|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ICS ##.###.### | VDI-RICHTLINIEN | | Mai  2017 |  |
| Zu beziehen durch Beuth Verlag GmbH, 10772 Berlin – Alle Rechte vorbehalten © Verein Deutscher Ingenieure e.V., Düsseldorf 20## | Verein Deutscher Ingenieure | Automatisierungstechnisches Engineering modularer Anlagen in der Prozessindustrie -  Basisbibliothek | | VDI/VDE/NAMUR 2658  Blatt 3  Entwurf  Internes Arbeitspapier | Vervielfältigung – auch für innerbetriebliche Zwecke – nicht gestattet |
| Automation engineering of modular systems in the process industry – Basic Library | | Einsprüche bis 20##-##-##   * vorzugsweise über das VDI-Richtlinien-Einspruchsportal http://www.vdi.de/einspruchsportal * in Papierform an ##### ##### Postfach 10 11 39 40002 Düsseldorf | |
|  | | | |
| VDI-Gesellschaft ###################  Fachbereich ################## | | | |
| VDI-Handbuch ###################### | | | |

*Frühere Ausgabe: ##.##*

**Inhalt** Seite

Vorbemerkung 3

Einleitung 4

1. Anwendungsbereich 5

2. Abkürzungen 6

3. Grundkonzepte der Schnittstellendefinitionen 7

3.1 Richtlinien-Konformität 7

3.2 Beschreibung der Interpretation von Schnittstellendefinitionen 7

3.3 Datentypen 7

3.4 Default Values 8

4. Allgemeine Definitionen 9

4.1 TagName 9

4.2 TagDescription 9

4.3 OSLevel 9

4.4 WQC 10

4.5 ScaleSettings 11

4.6 UnitSettings 12

4.7 ValueLimitation 14

4.8 OperationModes 15

4.9 Handshake Verfahren 23

4.10 Interlocks 24

4.11 Feedback Monitoring 25

4.12 Reset 26

4.13 Limit Monitoring 27

5. Schnittstellendefinitionen 29

5.1 DataAssembly 30

5.2 Analogwertanzeigen 31

5.3 Analogoperationen 33

5.4 Digitalwertanzeigen 36

5.5 Digitaloperationen 39

5.6 Binäranzeigen 42

5.7 Binäroperation 44

5.8 Regelungen 47

5.9 Bistabile Ventile 49

5.10 Analoge Ventile 52

5.11 Bistabile Antriebe 56

5.12 Analoge Antriebe 59

5.13 Verriegelungsanzeige 64

5.14 Zeichenkettenanzeigen 69

Anhang 70

Vorbemerkung

Der Inhalt dieser Richtlinie ist entstanden unter Beachtung der Vorgaben und Empfehlungen der Richtlinie VDI 1000. Alle Rechte, insbesondere die des Nachdrucks, der Fotokopie, der elektronischen Verwendung und der Übersetzung, jeweils auszugsweise oder vollständig, sind vorbehalten.

Die Nutzung dieser VDI-Richtlinie ist unter Wahrung des Urheberrechts und unter Beachtung der Lizenzbedingungen (www.vdi-richtlinien.de), die in den VDI-Merkblättern geregelt sind, möglich.

An der Erarbeitung dieser VDI-Richtlinie waren beteiligt:

Andreas Stutz, Siemens AG Karlsruhe

Mathias Maurmaier, Siemens AG Karlsruhe

Mario Hoernicke, ABB Ladenburg

Katharina Stark, ABB Ladenburg

Christian Schäfer, Merck Darmstadt

Polyana Santos, Evonik Marl

Henry Bloch, HSU / Universität der Bundeswehr Hamburg

Leon Urbas, TU Dresden (Vorsitz)

Stephan Hensel, TU Dresden

Markus Graube, TU Dresden

Alexander Kehl, Festo Esslingen

Thomas Holm, Wago Minden

Simon Löpler, Wago Minden

Claus Vothknecht, PHOENIX CONTACT Electronics, Bad Pyrmont

Thilo Glas, PHOENIX CONTACT Electronics, Bad Pyrmont

Oleg Makarov, PHOENIX CONTACT Electronics, Bad Pyrmont

Roland Gauweiler, HIMA Brühl

Simon Kronemeier, BASF Ludwigshafen

Klaus Erni, Emerson Wesseling

Allen, die ehrenamtlich an der Erarbeitung dieser VDI-Richtlinie mitgewirkt haben, sei gedankt.

Eine Liste der aktuell verfügbaren Blätter dieser Richtlinienreihe ist im Internet abrufbar unter www.vdi.de/2658.

Einleitung

Der Fachausschuss „Zukünftige Architekturen der Automatisierungstechnik“ der VDI-Gesellschaft für Mess- und Automatisierungstechnik hat sich zusammen mit der NAMUR und dem ZVEI mit der Erstellung dieser Richtlinie der Aufgabe angenommen, die Spezifikation von Datenschnittstellen, zur Verwendung in modularen Anlagen, zu definieren und syntaktisch, semantisch und pragmatisch zu beschreiben.

Modulare Anlagen werden in der Fertigungs- und Verfahrenstechnik vermehrt eingesetzt. Das Ziel hierbei ist die Planungszeit neuer Anlagen deutlich zu verkürzen und Umbauarbeiten an Anlagen zeitlich zu verkürzen. Hierdurch reduziert sich die Stillstandzeit, bzw. die Time-To-Market wird bei Neuanlagen deutlich verkürzt.

Da die Domänen Fertigungstechnik und Verfahrenstechnik hierbei sehr unterschiedliche Anforderungen an die Modularität stellen, wird in dieser Richtlinie vornehmlich die Verfahrenstechnik betrachtet.

Ausgehend von abgeschlossenen Projekten, wie F3 Factory, und bestehenden Empfehlungen und Anforderungen, veröffentlicht in der NE148, an verfahrenstechnische Module wird in dieser Richtlinie das Engineering der Automatisierungstechnik modularer Anlagen beschrieben. Hierbei wird sowohl das Modulengineering, als auch das Anlagenengineering der Automatisierungstechnik betrachtet.

Zur Beschreibung der Modultypen wird das Module-Type-Package (MTP) verwendet, welches die Schnittstellen und Funktionen der Automatisierungstechnik von Modulen definiert, beschreibt und letztlich die Integration von Modulen in eine Prozessführungsebene (PFE) ermöglicht.

Hierbei werden folgende Aspekte in Blatt 3 der vorliegenden Richtlinie fokussiert:

1. Syntaktische und semantische Definition von Datenschnittstellen für Sensoren
2. Syntaktische und semantische Definition von Datenschnittstellen für Aktoren
3. Syntaktische und semantische Definition von Datenschnittstellen für Wertevorgaben
4. Syntaktische und semantische Definition von Datenschnittstellen für Interlock-Logiken

Weitere (geplante) Blätter der Richtlinie greifen folgende Aspekte des automatisierungstechnischen Engineerings modularer Anlagen auf:

* Blatt 1: Allgemeines Konzept und Schnittstellen
* Blatt 2: Modellierung von Bedienbildern
* Blatt 4: Modellierung von Moduldiensten
* Blatt 5: Laufzeit- und Kommunikationsaspekte
* Blatt 6: Konzept Alarmmanagement
* Hinzu kommen Blätter zu den Themen: Diagnose, Alarmmanagement, funktionale Sicherheit sowie Validieren von MTP und Modulen.

Durch die zunehmende Vernetzung der Module werden weitere Themen hinzukommen, wie z.B. modulübergreifende funktionale Sicherheit, sichere Kommunikation zwischen Modulen.

# Anwendungsbereich

Diese Richtlinie definiert Schnittstellen, die die auszutauschenden Daten eines Moduls definieren. Das vorliegende Richtliniendokument definiert für unterschiedliche Schnittstellenfamilien entsprechende Schnittstellen, die ausgehend von einer Basis-Schnittstelle weitere Erweiterungen mit Hilfe von Ableitungen enthalten.

Zielgruppen, Anwendungsfälle, Definitionen und Richtlinienfokus sind die gleichen wie in VDI/VDE/NAMUR 2658 Blatt 1 definiert.

# Abkürzungen

Neben den Abkürzungen aus VDI/VDE/NAMUR 2658 Blatt 1 und Blatt 2 werden keine Abkürzungen verwendet.

# Grundkonzepte der Schnittstellendefinitionen

Die in diesem Blatt definierten Schnittstellen stellen einen integralen Bestandteil des MTP Konzepts dar. Zur Integration von Modulen in eine PFE werden die in diesem Blatt syntaktisch und semantisch definierten Schnittstellen verwendet, die sowohl auf Modulseite als auch auf PFE Seite bekannt sein müssen. Im Folgenden werden die Allgemeinen Definitionen und die Schnittstellen näher erläutert.

## Richtlinien-Konformität

### Konformität für die Modulautomatisierung

Ein Modulautomatisierungssystem gilt als richtlinien-konform, wenn für alle Schnittstellenfamilien mindestens eine Schnittstellendefinition, unabhängig dessen Ableitungsstufe, realisierbar ist.

### Konformität für die Prozessführungssysteme

Ein Prozessführungssystem gilt als richtlinien-konform, wenn für alle Schnittstellenfamilien mindestens die Basistyp-Definition realisierbar ist.

## Beschreibung der Interpretation von Schnittstellendefinitionen

Jede Schnittstellendefinition besteht aus einem einleitenden Erklärungstext und einer detaillierten Tabelle mit einer Auflistung der für die Schnittstelle spezifizierten Informations-Variablen.

Kursiv gesetzte Variablen sind bereits in einer allgemeineren Definition vorhanden und werden in der entsprechenden Ableitung mit einer erweiterten Funktion belegt.

Im Folgenden werden die einzelnen Spalten der Tabellen erläutert:

|  |  |
| --- | --- |
| **Spalte** | **Definition** |
| Alias | Name des Informationselements |
| Access | Definition der Richtung des Informationsfluss  **PFE 🡨 MOD**  Information geht von Modul zur PFE (readonly)  **PFE 🡪 MOD**  Information geht von der PFE zum Modul (read/write)  **PFE 🡨🡪 MOD**  Informationselement ist bitgranular und beinhaltet Bits, die sowohl readonly als auch read/write sind. Das Element selbst muss hierfür als read/write deklariert werden. |
| DataType | Datentyp des Informationselements, siehe 3.3 Datentypen |
| Description | Beschreibung/Definition der Interpretation für das Informationselement. Hier können auch Hinweise zu Abhängigkeiten anderer Variablen der Schnittstelle beschrieben sein. |

## Datentypen

In der AML-Datei und der Steuerung werden teilweise verschiedene Datentypen für dieselbe Variable verwendet. Dies begründet sich darin, dass einige Datenformate nicht in AML unterstützt werden und deshalb durch andere, vergleichbare Datentypen ersetzt werden müssen. Die folgende Tabelle stellt die verwendeten Datentypen übersichtlich dar. In den weiteren Kapiteln dieser Dokumentation werden die Variablen mit den Datentypen der SPS deklariert.

|  |  |
| --- | --- |
| **SPS (Datentypen der IEC 61131-3)** | **AML** |
| REAL | xs:double |
| DWORD | xs:unsignedLong |
| DINT | xs:long |
| BYTE | xs:byte |
| BOOL | xs:boolean |
| INT | xs:integer |

## Default Values

Default Values werden im Merkmal *DefaultValue* des betreffenden Attributes der Schnittstellenvariable definiert und sind somit bereits mit dem MTP in der Planungsphase einer modularen Anlage verfügbar.

Das Merkmal *DefaultValue* muss mit dem Wert versehen werden, mit dem die PEA ausgeliefert wird. Wenn kein sinnvoller Default angegeben werden kann, beispielsweise für den Wert einer Prozessvariablen, ist dieses Merkmal leer zu lassen.

Wenn der Wert zur Laufzeit geändert werden kann, muss das Merkmal *AttributeDataType* den Wert xs:IDREF haben und das Merkmal *Value* verweist auf eine mindestens lesbare Variable (Link Mechanismus, definiert in Blatt 1). Bei unveränderlichen Werten müssen die Merkmale *Value* und *DefaultValue* auf den gleichen Wert gesetzt sein.

|  |
| --- |
| Grafik: Abbildung eines AML-Attribut aus einem Beispiel-MTP zur dynamischen Übertragung einer Temperatureinheit mit dem Offline-verfügbaren DefaultValue „1001“ für Grad Celsius. |
|  |

# Allgemeine Definitionen

Dieser Absatz definiert gewisse Bestandteile der Schnittstellen, die in mehreren Schnittstellenfamilien Anwendung finden und somit dokumentenweit Geltung haben.

## TagName

Die Variable TagName beinhaltet den Identifier für die hier realisierte PLT-Stelle innerhalb des Moduls. Der Wert dieser Variable ist im Normalfall statisch auszuführen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| TagName | PFE 🡨 MOD | STRING | Name of the represented Entity – i.e. TIC111 |

## TagDescription

Die Variable TagDescription beinhaltet die Beschreibung für die hier realisierte PLT-Stelle. Der Wert dieser Variable ist im Normalfall statisch auszuführen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| TagDescription | PFE 🡨 MOD | STRING | Description of the represented Entity – i.e. Reactor Temperature Inside |

## OSLevel

Die Variable OSLevel ermöglicht einen einfachen Mechanismus, mit dem das Schreiben mehrerer Bedienungsebenen überwacht werden kann. Jede Bedienerebene (z.b. lokaler Leitstand / zentraler Leitstand) wird einer entsprechenden Zahl zugeordnet. Wenn die Zahl der Bedienungsebene und die Zahl dieser Variable übereinstimmen, sollen die Bedienelemente der Bedienungseinrichtung den Schreibbefehl zulassen, andernfalls, sollen die Eingabefelder inaktiv sein. Somit werden konfligierende Schreibzugriffe verhindert.

Die Variable wird im Modul keinerlei Funktionalität auslösen, sondern dient lediglich einem zentralen Ort der Persistierung, damit alle Bedienungsstationen auf ein und denselben Wert zugreifen können. Für Lieferanten der Modulautomatisierung ist lediglich die Variable in der Schnittstelle vorzusehen.

Die Modulautomatisierung hat in diesen einfachen Mechanismus keine Möglichkeit zu prüfen, wer aktuell auf die Variablen schreibt. Die Überprüfung ist Aufgabe des Bildbausteins einer jeden Bedienebene selbst und fällt somit in den Realisierungsumfang der Lieferanten eines Prozessführungssystems.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| OSLevel | PFE 🡪 MOD | BYTE | Allowed Values 1..255 – Plant-wide Level Definitions must be defined |

## WQC

Der Worst Quality Code beinhaltet einen verarbeiteten, zusammenfassenden Quality Code, der den schlechtesten Quality Code aller eingehenden Quality Codes beinhaltet. Der schlechteste Quality Code mit der höchsten Priorität wird als Worst Quality Code ausgegeben. Jede Datenstruktur besitzt immer nur eine Instanz dieser Variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| WQC | PFE 🡨 MOD | BYTE | Worst Quality code – contains the worst quality code |

Jeder Prozess- und/oder Steuerwert besitzt, neben dem eigentlichen Wert, einen Parameter zur Übermittlung der Prozesswertqualität. Die Prozesswertqualität gibt Aufschluss darüber, unter welchen Bedingungen ein Wert ermittelt wurde. Dieser Qualitätswert wird meist von den Sensoren oder den Eingangsbaugruppen des Automatisierungssystems gebildet und beschreibt, unter welchen Bedingungen der Wert erfasst wurde und ob diesem Wert vertraut werden kann. Die höchste Priorität entspricht 0, die niedrigste Priorität entspricht 7.

| **Quality Code Enumeration** | | |
| --- | --- | --- |
| **Byte** | **Prio.** | **Description** |
| 16#60 | 0 | Simulation Value |
| 16#00 | 1 | Bad, conditioned by device |
| 16#28 | 2 | Bad, conditioned by process |
| 16#68 | 3 | Uncertain, conditioned by device |
| 16#78 | 4 | Uncertain, conditioned by process |
| 16#A4 | 5 | Maintenance Request |
| 16#80 | 6 | Good |
| 16#FF | 7 | No QC available (default) |

## ScaleSettings

Zum Festlegen der Anzeigegrenzen eines Analogwerts werden jeweils eine Min und eine Max Variable verwendet. Diese bietet die Möglichkeit den Wertebereich eines Analogwerts zu beschreiben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| SclMin | PFE 🡨 MOD | REAL/DINT | Low Limit Value e.g. for the Bar Graph |
| SclMax | PFE 🡨 MOD | REAL/DINT | High Limit Value e.g. for the Bar Graph |

## UnitSettings

Die Einheit eines Wertes wird mit Hilfe einer Unit Variable beschrieben. Die Einheiten werden nach IEC61158 beschrieben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| Unit | PFE 🡨 MOD | INT | Enumeration Value of the Unit List |

Folgende Auflistung beinhaltet die meistgenutzten Einheiten nach IEC61158.

|  |  |  |
| --- | --- | --- |
| Identifier | Einheit-Kürzel | Einheit |
| 1000 | K | Kelvin |
| 1001 | °C | Grad Celsius |
| 1002 | °F | Grad Fahrenheit |
| 1005 | ° | Grad |
| 1006 | ' | Minute |
| 1007 | '' | Sekunde |
| 1010 | m | Meter |
| 1013 | mm | Millimeter |
| 1018 | ft | Fuß |
| 1023 | m2 | Quadratmeter |
| 1038 | L | Liter |
| 1041 | hl | Hektoliter |
| 1054 | s | Sekunde |
| 1058 | min | Minute |
| 1059 | h | Stunde |
| 1060 | d | Tag |
| 1061 | m/s | Meter pro Sekunde |
| 1077 | Hz | Hertz |
| 1081 | kHz | Kilohertz |
| 1082 | 1/s | Pro Sekunde |
| 1083 | 1/min | Pro Minute |
| 1088 | kg | Kilogramm |
| 1092 | t | Metrische Tonne |
| 1100 | g/cm3 | Gramm pro Kubikzentimeter |
| 1105 | g/L | Gramm pro Liter |
| 1120 | N | Newton |
| 1123 | mN | Millinewton |
| 1130 | Pa | Pascal |
| 1133 | kPa | Kilopascal |
| 1137 | bar | Bar |
| 1138 | mbar | Millibar |
| 1149 | mmH2O | Millimeter Wassersäule |
| 1175 | W·h | Wattstunde |
| 1179 | kW·h | Kilowattstunde |
| 1181 | kcalth | Kilokalorien |
| 1190 | kW | Kilowatt |
| 1209 | A | Ampere |
| 1211 | mA | Milliampere |
| 1221 | A·h | Amperestunde |
| 1240 | V | Volt |
| 1342 | % | Prozent |
| 1349 | m3/h | Kubikmeter pro Stunde |
| 1353 | L/h | Liter pro Stunde |
| 1384 | mol | Mol |
| 1422 | pH | pH-Wert |

## ValueLimitation

Diese Variablen beschreiben eine Sollwertbegrenzung für Analogwerte. Werte außerhalb dieser Variablen werden nicht akzeptiert und eine Werteänderung muss verhindert werden. Die Interpretation des Wertes ergibt sich bei der Kombination aus dieser Standard-Funktion in Zusammenhang mit der Standard-Funktion Unit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| Min | PFE 🡨 MOD | REAL/DINT | Low Limit Value for Value Input |
| Max | PFE 🡨 MOD | REAL/DINT | High Limit Value for Value Input |

## OperationModes

### BaseOperationMode

Die Variablen des BaseOperationMode realisieren einen Zustandsautomat mit zwei Zuständen, der festlegt, ob eine Datenstruktur im Online oder Offline Modus ist. Anwendung findet diese Schnittstelle zur Zustandssteuerung von passiven Entitäten, wie zum Beispiel Messstellen oder Wertevorgaben. In den Zustand Offline kann der Automat sowohl vom Operator mit Hilfe der Op-Variablen, als auch modulintern, mit Hilfe der Li-Variablen gesetzt werden. In den Zustand Online kann nur der Operator schalten Der aktuelle Zustand wird über die beiden Statusausgänge signalisiert. In dieser Schnittstelle sind die Variablen des Operators und der internen Logik gleichberechtigt. Einen Default-Wert gibt es als globale Festlegung nicht, da er je nach Anwendungsfall der BaseOperationMode Schnittstelle unterschiedlich sein kann. Demnach ist in der konkreten Schnittstelle zu prüfen, ob dort ein Default-Zustand definiert wird. Grundsätzlich kann es aber auch sinnvoll sein, dass der Default-Wert des Zustandes durch den Modulautomatisierer definiert wird.

Die Variablen werden in eine Operation Mode Variable, kurz OpMode, zusammengefasst (siehe OpMode).

Der BaseOperationMode darf nur eingesetzt werden, wenn ExtendedOperationMode nicht verwendet wird.

|  |
| --- |
| Grafik: Zustandsautomat der BaseOperationMode Schnittstelle |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Definition** | | | |  |
| **Alias** | **Access** | **Type** | **Description** | **Mapping to OpMode** |
| StateOffLi | PFE 🡨 MOD | BOOL | Set Operation Mode to Offline by internal interaction  1:Operation Mode is set to Offline  0: no operation | Bit 1 |
| StateOffOp | PFE 🡪 MOD | BOOL | Set Operation Mode to Offline by operator interaction  0🡪1: request to set Operation Mode to Offline  1🡪0: acknowledge by Module | Bit 2 |
| StateOnOp | PFE 🡪 MOD | BOOL | Set Operation Mode to Online by operator interaction  0🡪1: request to set Operation Mode is to Online  1🡪0: acknowledge by Module | Bit 4 |
| StateOnAct | PFE 🡨 MOD | BOOL | Current State is Online  1:State is Online  0: State is not Online | Bit 7 (1:on) |
| StateOffAct | PFE 🡨 MOD | BOOL | Current State is Offline  1: State is Offline  0: State is not Offline | Bit 7 (0:off) |

### ExtendedOperationMode

Die Variablen des ExtendedOperationMode realisieren einen Zustandsautomaten mit drei Zuständen, der festlegt, ob eine Schnittstelle im Offline, Manual oder Automatic Mode sich befindet. Diese Variante findet Anwendung bei aktiven Entitäten, wie zum Beispiel Antriebe oder Regler. Ein Zustandswechsel kann immer nur zwischen Offline und Manual oder Manual und Automatic durchgeführt werden. Ein direkter Übergang von Offline in Automatic ist nicht erlaubt. Die Modullogik definiert, ob der Operator oder eine weitere Modullogik die Betriebsarten-Umschaltung durchführen darf, hierzu dient die Variable StateLiOp. In Abhängigkeit dieser Variable sind entweder die Operator-Signale (\*Op-Variablen) oder die modul-internen Variablen (\*Li-Variablen) ausschlaggebend. Mit Hilfe der StateLiOp Variable kann der Zustand bzw. die Gegebenheiten im Modul berücksichtigt werden, wann das Modul den Operator-Zugriff erlaubt und wann dieser verboten wird. Die aktuellen Zustände werden über entsprechende Signale ausgegeben. Auch hier ist es je nach Anwendungsfall und Verwendung innerhalb einer konkreten Schnittstelle abhängig, welcher der drei Zustände als Default-Wert definiert wird, aus diesem Grund wird auf die Schnittstellen spezifischen Beschreibungen verwiesen. Die Variablen werden in eine Operation Mode Variable, kurz OpMode, zusammengefasst (siehe OpMode).

Der ExtendedOperationMode darf nur eingesetzt werden, wenn BaseOperationMode nicht verwendet wird.

Weitere Besonderheiten im Zusammenhang mit dem Extended Operation Mode sind in den einzelnen Schnittstellenbeschreibungen ergänzt, da diese anwendungsspezifisch sind.

|  |
| --- |
| Grafik: Zustandsautomat der ExtendedOperationMode Schnittstelle |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Definition** | | | |  |
| **Alias** | **Access** | **Type** | **Description** | **Mapping to OpMode** |
| StateLiOp | PFE 🡨 MOD | BOOL | Selection of the active Operation Mode interaction channel  0: the operator switches (\*Op) shall be used  1: the internal switches (\*Li) shall be used | Bit 0 |
| StateOffLi | PFE 🡨 MOD | BOOL | Set Operation Mode to Offline by internal interaction  (relevant, if StateLiOp is true)  1: Operation Mode is set to Offline  0: no operation | Bit 1 |
| StateManLi | PFE 🡨 MOD | BOOL | Set Operation Mode to Manual by internal interaction  (relevant, if StateLiOp is true)  1: Operation Mode is set to Manual  0: no operation | Bit 3 |
| StateAutLi | PFE 🡨 MOD | BOOL | Set OperationMode to Automatic by internal interaction  (relevant, if StateLiOp is true)  1: Operation Mode is set to Automatic  0: no operation | Bit 5 |
| StateOffOp | PFE 🡪 MOD | BOOL | Set Operation Mode to Offline by operator interaction  (relevant, if StateLiOp is false)  0🡪1: request to set Operation Mode to Offline  1🡪0: acknowledge by Module | Bit 2 |
| StateManOp | PFE 🡪 MOD | BOOL | Set OperationMode to manual by operator interaction  (relevant, if StateLiOp is false)  0🡪1: request to set Operation Mode to Manual  1🡪0: acknowledge by Module | Bit 4 |
| StateAutOp | PFE 🡪 MOD | BOOL | Set OperationMode to Automatic by operator interaction  (active, if StateLiOp is false)  0🡪1: request to set Operation Mode is to Automatic  1🡪0: acknowledge by Module | Bit 6 |
| StateManAct | PFE 🡨 MOD | BOOL | 1: current mode is Manual  0: current mode is not Manual | Bit 7 (1:manual) |
| StateAutAct | PFE 🡨 MOD | BOOL | 1: current mode is Automatic  0: current mode is not Automatic | Bit 8 (1:automatic) |
| StateOffAct | PFE 🡨 MOD | BOOL | 1: current mode is Offline  0: current mode is not Offline | Bit 7/8 (0:off) |

### SourceMode

Die Variablen der SourceMode-Schnittstelle realisieren einen Zustandsautomaten zur Auswahl einer Quelle bei Schnittstellen die von zwei unterschiedlichen Quellen beeinflusst werden können. Es wird hierbei unterschieden zwischen Intern (modul-intern) und Extern (modul-extern). Die Modullogik definiert mit Hilfe der SrcLiOp-Variable, ob die Auswahl durch eine weitere Modul-Logik oder durch die Operator-Variablen vorgenommen werden darf. In Abhängigkeit dieser Variable sind entweder die Operator-Signale (\*Op-Variablen) oder die modul-internen Variablen (\*Li-Variablen) ausschlaggebend. Der Zeitpunkt, in dem umgeschaltet werden darf kann mit Hilfe der SrcLiOp-Variable durch das Modul definiert werden. Die aktuelle Quelle wird über entsprechende Signale ausgegeben. Die Default-Quelle kann je nach Anwendung in einer konkreten Schnittstelle unterschiedlich sinnvoll sein, demnach ist die Default-Definition in den Beschreibungen der konkreten Schnittstellen zu entnehmen.

Die Variablen werden in eine Operation Mode Variable, kurz OpMode, zusammengefasst (siehe OpMode).

Weitere Besonderheiten im Zusammenhang mit dem Source Mode sind in den einzelnen Schnittstellenbeschreibungen ergänzt, da diese anwendungsspezifisch sind.

|  |
| --- |
| Grafik: Zustandsautomat der SourceMode Schnittstelle |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Definition** | | | |  |
| **Alias** | **Access** | **Type** | **Description** | **Mapping to OpMode** |
| SrcLiOp | PFE 🡨 MOD | BOOL | Selection of the active Source Mode interaction channel  0: the operator switches (\*Op) shall be used  1: the internal switches (\*Li) shall be used | Bit 9 |
| SrcExtLi | PFE 🡨 MOD | BOOL | Set SourceMode to external by internal interaction  (relevant, if SrcLiOp is true)  1 :SourceMode is set to external  0: no operation | Bit 10 |
| SrcIntLi | PFE 🡨 MOD | BOOL | Set SourceMode to internal by internal interaction  (relevant, if SrcLiOp is true)  1 : SourceMode is set to internal  0: no operation | Bit 11 |
| SrcIntOp | PFE 🡪 MOD | BOOL | Set SourceMode to internal by operator interaction  (relevant, if SrcLiOp is false)  0🡪1: request to set Operation Mode to Internal  1🡪0: acknowledge by Module | Bit 12 |
| SrcExtOp | PFE 🡪 MOD | BOOL | Set SourceMode to external by operator interaction  (relevant, if SrcLiOp is false)  0🡪1: request to set Operation Mode to External  1🡪0: acknowledge by Module | Bit 13 |
| SrcIntAct | PFE 🡨 MOD | BOOL | 1: current mode is Intern  0: current mode is not Intern | Bit 14 (1:intern) |
| SrcExtAct | PFE 🡨 MOD | BOOL | 1: current mode is Extern  0: current mode is not Intern | Bit 14 (0:extern) |

### CollectiveSystemFault

Die Collective System Fault Variable dient zur Übertragung eines intern aufgetretenen Fehlers, der sowohl von der Logik der Datenstruktur selbst kommen kann, oder aber auch durch andere in Abhängigkeit stehende Datenstrukturen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Definition** | | | |  |
| **Alias** | **Access** | **Type** | **Description** | **Mapping to OpMode** |
| CSF | PFE 🡨 MOD | BOOL | 1: an internal fault logic recognizes a problem  0: no internal fault recognized | Bit 15 |

### OperationMode

Die Variable Operation Mode, kurz OpMode, ist eine Zusammenfassung der Definitionen aus „BaseOperationMode“, „ExtendedOperationMode, „SourceMode“ und „Collective System Fault“.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | See OpMode Mapping Definition |

| **Operation Mode Mapping Table** | | | |
| --- | --- | --- | --- |
| Bit | Name | Description | Elementary Structure |
| 0 | StateLiOp | 0: the operator switches (\*Op) shall be used  1: the internal switches (\*Li) shall be used | ExtendedOperationMode |
| 1 | StateOffLi | 1: Operation Mode is set to Offline  0: no operation | BaseOperationMode |
| (relevant, if StateLiOp is true)  1: Operation Mode is set to Offline  0: no operation | ExtendedOperationMode |
| 2 | StateOffOp | 0🡪1: request to set Operation Mode to Offline  1🡪0: acknowledge by Module | BaseOperationMode |
| (relevant, if StateLiOp is false)  0🡪1: request to set Operation Mode to Offline  1🡪0: acknowledge by Module | ExtendedOperationMode |
| 3 | StateManLi | (relevant, if StateLiOp is true)  1: Operation Mode is set to Manual  0: no operation | ExtendedOperationMode |
| 4 | StateManOp / StateOnOp | 0🡪1: request to set Operation Mode to Online  1🡪0: acknowledge by Module | BaseOperationMode |
| (relevant, if StateLiOp is false)  0🡪1: request to set Operation Mode to Manual  1🡪0: acknowledge by Module | ExtendedOperationMode |
| 5 | StateAutLi | (relevant, if StateLiOp is true)  1: Operation Mode is set to Automatic  0: no operation | ExtendedOperationMode |
| 6 | StateAutOp | (relevant, if StateLiOp is false)  0🡪1: request to set Operation Mode to Automatic  1🡪0: acknowledge by Module | ExtendedOperationMode |
| 7/8 | StateOffAct | Bit 7 = 0: current mode is Offline. | BaseOperationMode |
| StateOnAct | Bit 7 = 1: current mode is Online. | BaseOperationMode |
| StateOffAct | Bit 7 = 0, Bit 8 = 0: current mode is Offline. | ExtendedOperationMode |
| StateManAct | Bit 7 =1, Bit 8=0: current mode is Manual. | ExtendedOperationMode |
| StateAutAct | Bit 7 =0, bit 8 =1, current mode is Automatic | ExtendedOperationMode |
| 9 | SrcLiOp | 0: the operator switches (\*Op) shall be used  1: the internal switches (\*Li) shall be used | SourceMode |
| 10 | SrcExtLi | (relevant, if SrcLiOp is true)  1 :SourceMode is set to external  0: no operation | SourceMode |
| 11 | SrcIntLi | (relevant, if SrcLiOp is true)  1 : SourceMode is set to internal  0: no operation | SourceMode |
| 12 | SrcIntOp | (relevant, if SrcLiOp is false)  1: Source Mode is set to internal | SourceMode |
| 13 | SrcExtOp | (relevant, if SrcLiOp is false)  1: Source Mode is set to external | SourceMode |
| 14 | SrcExtAct / SrcIntAct | 1: current mode is internal  0: current mode is external. | SourceMode |
| 15 | CSF | 1: a system fault has been detected  0: no system fault has been detected | SystemFault |
| 16+ | Reserve | | |

## Handshake Verfahren

Bei einigen Variablen wird eine Information zwischen dem Bildbaustein des Operators und der Schnittstelle ausgetauscht. Die dahinter liegende Implementierung der Funktion wird somit angesprochen. Der Transport der Information wird mit Hilfe eines Handshake-Verfahrens realisiert. Der Bildbaustein schreibt einen Wert, z.B. logisch „1“, auf die Variable und die Schnittstelle bzw. deren Realisierung setzt diesen nach Bearbeitung wieder auf eine logische „0“. Bearbeitung heißt in diesem Fall aber nicht, dass die Funktion, wie durch den Operator angefordert, erfolgreich ausgeführt wurde, sondern lediglich, dass die Funktion die Anforderung vernommen hat. Die erfolgreiche Ausführung wird mit Hilfe der vorhandenen zustandsbeschreibenden Variablen realisiert.

Das Handshake-Verfahren kommt immer dann zur Anwendung, wenn in der Variablen-Beschreibung die entsprechende Flankenwechsel-Notation beschrieben ist (0🡪1: do something, 1🡪0 acknowledge).

|  |
| --- |
| Grafik: Sequenzdiagramm der Interaktion im Handshake-Verfahren am Beispiel eines einfachen RS-Glieds |
|  |

## Interlocks

Die Standard-Funktion Interlocks beinhaltet ein dreistufiges System bestehend aus einem Permit, einem Interlock und einem Protect.

Permit realisiert eine Einschaltverriegelung verhindert bei verriegeltem Zustand das aktivieren.

Interlock realisiert eine Verriegelung, die das Einschalten verhindert, als auch eine Entität in deren Sicherheitsposition versetzt. Ein aktiver Interlock muss nicht zurückgesetzt werden.

Protect realisiert eine Verriegelung, vergleichbar zum Interlock, mit dem Unterschied, dass die Protection zurückgesetzt werden muss.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| PermEn | PFE 🡨 MOD | BOOL | Enables the Permission Lock  1:enabled  0:disabled |
| Permit | PFE 🡨 MOD | BOOL | Permit locks the Entity to be activated  1:permission is given  0:permission is not given |
| IntlEn | PFE 🡨 MOD | BOOL | Enables the Interlock Lock  1:enabled  0:disabled |
| Interlock | PFE 🡨 MOD | BOOL | Interlock locks the Entity to the safe state  1: interlock is not active  0: interlocking is active, must not be reset (see Reset) |
| ProtEn | PFE 🡨 MOD | BOOL | Enables the Protection Lock  1:enabled  0:disabled |
| Protect | PFE 🡨 MOD | BOOL | Protect locks the Entity to the safe state – a reset is required  1: protection is not active  0: protection is active, must be reset (see Reset) |

## Feedback Monitoring

Die Standard-Funktion Feedback-Monitoring unterscheidet in eine statische und eine dynamische Fehlfunktion.

Eine statische Fehlfunktion liegt vor, wenn eine Datenstruktur den Zustand ändert ohne dass eine Änderung der Ansteuerung erfolgte. Ein Beispiel wäre eine Änderung des Ventilzustands ohne dass das Ansteuersignal geändert wurde. Dieser Fehler deutet zum Beispiel auf den Verlust der Versorgungsenergie hin.

Eine dynamische Fehlfunktion liegt vor, wenn eine Datenstruktur den Zustand nicht ändert, obwohl eine Änderung der Ansteuerung erfolgte. Ein Beispiel wäre die Änderung der Ansteuerung eines Ventils, jedoch ändert sich das entsprechende Rückführsignal des Zustands nicht. Dieser Fehler deutet zum Beispiel auf einen mechanischen Fehler des Ventils hin.

Über die Variable MonSafePos wird angezeigt, ob eine Datenstruktur in ihre Sicherheitsposition geht, wenn ein Fehler erkannt wurde. Bei Antrieben ist diese immer „true“, was bedeutet, dass Antriebe immer in die Sicherheitsposition gehen, wenn ein Fehler erkannt wird.

Über die beiden Zeiten MonTiStat und MonTiDyn wird eingestellt, wie lange gewartet wird, bis der geforderte Zustand der Ansteuerung, sich an der entsprechenden Datenstruktur eingestellt hat. Falls sich der geforderte Zustand an der Datenstruktur einstellt, bevor die Zeit abgelaufen ist, liegt keine Fehlfunktion vor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| MonEn | PFE 🡨 MOD | BOOL | Enables the Monitoring Feature  1:enabled  0:disabled |
| MonSafePos | PFE 🡨 MOD | BOOL | Set the Error Behaviour, if an error occurs  1: after an error occurs, the safe pos will be set  0: after an error occurs, the entity holds the state |
| MonStatErr | PFE 🡨 MOD | BOOL | Static Error occured  1:error occurred  0: no error occured |
| MonDynErr | PFE 🡨 MOD | BOOL | Dynamic Error occured  1:error occurred  0: no error occured |
| MonStatTi | PFE 🡪 MOD | REAL | Monitor Time for uncontrolled changes [s] |
| MonDynTi | PFE 🡪 MOD | REAL | Monitor Time for controlled changes [s] |

## Reset

Die Standard-Funktion Reset beinhaltet zwei Variablen die ein Rücksetzen von Fehlervariablen und dem Protection-Lock ermöglicht. Rücksetzen kann sowohl von einer internen Logik im Modulautomatisierungssystem als auch vom Operator ausgeführt werden. Beide Reset-Kanäle sind gleichberechtigt.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definitions** | | | |
| **Alias** | **Access** | **Type** | **Description** |
| ResetOp | PFE 🡪 MOD | BOOL | Reset by Operator (Handshake)  0🡪1: request from PFE  1🡪0: acknowledge from Module |
| ResetLi | PFE 🡨 MOD | BOOL | Reset by internal link  1: Reset executed  0: no operation |

## Limit Monitoring

Die Standard-Funktion Limit Monitoring überwacht einen Analogwert auf bis zu 6 Grenzen. Die Grenzen unterteilen sich einmal in Ober- und Untergrenzen und einmal in Toleranz-, Warnungs- und Alarmgrenzen. Mit Hilfe der Enable Variablen, kann eine Grenzwertüberwachung aktiviert oder deaktiviert werden. Mit den entsprechenden Limit Variablen wird der Grenzwert festgelegt. Sowohl die Enable- als auch die Limit-Variablen können durch den Operator verändert werden. Die Active Variablen signalisieren eine Verletzung der Grenzwerte.

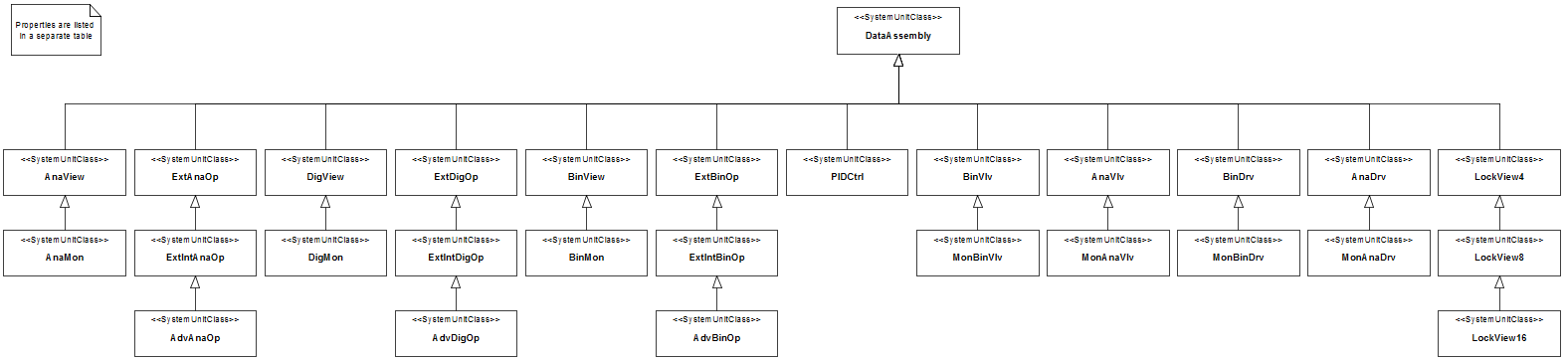
|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Definition** | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| AHEn | PFE 🡨 MOD | BOOL | Enables Alarm High Limit  1:enabled  0:disabled |
| AHLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Alarm High |
| AHAct | PFE 🡨 MOD | BOOL | Alarm High active  1:active  0:inactive |
| WHEn | PFE 🡨 MOD | BOOL | Enables Warning High Limit  1:enabled  0:disabled |
| WHLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Warning High |
| WHAct | PFE 🡨 MOD | BOOL | Warning High active  1:active  0:inactive |
| THEn | PFE 🡨 MOD | BOOL | Enables Tolerance High Limit  1:enabled  0:disabled |
| THLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Tolerance High |
| THAct | PFE 🡨 MOD | BOOL | Tolerance High active  1:active  0:inactive |
| TLEn | PFE 🡨 MOD | BOOL | Enables Tolerance Low Limit  1:enabled  0:disabled |
| TLLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Tolerance Low |
| TLAct | PFE 🡨 MOD | BOOL | Tolerance Low active  1:active  0:inactive |
| WLEn | PFE 🡨 MOD | BOOL | Enables Warning Low Limit  1:enabled  0:disabled |
| WLLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Warning Low |
| WLAct | PFE 🡨 MOD | BOOL | Warning Low active  1:active  0:inactive |
| ALEn | PFE 🡨 MOD | BOOL | Enables Alarm Low Limit  1:enabled  0:disabled |
| ALLim | PFE 🡪 MOD | REAL/DINT | Limit Value for Alarm Low |
| ALAct | PFE 🡨 MOD | BOOL | Alarm Low active  1:active  0:inactive |

# Schnittstellendefinitionen

Dieser Absatz definiert die Schnittstellen, die in entsprechenden Schnittstellenfamilien organisiert werden. Diese Schnittstellen legen den zwischen Prozessführungssystem und Modulautomatisierungssystem auszutauschenden Datenumfang, sowohl syntaktisch als auch semantisch fest.

Die in der Richtlinie definierten Schnittstellenfamilien, sowie deren Basis- und Erweiterungsschnittstellen sind in Tabelle X aufgelistet.

|  |  |
| --- | --- |
| **Schnittstellenfamilie** | **Basis-Schnittstelle, (Erweiterungen), (…)** |
| Analogwertanzeigen | AnaView, (AnaMon) |
| Analogwertoperationen | ExtAnaOp, (ExtIntAnaOp), (AdvAnaOp) |
| Binärwertanzeigen | BinView, (BinMon) |
| Binäroperationen | ExtBinOp, (ExtIntBinOp), (AdvBinOp) |
| Zeichenkettenanzeige | StrView |
| Regelungen | PIDCtrl |
| Bistabile Ventile | BinVlv, (MonBinVlv) |
| Regelbare Ventile | AnaVlv, (MonAnaVlv) |
| Bistabile Antriebe | BinDrv, (MonBinDrv) |
| Regelbare Antriebe | AnaDrv, (MonAnaDrv) |
| Verriegelungsanzeigen | LockView4, (LockView8), (LockView16) |

Abbildung : Ableitungshierarchie der SystemUnitClass-Definitionen der in Blatt 3 definierten Schnittstellen

Die Schnittstellendefinitionen dieser Richtlinie werden in Form einer AutomationML SystemUnitClassLibrary, mit dem Namen *MTPDataObjectSUCLib*, modelliert. Jede Basis-Schnittstelle wird von der DataAssembly SystemUnitClass abgeleitet. Die erweiterten Schnittstellen leiten von der SystemUnitClass der Eltern-Schnittstelle ab und erweitern die Attribute der SystemUnitClass.

## DataAssembly

Das DataAssembly ist das Root-Objekt eines jeden Datensatz-Elements im ModuleTypePackage. Alle weiteren Datenpakete müssen von diesem Element erben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| TagName | PFE 🡨 MOD | STRING | TagName Field (see TagName) |
| TagDescription | PFE 🡨 MOD | STRING | TagDescription Field (see TagDescription) |
| OSLevel | PFE 🡪 MOD | BYTE | OSLevel Variable (see OSLevel) |
| WQC | PFE 🡨 MOD | BYTE | Worst Quality Code Variable (see WQC) |

## Analogwertanzeigen

Analoganzeigen werden zum einen verwendet um gemessene Werte in der PFE zu visualisieren und zum anderen um einfache analoge Werte von der Modulautomatisierung an die PFE zu übertragen.

### AnaView

AnaView dient zur Anzeige eines analogen Wertes der Modulautomation (beispielsweise Prozesswerte und Parameter). Dazu gehören neben dem aktuellen Wert, die Einheit, der minimale und maximale Skalenwert, sowie die Qualität des Wertes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| V | PFE 🡨 MOD | REAL | Value |
| VSclMin | PFE 🡨 MOD | REAL | Value Scale Low Limit (see ScaleSettings) |
| VSclMax | PFE 🡨 MOD | REAL | Value Scale High Limit (see ScaleSettings) |
| VUnit | PFE 🡨 MOD | INT | Value Unit (see UnitSettings) |

### AnaMon

AnaMon erweitert die Schnittstelle AnaView um die Grenzwertprüfung entsprechend der Standard-Funktion „Limit Monitoring“.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/AnaView | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VAHEn | PFE 🡨 MOD | BOOL | Enable Alarm High Limit  1:enabled  0:disabled |
| VAHLim | PFE 🡪 MOD | REAL | Limit Value for Alarm High |
| VAHAct | PFE 🡨 MOD | BOOL | Alarm High Active  1:active  0:inactive |
| VWHEn | PFE 🡨 MOD | BOOL | Enable Warning High Limit  1:enabled  0:disabled |
| VWHLim | PFE 🡪 MOD | REAL | Limit Value for Warning High |
| VWHAct | PFE 🡨 MOD | BOOL | Warning High Active  1:active  0:inactive |
| VTHEn | PFE 🡨 MOD | BOOL | Enables Tolerance High Limit  1:enabled  0:disabled |
| VTHLim | PFE 🡪 MOD | REAL | Limit Value for Tolerance High |
| VTHAct | PFE 🡨 MOD | BOOL | Tolerance High Active  1:active  0:inactive |
| VTLEn | PFE 🡨 MOD | BOOL | Enables Tolerance Low Limit  1:enabled  0:disabled |
| VTLLim | PFE 🡪 MOD | REAL | Limit Value for Tolerance Low |
| VTLAct | PFE 🡨 MOD | BOOL | Tolerance Low Active  1:active  0:inactive |
| VWLEn | PFE 🡨 MOD | BOOL | Enables Warning Low Limit  1:enabled  0:disabled |
| VWLLim | PFE 🡪 MOD | REAL | Limit Value for Warning Low |
| VWLAct | PFE 🡨 MOD | BOOL | Warning Low Active  1:active  0:inactive |
| VALEn | PFE 🡨 MOD | BOOL | Enables Alarm Low Limit  1:enabled; 0:disabled |
| VALLim | PFE 🡪 MOD | REAL | Limit Value for Alarm Low |
| VALAct | PFE 🡨 MOD | BOOL | Alarm Low Active  1:active  0:inactive |

## Analogoperationen

Analogoperationen werden verwendet um analoge Werte an die Modulautomatisierung zu übertragen.

### ExtAnaOp

ExtAnaOp ist eine Schnittstelle zum Übertragen eines analogen Wertes durch die PFE in die Modulautomatisierung. Der übertragene Wert enthält einen minimalen und maximalen Skalenwert. Informationen wie die Einheit, sowie minimaler und maximaler Eingabewert werden in dem Modul selbst definiert. Zur Überprüfung der Eingabe, werden alle Werte als Rückgabewert zur Verfügung gestellt. Zudem verfügt die Schnittstelle über die Standard-Funktionen OSLevel und Worst Quality Code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VOut | PFE 🡨 MOD | REAL | Value Output |
| VSclMin | PFE 🡨 MOD | REAL | Value Scale Low Limit (see ScaleSettings) |
| VSclMax | PFE 🡨 MOD | REAL | Value Scale High Limit (see ScaleSettings) |
| VUnit | PFE 🡨 MOD | INT | Value Unit (see UnitSettings) |
| VExt | PFE 🡪 MOD | REAL | External Value |
| VMin | PFE 🡨 MOD | REAL | Value Low Limit (see ValueLimitation) |
| VMax | PFE 🡨 MOD | REAL | Value High Limit (see ValueLimitation) |
| VRbk | PFE 🡨 MOD | REAL | Readback Value |

### ExtIntAnaOp

ExtIntAnaOp wird zur Vorgabe eines analogen Wertes von einem externen System, als auch von der Modulautomatisierung selbst kommend, verwendet. Die Schnittstelle ExtAnaOp wird hier um die interne Wertevorgabe und einen Operation Mode erweitert. Wird der Baustein auf den externen Wert geschaltet, so wird anstelle der internen Sollwertvorgabe der externe Sollwert verwendet. Hierdurch kann ein analoger Sollwert in die Modulautomatisierung geschrieben werden.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtAnaOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *VExt* | *PFE 🡪 MOD* | *REAL* | *(relevant, if external source is active, see OperationMode)*  *External Value* |
| VInt | PFE 🡨 MOD | REAL | (relevant, if internal source is active, see OperationMode)  Internal Value |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OperationMode, only Source Mode used) |

### AdvAnaOp

AdvAnaOp ist eine Erweiterung zu ExtIntAnaOp und besitzt grundsätzlich keine anderen Variablen, im Vergleich zum ExtIntAnaOp. Diese Schnittstelle besitzt einen um die BaseOperationMode erweiterten Funktionsumfang.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtAnaOp/ExtIntAnaOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *OpMode* | *PFE 🡨🡪 MOD* | *DWORD* | Operation Mode Mask  (see OperationMode, BaseOperation and Source Mode used) |

## Digitalwertanzeigen

Digitalwertanzeigen werden zum einen verwendet um ganzzahlige Werte in der PFE zu visualisieren und zum anderen um einfache digitale Werte von der Modulautomatisierung an die PFE zu übertragen.

### DigView

DigView dient zur Anzeige eines digitalen Wertes der Modulautomation (beispielsweise ganzzahlige Prozesswerte und Parameter). Dazu gehören neben dem aktuellen Wert, die Einheit, der minimale und maximale Skalenwert, sowie die Qualität des Wertes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| V | PFE 🡨 MOD | DINT | Value |
| VSclMin | PFE 🡨 MOD | DINT | Value Scale Low Limit (see ScaleSettings) |
| VSclMax | PFE 🡨 MOD | DINT | Value Scale High Limit (see ScaleSettings) |
| VUnit | PFE 🡨 MOD | INT | Value Unit (see UnitSettings) |

### DigMon

DigMon erweitert die Schnittstelle DigView um die Grenzwertprüfung entsprechend der Standard-Funktion „Limit Monitoring“.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/DigView | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VAHEn | PFE 🡨 MOD | BOOL | Enable Alarm High Limit  1:enabled  0:disabled |
| VAHLim | PFE 🡪 MOD | DINT | Limit Value for Alarm High |
| VAHAct | PFE 🡨 MOD | BOOL | Alarm High Active  1:active  0:inactive |
| VWHEn | PFE 🡨 MOD | BOOL | Enable Warning High Limit  1:enabled  0:disabled |
| VWHLim | PFE 🡪 MOD | DINT | Limit Value for Warning High |
| VWHAct | PFE 🡨 MOD | BOOL | Warning High Active  1:active  0:inactive |
| VTHEn | PFE 🡨 MOD | BOOL | Enables Tolerance High Limit  1:enabled  0:disabled |
| VTHLim | PFE 🡪 MOD | DINT | Limit Value for Tolerance High |
| VTHAct | PFE 🡨 MOD | BOOL | Tolerance High Active  1:active  0:inactive |
| VTLEn | PFE 🡨 MOD | BOOL | Enables Tolerance Low Limit  1:enabled  0:disabled |
| VTLLim | PFE 🡪 MOD | DINT | Limit Value for Tolerance Low |
| VTLAct | PFE 🡨 MOD | BOOL | Tolerance Low Active  1:active  0:inactive |
| VWLEn | PFE 🡨 MOD | BOOL | Enables Warning Low Limit  1:enabled  0:disabled |
| VWLLim | PFE 🡪 MOD | DINT | Limit Value for Warning Low |
| VWLAct | PFE 🡨 MOD | BOOL | Warning Low Active  1:active  0:inactive |
| VALEn | PFE 🡨 MOD | BOOL | Enables Alarm Low Limit  1:enabled  0:disabled |
| VALLim | PFE 🡪 MOD | DINT | Limit Value for Alarm Low |
| VALAct | PFE 🡨 MOD | BOOL | Alarm Low Active  1:active  0:inactive |

## Digitaloperationen

Digitaloperationen werden verwendet um ganzzahlige Werte an die Modulautomatisierung zu übertragen.

### ExtDigOp

ExtDigOp ist eine Schnittstelle zum Übertragen eines ganzzahligen Wertes durch die PFE in die Modulautomatisierung. Der übertragene Wert enthält einen minimalen und maximalen Skalenwert. Informationen wie die Einheit, sowie minimaler und maximaler Eingabewert werden in dem Modul selbst definiert. Zur Überprüfung der Eingabe, werden alle Werte als Rückgabewert zur Verfügung gestellt. Zudem verfügt die Schnittstelle über die Standard-Funktionen OSLevel und Worst Quality Code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VOut | PFE 🡨 MOD | DINT | Value Output |
| VSclMin | PFE 🡨 MOD | DINT | Value Scale Low Limit (see ScaleSettings) |
| VSclMax | PFE 🡨 MOD | DINT | Value Scale High Limit (see ScaleSettings) |
| VUnit | PFE 🡨 MOD | INT | Value Unit (see UnitSettings) |
| VExt | PFE 🡪 MOD | DINT | External Value |
| VMin | PFE 🡨 MOD | DINT | Value Low Limit (see ValueLimitation) |
| VMax | PFE 🡨 MOD | DINT | Value High Limit (see ValueLimitation |
| VRbk | PFE 🡨 MOD | DINT | Readback Value |

### ExtIntDigOp

ExtIntDigOp wird zur Vorgabe eines ganzzahligen Wertes von einem externen System, als auch von der Modulautomatisierung selbst kommend, verwendet. Die Schnittstelle ExtDigOp wird hier um die interne Wertevorgabe und einen Operation Mode erweitert. Wird der Baustein auf den externen Wert geschaltet, so wird anstelle der internen Sollwertvorgabe der externe Sollwert verwendet. Hierdurch kann ein ganzzahliger Sollwert in die Modulautomatisierung geschrieben werden.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtDigOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *VExt* | *PFE 🡪 MOD* | *DINT* | *(relevant, if external source is active, see OperationMode)*  *External Value* |
| VInt | PFE 🡨 MOD | DINT | (relevant, if internal source is active, see OperationMode)  Internal Value |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see Operation Mode, only Source Mode used) |

### AdvDigOp

AdvDigOp ist eine Erweiterung zu ExtIntDigOp und besitzt grundsätzlich keine anderen Variablen, im Vergleich zum ExtIntDigOp. Diese Schnittstelle besitzt einen um die BaseOperationMode erweiterten Funktionsumfang.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtDigOp/ExtIntDigOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *OpMode* | *PFE 🡨🡪 MOD* | *DWORD* | Operation Mode Mask  (see Operation Mode, BaseOperation- and Source Mode used) |

## Binäranzeigen

Binäranzeigen werden verwendet, um einen erfassten Wert in der PFE zu visualisieren.

### BinView

BinView dient der Anzeige eines einfachen binären Werts. Zusätzlich zum Wert können Statustexte für den Wert 0 und 1 mitgeliefert werden. Die Schnittstelle besitzt die Standard-Funktionen OSLevel und Worst Quality Code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/BinView | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| V | PFE 🡨 MOD | BOOL | Binary Value |
| VState0 | PFE 🡨 MOD | STRING | Text Replacement for false |
| VState1 | PFE 🡨 MOD | STRING | Text Replacement for true |

### BinMon

BinMon erweitert die Schnittstelle BinView um die Überwachung eines flatternden Signals.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/BinView | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VFlutEn | PFE 🡨 MOD | BOOL | Enable Fluttering Recognition  1:enabled; 0:disabled |
| VFlutTi | PFE 🡨🡪 MOD | REAL | Period of an active signal before it will be recognized as flutter-free [s] |
| VFlutCnt | PFE 🡨🡪 MOD | INT | Counts of the allowed fluttering signals in the defined period VFlutTi |
| VFlutAct | PFE 🡨 MOD | BOOL | Fluttering Signal recognized  1:Fluttering Signal recognized |

## Binäroperation

### ExtBinOp

Diese Schnittstelle ermöglicht es einen binären Wert im Modulautomatisierungssystem zu setzen oder rückzusetzen. Der Binärwert besitzt zusätzliche Statustexte. Zudem verfügt die Schnittstelle über die Standard-Funktionen OSLevel und Worst Quality Code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| VState0 | PFE 🡨 MOD | STRING | Text Replacement for false |
| VState1 | PFE 🡨 MOD | STRING | Text Replacement for true |
| VExt | PFE 🡪 MOD | BOOL | Value from Operator |
| VOut | PFE 🡨 MOD | BOOL | Binary Value Output |
| VRbk | PFE 🡨 MOD | BOOL | Binary Value Readback |

### ExtIntBinOp

Diese Schnittstelle erweitert ExtBinOp um zusätzliche Variablen für die Anzeige einer internen Wertevorgabe und der dazugehörigen Operation Mode (SourceMode) Funktionalität zur Auswahl der Quelle.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtBinOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *VExt* | *PFE 🡪 MOD* | *BOOL* | *(relevant, if external source is active, see OperationMode)*  *Value from Operator* |
| VInt | PFE 🡨 MOD | BOOL | (relevant, if internal source is active, see OperationMode)  Value from Internal Link |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OperationMode, only Source Mode used) |

### AdvBinOp

AdvBinOp ist eine Erweiterung des ExtIntBinOp und besitzt grundsätzlich keine anderen Variablen, im Vergleich zum ExtIntBinOp. Diese Schnittstelle besitzt einen um BaseOperationMode erweiterten Funktionsumfang.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/ExtBinOp/ExtIntBinOp | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| *OpMode* | PFE 🡨🡪 MOD | *DWORD* | Operation Mode Mask  (see Operation Mode, BaseOperation- and Source Mode used) |

## Regelungen

### PIDCtrl

Die Schnittstelle PIDCtrl stellt die Informationen für einen kontinuierlichen PID Regler bereit, der die typischen Größen (Prozesswert PV, Sollwert SP, Regelgröße MV) entsprechend verarbeitet. Die Schnittstelle beinhaltet für den Sollwert eine Umschaltung, um sowohl einen intern vorgegebenen, als auch einen extern vorgegebenen Sollwert zu verwenden. Hierbei kommt die Schnittstelle Source Mode zum Einsatz. Der Regler selbst besitzt drei Zustände – Offline, Manual und Automatic. Im Zustand Manual, wird die Regelgröße MV über die Variable MVOp durch den Operator vorgegeben. Im Zustand Automatic ist der Regelungsalgorithmus in Kraft und kann entsprechend des internen oder externen Sollwerts eine manuelle Sollwertvorgabe oder eine Kaskaden-Schaltung realisieren. Mit Hilfe der Min- und Max-Variablen werden Begrenzungsfunktionen für Sollwerte und Stellgröße dargestellt. Der Regelungsalgorithmus kann über die drei Variablen P, TI und TD parametriert werden. Darüber hinaus besitzt die Schnittstelle die Standard-Funktionen OSLevel, WQC und Operation Mode.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Zustände für Operation Mode und Source Mode werden im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OeprationModes, ExtendedOperation- and SourceMode used) |
| PV | PFE 🡨 MOD | REAL | Process Value |
| PVSclMin | PFE 🡨 MOD | REAL | Process Value Scale Low Limit (see ScaleSettings) |
| PVSclMax | PFE 🡨 MOD | REAL | Process Value Scale High Limit (see ScaleSettings) |
| PVUnit | PFE 🡨 MOD | INT | Process Value Unit (see UnitSettings) |
| SPExt | PFE 🡪 MOD | REAL | (relevant, if external source and automatic mode is active)  External Setpoint |
| SPInt | PFE 🡨 MOD | REAL | (relevant, if internal source and automatic mode is active)  Internal Setpoint |
| SPSclMin | PFE 🡨 MOD | REAL | Setpoint Scale Low Limit (see ScaleSettings) |
| SPSclMax | PFE 🡨 MOD | REAL | Setpoint Scale High Limit (see ScaleSettings) |
| SPUnit | PFE 🡨 MOD | INT | Setpoint Unit (see UnitSettings) |
| SPIntMin | PFE 🡨 MOD | REAL | Internal Setpoint Low Limit (see ValueLimitation) |
| SPIntMax | PFE 🡨 MOD | REAL | Internal Setpoint High Limit (see ValueLimitation) |
| SPExtMin | PFE 🡨 MOD | REAL | External Setpoint Low Limit (see ValueLimitation) |
| SPExtMax | PFE 🡨 MOD | REAL | External Setpoint High Limit (see ValueLimitation) |
| SP | PFE 🡨 MOD | REAL | (depends on the active selected source)  Setpoint |
| MVOp | PFE 🡪 MOD | REAL | (relevant, if manual mode is active)  Manipulated Value from Operator (only in manual mode used) |
| MV | PFE 🡨 MOD | REAL | Manipulated Value |
| MVMin | PFE 🡨 MOD | REAL | Minimal Manipulated Value (see ValueLimitation) |
| MVMax | PFE 🡨 MOD | REAL | Maximal Manipulated Value (see ValueLimitation) |
| MVUnit | PFE 🡨 MOD | INT | Manipulated Value Unit (see UnitSettings) |
| MVSclMin | PFE 🡨 MOD | REAL | Manipulated Value Scale Low Limit (see ScaleSettings) |
| MVSclMax | PFE 🡨 MOD | REAL | Manipulated Value Scale High Limit (see ScaleSettings) |
| P | PFE 🡨🡪 MOD | REAL | (relevant, if automatic mode is active)  Proportional Parameter |
| Ts | PFE 🡨🡪 MOD | REAL | (relevant, if automatic mode is active)  Integration Parameter |
| Td | PFE 🡨🡪 MOD | REAL | (relevant, if automatic mode is active)  Derivation Parameter |

## Bistabile Ventile

Bistabile Ventile können nur geöffnet bzw. geschlossen werden.

### BinVlv

Die Schnittstelle BinVlv beschreibt ein bistabiles Ventil mit der Möglichkeit aus dem Modul oder durch den Operator gesteuert werden zu können. Zusätzlich beinhaltet er entsprechende Feedback-Signale mit denen der tatsächliche Ventilzustand angezeigt werden kann. Darüber hinaus enthält die Schnittstelle die Standard-Funktion ExtendedOperationMode.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OeprationModes, only ExtendedOperationMode used) |
| SafePos | PFE 🡨 MOD | BOOL | Safe Position  1:open; 0:close |
| OpenOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationMode)  Open Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| CloseOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationMode)  Close Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| OpenLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active, see OperationMode)  Open Command from Internal Link  1: set open  0: no operation |
| CloseLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active)  Close Command from Internal Link  1: set close  0: no operation |
| Ctrl | PFE 🡨 MOD | BOOL | Valve Control  1:active  0:inactive |
| OpenFbkEn | PFE 🡨 MOD | BOOL | Open Feedback Enable  1:enabled  0:disabled |
| CloseFbkEn | PFE 🡨 MOD | BOOL | Close Feedback Enable  1:enabled  0:disabled |
| OpenFbk | PFE 🡨 MOD | BOOL | Open Feedback Signal  1: Open , if OpenFbkEn = false, Fbk will be calculated from Ctrl and SafePos  0: is not opened |
| CloseFbk | PFE 🡨 MOD | BOOL | Close Feedback Signal  1: Close, if OpenFbkEn = false, Fbk will be calculated from Ctrl and SafePos  0: is not closed |
| PermEn | PFE 🡨 MOD | BOOL | Enables the Permission Lock  1:enabled  0:disabled |
| Permit | PFE 🡨 MOD | BOOL | Permit locks the Entity to be activated  1:permission is given  0:permission is not given |
| IntlEn | PFE 🡨 MOD | BOOL | Enables the Interlock Lock  1:enabled  0:disabled |
| Interlock | PFE 🡨 MOD | BOOL | Interlock locks the Entity to the safe state  1: interlock is not active  0: interlocking is active, must not be reset (see Reset)  In this case deactivation (interlock = 0) means the valve will be set to safe position. |
| ProtEn | PFE 🡨 MOD | BOOL | Enables the Protection Lock  1:enabled; 0:disabled |
| Protect | PFE 🡨 MOD | BOOL | Protect locks the Entity to the safe state – a reset is required  1: protection is not active; 0: protection is active, must be reset (see Reset)  In this case deactivation (protect = 0) means the valve will be set to safe position. |
| ResetOp | PFE 🡪 MOD | BOOL | Reset from Operator (see Handshake)  0🡪1: request from PFE  1🡪0: acknowledge from Module |
| ResetLi | PFE 🡨 MOD | BOOL | Reset from internal link  1:Reset executed  0: no Operation |

### MonBinVlv

Die Schnittstelle MonBinVlv erweitert die Schnittstelle BinVlv um die Standard-Funktion Feedback Monitoring. Diese definiert übergreifend die Überwachung von gesteuerten (dynamische Fehlfunktion) oder ungesteuerten (statische Fehlfunktion) Zustandsänderungen inklusive deren Parametrierung.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/BinVlv | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| MonEn | PFE 🡪 MOD | BOOL | Monitor Enable  1:enabled  0:disabled |
| MonSafePos | PFE 🡪 MOD | BOOL | Set the Error Behaviour, if an error occurs  1: after an error occurs, the safe pos will be set  0: after an error occurs, the entity holds the state |
| MonStatErr | PFE 🡪 MOD | BOOL | Static Error active  1:active  0:inactive |
| MonDynErr | PFE 🡪 MOD | BOOL | Dynamic Error active  1:active  0:inactive |
| MonStatTi | PFE 🡨🡪 MOD | REAL | Monitor Time for uncontrolled changes [s] |
| MonDynTi | PFE 🡨🡪 MOD | REAL | Monitor Time for controlled changes [s] |

## Analoge Ventile

Ventile können geöffnet bzw. geschlossen, sowie frei-positionierbar betrieben werden.

### AnaVlv

Die Schnittstelle AnaVlv stellt ein frei-positionierbares Analogventil dar. Es bietet die Möglichkeit sowohl den Zustand als auch die Position durch den Operator oder das modul-interne Automatisierungssystem steuern zulassen. Es verfügt über die Standard-Funktionen OSLevel, WQC, Operation Mode (ExtendedOperationMode und SourceMode).

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OeprationModes, ExtendedOperation- and SourceMode used) |
| OpenOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationModes)  Open Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| CloseOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationModes)  Close Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| OpenLi | PFE 🡨MOD | BOOL | (relevant, if automatic mode is active, see OperationModes)  Open Command from Internal Link  1: Open Command execute  0: no operation |
| CloseLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active, see OperationModes)  Close Command from Internal Link  1: Close Command execute  0: no operation |
| Ctrl | PFE 🡨 MOD | BOOL | Valve Control  1: PosCtrl = PosInt (if SrcIntAct = true) || PosExt (if SrcExtAct = true)  0: PosCtrl = 0.0; |
| OpenFbkEn | PFE 🡨 MOD | BOOL | Open Feedback Enable  1:enabled  0:disabled |
| CloseFbkEn | PFE 🡨 MOD | BOOL | Close Feedback Enable  1:enabled  0:disabled |
| OpenFbk | PFE 🡨 MOD | BOOL | Open Feedback Signal  1: Open End Position achieved, if OpenFbkEn = false, OpenFbk = Ctrl  0: Open End Position not achieved |
| CloseFbk | PFE 🡨 MOD | BOOL | Close Feedback Signal  1: Close End Position achieved, if CloseFbkEn = false, CloseFbk = !Ctrl  0: Close End Positon not achieved |
| PosSclMin | PFE 🡨 MOD | REAL | Position Setpoint Scale Low Limit (see ScaleSettings) |
| PosSclMax | PFE 🡨 MOD | REAL | Position Setpoint Scale High Limit (see ScaleSettings) |
| PosUnit | PFE 🡨 MOD | INT | Position Setpoint Unit (see UnitSettings) |
| PosInt | PFE 🡨 MOD | REAL | (relevant, if internal source is active, see OperationModes)  Position Internal Setpoint (from Module Automation) |
| PosExt | PFE 🡪 MOD | REAL | (relevant, if external source is active, see OperationModes)  Position External Setpoint (from Process Control Level) |
| PosMin | PFE 🡨 MOD | REAL | Position Setpoint Low Limit (see ValueLimitation) |
| PosMax | PFE 🡨 MOD | REAL | Position Setpoint High Limit (see ValueLimitation) |
| SafePos | PFE 🡨 MOD | BOOL | Safe Position  1: PosMax  0: PosMin |
| PosCtrl | PFE 🡨 MOD | REAL | Position Setpoint  (see Ctrl Description) |
| PosFbkEn | PFE 🡨 MOD | BOOL | Position Feedback Enable  1:enabled  0:disabled |
| PosFbk | PFE 🡨 MOD | REAL | Position Feedback Signal  If PosFbkEn = false, PosFbk = Pos |
| PermEn | PFE 🡨 MOD | BOOL | Enables the Permission Lock  1:enabled  0:disabled |
| Permit | PFE 🡨 MOD | BOOL | Permit locks the Entity to be activated  1:permission is given  0:permission is not given |
| IntlEn | PFE 🡨 MOD | BOOL | Enables the Interlock Lock  1:enabled  0:disabled |
| Interlock | PFE 🡨 MOD | BOOL | Interlock locks the Entity to the safe state  1: interlock is not active  0: interlocking is active, must not be reset (see Reset) |
| ProtEn | PFE 🡨 MOD | BOOL | Enables the Protection Lock  1:enabled; 0:disabled |
| Protect | PFE 🡨 MOD | BOOL | Protect locks the Entity to the safe state – a reset is required  1: protection is not active  0: protection is active, must be reset (see Reset) |
| ResetOp | PFE 🡪 MOD | BOOL | Reset from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| ResetLi | PFE 🡨 MOD | BOOL | Reset from internal link  1:Reset execute  0: no operation |

### MonAnaVlv

Die Schnittstelle MonAnaVlv erweitert die Schnittstelle AnaVlv um die Standard-Funktion Feedback Monitoring. Überwachung von gesteuerten (dynamische Fehlfunktion) oder ungesteuerten (statische Fehlfunktion) Zustandsänderungen inklusive deren Parametrierung, sowie einer Positionsüberwachung.

Das Position Monitoring erweitert die Standard-Funktion Feedback-Monitoring um Variablen zur Überwachung einer Position bei analogen Ventilen. Die Definitionen für eine dynamische oder statische Fehlfunktion bleiben auch hier gültig. Zusätzlich wird überwacht, ob sich ein Ventil öffnet, schließt oder seine Position erreicht hat. Vergleichbar wie bei der Erkennung dynamischer Fehlfunktionen, wird auch hier die Zeit gemessen zwischen der Änderung der Positionsansteuerung bis zu dem Zeitpunkt, an dem die Position erreicht wird. Ist die Zeit MonTiPos vor Erreichung abgelaufen, wird der Fehler MonPosErr ausgelöst.

Die Feedback-Signale für den geöffneten und geschlossenen Zustand werden in dieser Schnittstelle, im Vergleich zu AnaVlv, unterschiedlich ermittelt. Für den Fall, dass es keine binären Endlagesensoren für Offen und Geschlossen gibt, wird das OpenFbk und CloseFbk über die Positionsrückmeldung gebildet.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/AnaVlv | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| MonEn | PFE 🡨🡪 MOD | BOOL | Monitor Enable  1:enabled  0:disabled |
| MonSafePos | PFE 🡨 MOD | BOOL | Set the Error Behaviour, if an error occurs  1: after an error occurs, the safe pos will be set  0: after an error occurs, the entity holds the state |
| MonStatErr | PFE 🡨 MOD | BOOL | Static Error active  1:active  0:inactive |
| MonDynErr | PFE 🡨 MOD | BOOL | Dynamic Error active  1:active  0:inactive |
| MonStatTi | PFE 🡨🡪 MOD | REAL | Monitor Time for uncontrolled changes [s] |
| MonDynTi | PFE 🡨🡪 MOD | REAL | Monitor Time for controlled changes [s] |
| PosOpngFbk | PFE 🡨 MOD | BOOL | 1: Position is changing to open  0: Positon is not changing to open |
| PosClsngFbk | PFE 🡨 MOD | BOOL | 1:Position is changing to close  0: Positon is not changing to close |
| PosReachedFbk | PFE 🡨 MOD | BOOL | Position reached  Position is reached, ReachedFbk = (PosFbk – Pos) with in +/- PosTolerance |
| PosTolerance | PFE 🡨🡪 MOD | REAL | Position Tolearance Value for Calculation |
| PosDefClose | PFE 🡨🡪 MOD | REAL | Position to define close position |
| PosDefOpen | PFE 🡨🡪 MOD | REAL | Position to define open position |
| MonPosTi | PFE 🡨🡪 MOD | REAL | Monitor Time until position is reached [s] |
| MonPosErr | PFE 🡨 MOD | BOOL | Position Error active  1: if the MonTiPos is elapsed before ReachedFbk = true  0: no error |
| *CloseFbk* | PFE 🡨 MOD | *BOOL* | Position Close achieved  If CloseFbkEn = false, CloseFbk = PosFbk <= PosDiClose |
| *OpenFbk* | PFE 🡨 MOD | *BOOL* | Position Open achieved  If OpenFbkEn = false, CloseFbk = PosFbk >= PosDiOpen |

## Bistabile Antriebe

Bistabile Antriebe können sowohl für Rechts- und Linkslauf nur ein- und ausgeschaltet werden.

### BinDrv

BinDrv ermöglicht der PFE das Ein- und Ausschalten eines Antriebs sowie die Vorgabe der Drehrichtung. Für die Drehrichtung gibt es jeweils einen booleschen Ausgangswert, der rechts oder linksdrehend signalisiert. Die Drehrichtung wird entweder intern oder extern ausgewählt in Abhängigkeit von dem Operation Mode. Darüber hinaus kann der Betriebsmodus ausgewählt werden. Es stehen die beiden Modi „Auto“ und „Manuell“ zur Verfügung. Im „Auto“-Modus werden die internen Steuersignale angewendet und im „Manuell“-Modus werden die externen Signale verwendet. Der Antrieb kann in beide Richtungen betrieben werden. Die Modulautomatisierung meldet vorliegende Werte zurück.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OeprationModes, only ExtendedOperationMode used) |
| FwdEn | PFE 🡨 MOD | BOOL | Forward Enable  1:enabled  0:disabled |
| RevEn | PFE 🡨 MOD | BOOL | Reverse Enable  1:enabled  0:disabled |
| StopOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationModes)  Stop Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| FwdOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active and FwdEn is true, see OperationModes)  Forward Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| RevOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active and RevEn is true, see OperationModes)  Reverse Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| StopLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active, see OperationModes)  Stop Command from Internal Link  1:Stop Command execute  0: no operation |
| FwdLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active and FwdEn is true, see OperationModes)  Forward Command from Internal Link  1:Forward Command execute  0: no operation |
| RevLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active and RevEn is true, see OperationModes)  Reverse Command from Internal Link  1:Reverse Command execute  0: no operation |
| FwdCtrl | PFE 🡨 MOD | BOOL | Forward Control  1:active; 0:inactive |
| RevCtrl | PFE 🡨 MOD | BOOL | Reverse Control  1:active; 0:inactive |
| RevFbkEn | PFE 🡨 MOD | BOOL | Reverse Feedback Enable  1:Reverse Feedback Signal available |
| FwdFbkEn | PFE 🡨 MOD | BOOL | Forward Feedback Enable  1:Forward Feedback Signal available |
| RevFbk | PFE 🡨 MOD | BOOL | Reverse Feedback Signal  1: Reverse running , if RevFbkEn = false, RevFbk = RevCtrl  0: not reverse running |
| FwdFbk | PFE 🡨 MOD | BOOL | Forward Feedback Signal  1:Forward running , if FwdFbkEn = false, FwdFbk = FwdCtrl  0: not forward running |
| SafePos | PFE 🡨 MOD | BOOL | Safe Position  1:running; 0:stopped |
| Trip | PFE 🡨 MOD | BOOL | Drive Protection Indicator  0:tripped, 1:no error |
| PermEn | PFE 🡨 MOD | BOOL | Enables the Permission Lock  1:enabled; 0:disabled |
| Permit | PFE 🡨 MOD | BOOL | Permit locks the Entity to be activated  1:permission is given; 0:permission is not given |
| IntlEn | PFE 🡨 MOD | BOOL | Enables the Interlock Lock  1:enabled; 0:disabled |
| Interlock | PFE 🡨 MOD | BOOL | Interlock locks the Entity to the safe state  1: interlock is not active; 0: interlocking is active, must not be reset (see Reset)  In this case deactivation (interlock = 0) means motor will be shutdown. |
| ProtEn | PFE 🡨 MOD | BOOL | Enables the Protection Lock  1:enabled; 0:disabled |
| Protect | PFE 🡨 MOD | BOOL | Protect locks the Entity to the safe state – a reset is required  1: protection is not active; 0: protection is active, must be reset (see Reset)  In this case deactivation (protect = 0) means motor will be shutdown. |
| ResetOp | PFE 🡪 MOD | BOOL | Reset from Operator  0🡪1: request from PFE, 1🡪0: acknowledge from Module (see Handshake) |
| ResetLi | PFE 🡨 MOD | BOOL | Reset from internal link  1:Reset execute |

### MonBinDrv

Die Schnittstelle MonBinDrv erweitert die Schnittstelle BinDrv um die Standard-Funktion Feedback Monitoring. Diese definiert übergreifend die Überwachung von gesteuerten (dynamische Fehlfunktion) oder ungesteuerten (statische Fehlfunktion) Zustandsänderungen inklusive deren Parametrierung. Die Sicherheitsposition eines Antriebs ist immer die Ruhelage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/BinDrv | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| MonEn | PFE 🡨🡪 MOD | BOOL | Monitor Enable  1:enabled; 0:disabled |
| MonSafePos | PFE 🡨 MOD | BOOL | Monitor SafePos  1: go to SafePos, if Monitoring Error occurs |
| MonStatErr | PFE 🡨 MOD | BOOL | Static Error active  1:active; 0:inactive |
| MonDynErr | PFE 🡨 MOD | BOOL | Dynamic Error active  1:active; 0:inactive |
| MonStatTi | PFE 🡨🡪 MOD | REAL | Monitor Time for uncontrolled changes [s] |
| MonDynTi | PFE 🡨🡪 MOD | REAL | Monitor Time for controlled changes [s] |

## Analoge Antriebe

Analoge Antriebe können für Rechts- und Linkslauf ein- und ausgeschaltet und mit variabler Geschwindigkeit betrieben werden.

### AnaDrv

AnaDrv ermöglicht der PFE das Ein- und Ausschalten eines Antriebs, sowie die Vorgabe einer entsprechenden Drehzahl und Drehrichtung. Für die Drehrichtung gibt es jeweils einen booleschen Ausgangswert, der rechts oder linksdrehend signalisiert. Die Drehrichtung wird entweder intern oder extern ausgewählt in Abhängigkeit von dem Operation Mode. Darüber hinaus kann der Betriebsmodus ausgewählt werden. Es stehen die beiden Modi „Auto“ und „Manuell“ zur Verfügung. Im „Auto“-Modus werden die internen Steuersignale angewendet und im „Manuell“-Modus werden die externen Signale verwendet. Eine Analogwertvorgabe der Drehzahl ermöglicht das Einstellen einer Drehzahl. Quelle der Drehzahlvorgabe kann nun durch die Auswahl Extern oder Intern vorgenommen werden. Ist der Motor deaktiviert, wird der Sollwert für die Drehzahl auf 0 gesetzt.

In der Spalte der spezifischen Beschreibung sind zusätzliche Bemerkungen enthalten, wie die Operation Moder Schnittstellen Auswirkungen auf die anderen Schnittstellen-Bestandteile hat. Die Default-Quelle wird im Programm der Modulautomatisierung, je nach Verwendung, definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| OpMode | PFE 🡨🡪 MOD | DWORD | Operation Mode Mask  (see OeprationModes, ExtendedOperation- and SourceMode used) |
| FwdEn | PFE 🡨 MOD | BOOL | Forward Enable  1:enabled  0:disabled |
| RevEn | PFE 🡨 MOD | BOOL | Reverse Enable  1:enabled  0:disabled |
| StopOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active, see OperationModes)  Stop Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| FwdOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active and FwdEn is true, see OperationModes)  Forward Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| RevOp | PFE 🡪 MOD | BOOL | (relevant, if manual mode is active and RevEn is true, see OperationModes)  Reverse Command from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| StopLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active, see OperationModes)  Stop Command from Internal Link  1:Stop Command executed  0: no operation |
| FwdLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active and FwdEn is true, see OperationModes)  Forward Command from Internal Link  1:Forward Command executed  0: no operation |
| RevLi | PFE 🡨 MOD | BOOL | (relevant, if automatic mode is active and RevEn is true, see OperationModes)  Reverse Command from Internal Link  1:Reverse Command executed  0: no operation |
| FwdCtrl | PFE 🡨 MOD | BOOL | Forward Control  1:active  0:inactive |
| RevCtrl | PFE 🡨 MOD | BOOL | Reverse Control  1:active  0:inactive |
| RevFbkEn | PFE 🡨 MOD | BOOL | Reverse Feedback Enable  1: Reverse Feedback available  0: no reverse feedback available |
| FwdFbkEn | PFE 🡨 MOD | BOOL | Forward Feedback Enable  1: Forward Feedback available  0: no forward feedback available |
| RevFbk | PFE 🡨 MOD | BOOL | Reverse Feedback Signal  1: Reverse running , if RevFbkEn = false, RevFbk = RevCtrl  0: is not reverse running |
| FwdFbk | PFE 🡨 MOD | BOOL | Forward Feedback Signal  1: Forward running , if FwdFbkEn = false, FwdFbk = FwdCtrl  0: is not forward running |
| SafePos | PFE 🡨 MOD | BOOL | Safe Position  1: running  0: stopped |
| Trip | PFE 🡨 MOD | BOOL | Drive Protection Indicator  1: no problem  0: Drive Protection triggered |
| RpmSclMin | PFE 🡨 MOD | REAL | RPM Setpoint Scale Low Limit (see ScaleSettings) |
| RpmSclMax | PFE 🡨 MOD | REAL | RPM Setpoint Scale High Limit (see ScaleSettings) |
| RpmUnit | PFE 🡨 MOD | INT | RPM Setpoint Unit (see UnitSettings) |
| RpmInt | PFE 🡨 MOD | REAL | (relevant, if internal source is active, see OperationModes)  RPM Internal Setpoint |
| RpmExt | PFE 🡪 MOD | REAL | (relevant, if external source is active, see OperationModes)  RPM External Setpoint |
| RpmMin | PFE 🡨 MOD | REAL | RPM Setpoint Low Limit (see ValueLimitation) |
| RpmMax | PFE 🡨 MOD | REAL | RPM Setpoint High Limit (see ValueLimitation) |
| Rpm | PFE 🡨 MOD | REAL | RPM Setpoint |
| RpmFbk | PFE 🡨 MOD | REAL | RPM Feedback Signal |
| PermEn | PFE 🡨 MOD | BOOL | Enables the Permission Lock  1:enabled  0:disabled |
| Permit | PFE 🡨 MOD | BOOL | Permit locks the Entity to be activated  1:permission is given  0:permission is not given |
| IntlEn | PFE 🡨 MOD | BOOL | Enables the Interlock Lock  1:enabled  0:disabled |
| Interlock | PFE 🡨 MOD | BOOL | Interlock locks the Entity to the safe state  1: interlock is not active  0: interlocking is active, must not be reseted (see Reset)  In this case deactivation (interlock = 0) means motor will be shutdown. |
| ProtEn | PFE 🡨 MOD | BOOL | Enables the Protection Lock  1:enabled  0:disabled |
| Protect | PFE 🡨 MOD | BOOL | Protect locks the Entity to the safe state – a reset is required  1: protection is not active  0: protection is active, must be reset (see Reset)  In this case deactivation (protect = 0) means motor will be shutdown. |
| ResetOp | PFE 🡪 MOD | BOOL | Reset from Operator  0🡪1: request from PFE  1🡪0: acknowledge from Module (see Handshake) |
| ResetLi | PFE 🡨 MOD | BOOL | Reset from Module  1:Reset execute  0: no operation |

### MonAnaDrv

Die Schnittstelle MonAnaDrv erweitert die Schnittstelle AnaDrv um die Standard-Funktion Feedback Monitoring. Überwachung von gesteuerten (dynamische Fehlfunktion) oder ungesteuerten (statische Fehlfunktion) Zustandsänderungen inklusive deren Parametrierung, sowie einer Drehzahlüberwachung.

Die Drehzahlüberwachung beinhaltet Variablen zur Überwachung einer Drehzahl, bzw. der Drehzahlabweichung, bei analogen Antrieben. Die Drehzahlabweichung wird auf eine Ober- und eine Untergrenze überwacht.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly/AnaDrv | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| MonEn | PFE 🡨🡪 MOD | BOOL | Monitor Enable  1:enabled  0:disabled |
| MonSafePos | PFE 🡨 MOD | BOOL | Set the Error Behaviour, if an error occurs  1: after an error occurs, the safe pos will be set  0: after an error occurs, the entity holds the state |
| MonStatErr | PFE 🡨 MOD | BOOL | Static Error active  1:active  0:inactive |
| MonDynErr | PFE 🡨 MOD | BOOL | Dynamic Error active  1:active  0:inactive |
| MonStatTi | PFE 🡨🡪 MOD | REAL | Monitor Time for uncontrolled changes [s] |
| MonDynTi | PFE 🡨🡪 MOD | REAL | Monitor Time for controlled changes [s] |
| RpmErr | PFE 🡪 MOD | REAL | RPM Error  RpmErr = Rpm – RpmFbk |
| RpmAHEn | PFE 🡨🡪 MOD | BOOL | Enable Alarm High Limit  1:enabled  0:disabled |
| RpmALEn | PFE 🡨🡪 MOD | BOOL | Enable Alarm Low Limit  1:enabled  0:disabled |
| RpmAHAct | PFE 🡨 MOD | BOOL | Alarm High Active  1:active  0:inactive |
| RpmALAct | PFE 🡨 MOD | BOOL | Alarm Low Active  1:active  0:inactive |
| RpmAHLim | PFE 🡨🡪 MOD | REAL | Limit Value for Alarm High |
| RpmALLim | PFE 🡨🡪 MOD | REAL | Limit Value for Alarm Low |
| StartingFbk | PFE 🡨 MOD | BOOL | Starting Feedback  1: drive is starting  0: drive is not starting |
| StoppingFbk | PFE 🡨 MOD | BOOL | Stopping Feedback  1:drive is stopping  0: drive is not stopping |

## Verriegelungsanzeige

Verriegelungsanzeigen visualisieren den aktuellen Zustand einer Verriegelung

### LockView4

LockView4 repräsentiert eine Anzeige einer logischen Grundoperation (UND / ODER) mit vier Eingängen, die jeweils nochmals invertiert werden können. Folgend aus den Interlock-, Permit- und Protect-Variablen an den Schnittstellen mit Verriegelungsfunktion sind auch hier die Signale als Low-Active (Drahtbruch-Stil) auszuführen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| Logic | PFE 🡨 MOD | BOOL | Logical Behavior  0: OR  1: AND |
| In1En | PFE 🡨 MOD | BOOL | Locking Input 1 – Enable  0: Input not used  1: Input is used |
| In1 | PFE 🡨 MOD | BOOL | Locking Input 1 - Value |
| In1QC | PFE 🡨 MOD | BYTE | Locking Input 1 – Quality Code |
| In1Inv | PFE 🡨 MOD | BOOL | Locking Input 1 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In1Txt | PFE 🡨 MOD | STRING | Locking Input 1 – Additional Text |
| In2En | PFE 🡨 MOD | BOOL | Locking Input 2 – Enable  0: Input not used  1: Input is used |
| In2 | PFE 🡨 MOD | BOOL | Locking Input 2 - Value |
| In2QC | PFE 🡨 MOD | BYTE | Locking Input 2 – Quality Code |
| In2Inv | PFE 🡨 MOD | BOOL | Locking Input 2 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In2Txt | PFE 🡨 MOD | STRING | Locking Input 2 – Additional Text |
| In3En | PFE 🡨 MOD | BOOL | Locking Input 3 – Enable  0: Input not used  1: Input is used |
| In3 | PFE 🡨 MOD | BOOL | Locking Input 3 - Value |
| In3QC | PFE 🡨 MOD | BYTE | Locking Input 3 – Quality Code |
| In3Inv | PFE 🡨 MOD | BOOL | Locking Input 3 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In3Txt | PFE 🡨 MOD | STRING | Locking Input 3 – Additional Text |
| In4En | PFE 🡨 MOD | BOOL | Locking Input 4 – Enable  0: Input not used  1: Input is used |
| In4 | PFE 🡨 MOD | BOOL | Locking Input 4 - Value |
| In4QC | PFE 🡨 MOD | BYTE | Locking Input 4 – Quality Code |
| In4Inv | PFE 🡨 MOD | BOOL | Locking Input 4 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In4Txt | PFE 🡨 MOD | STRING | Locking Input 4 – Additional Text |
| Out | PFE 🡨 MOD | BOOL | Logical Result Output |
| OutQC | PFE 🡨 MOD | BYTE | Locking Result Output – Quality Code |

### LockView8

LockView8 repräsentiert eine Anzeige einer logischen Grundoperation (UND / ODER) mit acht Eingängen, die jeweils nochmals invertiert werden können. Sie baut auf der Basis-Schnittstelle LockView4 auf.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| In5En | PFE 🡨 MOD | BOOL | Locking Input 5 – Enable  0: Input not used  1: Input is used |
| In5 | PFE 🡨 MOD | BOOL | Locking Input 5 - Value |
| In5QC | PFE 🡨 MOD | BYTE | Locking Input 5 – Quality Code |
| In5Inv | PFE 🡨 MOD | BOOL | Locking Input 5 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In5Txt | PFE 🡨 MOD | STRING | Locking Input 5 – Additional Text |
| In6En | PFE 🡨 MOD | BOOL | Locking Input 6 – Enable  0: Input not used  1: Input is used |
| In6 | PFE 🡨 MOD | BOOL | Locking Input 6 - Value |
| In6QC | PFE 🡨 MOD | BYTE | Locking Input 6 – Quality Code |
| In6Inv | PFE 🡨 MOD | BOOL | Locking Input 6 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In6Txt | PFE 🡨 MOD | STRING | Locking Input 6 – Additional Text |
| In7En | PFE 🡨 MOD | BOOL | Locking Input 7 – Enable  0: Input not used  1: Input is used |
| In7 | PFE 🡨 MOD | BOOL | Locking Input 7 - Value |
| In7QC | PFE 🡨 MOD | BYTE | Locking Input 7 – Quality Code |
| In7Inv | PFE 🡨 MOD | BOOL | Locking Input 7 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In7Txt | PFE 🡨 MOD | STRING | Locking Input 7 – Additional Text |
| In8En | PFE 🡨 MOD | BOOL | Locking Input 8 – Enable  0: Input not used  1: Input is used |
| In8 | PFE 🡨 MOD | BOOL | Locking Input 8 - Value |
| In8QC | PFE 🡨 MOD | BYTE | Locking Input 8 – Quality Code |
| In8Inv | PFE 🡨 MOD | BOOL | Locking Input 8 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In8Txt | PFE 🡨 MOD | STRING | Locking Input 8 – Additional Text |

### LockView16

LockView16 repräsentiert eine Anzeige einer logischen Grundoperation (UND / ODER) mit sechzehn Eingängen, die jeweils nochmals invertiert werden können. Sie baut auf der Basis-Schnittstelle LockView8 auf.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| In9En | PFE 🡨 MOD | BOOL | Locking Input 9 – Enable  0: Input not used  1: Input is used |
| In9 | PFE 🡨 MOD | BOOL | Locking Input 9 - Value |
| In9QC | PFE 🡨 MOD | BYTE | Locking Input 9 – Quality Code |
| In9Inv | PFE 🡨 MOD | BOOL | Locking Input 9 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In9Txt | PFE 🡨 MOD | STRING | Locking Input 9 – Additional Text |
| In10En | PFE 🡨 MOD | BOOL | Locking Input 10 – Enable  0: Input not used  1: Input is used |
| In10 | PFE 🡨 MOD | BOOL | Locking Input 10 - Value |
| In10QC | PFE 🡨 MOD | BYTE | Locking Input 10 – Quality Code |
| In10Inv | PFE 🡨 MOD | BOOL | Locking Input 10 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In10Txt | PFE 🡨 MOD | STRING | Locking Input 10 – Additional Text |
| In11En | PFE 🡨 MOD | BOOL | Locking Input 11 – Enable  0: Input not used  1: Input is used |
| In11 | PFE 🡨 MOD | BOOL | Locking Input 11 - Value |
| In11QC | PFE 🡨 MOD | BYTE | Locking Input 11 – Quality Code |
| In11Inv | PFE 🡨 MOD | BOOL | Locking Input 11 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In11Txt | PFE 🡨 MOD | STRING | Locking Input 11 – Additional Text |
| In12En | PFE 🡨 MOD | BOOL | Locking Input 12 – Enable  0: Input not used  1: Input is used |
| In12 | PFE 🡨 MOD | BOOL | Locking Input 12 - Value |
| In12QC | PFE 🡨 MOD | BYTE | Locking Input 12 – Quality Code |
| In12Inv | PFE 🡨 MOD | BOOL | Locking Input 12 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In12Txt | PFE 🡨 MOD | STRING | Locking Input 12 – Additional Text |
| In13En | PFE 🡨 MOD | BOOL | Locking Input 13 – Enable  0: Input not used  1: Input is used |
| In13 | PFE 🡨 MOD | BOOL | Locking Input 13 - Value |
| In13QC | PFE 🡨 MOD | BYTE | Locking Input 13 – Quality Code |
| In13Inv | PFE 🡨 MOD | BOOL | Locking Input 13 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In13Txt | PFE 🡨 MOD | STRING | Locking Input 13 – Additional Text |
| In14En | PFE 🡨 MOD | BOOL | Locking Input 14 – Enable  0: Input not used  1: Input is used |
| In14 | PFE 🡨 MOD | BOOL | Locking Input 14 - Value |
| In14QC | PFE 🡨 MOD | BYTE | Locking Input 14 – Quality Code |
| In14Inv | PFE 🡨 MOD | BOOL | Locking Input 14 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In14Txt | PFE 🡨 MOD | STRING | Locking Input 14 – Additional Text |
| In15En | PFE 🡨 MOD | BOOL | Locking Input 15 – Enable  0: Input not used  1: Input is used |
| In15 | PFE 🡨 MOD | BOOL | Locking Input 15 - Value |
| In15QC | PFE 🡨 MOD | BYTE | Locking Input 15 – Quality Code |
| In15Inv | PFE 🡨 MOD | BOOL | Locking Input 15 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In15Txt | PFE 🡨 MOD | STRING | Locking Input 15 – Additional Text |
| In16En | PFE 🡨 MOD | BOOL | Locking Input 16 – Enable  0: Input not used  1: Input is used |
| In16 | PFE 🡨 MOD | BOOL | Locking Input 16 - Value |
| In16QC | PFE 🡨 MOD | BYTE | Locking Input 16 – Quality Code |
| In16Inv | PFE 🡨 MOD | BOOL | Locking Input 16 – Inversion Enable  0: Input not inverted  1: Input inverted |
| In16Txt | PFE 🡨 MOD | STRING | Locking Input 16 – Additional Text |

## Zeichenkettenanzeigen

### StrView

StrView definiert eine Schnittstelle zur Anzeige eines Strings aus dem Modulautomatisierungssystem in einem Visualisierungssystem

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface Definition** | | | |
| **Parent Interface** | | | |
| MTPDataObjectSUCLib/DataAssembly | | | |
| **Alias** | **Access** | **Type** | **Specific Description** |
| Text | PFE 🡨 MOD | STRING | Text Value out of the module automation |

# Anhang