ResetAut <i>BOOL</i>	

Der Funktionsbaustein `FB_MTP_BinDrv` ist ein Objekt zur Steuerung eines binären Antriebs von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z.B. über OPC-UA). Schaltanfragen werden über den Zustandsautomaten des [Operation Mode \[► 13\]](#) verwaltet. Die OPC-UA-Zugriffsrechte sind in den Variablen Tabellen beschrieben.

#### Ansteuern des Antriebs

Die Vorgabe der Drehrichtung des Antriebs erfolgt über die Schaltanfragen `Fwd*`, `Rev*` und `Stop*`. Der Zustandsautomat des [Operation Mode \[► 13\]](#) verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Fwd*` und `Rev*` sind beide gleich priorisiert und der Antrieb bleibt in seinem Zustand. `Stop*` ist bei `SafePos = FALSE` bei gleichzeitigen Anfragen von `Fwd*`, `Rev*` und `Stop*` am höchsten priorisiert. Ist `SafePos = TRUE` sind die Schaltanfragen `Fwd*`, `Rev*` und `Stop*` gleich priorisiert und der Antrieb bleibt bei gleichzeitigen Anfragen in seinem aktuellen Zustand.

Bei aktiver [Verriegelung \[► 20\]](#) ist, je nach Verriegelungsart, ein Steuern des Antriebs nicht mehr möglich.

#### Sicherheitsposition

Die Sicherheitsposition wird über die Variablen `SafePos`, `SafePosDirEn` und `SafePosDir` definiert.



Mit der Variablen `SafePos` wird beschrieben, ob die Sicherheitsposition des Antriebs der energielose Zustand oder ein aktiver Zustand ist.



Die Variablen `SafePosDirEn` und `SafePosDir` sind nicht Bestandteil der MTP-Richtlinie. Sie wurden zusätzlich implementiert und können zur Definition einer Sicherheitsposition im aktiven Zustand verwendet werden. Diese Variablen werden nicht in der MTP-Datei aufgeführt und nicht über OPC-UA zu Verfügung gestellt.

**Sicherheitsposition: Energieloser Zustand**

Im energielosen Zustand ist die Sicherheitsposition des Antriebs: `FwdCtrl = FALSE` und `RevCtrl = FALSE`.

**Sicherheitsposition: Aktiver Zustand**

Für den aktiven Zustand kann die Vorgabe der Drehrichtung aktiviert (`SafePosDirEn = TRUE`) und die Drehrichtung (`SafePosDir`) vorgegeben werden.

Wird die Vorgabe der Drehrichtung deaktiviert `SafePosDirEn = FALSE`, so ist der aktuelle Bewegungszustand (Vorwärts, Rückwärts oder Stopp) die Sicherheitsposition.

Bei aktivierter Vorgabe ist die Sicherheitsposition `SafePosDirEn = TRUE` vorwärts (`SafePosDir = FALSE`) oder rückwärts (`SafePosDir = TRUE`).

**Weitere Eigenschaften**

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[OSLevel \[► 18\]](#)

[Rückmeldung \[► 20\]](#)

[Verriegelung \[► 20\]](#)

[Reset \[► 23\]](#)

**Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
FwdEn	BOOL	Vorwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
RevEn	BOOL	Rückwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
StopAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Stopp zu setzen: (Relevant, wenn <code>StateAutAct</code> ) 1: Stopp ausführen.	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		0: Kein Stopp	
FwdAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen
RevAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Rückwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen
FwdFbkCalc	BOOL	Quelle des Feedback Signals Vorwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
FwdFbk	BOOL	Feedback Signal Vorwärtsbewegung: 1: Vorwärtsbewegung 0: Keine Vorwärtsbewegung	Lesen
RevFbkCalc	BOOL	Quelle des Feedback Signals Rückwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
RevFbk	BOOL	Feedback Signal Rückwärtsbewegung: 1: Rückwärtsbewegung 0: Keine Rückwärtsbewegung	Lesen
Perm* Int* Prot*		Siehe <a href="#">Verriegelung [► 20]</a>	
Trip	BOOL	Meldekontakt Motorschutz: 1: Kein Fehler 0: Motorschutz angesprochen.	Lesen
SafePos	BOOL	1: Behalte aktuellen Zustand bei. 0: Stopp	Lesen
SafePosDirEn	BOOL	1: Drehrichtungsvorgabe für SafePos = TRUE über die Variable SafePosDir 0: keine Drehrichtungsvorgabe	-
SafePosDir	BOOL	Drehrichtungsvorgabe für SafePos = TRUE und SafePosDirRpmEn = TRUE 1: Rückwärts 0: Vorwärts	-
ResetAut	BOOL	Automatic Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
FwdCtrl	BOOL	Vorwärtsbewegung:	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Aktiv 0: Inaktiv	
RevCtrl	BOOL	Rückwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
SafePosAct	BOOL	Sicherheitsposition aktiviert: 1: Sicherheitsposition ist aktiv. 0: Sicherheitsposition ist nicht aktiv.	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
StopOp	BOOL	Operator Schaltanfrage den Antrieb in Stopp zu setzen: (Relevant, wenn StateOpAct) 1: Stopp ausführen. 0: Keinen Stopp ausführen.	Lesen/schreiben
FwdOp	BOOL	Operator Schaltanfrage den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen/schreiben
RevOp	BOOL	Operator Schaltanfrage den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

 **Methoden**

Name	Beschreibung
SetOffline SetOperator SetAutomatic	Siehe <a href="#">Operation Mode [► 13]</a>

### 5.3.1.2 FB\_MTP\_MonBinDrv

FB_MTP_MonBinDrv		
WQC	BYTE	BOOL StateOpAct
OSLevel	BYTE	BOOL StateAutAct
StateChannel	BOOL	BOOL StateOffAct
StateOffAut	BOOL	BOOL FwdCtrl
StateOpAut	BOOL	BOOL RevCtrl
StateAutAut	BOOL	BOOL SafePosAct
FwdEn	BOOL	BOOL MonStatErr
RevEn	BOOL	BOOL MonDynErr
StopAut	BOOL	
FwdAut	BOOL	
RevAut	BOOL	
RevFbk	BOOL	
RevFbkCalc	BOOL	
FwdFbk	BOOL	
FwdFbkCalc	BOOL	
SafePos	BOOL	
Trip	BOOL	
PermEn	BOOL	
Permit	BOOL	
IntlEn	BOOL	
Interlock	BOOL	
ProtEn	BOOL	
Protect	BOOL	
ResetAut	BOOL	
MonStatTi	REAL	
MonDynTi	REAL	
MonSafePos	BOOL	

Der Funktionsbaustein `FB_MTP_MonBinDrv` ist ein Objekt zur Steuerung eines binären Antriebs von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z.B. über OPC-UA).

Schaltanfragen werden über den Zustandsautomaten des [Operation Mode \[13\]](#) verwaltet. Zusätzlich beinhaltet er die Möglichkeit die Ausgangswerte mit den jeweiligen Rückmeldungen zu überwachen. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.

#### Ansteuern des Antriebs

Die Vorgabe der Drehrichtung des Antriebs erfolgt über die Schaltanfragen `Fwd*`, `Rev*` und `Stop*`. Der Zustandsautomat des [Operation Mode \[13\]](#) verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Fwd*` und `Rev*` sind beide gleich priorisiert und der Antrieb bleibt in seinem Zustand. `Stop*` ist bei `SafePos = FALSE` bei gleichzeitigen Anfragen von `Fwd*`, `Rev*` und `Stop*` am höchsten priorisiert. Ist `SafePos = TRUE` sind die Schaltanfragen `Fwd*`, `Rev*` und `Stop*` gleich priorisiert und der Antrieb bleibt bei gleichzeitigen Anfragen in seinem aktuellen Zustand.

Bei aktiver [Verriegelung \[20\]](#) ist, je nach Verriegelungsart, ein Steuern des Antriebs nicht mehr möglich.

#### Sicherheitsposition

Die Sicherheitsposition wird über die Variablen `SafePos`, `SafePosDirEn` und `SafePosDir` definiert.

Mit der Variablen `SafePos` wird beschrieben, ob die Sicherheitsposition des Antriebs der energielose Zustand oder ein aktiver Zustand ist.



Die Variablen `SafePosDirEn` und `SafePosDir` sind nicht Bestandteil der MTP-Richtlinie. Sie wurden zusätzlich implementiert und können zur Definition einer Sicherheitsposition im aktiven Zustand verwendet werden. Diese Variablen werden nicht in der MTP-Datei aufgeführt und nicht über OPC-UA zu Verfügung gestellt.

#### Sicherheitsposition: Energieloser Zustand

Im energielosen Zustand ist die Sicherheitsposition des Antriebs: `FwdCtrl = FALSE` und `RevCtrl = FALSE`.

#### Sicherheitsposition: Aktiver Zustand

Für den aktiven Zustand kann die Vorgabe der Drehrichtung aktiviert (`SafePosDirEn = TRUE`) und die Drehrichtung (`SafePosDir`) vorgegeben werden.

Wird die Vorgabe der Drehrichtung deaktiviert `SafePosDirEn = FALSE`, so ist der aktuelle Bewegungszustand (Vorwärts, Rückwärts oder Stopp) die Sicherheitsposition.

Bei aktivierter Vorgabe ist die Sicherheitsposition `SafePosDirEn = TRUE` vorwärts (`SafePosDir = FALSE`) oder rückwärts (`SafePosDir = TRUE`).

**Überwachung**

Die Ansteuerung und die jeweilige [Rückmeldung \[▶ 20\]](#) kann über die [Rückmeldeüberwachung \[▶ 21\]](#) überwacht werden.

**Weitere Eigenschaften**

[Name des Objekts \[▶ 18\]](#)

[Objektbeschreibung \[▶ 18\]](#)

[WQC \[▶ 19\]](#)

[OSLevel \[▶ 18\]](#)

[Verriegelung \[▶ 20\]](#)

[Reset \[▶ 23\]](#)

**Vererbungshierarchie**

FB\_MTP\_BinDrv

    FB\_MTP\_MonBinDrv

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [▶ 13]</a>	
FwdEn	BOOL	Vorwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
RevEn	BOOL	Rückwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
StopAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Stopp zu setzen: (Relevant, wenn <code>StateAutAct</code> ) 1: Stopp ausführen. 0: Kein Stopp	Lesen
FwdAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn <code>StateAutAct &amp; FwdEn</code> )	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	
RevAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen
FwdFbkCalc	BOOL	Quelle des Feedback Signals Vorwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
FwdFbk	BOOL	Feedback Signal Vorwärtsbewegung: 1: Vorwärtsbewegung 0: Keine Vorwärtsbewegung	Lesen
RevFbkCalc	BOOL	Quelle des Feedback Signals Rückwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
RevFbk	BOOL	Feedback Signal Rückwärtsbewegung: 1: Rückwärtsbewegung 0: Keine Rückwärtsbewegung	Lesen
Perm* Int* Prot*		Siehe <a href="#">Verriegelung [► 20]</a>	
Trip	BOOL	Meldekontakt Motorschutz: 1: Kein Fehler 0: Motorschutz angesprochen.	Lesen
SafePos	BOOL	1: Behalte aktuellen Zustand bei. 0: Stopp	Lesen
SafePosDirEn	BOOL	1: Drehrichtungsvorgabe für SafePos = TRUE über die Variable SafePosDir 0: keine Drehrichtungsvorgabe	-
SafePosDir	BOOL	Drehrichtungsvorgabe für SafePos = TRUE und SafePosDirRpmEn = TRUE 1: Rückwärts 0: Vorwärts	-
MonStatTi	REAL	Überwachungszeit für statische Fehler [s]	Lesen
MonDynTi	REAL	Überwachungszeit für dynamische Fehler [s]	Lesen
MonSafePos	BOOL	Verhalten der Schnittstelle nach Auftreten eines Fehlers: 1: Sichere Position soll angefahren werden. 0: Aktueller Zustand wird beibehalten.	Lesen
ResetAut	BOOL	Automatic Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	

Name	Typ	Beschreibung	OPC-UA-Zugriff
FwdCtrl	BOOL	Vorwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
RevCtrl	BOOL	Rückwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
SafePosAct	BOOL	Sicherheitsposition aktiviert: 1: Sicherheitsposition ist aktiv. 0: Sicherheitsposition ist nicht aktiv.	Lesen
MonStatErr	REAL	Statischer Fehler: 1: aktiv 0: inaktiv	Lesen
MonDynErr	REAL	Dynamischer Fehler: 1: aktiv 0: inaktiv	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
StopOp	BOOL	Operator Schaltanfrage den Antrieb in Stopp zu setzen: (Relevant, wenn StateOpAct) 1: Stopp ausführen. 0: Keinen Stopp ausführen.	Lesen/schreiben
FwdOp	BOOL	Operator Schaltanfrage den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen/schreiben
RevOp	BOOL	Operator Schaltanfrage den Antrieb in Rückwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen/schreiben
MonEn	BOOL	FeedbackMonitoring aktivieren: 1: FeedbackMonitoring aktiv 0: FeedbackMonitoring inaktiv	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

 **Methoden**

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	

Name	Beschreibung
SetAutomatic	

### 5.3.1.3 FB\_MTP\_AnaDrv

FB_MTP_AnaDrv			
WQC	BYTE	BOOL	StateOpAct
OSLevel	BYTE	BOOL	StateAutAct
StateChannel	BOOL	BOOL	StateOffAct
StateOffAut	BOOL	BOOL	SrcIntAct
StateOpAut	BOOL	BOOL	SrcManAct
StateAutAut	BOOL	BOOL	FwdCtrl
SrcChannel	BOOL	BOOL	RevCtrl
SrcManAut	BOOL	REAL	Rpm
SrcIntAut	BOOL	BOOL	SafePosAct
FwdEn	BOOL		
RevEn	BOOL		
StopAut	BOOL		
FwdAut	BOOL		
RevAut	BOOL		
RevFbk	BOOL		
RevFbkCalc	BOOL		
FwdFbk	BOOL		
FwdFbkCalc	BOOL		
SafePos	BOOL		
Trip	BOOL		
RpmScMin	REAL		
RpmScMax	REAL		
RpmUnit	INT		
RpmInt	REAL		
RpmMin	REAL		
RpmMax	REAL		
RpmFbk	REAL		
RpmFbkCalc	BOOL		
PermEn	BOOL		
Permit	BOOL		
IntlEn	BOOL		
Interlock	BOOL		
ProtEn	BOOL		
Protect	BOOL		
ResetAut	BOOL		

Der Funktionsbaustein FB\_MTP\_AnaDrv ist ein Objekt zur Steuerung eines Antriebs mit variabler Drehzahl von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z.B. über OPC-UA). Schaltanfragen und Wertvorgaben werden unabhängig voneinander über die Zustandsautomaten des [Operation Mode](#) [► 13] und des [Source Mode](#) [► 15] verwaltet. Die OPC-UA-Zugriffsrechte sind in den Variablen tabellen beschrieben.

#### Ansteuern des Antriebs

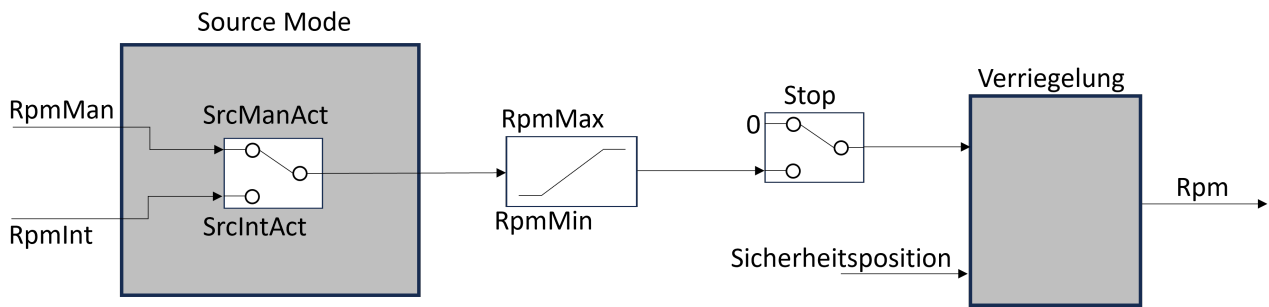
Die Vorgabe der Drehrichtung des Antriebs erfolgt über die Schaltanfragen Fwd\*, Rev\* und Stop\*. Der Zustandsautomat des [Operation Mode](#) [► 13] verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von Fwd\* und Rev\* sind beide gleich priorisiert und der Antrieb bleibt in seinem Zustand. Stop\* ist bei SafePos = FALSE bei gleichzeitigen Anfragen von Fwd\*, Rev\* und Stop\* am höchsten priorisiert. Ist SafePos = TRUE sind die Schaltanfragen Fwd\*, Rev\* und Stop\* gleich priorisiert und der Antrieb bleibt bei gleichzeitigen Anfragen in seinem aktuellen Zustand.

Bei aktiver [Verriegelung](#) [► 20] ist, je nach Verriegelungsart, ein Steuern des Antriebs nicht mehr möglich.

#### Sollwertvorgabe

Die Drehzahlvorgabe erfolgt über Rpm\* Variablen. Die Drehzahlvorgabe wird von dem Zustandsautomaten [Source Mode](#) [► 15] verwaltet und nach dem untenstehenden Schema ausgegeben:





**Sicherheitsposition**

Die Sicherheitsposition wird über die Variablen `SafePos`, `SafePosDirRpmEn`, `SafePosDir` und `SafePosRpm` definiert.

Mit der Variablen `SafePos` wird beschrieben, ob die Sicherheitsposition des Antriebs der energielose Zustand oder ein aktiver Zustand ist.

**i** Die Variablen `SafePosDirRpmEn`, `SafePosDir` und `SafePosRpm` sind nicht Bestandteil der MTP-Richtlinie. Sie wurden zusätzlich implementiert und können zur Definition einer Sicherheitsposition im aktiven Zustand verwendet werden. Diese Variablen werden nicht in der MTP-Datei aufgeführt und werden nicht über OPC-UA zu Verfügung gestellt.

**Sicherheitsposition: Energieloser Zustand (`SafePos = FALSE`)**

Im Energielosen Zustand ist der Antrieb gestoppt (`FwdCtrl = FALSE`, `RevCtrl = FALSE` und `Rpm = 0`)

**Sicherheitsposition: Aktiver Zustand (`SafePos = TRUE`)**

Bei einem aktiven Zustand kann die Vorgabe der Drehrichtung und Drehzahl aktiviert (`SafePosDirRpmEn = TRUE`) und die Drehrichtung (`SafePosDir`) bzw. Drehzahl (`SafePosRpm`) vorgegeben werden.

Wenn `SafePosDirRpmEn = FALSE`, ist der aktuelle Bewegungszustand (Vorwärts, Rückwärts oder Stopp) die Sicherheitsposition.

Bei aktivierter Vorgabe (`SafePosDirRpmEn = TRUE`) ist die Sicherheitsposition Vorwärts (`SafePosDir = FALSE`) oder Rückwärts (`SafePosDir = TRUE`) mit der definierten Sicherheitsdrehzahl (`SafePosRpm`).

Nachfolgend finden Sie eine Tabelle mit den Bedienmöglichkeiten des Antriebs (`Fwd*`, `Rev*`, `Stop*` und `Rpm*`) bei aktivem `Permit` in Abhängigkeit der Ausgangslage des Antriebs (Antrieb Stoppen, Antrieb in Vorwärtsbewegung setzen und Antrieb in Rückwärtsbewegung setzen) und der definierten Sicherheitsposition:

SafePos	SafePos DirRpmEn	SafePos-Dir	Drive set to Stop	Drive set to Forward	Drive set to Reverse
FALSE	-	-	SafePosAct = TRUE Rev* + Fwd*: X Rpm*: X	SafePosAct = FALSE Rev* + Stop*: ✓ Rpm*: ✓	SafePosAct = FALSE Fwd* + Stop*: ✓ Rpm*: ✓
TRUE	FALSE	-	SafePosAct = TRUE Rev* + Fwd*: X Rpm*: X	SafePosAct = TRUE Rev* + Stop*: X Rpm*: X	SafePosAct = TRUE Fwd* + Stop*: X Rpm*: X
TRUE	TRUE	FALSE (Fwd)	SafePosAct = FALSE Rev* + Fwd*: ✓	SafePosAct = TRUE Rev* + Stop*: X Rpm*: X (SafePosRpm)	SafePosAct = FALSE Fwd* + Stop*: ✓

SafePos	SafePos DirRpmEn	SafePos-Dir	Drive set to Stop	Drive set to Forward	Drive set to Reverse
			Rpm*: ✓		Rpm*: ✓
TRUE	TRUE	TRUE (Rev)	SafePosAct = FALSE Rev* + Fwd*: ✓ Rpm*: ✓	SafePosAct = FALSE Rev* + Stop*: ✓ Rpm*: ✓	SafePosAct = TRUE Fwd* + Stop*: X Rpm*: X (SafePosRpm)

x: Schaltanfragen und/oder Wertvorgaben werden nicht mehr berücksichtigt

✓: Schaltanfragen und/oder Wertvorgaben werden berücksichtigt

### Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[OSLevel \[► 18\]](#)

[Einheit \[► 19\]](#)

[Rückmeldung \[► 20\]](#)

[Verriegelung \[► 20\]](#)

[Reset \[► 23\]](#)

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
FwdEn	BOOL	Vorwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
RevEn	BOOL	Rückwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
StopAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Stopp zu setzen: (Relevant, wenn StateAutAct) 1: Stopp ausführen. 0: Kein Stopp	Lesen
FwdAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & FwdEn) 1: Vorwärtsbewegung ausführen.	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		0: Keine Vorwärtsbewegung ausführen.	
RevAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen
FwdFbkCalc	BOOL	Quelle des Feedback Signals Vorwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
FwdFbk	BOOL	Feedback Signal Vorwärtsbewegung: 1: Vorwärtsbewegung 0: Keine Vorwärtsbewegung	Lesen
RevFbkCalc	BOOL	Quelle des Feedback Signals Rückwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
RevFbk	BOOL	Feedback Signal Rückwärtsbewegung: 1: Rückwärtsbewegung 0: Keine Rückwärtsbewegung	Lesen
Src*		Siehe <a href="#">Source Mode</a> [► 15]	
RpmInt	REAL	Internal Drehzahl-Sollwertvorgabe	Lesen
RpmUnit	INT	Drehzahl-Einheit	Lesen
RpmMin	REAL	Oberer Grenzwert des Drehzahl-Sollwerts	Lesen
RpmMax	REAL	Unterer Grenzwert des Drehzahl-Sollwerts	Lesen
RpmFbkCalc	BOOL	Quelle des Drehzahl-Istwerts: 1: Berechnet 0: Sensor	Lesen
RpmFbk	REAL	Drehzahl-Istwert	Lesen
RpmScIMin	REAL	Drehzahl-Skalenanfang	Lesen
RpmScIMax	REAL	Drehzahl-Skalenende	Lesen
Perm* Int* Prot*		Siehe <a href="#">Verriegelung</a> [► 20]	
Trip	BOOL	Meldekontakt Motorschutz: 1: Kein Fehler 0: Motorschutz angesprochen.	Lesen
SafePos	BOOL	1: Behalte aktuellen Zustand bei. 0: Stopp	Lesen
SafePosDirRpmEn	BOOL	1: Drehrichtungsvorgabe für SafePos = TRUE über die Variable SafePosDir 0: keine Drehrichtungsvorgabe	-
SafePosDir	BOOL	Drehrichtungsvorgabe für SafePos = TRUE und SafePosDirRpmEn = TRUE 1: Rückwärts 0: Vorwärts	-
SafePosRpm	REAL	Drehzahlvorgabe für Sicherheitsposition, wenn SafePos = TRUE und SafePosDirRpmEn = TRUE	-
ResetAut	BOOL	Automatic Reset Schaltanforderung:	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Reset angefordert. 0: Kein Reset angefordert.	

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
FwdCtrl	BOOL	Vorwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
RevCtrl	BOOL	Rückwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
Rpm	REAL	Drehzahl-Sollwert an Antrieb	Lesen
SafePosAct	BOOL	Sicherheitsposition aktiviert: 1: Sicherheitsposition ist aktiv. 0: Sicherheitsposition ist nicht aktiv.	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
StopOp	BOOL	Operator Schaltanfrage den Antrieb in Stopp zu setzen: (Relevant, wenn StateOpAct) 1: Stopp ausführen. 0: Keinen Stopp ausführen.	Lesen/schreiben
FwdOp	BOOL	Operator Schaltanfrage den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen/schreiben
RevOp	BOOL	Operator Schaltanfrage den Antrieb in Rückwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
RpmMan	REAL	Manual Sollwertvorgabe für die Drehzahl	Lesen/schreiben
RpmRbk	REAL	Unbearbeiteter Wert der Operator Drehzahl Wertevorgabe	Lesen
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

Methoden

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	
SetAutomatic	
SetInternal	Siehe <a href="#">Source Mode [► 15]</a>
SetManual	

5.3.1.4 FB\_MTP\_MonAnaDrv

FB_MTP_MonAnaDrv			
WQC	BYTE	BOOL	StateOpAct
OSLevel	BYTE	BOOL	StateAutAct
StateChannel	BOOL	BOOL	StateOffAct
StateOffAut	BOOL	BOOL	SrcIntAct
StateOpAut	BOOL	BOOL	SrcManAct
StateAutAut	BOOL	BOOL	FwdCtrl
SrcChannel	BOOL	BOOL	RevCtrl
SrcManAut	BOOL	REAL	Rpm
SrcIntAut	BOOL	BOOL	SafePosAct
FwdEn	BOOL	BOOL	MonStatErr
RevEn	BOOL	BOOL	MonDynErr
StopAut	BOOL	REAL	RpmErr
FwdAut	BOOL	BOOL	RpmAHAct
RevAut	BOOL	BOOL	RpmALAct
RevFbk	BOOL		
RevFbkCalc	BOOL		
FwdFbk	BOOL		
FwdFbkCalc	BOOL		
SafePos	BOOL		
Trip	BOOL		
RpmSclMin	REAL		
RpmSclMax	REAL		
RpmUnit	INT		
RpmInt	REAL		
RpmMin	REAL		
RpmMax	REAL		
RpmFbk	REAL		
RpmFbkCalc	BOOL		
PermEn	BOOL		
Permit	BOOL		
IntlEn	BOOL		
Interlock	BOOL		
ProtEn	BOOL		
Protect	BOOL		
ResetAut	BOOL		
MonSafePos	BOOL		
RpmAHEn	BOOL		
RpmALEn	BOOL		
MonStatTi	REAL		
MonDynTi	REAL		

Der Funktionsbaustein FB\_MTP\_MonAnaDrv ist ein Objekt zur Steuerung eines Antriebs mit variabler Drehzahl von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z.B. über OPC-UA). Schaltanfragen und Wertvorgaben werden unabhängig voneinander über die Zustandsautomaten des [Operation Mode \[► 13\]](#) und des [Source Mode \[► 15\]](#) verwaltet. Zusätzlich beinhaltet er die Möglichkeit die Ausgangswerte mit den jeweiligen Rückmeldungen zu überwachen. Die OPC-UA-Zugriffsrechte sind in den Variablen tabellen beschrieben.

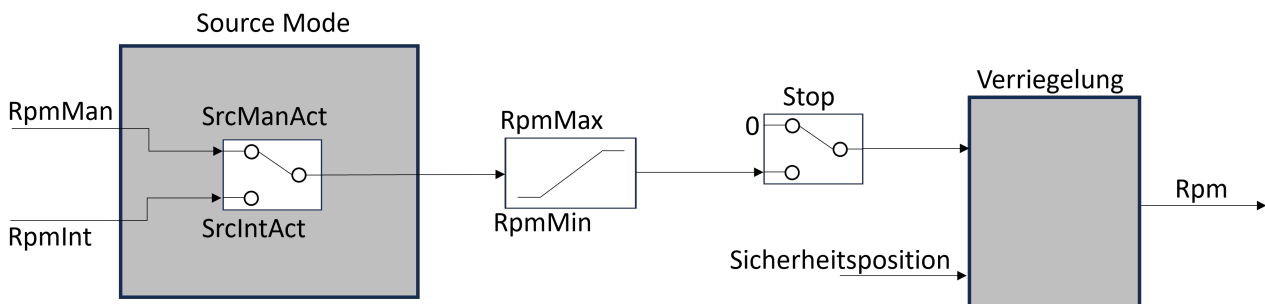
## Ansteuern des Antriebs

Die Vorgabe der Drehrichtung des Antriebs erfolgt über die Schaltanfragen  $Fwd^*$ ,  $Rev^*$  und  $Stop^*$ . Der Zustandsautomat des [Operation Mode](#) [► 13] verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von  $Fwd^*$  und  $Rev^*$  sind beide gleich priorisiert und der Antrieb bleibt in seinem Zustand.  $Stop^*$  ist bei  $SafePos = FALSE$  bei gleichzeitigen Anfragen von  $Fwd^*$ ,  $Rev^*$  und  $Stop^*$  am höchsten priorisiert. Ist  $SafePos = TRUE$  sind die Schaltanfragen  $Fwd^*$ ,  $Rev^*$  und  $Stop^*$  gleich priorisiert und der Antrieb bleibt bei gleichzeitigen Anfragen in seinem aktuellen Zustand.

Bei aktiver [Verriegelung](#) [► 20] ist, je nach Verriegelungsart, ein Steuern des Antriebs nicht mehr möglich.

## Sollwertvorgabe

Die Drehzahlvorgabe erfolgt über  $Rpm^*$  Variablen. Die Drehzahlvorgabe wird von dem Zustandsautomaten [Source Mode](#) [► 15] verwaltet und nach dem untenstehenden Schema ausgegeben:



## Sicherheitsposition

Die Sicherheitsposition wird über die Variablen  $SafePos$ ,  $SafePosDirRpmEn$ ,  $SafePosDir$  und  $SafePosRpm$  definiert.

Mit der Variablen  $SafePos$  wird beschrieben, ob die Sicherheitsposition des Antriebs der energielose Zustand oder ein aktiver Zustand ist.



Die Variablen  $SafePosDirRpmEn$ ,  $SafePosDir$  und  $SafePosRpm$  sind nicht Bestandteil der MTP-Richtlinie. Sie wurden zusätzlich implementiert und können zur Definition einer Sicherheitsposition im aktiven Zustand verwendet werden. Diese Variablen werden nicht in der MTP-Datei aufgeführt und werden nicht über OPC-UA zu Verfügung gestellt.

### Sicherheitsposition: Energieloser Zustand ( $SafePos = FALSE$ )

Im Energielosen Zustand ist der Antrieb gestoppt ( $FwdCtrl = FALSE$ ,  $RevCtrl = FALSE$  und  $Rpm = 0$ )

### Sicherheitsposition: Aktiver Zustand ( $SafePos = TRUE$ )

Bei einem aktiven Zustand kann die Vorgabe der Drehrichtung und Drehzahl aktiviert ( $SafePosDirRpmEn = TRUE$ ) und die Drehrichtung ( $SafePosDir$ ) bzw. Drehzahl ( $SafePosRpm$ ) vorgegeben werden.

Wenn  $SafePosDirRpmEn = FALSE$ , ist der aktuelle Bewegungszustand (Vorwärts, Rückwärts oder Stopp) die Sicherheitsposition.

Bei aktivierter Vorgabe ( $SafePosDirRpmEn = TRUE$ ) ist die Sicherheitsposition Vorwärts ( $SafePosDir = FALSE$ ) oder Rückwärts ( $SafePosDir = TRUE$ ) mit der definierten Sicherheitsdrehzahl ( $SafePosRpm$ ).

Nachfolgend finden Sie eine Tabelle mit den Bedienmöglichkeiten des Antriebs ( $Fwd^*$ ,  $Rev^*$ ,  $Stop^*$  und  $Rpm^*$ ) bei aktivem  $Permit$  in Abhängigkeit der Ausgangslage des Antriebs (Antrieb Stoppen, Antrieb in Vorwärtsbewegung setzen und Antrieb in Rückwärtsbewegung setzen) und der definierten Sicherheitsposition:

SafePos	SafePos DirRpmEn	SafePos-Dir	Drive set to Stop	Drive set to Forward	Drive set to Reverse
FALSE	-	-	SafePosAct = TRUE Rev* + Fwd*: X Rpm*: X	SafePosAct = FALSE Rev* + Stop*: ✓ Rpm*: ✓	SafePosAct = FALSE Fwd* + Stop*: ✓ Rpm*: ✓
TRUE	FALSE	-	SafePosAct = TRUE Rev* + Fwd*: X Rpm*: X	SafePosAct = TRUE Rev* + Stop*: X Rpm*: X	SafePosAct = TRUE Fwd* + Stop*: X Rpm*: X
TRUE	TRUE	FALSE (Fwd)	SafePosAct = FALSE Rev* + Fwd*: ✓ Rpm*: ✓	SafePosAct = TRUE Rev* + Stop*: X Rpm*: X (SafePosRpm)	SafePosAct = FALSE Fwd* + Stop*: ✓ Rpm*: ✓
TRUE	TRUE	TRUE (Rev)	SafePosAct = FALSE Rev* + Fwd*: ✓ Rpm*: ✓	SafePosAct = FALSE Rev* + Stop*: ✓ Rpm*: ✓	SafePosAct = TRUE Fwd* + Stop*: X Rpm*: X (SafePosRpm)

x: Schaltanfragen und/oder Wertvorgaben werden nicht mehr berücksichtigt

✓: Schaltanfragen und/oder Wertvorgaben werden berücksichtigt

**Überwachung**

Die Ansteuerung und die jeweilige Rückmeldung [▶ 20] kann über die Rückmeldeüberwachung [▶ 21] überwacht werden.

Der Ausgang RpmErr gibt die aktuelle Drehzahlabweichung  $RpmErr = Rpm - RpmFbk$  an. Die Drehzahlabweichung kann mit der Grenzwertüberwachung [▶ 22] auf eine obere und untere Grenze (Alarm High und Alarm Low) überwacht werden.

**Weitere Eigenschaften**

[Name des Objekts \[▶ 18\]](#)

[Objektbeschreibung \[▶ 18\]](#)

[WQC \[▶ 19\]](#)

[OSLevel \[▶ 18\]](#)

[Wertskalierung \[▶ 19\]](#)

[Wertbegrenzung \[▶ 20\]](#)

[Verriegelung \[▶ 20\]](#)

[Reset \[▶ 23\]](#)

**Vererbungshierarchie**

FB\_MTP\_AnaDrv

    FB\_MTP\_MonAnaDrv

 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [►_13]</a>	
FwdEn	BOOL	Vorwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
RevEn	BOOL	Rückwärtsbewegung freigeben: 1: Freigabe 0: Gesperrt	Lesen
StopAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Stopp zu setzen: (Relevant, wenn StateAutAct) 1: Stopp ausführen. 0: Kein Stopp	Lesen
FwdAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen
RevAut	BOOL	Automatic Schaltanfrage, um den Antrieb in Rückwärtsbewegung zu setzen: (Relevant, wenn StateAutAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen
FwdFbkCalc	BOOL	Quelle des Feedback Signals Vorwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
FwdFbk	BOOL	Feedback Signal Vorwärtsbewegung: 1: Vorwärtsbewegung 0: Keine Vorwärtsbewegung	Lesen
RevFbkCalc	BOOL	Quelle des Feedback Signals Rückwärtsbewegung: 1: Berechnet 0: Sensor	Lesen
RevFbk	BOOL	Feedback Signal Rückwärtsbewegung: 1: Rückwärtsbewegung 0: Keine Rückwärtsbewegung	Lesen
Src*		Siehe <a href="#">Source Mode [►_15]</a>	
RpmInt	REAL	Internal Drehzahl-Sollwertvorgabe	Lesen
RpmUnit	INT	Drehzahl-Einheit	Lesen
RpmMin	REAL	Oberer Grenzwert des Drehzahl-Sollwerts	Lesen
RpmMax	REAL	Unterer Grenzwert des Drehzahl-Sollwerts	Lesen
RpmFbkCalc	BOOL	Quelle des Drehzahl-Istwerts:	Lesen



Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Berechnet 0: Sensor	
RpmFbk	REAL	Drehzahl-Istwert	Lesen
RpmScIMin	REAL	Drehzahl-Skalenanfang	Lesen
RpmScIMax	REAL	Drehzahl-Skalenende	Lesen
Perm* Int* Prot*		Siehe <a href="#">Verriegelung [► 20]</a>	
Trip	BOOL	Meldekontakt Motorschutz: 1: Kein Fehler 0: Motorschutz angesprochen.	Lesen
SafePos	BOOL	1: Behalte aktuellen Zustand bei. 0: Stopp	Lesen
SafePosDirRpmEn	BOOL	1: Drehrichtungsvorgabe für SafePos = TRUE über die Variable SafePosDir 0: keine Drehrichtungsvorgabe	-
SafePosDir	BOOL	Drehrichtungsvorgabe für SafePos = TRUE und SafePosDirRpmEn = TRUE 1: Rückwärts 0: Vorwärts	-
SafePosRpm	REAL	Drehzahlvorgabe für Sicherheitsposition, wenn SafePos = TRUE und SafePosDirRpmEn = TRUE	-
MonStatTi	REAL	Überwachungszeit für statische Fehler [s]	Lesen
MonDynTi	REAL	Überwachungszeit für dynamische Fehler [s]	Lesen
MonSafePos	BOOL	Verhalten der Schnittstelle nach Auftreten eines Fehlers: 1: Sichere Position soll angefahren werden. 0: Aktueller Zustand wird beibehalten.	Lesen
RpmAHEn	BOOL	Grenzwert Alarm High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
RpmALEn	BOOL	Grenzwert Alarm Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
ResetAut	BOOL	Automatic Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
FwdCtrl	BOOL	Vorwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen
RevCtrl	BOOL	Rückwärtsbewegung: 1: Aktiv 0: Inaktiv	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
Rpm	REAL	Drehzahl-Sollwert an Antrieb	Lesen
MonStatErr	REAL	Statischer Fehler: 1: aktiv 0: inaktiv	Lesen
MonDynErr	REAL	Dynamischer Fehler: 1: aktiv 0: inaktiv	Lesen
RpmErr	REAL	Drehzahlabweichung: $RpmErr = Rpm - RpmFbk$	
RpmAHAct	BOOL	Alarm High Grenzwert überschritten.	Lesen
RpmALAct	BOOL	Alarm Low Grenzwert unterschritten.	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
StopOp	BOOL	Operator Schaltanfrage den Antrieb in Stopp zu setzen: (Relevant, wenn StateOpAct) 1: Stopp ausführen. 0: Keinen Stopp ausführen.	Lesen/schreiben
FwdOp	BOOL	Operator Schaltanfrage den Antrieb in Vorwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & FwdEn) 1: Vorwärtsbewegung ausführen. 0: Keine Vorwärtsbewegung ausführen.	Lesen/schreiben
RevOp	BOOL	Operator Schaltanfrage den Antrieb in Rückwärtsbewegung zu setzen: (Relevant, wenn StateOpAct & RevEn) 1: Rückwärtsbewegung ausführen. 0: Keine Rückwärtsbewegung ausführen.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
RpmMan	REAL	Manual Sollwertvorgabe für die Drehzahl	Lesen/schreiben
RpmRbk	REAL	Unbearbeiteter Wert der Operator Drehzahl Wertevorgabe	Lesen
MonEn	BOOL	FeedbackMonitoring aktivieren: 1: FeedbackMonitoring aktiv 0: FeedbackMonitoring inaktiv	Lesen/schreiben
RpmAHLim	REAL	Alarm High Grenzwert (>0)	Lesen/schreiben
RpmALLim	REAL	Alarm Low Grenzwert (<0)	Lesen/schreiben

### Methoden

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	
SetAutomatic	
SetInternal	Siehe <a href="#">Source Mode [► 15]</a>

Name	Beschreibung
SetManual	

### 5.3.1.5 FB\_MTP\_BinVlv

FB_MTP_BinVlv		
WQC	BYTE	BOOL StateOpAct
OSLevel	BYTE	BOOL StateAutAct
StateChannel	BOOL	BOOL StateOffAct
StateOffAut	BOOL	BOOL Ctrl
StateOpAut	BOOL	BOOL SafePosAct
StateAutAut	BOOL	
SafePos	BOOL	
SafePosEn	BOOL	
OpenAut	BOOL	
CloseAut	BOOL	
OpenFbkCalc	BOOL	
CloseFbkCalc	BOOL	
OpenFbk	BOOL	
CloseFbk	BOOL	
PermEn	BOOL	
Permit	BOOL	
IntlEn	BOOL	
Interlock	BOOL	
ProtEn	BOOL	
Protect	BOOL	
ResetAut	BOOL	

Der Funktionsbaustein FB\_MTP\_BinVlv ist ein Objekt zur Steuerung eines binären Ventils von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z. B. über OPC-UA). Schaltanfragen werden über den Zustandsautomaten Operation Mode [► 13] verwaltet. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.

#### Ansteuern des Ventils

Die Vorgabe zum Öffnen und Schließen des Ventils erfolgt über die Schaltanfragen `Open*` und `Close*`. Der Zustandsautomat des Operation Mode [► 13] verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Open*` und `Close*` ist `Close*` am höchsten priorisiert. Bei aktiver Verriegelung [► 20] wird ein Ansteuern der Ausgänge über die Schaltanfragen verhindert bzw. die Sicherheitsposition eingenommen.

#### Sicherheitsposition

Die Sicherheitsposition wird über die Variablen `SafePosEn` und `SafePos` definiert.

Mit der Variablen `SafePosEn` wird beschrieben, ob das Ventil eine feste Sicherheitsposition hat (`SafePosEn = TRUE`) oder die aktuelle Position beibehalten werden soll (`SafePosEn = FALSE`).

Mit `SafePos` wird die Lage der Sicherheitsposition beschrieben:

`SafePos = FALSE`: Sicherheitsposition des Ventils: geschlossen

`SafePos = TRUE`: Sicherheitsposition des Ventils: offen

#### Weitere Eigenschaften

Name des Objekts [► 18]

Objektbeschreibung [► 18]

WQC [► 19]

OSLevel [► 18]

[Rückmeldung \[► 20\]](#)[Verriegelung \[► 20\]](#)[Reset \[► 23\]](#)**🔴 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenAut	BOOL	<i>Automatic</i> Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen
CloseAut	BOOL	<i>Automatic</i> Schaltanfrage zum Schließen des Ventils.	Lesen
OpenFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil offen: 1: Berechnet 0: Sensor	Lesen
OpenFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil offen	Lesen
CloseFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil geschlossen: 1: Berechnet 0: Sensor	Lesen
CloseFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil geschlossen.	Lesen
Perm*		Siehe <a href="#">Verriegelung [► 20]</a>	
Int*			
Prot*			
SafePosEn	BOOL	Sichere Position des Ventils verwenden: 1: Sichere Position verwenden. 0: Sichere Position nicht verwenden.	Lesen
SafePos	BOOL	Sichere Position des Ventils: 1: Offen 0: Geschlossen	Lesen
ResetAut	BOOL	<i>Automatic</i> Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

**🔴 Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
Ctrl	BOOL	Schaltbefehl Ventil: 1: Öffnen 0: Schließen	Lesen
SafePosAct	BOOL	Sichere Position:	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Sichere Position anfahren. 0: Normaler Betrieb	

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <u>Operation Mode</u> [► 13]	
OpenOp	BOOL	Operator Schaltanfrage zum Öffnen des Ventils.	Lesen/schreiben
CloseOp	BOOL	Operator Schaltanfrage zum Schließen des Ventils.	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

**Methoden**

Name	Beschreibung
SetOffline	Siehe <u>Operation Mode</u> [► 13]
SetOperator	
SetAutomatic	

**5.3.1.6 FB\_MTP\_MonBinVlv**

FB_MTP_MonBinVlv	
WQC	BYTE BOOL StateOpAct
OSLevel	BYTE BOOL StateAutAct
StateChannel	BOOL BOOL StateOffAct
StateOffAut	BOOL BOOL Ctrl
StateOpAut	BOOL BOOL SafePosAct
StateAutAut	BOOL BOOL MonStatErr
SafePos	BOOL BOOL MonDynErr
SafePosEn	BOOL
OpenAut	BOOL
CloseAut	BOOL
OpenFbkCalc	BOOL
CloseFbkCalc	BOOL
OpenFbk	BOOL
CloseFbk	BOOL
PermEn	BOOL
Permit	BOOL
IntlEn	BOOL
Interlock	BOOL
ProtEn	BOOL
Protect	BOOL
ResetAut	BOOL
MonSafePos	BOOL
MonStatTi	REAL
MonDynTi	REAL

Der Funktionsbaustein FB\_MTP\_MonBinVlv ist ein Objekt zur Steuerung eines binären Ventils von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z. B. über OPC-UA). Schaltanfragen werden über den Zustandsautomaten Operation Mode [► 13] verwaltet. Zusätzlich beinhaltet er die Möglichkeit den Ausgangswert mit der jeweiligen Rückmeldung zu überwachen. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.

## Ansteuern des Ventils

Die Vorgabe zum Öffnen und Schließen des Ventils erfolgt über die Schaltanfragen `Open*` und `Close*`. Der Zustandsautomat des [Operation Mode \[▶ 13\]](#) verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Open*` und `Close*` ist `Close*` am höchsten priorisiert. Bei aktiver [Verriegelung \[▶ 20\]](#) wird ein Ansteuern der Ausgänge über die Schaltanfragen verhindert bzw. die Sicherheitsposition eingenommen.

## Sicherheitsposition

Die Sicherheitsposition wird über die Variablen `SafePosEn` und `SafePos` definiert.

Mit der Variablen `SafePosEn` wird beschrieben, ob das Ventil eine feste Sicherheitsposition hat (`SafePosEn = TRUE`) oder die aktuelle Position beibehalten werden soll (`SafePosEn = FALSE`).

Mit `SafePos` wird die Lage der Sicherheitsposition beschrieben:

`SafePos = FALSE`: Sicherheitsposition des Ventils: geschlossen

`SafePos = TRUE`: Sicherheitsposition des Ventils: offen

## Überwachung

Die Ansteuerung und die jeweilige [Rückmeldung \[▶ 20\]](#) kann mit der [Rückmeldeüberwachung \[▶ 21\]](#) überwacht werden.

## Weitere Eigenschaften

[Name des Objekts \[▶ 18\]](#)

[Objektbeschreibung \[▶ 18\]](#)

[WQC \[▶ 19\]](#)

[OSLevel \[▶ 18\]](#)

[Verriegelung \[▶ 20\]](#)

[Reset \[▶ 23\]](#)

## Vererbungshierarchie

FB\_MTP\_BinVlv

    FB\_MTP\_MonBinVlv

## 🔌 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode [▶ 13]</a>	
OpenAut	BOOL	<code>Automatic</code> Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen
CloseAut	BOOL	<code>Automatic</code> Schaltanfrage zum Schließen des Ventils.	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
OpenFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil offen: 1: Berechnet 0: Sensor	Lesen
OpenFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil offen	Lesen
CloseFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil geschlossen: 1: Berechnet 0: Sensor	Lesen
CloseFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil geschlossen.	Lesen
Perm* Int* Prot*		Siehe <a href="#">Verriegelung [► 20]</a>	
SafePosEn	BOOL	Sichere Position des Ventils verwenden: 1: Sichere Position verwenden. 0: Sichere Position nicht verwenden.	Lesen
SafePos	BOOL	Sichere Position des Ventils: 1: Offen 0: Geschlossen	Lesen
ResetAut	BOOL	Automatic Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen
MonStatTi	REAL	Überwachungszeit für statische Fehler [s]	Lesen
MonDynTi	REAL	Überwachungszeit für dynamische Fehler [s]	Lesen
MonSafePos	BOOL	Verhalten der Schnittstelle nach Auftreten eines Fehlers: 1: Sichere Position soll angefahren werden. 0: Aktueller Zustand wird beibehalten.	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
Ctrl	BOOL	Schaltbefehl Ventil: 1: Öffnen 0: Schließen	Lesen
SafePosAct	BOOL	Sichere Position: 1: Sichere Position anfahren. 0: Normaler Betrieb	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenOp	BOOL	Operator Schaltanfrage zum Öffnen des Ventils.	Lesen/schreiben
CloseOp	BOOL	Operator Schaltanfrage zum Schließen des Ventils.	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1→0: Anfrage wurde verarbeitet.	
MonEn	BOOL	FeedbackMonitoring aktivieren: 1: FeedbackMonitoring aktiv 0: FeedbackMonitoring inaktiv	Lesen/schreiben

### Methoden

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	
SetAutomatic	

### 5.3.1.7 FB\_MTP\_AnaVlv

FB_MTP_AnaVlv			
WQC	BYTE	BOOL	StateOpAct
OSLevel	BYTE	BOOL	StateAutAct
StateChannel	BOOL	BOOL	StateOffAct
StateOffAut	BOOL	BOOL	SrcIntAct
StateOpAut	BOOL	BOOL	SrcManAct
StateAutAut	BOOL	REAL	Pos
SrcChannel	BOOL	BOOL	OpenAct
SrcManAut	BOOL	BOOL	CloseAct
SrcIntAut	BOOL	BOOL	SafePosAct
OpenAut	BOOL		
CloseAut	BOOL		
OpenFbk	BOOL		
CloseFbk	BOOL		
OpenFbkCalc	BOOL		
CloseFbkCalc	BOOL		
PosScdMin	REAL		
PosScdMax	REAL		
PosUnit	INT		
PosInt	REAL		
PosMin	REAL		
PosMax	REAL		
SafePos	BOOL		
SafePosEn	BOOL		
PosFbk	REAL		
PosFbkCalc	BOOL		
PermEn	BOOL		
Permit	BOOL		
IntlEn	BOOL		
Interlock	BOOL		
ProtEn	BOOL		
Protect	BOOL		
ResetAut	BOOL		

Der Funktionsbaustein FB\_MTP\_AnaVlv dient als Schnittstelle zur Steuerung eines analogen Ventils von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z. B. über OPC-UA). Schaltafragen und Wertvorgaben werden unabhängig voneinander über die Zustandsautomaten des [Operation Mode \[► 13\]](#) und des [Source Mode \[► 15\]](#) verwaltet. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.

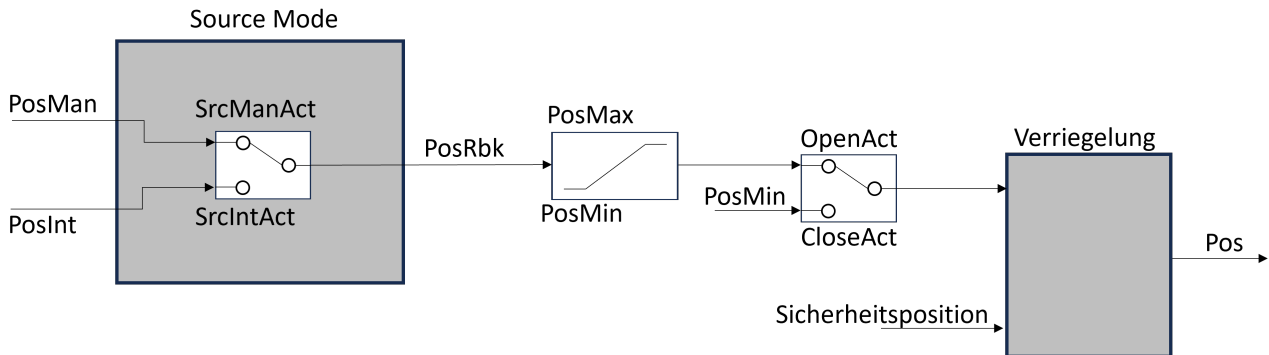


**Ansteuern des Ventils**

Die Vorgabe zum Öffnen und Schließen des Ventils erfolgt über die Schaltanfragen `Open*` und `Close*`. Der Zustandsautomat des [Operation Mode](#) [▶ 13] verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Open*` und `Close*` ist `Close*` am höchsten priorisiert. Bei aktiver [Verriegelung](#) [▶ 20] wird ein Ansteuern der Ausgänge über die Schaltanfragen verhindert bzw. die Sicherheitsposition eingenommen.

**Sollwertvorgabe**

Die Positions Vorgabe erfolgt über `Pos*` Variablen. Die Positions Vorgabe wird von dem Zustandsautomaten des [Source Mode](#) [▶ 15] verwaltet und nach dem untenstehenden Schema ausgegeben:



**Sicherheitsposition**

Die Sicherheitsposition wird über die Variablen `SafePosEn` und `SafePos` definiert.

Mit der Variablen `SafePosEn` wird beschrieben, ob das Ventil eine feste Sicherheitsposition hat (`SafePosEn = TRUE`) oder die aktuelle Position beibehalten werden soll (`SafePosEn = FALSE`)

Mit `SafePos` wird die Lage der Sicherheitsposition beschrieben:

`SafePos = FALSE`: Sicherheitsposition des Ventils: `PosMin`

`SafePos = TRUE`: Sicherheitsposition des Ventils: `PosMax`

**Weitere Eigenschaften**

[Name des Objekts](#) [▶ 18]

[Objektbeschreibung](#) [▶ 18]

[WQC](#) [▶ 19]

[OSLevel](#) [▶ 18]

[Operation Mode](#) [▶ 13]

[Source Mode](#) [▶ 15]

[Rückmeldung](#) [▶ 20]

[Einheit](#) [▶ 19]

[Wertskalierung](#) [▶ 19]

[Wertbegrenzung](#) [▶ 20]

[Verriegelung](#) [▶ 20]

[Reset](#) [▶ 23]

 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <u>Operation Mode</u> [► 13]	
OpenAut	BOOL	<i>Automatic</i> Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen
CloseAut	BOOL	<i>Automatic</i> Schaltanfrage zum Schließen des Ventils.	Lesen
OpenFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil offen: 1: Berechnet 0: Sensor	Lesen
OpenFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil offen	Lesen
CloseFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil geschlossen: 1: Berechnet 0: Sensor	Lesen
CloseFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil geschlossen.	Lesen
Src*		Siehe <u>Source Mode</u> [► 15]	
PosInt	REAL	<i>Internal</i> Wertvorgabe für die Position	Lesen
PosUnit	INT	Einheit der Position	Lesen
PosMin	REAL	Untere Wertbegrenzung der Wertvorgabe für die Ventil-Position	Lesen
PosMax	REAL	Obere Wertbegrenzung der Wertvorgabe für die Ventil-Position	Lesen
PosFbkCalc	BOOL	Quelle des Rückmeldewerts der Position: 1: Berechnet 0: Sensor	Lesen
PosFbk	REAL	Rückmeldewert der Position	Lesen
PosScIMin	REAL	Skalenanfang der Positionsdarstellung	Lesen
PosScIMax	REAL	Skalenende der Positionsdarstellung	Lesen
Perm*		Siehe <u>Verriegelung</u> [► 20]	
Int*			
Prot*			
SafePosEn	BOOL	Sichere Position des Ventils verwenden: 1: Sichere Position verwenden. 0: Sichere Position nicht verwenden (Position Halten).	Lesen
SafePos	BOOL	Sichere Position des Ventils: 1: PosMax 0: PosMin	Lesen
ResetAut	BOOL	<i>Automatic</i> Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenAct	BOOL	Schaltbefehl Ventil öffnen.	Lesen
CloseAd	BOOL	Schaltbefehl Ventil schließen.	Lesen
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
Pos	REAL	Positionssollwert an Ventil	Lesen
SafePosAct	BOOL	Sichere Position: 1: Sichere Position anfahren 0: Normaler Betrieb	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenOp	BOOL	Operator Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen/schreiben
CloseOp	BOOL	Operator Schaltanfrage zum Schließen des Ventils.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
PosMan	BOOL	Manual Wertvorgabe für die Position	Lesen/schreiben
PosRbk	REAL	Unbearbeiteter Wert der Operator Wertevorgabe für die Position	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben

 **Methoden**

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	
SetAutomatic	
SetInternal	Siehe <a href="#">Source Mode [► 15]</a>
SetManual	

### 5.3.1.8 FB\_MTP\_MonAnaVlv

FB_MTP_MonAnaVlv			
WQC	BYTE	BOOL	StateOpAct
OSLevel	BYTE	BOOL	StateAutAct
StateChannel	BOOL	BOOL	StateOffAct
StateOffAut	BOOL	BOOL	SrcIntAct
StateOpAut	BOOL	BOOL	SrcManAct
StateAutAut	BOOL	REAL	Pos
SrcChannel	BOOL	BOOL	OpenAct
SrcManAut	BOOL	BOOL	CloseAct
SrcIntAut	BOOL	BOOL	SafePosAct
OpenAut	BOOL	BOOL	MonStatErr
CloseAut	BOOL	BOOL	MonDynErr
OpenFbk	BOOL	BOOL	MonPosErr
CloseFbk	BOOL	BOOL	PosReachedFbk
OpenFbkCalc	BOOL		
CloseFbkCalc	BOOL		
PosSclMin	REAL		
PosSclMax	REAL		
PosUnit	INT		
PosInt	REAL		
PosMin	REAL		
PosMax	REAL		
SafePos	BOOL		
SafePosEn	BOOL		
PosFbk	REAL		
PosFbkCalc	BOOL		
PermEn	BOOL		
Permit	BOOL		
IntlEn	BOOL		
Interlock	BOOL		
ProtEn	BOOL		
Protect	BOOL		
ResetAut	BOOL		
MonSafePos	BOOL		
PosTolerance	REAL		
MonStatTi	REAL		
MonDynTi	REAL		
MonPosTi	REAL		

Der Funktionsbaustein FB\_MTP\_MonAnaVlv ist ein Objekt zur Steuerung eines analogen Ventils von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (Zugriff über OPC-UA).

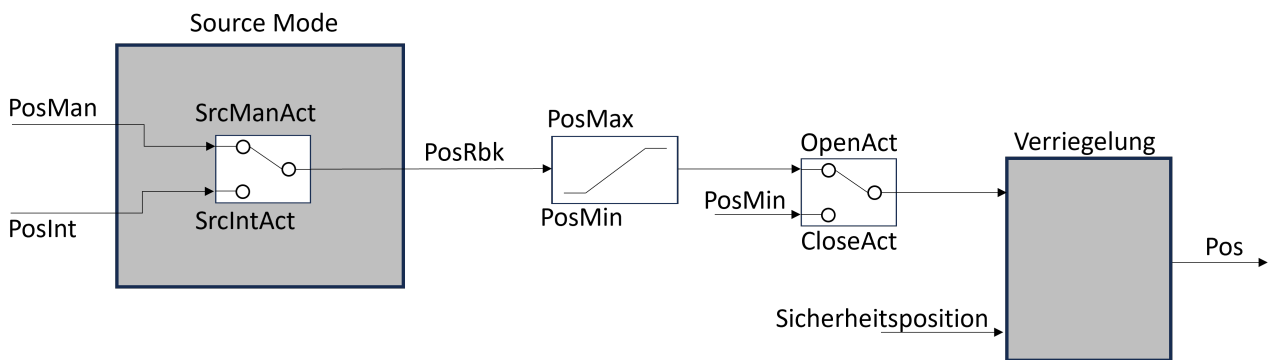
Schaltanfragen und Wertvorgaben werden unabhängig voneinander über die Zustandsautomaten [Operation Mode](#) [▶ 13] und [Source Mode](#) [▶ 15] verwaltet. Zusätzlich beinhaltet er die Möglichkeit die Ausgangswerte mit den jeweiligen Rückmeldungen zu überwachen. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.

#### Ansteuern des Ventils

Die Vorgabe zum Öffnen und Schließen des Ventils erfolgt über die Schaltanfragen `Open*` und `Close*`. Der Zustandsautomat des [Operation Mode](#) [▶ 13] verwaltet, ob und von welcher Quelle neue Schaltanfragen verarbeitet werden. Bei gleichzeitigen Anfragen von `Open*` und `Close*` ist `Close*` am höchsten priorisiert. Bei aktiven [Verriegelung](#) [▶ 20] wird ein Ansteuern der Ausgänge über die Schaltanfragen verhindert bzw. die Sicherheitsposition eingenommen.

#### Sollwertvorgabe

Die Positionsvorgabe erfolgt über `Pos*` Variablen. Die Positionsvorgabe wird von dem Zustandsautomaten des [Source Mode](#) [▶ 15] verwaltet und nach dem untenstehenden Schema ausgegeben:



### Sicherheitsposition

Die Sicherheitsposition wird über die Variablen `SafePosEn` und `SafePos` definiert.

Mit der Variablen `SafePosEn` wird beschrieben, ob das Ventil eine feste Sicherheitsposition hat (`SafePosEn = TRUE`) oder die aktuelle Position beibehalten werden soll (`SafePosEn = FALSE`)

Mit `SafePos` wird die Lage der Sicherheitsposition beschrieben:

`SafePos = FALSE`: Sicherheitsposition des Ventils: `PosMin`

`SafePos = TRUE`: Sicherheitsposition des Ventils: `PosMax`

### Überwachung

Die Ansteuerung und die jeweilige [Rückmeldung](#) [► 20] kann über die [Rückmeldeüberwachung](#) [► 21] überwacht werden.

Zudem verfügt die Schnittstelle über ein Positionsüberwachung, welches die [Rückmeldeüberwachung](#) [► 21] um zwei weitere Ausgänge erweitert:

Der Ausgang `PosReachedFbk` gibt an, ob die Sollposition erreicht wurde:

$$\text{PosReachedFbk} = (\text{PosFbk} - \text{Pos}) \leq \text{PosTolerance}$$

Über den Eingang `MonPosTi` wird die Zeit zum Erreichen der Position vorgegeben. Bei Überschreiten der Zeit wird der Ausgang `MonPosErr` auf `TRUE` gesetzt.

### Weitere Eigenschaften

[Name des Objekts](#) [► 18]

[Objektbeschreibung](#) [► 18]

[WQC](#) [► 19]

[OSLevel](#) [► 18]

[Wertskalierung](#) [► 19]

[Wertbegrenzung](#) [► 20]

[Verriegelung](#) [► 20]

[Reset](#) [► 23]

### Vererbungshierarchie

FB\_MTP\_AnaVlv

    FB\_MTP\_MonAnaVlv

 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <u>Operation Mode</u> [► 13]	
OpenAut	BOOL	<i>Automatic</i> Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen
CloseAut	BOOL	<i>Automatic</i> Schaltanfrage zum Schließen des Ventils.	Lesen
OpenFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil offen: 1: Berechnet 0: Sensor	Lesen
OpenFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil offen	Lesen
CloseFbkCalc	BOOL	Quelle des Rückmeldesignals Endlagenschalter Ventil geschlossen: 1: Berechnet 0: Sensor	Lesen
CloseFbk	BOOL	Rückmeldesignal Endlagenschalter Ventil geschlossen.	Lesen
Src*		Siehe <u>Source Mode</u> [► 15]	
PosInt	REAL	<i>Internal</i> Wertvorgabe für die Position	Lesen
PosUnit	INT	Einheit der Position	Lesen
PosMin	REAL	Untere Wertbegrenzung der Wertvorgabe für die Ventil-Position	Lesen
PosMax	REAL	Obere Wertbegrenzung der Wertvorgabe für die Ventil-Position	Lesen
PosFbkCalc	BOOL	Quelle des Rückmeldewerts der Position: 1: Berechnet 0: Sensor	Lesen
PosFbk	REAL	Rückmeldewert der Position	Lesen
PosScIMin	REAL	Skalenanfang der Positionsdarstellung	Lesen
PosScIMax	REAL	Skalenende der Positionsdarstellung	Lesen
Perm*		Siehe <u>Verriegelung</u> [► 20]	
Int*			
Prot*			
SafePosEn	BOOL	Sichere Position des Ventils verwenden: 1: Sichere Position verwenden. 0: Sichere Position nicht verwenden.	Lesen
SafePos	BOOL	Sichere Position des Ventils: 1: Offen 0: Geschlossen	Lesen
ResetAut	BOOL	<i>Automatic</i> Reset Schaltanforderung: 1: Reset angefordert. 0: Kein Reset angefordert.	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenAct	BOOL	Schaltbefehl Ventil öffnen.	Lesen
CloseAd	BOOL	Schaltbefehl Ventil schließen.	Lesen
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
Pos	REAL	Positionssollwert an Ventil	Lesen
SafePosAct	BOOL	Sichere Position: 1: Sichere Position anfahren 0: Normaler Betrieb	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
OpenOp	BOOL	Operator Schaltanfrage zum Anfahren der ausgewählten Position des Ventils.	Lesen/schreiben
CloseOp	BOOL	Operator Schaltanfrage zum Schließen des Ventils.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
PosMan	BOOL	Manual Wertvorgabe für die Position	Lesen/schreiben
PosRbk	REAL	Unbearbeiteter Wert der Operator Wertevorgabe für die Position	Lesen/schreiben
ResetOp	BOOL	Operator Reset Schaltanforderung: 0→1: Operator Anfrage 1→0: Anfrage wurde verarbeitet.	Lesen/schreiben
MonEn	BOOL	FeedbackMonitoring aktivieren: 1: FeedbackMonitoring aktiv 0: FeedbackMonitoring inaktiv	Lesen/schreiben

 **Methoden**

Name	Beschreibung
SetOffline	Siehe <a href="#">Operation Mode [► 13]</a>
SetOperator	
SetAutomatic	
SetInternal	Siehe <a href="#">Source Mode [► 15]</a>
SetManual	

### 5.3.1.9 FB\_MTP\_PIDCtrlBase

FB_MTP_PIDCtrlBase		
WQC	BYTE	BOOL StateOpAct
OSLevel	BYTE	BOOL StateAutAct
StateChannel	BOOL	BOOL StateOffAct
StateOffAut	BOOL	BOOL SrcIntAct
StateOpAut	BOOL	BOOL SrcManAct
StateAutAut	BOOL	REAL SP
SrcChannel	BOOL	REAL MV
SrcManAut	BOOL	
SrcIntAut	BOOL	
PV	REAL	
PVScMin	REAL	
PVScMax	REAL	
PVUnit	INT	
SPInt	REAL	
SPScMin	REAL	
SPScMax	REAL	
SPUnit	INT	
SPIntMin	REAL	
SPIntMax	REAL	
SPManMin	REAL	
SPManMax	REAL	
MVMin	REAL	
MVMax	REAL	
MVUnit	INT	
MVScMin	REAL	
MVScMax	REAL	
P	REAL	
Ti	REAL	
Td	REAL	

Der Funktionsbaustein FB\_MTP\_PIDCtrlBase dient als Schnittstelle zur Verwendung eines PID-Reglers von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z. B. über OPC-UA). Wertvorgaben werden unabhängig voneinander von den Zustandsautomaten des [Operation Mode \[► 13\]](#) und des [Source Mode \[► 15\]](#) verwaltet. Die OPC-UA-Zugriffsrechte sind in den Variablen-Tabellen beschrieben.



Bei dem FB\_MTP\_PIDCtrlBase handelt es sich um einen abstrakten Funktionsbaustein. Dieser kann nicht instanziiert, sondern muss abgeleitet werden (siehe Syntax)!

#### Zustandsautomaten

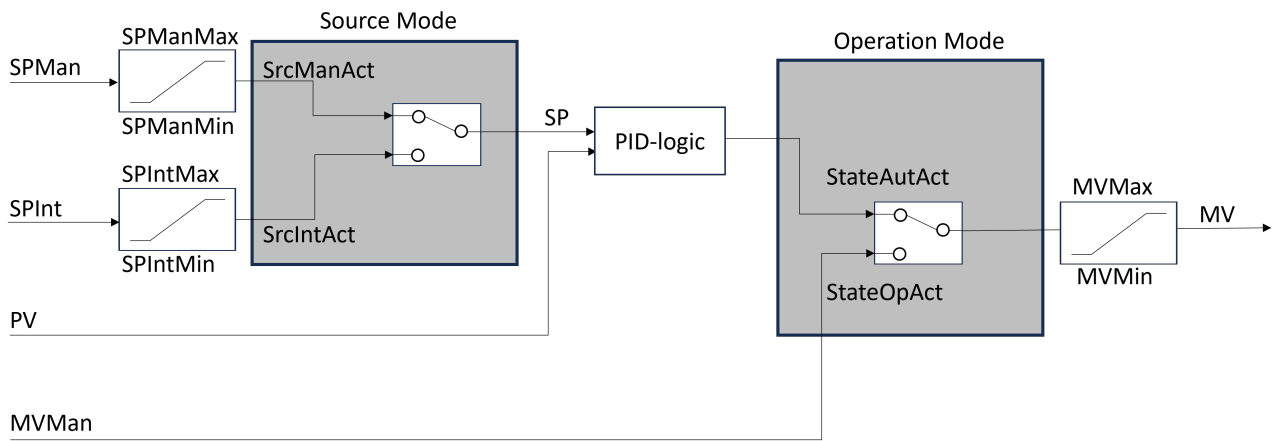
Über den Zustandsautomaten des [Source Mode \[► 15\]](#) wird die Quelle des Sollwerts  $SP^*$  verwaltet. Über den Zustandsautomaten des [Operation Mode \[► 13\]](#) wird die Quelle der ausgegebenen Stellgröße  $MV^*$  verwaltet: Ausgangswert der PID-Logik oder Vorgabe durch manuelle Bedienung  $MV_{Man}$ .

#### PID-Regler

Die PID-Logik (`PIDLogic()`) kann frei gewählt werden. Der Aufruf der PID-Logik erfolgt über die Methode `PIDLogic()`. Über die Eingänge  $P$ ,  $Ti$  und  $Td$  werden die Parameter für den Regler beschrieben. Der Ausgang  $MV$  beschreibt die aktuelle Stellgröße in Abhängigkeit der aktuellen Zustände der Zustandsautomaten. Der Ausgang  $SP$  gibt den aktuell verwendeten Sollwert der PID-Logik wieder.

Der Funktionsbaustein arbeitet nach dem folgenden Schema:





**Weitere Eigenschaften**

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]

OSLevel [[▶ 18](#)]

Wertskalierung [[▶ 19](#)]

Wertbegrenzung [[▶ 20](#)]

Einheit [[▶ 19](#)]

**Syntax**

```
FUNCTION_BLOCK FB_MTP_PIDCtrl EXTENDS FB_MTP_PIDCtrlBase
VAR
    ///Controller Toolbox (delete if you want to use own PID Logic)
    CTRL_PID: FB_CTRL_PID;
    PIDParams: ST_Ctrl_PID_PARAMS;
END_VAR
SUPER^();
```

**Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Operation Mode ▶ 13</a>	
MVUnit	INT	Einheit der Stellgröße	Lesen
MVSciMin	REAL	Minimalwert der Darstellung der Stellgröße	Lesen
MVSciMax	REAL	Maximalwert der Darstellung der Stellgröße	Lesen
MVMin	REAL	Minimalwert der Stellgröße	Lesen
MVMax	REAL	Maximalwert der Stellgröße	Lesen
Src*		Siehe <a href="#">Source Mode ▶ 15</a>	
SPInt	REAL	internal Sollwertvorgabe	Lesen
SPUnit	INT	Einheit des Sollwerts	Lesen
SPIntMin	REAL	Unterer Grenzwert der internal Sollwertvorgabe	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
SPIntMax	REAL	Oberer Grenzwert der PEA internen Sollwertvorgabe	Lesen
SPManMin	REAL	Unterer Grenzwert der manuellen Sollwertvorgabe	Lesen
SPManMax	REAL	Oberer Grenzwert der manuellen Sollwertvorgabe	Lesen
SPScImin	REAL	Minimalwert für die Darstellung der Sollwertvorgabe	Lesen
SPScIMax	REAL	Maximalwert für die Darstellung der Sollwertvorgabe	Lesen
PV	REAL	Istwert	Lesen
PVUnit	INT	Einheit des Istwerts	Lesen
PVScImin	REAL	Minimalwert für die Darstellung des Istwerts	Lesen
PVScIMax	REAL	Maximalwert für die Darstellung des Istwerts	Lesen
P	REAL	Proportionale Verstärkung des Reglers	Lesen
Ti	REAL	Nachstellzeit des Reglers [s]	Lesen
Td	REAL	Vorhaltezeit des Reglers [s]	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
MV	REAL	Aktuelle Stellgröße des Reglers	
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
SP	REAL	Aktuell verwendeter Sollwert des Reglers.	

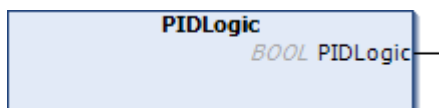
### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Operation Mode [► 13]</a>	
MVMan	REAL	Manual Vorgabe der Stellgröße	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
SPMan	REAL	Manual Sollwertvorgabe	Lesen/schreiben

### Methoden

Name	Beschreibung
PIDLogic	Implementierung der Logik zum Aufruf des PID-Reglers
ResetPID	Zurücksetzen des PID-Reglers

#### 5.3.1.9.1 PIDLogic



Diese Methode ermöglicht die Implementierung der Logik des PID-Reglers.

#### Syntax

```
METHOD PIDLogic : BOOL
```

Aufruf Beckhoff PID-Regler (Tc2\_ControllerToolbox wird benötigt)

```
//Set PID Params
PIDParams.tCtrlCycleTime := LREAL_TO_TIME(fCycleTime * 1000);
PIDParams.tTaskCycleTime := LREAL_TO_TIME(fCycleTime * 1000);
PIDParams.fKp             := P;
PIDParams.tTn            := REAL_TO_TIME(TI * 1000);
PIDParams.tTv            := REAL_TO_TIME(TD * 1000);
```

```
PIDParams.fOutMaxLimit := MVMax;
PIDParams.fOutMinLimit := MVMin;
PIDParams.bARWOnIPartOnly := TRUE;

//Call Controller Toolbox PID function block
CTRL_PID(
  fSetpointValue := SP,
  fActualValue := PV,
  eMode := eCTRL_MODE_ACTIVE,
  stParams := PIDParams);

//Write resulting Manipulated Value to MTP function block
MV:= LREAL_TO_REAL(CTRL_PID.fOut);
```

 Rückgabewert

Name	Typ	Beschreibung
PIDLogic	BOOL	

### 5.3.1.9.2 ResetPID

Diese Methode erlaubt die Implementierung einer Logik zum Zurücksetzen der PID-Logik.

**Syntax**

```
METHOD ResetPID : BOOL

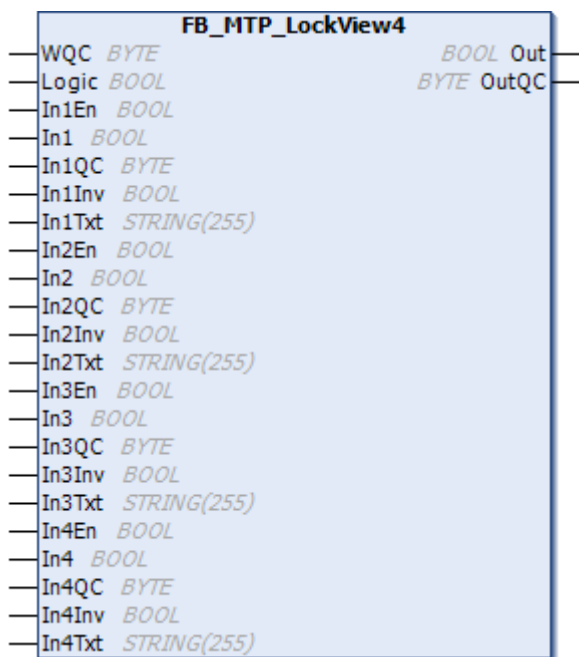
CTRL_PID(
  eMode := eCTRL_MODE_Reset,
  stParams:= PIDParams
);

//Write resulting Manipulated Value to MTP function block
MV:= LREAL_TO_REAL(CTRL_PID.fOut);
```

## 5.3.2 DiagnosticElements

Verriegelungsanzeigen visualisieren den aktuellen Zustand einer logischen Grundoperation AND/OR.

### 5.3.2.1 FB\_MTP\_LockView4



Der Funktionsbaustein `FB_MTP_LockView4` ist ein Objekt zur Visualisierung einer logischen Grundoperation `AND/OR` mit vier Eingängen. Diese wird über OPC-UA zur Verfügung gestellt.

Jeder Eingang kann aktiviert `*En`, invertiert `*Inv` und mit einem Text `*Txt` beschrieben werden. Zudem enthält jeder Eingang einen Qualitycode `*QC`. Er beschreibt unter welchen Bedingungen der Wert der Schnittstelle erfasst wurde und wie vertrauenswürdig dieser Wert ist.

Die Eingänge können als `AND`-Logik (`Logic = TRUE`) oder als `OR`-Logik (`Logic = FALSE`) ausgewertet werden. Das Verknüpfungsergebnis aus den Eingängen wird am Ausgang `Out` ausgegeben. Hierfür werden nur aktivierte Eingänge berücksichtigt. Das Ergebnis der `QC` der Eingänge wird am Ausgang `OutQC` ausgegeben.

### Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

### Eingänge

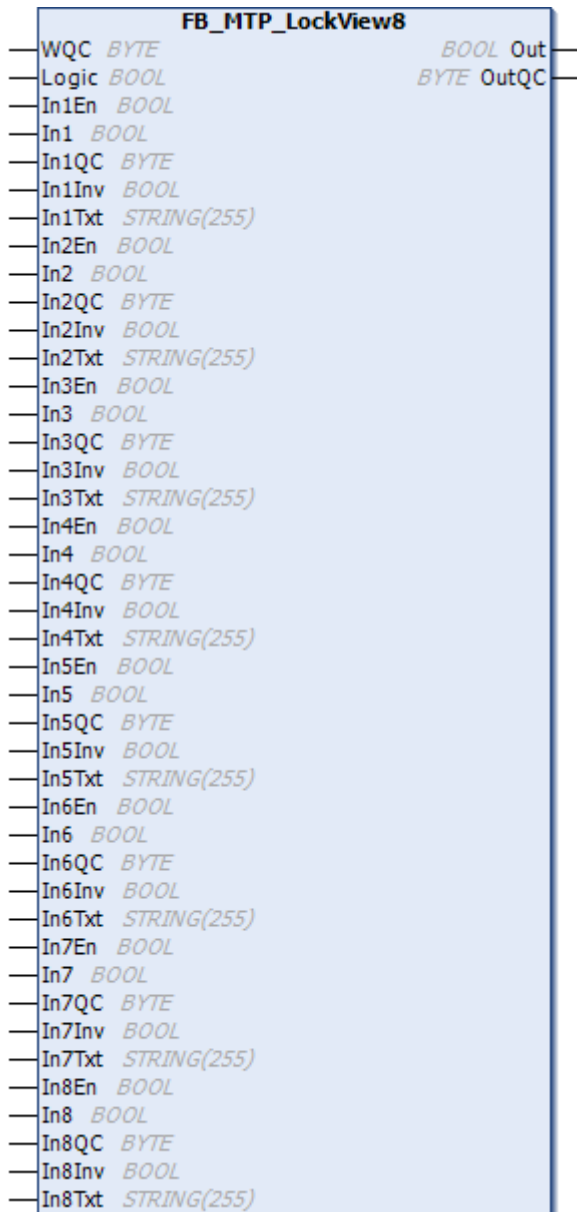
Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
Logic	BOOL	Logische Operation: 0: OR 1: AND	Lesen
In1En	BOOL	Eingang 1 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert	Lesen
In1	BOOL	Eingang 1 - Wert	Lesen
In1QC	BYTE	Eingang 1 – Quality Code	Lesen
In1Inv	BOOL	Eingang 1 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In1Txt	STRING	Eingang 1 - Beschreibungstext	Lesen
In2En	BOOL	Eingang 2 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In2	BOOL	Eingang 2 - Wert	Lesen
In2QC	BYTE	Eingang 2 – Quality Code	Lesen
In2Inv	BOOL	Eingang 2 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In2Txt	STRING	Eingang 2 - Beschreibungstext	Lesen
In3En	BOOL	Eingang 3 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In3	BOOL	Eingang 3 – Wert	Lesen
In3QC	BYTE	Eingang 3 – Quality Code	Lesen
In3Inv	BOOL	Eingang 3 invertieren: 0: Nicht invertiert	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		1: Invertiert	
In3Txt	STRING	Eingang 3 - Beschreibungstext	Lesen
In4En	BOOL	Eingang 4 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In4	BOOL	Eingang 4 - Wert	Lesen
In4QC	BYTE	Eingang 4 - <i>Quality Code</i>	Lesen
In4Inv	BOOL	Eingang 4 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In4Txt	STRING	Eingang 4 - Beschreibungstext	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
Out	BOOL	Ausgabe des Verknüpfungsergebnisses	Lesen
OutQC	BOOL	QC des Verknüpfungsergebnisses	Lesen

### 5.3.2.2 FB\_MTP\_LockView8



Der Funktionsbaustein **FB\_MTP\_LockView8** ist ein Objekt zur Visualisierung einer logischen Grundoperation **AND/OR** mit acht Eingängen. Diese wird über OPC-UA zur Verfügung gestellt.

Jeder Eingang kann aktiviert *\*En*, invertiert *\*Inv* und mit einem Text *\*Txt* beschrieben werden. Zudem enthält jeder Eingang einen Qualitycode *\*QC*. Er beschreibt unter welchen Bedingungen der Wert der Schnittstelle erfasst wurde und wie vertrauenswürdig dieser Wert ist.

Die Eingänge können als **AND-Logik** (*Logic = TRUE*) oder als **OR-Logik** (*Logic = FALSE*) ausgewertet werden. Das Verknüpfungsergebnis aus den Eingängen wird am Ausgang *Out* ausgegeben. Hierfür werden nur aktivierte Eingänge berücksichtigt. Das Ergebnis der QC der Eingänge wird am Ausgang *OutQC* ausgegeben.

#### Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
Logic	BOOL	Logische Operation: 0: OR 1: AND	Lesen
In1En	BOOL	Eingang 1 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert	Lesen
In1	BOOL	Eingang 1 - Wert	Lesen
In1QC	BYTE	Eingang 1 – Quality Code	Lesen
In1Inv	BOOL	Eingang 1 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In1Txt	STRING	Eingang 1 - Beschreibungstext	Lesen
In2En	BOOL	Eingang 2 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In2	BOOL	Eingang 2 - Wert	Lesen
In2QC	BYTE	Eingang 2 – Quality Code	Lesen
In2Inv	BOOL	Eingang 2 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In2Txt	STRING	Eingang 2 - Beschreibungstext	Lesen
In3En	BOOL	Eingang 3 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In3	BOOL	Eingang 3 – Wert	Lesen
In3QC	BYTE	Eingang 3 – Quality Code	Lesen
In3Inv	BOOL	Eingang 3 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In3Txt	STRING	Eingang 3 - Beschreibungstext	Lesen
In4En	BOOL	Eingang 4 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In4	BOOL	Eingang 4 - Wert	Lesen
In4QC	BYTE	Eingang 4 – Quality Code	Lesen
In4Inv	BOOL	Eingang 4 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In4Txt	STRING	Eingang 4 - Beschreibungstext	Lesen
In5En	BOOL	Eingang 5 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In5	BOOL	Eingang 5 – Wert	Lesen
In5QC	BYTE	Eingang 5 – Quality Code	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
In5Inv	BOOL	Eingang 5 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In5Txt	STRING	Eingang 5 - Beschreibungstext	Lesen
In6En	BOOL	Eingang 6 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In6	BOOL	Eingang 6 - Wert	Lesen
In6QC	BYTE	Eingang 6 – <i>Quality Code</i>	Lesen
In6Inv	BOOL	Eingang 6 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In6Txt	STRING	Eingang 6 - Beschreibungstext	Lesen
In7En	BOOL	Eingang 7 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In7	BOOL	Eingang 7 - Wert	Lesen
In7QC	BYTE	Eingang 7 – <i>Quality Code</i>	Lesen
In7Inv	BOOL	Eingang 7 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In7Txt	STRING	Eingang 7 - Beschreibungstext	Lesen
In8En	BOOL	Eingang 8 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In8	BOOL	Eingang 8 – Wert	Lesen
In8QC	BYTE	Eingang 8 – <i>Quality Code</i>	Lesen
In8Inv	BOOL	Eingang 8 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In8Txt	STRING	Eingang 8 - Beschreibungstext	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
Out	BOOL	Ausgabe des Verknüpfungsergebnisses	Lesen
OutQC	BOOL	QC des Verknüpfungsergebnisses	Lesen



### 5.3.2.3 FB\_MTP\_LockView16



Der Funktionsbaustein FB\_MTP\_LockView16 ist ein Objekt zur Visualisierung einer logischen Grundoperation AND/OR mit sechzehn Eingängen. Diese wird über OPC-UA zur Verfügung gestellt.

Jeder Eingang kann aktiviert \*En, invertiert \*Inv und mit einem Text \*Txt beschrieben werden. Zudem enthält jeder Eingang einen Qualitycode \*QC. Er beschreibt unter welchen Bedingungen der Wert der Schnittstelle erfasst wurde und wie vertrauenswürdig dieser Wert ist.

Die Eingänge können als AND-Logik (`Logic = TRUE`) oder als OR-Logik (`Logic = FALSE`) ausgewertet werden. Das Verknüpfungsergebnis aus den Eingängen wird am Ausgang `Out` ausgegeben. Hierfür werden nur aktivierte Eingänge berücksichtigt. Das Ergebnis der QC der Eingänge wird am Ausgang `OutQC` ausgegeben.

### Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
Logic	BOOL	Logische Operation: 0: OR 1: AND	Lesen
In1En	BOOL	Eingang 1 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert	Lesen
In1	BOOL	Eingang 1 - Wert	Lesen
In1QC	BYTE	Eingang 1 – <code>Quality Code</code>	Lesen
In1Inv	BOOL	Eingang 1 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In1Txt	STRING	Eingang 1 - Beschreibungstext	Lesen
In2En	BOOL	Eingang 2 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In2	BOOL	Eingang 2 - Wert	Lesen
In2QC	BYTE	Eingang 2 – <code>Quality Code</code>	Lesen
In2Inv	BOOL	Eingang 2 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In2Txt	STRING	Eingang 2 - Beschreibungstext	Lesen
In3En	BOOL	Eingang 3 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In3	BOOL	Eingang 3 – Wert	Lesen
In3QC	BYTE	Eingang 3 – <code>Quality Code</code>	Lesen
In3Inv	BOOL	Eingang 3 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In3Txt	STRING	Eingang 3 - Beschreibungstext	Lesen
In4En	BOOL	Eingang 4 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In4	BOOL	Eingang 4 - Wert	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
In4QC	BYTE	Eingang 4 – <i>Quality Code</i>	Lesen
In4Inv	BOOL	Eingang 4 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In4Txt	STRING	Eingang 4 - Beschreibungstext	Lesen
In5En	BOOL	Eingang 5 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In5	BOOL	Eingang 5 – Wert	Lesen
In5QC	BYTE	Eingang 5 – <i>Quality Code</i>	Lesen
In5Inv	BOOL	Eingang 5 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In5Txt	STRING	Eingang 5 - Beschreibungstext	Lesen
In6En	BOOL	Eingang 6 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In6	BOOL	Eingang 6 - Wert	Lesen
In6QC	BYTE	Eingang 6 – <i>Quality Code</i>	Lesen
In6Inv	BOOL	Eingang 6 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In6Txt	STRING	Eingang 6 - Beschreibungstext	Lesen
In7En	BOOL	Eingang 7 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In7	BOOL	Eingang 7 - Wert	Lesen
In7QC	BYTE	Eingang 7 – <i>Quality Code</i>	Lesen
In7Inv	BOOL	Eingang 7 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In7Txt	STRING	Eingang 7 - Beschreibungstext	Lesen
In8En	BOOL	Eingang 8 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In8	BOOL	Eingang 8 – Wert	Lesen
In8QC	BYTE	Eingang 8 – <i>Quality Code</i>	Lesen
In8Inv	BOOL	Eingang 8 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In8Txt	STRING	Eingang 8 - Beschreibungstext	Lesen
In9En	BOOL	Eingang 9 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In9	BOOL	Eingang 9 – Wert	Lesen
In9QC	BYTE	Eingang 9 – <i>Quality Code</i>	Lesen
In9Inv	BOOL	Eingang 9 invertieren:	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
		0: Nicht invertiert 1: Invertiert	
In9Txt	STRING	Eingang 9 - Beschreibungstext	Lesen
In10En	BOOL	Eingang 10 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In10	BOOL	Eingang 10 - Wert	Lesen
In10QC	BYTE	Eingang 10 – <i>Quality Code</i>	Lesen
In10Inv	BOOL	Eingang 10 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In10Txt	STRING	Eingang 10 - Beschreibungstext	Lesen
In11En	BOOL	Eingang 11 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In11	BOOL	Eingang 11 - Wert	Lesen
In11QC	BYTE	Eingang 11 – <i>Quality Code</i>	Lesen
In11Inv	BOOL	Eingang 11 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In11Txt	STRING	Eingang 11 - Beschreibungstext	Lesen
In12En	BOOL	Eingang 12 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In12	BOOL	Eingang 12 - Wert	Lesen
In12QC	BYTE	Eingang 12 – <i>Quality Code</i>	Lesen
In12Inv	BOOL	Eingang 12 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In12Txt	STRING	Eingang 12 - Beschreibungstext	Lesen
In13En	BOOL	Eingang 13 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In13	BOOL	Eingang 13 – Wert	Lesen
In13QC	BYTE	Eingang 13 – <i>Quality Code</i>	Lesen
In13Inv	BOOL	Eingang 13 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In13Txt	STRING	Eingang 13 - Beschreibungstext	Lesen
In14En	BOOL	Eingang 14 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In14	BOOL	Eingang 14 – Wert	Lesen
In14QC	BYTE	Eingang 14 – <i>Quality Code</i>	Lesen
In14Inv	BOOL	Eingang 14 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen

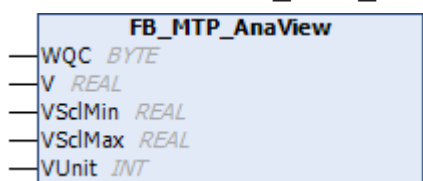
Name	Typ	Beschreibung	OPC-UA-Zugriff
In14Txt	STRING	Eingang 14 - Beschreibungstext	Lesen
In15En	BOOL	Eingang 15 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In15	BOOL	Eingang 15 - Wert	Lesen
In15QC	BYTE	Eingang 15 – Quality Code	Lesen
In15Inv	BOOL	Eingang 15 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In15Txt	STRING	Eingang 15 - Beschreibungstext	Lesen
In16En	BOOL	Eingang 16 aktivieren: 0: Eingang nicht aktiviert. 1: Eingang aktiviert.	Lesen
In16	BOOL	Eingang 16 – Wert	Lesen
In16QC	BYTE	Eingang 16 – Quality Code	Lesen
In16Inv	BOOL	Eingang 16 invertieren: 0: Nicht invertiert 1: Invertiert	Lesen
In16Txt	STRING	Eingang 16 - Beschreibungstext	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
Out	BOOL	Ausgabe des Verknüpfungsergebnisses	Lesen
OutQC	BOOL	QC des Verknüpfungsergebnisses	Lesen

### 5.3.3 IndicatorElements

#### 5.3.3.1 FB\_MTP\_AnaView



Der Funktionsbaustein FB\_MTP\_AnaView stellt den analogen Eingangswert *v* zusammen mit Skalenwerten und Einheit über OPC-UA zur Verfügung.

**Weitere Eigenschaften**

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

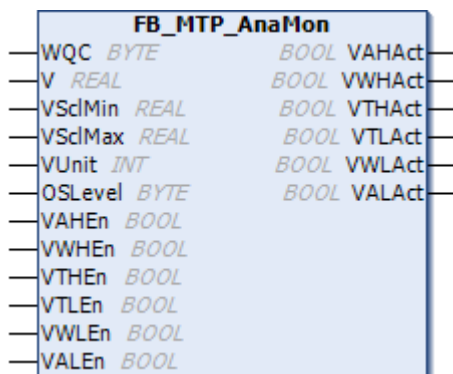
[Wertskalierung \[► 19\]](#)

[Einheit \[► 19\]](#)

### 🚩 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
V	REAL	Wert	Lesen
VUnit	INT	Einheit des Werts	Lesen
VScImin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen

### 5.3.3.2 FB\_MTP\_AnaMon



Der Funktionsbaustein FB\_MTP\_AnaMon stellt den analogen Eingangswert *v* für eine manuelle Bedienung (z. B. über OPC-UA) zur Verfügung. Dieser Wert *v* kann auf bis zu sechs Grenzen überwacht werden.

#### Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[OSLevel \[► 18\]](#)

[Wertskalierung \[► 19\]](#)

[Einheit \[► 19\]](#)

[Grenzwertüberwachung \[► 22\]](#)

#### Vererbungshierarchie

FB\_MTP\_AnaView

    FB\_MTP\_AnaMon

### 🚩 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
V	REAL	Wert	Lesen
VUnit	INT	Einheit des Werts	Lesen
VSciMin	REAL	Skalenanfang für Wertdarstellung	Lesen
VSciMax	REAL	Skalenende für Wertdarstellung	Lesen
VAHEn	BOOL	Grenzwert Alarm High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VALEn	BOOL	Grenzwert Alarm Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VWHEn	BOOL	Grenzwert Warning High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VWLEn	BOOL	Grenzwert Warning Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VTHEn	BOOL	Grenzwert Tolerance High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VTLEn	BOOL	Grenzwert Tolerance Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VAHAct	BOOL	Alarm High Grenzwert überschritten.	Lesen
VALAct	BOOL	Alarm Low Grenzwert unterschritten.	Lesen
VWHAAct	BOOL	Warning High Grenzwert überschritten.	Lesen
VWLAct	BOOL	Warning Low Grenzwert unterschritten.	Lesen
VTHAct	BOOL	Tolerance High Grenzwert überschritten.	Lesen
VTLAct	BOOL	Tolerance Low Grenzwert unterschritten.	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VAHLim	REAL	Alarm High Grenzwert	Lesen/schreiben
VALLim	REAL	Alarm Low Grenzwert	Lesen/schreiben
VWHLim	REAL	Warning High Grenzwert	Lesen/schreiben
VWLLim	REAL	Warning Low Grenzwert	Lesen/schreiben
VTHLim	REAL	Tolerance High Grenzwert	Lesen/schreiben
VTLLim	REAL	Tolerance Low Grenzwert	Lesen/schreiben

### 5.3.3.3 FB\_MTP\_DIntView



Der Funktionsbaustein FB\_MTP\_DIntView stellt den ganzzahligen Eingangswert  $v$  über OPC-UA zur Verfügung.

#### Weitere Eigenschaften

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]

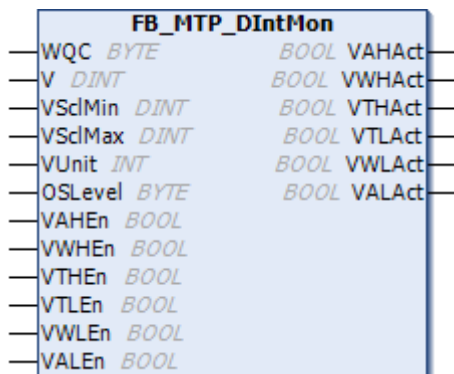
Wertskalierung [[▶ 19](#)]

Einheit [[▶ 19](#)]

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
V	DINT	Wert	Lesen
VUnit	INT	Einheit des Werts	Lesen
VScMin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScMax	DINT	Skalenende für Wertdarstellung	Lesen

### 5.3.3.4 FB\_MTP\_DIntMon



Der Funktionsbaustein FB\_MTP\_DIntMon stellt den ganzzahligen Eingangswert  $v$  für eine manuelle Bedienung (z. B. über OPC-UA) zur Verfügung. Dieser Wert  $v$  kann auf bis zu sechs Grenzen überwacht werden.

#### Weitere Eigenschaften

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]



[OSLevel \[▶\\_18\]](#)

[Wertskalierung \[▶\\_19\]](#)

[Grenzwertüberwachung \[▶\\_22\]](#)

**Vererbungshierarchie**

FB\_MTP\_DIntView

    FB\_MTP\_DIntMon

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
V	DINT	Wert	Lesen
VUnit	INT	Einheit des Werts	Lesen
VScImin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScImax	DINT	Skalenende für Wertdarstellung	Lesen
VAHEn	BOOL	Grenzwert Alarm High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VALEn	BOOL	Grenzwert Alarm Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VWHEn	BOOL	Grenzwert Warning High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VWLEn	BOOL	Grenzwert Warning Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VTHEn	BOOL	Grenzwert Tolerance High überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen
VTLEn	BOOL	Grenzwert Tolerance Low überwachen: 1: Überwachung aktiv 0: Keine Überwachung	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VAHAct	BOOL	Alarm High Grenzwert überschritten.	Lesen
VALAct	BOOL	Alarm Low Grenzwert unterschritten.	Lesen
VWHAct	BOOL	Warning High Grenzwert überschritten.	Lesen
VWLAct	BOOL	Warning Low Grenzwert unterschritten.	Lesen
VTHAct	BOOL	Tolerance High Grenzwert überschritten.	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
VTLAct	BOOL	Tolerance Low Grenzwert unterschritten.	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
VAHLim	DINT	Alarm High Grenzwert	Lesen/schreiben
VALLim	DINT	Alarm Low Grenzwert	Lesen/schreiben
VWHLim	DINT	Warning High Grenzwert	Lesen/schreiben
VWLLim	DINT	Warning Low Grenzwert	Lesen/schreiben
VTHLim	DINT	Tolerance High Grenzwert	Lesen/schreiben
VTLLim	DINT	Tolerance Low Grenzwert	Lesen/schreiben

### 5.3.3.5 FB\_MTP\_BinView



Der Funktionsbaustein `FB_MTP_BinView` stellt den binären Eingangswert `v` über OPC-UA zur Verfügung. Den Zuständen `TRUE` und `FALSE` des Eingangswerts kann über die Eingänge `VState0` und `VState1` ein Text zugewiesen werden.

#### Weitere Eigenschaften

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]

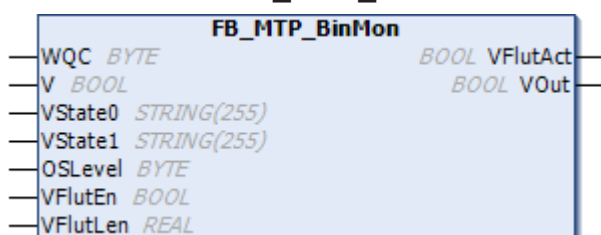
Wertskalierung [[▶ 19](#)]

Einheit [[▶ 19](#)]

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
V	BOOL	Wert	Lesen
VState0	STRING	Text für Wert <code>FALSE</code>	Lesen
VState1	STRING	Text für Wert <code>TRUE</code>	Lesen

### 5.3.3.6 FB\_MTP\_BinMon



Der Funktionsbaustein `FB_MTP_BinMon` stellt den binären Eingangswert `V` für eine manuelle Bedienung (z.B. über OPC-UA) zur Verfügung. Den Zuständen `TRUE` und `FALSE` des Eingangswerts kann über die Eingänge `VState0` und `VState1` ein Text zugewiesen werden.

**Überwachung**

Der binäre Eingangswert `V` kann auf Flattern (Wechsel des Eingangssignals) überwacht werden. Der Ausgang `VFlutAct` geht auf `TRUE`, wenn der Wechsel des Eingangssignals pro Zeitintervall `VFlutTi` den Maximalwert `VFlutCnt` überschreitet.

Das Ausgangssignal `VOut` entspricht dem entprellten Eingangswert `V.VOut` und folgt einem Wechsel des Eingangssignals `V` nach Ablauf von `VFlutLen`. Dabei muss das Eingangssignal `V` seinen Wert für diese Zeit beibehalten.

**Weitere Eigenschaften**

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]

OSLevel [[▶ 18](#)]

**Vererbungshierarchie**

FB\_MTP\_BinView

    FB\_MTP\_BinMon

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
V	BOOL	Wert	Lesen
VState0	STRING	Text für Wert <code>FALSE</code>	Lesen
VState1	STRING	Text für Wert <code>TRUE</code>	Lesen
VFlutEn	BOOL	Flutterüberwachung aktivieren: 1: Aktiviert 0: Deaktiviert	Lesen
VFlutLen	REAL	Prellzeit	Lesen

 **Ausgänge**

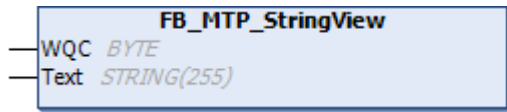
Name	Typ	Beschreibung	OPC-UA-Zugriff
VOut	BOOL	Ausgangswert	Lesen
VFlutAct	BOOL	Eingangswert flattert	Lesen
Vout	BOOL	Entprellter Eingangswert	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VFlutTi	REAL	Dauer der Zeitintervalls	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
VFlutCnt	INT	Maximale Eingangssignalwechsel pro Zeitintervall	Lesen/schreiben

### 5.3.3.7 FB\_MTP\_StringView



Der Funktionsbaustein `FB_MTP_StringView` stellt die Eingangsvariable `Text` über OPC-UA zur Verfügung.

#### Weitere Eigenschaften

Name des Objekts [\[► 18\]](#)

Objektbeschreibung [\[► 18\]](#)

WQC [\[► 19\]](#)

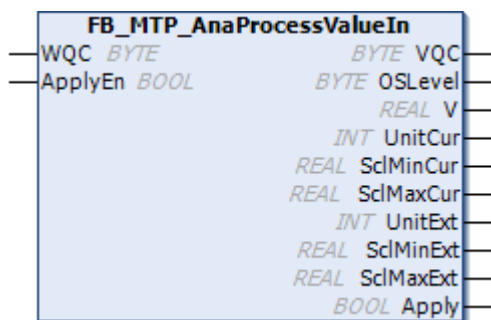
#### 🔴 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
Text	STRING	Dynamischer Text	Lesen

## 5.3.4 InputElements

`InputElements` sind prozesstechnische Eingangswerte, welche z. B. über OPC-UA als Eingangswerte geschrieben werden können. Es ist möglich, den Wertebereich (Einheit mit Wertskalierung) anzupassen, um eine korrekte Darstellung der Werte zu gewährleisten.

### 5.3.4.1 FB\_MTP\_AnaProcessValueIn



Der Funktionsbaustein `FB_MTP_AnaProcessValueIn` stellt einen analogen Prozesswert `V` mit zugehörigem `ValueQualityCode VQC` von einem übergeordneten System (z. B. über OPC-UA) zur Verfügung.

Über die Variablen `UnitCur`, `SclMinCur` und `SclMaxCur` werden die aktuelle Einheit und die aktuelle Wertskalierung angezeigt.

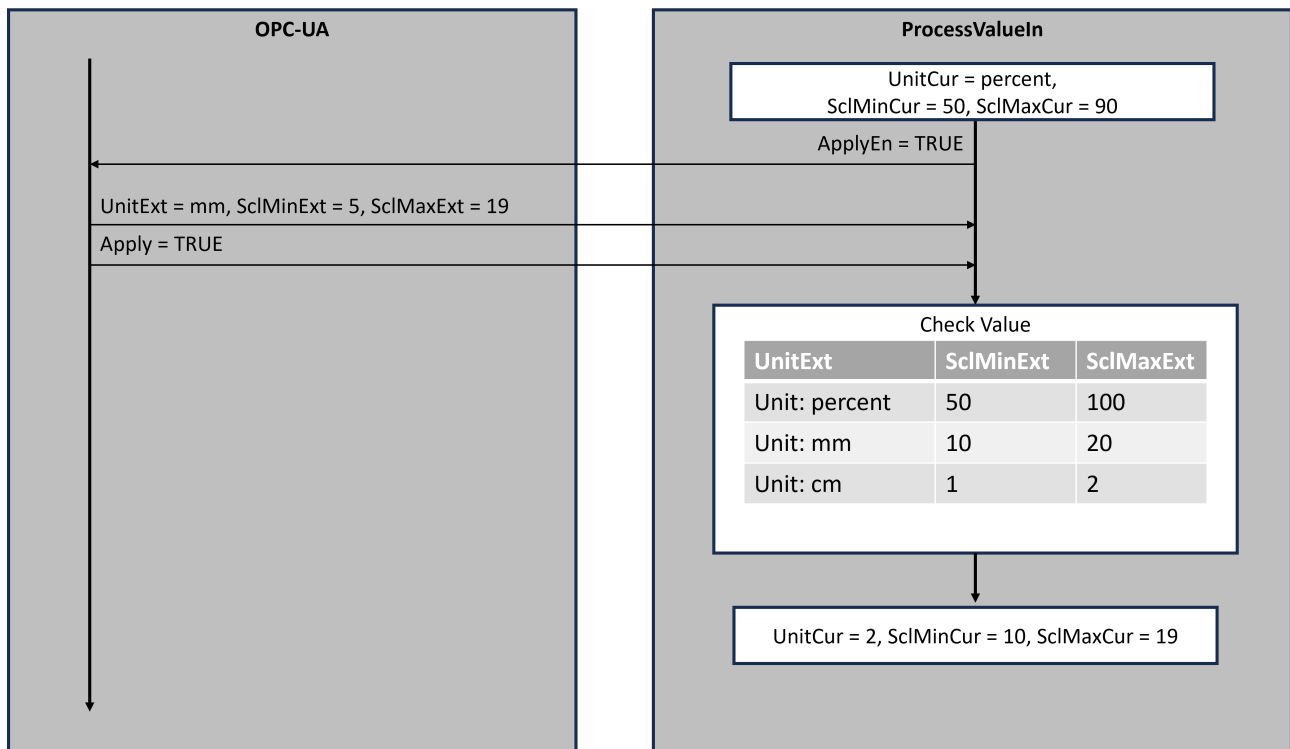
#### Übernahme einer neuen Konfiguration

Die Vorgabe einer neuen Konfiguration erfolgt über die Variablen `UnitExt`, `SclMinExt` und `SclMaxExt`.

Für eine Änderung der Konfiguration (Einheit und Wertskalierung) wird SPS-seitig über die Eingangsvariable `Configs` ein Array vom Typ `ST_MTP_InputElementConfig` [\[► 119\]](#) mit einer Liste mit unterstützten Konfigurationen übergeben. In diesem Array werden die Wertskalierungen für die jeweiligen Einheiten hinterlegt.

Die Freigabe für die Übernahme der Konfiguration wird SPS-seitig über die Variable `ApplyEn = TRUE` signalisiert. Anschließend kann über Variable `Apply = TRUE` die Übernahme angefordert werden.

Während der Übernahme wird `UnitExt` mit den hinterlegten möglichen Grenzwerten in `Configs` verglichen. Bei Überschreitung bzw. Unterschreitung wird der hinterlegte obere bzw. untere Grenzwert verwendet.



**Eingänge**

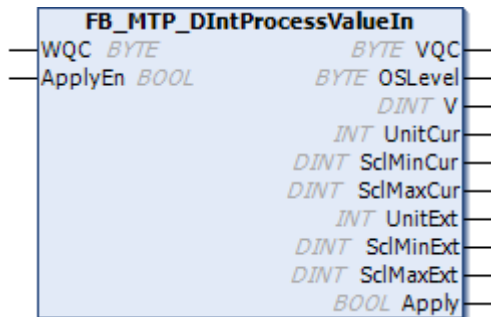
Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
Configs	ARRAY [*] OF ST_MT P_InputE lementsC onfig	Mögliche Konfigurationen	-
ApplyEn	BOOL	Freigabe für Anforderung: Konfiguration übernehmen.	Lesen

**Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VQC	BYTE	Quality Code für den Wert	Lesen/schreiben
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
V	REAL	Eingehender Prozesswert über OPC-UA	Lesen/schreiben
UnitCur	INT	Aktuelle Einheit	Lesen
SclMinCur	REAL	Aktueller Skalenanfang	Lesen
SclMaxCur	REAL	Aktuelles Skalenende	Lesen
UnitExt	INT	Konfigurationseingang für die Einheit	Lesen/schreiben
SclMinExt	REAL	Konfigurationseingang für den Skalenanfang	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
SclMaxExt	REAL	Konfigurationseingang für das Skaleneende	Lesen/schreiben
Apply	BOOL	Anforderung: Konfiguration (Unit* und Scl*) übernehmen.	Lesen/schreiben

### 5.3.4.2 FB\_MTP\_DIntProcessValueIn



Der Funktionsbaustein `FB_MTP_DIntProcessValueIn` stellt einen ganzzahligen Prozesswert `V` mit zugehörigem `ValueQualityCode VQC` von einem übergeordneten System (z. B. über OPC-UA) zur Verfügung.

Über die Variablen `UnitCur`, `SclMinCur` und `SclMaxCur` werden die aktuelle Einheit und die aktuelle Wertskalierung angezeigt.

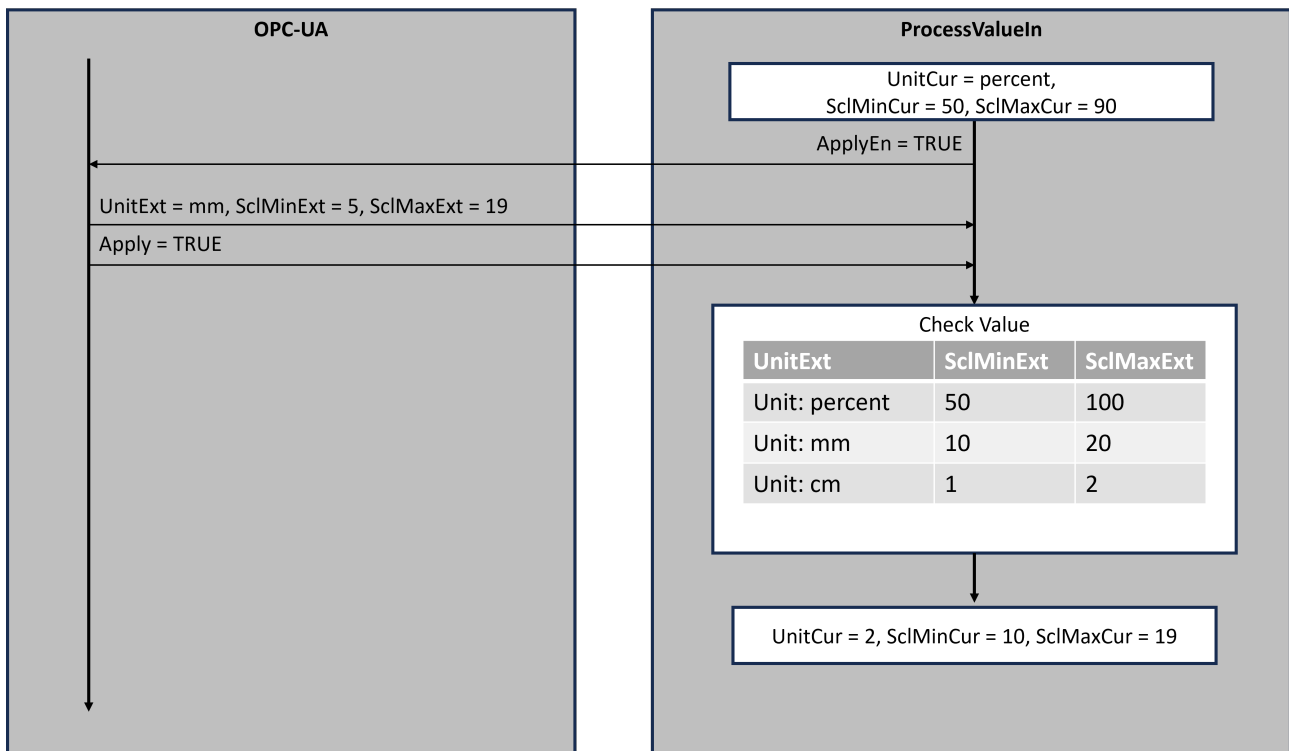
#### Übernahme einer neuen Konfiguration

Die Vorgabe einer neuen Konfiguration erfolgt über die Variablen `UnitExt`, `SclMinExt` und `SclMaxExt`.

Für eine Änderung der Konfiguration (Einheit und Wertskalierung) wird SPS-seitig über die Eingangsvariable `Configs` ein Array vom Typ `ST_MTP_InputElementConfig` [▶ 119] mit einer Liste mit unterstützten Konfigurationen übergeben. In diesem Array werden die Wertskalierungen für die jeweiligen Einheiten hinterlegt.

Die Freigabe für die Übernahme der Konfiguration wird SPS-seitig über die Variable `ApplyEn = TRUE` signalisiert. Anschließend kann über Variable `Apply = TRUE` die Übernahme angefordert werden.

Während der Übernahme wird `UnitExt` mit den hinterlegten möglichen Grenzwerten in `Configs` verglichen. Bei Überschreitung bzw. Unterschreitung wird der hinterlegte obere bzw. untere Grenzwert verwendet.



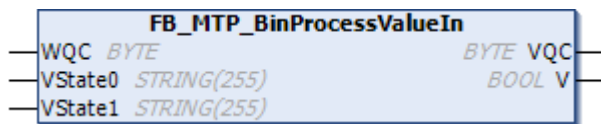
**Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
Configs	ARRAY [*] OF ST_MTP_InputElementsConfig	Mögliche Konfigurationen	-
ApplyEn	BOOL	Freigabe für Anforderung: Konfiguration übernehmen.	Lesen

**Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VQC	BYTE	Quality Code für den Wert	Lesen/schreiben
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
V	DINT	Eingehender Prozesswert über OPC-UA	Lesen/schreiben
UnitCur	INT	Aktuelle Einheit des Prozesswerts	Lesen
SclMinCur	DINT	Aktueller Skalenanfang	Lesen
SclMaxCur	DINT	Aktuelles Skalende	Lesen
UnitExt	INT	Konfigurationseingang für die Einheit	Lesen/schreiben
SclMinExt	DINT	Konfigurationseingang für den Skalenanfang	Lesen/schreiben
SclMaxExt	DINT	Konfigurationseingang für das Skalende	Lesen/schreiben
Apply	BOOL	Anforderung: Konfiguration (Scl* und Unit*) übernehmen.	Lesen/schreiben

### 5.3.4.3 FB\_MTP\_BinProcessValueIn



Der Funktionsbaustein `FB_MTP_BinProcessValueIn` stellt den binären Wert `v` von einem übergeordneten System (z. B. über OPC-UA) zur Verfügung. Den Zuständen `TRUE` und `FALSE` des Werts kann über die Eingänge `VState0` und `VState1` ein Text zugewiesen werden.

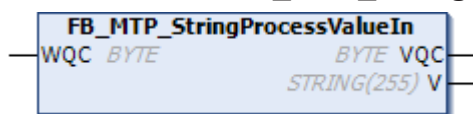
#### 🔌 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
VState0	String	Text für Wert = <code>FALSE</code>	Lesen
VState1	String	Text für Wert = <code>TRUE</code>	Lesen

#### 🔌 Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
VQC	BYTE	Quality Code für den Wert	Lesen/schreiben
V	BOOL	Wert	Lesen/schreiben

### 5.3.4.4 FB\_MTP\_StringProcessValueIn



Der Funktionsbaustein `FB_MTP_StringProcessValueIn` stellt einen Prozesswert `v` von einem übergeordneten System (z. B. über OPC-UA) zur Verfügung.

#### 🔌 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen

#### 🔌 Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
VQC	BYTE	Quality Code für den Wert	Lesen/schreiben
V	STRING	Eingehender Prozesswert über OPC-UA	Lesen/schreiben

## 5.3.5 OutputElements

`OutputElements` sind Instanzen von `IndicatorElements` [► 69], welche in der MTP-Datei als `OutputElements` definiert werden. Sie können Informationen für andere Anlagenbereiche beinhalten und werden z. B. einem Leitsystem über OPC-UA zur Verfügung gestellt.

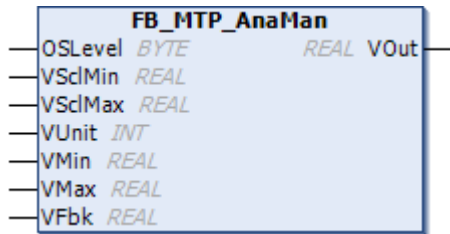




Für die Verwendung von OutputElements wird die Verwendung des MTP-Engineerings und des darin enthaltenen MTP-Exports benötigt!

### 5.3.6 OperationElements

#### 5.3.6.1 FB\_MTP\_AnaMan



Der Funktionsbaustein FB\_MTP\_AnaMan ist ein Objekt für eine analoge Wertvorgabe einer manuellen Bedienung VMan (z. B. über OPC-UA). Der ausgegebene Wert VOut entspricht der durch die Wertbegrenzung [▶ 20] limitierten Wertvorgabe.

#### Weitere Eigenschaften

Name des Objekts [▶ 18]

Objektbeschreibung [▶ 18]

OSLevel [▶ 18]

Rückmeldung [▶ 20]

Wertskalierung [▶ 19]

Einheit [▶ 19]

#### 🔌 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
VUnit	INT	Einheit des Werts	Lesen
VMin	REAL	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	REAL	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	REAL	Rückmeldewert	Lesen
VScIMin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen

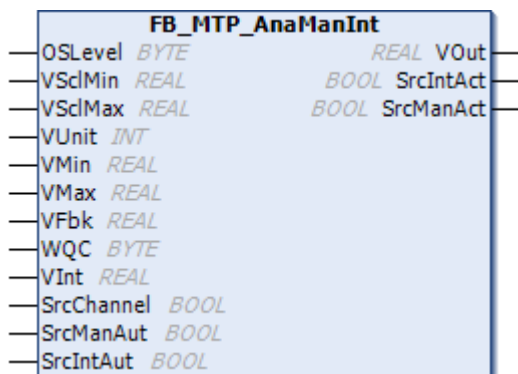
#### 🔌 Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
VOut	REAL	Ausgegebener Wert der aktuellen Wertvorgabe unter Berücksichtigung der Grenzwerte	Lesen

## Externe Variablen

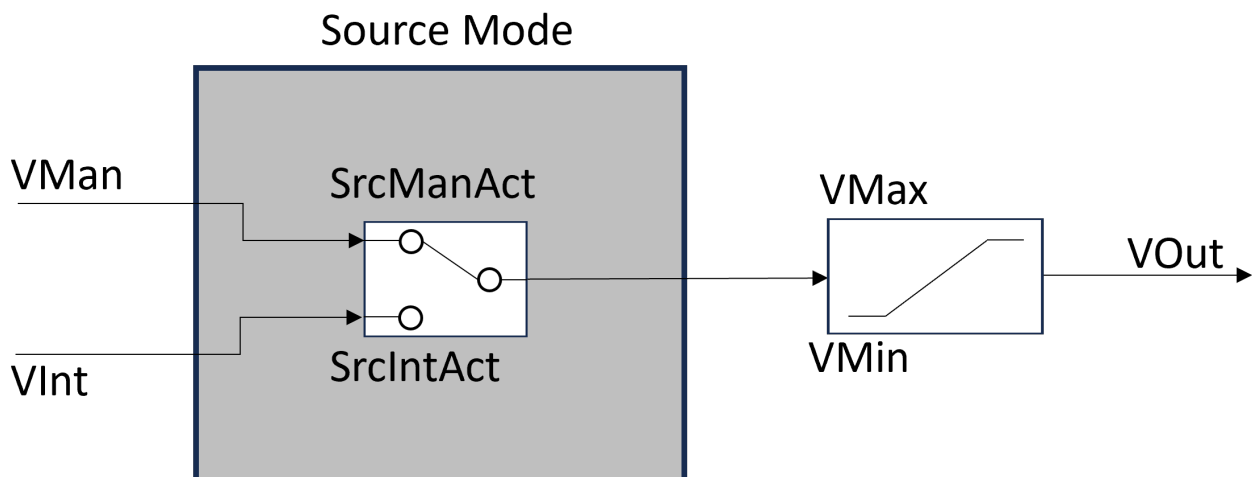
Name	Typ	Beschreibung	OPC-UA-Zugriff
VMan	BOOL	Manual Wertvorgabe	Lesen/schreiben

## 5.3.6.2 FB\_MTP\_AnaManInt



Der Funktionsbaustein FB\_MTP\_AnaManInt ist ein Objekt für eine analoge Wertvorgabe von unterschiedlichen Quellen: interne SPS-Logik VInt oder manuelle Bedienung VMan (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten des [Source Mode](#) [► 15] verwaltet.

Der ausgegebene Wert VOut entspricht der durch die [Wertbegrenzung](#) [► 20] limitierten Wertvorgabe.



## Weitere Eigenschaften

[Name des Objekts](#) [► 18]

[Objektbeschreibung](#) [► 18]

[WQC](#) [► 19]

[OSLevel](#) [► 18]

[Rückmeldung](#) [► 20]

[Wertskalierung](#) [► 19]

[Einheit](#) [► 19]

## Vererbungshierarchie

FB\_MTP\_AnaMan

FB\_MTP\_AnaManInt

**Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VInt	REAL	Internal Wertvorgabe. (Relevant, wenn SrcIntAct = TRUE)	Lesen
VUnit	INT	Einheit des Werts	Lesen
VMin	REAL	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	REAL	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	REAL	Rückmeldewert	Lesen
VScIMin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen

**Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VOut	REAL	Ausgegebener Wert der aktuellen Wertvorgabe unter Berücksichtigung der Grenzwerte	Lesen

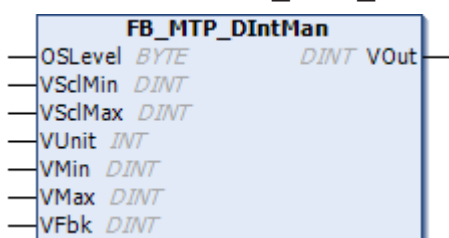
**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
VMan	REAL	Manual Wertvorgabe (Relevant, wenn SrcManAct = TRUE)	Lesen
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	

**Methoden**

Name	Beschreibung
SetInternal	Siehe <a href="#">Source Mode [▶ 15]</a>
SetManual	

**5.3.6.3 FB\_MTP\_DIntMan**



Der Funktionsbaustein `FB_MTP_DIntMan` ist ein Objekt für eine ganzzahlige Wertvorgabe einer manuellen Bedienung `VMan` (z. B. über OPC-UA). Der ausgegebene Wert `VOut` entspricht der durch die Wertbegrenzung [► 20] limitierten Wertvorgabe.

### Weitere Eigenschaften

Name des Objekts [► 18]

Objektbeschreibung [► 18]

OSLevel [► 18]

Rückmeldung [► 20]

Wertskalierung [► 19]

Einheit [► 19]

### 🔌 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
VUnit	INT	Einheit des Werts	Lesen
VMin	DINT	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	DINT	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	DINT	Rückmeldewert	Lesen
VScImin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScImax	DINT	Skalenende für Wertdarstellung	Lesen

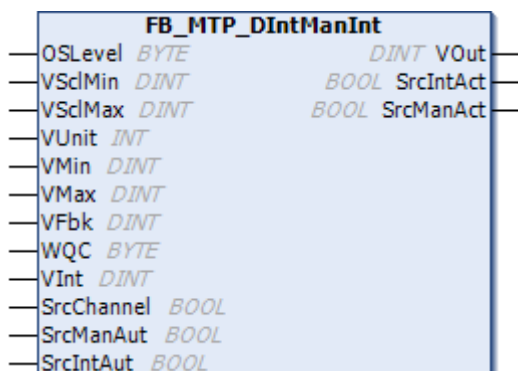
### 🔌 Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
VOut	DINT	Ausgegebener Wert der aktuellen Wertvorgabe unter Berücksichtigung der Grenzwerte	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
VMan	REAL	Manual Wertvorgabe	Lesen/schreiben

#### 5.3.6.4 FB\_MTP\_DIntManInt



Der Funktionsbaustein `FB_MTP_DIntManInt` ist ein Objekt für eine ganzzahlige Wertvorgabe von unterschiedlichen Quellen: interne SPS-Logik `VInt` oder manuelle Bedienung `VMan` (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten des [Source Mode \[▶ 15\]](#) verwaltet.

Der ausgegebene Wert `VOut` entspricht der durch die [Wertbegrenzung \[▶ 20\]](#) limitierten Wertvorgabe.

**Weitere Eigenschaften**

[Name des Objekts \[▶ 18\]](#)

[Objektbeschreibung \[▶ 18\]](#)

[WQC \[▶ 19\]](#)

[OSLevel \[▶ 18\]](#)

[Rückmeldung \[▶ 20\]](#)

[Wertskalierung \[▶ 19\]](#)

[Einheit \[▶ 19\]](#)

**Vererbungshierarchie**

FB\_MTP\_DIntMan

    FB\_MTP\_DIntManInt

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VInt	DINT	Internal Wertvorgabe (Relevant, wenn <code>SrcIntAct = TRUE</code> )	Lesen
VUnit	INT	Einheit des Werts	Lesen
VMin	DINT	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	DINT	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	DINT	Rückmeldewert	Lesen
VScIMin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScIMax	DINT	Skalenende für Wertdarstellung	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VOut	DINT	Ausgegebener Wert der aktuellen Wertvorgabe unter Berücksichtigung der Grenzwerte	Lesen

**Externe Variablen**

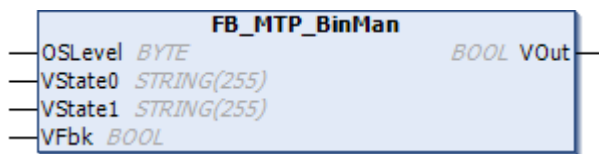
Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	

Name	Typ	Beschreibung	OPC-UA-Zugriff
VMan	REAL	Manual Wertvorgabe. (Relevant, wenn SrcManAct = TRUE)	Lesen

### Methoden

Name	Beschreibung
SetInternal	Siehe <a href="#">Source Mode</a> [▶ 15]
SetManual	

### 5.3.6.5 FB\_MTP\_BinMan



Der Funktionsbaustein FB\_MTP\_BinMan ist ein Objekt für eine binäre Wertvorgabe einer manuellen Bedienung VMan (z. B. über OPC-UA). Den Zuständen TRUE und FALSE der Wertvorgabe werden über die Eingangsvariablen VState0 und VState1 Zustandstexte zugewiesen.

#### Weitere Eigenschaften

[Name des Objekts](#) [▶ 18]

[Objektbeschreibung](#) [▶ 18]

[OSLevel](#) [▶ 18]

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
VState0	STRING	Text für Wert VOut = FALSE	Lesen
VState1	STRING	Text für Wert VOut = TRUE	Lesen
VFbk	BOOL	Rückmeldewert	Lesen

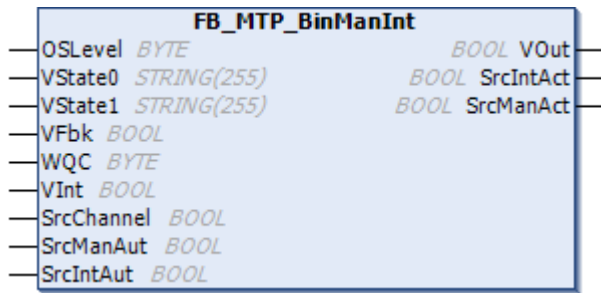
#### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
VOut	REAL	Ausgegebener Wert der aktuellen Wertvorgabe	Lesen

#### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
VMan	BOOL	Manual Wertvorgabe	Lesen

### 5.3.6.6 FB\_MTP\_BinManInt



Der Funktionsbaustein FB\_MTP\_BinManInt ist ein Objekt für eine binäre Wertvorgabe aus unterschiedlichen Quellen: interne SPS-Logik VInt oder manuelle Bedienung VMan (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten des [Source Mode \[▶ 15\]](#) verwaltet. Den Zuständen TRUE und FALSE der Wertvorgabe werden über die Eingangsvariablen VState0 und VState1 Zustandstexte zugewiesen.

#### Weitere Eigenschaften

[Name des Objekts \[▶ 18\]](#)

[Objektbeschreibung \[▶ 18\]](#)

[OSLevel \[▶ 18\]](#)

#### Vererbungshierarchie

```

FB_MTP_BinMan
  FB_MTP_BinManInt
  
```

#### Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
VUnit	INT	Einheit des Werts	Lesen
VMin	DINT	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	DINT	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	DINT	Rückmeldewert	Lesen
VScImin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScImax	DINT	Skalenende für Wertdarstellung	Lesen
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VInt	BOOL	Internal Wertvorgabe (Relevant, wenn SrcIntAct = True)	Lesen

#### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [▶ 15]</a>	
VOut	DINT	Ausgegebener Wert der aktuellen Wertvorgabe unter Berücksichtigung der Grenzwerte	Lesen

## Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*		Siehe <a href="#">Source Mode [► 15]</a>	
VMan	BOOL	Manual Wertvorgabe (Relevant, wenn SrcManAct = True)	Lesen

## Methoden

Name	Beschreibung
SetInternal	Siehe <a href="#">Source Mode [► 15]</a>
SetManual	

## 5.3.7 ParameterElements

`ParameterElements` sind Objekte für Wertvorgaben durch Parameter von unterschiedlichen Quellen: interne SPS-Logik und manuelle Bedienung (z.B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten [Service Mode \[► 17\]](#) verwaltet.

Die Parameter können in Konfigurationsparameter oder Prozedurparameter unterschieden werden. Konfigurationsparameter werden einem Dienst zugeordnet und können von allen Prozeduren verwendet werden. Prozedurparameter werden einer oder mehreren Prozeduren innerhalb eines Dienstes zugewiesen.

Parameterinstanzen die am Eingang `ConfParameters` eines Dienstes übergeben werden, werden als Konfigurationsparameter betrachtet.

Parameterinstanzen die am Eingang `ProcParameters` einer oder mehreren Prozedur/en des Dienstes übergeben werden, werden als Prozedurparameter betrachtet.

Die Summe aller Prozedurparameter der Prozeduren eines Dienstes müssen zusätzlich gesammelt am Eingang `ProcParameters` des übergeordneten Dienstes übergeben werden.



Für die Verwendung von Parametern wird die Verwendung des MTP-Engineerings und die darin enthaltene automatische Codegenerierung empfohlen!

### 5.3.7.1 FB\_MTP\_AnaServParam

FB_MTP_AnaServParam	
WQC	BYTE
OSLevel	BYTE
SrcChannel	BOOL
SrcExtAut	BOOL
SrcIntAut	BOOL
StateChannel	BOOL
StateOffAut	BOOL
StateOpAut	BOOL
StateAutAut	BOOL
ApplyEn	BOOL
ApplyInt	BOOL
Sync	BOOL
VScMin	REAL
VScMax	REAL
VUnit	INT
VMin	REAL
VMax	REAL
VFbk	REAL
VInt	REAL



Der Funktionsbaustein `FB_MTP_AnaServParam` ist ein Objekt für analoge Parameter-Wertvorgaben von unterschiedlichen Quellen: interne SPS-Logik oder manuelle Bedienung (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten `Service Mode` [► 17] verwaltet.

Der ausgegebene Wert `VOut` entspricht der auf Grenzwerte limitierten aktuellen Wertvorgabe und kann innerhalb eines Dienstes verwendet werden (siehe `ParameterElements` [► 88]).

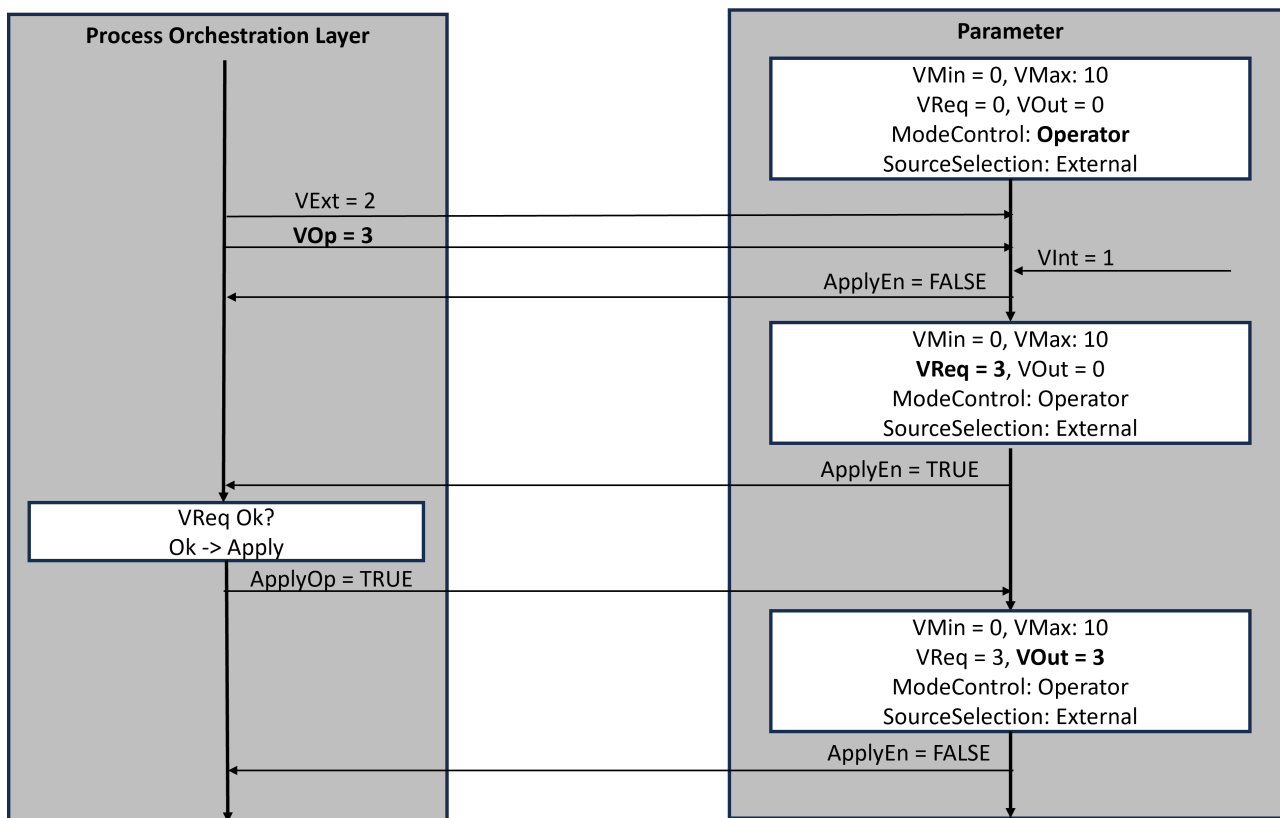
### Werteübernahme

Jeder Parameter folgt dem gleichen Ablauf bei der Werteübernahme. Die Werteübernahme erfolgt in zwei Stufen:

1. Vorbereitung des Werts
2. Übernahme des Werts

Die Vorbereitung des Werts beginnt mit der Wertvorgabe über die Variablen `VOp`, `VInt`, `VExt`. In Abhängigkeit des Zustands des `Service Mode` [► 17] wird ein Wert ausgewählt und über die Wertbegrenzung begrenzt. Über die Variable `VReq` wird der resultierende angeforderte Wert angezeigt.

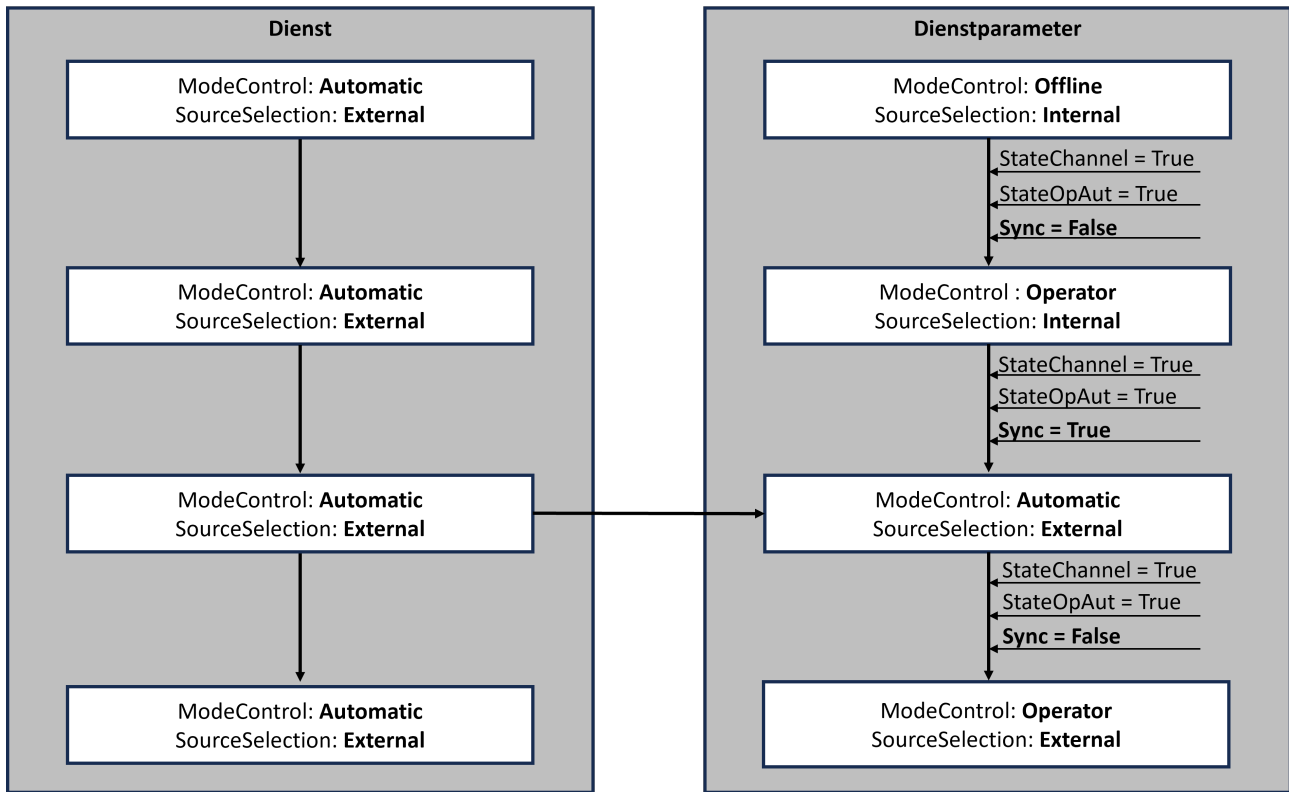
Für die Übernahme des Werts wird SPS-seitig die Freigabe über die Variable `ApplyEn = TRUE` signalisiert. Anschließend kann über Variablen `ApplyOp`, `ApplyInt` oder `ApplyExt`, je nach Zustand des `Service Mode` [► 17], die Übernahme bestätigt werden. Nach erfolgreicher Übernahme entspricht der Wert der Variable `VOut` dem Wert von `VReq`.



Die Werteübernahme kann auch auf Dienstebene nach dem gleichen Ablauf erfolgen. Siehe hierzu den Abschnitt `Parameter` in `FB_MTP_ServiceControl` [► 107].

### Betriebsarten-Synchronisierung mit übergeordnetem Dienst

Der `Service Mode` des Parameters kann über die Variable `Sync` mit dem übergeordneten Dienst synchronisiert werden. Dabei werden die Eingangsvariablen zum Ansteuern des Zustandsautomaten ignoriert und der Zustandsautomat des Parameters nimmt automatisch den Zustand des `Service Mode` [► 17] des Dienstes an.



**Weitere Eigenschaften**

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[OSLevel \[► 18\]](#)

[Service Mode \[► 17\]](#)

[Wertskalierung \[► 19\]](#)

[Einheit \[► 19\]](#)

[Wertbegrenzung \[► 20\]](#)

[Rücklesen \[► 20\]](#)

**🔌 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Service Mode [► 17]</a>	
Src*			
ApplyEn	BOOL	Freigabe Parameterübernahme: 1: Parameter können übernommen werden. 0: Parameter können nicht übernommen werden.	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
ApplyInt	BOOL	Internal Schaltanfrage, um angeforderte Parameter zu übernehmen (relevant, wenn <code>StateAutAct = TRUE</code> und <code>SrcIntAct = TRUE</code> ): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen
Sync	BOOL	Betriebsarten-Synchronisierung: 1: Mit Dienst synchronisieren. 0: Eigenständiger Betrieb	Lesen/schreiben
VInt	REAL	Internal Wertvorgabe (relevant, wenn <code>StateAutAct = TRUE</code> und <code>SrcIntAct = TRUE</code> )	Lesen
VUnit	INT	Einheit des Werts	Lesen
VMin	REAL	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	REAL	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	REAL	Rückmeldewert	Lesen
VScIMin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode</a> [▶ 17]	
Src*			
VOut	REAL	Aktueller Parameterwert	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode</a> [▶ 17]	
Src*			
ApplyOp	BOOL	Operator Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn <code>StateOpAct = TRUE</code> ): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
ApplyExt	BOOL	External Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn <code>SrcExtAct = TRUE</code> und <code>StateAutAct = TRUE</code> ): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
VOp	REAL	Operator Wertvorgabe (relevant, wenn <code>StateOpAct = TRUE</code> )	Lesen/schreiben
VExt	REAL	External Wertvorgabe (relevant, wenn <code>SrcExtAct = TRUE</code> und <code>StateAutAct = TRUE</code> )	Lesen/schreiben

### 5.3.7.2 FB\_MTP\_DIntServParam

FB_MTP_DIntServParam		
WQC	BYTE	BOOL SrcIntAct
OSLevel	BYTE	BOOL SrcExtAct
SrcChannel	BOOL	BOOL StateOpAct
SrcExtAut	BOOL	BOOL StateAutAct
SrcIntAut	BOOL	BOOL StateOffAct
StateChannel	BOOL	DINT VOut
StateOffAut	BOOL	
StateOpAut	BOOL	
StateAutAut	BOOL	
ApplyEn	BOOL	
ApplyInt	BOOL	
Sync	BOOL	
VScMin	DINT	
VScMax	DINT	
VUnit	INT	
VMin	DINT	
VMax	DINT	
VFbk	DINT	
VInt	DINT	

Der Funktionsbaustein `FB_MTP_DIntServParam` ist ein Objekt für ganzzahlige Parameter-Wertvorgaben von unterschiedlichen Quellen: interne SPS-Logik und manuelle Bedienung (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten [Service Mode \[► 17\]](#) verwaltet.

Der ausgegebene Wert `VOut` entspricht der auf Grenzwerte limitierten aktuellen Wertvorgabe und kann innerhalb eines Dienstes verwendet werden (siehe [ParameterElements \[► 88\]](#)).

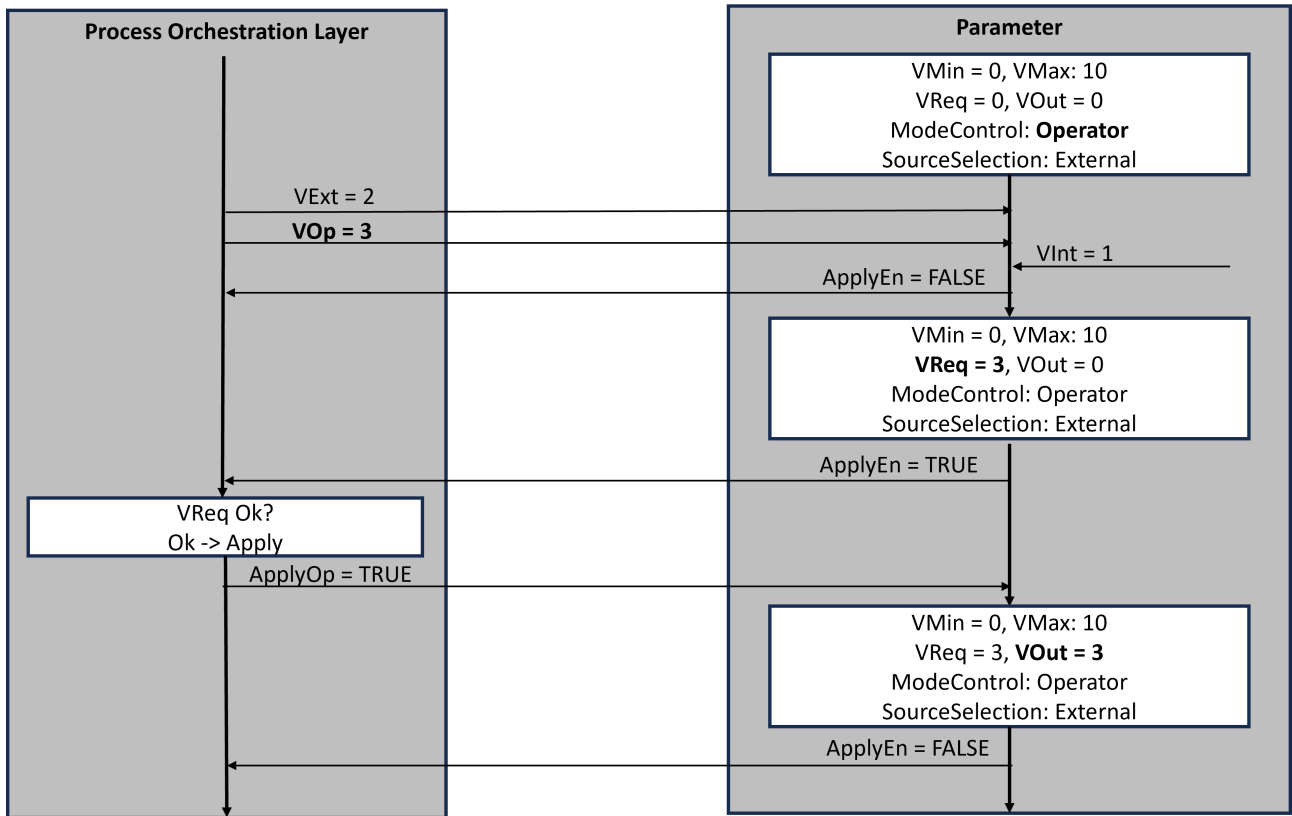
#### Werteübernahme

Jeder Parameter folgt dem gleichen Ablauf bei der Werteübernahme. Die Werteübernahme erfolgt in zwei Stufen:

1. Vorbereitung des Werts
2. Übernahme des Werts

Die Vorbereitung des Werts beginnt mit der Wertvorgabe über die Variablen `VOp`, `VInt`, `VExt`. In Abhängigkeit des Zustands des [Service Mode \[► 17\]](#) wird ein Wert ausgewählt und über die Wertbegrenzung begrenzt. Über die Variable `VReq` wird der resultierende angeforderte Wert angezeigt.

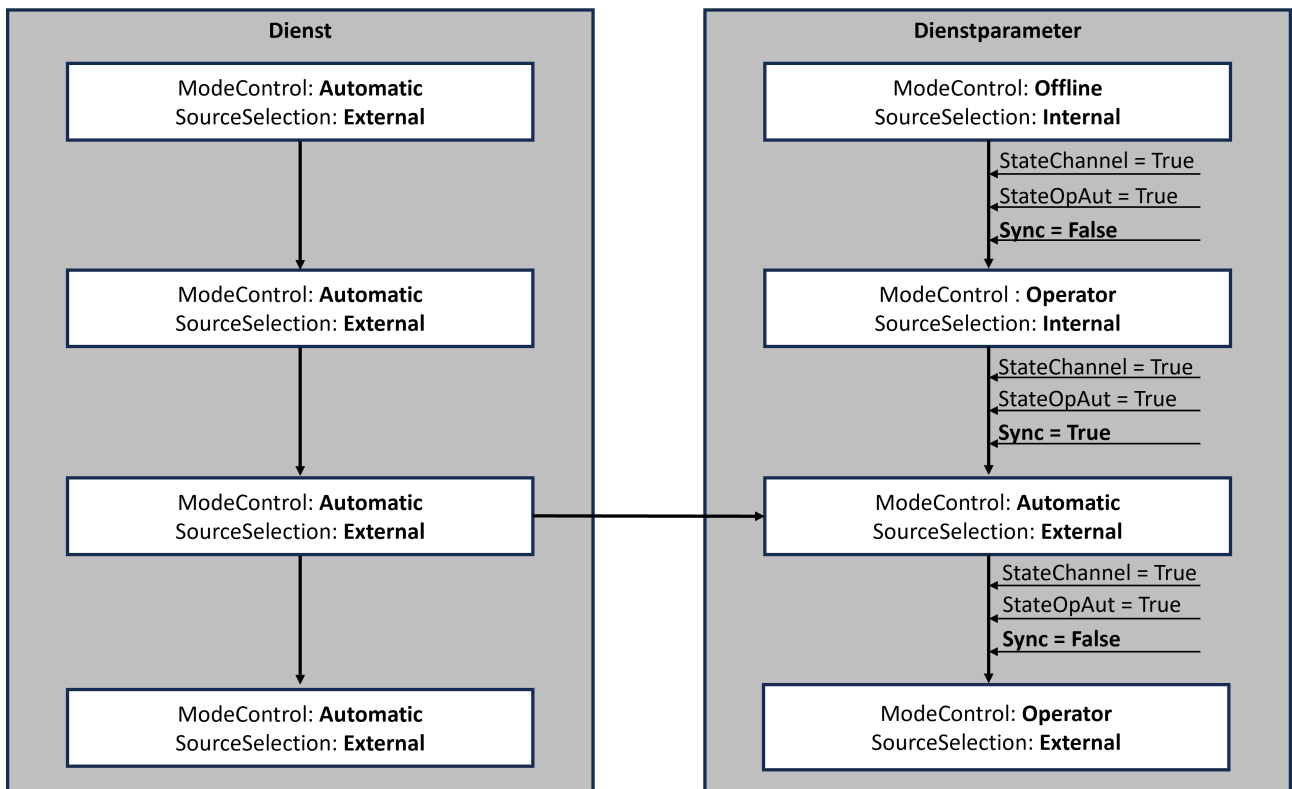
Für die Übernahme des Werts wird SPS-seitig die Freigabe über die Variable `ApplyEn = TRUE` signalisiert. Anschließend kann über Variablen `ApplyOp`, `ApplyInt` oder `ApplyExt`, je nach Zustand des [Service Mode \[► 17\]](#), die Übernahme bestätigt werden. Nach erfolgreicher Übernahme entspricht der Wert der Variable `VOut` dem Wert von `VReq`.



Die Werteübernahme kann auch auf Dienstebene nach dem gleichen Ablauf erfolgen. Siehe hierzu den Abschnitt Parameter in [FB\\_MTP\\_ServiceControl](#) [▶ 107].

**Betriebsarten-Synchronisierung mit übergeordnetem Dienst**

Der *Service Mode* des Parameters kann über die Variable *Sync* mit dem übergeordneten Dienst synchronisiert werden. Dabei werden die Eingangsvariablen zum Ansteuern des Zustandsautomaten ignoriert und der Zustandsautomat des Parameters nimmt automatisch den Zustand des *Service Mode* [▶ 17] des Dienstes an.



**Weitere Eigenschaften**[Name des Objekts \[► 18\]](#)[Objektbeschreibung \[► 18\]](#)[WQC \[► 19\]](#)[OSLevel \[► 18\]](#)[Service Mode \[► 17\]](#)[Wertskalierung \[► 19\]](#)[Einheit \[► 19\]](#)[Wertbegrenzung \[► 20\]](#)[Rücklesen \[► 20\]](#)**🔌 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State* Src*		Siehe <a href="#">Service Mode [► 17]</a>	
ApplyEn	BOOL	Freigabe Parameterübernahme: 1: Parameter können übernommen werden. 0: Parameter können nicht übernommen werden.	Lesen
ApplyInt	BOOL	Internal Schaltanfrage, um angeforderte Parameter zu übernehmen (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen
Sync	BOOL	Betriebsarten-Synchronisierung: 1: Mit Dienst synchronisieren. 0: Eigenständiger Betrieb	Lesen/schreiben
VInt	DINT	Internal Wertvorgabe (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE)	Lesen
VUnit	INT	Einheit des Werts	Lesen
VMin	DINT	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	DINT	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	DINT	Rückmeldewert	Lesen
VScIMin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScIMax	DINT	Skalenende für Wertdarstellung	Lesen

**🔌 Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode [► 17]</a>	

Name	Typ	Beschreibung	OPC-UA-Zugriff
Src*			
Vout	DINT	Aktueller Parameterwert	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode [► 17]</a>	
Src*			
ApplyOp	BOOL	Operator Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn StateOpAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
ApplyExt	BOOL	External Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
VOp	DINT	Operator Wertvorgabe (relevant, wenn StateOpAct = TRUE)	Lesen/schreiben
VExt	DINT	External Wertvorgabe (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE)	Lesen/schreiben

**5.3.7.3 FB\_MTP\_BinServParam**

FB_MTP_BinServParam	
WQC	BYTE BOOL SrcIntAct
OSLevel	BYTE BOOL SrcExtAct
SrcChannel	BOOL BOOL StateOpAct
SrcExtAut	BOOL BOOL StateAutAct
SrcIntAut	BOOL BOOL StateOffAct
StateChannel	BOOL BOOL VOut
StateOffAut	BOOL
StateOpAut	BOOL
StateAutAut	BOOL
ApplyEn	BOOL
ApplyInt	BOOL
Sync	BOOL
VState0	STRING(255)
VState1	STRING(255)
VFbk	BOOL
VInt	BOOL

Der Funktionsbaustein FB\_MTP\_BinServParam ist ein Objekt für binäre Parameter-Wertvorgaben von unterschiedlichen Quellen: interne PLC-Logik und manuelle Bedienung (z. B. über OPC-UA). Die Wertvorgaben werden über den Zustandsautomaten [Service Mode \[► 17\]](#) verwaltet.

Der ausgegebene Wert VOut entspricht der auf Grenzwerte limitierten aktuellen Wertvorgabe und kann innerhalb eines Dienstes verwendet werden (siehe [ParameterElements \[► 88\]](#)).

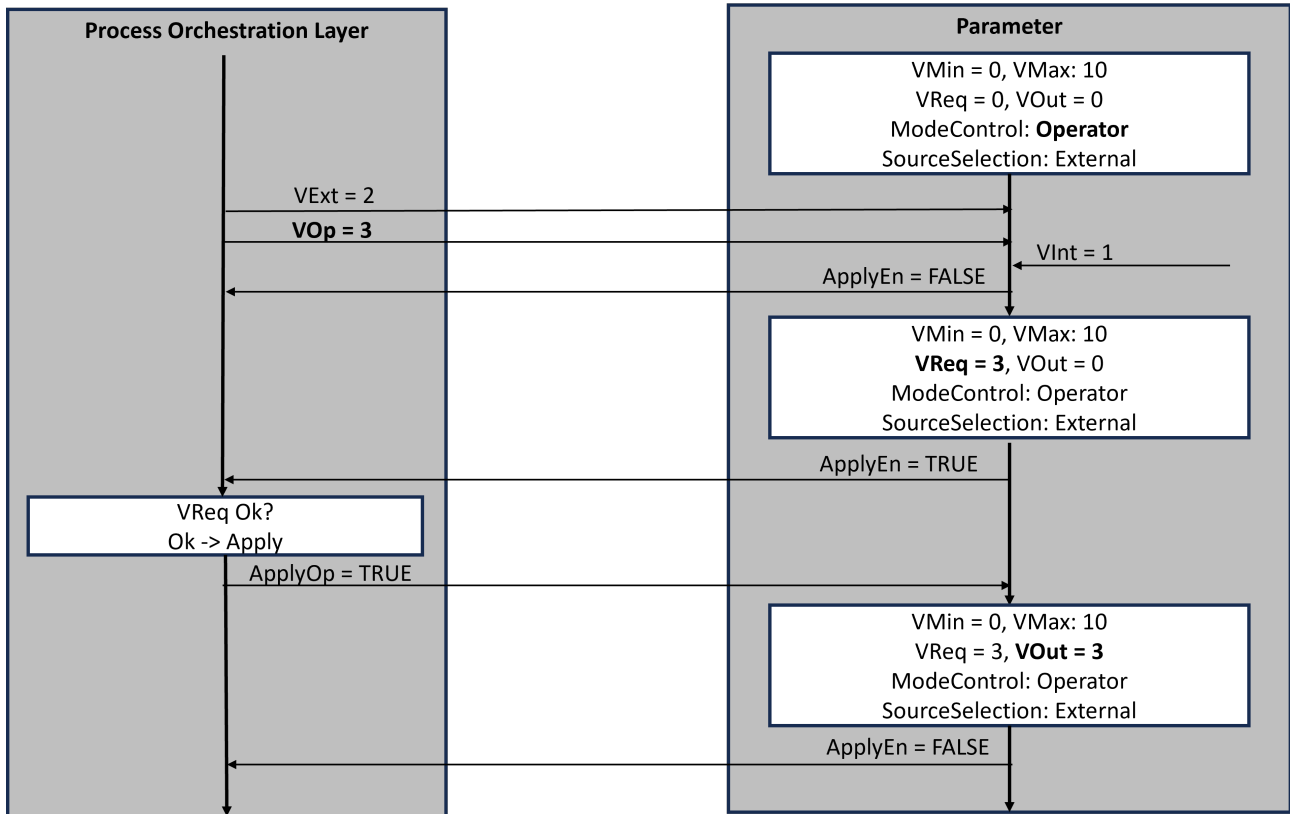
**Werteübernahme**

Jeder Parameter folgt dem gleichen Ablauf bei der Werteübernahme. Die Werteübernahme erfolgt in zwei Stufen:

1. Vorbereitung des Werts
2. Übernahme des Werts

Die Vorbereitung des Werts beginnt mit der Wertvorgabe über die Variablen  $v_{Op}$ ,  $v_{Int}$ ,  $v_{Ext}$ . In Abhängigkeit des Zustands des Service Mode [► 17] wird ein Wert ausgewählt und über die Wertbegrenzung begrenzt. Über die Variable  $v_{Req}$  wird der resultierende angeforderte Wert angezeigt.

Für die Übernahme des Werts wird SPS-seitig die Freigabe über die Variable  $ApplyEn = TRUE$  signalisiert. Anschließend kann über Variablen  $ApplyOp$ ,  $ApplyInt$  oder  $ApplyExt$ , je nach Zustand des Service Mode [► 17], die Übernahme bestätigt werden. Nach erfolgreicher Übernahme entspricht der Wert der Variable  $v_{Out}$  dem Wert von  $v_{Req}$ .

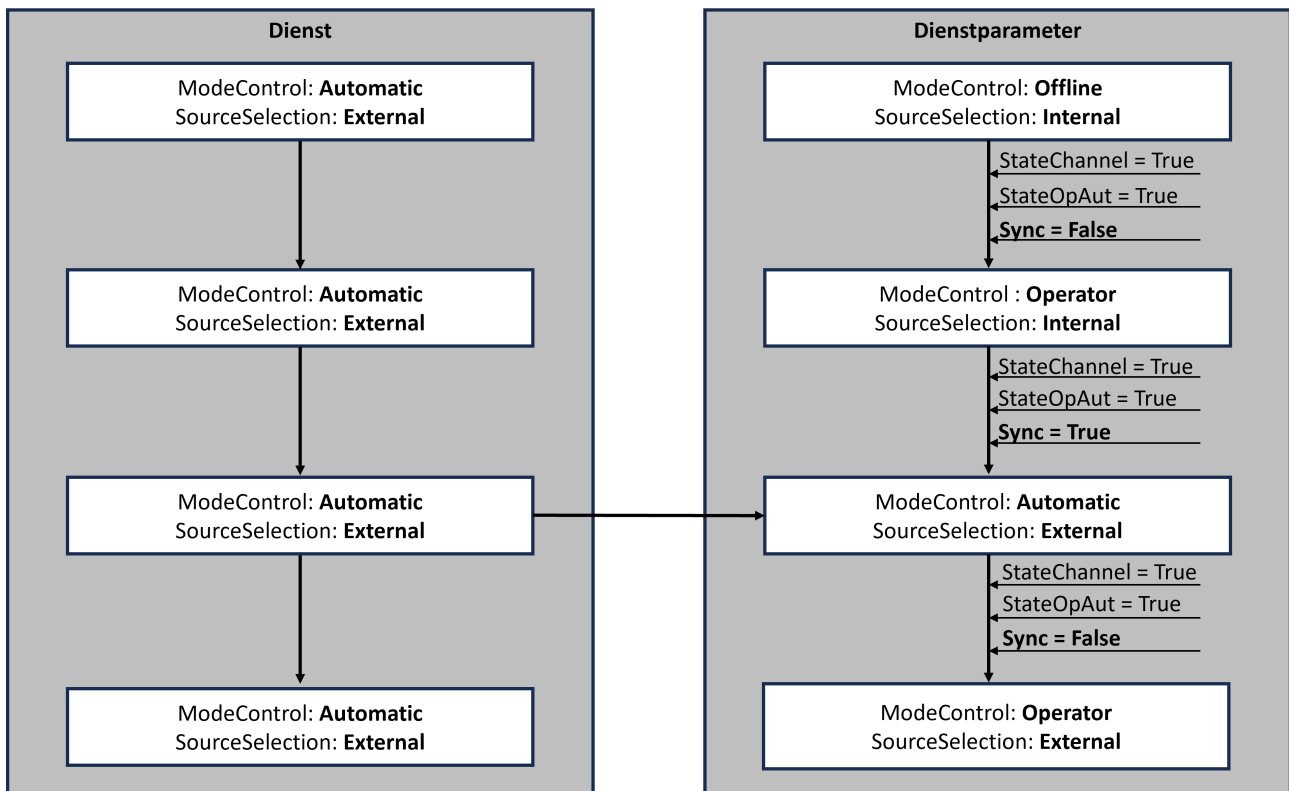


Die Werteübernahme kann auch auf Dienstebene nach dem gleichen Ablauf erfolgen. Siehe hierzu den Abschnitt Parameter in FB MTP ServiceControl [► 107].

### Betriebsarten-Synchronisierung mit übergeordnetem Dienst

Der Service Mode des Parameters kann über die Variable  $Sync$  mit dem übergeordneten Dienst synchronisiert werden. Dabei werden die Eingangsvariablen zum Ansteuern des Zustandsautomaten ignoriert und der Zustandsautomat des Parameters nimmt automatisch den Zustand des Service Mode [► 17] des Dienstes an.





**Weitere Eigenschaften**

Name des Objekts [[▶ 18](#)]

Objektbeschreibung [[▶ 18](#)]

WQC [[▶ 19](#)]

OSLevel [[▶ 18](#)]

Service Mode [[▶ 17](#)]

Rücklesen [[▶ 20](#)]

**📁 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
State*		Siehe <a href="#">Service Mode [▶ 17]</a>	
Src*			
ApplyEn	BOOL	Freigabe Parameterübernahme: 1: Parameter können übernommen werden. 0: Parameter können nicht übernommen werden.	Lesen
ApplyInt	BOOL	Internal Schaltanfrage, um angeforderte Parameter zu übernehmen (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen

Name	Typ	Beschreibung	OPC-UA-Zugriff
Sync	BOOL	Betriebsarten-Synchronisierung: 1: Mit Dienst synchronisieren. 0: Eigenständiger Betrieb	Lesen/schreiben
VInt	BOOL	Internal Wertvorgabe (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE)	Lesen
VState0	STRING	Text für Wert VOut = FALSE	Lesen
VState1	STRING	Text für Wert VOut = TRUE	Lesen
VFbk	BOOL	Rückmeldewert	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe	
Src*			
Vout	BOOL	Aktueller Parameterwert	Lesen

### Externe Variablen

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode [▶ 17]</a>	
Src*			
ApplyOp	BOOL	Operator Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn StateOpAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
ApplyExt	BOOL	External Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
VOp	REAL	Operator Wertvorgabe (relevant, wenn StateOpAct = TRUE)	Lesen/schreiben
VExt	REAL	External Wertvorgabe (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE)	Lesen/schreiben

#### 5.3.7.4 FB\_MTP\_StringServParam

FB_MTP_StringServParam	
WQC	BYTE
OSLevel	BYTE
SrcChannel	BOOL
SrcExtAut	BOOL
SrcIntAut	BOOL
StateChannel	BOOL
StateOffAut	BOOL
StateOpAut	BOOL
StateAutAut	BOOL
ApplyEn	BOOL
ApplyInt	BOOL
Sync	BOOL
VFbk	STRING(255)
VInt	STRING(255)

Der Funktionsbaustein `FB_MTP_StringServParam` ist ein Objekt für Zeichenketten Parameter-Wertvorgaben von unterschiedlichen Quellen: interne PLC-Logik und manuelle Bedienung (z. B. über OPC-UA). Die Wertvorgaben werden über die Zustandsautomaten Service Mode [▶ 17] verwaltet.

Der ausgegebene Wert `VOut` entspricht der auf Grenzwerte limitierten aktuellen Wertvorgabe und kann innerhalb eines Dienstes verwendet werden (siehe ParameterElements [▶ 88]).

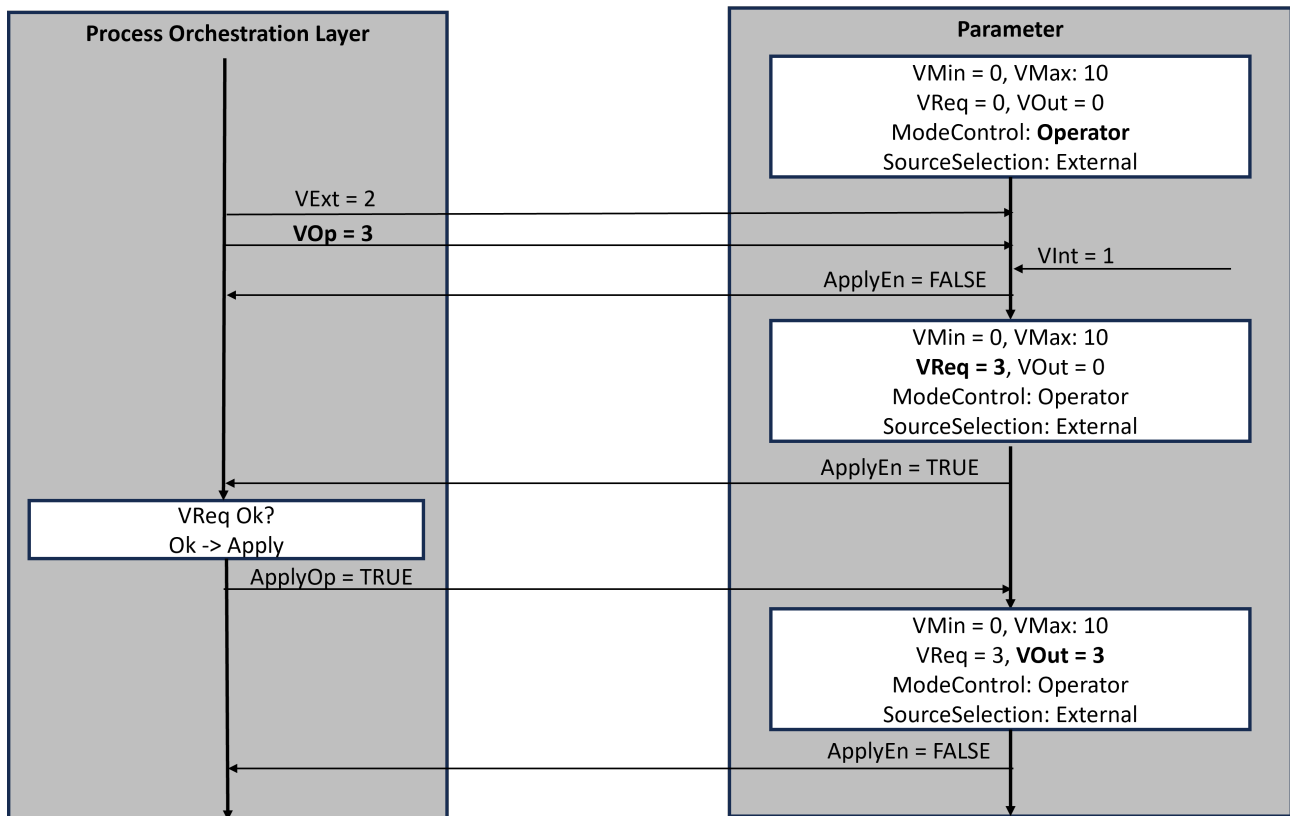
### Werteübernahme

Jeder Parameter folgt dem gleichen Ablauf bei der Werteübernahme. Die Werteübernahme erfolgt in zwei Stufen:

1. Vorbereitung des Werts
2. Übernahme des Werts

Die Vorbereitung des Werts beginnt mit der Wertvorgabe über die Variablen `VOp`, `VInt`, `VExt`. In Abhängigkeit des Zustands des Service Mode [▶ 17] wird ein Wert ausgewählt und über die Wertbegrenzung begrenzt. Über die Variable `VReq` wird der resultierende angeforderte Wert angezeigt.

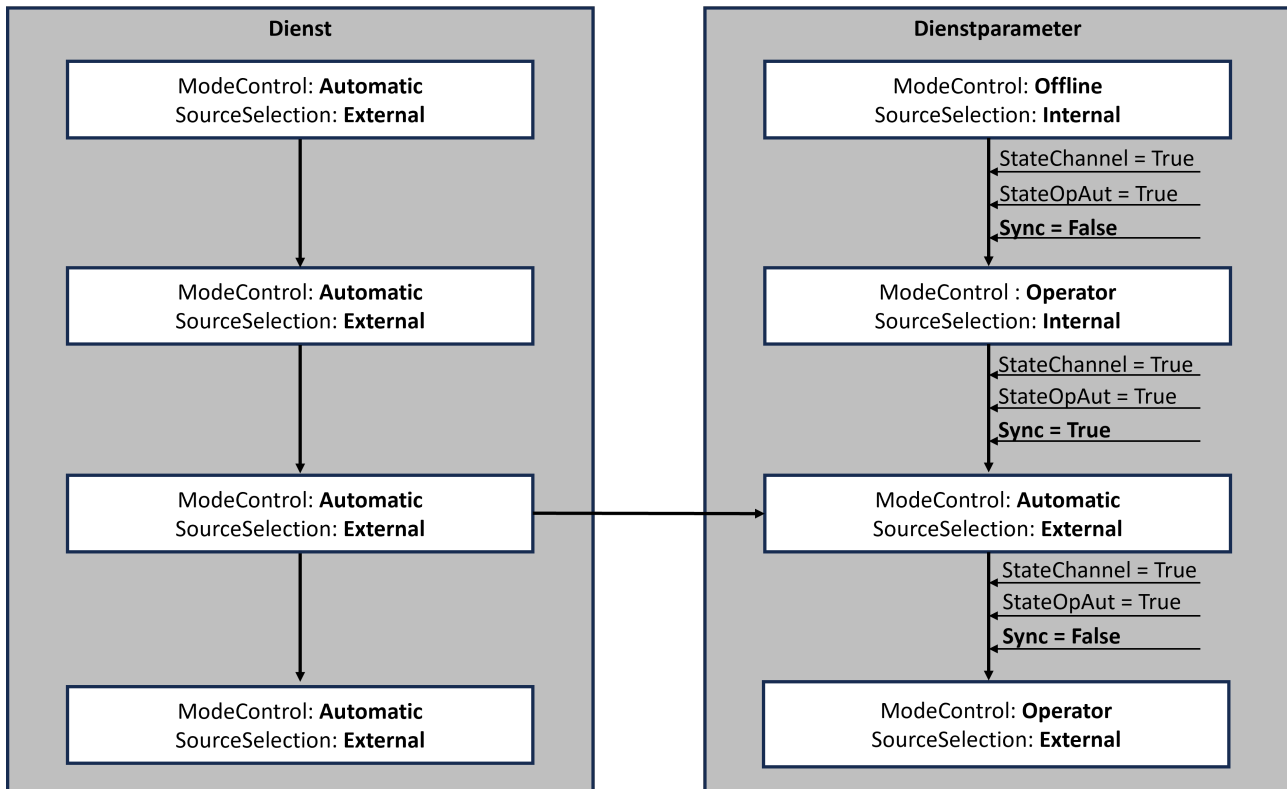
Für die Übernahme des Werts wird SPS-seitig die Freigabe über die Variable `ApplyEn = TRUE` signalisiert. Anschließend kann über Variablen `ApplyOp`, `ApplyInt` oder `ApplyExt`, je nach Zustand des Service Mode [▶ 17], die Übernahme bestätigt werden. Nach erfolgreicher Übernahme entspricht der Wert der Variable `VOut` dem Wert von `VReq`.



Die Werteübernahme kann auch auf Dienstebene nach dem gleichen Ablauf erfolgen. Siehe hierzu den Abschnitt Parameter in FB\_MTP\_ServiceControl [▶ 107].

### Betriebsarten-Synchronisierung mit übergeordnetem Dienst

Der `Service Mode` des Parameters kann über die Variable `Sync` mit dem übergeordneten Dienst synchronisiert werden. Dabei werden die Eingangsvariablen zum Ansteuern des Zustandsautomaten ignoriert und der Zustandsautomat des Parameters nimmt automatisch den Zustand des Service Mode [▶ 17] des Dienstes an.



### Weitere Eigenschaften

Name des Objekts [[► 18](#)]

Objektbeschreibung [[► 18](#)]

WQC [[► 19](#)]

OSLevel [[► 18](#)]

Service Mode [[► 17](#)]

Rücklesen [[► 20](#)]

### 📁 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
VInt	STRING	Internal Wertvorgabe (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE)	Lesen
VUnit	INT	Einheit des Werts	Lesen
VMin	REAL	Oberer Grenzwert der Wertvorgabe	Lesen
VMax	REAL	Unterer Grenzwert der Wertvorgabe	Lesen
VFbk	REAL	Rückmeldewert	Lesen
VScImin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen

**Ausgänge**

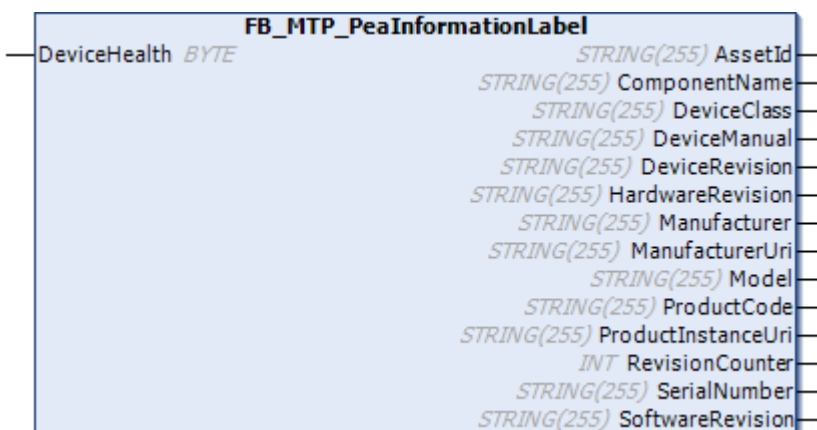
Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode</a> [▶ 17]	
Src*			
Vout	STRING	Aktueller Parameterwert	Lesen

**Externe Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State*		Siehe <a href="#">Service Mode</a> [▶ 17]	
Src*			
ApplyOp	BOOL	Operator Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn StateOpAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
ApplyExt	BOOL	External Schaltanfrage, um aktuell angeforderte Parameter zu übernehmen (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE): 1: Wertübernahme durchführen. 0: Keine Wertübernahme	Lesen/schreiben
VOp	STRING	Operator Wertvorgabe (relevant, wenn StateOpAct = TRUE)	Lesen/schreiben
VExt	STRING	External Wertvorgabe (relevant, wenn SrcExtAct = TRUE und StateAutAct = TRUE)	Lesen/schreiben

### 5.3.8 PeaElements

#### 5.3.8.1 FB\_MTP\_PeaInformationLabel



Der Funktionsbaustein FB\_MTP\_PeaInformationLabel ist ein Objekt, welches Informationen über eine PEA während der Laufzeit bereitstellt (z. B. über OPC-UA). PEA steht für Process Equipment Assembly und stellt ein prozesstechnisches Modul dar. Pro Modul wird eine Instanz dieses Objekts angelegt. Die Informationen am Ausgang dieses FBs werden bei der Instanziierung des FBs initialisiert. Diese Werte können während der Laufzeit von einer manuellen Bedienung (z. B. über OPC-UA) gelesen und geändert werden (siehe Zugriffsrechte in den Variablentabellen).



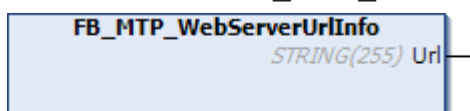
Die Variablen AssetId, ComponentName und DeviceClass sind als persistent deklariert.

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
DeviceHealth	BYTE	Zusammengefasster WQC auf Modulebene (siehe auch <a href="#">WQC (► 19)</a> )	Lesen

### Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
AssetId	STRING	Eindeutige ID innerhalb eines Systems. Diese kann vom Anlagenbetreiber vorgegeben werden.	Lesen/schreiben
ComponentName	STRING	Modulname innerhalb eines Systems. Dieser kann vom Anlagenbetreiber vorgegeben werden.	Lesen/schreiben
DeviceClass	STRING	Die Geräteklasse kann vom PEA-Hersteller vordefiniert werden. Dieser Wert kann vom Anlagenbetreiber überschrieben werden.	Lesen/schreiben
DeviceManual	STRING	Verweis auf modulbezogene Dokumente (z. B. Webseiten)	Lesen
DeviceRevision	STRING	Versionsinformation der informationstechnischen Schnittstelle. Diese wird als Major-Minor-Patch-Notation ausgeführt (z. B. 1.2.3).	Lesen
HardwareRevision	STRING	Versionsinformation des konstruktiven Aufbaus einer PEA. Diese wird als Major-Minor-Patch-Notation ausgeführt (z. B. 1.2.3).	Lesen
Manufacturer	STRING	Name des Herstellers des Moduls	Lesen
ManufacturerUri	STRING	Adresse der Webseite des Herstellers (z. B. <a href="http://www.Beckhoff.de">www.Beckhoff.de</a> )	Lesen
Model	STRING	Name der PEA. Dieser Wert ist identisch mit dem Namen des MTP.	Lesen
ProductCode	STRING	Eindeutige Produktbezeichnung der PEA	Lesen
ProductInstanceUri	STRING	Eindeutige Kennung (URI) der PEA (gemäß DIN SPEC 91406)	Lesen
RevisionCounter	STRING	Revisionszähler	Lesen
SerialNumber	STRING	Seriennummer der PEA (Zugewiesen durch den PEA-Hersteller.)	Lesen
SoftwareRevision	STRING	Versionsinformation der SPS-Software. Diese wird als Major-Minor-Patch-Notation ausgeführt (z. B. 1.2.3).	Lesen

### 5.3.8.2 FB\_MTP\_WebServerUrlInfo



Der Funktionsbaustein `FB_MTP_WebServerUrlInfo` ist ein Objekt zur Bereitstellung eines Endpunkts eines vorhandenen Webservers.

Wenn ein Webserver angegeben werden soll, kann über die Variable `Url` der Endpunkt des Webservers bei der Initialisierung des Objekts instanziiert werden.

 **Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
Url	STRING	Endpunkt des Webservers	Lesen

### 5.3.9 ServiceElements

Anlagenfunktionen können in gekapselten Diensten [FB\\_MTP\\_ServiceControl \[▶ 107\]](#) ausgeführt werden. Dabei bündelt der Dienst verschiedene Prozeduren, Parameter und Berichtswerte. Die Ausführung erfolgt in Prozeduren. Prozeduren sind unterschiedliche Ausführungsvarianten eines Dienstes.

Ein Beispiel wäre das Befüllen eines Tanks: Der Dienst könnte `Filling` genannt werden. Die Prozeduren dieses Dienstes könnten `Volume_Filling` und `Duration_Filling` heißen, denen unterschiedliche Parameter zugeordnet sind, um ein bestimmtes Volumen oder für eine bestimmte Zeit zu Befüllen.

Dienste beschreiben die Funktion und führen diese in Prozeduren aus, daher muss ein Dienst immer mindestens eine Prozedur [FB\\_MTP\\_Procedure \[▶ 103\]](#) besitzen.

Für die Ausführung können folgende Parameter den Prozeduren übergeben werden:

#### Konfigurationsparameter

Konfigurationsparameter werden allen Prozeduren eines Dienstes zugeordnet. Sie werden typischerweise bei der Inbetriebnahme gesetzt.

#### Prozedurparameter

Prozedurparameter können im Gegensatz zu Konfigurationsparametern einer oder mehreren Prozeduren eines Dienstes zugeordnet werden. Sie werden typischerweise vorm Starten eines Dienstes gesetzt.

#### Reportwerte

`ReportValues` werden einer oder mehreren Prozeduren eines Dienstes zugeordnet. Sie können z.B. für das Dokumentieren wichtiger Prozesswerte verwendet werden.

#### 5.3.9.1 FB\_MTP\_Procedure



Der Funktionsbaustein `FB_MTP_Procedure` enthält alle Attribute und Methoden zum Beschreiben einer Prozedur. Eine Prozedur ist eine Ausführungsvariante eines Dienstes und kann daher nur einem Dienst zugewiesen werden. Alle Prozeduren erhalten eine im Dienst einmalige `ProcedureID`. Diese müssen Sie durch eine natürliche Zahl beschreiben, wobei die Zahl 0 nicht verwendet werden darf! Über die Variable `IsSelfCompleting` wird angegeben, ob die prozesstechnische(n) Funktion(en) der Prozedur sich selbst beendet/beenden (z. B. nach Ablauf einer Zeit).

Die Ausgänge `IsSelfCompleting` und `ProcedureID` werden bei der Instanziierung dieses Funktionsbausteins initialisiert.

`CommandInfo` ist nach `E_MTP_CommandEn` codiert und wird für jede Prozedur, die sich im Zustand `Automatic` oder `Operator` befindet, separat und unabhängig vom Zustand des Dienstes aktualisiert. Sie gibt an, welche Steuerbefehle aktuell freigegeben sind.

Die Variable `CommandAdapt` ist nach `E_MTP_CommandEn` codiert. Mit ihr kann die Freigabe von Steuerbefehlen in `CommandInfo` zusätzlich eingeschränkt werden.

**i** Die Variable `CommandAdapt` ist kein Bestandteil der MTP-Richtlinie. Sie wurde zusätzlich implementiert. Diese Variable wird nicht in der MTP-Datei aufgeführt und nicht über OPC-UA zu Verfügung gestellt.

**i** Bei dem `FB_MTP_Procedure` handelt es sich um einen abstrakten Funktionsbaustein. Dieser kann nicht instanziiert, sondern muss abgeleitet werden (siehe Abschnitt Syntax)!

**i** Für die Verwendung von Prozeduren wird die Verwendung des MTP-Engineering und die darin enthaltene automatische Codegenerierung empfohlen!

### Benötigte Parameter und Betriebsmittel

Der Funktionsbaustein bekommt die zur Ausführung benötigten Prozessparameter `ProcParameters` als `Array[*] OF POINTER TO FB_MTP_ParameterElement` übergeben. Die benötigten Betriebsmittel `RequiredEquipment` werden als `Array[*] OF POINTER TO I_MTP_ReqEq` übergeben.

**i** Sollten keine `ProcParameter` bzw. kein `RequiredEquipment` benötigt werden, müssen Sie ein leeres Array übergeben!

### Dienste-Zustandsautomat

Die Zustände des Dienstzustandsautomaten werden in den gleichnamigen Methoden dieses Funktionsblocks ausgeführt. Der Aufruf der Methoden erfolgt durch den Dienstzustandsautomat.

### Aufruf der Prozedur

Die Prozeduren können vom übergeordneten Dienst aufgerufen werden. Im Funktionsblock `FB_MTP_ServiceControl` [► 107] ist ein möglicher Aufruf beschrieben.

### Weitere Eigenschaften

[WQC](#) [► 19]

### Syntax

```
FUNCTION_BLOCK FB_MTP_Procedure_1_Service EXTENDS FB_MTP_Procedure
```

### 📁 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
CommandInfo	DWORD	Aktuell freigegebene Steuerbefehle	Lesen



Name	Typ	Beschreibung	OPC-UA-Zugriff
ProcParameters	ARRAY [*] OF POINTER TO POINTER TO FB_MTP_ Parameter Element	Benötigte Prozedurparameter	-
RequiredEquipment	ARRAY [*] OF POINTER TO I_MTP_Re qEq	Benötigte Betriebsmittel	-
CommandAdapt	DWORD	Variable zum Sperren von Zustandsübergängen	-

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
IsSelfCompleting	BOOL	1: Prozedur ist selbstbeendend. 0: Prozedur wird über den Steuerbefehl beendet.	Lesen
ProcedureID	DWORD	Eindeutige Identifikation der Prozedur innerhalb des Dienstes	Lesen

 **Methoden**

Name	Typ	Beschreibung
Aborted	BOOL	Implementierung der Logik für den Zustand <code>Aborted</code> des Dienst-Zustandsautomaten.
Aborting	BOOL	Implementierung der Logik für den Zustand <code>Aborting</code> des Dienst-Zustandsautomaten.
Completed	BOOL	Implementierung der Logik für den Zustand <code>Completed</code> des Dienst-Zustandsautomaten.
Completing	BOOL	Implementierung der Logik für den Zustand <code>Completing</code> des Dienst-Zustandsautomaten.
Execute	BOOL	Implementierung der Logik für den Zustand <code>Execute</code> des Dienst-Zustandsautomaten.
Held	BOOL	Implementierung der Logik für den Zustand <code>Held</code> des Dienst-Zustandsautomaten.
Holding	BOOL	Implementierung der Logik für den Zustand <code>Holding</code> des Dienst-Zustandsautomaten.
Paused	BOOL	Implementierung der Logik für den Zustand <code>Paused</code> des Dienst-Zustandsautomaten.
Pausing	BOOL	Implementierung der Logik für den Zustand <code>Pausing</code> des Dienst-Zustandsautomaten
Resetting	BOOL	Implementierung der Logik für den Zustand <code>Resetting</code> des Dienst-Zustandsautomaten.
Resuming	BOOL	Implementierung der Logik für den Zustand <code>Resuming</code> des Dienst-Zustandsautomaten.
Starting	BOOL	Implementierung der Logik für den Zustand <code>Starting</code> des Dienst-Zustandsautomaten.
Stopping	BOOL	Implementierung der Logik für den Zustand <code>Stopping</code> des Dienst-Zustandsautomaten.

Name	Typ	Beschreibung
Stopped	BOOL	Implementierung der Logik für den Zustand <code>Stopped</code> des Dienst-Zustandsautomaten.
Unholding	BOOL	Implementierung der Logik für den Zustand <code>Unholding</code> des Dienst-Zustandsautomaten.
IsActuatorsInternal	BOOL	1: Alle der Prozedur zugeordneten Aktoren befinden sich im Zustand <code>Internal</code> .
IsActuatorsManual	BOOL	1: Alle der Prozedur zugeordneten Aktoren befinden sich im Zustand <code>Manual</code> .
IsActuatorsOffline	BOOL	1: Alle der Prozedur zugeordneten Aktoren befinden sich im Zustand <code>Offline</code> .
IsActuatorsOperator	BOOL	1: Alle der Prozedur zugeordneten Aktoren befinden sich im Zustand <code>Operator</code> .
IsActuatorsAutomatic	BOOL	1: Alle der Prozedur zugeordneten Aktoren befinden sich im Zustand <code>Automatic</code> .
IsReqEqInternal	BOOL	1: Alle der Prozedur zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Internal</code> .
IsReqEqManual	BOOL	1: Alle der Prozedur zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Manual</code> .
IsReqEqOffline	BOOL	1: Alle der Prozedur zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Offline</code> .
IsReqEqOperator	BOOL	1: Alle der Prozedur zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Operator</code> .
IsReqEqAutomatic	BOOL	1: Alle der Prozedur zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Automatic</code> .
SetActuatorsInternal	BOOL	Alle der Prozedur zugeordneten Aktoren werden in den Zustand <code>Internal</code> überführt.
SetActuatorsManual	BOOL	Alle der Prozedur zugeordneten Aktoren werden in den Zustand <code>Manual</code> überführt.
SetActuatorsOffline	BOOL	Alle der Prozedur zugeordneten Aktoren werden in den Zustand <code>Offline</code> überführt.
SetActuatorsOperator	BOOL	Alle der Prozedur zugeordneten Aktoren werden in den Zustand <code>Operator</code> überführt.
SetActuatorsAutomatic	BOOL	Alle der Prozedur zugeordneten Aktoren werden in den Zustand <code>Automatic</code> überführt.
SetReqEqInternal	BOOL	Alle der Prozedur zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Internal</code> überführt.
SetReqEqManual	BOOL	Alle der Prozedur zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Manual</code> überführt.
SetReqEqOffline	BOOL	Alle der Prozedur zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Offline</code> überführt.
SetReqEqOperator	BOOL	Alle der Prozedur zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Operator</code> überführt.
SetReqEqAutomatic	BOOL	Alle der Prozedur zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Automatic</code> überführt.

### 5.3.9.2 FB\_MTP\_ServiceControl

FB_MTP_ServiceControl	
WQC <i>BYTE</i>	<i>DWORD</i> StateCur
Procedures <i>POINTER TO POINTER TO FB_MTP_Procedure</i>	<i>DWORD</i> ProcedureCur
ConfParameters <i>POINTER TO POINTER TO FB_MTP_ParameterElement</i>	<i>DWORD</i> ProcedureReq
ProcParameters <i>POINTER TO POINTER TO FB_MTP_ParameterElement</i>	<i>BOOL</i> StateOpAct
ReportValues <i>POINTER TO POINTER TO FB_MTP_ReportValue</i>	<i>BOOL</i> StateAutAct
OSLevel <i>BYTE</i>	<i>BOOL</i> StateOffAct
CommandInt <i>DWORD</i>	<i>BOOL</i> SrcIntAct
ProcedureInt <i>DWORD</i>	<i>BOOL</i> SrcExtAct
CommandEn <i>DWORD</i>	<i>BOOL</i> ProcParamApplyEn
StateChannel <i>BOOL</i>	<i>BOOL</i> ConfigParamApplyEn
StateOffAut <i>BOOL</i>	
StateOpAut <i>BOOL</i>	
StateAutAut <i>BOOL</i>	
SrcChannel <i>BOOL</i>	
SrcExtAut <i>BOOL</i>	
SrcIntAut <i>BOOL</i>	
PosTextID <i>DWORD</i>	
InteractQuestionID <i>DWORD</i>	
InteractAddInfo <i>STRING(255)</i>	
ProcParamApplyInt <i>BOOL</i>	
ConfigParamApplyInt <i>BOOL</i>	
ReportValueFreeze <i>BOOL</i>	

Der Funktionsbaustein FB\_MTP\_ServiceControl ermöglicht das Ausführen von Prozeduren aus unterschiedlichen Quellen: interne SPS-Logik, manuelle Bedienung (z. B. über OPC-UA) oder Leitsystem (z. B. über OPC-UA). Schaltanfragen und Wertevorgaben werden über den Service Mode [► 17] verwaltet.



Bei dem FB\_MTP\_ServiceControl handelt es sich um einen abstrakten Funktionsbaustein. Dieser kann nicht instanziiert, sondern muss abgeleitet werden (siehe Abschnitt Syntax)!



Für die Verwendung von Diensten wird die Verwendung des MTP-Engineerings und die darin enthaltene automatische Codegenerierung empfohlen!

#### Prozeduren

Am Eingang `Procedures` werden alle dem Dienst zugehörigen Prozeduren als `ARRAY[*] OF POINTER TO FB_MTP_Procedure` übergeben. Ein Dienst benötigt mindestens eine Prozedur.

Die Auswahl einer Prozedur erfolgt über die Variablen `ProcedureInt`, `ProcedureOp`, `ProcedurExt` in Abhängigkeit des Zustands des Service Mode [► 17] des Dienstes. Über `ProcedureReq` wird die aktuell vorgewählte Prozedur-ID angezeigt, welche beim Start des Dienstes verwendet wird. Bei ungültiger Prozedur-ID wird hier eine 0 ausgegeben. `ProcedureCur` zeigt die aktuell gestartete Prozedur an. Es kann immer nur eine Prozedur zur gleichen Zeit ausgeführt werden.

#### Parameter

Am Eingang `ConfParameters` werden alle dem Dienst zugehörigen Konfigurationsparameter als `ARRAY[*] OF POINTER TO FB_MTP_ParameterElement` übergeben.

Am Eingang `ProcParameter` werden alle dem Dienst zugehörigen Prozedurparameter als `ARRAY[*] OF POINTER TO FB_MTP_ParameterElement` übergeben.

Die dienstweite Übernahme neuer Werte für die Parameter erfolgt nach dem gleichen Prinzip wie in den Unterkapiteln von ParameterElements [► 88] beschrieben. Es werden die Wertvorgaben aller dem Dienst zugewiesenen Parameter auf einmal übernommen. Über die Variable `vReq` der Parameter können die zu übernehmenden Werte vorher geprüft werden.

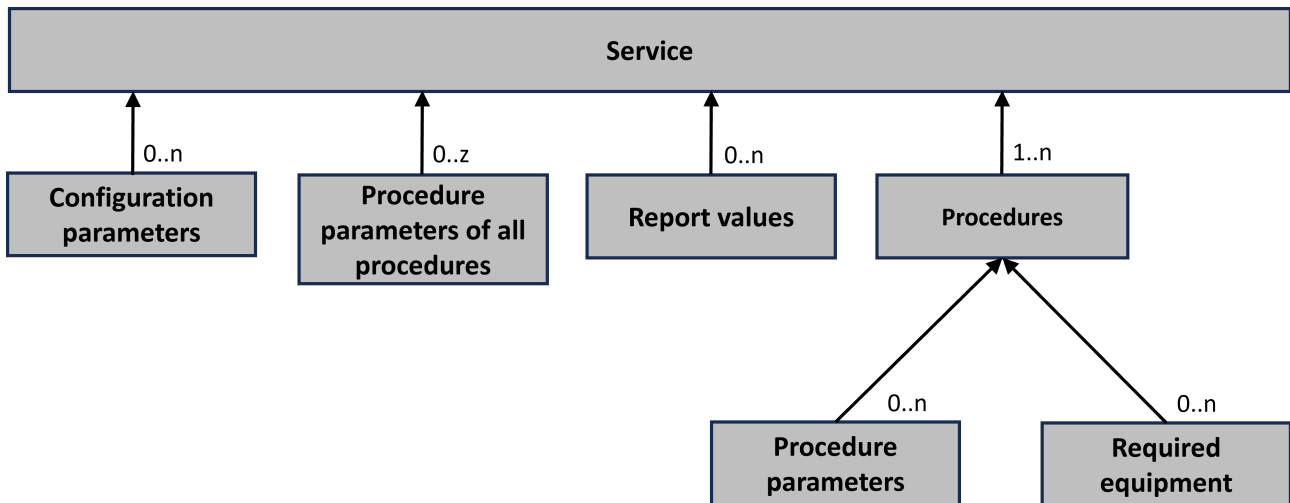
Die Freigabe für die Übernahme erfolgt über die Variable `ProcParameterApplyEn` für Prozedurparameter bzw. über die Variable `ConfigParameterApplyEn` für Konfigurationsparameter. Eine dienstweite Freigabe ist nur möglich, wenn alle Prozedurparameter bzw. Konfigurationsparameter die Freigabe `ApplyEn = TRUE` erteilt haben.

Je nach Zustand des `Service Mode` erfolgt die dienstweite Werteübernahme für Prozedurparameter über die Variablen `ProcParameterApplyInt`, `ProcParameterApplyOp` oder `ProcParameterApplyExt` bzw. über die Variablen `ConfigParameterApplyInt`, `ConfigParameterApplyOp` oder `ConfigParameterApplyExt` für die Konfigurationsparameter, wenn `Apply = TRUE` ist.

## Berichtswerte

Am Eingang `ReportValues` werden alle dem Dienst zugehörigen Berichtswerte als `ARRAY[*] OF POINTER TO FB_MTP_ReportValue` übergeben. Über den Eingang `ReportValueFreeze` können alle Reportwerte `V` des Dienstes eingefroren werden. Sollte sich der Wert des Eingangswerts `VIn` in dieser Zeit mehr als einmal ändern, wird dies am Ausgang `MissedValue = TRUE` des Parameters angezeigt.

Das untenstehende Bild zeigt eine Übersicht der zu übergebenen Arrays:

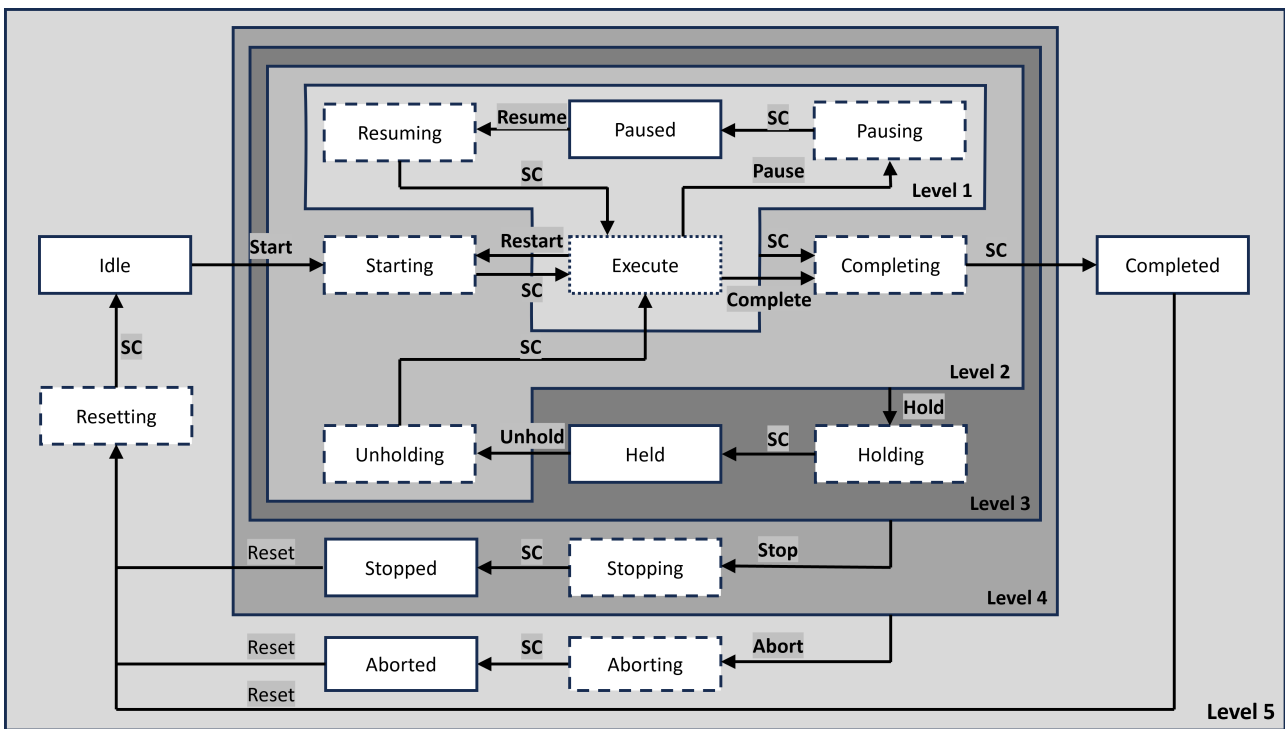


Sollten keine `ConfParameter`, `ProcParameter` oder kein `RequiredEquipment` benötigt werden, übergeben Sie ein leeres Array.

## Dienstzustandsautomat

Der Dienstzustandsautomat hat 16 Zustände auf fünf Ebenen (Level). Die Zustände sind unterteilt in transiente und nicht-transiente Zustände. Die transienten Zustände werden über Steuerbefehle des Steuerworts erreicht. Nicht transiente Zustände werden nach erfolgreicher Abarbeitung der transienten Zustände (SC) erreicht. Eine Ausnahme ist der Zustand `Execute`. Dieser kann sowohl transient, als auch nicht-transient sein. Dies wird über die Variable `IsSelfCompleting` in der [Prozedur \[103\]](#) definiert.

Der Dienstzustandsautomat ist nach dem untenstehenden Schema aufgebaut:



Der Dienstzustandsautomat kann in unterschiedlichen Schleifen betrachtet werden:

**Hauptschleife**

Die Hauptschleife zum Ausführen der prozesstechnischen Hauptfunktion(en) des Automaten beinhaltet die folgenden Zustände:

Idle→Starting→Execute→Completing→Completed→Resetting→Idle

Daneben können noch folgende Schleifen betrachtet werden:

**Pause-Schleife**

Die Pause-Schleife kann nur aus dem Zustand `Execute` gestartet werden. Beendet wird sie durch den Befehl `Resume` oder durch einen Befehl, der in einen Zustand höherer Ebene führt.

Beispiel:

Execute→Pausing→Paused→Resuming→Execute

**Hold-Schleife**

Die Hold-Schleife kann aus den Zuständen aus Ebene 2 oder geringer gestartet werden. Beendet wird sie durch den Befehl `Unhold` oder durch einen Befehl, der in einen Zustand höherer Ebene führt.

Beispiel:

Execute→Holding→Held→Unholding→Execute



In der MTP-Richtlinie dient die Hold-Schleife der Implementierung der in einem Fehlerfall auszuführenden Logik.

**Stop-Schleife**

Die Stop-Schleife kann aus den Zuständen aus Ebene 3 oder geringer gestartet werden. Beendet wird sie durch den Befehl `Reset` oder `Abort`.

Execute→Stopping→Stopped→Resetting→Idle



In der MTP-Richtlinie dient die Stop-Schleife der Implementierung der Logik zum Stoppen des Dienstes.

---

### Abort-Schleife

Die Abort-Schleife kann aus den Zuständen aus Ebene 4 oder geringer gestartet werden. Beendet wird sie durch den Befehl `Reset`.

`Execute`→`Aborting`→`Aborted`→`Resetting`→`Idle`

---



In der MTP-Richtlinie dient die Abort-Schleife der Implementierung der Logik zum schnellen Stoppen des Dienstes.

---

### Steuerung des Dienstzustandsautomaten

Die Variable `CommandEn` gibt die einzelnen Steuerbefehle des Steuerworts frei. Über die Variablen `CommandInt`, `CommandOp` und `CommandExt` kann das Steuerwort (siehe `E_MTP_Command`) in Abhängigkeit des Zustands des `Service Mode` [► 17] für den Dienstzustandsautomaten vorgegeben werden.

Alle Zustände können mit einem Steuerbefehl, sofern dieser freigegeben ist, in einen Zustand höherer Ebene überführt werden: z. B. können mit dem Befehl `Hold` alle Zustände der Ebenen 1 und 2 in den Zustand  `Holding` überführt werden.

### Ausführung von Prozeduren

Erfolgt der Befehl `Start`, wird die zu ausgewählte Prozedur in den Zustand `Starting` überführt. Alle anderen Prozeduren, die dem Dienst zugewiesen sind, werden gesperrt.

Die Ausführung der Zustände, mit Ausnahme von `Idle`, erfolgt in den gleichnamigen Methoden der ausgewählten Prozedur. Bei der Ausführung von transienten Zuständen erfolgt der Zustandsübergang (SC), wenn die jeweilige Methode `TRUE` zurückgibt. Nach erfolgreicher Ausführung gehen sie in ihren Folgezustand über.

Erst wenn der Dienstzustandsautomat sich wieder im Zustand `Idle` befindet, kann eine andere Prozedur ausgewählt und gestartet werden. Über die Variable `StateCur` wird der aktuelle Zustand des Dienstzustandsautomaten angezeigt (siehe `E_MTP_State`).

### Abhängigkeiten mit Service Mode

Wird der `Service Mode` aus dem Zustand `Offline` in einen der beiden anderen Zustände `Operator` oder `Automatic*` überführt, befindet sich der Dienstzustandsautomat immer im Zustand `Idle`. Ein Zurückführen des `Service Mode` in den Zustand `Offline` ist nur möglich, wenn der Dienstzustandsautomat sich im Grundzustand `Idle` befindet. Ein Umschalten zwischen `Operator` und `Automatic*` ist unabhängig vom Zustand des Dienstzustandsautomaten möglich und verändert nicht dessen aktuellen Zustand.

### Positionsbeschreibung

Um den aktuellen Stand des Dienstes näher zu beschreiben, kann eine Textliste angelegt werden. In dieser können die einzelnen Arbeitsschritte näher beschrieben werden. Über die Variable `PosTextID` kann auf die einzelnen Arbeitsschritte in der Textliste verwiesen werden.



Für die Positionsbeschreibung wird die Verwendung des MTP-Engineerings und des darin enthaltenen MTP-Exports empfohlen, da die Textliste nicht über OPC-UA zur Verfügung gestellt wird, sondern in der MTP-Datei beschrieben wird!

---

## Dienst-Bediener-Interaktion

Während der Laufzeit kann eine Interaktion zwischen Dienst und Bediener erfolgen. Hierzu werden Textlisten mit Fragen und zugehörigen Antworten angelegt. Über die Variable `QuestionID` verweist der Dienst auf eine Frage in der Textliste. Sie bekommen zu der Frage Antwortmöglichkeiten angezeigt. Bei Auswahl einer Antwort wird diese über die Variable `InteractQuestionID` an den Dienst geschrieben. Die Variable `AdditionalInfo` ermöglicht es, weitere Informationen in Textform bereitzustellen.



Für die Dienst-Bediener-Interaktion wird die Verwendung des MTP-Engineerings und des darin enthaltenen MTP-Exports empfohlen, da die Textlisten nicht über OPC-UA zur Verfügung gestellt werden, sondern in der MTP-Datei beschrieben werden!

Im untenstehenden Beispiel werden die Prozeduren, Konfigurationsparameter und Reportwerte im Deklarationsteil einer Ableitung des `FB_MTP_ServiceControl` deklariert und in Arrays zusammengefasst. Anschließend werden diese im Rumpf des `FB_MTP_ServiceControl` aufgerufen. Somit sind alle dienstrelevanten Prozeduren, Parameter und Reportwerte an einem Ort zusammengefasst.

Dabei kann immer nur eine Prozedur zur gleichen Zeit ausgeführt werden. Das Starten einer Prozedur kann nur erfolgen, wenn der Dienst im Zustand `Idle` ist.

## Syntax

```
FUNCTION_BLOCK FB_MTP_Service EXTENDS FB_MTP_ServiceControl
VAR
  ///Procedures - Parameters and Required Equipment
  ///Procedure 1
  <ProcParametersArray1>      : ARRAY[0..<n>] OF POINTER TO FB_MTP_ParameterElement :=
  [ADR(<ProcParam_0>), ADR(<ProcParam_1>)];
  <RequiredEquipmentArray1>  : ARRAY[0..<n>] OF I_MTP_ReqEq := [ADR(<ReqEq_0>), ADR(<ReqEq_1>)];

  ///Procedure 2
  <ProcParametersArray2>      : ARRAY[0..<n>] OF POINTER TO FB_MTP_ParameterElement := [ADR(<ProcPara
  m_1>), ADR(<ProcParam_2>)];

  <RequiredEquipmentArray2>  : ARRAY[0..<n>] OF I_MTP_ReqEq := [ADR(<ReqEq_0>), ADR(<ReqEq_1>)];

  ///Procedures
  ProcedureArray
    : ARRAY[1..<n>] OF POINTER TO FB_MTP_Procedure := [ADR(<Procedure_1>), ADR(<Procedure_1>)];

  ///ConfigurationParameters
  ConfParameterArray         : ARRAY[0..<n>] OF POINTER TO FB_MTP_ParameterElement := [ADR(<ConfParame
  ter_0>)]

  ///ProcedureParameters
  ProcParameterArray         : ARRAY[0..<z>] OF POINTER TO FB_MTP_ParameterElement := [ADR(<ProcParame
  ter_0>), ADR(<ProcParameter_1>), ADR(<ProcParameter_2>)];

  ///ReportValues
  ReportValueArray           : ARRAY[0..<n>] OF POINTER TO FB_MTP_ReportValue [ADR(<ReportValue_0>)]
END_VAR

VAR_INPUT
  ///Procedures
  <Procedure_1>               : FB_Procedure := (<initial values>);
  <Procedure_2>               : FB_Procedure := (<initial values>);

  ///ConfigurationParameters
  < ConfParameter_0>         : FB_MTP_AnaServParam := (<initial values>);

  ///ProcedureParameters
  (<ProcParameter_0>         : FB_MTP_AnaServParam := (<initial values>);
  <ProcParameter_1>         : FB_MTP_AnaServParam := (<initial values>);
  <ProcParameter_2>         : FB_MTP_AnaServParam := (<initial values>);

  ///ReportValues
  (<ReportValue_0>: FB_MTP_AnaReportValue := ((<initial values>);
END_VAR

<Procedure_1>(
  ProcParameters      := <ProcParametersArray_1>,
  RequiredEquipment  := <RequiredEquipmentArray_1>);

<Procedure_2>(
```

```

ProcParameters := <ProcParametersArray_2>,
RequiredEquipment := <RequiredEquipmentArray_2>;

SUPER^ (
  Procedures := ProcedureArray,
  ConfParameters := ConfParameterArray,
  ProcParameters := ProcParameterArray,
  ReportValues := ReportValueArray);

```

## Weitere Eigenschaften

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[OSLevel \[► 18\]](#)

[Service Mode \[► 17\]](#)

## Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
OSLevel	BYTE	Level muss für Anlage definiert werden. Wert 0 ist reserviert für lokale Bedienung.	Lesen/schreiben
Procedures	ARRAY [*] OF POINTER TO FB_MTP_Procedure	Array mit allen zum Dienst gehörenden Prozeduren	-
ConfParameters	ARRAY [*] OF POINTER TO FB_MTP_ParameterElement	Array mit allen zum Dienst gehörenden Konfigurationsparametern	-
ProcParameters	ARRAY [*] OF POINTER TO FB_MTP_ParameterElement	Array mit allen zum Dienst gehörenden Prozedurparametern	-
ReportValues	ARRAY [*] OF POINTER TO FB_MTP_ReportValue	Array mit allen zum Dienst gehörenden ReportValues	-
CommandInt	DWORD	Internal Steuerwort (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE)	Lesen
ProcedureInt	DWORD	Internal Prozedurvorgang (relevant, wenn StateAutAct = TRUE und SrcIntAct = TRUE)	Lesen
CommandEn	DWORD	Freigabe von Steuerbefehlen des Steuerworts	Lesen
State*		Siehe <a href="#">Service Mode [► 17]</a>	
Src*			
PosTextID	DWORD	ID-Variable für den Positionstext	Lesen



Name	Typ	Beschreibung	OPC-UA-Zugriff
InteractQuestionID	DWORD	ID-Variable für den Fragetext	Lesen
InteractAddInfo	STRING	Zusätzliche Information zur aktuellen InteractQuestionID	Lesen
ProcParamApplyInt	BOOL	Alle zum Dienst gehörenden <i>Internal</i> Prozessparameterwerte <i>VInt</i> übernehmen. (Relevant, wenn <i>StateAutAct = TRUE</i> und <i>SrcIntAct = TRUE</i> )	Lesen
ConfigParamApplyInt	BOOL	Alle zum Dienst gehörenden <i>Internal</i> Konfigurationsparameterwerte <i>VInt</i> übernehmen. (Relevant, wenn <i>StateAutAct = TRUE</i> und <i>SrcIntAct = TRUE</i> )	Lesen
ReportValueFreeze	BOOL	1: Einfrieren aller Reportwerte des Dienstes 0: Reportwerte des Dienstes nicht einfrieren	Lesen/schreiben

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
StateCur	DWORD	Aktueller Zustand des Dienstzustandsautomaten	Lesen
ProcedureCur	DWORD	Aktuell verwendete Prozedur	Lesen
ProcedureReq	DWORD	Aktuell vorgewählte Prozedur, welche bei Start ausgeführt wird.	Lesen
State* Src*		Siehe <a href="#">Service Mode [▶ 17]</a>	
ProcParamApplyEn	BOOL	Freigabe Prozedurparameterübernahme: 1: Parameter können übernommen werden. 0: Parameter können nicht übernommen werden.	Lesen
ConfigParamApplyEn	BOOL	Freigabe Konfigurationsparameterübernahme: 1: Parameter können übernommen werden. 0: Parameter können nicht übernommen werden.	Lesen

**Interne Variablen**

Name	Typ	Beschreibung	OPC-UA-Zugriff
State* Src*	BOOL	Siehe <a href="#">Service Mode [▶ 17]</a>	
CommandOp	DWORD	Operator Steuerwort (relevant, wenn <i>StateOpAct = TRUE</i> )	Lesen/schreiben
CommandExt	DWORD	External Steuerwort (relevant, wenn <i>StateAutAct = TRUE</i> und <i>SrcExtAct = TRUE</i> )	Lesen/schreiben
ProcedurOp	DWORD	Operator Prozedurvorgwahl (relevant, wenn <i>StateOpAct = TRUE</i> )	Lesen/schreiben
ProcedureExt	DWORD	External Prozedurvorgwahl (relevant, wenn <i>StateAutAct = TRUE</i> und <i>SrcExtAct = TRUE</i> )	Lesen/schreiben
InteractAnswerID	DWORD	ID-Variable für den Antworttext	Lesen/schreiben

Name	Typ	Beschreibung	OPC-UA-Zugriff
ProcParamApplyOp	BOOL	Alle zum Dienst gehörenden <code>Operator</code> Prozessparameterwerte <code>VOp</code> übernehmen. (Relevant, wenn <code>StateOpAct = TRUE</code> )	Lesen/schreiben
ProcParamApplyExt	BOOL	Alle zum Dienst gehörenden <code>External</code> Prozessparameterwerte <code>VExt</code> übernehmen. (Relevant, wenn <code>StateAutAct = TRUE</code> und <code>SrcExtAct = TRUE</code> )	Lesen/schreiben
ConfigParamApplyOp	BOOL	Alle zum Dienst gehörenden <code>Operator</code> Konfigurationsparameterwerte <code>VOp</code> übernehmen. (Relevant, wenn <code>StateOpAct = TRUE</code> )	Lesen/schreiben
ConfigParamApplyExt	BOOL	Alle zum Dienst gehörenden <code>External</code> Konfigurationsparameterwerte <code>VExt</code> übernehmen. (Relevant, wenn <code>StateAutAct = TRUE</code> und <code>SrcExtAct = TRUE</code> )	Lesen/schreiben

## Methoden

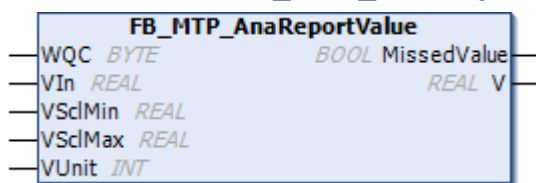
Name	Typ	Beschreibung
Idle		Nicht transienter Initialzustand. Es wird keine prozesstechnische Funktion ausgeführt.
IsActuatorsInternal	BOOL	1: Alle dem Dienst zugeordneten Aktoren befinden sich im Zustand <code>Internal</code> .
IsActuatorsManual	BOOL	1: Alle dem Dienst zugeordneten Aktoren befinden sich im Zustand <code>Manual</code> .
IsActuatorsOffline	BOOL	1: Alle dem Dienst zugeordneten Aktoren befinden sich im Zustand <code>Offline</code> .
IsActuatorsOperator	BOOL	1: Alle dem Dienst zugeordneten Aktoren befinden sich im Zustand <code>Operator</code> .
IsActuatorsAutomatic	BOOL	1: Alle dem Dienst zugeordneten Aktoren befinden sich im Zustand <code>Automatic</code> .
IsReqEqInternal	BOOL	1: Alle dem Dienst zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Internal</code>
IsReqEqManual	BOOL	1: Alle dem Dienst zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Manual</code>
IsReqEqOffline	BOOL	1: Alle dem Dienst zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Offline</code>
IsReqEqOperator	BOOL	1: Alle dem Dienst zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Operator</code>
IsReqEqAutomatic	BOOL	1: Alle dem Dienst zugeordneten Aktoren und PID-Logik befinden sich im Zustand <code>Automatic</code>
SetActuatorsInternal	BOOL	Alle dem Dienst zugeordneten Aktoren werden in den Zustand <code>Internal</code> überführt.
SetActuatorsManual	BOOL	Alle dem Dienst zugeordneten Aktoren werden in den Zustand <code>Manual</code> überführt.
SetActuatorsOffline	BOOL	Alle dem Dienst zugeordneten Aktoren werden in den Zustand <code>Offline</code> überführt.
SetActuatorsOperator	BOOL	Alle dem Dienst zugeordneten Aktoren werden in den Zustand <code>Operator</code> überführt.
SetActuatorsAutomatic	BOOL	Alle dem Dienst zugeordneten Aktoren werden in den Zustand <code>Automatic</code> überführt.
SetReqEqInternal	BOOL	Alle dem Dienst zugeordneten Aktoren und PID-Logik werden in den Zustand <code>Internal</code> überführt.

Name	Typ	Beschreibung
SetReqEqManual	BOOL	Alle dem Dienst zugeordneten Aktoren und PID-Logik werden in den Zustand <i>Manual</i> überführt.
SetReqEqOffline	BOOL	Alle dem Dienst zugeordneten Aktoren und PID-Logik werden in den Zustand <i>Offline</i> überführt.
SetReqEqOperator	BOOL	Alle dem Dienst zugeordneten Aktoren und PID-Logik werden in den Zustand <i>Operator</i> überführt.
SetReqEqAutomatic	BOOL	Alle dem Dienst zugeordneten Aktoren und PID-Logik werden in den Zustand <i>Automatic</i> überführt.
SetOffline	BOOL	Den Dienst in den Zustand <i>Offline</i> überführen.
SetOperator	BOOL	Den Dienst in den Zustand <i>Operator</i> überführen.
SetAutomaticIntern	BOOL	Den Dienst in den Zustand <i>AutomaticIntern</i> überführen.
SetAutomaticExtern	BOOL	Den Dienst in den Zustand <i>AutomaticExtern</i> überführen.
SetProcParamApplyEn	BOOL	Wertvorgabe (Eingangsvariable der Methode <i>ApplyEn</i> ) für Eingangsvariable <i>ApplyEn</i> der Prozedurparameter: Alle dem Dienst zugeordneten Dienstparameter (wenn <i>SyncOnly</i> = FALSE) Alle dem Dienst zugeordneten Dienstparameter mit Eingangsvariable <i>Sync</i> = TRUE (wenn Eingangsvariable der Methode <i>SyncOnly</i> = TRUE)
SetConfigParamApplyEn	BOOL	Wertvorgabe (Eingangsvariable Methode <i>ApplyEn</i> ) für Konfigurationsparameter: Alle dem Dienst zugeordneten Dienstparameter (wenn <i>SyncOnly</i> = FALSE) Alle dem Dienst zugeordneten Dienstparameter mit Eingangsvariable <i>Sync</i> = TRUE (wenn Eingangsvariable der Methode <i>SyncOnly</i> = TRUE)
ToOnline	BOOL	Implementierung von Logik bei Verlassen des Zustands <i>Offline</i> .
ToOffline	BOOL	Implementierung von Logik bei Betreten des Zustands <i>Offline</i> .

### 5.3.10 Report Values

Berichtswerte werden einem Dienst zugeordnet. Zum Aktualisieren der Berichtswerte müssen neue Werte an den Funktionsbaustein übergeben und der Funktionsbaustein muss manuell aufgerufen werden.

#### 5.3.10.1 FB\_MTP\_AnaReportValue



Der Funktionsbaustein **FB\_MTP\_AnaReportValue** stellt einen analogen Eingangswert *VIn* einer manuellen Bedienung (z. B. über OPC-UA) für die Dokumentation zur Verfügung (Ausgang *V*). Dieser Wert wird genau einem Dienst zugewiesen. Innerhalb dieses Dienstes kann er einer oder mehreren Prozeduren zugewiesen werden.

#### Wert einfrieren

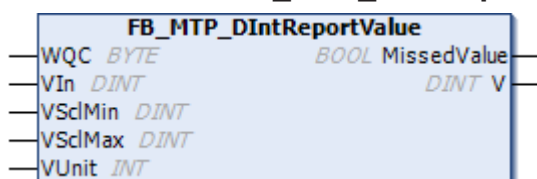
Über den übergeordneten Dienst kann der Wert *v* eingefroren werden (siehe [FB\\_MTP\\_ServiceControl](#) [► 107]). Sollte sich der Eingangswert *VIn* während dieser Zeit mehr als einmal ändern, wird dies über den Ausgang *MissedValue* = TRUE signalisiert.

**Weitere Eigenschaften**[Name des Objekts \[► 18\]](#)[Objektbeschreibung \[► 18\]](#)[WQC \[► 19\]](#)[Wertskalierung \[► 19\]](#)[Einheit \[► 19\]](#)**🔴 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
VIn	REAL	Eingangswert	-
VScIMin	REAL	Skalenanfang für Wertdarstellung	Lesen
VScIMax	REAL	Skalenende für Wertdarstellung	Lesen
VUnit	INT	Einheit des Werts	Lesen

**🔴 Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
MissedValue	BOOL	Änderung des Eingangswerts $V_{In}$ während des Einfrierens von $V$ :  1: Der Eingangswert $V_{In}$ hat sich seit dem Einfrieren mindestens zweimal geändert.  0: Der Eingangswert $V_{In}$ hat sich seit dem Einfrieren nicht mehr als einmal geändert.	Lesen
V	REAL	Wert, der über OPC-UA zur Verfügung gestellt wird.	Lesen

**5.3.10.2 FB\_MTP\_DIntReportValue**

Der Funktionsbaustein `FB_MTP_DIntReportValue` stellt einen ganzzahligen Eingangswert  $V_{In}$  einer manuellen Bedienung (z. B. über OPC-UA) für die Dokumentation zur Verfügung (Ausgang  $V$ ). Dieser Wert wird genau einem Dienst zugewiesen. Innerhalb dieses Dienstes kann er einer oder mehreren Prozeduren zugewiesen werden.

**Wert einfrieren**

Über den übergeordneten Dienst kann der Wert  $V$  eingefroren werden (siehe [FB\\_MTP\\_ServiceControl \[► 107\]](#)). Sollte sich der Eingangswert  $V_{In}$  während dieser Zeit mehr als einmal ändern, wird dies über den Ausgang `MissedValue = TRUE` signalisiert.

**Weitere Eigenschaften**[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

[Wertskalierung \[► 19\]](#)

[Einheit \[► 19\]](#)

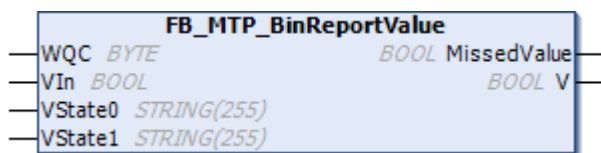
**🔴 Eingänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
VIn	DINT	Eingangswert	-
VScIMin	DINT	Skalenanfang für Wertdarstellung	Lesen
VScIMax	DINT	Skalenende für Wertdarstellung	Lesen
VUnit	INT	Einheit des Werts	Lesen

**🔴 Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
MissedValue	BOOL	Änderung des Eingangswerts VIn während des Einfrierens von V:  1: Der Eingangswert VIn hat sich seit dem Einfrieren mindestens zweimal geändert.  0: Der Eingangswert VIn hat sich seit dem Einfrieren nicht mehr als einmal geändert.	Lesen
V	DINT	Wert, der über OPC-UA zur Verfügung gestellt wird.	Lesen

**5.3.10.3 FB\_MTP\_BinReportValue**



Der Funktionsbaustein `FB_MTP_BinReportValue` stellt einen binären Eingangswert `VIn` einer manuellen Bedienung (z. B. über OPC-UA) für die Dokumentation zur Verfügung (Ausgang `V`). Dieser Wert wird genau einem Dienst zugewiesen. Innerhalb dieses Dienstes kann er einer oder mehreren Prozeduren zugewiesen werden.

**Wert einfrieren**

Über den übergeordneten Dienst kann der Wert `V` eingefroren werden (siehe [FB\\_MTP\\_ServiceControl \[► 107\]](#)). Sollte sich der Eingangswert `VIn` während dieser Zeit mehr als einmal ändern, wird dies über den Ausgang `MissedValue = TRUE` signalisiert.

**Weitere Eigenschaften**

[Name des Objekts \[► 18\]](#)

[Objektbeschreibung \[► 18\]](#)

[WQC \[► 19\]](#)

### 🔍 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
VIn	BOOL	Eingangswert	-
VState0	STRING	Text für VIn = FALSE	Lesen
VState1	STRING	Text für VIn = TRUE	Lesen

### 🔍 Ausgänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
MissedValue	BOOL	Änderung des Eingangswerts VIn während des Einfrierens von V:  1: Der Eingangswert VIn hat sich seit dem Einfrieren mindestens zweimal geändert.  0: Der Eingangswert VIn hat sich seit dem Einfrieren nicht mehr als einmal geändert.	Lesen
V	BOOL	Wert, der über OPC-UA zur Verfügung gestellt wird.	Lesen

#### 5.3.10.4 FB\_MTP\_StringReportValue



Der Funktionsbaustein `FB_MTP_StringReportValue` stellt die Eingangsvariable `TextIn` einer manuellen Bedienung (z. B. über OPC-UA) für die Dokumentation zur Verfügung (Ausgang `Text`). Dieser Wert wird genau einem Dienst zugewiesen. Innerhalb dieses Dienstes kann er einer oder mehreren Prozeduren zugewiesen werden.

#### Wert einfrieren

Über den übergeordneten Dienst kann der Wert `Text` eingefroren werden (siehe [FB\\_MTP\\_ServiceControl](#) [▶ 107]). Sollte sich der Eingangswert `TextIn` während dieser Zeit mehr als einmal ändern, wird dies über den Ausgang `MissedValue = TRUE` signalisiert.

#### Weitere Eigenschaften

[Name des Objekts](#) [▶ 18]

[Objektbeschreibung](#) [▶ 18]

[WQC](#) [▶ 19]

### 🔍 Eingänge

Name	Typ	Beschreibung	OPC-UA-Zugriff
TagName	STRING	Name der Schnittstelle	-
TagDescription	STRING	Beschreibung der Schnittstelle	-
WQC	BYTE	Worst Quality Code	Lesen
TextIn	STRING	Eingangstext	-

 **Ausgänge**

Name	Typ	Beschreibung	OPC-UA-Zugriff
MissedValue	BOOL	Änderung des Eingangswerts <code>TextIn</code> während des Einfrierens von <code>Text</code> :  1: Der Eingangswert <code>TextIn</code> hat sich seit dem Einfrieren mindestens zweimal geändert.  0: Der Eingangswert <code>TextIn</code> hat sich seit dem Einfrieren nicht mehr als einmal geändert.	Lesen
Text	STRING	Wert, der über OPC-UA zur Verfügung gestellt wird.	Lesen

## 5.4 Datentypen

In diesem Kapitel werden die verwendeten Datentypen beschrieben.

### 5.4.1 ST\_MTP\_InputElementConfig

`ST_MTP_InputElementConfig` ist eine Konfiguration für `InputElements`. Sie beinhaltet Einheiten mit zugehörigen Wertdarstellungen und wird als Array übergeben. Bei der Übergabe an `DIntProcessValueIn` werden die Variablen `Sc1MinExt` und `Sc1MaxExt` in den Datentypen `DInt` umgewandelt.

**Werte**

Name	Typ	Beschreibung
UnitExt	E_MTP_Unit	Einheit
Sc1MinExt	REAL	Unterer Grenzwert der Wertdarstellung
Sc1MaxExt	REAL	Oberer Grenzwert der Wertdarstellung

## 6 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](http://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](mailto: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](mailto: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](mailto:info@beckhoff.com)

Internet: [www.beckhoff.com](http://www.beckhoff.com)



## **Trademark statements**

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Mehr Informationen:  
**[www.beckhoff.com/tf8400](http://www.beckhoff.com/tf8400)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

