



MANUAL

MANUAL DE INSTRUÇÕES DOS BLOCOS FUNCIONAIS

O-PAS™



OUT/25 - VERSÃO 2

smar
Technology Company

MANUAL DE INSTRUÇÕES DOS BLOCOS FUNCIONAIS



Consulte nossos
representantes



Rua Dr. Antônio Furlan Junior, 1028 - Sertãozinho, SP - CEP: 14170-480
orcamento@smar.com.br | +55 (16) 3946-3599 | www.smar.com.br

© Copyright 2025, Nova Smar S/A. Todos os direitos reservados. - 2025
Especificações e informações estão sujeitas a modificações.
Informações atualizadas dos endereços estão disponíveis em nosso site.

smar
Technology Company

ÍNDICE

Sumário

ÍNDICE	3
BLOCOS FUNCIONAIS O-PAS™	5
Introdução	5
Objetos comuns O-PAS™ – nível de bloco funcional	5
Tipos de Blocos Funcionais	6
Tipo FBOptions	8
Parâmetro FBShedOptionsType	8
OCFB_Type	9
OCFB_OutType	10
OCFB_InType	10
ControlFunctionBlockType	10
Lógica Funcional Comum	14
Entrada Analógica (AI)	15
Lógica Funcional	15
LinearizationType	22
Implementação de Referência IEC 61131 – Variáveis e Código	23
Descrição dos Parâmetros do Bloco	27
Saída Analógica (AO)	29
Lógica Funcional	29
Modelo de Informação OCF	31
Implementação de Referência IEC 61131 – Variáveis e Código	38
Descrição dos Parâmetros do Bloco	43
Controle PID (PID)	46
Lógica Funcional	46
Modelo de Informação OCF	50
Implementação de Referência IEC 61131 – Variáveis e Código	60
Descrição dos Parâmetros do Bloco	65
Entrada Discreta de Dois Estados (DI)	68
Lógica Funcional	68
Modelo de Informação OCF	69
Implementação de Referência IEC 61131 – Variáveis e Código	74
Descrição dos Parâmetros do Bloco	76
Saída Discreta de Dois Estados (DO)	77
Lógica Funcional	77
Modelo de Informação OCF	78
Implementação de Referência IEC 61131 – Variáveis e Código	83
Descrição dos Parâmetros do Bloco	86
Entrada Discreta Multiestados (MDI)	88
Lógica Funcional	88
Modelo de Informação OCF	89
Implementação de Referência IEC 61131 – Variáveis e Código	93
Descrição dos Parâmetros do Bloco	95
Saída Discreta Multiestados (MDO)	96
Lógica Funcional	96
Modelo de Informação OCF	97
Implementação de Referência IEC 61131 – Variáveis e Código	103
Descrição dos Parâmetros do Bloco	106

IMPORTANTE

O-PAS™ é uma marca registrada de **The Open Group**.

O-PAS™ (Open Process Automation Standard) é um conjunto de especificações desenvolvido pela **The Open Group**. Este documento contém referências e partes de conteúdo técnico baseados neste padrão. Trechos e figuras originados do **O-PAS™** são usados apenas para fins de referência técnica e não constituem redistribuição da norma.

O uso das informações técnicas aqui apresentadas está em conformidade com as políticas de uso e licenciamento aplicáveis aos membros do **Open Process Automation Forum (OPAF)**.

Nenhuma parte deste documento deve ser interpretada como uma certificação, endosso ou afiliação formal da **The Open Group**, exceto quando expressamente autorizado por ela.

FOUNDATION é uma marca registrada de Fieldbus Foundation.

OPC Foundation é uma marca registrada de OPC Foundation.

BLOCOS FUNCIONAIS O-PAS™

Introdução

O núcleo de uma aplicação de controle compatível com o padrão O-PAS™ consiste em um conjunto de blocos de funções, cada um realizando cálculos úteis. O exemplo clássico é um "loop PID" que é normalmente representado por um bloco funcional que lida com o sinal de entrada (escala, caracterização etc.), um bloco funcional para executar o algoritmo de controle e um bloco funcional para acionar a saída para o campo.

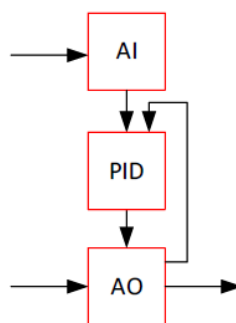


Figura 1 – Loop PID

No exemplo ilustrado na figura acima, temos três blocos funcionais especificados pelo O-PAS™. O bloco funcional de entrada analógica (AI) recebe o valor de um sensor ou outra fonte. O bloco funcional PID executa o cálculo e recebe feedback do bloco funcional saída analógica (AO). O bloco funcional AO recebe um sinal de leitura do campo e gera o valor do campo. O bloco funcional PID saberá, através de seu link de retorno, as informações e status do bloco AO e dispositivo de campo.

Objetos comuns O-PAS™ – nível de bloco funcional

OBJETO	DESCRIÇÃO
Pin do Bloco Funcional	Estas são as entradas e saídas do bloco funcional (dados e gatilhos).
Bloco Funcional	Os blocos funcionais descrevem a função entre variáveis de entrada e saída. As entradas e saídas dos blocos são conectadas entre si por meio de conexões ou linhas no ambiente de programação.
Entrada Trigger	Uma entrada conectável ao bloco funcional que aciona uma ação no bloco funcional. Pode ser conectado a uma saída de disparo de um bloco funcional. Por exemplo, um evento IEC 61499 ou um sinal de temporizador que aciona um bloco funcional cíclico.
Saída Trigger	Uma saída de um bloco funcional que pode ser uma entrada de disparo (trigger) para uma ou mais entradas de disparo. Por exemplo, um evento IEC 61499.
Referência de Evento	Associação definida entre uma entrada e saída de disparo (trigger).
Dado de Entrada	Uma entrada de dados para um bloco funcional que pode ser conectada a uma saída de dados de um bloco funcional.
Dado de Saída	Uma saída de dados de um bloco funcional que pode ser conectada a uma ou mais entradas de dados de bloco.
Referência de Sinal	Associação definida entre entrada e saída de dados.
Interação do Bloco Funcional	Representa uma interface que expõe parâmetros internos do bloco funcional. A intenção da interface é fornecer dados para monitoramento, operação, ajuste e configuração do bloco funcional para clientes OPC UA.
Parâmetros	Parâmetros de interação do bloco funcional. Destinado à operação e configuração.
Parâmetro de Configuração	Representa dados de blocos funcionais que definem o comportamento de aplicações O-PAS™. Não se destina a ser conectado a outro bloco funcional. A capacidade de alterar parâmetros de configuração pode depender do estado operacional do bloco funcional e da permissão de acesso do usuário.
Parâmetro de Ajuste	Um parâmetro que representa dados de blocos funcionais que otimiza o comportamento de aplicações O-PAS™. Não se destina a ser conectado a outro bloco funcional. Pode ser alterado durante a execução do bloco. A capacidade de

OBJETO	DESCRIÇÃO
	alteração pode depender do estado operacional do bloco funcional e da permissão de acesso do usuário.
Parâmetro Operacional	Parâmetros e métodos úteis durante a operação normal, como dados de processo.
Variável de Status	Uma variável que está disponível no bloco funcional. Representa dados de blocos funcionais que devem ser monitorados por aplicações O-PAS™. Não se destina a ser conectado a outro bloco funcional. Um exemplo de variável de status é erro de bloco ou modo atual do bloco funcional.

Tipos de Blocos Funcionais

Variáveis declaradas para um **FunctionBlockType** são referenciadas usando referências genéricas de **HasComponent**. Para diferenciar a finalidade das variáveis individuais, existem referências não hierárquicas adicionais dos grupos funcionais para as variáveis. Eles indicam a(s) função(ões) que uma variável assume no modelo de bloco funcional O-PAS™. Todas as referências não hierárquicas começando nos grupos funcionais devem apontar para variáveis (ou filhos de variáveis) no nó do bloco funcional que contém esse grupo funcional. As descrições dos grupos funcionais O-PAS™ estão listadas na Tabela **Descrições de grupos funcionais O-PAS™**.

O componente **<FunctionBlocks>** de **FunctionBlockType** é um espaço reservado opcional que indica que um bloco funcional O-PAS™ pode conter um ou mais blocos funcionais. Este conceito também é conhecido como bloco funcional composto na IEC 61499; não há conceito equivalente na IEC 61131 em que todas as Unidades de Organização de Programa (POUs) podem conter blocos funcionais.

Um produto que implementa **FunctionBlockType** deve incluir pelo menos um “OptionalPlaceholder” OPC UA com os seguintes BrowseNames: **<TriggerInputs>**, **<TriggerOutputs>**, **<DataOutputs>**, **<DataInputs>**, **<ConfigurationParams>**, **<TuningParams>**, **<StatusVariables>**, **<DiagnosticParams>** e **<OperationalParams>**

References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de 0:BaseObjectType definido em OPC 10000-5.					
0:HasSubtype	ObjectType	OCFB_Type	Definido na seção OCFB_Type.		
0:HasComponent	Object	<FunctionBlocks>		FunctionBlockType	OptionalPlaceholder
0:HasComponent	Variable	Error	0:BaseDataType	FBErrorType	Optional
0:HasComponent	Variable	ExtendedError	ExtErrorStructure	0:BaseDataVariable	Optional
0:HasComponent	Variable	Mode	0:UInt	FBModeType	Optional
0:HasComponent	Object	Options		FBOptionsType	Optional
0:HasComponent	Object	ShedOptions		FBShedOptionsType	Optional
0:HasComponent	Variable	<TriggerInputs>	0:UInt32	TriggerType	OptionalPlaceholder
0:HasComponent	Variable	<TriggerOutputs>	0:UInt32	TriggerType	OptionalPlaceholder
0:HasComponent	Variable	<DataOutputs>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<DataInputs>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<ConfigurationParams>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<TuningParams>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<StatusVariables>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<OperationalParams>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Variable	<DiagnosticParams>	0:BaseDataType	0:BaseDataVariableType	OptionalPlaceholder
0:HasComponent	Object	Inputs		2:FunctionalGroupType	Optional
0:HasComponent	Object	Outputs		2:FunctionalGroupType	Optional
0:HasComponent	Object	Status		2:FunctionalGroupType	Optional
0:HasComponent	Object	Operational		2:FunctionalGroupType	Optional
0:HasComponent	Object	Configuration		2:FunctionalGroupType	Optional
0:HasComponent	Object	Tuning		2:FunctionalGroupType	Optional
0:HasComponent	Object	Diagnostics		2:FunctionalGroupType	Optional

Tabela 1 - Definição de FunctionBlockType

Error é uma instância de **FBErrorType**, subtipo de **OPC UA OptionSetType**, definido por um conjunto padrão de erros, mas os usuários podem adicionar seus próprios.

ExtendedError permite que subtipos forneçam informações mais detalhadas sobre os erros.

Os grupos funcionais neste tipo são marcados como opcionais. Qualquer parâmetro pode ser organizado em vários grupos funcionais ao mesmo tempo.

OBJETO	DESCRIÇÃO (DE OPC 10000-100)
Configuration	Parâmetros que representam os itens de configuração do TopologyElement , definidos no OPC 10000-5. Se o bit CurrentWrite estiver definido no atributo AccessLevel , eles poderão ser modificados por clientes que tenham as permissões corretas.
Tuning	Parâmetros e métodos para otimizar o comportamento do TopologyElement .
Status	Parâmetros que descrevem o funcionamento geral do TopologyElement . Isso pode incluir parâmetros de diagnóstico.
Operational	Parâmetros e métodos úteis durante a operação normal, como dados de processo.
Diagnostics	Parâmetros e métodos de diagnóstico.

Tabela 2 - Descrições de grupos funcionais O-PAS™

Os blocos funcionais com referências não hierárquicas são mostrados na tabela seguinte.

SOURCE PATH	REFERENCETYPE	IS FORWARD	TARGET PATH
Inputs	0:Organizes	True	<DataInputs> or <TriggerInputs>
Outputs	0:Organizes	True	<DataOutputs> or <TriggerOutputs>
Tuning	0:Organizes	True	<TuningParams>
Configuration	0:Organizes	True	<ConfigurationParams>
Status	0:Organizes	True	<StatusVariables>
Operational	0:Organizes	True	<OperationalVariables> or <MethodCall>
Diagnostics	0:Organizes	True	<DiagnosticParams>

Tabela 3 - Referências não hierárquicas de grupos funcionais de blocos funcionais

Os componentes de <DataInputs> e <TriggerInputs> possuem referências não hierárquicas adicionais que são definidas na Tabela seguinte.

SOURCE PATH	REFERENCETYPE	IS FORWARD	TARGET PATH
<DataInputs>	ConnectedToOutput	True	<DataOutputs>
<TriggerInputs>	ConnectedToOutput	True	<TriggerOutputs>

Tabela 4 - Blocos Funcionais DataInputs e TriggerInputs Referências Adicionais

As referências não hierárquicas adicionais acima fornecem ligações de entradas de bloco para saídas de bloco e entradas de gatilho para saídas de gatilho (trigger).

A estrutura **ExtErrorStructure** contém um código de erro numérico e a descrição do erro em uma cadeia de caracteres legível. A estrutura está definida na tabela seguinte.

NOME ESTRUTURA	TIPO	DESCRIÇÃO
ExtErrorStructure	0:Structure	Subtipo de estrutura definida em OPC 10000-5.
ErrorCode	0:UInt32	Código de erro estendido do bloco funcional.
ErrorString	0:LocalizedText	O bloco de funções estendido como string de erro legível.

Tabela 5 - Estrutura de erro ExtErrorStructure

A representação **ExtErrorStructure** no **AddressSpace** é definida na tabela seguinte.

Attribute			Value		
BrowseName			ExtErrorStructure		
IsAbstract			FALSE		
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de estrutura definida em OPC 10000-5.					

Tabela 6 - Definição de ExtErrorStructure

Tipo FBOptions

O **FBOptionsType** é um tipo abstrato que fornece opções de funcionalidade comuns usadas por vários tipos de blocos funcionais. Cada tipo de bloco funcional definirá seu tipo de opção específico derivando desse tipo base e alterando a regra de modelagem da(s) opção(ões) apropriada(s) para **Mandatory**. O **FBOptionsType** é formalmente definido na tabela seguinte.

Attribute			Value		
BrowseName			FBOptionsType		
IsAbstract			TRUE		
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de 0:BaseObjectType de OPC 10000-5.					
0:HasProperty	Variable	BadIfLimited	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	DirectActing	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	IncreaseToClose	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	Invert	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	SpPvTrackInManual	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	TrackOn	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	UncertainIfLimited	0:Boolean	0:PropertyType	Optional
0:HasProperty	Variable	UseUncertainAsGood	0:Boolean	0:PropertyType	Optional

Tabela 7 - Definição de FBOptionsType

Esta lista não exclusiva de opções de bloco funcional é típica para controle de processo. O conjunto padrão de blocos funcionais O-PAS™, definido na seção “Information and Exchange Models: Function Blocks” da norma O-PAS™, usa esta lista curta e especifica opções adicionais que são usadas na implementação de referência. A tabela seguinte fornece as descrições para cada uma das opções definidas em **FBOptionsType**

OPÇÃO	DESCRIÇÃO
BadIfLimited	O status de qualidade de um valor é definido como ruim se seu valor de entrada estiver acima de seu limite alto ou abaixo de seu limite baixo. Esta opção tem prioridade sobre UncertainIfLimited.
DirectActing	A saída do bloco funcional aumenta com um aumento no valor do processo. Quando falso, a saída diminui com um aumento no valor do processo.
IncreaseToClose	Inverte a relação padrão da saída do bloco funcional para a atuação; ou seja, 100% é fechado e 0% é aberto.
Invert	Para valores de entrada ou saída discretos booleanos, altera 0 para 1 ou 1 para 0.
SpPvTrackInManual	O setpoint segue o valor do processo quando o bloco funcional está no modo manual.
TrackOn	Para um bloco funcional que fornece uma entrada configurável a ser rastreada pelo valor primário do bloco, esta opção permite que o rastreamento ocorra.
UncertainIfLimited	O status de qualidade de um valor é definido como ruim se o valor de entrada estiver acima do limite alto ou abaixo do limite baixo.
UseUncertainAsGood	Se o status de qualidade de um valor for incerto, trate-o como bom. Caso contrário, trate-o como ruim.

Tabela 8 - Descrições das opções dos blocos funcionais

Parâmetro FBShedOptionsType

O **FBShedOptionsType** é um parâmetro usado para configurar o comportamento desejado caso o tempo exceda, indicando comunicação perdida. Este parâmetro determina o modo shed real quando o setpoint não é atualizado dentro de um limite de tempo definido no modo remote-cascade. Além disso, ele determina se o modo shed será mantido caso o parâmetro remote-cascade seja atualizado após o evento ter ocorrido. O **FBShedOptionsType** é formalmente definido na tabela seguinte.

Attribute			Value		
BrowseName			FBShedOptionsType		
IsAbstract			FALSE		
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 0:OptionSetType definido em OPC 10000-5.					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText[]	0:PropertyType	Mandatory

Tabela 9 - Definição de FBShedOptionsType

Os tipos de dados para **FBShedOptionsType** são formalmente definidos na tabela seguinte:

NOME	VALOR	DESCRIÇÃO
Undefined	0	Inválido
NormalShedNormalReturn	1	O modo atual muda para o próximo modo não remoto de menor prioridade permitido, mas retorna ao modo remoto objetivo quando o computador remoto conclui o handshake de inicialização.
NormalShedNoReturn	2	O modo objetivo muda para o próximo modo não remoto de menor prioridade permitido. O modo remoto objetivo é perdido, então não há retorno a ele.
ShedToAutoNormalReturn	3	O modo atual muda para Automático, mas retorna ao modo remoto objetivo quando o computador remoto conclui o handshake de inicialização.
ShedToAutoNoReturn	4	O modo objetivo muda para Automático ao detectar uma condição shed.
ShedToManual	5	O modo atual muda para Manual, mas retorna ao modo remoto objetivo quando o computador remoto conclui o handshake de inicialização.
ShedToManualNoReturn	6	O modo objetivo muda para Manual na detecção de uma condição shed. Quando o modo objetivo é definido como Manual, os bits retidos serão definidos como zero (0).
ShedToRetainedTargetNormalReturn	7	O modo atual muda para o modo objetivo retido, mas retorna ao modo objetivo remoto quando o computador remoto conclui o handshake de inicialização.
ShedToRetainedTargetNoReturn	8	O modo objetivo muda para o modo objetivo retido na detecção de uma condição shed.

Tabela 10 – Tipos de dados de FBShedOptionsType

As opções de enumeração do modo shed 3, 4, 5 e 6 prevalecerão, mesmo se o atributo permitido do parâmetro do modo de bloco não incluir o modo shed especificado.

Se o modo shed especificado não for um modo permitido, então aparecerá a indicação “Erro de configuração” no erro de bloco.

OCFB_Type

Simplificadamente, **OCFB_Type** é uma realização de um bloco funcional de comunicação. A propriedade **TagVariableAlias** é usada por um mecanismo de execução para registrar ou descobrir um sinal no OCF.

Attribute			Value		
BrowseName			OCFB_Type		
IsAbstract			TRUE		
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de FunctionBlockType.					
0:HasSubtype			OCFB_OutType		
0:HasSubtype			OCFB_InType		
0:HasProperty	Variable	TagVariableAlias	0:String	0:PropertyType	Optional

Tabela 11 – Definição de OCFB_Type

OCFB_OutType

O **OCFB_OutType** é um subtipo de **OCFB_Type** definido na Seção “O-PAS™ OPC UA ObjectTypes” da norma O-PAS™, parte “Information and Exchange Models: Basic Configuration”. Ele é usado para tornar o **DataOutput** de um aplicativo O-PAS™ Layer-F disponível como um sinal OCF.

Attribute			Value		
BrowseName			OCFB_OutType		
IsAbstract			FALSE		
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de OCFB_Type.					
0:HasComponent	Variable	OutgoingVariable	0:BaseDataType	0:BaseDataVariableType	Mandatory

Tabela 12 – Definição OCFB_OutType

OCFB_InType

O **OCFB_InType** é um subtipo de **OCFB_Type** definido na Seção “O-PAS™ OPC UA ObjectTypes” da norma O-PAS™, parte “Information and Exchange Models: Basic Configuration”. Ele é usado para conectar a um sinal OCF, tornando os dados do sinal disponíveis para um aplicativo O-PAS™ Layer-F.

Attribute			Value		
BrowseName			OCFB_InType		
IsAbstract			FALSE		
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de OCFB_Type.					
HasComponent	Variable	ReceivedVariable	BaseDataType	0:BaseDataVariableType	Mandatory
HasComponent	Variable	UpdateInterval	Duration	0:BaseDataVariableType	Mandatory
HasComponent	Variable	DeadbandType	UInt32	0:BaseDataVariableType	Optional
HasComponent	Variable	DeadbandValue	Double	0:BaseDataVariableType	Optional

Tabela 13 – Definição OCFB_InType

ControlFunctionBlockType

Esta seção descreve e especifica o padrão do bloco funcional O-PAS™ definido para casos de uso de **controle de malha fechada**. O **ControlFunctionBlockType** é definido como um subtipo do **FunctionBlockType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration”.

O **ControlFunctionBlockType** torna obrigatórias as variáveis necessárias para o seguinte:

- Inicialização em cascata de controle “sem impacto”
- Proteção anti-windup
- Configuração do comportamento do usuário durante condições de falha
- Outras características esperadas de aplicações de controle de malha fechada

O conceito chave para controle de malha fechada é a interface de coordenação entre blocos que requer o uso de variáveis de Modo, Opções e Status conforme especificado nas subseções a seguir.

Os blocos de função O-PAS™ do perfil IEC 61131 são projetados para execução periódica. Em contraste, os blocos de função O-PAS™ do perfil IEC 61499 são projetados para execução baseada em eventos.

Em relação às unidades de conformidade normativa, o princípio geral é que o padrão O-PAS™ especifica os modelos de dados e interfaces entre blocos, mas não especifica ou exige a divulgação da implementação interna de um bloco funcional ou de qualquer módulo de software.

Attribute	Value				
BrowseName	ControlFunctionBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 4:FunctionBlockType definido em O-PAS™ Parte "Information and Exchange Models: Basic Configuration".					
0:HasComponent	Variable	4:Error	0:BaseData Type	4:FBErrorType	Mandatory
0:HasComponent	Variable	4:Mode	0:UInt	4:FBModeType	Mandatory
0:HasComponent	Object	4:Options		4:FBOptionsType	Mandatory
0:HasSubtype	ObjectType	AnalogInputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks"		
0:HasSubtype	ObjectType	AnalogOutputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		
0:HasSubtype	ObjectType	PIDControlBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		
0:HasSubtype	ObjectType	TwoStateDiscreteInputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		
0:HasSubtype	ObjectType	TwoStateDiscreteOutputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		
0:HasSubtype	ObjectType	MultiStateDiscreteInputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		
0:HasSubtype	ObjectType	MultistateDiscreteOutputBlockType	Definido na seção "O-PAS™ Function Blocks" da Parte "Information and Exchange Models: Function Blocks".		

Tabela 14 – Definição de O-PAS™ ControlFunctionBlockType

FBMode, **FBError** e **FBOptions** são comuns a todos os blocos no conjunto padrão de blocos de funções O-PAS™. **FBMode** e **FBError** são **VariableTypes** e **FBOptions** é um **ObjectType**. Todos estão formalmente definidos no O-PAS™ Parte "Information and Exchange Models: Basic Configuration".

FBOptions, que é uma abreviação de opções de bloco funcional, são usados para especificar as E/S ou o comportamento lógico de controle do conjunto padrão de bloco funcional O-PAS™. Embora as definições das **FBOptions** sejam definidas no O-PAS™ Parte "Information and Exchange Models: Basic Configuration", a explicação de cada uma das **FBOptions** é fornecida na tabela seguinte.

FBOptions BrowseName	Pertinência	Descrição
ActOnIR	PID	Quando a solicitação de inicialização for recebida via BackCallIn.Status , o SP.Value será ajustado dentro dos limites do setpoint para fornecer transferência sem sobressaltos (bumpless) quando a cascata for fechada.
BadIfLimited	AI	Se <i>In.Status=HighLimited</i> ou <i>LowLimited</i> , então <i>Out.Status=Bad</i> . <i>BadIfLimited</i> tem prioridade sobre <i>UncertainIfLimited</i> .
BypassEnable	PID	Habilita o recurso que permite ignorar o cálculo da lei de controle no bloco funcional de controle PID.
DirectActing	PID	Aplica a função do algoritmo de ação direta aumentando a saída ao aumentar PV.
FaultStateToValue	AO, DO, MDO	Se <i>In.Status=InitiateFaultState</i> , então atribua <i>FaultStateSP.Value</i> a <i>Out.Value</i> . Caso contrário, use o Hold Last Value padrão.
IFSIfBadCasIn	PID	Se <i>CascadeSP.Status=Bad</i> , então <i>Out.Status=InitiateFaultState</i> .
IFSIfBadIn	PID	Se <i>In.Status=Bad</i> , então <i>Out.Status=InitiateFaultState</i> .
IFSIfBadTrkOn	PID	Se <i>TrackValue.Status=Bad</i> e se <i>TrackEnable</i> for selecionado, então <i>Out.Status=InitiateFaultState</i> .
IFSIfBadTrkVal	PID	Inicia estado de falha se o valor de <i>TrackValue</i> for ruim.
IncreaseToClose	AO	O valor de saída é invertido antes de definir <i>Out.value</i> .
Invert	DI, DO	Inverte logicamente o valor discreto.
LowCutEnable	AI	Habilita o recurso de corte baixo.
ManuallfBadTrkVal	PID, AO	Vá para Manual se o rastreamento estiver ativo e o valor de <i>TrackValue</i> estiver ruim.
NoOutLimitsInManual	PID	Ignora <i>PIDTuning.OutputLimitHigh</i> e <i>OutputLimitLow</i> se <i>Mode.Actual=Manual</i> .
ObeySpLimitsIfCasOrRCas	PID	Restringe o setpoint a limites absolutos se <i>Mode.Actual=Cascade</i> ou <i>RemoteCascade</i> .
PropagateFaultBackward	AO, DO, MDO	Se <i>ReadBackIn.Status=Bad</i> ou <i>DeviceFailure</i> , ou se <i>Mode.Actual=FaultStateActive</i> ou <i>LocalOverride</i> , então propague esses parâmetros para <i>BackCalOut</i> , dando ao usuário a opção de iniciar um <i>BlockAlert</i> dentro deste bloco funcional ou em blocos funcionais anteriores.
PropagateFaultForward	AI, DI, MDI	Se <i>In.Status=DeviceFailure</i> ou <i>SensorFailure</i> , então propague o status para <i>Out.Status</i> , dando ao usuário a opção de iniciar um <i>BlockAlert</i> dentro deste bloco funcional ou em blocos funcionais posteriores.
SpPvTrackInLO	AO, DO, MDO	Setpoint segue a variável de processo quando <i>Mode.Actual=LocalOverride</i> .
SpPvTrackInLoorlMan	PID	Setpoint segue a variável de processo quando <i>Mode.Actual=LocalOverride</i> ou <i>InitializationManual</i> .

FBOptions BrowseName	Pertinência	Descrição
SpPvTrackInManual	AO, DO, PID, MDO	Setpoint segue a variável de processo quando <i>Mode.Target=Manual</i> .
SpTrackRetainedTarget	AO, DO, PID, MDO	O Setpoint rastreia o parâmetro In, CascadeSP ou RemoteCascadeSP com base no <i>Mode.Target</i> retido quando <i>Mode.Actual=Manual</i> , <i>LocalOverride</i> ou <i>InitializationManual</i> .
TargetToManualIfBadIn	PID	Mantém <i>Mode.Target=Manual</i> se <i>In.Status=Bad</i> .
TargetToManualIfBadTrkOn	PID	Mantém <i>Mode.Target=Manual</i> se <i>TrkOn.Status=Bad</i> e <i>TrackEnable</i> for selecionado.
TargetToManualIfBadTrkVal	PID	Mantém <i>Mode.Target=Manual</i> se <i>TrackValue.Status=Bad</i> e <i>TrackEnable</i> for selecionado.
TargetToManualIfFaultStateActive	AO, DO, MDO	Mantém <i>Mode.Target=Manual</i> se o bloco funcional anterior estiver invocando <i>InitiateFaultState</i> .
TargetToManualIfTrkActive	PID	Muda <i>Mode.Target</i> para Manual se o valor de <i>Tracking</i> estiver ativo.
TargetToNextPermittedModeIfBadCasIn	AO, DO, PID, MDO	Mantém <i>Mode.Target=next Permitted Mode</i> se <i>Mode.Target=Cascade</i> e <i>CascadeSP.Status=Bad</i> .
TrackEnable	PID	Habilita o recurso de rastreamento externo.
TrackIfBadTrkOn	PID	Se <i>TrackEnable</i> está ativo e se <i>Mode.Target≠Manual</i> e se <i>TrackValue.Status=Bad</i> , então <i>TrackValue</i> é atribuído a <i>Out.Value</i> .
TrackInManual	PID	Se <i>TrackEnable</i> for selecionado e se <i>Mode.Target=Manual</i> e se <i>TrackOn=True</i> , então <i>TrackValue</i> é atribuído a <i>Out.Value</i> .
UncertainIfLimited	AI	Se <i>In.Status=HighLimited</i> ou <i>LowLimited</i> , então <i>Out.Status=Uncertain</i> .
UncertainIfManual	AI, DI, MDI	Se <i>Mode.Actual=Manual</i> , então <i>Out.Status=Uncertain</i> .
UseFaultStateValueOnRestart	AO, DO, MDO	Aplica <i>FaultStateSP.Value</i> na reinicialização do dispositivo. Caso contrário, use o valor padrão (não volátil).
UseOutForReadBackIn	AO, DO, MDO	Calcula a PV do bloco a partir de sua própria variável Out em vez de pelo feedback do atuador.
UsePvForBackCalOut	AO, DO, PID, MDO	Substitui PV por SP de trabalho. Também aplica PV a <i>RemoteCascadeOut</i> para bloco funcional PID se <i>Mode.Actual=Cascade</i> .
UseUncertainAsGood	AI, DI, MDI, PID	Se <i>In.Status=Uncertain</i> , então substitua como Good. Caso contrário, use o padrão Uncertain como Bad.

Tabela 15 – Descrições de FBOptions

Lógica Funcional Comum

Esta seção especifica a lógica funcional comum a todos os blocos funcionais O-PAS™.

A proteção contra erros de cálculo, como divisão por zero ou raiz quadrada de um número negativo, deve ser fornecida nos mecanismos de execução do O-PAS™. Normalmente, o padrão de bits definido pelo padrão IEEE para aritmética de ponto flutuante (IEEE 754) para Not a Number (NaN) é usado para essa finalidade. NaN é o resultado de operações incalculáveis, como dividir por zero. NaN não é propagado por meio de operações aritméticas subsequentes.

No entanto, como o NaN não é representado nos tipos de dados definidos no Texto Estruturado IEC 61131-3, nenhum requisito de conformidade envolvendo o uso de NaN como proteção contra erros computacionais é especificado na Norma O-PAS™, Versão 2.1. Esta questão será revisitada em uma versão futura do Padrão O-PAS™.

Entrada Analógica (AI)

Um bloco funcional de entrada analógica (AI) recebe dados de **valor, status, registro de data e hora (VST), faixa e unidades de engenharia de um transdutor de sensor em sua entrada (In)** como um sinal O-PAS™ por meio do OCF. Condiciona os dados de entrada e disponibiliza os dados pós-condicionados através de sua saída (Out).

As unidades de conformidade normativa para um bloco funcional de entrada analógica O-PAS™ são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação entre blocos. A lógica funcional expressa no tópico seguinte reflete a implementação de referência, mas não é normativa.

Lógica Funcional

A lógica funcional do **AnalogInputBlockType O-PAS™** é representada na figura seguinte e detalhada abaixo.

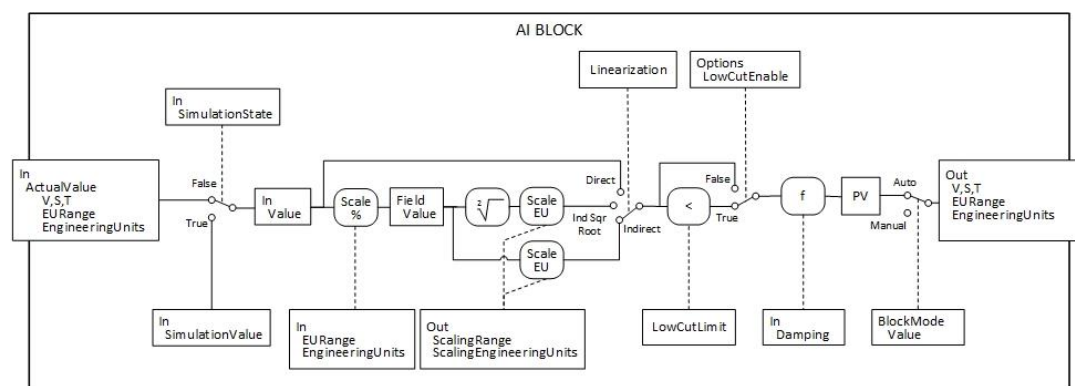


Figura 2: Esquemático do bloco funcional de entrada analógica

As variáveis **PV** e **Out** fornecem os valores primários calculados durante a execução do bloco funcional quando este está em serviço. Quando o **Modo** é **Automático**, **Out.Value** é atribuído a partir de **PV.Value**. Quando o bloco está no modo **Manual**, a variável **Out** é desconectada da **PV** calculada pelo bloco funcional e **Out.Value** pode ser gravado através do OCF, normalmente por um operador de uma IHM.

A variável **In** fornece a entrada primária para o bloco funcional. O índice VST, faixa (superior e inferior) e unidades de engenharia são adquiridos, respectivamente, em **In.ActualValue.Value**, **In.ActualValue.Status**, **In.ActualValue.Timestamp**, **In.ActualValue.EURange.High**, **In.ActualValue.EURange.Low**, e **In.ActualValue.EngineeringUnits.UnitID**.

Se as unidades de engenharia do transmissor (**In.ActualValue.EngineeringUnits.UnitID**) não forem iguais às unidades de engenharia configuradas do bloco funcional de entrada analógica (**In.Value.EngineeringUnits.UnitID**), **Error.Value** será definido como **ConfigurationError**, **Mode.Actual** está definido como **OutOfService** e **Out.Status** está definido como **Bad**, **OutOfService**, **NotLimited**.

Durante a operação normal, **In.SimulationState** é definido como falso. **In.ActualValue.Value**, **In.ActualValue.Status** e **In.ActualValue.Timestamp** são atribuídos a **In.Value**, **In.Status** e **In.Timestamp**. Além disso, **In.SimulationValue.Value** e **In.SimulationValue.Status** são atribuídos a partir de **In.ActualValue.Value** e **In.ActualValue.Status** para fins de rastreamento e inicialização sem problemas de simulação.

Quando **In.SimulationState** é definido como verdadeiro, os dados do transmissor não são processados. **In.SimulationValue.Value**, **In.SimulationValue.Status** e **In.SimulationValue.Timestamp** são atribuídos a **In.Value**, **In.Status** e **In.Timestamp**.

Os dados de **In.Value** podem ser redimensionados e transformados dependendo da configuração da opção **Linearization** (Linearização).

Quando a linearização é **Direct** (direta), nenhuma mudança de escala ou transformação é executada, ou seja:

$$result = In.Value$$

Quando a linearização é **Indirect** (indireta) ou **IndirectSquareRoot**, **In.Value** é convertido de unidades de engenharia em porcentagem (%) usando os valores **High** (alto) e **Low** (baixo) de **In.Value.EURange**.

$$FIELD.Value = 100. \times \frac{(IN.Value - IN.Value.EURange.Low)}{(IN.Value.EURange.High - IN.Value.EURange.Low)}$$

Quando a linearização é **Indirect** (indireta):

$$result = \left(\frac{FIELD.Value}{100.} \right) \times (OUT.Value.EURange.High - OUT.Value.EURange.Low) + OUT.Value.EURange.Low$$

Quando a linearização é **IndirectSquareRoot**:

$$result = Sqrt\left(\frac{FIELD.Value}{100.}\right) \times (OUT.Value.EURange.High - OUT.Value.EURange.Low) + OUT.Value.EURange.Low$$

LowCutEnable permite o condicionamento do resultado do cálculo de linearização para mitigar o ruído de medição em faixas baixas. Quando **LowCutEnable** é definido como verdadeiro e quando o resultado é menor ou igual a **LowCutLimit**, o resultado é definido como **zero**.

Para remover ruído de alta frequência no valor do transmissor, um filtro digital exponencial passa-baixa pode ser aplicado inserindo um valor diferente de zero em **In.Damping**.

$$filter\ factor = exp\left(\frac{-\ scan\ time}{In.Damping}\right)$$

$$result = (f \times previous\ value) + ((1 - f) \times current\ value)$$

onde f = fator do filtro

O resultado dos cálculos acima é atribuído ao **PV.Value**.

O bloco funcional de entrada analógica suporta os três modos: **OutOfService**, **Manual** e **Automático**.

Se o **Modo** for **Automático**, **PV.Value** e **PV.Status** serão atribuídos a **Out.Value** e **Out.Status**, respectivamente. Uma chamada ao relógio do sistema é feita com o cálculo de **Out** para atribuir o valor de **Out.Timestamp**.

Se o **Modo** for **Manual**, **PV.Value** não será atribuído a **Out.Value**. Em vez disso, um valor adquirido de uma fonte externa, normalmente uma IHM, é atribuído a **Out.Value**. Uma chamada ao relógio do sistema é feita com o cálculo de **Out** para atribuir o valor de **Out.Timestamp**.

Um bloco funcional de entrada analógica O-PAS™ deve estar em conformidade com o modelo de informação OCF especificado na Figura 3, Tabelas 16 e 17. Deverá suportar, no mínimo, os seguintes Modos:

- Out of service (Fora de serviço)
- Manual
- Automático

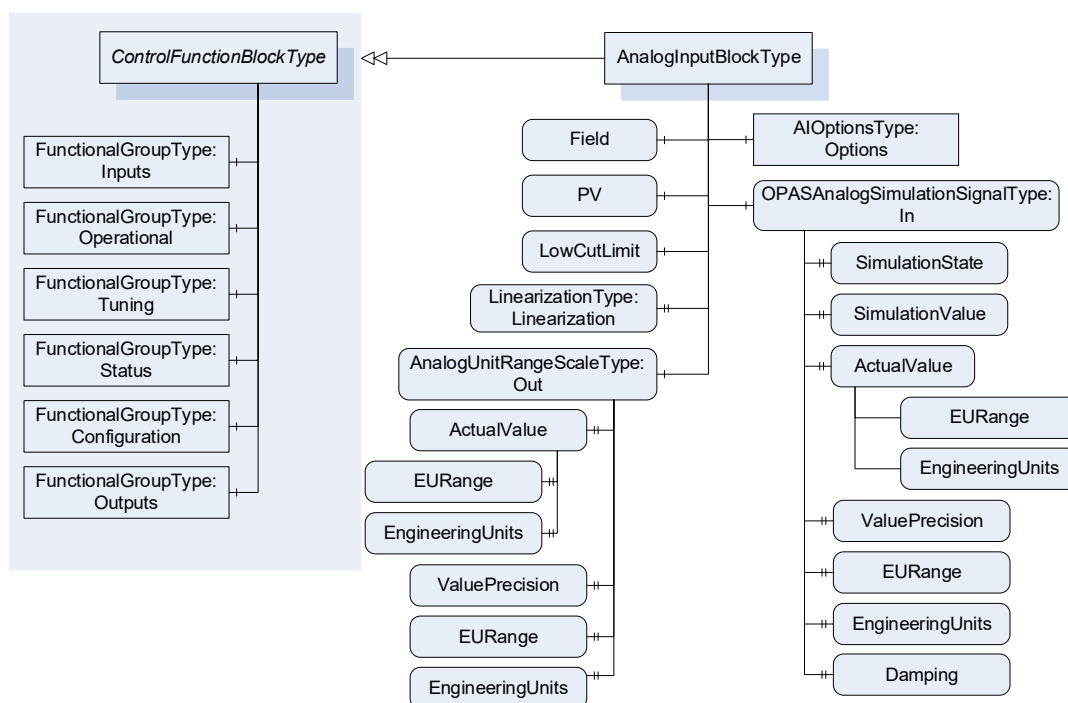


Figura 3: Modelo de dados de AnalogInputBlockType

Attribute	Value				
BrowseName	AnalogInputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da Seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Number	4:OPASAnalogSimulationSignalType	Mandatory
0:HasComponent	Variable	Out	0:Number	4:AnalogUnitRangeScaleType	Mandatory
0:HasComponent	Variable	Field	0:Number	0:BaseDataVariableType	Mandatory
0:HasProperty	Variable	Linearization	LinearizationType	0:PropertyType	Mandatory
0:HasComponent	Variable	LowCutLimit	0:Number	0:BaseDataVariableType	Mandatory
0:HasComponent	Object	Options		AIOptionType	Mandatory
0:HasComponent	Variable	PV	0:Number	0:BaseAnalogType	Mandatory

Tabela 16 - Definição de AnalogInputBlockType

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In ActualValue
Outputs	0:Organizes	True	Out
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	In EngineeringUnits
Configuration	0:Organizes	True	In EURange
Configuration	0:Organizes	True	In ValuePrecision
Configuration	0:Organizes	True	Linearization
Configuration	0:Organizes	True	LowCutLimit ValuePrecision
Configuration	0:Organizes	True	Options
Configuration	0:Organizes	True	Out EngineeringUnits
Configuration	0:Organizes	True	Out EURange
Configuration	0:Organizes	True	Out ValuePrecision
Configuration	0:Organizes	True	PV ValuePrecision
Tuning	0:Organizes	True	In Damping
Tuning	0:Organizes	True	LowCutLimit
Operational	0:Organizes	True	Mode Target
Operational	0:Organizes	True	In SimulationState
Operational	0:Organizes	True	In SimulationValue
Operational	0:Organizes	True	Out Value
Status	0:Organizes	True	Error

Source Path	Reference Type	Is Forward	Target Path
Status	0:Organizes	True	Mode
Status	0:Organizes	True	Field
Status	0:Organizes	True	In
Status	0:Organizes	True	PV

Tabela 17 - Referências adicionais de AnalogInputBlockType

Um bloco funcional de entrada analógica O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 5 e 6.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável neste modelo de informação OCF. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte "O-PAS™ Connectivity Framework (OCF)") de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte "Security").

Os modos e a interface de coordenação entre blocos de um bloco de funções de entrada analógica O-PAS™ são especificados nas Figuras 4, 5 e 6.

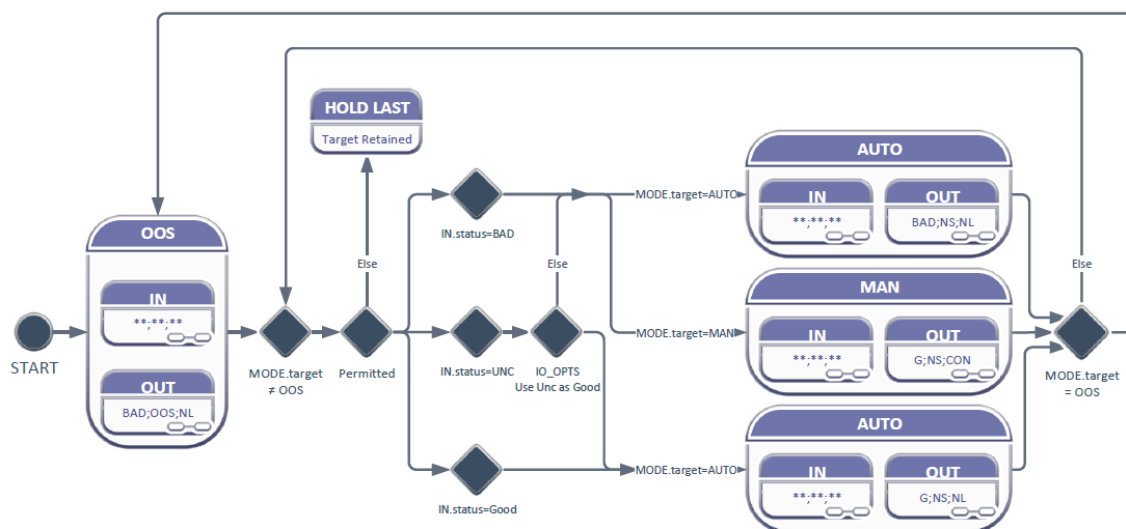


Figura 4: Diagrama de transição de estado do bloco funcional de entrada analógica O-PAS™ IEC 61131

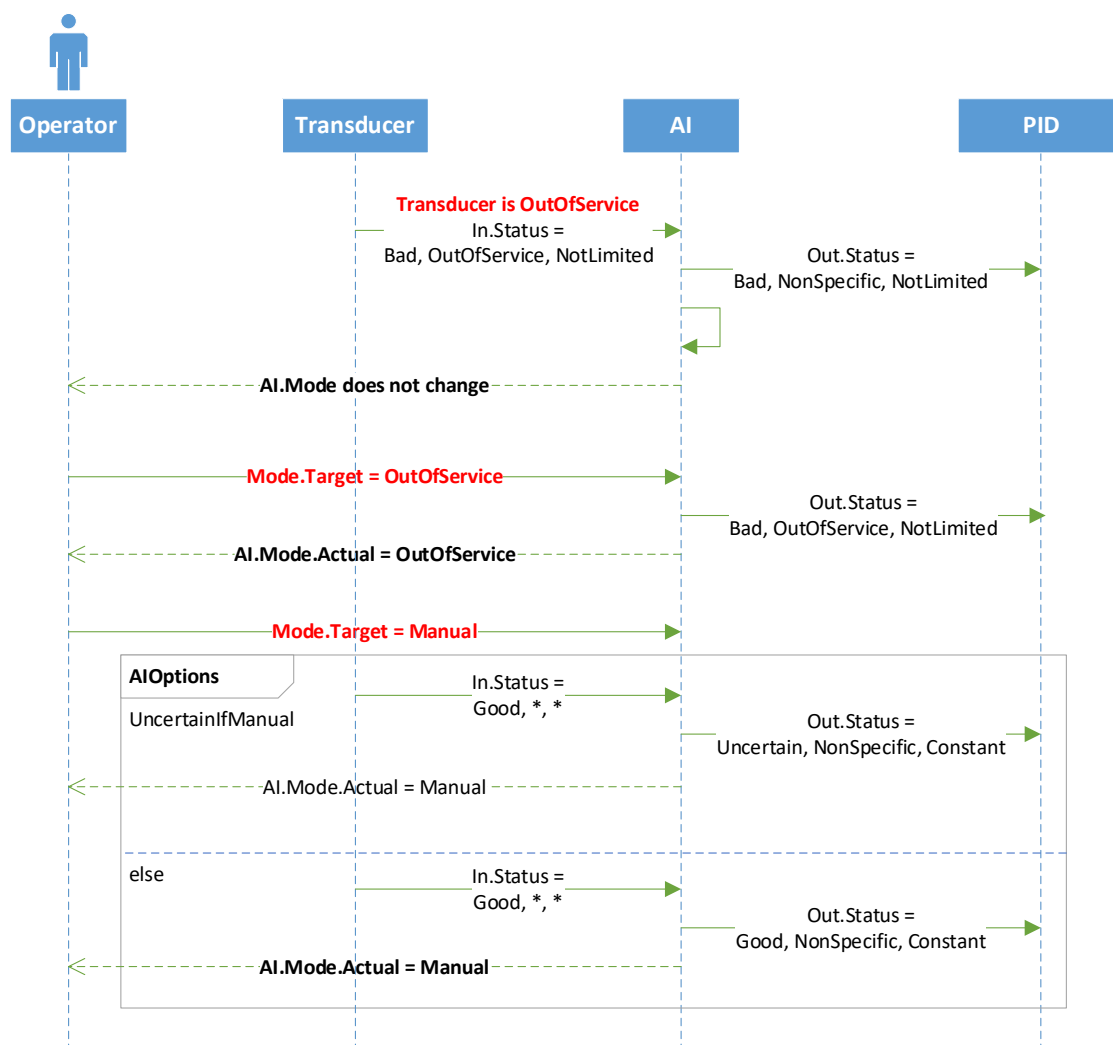


Figura 5: Diagrama de sequência do modo de bloco funcional de entrada analógica O-PAS™ e da interface de coordenação entre blocos – Parte 1 de 2 (consulte a Seção de Referências “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas de sintaxe UML)



Figura 6: Diagrama de sequência do modo de bloco funcional de entrada analógica O-PAS™ e da interface de coordenação entre blocos – Parte 1 de 2 (consulte a Seção de Referências “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas de sintaxe UML)

O **AIOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration”.com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 18.

Attribute	Value				
BrowseName	AIOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 4: <i>FBOptionsType</i> definido em O-PAS™ Parte “Information and Exchange Models: Basic Configuration”.					
0:HasProperty	Variable	4:BadIfLimited	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:LowCutEnable	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:PropagateFaultForward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UncertainIfLimited	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UncertainIfManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseUncertainAsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 18 - Definição de AIOptionsType

LinearizationType

Este **Data Type** é uma enumeração que identifica uma configuração de linearização a ser usada por um bloco funcional. Está definido na Tabela 19.

Attribute	Value				
BrowseName	LinearizationType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipp of 0: <i>Enumeration</i> definido em OPC 10000-5.					
0:HasProperty	Variable	0:EnumStrings	0:LocalizedText[]	0:Property	Mandatory

Tabela 19 - Definição de LinearizationType

Três valores de Linearização são definidos na Tabela 20. O condicionamento direto não aplica processamento. O condicionamento indireto aplica um re-escalonamento. Condicionamento SquareRoot condições adicionais: Indireto com extração de raiz quadrada.

Nome	Valor	Descrição
Direct	0	Nenhum processamento de escala é aplicado.
Indirect	1	A escala é aplicada.
SquareRoot	2	São aplicados escala e extração de raiz quadrada.

Tabela 20 - Valores de Linearization (Linearização)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informação **AnalogInputBlockType** OCF e as variáveis do programa na implementação de referência do bloco funcional de entrada analógica O-PAS™ IEC 61131 é fornecido na Tabela 21.

As colunas 1 e 2 listam as variáveis do bloco funcional que são definidas no esquema lógico (Figura 3), no modelo de dados **BlockType** (Figura 4) e na definição **BlockType** (Tabela 5). As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais do FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

[illegible]

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
Error	Status	BLOCK_ERR	WORD	na	Sequência de bits na qual vários erros de bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted, e Normal.
Field Value Status	Status Status	FIELD .Value .Status	REAL UDINT	% na	Um valor interno (contained parameter), incluindo status associado, do valor bruto normalizado do dispositivo de campo, em % da faixa de PV, conforme calculado durante a execução do bloco funcional. O cálculo é realizado antes da aplicação de L_TYPE ou DAMPING.
Linearization	Configuration	L_TYPE	USINT	na	Caracterização de sinal selecionável pelo usuário do valor de entrada do OSS para o bloco funcional de entrada analógica. Determina se deve usar o valor OSS diretamente (Direct) com base em IN.Range, ou usar o processamento de escala do bloco funcional de FIELD para linearizar (Indirect), ou aplicar extração de raiz quadrada (Ind Sqr Root) com base em OUT.Range.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
LowCutLimit	Tuning	LOW_CUT	REAL	EU of In	Substituição de valor selecionável pelo usuário para evitar o processamento de raiz quadrada de valores fornecidos pelo OSS. A saída calculada abaixo do valor de corte inferior será alterada para zero. Mais útil para eliminar ruído de medição de fluxo próximo a zero. LOW_CUT é habilitado/desabilitado em IO_OPTS.
Options	Configuration	IO_OPTS STATUS_OPTS	BOOL	na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional.
PV Value Status	Status Status	PV .Value .Status	REAL UDINT	EU of Out na	Um valor interno (contained parameter), incluindo status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.

Tabela 21 - Dicionário de dados do bloco funcional de entrada analógica O-PAS™ IEC 61131

- (1) DATE_AND_TIME
 (2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável **Value** como uma estrutura contendo três atributos; ou seja, (valor, status, carimbo de data/hora).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo Dado	Faixa Válida/ Opções	Unid.	Memória /Modo	Descrição
ERROR	Word		NA	D / RO	Sequência de bits na qual vários erros de bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
FIELD .Value .Status	Real	0 a 100 Status	%	D / RO	Um valor interno (contained parameter), incluindo status associado, do valor bruto normalizado do dispositivo de campo, em % da faixa de PV, conforme calculado durante a execução do bloco funcional. O cálculo é realizado antes da aplicação de L_TYPE ou DAMPING.
IN .ActualValue - EngineeringUnits - EURange - High - Low .Damping .EngineeringUnits .EURange - High - Low .SimulationState .SimulationValue .ValuePrecision	UDINT Real Real Real positivo UDINT Real Real Bool Real Real positivo				Valor de entrada do bloco funcional, status (gravidade, subcódigo, limite), registro de data e hora, intervalo e unidades de engenharia do OSS que está lendo de um sensor do dispositivo de campo. Os limites de faixa superior e inferior, incluindo as unidades de engenharia da variável IN do bloco funcional AI. A faixa pode ser diferente da faixa do sensor. Permite que o valor de entrada (IN) e o status sejam fornecidos manualmente quando a simulação estiver habilitada. Quando a simulação estiver desabilitada, o valor e o status da simulação seguem o valor e o status reais de IN. Suavização exponencial configurável pelo usuário (Tempo de filtro, em segundos) aplicada à PV, mas não ao campo. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
LINEARIZATION	USINT	Indirect SquareRoot Direct	NA	S / Man	Caracterização de sinal selecionável pelo usuário do valor de entrada do OSS para o bloco funcional de entrada analógica. Determina se deve usar o valor OSS diretamente (Direct) com base em IN.Range, ou usar o processamento de escala do bloco funcional de FIELD para linearizar (Indirect), ou aplicar extração de raiz quadrada (Ind Sqr Root) com base em OUT.Range.
LOWCUTLIMIT .ValuePrecision	Real positivo		IN	S	Substituição de valor selecionável pelo usuário para evitar o processamento de raiz quadrada de valores fornecidos pelo OSS. A saída calculada abaixo do valor de corte inferior será alterada para zero. Mais útil para eliminar ruído de medição de fluxo próximo a zero. LOW_CUT é habilitado/desabilitado em IO_OPTS.
MODE .Normal .Target	DINT	1 = OutofService 2 = InitializationManual 3 = LocalOverride 4 = Manual 5 = Automatic 6 = Cascade 7 = RemoteCascade	NA	S	Veja Parâmetro de Modo
OPTIONS	Bool	BadIfLimited	NA	S / O/S	Veja Opções de Blocos

Parâmetro O-PAS™	Tipo Dado	Faixa Válida/ Opções	Unid.	Memória /Modo	Descrição
		LowCutEnable PropagateFaultForward UncertainIfLimited UncertainIfManual UseUncertainAsGood			
OUT .EngineeringUnits .EURange - High - Low .ScalingEngineeringUnits .ScalingRange - High - Low .ValuePrecision	UDINT Real Real UDINT Real Real	OUT_SCALE +/- 10%	OUT	D / Man	O valor de saída (transmitido), status (severidade, subcódigo, limite), timestamp, faixa e unidades de engenharia do valor do processo conforme calculado durante a execução do bloco funcional. No Modo Manual, ActualValue é escrito a partir da IHM. Os limites superior e inferior da faixa e as unidades de engenharia a serem usadas ao exibir o valor de OUT. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
PV .ValuePrecision	Real		PV	D / RO	Um valor interno (contained parameter), incluindo status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.

Tabela 22 – Descrição dos parâmetros do bloco AI

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

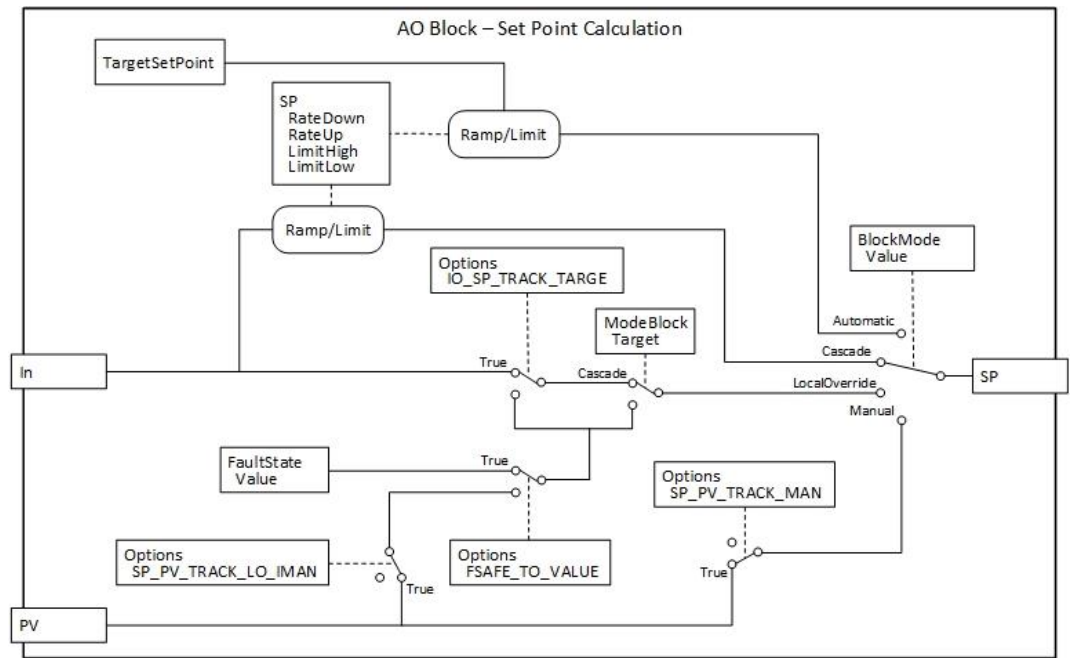


Figura 8 - Esquemático do bloco funcional de saída analógica

Existem três fluxos de dados primários no bloco funcional Saída Analógica. O resumo geral desses três fluxos de dados é o seguinte:

A entrada **In** é recebida de um bloco funcional upstream. Após o escalonamento ou condicionamento, este valor é passado para a saída **Out** para transmissão final a um atuador. Normalmente, **In** é conectada a um bloco funcional upstream, como controle PID.

A entrada **ReadBackIn** pode ser adquirida de um atuador para uso pelo bloco funcional de Saída Analógica para rastrear o valor real do atuador com o propósito de inicialização sem interrupções (bumpless) do próprio bloco AO. Se o atuador e os dispositivos de E/S suportarem o envio de dados, como a posição da válvula, a opção **UseOutForReadBackIn** poderá ser definida como falsa, o que roteia o valor da posição da válvula através dos cálculos de conversão de unidades de engenharia para calcular PV.

A saída **BackCalOut** é calculada para transmissão no caminho de fluxo de dados reverso para um bloco funcional upstream com a finalidade de inicialização sem interrupções (bumpless) do referido bloco funcional upstream.

O bloco funcional de saída analógica suporta os seis modos seguintes: **OutOfService**, **InitializationManual**, **LocalOverride**, **Manual**, **Automatic** e **Cascade**.

Cascade é o modo típico para uso online. **In.Value** é adquirido de um bloco funcional upstream. Este valor pode ser aumentado linearmente a partir de seu último valor de acordo com os valores de **SP.RateDown** e **SP.RateUp**, ou fixado para ficar dentro dos valores de **SP.LimitHigh** ou **SP.LimitLow** e, em seguida, atribuído a **SP.Value**. **SP.Value** de **In** em unidades de % pode ser reorganizado usando os valores de **Out.ScanningRange.High** e **Out.ScanningRange.Low**. Este valor em unidades de engenharia é atribuído a **Out.Value**.

Se as unidades de engenharia do bloco funcional upstream (normalmente um bloco PID) que está inserindo dados em **In (In.ActualValue.EngineeringUnits.UnitID)** não forem iguais às unidades de engenharia configuradas no bloco funcional de saída analógica (**In.Value.EngineeringUnits.UnitID**), **Error.Value** está definido como **ConfigurationError**, **Mode.Actual** está definido como **OutOfService** e **Out.Status** está definido como **Bad**, **OutOfService**, **NotLimited**.

Quando o Modo é Automático, o valor do SP alvo é gravado na interface de comunicação vertical OPC UA do bloco funcional de Saída Analógica. O valor do SP alvo é processado então, exatamente como descrito no parágrafo **Modo Cascade** acima. O valor do **SP** alvo pode ser aumentado ou fixado antes da atribuição a **SP.Value**, e reajustado e redimensionado antes da atribuição a **Out.Value**.

Quando o modo é **Manual**, o valor de **Out.Value** é gravado na interface de comunicação vertical OPC UA do bloco funcional de Saída Analógica para transmissão ao elemento final de controle. Para obter uma inicialização bumpless do modo manual, **FunctionBlockOption SpPvTrackInManual** é definido como verdadeiro. O rastreamento é feito usando o valor real do atuador que é adquirido através da entrada **ReadBackIn.ActualValue.Value**. Este valor em% é redefinido usando **Out.ScalingRange.High** e **Out.ScalingRange.Low**. Este valor é redimensionado para unidades de engenharia usando **In.EURange.High** e **In.EURange.Low**. O resultado deste cálculo é atribuído ao **PV.Value**. O cálculo de reorganização/reescalamento é então revertido. **PV.Value** é redimensionado para % usando **In.EURange.High** e **In.EURange.Low**. Este valor é redimensionado para unidades de engenharia usando **Out.ScalingRange.High** e **Out.ScalingRange.Low**. Consequentemente, o rastreamento da posição real do atuador no sinal de comando de **Out.Value** no bloco AO é alcançado

O modo **LocalOverride** não resulta de uma solicitação do operador. Em vez disso, resulta do estado do bloco funcional de supervisão gravando dados na variável **In** da saída analógica. Especificamente, o bloco funcional de entrada analógica vai para o modo **LocalOverride** quando o substatus de **In.Status** é **FaultStateSP**. As três **AOOptions** a seguir determinam o comportamento do bloco funcional quando em **LocalOverride**: **IFSIfBadIn**, **IFSIfBadCasIn** e **IFSIfBadTrkOn**. O caso típico é **IFSIfBadIn** definido como verdadeiro. Neste caso, **FaultStateSP.Value** é atribuído a **SP.Value**.

Modelo de Informação OCF

O **AnalogOutputBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção O-PAS™ “ControlFunctionBlockType Definition”. Um bloco funcional de saída analógica O-PAS™ deve estar em conformidade com o modelo de informação OCF especificado na Figura 9, Tabela 23 e Tabela 24. Deverá suportar, no mínimo, os seguintes Modos:

- OutOfService
- InitializationManual
- LocalOverride
- Manual
- Automatic
- Cascade

Um bloco funcional de saída analógica O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 10 e 11.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável neste modelo de informação OCF. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

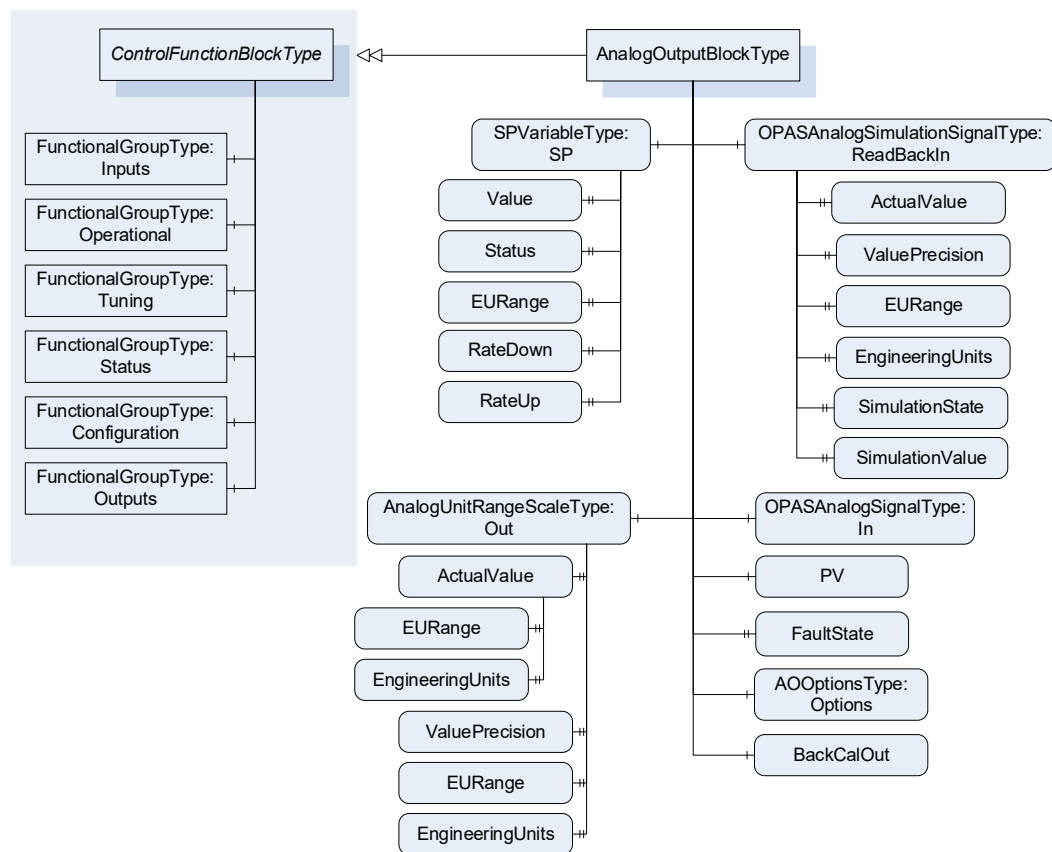


Figura 9: Modelo de dados AnalogOutputBlockType

Attribute	Value				
BrowseName	AnalogOutputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de ControlFunctionBlockType da Seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	ReadBackIn	0:Number	4:OPASAnalogSimulationSignalType	Mandatory
0:HasComponent	Variable	BackCalOut	0:Number	0:AnalogUnitRangeType	Mandatory
0:HasComponent	Variable	Out	0:Number	4:AnalogUnitRangeScaleType	Mandatory
0:HasProperty	Variable	FaultStateSP		FaultStateSPType	Mandatory
0:HasComponent	Object	Options		AOptionsType	Mandatory
0:HasComponent	Variable	PV	0:Number	0:DataItemType	Mandatory
0:HasComponent	Variable	SP	0:Number	SPVariableType	Mandatory

Tabela 23 - Definição de AnalogOutputBlockType

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In
Inputs	0:Organizes	True	ReadBackIn ActualValue
Outputs	0:Organizes	True	BackCalOut
Configuration	0:Organizes	True	BackCalOut ValuePrecision
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	Options
Configuration	0:Organizes	True	In ValuePrecision
Configuration	0:Organizes	True	Out EngineeringUnits
Configuration	0:Organizes	True	Out EURange
Configuration	0:Organizes	True	Out ValuePrecision
Configuration	0:Organizes	True	ReadBackIn ValuePrecision
Tuning	0:Organizes	True	FaultStateSP
Tuning	0:Organizes	True	ReadBackIn SimulationValue
Tuning	0:Organizes	True	SP LimitHigh
Tuning	0:Organizes	True	SP LimitLow
Tuning	0:Organizes	True	SP RateDown
Tuning	0:Organizes	True	SP RateUp
Operational	0:Organizes	True	Mode Target
Operational	0:Organizes	True	ReadBackIn SimulationState

Source Path	Reference Type	Is Forward	Target Path
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	PV
Status	0:Organizes	True	SP

Tabela 24 - Referências Adicionais AnalogOutputBlockType

O **AOOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte "Information and Exchange Models: Function Blocks" com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 25.

Attribute	Value				
BrowseName	AOOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de FBOptionsType definido em O-PAS™ Parte "Information and Exchange Models: Function Blocks".					
0:HasProperty	Variable	4:FaultStateTo Value	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IFSIfBadTrkVal	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IncreaseToClose	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:ManualIfBadTrkVal	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:PropagateFaultBackward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInLO	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpTrackRetainedTarget	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfFaultStateActive	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToNextPermittedModelIfBadCasIn	0:Boolean	0:PropertyType	Mandatory

Attribute	Value				
BrowseName	AOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
0:HasProperty	Variable	4:UseFaultState ValueOnRestart	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseOutForRe adBackIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UsePvForBac kCalOut	0:Boolean	0:PropertyType	Mandatory

Tabela 25 - Definições de AOptionsType

FaultStateSPTType é um tipo de dado, subtipo de **AnalogItem**. Está definido na tabela 26.

Attribute	Value				
BrowseName	FaultStateSPTType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 0:BaseDataVariableType definido em OPC 10000-5.					
0:HasProperty	Variable	Time	0:Duration	0:PropertyType	Mandatory

Tabela 26 - Definições de FaultStateSPTType

FaultStateSP.Value é o setpoint de trabalho fixo e configurável pelo usuário, em unidades de PV, que é substituído pelo setpoint real após um período especificado por **FaultStateSP.Time** após uma solicitação **InitiateFaultState (IFS)** ser recebida de um bloco funcional de classe de controle.

A condição **InitiateFaultState** deve permanecer continuamente ativa até que o cronômetro de contagem regressiva atinja **FaultStateSP.Time** para que **FaultStateSP.Value** seja propagado. O cronômetro de contagem regressiva reinicia se a condição **InitiateFaultState** for limpa antes que a duração de **FaultStateSP.Time** seja atingida.

A propagação de **FaultStateSP.Value** é habilitada/desabilitada por **FaultStateToValue** em **AOptions**.

SPVariableType é um tipo de dado, subtipo de **AnalogItem**. Está definido na tabela 26.

Attribute	Value				
BrowseName	SPVariableType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 0:AnalogItem definido em OPC 10000-5.					
0:HasProperty	Variable	LimitHigh	0:Number	0:PropertyType	Mandatory

Attribute	Value				
BrowseName	SPVariableType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
0:HasProperty	Variable	LimitLow	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	RateDown	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	RateUp	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	ValuePrecision	0:Number	0:PropertyType	Mandatory

Tabela 27 - Definições de SPVariableType

Value é o dado inserido manualmente para a entrada do bloco funcional. **LimitHigh** e **LimitLow** são os limites válidos para **Value**. **RateDown** e **RateUp** são taxas de rampa para alterar a saída do bloco funcional do valor antigo para o novo. **ValuePrecision** deve ser usado para controlar o número de dígitos usados para exibir **SP.Value** em uma IHM.

Os modos e a interface de coordenação de um bloco funcional de saída analógica O-PAS™ são especificados na Figura 10 e na Figura 11.

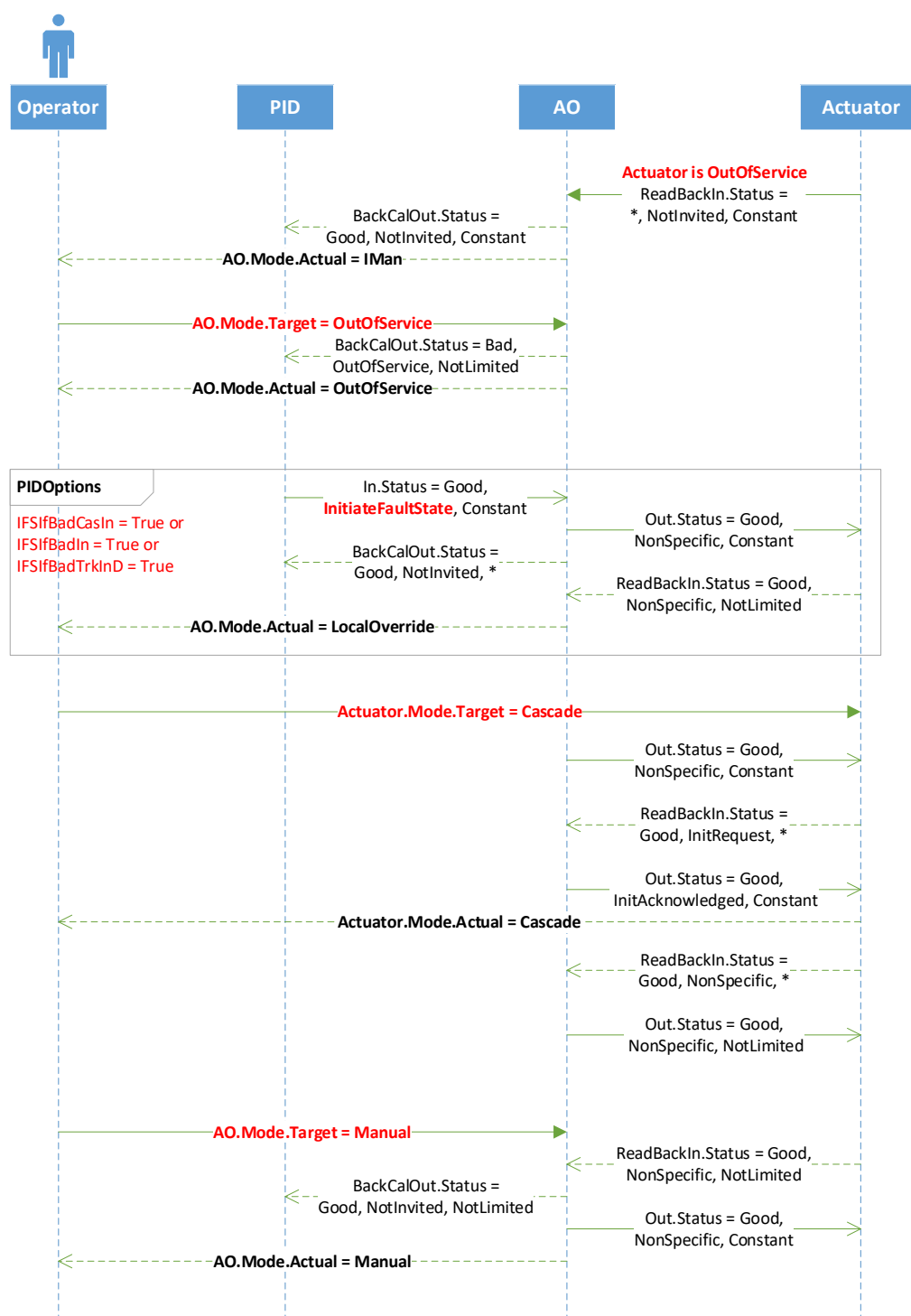


Figura 10: O Diagrama de sequência de modo de bloco funcional de saída analógica O-PAS™ e interface de coordenação entre blocos - Parte 1 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas de sintaxe UML)

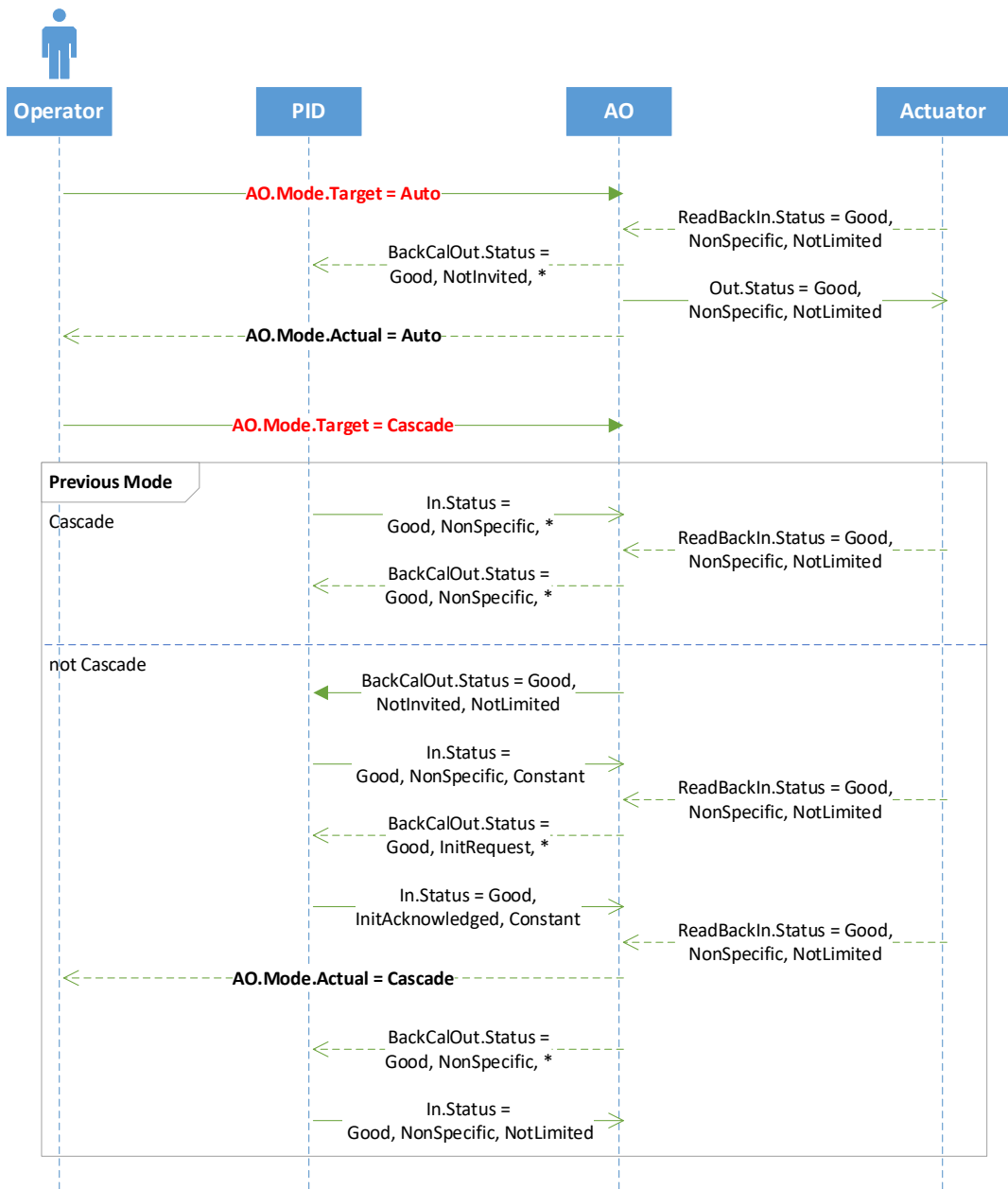


Figura 11: O Diagrama de sequência de modo de bloco funcional de saída analógica O-PAS™ e interface de coordenação entre blocos - Parte 2 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas de sintaxe UML)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informação **AnalogOutputBlockType** OCF e as variáveis do programa na implementação de referência do bloco funcional de saída analógica O-PAS™ IEC 61131 é fornecido na Tabela 28.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados **BlockType** e na definição **BlockType**. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais do FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID	Input Input Input Input Input Input	<i>Structure</i> CAS_IN .Value .Status .Timestamp .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT ¹ REAL REAL UDINT	EU of In na utcTime EU of In EU of In na	Valor de entrada do bloco funcional, status (gravidade, subcódigo, limite) e timestamp do setpoint fornecido por um bloco funcional de classe de controlador (por exemplo, PID).
Value Status Timestamp EURange High Low EngineeringUnits UnitID	Status Status Status Configuration Configuration Configuration	PV_SCALE .EU_100 .EU_0 .UNITS_INDEX	REAL REAL UDINT	EU of In na utcTime EU of In EU of In na	Os limites superior e inferior da faixa, incluindo as unidades de engenharia, serão usados ao exibir o valor de PV.
ReadBackIn ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID	Input Input Input Input Input Input	<i>Structure</i> RDBK_IN .ActualValue .ActualStatus .Timestamp .ActualEU_100 .ActualEU_0 .ActualUNITS_IND	REAL UDINT LDT ¹ REAL REAL UDINT	EU of Out na utcTime EU of Out EU of Out na	Se disponível, RDBK_IN é um valor de entrada do bloco funcional, incluindo status e timestamp, contendo a posição real de leitura da válvula de controle (ou outro atuador) em unidades de engenharia do OSS (Sink).
Value Status Timestamp EURange High Low EngineeringUnits UnitID SimulationState SimulationValue Value Status	Operational Tuning Tuning	EX SIMULATE .Enable .Value .Status	BOOL REAL UDINT	na EU of Out na	Se não estiver disponível, RDBK_IN será gerado a partir do valor OUT do bloco funcional AO.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
BackCalOut Value Status Timestamp EURange High Low EngineeringUnits UnitID	Output Output Output Output Output Output	<i>Structure</i> BKCAL_OUT .Value .Status .Timestamp .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT¹ REAL REAL UDINT	EU of In na utcTime EU of Out EU of Out na	Valor de saída do bloco funcional, incluindo status e registro de data e hora, que é fornecido a um bloco funcional de classe de controle upstream com a finalidade de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless.
Out Value Status Timestamp EURange High Low EngineeringUnits UnitID ScalingRange High Low ScalingEngineeringUnits UnitID	Output Output Output Output Output Output Configuration Configuration Configuration	<i>Structure</i> OUT .ActualValue .ActualStatus .Timestamp .ActualEU_100 .ActualEU_0 .ActualUNITS_INDEX XD_SCALE .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT¹ REAL REAL UDINT REAL REAL UDINT	EU of Out na utcTime EU of Out EU of Out na EU of Out EU of Out na	O valor de saída (transmitido), incluindo status associado e timestamp, do valor analógico primário conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa e as unidades de engenharia de OUT.
Error	Status	BLOCK_ERR	WORD	na	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte “Information and Exchange Models: Alarm and Events Configuration”)
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	OptionsType em que os modos do bloco funcional estão contidos, incluindo Target, Actual, Permission e Normal.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
FaultStateSP Value Time	Tuning Tuning	FSTATE_VAL FSTATE_TIME	REAL TIME	EU of In Sec	Setpoint de trabalho fixo configurável pelo usuário, em unidades de PV, que é substituído pelo setpoint real mediante solicitação do IFS imediatamente ou após FSTATE_TIME ter expirado. FSTATE_VAL é ativado/desativado por FaultStateToValue em IO_OPTS. Temporizador de contagem regressiva configurável pelo usuário, em segundos, desde o momento em que uma ação Initiate Fault State (IFS) é invocada (por um bloco funcional de classe de controle) até o momento em que o FaultState é ativado no bloco funcional de classe de saída. A condição FaultState deve permanecer continuamente ativa para que o temporizador faça a contagem regressiva. O cronômetro de contagem regressiva reinicia se a solicitação IFS for cancelada antes que o cronômetro chegue a zero.
Options	Configuration	IO_OPTS STATUS_OPTS	WORD WORD	na na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional.
PV Value Status	Status Status	PV .Value .Status	REAL UDINT	EU of In na	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa, incluindo as unidades de engenharia e a precisão do ponto decimal a serem usados ao exibir o valor de OUT.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
SP Value Status EURange High Low RateDown RateUp	Status Status Tuning Tuning Tuning Tuning	SP .Value .Status .SP_HI_LIM .SP_LO_LIM .SP_RATE_DN .SP_RATE_UP	REAL UDINT REAL REAL REAL REAL	EU of In na EU of In EU of In 1/Sec 1/Sec	O valor interno (parâmetro contido), incluindo o status associado, do setpoint para este bloco funcional conforme calculado durante a execução do bloco funcional. Por padrão, SP (após limitação) é usado para o valor BKCAL_OUT. Limites do valor de setpoint mais alto ou mais baixo que um operador pode inserir. Inclinação, em unidades PV por segundo, na qual é aplicado um setpoint decrescente. Nos blocos funcionais de classe de saída, as taxas de rampa se aplicam enquanto o bloco funcional está em Automático, Cas ou RCas. Se SP_RATE_DN = 0, o setpoint é aplicado diretamente.

Tabela 28 - Dicionário de dados do bloco funcional de saída analógica O-PAS™ IEC 61131

(1) DATE_AND_TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável Value como uma estrutura contendo três atributos; ou seja, (valor, status, timestamp).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo de dado	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
BACKCALOUT EngineeringUnits EURange High Low ScalingEngineeringUnits ScalingRange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		PV	D / RO	Valor de saída do bloco funcional, incluindo status e registro de data e hora, que é fornecido a um bloco funcional de classe de controle upstream com a finalidade de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless. Os parâmetros "Scaling" podem ser escritos e estão sempre nas saídas de blocos que podem fazer conversão de escala. O usuário configura o parâmetro scaling de forma que, se estiver tudo certo, esse valor passa para o Engineering units ou EURange, que é o que o bloco usa de fato. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
ERROR	WORD		NA	D/RO	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration")
FAULTSTATESP EURange High Low Time	REAL TIME		PV s	S S	Setpoint de trabalho fixo configurável pelo usuário, em unidades de PV, que é substituído pelo setpoint real mediante solicitação do IFS (Initiate Fault State) imediatamente ou após FSTATE_TIME ter expirado. FSTATE_VAL é ativado/desativado por FaultStateToValue em IO_OPTS. Temporizador de contagem regressiva configurável pelo usuário, em segundos, desde o momento em que uma ação IFS é invocada (por um bloco funcional de classe de controle) até o momento em que o FaultState é ativado no bloco funcional de classe de saída. A condição FaultState deve permanecer continuamente ativa para que o temporizador faça a contagem regressiva. O cronômetro de contagem regressiva reinicia se a solicitação IFS for cancelada antes que o cronômetro chegue a zero. O FaultStateSP é um setpoint de reserva que o bloco AO pode usar automaticamente quando ocorre uma falha no sinal de entrada ou no próprio bloco. Ele faz parte do mecanismo de Fault State — ou seja, como o bloco lida com falhas.

Parâmetro O-PAS™	Tipo de dado	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
IN ActualValue EngineeringUnits EURange High Low EngineeringUnits EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT			D	Valor de entrada do bloco funcional, status (gravidade, subcódigo, limite) e timestamp do setpoint fornecido por um bloco funcional de classe de controlador (por exemplo, PID). Os limites superior e inferior da faixa, incluindo as unidades de engenharia, serão usados ao exibir o valor de PV. Existem as unidades e faixa do ActualValue, que é um parâmetro “filho” de In e existem as unidades e faixa do próprio In. Ou seja, ActualValue representa a entrada para o bloco, já o In é o valor que o bloco processa de fato. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
MODE Normal Target	DINT DINT	1 = OutofService 2 = InitializationManual 3 = LocalOverride 4 = Manual 5 = Automatic 6 = Cascade	Na	S	OptionsType em que os modos do bloco funcional estão contidos, incluindo Target, Actual, Permission e Normal.
OPTIONS	WORD	- FaultStateToValue - IncreaseToClose - PropagateFaultBackward - SpPvTrackInLO - SpPvTrackInManual - SpTrackRetainedTarget - TargetToManualIfFaultStateActive - TargetToNextPermittedModelIfBadCasIn - UseFaultStateValueOnRestart - UseOutForReadBackIn - UsePvForBackCalOut	Na	S / O/S	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional.
OUT EngineeringUnits EURange High Low ScalingEngineeringUnits ScalingRange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		OUT	N / Man	O valor de saída (transmitido), incluindo status associado e timestamp, do valor analógico primário conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa e as unidades de engenharia de OUT. Os parâmetros “Scaling” podem ser escritos e estão sempre nas saídas de blocos que podem fazer conversão de escala. O usuário configura o parâmetro scaling de forma que, se estiver tudo certo, esse valor passa para o Engineering units ou EURange, que é o que o bloco usa de fato. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
PV ValuePrecision	REAL USINT		PV	D / RO	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa, incluindo as unidades de engenharia e a precisão do ponto decimal a serem usados ao exibir o valor de OUT. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.

Parâmetro O-PAS™	Tipo de dado	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
ReadBackIn ActualValue EngineeringUnits EURange High Low EngineeringUnits EURange High Low SimulationState SimulationValue ValuePrecision	UDINT REAL REAL UDINT REAL REAL BOOL REAL USINT		XD	D / RO	Se disponível, RDBK_IN é um valor de entrada do bloco funcional, incluindo status e timestamp, contendo a posição real de leitura da válvula de controle (ou outro atuador) em unidades de engenharia do OSS (Sink). Se não estiver disponível, RDBK_IN será gerado a partir do valor OUT do bloco funcional AO. Permite o valor e de retorno do transducer para ser manualmente fornecido quando a simulação está habilitada. Neste caso, o valor de simulação e status serão o valor PV. Existem as unidades e faixa do ActuaValue, que é um parâmetro "filho" de In e existem as unidades e faixa do próprio In. Ou seja, ActualValue representa a entrada para o bloco, já o In é o valor que o bloco processa de fato. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
SP LimitHigh LimitLow RateDown RateUp ValuePrecision	REAL REAL REAL REAL REAL USINT		PV	N / Auto	Setpoint analógico. Pode ser configurado manualmente, automaticamente através da interface do equipamento ou outro equipamento de campo. valor interno (parâmetro contido), incluindo o status associado, do setpoint para este bloco funcional conforme calculado durante a execução do bloco funcional. Por padrão, SP (após limitação) é usado para o valor BKCAL_OUT. Limites do valor de setpoint mais alto ou mais baixo que um operador pode inserir. Inclinação, em unidades PV por segundo, na qual é aplicado um setpoint decrescente. Nos blocos funcionais de classe de saída, as taxas de rampa se aplicam enquanto o bloco funcional está em Automático, Cas ou RCas. Se SP_RATE_DN = 0, o setpoint é aplicado diretamente. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.

Tabela 29 – Descrição dos parâmetros do bloco AO

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

Controle PID (PID)

Um bloco funcional de controle PID assume um valor de setpoint, um valor de processo, seus dados de status, parâmetros de ajuste de controle de feedback, etc.; calcula um valor para um bloco funcional downstream usando uma equação de lei de controle PID; e transmite o valor para um bloco funcional downstream. Em última análise, o valor de saída é usado para ajustar a posição de um atuador.

As unidades de conformidade normativa para um bloco funcional de controle O-PAS™ PID são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação interblocos. A lógica funcional expressa no próximo tópico reflete a implementação de referência, mas não é normativa.

Lógica Funcional

O bloco funcional de controle PID implementa a malha de controle de feedback e feedforward mostrada na Figura 12.

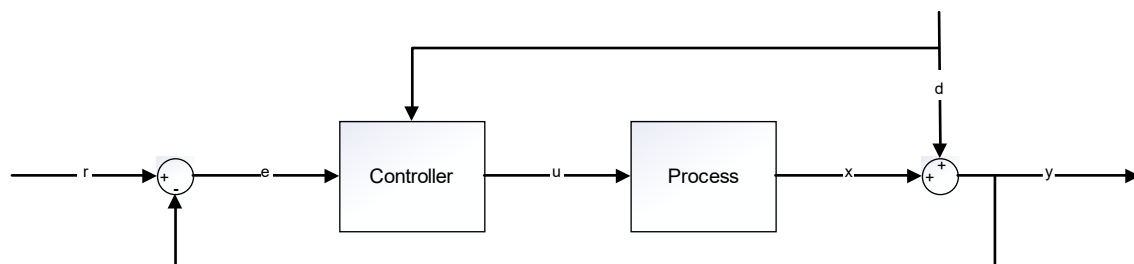


Figura 12: Malha de controle Feedback – Feedforward

A lógica funcional do bloco funcional de controle O-PAS™ PID é mostrada na Figura 13.

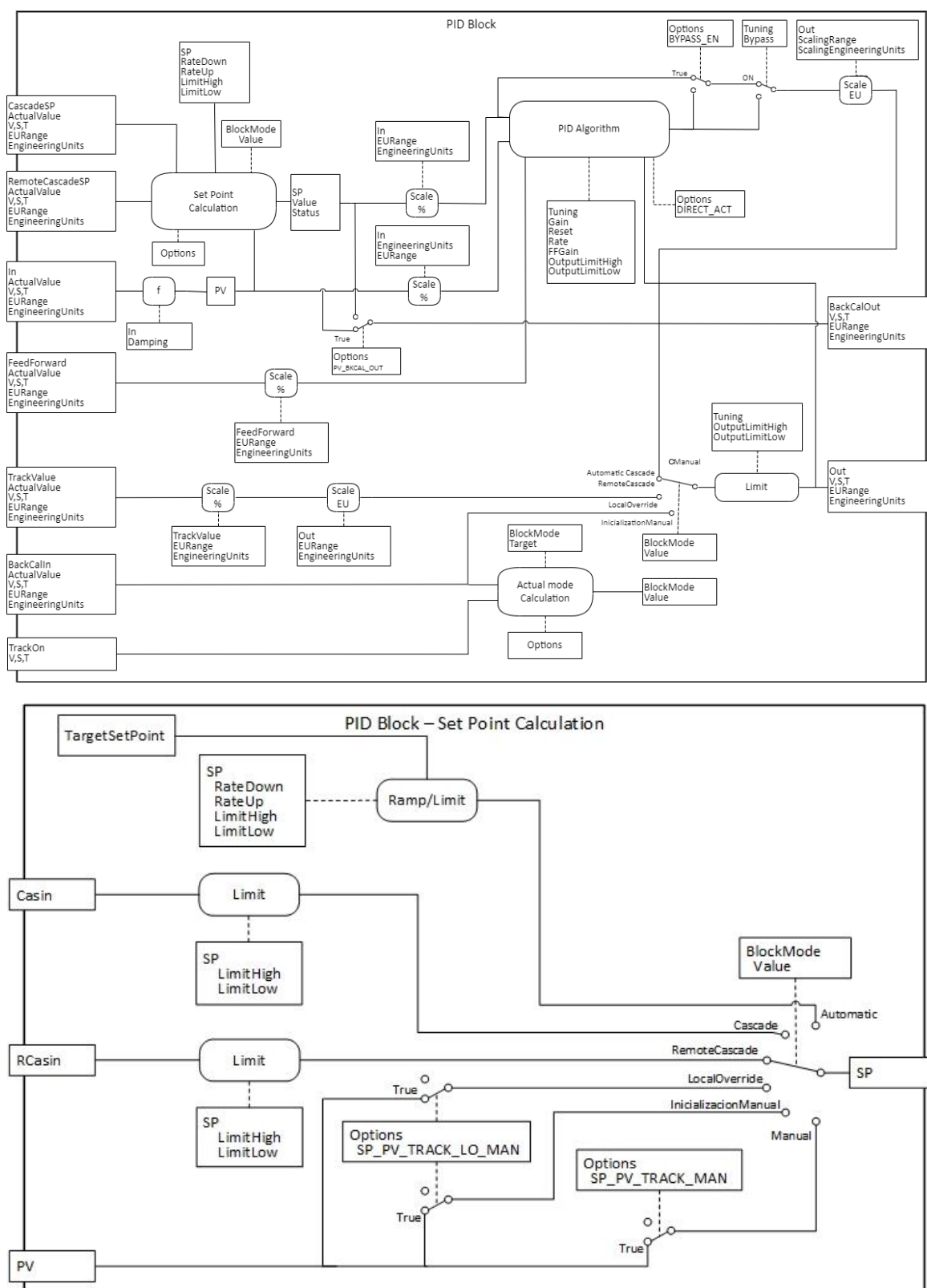


Figura 13 - Esquema do bloco funcional de controle O-PAS™ PID (consulte a Seção Syntax Notes for Function Block Logic Diagrams para notas de sintaxe)

Existem sete fluxos de dados primários no bloco funcional de controle PID. O resumo geral desses fluxos de dados é o seguinte:

1 - As entradas **CascadeSP** e **RemoteCascadeSP** são os valores alvo ou setpoint do valor do processo que o bloco funcional de controle PID está trabalhando para fornecer. **CascadeSP** e **RemoteCascadeSP** são adquiridos de blocos funcionais upstream.

2 - A entrada **In** é o valor do valor do processo que o bloco funcional de controle PID está trabalhando para levar ao valor do setpoint pela equação da lei de controle PID que calcula um sinal de comando para o atuador através da saída **Out.Value**.

Se as unidades de engenharia do bloco funcional upstream (normalmente um bloco funcional de entrada analógica) que está inserindo dados em **In (In.ActualValue.EUInformation.UnitID)** não forem iguais às unidades de engenharia configuradas do bloco funcional de controle PID (**In.EUInformation. UnitID**), **Error. Value** é definido como **ConfigurationError**, **Mode.Actual** é definido como **OutOfService** e **Out.Status** é definido como **Bad**, **OutOfService**, **Not limited**.

3 - A entrada opcional **Feedforward** é o valor de uma perturbação mensurável da variável do processo que pode ser rejeitada por um termo aditivo à equação de controle PID.

4 - A entrada **BackCallIn** pode ser adquirida de um atuador para uso pelo bloco funcional de controle PID para rastrear o valor real do atuador com a finalidade de inicialização bumpless do próprio bloco funcional de controle PID.

5 - O **TrackValue** de entrada é definido como uma constante ou adquirido de outro bloco funcional com a finalidade de fazer com que o **Out** do bloco funcional de controle PID rastreie o valor real do objeto para o qual está sendo enviado. A entrada booleana **TrackOn** quando definida como true ativa o rastreamento de saída quando a opção de configuração **TrackEnable** permite.

6 - A saída **Out** é o resultado da equação da lei de controle ou do valor definido manualmente do sinal de comando para um atuador, normalmente por meio de um bloco funcional de Saída Analógica.

7 - A saída **BackCalOut** é calculada para transmissão no caminho de fluxo de dados reverso para um bloco funcional upstream com a finalidade de inicialização bumpless do referido bloco funcional upstream.

O setpoint, **SP.Value**, é calculado pela seguinte lógica.

- Se o Modo for **Automático**, o valor bruto do setpoint é adquirido da IHM. Se Modo for **Cascade**, o valor bruto do setpoint é adquirido de **CascadeSP.Value**. Se Modo for **RemoteCascade**, o valor bruto do setpoint é adquirido do **RemoteCascadeSP.Value**.
- O valor bruto do setpoint pode ser aumentado linearmente a partir de seu último valor de acordo com os valores de **SP.RateDown** e **SP.RateUp**, ou fixado para ficar dentro dos valores de **SP.LimitHigh** ou **SP.LimitLow**. Após a rampa e fixação opcionais, o valor bruto processado é atribuído a **SP.Value**. **SP.Status** é definido para indicar se **SP.Value** está fixado em **SP.LimitHigh** ou **SP.LimitLow**.

```

IF SP.raw > SP.LimitHigh THEN
    SP.raw = SP.LimitHigh
    SP.status.limit = HighLimited
ELSE IF SP.raw < SP.LimitLow THEN
    SP.raw = SP.LimitLow
    SP.status.limit = LowLimited
ELSE
    SP.status.limit = NotLimited
END IF

IF SP.RateUp > 0. AND SP.raw > SP.previous
    AND (SP.raw - SP.previous)/scan_time > SP.RateUp THEN
    SP.raw = SP.previous + SP.RateUp x scan_time
ELSE IF SP.RateDown > 0. AND SP.raw < SP.previous
    AND (SP.previous - SP.raw)/scan_time > SP.RateDown THEN
    SP.raw = SP.previous - SP.RateDown x scan_time
END IF

SP.previous = SP.raw

```

- Para uso no cálculo da lei de controle, **SP.Value** é redimensionado de unidades de engenharia para % usando **In.EURange.High** e **In.EURange.Low**.

$$setpoint = 100. \times \frac{(SP.Value - In.EURange.Low)}{(In.EURange.High - In.EURange.Low)}$$

O valor do processo, **PV.Value**, é calculado pela seguinte lógica:

- O valor bruto, o status e o timestamp são adquiridos de **In.Value**, **In.Status** e **In.Timestamp**.
- Para remover ruído de alta frequência no sinal de valor bruto, um filtro digital exponencial passa-baixa pode ser aplicado inserindo um valor diferente de zero para **In.Damping**.

$$filter\ factor = \exp\left(\frac{-scan\ time}{In.Damping}\right)$$

$$PV.Value = (f \times previous\ raw\ value)(f \times previous\ raw\ value) + ((1 - f) \times current\ raw\ value)$$

Onde f = filter factor (fator de filtro)

Para uso no cálculo da lei de controle, **PV.Value** é redimensionado de unidades de engenharia para % usando **In.EURange.High** e **In.EURange.Low**.

$$process\ value = 100. \times \frac{(PV.Value - In.EURange.Low)}{(In.EURange.High - In.EURange.Low)}$$

- Da mesma forma, **FeedForward.Value** é redimensionado de unidades de engenharia para % usando **FeedForward.EURange.High** e **FeedForward.EURange.Low**.

feedforward value

$$= 100. \times \frac{(FeedForward.Value - FeedForward.EURange.Low)}{(FeedForward.EURange.High - FeedForward.EURange.Low)}$$

O bloco funcional de controle PID suporta os sete modos a seguir: **OutOfService**, **InitializationManual**, **LocalOverride**, **Manual**, **Automatic**, **Cascade** e **RemoteCascade**.

Os modos de operação típicos são **Automático** e **Cascade**. **RemoteCascade** é funcionalmente semelhante ao **Cascade**. Nestes três modos, **Out.Value** é calculado pela equação de controle PID. O Modo **Manual** é usado ocasionalmente para desativar o cálculo da lei de controle e definir **Out.Value** manualmente. Os modos **LocalOverride** e **InitializationManual** resultam quando há erros nos dados de entrada do bloco funcional. **OutOfService** é usado para configurar o bloco funcional.

Nos modos **Automático**, **Cascade** e **RemoteCascade**, **SP.Value**, **PV.Value** e, opcionalmente, **FeedForward.Value**, todas as unidades em %, conforme calculado nos parágrafos anteriores, são usadas na equação de controle PID. A equação de controle PID usada na implementação de referência do bloco funcional de controle PID O PAS™ é o chamado algoritmo Não Interativo (Ideal, Padrão ou ISA). Utilizando a nomenclatura da Figura 12, a equação de controle PID não interativo expressa em tempo contínuo é especificada na Equação 1.

Nota: Para mitigar ações agressivas indesejadas na alteração do setpoint, o termo derivado é baseado na alteração na medição (dy/dt), não no erro (de/dt).

$$u = K_C \times \left(e + \frac{1}{\tau_I} \int e\ dt + \tau_D \frac{dy}{dt} \right) + K_{FF} \times d$$

Where	<i>u</i> is process input	(<i>Out.Value</i>)
	<i>e</i> is error (<i>r-y</i>)	(<i>SP.Value</i> - <i>PV.Value</i>)
	<i>r</i> is reference	(<i>SP.Value</i>)
	<i>y</i> is process output	(<i>PV.Value</i>)

d is disturbance	(<i>FeedForward.Value</i>)
K_C is proportional gain	(<i>PIDTuning.Gain</i>)
τ_I is integral time	(<i>PIDTuning.Reset</i>)
τ_D is derivative time	(<i>PIDTuning.Rate</i>)
K_{FF} is feedforward gain	(<i>PIDTuning.FeedForwardGain</i>)

Equação 1: Lei de Controle PID Não Interativo com Derivada na Medição e com Feedforward

Esta forma específica da equação de controle PID foi selecionada como base para o bloco funcional de controle PID padrão O-PAS™. Todas as implementações do bloco funcional de controle PID padrão OPAS™ devem implementar esta formulação.

No entanto, não se pretende proibir outras formulações da equação de controle PID em bibliotecas de blocos funcionais em conformidade com O-PAS™. Caso um fornecedor implemente formulações adicionais da lei de controle PID como opções em seu bloco funcional de controle PID, ele deverá documentar as formulações e identificar explicitamente a formulação que é igual à Equação 1.

A proteção anti-windup é fornecida evitando aumentos no erro integrado quando o valor de saída está em um limite superior, um limite inferior ou fixado em um valor constante.

O **PIDOption DirectActing** determina o “sinal do ganho” da equação de controle. Quando **DirectActing** for verdadeiro, o erro é calculado como (**PV.Value – SP.Value**). Quando **DirectActing** é falso, o erro é (**SP.Value – PV.Value**). Alternativamente, quando **DirectActing** é falso, a ação reversa é obtida.

O **PIDOption BypassEnable** permite que a variável de configuração **PIDTuning.Bypass** seja definida como verdadeira por uma fonte externa, normalmente da IHM. Normalmente usado em blocos funcionais de controle PID secundário em cascata quando seu **PV.Status** é **Bad**, quando **PIDTuning.Bypass** é verdadeiro **SP.Value** em % é atribuído a **Out.Value**.

O rastreamento da saída é feito usando a entrada **TrackValue**. O valor bruto de **TrackValue** é convertido de suas unidades de engenharia de entrada para as unidades de engenharia de **Out.Value** pelo cálculo a seguir.

$$result = 100. \times \frac{(TrackValue.Value - TrackValue.Value.EURange.Low)}{(TrackValue.Value.EURange.High - TrackValue.Value.EURange.Low)}$$

$$Out.Value = \left(\frac{result}{100.} \right) \times (OUT.Value.EURange.High - OUT.Value.EURange.Low) + OUT.Value.EURange.Low$$

Nos modos **Automático**, **Cascade** e **RemoteCascade**, quando a entrada **TrackOn** é verdadeira e quando a variável de configuração **PIDOptions TrackEnable** é definida como verdadeira, o resultado da equação de controle PID é substituído pelo valor de **Out.Value** calculado pela equação acima.

No Modo **Manual**, se o rastreamento da saída for solicitado e se o **PIDOption TrackInManual** estiver definido como verdadeiro, o Modo será rebaixado para **LocalOverride** e o rastreamento da saída ocorrerá.

Modelo de Informação OCF

O **PIDControlBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção O-PAS™ “ControlFunctionBlockType Definition”. Um bloco funcional de controle PID O-PAS™ deve estar em conformidade com o modelo de informação OCF especificado na Figura 14, Tabela 28 e Tabela 29. Deverá suportar, no mínimo, os seguintes Modos.

- OutOfService
- InitializationManual
- LocalOverride
- Manual
- Automatic
- Cascade
- RemoteCascade

Um bloco funcional de controle PID O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 15, 16 e 17.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável neste modelo de informação OCF. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

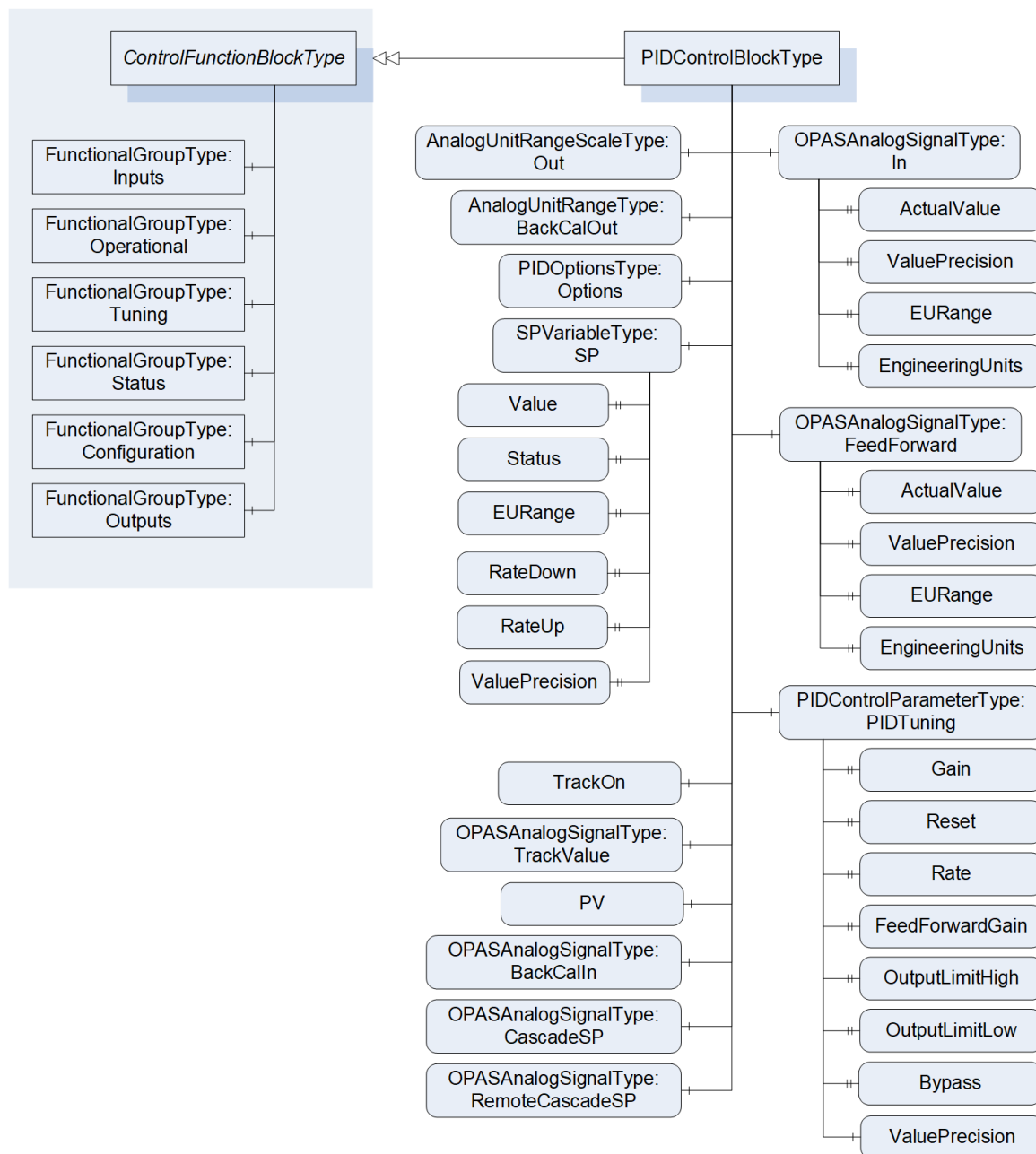


Figura 14: Modelo de dados PIDControlBlockType

Attribute	Value				
BrowseName	PIDControlBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da Seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	BackCallIn	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	CascadeSP	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	FeedForward	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	In	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	RemoteCascadeSP	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	TrackOn	0:Boolean	0:BaseDataVariableType	Mandatory
0:HasComponent	Variable	TrackValue	0:Number	4:OPASAnalogSignalType	Mandatory
0:HasComponent	Variable	BackCalOut	0:Number	0:AnalogUnitRangeType	Mandatory
0:HasComponent	Variable	Out	0:Number	4:AnalogUnitRangeScaleType	Mandatory
0:HasComponent	Object	Options		PIDOptionsType	Mandatory
0:HasComponent	Variable	PV	0:Number	0:DataItemType	Mandatory
0:HasComponent	Variable	SP	0:Number	SPVariableType	Mandatory
0:HasComponent	Variable	PIDTuning	0:Number	PIDControlParameterType	Mandatory

Tabela 30 - Definição de *PIDControlBlockType*

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	BackCallIn
Inputs	0:Organizes	True	CascadeSP
Inputs	0:Organizes	True	FeedForward
Inputs	0:Organizes	True	In
Inputs	0:Organizes	True	RemoteCascadeSP

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	TrackOn
Inputs	0:Organizes	True	TrackValue
Outputs	0:Organizes	True	BackCalOut
Outputs	0:Organizes	True	Out
Configuration	0:Organizes	True	BackCalIn ValuePrecision
Configuration	0:Organizes	True	BackCalOut ValuePrecision
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	CascadeSP ValuePrecision
Configuration	0:Organizes	True	FeedForward EngineeringUnits
Configuration	0:Organizes	True	FeedForward EURange
Configuration	0:Organizes	True	FeedForward ValuePrecision
Configuration	0:Organizes	True	In EngineeringUnits
Configuration	0:Organizes	True	In EURange
Configuration	0:Organizes	True	In ValuePrecision
Configuration	0:Organizes	True	Options
Configuration	0:Organizes	True	Out EngineeringUnits
Configuration	0:Organizes	True	Out EURange
Configuration	0:Organizes	True	Out ValuePrecision
Configuration	0:Organizes	True	PV ValuePrecision
Configuration	0:Organizes	True	RemoteCascadeSP ValuePrecision

Source Path	Reference Type	Is Forward	Target Path
Configuration	0:Organizes	True	SP ValuePrecision
Configuration	0:Organizes	True	TrackEnable
Configuration	0:Organizes	True	TrackValue EngineeringUnits
Configuration	0:Organizes	True	TrackValue EURange
Configuration	0:Organizes	True	TrackValue ValuePrecision
Configuration	0:Organizes	True	PIDTuning ValuePrecision
Tuning	0:Organizes	True	PIDTuning
Tuning	0:Organizes	True	SP LimitHigh
Tuning	0:Organizes	True	SP LimitLow
Tuning	0:Organizes	True	SP RateDown
Tuning	0:Organizes	True	SP RateUp
Operational	0:Organizes	True	Mode Target
Operational	0:Organizes	True	PIDTuning Bypass
Operational	0:Organizes	True	SP
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	PV

Tabela 31 - Referências Adicionais PIDControlBlockType

O **PIDOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte "ControlFunctionBlockType Definition".com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 30.

Attribute	Value				
BrowseName	PIDOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de <i>FBOptionsType</i> definido em O-PAS™ Part “ControlFunctionBlockType Definition”.					
0:HasProperty	Variable	4:BypassEnable	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:DirectActing	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IFSIfBadCasIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IFSIfBadIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IFSIfBadTrkOn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:IFSIfBadTrkVal	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:ManualIfBadTrkVal	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:NoOutLimitsInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:ObeySpLimitsIfCasOrRCas	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInLOrIMan	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpTrackRetainedTarget	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfBadIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfBadTrkOn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfBadTrkVal	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfTrkActive	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToNextPermittedModelIfBadCasIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TrackEnable	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TrackIfBadTrkOn	0:Boolean	0:PropertyType	Mandatory

Attribute	Value				
BrowseName	PIDOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
0:HasProperty	Variable	4:TrackInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UsePvForBackCalOut	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseUncertainAsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 32 - Definição de *PIDOptionsType*

Este **ObjectType** é um subtipo de **FolderType**. Está definido na Tabela 31.

Attribute	Value				
BrowseName	PIDControlParameterType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Sub-type of 0:FolderType defined in OPC 10000-5 §6.6.					
0:HasProperty	Variable	Gain	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	Reset	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	Rate	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	FeedForwardGain	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	OutputLimitHigh	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	OutputLimitLow	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	Bypass	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	ValuePrecision	0:Number	0:PropertyType	Mandatory

Tabela 33 - Definição de *PIDControlParameterType*

Gain, **Reset** e **Rate** são os parâmetros de ajuste para ação de controle proporcional, integral e derivativo, respectivamente. O ganho é adimensional (%/%). **Reset** e **Rate** têm unidades de engenharia de segundos. **FeedForwardGain** é um coeficiente multiplicativo adimensional de uma mudança em **FeedForward.Value** que é adicionado a **Out.Value**. **OutputLimitHigh** e **OutputLimitLow** são limites superior e inferior em **Out.Value**. Quando **Bypass** for verdadeiro, o cálculo do controle PID não é feito e o **SP.Value** é atribuído a **Out.Value**.

SPVariableType

Este **DataType** é um subtipo de **AnalogItemType**. Está definido na Tabela 32.

Attribute	Value				
BrowseName	SPVariableType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Sub-type of 0:AnalogItemType defined in OPC 10000-5.					
0:HasProperty	Variable	LimitHigh	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	LimitLow	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	RateDown	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	RateUp	0:Number	0:PropertyType	Mandatory
0:HasProperty	Variable	ValuePrecision	0:Number	0:PropertyType	Mandatory

Tabela 34 - Definição de SPVariableType

Value são os dados inseridos manualmente para o valor desejado da variável controlada (PV). **LimitHigh** e **LimitLow** são os limites de validade do **setpoint (SP)**. **RateDown** e **RateUp** são taxas de rampa para alterar o valor do setpoint na equação de controle PID. **ValuePrecision** deve ser usado para controlar o número de dígitos usados para exibir **SP.Value** em uma IHM.

Os modos e a interface de coordenação entre blocos de um bloco funcional de controle PID O-PAS™ são especificados na Figura 15, Figura 16 e Figura 17.

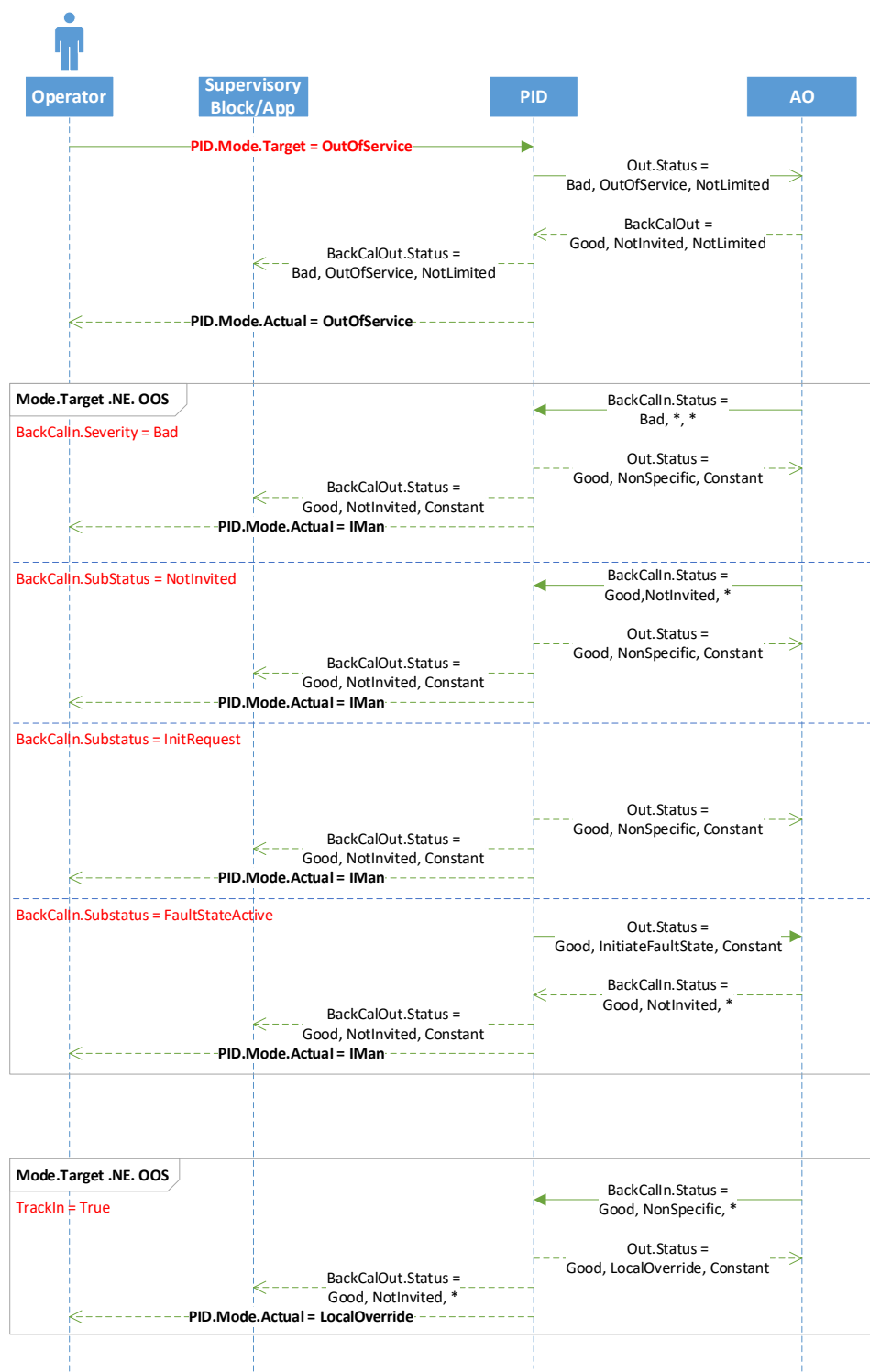


Figura 15: Diagrama de sequência de modo de bloco funcional de controle PID O-PAS™ e interface de coordenação entre blocos - Parte 1 de 3 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams”)

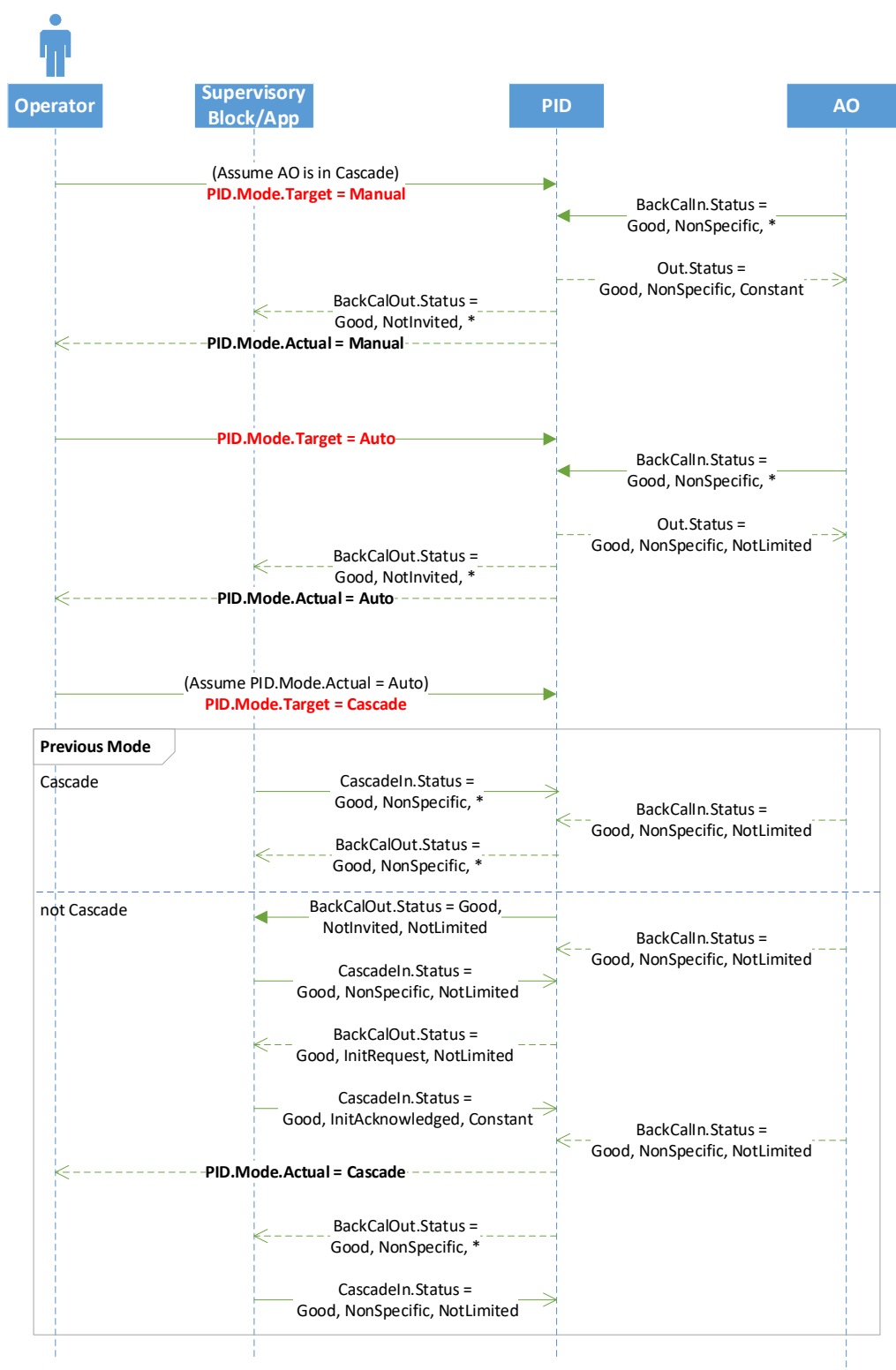


Figura 16: Diagrama de sequência de modo de bloco funcional de controle PID O-PAS™ e interface de coordenação entre blocos - Parte 2 de 3 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams”)

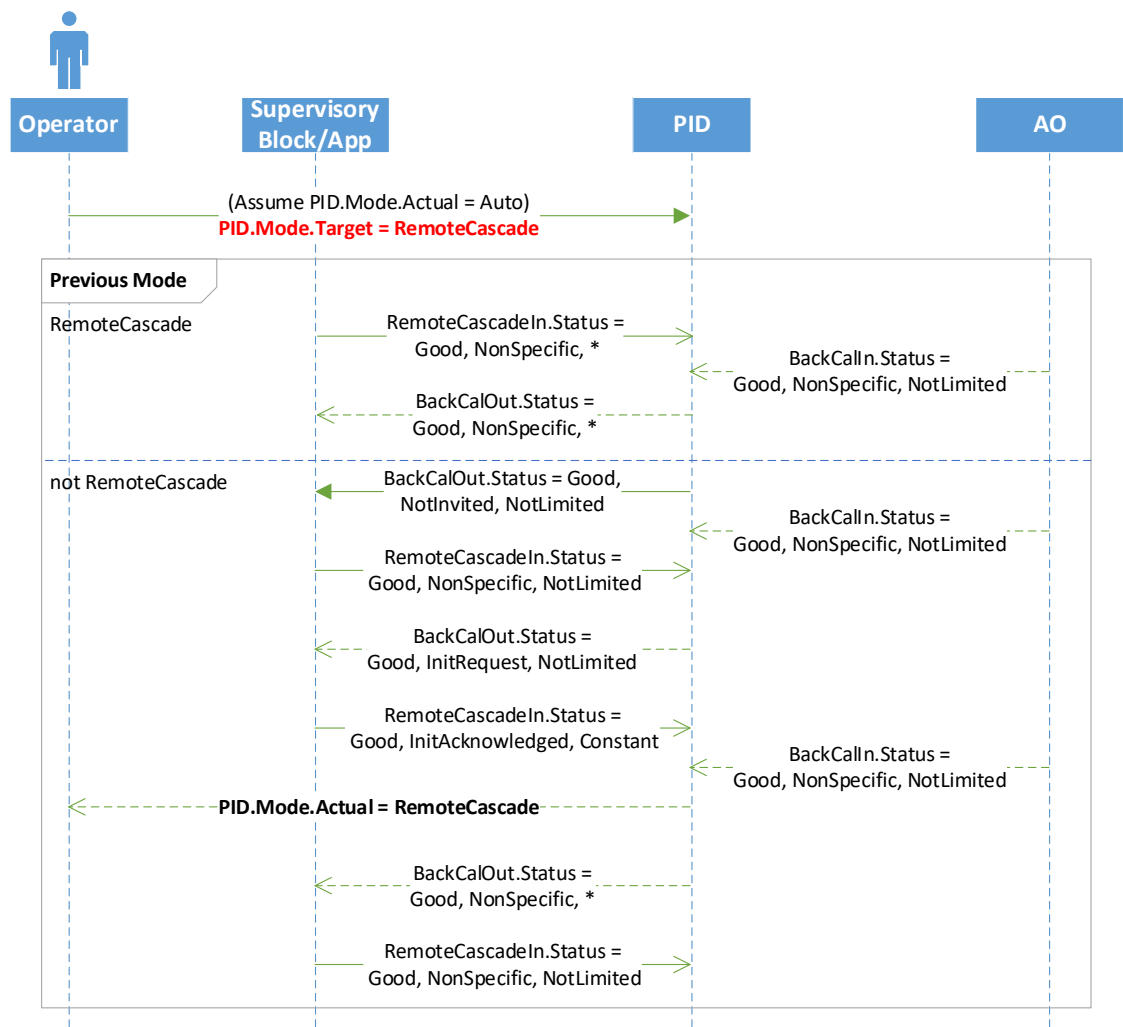


Figura 17: Diagrama de sequência de modo de bloco funcional de controle PID O-PAS™ e interface de coordenação entre blocos - Parte 3 de 3 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams”)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informação **PIDControlBlockType** OCF e as variáveis do programa na implementação de referência do bloco funcional de controle PID O-PAS™ IEC 61131 é fornecido na Tabela 33.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados **BlockType** e na definição **BlockType**. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais do FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
BackCallIn ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID Value EngineeringUnits UnitID	Input Input Input Input Input Input Configuration	Structure BKCAL_IN .Value .Status .Timestamp	REAL UDINT LDT ¹	EU of OUT na utcTime	Valor de entrada do bloco funcional, incluindo status e registro de data e hora, de BKCAL_OUT de um bloco funcional downstream para fins de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless.
CascadeSP ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID Value EngineeringUnits UnitID	Input Input Input Input Input Input Configuration	Structure CAS_IN .Value .Status .Timestamp	REAL UDINT LDT ¹	EU of PV na utcTime	Valor de entrada do bloco funcional PID secundário, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador (por exemplo, PID primário).
FeedForward ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID Value EURange High Low EngineeringUnits UnitID	Input Input Input Input Input Input Configuration Configuration Configuration	Structure FF_VAL .Value .Status .Timestamp FF_SCALE .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT ¹ REAL REAL UDINT	EU of FF na utcTime EU of FF EU of FF na	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, da variável feedforward. Os limites superior e inferior da faixa e as unidades de engenharia associadas a FF_VAL.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In ActualValue Value Status Timestamp EURange High Low EngineeringUnits NamespaceID UnitID Value EURange High Low EngineeringUnits UnitID Damping	Input Input Input Input Input Input Configuration Configuration Configuration Tuning	<i>Structure</i> IN .Value .Status .Timestamp PV_SCALE .EU_100 .EU_0 .UNITS_INDEX PV_FTIME	REAL UDINT LDT ¹ REAL REAL UDINT REAL	EU of PV na utcTime EU of PV EU of PV na Sec	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do valor de entrada primário fornecido por um bloco funcional de classe de entrada (por exemplo, AI). Os limites superior e inferior da faixa e as unidades de engenharia para PV e SP. O amortecimento é a constante de tempo do filtro exponencial configurável pelo usuário, em segundos, aplicada ao valor de In.
RemoteCascadeSP ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID Value EngineeringUnits UnitID	Input Input Input Input Input Input Configuration	<i>Structure</i> RCAS_IN .Value .Status .Timestamp	REAL UDINT LDT ¹	EU of PV na utcTime	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por uma aplicação online ou estação de supervisão (Host).
TrackOn Value	Input	TRK_IN_D .Value	BOOL	On/Off	Entrada discreta do bloco funcional recebida via link de uma fonte externa; por exemplo, IHM. Se TrackOn se tornar verdadeiro, o TrackValue fornecido externamente se tornará o valor OUT do bloco funcional PID.
TrackValue ActualValue Value Status Timestamp EURange High Low EngineeringUnits UnitID Value EURange High Low EngineeringUnits UnitID	Input Input Input Input Input Input Configuration Configuration Configuration	<i>Structure</i> TRK_VAL .Value .Status .Timestamp TRK_SCALE .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT ¹ REAL REAL UDINT	EU of TRK na utcTime	O valor substituto usado para OUT conforme fornecido por um bloco funcional vinculado externamente se TRK_IN_D for verdadeiro. Os limites superior e inferior da faixa, incluindo as unidades de engenharia e a precisão do ponto decimal, associados a TRK_VAL.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
BackCalOut Value Status Timestamp EURange High Low EngineeringUnits UnitID	Output Output Output Output Output Output	Structure BKCAL_OUT .Value .Status .Timestamp	REAL UDINT LDT ¹	EU of PV	Usado em relacionamentos de cascata secundária PID, BKCAL_OUT é o valor de saída do bloco funcional, incluindo status e registro de data/hora que é fornecido a um bloco funcional de classe de controle upstream para fins de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless.
Out Value Status Timestamp EURange High Low EngineeringUnits UnitID ScalingRange High Low ScalingEngineeringUnits UnitID	Output Output Output Output Output Output Configuration Configuration Configuration	Structure OUT .Value .Status .Timestamp OUT_SCALE .EU_100 .EU_0 .UNITS_INDEX	REAL UDINT LDT ¹ REAL REAL UDINT	EU of OUT	O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa e as unidades de engenharia de OUT e parâmetros com a mesma escala de OUT.
Error	Status	BLOCK_ERR	WORD	N/A	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (consulte O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
Options	Configuration	CONTROL_OPTS SHED_OPT STATUS_OPTS	WORD WORD WORD	na	Opções selecionáveis pelo usuário para alterar cálculos feitos no bloco funcional de controle. Opções selecionáveis pelo usuário para alterar a ação tomada no tempo limite (timeout) do dispositivo de controle remoto. Opções selecionáveis pelo usuário para alterar a ação tomada no status do bloco funcional.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
PV Value Status	Status Status	PV .Value .Status	REAL UDINT	EU of PV na	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.
SP Value Status EURange High Low RateDown RateUp	Operational Status Tuning Tuning Tuning Tuning	SP .Value .Status SP_HI_LIM SP_LO_LIM SP_RATE_DN SP_RATE_UP	REAL UDINT REAL REAL REAL REAL	EU of PV n.a. EU of PV EU of PV 1/sec 1/sec	O valor, incluindo o status associado, do valor desejado da variável de processo controlada conforme escrito pelo operador. Limites do valor do setpoint mais alto e mais baixo que um operador pode inserir. Inclinação, em unidades de PV por segundo, na qual uma alteração do setpoint é aplicada. Nos blocos funcionais da classe de controle, as taxas de rampa se aplicam enquanto o bloco funcional está no modo automático. Se SP_RATE_x = 0, o setpoint é aplicado diretamente.
PIDTuning Gain Reset Rate FeedForwardGain OutputLimitHigh OutputLimitLow Bypass	Tuning Tuning Tuning Tuning Tuning Tuning Operational	GAIN RESET RATE FF_GAIN OUT_HI_LIM OUT_LO_LIM BYPASS	REAL REAL REAL REAL REAL REAL REAL BOOL	%/% Sec Sec %//% EU of Out EU of Out na	Coefficiente proporcional adimensional usado pelo algoritmo do bloco funcional PID no cálculo da saída do bloco funcional. A constante de tempo integral, em segundos por repetição. A constante de tempo derivada, em segundos. Coefficiente proporcional pelo qual a variável feedforward é multiplicada antes de adicionar o produto à saída do controlador. Valores máximo e mínimo permitidos de OUT. Utilizado apenas em controladores secundários em cascata, ao receber uma variável de processo BAD em IN, o cálculo normal do controle pode ser contornado através deste parâmetro. Quando definido, o valor de SP, em % da faixa, é passado diretamente para a saída target e o valor de OUT é usado para BKCAL_OUT. Para evitar um impacto na transferência de/para o bypass, o setpoint será inicializado automaticamente para o valor de saída ou valor do processo, respectivamente, e o sinalizador de caminho quebrado (NotInvited) será definido para uma execução.

Tabela 35 - Dicionário de dados do bloco funcional de controle PID O-PAS™ IEC 61131

(1) DATE AND TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável Value como uma estrutura contendo três atributos; ou seja, (valor, status e timestamp).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo Dado	Faixa Válida/ Opções	Unidades	Memória / Modo	Descrição
BACKCALIN Actual Value Engineering Units EURange High Low Engineering Units EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		OUT	N	Valor de entrada do bloco funcional, incluindo status e registro de data e hora, de BKCAL_OUT de um bloco funcional downstream para fins de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
BACKCALOUT EngineeringUnits EURange High Low ScalingEngineeringUnits ScalingRange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		PV	D / RO	Usado em relacionamentos de cascata secundária PID, BKCAL_OUT é o valor de saída do bloco funcional, incluindo status e registro de data/hora que é fornecido a um bloco funcional de classe de controle upstream para fins de anti-reset windup e para facilitar a inicialização do circuito de controle e a transferência bumpless. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
CASCADESP Actual Value Engineering Units EURange High Low Engineering Units EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT			D	Valor de entrada do bloco funcional PID secundário, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador (por exemplo, PID primário). ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
ERROR	WORD		N/A	D / RO	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (consulte O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
FEEDFORWARD Actual Value Engineering Units EURange High Low Engineering Units EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		FF	D	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, da variável feedforward. Os limites superior e inferior da faixa e as unidades de engenharia associadas ao valor de feedforward. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
IN Actual Value Engineering Units EURange High Low Engineering Units EURange High Low	UDINT REAL REAL UDINT REAL REAL		PV	D	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do valor de entrada primário fornecido por um bloco funcional de classe de entrada (por exemplo, AI). Os limites superior e inferior da faixa e as unidades de engenharia para PV e SP. O amortecimento é a constante de tempo do filtro exponencial configurável pelo

Parâmetro O-PAS™	Tipo Dado	Faixa Válida/ Opções	Unidades	Memória / Modo	Descrição
ValuePrecision	USINT				usuário, em segundos, aplicada ao valor de In. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
Normal Target MODE	DINT DINT	1 = OutofService 2 = InitializationManual 3 = LocalOverride 4 = Manual 5 = Automatic 6 = Cascade 7 = RemoteCascade	Na	S	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
OPTIONS	WORD	BypassEnable DirectActing IFSIIfBadCasIn IFSIIfBadIn IFSIIfBadTrkInD NoOutLimitsInManual ObeySpLimitsIfCasOrRCas SpPvTrackInLOorlMan SpPvTrackInManual SpPvTrackRetainedTarget SpPvTrackRetainedTarget TargetToManualIfBadIn TargetToManualIfBadTrkInD TargetToManualIfTrkActive TargetToNextPermittedModelIfBadCasIn TrackEnable TrackIfBadTrkInD TrackInManual UsePvForBackCalOut UseUncertainAsGood	Na		Opções selecionáveis pelo usuário para alterar cálculos feitos no bloco funcional de controle. Opções selecionáveis pelo usuário para alterar a ação tomada no tempo limite (timeout) do dispositivo de controle remoto. Opções selecionáveis pelo usuário para alterar a ação tomada no status do bloco funcional.
OUT EngineeringUnits EURange High Low ScalingEngineeringUnits ScalingRange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		OUT	N / Man	O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional. Os limites superior e inferior da faixa e as unidades de engenharia de OUT e parâmetros com a mesma escala de OUT. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
PV ValuePrecision	REAL USINT		PV	D / RO	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
REMOTECASCADESP Actual Value Engineering Units EURange High Low Engineering Units EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		PV	D	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por uma aplicação online ou estação de supervisão (Host). ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
SP LimitHigh LimitLow RateDown	REAL REAL REAL		PV 1/s	N / Auto	O valor, incluindo o status associado, do valor desejado da variável de processo controlada conforme escrito pelo operador.

Parâmetro O-PAS™	Tipo Dado	Faixa Válida/ Opções	Unidades	Memória / Modo	Descrição
RateUp ValuePrecision	REAL USINT				Limites do valor do setpoint mais alto e mais baixo que um operador pode inserir. Inclinação, em unidades de PV por segundo, na qual uma alteração do setpoint é aplicada. Nos blocos funcionais da classe de controle, as taxas de rampa se aplicam enquanto o bloco funcional está no modo automático. Se SP_RATE_x = 0, o setpoint é aplicado diretamente. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
TRACKON	BOOL		On/Off	D	Entrada discreta do bloco funcional recebida via link de uma fonte externa; por exemplo, IHM. Se TrackOn se tornar verdadeiro, o TrackValue fornecido externamente se tornará o valor OUT do bloco funcional PID.
TRACKVALUE Actual Value Engineering Units EURange High Low Engineering Units EURange High Low ValuePrecision	UDINT REAL REAL UDINT REAL REAL USINT		TRK	D	O valor substituto usado para OUT conforme fornecido por um bloco funcional vinculado externamente se TRK_IN_D for verdadeiro. Os limites superior e inferior da faixa, incluindo as unidades de engenharia e a precisão do ponto decimal, associados a TRK_VAL. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.
TUNING Bypass FeedForwardGain Gain OutputLimitHigh OutputLimitLow Rate Reset ValuePrecision	BOOL REAL REAL REAL REAL REAL REAL USINT			S / Man	Coefficiente proporcional pelo qual a variável feedforward é multiplicada antes de adicionar o produto à saída do controlador. Coeficiente proporcional adimensional usado pelo algoritmo do bloco funcional PID no cálculo da saída do bloco funcional. Valores máximo e mínimo permitidos de OUT. A constante de tempo derivada, em segundos. A constante de tempo integral, em segundos por repetição. Usado apenas em controladores secundários em cascata. Quando a variável de processo (IN) estiver BAD, o cálculo PID é ignorado: o valor de SP (% da faixa) é enviado direto para a saída OUT, que por sua vez vai para BKCAL_OUT. Para evitar saltos, o SP é inicializado com o valor de saída ou do processo, e o sinalizador NotInvited é marcado por uma execução. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.

Tabela 36 – Descrição dos parâmetros do bloco PID

Legenda:

NA – Parâmetro Adimensional
RO – Somente Leitura
D – dinâmico
N – não volátil
S – estático

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);
DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);
BOOL: Um valor booleano (TRUE ou FALSE);
WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);
USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

Entrada Discreta de Dois Estados (DI)

Um bloco funcional de entrada discreta (DI) de dois estados recebe dados VST de um transdutor de sensor em sua entrada (In) como um sinal O-PAS™ por meio do OCF. Ele condiciona esses dados e os disponibiliza para outros blocos funcionais através de sua saída (Out).

As unidades de conformidade normativa para um bloco funcional de entrada discreta de dois estados O-PAS™ são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação entre blocos. A lógica funcional expressa no tópico seguinte reflete a implementação de referência, mas não é normativa.

Lógica Funcional

A lógica funcional do bloco funcional de entrada discreta de dois estados O-PAS™ é mostrada na Figura 18.

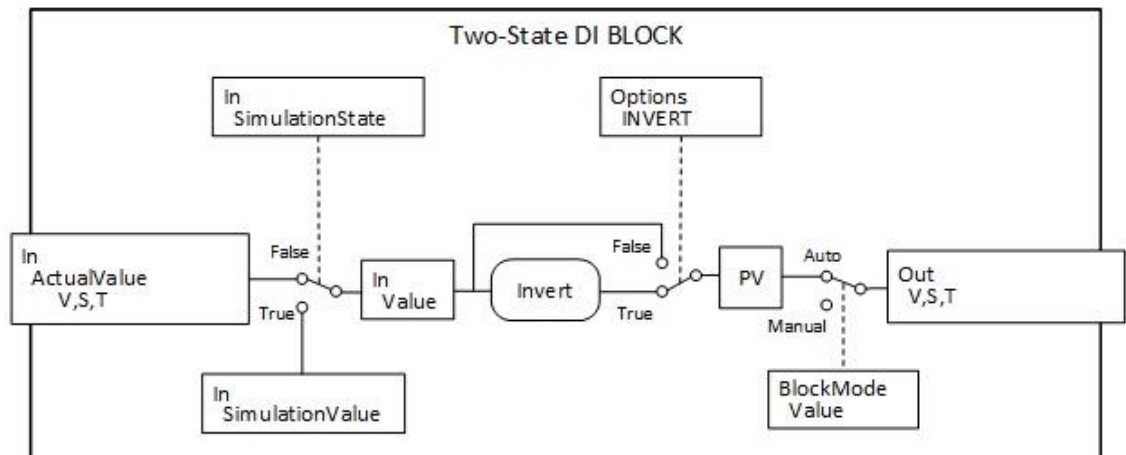


Figura 18: Esquemático do bloco funcional de entrada discreta de dois estados O-PAS™ (consulte a Seção “Syntax Notes for Function Block Logic Diagrams” para notas de sintaxe)

A entrada In é normalmente adquirida de um sensor por meio de um dispositivo de E/S. Um valor booleano, um número inteiro de status e um registro de data/hora (timestamp) são atribuídos a **In.ActualValue.Value**, **In.ActualValue.Status** e **In.ActualValue.Timestamp**.

Na operação normal, o valor de **In.SimulationState** é definido como falso e **In.ActualValue.Value**, **In.ActualValue.Status** e **In.ActualValue.Timestamp** são atribuídos a **In.Value**, **In.Status** e **In.Timestamp**. Se o valor de **In.SimulationState** for definido como verdadeiro, **In.SimulationValue**, **In.SimulationStatus** e **In.SimulationTimestamp** serão atribuídos a **In.Value**, **In.Status** e **In.Timestamp**.

Se **FunctionBlockOption Invert** for definido como verdadeiro, **In.Value** será alterado de verdadeiro para falso ou de falso para verdadeiro. Se **FunctionBlockOption Invert** estiver definido como falso, **In.Value** permanecerá inalterado.

Na implementação de referência, um filtro “debounce” é implementado definindo um valor positivo diferente de zero de Damping em unidades de um número inteiro de tempos de varredura do bloco funcional. Quando o filtro debounce está habilitado, o valor de trabalho pós-inversão de **In.Value** deve permanecer inalterado na janela inversa de amostras de amortecimento antes de ser alterado. O resultado dos cálculos do filtro de inversão e debounce é atribuído a **PV.Value**. **In.Status** é atribuído ao **PV.Status**.

O bloco funcional de entrada discreta de dois estados suporta os três modos a seguir: **OutOfService**, **Manual** e **Automatic**.

Se o modo for **Automatic**, **PV.Value** e **PV.Status** serão atribuídos a **Out.Value** e **Out.Status**. Uma consulta ao relógio do sistema no momento do cálculo de **Out** é feita para definir **Out.Timestamp**.

Se o modo for **Manual**, **PV.Value** não será atribuído a **Out.Value**. Em vez disso, um valor adquirido de uma fonte externa, normalmente a IHM, é atribuído a **Out.Value**. Uma consulta ao relógio do sistema no momento do cálculo de **Out** é feita para definir **Out.Timestamp**.

Modelo de Informação OCF

O **TwoStateDiscreteInputBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção “ControlFunctionBlockType Definition”. Um bloco funcional de entrada discreta de dois estados O-PAS™ deve estar em conformidade com o modelo de informações OCF especificado na Figura 19, Tabela 34 e Tabela 35. Deverá suportar, no mínimo, os seguintes Modos:

- Out of Service (Fora de serviço)
- Manual
- Automático

Um bloco funcional de entrada discreta de dois estados O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 20 e 21.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável neste modelo de informação OCF. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

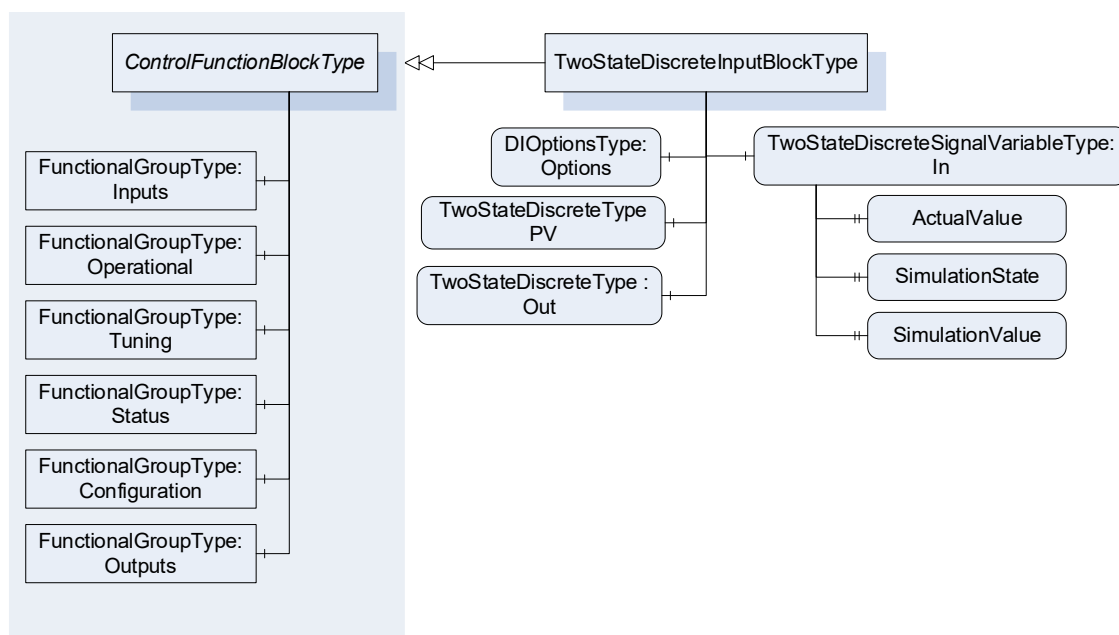


Figura 19: Modelo de dados TwoStateDiscreteInputBlockType

Attribute	Value				
BrowseName	TwoStateDiscreteInputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Boolean	3:TwoStateDiscreteSignalVariableType	Mandatory

Attribute	Value				
BrowseName	TwoStateDiscreteInputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
0:HasComponent	Variable	Out	0:Boolean	0:TwoStateDiscreteType	Mandatory
0:HasComponent	Variable	Damping	0:Number	0:BaseDataVariableType	Mandatory
0:HasComponent	Variable	Field	0:Boolean	0:BaseDataVariableType	Mandatory
0:HasComponent	Object	Options		DIOptionsType	Mandatory
0:HasComponent	Variable	PV	0:Boolean	0:TwoStateDiscreteType	Mandatory

Tabela 37 - Definição TwoStateDiscreteInputBlockType

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In ActualValue
Outputs	0:Organizes	True	Out
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	Options
Tuning	0:Organizes	True	Damping
Tuning	0:Organizes	True	In SimulationState
Tuning	0:Organizes	True	In SimulationValue
Operational	0:Organizes	True	Mode Target
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	In
Status	0:Organizes	True	PV

Tabela 38 - Referências Adicionais TwoStateDiscreteInputBlockType

O **DIOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration” com as opções aplicáveis obrigatórias. Está formalmente definido na **Tabela 37**.

Attribute	Value				
BrowseName	DIOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de 4: <i>FBOptionsType</i> definido em O-PAS™ Parte “Information and Exchange Models: Basic Configuration”.					
0:HasProperty	Variable	4:Invert	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:PropagateFaultForward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UncertainIfManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseUncertainAsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 39 - Definição de DIOptionsType

Os modos e a interface de coordenação entre blocos de um bloco funcional de entrada discreta de dois estados O-PAS™ são especificados nas Figuras 20 e 21.

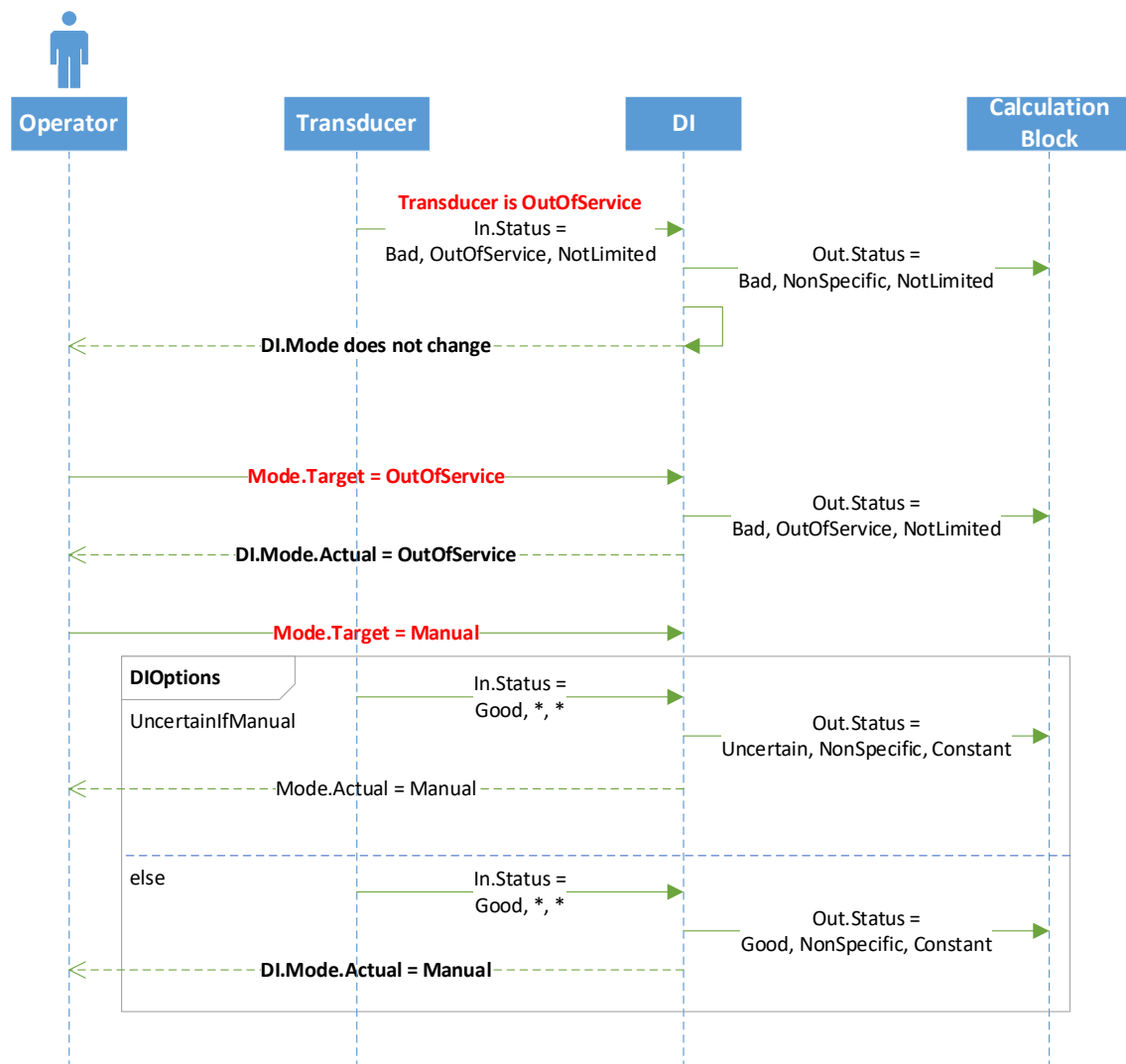


Figura 20 - Diagrama de sequência de modo de bloco funcional de entrada discreta de dois estados O-PAS™ e interface de coordenação entre blocos - Parte 1 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

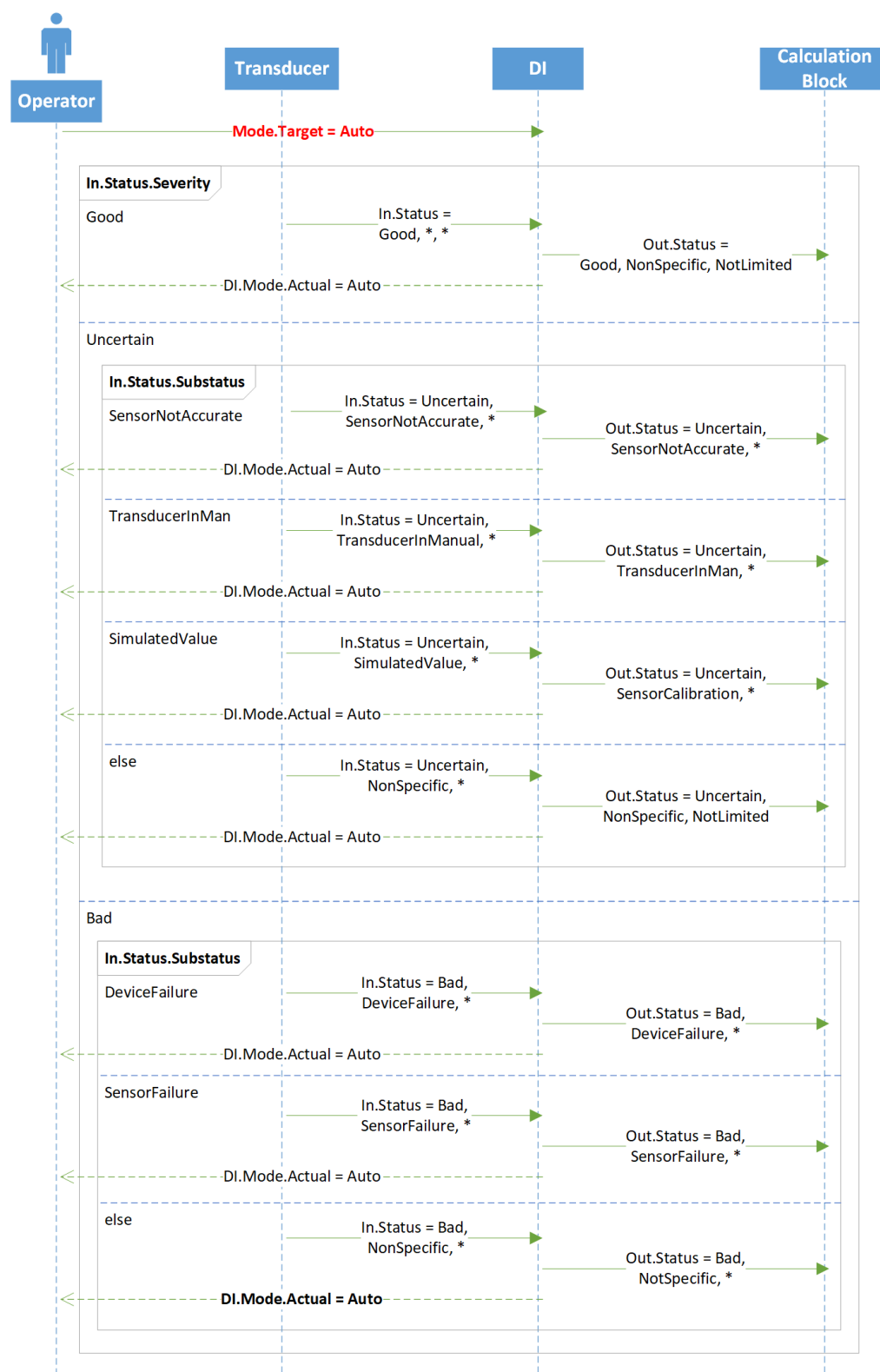


Figura 21 - Diagrama de sequência de modo de bloco funcional de entrada discreta de dois estados O-PAS™ e interface de coordenação entre blocos - Parte 2 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informação TwoStateDiscreteInputBlockType OCF e as variáveis do programa na implementação de referência O-PAS™ IEC 61131 é fornecido na Tabela 38.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados BlockType e na definição BlockType. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais do FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In ActualValue Value Status Timestamp Value Status Timestamp SimulationValue Value Status SimulationState	Input Input Input Status Status Status Tuning Tuning Tuning	Structure IN_D .ActualValue .ActualStatus .Timestamp FIELD_D .Value .Status .Timestamp SIMULATE_D .Value .Status .Enable	BOOL UDINT LDT ¹ BOOL UDINT LDT ¹ BOOL UDINT BOOL	EU of In na utcTime EU of In na utcTime EU of In na na	Valor de entrada do bloco de funções, status (gravidade, subcódigo, limite) e registro de data/hora do OSS que está lendo de um sensor de um dispositivo de campo.
Out Value Status Timestamp	Output Output Output	Structure OUT_D .Value .Status .Timestamp	REAL UDINT LDT ¹	EU of Out na utcTime	O valor de saída (transmitido), status (gravidade, subcódigo, limite) e registro de data/hora do valor do processo conforme calculado durante a execução do bloco funcional.
Damping	Tuning	PV_FTIME	REAL	na	Número de amostras que devem ser iguais antes que uma mudança de estado seja feita em In.Value.
Error	Status	BLOCK_ERR	WORD	na	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte “Information and Exchange Models: Alarm and Events Configuration”).

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Permitted e Normal. Modo é o valor real.
Options	Configuration	IO_OPTS STATUS_OPTS	BOOL	na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
PV Value Status	Status Status	PV .Value .Status	BOOL UDINT	EU of In na	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.

Tabela 40 - Dicionário de dados do bloco funcional de entrada discreta de dois estados O-PAS™ IEC 61131

(1) DATA_AND_TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável **Value** como uma estrutura contendo três atributos; ou seja, (valor, status, registro de data/hora).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
DAMPING	REAL		NA		Representa um tempo de filtragem em segundos (com casas decimais).
ERROR	WORD		NA		Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
FIELD	BOOL				Parâmetro interno do bloco que representa o valor de entrada antes de qualquer filtragem. Ou seja, é o valor Booleano da entrada antes de qualquer filtragem (Damping). Tem esse nome pois é como se fosse um valor de campo, aquilo que vem direto do equipamento.
IN Value SimulationState SimulationValue	BOOL		NA		In.Value é simplesmente o valor de entrada (0 ou 1). In.SimulationState representa ligar ou desligar a simulação. Se for 0, o bloco usa o valor de In.Value para ser processado, se for 1, o valor de In.Value é ignorado e substituído pelo valor de In.SimulationValue que o usuário escolher.
MODE Target RetainedTarget Normal	DINT	1 = OutOfService 2 = Manual 3 - Automatic	NA		Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, RetainedTarget e Normal. Modo é o valor real. RetainedTarget é uma memória do último modo válido que o bloco tinha antes de ir para uma condição especial (como OOS, falha ou perda de comando).
OPTIONS	BOOL	Invert PropagateFaultForward Uncertain UncertainIfManual	NA		Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
OUT Value	BOOL		NA		O valor de saída (transmitido), status (gravidade, subcódigo, limite) e registro de data/hora do valor do processo conforme calculado durante a execução do bloco funcional.
PV Value	BOOL		NA		Em bloco DI são processados valores digitais (0 ou 1), não analógicos. PV é a variável de processo. Em condições de operação Normal do bloco (MODE = AUTO) o valor de IN passa para PV que passa para OUT.

Tabela 41 – Descrição dos parâmetros do bloco DI

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

Saída Discreta de Dois Estados (DO)

Um bloco funcional de saída discreta de dois estados (DO) obtém um valor de setpoint booleano e status de um bloco funcional upstream em sua entrada (In) e o converte em um sinal em sua saída (Out) que é adequado para transmissão através do OCF para o atuador. O bloco funcional de saída discreta de dois estados também fornece feedback do atuador que é usado para inicialização sem interrupções (bumpless) da ação de controle.

As unidades de conformidade normativa para um bloco funcional de saída discreta de dois estados O-PAS™ são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação entre blocos. A lógica funcional expressa no tópico seguinte reflete a implementação de referência, mas não é normativa.

Lógica Funcional

A lógica funcional do bloco funcional de saída discreta de dois estados O-PAS™ é mostrada na Figura 22.

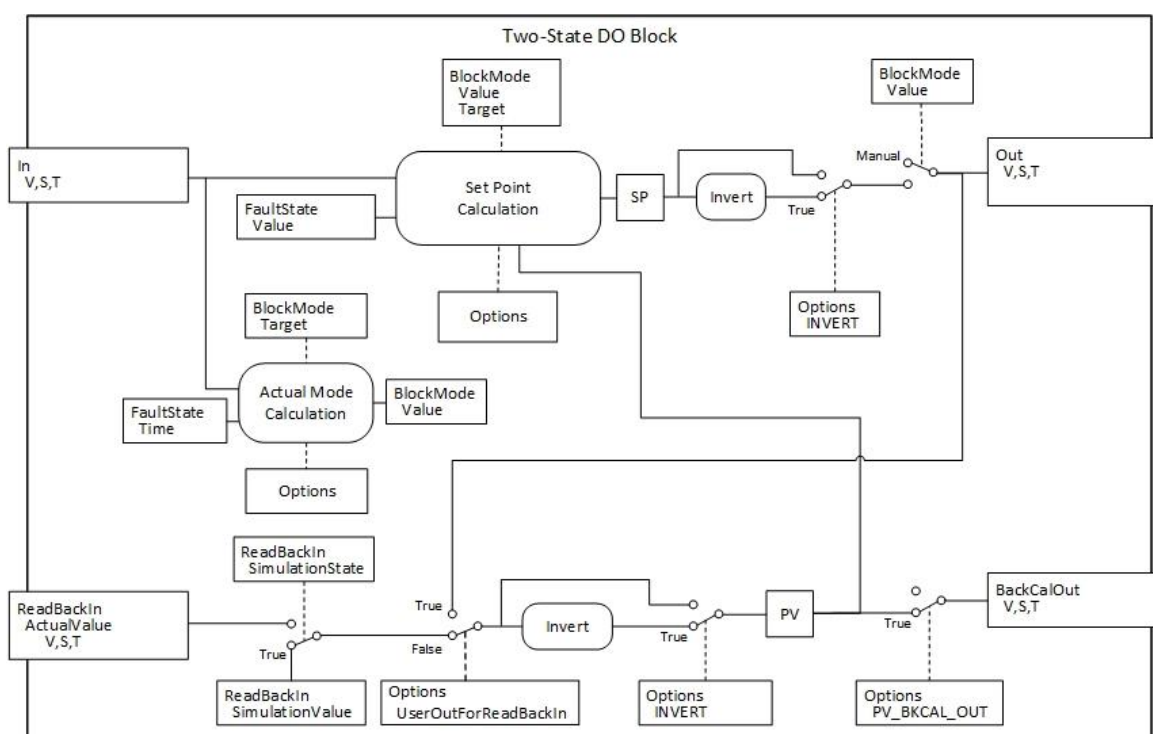


Figura 22 - Esquema do bloco funcional de saída discreta de dois estados O-PAS™ (consulte a Seção “Syntax Notes for Function Block Logic Diagrams” para notas de sintaxe)

Existem três fluxos de dados primários no bloco funcional de Saída Discreta de Dois Estados. O resumo geral desses três fluxos de dados é o seguinte:

1. A entrada **In** é recebida de um bloco funcional upstream. Este valor é passado para a saída **Out** para transmissão final a um atuador. Normalmente, **In** está conectado a um bloco funcional upstream que executa um cálculo de controle.
2. A entrada **ReadBackIn** é adquirida de um atuador. Ela é usada para calcular a variável interna **PV** e para rastrear o valor real do atuador para inicialização sem interrupções (bumpless) do bloco funcional de saída discreta de dois estados. Se o atuador e os dispositivos de E/S suportarem o envio de dados, como a posição da válvula, a opção **UseOutForReadBackIn** poderá ser definida como falsa, o que direciona o valor da posição da válvula para o valor **PV**.
3. A saída **BackCalOut** é calculada para transmissão no caminho de fluxo de dados reverso para um bloco funcional upstream com a finalidade de inicialização sem interrupções (bumpless) do referido bloco funcional upstream.

O bloco funcional de saída discreta de dois estados suporta os seis modos a seguir: **OutOfService**, **InitializationManual**, **LocalOverride**, **Manual**, **Automatic** e **Cascade**.

Cascade é o modo típico para uso online. **In.Value**, **In.Status** e **In.Timestamp** são adquiridos de um bloco funcional upstream. Os dados **In.Value** e **In.Status** são atribuídos a **SP.Value** e **SP.Status**. Se **DOOption Invert** for definido como verdadeiro, **SP.Value** será alterado de verdadeiro para falso ou de falso para verdadeiro, e o valor resultante será atribuído a **Out.Value**. Se **Invert** for definido como falso, **SP.Value** será atribuído sem alteração a **Out.Value**. **SP.Status** é atribuído a **Out.Status**. Uma consulta ao relógio do sistema no momento do cálculo de **Out** é feita para definir **Out.Timestamp**.

Quando o Modo é **Automatic**, um valor adquirido de uma fonte externa, normalmente a IHM, é gravado em **SP.Value** em vez de ser propagado de **In**. Então, exatamente como descrito no parágrafo **Cascade** acima, o valor da IHM pode ser invertido antes de ser atribuído a **Out.Value**.

Quando o Modo é **Manual**, um valor adquirido de uma fonte externa, normalmente a IHM, é atribuído ao **Out.Value** para transmissão ao atuador.

A inicialização bumpless durante a transição do modo manual é feita fazendo com que **SP.Value** e **SP.Status** rastreiem o status do valor real do atuador que é adquirido por meio das entradas **ReadBackIn.ActualValue.Value** e **ReadBackIn.ActualValue.Status** pela seguinte lógica:

- O valor e status do atuador são adquiridos em **ReadBackIn.ActualValue.Value** e **ReadBackIn.ActualValue.Status**; **ReadBackIn.ActualValue.Timestamp** também pode ser adquirido ou atribuído
- Em operação normal, o valor de **ReadBackIn.SimulationState** é definido como falso e **ReadBackIn.ActualValue.Value**, **ReadBackIn.ActualValue.Status** e **ReadBackIn.ActualValue.Timestamp** são atribuídos a **ReadBackIn.Value**, **ReadBackIn.Status** e **ReadBackIn.Timestamp**. Se o valor de **ReadBackIn.SimulationState** for definido como verdadeiro, **ReadBackIn.SimulationValue** e **ReadBackIn.SimulationStatus** serão atribuídos a **ReadBackIn.Value** e **ReadBackIn.Status**.
- Quando **DOOption UseOutForReadBackIn** é definido como falso, o **ReadBackIn.Value** é passado para a função **Invert**. Se **DOOption Invert** for definido como verdadeiro, o valor **ReadBackIn** será alterado de verdadeiro para falso ou de falso para verdadeiro, e o valor resultante será atribuído a **PV.Value**. Se **Invert** for definido como falso, **ReadBackIn.Value** será atribuído sem alteração a **PV.Value**. **ReadBackIn.Status** é atribuído a **PV.Status**.
- Finalmente, se **DOOption SpPvTrackInManual** estiver definido como verdadeiro, **PV.Value** e **PV.Status** serão atribuídos a **SP.Value** e **SP.Status**.

Quando o Modo é **LocalOverride**, **SP.Value** e **SP.Status** são atribuídos de **In.Value** e **In.Status** ou de **FaultStateSP.Value** e **FaultStateSP.Status** pela lógica a seguir.

- Se **Mode.Target** for **Cascade** e se **DOOption FaultStateToValue** estiver definido como verdadeiro, **FaultStateSP.Value** será atribuído a **SP.Value**

Os valores de **BackCalOut.Value** e **BackCalOut.Status** são atribuídos de **SP.Value** e **SP.Status** ou de **PV.Value** e **PV.Status** dependendo do valor de **DOOption UsePvForBackCalOut**. Os valores de **SP.Value** e **SP.Status** dependem do Modo descrito acima. Os valores de **PV.Value** e **PV.Status** são determinados pela lógica descrita no parágrafo Modo **Manual** acima. Quando **UsePvForBackCalOut** é definido como verdadeiro, **PV.Value** e **PV.Status** são atribuídos a **BackCalOut.Value** e **BackCalOut.Status**. Quando **UsePvForBackCalOut** é definido como falso, **SP.Value** e **SP.Status** são atribuídos a **BackCalOut.Value** e **BackCalOut.Status**. Uma consulta ao relógio do sistema no momento do cálculo do **BackCalOut** é feita para definir **BackCalOut.Timestamp**.

Modelo de Informação OCF

O **TwoStateDiscreteOutputBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção "ControlFunctionBlockType Definition". Um bloco funcional de saída discreta de dois estados O-PAS™ deve estar em conformidade com o modelo de informações OCF especificado na Figura 23, Tabela 40 e Tabela 41. Deverá suportar, no mínimo, os seguintes Modos:

- **OutOfService**
- **InitializationManual**

- LocalOverride
- Manual
- Automatic
- Cascade

Um bloco funcional de saída discreta de dois estados O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 24 e 25.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

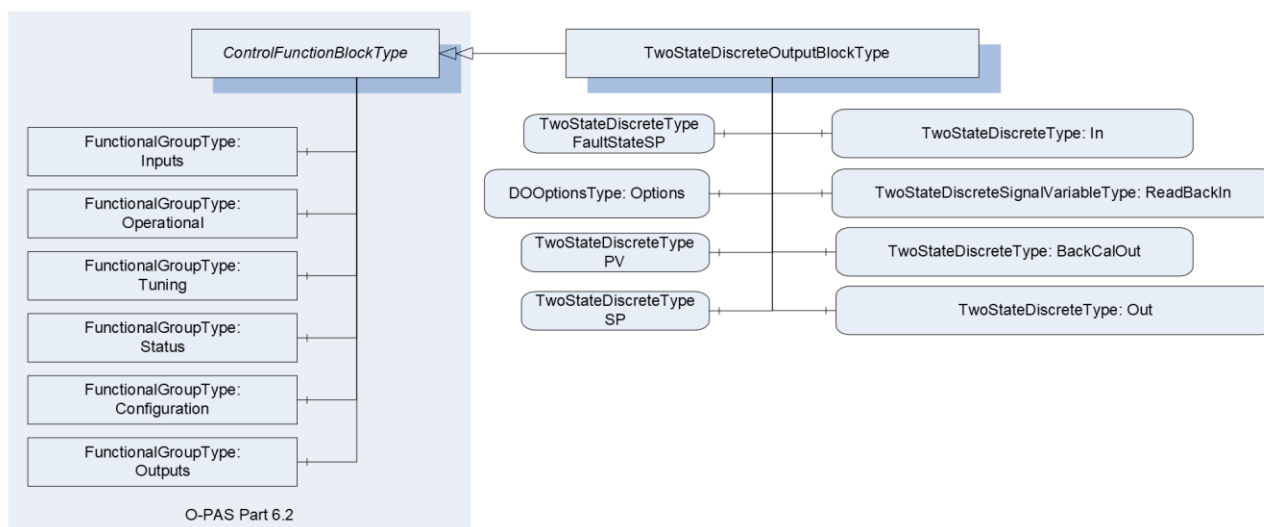


Figura 23 - Modelo de dados TwoStateDiscreteOutputBlockType

Attribute	Value				
BrowseName	TwoStateDiscreteOutputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da Seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Boolean	0:TwoStateDiscreteType	Mandatory
0:HasComponent	Variable	ReadBackIn	0:Boolean	3:TwoStateDiscreteSignalVariableType	Mandatory
0:HasComponent	Variable	BackCalOut	0:Boolean	0:TwoStateDiscreteType	Mandatory
0:HasComponent	Variable	Out	0:Boolean	0:TwoStateDiscreteType	Mandatory
0:HasComponent	Variable	FaultStateSP	0:Boolean	FaultStateSPType	Mandatory
0:HasComponent	Object	Options		DDOptionsType	Mandatory

Attribute	Value				
BrowseName	TwoStateDiscreteOutputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
0:HasComponent	Variable	PV	0:Boolean	0:TwoStateDiscreteT ype	Mandatory
0:HasComponent	Variable	SP	0:Boolean	0:TwoStateDiscreteT ype	Mandatory

Tabela 42 - Definição TwoStateDiscreteOutputBlockType

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In
Inputs	0:Organizes	True	ReadBackIn
Outputs	0:Organizes	True	BackCalOut
Outputs	0:Organizes	True	Out
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	Options
Tuning	0:Organizes	True	FaultStateSP
Tuning	0:Organizes	True	ReadBackIn SimulationState
Tuning	0:Organizes	True	ReadBackIn SimulationValue
Operational	0:Organizes	True	Mode Target
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	PV
Status	0:Organizes	True	SP

Tabela 43 - Referências Adicionais TwoStateDiscreteOutputBlockType

O **DOOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration” com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 42.

Attribute	Value				
BrowseName	DOOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de 4: <i>FBOptionsType</i> definido em O-PAS™ Parte "Information and Exchange Models: Basic Configuration".					
0:HasProperty	Variable	4:FaultStateTo Value	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:Invert	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:PropagateFaultBackward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInLO	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpTrackRetainedTarget	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfFaultStateActive	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToNextPermittedModelIfBadCaseIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseFaultStateValueOnRestart	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseOutForReadBackIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UsePvForBackCalcOut	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseUncertainAsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 44 - Definição *DOOptionsType*

Os modos e a interface de coordenação entre blocos de um bloco funcional de saída discreta de dois estados O-PAS™ são especificados nas Figuras 24 e 25.

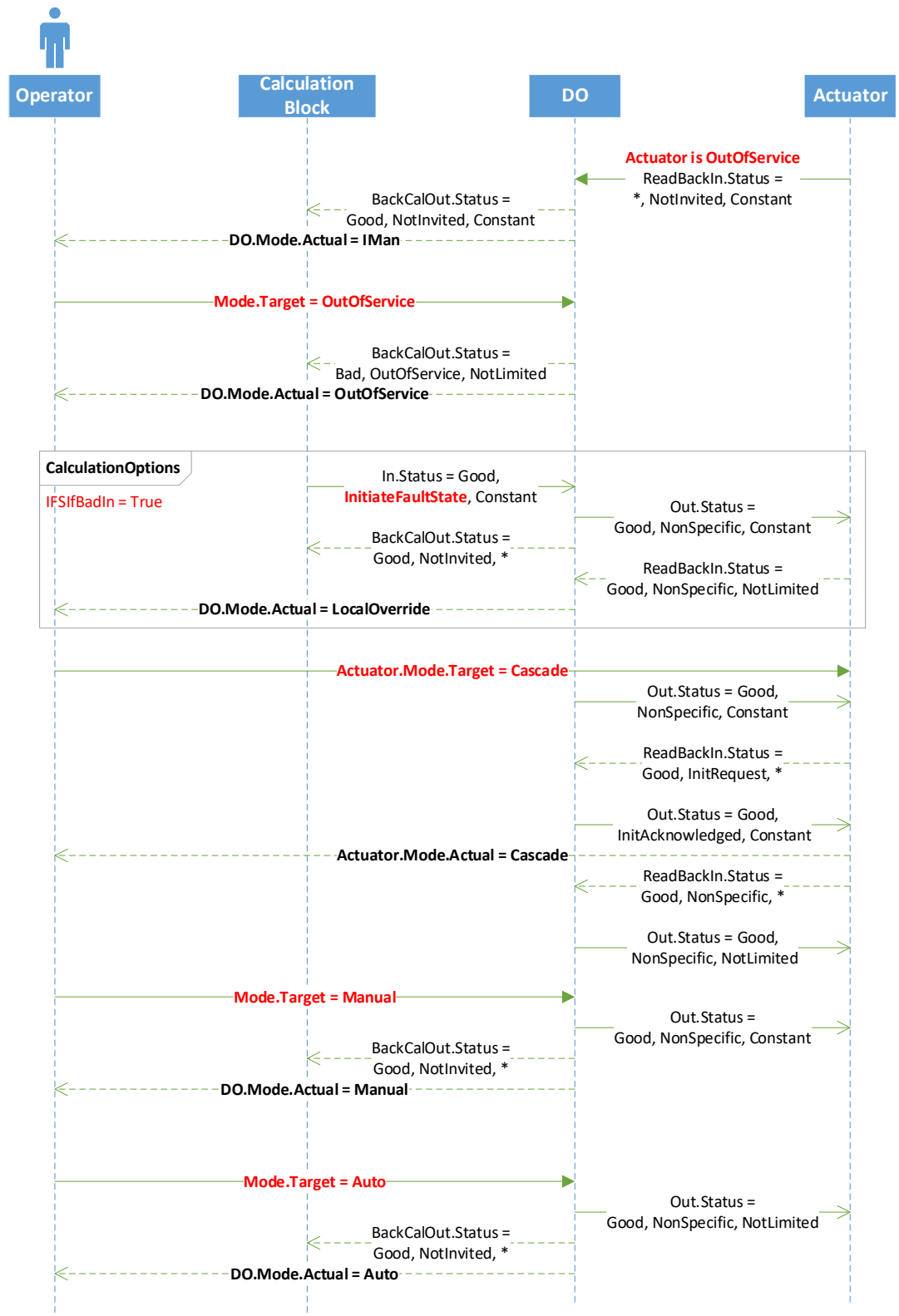


Figura 24 - Diagrama de sequência de modo de bloco funcional de saída discreta de dois estados O-PAS™ e interface de coordenação entre blocos - Parte 1 de 2 (consulte a Seção "Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams" para notas sobre sintaxe UML)

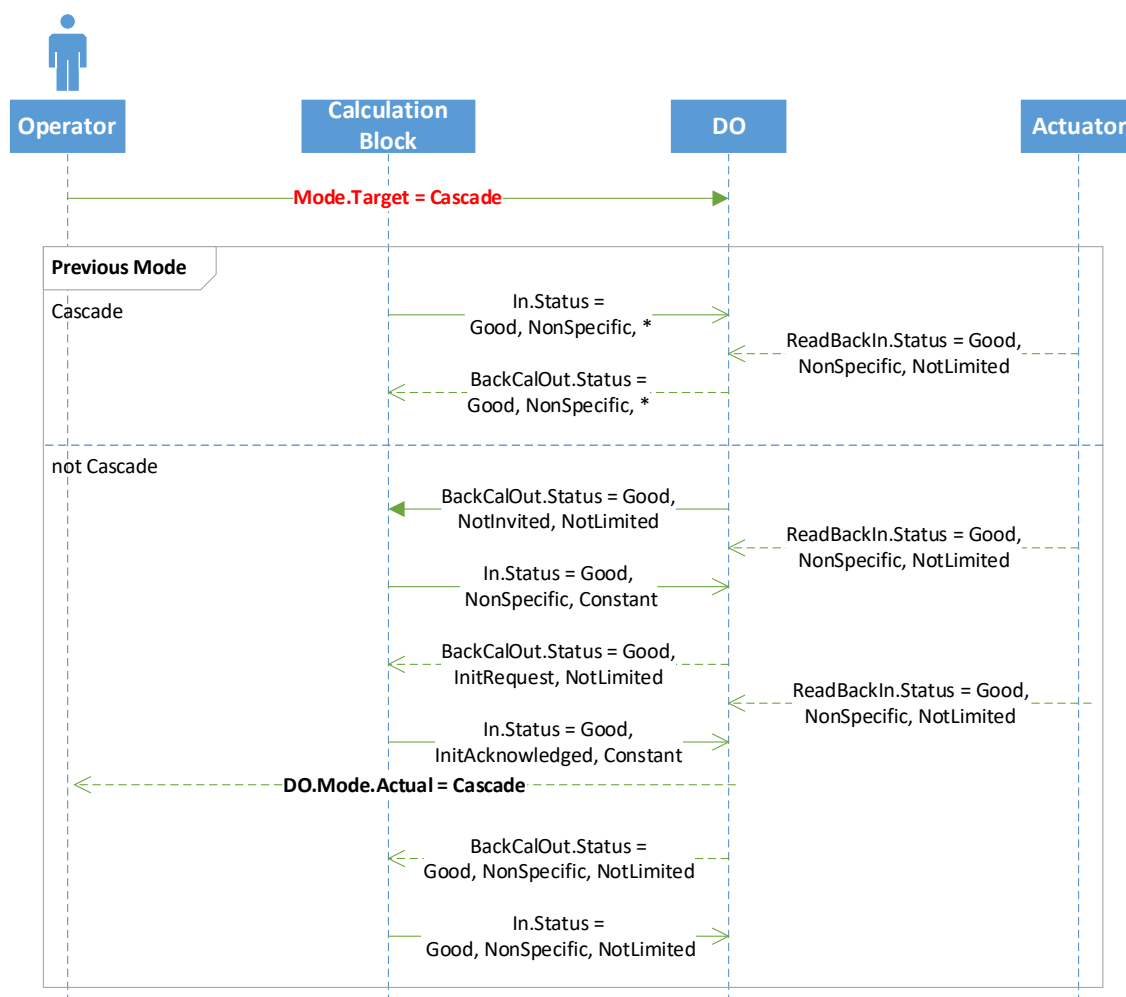


Figura 25 - Diagrama de sequência de modo de bloco funcional de saída discreta de dois estados O-PAS™ e interface de coordenação entre blocos - Parte 2 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informação TwoStateDiscreteOutputBlockType OCF e as variáveis do programa na implementação de referência O-PAS™ IEC 61131 é fornecido na Tabela 43.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados **BlockType** e na definição **BlockType**. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In Value Status Timestamp	Input Input Input	Structure CAS_IN_D .Value .Status .Timestamp	BOOL UDINT LDT ¹		Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador.
ReadBackIn ActualValue Value Status Timestamp Value Status Timestamp SimulationValue Value Status SimulationState	Input Input Input Status Status Status Tuning Tuning Tuning	Structure RDBK_IN_D .Value .Status .Timestamp SIMULATE_D .Value .Status .Enable	BOOL UDINT LDT ¹ BOOL UDINT LDT BOOL UDINT BOOL		Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação seguem o valor e o status reais do OSS (Fonte).
BackCalln Value Status Timestamp	Output Output Output	Structure BKCAL_D .Value .Status .Timestamp	BOOL UDINT LDT ¹		
Out Value Status Timestamp	Output Output Output	Structure OUT_D .Value .Status .Timestamp	BOOL UDINT LDT ¹		O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.
Error	Status	BLOCK_ERR	BOOL	na	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte Information and Exchange Models: Alarm and Events Configuration).

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
Mode .Target EnumValues .Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
FaultStateSP Time Value	Tuning Tuning	FSTATE_TIME FSTATE_VAL_D	TIME BOOL	Sec na	Tempo em segundos até o tempo limite (timeout) em CAS_IN. Valor a ser usado em FAULT_STATE.
Options	Configuration	IO_OPTS_DO SHED_OPT STATUS_OPTS_D	BOOL	na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
PV Value Status	Status Status	PV_D .Value .Status	BOOL UDINT	EU of PV	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.
SP Value Status	Status Status	SP_D .Value .Status	BOOL UDINT		

Tabela 45 - Dicionário de dados do bloco funcional de saída discreta de dois estados O-PAS™ IEC 61131

(1) DATA_AND_TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável Value como uma estrutura contendo três atributos; ou seja, (valor, status, registro de data/hora).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
BACKCALOUT Value	BOOL	True ou False	NA		A saída BackCalOut é calculada para transmissão no caminho reverso de fluxo de dados para um bloco funcional upstream com a finalidade de inicialização sem interrupções (bumpless) do referido bloco funcional upstream.
ERROR	WORD		NA		Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte Information and Exchange Models: Alarm and Events Configuration).
FAULTSTATESP Time	TIME		s		Tempo em segundos até o tempo limite (timeout) em CAS_IN. Valor a ser usado em FAULT_STATE.
IN Value	BOOL	True ou False	NA		Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador.
MODE Normal Target	DINT	OutOfService InitializationManual LocalOverride Manual Automatic Cascade	NA		Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
OPTIONS	BOOL	FaultStateToValue Invert PropagateFaultBackward SpPvTrackInLO SpPvTrackInManual SpTrackRetainedTarget TargetToManualIfFaultStateActive TargetToNextPermittedModelIfBad CasIn UseFaultStateValueOnRestart UseOutForReadBackIn UsePvForBackCalOut UseUncertainAsGood	NA		Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
OUT Value	BOOL	True ou False	NA		O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.
PV Value	BOOL	True ou False	EU of PV		Valor interno (parâmetro contido) que indica o estado discreto atual da saída do bloco (por exemplo, LIGADO/DESLIGADO), incluindo o status associado. O valor é determinado durante a execução do bloco funcional conforme o modo de operação e os comandos de controle.

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
READBACKIN Value SimulationValue SimulationState	BOOL	True ou False	NA		Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação seguem o valor e o status reais do OSS (Fonte).
SP Value	BOOL	True ou False	NA		Estado discreto desejado (por exemplo, LIGADO/DESLIGADO) para a saída do bloco. Representa o comando de controle solicitado pelo operador, outro bloco ou sistema superior.

Tabela 46 – Descrição dos parâmetros do bloco DO

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

Entrada Discreta Multiestados (MDI)

Um bloco funcional de entrada discreta multiestados (MDI) recebe dados VST de um transdutor de sensor em sua entrada (In) como um sinal O-PAS™ por meio do OCF. Disponibiliza esses dados para outros blocos funcionais através de sua saída (Out).

As unidades de conformidade normativa para um bloco funcional de entrada discreta multiestados O-PAS™ são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação entre blocos. A lógica funcional expressa no tópico seguinte reflete a implementação de referência, mas não é normativa.

Lógica Funcional

A lógica funcional do bloco funcional de entrada discreta multiestado O-PAS™ é mostrada na Figura 26.

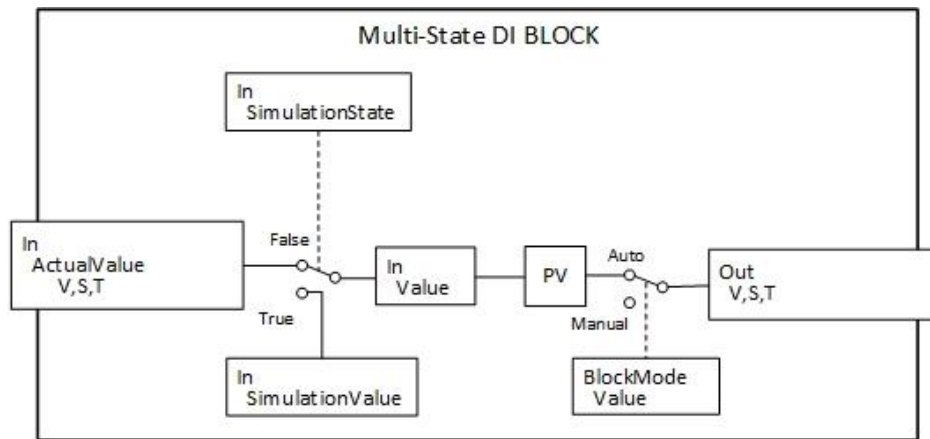


Figura 26 - Esquema do bloco funcional de entrada discreta multiestados O-PAS™ (consulte a Seção “Syntax Notes for Function Block Logic Diagrams” para notas de sintaxe)

A entrada In é normalmente adquirida de um sensor por meio de um dispositivo de E/S. Um valor inteiro representando uma enumeração, um número inteiro de status e um registro de data/hora são atribuídos a **In.ActualValue.Value**, **In.ActualValue.Status** e **In.ActualValue.Timestamp**.

Em operação normal, o valor de **In.SimulationState** é definido como falso e **In.ActualValue.Value**, **In.ActualValue.Status** e **In.ActualValue.Timestamp** são atribuídos a **In.Value**, **In.Status** e **In.Timestamp**. Se o valor de **In.SimulationState** for definido como verdadeiro, **In.SimulationValue** e **In.SimulationStatus** serão atribuídos a **In.Value** e **In.Status**.

Na implementação de referência, um filtro “debounce” é implementado definindo um valor positivo diferente de zero de Damping em unidades de um número inteiro de tempos de varredura do bloco funcional. Quando o filtro debounce está habilitado, o valor de trabalho de **In.Value** deve permanecer inalterado na janela inversa de amostras de amortecimento antes de ser alterado. O resultado dos cálculos do filtro de inversão e debounce é atribuído a **PV.Value**. **In.Status** é atribuído ao **PV.Status**.

O bloco funcional de entrada discreta multiestados suporta e pode operar nos modos OutOfService, Manual e Automatic.

Se o Modo for Automático, **PV.Value** e **PV.Status** serão atribuídos a **Out.Value** e **Out.Status**. Uma consulta ao relógio do sistema é feita no momento do cálculo de **Out** para definir **Out.Timestamp**.

Se o Modo for Manual, **PV.Value** não será atribuído a **Out.Value**. Em vez disso, um valor adquirido de uma fonte externa, normalmente a IHM, é atribuído a **Out.Value**. Uma consulta ao relógio do sistema é feita no momento do cálculo de **Out** para definir **Out.Timestamp**.

Modelo de Informação OCF

O **MultiStateDiscreteInputBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção “ControlFunctionBlockType Definition”. Um bloco funcional de entrada discreta multiestado O-PAS™ deve estar em conformidade com o modelo de informação OCF especificado na Figura 27, Tabela 45 e Tabela 46. Deverá suportar, no mínimo, os seguintes Modos:

- OutOfService
- Manual
- Automatic

Um bloco funcional de entrada discreta multiestado O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 28 e 29.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

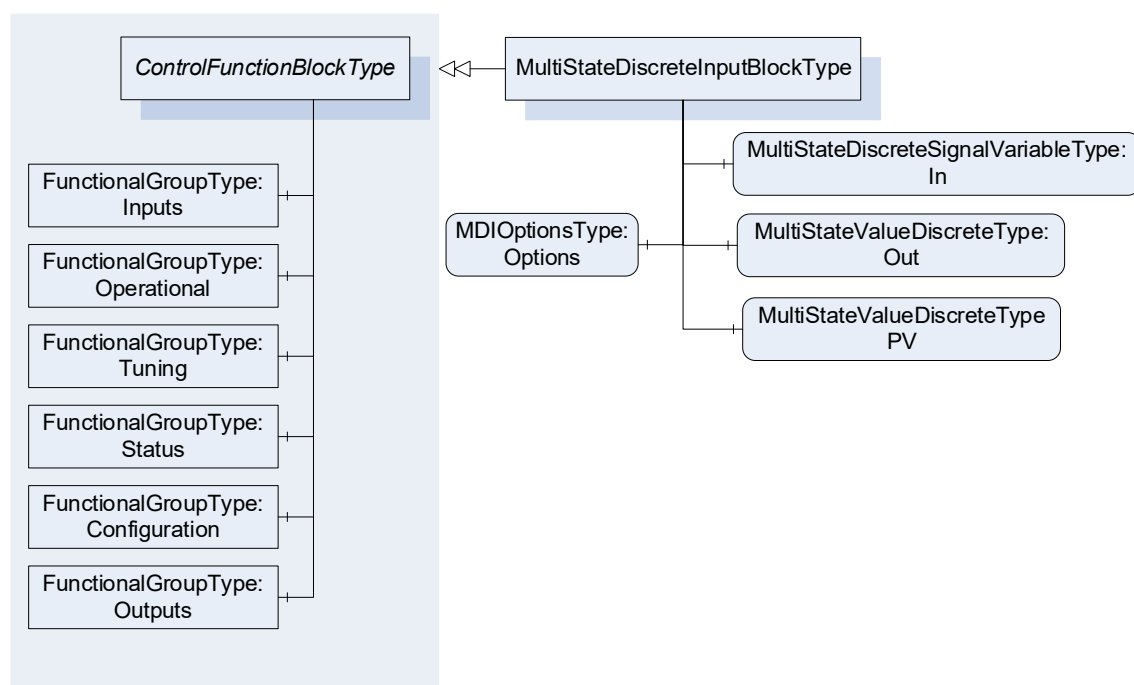


Figura 27 - Modelo de dados MultiStateDiscreteInputBlockType

Attribute	Value				
BrowseName	MultiStateDiscreteInputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Number	3:MultiStateDiscreteSignalVariableType	Mandatory
0:HasComponent	Variable	Out	0:Number	0:MultiStateValueDiscreteType	Mandatory

Attribute	Value				
BrowseName	MultiStateDiscreteInputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
0:HasComponent	Object	Options		MDIOptionsType	Mandatory
0:HasComponent	Variable	PV	0:Number	0:MultiStateValueDiscreteType	Mandatory

Tabela 47 - Definição MultiStateDiscreteInputBlockType

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In ActualValue
Outputs	0:Organizes	True	Out
Configuration	0:Organizes	True	In EnumValues
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	Options
Configuration	0:Organizes	True	Out EnumValues
Tuning	0:Organizes	True	Damping
Tuning	0:Organizes	True	In SimulationState
Tuning	0:Organizes	True	In SimulationValue
Operational	0:Organizes	True	Mode Target
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	In Value
Status	0:Organizes	True	PV

Tabela 48 - Referências Adicionais MultiStateDiscreteInputBlockType

O **MDIOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration” com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 47.

Attribute	Value				
BrowseName	MDIOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	Type Definition	Modeling Rule
Subtipo de 4: <i>FBOptionsType</i> definido em O-PAS™ Parte Information and Exchange Models: Basic Configuration.					
0:HasProperty	Variable	4:PropagateFaultForward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UncertainIfManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseUncertainAsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 49 - Definição de MDIOptionsType

Os modos e a interface de coordenação entre blocos de um bloco funcional de entrada discreta multiestado O-PAS™ são especificados nas Figuras 28 e 29.

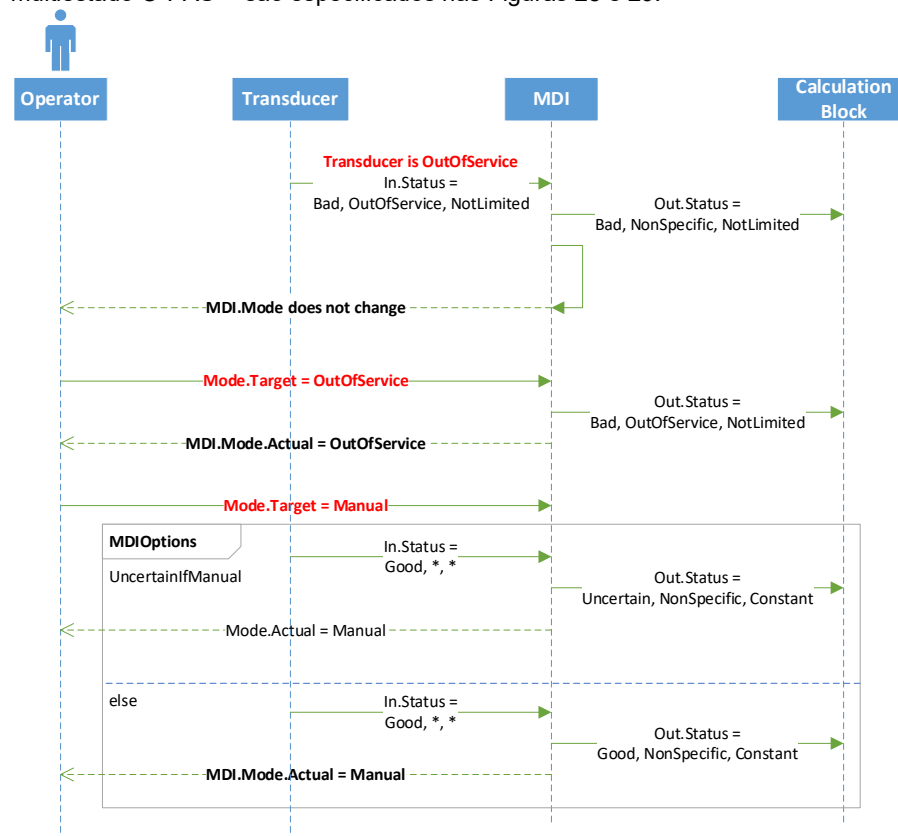


Figura 28 - Diagrama de sequência de modo de bloco funcional de entrada discreta multiestado O-PAS™ e interface de coordenação entre blocos - Parte 1 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

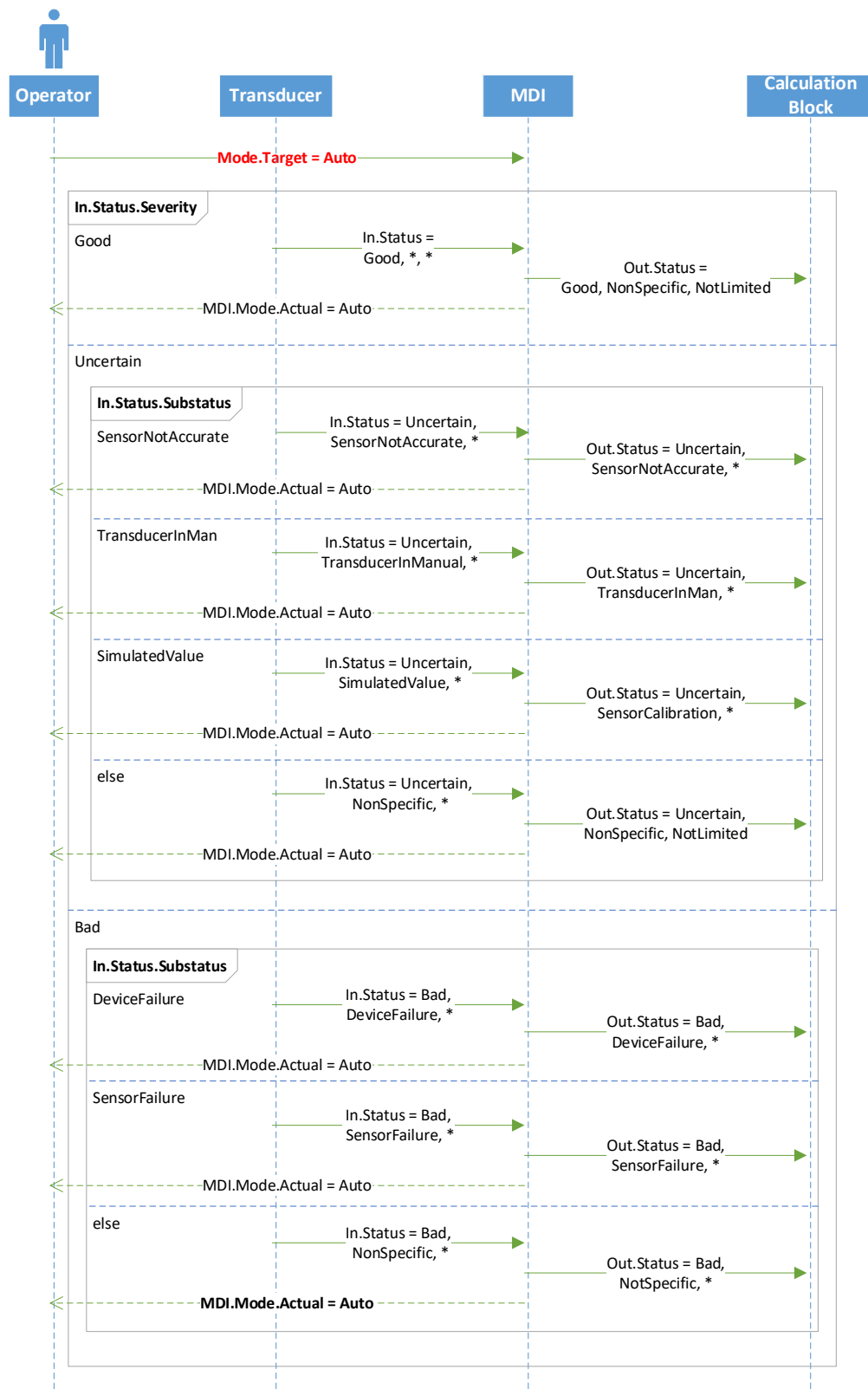


Figura 29 - Diagrama de sequência de modo de bloco funcional de entrada discreta multiestado O-PAS™ e interface de coordenação entre blocos - Parte 2 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informações **MultiStateDiscreteInputBlockType** OCF e as variáveis do programa na implementação de referência do bloco funcional de entrada discreta multiestado O-PAS™ IEC 61131 é fornecido na Tabela 48.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados **BlockType** e na definição **BlockType**. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In ActualValue Value Status Timestamp Value Status Timestamp EnumValues ValueAsText SimulationValue Value Status SimulationState	Input Input Input Status Status Status Configuration Configuration Tuning Tuning Tuning	Structure IN_D .ActualValue .ActualStatus .Timestamp FIELD_D .Value .Status .Timestamp XD_STATE SIMULATE_D .Value .Status .Enable	ULINT UDINT LDT ¹ ULINT UDINT LDT ¹ ULINT STRING ULINT UDINT BOOL	EU of In na utcTime EU of In na utcTime EU of In EU of In na na	Valor de entrada do bloco funcional, status e registro de data/hora do OSS (Fonte) que está lendo de um sensor de um dispositivo de campo. EnumValues é uma lista de possíveis estados de In. Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação rastreiam o valor e o status reais do OSS (Fonte).
Out Value Status Timestamp EnumValues ValueAsText	Output Output Output Configuration Configuration	OUT_D .Value .Status .Timestamp OUT_STATE	ULINT UDINT LDT ¹ ULINT STRING	EU of PV	O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.
Damping	Tuning	PV_FTIME	REAL	na	Número de amostras que devem ser iguais antes que uma mudança de estado seja feita em In.Value.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
Error	Status	BLOCK_ERR	WORD	na	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
Options	Configuration	IO_OPTS STATUS_OPTS	WORD	na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
PV Value Status EnumValue ValueAsText	Status Status Status Status	PV_D .Value .Status	ULINT UDINT ULINT STRING	EU of In na	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.

Tabela 50 - Dicionário de dados do bloco funcional de entrada discreta multiestado O-PAS™

(1) DATA_AND_TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável Value como uma estrutura contendo três atributos; ou seja, (valor, status, registro de data/hora).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
DAMPING	REAL		NA		Representa um tempo de filtragem em segundos (com casas decimais).
ERROR	WORD		NA		Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
FIELD					Valor interno (parâmetro contido) que representa o estado bruto das entradas discretas de campo, conforme lido diretamente do hardware ou canal físico do dispositivo, antes da aplicação de qualquer filtragem, verificação de status ou processamento lógico.
IN SimulationState SimulationValue ValuePrecision	BOOL ULINT UDINT				Valor de entrada do bloco funcional, status e registro de data/hora do OSS (Fonte) que está lendo de um sensor de um dispositivo de campo. Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação rastreiam o valor e o status reais do OSS (Fonte). ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC. No entanto, o bloco MDI trabalha apenas com números inteiros.
MODE Normal Target	DINT	OutOfService Manual Automatic Cascade	NA		Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
OPTIONS	BOOL	PropagateFaultForward UncertainIfManual UseUncertainAsGood	NA		Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
OUT Value	ULINT		EU de PV		O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.
PV Value	ULINT		EU de In		Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.

Tabela 51 – Descrição dos parâmetros do bloco MDI

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

ULINT: Um valor inteiro sem sinal de 64 bits

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

Saída Discreta Multiestados (MDO)

Um bloco funcional de saída discreta multiestados (MDO) pega um valor de setpoint inteiro de vários valores e status de um bloco funcional upstream em sua entrada (In) e o converte em um sinal em sua saída (Out) que é adequado para transmissão via o OCF, em última análise, para o atuador. O bloco funcional de saída discreta multiestado também fornece feedback do atuador que é usado para inicialização contínua da ação de controle.

As unidades de conformidade normativa para um bloco funcional de saída discreta multiestado O-PAS™ são especificadas em termos do modelo de informação OCF, dos modos suportados e da interface de coordenação entre blocos. A lógica funcional expressa no tópico seguinte reflete a implementação de referência, mas não é normativa.

Lógica Funcional

A lógica funcional do bloco funcional de saída discreta multiestado O-PAS™ é mostrada na Figura 30.

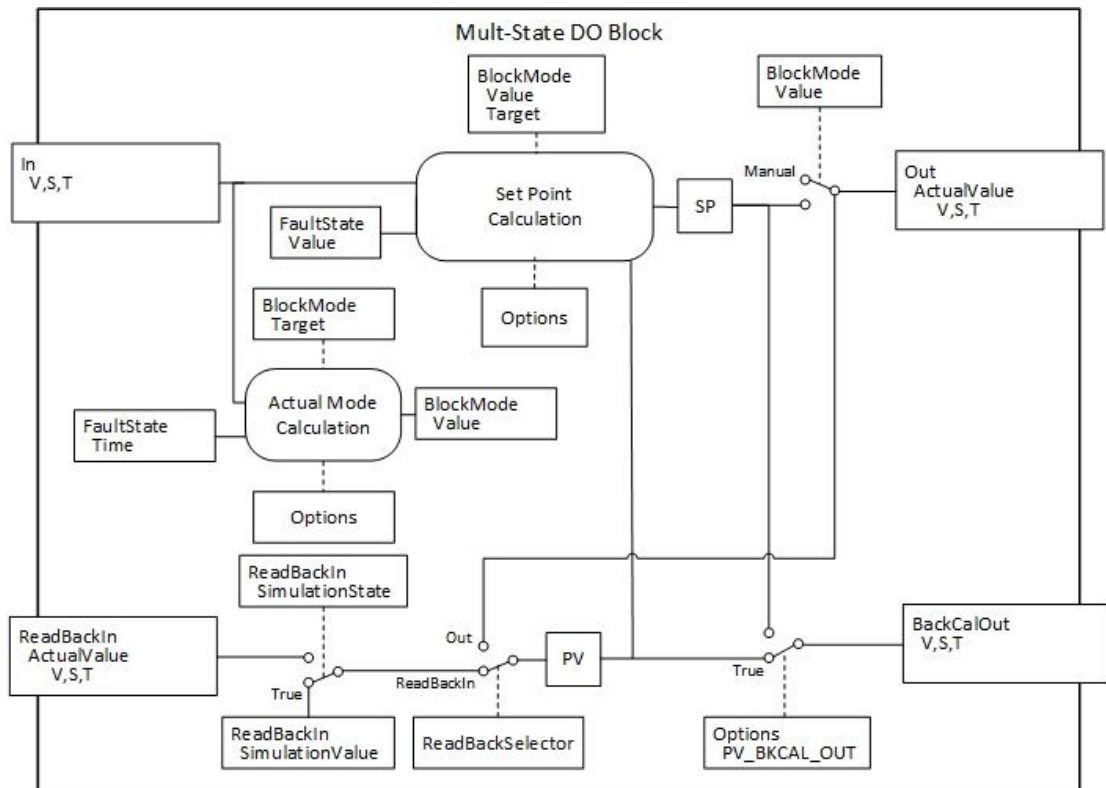


Figura 30 - Esquema do bloco funcional de saída discreta multiestado O-PAS™ (consulte a Seção “Syntax Notes for Function Block Logic Diagrams” para notas de sintaxe)

Existem três fluxos de dados primários no bloco funcional de saída discreta multiestados. O resumo geral desses três fluxos de dados é o seguinte:

1. A entrada **In** é recebida de um bloco funcional upstream. Este valor é passado para a saída **Out** para transmissão final a um atuador. Normalmente, **In** está conectada a um bloco funcional upstream que executa um cálculo de controle.
2. A entrada **ReadBackIn** é adquirida de um atuador. Ela é usada para calcular a variável interna **PV** e para rastrear o valor real do atuador para inicialização sem interrupções (bumpless) do bloco funcional de saída discreta multiestados. Se o atuador e os dispositivos de E/S suportarem o envio de dados, como a posição da válvula, a opção **UseOutForReadBackIn** poderá ser definida como falsa, o que direciona o valor da posição da válvula para o valor de **PV**.

3. A saída **BackCalOut** é calculada para transmissão no caminho de fluxo de dados reverso para um bloco funcional upstream com a finalidade de inicialização sem interrupções (bumpless) do referido bloco funcional upstream.

O bloco funcional de saída discreta multiestados suporta os seis modos a seguir: **OutOfService**, **InitializationManual**, **LocalOverride**, **Manual**, **Automatic** e **Cascade**.

Cascade é o modo típico para uso online. **In.Value**, **In.Status** e **In.Timestamp** são adquiridos de um bloco funcional upstream. Os dados **In.Value** e **In.Status** são atribuídos a **SP.Value** e **SP.Status**. **SP.Value** e **SP.Status** são atribuídos a **Out.Value** e **Out.Status**. Uma consulta ao relógio do sistema é feita no momento do cálculo de **Out** para definir **Out.Timestamp**.

Quando o Modo é **Automático**, um valor adquirido de uma fonte externa, normalmente a IHM, é gravado em **SP.Value** em vez de ser propagado de **In**. Então, exatamente como descrito no parágrafo acima, o valor da IHM é atribuído a **Out.Value**. Uma consulta ao relógio do sistema é feita no momento do cálculo de **Out** para definir **Out.Timestamp**.

Quando o Modo é **Manual**, um valor adquirido de uma fonte externa, normalmente a IHM, é atribuído a **Out.Value** para transmissão ao atuador. Uma consulta ao relógio do sistema é feita no momento do cálculo de **Out** para definir **Out.Timestamp**.

A inicialização sem interrupções (bumpless) durante a transição do modo manual é feita fazendo com que **SP.Value** e **SP.Status** rastreiem o status do valor real do atuador que é adquirido por meio das entradas **ReadBackIn.ActualValue.Value** e **ReadBackIn.ActualValue.Status** pela lógica a seguir.

- O valor, o status e o registro de data/hora do atuador são adquiridos em **ReadBackIn.ActualValue.Value**, **ReadBackIn.ActualValue.Status** e **ReadBackIn.ActualValue.Timestamp**
 - Na operação normal, o valor de **ReadBackIn.SimulationState** é definido como falso e **ReadBackIn.ActualValue.Value**, **ReadBackIn.ActualValue.Status** e **ReadBackIn.ActualValue.Timestamp** são atribuídos a **ReadBackIn.Value**, **ReadBackIn.Status** e **ReadBackIn.Timestamp**.
- Se o valor de **ReadBackIn.SimulationState** for definido como verdadeiro, **ReadBackIn.SimulationValue**, **ReadBackIn.SimulationStatus** e **ReadBackIn.SimulationTimestamp** serão atribuídos a **ReadBackIn.Value**, **ReadBackIn.Status** e **ReadBackIn.Timestamp**.
- Quando **MDOOption UseOutForReadBackIn** é definido como falso, **ReadBackIn.Value** e **ReadBackIn.Status** são atribuídos a **PV.Value** e **PV.Status**.
 - Finalmente, se **MDOOption SpPvTrackInManual** estiver definido como verdadeiro, **PV.Value** e **PV.Status** serão atribuídos a **SP.Value** e **SP.Status**.

Quando o modo é **LocalOverride**, **SP.Value** e **SP.Status** são atribuídos de **In.Value** e **In.Status** ou de **FaultStateSP.Value** e **FaultStateSP.Status**. Os dados **FaultStateSP** são atribuídos aos dados do setpoint quando **Mode.Target** é **Cascade** e **MDOOption FaultStateToValue** é definido como verdadeiro.

Os valores de **BackCalOut.Value** e **BackCalOut.Status** são atribuídos de **SP.Value** e **SP.Status** ou de **PV.Value** e **PV.Status** dependendo do valor de **FunctionBlockOption UsePvForBackCalOut**.

Os valores de **SP.Value** e **SP.Status** dependem do Modo conforme descrito nos diversos parágrafos acima. Os valores de **PV.Value** e **PV.Status** são determinados pela lógica descrita no parágrafo Modo Manual acima. Quando **UsePvForBackCalOut** é definido como verdadeiro, **PV.Value** e **PV.Status** são atribuídos a **BackCalOut.Value** e **BackCalOut.Status**. Quando **UsePvForBackCalOut** é definido como falso, **SP.Value** e **SP.Status** são atribuídos a **BackCalOut.Value** e **BackCalOut.Status**. Uma chamada para o relógio do sistema no momento do cálculo do **BackCalOut** é usada para definir **BackCalOut.Timestamp**.

Modelo de Informação OCF

O **MultiStateDiscreteOutputBlockType** é um subtipo de **ControlFunctionBlockType** definido na Seção **ControlFunctionBlockType Definition**. Um bloco funcional de saída discreta multiestado O-PAS™ deve estar em conformidade com o modelo de informação OCF especificado na Figura 31, Tabela 50 e Tabela 51. Deverá suportar, no mínimo, os seguintes Modos:

- **OutOfService**
- **InitializationManual**

- LocalOverride
- Manual
- Automatic
- Cascade

Um bloco funcional de saída discreta multiestado O-PAS™ deve estar em conformidade com a interface de coordenação entre blocos especificada nas Figuras 32 e 33.

Nenhuma especificação é fornecida para acesso baseado em função a cada variável. O controle de acesso deverá ser implementado utilizando mecanismos do OCF (O-PAS™ Parte “O-PAS™ Connectivity Framework (OCF)”) de acordo com a especificação de Segurança do O-PAS™ (O-PAS™ Parte “Security”).

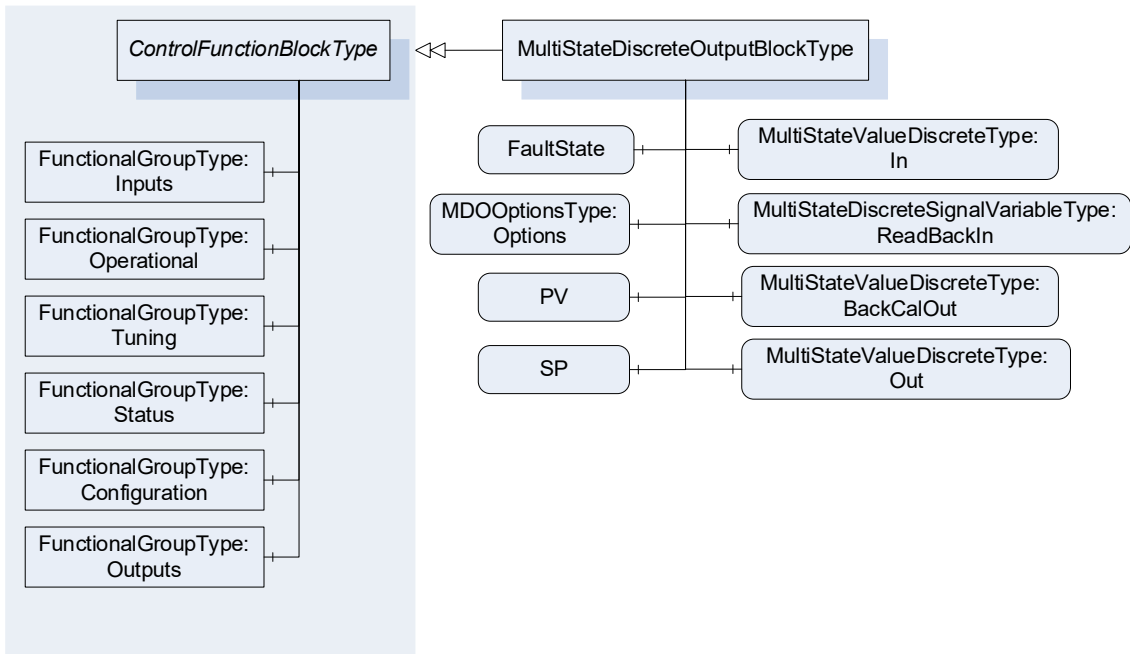


Figura 31 - Modelo de dados MultiStateDiscreteOutputBlockType

Attribute	Value				
BrowseName	MultiStateDiscreteOutputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da Seção “ControlFunctionBlockType Definition”.					
0:HasComponent	Variable	In	0:Number	0:MultiStateValueDiscreteType	Mandatory
0:HasComponent	Variable	ReadBackIn	0:Number	3:MultiStateDiscreteSignalVariableType	Mandatory
0:HasComponent	Variable	BackCalOut	0:Number	0:MultiStateValueDiscreteType	Mandatory
0:HasComponent	Variable	Out	0:Number	0:MultiStateValueDiscreteType	Mandatory

Attribute	Value				
BrowseName	MultiStateDiscreteOutputBlockType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de <i>ControlFunctionBlockType</i> da Seção “ControlFunctionBlockType Definition”.					
0:HasProperty	Variable	FaultStateSP	0:Number	FaultStateSPType	Mandatory
0:HasComponent	Object	Options		MDOOptionsType	Mandatory
0:HasComponent	Variable	PV	0:Number	0:MultiStateValueDiscreteType	Mandatory
0:HasComponent	Variable	SP	0:Number	0:MultiStateValueDiscreteType	Mandatory

Tabela 52 - Definição *MultiStateDiscreteOutputBlockType*

Source Path	Reference Type	Is Forward	Target Path
Inputs	0:Organizes	True	In
Inputs	0:Organizes	True	ReadBackIn ActualValue
Outputs	0:Organizes	True	BackCalOut
Outputs	0:Organizes	True	Out ActualValue
Configuration	0:Organizes	True	In EnumValues
Configuration	0:Organizes	True	Mode Normal
Configuration	0:Organizes	True	Mode Target EnumValues[]
Configuration	0:Organizes	True	Options
Configuration	0:Organizes	True	Out EnumValues
Tuning	0:Organizes	True	FaultStateSP
Tuning	0:Organizes	True	ReadBackIn SimulationState
Tuning	0:Organizes	True	ReadBackIn SimulationValue
Operational	0:Organizes	True	Mode Target

Source Path	Reference Type	Is Forward	Target Path
Status	0:Organizes	True	Error
Status	0:Organizes	True	Mode
Status	0:Organizes	True	Out
Status	0:Organizes	True	PV
Status	0:Organizes	True	SP

Tabela 53 - Referências Adicionais MultiStateDiscreteOutputBlockType

O **MDOOptionsType** é um subtipo de **FBOptionsType** definido no O-PAS™ Parte “Information and Exchange Models: Basic Configuration” com as opções aplicáveis obrigatórias. Está formalmente definido na Tabela 52.

Attribute	Value				
BrowseName	MDOOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Modeling Rule
Subtipo de 4:FBOptionsType definido em O-PAS™ Parte “Information and Exchange Models: Basic Configuration”					
0:HasProperty	Variable	4:FaultStateTo Value	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:PropagateFaultBackward	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInLO	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpPvTrackInManual	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:SpTrackRetainedTarget	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToManualIfFaultStateActive	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:TargetToNextPermittedModelIfBadCaseIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseFaultStateValueOnRestart	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UseOutForReadBackIn	0:Boolean	0:PropertyType	Mandatory
0:HasProperty	Variable	4:UsePvForBackCalcOut	0:Boolean	0:PropertyType	Mandatory

Attribute	Value				
BrowseName	MDOOptionsType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	Modeling Rule
0:HasProperty	Variable	4:UseUncertain AsGood	0:Boolean	0:PropertyType	Mandatory

Tabela 54 - Definição de MDOOptionsType

Os modos e a interface de coordenação entre blocos de um bloco funcional de saída discreta multiestados O-PASTM são especificados nas Figuras 32 e 33.

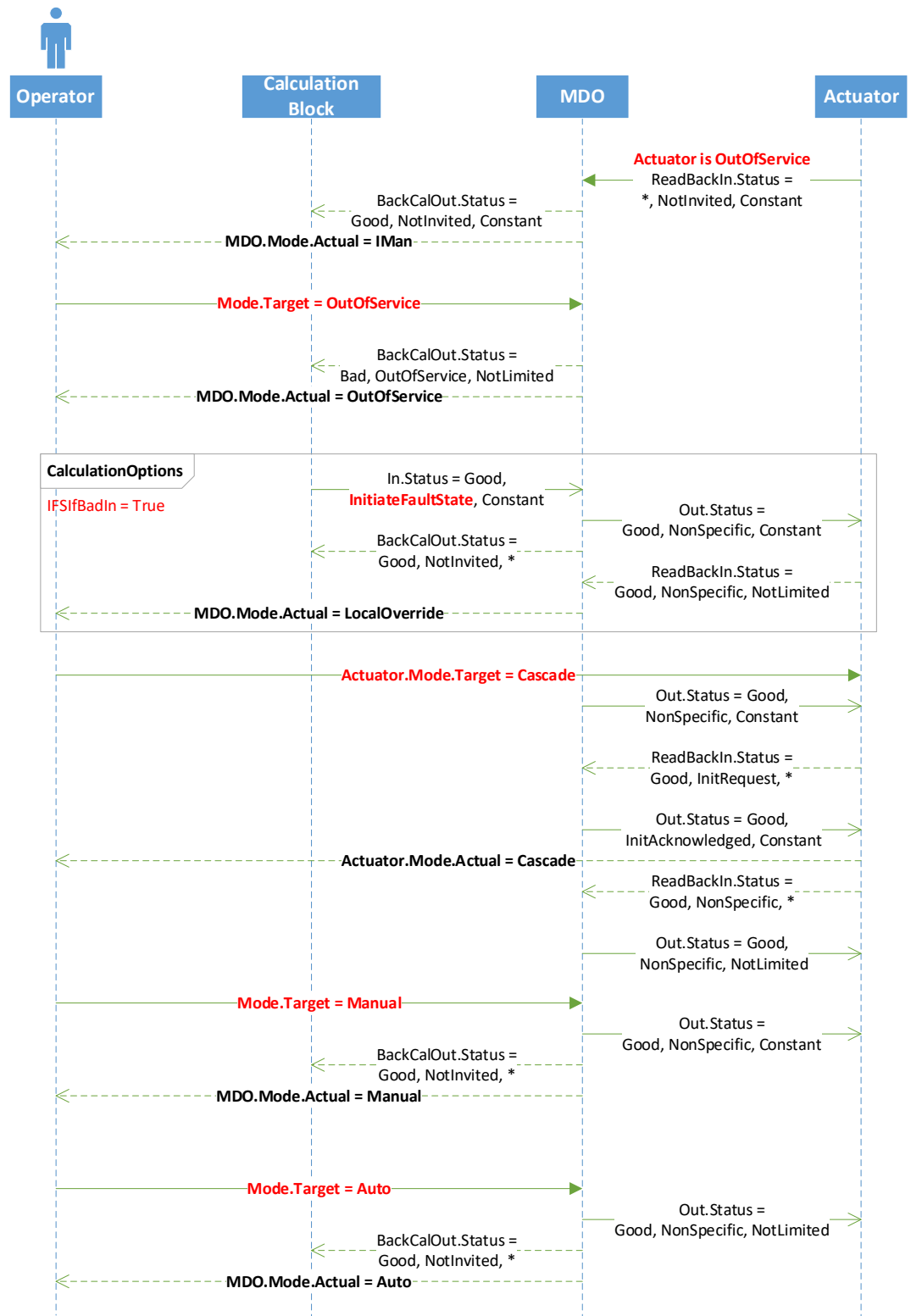


Figura 32 - Diagrama de sequência de modo de bloco funcional de saída discreta multiestados O-PAS™ e interface de coordenação entre blocos - Parte 1 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

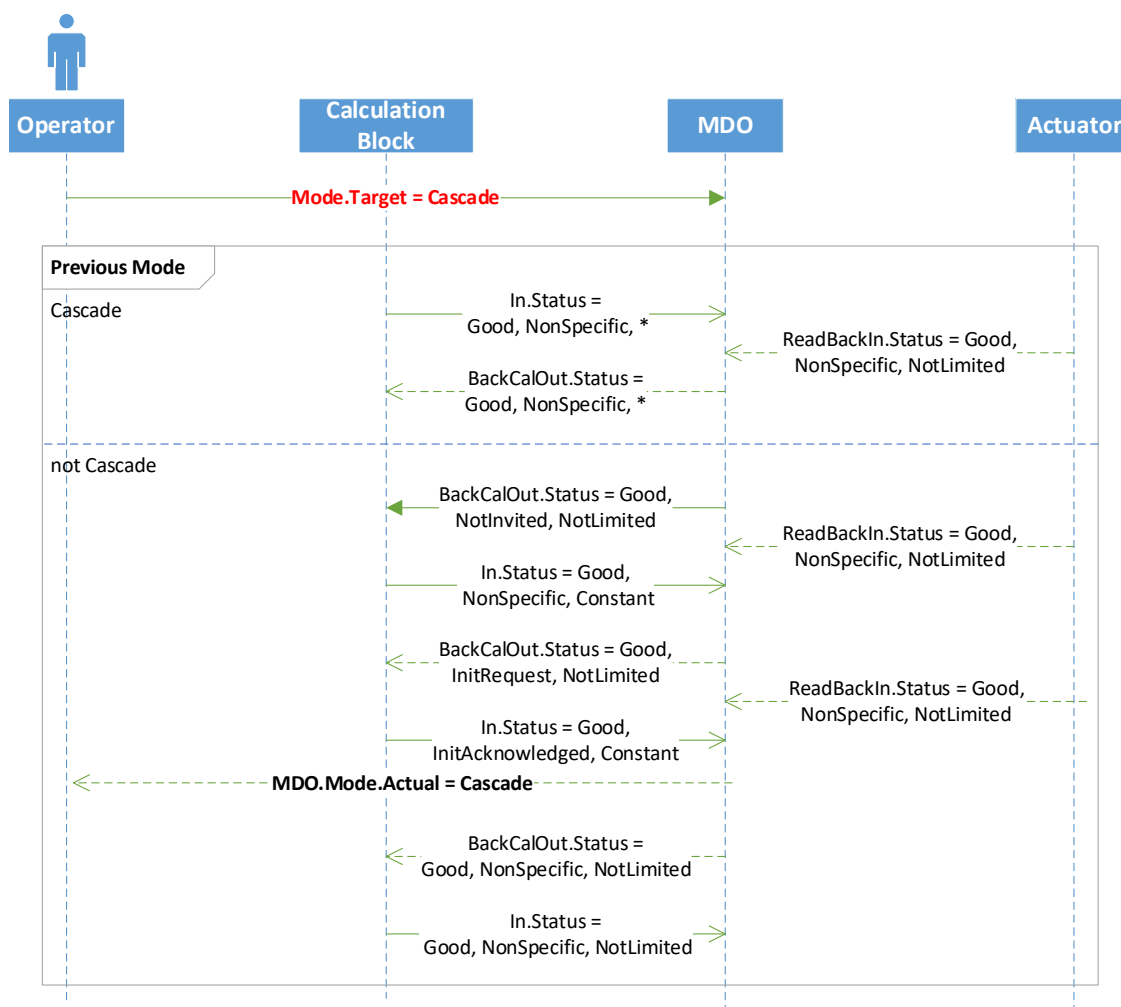


Figura 33 - Diagrama de sequência de modo de bloco funcional de saída discreta multiestados O-PAS™ e interface de coordenação entre blocos - Parte 2 de 2 (consulte a Seção “Syntax Notes for Interblock Coordination Interface UML Sequence Diagrams” para notas sobre sintaxe UML)

Implementação de Referência IEC 61131 – Variáveis e Código

O mapeamento entre o modelo de informações **MultiStateDiscreteOutputBlockType** OCF e as variáveis do programa na implementação de referência do bloco funcional de saída discreta multiestado O-PAS™ IEC 61131 é fornecido na Tabela 53.

As colunas 1 e 2 listam as variáveis do bloco funcional definidas no esquema lógico, no modelo de dados BlockType e na definição BlockType. As colunas 3 e 4 listam os nomes de variáveis e declarações de dados usados no código de implementação de referência. Além disso, os nomes das variáveis do código de implementação de referência foram selecionados para serem equivalentes aos blocos funcionais FOUNDATION Fieldbus para facilitar a compreensão do usuário. Finalmente, para vincular os seis atributos de cada variável de entrada e saída em conexões únicas “fixadas”, um tipo de dados de estrutura é usado, conforme indicado na Coluna 3.

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
In Value Status Timestamp EnumValues ValueAsText	Input Input Input Configuration Configuration	Structure CAS_IN_D .Value .Status .Timestamp XD_STATE	ULINT UDINT LDT ¹ ULINT STRING	EU of In na utcTime	Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador. Enumeração de valores de estado de entrada/saída.
ReadBackIn ActualValue Value Status Timestamp SimulationValue Value Status SimulationState	Input Input Input Tuning Tuning Tuning	Structure RDBK_IN_D .Value .Status .Timestamp SIMULATE_D .Value .Status .Enable	ULINT UDINT LDT ¹ ULINT UDINT BOOL	EU of Out na utcTime EU of Out na utcTime	Informações de feedback do atuador. Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação seguem o valor e o status reais do OSS (Fonte).
BackCalOut Value Status Timestamp	Output Output Output	Structure BKCAL_D .Value .Status .Timestamp	ULINT UDINT LDT ¹	EU of In na utcTime	
Out Value Status Timestamp EnumValues ValueAsText	Output Output Output Status Status	Structure OUT_D .Value .Status .Timestamp PV_STATE	ULINT UDINT LDT ¹ ULINT STRING	EU of Out na utcTime EU of Out	O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.
Error	Status	BLOCK_ERR	WORD	na	Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte “Information and Exchange Models: Alarm and Events Configuration”).

OPC UA Target Path (2)	Tipo de grupo funcional O-PAS™	Equivalente FOUNDATION Fieldbus	Tipo de dados da variável do programa	Unidades de Engenharia	Descrição
Mode Target EnumValues Normal	Status Operational Configuration Configuration	MODE_BLK .Actual .Target .Permitted .Normal	DINT DINT DINT DINT	na na na na	Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal.
FaultStateSP Time Value	Tuning Tuning	FSTATE_TIME FSTATE_VAL_D	TIME ULINT	Sec EU of In	Tempo em segundos até o tempo limite em CAS_IN. Valor a ser usado quando FaultStateSP for verdadeiro.
Options	Configuration	IO_OPTS_DO STATUS_OPTS_D O SHED_OPT	BOOL	Na	Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
PV Value Status EnumValue ValueAsText	Status Status Status Status	PV_D .Value .Status	ULINT UDINT ULINT STRING	EU of In na	Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.
SP Value Status EnumValue ValueAsText	Status Status Status Status	SP_D .Value .Status	ULINT UDINT ULINT STRING	EU of Out na	

Tabela 55 - Dicionário de dados do bloco funcional de saída discreta multiestado O-PAS™

(1) DATA_AND_TIME

(2) Na coluna OPC UA Target Path, os valores VST são listados como variáveis escalares para corresponder aos nomes das variáveis do programa O-PAS™. O OPC UA transmite qualquer variável Value como uma estrutura contendo três atributos; ou seja, (valor, status, registro de data/hora).

Descrição dos Parâmetros do Bloco

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
BACKCALOUT Value	ULINT		EU de In		A saída BackCalOut é calculada para transmissão no caminho reverso de fluxo de dados para um bloco funcional upstream com a finalidade de inicialização sem interrupções (bumpless) do referido bloco funcional upstream.
ERROR	WORD		NA		Sequência de bits na qual vários erros do bloco funcional podem ser transmitidos conforme calculado com base nas condições físicas ou cibernéticas do bloco funcional. A provisão para alarme/alerta em BLOCK_ERR é fornecida no sistema de alarme O-PAS™ (O-PAS™ Parte "Information and Exchange Models: Alarm and Events Configuration").
FAULTSTATESP Time	TIME		segundos		Tempo em segundos até o tempo limite em CAS_IN. Valor a ser usado quando FaultStateSP for verdadeiro.
IN Value	ULINT		NA		Valor de entrada do bloco funcional, incluindo status e registro de data/hora, do setpoint fornecido por um bloco funcional de classe de controlador. Enumeração de valores de estado de entrada/saída.
MODE Normal RetainedTarget Target	DINT	OutOfService InitializationManual LocalOverride Manual Automatic Cascade	NA		Objeto no qual os modos do bloco funcional estão contidos, incluindo Target, Actual, Permitted e Normal. RetainedTarget é uma memória do último modo válido que o bloco tinha antes de ir para uma condição especial (como OOS, falha ou perda de comando).
OPTIONS	BOOL	FaultStateToValue PropagateFaultBackward SpPvTrackInLO SpPvTrackInManual SpPvTrackInRetainedTarget TargetToManualIfFaultStateActive TargetToNextPermittedModelIfBadCasIn UseFaultStateValueOnRestart UseOutForReadBackIn UsePvForBackCalOut UseUncertainAsGood	True ou False		Opções selecionáveis pelo usuário para alterar a ação tomada na entrada ou saída do bloco funcional ou no estado do bloco funcional.
OUT Value	ULINT		EU de Out		O valor de saída (transmitido), incluindo status associado e registro de data/hora, do valor analógico primário conforme calculado durante a execução do bloco funcional.

Parâmetro O-PAS™	Tipo de dados	Faixa Válida Opções	Unid.	Memória/ Modo	Descrição
PV Value	ULINT		EU de In		Um valor interno (parâmetro contido), incluindo o status associado, do valor analógico primário (ou valor associado), conforme calculado durante a execução do bloco funcional.
READBACKIN SimulationState SimulationValue	BOOL ULINT		True ou False EU of Out		Informações de feedback do atuador. Permite que a entrada OSS (fonte) (IN_D) ou a saída do bloco funcional (OUT_D), incluindo qualquer um dos respectivos status, sejam fornecidas manualmente quando a simulação estiver habilitada. Quando a simulação está desabilitada, o valor e o status da simulação seguem o valor e o status reais do OSS (Fonte).
SP LimitHigh LimitLow RateDown RateUp ValuePrecision			EU de Out NA		O valor, incluindo o status associado, do valor desejado da variável de processo controlada conforme escrito pelo operador. Limites do valor do setpoint mais alto e mais baixo que um operador pode inserir. Inclinação, em unidades de PV por segundo, na qual uma alteração do setpoint é aplicada. Nos blocos funcionais da classe de controle, as taxas de rampa se aplicam enquanto o bloco funcional está no modo automático. Se SP_RATE_x = 0, o setpoint é aplicado diretamente. ValuePrecision é um campo para configurar a quantidade de casas decimais para visualização em algum cliente OPC.

Tabela 56 – Descrição dos parâmetros do bloco MDO

Legenda:

NA – Parâmetro Adimensional

RO – Somente Leitura

D – dinâmico

N – não volátil

S – estático

USINT: Um valor inteiro sem sinal de 8 bits (0 a 255);

UDINT: Um valor inteiro sem sinal de 32 bits (0 a 4.294.967.295);

ULINT: Um valor inteiro sem sinal de 64 bits

DINT: Um valor inteiro com sinal de 32 bits (-2.147.483.648 para 2.147.483.647);

BOOL: Um valor booleano (TRUE ou FALSE);

WORD: Um valor inteiro sem sinal de 16 bits (0 a 65.535);

