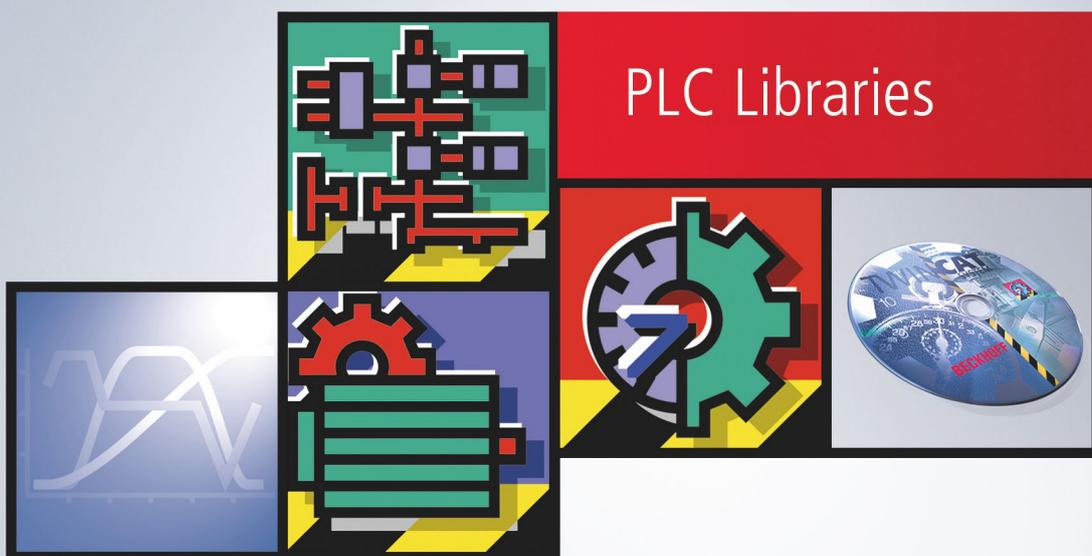


Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: TcMPBus



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Einleitung.....	8
3	Zielgruppen.....	9
4	MP-Bus	10
4.1	Topologie	10
4.2	Aktor Lösungen	10
5	Integration in TwinCAT	14
5.1	KL6771 - Verknüpfung mit dem TwinCAT System Manager	14
5.2	Integration in TwinCAT (CX9020)	16
5.3	Integration in TwinCAT (BC9191)	18
6	Programmierung	22
6.1	Allgemeine Informationen	25
6.2	Funktionsbausteine	25
6.2.1	KL6771	27
6.2.2	MP_Addressing.....	27
6.2.3	MP_CMV.....	29
6.2.4	MP_DamperLinearActuator.....	31
6.2.5	MP_EnergyValveV4_Configuration.....	32
6.2.6	MP_EnergyValveV4_Process	35
6.2.7	MP_EPIV.....	37
6.2.8	MP_EPIV_R6	39
6.2.9	MP_EPIV_R6_Parameter	40
6.2.10	MP_EV	42
6.2.11	MP_EV_Parameter	44
6.2.12	MP_MPX.....	45
6.2.13	MP_OperatingUnit.....	46
6.2.14	MP_OperatingUnit_ConfigurationCO2.....	48
6.2.15	MP_OperatingUnit_ConfigurationDisplay	49
6.2.16	MP_OperatingUnit_ConfigurationOffsetValues.....	50
6.2.17	MP_OperatingUnit_ConfigurationStatusIcons	51
6.2.18	MP_OperatingUnit_ConfigurationTemp	52
6.2.19	MP_OperatingUnit_ConfigurationVentilation	53
6.2.20	MP_PTH.....	54
6.2.21	MP_RoomSensor.....	55
6.2.22	MP_RoomSensor_Parameter	57
6.2.23	MP_Smoker	58
6.2.24	MP_TEM_Configuration.....	59
6.2.25	MP_TEM_Process	61
6.2.26	MP_UST_3.....	62
6.2.27	MP_VAV.....	65

6.2.28	MP_VRU_Configuration.....	67
6.2.29	MP_VRU_Process	68
6.2.30	MP_Window	70
6.2.31	MPL_DamperLinearActuator.....	71
6.3	Funktionen	72
6.3.1	NI1000_LuS_TO_INT : INT.....	72
6.3.2	NI1000_TO_INT : INT	72
6.3.3	NTC_TO_INT : INT	73
6.3.4	PT1000_TO_INT : INT	73
6.4	Datentypen	74
6.4.1	Enums	74
6.4.2	Strukturen.....	85
6.5	Fehlercodes	88
7	Anhang.....	90
7.1	Beispiele.....	90
7.2	Support und Service.....	90

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Einleitung

Die MP-Bus-Bibliothek ist eine TwinCAT SPS Bibliothek zum Datenaustausch mit MP-Bus Geräten.

Alle Funktionsblöcke aus der Bibliothek müssen in derselben Task aufgerufen werden.

Diese Bibliothek ist nur in Verbindung mit einer MP-Bus-Masterklemme KL6771 einzusetzen.

3 Zielgruppen

Für den Nutzer dieser Bibliothek werden folgende Grundkenntnisse vorausgesetzt:

- TwinCAT PLC Control
- TwinCAT System Manager
- PC und Netzwerkkennnisse
- Aufbau und Eigenschaften der Beckhoff Embedded-PC und deren Busklemmensystem
- Technologie von MP-Bus Geräten
- Einschlägige Sicherheitsvorschriften der technischen Gebäudeausrüstung

Diese Softwarebibliothek ist für Gebäudeautomation-Systempartner der Beckhoff Automation GmbH & Co. KG. Die Systempartner sind tätig in dem Bereich Gebäudeautomation und beschäftigen sich mit Errichtung, Inbetriebsetzung, Erweiterung, Wartung und Service von mess-, steuer- und regelungstechnischen Anlagen der technischen Gebäudeausrüstung.

4 MP-Bus

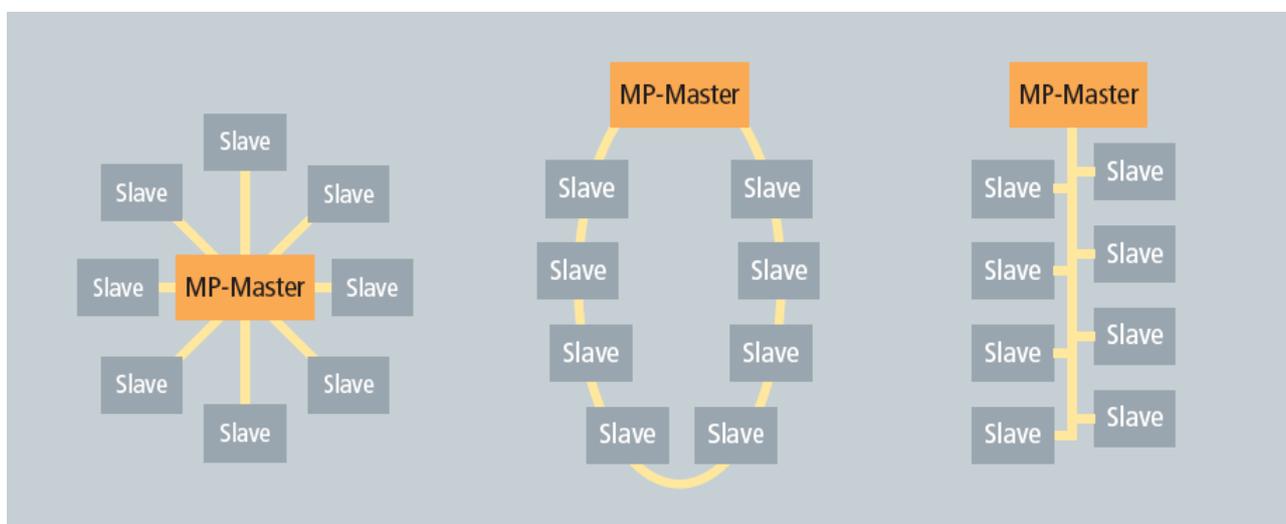
MP-Bus = Multi-Point Bus

Der MP-Bus (Multi-Point) ist ein einfacher Sensor-/Aktor-Bus, welcher für bestimmte Gewerke der Gebäudeautomation genutzt wird. Der MP-Bus dient zur Steuerung von Stellantrieben im HLK-Bereich, für Klappen, Regelventile und Volumenstromregler aus dem Produktportfolio von Belimo. Es können bis zu acht unterschiedliche Geräte aus dem Bereich HLK in 3-Leitertechnik an einen MP-Bus-Master angeschlossen werden. Zusätzlich kann an jeden dieser acht Teilnehmer ein Sensor angeschlossen werden, der durch den MP-Bus angesprochen wird. Eine zusätzliche Produktreihe mit MP-Bus-Anschluss ist das Fensterlüftungssystem FLS von Belimo (Anschluss der Belimo-Antriebe siehe Dokumentation Belimo).

Der MP-Bus wurde von der Firma Belimo für den Anschluss von Ventilen, Drossel-, Lüftungs- und Brandschutzklappen sowie für Fensterlüftungssysteme entwickelt.

4.1 Topologie

Es gibt keinerlei Einschränkungen bzgl. des topologischen Aufbaus der MP-Bus-Stränge: Stern-, Ring-, Baum- oder gemischte Topologien sind möglich. Die Länge des gesamten Busstrangs ist abhängig vom gewählten Kabelquerschnitt und Art sowie Anzahl der angeschlossenen Antriebe! Weitere Informationen enthält die Dokumentation der Firma Belimo.



4.2 Aktor Lösungen

Typen ¹	Einheit	bereit für erweiterten MP- Bus® ²	Beckhoff Funktions- baustein
Lüftungsanwendungen			
Drehantriebe			
CM24-MPL-L	2 Nm	•	MPL_DamperLinearActuator
CM24-MPL-R	2 Nm	•	
LM24A-MP	5 Nm		MP_DamperLinearActuator
NM24A-MP	10 Nm		
SM24A-MP	20 Nm		
GM24A-MP	40 Nm		
Drehantriebe mit Notstellfunktion			
TF24-MFT	2 Nm		MP_DamperLinearActuator
LF24-MFT2	4 Nm		
NF24A-MP	10 Nm		

Typen ¹	Einheit	bereit für erweiterter MP-Bus® ²	Beckhoff Funktionsbaustein
SF24A-MP	20 Nm		
EF24A-MP	30 Nm		
GK24A-MP	40 Nm		
Linearantriebe			
LH24A-MP... 60/100/200/300	150 Nm		MP_DamperLinearActuator
SH24A-MP... 100/200/300	450 Nm		
Linearantriebe mit Notstellfunktion			
LHK24A-MP100	150 Nm		MP_DamperLinearActuator
SHK24A-MP100	450 Nm		
Rotativantriebe			
LU24A-MP	3 Nm		MP_DamperLinearActuator
Schnellläufer-Drehantriebe			
NMC24A-MP	10 Nm		MP_DamperLinearActuator
SMC24A-MP	20 Nm		
Rotativantriebe für spezielle Anwendungen			
NM24P-MP	10 Nm		MP_DamperLinearActuator
SM24P-MP	20 Nm		
GM24G-MP-T	40 Nm		
Rotativantriebe mit Notstellfunktion für spezielle Anwendungen			
NF24G-MP-L	10 Nm		MP_DamperLinearActuator
SF24G-MP-L	20 Nm		
GK24G-MP	40 Nm		
Wasseranwendungen			
Antriebe für Regelkugelhahnen / Auf-Zu Regelkugelhahnen			
CQ24A-MPL	1 Nm	•	MPL_DamperLinearActuator
LR24A-MP	5 Nm		MP_DamperLinearActuator
NR24A-MP	10 Nm		
SR24A-MP	20 Nm		
Antriebe mit Notstellfunktion für Regelkugelhahnen / Auf-Zu Regelkugelhahnen			
TRF24-MFT	2 Nm		MP_DamperLinearActuator
LRF24-MP	4 Nm		
NRF24A-MP	10 Nm		
Antriebe für Hubventile			
LV24A-MP-TPC	500 N		MP_DamperLinearActuator
NV24A-MP-TPC	1000 N		
SV24A-MP-TPC	1500 N		
EV24A-MP-TPC	2500 N		
Antriebe mit Notstellfunktion für Hubventile			
NVK24A-MP-TPC	1000 N		MP_DamperLinearActuator
AVK24A-MP-TPC	2000 N		
Schnellläufer-Drehantriebe für Hubventile			
LVC24A-MP-TPC	500 N		MP_DamperLinearActuator
NVC24A-MP-TPC	1000 N		
SVC24A-MP-TPC	1500 N		
Schnellläufer-Drehantriebe mit Notstellfunktion für Hubventile			

Typen ¹	Einheit	bereit für erweiterten MP- Bus® ²	Beckhoff Funktions- baustein
NVKC24A-MP-TPC	1000 N		MP_DamperLinearActuator
Antriebe für Drosselklappen			
SR24A-MP-5	20 Nm		MP_DamperLinearActuator
GR24A-MP-5/7	40 Nm		
DR24A-MP-...5/7	90 Nm		
PRCA-BAC-S2-T	160 Nm		
PRKCA-BAC-S2-T	160 Nm		
SY2-24-MP-T	90 Nm		
SY2-230-MP-T	90 Nm		
SY3-24-MP-T	150 Nm		
SY3-230-MP-T	150 Nm		
SY4-24-MP-T	400 Nm		
SY4-230-MP-T	400 Nm		
SY5-24-MP-T	500 Nm		
Sicherheitsanwendungen			
"BF-TopLine" Antriebe für Brandschutzklappen			
BKN230-24MP zur Anbindung von BF(G)24TL-T-ST	11 / 18 Nm		MP_Smoker
Standardantrieb für Brandschutzklappen			
BKN230-24-C-MP zur Anbindung von BF..24-..ST	4 / 9 / 11 / 18 Nm		MP_Smoker
Raum- und Systemanwendungen			
VAV-Compact Drehantriebe			
LMV-D3-MP	5 Nm		MP_VAV
NMV-D3-MP	10 Nm		
SMV-D3-MP	20 Nm		
VAV-Compact Linearantriebe			
LHV-D3-MP	150 N		MP_VAV
VAV-Universal			
VRP-M			MP_VAV
VRU-D3-BAC			MP_VRU_Process
VRU-M1-BAC			MP_VRU_Configuration
VRU-M1R-BAC			
VAV Steuerungssystem für geregeltes Raumklima			
CMV-100-MP	DN 100	•	MP_CMV
CMV-125-MP	DN 125	•	
CMV-150-MP	DN 150	•	
CMV-160-MP	DN 160	•	
HVAC Leistungsgeräte			
EPIV			
EP..R-R6+BAC	DN 15-20	•	MP_EPIV_R6 MP_EPIV_R6_Parameter
EP..R+MP	DN 15-20		MP_EPIV
P..W..E-MP	DN 65-150		
Energy Valve™			

Typen ¹	Einheit	bereit für erweiterterten MP-Bus® ²	Beckhoff Funktionsbaustein
EV..R+BAC	DN 15-50	•	MP_EV
P..W..EV-BAC	DN 65-150	•	MP_EV_Parameter
Energy Valve™ V4			
EV..R2+..	DN 15-50	•	MP_EnergyValveV4_Process MP_EnergyValveV4_Configuration
Thermal Energy Meter			
22PE- ..		•	MP_TEM_Process
22PEM-..		•	MP_TEM_Configuration
Sensoren			
Raumsensoren in der Komfortzone			
22RT-19-1 (Temp.)		•	MP_RoomSensor
22RTH-19-1 (Temp., rH)		•	MP_RoomSensor_Parameter
22RTM-19-1 (Temp., CO ₂ , rH)		•	
MS24A-R02-MPX (Temp., CO ₂)		•	MP_MPX
MS24A-R08-MPX (Temp., VOC, CO ₂ , rH)		•	
Raumbediengeräte			
P-22Rxx-1900x-1		•	MP_OperatingUnit MP_OperatingUnit_ConfigurationCO2 MP_OperatingUnit_ConfigurationTemp MP_OperatingUnit_ConfigurationDisplay MP_OperatingUnit_ConfigurationVentilation MP_OperatingUnit_ConfigurationStatusIcons MP_OperatingUnit_ConfigurationOffsetValues

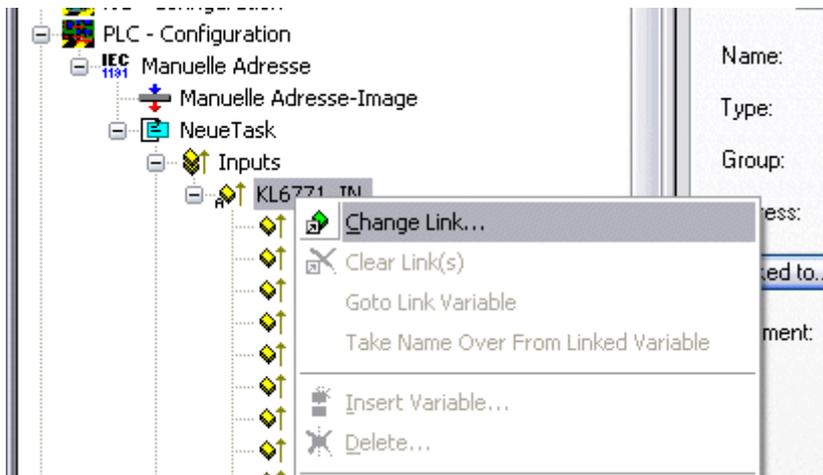
¹ Die aktuell verfügbare Produktpalette finden Sie online unter www.belimo.eu.

² Die markierten Geräte unterstützen den erweiterten Adressbereich. Es können bis zu 16 MP-Geräte (adressiert MP1 ... MP16) an eine Datenleitung angeschlossen werden. Wenn nicht markierte Geräte an dieselbe Datenleitung angeschlossen sind, ist der gemeinsame Adressbereich auf 8-MP-Geräte zu beschränken.

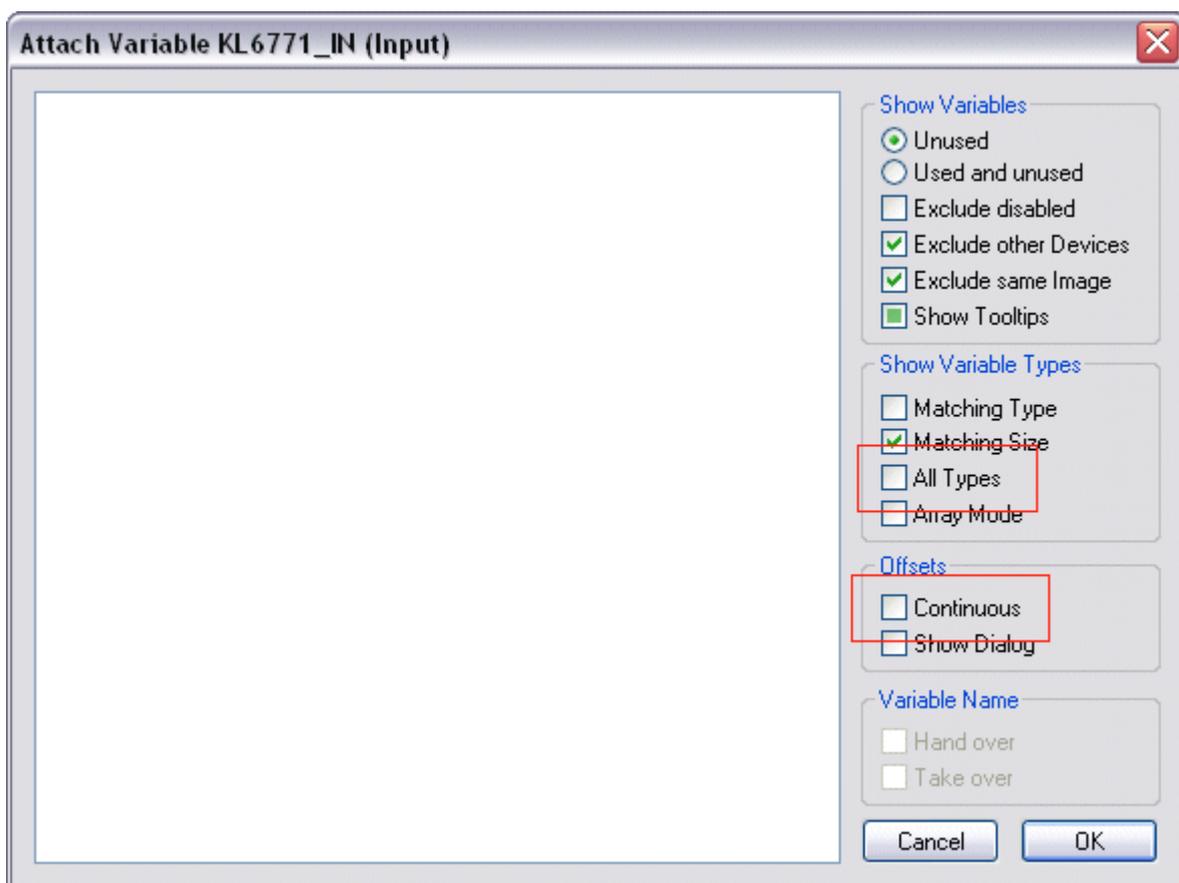
5 Integration in TwinCAT

5.1 KL6771 - Verknüpfung mit dem TwinCAT System Manager

Wie verknüpfe ich die KL6771 mit dem System Manager?



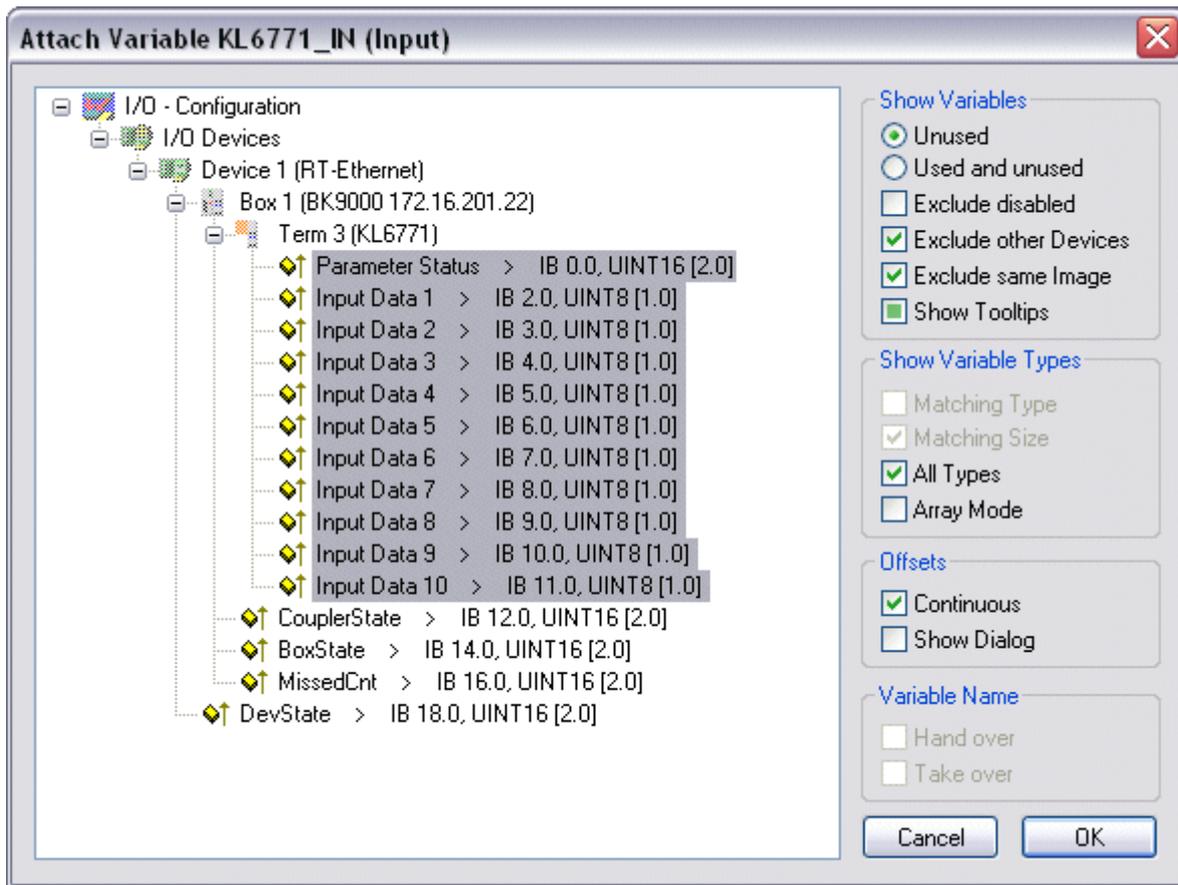
1. Wählen Sie **All Types** und **Continuous** an



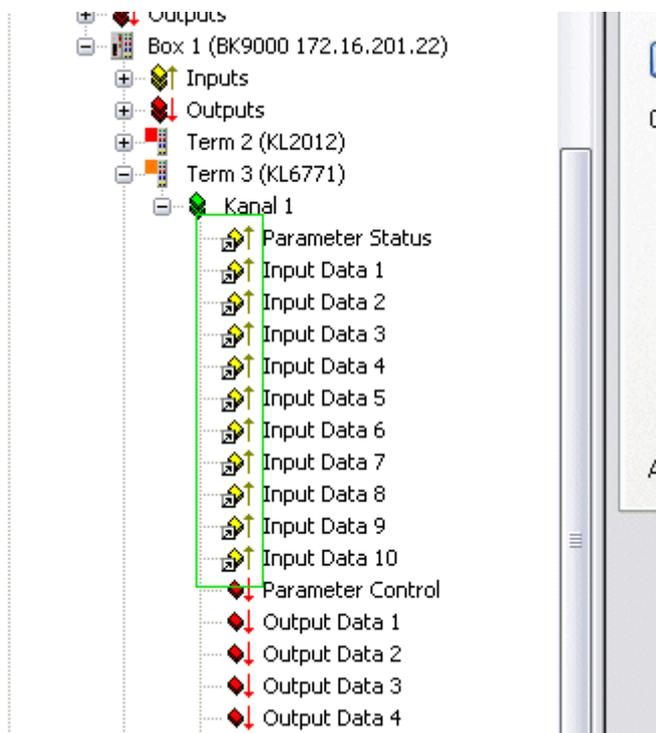
2. Klicken Sie mit der Maus auf die erste Variable der KL6771 **Parameter Status**. Drücken Sie dann die [SHIFT] Taste und halten Sie diese gedrückt. Gehen Sie mit dem Mauszeiger auf die letzte Variable der KL6771 **Input Data 10** und klicken Sie wiederum die linke Maustaste. Jetzt lassen Sie die [SHIFT] Taste wieder los.

⇒ Alle Daten der Klemme müssen jetzt markiert sein (siehe Bild).

3. Drücken Sie anschließend den Button **OK**.



4. Sie können die Verknüpfung jetzt kontrollieren. Gehen Sie dazu auf die KL6771 und öffnen Sie diese. Alle Daten der Klemme müssen jetzt mit einem kleinen Pfeil markiert sein (siehe Bild).



⇒ Ist dies der Fall fahren Sie genauso mit den Ausgängen fort.

5.2 Integration in TwinCAT (CX9020)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für MP-Bus in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll ein einzelner Klappenantrieb angesteuert und per Taster verändert werden.

<https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063685643.zip> <https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063685643.zip>

Hardware

Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Busklemmen Controller [CX9020](#)
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Öffnen und Schließen Funktion)
- 1x MP-Bus-Masterklemme [KL6771](#)
- 1x Endklemme KL9010

Richten Sie die Hardware sowie die MP-Bus-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass die Adresse vom Klappenantrieb bekannt ist.

Software

Erstellung des SPS-Programms

Erstellen Sie ein neues SPS-Projekt für PC-basierte Systeme (ARM) und fügen die Bibliothek *TcMPBus.lib* hinzu.

Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
  bOpen      AT %I*    : BOOL;
  bClose     AT %I*    : BOOL;
  arrKL6771_IN AT %I*  : ARRAY[0..11] OF BYTE;
  arrKL6771_OUT AT %Q* : ARRAY[0..11] OF BYTE;
  stDataKL6771 : DataKL6771;
END_VAR
```

bOpen: Eingangsvariable für den Öffnen-Taster.

bClose: Eingangsvariable für den Schließen-Taster.

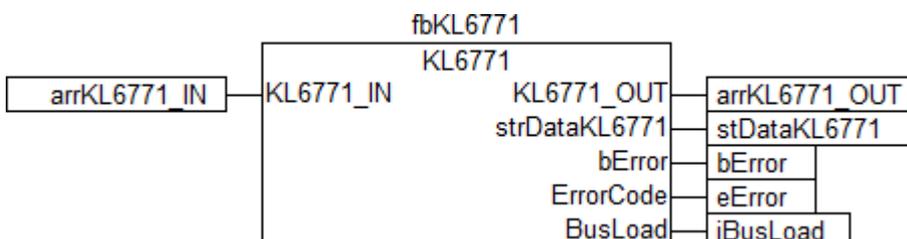
arrKL6771_IN: Eingangsvariable für die MP-Bus-Klemme.

arrKL6771_OUT: Ausgangsvariable für die MP-Bus-Klemme.

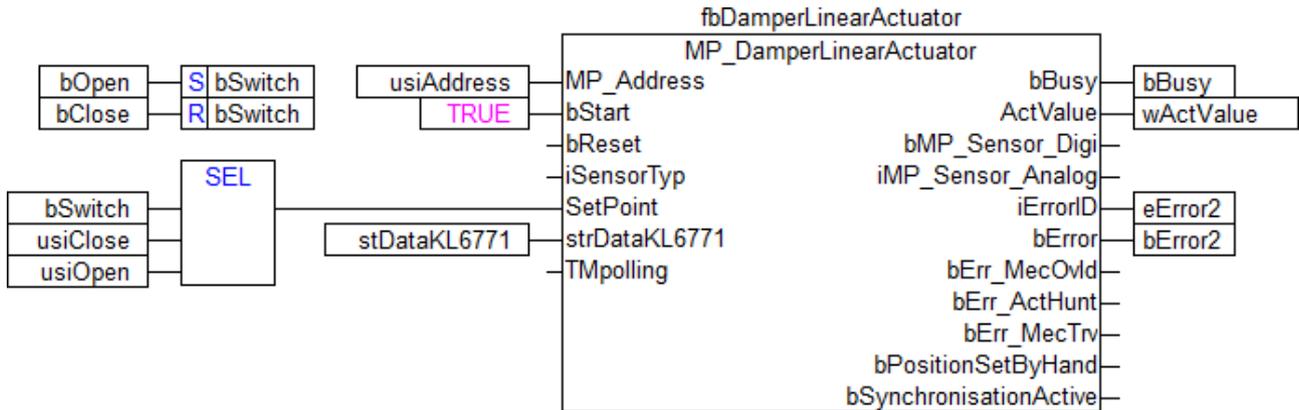
stDataKL6771: Wird für die Kommunikation mit MP-Bus benötigt (siehe [DataKL6771](#) [► 85]).

Alle Bausteine bei MP-Bus müssen in derselben Task aufgerufen werden.

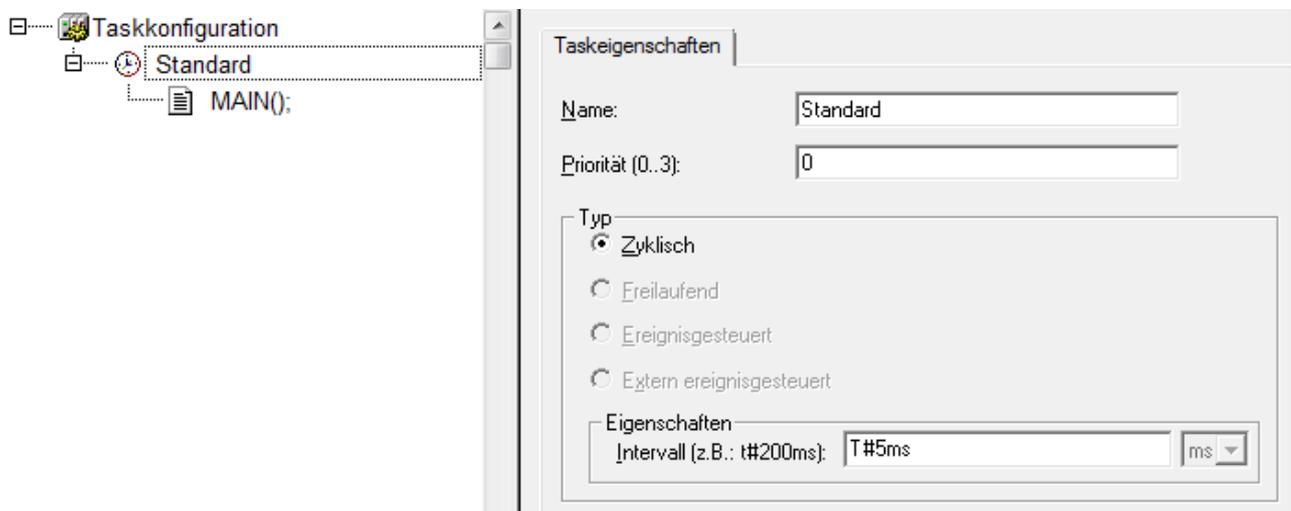
Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine [KL6771\(\)](#) [► 27] und [MP_DamperLinearActuator\(\)](#) [► 31] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *arrKL6771_IN*, *arrKL6771_OUT* und *stDataKL6771* zu verknüpfen.



Der Eingang *SetPoint* wird in Abhängigkeit der gewählten Funktion gesetzt. Verknüpfen Sie dazu die globalen Variablen *bOpen* und *bClose* mit einer Hilfsvariable.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins [KL6771\(\)](#) [27].

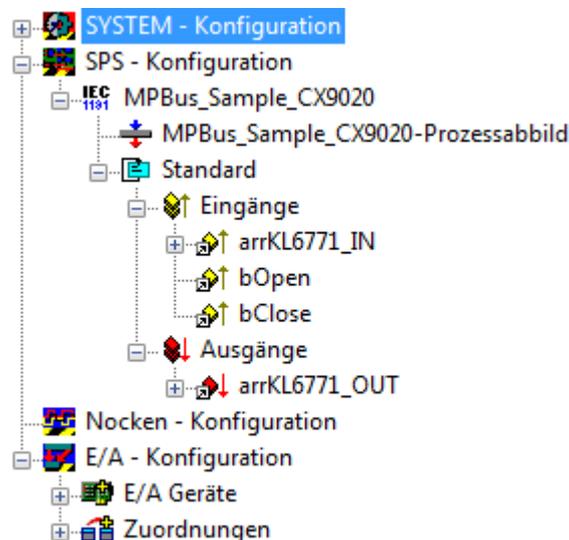


Laden Sie das Projekt als Bootprojekt auf den CX und speichern Sie es ab.

Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den CX und lassen Sie sich nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.

Ihr CX ist jetzt einsatzbereit.

Nach Betätigen der Taster kann der Klappenantrieb gefahren werden. Die aktuelle Position steht in der Variable *wActValue*.

5.3 Integration in TwinCAT (BC9191)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für MP-Bus in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll ein einzelner Klappenantrieb angesteuert und per Taster verändert werden.

<https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063687051.zip> <https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063687051.zip>

Hardware

Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Busklemmen Controller [BC9191](#)
- 1x Potenzialeinspeiseklemme 24V DC
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Öffnen und Schließen Funktion)
- 1x MP-Bus-Masterklemme [KL6771](#)
- 1x Endklemme KL9010

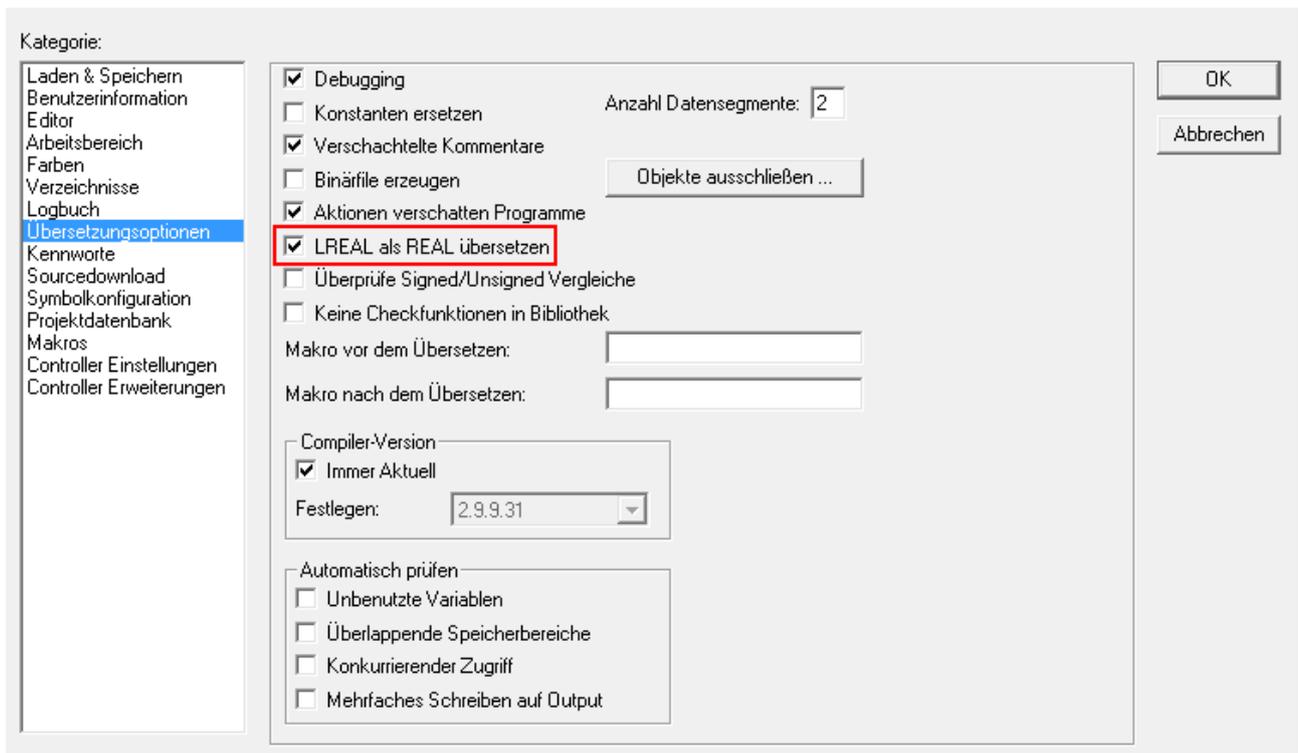
Richten Sie die Hardware sowie die MP-Bus-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass die Adresse vom Klappenantrieb bekannt ist.

Software

Erstellung des SPS-Programms

Erstellen Sie ein neues SPS-Projekt für BC-basierte Systeme (BCxx50 über AMS) und fügen die Bibliotheken *TcMPBus.lbx* und *TcSystemBCxx50.lbx* hinzu. Gehen Sie danach im Menü auf *Projekt* → *Optionen...* → *Übersetzungsoptionen* und wählen *LREAL als REAL übersetzen* an.



Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
  bOpen      AT %I*    : BOOL;
  bClose     AT %I*    : BOOL;
  arrKL6771_IN AT %I*  : ARRAY[0..11] OF BYTE;
  arrKL6771_OUT AT %Q* : ARRAY[0..11] OF BYTE;
  stDataKL6771 : DataKL6771;
END_VAR
```

bOpen: Eingangsvariable für den Öffnen-Taster.

bClose: Eingangsvariable für den Schließen-Taster.

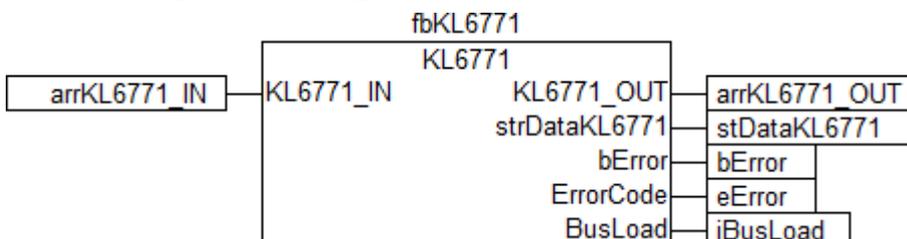
arrKL6771_IN: Eingangsvariable für die MP-Bus-Klemme.

arrKL6771_OUT: Ausgangsvariable für die MP-Bus-Klemme.

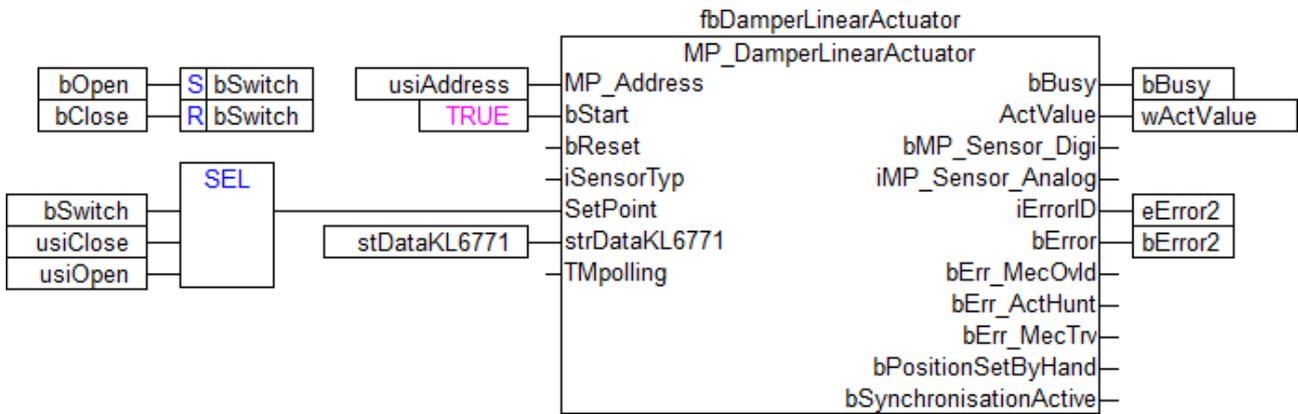
stDataKL6771: Wird für die Kommunikation mit MP-Bus benötigt (siehe [DataKL6771](#) [▶ 85]).

Alle Bausteine bei MP-Bus müssen in einer Task ausgeführt werden.

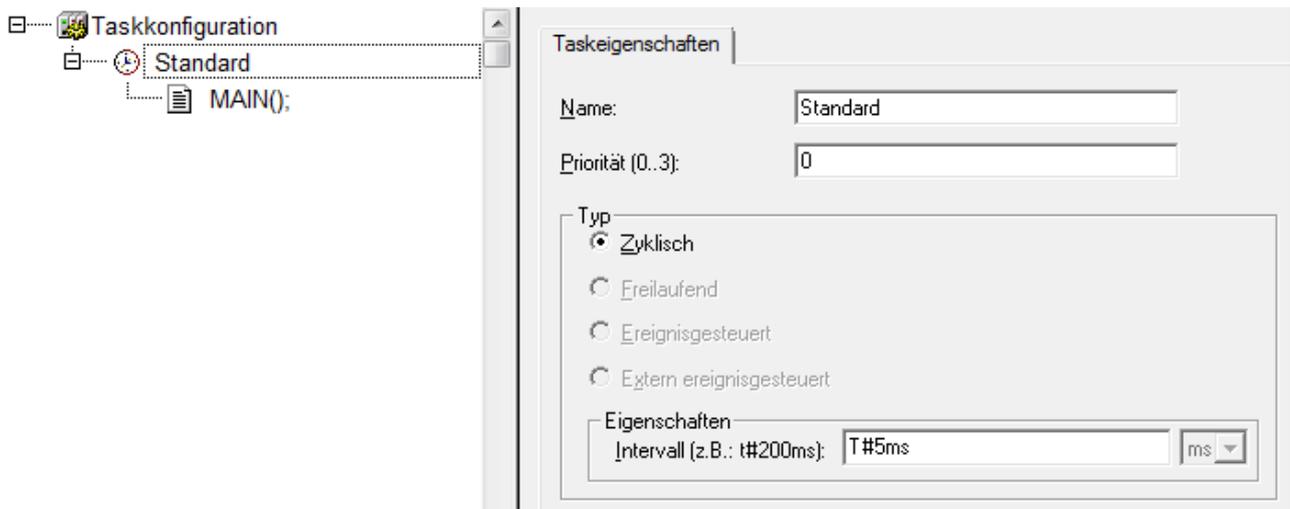
Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine [KL6771\(0\)](#) [▶ 27] und [MP_DamperLinearActuator\(\)](#) [▶ 31] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *arrKL6771_IN*, *arrKL6771_OUT* und *stDataKL6771* zu verknüpfen.



Der Eingang *SetPoint* wird in Abhängigkeit der gewählten Funktion gesetzt. Verknüpfen Sie dazu die globalen Variablen *bOpen* und *bClose* mit einer Hilfsvariable.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins [KL6771\(\)](#) [▶ 27].

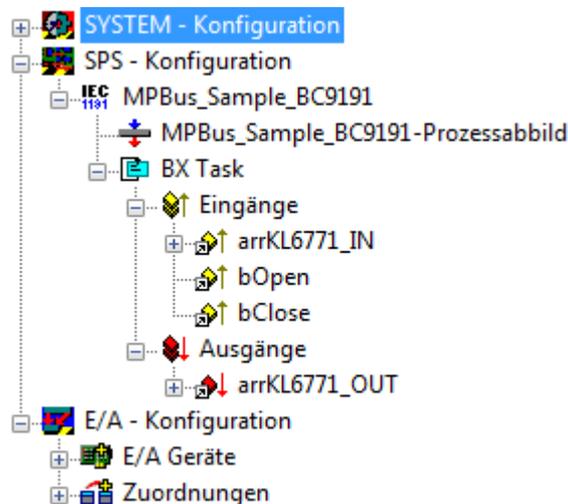


Laden Sie das Projekt als Bootprojekt auf den BC und speichern Sie es ab.

Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den BC und lassen Sie nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.

Ihr BC ist jetzt einsatzbereit.

Nach Betätigen der Taster kann der Klappenantrieb gefahren werden. Die aktuelle Position steht in der Variable *wActValue*.

6 Programmierung

Inhalt
Allgemeine Informationen [▶ 25]
Verknüpfung mit dem TwinCAT System Manager [▶ 14]

Bausteine

Bausteine	Beschreibung
KL6771 [▶ 27]	Kommunikation mit einer MP-Bus-Masterklemme KL6771

Funktionsbausteine

Bausteine	Beschreibung
KL6771 [▶ 27]	Kommunikation mit einer MP-Bus-Masterklemme KL6771 .
MP Addressing [▶ 27]	Dieser Funktionsbaustein ermöglicht das Adressieren eines MP-Bus Slaves.
MP CMV [▶ 29]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.
MP DamperLinearActuator [▶ 31]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantriebs.
MP EnergyValveV4 Configuration [▶ 32]	Dieser Funktionsbaustein dient zur Konfiguration der Energy Valve Antriebe EV..R2+.. (V4).
MP EnergyValveV4 Process [▶ 35]	Dieser Funktionsbaustein ist geeignet für die Energy Valve Antriebe EV..R2+.. (V4).
MP EPIV [▶ 37]	Dieser Funktionsbaustein dient der Ansteuerung eines druckunabhängigen Regelkugelhahns.
MP EPIV R6 [▶ 39]	Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe EP..R-R6+BAC.
MP EPIV R6 Parameter [▶ 40]	Dieser Funktionsbaustein dient zur Parametrierung von Antrieben der Baureihe EP..R-R6+BAC.
MP EV [▶ 42]	Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe P6..W..EV-BAC.
MP EV Parameter [▶ 44]	Dieser Funktionsbaustein dient zur Parametrierung von Antrieben.
MP MPX [▶ 45]	Für BELIMO Raum Sensor MS24A-R..-MPX.
MP OperatingUnit [▶ 46]	Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationCO2 [▶ 48]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationDisplay [▶ 49]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationOffsetValues [▶ 50]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationStatusIcons [▶ 51]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationTemp [▶ 52]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP OperatingUnit ConfigurationVentilation [▶ 53]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP PTH [▶ 54]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines PTH Sensors.

Bausteine	Beschreibung
MP_RoomSensor [▶ 55]	Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren.
MP_RoomSensor_Parameter [▶ 57]	Dieser Funktionsbaustein dient zur Parametrierung von Raumsensoren.
MP_Smoker [▶ 58]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung einer Brandschutzklappe.
MP_TEM_Configuration [▶ 59]	Dieser Funktionsbaustein dient zur Konfiguration der Thermal Energy Meter Typ 22PE-.. und 22PEM-.. .
MP_TEM_Process [▶ 61]	Dieser Funktionsbaustein ist geeignet für die Thermal Energy Meter Typ 22PE-.. und 22PEM-.. .
MP_UST_3 [▶ 62]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Multi IO Moduls UST3.
MP_VAV [▶ 65]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.
MP_VRU_Configuration [▶ 67]	Dieser Funktionsbaustein dient zur Konfiguration der VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC.
MP_VRU_Process [▶ 68]	Dieser Funktionsbaustein ist geeignet für die VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC.
MP_Window [▶ 70]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Fensterlüftungssystems FLS.
MPL_DamperLinearActuator [▶ 71]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantriebs.

Funktionen

Bausteine	Beschreibung
NI1000_LuS_TO_INT [▶ 72]	Diese Funktion berechnet aus einem Widerstand NI1000 L&S eine Temperatur.
NI1000_TO_INT [▶ 72]	Diese Funktion berechnet aus einem Widerstand NI1000 eine Temperatur.
NTC_TO_INT [▶ 73]	Diese Funktion berechnet aus einem Widerstand NTC eine Temperatur.
PT1000_TO_INT [▶ 73]	Diese Funktion berechnet aus einem Widerstand PT1000 eine Temperatur.

Slave Adressierung

Bausteine	Beschreibung
MP_Addressung [▶ 27]	Dieser Funktionsbaustein ermöglicht das Adressieren eines MP-Bus Slaves.

Enums

Datentypen	Beschreibung
Data_Window [▶ 74]	Lüftungsart.
E_MP_AirQualityStatus [▶ 74]	Status der gemessenen Luftqualität.
E_MP_DisplayBackground [▶ 75]	Hintergrundfarbe des Displays.
E_MP_DisplayModeButton [▶ 75]	Anzeigemodus der Tasten.
E_MP_DisplayModeHeatingCooling [▶ 75]	Anzeigemodus von Heiz- oder Kühlsymbolen.
E_MP_DisplayModelconWarning [▶ 75]	Anzeigemodus des Warnsymbols.
E_MP_DisplayModelconWindow [▶ 76]	Anzeigemodus des Fenstersymbols.

Datentypen	Beschreibung
E_MP_DisplayModeTemp [▶ 76]	Anzeigemodus Temperatur.
E_MP_DisplayModeTempUnit [▶ 76]	Anzeigemodus der Einheit für die Temperatur.
E_MP_DisplayModeVentilationStage [▶ 76]	Anzahl der auf dem Display angezeigten einstellbaren Lüftungsstufen.
E_MP_DisplayVisibility [▶ 77]	Aktivierungsstatus.
E_MP_EnabledStatus [▶ 77]	Skalierung.
E_MP_EP_R_R6_UnitSel [▶ 77]	Skalierung.
E_MP_EV_V4_Command [▶ 78]	Befehl für Service- und Testfunktionen des Antriebs.
E_MP_EV_V4_ControlMode [▶ 78]	Steuerungsmodus.
E_MP_EV_V4_DeltaTLimitation [▶ 78]	Reaktion auf ein niedriges Delta T.
E_MP_EV_V4_DeltaTManagerStatus [▶ 78]	Status vom Delta T Manager.
E_MP_EV_V4_OverrideControl [▶ 79]	Überschreiben des Sollwerts.
E_MP_EV_V4_Sensor1Type [▶ 79]	Externer Sensor an Eingang S1.
E_MP_SystemOperationMode [▶ 79]	Betriebsart des Systems.
E_MP_VRU_Application [▶ 80]	Visualisierung der vom Hersteller gewählten Anwendung.
E_MP_VRU_Command [▶ 80]	Kommandos für Service- und Testfunktionen des Antriebs.
E_MP_VRU_OverrideControl [▶ 80]	Überschreiben des Sollwerts.
E_MP_VRU_RoomPressureCascade [▶ 81]	Raumdruck-Kaskadenregelung.
E_MP_VRU_Sensor1Type [▶ 81]	Externer Sensor an Eingang S1.
E_MPBus_ControlMode [▶ 81]	Regelmodus.
E_MPBus_DeltaTLimitation [▶ 81]	Delta T (dT) Limitierung.
E_MPBus_Override [▶ 82]	Überschreiben Modus.
E_MPBus_Override_6wayMPIV [▶ 82]	Überschreiben Regelmodus.
MP_Error [▶ 82]	Fehlermeldungen.
UST3_Ex [▶ 84]	Skalierung der Spannung.
UST3_R_set [▶ 84]	Skalierung des Widerstandes.

Strukturen

Datentypen	Beschreibung
DataKL6771 [▶ 85]	Verbindet die Sende- und Empfangs-Bausteine mit dem Baustein KL6771.
MP_BUS_MPX_ERROR [▶ 85]	Fehlermeldungen der "MPX" Sensoren.
MP_Serial_Number [▶ 85]	Seriennummer des Gerätes.
St_MP_EV_V4_MalfunctionServiceInfo [▶ 86]	Störungs- und Serviceinformationen.
St_MP_VRU_ServiceInfo [▶ 86]	Störungs- und Serviceinformationen.
St_StateEV [▶ 87]	Auskunft über den Zustand vom EV.
UST3_SET [▶ 87]	Datenstruktur zur Einstellung der Skalierung und dem Einstellen der Widerstandsmessung.

6.1 Allgemeine Informationen

● Installation

i Ab TwinCAT 2.11 Build 2229 (R3 und x64 Engineering) werden die Bibliotheken "TcMPBus.lib/.lb6/.lbox" standardmäßig mitinstalliert.

● Name der Bibliothek

i Diese Bibliothek ersetzt die "TcKL6771.lib/.lb6/.lbox". Es hat sich nur der Name der Bibliothek geändert. Die Bausteine sind kompatibel.

Hardware Dokumentation im Beckhoff Information System: [KL6771 - MP-Bus-Masterklemme](#)

Weitere erforderliche Bibliotheken

Für PC-Systeme (x86) und Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib

Für Busklemmen-Controller der Serie BCxx00:

- Standard.lb6
- PlcHelperBC.lb6

Für Busklemmen-Controller der Serie BCxx50, BCxx20 und BC9191:

- Standard.lbx
- TcBaseBCxx50.lbx
- TcSystemBCxx50.lbx

Für Busklemmen-Controller der Serie BXxx00:

- Standard.lbx
- TcBaseBX.lbx
- TcSystemBX.lbx

● Speicherauslastung

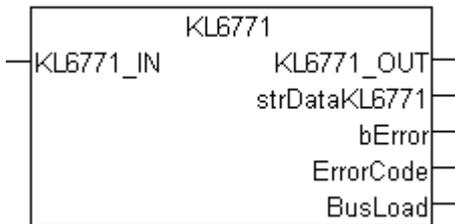
i Durch Einbinden der Bibliothek wird bereits SPS-Programmspeicher verbraucht. Abhängig vom Applikationsprogramm kann daher der verbleibende Speicher nicht ausreichend sein.

6.2 Funktionsbausteine

Bausteine	Beschreibung
KL6771 [▶ 27]	Kommunikation mit einer MP-Bus-Masterklemme KL6771.
MP Addressing [▶ 27]	Dieser Funktionsbaustein ermöglicht das Adressieren eines MP-Bus Slaves.
MP CMV [▶ 29]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.
MP DamperLinearActuator [▶ 31]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantriebs.
MP EnergyValveV4 Configuration [▶ 32]	Dieser Funktionsbaustein dient zur Konfiguration der Energy Valve Antriebe EV..R2+.. (V4).
MP EnergyValveV4 Process [▶ 35]	Dieser Funktionsbaustein ist geeignet für die Energy Valve Antriebe EV..R2+.. (V4).
MP EPIV [▶ 37]	Dieser Funktionsbaustein dient der Ansteuerung eines druckunabhängigen Regelkugelhahns.

Bausteine	Beschreibung
MP_EPIV_R6 [▶ 39]	Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe EP..R-R6+BAC.
MP_EPIV_R6_Parameter [▶ 40]	Dieser Funktionsbaustein dient zur Parametrierung von Antrieben der Baureihe EP..R-R6+BAC.
MP_EV [▶ 42]	Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe P6..W..EV-BAC.
MP_EV_Parameter [▶ 44]	Dieser Funktionsbaustein dient zur Parametrierung von Antrieben.
MP_MPX [▶ 45]	Für BELIMO Raum Sensor MS24A-R..-MPX.
MP_OperatingUnit [▶ 46]	Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationCO2 [▶ 48]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationDisplay [▶ 49]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationOffsetValues [▶ 50]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationStatusIcons [▶ 51]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationTemp [▶ 52]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_OperatingUnit_ConfigurationVentilation [▶ 53]	Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1.
MP_PTH [▶ 54]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines PTH Sensors.
MP_RoomSensor [▶ 55]	Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren.
MP_RoomSensor_Parameter [▶ 57]	Dieser Funktionsbaustein dient zur Parametrierung von Raumsensoren.
MP_Smoker [▶ 58]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung einer Brandschutzklappe.
MP_TEM_Configuration [▶ 59]	Dieser Funktionsbaustein dient zur Konfiguration der Thermal Energy Meter Typ 22PE-.. und 22PEM-..
MP_TEM_Process [▶ 61]	Dieser Funktionsbaustein ist geeignet für die Thermal Energy Meter Typ 22PE-.. und 22PEM-..
MP_UST_3 [▶ 62]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Multi IO Moduls UST3.
MP_VAV [▶ 65]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.
MP_VRU_Configuration [▶ 67]	Dieser Funktionsbaustein dient zur Konfiguration der VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC.
MP_VRU_Process [▶ 68]	Dieser Funktionsbaustein ist geeignet für die VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC.
MP_Window [▶ 70]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Fensterlüftungssystems FLS.
MPL_DamperLinearActuator [▶ 71]	Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantriebs.

6.2.1 KL6771



Dieser Funktionsbaustein übernimmt die Kommunikation mit der MP-Bus-Masterklemme KL6771. Über diesen Baustein wird die KL6771 konfiguriert und der Datenaustausch mit dem MP-Netzwerk gestartet.

Einschränkungen:

- Dieser Baustein darf nur einmal pro KL6771 aufgerufen werden!
- Er muss in derselben Task aufgerufen werden, wie die Aktor Bausteine!
- Im PLC-Projekt darf dieser Baustein nur einmal pro Zyklus aufgerufen werden!

VAR_INPUT

```
KL6771_IN : ARRAY [0..11] OF BYTE;
```

KL6771_IN: Wird mit dem Eingangsprozessabbild der MP-Bus-Masterklemme verbunden.

VAR_OUTPUT

```
KL6771_OUT : ARRAY [0..11] OF BYTE;
strDataKL6771 : DataKL6771;
bError : BOOL;
ErrorCode : MP_Error;
BusLoad : INT := -1;
```

KL6771_OUT: Wird mit dem Ausgangsprozessabbild der MP-Bus-Masterklemme verbunden.

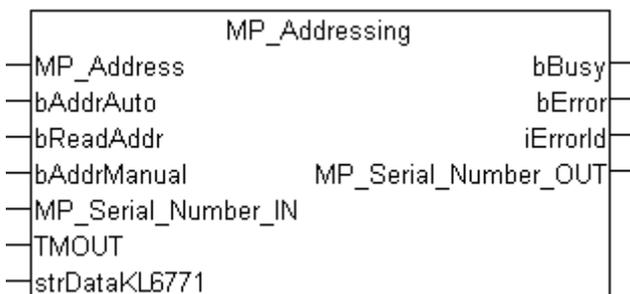
strDataKL6771: Datenstruktur, die mit den verschiedenen MP-Bus-Funktionsbausteinen verbunden wird und die Kommunikation mit dem KL6771 Baustein beinhaltet (siehe [DataKL6771 \[► 85\]](#)).

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *ErrorCode* beschrieben.

ErrorCode: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

BusLoad: Auslastung des MP-Bus in Prozent.

6.2.2 MP_Adressing



Dieser Funktionsbaustein ermöglicht das Adressieren eines MP-Bus Slaves. Des Weiteren kann über diesen Baustein auch die Seriennummer der Slaves ausgelesen werden.

Mit **MP_Address** wird entweder die Slave Adresse übergeben, die der Slave beim Adressieren bekommen soll, oder die Adresse, bei welchem Teilnehmer die Seriennummer ausgelesen werden soll. Mit steigender Flanke des **bReadAddr** wird die Seriennummer ausgelesen. Diese wird über **MP_Serial_Numer_Out**

ausgegeben. Eine positive Flanke von **bAddrAuto** adressiert einen Slave mit der Adresse **MP_Address**. Der Baustein wartet mit der eingestellten **TMOU**T Zeit auf das Senden des Slaves mit seiner Seriennummer. Das Auslösen der Übermittlung der Seriennummer ist von Slave zu Slave unterschiedlich. Bitte entnehmen Sie der Dokumentation der MP-Bus Teilnehmer, wie die Seriennummer verschickt werden kann (meist ist ein Schalter auf dem Antrieb, der dies durch Drücken auslöst). Während der **TMOU**T Zeit werden keine Telegramme zu den Slaves versendet.

Eine positive Flanke von **bAddrManual** löst eine manuelle Adressierung aus. Dafür muss die Seriennummer des Antriebs in **MP_Serial_Number_In** hinterlegt sein. Die Seriennummer der Slaves sind einem Aufkleber am Antrieb zu entnehmen.

Beispiel: 00234-00016-002-031 In der Variable **MP_Serial_Numer_IN** ist folgendes einzutragen:

YearAndWeek = 234

DayAndNumber = 16

DeviceFamily = 2

TestStation = 31

FamilySuffix = wird nicht ausgewertet und ist daher nicht zu berücksichtigen.

bBusy wird gesetzt solange der Baustein aktiv ist. Ein Fehler wird mit **bError** angezeigt und die Fehlernummer kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bAddrAuto      : BOOL;
bReadAddr      : BOOL;
bAddrManual    : BOOL;
MP_Serial_Number_IN : MP_Serial_Number;
TMOU           : TIME := t#30s;
strDataKL6771  : DataKL6771;
```

MP_Address: Die MP Adresse die für die Adressierung verwendet werden soll oder für das Lesen der Seriennummer. Gültige Werte (1...8).

bAddrAuto: Positive Flanke startet den Baustein. Der Baustein unterbindet alle anderen MP-Bus Kommunikationen und wartet mit der eingestellten **TMOU**T Zeit, bis ein MP-Bus Slave seine Seriennummer zum Beispiel durch Drücken eines Schalters übermittelt hat. Es wird dann die MP-Bus Adresse an den Slave übertragen, die in der Variable **MP_Address** konfiguriert wurde.

bReadAddr: Positive Flanke startet den Baustein. Der Baustein liest die Seriennummer des MP-Bus Slaves mit der Adresse **MP_Address** aus.

bAddrManual: Positive Flanke startet den Baustein. Der Baustein adressiert den Slave mit der Seriennummen **MP_Serial_Number_IN**. Die Soll-Adresse des Slave wird mit **MP_Address** konfiguriert.

MP_Serial_Number_IN: Seriennummer des Gerätes (siehe [MP_Serial_Number](#) [▶ 85]).

TMOU: Zeit nach der die automatische Adressierung abgebrochen wird.

strDataKL6771: Datenstruktur die mit dem [KL6771\(0\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
MP_Serial_Number_OUT : MP_Serial_Number;
```

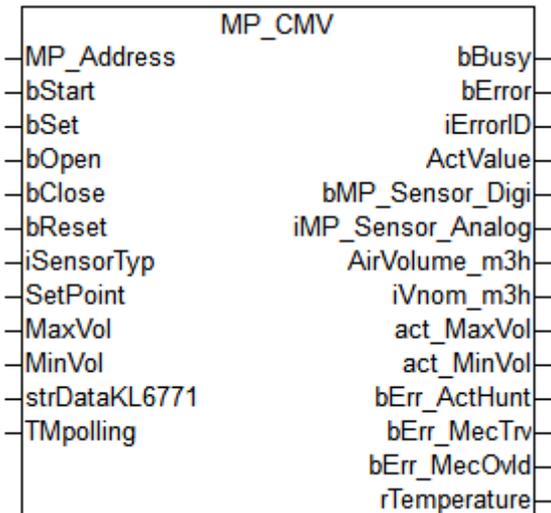
bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable **iErrorId** beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [▶ 82]). Gleichzeitig wird *bError* TRUE.

MP_Serial_Number_OUT: Die Seriennummer vom adressierten oder angefragten Slave (siehe [MP_Serial_Number](#) [▶ 85]).

6.2.3 MP_CMV



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
bSet            : BOOL;
bOpen          : BOOL;
bClose         : BOOL;
bReset         : BOOL;
iSensorTyp     : INT;
SetPoint       : USINT;
MaxVol         : WORD;
MinVol         : WORD;
strDataKL6771  : DataKL6771;
Tmpolling      : TIME:= t#10S;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

bSet: Positive Flanke schreibt die Daten *MaxVol* und *MinVol* zum Antrieb.

bOpen: Positive Flanke öffnet die Klappen des Antriebs, eine negative Flanke löscht die Zwangsbelüftung.

bClose: Positive Flanke schließt die Klappen des Antriebs, eine negative Flanke löscht die Zwangsverschließung.

bReset: Positive Flanke setzt die Fehlermeldungen des Antriebs zurück.

iSensorTyp: 0: kein Sensor angeschlossen, 1: digitaler Sensor angeschlossen, 2: analoger Sensor angeschlossen (0...35 V), 3...6: Ausgabe eines Widerstandes in Ohm (3..5 gilt für PT1000, NI1000 und NI1000LuS; 6 gilt für NTC). Zum Umrechnen in eine Temperatur sind die entsprechenden Konvertierungsfunktionen zu verwenden.

SetPoint: Soll-Volumenstrom (0...100 %).

MaxVol: Maximaler-Volumenstrom (30...100 %).

MinVol: Minimaler-Volumenstrom (0...100 %).

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[85](#)]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
ActValue       : WORD;
bMP_Sensor_Digi : BOOL;
iMP_Sensor_Analog : INT;
AirVolume_m3h  : WORD;
iVnom_m3h      : INT;
act_MaxVol     : INT;
act_MinVol     : INT;
bErr_ActHunt   : BOOL;
bErr_MecTrv    : BOOL;
bErr_MecOvld   : BOOL;
rTemperature   : LREAL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) [[82](#)]). Gleichzeitig wird *bError* TRUE.

ActValue: Enthält die aktuelle Position (0...100 %) des Antriebs.

bMP_Sensor_Digi: Ist ein digitaler Sensor angeschlossen, wird der Zustand über diese Variable angezeigt. *iSensorTyp* muss "1" sein.

iMP_Sensor_Analog: Ist ein analoger Sensor angeschlossen, wird der Wert über diese Variable angezeigt. *iSensorTyp* muss "2...6" sein.

AirVolume_m3h: Ausgabe des Volumenstroms in m³/h.

iVnom_m3h: Nominal Luftvolumenstrom in m³/h. Dieser Ausgang ist ab Version 1.12.0 verfügbar. VAV wird gelesen und muss > 0 sein. Wenn gleich 0, dann ist die Berechnung von *AirVolume_m3h* nicht korrekt.

act_MaxVol: Maximal eingestellter Volumenstrom in %.

act_MinVol: Minimal eingestellter Volumenstrom in %.

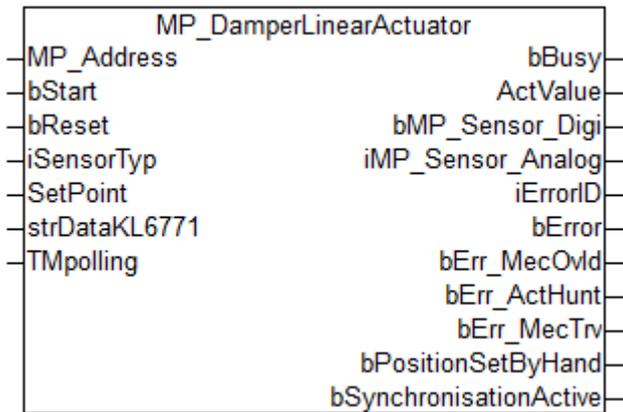
bErr_ActHunt: Antriebsfehler "Regelschwingung", Antrieb pendelt hin und her.

bErr_MecTrv: Antriebsfehler "Stellwinkel überschritten", Antriebsdrehwinkel gegenüber Adaption um mehr als 10° überschritten.

bErr_MecOvld: Antriebsfehler "Überlast", Sollposition konnte nicht erreicht werden.

rTemperature: Temperatur im Kanal in °C.

6.2.4 MP_DamperLinearActuator



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantrieb.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMpolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 sec einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorID** ausgelesen werden.

Positive Flanke von **bReset** löscht anstehende Fehlermeldungen des Antriebs. Damit werden Fehler zurückgesetzt, die die Ausgangsvariablen **bErr_MecOvld**, **bErr_ActHunt** und **bErrMecTrv** betreffen. Sollte der Fehler noch anstehen, werden diese Fehlerbits vom Antrieb erneut gesetzt.

Mit **SetPoint** wird die Stellung der Klappe von 0...100 % eingestellt. Die aktuelle Stellung des Antriebs wird über **ActValue** ausgelesen.

Sollte am Antrieb ein Sensor angeschlossen sein, ist über **iSensorTyp** anzugeben welcher Sensor dies ist. Ist kein Sensor angeschlossen, so ist der Wert "0" einzutragen oder die Variable wird offengelassen. Ein digitaler Sensor ist mit "1" zu parametrieren. Der Zustand des Sensors kann dann über **bMP_Sensor_Digi** abgefragt werden. Analoge Sensoren "2...6" werden in der Variable **iMP_Sensor_Analog** ausgegeben.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
bReset          : BOOL;
iSensorTyp      : INT;
SetPoint        : USINT;
strDataKL6771   : DataKL6771;
Tm polling      : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMpolling* der Baustein zyklisch aktiv.

bReset: Positive Flanke setzt die Fehlermeldungen des Antriebs zurück.

iSensorTyp: 0: kein Sensor angeschlossen, 1: digitaler Sensor angeschlossen, 2: analoger Sensor angeschlossen (0...35 V), 3...6: Ausgabe eines Widerstandes in Ohm (3...5 gilt für PT1000, NI1000 und NI1000LuS; 6 gilt für NTC). Zum Umrechnen in eine Temperatur sind die entsprechenden Konvertierungsfunktionen zu verwenden.

SetPoint: 0...100 % Soll-Klappenstellung des Antriebs.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [27] Baustein verbunden sein muss (siehe [DataKL6771](#) [85]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy          : BOOL;
ActValue       : WORD;
bMP_Sensor_Digi : BOOL;
iMP_Sensor_Analog : INT;
iErrorID       : MP_Error;
bError         : BOOL;
bErr_MecOvld   : BOOL;
bErr_ActHunt   : BOOL;
bErr_MecTrv    : BOOL;
bPositionSetByHand : BOOL;
bSynchronisationActive : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

ActValue: Enthält die aktuelle Position (0...100 %) des Antriebs.

bMP_Sensor_Digi: Ist ein digitaler Sensor angeschlossen, wird der Zustand über diese Variable angezeigt. *iSensorTyp* muss "1" sein.

iMP_Sensor_Analog: Ist ein analoger Sensor angeschlossen, wird der Wert über diese Variable angezeigt. *iSensorTyp* muss "2...6" sein.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe MP_Error [▶ 82]). Gleichzeitig wird *bError* TRUE.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

bErr_MecOvld: Antriebsfehler "Überlast", Sollposition konnte nicht erreicht werden.

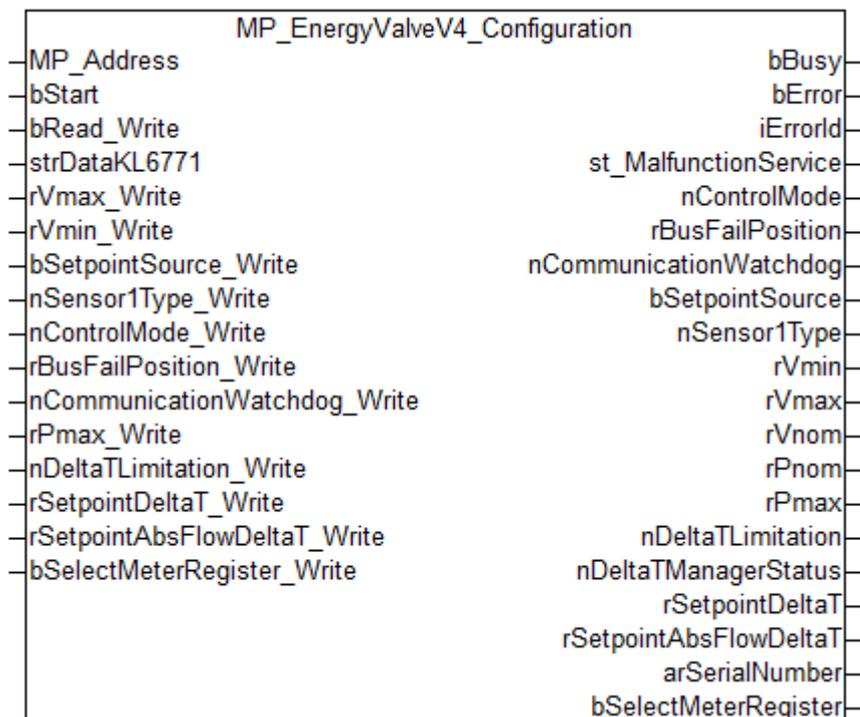
bErr_ActHunt: Antriebsfehler "Regelschwingung", Antrieb pendelt hin und her.

bErr_MecTrv: Antriebsfehler "Stellwinkel überschritten", Antriebsdrehwinkel gegenüber Adaption um mehr als 10° überschritten.

bPositionSetByHand: Die Position des Antriebs wurde von Hand verändert.

bSynchronisationActive: Der Antrieb synchronisiert sich.

6.2.5 MP_EnergyValveV4_Configuration



Dieser Funktionsbaustein dient zur Konfiguration der Energy Valve Antriebe EV..R2+.. (V4). Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
bRead_Write    : BOOL;
strDataKL6771  : DataKL6771;
rVmax_Write    : LREAL := 100;
rVmin_Write    : LREAL;
bSetpointSource_Write : BOOL := TRUE;
nSensor1Type_Write : E_MP_EV_V4_Sensor1Type;
nControlMode_Write : E_MP_EV_V4_ControlMode := 1;
rBusFailPosition_Write : LREAL;
nCommunicationWatchdog_Write : UINT;
rPmax_Write    : LREAL := 100;
nDeltaTLimitation_Write : E_MP_EV_V4_DeltaTLimitation;
rSetpointDeltaT_Write : LREAL := 1.0;
rSetpointAbsFlowDeltaT_Write : LREAL;
bSelectMeterRegister_Write : BOOL;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

bRead_Write: Wenn FALSE dann nur LESEN, bei TRUE LESEN und SCHREIBEN.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

rVmax_Write: Max-Sollwert in % (25...100). Standard 100 %. Bezieht sich auf Vnom und wird berücksichtigt, wenn Steuerungsmodus = Durchflusssteuerung.

rVmin_Write: Min-Sollwert in % (0...Vmax). Vmin muss kleiner sein als Vmax.

bSetpointSource_Write: TRUE = Bus; FALSE = Analog.

nSensor1Type_Write: Sensor 1 Typ (siehe [E_MP_EV_V4_Sensor1Type](#) [► 79]).

nControlMode_Write: Steuerungsmodus (siehe [E_MP_EV_V4_ControlMode](#) [► 78]).

rBusFailPosition_Write: Bus-Ausfallposition in % (0...100). Nicht funktionell (reserviert für zukünftige Erweiterungen).

nCommunicationWatchdog_Write: Kommunikationsüberwachung in s (0...3600). Nicht funktionell (reserviert für zukünftige Erweiterungen).

rPmax_Write: Max-Sollwert in % (5...100). Standard 100 %. Bezieht sich auf Pnom und wird berücksichtigt, wenn Steuerungsmodus = Leistungssteuerung.

nDeltaTLimitation_Write: Legt fest, ob das Gerät auf ein niedriges delta T reagiert (siehe [E_MP_EV_V4_DeltaTLimitation](#) [► 78]).

rSetpointDeltaT_Write: Sollwert Delta T in K (0...60). Standard 1.0K. Wird berücksichtigt, wenn die Delta T Limitierung aktiv ist (nicht deaktiviert).

rSetpointAbsFlowDeltaT_Write: Sollwert Abs Durchfluss Delta T in l/s (0...100.000). Wird berücksichtigt, wenn die Delta T Limitierung auf "Delta T Manager skaliert" gesetzt ist.

bSelectMeterRegister_Write: FALSE = zertifiziertes Zählwerk; TRUE = Lebensdauer Zählwerk.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
st_MalfunctionService : St_MP_EV_V4_MalfunctionServiceInfo;
nControlMode   : E_MP_EV_V4_ControlMode;
rBusFailPosition : LREAL;
nCommunicationWatchdog : UINT;
bSetpointSource : BOOL;
nSensor1Type   : E_MP_EV_V4_Sensor1Type;
rVmin          : LREAL;
rVmax          : LREAL;
rVnom          : LREAL;
rPnom          : LREAL;
rPmax          : LREAL;
nDeltaTLimitation : E_MP_EV_V4_DeltaTLimitation;
nDeltaTManagerStatus : E_MP_EV_V4_DeltaTManagerStatus;
rSetpointDeltaT : LREAL;
rSetpointAbsFlowDeltaT : LREAL;
arSerialNumber : ARRAY[0..1] OF DWORD;
bSelectMeterRegister : BOOL;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

st_MalfunctionService: Störungs- und Serviceinformationen (siehe [St_MP_EV_V4_MalfunctionServiceInfo \[► 86\]](#)).

nControlMode: Steuerungsmodus (siehe [E_MP_EV_V4_ControlMode \[► 78\]](#)).

rBusFailPosition: Bus-Ausfallposition in % (0...100). Nicht funktionell (reserviert für zukünftige Erweiterungen).

nCommunicationWatchdog: Kommunikationsüberwachung in Sek (0...3600). Nicht funktionell (reserviert für zukünftige Erweiterungen).

bSetpointSource: TRUE = Bus; FALSE = Analog.

nSensor1Type: Sensor 1 Typ (siehe [E_MP_EV_V4_Sensor1Type \[► 79\]](#)).

rVmin: Min-Sollwert in % (0...Vmax). Bezieht sich auf Vnom und wird berücksichtigt, wenn Steuerungsmodus = Durchflusssteuerung.

rVmax: Max-Sollwert in % (25...100). Bezieht sich auf Vnom und wird berücksichtigt, wenn Steuerungsmodus = Durchflusssteuerung.

rVnom: Nennvolumenstrom in l/s (0...100).

rPnom: Nennleistung in kW (0...21.5).

rPmax: Max-Sollwert in % (5...100). Bezieht sich auf Pnom und wird berücksichtigt, wenn Steuerungsmodus = Leistungssteuerung.

nDeltaTLimitation: Gibt an, ob das Gerät auf ein niedriges delta T reagiert (siehe [E_MP_EV_V4_DeltaTLimitation \[► 78\]](#)).

nDeltaTManagerStatus: Gibt den Status vom DeltaT Manager an (siehe [E_MP_EV_V4_DeltaTManagerStatus \[► 78\]](#)).

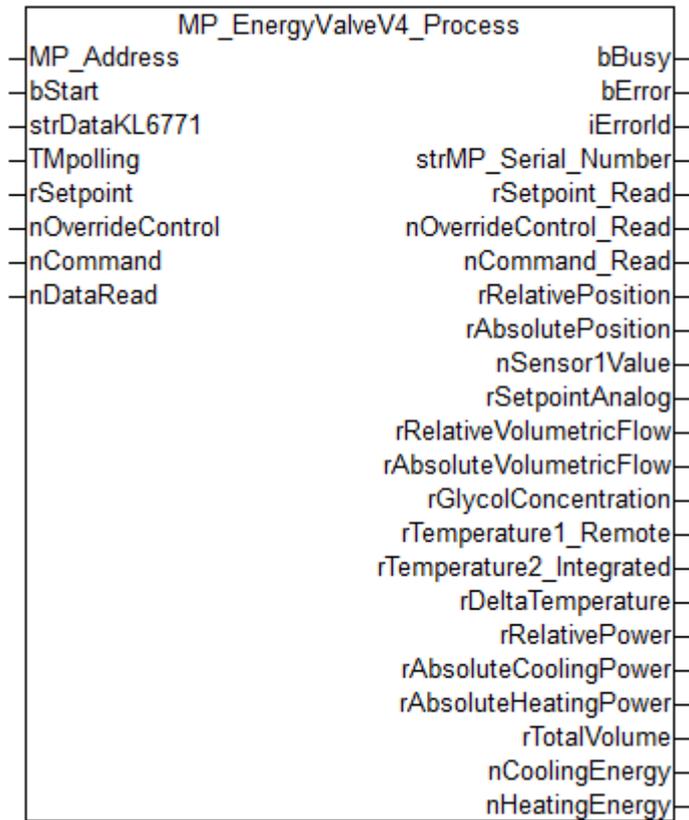
rSetpointDeltaT: Sollwert DeltaT in K (0...60). Wird berücksichtigt, wenn DeltaT Limitierung aktiv (nicht deaktiviert).

rSetpointAbsFlowDeltaT: Sollwert Abs Durchfluss DeltaT in l/s (0...100.000). Wird berücksichtigt, wenn die DeltaT Limitierung auf DeltaT-Manager-Skaliert gesetzt ist.

arSerialNumber: Seriennummer des Geräts.

bSelectMeterRegister: FALSE = zertifiziertes Zählwerk; TRUE = Lebensdauer Zählwerk.

6.2.6 MP_EnergyValveV4_Process



Dieser Funktionsbaustein ist geeignet für die Energy Valve Antriebe EV..R2+.. (V4). Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMpolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
TMpolling       : TIME:= t#10s;
rSetpoint       : LREAL;
nOverrideControl : E_MP_EV_V4_OverrideControl;
nCommand        : E_MP_EV_V4_Command;
nDataRead       : WORD := 16#FFFF;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMpolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMpolling: Zeit, mit der der Baustein den Antrieb ansprechen soll. Default 10 s. Minimale Zeit 1 s.

rSetpoint: Sollwert in % (0...100).

nOverrideControl: Überschreiben des Sollwerts mit definierten Werten (siehe [E_MP_EV_V4 OverrideControl](#) [[▶ 78](#)]).

nCommand: Befehl für Service- und Testfunktionen des Antriebs (siehe [E_MP_EV_V4 Command](#) [[▶ 78](#)]).

nDataRead: 0xFFFF - Lese alle Daten; Bit 0 – Lese relative Position; Bit 1 – Lese absolute Position; Bit 2 – Lese Wert Sensor 1; Bit 3 – Lese analogen Sollwert; Bit 4 – Lese relativen Volumenstrom; Bit 5 – Lese absoluten Volumenstrom; Bit 6 – Lese Glykol Konzentration; Bit 7 – Lese Temperatur1 remote; Bit 8 – Lese Temperature2 integriert; Bit 9 – Lese Temperatur Delta; Bit 10 – Lese relative Leistung; Bit 11 – Lese absolute Kühlleistung; Bit 12 – Lese absolute Heizleistung; Bit 13 - Gesamtvolumen; Bit 14 - Kühlenergie; Bit 15 - Heizenergie

VAR_OUTPUT

```

bBusy           : BOOL;
bError          : BOOL;
iErrorId       : MP_Error;
strMP_Serial_Number : MP_Serial_Number;
rSetpoint_Read  : LREAL;
nOverrideControl_Read : E_MP_EV_V4_OverrideControl;
nCommand_Read  : E_MP_EV_V4_Command;
rRelativePosition : LREAL;
rAbsolutePosition : LREAL;
nSensor1Value   : DINT;
rSetpointAnalog : LREAL;
rRelativeVolumetricFlow : LREAL;
rAbsoluteVolumetricFlow : LREAL;
rGlycolConcentration : LREAL;
rTemperature1_Remote : LREAL;
rTemperature2_Integrated : LREAL;
rDeltaTemperature : LREAL;
rRelativePower  : LREAL;
rAbsoluteCoolingPower : LREAL;
rAbsoluteHeatingPower : LREAL;
rTotalVolume    : LREAL;
nCoolingEnergy  : DINT;
nHeatingEnergy  : DINT;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [[▶ 82](#)]). Gleichzeitig wird *bError* TRUE.

strMP_Serial_Number: Struktur für die Seriennummer (siehe [MP_Serial_Number](#) [[▶ 85](#)]).

rSetpoint_Read: Sollwert in % (0...100).

nOverrideControl_Read: Überschriebener Sollwert (siehe [E_MP_EV_V4 OverrideControl](#) [[▶ 79](#)]).

nCommand_Read: Befehl (siehe [E_MP_EV_V4 Command](#) [[▶ 78](#)]).

rRelativePosition: Relative Position in % (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.0 = TRUE](#)).

rAbsolutePosition: Absolute Position in ° (0...96). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.1 = TRUE](#)).

nSensor1Value: Sensor 1 Wert in mV/Ohm (0...65535). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.2 = TRUE](#)).

rSetpointAnalog: Analoger Sollwert in % (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.3 = TRUE](#)).

rRelativeVolumetricFlow: Relativer Volumenstrom in % (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.4 = TRUE](#)).

rAbsoluteVolumetricFlow: Absoluter Volumenstrom in l/s (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe [VAR_INPUT nDataRead.5 = TRUE](#)).

rGlycolConcentration: Glykol Konzentration in % (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.6 = TRUE).

rTemperature1_Remote: Temperatur1 remote in °C (-20...12). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.7 = TRUE).

rTemperature2_Integrated: Temperature2 integriert in °C (-20...12). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.8 = TRUE).

rDeltaTemperature: Temperature Delta in K (0...14). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.9 = TRUE).

rRelativePower: Relative Leistung in % (0...100). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.10 = TRUE).

rAbsoluteCoolingPower: Absolute Kühlleistung in kW (0...21.5). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.11 = TRUE).

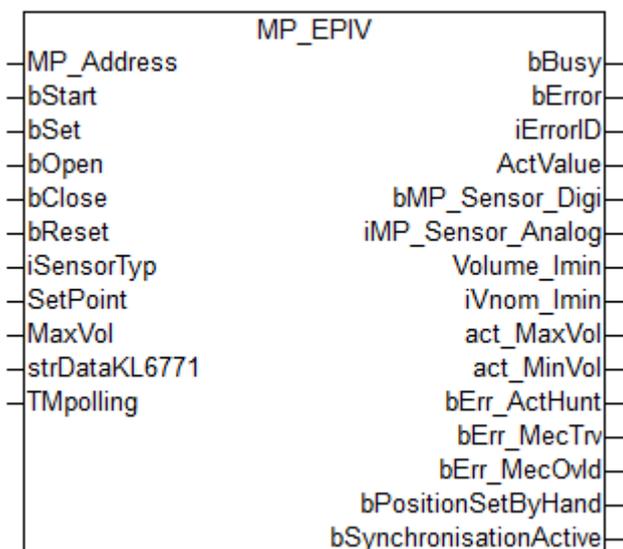
rAbsoluteHeatingPower: Absolute Heizleistung in kW (0...21.5). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.12 = TRUE).

rTotalVolume: Gesamtvolumen in m³ (0...214748.36). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.13 = TRUE).

nCoolingEnergy: Kühlenergie in kWh (0...21474836). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.14 = TRUE).

nHeatingEnergy: Heizenergie in kWh (0...21474836). Ein Wert von -1 bedeutet, dass die Daten deaktiviert sind (siehe VAR_INPUT nDataRead.15 = TRUE).

6.2.7 MP_EPIV



Dieser Funktionsbaustein dient der Ansteuerung eines druckunabhängigen Regelkugelhahns. Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
bSet            : BOOL;
```

```
bOpen      : BOOL;
bClose     : BOOL;
bReset     : BOOL;
iSensorTyp : INT;
SetPoint   : USINT;
MaxVol     : WORD;
strDataKL6771 : DataKL6771;
TMPolling  : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

bSet: Positive Flanke schreibt die Daten von *MaxVol* zum Antrieb.

bOpen: Positive Flanke öffnet die Klappen des Antriebs, eine negative Flanke löscht die Zwangsbelüftung.

bClose: Positive Flanke schließt die Klappen des Antriebs, eine negative Flanke löscht die Zwangsverschließung.

bReset: Positive Flanke setzt die Fehlermeldungen des Antriebs zurück.

iSensorTyp: 0: kein Sensor angeschlossen, 1: digitaler Sensor angeschlossen, 2: analoger Sensor angeschlossen (0...35 V), 3...6: Ausgabe eines Widerstandes in Ohm (3...5 gilt für PT1000, NI1000 und NI1000LuS; 6 gilt für NTC). Zum Umrechnen in eine Temperatur sind die entsprechenden Konvertierungsfunktionen zu verwenden.

SetPoint: Soll-Volumenstrom (0...100 %).

MaxVol: Maximaler-Volumenstrom (30...100 %).

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
iErrorId   : MP_Error;
ActValue   : WORD;
bMP_Sensor_Digi : BOOL;
iMP_Sensor_Analog : INT;
Volume_lmin : WORD;
iVnom_lmin  : INT;
act_MaxVol  : INT;
act_MinVol  : INT;
bErr_ActHunt : BOOL;
bErr_MecTrv : BOOL;
bErr_MecOvld : BOOL;
bPositionSetByHand : BOOL;
bSynchronisationActive : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [▶ 82]). Gleichzeitig wird *bError* TRUE.

ActValue: Enthält die aktuelle Position (0...100 %) des Antriebs.

bMP_Sensor_Digi: Ist ein digitaler Sensor angeschlossen, wird der Zustand über diese Variable angezeigt. *iSensorTyp* muss 1 sein.

iMP_Sensor_Analog: Ist ein analoger Sensor angeschlossen, wird der Wert über diese Variable angezeigt. *iSensorTyp* muss 2...6 sein.

Volume_lmin: Ausgabe vom Volumenstrom in lmin.

iVnom_Imin: Nominal Luftvolumenstrom in Imin. Dieser Ausgang ist ab Version 1.12.0 verfügbar.

act_MaxVol: Maximal eingestellter Volumenstrom in %.

act_MinVol: Minimal eingestellter Volumenstrom in %.

bErr_ActHunt: Antriebsfehler "Regelschwingung": Antrieb pendelt hin und her.

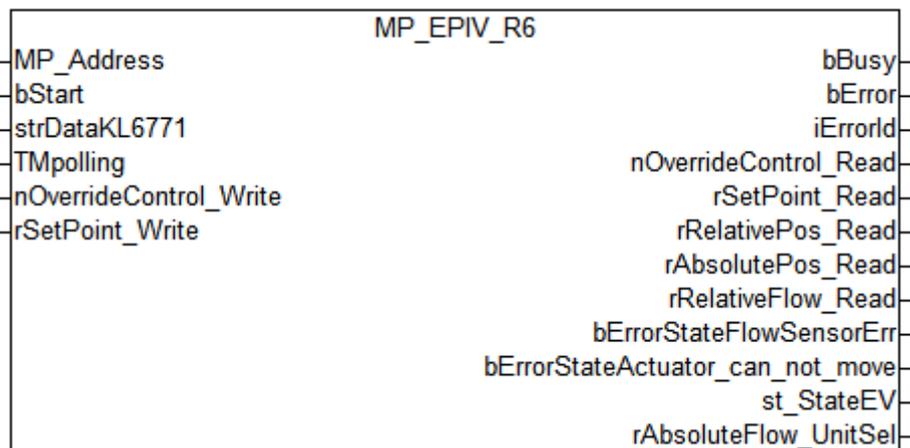
bErr_MecTrv: Antriebsfehler "Stellwinkel überschritten": Antriebsdrehwinkel gegenüber Adaption um mehr als 10° überschritten.

bErr_MecOvld: Antriebsfehler "Überlast": Sollposition konnte nicht erreicht werden.

bPositionSetByHand: Die Position des Antriebs wurde manuell geändert.

bSynchronisationActive: Der Antrieb synchronisiert sich.

6.2.8 MP_EPIV_R6



Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe EP..R-R6+BAC.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMpolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
TMpolling       : TIME := t#10s;
nOverrideControl_Write : E_MPBus_Override_6wayMPIV := MPBus_6wayMPIV_None;
rSetPoint_Write : LREAL;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMpolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

nOverrideControl_Write: Der relative Sollwert wird im Überschreiben-Steuermodus ignoriert (siehe [E_MPBus_Override_6wayMPIV](#) [▶ 82]). Wird der Befehl nicht innerhalb von 120min wiederholt, deaktiviert sich das Überschreiben.

rSetPoint_Write: 0...100 %.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
nOverrideControl_Read : E_MPBus_Override_6wayMPIV;
rSetPoint_Read : LREAL;
rRelativePos_Read : LREAL;
rAbsolutePos_Read : LREAL;
rRelativeFlow_Read : LREAL;
bErrorStateFlowSensorErr : BOOL;
bErrorStateActuator_can_not_move : BOOL;
st_StateEV     : st_StateEV;
rAbsoluteFlow_UnitSel : LREAL;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

nOverrideControl_Read: Aktueller Überschreiben-Steuerungsmodus (siehe [E_MPBus_Override_6wayMPIV \[► 82\]](#)).

rSetPoint_Read: Sollwert in % (0...100 %).

rRelativePos_Read: Relative Position in % (0...100 %).

rAbsolutePos_Read: Absolute Position in °.

rRelativeFlow_Read: Relativer Durchfluss in % (0...100 %).

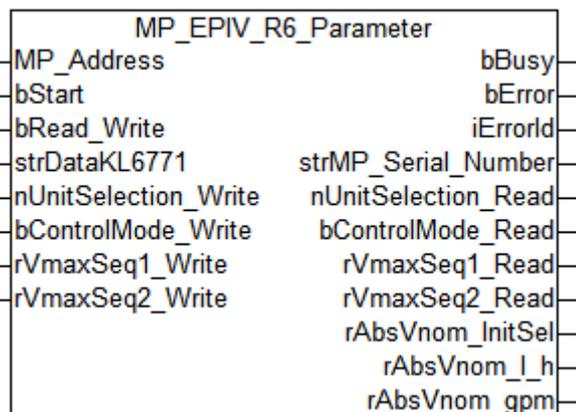
bErrorStateFlowSensorErr: Durchfluss Sensor hat einen Fehler .

bErrorStateActuator_can_not_move: Antrieb kann sich nicht bewegen.

st_StateEV: Nur Geräte ab 27.03.2014 (siehe [St_StateEV \[► 87\]](#)).

rAbsoluteFlow_UnitSel: Absoluter Durchfluss in UnitSel (0...4294967295).

6.2.9 MP_EPIV_R6_Parameter



Dieser Funktionsbaustein dient zur Parametrierung von Antrieben der Baureihe EP..R-R6+BAC.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
bRead_Write    : BOOL;
strDataKL6771  : DataKL6771;
nUnitSelection_Write : E_MP_EP_R_R6_UnitSel;
bControlMode_Write : BOOL;
rVmaxSeq1_Write : LREAL;
rVmaxSeq2_Write : LREAL;

```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

bRead_Write: Wenn FALSE dann nur LESEN, bei TRUE LESEN und SCHREIBEN.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

nUnitSelection_Write: Skalierung für *rAbsoluteFlow_UnitSel* (siehe [E_MP_EP_R_R6_UnitSel](#) [► 77]).

bControlMode_Write: FALSE: positionsgesteuert, TRUE: durchflussgesteuert.

rVmaxSeq1_Write: 0...100 %.

rVmaxSeq2_Write: 0...100 %.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
strMP_Serial_Number : MP_Serial_Number;
nUnitSelection_Read : E_MP_EP_R_R6_UnitSel;
bControlMode_Read : BOOL;
rVmaxSeq1_Read : LREAL;
rVmaxSeq2_Read : LREAL;
rAbsVnom_InitSel : LREAL;
rAbsVnom_l_h   : LREAL;
rAbsVnom_gpm   : LREAL;

```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) [► 82]). Gleichzeitig wird *bError* TRUE.

strMP_Serial_Number: Struktur für die Seriennummer (siehe [MP_Serial_Number](#) [► 85]).

nUnitSelection_Read: Setzen der Skalierung (siehe [E_MP_EP_R_R6_UnitSel](#)).

bControlMode_Read: FALSE: positionsgesteuert, TRUE: durchflussgesteuert.

rVmaxSeq1_Read: Maximale Sequenzgeschwindigkeit 1 in % (0...100 %).

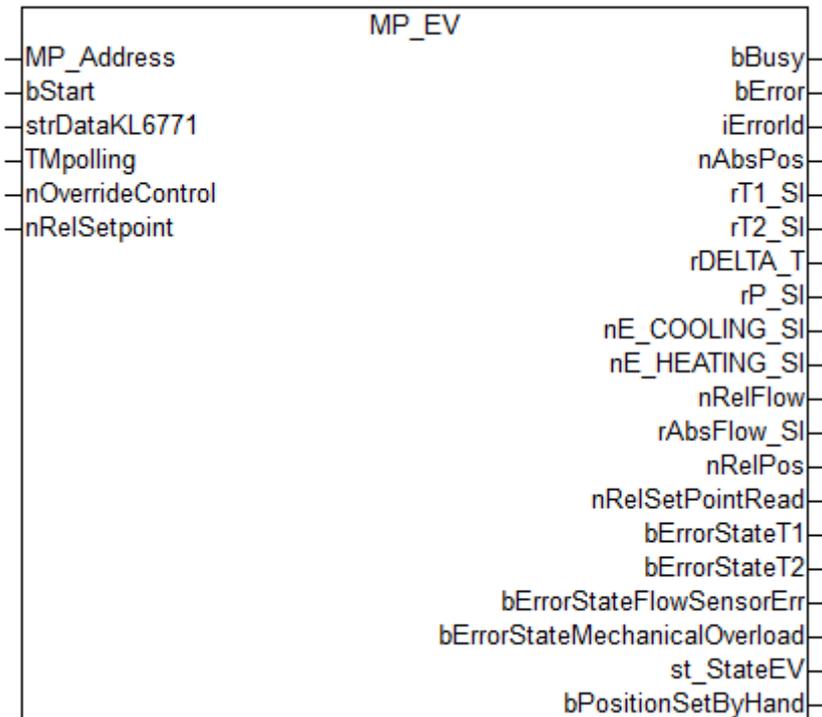
rVmaxSeq2_Read: Maximale Sequenzgeschwindigkeit 2 in % (0...100 %).

rAbsVnom_InitSel: Volumen nominal (siehe *rAbsoluteFlow_UnitSel*).

rAbsVnom_l_h: Volumen nominal in l/h (0...4294967295).

rAbsVnom_gpm: Volumen nominal in gpm (0...4294967295).

6.2.10 MP_EV



Dieser Funktionsbaustein dient der Ansteuerung eines Regelkugelhahns der Baureihe P6..W..EV-BAC. Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMpolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771  : DataKL6771;
TMpolling      : TIME := t#10s;
nOverrideControl : E_MPBus_Override := MPBus_Override_Auto;
nRelSetpoint   : INT;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMpolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(0\)](#) ([▶ 271](#)) Baustein verbunden sein muss (siehe [DataKL6771](#) ([▶ 85](#))).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

nOverrideControl: Der relative Sollwert *nRelSetpoint* wird im Überschreiben Modus ignoriert (siehe [E_MPBus_Override](#) ([▶ 82](#))). Wird der Befehl nicht innerhalb von 120 min wiederholt, deaktiviert sich das Überschreiben.

nRelSetpoint: Der Sollwert wird entweder als Positions-Sollwert oder als Vorlauf-Sollwert interpretiert.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
nAbsPos        : INT;
```

```

rT1_SI          : REAL;
rT2_SI          : REAL;
rDELTA_T       : REAL;
rP_SI          : REAL;
nE_COOLING_SI  : DINT;
nE_HEATING_SI  : DINT;
nRelFlow       : INT;
rAbsFlow_SI    : REAL;
nRelPos        : INT;
nRelSetPointRead : INT;
bErrorStateT1  : BOOL;
bErrorStateT2  : BOOL;
bErrorStateFlowSensorErr : BOOL;
bErrorStateMechanicalOverload : BOOL;
st_StateEV     : St_StateEV;
bPositionSetByHand : BOOL;

```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

nAbsPos: Absolute Position in °.

rT1_SI: Temperatur 1 (remote) in °C.

rT2_SI: Temperatur 2 (embedded) in °C.

rDELTA_T: Delta Temperature in °C.

rP_SI: Energie in kW.

nE_COOLING_SI: Kühlenergie in kWh.

nE_HEATING_SI: Heizenergie in kWh.

nRelFlow: Relativer Durchfluss in %.

rAbsFlow_SI: Absoluter Durchfluss in l/min.

nRelPos: Relative Position in %.

nRelSetPointRead: Der Sollwert wird entweder als Positions-Sollwert oder als Vorlauf-Sollwert (bezogen auf Vmax) interpretiert in %.

bErrorStateT1: Fehler Temperatur Sensor T1.

bErrorStateT2: Fehler Temperatur Sensor T2.

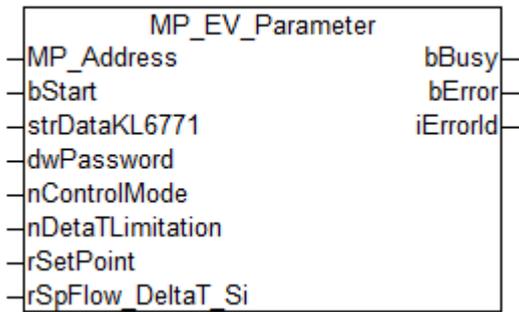
bErrorStateFlowSensorErr: Durchfluss Sensor hat einen Fehler.

bErrorStateMechanicalOverload: Mechanische Überlast erkannt.

st_StateEV: Nur Geräte ab 27.03.2014 (siehe [St_StateEV \[► 87\]](#)).

bPositionSetByHand: Die Position des Antriebs wurde manuell geändert.

6.2.11 MP_EV_Parameter



Dieser Funktionsbaustein dient zur Parametrierung von Antrieben.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771  : DataKL6771;
dwPassword     : DWORD;
nControlMode   : E_MPBus_ControlMode := MPBus_ControlMode_Disable;
nDeltaLimitation : E_MPBus_DeltaTLimitation := MPBus_DeltaTLimitation_Disable;
rSetPoint      : REAL := 0.0;
rSpFlow_DeltaT_Si : REAL := 0.0;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[85](#)]).

dwPassword: Das Passwort bei den Antrieben ist in der Regel 0x0000.

nControlMode: Legt den Regelmodus fest (siehe [E_MPBus_ControlMode](#) [[81](#)]).

nDeltaLimitation: dT Limitierung (siehe [E_MPBus_DeltaTLimitation](#) [[81](#)]).

rSetPoint: dT Grenzwert.

rSpFlow_DeltaT_Si: Durchfluss bei Sättigung.

VAR_OUTPUT

```

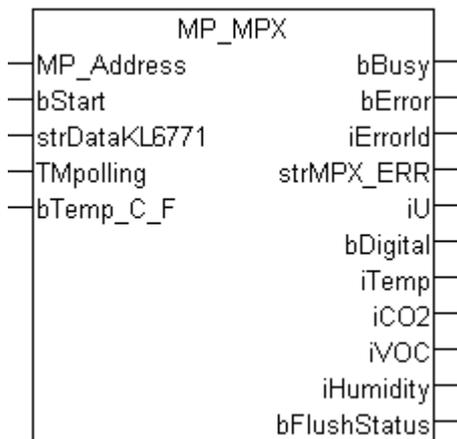
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [[82](#)]). Gleichzeitig wird *bError* TRUE.

6.2.12 MP_MPX



Für BELIMO Raum Sensor MS24A-R..-MPX.

MS24A-R01-MPX Temperature

MS24A-R02-MPX Temperature, CO2

MS24A-R03-MPX Temperature, VOC

MS24A-R04-MPX Temperature, CO2, VOC

MS24A-R05-MPX Temperature, Humidity

MS24A-R06-MPX Temperature, Humidity, CO2

MS24A-R07-MPX Temperature, Humidity, VOC

MS24A-R08-MPX Temperature, Humidity, CO2, VOC

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
TMPolling       : TIME := t#10s;
bTemp_C_F      : BOOL;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) ([27](#)) Baustein verbunden sein muss (siehe [DataKL6771](#) ([85](#))).

TMpolling: Zeit mit der der Baustein den Sensor ansprechen soll. Default 10 s, minimale Zeit 1 s.

bTemp_C_F: FALSE = °C / TRUE = °F.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
iErrorId        : MP_Error;
strMPX_ERR      : MP_BUS_MPX_ERROR;
iU              : INT;
bDigital        : BOOL;
iTemp           : INT;
iCO2            : INT;
```

```
iVOC      : INT;
iHumidity : INT;
bFlushStatus : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error \[▶ 82\]](#)). Gleichzeitig wird *bError* TRUE.

strMPX_ERR: Fehlermeldungen des Sensors (siehe [MP_BUS_MPX_ERROR \[▶ 85\]](#)).

iU: 0...10 V UNIT 1 mV.

bDigital: DI 24 V.

iTemp: 0...50 °C Unit: 0.01 °C.

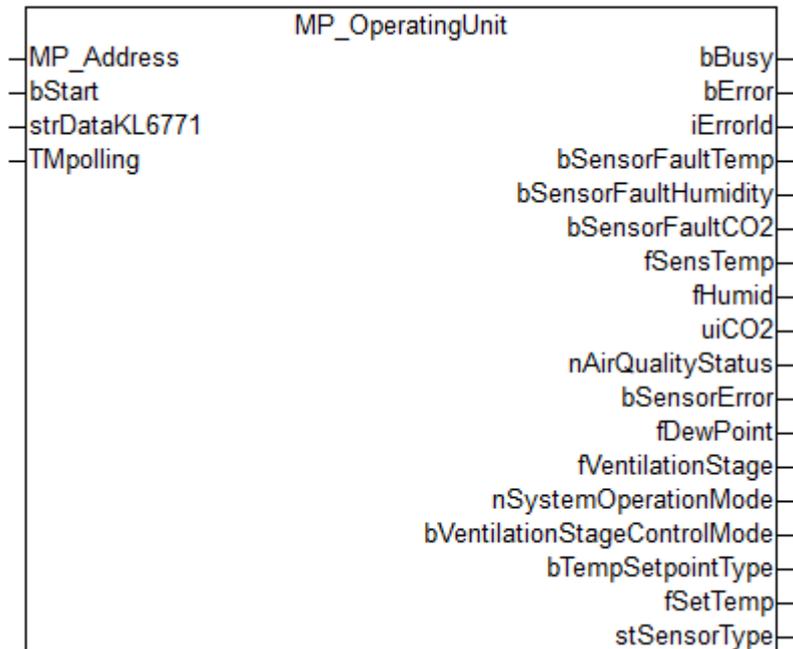
iCO2: 0...2000 ppm Unit: 1 ppm.

iVOC: 0...2000 ppm Unit: 1 ppm (pseudo).

iHumidity: 10...90 % Unit: 0.01 %.

bFlushStatus: VOC Gradienten Schwelle überschritten, FALSE = Luftqualität OK, TRUE = Luftqualität nicht OK, flush.

6.2.13 MP_OperatingUnit



Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor \[▶ 55\]](#)).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address    : USINT := 1;
bStart        : BOOL;
strDataKL6771 : DataKL6771;
TMPolling     : TIME := t#10s;

```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMPolling: Zeit, mit der der Baustein den Antrieb ansprechen soll. Default 10s. Minimale Zeit 1s.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
bSensorFaultTemp      : BOOL;
bSensorFaultHumidity  : BOOL;
bSensorFaultCO2       : BOOL;
fSensTemp         : LREAL;
fHumid           : LREAL;
uiCO2            : UINT;
nAirQualityStatus : E_MP_AirQualityStatus;
bSensorError     : BOOL;
fDewPoint        : LREAL;
fVentilationStage : LREAL;
nSystemOperationMode : E_MP_SystemOperationMode;
bVentilationStageControlMode : BOOL;
bTempSetpointType : BOOL;
fSetTemp         : LREAL;
stSensorType     : STRING;

```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [▶ 82]). Gleichzeitig wird *bError* TRUE.

bSensorFaultTemp: Temperatursensor hat einen Fehler.

bSensorFaultHumidity: Luftfeuchtigkeitssensor hat einen Fehler.

bSensorFaultCO2: CO2-Sensor hat einen Fehler.

fSensTemp: Raumtemperatur in °C oder °F (0..50 oder 32..122).

fHumid: Relative Luftfeuchtigkeit in % (0..100).

uiCO2: CO2-Wert in ppm (0..2000).

nAirQualityStatus: Zustand der Luftqualität (siehe [E_MP_AirQualityStatus](#) [▶ 74]).

bSensorError: Einer der Sensoren hat einen Fehler.

fDewPoint: Taupunkttemperatur in °C (-50..50).

fVentilationStage: Lüftungsstufe in % (0..100).

nSystemOperationMode: Betriebsart des Systems (siehe [E_MP_SystemOperationMode](#) [▶ 79]).

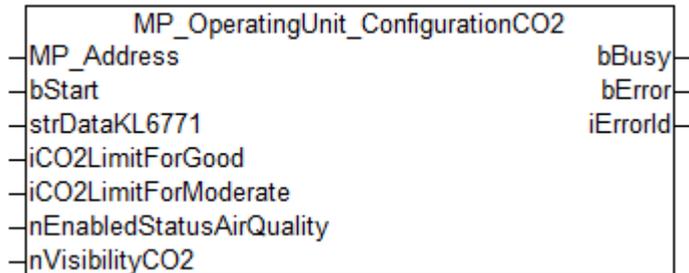
bVentilationStageControlMode: FALSE = Manuell; TRUE = Automatisch.

bTempSetpointType: FALSE = Absolut; TRUE = Relativ.

fSetTemp: bTempSetpointType FALSE = Sollwert Raumtemperatur in °C; bTempSetpointType TRUE = Sollwert relative Raumtemperatur in °C.

stSensorType: DSensortyp. '?' = nicht gelesen; 'unknown' = Nummer unbekannt.

6.2.14 MP_OperatingUnit_ConfigurationCO2



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor](#) Parameter [► 57]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
iCO2LimitForGood : UINT := 1000;
iCO2LimitForModerate : UINT := 1500;
nEnabledStatusAirQuality : E_MP_EnabledStatus;
nVisibilityCO2  : E_MP_DisplayVisibility;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

iCO2LimitForGood: CO2-Grenzwert für „Good“ Luftqualität in ppm (600..1249).

iCO2LimitForModerate: CO2-Grenzwert für „Moderate“ Luftqualität in ppm (1250..2000).

nEnabledStatusAirQuality: Status der Luftqualitätsanzeige (siehe [E_MP_EnabledStatus](#) [► 77]).

nVisibilityCO2: Sichtbarkeit des CO2-Wertes (siehe [E_MP_DisplayVisibility](#) [► 77]).

VAR_OUTPUT

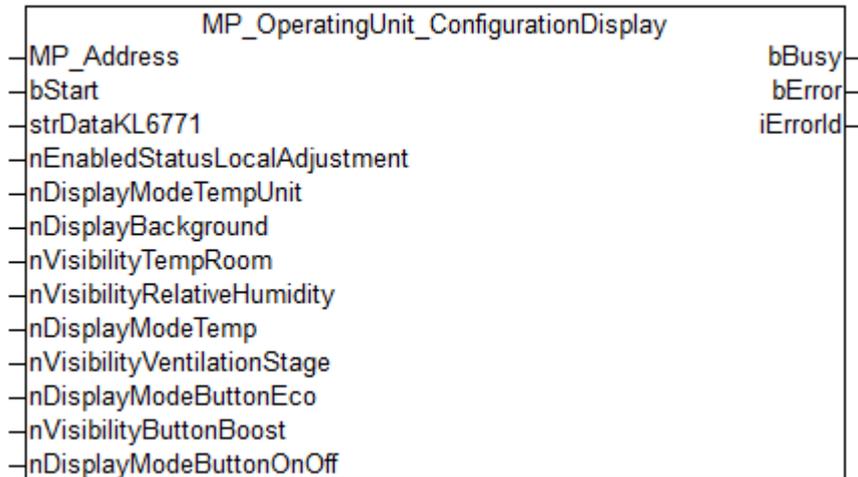
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId   : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [► 82]). Gleichzeitig wird *bError* TRUE.

6.2.15 MP_OperatingUnit_ConfigurationDisplay



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor_Parameter](#) [► 57]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
nEnabledStatusLocalAdjustment : E_MP_EnabledStatus;
nDisplayModeTempUnit : E_MP_DisplayModeTempUnit;
nDisplayBackground : E_MP_DisplayBackground;
nVisibilityTempRoom : E_MP_DisplayVisibility;
nVisibilityRelativeHumidity : E_MP_DisplayVisibility;
nDisplayModeTemp : E_MP_DisplayModeTemp;
nVisibilityVentilationStage : E_MP_DisplayVisibility;
nDisplayModeButtonEco : E_MP_DisplayModeButton;
nVisibilityButtonBoost : E_MP_DisplayVisibility;
nDisplayModeButtonOnOff : E_MP_DisplayModeButton;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

nEnabledStatusLocalAdjustment: Status der Zugriffserlaubnis für lokale Anpassungen (siehe [E_MP_EnabledStatus](#) [► 77]).

nDisplayModeTempUnit: Anzeigemodus der Einheit für die Temperatur im Display (siehe [E_MP_DisplayModeTempUnit](#) [► 76]).

nDisplayBackground: Hintergrundfarbe des Displays (siehe [E_MP_DisplayBackground](#) [► 75]).

nVisibilityTempRoom: Sichtbarkeit der Raumtemperatur (siehe [E_MP_DisplayVisibility](#) [► 77]).

nVisibilityRelativeHumidity: Sichtbarkeit der relativen Luftfeuchtigkeit (siehe [E_MP_DisplayVisibility](#) [► 77]).

nDisplayModeTemp: Anzeigemodus der Temperatur (siehe [E_MP_DisplayModeTemp](#) [► 76]).

nVisibilityVentilationStage: Sichtbarkeit der Lüftungsstufe (siehe [E_MP_DisplayVisibility](#) [► 77]).

nDisplayModeButtonEco: Anzeigemodus der Eco-Taste (siehe [E_MP_DisplayModeButton](#) [► 75]).

nVisibilityButtonBoost: Anzeigemodus der Boost-Taste (siehe [E_MP_DisplayVisibility](#) [► 77]).

nDisplayModeButtonOnOff: Anzeigemodus der OnOff-Taste (siehe [E_MP_DisplayModeButton](#) [► 75]).

VAR_OUTPUT

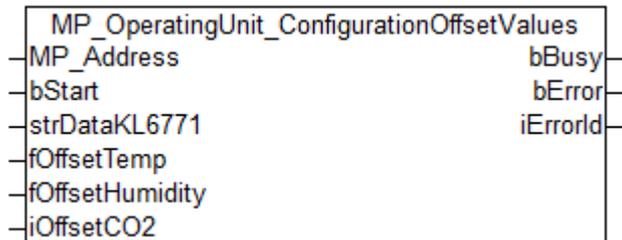
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId  : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [► 82]). Gleichzeitig wird *bError* TRUE.

6.2.16 MP_OperatingUnit_ConfigurationOffsetValues



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor Parameter](#) [► 57]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
fOffsetTemp     : LREAL := 0;
fOffsetHumidity : LREAL := 0;
iOffsetCO2      : INT := 0;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

fOffsetTemp: Auf die gemessene Temperatur angewandter Offset in °C oder °F (-15..15 oder -27..27).

fOffsetHumidity: Auf die gemessene relative Luftfeuchtigkeit angewandter Offset in % (-20..20).

iOffsetCO2: Auf den gemessenen CO2-Gehalt angewandter Offset in ppm (-500..500).

VAR_OUTPUT

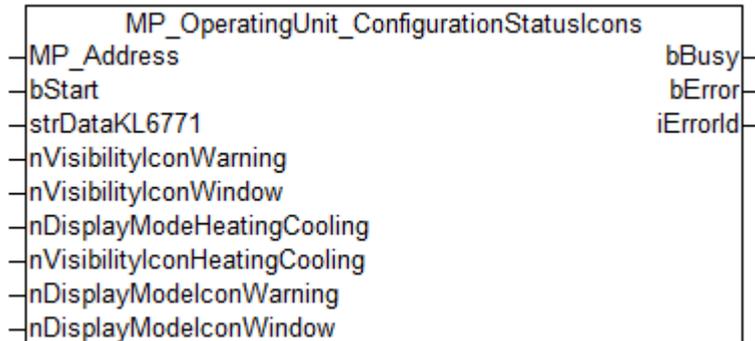
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId  : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [▶ 82]). Gleichzeitig wird *bError* TRUE.

6.2.17 MP_OperatingUnit_ConfigurationStatusIcons



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor Parameter](#) [▶ 57]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
nVisibilityIconWarning : E_MP_DisplayVisibility;
nVisibilityIconWindow   : E_MP_DisplayVisibility;
nDisplayModeHeatingCooling : E_MP_DisplayModeHeatingCooling;
nVisibilityIconHeatingCooling : E_MP_DisplayVisibility;
nDisplayModeIconWarning   : E_MP_DisplayModeIconWarning;
nDisplayModeIconWindow    : E_MP_DisplayModeIconWindow;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

nVisibilityIconWarning: Sichtbarkeit des Warnsymbols. Gilt, wenn [nDisplayModelIconWarning](#) auf 1 gesetzt ist (siehe [E_MP_DisplayVisibility](#) [▶ 77]).

nVisibilityIconWindow: Sichtbarkeit des Fenstersymbols. Gilt, wenn [nDisplayModelIconWindow](#) auf 1 gesetzt ist (siehe [E_MP_DisplayVisibility](#) [▶ 77]).

nDisplayModeHeatingCooling: Anzeigemodus von Heiz- oder Kühlsymbolen (siehe [E_MP_DisplayModeHeatingCooling](#) [▶ 75]).

nVisibilityIconHeatingCooling: Sichtbarkeit von Heiz- oder Kühlsymbolen (siehe [E_MP_DisplayVisibility](#) [▶ 77]).

nDisplayModelIconWarning: Anzeigemodus des Warnsymbols (siehe [E_MP_DisplayModelIconWarning](#) [▶ 75]).

nDisplayModelIconWindow: Anzeigemodus des Fenstersymbols (siehe [E_MP_DisplayModelIconWindow](#) [[▶ 76](#)]).

VAR_OUTPUT

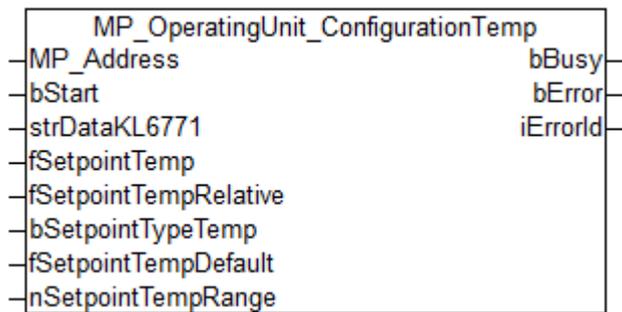
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId  : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [[▶ 82](#)]). Gleichzeitig wird *bError* TRUE.

6.2.18 MP_OperatingUnit_ConfigurationTemp



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor Parameter](#) [[▶ 57](#)]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
fSetpointTemp   : LREAL := 20;
fSetpointTempRelative : LREAL := 0;
bSetpointTypeTemp : BOOL;
fSetpointTempDefault : LREAL := 20;
nSetpointTempRange : USINT := 10;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[▶ 27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[▶ 85](#)]).

fSetpointTemp: Sollwert für die Raumtemperatur in °C (5..45).

fSetpointTempRelative: Sollwert für die relative Raumtemperatur in °C (-5.6..5.6).

bSetpointTypeTemp: FALSE = Absolut; TRUE = Relativ.

fSetpointTempDefault: Standard-Sollwert für die Raumtemperatur in °C (15..35).

nSetpointTempRange: Sollwertbereich für die Temperatur in °C (0..10).

VAR_OUTPUT

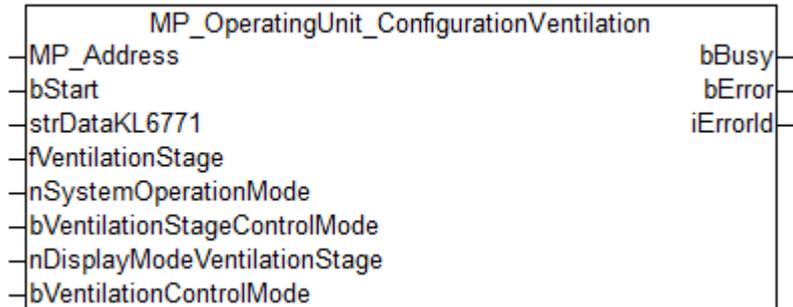
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId   : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[▶ 82\]](#)). Gleichzeitig wird *bError* TRUE.

6.2.19 MP_OperatingUnit_ConfigurationVentilation



Dieser Funktionsbaustein dient zur Konfiguration von Raumsensoren der Typen P-22Rxx-1900x-1. Er ist kompatibel mit Raumsensoren der Typen 22Rxx-19-1, die ab Mai 2022 produziert wurden (vor Mai 2022, siehe [MP_RoomSensor_Parameter \[▶ 57\]](#)).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771   : DataKL6771;
fVentilationStage : LREAL;
nSystemOperationMode : E_MP_SystemOperationMode;
bVentilationStageControlMode : BOOL;
nDisplayModeVentilationStage : E_MP_DisplayModeVentilationStage := MPBus_DisplayModeVentilationStage_7;
bVentilationControlMode : BOOL;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\) \[▶ 27\]](#) Baustein verbunden sein muss (siehe [DataKL6771 \[▶ 85\]](#)).

fVentilationStage: Lüftungsstufe im Raum/Zone in % (0..100).

nSystemOperationMode: Betriebsart des Systems (siehe [E_MP_SystemOperationMode \[▶ 79\]](#)).

bVentilationStageControlMode: Modus der Lüftungsstufenregelung. Gilt, wenn *bVentilationControlMode* auf TRUE gesetzt ist.

nDisplayModeVentilationStage: Anzeigemodus der Lüftungsstufe (siehe [E_MP_DisplayModeVentilationStage \[▶ 76\]](#)).

bVentilationControlMode: FALSE = nur manueller Modus; TRUE = hybrider Regelungsmodus (Sollwert im Auto-Modus unsichtbar).

VAR_OUTPUT

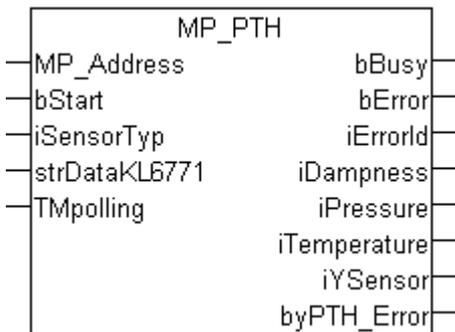
```
bBusy      : BOOL;
bError     : BOOL;
iErrorId  : MP_Error;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

6.2.20 MP_PTH



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines PTH Sensors.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll.

bStart aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Sensor angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

Sollte am Sensor ein Externer-Sensor angeschlossen sein, ist über **iSensorTyp** anzugeben, welcher Sensor dies ist. Ist kein Sensor angeschlossen, so ist der Wert "0" einzutragen oder die Variable wird offengelassen. Ein digitaler Sensor ist mit "3" zu parametrieren. Der Zustand des Sensors wird über die Variable **iYSensor** ausgegeben.

VAR_INPUT

```
MP_Address  : USINT := 1;
bStart      : BOOL;
iSensorTyp  : INT;
strDataKL6771 : DataKL6771;
Tmpolling   : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

iSensorTyp: "0" oder offen - kein Sensor angeschlossen, "1" analoger Sensor angeschlossen Spannung Ausgabe in mV, "2" Ausgabe eines Widerstandes in Ohm - 1,0 Ohm , "3" Ausgabe eines Widerstandes in Ohm - 0,1 Ohm, "4" digitaler Sensor.

strDataKL6771: Datenstruktur die mit dem [KL6771\(0\) \[► 27\]](#) Baustein verbunden sein muss (siehe [DataKL6771 \[► 85\]](#)).

Tmpolling: Zeit mit der der Baustein den Sensor ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
iErrorId   : MP_Error;
iDampness  : INT;
iPressure  : INT;
iTemperature : INT;
iYSensor   : INT;
byPTH_Error : BYTE;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe MP_Error [▶ 82]). Gleichzeitig wird *bError* TRUE.

iDampness: Relative Feuchte in 0,01 %.

iPressure: Differenzdruck, Ausgabe in 0,1 Pa.

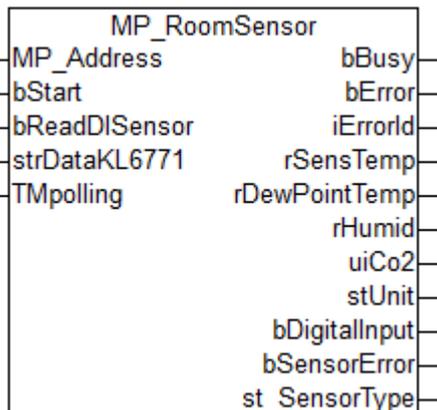
iTemperature: Temperatur in 0,01 °C.

iYSensor: Y-Eingang, iSensorTyp = "1" - Spannung 0...10V Ausgabe in [mV], iSensorTyp = "2" Widerstand Ausgabe in 1,0 Ohm, iSensorTyp = "3" Widerstand Ausgabe in 0,1 Ohm, iSensorTyp = "4" digitaler Schalter 0 oder 1.

byPTH_Error: Sensor Fehler - 0 - kein Fehler.

byPTH_Error	Beschreibung
Bit 0	Wartungsfehler
Bit 1	Störmeldung, Fühlerdefekt
Bit 2	-
Bit 3	-
Bit 4	Sensor (Temperatur/Feuchte) defekt
Bit 5	AD-Wandler (Druck) defekt
Bit 6	AD-Wandler (Y-Eingang) defekt
Bit 7	-

6.2.21 MP_RoomSensor



Dieser Funktionsbaustein dient zum Auslesen von Raumsensoren der Typen 22Rxx-19-1, die vor Mai 2022 produziert wurden (ab Mai 2022, siehe MP_OperatingUnit [▶ 46]).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
bReadDISensor  : BOOL;
strDataKL6771  : DataKL6771;
TMPolling      : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

bReadDISensor: Wenn TRUE, dann wird der DI Sensor ausgelesen und das Ergebnis steht in *bDigitalInput*.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) ([▶ 27](#)) Baustein verbunden sein muss (siehe [DataKL6771](#) ([▶ 85](#))).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
rSensTemp      : LREAL;
rDewPointTemp  : LREAL;
rHumid         : LREAL;
uiCo2          : UINT;
stUnit         : STRING;
bDigitalInput  : BOOL;
bSensorError   : BOOL;
st_SensorType  : STRING;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) ([▶ 82](#))). Gleichzeitig wird *bError* TRUE.

rSensTemp: Sensortemperatur in °C oder °F.

rDewPointTemp: Temperatur des berechneten Taupunktes in °C oder °F.

rHumid: Luftfeuchtigkeit in Prozent (% 0.01).

uiCo2: CO2-Gehalt in ppm.

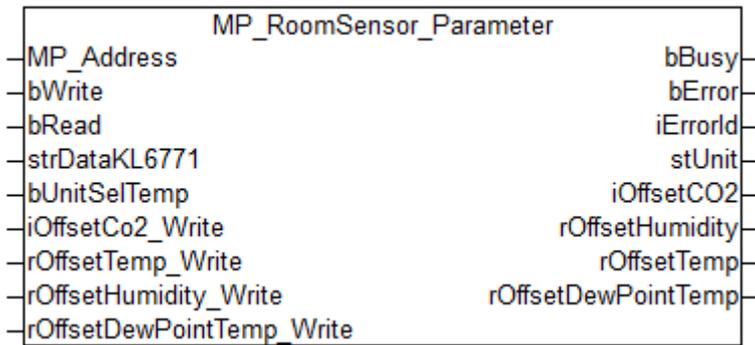
stUnit: C = °C oder F = °F, ? = nicht gelesen.

bDigitalInput: Ausgelesene DI-Sensor, wenn *bReadDISensor* TRUE ist.

bSensorError: Einer der Sensoren hat einen Fehler.

st_SensorType: Sensortyp. '?' = nicht gelesen / Typ / 'unknown' = Nummer unbekannt.

6.2.22 MP_RoomSensor_Parameter



Dieser Funktionsbaustein dient zur Parametrierung von Raumsensoren der Typen 22Rxx-19-1, die vor Mai 2022 produziert wurden (ab Mai 2022, siehe MP_OperatingUnit_Configuration).

Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bRead** liest die Parameter aus, **bWrite** schreibt sie in den Raumsensor. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bWrite          : BOOL;
bRead           : BOOL;
strDataKL6771  : DataKL6771;
bUnitSelTemp   : BOOL;
iOffsetCo2_Write : INT;
rOffsetTemp_Write : LREAL;
rOffsetHumidity_Write : LREAL;
rOffsetDewPointTemp_Write : LREAL;
```

MP_Address: MP-Bus Adresse des Slaves.

bWrite: Positive Flanke startet den Baustein und schreibt den Parameter.

bRead: Positive Flanke startet den Baustein und liest den Parameter.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [► 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [► 85]).

bUnitSelTemp: FALSE = °C, TRUE = °F.

iOffsetCo2_Write: OffsetCO2 [ppm] -500...500.

rOffsetTemp_Write: OffsetTemp [UnitSel] -15...15 °C (-27...27 °F)

rOffsetHumidity_Write: OffsetHumidity [%] -20...+20.

rOffsetDewPointTemp_Write: OffsetDewPointTemp [UnitSel] -15...15 °C (-27...27 °F).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
stUnit         : STRING;
iOffsetCO2     : INT;
rOffsetHumidity : LREAL;
rOffsetTemp    : LREAL;
rOffsetDewPointTemp : LREAL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR \[► 82\]](#)). Gleichzeitig wird *bError* TRUE.

stUnit: C = °C oder F = °F, ? = nicht gelesen.

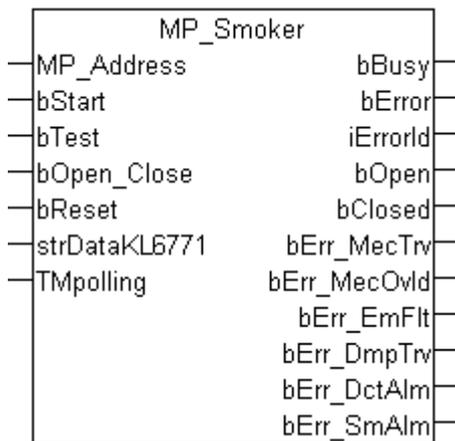
iOffsetCO2: OffsetCO2 [ppm].

rOffsetHumidity: OffsetHumidity [%] 0.01.

rOffsetTemp: OffsetTemp [°C oder °F] 0.01.

rOffsetDewPointTemp: DewOffsetTemp [°C oder °F] 0.01.

6.2.23 MP_Smoker



Dieser Funktionsbaustein dient zur Steuerung und Überwachung einer Brandschutzklappe.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen, maximal 30s. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorID** ausgelesen werden.

Positive Flanke von **bReset** löscht anstehende Fehlermeldungen des Antriebs. Damit wird ein Telegramm an den Antrieb versendet, dass die Fehler in dem Antrieb quittiert. Liegen diese weiterhin an, so bleiben diese gesetzt. Das gilt für alle *bErr_** Fehlerbits.

Über **bOpen_Close** wird die Brandschutzklappe geöffnet oder geschlossen. Ein TRUE bewirkt das Öffnen einer Brandschutzklappe, ein FALSE ein Schließen der Klappe. Über **bOpen** wird angezeigt, dass die Klappe geöffnet ist und über **bClosed** geschlossen ist. Sind beide Bits auf FALSE, öffnet oder schließt der Antrieb grade.

Mit einer positiven Flanke von **bTest** wird ein Testlauf der Brandschutzklappe angestoßen. Gesetzte Fehler können hiermit zurückgenommen werden, wenn sie nicht mehr vorliegen.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL  := TRUE;
bTest           : BOOL;
bOpen_Close    : BOOL;
bReset         : BOOL;
strDataKL6771  : DataKL6771;
TMPolling      : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

bTest: Positive Flanke startet den Testlauf einer Brandschutzklappe.

bOpen_Close: TRUE öffnet eine Klappe, FALSE schließt eine Klappe.

bReset: Positive Flanke setzt die Fehlermeldungen des Antriebs zurück.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[85](#)]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
iErrorId  : MP_Error;
bOpen     : BOOL;
bClosed   : BOOL;
bErr_MecTrv : BOOL;
bErr_MecOvld : BOOL;
bErr_EmFlt : BOOL;
bErr_DmpTrv : BOOL;
bErr_DctAlm : BOOL;
bErr_SmAlm : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) [[82](#)]). Gleichzeitig wird *bError* TRUE.

bOpen: Brandschutzklappe ist geöffnet.

bClosed: Brandschutzklappe ist geschlossen.

bErr_MecTrv: Antriebsfehler "Stellwinkel überschritten", Antriebsdrehwinkel gegenüber Adaption um mehr als 10° überschritten.

bErr_MecOvld: Antriebsfehler "Überlast", Sollposition konnte nicht erreicht werden.

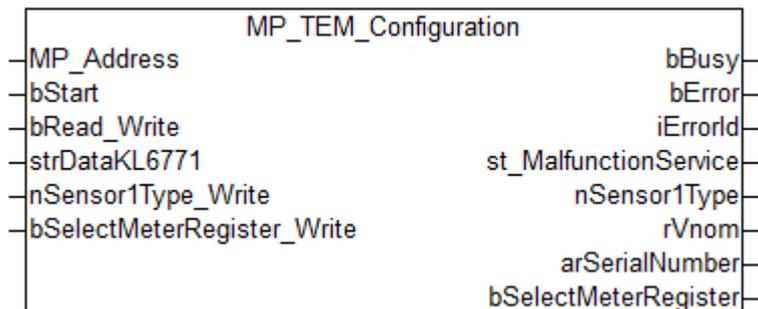
bErr_EmFlt: Antriebsfehler "Sicherheitsrelevante Störung", Umgebungstemperatur über 72°C oder Motortemperatur größer 85°C, Fehler kann nur im Werk zurückgesetzt werden.

bErr_DmpTrv: Antriebsfehler "Klappengängigkeitsfehler", wird gelöscht sollte der Testlauf in Ordnung sein.

bErr_DctAlm: Antriebsfehler "Kanaltemperatur zu groß", Antrieb pendelt hin und her.

bErr_SmAlm: Antriebsfehler "Alarm Rauchmelder".

6.2.24 MP_TEM_Configuration



Dieser Funktionsbaustein dient zur Konfiguration der Thermal Energy Meter Typ 22PE-.. und 22PEM-... . Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
bRead_Write    : BOOL;
strDataKL6771  : DataKL6771;
nSensor1Type_Write : E_MP_EV_V4_Sensor1Type;
bSelectMeterRegister_Write : BOOL;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

bRead_Write: Wenn FALSE dann nur LESEN, bei TRUE LESEN und SCHREIBEN.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[85](#)]).

nSensor1Type_Write: Sensor 1 Typ (siehe [E_MP_EV_V4_Sensor1Type](#) [[79](#)]).

bSelectMeterRegister_Write: FALSE = zertifiziertes Zählwerk; TRUE = Lebensdauer Zählwerk.

VAR_OUTPUT

```
bBusy          : BOOL
bError         : BOOL
iErrorId       : MP_Error
st_MalfunctionService : St_MP_EV_V4_MalfunctionServiceInfo
nSensor1Type   : E_MP_EV_V4_Sensor1Type
rVnom          : LREAL
arSerialNumber : ARRAY[0..1] OF DWORD
bSelectMeterRegister : BOOL
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [[82](#)]). Gleichzeitig wird *bError* TRUE.

st_MalfunctionService: Störungs- und Serviceinformationen (siehe [St_MP_EV_V4_MalfunctionServiceInfo](#) [[86](#)]).

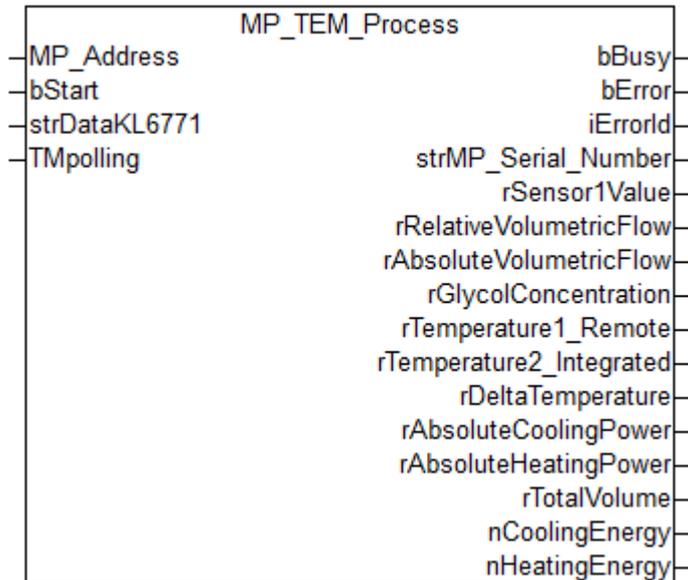
nSensor1Type: Sensor 1 Typ (siehe [E_MP_EV_V4_Sensor1Type](#) [[79](#)]).

rVnom: Nennvolumenstrom in l/s (0...100).

arSerialNumber: Seriennummer des Geräts.

bSelectMeterRegister: FALSE = zertifiziertes Zählwerk; TRUE = Lebensdauer Zählwerk.

6.2.25 MP_TEM_Process



Dieser Funktionsbaustein ist geeignet für die Thermal Energy Meter Typ 22PE-.. und 22PEM-... . Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Bus-Teilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771  : DataKL6771;
Tm polling      : TIME:= t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) ([▶ 27](#)) Baustein verbunden sein muss (siehe [DataKL6771](#) ([▶ 85](#))).

Tm polling: Zeit, mit der der Baustein den Antrieb ansprechen soll. Default 10 s. Minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
iErrorId        : MP_Error;
strMP_Serial_Number : MP_Serial_Number;
rSensor1Value   : LREAL;
rRelativeVolumetricFlow : LREAL;
rAbsoluteVolumetricFlow : LREAL;
rGlycolConcentration : LREAL;
rTemperature1_Remote : LREAL;
rTemperature2_Integrated : LREAL;
rDeltaTemperature : LREAL;
rAbsoluteCoolingPower : LREAL;
rAbsoluteHeatingPower : LREAL;
rTotalVolume    : LREAL;
nCoolingEnergy  : DINT;
nHeatingEnergy  : DINT;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe MP_ERROR [▶ 82]). Gleichzeitig wird *bError* TRUE.

strMP_Serial_Number: Struktur für die Seriennummer (siehe MP Serial Number [▶ 85]).

rSensor1Value: Sensor 1 Wert in mV/Ohm (0...65535).

rRelativeVolumetricFlow: Relativer Volumenstrom in % (0...100).

rAbsoluteVolumetricFlow: Absoluter Volumenstrom in l/s (0...100).

rGlycolConcentration: Glykol Konzentration in % (0...100).

rTemperature1_Remote: Temperatur1 remote in °C (-20...12).

rTemperature2_Integrated: Temperature2 integriert in °C (-20...12).

rDeltaTemperature: Temperature Delta in K (0...14).

rAbsoluteCoolingPower: Absolute Kühlleistung in kW (0...21.5).

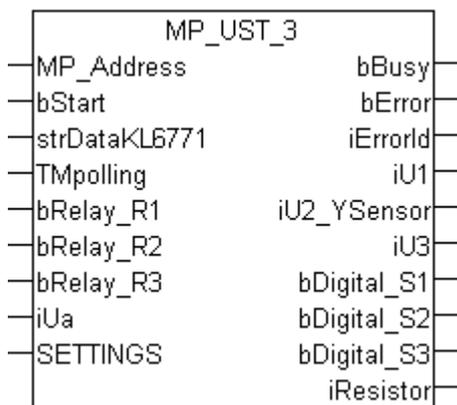
rAbsoluteHeatingPower: Absolute Heizleistung in kW (0...21.5).

rTotalVolume: Gesamtvolumen in m³ (0...214748.36).

nCoolingEnergy: Kühlenergie in kWh (0...21474836).

nHeatingEnergy: Heizenergie in kWh (0...21474836).

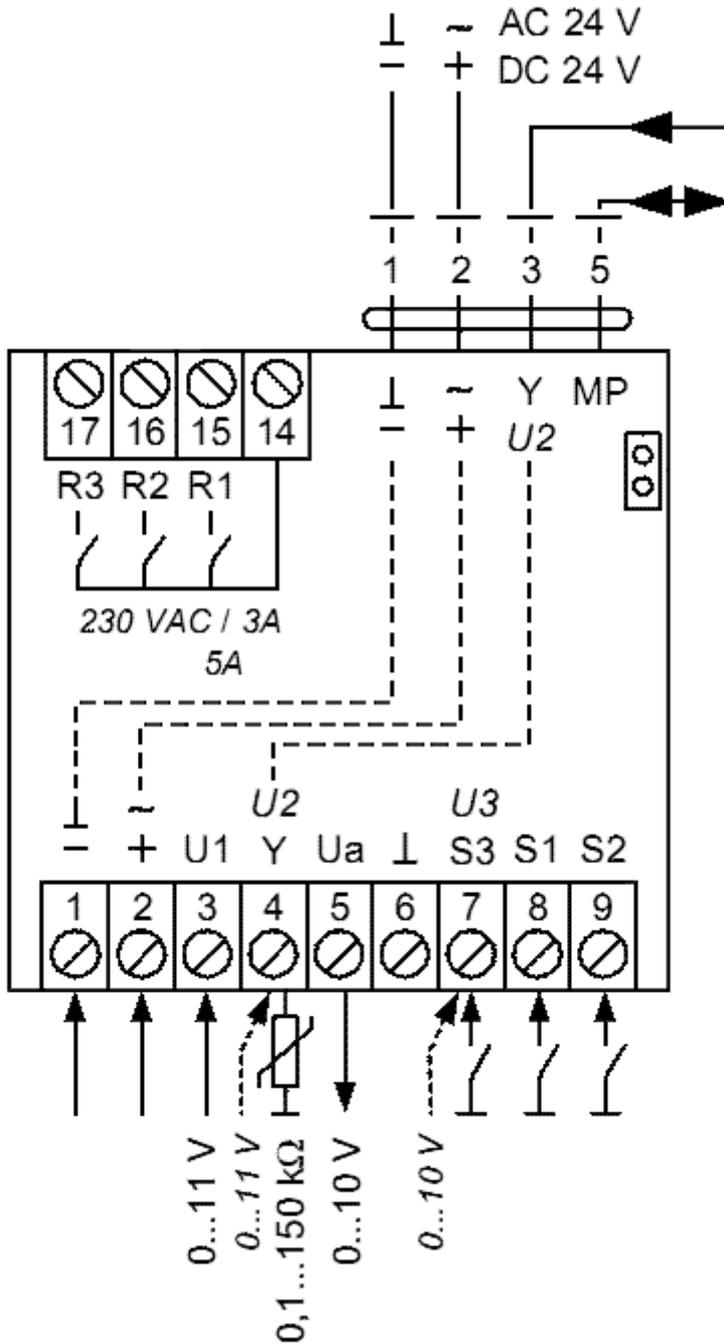
6.2.26 MP_UST_3



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Multi IO Moduls UST3.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus-Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer als eine Sekunde einzustellen. **bError** zeigt einen Fehler in der Kommunikation mit dem Antrieb an. Die Art des Fehlers kann mit **iErrorID** ausgelesen werden.

<https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063688459.gif>



Die Eingangsdaten **bRelay_R1** bis **bRelay_R3** schalten die Relais R1 bis R3 (PIN 15 bis PIN 17). Die Variable **iUa** schaltet den analogen Ausgang 0...10 V auf PIN 5. Dabei entspricht ein Digit einem mV.

Die Datenstruktur **SETTINGS** wird für die Parametrierung des UST3 verwendet. Es kann die Skalierung der analogen Eingangsdaten eingestellt, sowie die Widerstandsmessung auf PIN4 aktiviert werden. Für die Widerstandsmessung kann die Skalierung des Widerstandsmesswert geändert werden. Dies kann auch im laufenden Betrieb erfolgen. **iU1** ist der Analoge Eingang auf PIN 3. Dabei entspricht ein Digit einem mV oder, wenn man die Skalierung in der Datenstruktur **SETTINGS** umgestellt hat, 250 µV. Das gleiche gilt auch für die analogen Eingänge **iU2_YSensor** (PIN 4) oder auch **iU3** (PIN 7). **iU2_YSensor** kann auch als Widerstandsmessung genutzt werden. Dies muss über die Datenstruktur **SETTINGS** eingestellt werden. **bDigital_S1** bis **bDigital_S3** entsprechen den digitalen Eingängen des UST3, PIN 7 bis PIN 9.

Alle Daten werden von der MP-Bus-Masterklemme KL6771 automatisch gepollt. Die Pollgeschwindigkeit hängt ab von der Anzahl der angeschlossenen MP-Bus-Teilnehmer und der eingestellten Pollzeit. Die digitalen Eingänge **eignen sich nicht** für den Anschluss von Tastern oder Sensoren die nur kurze Impulse ausgeben. Um einen Pegelwechsel sicher mitzubekommen, muss dieser minimal eine Sekunde anliegen.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771  : DataKL6771;
TMPolling      : TIME := t#10s;
bRelay_R1      : BOOL;
bRelay_R2      : BOOL;
bRelay_R3      : BOOL;
iUa            : UINT;
SETTINGS       : UST3_SET;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [[27](#)] Baustein verbunden sein muss (siehe [DataKL6771](#) [[85](#)]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

bRelay_R1: Relais PIN 15.

bRelay_R2: Relais PIN 16.

bRelay_R3: Relais PIN 17.

iUa: Analoger Ausgang PIN 5 (1 mV = 1 Digit).

SETTINGS: Datenstruktur zum Einstellen der Skalierung und der Widerstandsmessung (siehe [UST3_SET](#) [[87](#)]).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
iU1            : INT;
iU2_YSensor    : INT;
iU3            : INT;
bDigital_S1    : BOOL;
bDigital_S2    : BOOL;
bDigital_S3    : BOOL;
iResistor      : INT;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) [[82](#)]). Gleichzeitig wird *bError* TRUE.

iU1: Analoger Eingang PIN 3 (1 Digit = 1 mV oder 1 Digit = 250 µV).

iU2_YSensor: Analoger Eingang PIN 4 (1 Digit = 1 mV oder 1 Digit = 250 µV).

iU3: Analoger Eingang PIN 7 (1 Digit = 1 mV oder 1 Digit = 250 µV).

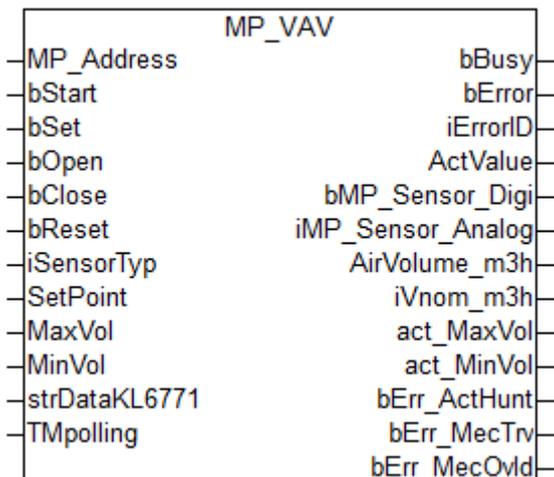
bDigital_S1: Digitaler Eingang PIN 8.

bDigital_S2: Digitaler Eingang PIN 9.

bDigital_S3: Digitaler Eingang PIN 7.

iResistor: Widerstandswert PIN 4.

6.2.27 MP_VAV



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Volumenstromreglers.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorID** ausgelesen werden.

Positive Flanke von **bReset** löscht anstehende Fehlermeldungen des Antriebs. Damit werden Fehler zurückgesetzt, die die Ausgangsvariablen **bErr_MecOvld**, **bErr_ActHunt** und **bErrMecTrv** betreffen. Sollte der Fehler noch anstehen, werden diese Fehlerbits vom Antrieb erneut gesetzt.

Mit **SetPoint** wird der Volumenstrom von 0...100 % eingestellt. Die aktuelle Stellung der Klappe wird über **ActValue** ausgelesen.

Sollte am Antrieb ein Sensor angeschlossen sein, ist über **iSensorTyp** anzugeben welcher Sensor dies ist. Ist kein Sensor angeschlossen, so ist der Wert "0" einzutragen oder die Variable wird offengelassen. Ein digitaler Sensor ist mit "1" zu parametrieren. Der Zustand des Sensors kann dann über **bMP_Sensor_Digi** abgefragt werden. Analoge Sensoren "2...6" werden in der Variable **iMP_Sensor_Analog** ausgegeben.

Eine positive Flanke der Eingänge **bOpen** oder **bClose** öffnet bzw. schließt die Klappe des Antriebs. Eine negative Flanke der beiden Eingänge löscht den Befehl wieder.

Mit **MaxVol** und **MinVol** kann ein maximaler und ein minimaler Volumenstrom im Antrieb hinterlegt werden. Eine positive Flanke von **bSet** schreibt die Daten zum Antrieb. Den aktuellen Wert erhalten Sie in den Ausgangsdaten **act_MaxVol** und **act_MinVol**. Der aktuellen Volumenstrom wird in der Variable **AirVolume_m3h** ausgegeben.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
bSet            : BOOL;
bOpen          : BOOL;
bClose         : BOOL;
bReset         : BOOL;
iSensorTyp     : INT;
SetPoint       : USINT;
MaxVol         : WORD;
MinVol         : WORD;
strDataKL6771  : DataKL6771;
TMPolling      : TIME := t#10s;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit **TMPolling** der Baustein zyklisch aktiv.

bSet: Positive Flanke schreibt die Daten *MaxVol* und *MinVol* zum Antrieb.

bOpen: Positive Flanke öffnet die Klappen des Antriebs, eine negative Flanke löscht die Zwangsbelüftung.

bClose: Positive Flanke schließt die Klappen des Antriebs, eine negative Flanke löscht die Zwangsverschließung.

bReset: Positive Flanke setzt die Fehlermeldungen des Antriebs zurück.

iSensorTyp: 0: kein Sensor angeschlossen, 1: digitaler Sensor angeschlossen, 2: analoger Sensor angeschlossen (0...35 V), 3...6: Ausgabe eines Widerstandes in Ohm (3..5 gilt für PT1000, NI1000 und NI1000LuS; 6 gilt für NTC). Zum Umrechnen in eine Temperatur sind die entsprechenden Konvertierungsfunktionen zu verwenden.

SetPoint: Soll-Volumenstrom (0...100 %).

MaxVol: Maximaler-Volumenstrom (30...100 %).

MinVol: Minimaler-Volumenstrom (0...100 %).

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) ([▶ 27](#)) Baustein verbunden sein muss (siehe [DataKL6771](#) ([▶ 85](#))).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorID       : MP_Error;
ActValue       : WORD;
bMP_Sensor_Digi : BOOL;
iMP_Sensor_Analog : INT;
AirVolume_m3h  : WORD;
iVnom_m3h      : INT;
act_MaxVol     : INT;
act_MinVol     : INT;
bErr_ActHunt   : BOOL;
bErr_MecTrv    : BOOL;
bErr_MecOvld   : BOOL;
    
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) ([▶ 82](#))). Gleichzeitig wird *bError* TRUE.

ActValue: Enthält die aktuelle Position (0...100 %) des Antriebs.

bMP_Sensor_Digi: Ist ein digitaler Sensor angeschlossen, wird der Zustand über diese Variable angezeigt. *iSensorTyp* muss "1" sein.

iMP_Sensor_Analog: Ist ein analoger Sensor angeschlossen, wird der Wert über diese Variable angezeigt. *iSensorTyp* muss "2...6" sein.

AirVolume_m3h: Ausgabe des Volumenstroms in m³/h.

iVnom_m3h: Nominal Luftvolumenstrom in m³/h. Dieser Ausgang ist ab Version 1.12.0 verfügbar. VAV wird gelesen und muss > 0 sein. Wenn gleich 0, dann ist die Berechnung von *AirVolume_m3h* nicht korrekt.

act_MaxVol: Maximal eingestellter Volumenstrom in %.

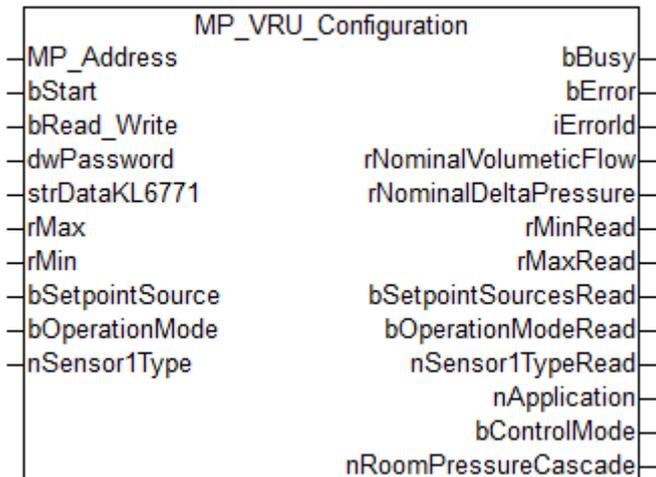
act_MinVol: Minimal eingestellter Volumenstrom in %.

bErr_ActHunt: Antriebsfehler "Regelschwingung", Antrieb pendelt hin und her.

bErr_MecTrv: Antriebsfehler "Stellwinkel überschritten", Antriebsdrehwinkel gegenüber Adaption um mehr als 10° überschritten.

bErr_MecOvld: Antriebsfehler "Überlast", Sollposition konnte nicht erreicht werden.

6.2.28 MP_VRU_Configuration



Dieser Funktionsbaustein dient zur Konfiguration der VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC (max. 8 Slaves). Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```

MP_Address      : USINT := 1;
bStart          : BOOL;
bRead_Write     : BOOL;
dwPassword      : DWORD;
strDataKL6771   : DataKL6771;
rMax            : LREAL := 100;
rMin            : LREAL;
bSetpointSource : BOOL;
bOperationMode  : BOOL := TRUE;
nSensor1Type    : E_MP_VRU_Sensor1Type := MPBus_VRU_Sensor_Active;
    
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein.

bRead_Write: FALSE = nur LESEN. TRUE = LESEN und SCHREIBEN.

dwPassword: Das Passwort für die Antriebe. Üblicherweise 0x0000.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

rMax: Max in % (20...100 %).

rMin: Min in % (0...rMax). *rMin* muss kleiner sein als *rMax*.

bSetpointSource: TRUE = Bus; FALSE = analog

bOperationMode: TRUE = Überdruck; FALSE = Unterdruck

nSensor1Type: [Sensor 1 Typ](#) [▶ 81].

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
rNominalVolumetricFlow : LREAL;
rNominalDeltaPressure : LREAL;
rMinRead       : LREAL;
    
```

```
rMaxRead      : LREAL;
bSetpointSourcesRead : BOOL;
bOperationModeRead : BOOL;
nSensor1TypeRead : E_MP_VRU_Sensor1Type;
nApplication : E_MP_VRU_Application;
bControlMode : BOOL;
nRoomPressureCascade : E_MP_VRU_RoomPressureCascade;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe MP_ERROR [▶ 82]). Gleichzeitig wird *bError* TRUE.

rNominalVolumetricFlow: Nennvolumenstrom in m³/h (0...60.000).

rNominalDeltaPressure: Nenn-Differenzdruck in Pa (0...10.000).

rMinRead: Min in % (0...*rMax*). *rMin* muss kleiner sein als *rMax*.

rMaxRead: Max in % (20...100 %).

bSetpointSourcesRead: TRUE = Bus; FALSE = Analog

bOperationModeRead: TRUE = Überdruck; FALSE = Unterdruck

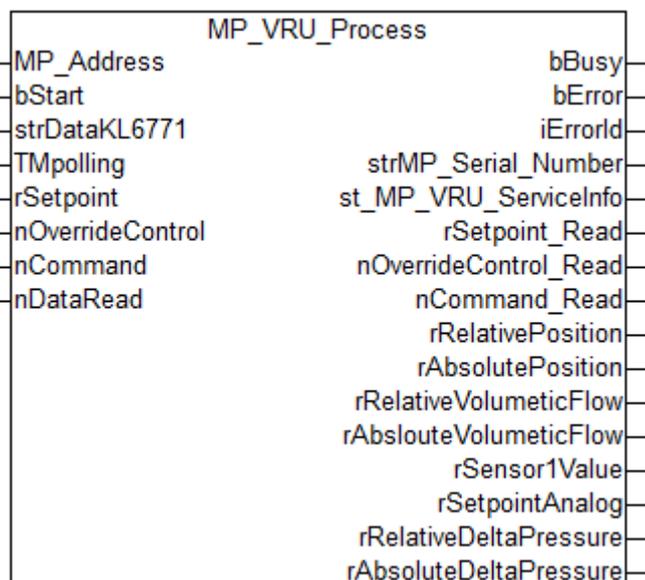
nSensor1TypeRead: Sensor 1 Typ [▶ 81].

nApplication: Visualisierung der vom Hersteller gewählten Anwendung [▶ 80].

bControlMode: Visualisierung der vom Hersteller gewählten Regelfunktion. TRUE = Volumenstromregelung; FALSE = Positionsregelung

nRoomPressureCascade: Raumdruck-Kaskadenregelung [▶ 81].

6.2.29 MP_VRU_Process



Dieser Funktionsbaustein ist geeignet für die VAV Antriebe VRU-D3-BAC, VRU-M1-BAC and VRU-M1R-BAC (max. 8 Slaves). Weitere Informationen finden Sie unter www.belimo.ch.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 s einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorId** ausgelesen werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
strDataKL6771  : DataKL6771;
TMpolling      : TIME := t#10s;
rSetpoint      : LREAL;
nOverrideControl : E_MP_VRU_OverrideControl;
nCommand       : E_MP_VRU_Command;
nDataRead      : BYTE;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [\[▶ 27\]](#) Baustein verbunden sein muss (siehe [DataKL6771](#) [\[▶ 85\]](#)).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

rSetpoint: Wert in % (0...100 %).

nOverrideControl: [Überschreiben des Sollwerts](#) [\[▶ 80\]](#).

nCommand: [Kommando](#) [\[▶ 80\]](#) für Service- und Testfunktionen des Antriebs.

nDataRead: 0xFF - Lese alle Daten; Bit 0 - Lese relative Position; Bit 1 - Lese absolute Position; Bit 2 - Lese relativen Volumenstrom; Bit 3 - Lese absoluten Volumenstrom; Bit 4 - Lese Wert Sensor 1; Bit 5 - Lese analogen Sollwert; Bit 6 - Lese relativen Differenzdruck; Bit 7 - Lese absoluten Differenzdruck

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorId       : MP_Error;
strMP_Serial_Number : MP_Serial_Number;
st_MP_VRU_ServiceInfo : St_MP_VRU_ServiceInfo;
rSetpoint_Read : LREAL;
nOverrideControl_Read : E_MP_VRU_OverrideControl;
nCommand_Read : E_MP_VRU_Command;
rRelativePosition : LREAL;
rAbsolutePosition : LREAL;
rRelativeVolumetricFlow : LREAL;
rAbsoluteVolumetricFlow : LREAL;
rSensor1Value   : LREAL;
rSetpointAnalog : LREAL;
rRelativeDeltaPressure : LREAL;
rAbsoluteDeltaPressure : LREAL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [\[▶ 82\]](#)). Gleichzeitig wird *bError* TRUE.

strMP_Serial_Number: [Struktur](#) [\[▶ 85\]](#) für die Seriennummer.

st_MP_VRU_ServiceInfo: [Störungs- und Serviceinformationen](#) [\[▶ 86\]](#).

rSetpoint_Read: Sollwert.

nOverrideControl_Read: [Übersteuerung](#) [\[▶ 80\]](#).

nCommand_Read: Kommando [► 80].

rRelativePosition: Relative Position in %. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rAbsolutePosition: Absolute Position in °. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rRelativeVolumetricFlow: Relativer Volumenstrom in %. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rAbslouteVolumetricFlow: Absoluter Volumenstrom in m³/h. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

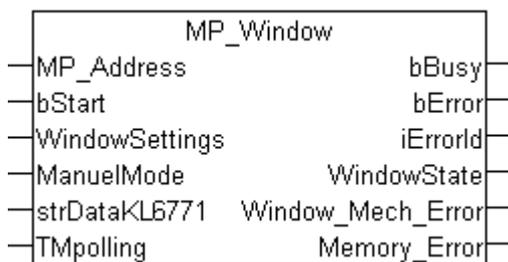
rSensor1Value: Wert von Sensor 1 in mV/Ohm. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rSetpointAnalog: Analoger Sollwert in %. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rRelativeDeltaPressure: Relativer Differenzdruck %. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

rAbsoluteDeltaPressure: Absoluter Differenzdruck in Pa. Wert von -1 bedeutet Daten sind deaktiviert (siehe VAR_INPUT *nDataRead*).

6.2.30 MP_Window



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Fensterlüftungssystems FLS.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 sec einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem FLS angezeigt. Die Art des Fehlers kann mit **iErrorld** ausgelesen werden.

Über **ManuelMode** kann man die Handbedienung aktivieren bzw. deaktivieren. Mit **WindowSettings** kann die Art des Lüftens vorgegeben werden.

VAR_INPUT

```
MP_Address      : USINT := 1;
bStart          : BOOL;
WindowSettings  : Data_Window;
ManuelMode      : BOOL;
strDataKL6771   : DataKL6771;
TMPolling       : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

WindowSettings: Soll-Einstellungen für das Lüften (siehe *Data_Window* [► 74]).

ManuelMode: FALSE: Handbedienung zugelassen. TRUE: Handbedienung deaktiviert.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
iErrorId   : MP_Error;
WindowState : Data_Window;
Window_Mech_Error : BOOL;
Memory_Error : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

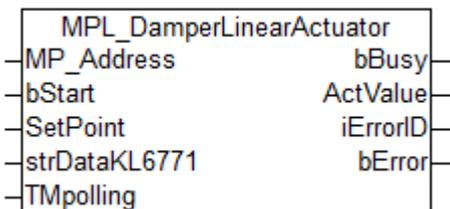
iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error](#) [▶ 82]). Gleichzeitig wird *bError* TRUE.

WindowState: Ist-Einstellungen für das Lüften (siehe [Data_Window](#) [▶ 74]).

Window_Mech_Error: Fenster oder Antrieb ist blockiert.

Memory_Error: Antrieb hat einen Memory Fehler. Neu programmieren oder ersetzen.

6.2.31 MPL_DamperLinearActuator



Dieser Funktionsbaustein dient zur Steuerung und Überwachung eines Klappen- und Hubventilantrieb.

Über **MP_Address** wird vorgegeben, mit welchem MP-Busteilnehmer der Baustein kommunizieren soll. **bStart** aktiviert die Kommunikation mit dem MP-Bus Teilnehmer. Über **bBusy** wird angezeigt, dass der Baustein aktiv ist. Bleibt **bStart** auf TRUE, wird mit der Zeit **TMPolling** der Teilnehmer zyklisch angesprochen. Die Zeit ist größer 1 sec einzustellen. Mit **bError** wird ein Fehler in der Kommunikation mit dem Antrieb angezeigt. Die Art des Fehlers kann mit **iErrorID** ausgelesen werden.

Mit **SetPoint** wird die Stellung der Klappe von 0...100 % eingestellt. Die aktuelle Stellung des Antriebs wird über **ActValue** ausgelesen.

VAR_INPUT

```
MP_Address : USINT := 1;
bStart     : BOOL;
SetPoint   : USINT;
strDataKL6771 : DataKL6771;
TMpolling  : TIME := t#10s;
```

MP_Address: MP-Bus Adresse des Slaves.

bStart: Positive Flanke startet den Baustein. Liegt dauerhaft ein TRUE an, so wird mit der Zeit *TMPolling* der Baustein zyklisch aktiv.

SetPoint: 0...100 % Soll-Klappenstellung des Antriebs.

strDataKL6771: Datenstruktur die mit dem [KL6771\(\)](#) [▶ 27] Baustein verbunden sein muss (siehe [DataKL6771](#) [▶ 85]).

TMpolling: Zeit mit der der Baustein den Antrieb ansprechen soll. Default 10 s, minimale Zeit 1 s.

VAR_OUTPUT

```
bBusy      : BOOL;
ActValue   : WORD;
iErrorID   : MP_Error;
bError     : BOOL;
```

bBusy: Dieses Bit wird so lange gesetzt, wie der Baustein aktiv ist.

ActValue: Enthält die aktuelle Position (0...100 %) des Antriebs.

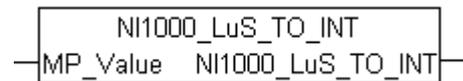
iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_Error \[▶ 82\]](#)). Gleichzeitig wird *bError* TRUE.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

6.3 Funktionen

Bausteine	Beschreibung
NI1000_LuS_TO_INT [▶ 72]	Diese Funktion berechnet aus einem Widerstand NI1000 L&S eine Temperatur.
NI1000_TO_INT [▶ 72]	Diese Funktion berechnet aus einem Widerstand NI1000 eine Temperatur.
NTC_TO_INT [▶ 73]	Diese Funktion berechnet aus einem Widerstand NTC eine Temperatur.
PT1000_TO_INT [▶ 73]	Diese Funktion berechnet aus einem Widerstand PT1000 eine Temperatur.

6.3.1 NI1000_LuS_TO_INT : INT



Diese Funktion berechnet aus einem Widerstand NI1000 L&S eine Temperatur.

Schließen Sie diese Funktion an **iMP_Sensor_Analog** an. Als Ausgabewert erhalten Sie eine INT Variable, die die Temperatur in einer Auflösung von 0.01°C ausgibt (Beispiel 20,5 °C entspricht 2050).

Kleinster, gültiger Wert 872 Ohm entspricht -30 °C, ist der Wert kleiner so wird 16#7FFD ausgegeben.

Größter, gültiger Wert 1586 Ohm entspricht 115 °C, ist der Wert größer so wird 16#7FFE ausgegeben.

VAR_INPUT

```
MP_Value   : WORD;
```

MP_Value: Eingang für einen ohmschen NI1000 L&S Sensor.

NI1000_LuS_TO_INT: Temperatur in 0.01°C.

6.3.2 NI1000_TO_INT : INT



Diese Funktion berechnet aus einem Widerstand NI1000 eine Temperatur.

Schließen Sie diese Funktion an **iMP_Sensor_Analog** an. Als Ausgabewert erhalten Sie eine INT Variable, die die Temperatur in einer Auflösung von 0.01 °C ausgibt (Beispiel 20,5 °C entspricht 2050).

Kleinster, gültiger Wert 867 Ohm entspricht -25 °C, ist der Wert kleiner so wird 16#7FFD ausgegeben.

Größter, gültiger Wert 1583 Ohm entspricht 95 °C, ist der Wert größer so wird 16#7FFE ausgegeben.

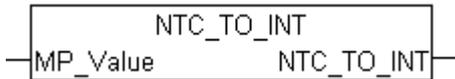
VAR_INPUT

MP_Value : WORD;

MP_Value: Eingang für einen ohmschen NI1000 Sensor

NI1000_TO_INT: Temperatur in 0.01 °C.

6.3.3 NTC_TO_INT : INT



Diese Funktion berechnet aus einem Widerstand NTC eine Temperatur.

Schließen Sie diese Funktion an **iMP_Sensor_Analog** an. Als Ausgabewert erhalten Sie eine INT Variable, die die Temperatur in einer Auflösung von 0.01 °C ausgibt (Beispiel 20,5 °C entspricht 2050).

Kleinster, gültiger Wert 104 Ohm entspricht 145 °C, ist der Wert kleiner so wird 16#7FFD ausgegeben.

Größter, gültiger Wert 48555 Ohm entspricht -20 °C, ist der Wert größer so wird 16#7FFE ausgegeben.

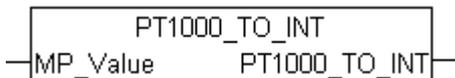
VAR_INPUT

MP_Value : WORD;

MP_Value: Eingang für einen ohmschen NTC Sensor.

NTC_TO_INT: Temperatur in 0.01°C.

6.3.4 PT1000_TO_INT : INT



Diese Funktion berechnet aus einem Widerstand PT1000 eine Temperatur.

Schließen Sie diese Funktion an **iMP_Sensor_Analog** an. Als Ausgabewert erhalten Sie eine INT Variable, die die Temperatur in einer Auflösung von 0.01 °C ausgibt (Beispiel 20,5 °C entspricht 2050).

Kleinster, gültiger Wert 862 Ohm entspricht -35 °C, ist der Wert kleiner so wird 16#7FFD ausgegeben.

Größter, gültiger Wert 1592 Ohm entspricht 155 °C, ist der Wert größer so wird 16#7FFE ausgegeben.

VAR_INPUT

MP_Value : WORD;

MP_Value: Eingang für einen ohmschen PT1000 Sensor.

PT1000_TO_INT: Temperatur in 0.01 °C.

6.4 Datentypen

6.4.1 Enums

6.4.1.1 Data_Window

Mit diesem ENUM kann die Art des Lüftens vorgegeben werden.

```

TYPE Data_Window :
(
  Window_Close      := 8,
  Window_Unlock    := 9,
  Window_Open      := 16#0A,
  Window_20        := 16#0B,
  Window_40        := 16#0C,
  Window_60        := 16#0D,
  Window_80        := 16#0E,
  Window_100       := 16#0F,
  Auto_Close       := 1,
  Auto_5_15min     := 2,
  Auto_8_30min     := 3,
  Auto_10_50min    := 4,
  Auto_open        := 5,
  Auto             := 0,
)
END_TYPE
    
```

Window_Close: Schließt das Fenster.

Window_Unlock: Wenn Sie "Unlock" benutzen, nehmen Sie den Schalter im Fenster, um die MP Bus Kommunikation neu zu starten.

Window_Open: Öffnet das Fenster.

Window_20: Öffnet das Fenster 20 %.

Window_40: Öffnet das Fenster 40 %.

Window_60: Öffnet das Fenster 60 %.

Window_80: Öffnet das Fenster 80 %.

Window_100: Öffnet das Fenster 100 %.

Auto_Close: Automatisches Schließen.

Auto_5_15min: Automatisches Lüften alle 5...15 min.

Auto_8_30min: Automatisches Lüften alle 8...30 min.

Auto_10_50min: Automatisches Lüften alle 10...50 min.

Auto_open: Automatisches Lüften öffnen.

Auto: Automatikbetrieb.

6.4.1.2 E_MP_AirQualityStatus

Status der gemessenen Luftqualität.

```

TYPE E_MP_AirQualityStatus :
(
  MPBus_AirQualityStatus_Deactivated := 0,
  MPBus_AirQualityStatus_OK         := 1,
  MPBus_AirQualityStatus_Warning    := 2,
  MPBus_AirQualityStatus_Alarm      := 3,
);
END_TYPE
    
```

MPBus_AirQualityStatus_Deactivated: Deaktiviert.

MPBus_AirQualityStatus_OK: OK.

MPBus_AirQualityStatus_Warning: Warnung.

MPBus_AirQualityStatus_Alarm: Alarm.

6.4.1.3 E_MP_DisplayBackground

Hintergrundfarbe des Displays.

```
TYPE E_MP_DisplayBackground :
(
  MPBus_DisplayBackground_WhiteOnBlack := 0,
  MPBus_DisplayBackground_BlackOnWhite := 1
);
END_TYPE
```

MPBus_DisplayBackground_WhiteOnBlack: Weiß auf schwarz.

MPBus_DisplayBackground_BlackOnWhite: Schwarz auf weiß.

6.4.1.4 E_MP_DisplayModeButton

Anzeigemodus der Tasten.

```
TYPE E_MP_DisplayModeButton :
(
  MPBus_DisplayModeButton_Invisible := 0,
  MPBus_DisplayModeButton_Status    := 1,
  MPBus_DisplayModeButton_Setpoint  := 2
);
END_TYPE
```

MPBus_DisplayModeButton_Invisible: Unsichtbar.

MPBus_DisplayModeButton_Status: Status.

MPBus_DisplayModeButton_Setpoint: Sollwert.

6.4.1.5 E_MP_DisplayModeHeatingCooling

Anzeigemodus von Heiz- oder Kühlsymbolen.

```
TYPE E_MP_DisplayModeHeatingCooling :
(
  MPBus_DisplayModeHeatingCooling_None    := 0,
  MPBus_DisplayModeHeatingCooling_Heating := 1,
  MPBus_DisplayModeHeatingCooling_Cooling := 2
);
END_TYPE
```

MPBus_DisplayModeHeatingCooling_None: Keine.

MPBus_DisplayModeHeatingCooling_Heating: Heizen.

MPBus_DisplayModeHeatingCooling_Cooling: Kühlen.

6.4.1.6 E_MP_DisplayModelconWarning

Anzeigemodus des Warnsymbols.

```
TYPE E_MP_DisplayModeIconWarning :
(
  MPBus_DisplayModeIconWarning_Invisible := 0,
  MPBus_DisplayModeIconWarning_Icon     := 1,
  MPBus_DisplayModeIconWarning_DeviceErrorState := 2
);
END_TYPE
```

MPBus_DisplayModelconWarning_Invisible: Unsichtbar.

MPBus_DisplayModelconWarning_Icon: Symbol.

MPBus_DisplayModelconWarning_DeviceErrorState: Fehlerstatus vom Gerät.

6.4.1.7 E_MP_DisplayModelconWindow

Anzeigemodus des Fenstersymbols.

```
TYPE E_MP_DisplayModeIconWindow :
(
  MPBus_DisplayModeIconWindow_Invisible := 0,
  MPBus_DisplayModeIconWindow_Icon     := 1,
  MPBus_DisplayModeIconWindow_Reserve   := 2
);
END_TYPE
```

MPBus_DisplayModelconWindow_Invisible: Unsichtbar.

MPBus_DisplayModelconWindow_Icon: Symbol.

MPBus_DisplayModelconWindow_Reserve: Reserve.

6.4.1.8 E_MP_DisplayModeTemp

Anzeigemodus Temperatur.

```
TYPE E_MP_DisplayModeTemp :
(
  MPBus_DisplayModeTemp_Invisible           := 0,
  MPBus_DisplayModeTemp_ActualRoomTemp     := 1,
  MPBus_DisplayModeTemp_RoomTempSetpoint   := 2
);
END_TYPE
```

MPBus_DisplayModeTemp_Invisible: Unsichtbar.

MPBus_DisplayModeTemp_ActualRoomTemp: Aktuelle Raumtemperatur.

MPBus_DisplayModeTemp_RoomTempSetpoint: Sollwert der Raumtemperatur.

6.4.1.9 E_MP_DisplayModeTempUnit

Anzeigemodus der Einheit für die Temperatur.

```
TYPE E_MP_DisplayModeTempUnit :
(
  MPBus_DisplayModeTempUnit_C           := 0,
  MPBus_DisplayModeTempUnit_Reserve     := 1,
  MPBus_DisplayModeTempUnit_F           := 2
);
END_TYPE
```

MPBus_DisplayModeTempUnit_C: °C.

MPBus_DisplayModeTempUnit_Reserve: Reserve.

MPBus_DisplayModeTempUnit_F: °F.

6.4.1.10 E_MP_DisplayModeVentilationStage

Anzahl der auf dem Display angezeigten einstellbaren Lüftungsstufen.

```
TYPE E_MP_DisplayModeVentilationStage :
(
  MPBus_DisplayModeVentilationStage_Reserve1 := 0,
  MPBus_DisplayModeVentilationStage_Reserve2 := 1,
  MPBus_DisplayModeVentilationStage_3       := 2,
  MPBus_DisplayModeVentilationStage_4       := 3,
  MPBus_DisplayModeVentilationStage_7       := 4
);
END_TYPE
```

MPBus_DisplayModeVentilationStage_Reserve1: Reserve 1.

MPBus_DisplayModeVentilationStage_Reserve2: Reserve 2.

MPBus_DisplayModeVentilationStage_3: 3 Lüftungsstufen.

MPBus_DisplayModeVentilationStage_4: 4 Lüftungsstufen.

MPBus_DisplayModeVentilationStage_7: 7 Lüftungsstufen.

6.4.1.11 E_MP_DisplayVisibility

Sichtbarkeit auf dem Display.

```
TYPE E_MP_DisplayVisibility :  
(  
  MPBus_DisplayVisibility_Invisible := 0,  
  MPBus_DisplayVisibility_Visible  := 1  
) ;  
END_TYPE
```

MPBus_DisplayVisibility_Invisible: Unsichtbar.

MPBus_DisplayVisibility_Visible: Sichtbar.

6.4.1.12 E_MP_EnabledStatus

Aktivierungsstatus.

```
TYPE E_MP_EnabledStatus :  
(  
  MPBus_EnabledStatus_Disabled := 0,  
  MPBus_EnabledStatus_Enabled  := 1  
) ;  
END_TYPE
```

MPBus_EnabledStatus_Disabled: Deaktiviert.

MPBus_EnabledStatus_Enabled: Aktiviert.

6.4.1.13 E_MP_EP_R_R6_UnitSel

Skalierung.

```
TYPE E_MP_EP_R_R6_UnitSel :  
(  
  E_MP_m3_s := 0,  
  E_MP_m3_h := 1,  
  E_MP_l_s  := 2,  
  E_MP_l_min := 3,  
  E_MP_l_h  := 4,  
  E_MP_gpm  := 5,  
  E_MP_cfm  := 6  
) ;  
END_TYPE
```

E_MP_m3_s: Setzt Skalierung auf m3/s.

E_MP_m3_h: Setzt Skalierung auf m3/h.

E_MP_l_s: Setzt Skalierung auf l/s.

E_MP_l_min: Setzt Skalierung auf l/min.

E_MP_l_h: Setzt Skalierung auf l/h.

E_MP_gpm: Setzt Skalierung auf gpm.

E_MP_cfm: Setzt Skalierung auf cfm.

6.4.1.14 E_MP_EV_V4_Command

Befehl für Service- und Testfunktionen des Antriebs.

```
TYPE E_MP_EV_V4_Command :
(
  MPBus_EV_Command_None := 0,
  MPBus_EV_Command_Sync := 2
);
END_TYPE
```

MPBus_EV_Command_None: Kein.

MPBus_EV_Command_Sync: Sync.

6.4.1.15 E_MP_EV_V4_ControlMode

Steuerungsmodus.

```
TYPE E_MP_EV_V4_ControlMode :
(
  MPBus_EV_PositionControl := 0,
  MPBus_EV_FlowControl     := 1,
  MPBus_EV_PowerControl    := 2
);
END_TYPE
```

MPBus_EV_PositionControl: Positionssteuerung.

MPBus_EV_FlowControl: Durchflusssteuerung.

MPBus_EV_PowerControl: Leistungssteuerung.

6.4.1.16 E_MP_EV_V4_DeltaTLimitation

Reaktion auf ein niedriges Delta T.

```
TYPE E_MP_EV_V4_DeltaTLimitation :
(
  MPBus_EV_Disabled           := 0,
  MPBus_EV_DeltaT_Manager    := 1,
  MPBus_EV_DeltaT_Manager_Scaled := 2
);
END_TYPE
```

MPBus_EV_Disabled: Deaktiviert.

MPBus_EV_DeltaT_Manager: Delta T Manager.

MPBus_EV_DeltaT_Manager_Scaled: Delta T Manager skaliert.

6.4.1.17 E_MP_EV_V4_DeltaTManagerStatus

Status vom Delta T Manager.

```
TYPE E_MP_EV_V4_DeltaTManagerStatus :
(
  MPBus_EV_NotSelect := 0,
  MPBus_EV_Standby   := 1,
  MPBus_EV_Active    := 2
);
END_TYPE
```

MPBus_EV_NotSelect: Nicht ausgewählt.

MPBus_EV_Standby: Standby.

MPBus_EV_Active: Aktiv.

6.4.1.18 E_MP_EV_V4_OverrideControl

Überschreiben des Sollwerts.

```
TYPE E_MP_EV_V4_OverrideControl :  
(  
  MPBus_EV_Override_None      := 0,  
  MPBus_EV_Override_Open     := 1,  
  MPBus_EV_Override_Close    := 2,  
  MPBus_EV_Override_MinFlow  := 3,  
  MPBus_EV_Override_MaxFlow  := 5,  
  MPBus_EV_Override_NomFlow  := 6,  
  MPBus_EV_Override_Motor_Stop := 10  
);  
END_TYPE
```

MPBus_EV_Override_None: Kein.

MPBus_EV_Override_Open: Öffnen.

MPBus_EV_Override_Close: Schließen.

MPBus_EV_Override_MinFlow: Minimaler Durchfluss.

MPBus_EV_Override_MaxFlow: Maximaler Durchfluss.

MPBus_EV_Override_NomFlow: Nominaler Durchfluss.

MPBus_EV_Override_Motor_Stop: Motorstopp.

6.4.1.19 E_MP_EV_V4_Sensor1Type

Externer Sensor an Eingang S1.

```
TYPE E_MP_EV_V4_Sensor1Type :  
(  
  MPBus_EV_Sensor_None      := 0,  
  MPBus_EV_Sensor_Active   := 1,  
  MPBus_EV_Sensor_Passive  := 3,  
  MPBus_EV_Sensor_Switch   := 4  
);  
END_TYPE
```

MPBus_EV_Sensor_None: Keiner.

MPBus_EV_Sensor_Active: Aktiv.

MPBus_EV_Sensor_Passive: Passiv.

MPBus_EV_Sensor_Switch: Switch.

6.4.1.20 E_MP_SystemOperationMode

Betriebsart des Systems.

```
TYPE E_MP_SystemOperationMode :  
(  
  MPBus_SystemOperationMode_OffProtection := 0,  
  MPBus_SystemOperationMode_OnComfort    := 1,  
  MPBus_SystemOperationMode_Eco         := 2,  
  MPBus_SystemOperationMode_Boost       := 3  
);  
END_TYPE
```

MPBus_SystemOperationMode_OffProtection: Aus/Schutz.

MPBus_SystemOperationMode_OnComfort: Ein/Komfort.

MPBus_SystemOperationMode_Eco: Eco-Modus.

MPBus_SystemOperationMode_Boost: Boost-Modus.

6.4.1.21 E_MP_VRU_Application

Visualisierung der vom Hersteller gewählten Anwendung.

```
TYPE E_MP_VRU_Application :
(
  MPBus_VRU_Application_FlowControl      := 0,
  MPBus_VRU_Application_PressureControl  := 1,
  MPBus_VRU_Application_RoomPressureControl := 2,
  MPBus_VRU_Application_FlowMeasurement := 4
);
END_TYPE
```

MPBus_VRU_Application_FlowControl: Durchflussregelung.

MPBus_VRU_Application_PressureControl: Druckregelung.

MPBus_VRU_Application_RoomPressureControl: Raumdruckregelung.

MPBus_VRU_Application_FlowMeasurement: Durchflussmessung.

6.4.1.22 E_MP_VRU_Command

Kommandos für Service- und Testfunktionen des Antriebs.

```
TYPE E_MP_VRU_Command :
(
  MPBus_VRU_Command_None      := 0,
  MPBus_VRU_Command_Adaption  := 1,
  MPBus_VRU_Command_Test      := 2,
  MPBus_VRU_Command_Sync      := 3
);
END_TYPE
```

MPBus_VRU_Command_None: Keines.

MPBus_VRU_Command_Adaption: Adaption.

MPBus_VRU_Command_Test: Test.

MPBus_VRU_Command_Sync: Sync.

6.4.1.23 E_MP_VRU_OverrideControl

Überschreiben des Sollwerts.

```
TYPE E_MP_VRU_OverrideControl :
(
  MPBus_VRU_Override_None      := 0,
  MPBus_VRU_Override_Open      := 1,
  MPBus_VRU_Override_Close     := 2,
  MPBus_VRU_Override_Max       := 3,
  MPBus_VRU_Override_Min       := 4,
  MPBus_VRU_Override_Reserve   := 5,
  MPBus_VRU_Override_Motor_Stop := 6,
  MPBus_VRU_Override_Vnom_Pnom := 7
);
END_TYPE
```

MPBus_VRU_Override_None: Keine.

MPBus_VRU_Override_Open: Öffnen.

MPBus_VRU_Override_Close: Schließen.

MPBus_VRU_Override_Max: Maximum.

MPBus_VRU_Override_Min: Minimum.

MPBus_VRU_Override_Reserve: Reserve.

MPBus_VRU_Override_Motor_Stop: Motorstopp.

MPBus_VRU_Override_Vnom_Pnom: Vnom / Pnom.

6.4.1.24 E_MP_VRU_RoomPressureCascade

Raumdruck-Kaskadenregelung.

```
TYPE E_MP_VRU_RoomPressureCascade :
(
  MPBus_VRU_RoomPressureCascade_Disabled := 0,
  MPBus_VRU_RoomPressureCascade_Enabled := 1,
  MPBus_VRU_RoomPressureCascade_EnabledFast := 2
);
END_TYPE
```

MPBus_VRU_RoomPressureCascade_Disabled: Raumdruck-Kaskadenregelung deaktiviert.

MPBus_VRU_RoomPressureCascade_Enabled: Raumdruck-Kaskadenregelung aktiviert.

MPBus_VRU_RoomPressureCascade_EnabledFast: Raumdruck-Kaskadenregelung aktiviert (schnell).

6.4.1.25 E_MP_VRU_Sensor1Type

Externer Sensor an Eingang S1.

```
TYPE E_MP_VRU_Sensor1Type :
(
  MPBus_VRU_Sensor_None := 0,
  MPBus_VRU_Sensor_Active := 1,
  MPBus_VRU_Sensor_Passive := 2,
  MPBus_VRU_Sensor_Switch := 4
);
END_TYPE
```

MPBus_VRU_Sensor_None: Keiner.

MPBus_VRU_Sensor_Active: Aktiv.

MPBus_VRU_Sensor_Passive: Passiv.

MPBus_VRU_Sensor_Switch: Switch.

6.4.1.26 E_MPBus_ControlMode

Regelmodus.

```
TYPE E_MPBus_ControlMode :
(
  MPBus_ControlMode_PosCtrl := 0,
  MPBus_ControlMode_FlowCtrl := 1,
  MPBus_ControlMode_PowerCtrl := 2,
  MPBus_ControlMode_Disable := 16#FF
);
END_TYPE
```

MPBus_ControlMode_PosCtrl: Regelt nach Position.

MPBus_ControlMode_FlowCtrl: Regelt nach Durchflussmenge.

MPBus_ControlMode_PowerCtrl: Regelt nach Energiemenge.

MPBus_ControlMode_Disable: Deaktiviert.

6.4.1.27 E_MPBus_DeltaTLimitation

Delta T (dT) Limitierung. Details sind der Dokumentation von Belimo Energy Valves zu entnehmen.

```
TYPE E_MPBus_DeltaTLimitation :
(
  MPBus_DeltaTLimitation_Disable := 0,
  MPBus_DeltaTLimitation_dT_Manager := 1,
);
```

```
MPBus_DeltaTLimitation_dT_ManagerScal := 2
);
END_TYPE
```

MPBus_DeltaTLimitation_Disable: dT deaktiviert.

MPBus_DeltaTLimitation_dT_Manager: Einfache dT Limitierung.

MPBus_DeltaTLimitation_dT_ManagerScal: Erweiterte dT Limitierung.

6.4.1.28 E_MPBus_Override

Überschreiben Modus.

```
TYPE E_MPBus_Override :
(
  MPBus_Override_None := 0,
  MPBus_Override_Auto := 1,
  MPBus_Override_Close := 2,
  MPBus_Override_Open := 3,
  MPBus_Override_Vnom := 4,
  MPBus_Override_Vmax := 5,
  MPBus_Override_Stop := 6,
  MPBus_Override_Pnom := 7,
  MPBus_Override_Pmax := 8
);
END_TYPE
```

6.4.1.29 E_MPBus_Override_6wayMPIV

Überschreiben Regelmodus.

```
TYPE E_MPBus_Override_6wayMPIV :
(
  MPBus_6wayMPIV_None := 0,
  MPBus_6wayMPIV_Seq1Open := 1,
  MPBus_6wayMPIV_Seq2Open := 2,
  MPBus_6wayMPIV_Close := 3,
  MPBus_6wayMPIV_Seq1Vmax := 4,
  MPBus_6wayMPIV_Seq2Vmax := 5
);
END_TYPE
```

6.4.1.30 MP_ERROR

Fehlermeldungen der Bibliothek.

```
TYPE MP_ERROR :
(
  NO_MP_ERROR := 0,
  WRONG_TERMINAL := 1,
  NO_ANSWER_FROM_KL6771 := 2,
  NO_LINK_TO_STRUCTURE_strDataKL6771 := 3,
  WRONG_MP_ADDRESS_IS_0 := 10,
  WRONG_MP_ADDRESS := 11,
  WRONG_SET_POINT := 21,
  MP_BUS_TIMEOUT_NO_ANSWER_FROM_SLAVE := 25,
  MP_BUS_SETPOINT_DIFF_TOO_HIGH := 26,
  KL6771_TIME_OUT := 31,
  MP_ADDRESS_IS_IN_USE := 32,
  MP_DISABLED := 33,
  MP_BUS_ERROR := 87,
  MP_NO_ANSWER_ON_EVENT := 88,
  MP_NO_ANSWER := 89,
  MP_COM_BREAK := 90,
  MP_LENGTH_PARITY_ERROR := 98,
  MP_CROSS_PARITY_ERROR := 99,
  MP_MASTER_CONFLICT_ERROR := 101,
  MP_GAP_TIMEOUT_ERROR := 102,
  MP_NO_ANSWER_SLAVE := 103,
  MP_ANSWER_ERROR_FLAG := 110,
  MP_ANSWER_WRONG_LEN := 111,
  MP_ANSWER_WRONG_TELEG := 112,
  MP_ANSWER_WITH_ERROR := 115,
  MP_ERROR_WrongDeviceFamily := 200,
```

```

MP_CONF_ERROR_CO2LimitGood           := 300,
MP_CONF_ERROR_CO2LimitModerate       := 301,
MP_CONF_ERROR_CO2Limit                := 302,
MP_CONF_ERROR_WRONG_TEMP_OFFSET      := 303,
MP_CONF_ERROR_WRONG_HUMIDITY_OFFSET  := 304,
MP_CONF_ERROR_WRONG_CO2_OFFSET       := 305,
MP_CONF_ERROR_WRONG_TempSetpoint      := 306,
MP_CONF_ERROR_WRONG_RelativeTempSetpoint := 307,
MP_CONF_ERROR_WRONG_DefaultTempSetpoint := 308,
MP_CONF_ERROR_WRONG_TempSetpointRange := 309,
MP_ANSWER_Reserve                     := 16#800A,
MP_ANSWER_UnknowCommand               := 16#800B,
MP_ANSWER_WrongOrNoPassword           := 16#800C,
MP_ANSWER_CommandExecution            := 16#800D,
MP_ANSWER_ParameterError              := 16#800E,
MP_ANSWER_UnknowId                    := 16#800F,
MP_ANSWER_SizeMismatch                := 16#8010,
MP_ANSWER_IllegalBlockNr              := 16#8011,
MP_ANSWER_InternalBusBusy             := 16#8012,
MP_ANSWER_ReservedForFuture           := 16#80FF
)
END_TYPE

```

NO_MP_ERROR: Kein Fehler.

WRONG_TERMINAL: Falsche Klemme angeschlossen.

NO_ANSWER_FROM_KL6771: Keine Antwort von der KL6771. Üblicherweise gibt es bei dieser Meldung keine Verbindung zur Klemme. Sind die E/A-Variablen der Klemme verknüpft? Klemme falsch gesteckt? Alles bereinigt, alles übersetzt und neu eingelesen?

NO_LINK_TO_STRUCTURE_strDataKL6771: Link zur Struktur DataKL6771 prüfen.

WRONG_MP_ADDRESS_IS_0: MP-Bus-Adresse ist 0. Es sind nur Adressen von 1-8 erlaubt.

WRONG_MP_ADDRESS: MP-Bus-Adresse ist >8. Es sind nur Adressen von 1-8 erlaubt.

WRONG_SET_POINT: Falscher Setpoint.

MP_BUS_TIMEOUT_NO_ANSWER_FROM_SLAVE: MP-Bus Timeout, keine Antwort vom Slave.

MP_BUS_SETPOINT_DIFF_TOO_HIGH: Differenz vom Setpoint zu groß.

KL6771_TIME_OUT: KL6771 Timeout.

MP_ADDRESS_IS_IN_USE: MP-Bus-Adresse ist im Gebrauch.

MP_DISABLED: MP-Bus gesperrt.

MP_BUS_ERROR: MP-Bus Fehler.

MP_NO_ANSWER_ON_EVENT: MP-Bus keine Antwort auf ein Ereignis.

MP_NO_ANSWER: MP-Bus keine Antwort.

MP_COM_BREAK: MP-Bus Kommunikationsabbruch.

MP_LENGTH_PARITY_ERROR: MP-Bus length parity Fehler.

MP_CROSS_PARITY_ERROR: MP-Bus cross parity Fehler.

MP_MASTER_CONFLICT_ERROR: MP-Bus MASTER_CONFLICT_ERROR.

MP_GAP_TIMEOUT_ERROR: MP-Bus GAP Timeout.

MP_NO_ANSWER_SLAVE: MP-Bus keine Antwort vom Slave.

MP_ANSWER_ERROR_FLAG: MP-Bus Fehlerbit im Antworttelegramm ist gesetzt.

MP_ANSWER_WRONG_LEN: MP-Bus falsche Telegrammlänge.

MP_ANSWER_WRONG_TELEG: MP-Bus falsches Telegramm empfangen.

MP_ANSWER_WITH_ERROR: MP-Bus Antwort enthält einen Fehler.

MP_ERROR_WrongDeviceFamily: Falsche Gerätefamilie.

MP_CONF_ERROR_CO2LimitGood: Konfigurationsfehler für CO2 Begrenzung "Good".

MP_CONF_ERROR_CO2LimitModerate: Konfigurationsfehler für CO2 Begrenzung "Moderate".

MP_CONF_ERROR_CO2Limit: Konfigurationsfehler für CO2 Begrenzung.

MP_CONF_ERROR_WRONG_TEMP_OFFSET: Konfigurationsfehler für Temperaturabweichung.

MP_CONF_ERROR_WRONG_HUMIDITY_OFFSET: Konfigurationsfehler für Feuchtigkeitsabweichung.

MP_CONF_ERROR_WRONG_CO2_OFFSET: Konfigurationsfehler für CO2 Abweichung.

MP_CONF_ERROR_WRONG_TempSetpoint: Konfigurationsfehler für Temperatursollwert.

MP_CONF_ERROR_WRONG_RelativeTempSetpoint: Konfigurationsfehler für relativen Temperatursollwert.

MP_CONF_ERROR_WRONG_DefaultTempSetpoint: Konfigurationsfehler für voreingestellten Temperatursollwert.

MP_CONF_ERROR_WRONG_TempSetpointRange: Konfigurationsfehler für Temperatursollwertbereich.

MP_ANSWER_Reserve: Reserve.

MP_ANSWER_UnknowCommand: Unbekannter Befehl.

MP_ANSWER_WrongOrNoPassword: Kein oder falsches Passwort.

MP_ANSWER_CommandExecution: Ausführung des Befehls.

MP_ANSWER_ParameterError: Parameterfehler.

MP_ANSWER_UnknowId: Unbekannte Id.

MP_ANSWER_SizeMismatch: Größe stimmt nicht überein.

MP_ANSWER_IllegalBlockNr: Ungültige Blocknummer.

MP_ANSWER_InternalBusBusy: Interner Bus ist belegt.

MP_ANSWER_ReservedForFuture: Reserve.

6.4.1.31 UST3_Ex

Skalierung der Spannung.

```
TYPE UST3_Ex :
(
  Ex_1mV      := 0,
  Ex_250uV    := 1,
)
END_TYPE
```

Ex_1mV: Skalierung 0...11 V.

Ex_250uV: Skalierung 0...3 V.

6.4.1.32 UST3_R_set

Skalierung des Widerstandes.

```
TYPE UST3_R_set :
(
  R_10hm      := 0,
  R_250mOhm   := 1,
  R_40hm      := 2,
)
END_TYPE
```

R_10hm: Skalierung 0...20 kOhm.

R_250mOhm: Skalierung 0...5 kOhm.

R_4Ohm: Skalierung 0...262 kOhm.

6.4.2 Strukturen

6.4.2.1 DataKL6771

Verbindet die Sende- und Empfangs-Bausteine mit dem Baustein [KL6771\(\)](#) [[▶ 27](#)].

```
TYPE DataKL6771 :
STRUCT
  OrderNumber : BYTE;
  ReciveData  : BOOL;
  SendData    : BOOL;
  Error       : BOOL;
  ErrorID     : MP_Error;
  pNumber     : DWORD;
END_STRUCT
END_TYPE
```

OrderNumber: Internes Byte.

ReciveData: Daten werden empfangen.

SendData: Daten werden gesendet.

Error: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *ErrorID* beschrieben.

ErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [MP_ERROR](#) [[▶ 82](#)]). Gleichzeitig wird *Error* TRUE.

pNumber: Interner Pointer.

6.4.2.2 MP_BUS_MPX_ERROR

Fehlermeldungen der "MPX" Sensoren (Funktionsbaustein [MP_MPX](#) [[▶ 45](#)]).

```
TYPE MP_BUS_MPX_ERROR :
STRUCT
  MP_BUS_MPX_TempSensorErr : BOOL;
  MP_BUS_MPX_HumiditySensorErr : BOOL;
  MP_BUS_MPX_CO2SensorErr : BOOL;
  MP_BUS_MPX_VocSensorErr : BOOL;
END_STRUCT
END_TYPE
```

MP_BUS_MPX_TempSensorErr: Der Temperatursensor ist fehlerhaft.

MP_BUS_MPX_HumiditySensorErr: Der Feuchtigkeitssensor ist fehlerhaft.

MP_BUS_MPX_CO2SensorErr: Der CO2 Sensor ist fehlerhaft.

MP_BUS_MPX_VocSensorErr: Der VOC Sensor ist fehlerhaft.

6.4.2.3 MP_Serial_Number

Seriennummer des Gerätes.

```
TYPE MP_Serial_Number :
STRUCT
  YearAndWeek : WORD;
  DayAndNumber : WORD;
  DeviceFamily : BYTE;
  TestStation : BYTE;
  FamilySuffix : BYTE;
END_STRUCT
END_TYPE
```

YearAndWeek: Jahr und Woche.

DayAndNumber: Tag und Nummer.

DeviceFamily: Gerätefamilie.

TestStation: Prüfstation.

FamilySuffix: Gerätefamilie Suffix.

6.4.2.4 St_MP_EV_V4_MalfunctionServiceInfo

Störungs- und Serviceinformationen.

```

TYPE St_MP_EV_V4_MalfunctionServiceInfo :
STRUCT
  bNoCommunicationToActuator : BOOL;
  bGearDisengaged           : BOOL;
  bActuatorCannotMove       : BOOL;
  bReverseFlow               : BOOL;
  bFlowSetpointNotReached    : BOOL;
  bFlowWithClosedValve       : BOOL;
  bActualFlowVnom            : BOOL;
  bFlowMeasurementError      : BOOL;
  bRemoteTempError           : BOOL;
  bIntegratedTempError        : BOOL;
  bCommToSensorInterrupted   : BOOL;
  bFreezeWarning             : BOOL;
  bGlycolDetected            : BOOL;
  bPowerSetpointNotReached    : BOOL;
END_STRUCT
END_TYPE
    
```

bNoCommunicationToActuator: Keine Kommunikation zum Aktor.

bGearDisengaged: Getriebe ausgekuppelt.

bActuatorCannotMove: Aktor kann sich nicht bewegen.

bReverseFlow: Umgekehrter Durchfluss.

bFlowSetpointNotReached: Sollwert für Durchfluss nicht erreicht.

bFlowWithClosedValve: Durchfluss mit geschlossenem Ventil.

bActualFlowVnom: Aktueller Durchfluss > Nennvolumenstrom.

bFlowMeasurementError: Fehler bei der Durchflussmessung.

bRemoteTempError: Fehler bei remote Temperatur.

bIntegratedTempError: Fehler bei integrierter Temperatur.

bCommToSensorInterrupted: Kommunikation zum Sensor ist unterbrochen.

bFreezeWarning: Frostwarnung.

bGlycolDetected: Glykol erkannt.

bPowerSetpointNotReached: Leistungssollwert nicht erreicht.

6.4.2.5 St_MP_VRU_ServiceInfo

Störungs- und Serviceinformationen.

```

TYPE St_MP_VRU_ServiceInfo :
STRUCT
  bError_dP_Sensor           : BOOL;
  bReverseAirflowDetected    : BOOL;
  bAirflowNotReached         : BOOL;
  bFlowInClosedPosition      : BOOL;
  bInternalActivity           : BOOL;
  bGearDisengaged            : BOOL;
    
```

```

bBusWatchdogTriggered      : BOOL;
bActuatorDoseNotFitToApplication : BOOL;
bPressSensorWrongConnected : BOOL;
bPressureSensorNotReached   : BOOL;
bError_dP_SensorOutOfRange  : BOOL;
END_STRUCT
END_TYPE

```

bError_dP_Sensor: Fehler dp-Sensor.

bReverseAirflowDetected: Umgekehrter Luftstrom erkannt.

bAirflowNotReached: Luftstrom nicht erreicht.

bFlowInClosedPosition: Durchfluss in geschlossener Position.

bInternalActivity: Interne Aktivität.

bGearDisengaged: Getriebe ausgekuppelt.

bBusWatchdogTriggered: Bus-Watchdog ausgelöst.

bActuatorDoseNotFitToApplication: Aktuator passt nicht zur Anwendung.

bPressSensorWrongConnected: Drucksensor falsch angeschlossen.

bPressureSensorNotReached: Drucksensor nicht erreicht.

bError_dP_SensorOutOfRange: dP-Sensor außerhalb des Bereichs.

6.4.2.6 St_StateEV

Auskunft über den Zustand vom EV.

```

TYPE St_StateEV :
STRUCT
  bFlow_with_closed_valve : BOOL;
  bAir_bubbles            : BOOL;
  bFlow_not_reached       : BOOL;
  bPower_not_realized     : BOOL;
  bGear_disengaged        : BOOL;
END_STRUCT
END_TYPE

```

bFlow_with_closed_valve: Durchfluss bei geschlossenem Ventil. Ventil überprüfen.

bAir_bubbles: Zu viele Luftblasen (System ungenügend entlüftet). Durchflussmessung nicht mehr exakt. EV wechselt von Volumenstromregelung auf Positionsregelung.

bFlow_not_reached: Durchfluss nicht erreicht, obwohl der Kugelhahn vollständig geöffnet ist. Hydraulik prüfen, Pumpe einschalten oder Pumpendruck erhöhen.

bPower_not_realized: Leistung nicht erreicht. Das EV kann neben der Positionsregelung und Volumenstromregelung direkt eine Leistung zuführen. Primärseite liefert zu wenig Durchfluss oder dT.

bGear_disengaged: Getriebeausrüstung aktiv. Handverstellung vor Ort möglich. Bei anderen MP-Antrieben kann diese Information mit MP_Get_State gelesen werden.

6.4.2.7 UST3_SET

Datenstruktur zum Einstellen der Skalierung und der Widerstandsmessung.

```

TYPE UST3_SET :
STRUCT
  E1 : UST3_Ex;
  E2 : UST3_Ex;
  E3 : UST3_Ex;
  R_SET : UST3_R_set;
  R_ON_OFF : BOOL;
END_STRUCT
END_TYPE

```

E1: Parameter U1 (siehe [UST3_Ex \[▶ 84\]](#)).

E2: Parameter U2 (siehe [UST3_Ex \[▶ 84\]](#)).

E3: Parameter U3 (siehe [UST3_Ex \[▶ 84\]](#)).

R_SET: Parameter Y (siehe [UST3_R_set \[▶ 84\]](#)).

R_ON_OFF: Messung R oder U.

6.5 Fehlercodes

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x0000	0	NO_MP_ERROR	Kein Fehler.
0x0001	1	WRONG_TERMINAL	Falsche Klemme angeschlossen.
0x0002	2	NO_ANSWER_FROM_KL6771	Keine Antwort der MP-Bus-Masterklemme KL6771.
0x0003	3	NO_LINK_TO_STRUCTURE_strDataKL6771	Link zur Struktur DataKL6771 [▶ 85] prüfen.
0x000A	10	WRONG_MP_ADDRESS_IS_0	MP-Bus Adresse ist 0. Es sind nur Adressen von 1-8 erlaubt.
0x000B	11	WRONG_MP_ADDRESS	MP-Bus Adresse ist >8. Es sind nur Adressen von 1-8 erlaubt.
0x0015	21	WRONG_SET_POINT	Falscher Setpoint.
0x0019	25	MP_BUS_TIMEOUT_NO_ANSWER_FROM_SLAVE	MP-Bus Timeout, keine Antwort vom Slave.
0x0020	26	MP_BUS_SETPOINT_DIFF_TOO_HIGH	Differenz vom Setpoint zu groß.
0x001F	31	KL6771_TIME_OUT	KL6771 Timeout.
0x0020	32	MP_ADDRESS_IS_IN_USE	MP-Bus Adresse ist im Gebrauch.
0x0021	33	MP_DISABLED	MP-Bus gesperrt.
0x0057	87	MP_BUS_ERROR	MP-Bus Fehler.
0x0058	88	MP_NO_ANSWER_ON_EVENT	MP-Bus keine Antwort auf ein Ereignis.
0x0059	89	MP_NO_ANSWER	MP-Bus keine Antwort.
0x005A	90	MP_COM_BREAK	MP-Bus Kommunikationsabbruch.
0x0062	98	MP_LENGTH_PARITY_ERROR	MP-Bus length parity Fehler.
0x0063	99	MP_CROSS_PARITY_ERROR	MP-Bus cross parity Fehler.
0x0065	101	MP_MASTER_CONFLICT_ERROR	MP-Bus MASTER_CONFLICT_ERROR.
0x0066	102	MP_GAP_TIMEOUT_ERROR	MP-Bus GAP Timeout.
0x0067	103	MP_NO_ANSWER_SLAVE	MP-Bus keine Antwort vom Slave.
0x006E	110	MP_ANSWER_ERROR_FLAG	MP-Bus Fehlerbit im Antworttelegramm ist gesetzt.
0x006F	111	MP_ANSWER_WRONG_LEN	MP-Bus falsche Telegrammlänge.
0x0070	112	MP_ANSWER_WRONG_TELEG	MP-Bus falsches Telegramm empfangen.
0x0073	115	MP_ANSWER_WITH_ERROR	Antwort enthält einen Fehler.
0x00C8	200	MP_ERROR_WrongDeviceFamily	Falsche Gerätefamilie.
0x012C	300	MP_CONF_ERROR_CO2LimitGood	Konfigurationsfehler für CO2 Begrenzung "Good".
0x012D	301	MP_CONF_ERROR_CO2LimitModerate	Konfigurationsfehler für CO2 Begrenzung "Moderate".
0x012E	302	MP_CONF_ERROR_CO2Limit	Konfigurationsfehler für CO2 Begrenzung.

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x012F	303	MP_CONF_ERROR_WRONG_TEMP_OFFSET	Konfigurationsfehler für Temperaturabweichung.
0x0130	304	MP_CONF_ERROR_WRONG_HUMIDITY_OFFSET	Konfigurationsfehler für Feuchtigkeitsabweichung.
0x0131	305	MP_CONF_ERROR_WRONG_CO2_OFFSET	Konfigurationsfehler für CO2 Abweichung.
0x0132	306	MP_CONF_ERROR_WRONG_TempSetpoint	Konfigurationsfehler für Temperatursollwert.
0x0133	307	MP_CONF_ERROR_WRONG_RelativeTempSetpoint	Konfigurationsfehler für relativen Temperatursollwert.
0x0134	308	MP_CONF_ERROR_WRONG_DefaultTempSetpoint	Konfigurationsfehler für voreingestellten Temperatursollwert.
0x0135	309	MP_CONF_ERROR_WRONG_TempSetpointRange	Konfigurationsfehler für Temperatursollwertbereich.
0x800A	32778	MP_ANSWER_Reserve	Reserve.
0x800B	32779	MP_ANSWER_UnknowCommand	Unbekannter Befehl.
0x800C	32780	MP_ANSWER_WrongOrNoPassword	Kein oder falsches Passwort.
0x800D	32781	MP_ANSWER_CommandExecution	Ausführung des Befehls.
0x800E	32782	MP_ANSWER_ParameterError	Parameterfehler.
0x800F	32783	MP_ANSWER_UnknowId	Unbekannte Id.
0x8010	32784	MP_ANSWER_SizeMismatch	Größe stimmt nicht überein.
0x8011	32785	MP_ANSWER_IllegalBlockNr	Ungültige Blocknummer.
0x8012	32786	MP_ANSWER_InternalBusBusy	Interner Bus ist belegt.
0x80FF	33023	MP_ANSWER_ReservedForFuture	Reserve.

7 Anhang

7.1 Beispiele

Voraussetzungen

Beispiel	Beschreibung
https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063689867.zip	TwinCAT-PLC-Projekt für die KL6771.
https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063691275.zip	TwinCAT-PLC-Projekt für die KL6771 (TwinCAT PLC Control).
https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063692683.zip	TwinCAT-PLC-Projekt für die KL6771 (TwinCAT System Manager).
https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063694091.zip	Manuelle Zuweisung der MP-Bus Adresse (TwinCAT PLC Control).
https://infosys.beckhoff.com/content/1031/tcplclibmpbus/Resources/12063695499.zip	Manuelle Zuweisung der MP-Bus Adresse (TwinCAT System Manager).

7.2 Support und Service

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

Downloadfinder

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

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

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

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

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

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

Hotline: +49 5246 963-157
E-Mail: support@beckhoff.com

Beckhoff Service

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

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

Hotline: +49 5246 963-460
E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.de/tx1200

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

