

**KOLLMORGEN**®

---

*Because Motion Matters™*

# **CNC AKC-PPC**

**Manual de Programação do PLC**

SOUZA, RENATO



## Contents

<b>1.0 Introdução</b>	<b>8</b>
<b>2.0 Linguagem ST</b>	<b>9</b>
<b>2.1 Fluxo do Programa</b>	<b>9</b>
<b>2.2 Tipos de Dados</b>	<b>10</b>
<b>2.3 Operadores e Expressões</b>	<b>11</b>
2.3.1 Operadores Aritméticos	11
2.3.2 Operadores Relacionais	11
2.3.3 Operadores Lógicos	12
2.3.4 Operador de Atribuição	12
2.3.5 Estruturas de controles condicionais e repetições	13
2.3.5.1 IF	13
2.3.5.2 CASE	14
2.3.5.3 WHILE	15
2.3.5.4 FOR	15
2.3.6 Expressões	16
2.3.6.1 Expressões lógicas	16
<b>3.0 Variáveis de Sistema</b>	<b>17</b>
<b>3.1 Geral</b>	<b>17</b>
3.1.1 svMemFirstCycle	17
3.1.2 svOnInitialization	17
3.1.3 svBlockInitialization	17
3.1.4 svSoftkeyBlockInitialization	17
3.1.5 svCncInitialized	17
3.1.6 svEmergencyCnc	18
3.1.7 svEmergencyOutput	18
3.1.8 svEmergencyState	18
3.1.9 svInitializationMode	18
3.1.10 svNewPLC	18
3.1.11 svParametersUpdated	18
3.1.12 svUserLevel	19
3.1.13 svVoid	19
3.1.14 svTransition500ms	19
3.1.15 svBlink500ms	19
3.1.16 svIntDraft1	19
3.1.17 svIntDraft2	19
3.1.18 svRealDraft1	20
3.1.19 svRealDraft2	20
3.1.20 svSaveMemory	20
3.1.21 svAxisInHoming	20
3.1.22 svAxisLagError	20
3.1.23 svClearLagError	20
3.1.24 svFeedholdCnc	20
3.1.25 svReferenceMode	21
3.1.26 svSelectOrigin	21
<b>3.2 Comandos de Execução de Programas ISO</b>	<b>22</b>
3.2.1 svStartCnc	22

3.2.2 svStopCnc	22
3.2.3 svOnKeyStart	22
3.2.4 svOnKeyStop	22
3.2.5 svBlockStart	22
3.2.6 svAbortCnc	22
<b>3.3 Execução de Programas ISO</b>	<b>23</b>
3.3.1 svCncInElementSimulation	23
3.3.2 svCncInSimulatedMode	23
3.3.3 svExecutionError	23
3.3.4 svExecutionHold	23
3.3.5 svExecutionInEmergency	23
3.3.6 svExecutionInInitialState	23
3.3.7 svExecutionInWaitState	24
3.3.8 svExecutionPause	24
3.3.9 svExecutionRun	24
3.3.10 svNewCodeM	24
3.3.11 svNewCodeS	24
3.3.12 svNewCodeT	24
3.3.13 svAnyFunctionMST	25
<b>3.4 Seleção de Programas ISO</b>	<b>26</b>
3.4.1 svSelectDirNumber	26
3.4.2 svSelectProgramNumber	26
3.4.3 svKeySelectProgram	26
3.4.4 svKeyPgmCall	26
<b>3.5 Potenciômetros</b>	<b>27</b>
3.5.1 svPotOverridePlcControl	27
3.5.2 svPotOverride	27
3.5.3 svPotSpindle	27
<b>3.6 Erros</b>	<b>28</b>
3.6.1 svCncError	28
3.6.2 svStoFault	28
3.6.3 svExternalClear	28
<b>3.7 Códigos de Tecla</b>	<b>29</b>
3.7.1 svNewCncKey	29
3.7.2 svRequestKeyPlc	29
3.7.3 svSendKeyToCnc	29
3.7.4 svLastCncKey	29
3.7.5 svOnKeyCE	29
<b>3.8 Telas</b>	<b>30</b>
3.8.1 svCurrentScreen	30
3.8.2 svLastScreen	30
<b>3.9 Árvores de Softkeys</b>	<b>31</b>
3.9.1 svCurrentHorizontalState	31
3.9.2 svCurrentVerticalState	31
3.9.3 svCncNewSoftkeyState	31
3.9.4 svCncNewHorizontalState	31
3.9.5 svPlcNewHorizontalState	31
3.9.6 svCncNewVerticalState	31
3.9.7 svPlcNewVerticalState	32

3.9.8 svRequestNewSoftkeyState	32
<b>4.0 Funções Nativas</b>	<b>33</b>
<b>4.1 Mensagens</b>	<b>34</b>
4.1.1 Alarme - ALARM	34
4.1.2 Piscante - BLINK_MESSAGE	34
4.1.3 Normal - MESSAGE	34
4.1.4 Temporizada - TIMED_MESSAGE	35
<b>4.2 Transição</b>	<b>36</b>
4.2.1 Transição de Subida - R_EDGE	36
4.2.2 Transição de Descida - F_EDGE	36
4.2.3 Qualquer Transição - RF_EDGE	36
<b>4.3 Parâmetros, Variáveis H e Variáveis Reservadas</b>	<b>37</b>
4.3.1 Ler Parâmetro - R_PARAM	37
4.3.2 Escrever no Parâmetro - W_PARAM	37
4.3.3 Ler Registrador H - R_HVAR	37
4.3.4 Escrever no Registrador H - W_HVAR	38
4.3.5 Ler Variável Reservada - R_COMMAND_VAR	38
4.3.6 Escrever na Variável Reservada - W_COMMAND_VAR	38
<b>4.4 Trigonômicas/Matemáticas</b>	<b>39</b>
4.4.1 Seno - SIN	39
4.4.2 Cosseno - COS	39
4.4.4 Tangente - TAN	39
4.4.5 Arco Tangente - ATAN	39
4.4.6 Valor da Constante Pi - PI	40
4.4.7 Valor Inteiro - TRUNC	40
4.4.8 Valor Fracionário - FRAC	40
4.4.9 Raiz Quadrada - SQRT	40
4.4.10 Valor Absoluto - ABS	41
<b>4.5 Conversões de tipos de dados</b>	<b>42</b>
4.5.1 Booleano para Real - BOOL_TO_REAL	42
4.5.6 Booleano para Inteiro - BOOL_TO_INT	42
4.5.2 Real para Booleano - REAL_TO_BOOL	42
4.5.3 Real para Inteiro - REAL_TO_INT	42
4.5.4 Real para Inteiro sem sinal - REAL_TO_UINT	43
4.5.5 Inteiro sem sinal para Inteiro - UINT_TO_INT	43
4.5.6 Inteiro para Inteiro sem sinal - INT_TO_UINT	43
4.5.7 Inteiro para Booleano - INT_TO_BOOL	43
4.5.8 Inteiro para Real - INT_TO_REAL	44
<b>5.0 Funções de Sistema</b>	<b>45</b>
<b>5.1 Mensagens</b>	<b>46</b>
5.1.1 Alarme - sfShowAlarmMessage	46
5.1.2 Piscante - sfShowBlinkingMessage	46
5.1.3 Normal - sfShowNormalMessage	47
5.1.4 Temporizada - sfShowTimedMessage	47
<b>5.2 Códigos e Navegação</b>	<b>48</b>
5.2.1 Identificar Código Recebido pelo CNC - sfKeyCode	48
5.2.2 Pegar Árvore Horizontal Seleccionada - sfCurrentHorizontalSFKState	48
5.2.3 Pegar Árvore Vertical Seleccionada - sfCurrentVerticalSoftkeyState	48

5.2.4 Verifica Tela Seleccionada - sfCurrentScreen	49
5.2.5 Bloqueio de Softkey - sfBlockSoftkey	49
5.2.6 Mudar Estado da Softkey - sfChangeSoftkey	49
<b>5.3 Execução</b>	<b>50</b>
5.3.1 sfFunctionM	50
5.3.2 sfFunctionS	50
5.3.3 sfFunctionT	51
<b>5.4 Transição</b>	<b>52</b>
5.4.1 Transição de Subida - sfRisingEdge	52
5.4.2 Transição de Descida - sfFallingEdge	52
5.4.3 Qualquer Transição - sfCheckingEdge	52
<b>5.5 Parâmetros, Variáveis H e Variáveis Reservadas</b>	<b>53</b>
5.5.1 Ler Parâmetro - sfReadParameter	53
5.5.2 Escrever no Parâmetro - sfWriteParameter	53
5.5.3 Ler Registrador H - sfReadHRegister	54
5.5.4 Escrever no Registrador H - sfWriteHRegister	54
5.5.5 Ler Variável Reservada - sfReadReservedVar	54
5.5.6 Escrever na Variável Reservada - sfWriteReservedVar	55
<b>5.6 Trigonométricas/Matemáticas</b>	<b>56</b>
5.6.1 Seno - sfMathSin	56
5.6.2 Cosseno - sfMathCos	56
5.6.4 Tangente - sfMathTan	56
5.6.5 Arco Tangente - sfMathAtan	56
5.6.6 Valor da Constante Pi - sfMathPi	57
5.6.7 Valor Inteiro - sfMathTrunc	57
5.6.8 Valor Fracionário - sfMathFrac	57
5.6.9 Raiz Quadrada - sfMathSqrt	57
5.6.10 Valor Absoluto - sfMathAbs	58
<b>5.7 Conversões de tipos de dados</b>	<b>59</b>
5.7.1 Booleano para Real - sfConvertBoolToReal	59
5.7.6 Booleano para Inteiro - sfConvertBoolToInt	59
5.7.2 Real para Booleano - sfConvertRealToBool	59
5.7.3 Real para Inteiro - sfConvertRealToInt	59
5.7.4 Real para Inteiro sem sinal - sfConvertRealToUint	60
5.7.5 Inteiro sem sinal para Inteiro - sfConvertUintToInt	60
5.7.6 Inteiro para Inteiro sem sinal - sfConvertIntToUint	60
5.7.7 Inteiro para Booleano - sfConvertIntToBool	60
5.7.8 Inteiro para Real - sfConvertIntToReal	61
<b>5.8 Eixos</b>	<b>62</b>
5.8.1 Eixo Habilitado - sfAxisEnabled	62
5.8.2 Eixo Habilitado para Movimento Manual - sfAxisEnabledForManualMove	62
5.8.3 Eixo em Movimento - sfAxisInMove	62
5.8.4 Eixo Definido - sfAxisIsDefined	62
5.8.5 Eixo é Spindle - sfAxisIsSpindle	63
5.8.6 Eixo está Referenciado - sfAxisReferenced	63
5.8.7 Executar Preset - sfPresetAxis	63
<b>6.0 Blocos de Sistema</b>	<b>64</b>
<b>6.1 Eixos</b>	<b>64</b>

6.1.1 Referenciamento - sbHoming	64
6.1.2 Fim de Curso - sbLimitSwitch	66
6.1.3 JOG - sbJog	67
6.1.4 Movimento Absoluto - sbMoveAbsolute	68
6.1.5 Movimento Incremental - sbMoveRelative	70
6.1.6 Manivela Remota - sbRemoteHandwheel	72
6.1.7 Acoplar Eixo - sbAxisGear	74
<b>6.2 Temporizadores</b>	<b>76</b>
6.2.1 Temporizador (Normalmente Desligado) - sbTimerOFF	76
6.2.2 Temporizador (Normalmente Ligado) - sbTimerON	77
6.2.3 Temporizador (Piscante) - sbTimerONOFF	78
<b>6.3 Contadores</b>	<b>79</b>
6.3.1 Contador Crescente - sbCounterUp	79
6.3.2 Contador Decrescente - sbCounterDown	80
6.3.3 Contador de Tempo - sbCounterTime	81
<b>6.4 Códigos e Softkeys</b>	<b>82</b>
6.4.1 Enviar Código - sbSendKey	82
6.4.2 Bloquear Código - sbBlockKeys	83
6.4.3 Enviar Estado de Softkeys - sbSendState	84
<b>6.5 Drives</b>	<b>85</b>
6.5.1 Ler Dados do Drive - sbReadDataDrive	85
6.5.2 Escrever Dados do Drive - sbWriteDataDrive	86
<b>6.6 Acesso a Tabelas</b>	<b>87</b>
6.6.1 Escrever no Campo da Tabela - sbWriteTable	87
6.6.2 Ler Campo da Tabela - ReadTable	88
<b>6.7 Programas CNC</b>	<b>89</b>
6.7.1 Controle de Programa do CNC - sbManageCncProgram	89
<b>6.8 Cloud</b>	<b>90</b>
6.8.1 Ler Dados da Nuvem - sbReadCloud	90
6.8.2 Escrever Dados na Nuvem - sbWriteCloud	91
<b>6.9 THC</b>	<b>92</b>
6.9.1 Controle de Altura - sbHeightControl	92
6.9.2 Toque Ôhmico - sbTouchTorch	95
<b>6.10 Controle</b>	<b>97</b>
6.10.1 Filtro Digital - sbDigital Filter	97
6.10.2 PWM - sbPWM	98
6.10.1 Ler Erros Logados - sbReadErrorLog	99
<b>6.11 Came</b>	<b>100</b>
6.11.1 Came Virtual - sbVirtualCamAxis	100
6.11.2 Came Virtual Contínuo - sbVirtualCamContinuousMove	101
6.11.3 Came Por Posição - sbVirtualCamPosition	102
6.11.4 Preset Came Virtual - sbVirtualCamPreset	103
6.11.5 Came Eixo Real - sbRealCamAxis	104
6.11.6 Preset Came Eixo Real - sbRealCamPreset	105
6.11.7 Acoplar Came Real - sbRealCamClutch	106
6.11.8 Desacoplar Came Real - sbRealCamDeclutch	107
6.11.9 Came Digital - sbDigitalCam	108

6.11.10 Acoplar Came Digital - sbDigitalCamClutch	110
6.11.11 Desacoplar Came Digital - sbDigitalCamDeclutch	111
6.11.12 Saída Came Digital - sbDigitalCamOutput	112
6.11.13 Offset Came Digital Real – sbDigitalCamRealOffset	113
6.11.14 Offset Came Digital Virtual – sbDigitalCamVirtualOffset	114
6.11.15 Came de Eixo - sbAxisCam	115
6.11.16 Acoplar Came de Eixo - sbAxisCamClutch	117
6.11.17 Desacoplar Came de Eixo - sbAxisCamDeclutch	118
6.11.18 Controle Came de Eixo - sbAxisCamControl	119
<b>6.12 Packaging</b>	<b>120</b>
6.12.1 Leitura de Entrada Rápida - TouchProbeS1	120
6.12.2 Registro de Marcas - PlasticBagRegist1	121
<b>6.13 Temperatura</b>	<b>123</b>
6.13.1 Ler Temperatura - ReadTemperature	123
<b>6.14 Spindle</b>	<b>124</b>
6.14.1 Eixo Árvore Geral - sbSpindle	124
6.14.2 Jog de Eixo Árvore - sbSpindleJog	126
6.14.3 Posicionamento do Eixo Árvore - sbSpindlePosition	127
6.14.4 Eixo Árvore Único - sbSpindleSingle	128
6.14.4 Eixo Árvore Simples - sbSpindleSimple	130
<b>7.0 Exemplo de PLC</b>	<b>132</b>



### 1.0 Introdução

Para construir o programa de PLC no CNC AKC-PPC utilizaremos o editor de PLC do AKC-PPC IDE.

A linguagem de programação utilizada para isso é o ST que é definida na norma IEC61131-3

Há muitos livros e treinamentos sobre a Programação de PLC definidas na norma IEC61131-3.

O objetivo desta documentação não é reproduzir o que já está amplamente definido na norma e em outras publicações, mas transmitir uma compreensão básica da linguagem de texto estruturado no ambiente do IDE ACK-PPC e do CNC AKC-PPC.

## 2.0 Linguagem ST

O texto estruturado é a linguagem de programação de PLC definida pelo “PLCOpen” na norma IEC 61131-3.

A linguagem de programação é baseada em texto, diferentemente do “Ladder” que é gráfica.

Pode parecer melhor usar uma linguagem de programação gráfica para programação de PLC, mas dependendo do tamanho do programa que será desenvolvido isso pode ser exatamente o contrário.

Para quem já programou em linguagens de alto nível o texto estruturado parecerá familiar.

A sintaxe do texto estruturado foi desenvolvida para se parecer com a sintaxe de uma linguagem de programação de alto nível com loops, variáveis, condições e operadores similares.

## 2.1 Fluxo do Programa

A execução de programa do PLC tem uma peculiaridade bem definida.

O programa do PLC é executado como se fosse um programa de loop infinito, ou seja, ele é executado uma vez e ao final da execução ele retorna para o início e faz isso novamente a cada vez.

Enquanto o CNC estiver ligado o PLC estará ativo e sempre executando.

No CNC temos um parâmetro para determinar o tempo de ciclo do programa do PLC.

Ele é o parâmetro 100 e está localizado na aba “Geral” do editor de parâmetros.

Neste parâmetro podemos escolher os seguintes tempos de execução:

0 = 16ms

1 = 4ms

2 = 8ms

Caso ao executar o programa do PLC o CNC perceber que o tempo fornecido não está suficiente uma mensagem será exibida para identificar isto.

Essa questão pode acontecer devido ao tamanho do programa desenvolvido, ou por alguma lógica errada.

Isso pode ser resolvido aumentando o tempo de execução do PLC ou verificando e otimizando o programa do PLC.

## 2.2 Tipos de Dados

Ao adicionar as variáveis no programa do PLC é necessário informar o tipo de dado da mesma.

O CNC suporta os seguintes tipos, descritos na tabela abaixo.

Tipo	Descrição	Valores Possíveis
<b>BOOL</b>	Booleano (bit)	FALSE ou TRUE
<b>SINT</b>	Inteiro de 8 bits com sinal	-128 a +127
<b>USINT</b>	Inteiro de 8 bits sem sinal	0 a 255
<b>INT</b>	Inteiro de 16 bits com sinal	-32768 a + 32767
<b>UINT</b>	Inteiro de 16 bits sem sinal	0 a +65535
<b>DINT</b>	Inteiro de 32 bits com sinal	-2147483648 a +2147483647
<b>UDINT</b>	Inteiro de 32 bits sem sinal	0 a +4294967295
<b>LINT</b>	Inteiro de 64 bits com sinal	$-(2^{63})$ a $+(2^{63}) - 1$
<b>ULINT</b>	Inteiro de 64 bits sem sinal	0 a $+(2^{64}) - 1$
<b>REAL</b>	Real de ponto flutuante de 32 bits	-3.4E38 a 3.4E38 e -3.4E-38 a 3.4E-38 (6 a 7 dígitos de precisão)
<b>ARRAY</b>	Vetor de valores INT ou REAL	De acordo com o tipo utilizado

Para adicionar, editar ou excluir variáveis consulte o Manual do usuário do AKC-PPC IDE e verifique a seção do editor de PLC.

## 2.3 Operadores e Expressões

Os operadores são comandos que aplicamos as expressões que escrevemos em nossos programas para executar uma determinada ação.

A seguir descreveremos os operadores e expressões e como devem ser utilizadas na programação do PLC para o CNC AKC-PPC.

### 2.3.1 Operadores Aritméticos

Os operadores aritméticos realizam operações matemáticas e retornam um valor como resultado. Caso seja necessário escrever expressões maiores ou mais complexas, podemos combinar esses operadores e criá-las. Isso nos permite executar todo tipo de cálculo que seja necessário.

Na tabela a seguir, vemos todos os operadores aritméticos disponíveis para a construção do programa do PLC.

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
**	Potenciação (Utilizado somente em números do tipo REAL)
<b>SQRT()</b>	Raiz Quadrada (Utilizado somente em números do tipo REAL)
<b>ABS()</b>	Valor Absoluto (Utilizado somente em números do tipo REAL)
<b>SIN()</b>	Seno (Utilizado somente em números do tipo REAL)
<b>COS()</b>	Cosseno (Utilizado somente em números do tipo REAL)
<b>TAN()</b>	Tangente (Utilizado somente em números do tipo REAL)
<b>ATAN()</b>	Arco Tangente (Utilizado somente em números do tipo REAL)

### 2.3.2 Operadores Relacionais

Todos os operandos relacionais analisam 2 operandos e retornam o tipo BOOL, independentemente do tipo de comparação que está sendo efetuada.

Os operadores relacionais sempre avaliam dois operandos do mesmo tipo.

Na tabela a seguir, vemos todos os operadores relacionais disponíveis para a construção do programa do PLC.

Operador	Descrição
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a
=	Igual
<>	Diferente

### 2.3.3 Operadores Lógicos

Os operadores lógicos nos permitem criar expressões lógicas maiores a partir da junção de duas ou mais expressões. Todos os operandos lógicos retornam o mesmo tipo em uma expressão. Por este motivo, em expressões com estes operadores não se deve utilizar variáveis de diferentes tipos.

Na tabela a seguir, vemos todos os operadores lógicos disponíveis para a construção do programa do PLC.

Operador	Descrição
<b>AND</b>	E Booleano
<b>OR</b>	OU Booleano
<b>XOR</b>	OU Exclusivo
<b>NOT</b>	Não

### 2.3.4 Operador de Atribuição

O operador de atribuição é utilizado para definir o valor uma variável.

Na tabela a seguir, vemos o operador de atribuição utilizado para a construção do programa do PLC.

Operador	Descrição
<b>:=</b>	Atribuição

Em seu uso, o operando à esquerda representa a variável para a qual desejamos atribuir o valor informado na expressão (ou variável) à direita.

### 2.3.5 Estruturas de controles condicionais e repetições

As estruturas de controles condicionais e repetições nos permitem tomar decisões importantes na execução do programa de PLC de acordo com uma expressão determinada.

#### 2.3.5.1 IF

IF é uma estrutura de controle condicional que executa as ações determinadas se a expressão condicional for satisfeita.

É possível também encadear com o comando ELSIF e o ELSE.

Sintaxes da estrutura de controle IF:

##### Somente IF

```
IF (Expressão 1) THEN
  Ações a serem executadas se a "Expressão 1" for satisfeita
END_IF;
```

##### IF em conjunto com ELSE

```
IF (Expressão 1) THEN
  Ações a serem executadas se a "Expressão 1" for satisfeita
ELSE
  Ações a serem executadas se a "Expressão 1" não for satisfeita
END_IF;
```

##### IF em conjunto com ELSIF e ELSE

```
IF (Expressão 1) THEN
  Ações a serem executadas se a "Expressão 1" for satisfeita
ELSIF (Expressão 2) THEN
  Ações a serem executadas se a "Expressão 1" não for satisfeita, mas a "Expressão 2"
  for satisfeita
ELSE
  Ações a serem executadas se a "Expressão 1" e a "Expressão 2" não forem satisfeitas
END_IF;
```

Exemplo de utilização:

```
IF var > 10 THEN
  valor := 1000;
  ligaLuzVerde := TRUE;
  ligaLuzAmarela := FALSE;
  ligaLuzVermelha := FALSE;
ELSIF (var > 0) AND (var <= 10) THEN
  valor := 100;
  ligaLuzVerde := FALSE;
  ligaLuzAmarela := TRUE;
  ligaLuzVermelha := FALSE;
ELSE
  valor := 1;
  ligaLuzVerde := FALSE;
  ligaLuzAmarela := FALSE;
  ligaLuzVermelha := TRUE;
END_IF;
```

### 2.3.5.2 CASE

CASE é uma estrutura de controle condicional que executa apenas um bloco de ações.

A validação do CASE é feita comparando o valor da variável de controle com o valor dos índices registrados.

Após a comparação será executado o índice que tem o valor igual ao da variável de controle.

Abaixo temos o formato de como se utiliza esta estrutura de controle.

Não é possível utilizar uma variável de controle do tipo REAL no CASE, apenas variáveis do tipo INT.

Sintaxes da estrutura de controle CASE:

```
CASE variável de controle OF
0:
    Ações a serem executadas se a "variável de controle" for igual a 0
1:
    Ações a serem executadas se a "variável de controle" for igual a 1
2:
    Ações a serem executadas se a "variável de controle" for igual a 2
.
.
.
N:
    Ações a serem executadas se a "variável de controle" for igual a N
END_CASE;
```

Exemplo de utilização:

```
CASE varControle OF
0:
    valor := var * varControle;
    ligaLuzVerde := TRUE;
    ligaLuzAmarela := FALSE;
    ligaLuzVermelha := FALSE;
1:
    valor := var * varControle;
    ligaLuzVerde := FALSE;
    ligaLuzAmarela := TRUE;
    ligaLuzVermelha := FALSE;
2:
    valor := var * varControle;
    ligaLuzVerde := FALSE;
    ligaLuzAmarela := FALSE;
    ligaLuzVermelha := TRUE;
END_CASE;
```

### 2.3.5.3 WHILE

WHILE é uma estrutura de repetição que executa um bloco de ações enquanto a condição de validação for verdadeira. A estrutura sempre testa a validação antes de executar as ações.

Sintaxes da estrutura de controle WHILE:

```
WHILE (Expressão 1) DO
    Ações a serem executadas enquanto a "Expressão 1" for satisfeita
END_WHILE;
```

Exemplo de utilização:

```
WHILE (varTemp > varTarget) DO
    varTemp := varTemp / 2;
    varCount := varCount + 1;
END_WHILE;
```

Deve-se tomar bastante cuidado com a utilização dessa estrutura, uma vez que dependendo da quantidade de repetições o PLC.

### 2.3.5.4 FOR

FOR é uma estrutura de repetição que executa um bloco de ações enquanto a condição de validação for verdadeira. A estrutura sempre testa a validação antes de executar as ações.

A condição de validação nesta estrutura é controlada pela própria estrutura

Ele utiliza uma variável para controlar a contagem e o incremento.

Sintaxes da estrutura de controle FOR:

```
FOR Index := Inicial TO Final BY Incremento DO
    Ações a serem executadas enquanto o valor do "Index" for menor que o valor do "Final"
END_FOR;
```

Sempre ao entrar nesta estrutura o "Index" irá assumir o valor "Inicial" e a cada repetição o "Index" será incrementado de acordo com o "Incremento" utilizado. No início de cada repetição é feita uma verificação se o valor do "Index" atingiu o valor "Final", se isso ocorreu é finalizada as repetições desta estrutura.

Exemplo de utilização:

```
FOR Index := 0 TO 9 BY 1 DO
    varCount := varCount + 1;
END_FOR;
```



### 2.3.6 Expressões

Expressões são definidas como um conjunto de funções e variáveis que produzem um resultado.

A variável que receberá o valor da expressão deve ser o mesmo tipo das variáveis e funções utilizadas na expressão. Portanto, não é possível somar duas variáveis do tipo INT e armazenar em uma variável do tipo REAL ou vice-versa. Se for necessário esse tipo de ação, as variáveis necessitarão de conversão para o tipo utilizado.

Exemplo de utilização:

```
varValor := varRep + 1;  
varValor := COS( 30 ) * varComprimento;
```

#### 2.3.6.1 Expressões lógicas

Expressões booleanas funcionam comparando operadores e gerando um resultado booleano (TRUE ou FALSE).

Importante lembrar que as atribuições e retornos do tipo booleano são TRUE e FALSE, não é permitido atribuir "1" a uma variável booleana.

A utilização de parêntesis é muito importante para organizar e permitir uma compreensão melhor da expressão. As comparações só podem ser feitas com variáveis do mesmo tipo, embora em uma expressão isso pode conter várias comparações com tipos diferentes.

Exemplo de utilização:

```
varValor := (varA > varB) AND ((varC <= varD) OR (varE == varF));
```

O resultado dessa expressão só será verdadeiro (TRUE), se todas as condições forem verdadeiras.

Note que "varA" e "varB" precisam ser do mesmo tipo. O mesmo ocorre com "varC" e "varD".

Mas "varA" e "varC" não precisam ser do mesmo tipo.

## 3.0 Variáveis de Sistema

Para facilitar a programação do PLC, o CNC possui diversas variáveis definidas.

Chamamos estas variáveis de “Variáveis de Sistema”, uma vez que o usuário não tem permissão para editá-las.

Todas essas variáveis começam com o prefixo “sv”.

Logo ao digitar “sv” o intellisense do editor do PLC mostrará todas as “Variáveis de Sistema” existentes.

### 3.1 Geral

#### 3.1.1 svMemFirstCycle

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Memória primeiro ciclo de PLC (ao ligar).

#### 3.1.2 svOnInitialization

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que ocorreu uma borda de subida (de 0 para 1) na inicialização do CNC

#### 3.1.3 svBlockInitialization

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Bloqueia inicialização do CNC.

Se o CNC receber o código de inicialização (1000) e esta variável estiver no estado “TRUE” a inicialização não será feita.

#### 3.1.4 svSoftkeyBlockInitialization

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Bit 1 de bloqueio de softkey, associado à softkey inicialização.

#### 3.1.5 svCncInitialized

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que o CNC já foi inicializado, saiu do modo de inicialização.

### 3.1.6 svEmergencyCnc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica estado de emergência para o CNC.

### 3.1.7 svEmergencyOutput

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Estado da saída de emergência.

### 3.1.8 svEmergencyState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Estado de emergência interna do CNC.

### 3.1.9 svInitializationMode

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Modo de inicialização.

### 3.1.10 svNewPLC

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que um novo PLC foi enviado ao CNC.

### 3.1.11 svParametersUpdated

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Variável que indica se houve uma atualização de parâmetros, novo PLC ou que o primeiro ciclo de PLC foi executado.

### 3.1.12 svUserLevel

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Informa o nível do usuário atual.

### 3.1.13 svVoid

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Variável auxiliar para retorno de todas funções que não necessitam de retorno.

### 3.1.14 svTransition500ms

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Fica em TRUE por 1 ciclo de PLC a cada 500ms.

### 3.1.15 svBlink500ms

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Pisca 500ms em estado "TRUE" e 500ms em estado "FALSE".

### 3.1.16 svIntDraft1

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Usado como rascunho para qualquer lógica.

### 3.1.17 svIntDraft2

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Usado como rascunho para qualquer lógica.

### 3.1.18 svRealDraft1

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Usado como rascunho para qualquer lógica.

### 3.1.19 svRealDraft2

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Usado como rascunho para qualquer lógica.

### 3.1.20 svSaveMemory

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Solicita salvar valores na área protegida por retenção

### 3.1.21 svAxisInHoming

**Tipo:**

BOOL

**Descrição:**

Variável que indica se eixo está fazendo referência.

### 3.1.22 svAxisLagError

**Tipo:**

BOOL

**Descrição:**

Indica o erro de lag do eixo.

### 3.1.23 svClearLagError

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Limpa erro de acompanhamento no CNC.

### 3.1.24 svFeedholdCnc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica feedhold geral para CNC.

### 3.1.25 svReferenceMode

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Modo de referência.

### 3.1.26 svSelectOrigin

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Seleção de origem para o preset.

## 3.2 Comandos de Execução de Programas ISO

### 3.2.1 svStartCnc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Comando "start" para o CNC.

### 3.2.2 svStopCnc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Comando "stop" para o CNC.

### 3.2.3 svOnKeyStart

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que ocorreu uma borda de subida (de 0 para 1) na tecla "start" do CNC.

### 3.2.4 svOnKeyStop

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que ocorreu uma borda de subida (de 0 para 1) na tecla "stop" do CNC.

### 3.2.5 svBlockStart

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Bloqueia o início de execução de um programa ISO.

Se o CNC receber o código de início de programa ISO (400) ou a variável de sistema "svStartCnc" for para o estado "TRUE" e esta variável estiver no estado "TRUE" a execução do programa ISO selecionado não será iniciada.

### 3.2.6 svAbortCnc

**Tipo:**

BOOL – Leitura/Escrita

**Descrição:**

Se o usuário atribuir o valor "TRUE" a essa variável, aborta a execução do programa ISO no CNC.

Em condições normais manter o valor desta variável em "FALSE".

### 3.3 Execução de Programas ISO

#### 3.3.1 svCncInElementSimulation

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que o CNC operando em modo de simulação de elemento.

#### 3.3.2 svCncInSimulatedMode

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que o CNC operando em modo de simulação.

#### 3.3.3 svExecutionError

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Erro de execução de programa.

#### 3.3.4 svExecutionHold

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Execução abortada ou aguardando autorização para executar.

#### 3.3.5 svExecutionInEmergency

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Execução em estado de emergência.

#### 3.3.6 svExecutionInInitialState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Execução em estado inicial.



### 3.3.7 svExecutionInWaitState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Execução em espera.

### 3.3.8 svExecutionPause

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que a execução de um programa foi pausada.

### 3.3.9 svExecutionRun

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que o CNC está executando um programa.

### 3.3.10 svNewCodeM

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Nova função M para PLC.

### 3.3.11 svNewCodeS

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Nova função S para PLC.

### 3.3.12 svNewCodeT

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Nova função T para PLC.

### 3.3.13 svAnyFunctionMST

**Tipo:**

BOOL – Somente Leitura

**Descrição:**

Constante para as Funções M, S e T usarem quando se quiser saber se uma função M, S ou T foi executada independente do código.

### 3.4 Seleção de Programas ISO

#### 3.4.1 svSelectDirNumber

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Número do diretório do programa que será selecionado pelo PLC

0 – Diretório DATA\PROGRAMS

1 – Diretório DATA\PROGRAMS\SUB.DIR

1 – Diretório DATA\PROGRAMS\CYC.DIR

#### 3.4.2 svSelectProgramNumber

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Número do programa que será selecionado pelo PLC

#### 3.4.3 svKeySelectProgram

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Seleciona código do programa. Os números do programa e do diretório devem ser programados em Select\_Program\_Number e Select\_Dir\_Number.

#### 3.4.4 svKeyPgmCall

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Chama a execução do programa #1, CYC.DIR.

### 3.5 Potenciômetros

#### 3.5.1 svPotOverridePlcControl

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Se o estado desta variável for “TRUE” determina que o controle do potenciômetro será feito pelo PLC.

#### 3.5.2 svPotOverride

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Porcentagem do potenciômetro de avanço, que será utilizado para controle de velocidade de todos os movimentos de eixos lineares e rotativos.

#### 3.5.3 svPotSpindle

**Tipo:**

BOOL

**Descrição:**

Leitura / Escrita.

Porcentagem do potenciômetro de spindle, que será utilizado para o controle de velocidade do eixo árvore.

### 3.6 Erros

#### 3.6.1 svCncError

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica o número do erro que estiver ocorrendo no CNC.

Se o valor desta variável for igual a "0", o CNC não possui erros.

#### 3.6.2 svStoFault

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que svCncError está com valor entre 50001 e 50003.

#### 3.6.3 svExternalClear

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Usado para executar "clear" por botão externo.

### 3.7 Códigos de Tecla

#### 3.7.1 svNewCncKey

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Nova tecla pressionada pelo usuário.

#### 3.7.2 svRequestKeyPlc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica qual a tecla que se quer passar para o CNC.

#### 3.7.3 svSendKeyToCnc

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Comando para enviar sinal com código de tecla ao CNC.

#### 3.7.4 svLastCncKey

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Última tecla pressionada ou recebida do CNC.

#### 3.7.5 svOnKeyCE

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que ocorreu uma borda de subida (de 0 para 1) na tecla CE.

### 3.8 Telas

#### 3.8.1 svCurrentScreen

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Código da tela atual em foco.

#### 3.8.2 svLastScreen

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Foco anterior ao atual em foco.

### 3.9 Árvores de Softkeys

#### 3.9.1 svCurrentHorizontalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Estado atual das softkeys horizontais.

#### 3.9.2 svCurrentVerticalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Estado atual das softkeys verticais.

#### 3.9.3 svCncNewSoftkeyState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica que um novo estado de softkeys foi chamado pelo usuário.

#### 3.9.4 svCncNewHorizontalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Novo estado de softkeys horizontais chamado por usuário.

#### 3.9.5 svPlcNewHorizontalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica qual o estado horizontal de softkeys que se quer chamar.

#### 3.9.6 svCncNewVerticalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Novo estado de softkeys verticais chamado pelo usuário.



### 3.9.7 svPlcNewVerticalState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Indica qual o estado vertical de softkeys que se quer chamar.

### 3.9.8 svRequestNewSoftkeyState

**Tipo:**

BOOL

**Descrição:**

Somente Leitura.

Requisita que o CNC configure um novo estado de softkeys.

## 4.0 Funções Nativas

As funções Nativas possuem esse nome pois foram criadas diretamente no firmware do PLC/CNC e não podem ser editadas ou customizadas.

As funções Nativas não possuem nenhum prefixo para identificação, mas pode-se identificá-las de outra forma.

Essas funções são descritas com letras maiúsculas.

As funções nativas podem ter mais de um dado de entrada, mas têm por característica comente um dado de saída (apenas para as funções que fazem leitura).

As funções nativas de escrita ou que são utilizadas para algum tipo de exibição não retornam nenhum valor.

## 4.1 Mensagens

As funções aqui descritas exibiram mensagens na parte superior do CNC AKC-PPC.

Essas mensagens são exibidas de acordo com o tipo das mesmas.

As mensagens exibidas aqui devem estar devidamente registradas no Editor de Alarmes do IDE AKC-PPC.

Para isto consulte o manual do usuário do IDE AKC-PPC.

### 4.1.1 Alarme - ALARM

#### 4.1.1.1 Funcionamento

Esta função exibirá na tela de forma contínua uma mensagem do tipo “Alarme” quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" receber uma transição de subida, ou seja, mudar seu valor do estado “FALSE” para o estado “TRUE”.

Quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" mudar para o estado “FALSE” a mensagem continuará sendo exibida.

Este tipo de mensagem só é removido da tela quando o código para limpar erros for enviado ao CNC (código 333).

Quando esta função for chamada a cor de fundo da mensagem exibida será vermelha.

Podem ser utilizados até 64 alarmes na aplicação PLC - ou seja, os índices vão de 0 a 63.

#### 4.1.1.2 Entradas

*INT\_NUMBER:*

Índice do alarme - definido no editor de alarmes.

*BOOL\_STATE:*

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 5.1.1.3 Chamada da função

ALARM(INT\_NUMBER, BOOL\_STATE);

### 4.1.2 Piscante - BLINK\_MESSAGE

#### 4.1.2.1 Funcionamento

Esta função exibirá na tela de forma piscante com intervalo de 2s uma mensagem do tipo “Piscante” quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" estiver no estado “TRUE”.

Quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" mudar para o estado “FALSE” a mensagem não será exibida mais.

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 32 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### 4.1.2.2 Entradas

*INT\_NUMBER:*

Índice do alarme - definido no editor de alarmes.

*BOOL\_STATE:*

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 4.1.2.3 Chamada da função

BLINK\_MESSAGE(INT\_NUMBER, BOOL\_STATE);

### 4.1.3 Normal - MESSAGE

#### 4.1.3.1 Funcionamento

Esta função exibirá na tela de forma contínua uma mensagem do tipo “Piscante” quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" estiver no estado “TRUE”.

Quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" mudar para o estado “FALSE” a mensagem não será exibida mais.

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 128 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 127.

#### **4.1.3.2 Entradas**

*INT\_NUMBER:*

Índice do alarme - definido no editor de alarmes.

*BOOL\_STATE:*

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### **4.1.3.3 Chamada da função**

MESSAGE(INT\_NUMBER, BOOL\_STATE);

### **4.1.4 Temporizada - TIMED\_MESSAGE**

#### **5.1.4.1 Funcionamento**

Apresenta uma mensagem com a cor amarela no topo da tela do CNC. Essa mensagem fica visível por 4 segundos após a função receber uma borda de subida (de FALSE para TRUE) na entrada "BOOL\_STATE". Podem ser utilizadas até 32 mensagens diferentes na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### **4.1.3.1 Funcionamento**

Esta função exibirá na tela por 4s uma mensagem do tipo "Temporizada" quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" receber uma transição de subida, ou seja, mudar seu valor do estado "FALSE" para o estado "TRUE".

Passado o tempo de exibição (4s) a mensagem não será exibida mais até que aconteça novamente uma transição de subida na memória do PLC que estiver vinculada a entrada "BOOL\_STATE".

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 32 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### **4.1.3.2 Entradas**

*INT\_NUMBER:*

Índice do alarme - definido no editor de alarmes.

*BOOL\_STATE:*

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### **4.1.3.3 Chamada da função**

TIMED\_MESSAGE(INT\_NUMBER, BOOL\_STATE);

## 4.2 Transição

### 4.2.1 Transição de Subida - R\_EDGE

#### 4.2.1.1 Funcionamento

Esta função detecta uma transição de subida, ou seja, a transição do estado FALSE para o estado TRUE de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 4.2.1.2 Entradas

*BOOL:*

Variável ou expressão booleana que se quer detectar a transição de subida.

#### 4.2.1.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição de subida no booleano da entrada.

#### 4.2.1.4 Chamada da função

```
R_EDGE(BOOL);
```

### 4.2.2 Transição de Descida - F\_EDGE

#### 4.2.2.1 Funcionamento

Esta função detecta uma transição de descida, ou seja, a transição do estado TRUE para o estado FALSE de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 4.2.2.2 Entradas

*BOOL:*

Variável ou expressão booleana que se quer detectar a transição de descida.

#### 4.2.2.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição de descida no booleano da entrada.

#### 4.2.2.4 Chamada da função

```
F_EDGE(BOOL);
```

### 4.2.3 Qualquer Transição - RF\_EDGE

#### 4.2.3.1 Funcionamento

Esta função detecta qualquer transição de estado (de subida e de descida) de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 4.2.3.2 Entradas

*BOOL:*

Variável ou expressão booleana que se quer detectar a transição.

#### 4.2.3.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição no booleano da entrada.

#### 4.2.3.4 Chamada da função

```
RF_EDGE(BOOL);
```

## 4.3 Parâmetros, Variáveis H e Variáveis Reservadas

### 4.3.1 Ler Parâmetro - R\_PARAM

#### 4.3.1.1 Funcionamento

Retorna o valor do parâmetro do CNC indicado na entrada da função.

#### 4.3.1.2 Entradas

*INT*:

Número inteiro referente ao índice do parâmetro que será lido. Aceita valores de 0 a 4199.

Parâmetros gerais: 0-499

Parâmetros de eixo:  $1000 - 4199$  ( $\text{parIndice} = 1000 + ((\text{eixo} - 1) * 100) + \text{parNum}$ )

Parâmetros PLC: 900-999

Índices de 500 a 899 não são utilizados.

#### 4.3.1.3 Saídas

Retorna o valor do parâmetro indicado no argumento "INT". O valor retornado é do tipo REAL.

#### 4.3.1.4 Chamada da função

R\_PARAM(INT);

### 4.3.2 Escrever no Parâmetro - W\_PARAM

#### 4.3.2.1 Funcionamento

Altera o valor do parâmetro indicado na entrada da função.

#### 4.3.2.2 Entradas

*INT*:

Número inteiro referente ao índice do parâmetro que será lido. Aceita valores de 0 a 4199.

Parâmetros gerais: 0-499

Parâmetros de eixo:  $1000 - 4199$  ( $\text{parIndice} = 1000 + ((\text{eixo} - 1) * 100) + \text{parNum}$ )

Parâmetros PLC: 900-999

Índices de 500 a 899 não são utilizados.

*REAL*:

Novo valor do tipo REAL a ser atribuído ao parâmetro indicado no argumento "INT".

#### 4.3.2.3 Saídas

Esta função não tem retorno.

#### 4.3.2.4 Chamada da função

W\_PARAM(INT, REAL);

### 4.3.3 Ler Registrador H - R\_HVAR

#### 4.3.3.1 Funcionamento

Retorna o valor do registrador H indicado na entrada da função.

#### 4.3.3.2 Entradas

*INT*:

Número do registrador H que se quer ler. Aceita valores de 0 a 999.

#### 4.3.3.3 Saídas

Retorna o valor do registrador H indicada no argumento "INT". O valor retornado é do tipo REAL.

#### 4.3.3.4 Chamada da função

R\_HVAR(INT);

#### **4.3.4 Escrever no Registrador H - W\_HVAR**

##### **4.3.4.1 Funcionamento**

Altera o valor do registrador H indicado na entrada da função.

##### **4.3.4.2 Entradas**

*INT:*

Número do registrador H que se quer alterar o valor. Aceita valores de 0 a 999.

*REAL:*

Novo valor a ser atribuído ao registrador H indicada no argumento "INT".

##### **4.3.4.3 Saídas**

Esta função não tem retorno.

##### **4.3.4.4 Chamada da função**

W\_HVAR(INT, REAL);

#### **4.3.5 Ler Variável Reservada - R\_COMMAND\_VAR**

##### **4.3.5.1 Funcionamento**

Retorna o valor da variável reservada indicada na entrada da função.

A lista de variáveis reservadas existentes pode ser vista no manual "Variáveis Reservadas".

##### **4.3.5.2 Entradas**

*INT:*

Número da variável reservada que se quer ler. Aceita valores de 10000 a 19999, que é o número de variáveis reservadas existentes. Nem todos os números de variáveis reservadas executam alguma ação.

##### **4.3.5.3 Saídas**

Retorna o valor da variável reservada indicada no argumento "INT". O valor retornado é do tipo REAL.

##### **4.3.5.4 Chamada da função**

R\_COMMAND\_VAR(INT);

#### **4.3.6 Escrever na Variável Reservada - W\_COMMAND\_VAR**

##### **4.3.6.1 Funcionamento**

Altera o valor da variável reservada indicada na entrada da função.

A lista de variáveis reservadas existentes pode ser vista no manual "Variáveis Reservadas".

##### **4.3.6.2 Entradas**

*INT:*

Número da variável reservada que se quer alterar o valor, podem ser programados valores de 10000 a 19999 que são os números de variáveis reservadas existentes.

*REAL:*

Novo valor a ser atribuído à variável reservada indicada no argumento "INT".

##### **4.3.6.3 Saídas**

Esta função não tem retorno.

##### **4.3.6.4 Chamada da função**

W\_COMMAND\_VAR(INT, REAL);

## 4.4 Trigonométricas/Matemáticas

### 4.4.1 Seno - SIN

#### 4.4.1.1 Funcionamento

Retorna o valor do seno do ângulo inserido.

#### 4.4.1.2 Entradas

##### **REAL:**

Valor real (em graus) do ângulo que se deseja calcular o seno.

#### 4.4.1.3 Saídas

Retorna o valor do seno do ângulo que foi inserido na entrada "REAL".

#### 4.4.1.4 Chamada da função

SIN(REAL)

### 4.4.2 Cosseno - COS

#### 4.4.2.1 Funcionamento

Retorna o valor do cosseno do ângulo inserido.

#### 4.4.2.2 Entradas

##### **REAL:**

Valor real (em graus) do ângulo que se deseja calcular o cosseno.

#### 4.4.2.3 Saídas

Retorna o valor do cosseno do ângulo que foi inserido na entrada "REAL".

#### 4.4.2.4 Chamada da função

COS(REAL)

### 4.4.4 Tangente - TAN

#### 4.4.4.1 Funcionamento

Retorna o valor da tangente do ângulo inserido.

#### 4.4.4.2 Entradas

##### **REAL:**

Valor real (em graus) do ângulo que se deseja calcular a tangente.

#### 4.4.4.3 Saídas

Retorna o valor da tangente do ângulo que foi inserido na entrada "REAL".

#### 4.4.4.4 Chamada da função

TAN(REAL)

### 4.4.5 Arco Tangente - ATAN

#### 4.4.5.1 Funcionamento

Retorna o valor do arco tangente do valor inserido.

#### 4.4.5.2 Entradas

##### **REAL:**

Valor real que se deseja calcular o arco tangente.

#### 4.4.5.3 Saídas

Retorna o valor do arco tangente do número que foi inserido na entrada "REAL".

#### 4.4.5.4 Chamada da função

ATAN(REAL)



#### 4.4.6 Valor da Constante Pi - PI

##### 4.4.6.1 Funcionamento

Retorna o valor da constante matemática PI.

##### 4.4.6.2 Entradas

Nessa função não há entrada.

##### 4.4.6.3 Saídas

Retorna o valor da constante matemática PI.

##### 4.4.6.4 Chamada da função

PI()

#### 4.4.7 Valor Inteiro - TRUNC

##### 4.4.7.1 Funcionamento

Retorna o valor inteiro de um número. Exemplo: TRUNC(35.33) retornará 35.

##### 4.4.7.2 Entradas

###### **REAL:**

Valor real do qual se deseja remover as casas decimais.

##### 4.4.7.3 Saídas

Retorna o valor inteiro do número que foi inserido na entrada "REAL".

##### 4.4.7.4 Chamada da função

TRUNC(24.4456)

#### 4.4.8 Valor Fracionário - FRAC

##### 4.4.8.1 Funcionamento

Retorna o valor da parte fracionária de um número real. Exemplo: FRAC(35.33) retornará 0.33.

##### 4.4.8.2 Entradas

###### **REAL:**

Valor real do qual se deseja obter a parte fracionária.

##### 4.4.8.3 Saídas

Retorna um número REAL apenas com a parte fracionária do número que foi inserido na entrada "REAL".

##### 4.4.8.4 Chamada da função

FRAC(45.354)

#### 4.4.9 Raiz Quadrada - SQRT

##### 4.4.9.1 Funcionamento

Retorna a raiz quadrada do número inserido.

##### 4.4.9.2 Entradas

###### **REAL:**

Valor REAL do qual se deseja obter a raiz quadrada.

##### 4.4.9.3 Saídas

Retorna o valor da raiz quadrada do número que foi inserido na entrada "REAL".

##### 4.4.9.4 Chamada da função

SQRT(9)

### **4.4.10 Valor Absoluto - ABS**

#### **4.4.10.1 Funcionamento**

Retorna o módulo de um número real.

#### **4.4.10.2 Entradas**

**REAL:**

Valor REAL do qual se deseja obter o módulo.

#### **4.4.10.3 Saídas**

Retorna o valor do módulo do número que foi inserido na entrada "REAL".

#### **4.4.10.4 Chamada da função**

ABS(-3)

## 4.5 Conversões de tipos de dados

### 4.5.1 Booleano para Real - BOOL\_TO\_REAL

#### 4.5.1.1 Funcionamento

Converte uma variável do tipo BOOL para o tipo REAL.

#### 4.5.1.2 Entradas

##### BOOL:

Valor do tipo BOOL que será convertido para o tipo REAL.

#### 4.5.1.3 Saídas

Retorna o valor booleano convertido para REAL.

#### 4.5.1.4 Chamada da função

BOOL\_TO\_REAL(BOOL)

### 4.5.6 Booleano para Inteiro - BOOL\_TO\_INT

#### 4.5.6.1 Funcionamento

Converte uma variável do tipo BOOL para o tipo INT.

#### 4.5.6.2 Entradas

##### BOOL:

Valor do tipo BOOL que será convertido para o tipo INT.

#### 4.5.6.3 Saídas

Retorna o valor booleano convertido para INT.

#### 4.5.6.4 Chamada da função

BOOL\_TO\_INT(BOOL);

### 4.5.2 Real para Booleano - REAL\_TO\_BOOL

#### 4.5.2.1 Funcionamento

Converte uma variável do tipo REAL para o tipo BOOL.

#### 4.5.2.2 Entradas

##### REAL:

Valor do tipo REAL que será convertido para o tipo BOOL.

#### 4.5.2.3 Saídas

Retorna o valor real convertido para BOOL.

#### 4.5.2.4 Chamada da função

REAL\_TO\_BOOL(REAL);

### 4.5.3 Real para Inteiro - REAL\_TO\_INT

#### 4.5.3.1 Funcionamento

Converte uma variável do tipo REAL para o tipo INT.

#### 4.5.3.2 Entradas

##### REAL:

Valor do tipo REAL que será convertido para o tipo INT.

#### 4.5.3.3 Saídas

Retorna o valor real convertido para INT.

#### 4.5.3.4 Chamadas das funções

REAL\_TO\_INT(REAL);

#### **4.5.4 Real para Inteiro sem sinal - REAL\_TO\_UINT**

##### **4.5.4.1 Funcionamento**

Converte uma variável do tipo REAL para o tipo UINT.

##### **4.5.4.2 Entradas**

###### **REAL:**

Valor do tipo REAL que será convertido para o tipo UINT.

##### **4.5.4.3 Saídas**

Retorna o valor real convertido para UINT.

##### **4.5.4.4 Chamadas das funções**

REAL\_TO\_UINT (REAL);

#### **4.5.5 Inteiro sem sinal para Inteiro - UINT\_TO\_INT**

##### **4.5.5.1 Funcionamento**

Converte uma variável do tipo UINT para o tipo INT.

##### **4.5.5.2 Entradas**

###### **UINT:**

Valor do tipo UINT que será convertido para o tipo INT.

##### **4.5.5.3 Saídas**

Retorna o valor UINT convertido para INT.

##### **4.5.5.4 Chamada da função**

UINT\_TO\_INT(UINT);

#### **4.5.6 Inteiro para Inteiro sem sinal - INT\_TO\_UINT**

##### **4.5.6.1 Funcionamento**

Converte uma variável do tipo INT para o tipo UINT.

##### **4.5.6.2 Entradas**

###### **INT:**

Valor do tipo INT que será convertido para o tipo UINT.

##### **4.5.6.3 Saídas**

Retorna o valor INT convertido para UINT.

##### **4.5.6.4 Chamada da função**

INT\_TO\_UINT(INT);

#### **4.5.7 Inteiro para Booleano - INT\_TO\_BOOL**

##### **4.5.7.1 Funcionamento**

Converte uma variável do tipo INT para o tipo BOOL.

##### **4.5.7.2 Entradas**

###### **INT:**

Valor do tipo INT que será convertido para o tipo BOOL.

##### **4.5.7.3 Saídas**

Retorna o valor INT convertido para BOOL.

##### **4.5.7.4 Chamada da função**

INT\_TO\_BOOL(INT);

### **4.5.8 Inteiro para Real - INT\_TO\_REAL**

#### **4.5.8.1 Funcionamento**

Converte uma variável do tipo INT para o tipo REAL.

#### **4.5.8.2 Entradas**

**INT:**

Valor do tipo INT que será convertido para o tipo REAL.

#### **4.5.8.3 Saídas**

Retorna o valor INT convertido para REAL.

#### **4.5.8.4 Chamada da função**

INT\_TO\_REAL (INT);

### 5.0 Funções de Sistema

Para facilitar a programação do PLC, o CNC possui diversas funções definidas.

As funções de “Funções de Sistema” possuem este nome pois o usuário não tem permissão para editá-las.

Todas as “Funções de Sistema” começam com o prefixo “sf”.

Logo ao digitar “sf” o intellisense mostrará todas as “Funções de Sistema” existentes.

As funções de sistema podem ter diversos dados de entrada, mas têm por característica apenas um dado de saída.

## 5.1 Mensagens

As funções aqui descritas exibirão mensagens na parte superior do CNC AKC-PPC.

Essas mensagens são exibidas de acordo com o tipo das mesmas.

As mensagens exibidas aqui devem estar devidamente registradas no Editor de Alarmes do IDE AKC-PPC.

Para isto consulte o manual do usuário do IDE AKC-PPC.

### 5.1.1 Alarme - `sfShowAlarmMessage`

#### 5.1.1.1 Funcionamento

Esta função exibirá na tela de forma contínua uma mensagem do tipo "Alarme" quando a memória do PLC que estiver vinculada a entrada "SHOW" receber uma transição de subida, ou seja, mudar seu valor do estado "FALSE" para o estado "TRUE".

Quando a memória do PLC que estiver vinculada a entrada " SHOW " mudar para o estado "FALSE" a mensagem continuará sendo exibida.

Este tipo de mensagem só é removido da tela quando o código para limpar erros for enviado ao CNC (código 333).

Quando esta função for chamada a cor de fundo da mensagem exibida será vermelha.

Podem ser utilizados até 64 alarmes na aplicação PLC - ou seja, os índices vão de 0 a 63.

#### 5.1.1.2 Entradas

##### **INDEX (INT):**

Índice do alarme - definido no editor de mensagens do IDE AKC-PPC.

##### **SHOW (BOOL):**

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 5.1.1.2 Saídas

Retorna um booleano, indicando se a mensagem está sendo exibida pela função ou não (TRUE ou FALSE).

#### 5.1.1.3 Chamada da função

```
sfShowAlarmMessage(INDEX, SHOW);
```

### 5.1.2 Piscante - `sfShowBlinkingMessage`

#### 5.1.2.1 Funcionamento

Esta função exibirá na tela de forma piscante com intervalo de 2s uma mensagem do tipo "Piscante" quando a memória do PLC que estiver vinculada a entrada " SHOW " estiver no estado "TRUE".

Quando a memória do PLC que estiver vinculada a entrada " SHOW " mudar para o estado "FALSE" a mensagem não será exibida mais.

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 32 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### 5.1.2.2 Entradas

##### **INDEX (INT):**

Índice da mensagem piscante - definido no editor de mensagens do IDE AKC-PPC.

##### **SHOW (BOOL):**

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 5.1.2.3 Chamada da função

```
sfShowBlinkingMessage(INDEX, SHOW);
```

### 5.1.3 Normal - sfShowNormalMessage

#### 5.1.3.1 Funcionamento

Esta função exibirá na tela de forma contínua uma mensagem do tipo “Piscante” quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" estiver no estado “TRUE”.

Quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" mudar para o estado “FALSE” a mensagem não será exibida mais.

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 128 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 127.

#### 5.1.3.2 Entradas

##### **INDEX (INT):**

Índice da mensagem normal - definido no editor de mensagens do IDE AKC-PPC.

##### **SHOW (BOOL):**

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 5.1.3.3 Chamada da função

```
sfShowNormalMessage(INDEX, SHOW);
```

### 5.1.4 Temporizada - sfShowTimedMessage

#### 5.1.4.1 Funcionamento

Apresenta uma mensagem com a cor amarela no topo da tela do CNC. Essa mensagem fica visível por 4 segundos após a função receber uma borda de subida (de FALSE para TRUE) na entrada "BOOL\_STATE".

Podem ser utilizadas até 32 mensagens diferentes na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### 5.1.3.1 Funcionamento

Esta função exibirá na tela por 4s uma mensagem do tipo “Temporizada” quando a memória do PLC que estiver vinculada a entrada "BOOL\_STATE" receber uma transição de subida, ou seja, mudar seu valor do estado “FALSE” para o estado “TRUE”.

Passado o tempo de exibição (4s) a mensagem não será exibida mais até que aconteça novamente uma transição de subida na memória do PLC que estiver vinculada a entrada "BOOL\_STATE".

Quando esta função for chamada a cor de fundo da mensagem exibida será amarela.

Podem ser utilizadas até 32 mensagens na aplicação PLC - ou seja, os índices vão de 0 a 31.

#### 5.1.3.2 Entradas

##### **INDEX (INT):**

Índice da mensagem temporizada - definido no editor de mensagens do IDE AKC-PPC.

##### **SHOW (BOOL):**

Habilita a exibição da mensagem quando estiver no estado TRUE.

#### 5.1.3.3 Chamada da função

```
sfShowTimedMessage(INDEX, SHOW);
```



## 5.2 Códigos e Navegação

### 5.2.1 Identificar Código Recebido pelo CNC - `sfKeyCode`

#### 5.2.1.1 Funcionamento

Quando o CNC receber um código esta função compara se o código recebido pelo CNC é igual ao código informado na entrada "Code".

Os códigos podem ser enviados por componentes de tela da IHM, por um teclado físico, por softkeys ou por scripts.

#### 5.2.1.2 Entradas

##### **Code (INT):**

Código que será comparado com o código recebido pelo CNC.

#### 5.2.1.3 Saídas

Esta função retorna um booleano.

Se o código recebido pelo CNC for igual ao código que foi informado na entrada "Code" esta função retorna TRUE, se não for igual retorna FALSE.

Se o CNC não receber nenhum código esta função retorna FALSE.

#### 5.2.1.4 Chamada da função

```
sfKeyCode(Code);
```

### 5.2.2 Pegar Árvore Horizontal Selecionada - `sfCurrentHorizontalSoftkeyState`

#### 5.2.2.1 Funcionamento

Busca o índice da árvore de softkeys horizontais que está selecionada.

#### 5.2.2.2 Entradas

Não possui entradas.

#### 5.2.2.3 Saídas

Retorna um número inteiro com o índice da árvore de softkeys horizontais que está selecionada.

#### 5.2.2.4 Chamada da função

```
sfCurrentHorizontalSoftkeyState()
```

### 5.2.3 Pegar Árvore Vertical Selecionada - `sfCurrentVerticalSoftkeyState`

#### 5.2.3.1 Funcionamento

Busca o índice da árvore de softkeys verticais que está selecionada.

#### 5.2.3.2 Entradas

Não possui entrada.

#### 5.2.3.3 Saídas

Retorna um número inteiro com o índice da árvore de softkeys verticais que está selecionada.

#### 5.2.3.4 Chamada da função

```
sfCurrentVerticalSoftkeyState()
```

### 5.2.4 Verifica Tela Selecionada - `sfCurrentScreen`

#### 5.2.4.1 Funcionamento

Verifica o índice da tela que está em selecionada.

Retorna "TRUE" quando o número da tela associada a variável de entrada "Screen\_Number" é igual a tela em foco.

#### 5.2.4.2 Entradas

##### **ScreenNumber (INT):**

Número da tela que será verificado.

O número da tela pode ser verificado no editor "IHM Studio" na árvore de projetos do IDE AKC-PPC.

#### 5.2.4.3 Saídas

Esta função retorna um booleano.

Se o índice da tela selecionada for igual ao que foi informado na entrada " ScreenNumber " esta função retorna TRUE, se não for igual retorna FALSE.

#### 5.2.4.4 Chamada da função

```
sfCurrentScreen(ScreenNumber);
```

### 5.2.5 Bloqueio de Softkey - `sfBlockSoftkey`

Esta função não é utilizada mais em versões acima da 3.0.

Ela está mantida por uma questão de compatibilidade.

As softkeys possuem uma propriedade chama LockBit e nesta propriedade utilizamos uma variável do PLC para fazer qualquer tipo de bloqueio necessário.

### 5.2.6 Mudar Estado da Softkey - `sfChangeSoftkey`

Esta função não é utilizada mais em versões acima da 3.0.

Ela está mantida por uma questão de compatibilidade.

As softkeys possuem uma propriedade chama MemoryBit e nesta propriedade utilizamos uma variável do PLC para controlar o estado que a softkey está.

## 5.3 Execução

Uma das formas de interação entre os programas ISO e o PLC são as funções M, S e T.

### 5.3.1 sfFunctionM

#### 5.3.1.1 Funcionamento

Esta função tem a finalidade de monitorar as funções M enviadas ao CNC pelos programas ISO.

Quando um programa ISO enviar um código M ao CNC esta função compara se o código recebido pelo CNC é igual ao código informado na entrada "FunctionCode". Se isso ocorrer o retorno desta função será TRUE, o que significa que a função M que estamos monitorando foi enviada ao PLC, caso contrário o retorno da função é FALSE.

A entrada "HoldFunctionM" segura a execução do programa ISO. Portanto se a entrada "HoldFunctionM" estiver em TRUE quando o a função M que estamos monitorando for detectada, a execução do CNC será bloqueada, não permitindo que o programa ISO avance para executar a próxima instrução até que entrada "HoldFunctionM" vá para FALSE.

#### 5.3.1.2 Entradas

##### **FunctionCode (INT):**

Código da função M que se quer monitorar.

##### **HoldFunctionM (BOOL):**

Caso esta entrada seja TRUE, a execução do CNC fica parada na linha desta função até que HoldFunctionM caia para FALSE.

#### 5.3.1.3 Saída

Esta função retorna um booleano que indica se a função M chamada em um programa CNC for igual à função cujo código foi passado na entrada "FunctionCode".

#### 5.3.1.4 Chamada da função

```
sfFunctionM(HoldFunctionM, FunctionCode);
```

### 5.3.2 sfFunctionS

#### 5.3.2.1 Funcionamento

Esta função tem a finalidade de monitorar as funções S enviadas ao CNC pelos programas ISO.

Quando um programa ISO enviar um código S ao CNC esta função compara se o código recebido pelo CNC é igual ao código informado na entrada "FunctionCode". Se isso ocorrer o retorno desta função será TRUE, o que significa que a função S que estamos monitorando foi enviada ao PLC, caso contrário o retorno da função será FALSE.

A entrada "HoldFunctionS" segura a execução do programa ISO. Portanto se a entrada "HoldFunctionS" estiver em TRUE quando o a função S que estamos monitorando for detectada, a execução do CNC será bloqueada, não permitindo que o programa ISO avance para executar a próxima instrução até que entrada "HoldFunctionS" vá para FALSE.

#### 5.3.2.2 Entradas

##### **FunctionCode (INT):**

Código da função S que se quer monitorar.

##### **HoldFunctionS (BOOL):**

Caso esta entrada seja TRUE, a execução do CNC fica parada na linha desta função até que HoldFunctionS caia para FALSE.

#### 5.3.2.3 Saída

Esta função retorna um booleano que indica se a função S chamada em um programa CNC for igual à função cujo código foi passado na entrada "FunctionCode".

#### 5.3.2.4 Chamada da função

```
sfFunctionS(HoldFunctionS, FunctionCode);
```

### 5.3.3 sfFunctionT

#### 5.3.3.1 Funcionamento

Esta função tem a finalidade de monitorar as funções T enviadas ao CNC pelos programas ISO.

Quando um programa ISO enviar um código T ao CNC esta função compara se o código recebido pelo CNC é igual ao código informado na entrada "FunctionCode". Se isso ocorrer o retorno desta função será TRUE, o que significa que a função T que estamos monitorando foi enviada ao PLC, caso contrário o retorno da função é FALSE.

A entrada "HoldFunctionT" segura a execução do programa ISO. Portanto se a entrada "HoldFunctionT" estiver em TRUE quando o a função T que estamos monitorando for detectada, a execução do CNC será bloqueada, não permitindo que o programa ISO avance para executar a próxima instrução até que entrada "HoldFunctionT" vá para FALSE.

#### 5.3.3.2 Entradas

##### **FunctionCode (INT):**

Código da função T que se quer monitorar. Caso se queira monitorar se existe uma função T independentemente do código, deve-se chamar a função com a constante svAnyFunctionMST.

##### **HoldFunctionT (BOOL):**

Caso esta entrada seja TRUE, a execução do CNC fica parada na linha desta função até que HoldFunctionT caia para FALSE.

#### 5.3.3.3 Saída

Esta função retorna um booleano que indica se a função T chamada em um programa CNC for igual à função cujo código foi passado na entrada "FunctionCode".

#### 5.3.3.4 Chamada da função

```
sfFunctionT(HoldFunctionT, FunctionCode);
```

## 5.4 Transição

### 5.4.1 Transição de Subida - sfRisingEdge

#### 5.4.1.1 Funcionamento

Esta função detecta uma transição de subida, ou seja, a transição do estado FALSE para o estado TRUE de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 5.4.1.2 Entradas

##### **Variable (BOOL):**

Variável ou expressão booleana que se quer detectar a transição de subida.

#### 5.4.1.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição de subida no booleano da entrada.

#### 5.4.1.4 Chamada da função

sfRisingEdge (Variable)

### 5.4.2 Transição de Descida - sfFallingEdge

#### 5.4.2.1 Funcionamento

Esta função detecta uma transição de descida, ou seja, a transição do estado TRUE para o estado FALSE de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 5.4.2.2 Entradas

##### **Variable (BOOL):**

Variável ou expressão booleana que se quer detectar a transição de descida.

#### 5.4.2.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição de descida no booleano da entrada.

#### 5.4.2.4 Chamada da função

sfFallingngEdge (Variable)

### 5.4.3 Qualquer Transição - sfCheckingEdge

#### 5.4.3.1 Funcionamento

Esta função detecta qualquer transição de estado (de subida e de descida) de uma variável ou expressão booleana.

Retorna TRUE por um ciclo de PLC.

#### 5.4.3.2 Entradas

##### **Variable (BOOL):**

Variável ou expressão booleana que se quer detectar a transição.

#### 5.4.3.3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que ocorreu uma transição no booleano da entrada.

#### 5.4.3.4 Chamada da função

sfCheckingEdge (Variable)

## 5.5 Parâmetros, Variáveis H e Variáveis Reservadas

### 5.5.1 Ler Parâmetro - sfReadParameter

#### 5.5.1.1 Funcionamento

Retorna o valor do parâmetro do CNC indicado na entrada da função.

#### 5.5.1.2 Entradas

**Axis (INT):**

Número do eixo do qual se deseja ler o parâmetro.

**Parameter (INT):**

Número do parâmetro que se deseja ler.

Se o tipo for geral 0: 0-499

Se o tipo for eixo ou PLC: 0-99

**Type (INT):**

Tipo do parâmetro que se deseja ler. Pode receber os seguintes valores:

Type = 0 - Geral

Type = 1 - Eixo

Type = 2 - PLC

#### 5.5.1.3 Saídas

Retorna o valor do parâmetro indicado nos argumentos de entrada. O valor retornado é do tipo REAL.

#### 5.5.1.4 Chamada da função

sfReadParameter(Axis, Parameter, Type);

### 5.5.2 Escrever no Parâmetro - sfWriteParameter

#### 5.5.2.1 Funcionamento

Altera o valor do parâmetro indicado na entrada da função.

#### 5.5.2.2 Entradas

**Axis (INT):**

Número do eixo do qual se deseja ler o parâmetro.

**Parameter (INT):**

Número do parâmetro que se deseja ler.

Se o tipo for geral 0: 0-499

Se o tipo for eixo ou PLC: 0-99

**Type (INT):**

Tipo do parâmetro que se deseja ler. Pode receber os seguintes valores:

Type = 0 - Geral

Type = 1 - Eixo

Type = 2 – PLC

**Value(REAL):**

Novo valor a ser atribuído ao parâmetro indicado.

#### 5.5.2.3 Saídas

Esta função retorna um booleano indicando que o parâmetro foi escrito.

#### 5.5.2.4 Chamada da função

sfWriteParameter(Axis, Parameter, Type, Value);

### 5.5.3 Ler Registrador H - sfReadHRegister

#### 5.5.3.1 Funcionamento

Retorna o valor do registrador H indicado na entrada da função.

#### 5.5.3.2 Entradas

**Register(INT):**

Número do registrador H que se quer ler. Aceita valores de 0 a 999.

#### 5.5.3.3 Saídas

Retorna o valor do registrador H indicada no argumento "Register". O valor retornado é do tipo REAL.

#### 5.5.3.4 Chamada da função

sfReadHRegister (Register)

### 5.5.4 Escrever no Registrador H - sfWriteHRegister

#### 5.5.4.1 Funcionamento

Altera o valor do registrador H indicado na entrada da função.

#### 5.5.4.2 Entradas

**Register(INT):**

Número do registrador H que se quer alterar o valor. Aceita valores de 0 a 999.

**Value(REAL):**

Novo valor a ser atribuído ao registrador H indicada no argumento "Register".

#### 5.5.4.3 Saídas

Esta função retorna um booleano indicando que o registrador foi escrito.

#### 5.5.4.4 Chamada da função

sfWriteHRegister (Register, Value)

### 5.5.5 Ler Variável Reservada - sfReadReservedVar

#### 5.5.5.1 Funcionamento

Retorna o valor da variável reservada indicada na entrada da função.

A lista de variáveis reservadas existentes pode ser vista no manual "Variáveis Reservadas".

#### 5.5.5.2 Entradas

**Var(INT):**

Número da variável reservada que se quer ler. Aceita valores de 10000 a 19999, que é o número de variáveis reservadas existentes. Nem todos os números de variáveis reservadas executam alguma ação.

#### 5.5.5.3 Saídas

Retorna o valor da variável reservada indicada no argumento "INT". O valor retornado é do tipo REAL.

#### 5.5.5.4 Chamada da função

sfReadReservedVar( Var )

### 5.5.6 Escrever na Variável Reservada - sfWriteReservedVar

#### 5.5.6.1 Funcionamento

Altera o valor da variável reservada indicada na entrada da função.

A lista de variáveis reservadas existentes pode ser vista no manual “Variáveis Reservadas”.

#### 5.5.6.2 Entradas

##### **Var(INT):**

Número da variável reservada que se quer alterar o valor, podem ser programados valores de 10000 a 19999 que são os números de variáveis reservadas existentes.

##### **Value(REAL):**

Novo valor a ser atribuído à variável reservada indicada no argumento “INT”.

#### 5.5.6.3 Saídas

Esta função retorna um booleano indicando que a variável reservada foi escrita.

#### 5.5.6.4 Chamada da função

sfWriteReservedVar( Var, Value )



## 5.6 Trigonométricas/Matemáticas

### 5.6.1 Seno - sfMathSin

#### 5.6.1.1 Funcionamento

Retorna o valor do seno do ângulo inserido.

#### 5.6.1.2 Entradas

##### **Ang(REAL):**

Valor real (em graus) do ângulo que se deseja calcular o seno.

#### 5.6.1.3 Saídas

Retorna o valor do seno do ângulo que foi inserido na entrada "Ang".

#### 5.6.1.4 Chamada da função

sfMathSin(Ang)

### 5.6.2 Cosseno - sfMathCos

#### 5.6.2.1 Funcionamento

Retorna o valor do cosseno do ângulo inserido.

#### 5.6.2.2 Entradas

##### **Ang(REAL):**

Valor real (em graus) do ângulo que se deseja calcular o cosseno.

#### 5.6.2.3 Saídas

Retorna o valor do cosseno do ângulo que foi inserido na entrada "Ang".

#### 5.6.2.4 Chamada da função

sfMathCos(Ang)

### 5.6.4 Tangente - sfMathTan

#### 5.6.4.1 Funcionamento

Retorna o valor da tangente do ângulo inserido.

#### 5.6.4.2 Entradas

##### **Ang(REAL):**

Valor real (em graus) do ângulo que se deseja calcular a tangente.

#### 5.6.4.3 Saídas

Retorna o valor da tangente do ângulo que foi inserido na entrada "Ang".

#### 5.6.4.4 Chamada da função

sfMathTan(Ang)

### 5.6.5 Arco Tangente - sfMathAtan

#### 5.6.5.1 Funcionamento

Retorna o valor do arco tangente do valor inserido.

#### 5.6.5.2 Entradas

##### **Value(REAL):**

Valor real que se deseja calcular o arco tangente.

#### 5.6.5.3 Saídas

Retorna o valor do arco tangente do número que foi inserido na entrada "REAL".

#### 5.6.5.4 Chamada da função

sfMathAtan(Value)

### 5.6.6 Valor da Constante Pi - sfMathPi

#### 5.6.6.1 Funcionamento

Retorna o valor da constante matemática PI.

#### 5.6.6.2 Entradas

Nessa função não há entrada.

#### 5.6.6.3 Saídas

Retorna o valor da constante matemática PI.

#### 5.6.6.4 Chamada da função

sfMathPi()

### 5.6.7 Valor Inteiro - sfMathTrunc

#### 5.6.7.1 Funcionamento

Retorna o valor inteiro de um número. Exemplo: TRUNC(35.33) retornará 35.

#### 5.6.7.2 Entradas

##### **Value(REAL):**

Valor real do qual se deseja remover as casas decimais.

#### 5.6.7.3 Saídas

Retorna o valor inteiro do número que foi inserido na entrada "REAL".

#### 5.6.7.4 Chamada da função

sfMathTrunc(Value)

### 5.6.8 Valor Fracionário - sfMathFrac

#### 5.6.8.1 Funcionamento

Retorna o valor da parte fracionária de um número real. Exemplo: FRAC(35.33) retornará 0.33.

#### 5.6.8.2 Entradas

##### **Value(REAL):**

Valor real do qual se deseja obter a parte fracionária.

#### 5.6.8.3 Saídas

Retorna um número REAL apenas com a parte fracionária do número que foi inserido na entrada "REAL".

#### 5.6.8.4 Chamada da função

FRAC(45.354)

### 5.6.9 Raiz Quadrada - sfMathSqrt

#### 5.6.9.1 Funcionamento

Cálcula a raiz quadrada do número inserido.

#### 5.6.9.2 Entradas

##### **Value(REAL):**

Valor REAL do qual se deseja obter a raiz quadrada.

#### 5.6.9.3 Saídas

Retorna o valor da raiz quadrada do número que foi inserido na entrada "REAL".

#### 5.6.9.4 Chamada da função

SQRT(9)

### 5.6.10 Valor Absoluto - sfMathAbs

#### 5.6.10.1 Funcionamento

Retorna o módulo de um número real.

#### 5.6.10.2 Entradas

**Value(REAL):**

Valor REAL do qual se deseja obter o módulo.

#### 5.6.10.3 Saídas

Retorna o valor do módulo do número que foi inserido na entrada "REAL".

#### 5.6.10.4 Chamada da função

ABS(-3)

## 5.7 Conversões de tipos de dados

### 5.7.1 Booleano para Real - sfConvertBoolToReal

#### 5.7.1.1 Funcionamento

Converte uma variável do tipo BOOL para o tipo REAL.

#### 5.7.1.2 Entradas

##### **Value(BOOL):**

Valor do tipo BOOL que será convertido para o tipo REAL.

#### 5.7.1.3 Saídas

Retorna o valor booleano convertido para REAL.

#### 5.7.1.4 Chamada da função

sfConvertBoolToReal(Value)

### 5.7.6 Booleano para Inteiro - sfConvertBoolToInt

#### 5.7.6.1 Funcionamento

Converte uma variável do tipo BOOL para o tipo INT.

#### 5.7.6.2 Entradas

##### **Value(BOOL):**

Valor do tipo BOOL que será convertido para o tipo INT.

#### 5.7.6.3 Saídas

Retorna o valor booleano convertido para INT.

#### 5.7.6.4 Chamada da função

sfConvertBoolToInt(Value)

### 5.7.2 Real para Booleano - sfConvertRealToBool

#### 5.7.2.1 Funcionamento

Converte uma variável do tipo REAL para o tipo BOOL.

#### 5.7.2.2 Entradas

##### **Value(REAL):**

Valor do tipo REAL que será convertido para o tipo BOOL.

#### 5.7.2.3 Saídas

Retorna o valor real convertido para BOOL.

#### 5.7.2.4 Chamada da função

sfConvertRealToBool(Value)

### 5.7.3 Real para Inteiro - sfConvertRealToInt

#### 5.7.3.1 Funcionamento

Converte uma variável do tipo REAL para o tipo INT.

#### 5.7.3.2 Entradas

##### **Value(REAL):**

Valor do tipo REAL que será convertido para o tipo INT.

#### 5.7.3.3 Saídas

Retorna o valor real convertido para INT.

#### 5.7.3.4 Chamadas das funções

sfConvertRealToInt(Value)

### 5.7.4 Real para Inteiro sem sinal - sfConvertRealToUint

#### 5.7.4.1 Funcionamento

Converte uma variável do tipo REAL para o tipo UINT.

#### 5.7.4.2 Entradas

##### **Value(REAL):**

Valor do tipo REAL que será convertido para o tipo UINT.

#### 5.7.4.3 Saídas

Retorna o valor real convertido para UINT.

#### 5.7.4.4 Chamadas das funções

sfConvertRealToUint(Value)

### 5.7.5 Inteiro sem sinal para Inteiro - sfConvertUintToInt

#### 5.7.5.1 Funcionamento

Converte uma variável do tipo UINT para o tipo INT.

#### 5.7.5.2 Entradas

##### **Value(UINT):**

Valor do tipo UINT que será convertido para o tipo INT.

#### 5.7.5.3 Saídas

Retorna o valor UINT convertido para INT.

#### 5.7.5.4 Chamada da função

sfConvertUintToInt(Value)

### 5.7.6 Inteiro para Inteiro sem sinal - sfConvertIntToUint

#### 5.7.6.1 Funcionamento

Converte uma variável do tipo INT para o tipo UINT.

#### 5.7.6.2 Entradas

##### **Value(INT):**

Valor do tipo INT que será convertido para o tipo UINT.

#### 5.7.6.3 Saídas

Retorna o valor INT convertido para UINT.

#### 5.7.6.4 Chamada da função

sfConvertIntToUint(Value)

### 5.7.7 Inteiro para Booleano - sfConvertIntToBool

#### 5.7.7.1 Funcionamento

Converte uma variável do tipo INT para o tipo BOOL.

#### 5.7.7.2 Entradas

##### **Value(INT):**

Valor do tipo INT que será convertido para o tipo BOOL.

#### 5.7.7.3 Saídas

Retorna o valor INT convertido para BOOL.

#### 5.7.7.4 Chamada da função

sfConvertIntToBool(Value)

### 5.7.8 Inteiro para Real - sfConvertIntToReal

#### 5.7.8.1 Funcionamento

Converte uma variável do tipo INT para o tipo REAL.

#### 5.7.8.2 Entradas

**Value(INT):**

Valor do tipo INT que será convertido para o tipo REAL.

#### 5.7.8.3 Saídas

Retorna o valor INT convertido para REAL.

#### 5.7.8.4 Chamada da função

sfConvertIntToReal(Value)

## 5.8 Eixos

### 5.8.1 Eixo Habilitado - `sfAxisEnabled`

#### 5.8.1.1 Funcionamento

A função retorna um valor booleano indicando se o eixo indicado está habilitado ou não.

#### 5.8.1.2 Entradas

##### **Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

#### 5.8.1.3 Saídas

Retorna um booleano que indica se o eixo está habilitado.

#### 5.8.1.4 Chamada da função

`sfAxisEnabled (Axis)`

### 5.8.2 Eixo Habilitado para Movimento Manual - `sfAxisEnabledForManualMove`

#### 5.8.2.1 Funcionamento

A função retorna um valor booleano indicando se o eixo indicado está habilitado para movimento manual ou não.

#### 5.8.2.2 Entradas

##### **Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

#### 5.8.2.3 Saídas

Retorna um booleano que indica se o eixo está habilitado para movimento manual.

#### 5.8.2.4 Chamada da função

`sfAxisEnabledForManualMove(Axis)`

### 5.8.3 Eixo em Movimento - `sfAxisInMove`

#### 5.8.3.1 Funcionamento

A função retorna um valor booleano indicando se o eixo indicado está em movimento ou não.

#### 5.8.3.2 Entradas

##### **Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

#### 5.8.3.3 Saídas

Retorna um booleano que indica se o eixo está em movimento.

#### 5.8.3.4 Chamada da função

`sfAxisInMove(Axis)`

### 5.8.4 Eixo Definido - `sfAxisIsDefined`

#### 5.8.4.1 Funcionamento

A função retorna um valor booleano indicando se o eixo indicado está definido nos parâmetros ou não.

#### 5.8.4.2 Entradas

##### **Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

#### 5.8.4.3 Saídas

Retorna um booleano que indica se o eixo está definido nos parâmetros.

#### 5.8.4.4 Chamada da função

`sfAxisIsDefined(Axis)`

### 5.8.5 Eixo é Spindle - sfAxisIsSpindle

#### 5.8.5.1 Funcionamento

A função retorna um valor booleano indicando se o eixo indicado está definido como spindle ou não.

#### 5.8.5.2 Entradas

**Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

#### 5.8.5.3 Saídas

Retorna um booleano que indica se o eixo está definido como spindle.

#### 5.8.5.4 Chamada da função

sfAxisIsSpindle(Axis)

### 5.8.6 Eixo está Referenciado - sfAxisReferenced

#### 5.8.6.1 Funcionamento

Esta função permite informar ao CNC se o eixo está referenciado ou ler seu estado de referenciamento.

#### 5.8.6.2 Entradas

**Axis(INT):**

Número do eixo que se quer verificar. Aceita valores de 1 a 32.

**Status(BOOL):**

Estado de referenciamento que se deseja atribuir ao eixo (caso a entrada "Type" seja FALSE).

**Type(BOOL):**

TRUE: Lê o estado de referenciamento do eixo.

FALSE: Define novo estado de referenciamento do eixo.

#### 5.8.6.3 Saídas

Retorna um booleano que indica se o eixo está referenciado (se a entrada "Type" for TRUE). Caso a entrada "Type" seja FALSE, a função define um novo estado de referenciamento para o eixo e retornará sempre FALSE.

#### 5.8.6.4 Chamada da função

sfAxisReferenced(Axis, Status, Type)

### 5.8.7 Executar Preset - sfPresetAxis

#### 5.8.7 .1 Funcionamento

Esta função executa um preset de coordenada no eixo indicado na entrada "Axis".

#### 5.8.7 .2 Entradas

**Axis(INT):**

Número do eixo no qual se quer fazer o preset.

**Coordinate (REAL):**

Valor de coordenada para o qual se quer fazer o preset do eixo.

**Origin(INT):**

Podemos fazer este preset em 6 origens diferentes (valores de 54 a 59). Pode-se visualizar estas origens entrando na tela de "Origens" do CNC AKC-PPC (objeto 361 de telas do CNC).

**Preset(BOOL):**

O preset da coordenada é executado no eixo e na origem selecionada quando acontecer uma transição de subida nesta entrada (transição de FALSE para TRUE).

#### 5.8.7 .3 Saídas

Esta função retorna um booleano que fica TRUE por um ciclo de PLC, indicando que o preset foi executado.

#### 5.8.7 .4 Chamada da função

sfPresetAxis(Coordinate, Preset, Origin, Axis)



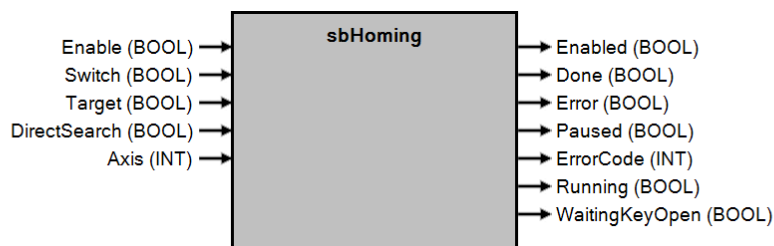
## 6.0 Blocos de Sistema

Blocos funcionam como variáveis: é necessário instanciá-los.

A "instância" é um objeto, ou seja, uma cópia da estrutura principal. Por isso, deve-se criar uma variável com um nome definido pelo usuário e selecionar o tipo desta variável de acordo com o bloco funcional que se deseja instanciar.

### 6.1 Eixos

#### 6.1.1 Referenciamento - sbHoming



##### 6.1.1.1 Funcionamento

Este bloco tem como função indicar ao CNC uma posição definida que será utilizada pelo CNC como referência para o eixo indicado.

No sistema de programação ISO (G-Codes) este dado está relacionado a origem G53.

Os parâmetros P09 e P10 do eixo indicado definem a posição de referência.

O Parâmetro P36 define o tipo marca de referência, para usar este bloco deve estar selecionado "PLC".

O Parâmetro P37 define a polaridade da marca de referência.

O Parâmetro P39 define o sentido do movimento que o eixo irá buscar referência.

O Parâmetro P40 define a velocidade do movimento na buscar referência.

O Parâmetro P41 define a velocidade reversa do movimento na buscar referência.

O procedimento de referenciamento deve anteceder qualquer tipo de operação automática ou manual que envolva movimentos para posições absolutas.

Para utilizar os limites de curso de software do CNC, definido nos parâmetros P48 ao P51 do eixo indicado, este procedimento também deve ter sido concluído.

##### 6.1.1.2 Entradas

###### **Enable (BOOL):**

Esta entrada habilita/inicia o processo de busca de referência do eixo indicado, quando receber o valor TRUE.

A variável de sistema *svHomingMode* é colocada em TRUE durante este processo.

Para cancelar o processo de busca de referência coloque o valor FALSE nesta entrada.

###### **Switch (BOOL):**

Esta entrada recebe o sinal de referência que o bloco irá aguardar quando iniciar o processo e o eixo movimentar.

Esta entrada normalmente será vinculada a uma entrada digital do drive ou cartão.

###### **Target (BOOL):**

Esta entrada define se a busca de referência será feita através de alvos.

Se o valor desta entrada for FALSE o CNC fará processo normal de busca de referência, movimentando o eixo até que a entrada Switch detecte o ponto de referência.

Se o valor desta entrada for TRUE o CNC assume que os eixos já estão posicionados na posição de referência, portanto não haverá nenhum movimento.

**DirectSearch (BOOL):**

Esta entrada define o sentido em que a busca de referência vai procurar a borda de descida do sinal de referência. Se o valor desta entrada for FALSE ao encontrar o sinal de referência o CNC inverte o sentido do movimento, com isso o eixo se movimentará no sentido oposto até que suma o sinal de referência.

Se o valor desta entrada for TRUE ao encontrar o sinal de referência o CNC continua o movimento no mesmo sentido até que suma o sinal de referência.

**Axis (REAL):**

Esta entrada define o número do eixo que irá buscar referência (1 a 32).

**6.1.1.3 Saídas****Enabled (BOOL):**

Esta saída indica se o bloco está habilitado caso tenha sido inicializado sem erros.

**Running (BOOL):**

Esta saída indica se o CNC está executando o processo de busca de referência no eixo definido.

**Paused (BOOL):**

Esta saída indica se o processo de busca de referência pausado.

**Error (BOOL):**

Esta saída indica se ocorreu algum erro durante o processo de busca de referência.

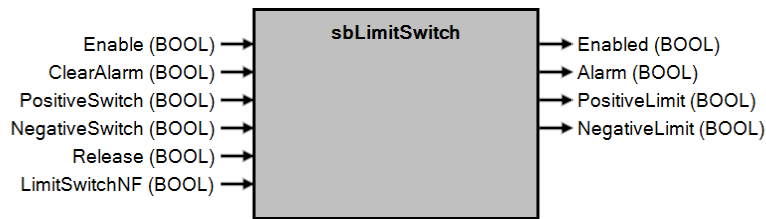
**ErrorCode (INT):**

A saída "ErrorCode" indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.1.1.4 Chamada do bloco**

sbHoming(Enable, Switch, Target, DirectSearch, Axis)

### 6.1.2 Fim de Curso - sbLimitSwitch



#### 6.1.2.1 Funcionamento

Este bloco tem a função de indicar se existe algum fim de curso físico acionado.

Caso um dos limites seja acionado após a inicialização do CNC, a saída "Alarm" também é acionada.

#### 6.1.2.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **LimitSwitchNF (BOOL):**

Esta entrada define a polaridade na detecção do fim de curso.

Se o valor desta entrada for TRUE a identificação de falhas de fim de curso ocorrerá quando as entradas "NegativeSwitch" ou "PositiveSwitch" estiverem na condição FALSE.

Se o valor desta entrada for FALSE a identificação de falhas de fim de curso ocorrerá quando as entradas "NegativeSwitch" ou "PositiveSwitch" estiverem na condição TRUE.

##### **NegativeSwitch (BOOL):**

Esta entrada deve ser atribuída a entrada digital que está conectada ao sensor de fim de curso negativo.

##### **PositiveSwitch (BOOL):**

Esta entrada deve ser atribuída a entrada digital que está conectada ao sensor de fim de curso positivo.

##### **Release (BOOL):**

Esta entrada define se todos os erros gerados por este bloco serão ignorados.

##### **ClearAlarm (BOOL):**

Esta entrada remove a indicação de erro da saída.

A transição do estado FALSE para TRUE da entrada "ClearAlarm" remove a indicação de erro da saída "Alarm".

Caso a condição que causou o alarme ainda esteja ativa, ClearAlarm não removerá a indicação de erro.

#### 6.1.2.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Alarm (BOOL):**

Esta saída indica se existe algum fim de curso acionado, esta saída só é usada quando o CNC estiver inicializado. Se o CNC não estiver inicializado, mesmo com algum fim de curso acionado, esta saída permanecerá no estado FALSE.

##### **NegativeSwitch (BOOL):**

Esta saída indica se o fim de curso negativo está acionado.

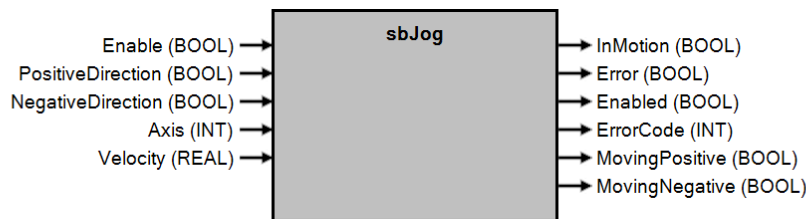
##### **PositiveSwitch (BOOL):**

Esta saída indica se o fim de curso positivo está acionado.

#### 6.1.2.4 Chamada do bloco

sbLimitSwitch(Enable, ClearAlarm, PositiveSwitch, NegativeSwitch, Release, LimitSwitchNF)

### 6.1.3 JOG - sbJog



#### 6.1.3.1 Funcionamento

A função deste bloco é o de realizar movimentos com velocidade controlada no eixo definido nos dois sentidos, enquanto a entrada "NegativeDirection" ou a entrada "PositiveDirection" estiverem em TRUE.

Se o CNC estiver referenciado e os limites de curso de software estiverem programados, ao utilizar este bloco o CNC respeitará esses limites e não deixará o eixo passar dos limites de curso.

Se o CNC não estiver referenciado e for necessário

Normalmente utilizamos este bloco com suas entradas de movimento conectadas a entradas digitais, algum botão ou softkey.

#### 6.1.3.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para executar movimento manual do eixos quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **Axis (INT):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

##### **NegativeDirection (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido negativo do eixo definido na entrada "Axis". O movimento permanece sendo executado enquanto este argumento estiver TRUE. Ao assumir o status FALSE, o movimento é interrompido.

##### **PositiveDirection (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido positivo do eixo definido na entrada "Axis". O movimento permanece sendo executado enquanto este argumento estiver TRUE. Ao assumir o status FALSE, o movimento é interrompido.

##### **Velocity (INT):**

Esta entrada define a velocidade com que o movimento será executado.

Se esta entrada for definida com o valor 0, a velocidade utilizada para os movimentos será a velocidade definida no parâmetro P27 do eixo definido.

#### 6.1.3.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **InMotion (BOOL):**

Esta saída indica se o eixo definido na entrada "Axis" está em movimento.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

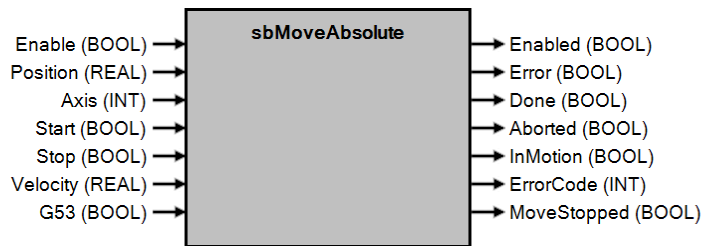
##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.1.3.4 Chamada do bloco

sbJog(Enable, PositiveDirection, NegativeDirection, Axis, Velocity)

### 6.1.4 Movimento Absoluto - sbMoveAbsolute



#### 6.1.4.1 Funcionamento

A função deste bloco é realizar um movimento com velocidade controlada no eixo definido na entrada "Axis" da posição atual do eixo até a posição definida na entrada "Position".

#### 6.1.4.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **Axis (INT):**

Esta entrada define o número do eixo que se quer movimentar.

##### **Position (REAL):**

Esta entrada define a posição absoluta final onde será posicionado o eixo definido.

##### **Start (BOOL):**

Esta entrada solicita ao bloco iniciar o movimento para a posição indicada em "Position".

##### **Stop (BOOL):**

Esta entrada solicita ao bloco parar o movimento que estiver sendo executado.

##### **Velocity (INT):**

Esta entrada define a velocidade com que será executado o movimento.

Se esta entrada for definida com o valor 0, a velocidade utilizada para os movimentos será a velocidade definida no parâmetro P27 do eixo definido.

##### **G53 (BOOL):**

Esta entrada define se o movimento será feito em relação a origem G53.

Se esta entrada for FALSE o movimento será feito em relação a origem selecionada (G54, G55, G56 ou G57)

Se esta entrada for TRUE o movimento será feito em relação a origem G53.

#### 6.1.4.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado e pronto para o uso.

##### **MoveStopped (BOOL):**

Esta saída indica se o movimento que estava sendo executado foi cancelado.

##### **Aborted (BOOL):**

Esta saída indica se um movimento que estava sendo executado foi interrompido. Esta saída é ativada após a saída o cancelamento do movimento.

##### **Done (BOOL):**

Esta saída indica se o movimento solicitado foi concluído.

##### **InMotion (BOOL):**

Esta saída indica se o eixo definido está em movimento.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante a execução do bloco.

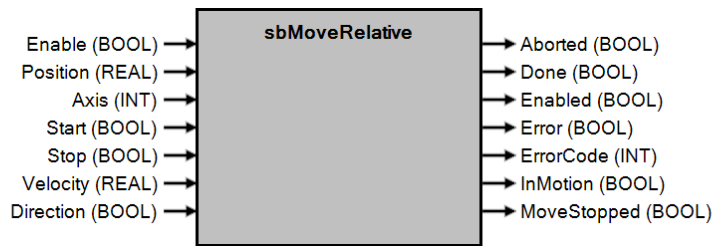
***ErrorCode (INT):***

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.1.4.4 Chamada do bloco**

sbMoveAbsolute(Enable, Position, Axis, Start, Stop, Velocity, G53)

### 6.1.5 Movimento Incremental - sbMoveRelative



#### 6.1.5.1 Funcionamento

A função deste bloco é realizar um movimento com velocidade controlada no eixo definido na entrada "Axis" da posição atual do eixo até a posição definida na entrada "Position".

#### 6.1.5.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **Axis (INT):**

Esta entrada define o número do eixo que se quer movimentar.

##### **Position (REAL):**

Esta entrada define o valor do incremento que será feito na posição atual do eixo. Este valor pode ser positivo para incrementar a posição, ou negativo para decrementar a posição atual.

##### **Direction (BOOL):**

Esta entrada define como o incremento será feito.

Se esta entrada for FALSE este o incremento utilizado será o definido na entrada "Position".

Se esta entrada for TRUE e o incremento definido na entrada "Position" for positivo, o movimento será feito para o sentido negativo.

##### **Start (BOOL):**

Esta entrada solicita iniciar o movimento para incrementar a posição.

##### **Stop (BOOL):**

Esta entrada solicita parar o movimento que estiver sendo executado.

##### **Velocity (INT):**

Esta entrada define a velocidade que o movimento será executado.

Se esta entrada for definida com o valor 0, a velocidade utilizada para os movimentos será a velocidade definida no parâmetro P27 do eixo definido.

#### 6.1.5.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **MoveStopped (BOOL):**

Esta saída indica se o movimento que estava sendo executado foi cancelado.

##### **Aborted (BOOL):**

Esta saída indica se um movimento que estava sendo executado foi interrompido. Esta saída é ativada após a saída o cancelamento do movimento.

##### **Done (BOOL):**

Esta saída indica se o movimento solicitado foi concluído.

##### **InMotion (BOOL):**

Esta saída indica se o eixo associado ao "Axis" está em movimento.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

- **ErrorCode (INT):**

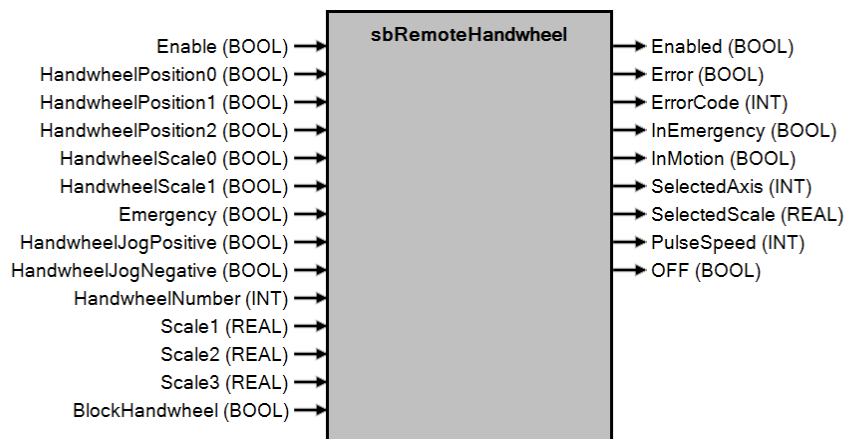
Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### **6.1.5.4 Chamada do bloco**

sbMoveRelative(Enable, Position, Axis, Start, Stop, Velocity, Direction)



### 6.1.6 Manivela Remota - sbRemoteHandwheel



#### 6.1.6.1 Funcionamento

A função deste bloco é controlar a manivela remota disponível ao CNC.

A manivela remota é um hardware vendido separadamente do CNC.

Com a manivela remota o CNC realiza movimentos incrementais ou de JOG de acordo com a escala selecionada.

A manivela remota possui um botão de emergência.

O seletor de eixo tem 8 posições, permitindo selecionar até 7 eixos.

O seletor de escala tem 4 posições, 3 valores de escala de incremento e JOG.

O CNC pode utilizar até 3 manivelas.

As manivelas remotas devem ser configuradas nos parâmetros gerais.

A configuração da Manivela remota 1 deve ser feita nos parâmetros gerais P36 a P39.

A configuração da Manivela remota 2 deve ser feita nos parâmetros gerais P40 a P43.

A configuração da Manivela remota 3 deve ser feita nos parâmetros gerais P44 a P47.

#### 6.1.6.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco, se este sinal for "FALSE" a manivela fica desabilitada.

##### **HandwheelPosition0 (BOOL):**

Esta entrada define o bit 0 da seleção de eixo da manivela remota. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

##### **HandwheelPosition1 (BOOL):**

Esta entrada define o bit1 da seleção de eixo da manivela remota. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

##### **HandwheelPosition2 (BOOL):**

Esta entrada define o bit 2 da seleção de eixo da manivela remota. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

##### **HandwheelScale0 (BOOL):**

Esta entrada define o bit 0 da seleção de escala da manivela remota. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

##### **HandwheelScale1 (BOOL):**

Esta entrada define o bit 1 da seleção de escala da manivela remota. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

##### **Emergency (BOOL):**

Esta entrada define o botão de emergência da manivela remota deve ser associada a esta entrada do bloco. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC.

**HandwheelJogPositive (BOOL):**

Esta entrada define o botão de movimento positivo deve ser associada a esta entrada do bloco. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC. Esta entrada realiza movimentos de JOG ou incrementais, dependendo do tipo de escala que estiver selecionada.

**HandwheelJogNegative (BOOL):**

Esta entrada define o botão de movimento negativo deve ser associada a esta entrada do bloco. Esta entrada utilizará uma entrada digital do CNC que será associada a ela no PLC. Esta entrada realiza movimentos de JOG ou incrementais, dependendo do tipo de escala que estiver selecionada.

**HandwheelNumber (INT):**

Esta entrada define o número da manivela remota (1 a 3, como mencionado anteriormente).

**Scale1 (REAL):**

Esta entrada define o valor do incremento que será utilizado na escala 1.

**Scale2 (REAL):**

Esta entrada define o valor do incremento que será utilizado na escala 2.

**Scale3 (REAL):**

Esta entrada define o valor do incremento que será utilizado na escala 3.

**BlockManWheel (BOOL):**

A entrada "BlockManWheel" é um booleano responsável por bloquear o funcionamento da manivela.

**6.1.6.3 Saídas****Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

**InEmergency (BOOL):**

Esta saída indica se o botão de emergência da manivela está pressionado.

**InMotion (BOOL):**

Esta saída indica se o eixo selecionado está em movimento.

**OFF (BOOL):**

Esta saída indica se nenhum eixo está selecionado na manivela remota.

**PulseSpeed (INT):**

Esta saída indica a velocidade da manivela remota em pulsos por amostragem a cada 16ms.

**SelectedAxis (INT):**

Esta saída indica o eixo que está selecionado.

**SelectedScale (REAL):**

Esta saída indica o valor da escala que está selecionada.

**Error (BOOL):**

Esta saída indica se ocorreu um erro durante a execução do bloco.

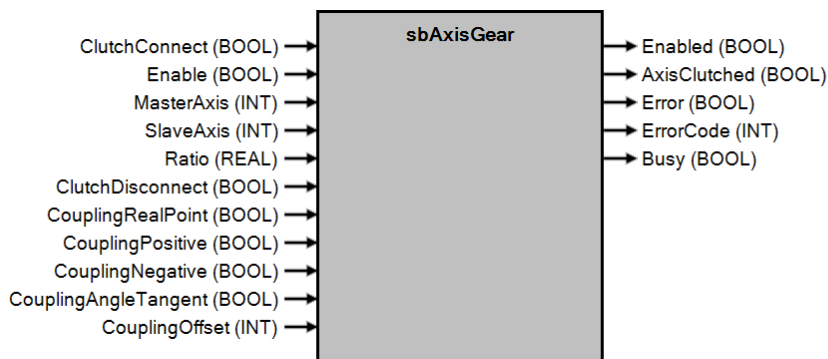
**ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.1.6.4 Chamada do bloco**

sbRemoteHandwheel(Enable, HandwheelPosition0, HandwheelPosition1, HandwheelPosition2, HandwheelScale0, HandwheelScale1, Emergency, HandwheelJogPositive, HandwheelJogNegative, HandwheelNumber, Scale1, Scale2, Scale3, BlockHandwheel)

### 6.1.7 Acoplar Eixo - sbAxisGear



#### 6.1.7.1 Funcionamento

Este bloco tem a função de fazer o acoplamento de um eixo escravo a um eixo mestre.

Com isso pode-se fazer um eixo escravo seguir os movimentos do mestre com uma relação de acoplamento definida. A relação de acoplamento entre os eixos pode ser positiva ou negativa. Se for positiva, o eixo escravo segue o eixo mestre no mesmo sentido do movimento do mestre. Valores negativos fazem o eixo escravo se movimentar no sentido inverso ao do movimento do eixo mestre.

O módulo da relação de acoplamento determina a relação entre o movimento do eixo mestre e do eixo escravo. Se a relação de acoplamento for maior que 0 e menor que 1 faz o eixo escravo se movimentar menos do que o movimento do mestre. Por exemplo, para relação igual a 0.5, para cada milímetro de movimento do eixo mestre, o escravo irá mover 0.5 mm.

Se a relação de acoplamento for igual a 1 o eixo escravo se movimentará na mesma proporção que o eixo mestre. Se a relação de acoplamento for maior que 1 o eixo escravo irá se movimentar mais do que o mestre. Por exemplo, para relação igual a 2, para cada milímetro de movimento do eixo mestre, o escravo irá mover 2 mm.

#### 6.1.7.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **ClutchConnect (BOOL):**

Esta entrada solicita ao CNC o acoplamento do eixo escravo ao mestre levando em consideração todos os valores atribuídos ao restante das entradas.

##### **ClutchDisconnect (BOOL):**

Esta entrada solicita ao CNC o desacoplamento do eixo escravo ao mestre, deixando-o liberado para movimentos independentes aos do mestre.

##### **CouplingAngleTangent (BOOL):**

Esta entrada define se o acoplamento do eixo escravo ao mestre será feito de forma tangencial.

Ao solicitar um acoplamento, se a entrada "CouplingAngleTangent" estiver em TRUE, o eixo escravo seguirá tangente à trajetória no plano definido pelo CNC via funções ISO G17 (plano XY), G18 (plano YZ) ou G19 (plano ZX), ou seja, no caso do plano XY sendo o eixo C (rotativo) o eixo escravo, caso o XY execute um movimento fazendo um círculo, o eixo C faria um movimento tangente à circunferência descrita por XY.

##### **CouplingNegative (BOOL):**

Esta entrada define se o acoplamento do eixo escravo ao mestre será feito somente quando o mestre se deslocar no sentido negativo.

##### **CouplingPositive (BOOL):**

Esta entrada define se o acoplamento do eixo escravo ao mestre será feito somente quando o mestre se deslocar no sentido positivo.

***CouplingOffset (INT):***

Esta entrada define se o CNC fará um deslocamento que será executado no eixo escravo em relação ao eixo mestre. Este movimento é executado apenas uma vez a cada solicitação de acoplamento.

***CouplingRealPoint (BOOL):***

Esta entrada define se o CNC fará o acoplamento pelo ponto real, ou seja, é acoplado a um encoder, resolver ou outro transdutor de posição.

Quando solicitado o acoplamento com este argumento em FALSE o acoplamento é feito pelo ponto teórico do CNC, ou seja, é acoplado ao ponto calculado pelo CNC.

***MasterAxis (INT):***

Esta entrada define o número do eixo correspondente ao eixo mestre.

***SlaveAxis (INT):***

Esta entrada define o número do eixo correspondente ao eixo mestre.

***Ratio (REAL):***

Esta entrada define o fator de acoplamento entre o escravo e o mestre.

Caso se acople um eixo linear a um eixo rotativo cada um grau corresponde a 0.0001 mm.

**6.1.7.3 Saídas*****Enabled (BOOL):***

Esta saída indica se o bloco está habilitado.

***AxisClutched (BOOL):***

Esta saída indica se o eixo escravo está acoplado ao respectivo eixo mestre.

***Error (BOOL):***

Esta saída indica se ocorreu um erro durante o processamento do bloco.

***ErrorCode (INT):***

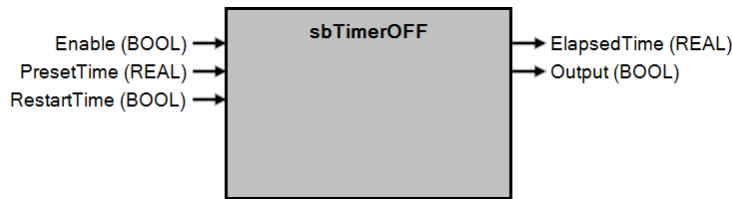
Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.1.7.4 Chamada do bloco**

sbAxisGear(ClutchConnect, Enable, MasterAxis, SlaveAxis, Ratio, ClutchDisconnect, CouplingRealPoint, CouplingPositive, CouplingNegative, CouplingAngleTangent, CouplingOffset)

## 6.2 Temporizadores

### 6.2.1 Temporizador (Normalmente Desligado) - sbTimerOFF



#### 6.2.1.1 Funcionamento

Este bloco tem a função de fazer a contagem do tempo e ativar uma saída ao final da contagem.

Quando o bloco é habilitado, a saída "Output" muda seu estado para TRUE e fica nesta condição pela duração de tempo definida em "PresetTime". Após esse tempo, a saída vai para o estado FALSE.

Para efetuar uma nova contagem de tempo mantenha o bloco habilitado e envie um pulso de pelo menos um ciclo na entrada "RestartTime". A contagem reiniciará quando a entrada "RestartTime" voltar para o estado FALSE.

#### 6.2.1.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **PresetTime (REAL):**

Esta entrada define o tempo para ser contado no temporizador, em milissegundos.

##### **RestartTime (BOOL):**

Esta entrada reinicia a contagem de tempo.

#### 6.2.1.3 Saídas

##### **ElapsedTime (REAL):**

Esta saída indica o tempo decorrido na contagem atual em milissegundos.

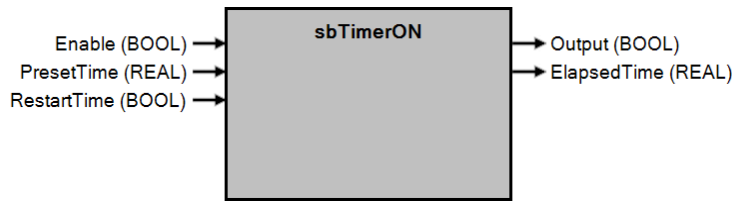
##### **Output (BOOL):**

Esta saída indica o status da contagem de tempo.

#### 6.2.1.4 Chamada do bloco

sbTimerOFF(Enable, PresetTime, RestartTime)

## 6.2.2 Temporizador (Normalmente Ligado) - sbTimerON



### 6.2.2.1 Funcionamento

Este bloco tem a função de fazer a contagem do tempo e ativar uma saída ao final da contagem.

Quando o bloco é habilitado, a saída "Output" muda seu estado para FALSE e fica nesta condição pela duração de tempo definida em "PresetTime". Após esse tempo, a saída vai para o estado TRUE.

Para efetuar uma nova contagem de tempo mantenha o bloco habilitado e envie um pulso de pelo menos um ciclo na entrada "RestartTime". A contagem reiniciará quando a entrada "RestartTime" voltar para o estado FALSE.

### 6.2.2.2 Entradas

#### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

#### **PresetTime (REAL):**

Esta entrada define o tempo para ser contado no temporizador, em milissegundos.

#### **RestartTime (BOOL):**

Esta entrada reinicia a contagem de tempo.

### 6.2.2.3 Saídas

#### **ElapsedTime (REAL):**

Esta saída indica o tempo decorrido na contagem atual em milissegundos.

#### **Output (BOOL):**

Esta saída indica o status da contagem de tempo.

### 6.2.2.4 Chamada do bloco

sbTimerON(Enable, PresetTime, RestartTime)

### 6.2.3 Temporizador (Piscante) - sbTimerONOFF



#### 6.2.3.1 Funcionamento

Este bloco tem a função de fazer a contagem do tempo e ativar uma saída ciclicamente.

Quando o bloco é habilitado, a saída "Output" permanece em TRUE durante o tempo definido na entrada "PresetTimeON".

Ao término desse tempo a saída vai para FALSE e permanece assim pela durante o tempo definido na entrada "PresetTimeOFF".

A contagem é cíclica até que a entrada "Enable" seja desabilitada (FALSE).

#### 6.2.3.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **PresetTimeON (REAL):**

Esta entrada define o tempo em milissegundo para ser contado no temporizador com a saída "Output" em TRUE.

##### **PresetTimeOFF (REAL):**

Esta entrada define o tempo em milissegundo para ser contado no temporizador com a saída "Output" em FALSE.

#### 6.2.3.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **ElapsedTime (REAL):**

Esta saída indica o tempo decorrido em milissegundos. Mostra o tempo que já se passou a partir do momento em que o temporizador iniciou a contagem. O valor vai de 0 até PresetTimeOFF + PresetTimeON e em seguida é zerado novamente.

##### **Output (BOOL):**

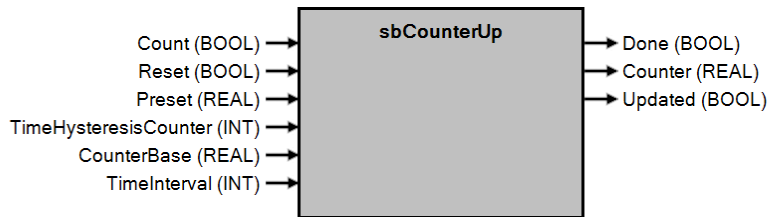
Esta saída indica o status da contagem de tempo.

#### 6.2.3.4 Chamada do bloco

sbTimerONOFF(Enable, PresetTimeOn, PresetTimeOff)

## 6.3 Contadores

### 6.3.1 Contador Crescente - sbCounterUp



#### 6.3.1.1 Funcionamento

Este bloco tem a função de fazer contagem de um valor numérico de forma crescente.

Ao receber uma transição de subida na entrada "Count" a saída "Counter" será incrementada com o valor definido na entrada "CounterBase".

Caso a entrada "TimeInterval" esteja em zero, ao manter a entrada "Count" em TRUE, o contador não é incrementado novamente, ou seja, a contagem fica apenas na transição de subida da entrada "Count".

Caso "TimeInterval" esteja com valor diferente de 0, o valor é incrementado automaticamente no tempo dado em "TimeInterval" se a entrada "Count" estiver em TRUE.

Quando o valor da saída "Counter" for maior ou igual ao da entrada "Preset" a saída "Done" irá para TRUE.

Mesmo após a saída "Done" ser igual a TRUE, se a entrada "Count" for acionada, o valor de "Counter" continua sendo incrementado até que o bloco receba TRUE na entrada "Reset".

#### 6.3.1.2 Entradas

##### **Count (BOOL):**

Esta entrada executa a contagem. Quando esta entrada receber uma transição de subida a saída "Counter" será incrementada do valor definido na entrada "CounterBase".

##### **Reset (BOOL):**

Esta entrada reinicia o contador, ou seja, a saída "Counter" vai para o valor 0.

##### **Preset (REAL):**

Esta entrada define o valor alvo da contagem. Ao atingir este valor a saída "Done" vai para o estado TRUE.

##### **TimeHysteresisCounter (INT):**

Esta entrada define o tempo de histerese em milissegundos para a entrada "Count". Este tempo é o atraso para iniciar o incremento automático se a entrada "Count" permanecer em TRUE.

##### **CounterBase (REAL):**

Esta entrada define o valor base que será usado para incrementar o contador.

##### **TimeInterval (INT):**

Esta entrada define o intervalo de tempo em milissegundos para executar incrementos automáticos se a entrada "Count" fica com valor TRUE. Caso este argumento seja zero, o incremento automático não acontece.

#### 6.3.1.3 Saídas

##### **Done (BOOL):**

Esta saída indica se a saída "Counter" atingiu o valor definido na entrada "Preset".

##### **Counter (REAL):**

Esta saída indica o valor atual da contagem. Mesmo após atingir o valor da entrada "Preset" a contagem continua, se o bloco não for reiniciado através da entrada "Reset".

##### **Updated (BOOL):**

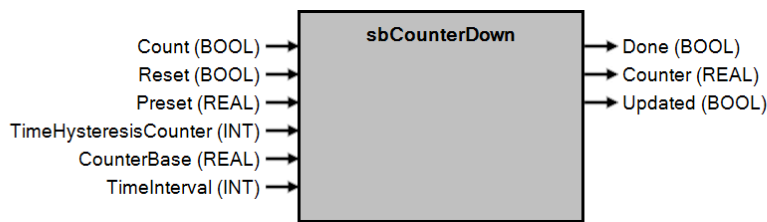
Indica que o valor da saída "Counter" foi atualizado. Permanece ativo por um ciclo de PLC.

#### 6.3.1.4 Chamada do bloco

sbCounterUp(Count, Reset, Preset, TimeHysteresisCounter, CounterBase, TimeInterval)



### 6.3.2 Contador Decrescente - sbCounterDown



#### 6.3.2.1 Funcionamento

Este bloco tem a função de fazer contagem de um valor numérico de forma decrescente.

Ao receber uma transição de subida na entrada "Count" a saída "Counter" será decrementada com o valor definido na entrada "CounterBase".

Caso a entrada "TimeInterval" esteja em zero, ao manter a entrada "Count" em TRUE, o contador não é decrementado novamente, ou seja, a contagem fica apenas na transição de subida da entrada "Count".

Caso "TimeInterval" esteja com valor diferente de 0, o valor é decrementado automaticamente no tempo dado em "TimeInterval" se a entrada "Count" estiver em TRUE.

Quando o valor da saída "Counter" for menor ou igual a zero, a saída "Done" irá para TRUE.

Mesmo após a saída "Done" ser igual a TRUE, se a entrada "Count" for acionada, o valor de "Counter" continua sendo decrementado até que o bloco receba TRUE na entrada "Reset".

#### 6.3.2.2 Entradas

##### **Count (BOOL):**

Esta entrada executa a contagem. Quando esta entrada receber uma transição de subida a saída "Counter" será decrementada do valor definido na entrada "CounterBase".

##### **Reset (BOOL):**

Esta entrada reinicia o contador, ou seja, a saída "Counter" recebe o valor definido na entrada "Preset".

##### **Preset (REAL):**

Esta entrada define o valor alvo da contagem. Ao atingir este valor a saída "Done" vai para o estado TRUE.

##### **TimeHysteresisCounter (INT):**

Esta entrada define o tempo de histerese em milissegundos para a entrada "Count". Este tempo é o atraso para iniciar o decremento automático se a entrada "Count" permanecer em TRUE.

##### **CounterBase (REAL):**

Esta entrada define o valor base que será usado para decrementar o contador.

##### **TimeInterval (INT):**

Esta entrada define o intervalo de tempo em milissegundos para executar decrementos automáticos se a entrada "Count" fica com valor TRUE. Caso este argumento seja zero, o decremento automático não acontece.

#### 6.3.2.3 Saídas

##### **Done (BOOL):**

Esta saída indica se a saída "Counter" atingiu o valor zero.

##### **Counter (REAL):**

Esta saída indica o valor atual da contagem. Mesmo após atingir o valor da entrada "Preset" a contagem continua, se o bloco não for reiniciado através da entrada "Reset".

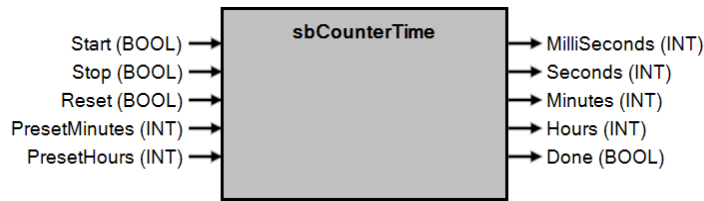
##### **Updated (BOOL):**

Indica que o valor da saída "Counter" foi atualizado. Permanece ativo por um ciclo de PLC.

#### 6.3.2.4 Chamada do bloco

sbCounterDown(Count, Reset, Preset, TimeHysteresisCounter, CounterBase, TimeInterval)

### 6.3.3 Contador de Tempo - sbCounterTime



#### 6.3.3.1 Funcionamento

Este bloco tem a função de fazer contagem de tempo, como em um cronômetro.

Uma transição de subida na entrada "Start" inicia a contagem de tempo, mostrando nas saídas do bloco os tempos em milissegundos, segundos, minutos e horas.

Uma transição de subida na entrada "Stop" faz a contagem ser pausada. Para reiniciá-la, o bloco deve receber uma nova transição de subida na entrada "Start".

Uma transição de subida na entrada "Reset", reinicia a contagem de tempo de todas as saídas.

Quando o valor das entradas de preset (minutos e horas) forem alcançados, a saída "Done" mudará para o estado TRUE. Mesmo após essas entradas serem atingidas, o tempo continuará sendo contado (O contador só parará quando receber uma transição de subida na entrada "Stop").

O tempo total de contagem é determinado pela combinação das 4 saídas de tempo (milissegundos, segundos, minutos e horas).

#### 6.3.3.2 Entradas

##### **Start (BOOL):**

Esta entrada inicia a contagem de tempo.

##### **Stop (BOOL):**

Esta entrada interrompe a contagem de tempo.

##### **Reset (BOOL):**

Esta entrada reinicia a contagem de tempo.

##### **PresetMinutes (INT):**

Esta entrada define o valor alvo para minutos.

##### **PresetHours (INT):**

Esta entrada define o valor alvo para horas.

#### 6.3.3.3 Saídas

##### **MilliSeconds (INT):**

Esta saída indica o tempo decorrido em milissegundos (0-999).

##### **Seconds (INT):**

Esta saída indica o valor decorrido em segundos (0-59).

##### **Minutes (INT):**

Esta saída indica o valor decorrido em minutos (0-59).

##### **Hours (INT):**

Esta saída indica o valor decorrido em horas (0-1092).

##### **Done (BOOL):**

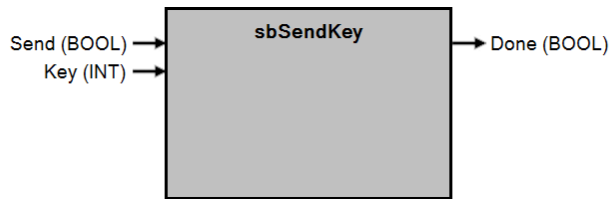
Esta saída indica se os valores programados nas entradas "PresetMinutes" e "PresetHours" foram atingidos.

#### 6.3.4.4 Chamada do bloco

sbCounterTime(Start, Stop, Reset, PresetMinutes, PresetHours)

## 6.4 Códigos e Softkeys

### 6.4.1 Enviar Código - sbSendKey



#### 6.4.1.1 Funcionamento

Este bloco tem a função de enviar o código definido na entrada "Key" ao CNC. O código pode ser um código de telas (foco), caracteres (ASCII ou não ASCII), de modo de operação ou qualquer outro código de teclas do CNC. Se o código for o de uma tela, o CNC irá atualizar a tela atual. Se for um modo, irá atualizar o modo atual de operação. Se for uma tecla (ASCII ou não ASCII), é como se aquela tecla específica tivesse sido pressionada no CNC.

#### 6.4.1.2 Entradas

##### **Key (INT):**

Esta entrada define o código a ser enviado para o CNC. Pode ser um código de telas (foco), caracteres (ASCII ou não ASCII), de modo de operação ou qualquer outro código do CNC.

##### **Send (BOOL):**

Esta entrada executa o envio do código ao CNC.

Na transição de subida desta entrada, o código definido na entrada "Key" será enviado ao CNC.

#### 6.4.1.3 Saídas

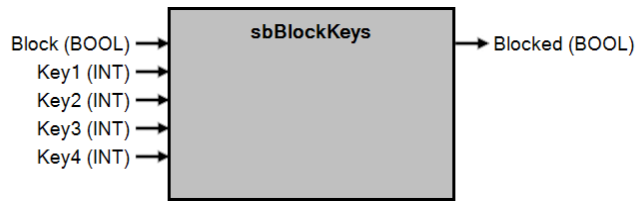
##### **Done (BOOL):**

Esta saída indica se o sinal foi enviado ao CNC.

#### 6.4.1.4 Chamada do bloco

sbSendKey(Send, Key)

### 6.4.2 Bloquear Código - sbBlockKeys



#### 6.4.2.1 Funcionamento

Este bloco tem a função de bloquear o recebimento dos códigos definidos nas entradas "Key1", "Key2", "Key3" e "Key4" ao CNC. O código pode ser um código de telas (foco), caracteres (ASCII ou não ASCII), de modo de operação ou qualquer outro código de teclas do CNC.

Quando se quiser bloquear 3 teclas ou menos, basta colocar as demais entradas "Key" em 0.

#### 6.4.2.2 Entradas

##### **Key1 (INT):**

Esta entrada define o código da tecla que se deseja bloquear. (1)

##### **Key2 (INT):**

Esta entrada define o código da tecla que se deseja bloquear. (2)

##### **Key3 (INT):**

Esta entrada define o código da tecla que se deseja bloquear. (3)

##### **Key4 (INT):**

Esta entrada define o código da tecla que se deseja bloquear. (4)

##### **Block (BOOL):**

Esta entrada bloqueia as teclas dos códigos que foram definidos nas entradas "Key1", "Key2", "Key3" e "Key4". O bloqueio estará ativo quando o valor desta entrada for TRUE,

#### 6.4.2.3 Saídas

##### **Blocked (BOOL):**

Esta saída indica se o bloco está bloqueando os códigos que foram definidos nas entradas "Key1", "Key2", "Key3" e "Key4".

#### 6.4.2.4 Chamada do bloco

sbBlockKeys(Block, Key1, Key2, Key3, Key4)

### 6.4.3 Enviar Estado de Softkeys - sbSendState



#### 6.4.3.1 Funcionamento

Este bloco altera o estado atual das softkeys do CNC - tanto horizontais quanto verticais.

#### 6.4.3.2 Entradas

##### **HorizontalState (INT):**

Esta entrada define o número do estado de softkeys horizontais que será selecionado.

##### **VerticalState (INT):**

Esta entrada define o número do estado de softkeys verticais que será selecionado.

##### **Send (BOOL):**

Esta entrada solicita a mudança de estado das softkeys.

Uma transição de subida nessa entrada altera o estado atual das softkeys do CNC para os estados definidos nas entradas "HorizontalState" e "VerticalState".

#### 6.4.3.3 Saídas

##### **Done (BOOL):**

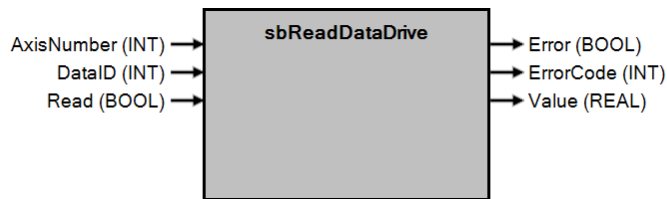
Esta saída indica se o bloco foi executado, com isso os estados das softkeys mudaram para os estados definidos nas entradas "HorizontalState" e "VerticalState".

#### 6.4.3.4 Chamada do bloco

sbSendState(HorizontalState, VerticalState, Send)

## 6.5 Drives

### 6.5.1 Ler Dados do Drive - sbReadDataDrive



#### 6.5.1.1 Funcionamento

Este bloco tem a função de fazer a leitura de informações dos drives AKD.

É possível ler posição, velocidade e torque.

Para ler uma destas informações, basta informar o número do drive na entrada "AxisNumber", o tipo da informação em "DataID" e colocar a entrada "Read" em TRUE. Em seguida, o valor é retornado na saída "Value".

#### 6.5.1.2 Entradas

##### **AxisNumber (INT):**

Esta entrada define o número do drive que se deseja ler a informação. Para saber o número do drive, verificar o parâmetro 12 (Número do drive EtherCAT / Canal Analógico) do eixo correspondente.

##### **DataID (INT):**

Esta entrada define o tipo de dado que se deseja ler. Os possíveis valores são:

0 = Posição

1 = Velocidade

2 = Torque

##### **Read (BOOL):**

Esta entrada solicita a leitura do dado.

#### 6.5.1.3 Saídas

##### **Value (REAL):**

Esta saída retorna o valor do dado solicitado.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

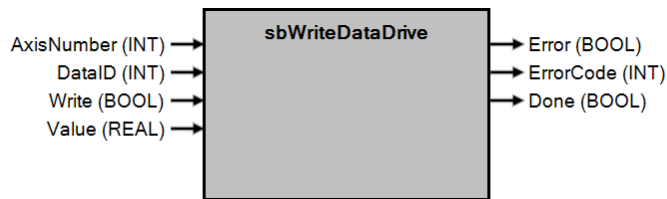
##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.5.1.4 Chamada do bloco

sbReadDataDrive(AxisNumber, DataID, Read)

## 6.5.2 Escrever Dados do Drive - sbWriteDataDrive



### 6.5.2.1 Funcionamento

Este bloco tem a função de fazer a escrita de informações nos drives AKD.

Atualmente, só é possível escrever o valor da saída analógica do drive.

Para escrever esta informação, basta definir o número do drive na entrada "AxisNumber", o ID da informação em "DataID", atribuir o novo valor à entrada "Value" e colocar a entrada "Write" em TRUE.

### 6.5.2.2 Entradas

#### AxisNumber (INT):

Esta entrada define o número do drive que se deseja escrever a informação. Para saber o número do drive, verificar o parâmetro 12 (Número do drive EtherCAT / Canal Analógico) do eixo correspondente.

#### DataID (INT):

Esta entrada define o tipo de dado que se deseja escrever. Os possíveis valores são:

0 = Saída analógica do drive

#### Write (BOOL):

Esta entrada solicita a escrita do dado.

#### Value (REAL):

Esta entrada define o valor que será escrito no drive.

### 6.5.2.3 Saídas

#### Done (BOOL):

Esta saída indica se a solicitação para escrever no drive foi enviada.

#### Error (BOOL):

Esta saída indica se ocorreu um erro durante o processamento do bloco.

#### ErrorCode (INT):

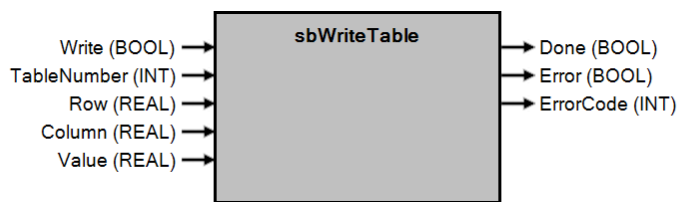
Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

### 6.5.2.4 Chamada do bloco

sbWriteDataDrive(AxisNumber, DataID, Write, Value)

## 6.6 Acesso a Tabelas

### 6.6.1 Escrever no Campo da Tabela - sbWriteTable



#### 6.6.1.1 Funcionamento

Este bloco tem a função de fazer a escrita de dados em tabelas do CNC.

Para escrever um valor em uma célula da tabela, o usuário deve informar nas entradas do bloco o número da tabela, a coluna e a linha onde se quer escrever o novo valor. Também deve-se informar o valor que será escrito nessa determinada célula.

#### 6.6.1.2 Entradas

##### **Write (BOOL):**

Esta entrada executa a escrita na tabela quando receber uma transição de subida.

##### **TableNumber (BOOL):**

Esta entrada define o número da tabela onde se deseja escrever o dado.

##### **Row (INT):**

Esta entrada define o número da linha da tabela.

##### **Column (INT):**

Esta entrada define o número da coluna da tabela.

##### **Value (REAL):**

Esta entrada define o valor a ser escrito na célula escolhida.

#### 6.6.1.3 Saídas

##### **Done (BOOL):**

Esta saída indica se foi feita a escrita na tabela.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

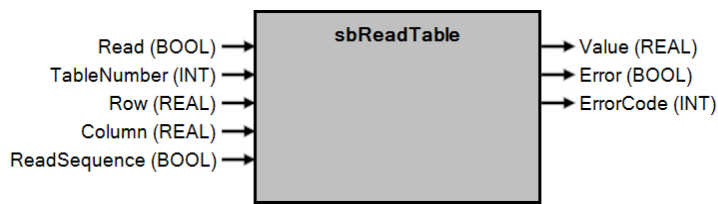
Esta saída indica o tipo do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.6.1.4 Chamada do bloco

sbWriteTable(Write, TableNumber, Row, Column, Value)



## 6.6.2 Ler Campo da Tabela - ReadTable



### 6.6.2.1 Funcionamento

Este bloco tem a função de fazer a leitura de dados em tabelas do CNC.

Para ler um valor em uma célula da tabela, o usuário deve informar nas entradas do bloco o número da tabela, a coluna e a linha onde se quer ler o valor.

### 6.6.2.2 Entradas

#### **Read (BOOL):**

Esta entrada executa a leitura na tabela quando receber uma transição de subida.

Caso se deseje ler o valor de uma outra posição da tabela ou de outra tabela, deve-se fornecer outra borda de subida nesse argumento.

Caso se deseje ler valores sequenciais em uma mesma tabela, há a opção de manter os valores de "Read" e "ReadSequence" em TRUE (ver explicação da entrada "ReadSequence").

#### **TableNumber (INT):**

Esta entrada define o número da tabela de onde se deseja ler o(s) dado(s).

#### **Row (REAL):**

Esta entrada define o número da linha da tabela.

#### **Column (REAL):**

Esta entrada define o número da coluna da tabela.

#### **ReadSequence (BOOL):**

Esta entrada define se a leitura a ser feita será de modo sequencial.

Ao se atribuir TRUE a esta entrada e TRUE a entrada "Read", o retorno de valores lidos se torna incremental.

No primeiro ciclo de PLC, o bloco retornará o valor da célula especificada nas entradas. Nos ciclos seguintes, o bloco incrementa o número da linha de onde se vai ler. Ao se atingir a última linha, o bloco incrementa o número da coluna e repete o processo. Quando se atinge o fim da tabela, o bloco retorna ao primeiro valor da tabela e o processo continua até que a leitura sequencial seja interrompida quando "ReadSequence" é colocado em FALSE.

### 6.6.2.3 Saídas

#### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

#### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### **Value (REAL):**

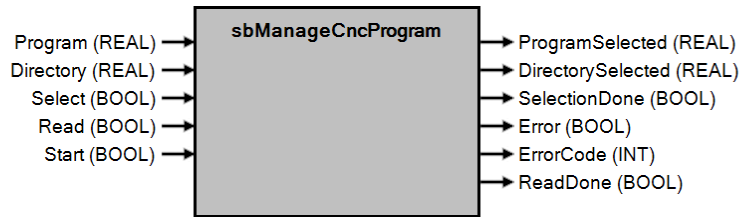
Esta saída retorna o valor lido da célula indicada. O valor é atualizado sempre após o "Read" sofrer transição de subida ou uma vez a cada ciclo de PLC caso "ReadSequence" esteja em "TRUE".

### 6.6.2.4 Chamada do bloco

sbReadTable(Read, TableNumber, Row, Column, ReadSequence)

## 6.7 Programas CNC

### 6.7.1 Controle de Programa do CNC - sbManageCncProgram



#### 6.7.1.1 Funcionamento

Este bloco tem a função de selecionar um programa no CNC e executá-lo ou ler qual programa está selecionado atualmente.

Para selecionar um novo programa, é necessário informar na entrada "Directory" o número do diretório onde o programa está localizado, informar o número do programa na entrada "Program" e colocar a entrada "Select" em TRUE.

Para retornar o programa atual selecionado no CNC, deve-se colocar a entrada "Read" em TRUE. O número do programa atual é retornado na saída "ProgramSelected" e seu diretório, na saída "DirectorySelected".

Nesse caso não é necessário informar os valores de "Directory" e "Program".

Para executar o programa selecionado, basta fornecer uma transição de subida na entrada "Start".

#### 6.7.1.2 Entradas

##### **Select (BOOL):**

Esta entrada seleciona um programa no CNC.

##### **Read (BOOL):**

Esta entrada solicita a leitura do programa que está selecionado no CNC.

##### **Program (REAL):**

Esta entrada define o número do programa que se quer selecionar no CNC.

##### **Directory (REAL):**

Esta entrada define o número do diretório que se quer selecionar o programa. Caso se deseje selecionar um programa que está na raiz do CNC, colocar 0 neste parâmetro.

##### **Start (BOOL):**

Esta entrada inicia a execução do programa selecionado no CNC, ao receber uma transição de subida.

#### 6.7.1.3 Saídas

##### **ProgramSelected (REAL):**

Esta saída retorna o número do programa selecionado pelo CNC, no caso de uma solicitação de leitura.

##### **DirectorySelected (REAL):**

Esta saída retorna o número do diretório selecionado pelo CNC, no caso de uma solicitação de leitura.

##### **SelectionDone (BOOL):**

Esta saída indica se o programa foi selecionado.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

##### **ReadDone (BOOL):**

Esta saída indica se a leitura foi concluída.

#### 6.7.1.4 Chamada do bloco

sbManageCncProgram(Program, Directory, Select, Read, Start);

## 6.8 Cloud

### 6.8.1 Ler Dados da Nuvem - sbReadCloud



#### 6.8.1.1 Funcionamento

Esse bloco tem a função de fazer a leitura de dados que estão disponíveis no serviço Cloud do CNC. As informações podem ser lidas passando o ID correspondente na entrada "ID" e colocando a entrada "Read" em TRUE. Quando a entrada "Read" é colocada em TRUE, o bloco faz a leitura da variável correspondente e disponibiliza o valor na saída "Value".

#### 6.8.1.2 Entradas

##### **Read (BOOL):**

Esta entrada executa a leitura da variável.

##### **ID (INT):**

Esta entrada define o tipo de dado que se deseja ler. Os possíveis valores são:

0 - Estado da máquina (0 - Idle, 1 - Preparação, 2 - Produzindo, 3 - Emergência, 4 - Manutenção)

1 - Total de peças/sacolas programado

2 - Total de peças/sacolas produzidos

3 - Número do pedido

4 - Tempo gasto por unidade

5 - Programa selecionado

6 - Diretório do programa selecionado (0 - Programs, 1 - SUB, 2 - CYC)

7 - Performance - % exec

8 - Performance - % plc

9 - Performance - % motion

10 - Performance - % ethercat

11 - Potência (Consumo)

#### 6.8.1.3 Saídas

##### **Done (BOOL):**

Esta saída indica se o bloco foi executado e o valor está disponível na saída "Value".

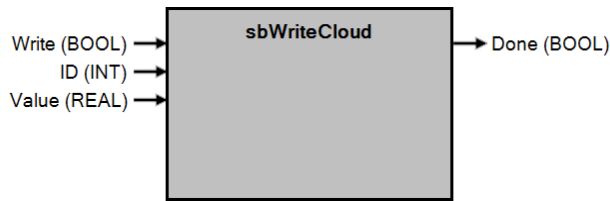
##### **Value (REAL):**

Esta saída retorna o valor lido, correspondente ao tipo de dado definido na entrada "ID".

#### 6.8.1.4 Chamada do bloco

sbReadCloud(Read, ID)

### 6.8.2 Escrever Dados na Nuvem- sbWriteCloud



#### 6.8.2.1 Funcionamento

Esse bloco tem a função de fazer a escrita de dados nas variáveis disponíveis no serviço Cloud do CNC. As informações podem ser escritas passando o ID correspondente na entrada "ID", o valor na entrada "Value" e colocando a entrada "Write" em TRUE.

#### 6.8.2.2 Entradas

##### **Write (BOOL):**

Esta entrada executa a escrita do valor na variável.

##### **ID (INT):**

Esta entrada define o tipo de dado que se deseja escrever. Os possíveis valores são:

- 0 - Estado da máquina (0 - Idle, 1 - Preparação, 2 - Produzindo, 3 - Emergência, 4 - Manutenção)
- 1 - Total de peças/sacolas programado
- 2 - Total de peças/sacolas produzidos
- 3 - Número do pedido
- 4 - Tempo gasto por unidade
- 5 - Programa selecionado
- 6 - Diretório do programa selecionado (0 - Programs, 1 - SUB, 2 - CYC)
- 7 - Performance - % exec
- 8 - Performance - % plc
- 9 - Performance - % motion
- 10 - Performance - % ethercat
- 11 - Potencia (Consumo)

##### **Value (Real):**

Esta entrada define o valor que se deseja escrever na variável especificada na entrada "ID".

#### 6.8.2.3 Saídas

##### **Done (BOOL):**

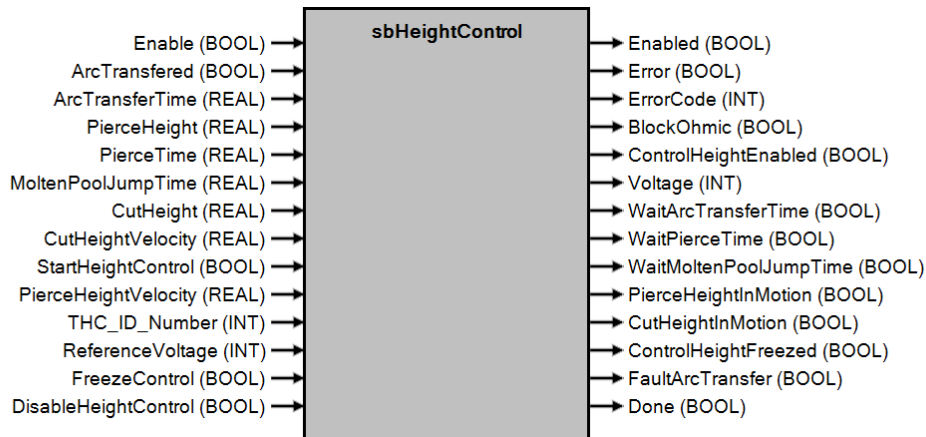
Esta saída indica se o bloco foi executado.

#### 6.8.2.4 Chamada do bloco

sbWriteCloud(Write, ID, Value)

## 6.9 THC

### 6.9.1 Controle de Altura - sbHeightControl



#### 6.9.1.1 Funcionamento

Este bloco tem a função de executar o controle do THC utilizado em máquinas de corte plasma, onde um eixo perpendicular ao plano de corte ajusta sua altura dependendo das ondulações da chapa. Esta correção é feita por um feedback analógico, ou seja, é passada ao CNC uma referência analógica correspondente à tensão de arco para que a correção da altura seja executada.

Uma vez habilitado o bloco, é solicitado iniciar o controle de altura. O processo consiste em:

- Aguardar o tempo para transferência de arco, em que se espera o retorno da informação de arco transferido (ArcTransferred) após iniciar a ignição da fonte plasma;
- Posicionar o eixo na altura de perfuração com velocidade controlada (PierceHeight, PierceHeightVelocity);
- Contar tempo de perfuração (PierceTime);
- Contar tempo de pula borra (MoltenPoolJumpTime);
- Posicionar na altura de corte com velocidade controlada (CutHeight, CutHeightVelocity);
- Assim que posicionado na altura de corte, iniciar o controle de altura.

#### Parâmetros de máquina

Existem alguns parâmetros de máquina que precisam ser programados para que este bloco funcione corretamente. Estes parâmetros devem ser programados na aba "PLC" da tela de parâmetros do CNC. Neste caso, o usuário deve adicionar os parâmetros na aplicação que estiver sendo desenvolvida.

P960 = Número do eixo associado ao THC1

P961 = Número do eixo associado ao THC2

P962 = Número do eixo associado ao THC3

P963 = Número do eixo associado ao THC4

P964 = Entrada analógica associada ao feedback analógico do controle de altura do THC1

P965 = Entrada analógica associada ao feedback analógico do controle de altura do THC2

P966 = Entrada analógica associada ao feedback analógico do controle de altura do THC3

P967 = Entrada analógica associada ao feedback analógico do controle de altura do THC4

P975 = janela de reação: Faixa de tensão em torno da tensão de controle dentro da qual o controle de altura é congelado automaticamente. O centro da faixa de controle é a tensão de controle.

P976 = janela de controle: Faixa de tensão em que o THC é efetivamente controlado. Fora desta janela, o controle de altura do THC fica congelado. A janela de controle deve ser obrigatoriamente maior do que a janela de reação.

P977 = Ganho proporcional para malha do controle de altura (feedback analógico).

P978 = Ganho integral para malha do controle de altura (feedback analógico).

P979 = Ganho diferencial para malha do controle de altura (feedback analógico).

P981 = Tensão de fundo de escala, ou seja, tensão na tocha correspondente a 10 volts na entrada analógica do THC.

P984 = Porcentagem de tensão para detectar mudanças bruscas na tensão de arco. Caso haja um aumento súbito na tensão equivalente ao valor deste parâmetro, o controle de altura do THC é congelado.

### 6.9.1.2 Entradas

**Enable (BOOL):**

Esta entrada habilita o bloco. Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

**StartHeightControl (BOOL):**

Esta entrada inicializa o processo de habilitação do controle de altura ao receber uma transição de subida.

**ArcTransferTime (REAL):**

Esta entrada define o tempo em segundos para aguardar a transferência de arco (ArcTransferred). Caso o arco não seja transferido até que este tempo se encerre, o processo é abortado e é indicado erro (FaultArcTransfer).

**ArcTransferred (BOOL):**

Esta entrada recebe o sinal aguardado durante o tempo de transferência de arco (ArcTransferTime). Durante este tempo, é esperado que este sinal passe para estado lógico TRUE.

**PierceHeight (REAL):**

Esta entrada define a posição absoluta onde será posicionado o THC para iniciar a perfuração da chapa.

**PierceHeightVelocity (REAL):**

Esta entrada define a velocidade com que o CNC posiciona o THC na posição de perfuração.

**PierceTime (REAL):**

Esta entrada define o tempo em que o eixo fica parado na posição de perfuração antes de ir para o próximo passo do processo.

**MoltenPoolJumpTime (REAL):**

Esta entrada define o tempo para pular borra. O controle de altura permanece na altura de perfuração pelo tempo estabelecido neste argumento.

**CutHeight (REAL):**

Esta entrada define a posição absoluta onde será posicionado o THC para que se inicie o corte.

**CutHeightVelocity (REAL):**

Esta entrada define a velocidade com que o CNC posiciona o THC para posição de corte.

**DisableHeightControl (BOOL):**

Esta entrada desabilita o controle de altura.

Na transição de subida é solicitado ao bloco que desabilite o controle de altura, ou seja, o CNC deixa de olhar o feedback analógico para corrigir a altura de corte.

**FreezeControl (BOOL):**

Esta entrada solicita ao bloco que congele a altura atual do THC, ou seja, na transição de subida o bloco mantém o THC na altura em que se encontrava antes dessa função ser solicitada. Esta função só é ativada caso o controle de altura esteja habilitado.

**ReferenceVoltage (INT):**

Esta entrada define a tensão de correspondente à altura de corte programada (CutHeight). Quando o CNC posiciona o THC na altura de corte, espera-se que a tensão de arco corresponda ao valor deste argumento. Ao se posicionar na altura de corte o controle de altura espera que a tensão de arco assuma este valor.

**THC\_ID\_Number (INT):**

Esta entrada define o número do THC que se quer habilitar. O CNC está preparado para controlar até 4 unidades de controle de altura, ou seja, podemos programar valores de 1 a 4.

### 6.9.1.3 Saídas

**Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

**BlockOhmic:**

Esta saída indica se o toque ôhmico está bloqueado.

**ControlHeightEnabled (BOOL):**

Esta saída indica se o processo para habilitar o controle foi concluído e está operando.

**ControlHeightFreezed (BOOL):**

Esta saída indica se o controle de altura está congelado.

**CutHeightInMotion (BOOL):**

Esta saída indica se o CNC está posicionando o THC na altura de corte.

***PierceHeightInMotion (BOOL):***

Esta saída indica se o CNC está posicionando o THC na altura de perfuração.

***Voltage (INT):***

Esta saída indica a tensão recebida pelo feedback analógico.

***WaitArcTransferTime (BOOL):***

Esta saída indica se o tempo de transferência de arco passou.

***WaitMoltenPoolJumpTime (BOOL):***

Esta saída indica se o tempo do processo de pular borra passou.

***WaitPierceTime (BOOL):***

Esta saída indica se o tempo do processo de perfuração passou.

***FaultArcTransfer (BOOL):***

Esta saída indica se houve falha enquanto aguardava o tempo para transferência de arco.

***Error (BOOL):***

Esta saída indica se ocorreu um erro durante o processamento do bloco.

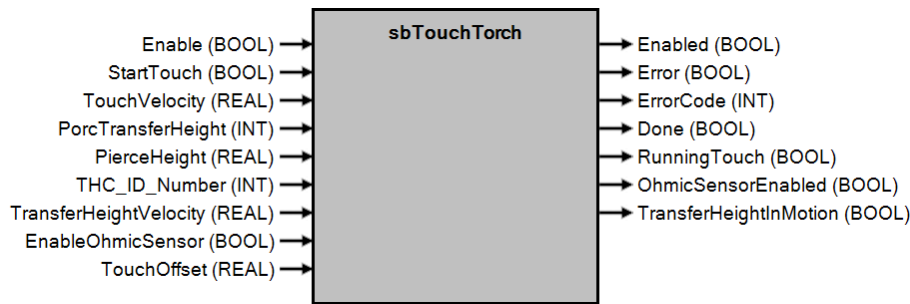
***ErrorCode (INT):***

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.9.1.4 Chamada do bloco**

sbHeightControl(Enable, ArcTransferred, ArcTransferTime, PierceHeight, PierceTime, MoltenPoolJumpTime, CutHeight, CutHeightVelocity, StartHeightControl, PierceHeightVelocity, THC\_ID\_Number, ReferenceVoltage, FreezeControl, DisableHeightControl)

## 6.9.2 Toque Ôhmico - sbTouchTorch



### 6.9.2.1 Funcionamento

Este bloco tem a função de executar a detecção de chapa e preset para zero na coordenada detectada.

O processo consiste em mover o eixo no sentido negativo até que se encontre a chapa. Após encontrá-la, é feito um preset de coordenada para zero e, em seguida, posiciona-se o eixo em uma altura de transferência.

A detecção da chapa pode se dar de duas formas: por sensor ôhmico ou por torque.

Quando for feita a detecção por sensor ôhmico, é feito o preset para zero assim que o sensor for acionado.

O sensor ôhmico é definidos nos parâmetros P968 (para THC 1), P969 (para THC 2), P970 (para THC 3) e P971 (para THC 4).

Quando for feita a detecção por torque, ao se detectar a chapa, a posição é somada a um offset (definido na entrada "TouchOffset") e então é feito o preset para zero.

O processo de detecção por torque é feito através da detecção de um atraso (lag - P974). Quando o atraso no processo de captura de chapa for maior que o definido no parâmetro, entende-se que a chapa foi encontrada.

A detecção por torque está sempre habilitada, enquanto que a por sensor ôhmico pode ser desabilitada colocando a entrada "EnableOhmicSensor" em FALSE. Este bloco é responsável por executar a detecção de chapa e preset para zero na coordenada detectada.

O processo consiste em mover o eixo no sentido negativo até que se encontre a chapa. Após encontrá-la, é feito um preset de coordenada para zero e, em seguida, posiciona-se o eixo em uma altura de transferência.

A detecção da chapa pode se dar de duas formas: por sensor ôhmico ou por torque.

Quando for feita a detecção por sensor ôhmico, é feito o preset para zero assim que o sensor for acionado.

O sensor ôhmico é definidos nos parâmetros P968 (para THC 1), P969 (para THC 2), P970 (para THC 3) e P971 (para THC 4).

Quando for feita a detecção por torque, ao se detectar a chapa, a posição é somada a um offset (definido na entrada "TouchOffset") e então é feito o preset para zero.

O processo de detecção por torque é feito através da detecção de um atraso (lag - P974). Quando o atraso no processo de captura de chapa for maior que o definido no parâmetro, entende-se que foi encontrada a chapa.

A detecção por torque está sempre habilitada, enquanto que a por sensor ôhmico pode ser desabilitada colocando a entrada "EnableOhmicSensor" em FALSE.

#### Parâmetros de máquina

P968 = Entrada digital associada ao sensor ôhmico do THC1

P969 = Entrada digital associada ao sensor ôhmico do THC2

P970 = Entrada digital associada ao sensor ôhmico do THC3

P971 = Entrada digital associada ao sensor ôhmico do THC4

P974 = Lag para detecção de toque por torque.

### 6.9.2.2 Entradas

#### **Enable (BOOL):**

Esta entrada habilita o bloco. Caso esta entrada seja FALSE, o bloco é desabilitado e todas as saídas recebem valores nulos.

#### **EnableOhmicSensor (BOOL):**

Esta entrada habilita a detecção de chapa por sensor ôhmico.



**PierceHeight (REAL):**

Esta entrada define a posição absoluta onde será posicionado o THC para iniciar a perfuração da chapa.

**PorcTransferHeight (INT):**

Esta entrada define a porcentagem de altura de transferência. Este argumento indica a porcentagem da altura de perfuração na qual será feito o posicionamento do eixo.

Exemplo: caso PorcTransferHeight seja 50 e PierceHeight seja 10, a altura de transferência será 50% de 10, ou seja, 5.

**StartTouch (BOOL):**

Esta entrada inicializa o processo de toque que faz a captura da chapa, ao receber uma transição de subida.

**TouchOffset (REAL):**

Esta entrada define o valor para ser somado à posição do eixo no momento em que a detecção por torque ocorrer.

**TouchVelocity (INT):**

Esta entrada define a velocidade do eixo durante a captura da chapa.

**TransferHeightVelocity (INT):**

Esta entrada define a velocidade com que o CNC posiciona o eixo para a posição de transferência.

**THC\_ID\_Number (INT):**

Esta entrada define o número do THC que se quer habilitar, existem 4 THCs possíveis (de 1 a 4).

**6.9.2.3 Saídas****Enabled (BOOL):**

Esta saída indica se o bloco está habilitado e pronto para o uso.

**Done (BOOL):**

Esta saída indica se o processamento do bloco foi concluído.

**OhmicSensorEnabled (BOOL):**

Esta saída indica se a detecção por sensor ôhmico está habilitada.

A captura deve ser desabilitada sempre que ocorrer uma situação em que o sensor possa ser acionado de forma incorreta, como em cortes embaixo de água.

**RunningTouch (BOOL):**

Esta saída indica se o processo de captura de chapa está em andamento (movimento no sentido negativo), mas chapa ainda não foi detectada.

**TransferHeightInMotion (BOOL):**

Esta saída indica que se o CNC está posicionando o eixo na posição de transferência.

**Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

**ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.9.2.4 Chamada do bloco**

sbTouchTorch(Enable, StartTouch, TouchVelocity, PorcTransferHeight, PierceHeight, THC\_ID\_Number, TransferHeightVelocity, EnableOhmicSensor, TouchOffset)

## 6.10 Controle

### 6.10.1 Filtro Digital - sbDigital Filter



#### 6.10.1.1 Funcionamento

Este bloco tem a função de filtrar um sinal analógico com base na fórmula:

Em qualquer sistema de controle com sinal de realimentação analógica presente, há o risco de ruídos que podem comprometer a integridade do sinal, produzindo um sistema menos robusto.

Este bloco é um filtro digital de primeira ordem. Ele filtra o sinal fornecido na entrada "Reference" e a intensidade da filtragem é selecionada na entrada "FilterGain".

#### 6.10.1.2 Entradas

**Reference (REAL):**

Esta entrada recebe o sinal que será filtrado.

**FilterGain (REAL):**

Esta entrada define o ganho utilizado para atenuar as oscilações do sinal de referência. Pode ser um valor de 0.05 até 1.00. Quanto menor, maior a atuação do filtro. Se esse valor for 1, o filtro está desabilitado e o valor da entrada é passado para a saída.

#### 6.10.1.3 Saídas

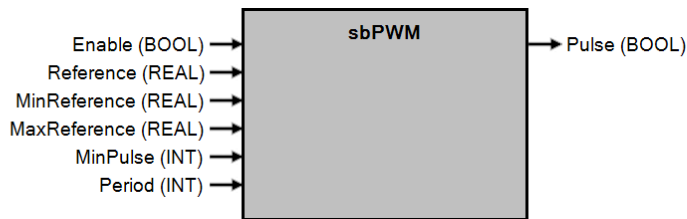
**FilterOutput (REAL):**

Esta saída é o sinal filtrado.

#### 6.10.1.4 Chamada do bloco

sbDigitalFilter(Reference, FilterGain)

### 6.10.2 PWM - sbPWM



#### 6.10.2.1 Funcionamento

Este bloco tem a função de gerar um sinal de PWM.

A modulação por largura de pulso, mais conhecida pela sigla em inglês PWM (Pulse-Width Modulation) é uma técnica de modulação usada para codificar uma mensagem em um sinal pulsante. Embora esta técnica de modulação possa ser utilizada para codificar informação para transmissão, a sua principal utilização é permitir o controle da potência fornecida a dispositivos elétricos, especialmente a cargas inerciais tais como motores e resistências de aquecimento.

A largura do pulso é calculada pela fórmula:

$$\text{Tempo}_{\text{ON}} = \text{Período} \times \text{Duty Cycle}$$

O termo Duty Cycle (ciclo de trabalho em português) descreve a proporção de tempo "ligado" em relação ao intervalo regular ou "período" de tempo: um ciclo de trabalho baixo corresponde a baixa potência, porque o fornecimento de potência está desligado a maior parte do tempo. O ciclo de trabalho é expresso em percentagem, sendo 100% totalmente ligado. O duty cycle pode ser calculado por:

$$\text{Duty cycle} = (\text{Referência} - \text{Referência}_{\text{MIN}}) / (\text{Referência}_{\text{MAX}} - \text{Referência}_{\text{MIN}})$$

Caso o tempo calculado para o pulso (ligado) seja menor que o valor definido na entrada "MinPulse", o valor calculado será substituído pelo tempo definido em "MinPulse" e o tempo complementar (desligado) será calculado por:

$$\text{Tempo}_{\text{OFF}} = (\text{MinPulse} / \text{Tempo}_{\text{ON Calculado}}) \times (\text{Período} - \text{Tempo}_{\text{ON Calculado}})$$

#### 6.10.2.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o cálculo da modulação. Manter ativo enquanto o cálculo for utilizado.

##### **Reference (REAL):**

Esta entrada recebe o sinal que em conjunto com as entradas de máximo e mínimo, definem a largura do pulso.

##### **MaxReference (REAL):**

Esta entrada define o valor máximo que a entrada de referência pode atingir.

##### **MinReference (REAL):**

Esta entrada define o valor mínimo que a entrada de referência pode atingir.

##### **MinPulse (INT):**

Esta entrada define o tempo mínimo permitido, em milissegundos, para tempo ligado.

##### **Period (INT):**

Esta entrada define o período do sinal, em milissegundos.

#### 6.10.2.3 Saídas

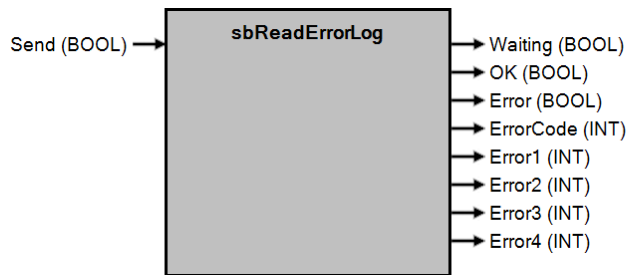
##### **Pulse (BOOL):**

Esta Saída é o PWM com ciclo baseado nos argumentos de entrada.

#### 6.10.2.4 Chamada do bloco

sbPWM(Enable, Reference, MinReference, MaxReference, MinPulse, Period)

### 6.10.1 Ler Erros Logados - sbReadErrorLog



#### 6.10.1.1 Entradas

Este bloco tem a função de retornar os 4 primeiros erros na lista de erros do CNC.

##### 6.10.1.1 Entradas

Send(bool)

##### **Send (BOOL):**

Esta entrada busca os 4 primeiros erros na lista de erros e envia para as respectivas saídas do bloco.

#### 6.10.1.2 Saídas

##### **Waiting (BOOL):**

Esta saída indica se o CNC está aguardando a leitura dos erros.

##### **OK (BOOL):**

Esta saída indica se a leitura dos erros foi feita.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

##### **Error1 (INT):**

Esta saída indica o primeiro erro da lista de erros do CNC.

##### **Error2 (INT):**

Esta saída indica o segundo erro da lista de erros do CNC.

##### **Error3 (INT):**

Esta saída indica o terceiro erro da lista de erros do CNC.

##### **Error4 (INT):**

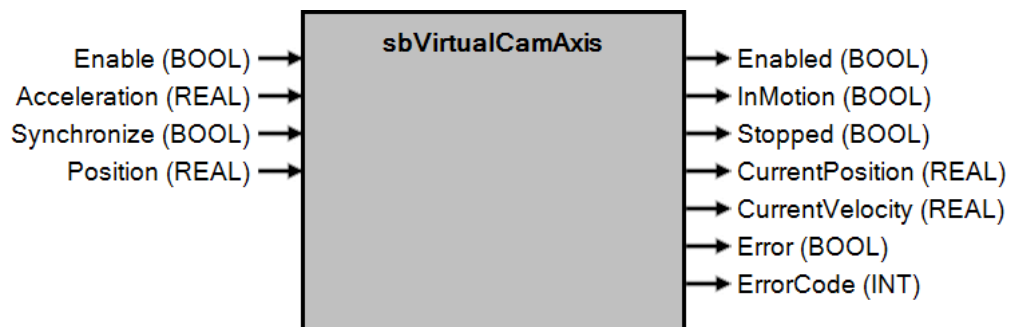
Esta saída indica o quarto erro da lista de erros do CNC.

#### 6.10.1.3 Chamada do bloco:

sbReadErrorLog(Send)

## 6.11 Came

### 6.11.1 Came Virtual - sbVirtualCamAxis



#### 6.11.1.1 Funcionamento

Este bloco tem a função de criar um came virtual, ou seja, um eixo virtual onde podemos associar funções, como acionamentos de saídas digitais por ângulo (came digital), vínculos a outros eixos para que em uma faixa angular estes outros eixos façam um deslocamento definido em tabela (came de eixos) e etc.

Só é possível ter uma instância deste bloco, ou seja, não é possível ter mais de um came virtual.

#### 6.11.1.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco. Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **Acceleration (INT):**

Esta entrada define o tempo de aceleração e desaceleração dos movimentos solicitados ao came em milissegundos.

##### **Synchronize (BOOL):**

Esta entrada solicita sincronizar o came virtual com o came real, o sincronismo é iniciado quando o came real passar pela posição definida em "Position". Após esta posição os comes virtual e real serão sincronizados.

##### **Position (REAL):**

Esta entrada define a posição absoluta onde será sincronizado o came real.

#### 6.11.1.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **InMotion (BOOL):**

Esta saída indica se o came virtual está em movimento.

##### **Stopped (BOOL):**

Esta saída indica se o movimento que estava sendo executado pelo came virtual foi interrompido.

##### **CurrentPosition (REAL):**

Esta saída indica a posição atual do came.

##### **CurrentVelocity (REAL):**

Esta saída indica a velocidade atual do came.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

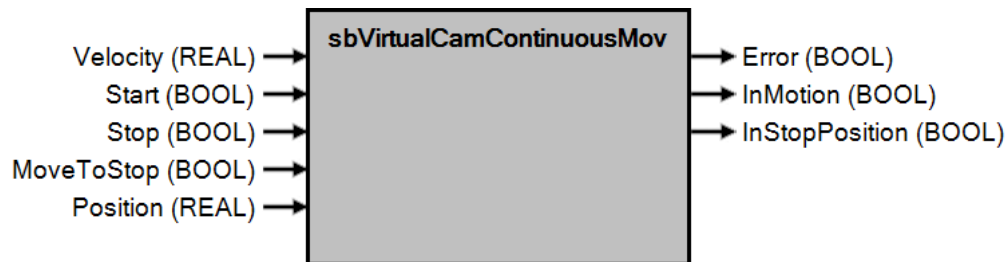
##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.1.4 Chamada do bloco:

sbVirtualCamAxis(Enable, Acceleration, Synchronize, Position)

### 6.11.2 Came Virtual Contínuo - sbVirtualCamContinuousMove



#### 6.11.2.1 Funcionamento

Este bloco tem a função de executar movimentos contínuos no came virtual com velocidade definida em "Velocity".

#### 6.11.2.2 Entradas

##### **Velocity (INT):**

Esta entrada define a velocidade com que serão executados os movimentos solicitados ao came (RPM).

##### **Start (BOOL):**

Esta entrada solicita iniciar o movimento contínuo do came virtual.

##### **Stop (BOOL):**

Esta entrada solicita a parada imediata do came, o came irá parar na posição em que estiver no momento da solicitação.

##### **MoveToStop (BOOL):**

Esta entrada solicita uma parada na posição definida em "Position".

##### **Position (REAL):**

Esta entrada define a posição absoluta onde será posicionado o came virtual ao ser solicitada uma parada "MoveToStop".

#### 6.11.2.3 Saídas:

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco. Verificar se existe uma instância habilitada do bloco "VirtualCamAxis" e se as entradas estão com valores válidos: a entrada "Position" entre 0 e 360 e a entrada "Velocity" diferente de 0.

Caso nenhuma das situações acima estejam inválidas, desabilite o bloco e habilite novamente.

##### **InMotion (BOOL):**

Esta saída indica se o came virtual está executando movimento contínuo ou o movimento até o ângulo de parada.

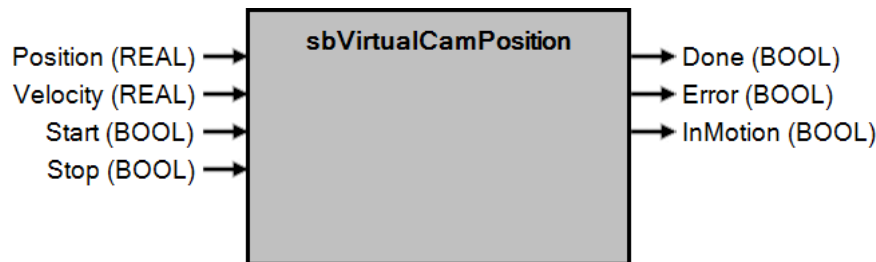
##### **InStopPosition (BOOL):**

Esta saída indica se o came virtual está parado.

#### 6.11.2.4 Chamada do bloco

sbVirtualCamContinuousMove(Velocity, Start, Stop, MoveToStop, Position)

### 6.11.3 Came Por Posição - sbVirtualCamPosition



#### 6.11.3.1 Funcionamento

Este bloco tem a função de controlar posicionamentos absolutos no came virtual, ele faz o posicionamento para a posição informada na entrada "Position" e na velocidade programada na entrada "Velocity".

#### 6.11.3.2 Entradas

##### **Position (REAL):**

Esta entrada define a posição absoluta onde será posicionado o came.

##### **Velocity (INT):**

Esta entrada define a velocidade com a qual o movimento será executado (RPM).

##### **Start (BOOL):**

Esta entrada solicita iniciar o movimento para a posição indicada em "Position".

##### **Stop (BOOL):**

Esta entrada solicita a parada imediata do came. O came irá parar na posição em que estiver no momento da solicitação.

#### 6.11.3.3 Saídas:

##### **Done (BOOL):**

Esta saída indica se movimento foi concluído.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

Verificar se existe uma instância habilitada do bloco "VirtualCamAxis" e se as entradas estão com valores válidos: a entrada "Position" entre 0 e 360 e a entrada "Velocity" diferente de 0.

Caso nenhuma das situações acima estejam inválidas, desabilite o bloco e habilite novamente.

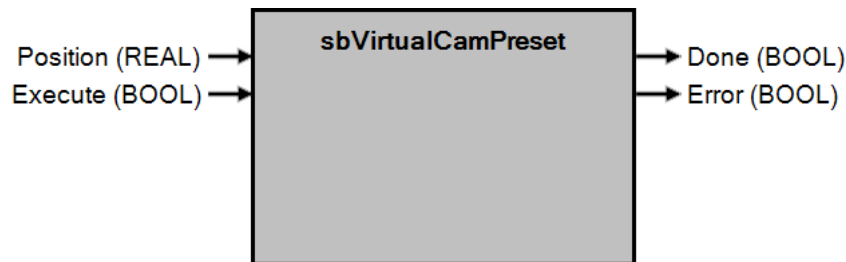
##### **InMotion (BOOL):**

Esta saída indica se o came virtual está executando o posicionamento.

#### 6.11.3.4 Chamada do bloco

sbVirtualCamPosition(Position, Velocity, Start, Stop)

#### 6.11.4 Preset Came Virtual - sbVirtualCamPreset



##### 6.11.4.1 Funcionamento

Este bloco tem a função de executar um preset no came virtual com o valor que está na entrada "Position".

##### 6.11.4.2 Entradas

###### Position (REAL):

Esta entrada define a posição que o came assumirá quando o bloco for executado.

###### Execute (BOOL):

Esta entrada solicita que o came assumira a posição definida na entrada "Position".

##### 6.11.4.3 Saídas:

###### Done (BOOL):

Esta saída indica se o movimento foi concluído.

###### Error (BOOL):

Esta saída indica se ocorreu um durante o processamento do bloco. Verificar se existe uma instância habilitada do bloco "VirtualCamAxis" e se o eixo está parado ao solicitar o preset. Caso o eixo esteja em movimento, o bloco gera falha.

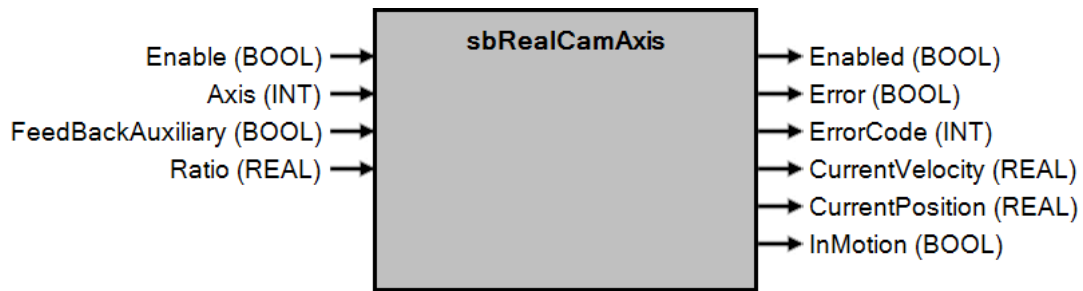
Caso nenhuma das situações acima estejam inválidas, desabilite o bloco e habilite novamente.

##### 6.11.4.4 Chamada do bloco

sbVirtualCamPreset(Position, Execute)



### 6.11.5 Came Eixo Real - sbRealCamAxis



#### 6.11.5.1 Funcionamento

Este bloco tem a função de criar um came real. É semelhante a um came virtual, com a diferença que nesse caso o eixo segue os movimentos do encoder ou eixo ao qual está acoplado.

#### 6.11.5.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **Axis (INT):**

Esta entrada define qual eixo será o mestre. No caso de utilizar encoder deve-se informar em qual eixo (AKD) está conectado o mesmo.

##### **Ratio (REAL):**

Esta entrada define o fator de acoplamento entre o escravo e o mestre. Pode ser programado 1 para uma relação 1:1, 2 para que o eixo escravo dê 2 voltas para 1 volta do mestre ou então 0.5 para que o eixo mestre de 2 voltas para 1 volta do escravo. Este fator é variável podendo se colocar o fator adequado para sua aplicação.

##### **FeedBackAuxiliary (BOOL):**

Caso esteja em FALSE, o feedback utilizado será um encoder auxiliar. Caso esteja em TRUE, utilizará o feedback do eixo definido na entrada "Axis".

#### 6.11.5.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica que o bloco está habilitado e pronto para o uso.

##### **CurrentPosition (REAL):**

Esta saída indica a posição atual do came.

##### **CurrentVelocity (REAL):**

Esta saída indica a velocidade atual do came.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.5.4 Chamada do bloco

sbRealCam(Enable, Axis, FeedBackAuxiliary, Ratio);

### 6.11.6 Preset Came Eixo Real - sbRealCamPreset



#### 6.11.6.1 Funcionamento

Este bloco tem a função de fazer um preset no came real com o valor que está na entrada "Position".

#### 6.11.6.2 Entradas

**Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

**Position (REAL):**

Esta entrada define a posição que o came assumirá quando o bloco for executado.

**Execute (BOOL):**

Esta entrada solicita que o came assuma a posição definida na entrada "Position".

#### 6.11.6.3 Saídas:

**Done (BOOL):**

Esta saída indica se movimento foi concluído.

**Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

**ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.6.4 Chamada do bloco

sbRealCamPreset(Enabled, Position, Execute);

### 6.11.7 Acoplar Came Real - sbRealCamClutch



#### 6.11.7.1 Funcionamento

Este bloco tem a função de fazer o acoplamento com o came real. Nele definimos se ocorre um acoplamento imediato ou após determinada posição. Caso se utilize a forma de acoplar após uma determinada posição, devemos defini-la na entrada "Position". Após configurar o número do eixo, o tipo de acoplamento e se houver habilitado o bloco, deve-se colocar a entrada "Execute" em TRUE.

#### 6.11.7.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **ClutchType (BOOL):**

Esta entrada define o tipo de acoplamento, onde:

0: acopla imediatamente;

1: acopla após passar pela posição determinada na entrada "Position".

##### **Position (REAL):**

Esta entrada define a posição utilizada para acoplar o came quando a entrada "ClutchType" estiver em TRUE.

##### **Execute (BOOL):**

Esta entrada solicita o acoplamento do came.

#### 6.11.7.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica que o bloco está habilitado.

##### **Coupled (BOOL):**

Esta saída indica que o came está acoplado.

##### **Error (BOOL):**

Esta saída indica que ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

##### **WaitingForCoupling (BOOL):**

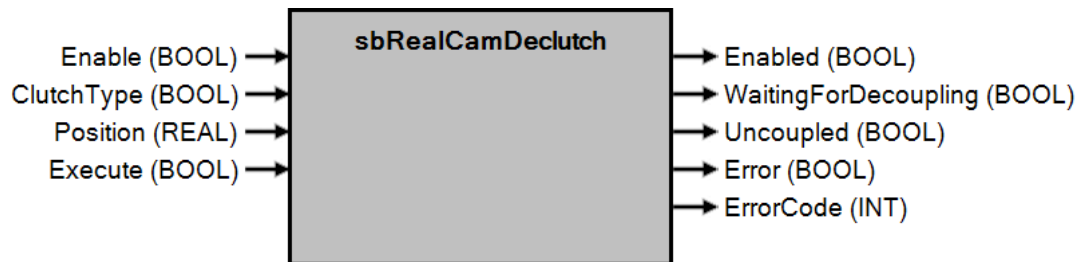
Esta saída indica se o came está aguardando passar pela posição programada na entrada

"Position" para poder acoplar. Quando for realizado o acoplamento, a saída "Coupled" fica em TRUE.

#### 6.11.7.4 Chamada do bloco

sbRealCamClutch (Enable, ClutchType, Position, Execute)

### 6.11.8 Desacoplar Came Real - sbRealCamDeclutch



#### 6.11.8.1 Funcionamento

Este bloco tem a função de desacoplar o came real. Nele definimos se ocorre um acoplamento imediato ou após determinada posição. Caso se utilize a forma de desacoplar após uma determinada posição, devemos defini-la na entrada "Position". Após configurar o número do eixo, o tipo de desacoplamento e se houver habilitado o bloco, deve-se colocar a entrada "Execute" em TRUE.

#### 6.11.8.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **ClutchType (BOOL):**

Esta entrada define o tipo de desacoplamento.

Se essa entrada estiver em FALSE, será feito um desacoplamento imediato. Caso esteja em TRUE, o desacoplamento será feito após passar pela posição programada na entrada "Position".

##### **Position (REAL):**

Esta entrada define a posição utilizada para desacoplar o came quando a entrada "ClutchType" estiver em TRUE.

##### **Execute (BOOL):**

Esta entrada solicita que o came faça o desacoplamento.

#### 6.11.8.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Uncoupled (BOOL):**

Esta saída indica se o came está desacoplado.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

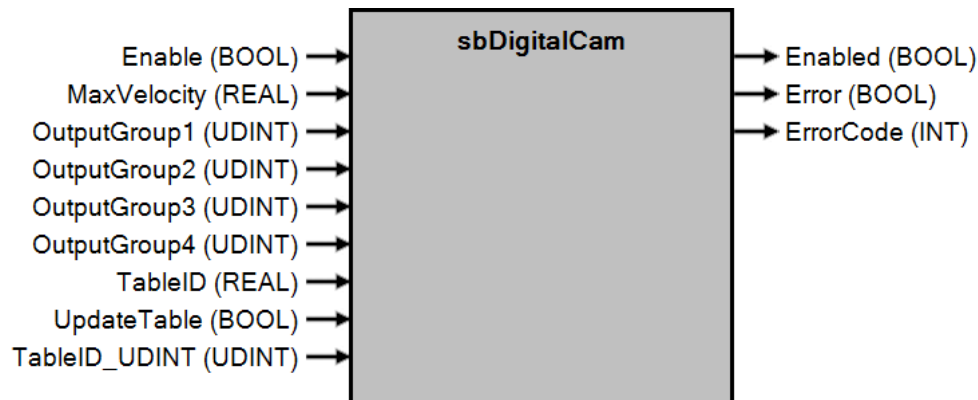
##### **WaitingForDecoupling (BOOL):**

Esta saída indica se o came está aguardando passar pela posição programada na entrada "Position" para poder desacoplar. Quando for realizado o desacoplamento, a saída "Uncoupled" fica em TRUE.

#### 6.11.8.4 Chamada do bloco

```
sbRealCamDeclutch(Enable, ClutchType, Position, Execute);
```

### 6.11.9 Came Digital - sbDigitalCam



#### 6.11.9.1 Funcionamento

Este bloco tem a função de criar um came digital com uma estrutura de até 32 saídas digitais. Através de uma tabela, configurada pelo usuário, é possível acionar estas saídas por ângulos ou por tempo. No editor de tabelas do AKC-PPC IDE, deve ser criada uma tabela de 32 linhas e 5 colunas, onde:

- A coluna 1 é referente ao ângulo inicial em graus
- A coluna 2 é referente ao ângulo final em graus ou tempo em milissegundos
- A coluna 3 é referente ao Shift (deslocamento do came em função da velocidade) em graus
- A coluna 4 é referente ao tipo do came mestre (0=Real, 1=Virtual)
- A coluna 5 é para selecionar o came por tempo (0=Graus, 1=Tempo)

#### 6.11.9.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **MaxVelocity (REAL):**

Esta entrada define a velocidade do came real (encoder, resolver, etc.) ou virtual - depende de qual came está vinculado ao came digital.

##### **OutputGroup1 (UDINT):**

Esta entrada define o número do grupo de saídas digitais que será associado as saídas de came 0 a 7.

##### **OutputGroup2 (UDINT):**

Esta entrada define o número do grupo de saídas digitais que será associado as saídas de came 8 a 15.

##### **OutputGroup3 (UDINT):**

Esta entrada define o número do grupo de saídas digitais que será associado as saídas de came 16 a 23.

##### **OutputGroup4 (UDINT):**

Esta entrada define o número do grupo de saídas digitais que será associado as saídas de came 24 a 31.

##### **TableID (REAL):**

Esta entrada define o número da tabela de configuração dos comes digitais.

##### **UpdateTable (BOOL):**

Esta entrada define o bit que será usado para atualizar a tabela de comes digitais.

##### **TableID\_UDINT (UDINT):**

Esta entrada define número da tabela (deve ser digitado novamente).

#### 6.11.9.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

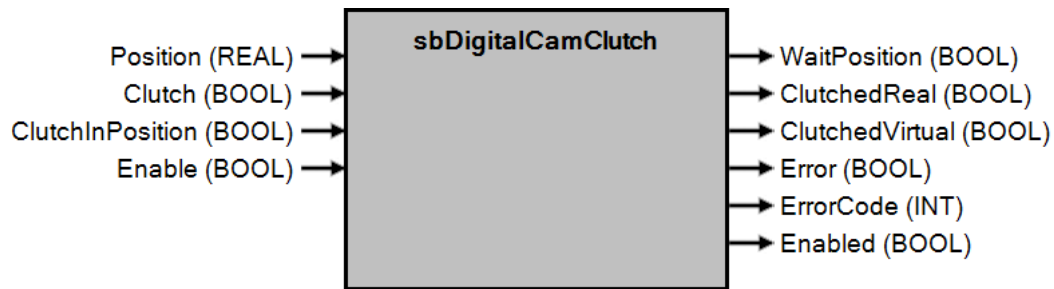
##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

### 6.11.9.4 Chamada do bloco

sbCameDigital(Enable, MaxVelocity, OutputGroup1, OutputGroup2, OutputGroup3, OutputGroup4, TableID, UpdateTable, TableID\_UDINT)

### 6.11.10 Acoplar Came Digital - sbDigitalCamClutch



#### 6.11.10.1 Funcionamento

Este bloco tem a função de executar o acoplamento do came digital criado em "sbDigitalCam" ao came mestre, seja virtual ou real.

#### 6.11.10.2 Entradas

##### **Position (REAL):**

Esta entrada define a posição utilizada para acoplar o came caso seja utilizado o acionamento na entrada "ClutchInPosition". Caso haja um offset aplicado ao came, a posição levará em conta este deslocamento.

##### **Clutch (BOOL):**

Esta entrada solicita acoplar imediatamente o came digital ao came associado (real ou virtual), ou seja, na posição atual, o came associado será acoplado ao came digital.

##### **ClutchInPosition (BOOL):**

Esta entrada solicita acoplar o came digital do came associado (real ou virtual) no momento em que a posição do came associado passar pela posição definida na entrada "Position".

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

#### 6.11.10.3 Saídas:

##### **WaitPosition (BOOL):**

Esta saída indica se o CNC está aguardando o came real chegar à posição definida na entrada "Position" para acoplar.

##### **ClutchedReal (BOOL):**

Esta saída indica se o came digital está acoplado ao came real.

##### **ClutchedVirtual (BOOL):**

Esta saída indica se o came digital está acoplado ao came virtual.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

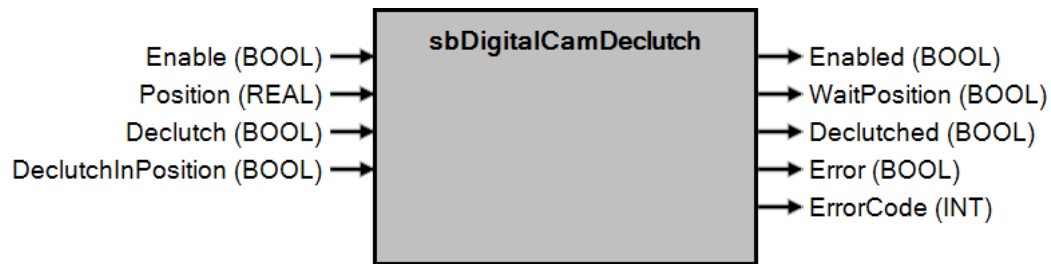
##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

#### 6.11.10.4 Chamada do bloco

sbDigitalCamClutch(Position, Clutch, ClutchInPosition, Enable)

### 6.11.11 Desacoplar Came Digital - sbDigitalCamDeclutch



#### 6.11.5.1 Funcionamento

Este bloco tem a função de executar o desacoplamento do came digital criado em "sbDigitalCam" ao came mestre, seja virtual ou real.

#### 6.11.11.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **Position (REAL):**

Esta entrada define a posição utilizada para desacoplar o came caso seja utilizado o acionamento na entrada "DeclutchInPosition". Caso haja um offset aplicado ao came, a posição levará em conta este deslocamento.

##### **Declutch (BOOL):**

Esta entrada solicita desacoplar imediatamente o came digital do came associado (real ou virtual), ou seja, na posição atual do came associado ele será desacoplado do came digital.

##### **DeclutchInPosition (BOOL):**

Esta entrada solicita desacoplar o came digital do came associado (real ou virtual) no momento em que a posição do came associado passar pela posição definida na entrada "Position".

#### 6.11.11.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Declutched (BOOL):**

Esta saída indica se o came digital está desacoplado do came associado (real ou virtual).

##### **WaitPosition (BOOL):**

Esta saída indica se o CNC está aguardando o came real chegar a uma posição definida em "Position" para então desacoplar.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

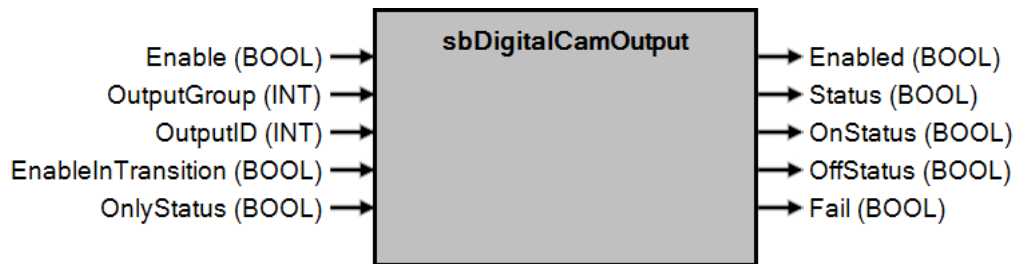
Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.11.4 Chamada do bloco

sbDigitalCamDeclutch (Enable, Position, Declutch, DeclutchInPosition)



### 6.11.12 Saída Came Digital - sbDigitalCamOutput



#### 6.11.12.1 Funcionamento

Este bloco tem a função de criar um came digital, ou seja, através dele é possível utilizar os estados de um came digital.

#### 6.11.12.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **OutputGroup (INT):**

Esta entrada define o número do grupo de saídas digitais que será associado às saídas que se quer verificar os estados.

##### **OutputID (INT):**

Esta entrada define o número da saída do grupo definida em "OutputGroup" que se quer verificar os estados.

##### **EnableInTransition (BOOL):**

Esta entrada habilita a saída na transição.

##### **OnlyStatus (BOOL):**

Esta entrada configura o bloco para não alterar o estado da saída digital (se estiver em TRUE).

#### 6.11.12.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado e pronto para o uso.

##### **OffStatus (BOOL):**

Esta saída indica o estado da borda de descida do came digital definido nas entradas "OutputGroup" e "OutputID". Ficará em TRUE por um ciclo de PLC caso o came digital sofra uma transição de TRUE para FALSE.

##### **OnStatus (BOOL):**

Esta saída indica o estado da borda de subida do came digital definido nas entradas "OutputGroup" e "OutputID". Ficará em TRUE por um ciclo de PLC caso o came digital sofra uma transição de FALSE para TRUE.

##### **Status (BOOL):**

Esta saída indica o atual estado do came definido em "OutputGroup" e "OutputID".

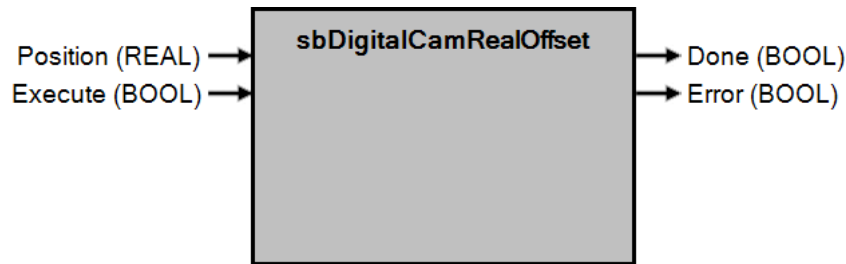
##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco. Verificar se existe uma instância habilitada do bloco "sbDigitalCam". Caso exista, o erro foi gerado internamente no CNC.

#### 6.11.12.4 Chamada do bloco

sbDigitalCamOutput(Enable, OutputGroup, OutputID, EnableInTransition, OnlyStatus)

### 6.11.13 Offset Came Digital Real – sbDigitalCamRealOffset



#### 6.11.13.1 Funcionamento

Este bloco tem a função de executar um deslocamento (Offset) entre o came digital e o came real, ou seja, caso o came digital esteja acoplado ao came real e queira aplicar uma diferença de posição entre estes comes se utiliza esta função.

#### 6.11.13.2 Entradas

**Position (REAL):**

Esta entrada define o valor do deslocamento, em graus, que se deseja aplicar entre o came digital e o real.

**Execute (BOOL):**

Esta entrada faz o bloco aplicar o offset definido na entrada "Position" ao came digital. Para isso execute uma transição de subida nesta entrada.

#### 6.11.13.3 Saídas:

**Done (BOOL):**

Esta saída indica que o offset foi concluído.

**Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

Verifique se existe uma instância habilitada do bloco "sbDigitalCam" ou se o came digital está acoplado ao came real. Caso exista a instância e os comes estiverem acoplados, o erro foi gerado internamente no CNC.

Desabilitar o bloco e habilitar novamente.

#### 6.11.13.4 Chamada do bloco

sbDigitalCamRealOffset(Position, Execute)

#### 6.11.14 Offset Came Digital Virtual – sbDigitalCamVirtualOffset



##### 6.11.14.1 Funcionamento

Este bloco tem a função de executar um deslocamento (Offset) entre o came digital e o came virtual.

##### 6.11.14.2 Entradas

###### **Position (REAL):**

Esta entrada define o valor do deslocamento, em graus, que se deseja aplicar entre o came digital e o virtual.

###### **Execute (BOOL):**

Esta entrada faz o bloco aplicar o offset definido na entrada "Position" ao came digital.

##### 6.11.14.3 Saídas:

###### **Done (BOOL):**

Esta saída indica se o offset foi concluído.

###### **Error (BOOL):**

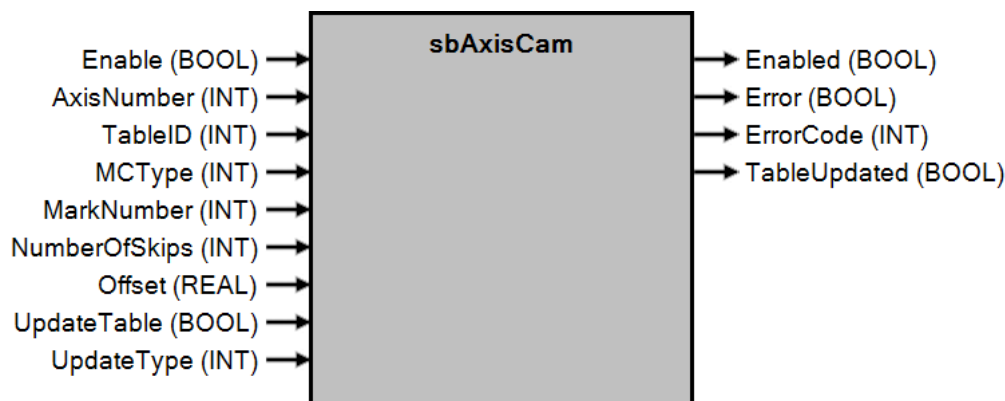
Esta saída indica se ocorreu um erro durante o processamento do bloco. Verificar se existe uma instância habilitada do bloco "sbDigitalCam" ou se o came digital está acoplado ao came virtual.

Caso exista a instância e os comes estiverem acoplados, o erro foi gerado internamente no CNC. Desabilitar o bloco e habilitar novamente.

##### 6.11.14.4 Chamada do bloco

sbDigitalCamVirtualOffset(Position, Execute)

### 6.11.15 Came de Eixo - sbAxisCam



#### 6.11.15.1 Funcionamento

Este bloco tem a função de criar um came de eixo. Esse tipo de came tem como característica primordial realizar o movimento de um eixo conforme o que foi programado na tabela correspondente. Esse came pode ser acoplado a um came mestre do tipo real ou virtual.

No editor de tabelas do AKC-PPC IDE, deve ser criada uma tabela de 8 linhas e 8 colunas, onde:

- A coluna 1 é referente ao ângulo inicial em graus
- A coluna 2 é referente ao ângulo final em graus ou tempo em milissegundos
- A coluna 3 é referente ao comprimento do vínculo em milímetros
- A coluna 4 é referente ao número da tabela com perfil complexo
- A coluna 5 é referente a zona de transição inicial em porcentagem do ângulo
- A coluna 6 é referente a zona de transição final em porcentagem do ângulo
- A coluna 7 é referente ao fator de acoplamento inicial em milímetros/grau
- A coluna 8 é referente ao fator de acoplamento final em milímetros/grau

#### 6.11.15.2 Entradas

##### Enable (BOOL):

Esta entrada habilita o bloco.

##### AxisNumber (INT):

Esta entrada define o número do eixo que será controlado pelo came.

##### TableID (INT):

Esta entrada define o número da tabela com configurações do came de eixo.

##### MCType (INT):

Define o tipo de came que será acoplado (0=came mestre real,1=came mestre virtual).

##### MarkNumber (INT):

Esta entrada define o número da entrada de foto que será utilizada para correção do came. Caso fique em 0 não será utilizado fotocélula.

##### NumberOfSkips (INT):

Esta entrada define o número de skips.

##### Offset (REAL):

Esta entrada define o valor para fazer um offset no came de eixo.

##### UpdateTable (BOOL):

Esta entrada programa a tabela de vínculos.

##### UpdateType (INT):

Esta entrada programa a tabela auxiliar com dados da tabela de vínculos

#### 6.11.15.3 Saídas

##### Enabled (BOOL):

Esta saída indica se bloco está habilitado.

##### Error (BOOL):

Esta saída indica se ocorreu um erro durante o processamento do bloco.

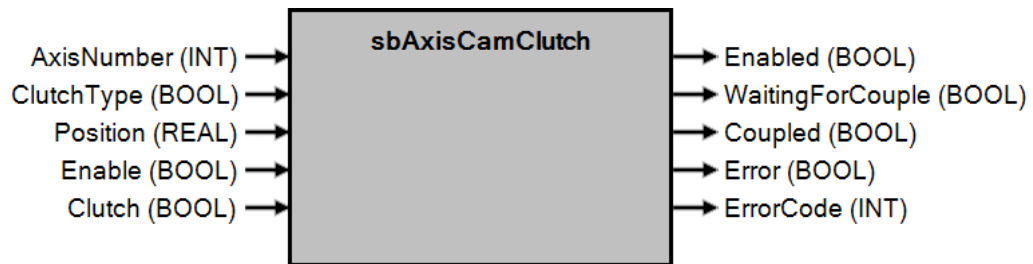
**ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**6.11.15.4 Chamada do bloco**

sbAxisCam(Enable, AxisNumber, TableID, MCType, MarkNumber, NumberOfSkips, Offset, UpdateTable, UpdateType)

### 6.11.16 Acoplar Came de Eixo - sbAxisCamClutch



#### 6.11.16.1 Funcionamento

Este bloco tem a função de acoplar o came de eixo. Nele definimos se ocorre um acoplamento imediato ou após determinada posição. Caso se utilize a forma de acoplar após posição, deve-se definir uma na entrada "Position". Após configurar o número do eixo, o tipo de acoplamento e habilitar o bloco, deve-se colocar a entrada "Clutch" em TRUE.

#### 6.11.16.2 Entradas

##### **AxisNumber (INT):**

Esta entrada define o número do eixo a ser controlado.

##### **ClutchType (BOOL):**

Esta entrada define o tipo de acoplamento, onde:

0: acopla imediatamente;

1: acopla após passar pela posição determinada na entrada "Position".

##### **Position (REAL):**

Esta entrada define a posição utilizada para acoplar o came quando a entrada "ClutchType" estiver em TRUE.

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **Clutch (BOOL):**

Esta entrada solicita o acoplamento do came.

#### 6.11.16.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **WaitingForCouple (BOOL):**

Esta saída indica se o came está esperando passar pela posição programada para poder acoplar.

##### **Coupled (BOOL):**

Esta saída indica se o came de eixo está acoplado ao came mestre.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.16.4 Chamada do bloco

sbAxisCamClutch(AxisNumber, ClutchType, Position, Enable, Clutch)

### 6.11.17 Desacoplar Came de Eixo - sbAxisCamDeclutch



#### 6.11.17.1 Funcionamento

Este bloco tem a função de desacoplar o came de eixo.

#### 6.11.17.2 Entradas

##### **AxisNumber (INT):**

Esta entrada define o número do eixo a ser controlado.

##### **DeclutchType (BOOL):**

Esta entrada define o tipo de desacoplamento, onde:

0: desacopla imediatamente;

1: desacopla após passar pela posição determinada na entrada "Position".

##### **Position (REAL):**

Esta entrada define a posição utilizada para desacoplar o came quando a entrada "DeclutchType" estiver em TRUE.

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **Declutch (BOOL):**

Esta entrada faz o desacoplamento se todas as entradas anteriores estiverem configuradas corretamente.

#### 6.11.17.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Uncoupled (BOOL):**

Esta saída indica se o came de eixo foi desacoplado do came mestre.

##### **WaitingForUncouple (BOOL):**

Essa saída indica se o came está esperando passar pela posição programada para poder desacoplar.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

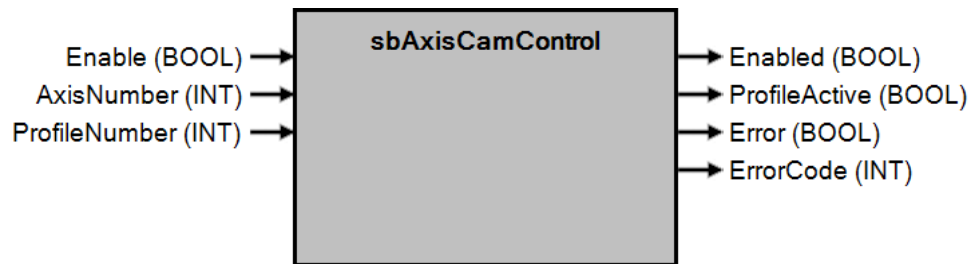
##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

#### 6.11.17.4 Chamada do bloco

sbAxisCamDeclutch(AxisNumber, DeclutchType, Position, Enable, Declutch)

### 6.11.18 Controle Came de Eixo - sbAxisCamControl



#### 6.11.18.1 Funcionamento

Este bloco tem a função de selecionar o perfil a ser executado.

#### 6.11.18.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco, para execução dos vínculos programados.

Caso o valor esteja em FALSE, o bloco é desabilitado e todas as saídas do bloco recebem valores nulos.

##### **AxisNumber (INT):**

Esta entrada define o número do eixo no qual será executado o vínculo.

##### **ProfileNumber (BOOL):**

Esta entrada define o número do perfil (linha da tabela).

#### 6.11.18.3 Saídas

##### **Enabled (BOOL):**

Esta saída indica se o bloco está habilitado.

##### **Error (BOOL):**

Esta saída indica se ocorreu um erro durante o processamento do bloco.

##### **ErrorCode (INT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

##### **ProfileActive (BOOL):**

Esta saída indica se o perfil está ativado.

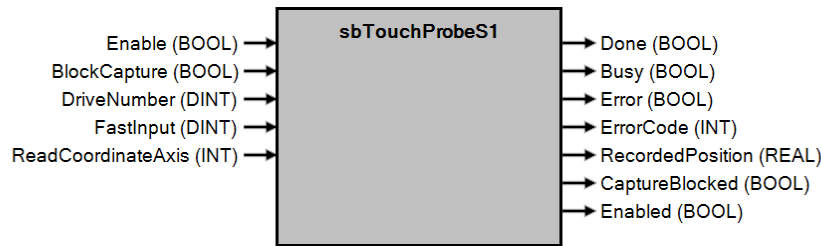
#### 6.11.18.4 Chamada do bloco

sbAxisCamControl(Enable, AxisNumber, ProfileNumber)



## 6.12 Packaging

### 6.12.1 Leitura de Entrada Rápida - TouchProbeS1



#### 6.12.1.1 Funcionamento

Este bloco tem a função de configurar uma entrada digital rápida do drive e retornar a posição capturada do eixo quando ocorrer uma transição de subida nessa entrada digital.

Este bloco de função não causa movimento no eixo.

#### 6.12.1.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita do bloco.

##### **DriveNumber (SINT):**

Esta entrada seleciona o eixo no qual a posição será capturada.

##### **FastInput (DINT):**

Esta entrada seleciona qual entrada digital rápida será usada para disparar o evento de captura.

##### **BlockCapture (BOOL):**

Esta entrada bloqueia o evento de captura.

##### **ReadCoordinateAxis (INT):**

Esta entrada define o número do eixo que o bloco irá capturar posição.

#### 6.12.1.3 Saídas

##### **Done (BOOL):**

Esta saída indica que a captura ocorreu com sucesso.

##### **CaptureBlocked (BOOL):**

Esta saída indica que a captura está bloqueada.

##### **Enabled (BOOL):**

Esta saída indica que o bloco está habilitado.

##### **Busy (BOOL):**

Esta saída indica que o bloco foi comandado para capturar a posição do eixo e está aguardando o evento de captura.

##### **Error (BOOL):**

Esta saída indica que um erro ocorreu durante o processamento do bloco.

##### **ErrorCode (BOOL):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

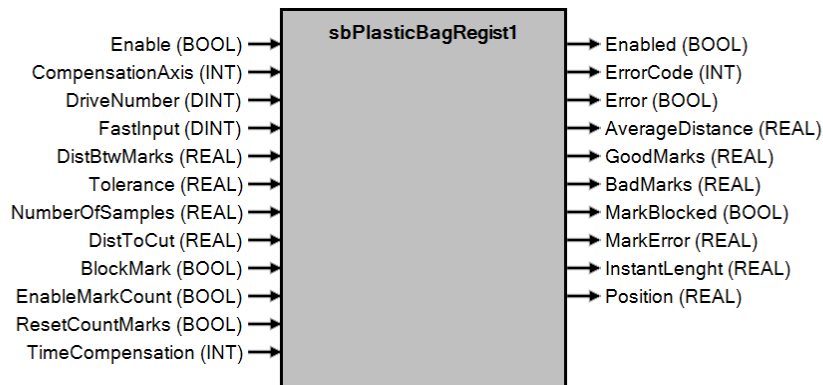
##### **RecordedPosition (REAL):**

Esta saída indica a posição capturada no eixo definido na entrada "DriveNumber" no momento da borda de subida da entrada definida em "FastInput".

#### 6.12.1.4 Chamada do bloco

sbTouchProbeS1(Enable, BlockCapture, DriveNumber, FastInput, ReadCoordinateAxis)

### 6.12.2 Registro de Marcas - PlasticBagRegist1



#### 6.12.2.1 Funcionamento

Este bloco tem a função de controlar o registro de marcas para o eixo especificado de acordo com leitura de marcas feita pela fotocélula configurada.

#### 6.12.2.2 Entradas

**Enable (BOOL):**

Esta entrada habilita a execução do bloco.

**DriverNumber (INT):**

Esta entrada define o número do drive em que a fotocélula está ligada.

**FastInput (INT):**

Esta entrada seleciona qual entrada digital rápida será usada para disparar o evento da captura.

**CompensationAxis (INT):**

Esta entrada define o Número do eixo no qual será aplicada a correção da fotocélula.

**DistBtwMarks (REAL):**

Esta entrada define a distância entre marcas.

**Tolerance (REAL):**

Esta entrada define a janela em torno da posição esperada de detecção de marca para considerar marca boa.

**DistToCut (REAL):**

Esta entrada define a distância entre a detecção da marca e o corte do material.

**NumberOfSamples (REAL):**

Esta entrada define o número de amostragens de distância entre marcas para calcular média.

**BlockMark (BOOL):**

Esta entrada bloqueia a leitura da fotocélula.

**ResetCountMarks (BOOL):**

Esta entrada reinicia a variável de saída que conta as marcas.

**TimeCompensation (INT):**

Esta entrada faz a compensação da leitura da posição da photo (em microssegundos).

**EnableMarkCount (BOOL):**

Esta entrada habilita a fila de marcas, no caso da fotocélula estar deslocada do ponto de atuação da correção/corte.

#### 6.12.2.3 Saídas

**Enabled (BOOL):**

Esta saída indica que o bloco está habilitado.

**Error (BOOL):**

Esta saída indica um erro ocorreu durante o processamento do bloco.

**ErrorCode (DINT):**

Esta saída indica o código do erro caso a saída "Error" seja TRUE. Os significados dos erros estão listados na tabela de erros dos blocos.

**AverageDistance (REAL):**

Esta saída indica a média da distância entre marcas.

**MarkError (REAL):**

Esta saída indica a diferença entre posição esperada e posição capturada.

**GoodMarks (REAL):**

Esta saída indica o número de marcas boas.

**BadMarks (REAL):**

Esta saída indica o número de marcas ruins.

**MarkBlocked (BOOL):**

Esta saída indica que a captura de marca pela fotocélula foi bloqueada.

**InstantLenght (REAL):**

Esta saída indica a posição instantânea medida.

**Position (REAL):**

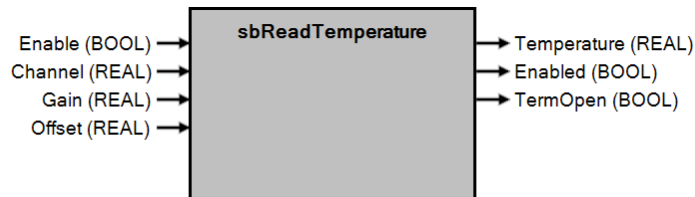
Esta saída indica a posição do eixo que está a fotocélula.

**6.12.2.4 Chamada do bloco**

sbPlasticBagRegist1(Enable, CompensationAxis, DriveNumber, FastInput, DistBtwMarks, Tolerance, NumberOfSamples, DistToCut, BlockMark, EnableMarkCount, ResetCountMarks, TimeCompensation)

## 6.13 Temperatura

### 6.13.1 Ler Temperatura - ReadTemperature



#### 6.13.1.1 Funcionamento

Este bloco tem a função de leitura de temperatura. Retorna a temperatura lida do termopar multiplicada pelo valor definido na entrada "Gain" e somado (ou subtraído) pelo valor definido na entrada "Offset".

Equação:

$$\text{Temperature} = (\text{ThermocoupleTemp} * \text{Gain}) + \text{Offset}$$

Onde:

**ThermocoupleTemp** é a temperatura lida do termopar

**Gain** é a entrada "Gain"

**Offset** é a entrada "Offset"

**Temperature** é a saída "Temperature"

#### 6.13.1.2 Entradas:

##### **Enable (BOOL):**

Esta entrada habilita o bloco.

##### **Channel (REAL):**

Esta entrada define o número do canal do qual se quer ler o termopar.

##### **Gain (REAL):**

Esta entrada define o ganho que será multiplicado pelo valor lido do termopar.

##### **Offset (REAL):**

Esta entrada define o offset que será somado (ou subtraído, se for negativo) ao resultado da multiplicação entre o valor lido do termopar e o ganho definido na entrada "Gain".

#### 6.13.1.3 Saídas:

##### **Enabled (BOOL):**

Esta saída indica que o bloco foi habilitado.

##### **Temperature (REAL):**

Esta saída indica o valor da temperatura lido, com ganho e offset já aplicados.

##### **TermOpen (REAL):**

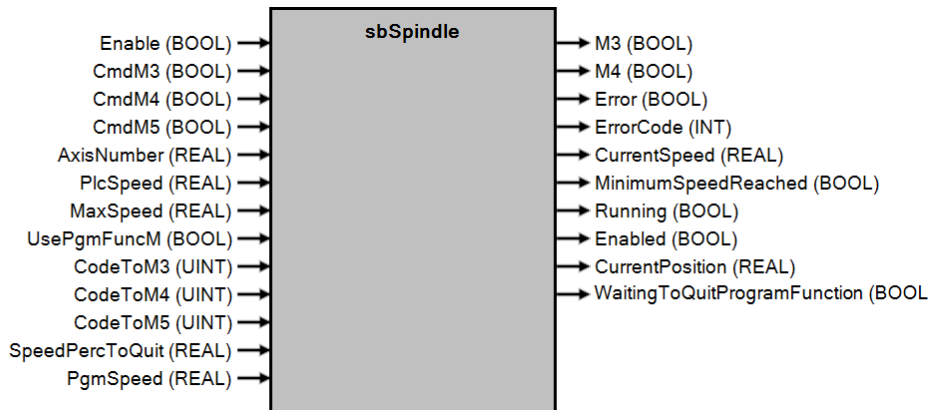
Esta saída indica se o termopar está aberto.

#### 6.13.1.4 Chamada do bloco

sbReadTemperature(Enable, Channel, Gain, Offset)

## 6.14 Spindle

### 6.14.1 Eixo Árvore Geral - sbSpindle



#### 6.14.1.1 Funcionamento

Este bloco tem a função de movimentar de forma contínua um eixo do tipo árvore.

O acionamento do spindle pode ser feito através do PLC utilizando as entradas "CmdM3", "CmdM4" e "CmdM5".

O acionamento do spindle também pode ser feito através da execução de programa ISO, onde os códigos das funções M são definidos nas entradas "CodeToM3", "CodeToM4" e "CodeToM5".

Este bloco pode ser utilizado em mais de um eixo do tipo árvore.

#### 6.14.1.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para utilização do eixo árvore quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **CmdM3 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM4 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido anti-horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM5 (BOOL):**

Esta entrada solicita ao bloco o cancelamento do movimento no eixo definido na entrada "AxisNumber".

##### **AxisNumber (REAL):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

##### **PlcSpeed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pelas entradas "CmdM3" ou "CmdM4".

##### **MaxSpeed (REAL):**

Esta entrada define a velocidade máxima de movimento do eixo. Essa entrada será utilizada internamente para definir o valor da saída analógica.

##### **UsePgmFuncM (BOOL):**

Esta entrada define se o bloco será utilizado para receber códigos de função M através do programa ISO na execução de programas pelo CNC.

##### **CodeToM3 (UINT):**

Esta entrada define o número do código da função M (enviado através do programa ISO) que executará o movimento no sentido horário do eixo definido na entrada "AxisNumber".

**CodeToM4 (UINT):**

Esta entrada define o número do código da função M (enviado através do programa ISO) que executará o movimento no sentido anti-horário do eixo definido na entrada "AxisNumber".

**CodeToM5 (UINT):**

Esta entrada define o número do código da função M (enviado através do programa ISO) que interromperá o movimento do eixo definido na entrada "AxisNumber".

**SpeedPercToQuit (REAL):**

Esta entrada define o valor em porcentagem que será utilizado para quitar a função M e continuar a execução do programa ISO.

**PgmSpeed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pelo programa ISO através dos códigos definidos nas entradas "CodeToM3" ou "CodeToM4".

**6.14.1.3 Saídas****M3 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido horário.

**M4 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido anti-horário.

**Error (BOOL):**

Esta saída indica que um erro aconteceu no eixo.

**ErrorCode (INT):**

Esta saída indica o código do erro que aconteceu no eixo.

**CurrentSpeed (REAL):**

Esta saída indica a velocidade atual que o eixo está se movimentando.

**MinimumSpeedReached (BOOL):**

Esta saída indica que o eixo está se movimentando e já atingiu a porcentagem de velocidade definida na entrada "SpeedPercToQuit".

**Running (BOOL):**

Esta saída indica que o eixo está em movimento.

**Enabled (BOOL):**

Esta saída indica que o bloco está habilitado e pronto para o uso.

**CurrentPosition (REAL):**

Esta saída indica a posição atual que o eixo está.

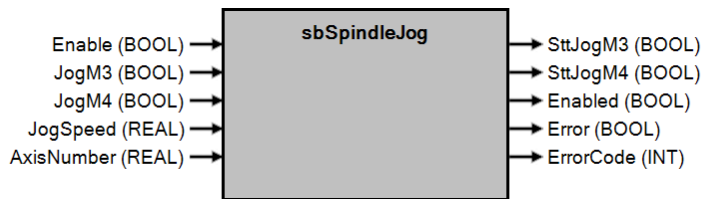
**WaitingToQuitProgramFunction (BOOL):**

Esta saída indica que o bloco recebeu um código do programa ISO para movimentar o eixo. O bloco já ligou o eixo e está aguardando atingir a porcentagem de velocidade definida na entrada "SpeedPercToQuit" para quitar a função M e continuar a execução do programa pelo CNC.

**6.14.1.4 Chamada do bloco**

sbSpindle( Enable, CmdM3, CmdM4, CmdM5, AxisNumber, PlcSpeed, MaxSpeed, UsePgmFuncM, CodeToM3, CodeToM4, CodeToM5, SpeedPercToQuit, PgmSpeed )

### 6.14.2 Jog de Eixo Árvore - sbSpindleJog



#### 6.14.2.1 Funcionamento

Este bloco tem a função de movimentar de forma momentânea um eixo do tipo árvore.

O acionamento do spindle pode ser feito através do PLC utilizando as entradas "CmdM3", "CmdM4" e "CmdM5".

Este bloco pode ser utilizado em mais de um eixo do tipo árvore.

#### 6.14.2.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para utilização do eixo árvore quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **JogM3 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento momentâneo no sentido horário do eixo definido na entrada "AxisNumber". O movimento será executado enquanto esta entrada estiver acionada.

##### **JogM4 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento momentâneo no sentido anti-horário do eixo definido na entrada "AxisNumber". O movimento será executado enquanto esta entrada estiver acionada.

##### **JogSpeed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pelas entradas "JogM3" ou "JogM4".

##### **AxisNumber (REAL):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

#### 6.14.2.3 Saídas

##### **SttJogM3 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido horário.

##### **SttJogM4 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido anti-horário.

##### **Enabled (BOOL):**

Esta saída indica que o bloco está habilitado e pronto para o uso.

##### **Error (BOOL):**

Esta saída indica que um erro aconteceu no eixo.

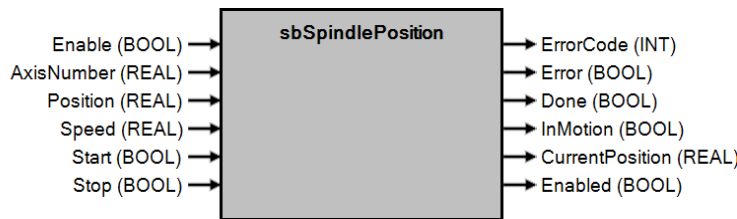
##### **ErrorCode (INT):**

Esta saída indica o código do erro que aconteceu no eixo.

#### 6.14.2.4 Chamada do bloco

sbSpindleJog(Enable, JogM3, JogM4, JogSpeed, AxisNumber)

### 6.14.3 Posicionamento do Eixo Árvore - sbSpindlePosition



#### 6.14.3.1 Funcionamento

Este bloco tem a função de movimentar um eixo do tipo árvore para uma posição definida na entrada “Position”. O acionamento do spindle deve ser feito através do PLC utilizando a entrada “Start”.

Este bloco pode ser utilizado em mais de um eixo do tipo árvore.

#### 6.14.3.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para utilização do eixo árvore quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **AxisNumber (REAL):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

##### **Position (REAL):**

Esta entrada define a posição final do movimento que será executado.

##### **Speed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pela entrada “Start”.

##### **Start (BOOL):**

Esta entrada solicita ao bloco a execução do movimento para a posição definido na entrada “Position” com velocidade definida na entrada “Speed”.

##### **Stop (BOOL):**

Esta entrada solicita ao bloco a interrupção do movimento que está sendo executado.

#### 6.14.3.3 Saídas

##### **ErrorCode (INT):**

Esta saída indica o código do erro que aconteceu no bloco.

##### **Error (BOOL):**

Esta saída indica que um erro aconteceu no processamento do bloco.

##### **Done (BOOL):**

Esta saída indica que o movimento solicitado foi concluído.

##### **InMotion (BOOL):**

Esta saída indica que o eixo está executando o movimento solicitado.

##### **CurrentPosition (REAL):**

Esta saída indica a posição atual do eixo definido na entrada “AxisNumber”.

##### **Enabled (BOOL):**

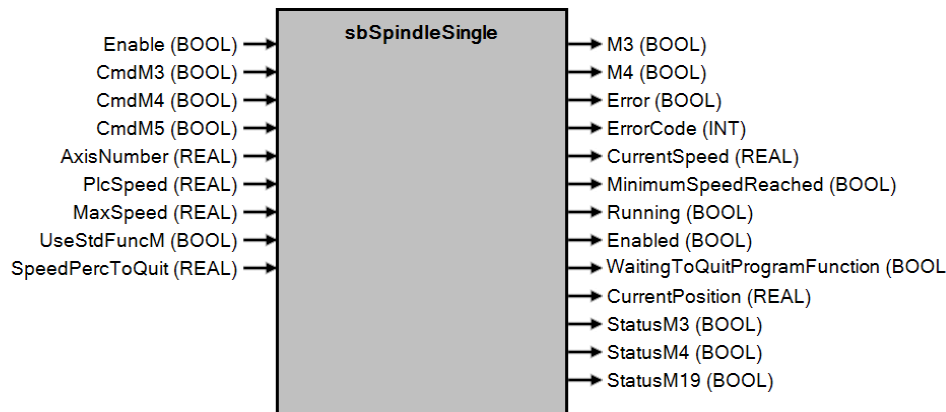
Esta saída indica que o bloco está habilitado e pronto para o uso.

#### 6.14.3.4 Chamada do bloco

sbSpindlePosition(Enable, AxisNumber, Position, Speed, Start, Stop)



### 6.14.4 Eixo Árvore Único - sbSpindleSingle



#### 6.14.4.1 Funcionamento

Este bloco tem a função de movimentar de forma contínua um eixo do tipo árvore.

O acionamento do spindle pode ser feito através do PLC utilizando as entradas "CmdM3", "CmdM4" e "CmdM5".

O acionamento do spindle também pode ser feito através da execução de programa ISO, onde os códigos das funções M padrões (M3, M4 e M5) serão utilizados se a entrada "UseStdFuncM" estiver em TRUE.

Este bloco pode ser utilizado em máquinas que possuam somente um eixo do tipo árvore.

#### 6.14.4.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para utilização do eixo árvore quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **CmdM3 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM4 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido anti-horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM5 (BOOL):**

Esta entrada solicita ao bloco o cancelamento do movimento no eixo definido na entrada "AxisNumber".

##### **AxisNumber (REAL):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

##### **PlcSpeed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pelas entradas "CmdM3" ou "CmdM4".

##### **MaxSpeed (REAL):**

Esta entrada define a velocidade máxima de movimento do eixo. Essa entrada será utilizada internamente para definir o valor da saída analógica.

##### **UseStdFuncM (BOOL):**

Esta entrada define se o bloco será utilizado para receber códigos de função M através do programa ISO na execução de programas pelo CNC.

##### **SpeedPercToQuit (REAL):**

Esta entrada define o valor em porcentagem que será utilizado para quitar a função M e continuar a execução do programa ISO.

**6.14.4.3 Saídas****M3 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido horário.

**M4 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido anti-horário.

**Error (BOOL):**

Esta saída indica que um erro aconteceu no eixo.

**ErrorCode (INT):**

Esta saída indica o código do erro que aconteceu no eixo.

**CurrentSpeed (REAL):**

Esta saída indica a velocidade atual que o eixo está se movimentando.

**MinimumSpeedReached (BOOL):**

Esta saída indica que o eixo está se movimentando e já atingiu a porcentagem de velocidade definida na entrada "SpeedPercToQuit".

**Running (BOOL):**

Esta saída indica que o eixo está em movimento.

**Enabled (BOOL):**

Esta saída indica que o bloco está habilitado e pronto para o uso.

**WaitingToQuitProgramFunction (BOOL):**

Esta saída indica que o bloco recebeu um código do programa ISO para movimentar o eixo. O bloco já ligou o eixo e está aguardando atingir a porcentagem de velocidade definida na entrada "SpeedPercToQuit" para quitar a função M e continuar a execução do programa pelo CNC.

**CurrentPosition (REAL):**

Esta saída indica a posição atual que o eixo está.

**StatusM3 (BOOL):**

Esta saída indica que o CNC recebeu um comando para movimentar o eixo árvore no sentido horário.

**StatusM4(BOOL):**

Esta saída indica que o CNC recebeu um comando para movimentar o eixo árvore no sentido anti-horário.

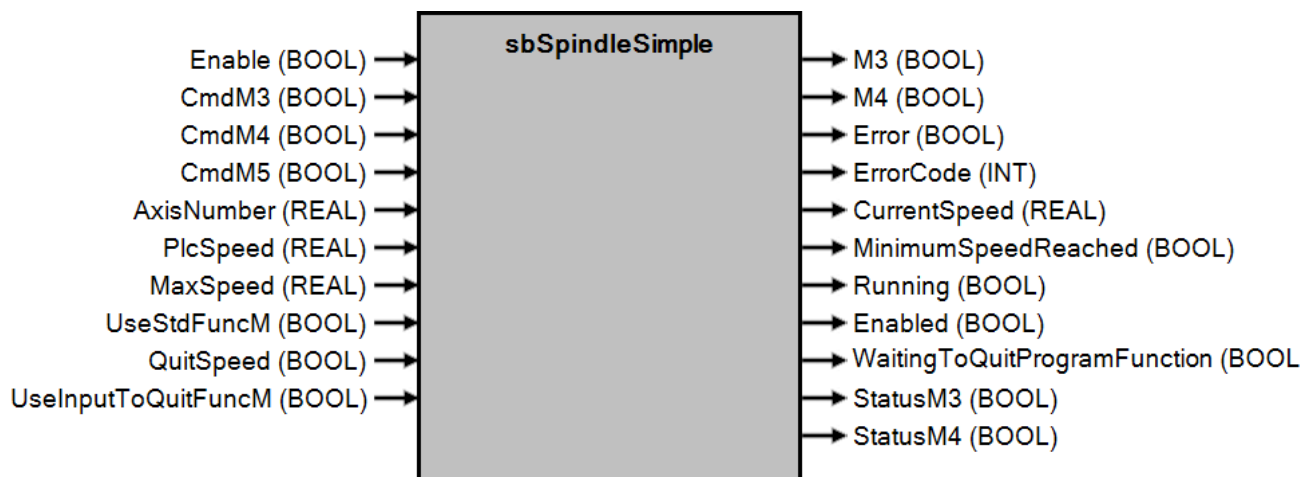
**StatusM19(BOOL):**

Esta saída indica que o CNC recebeu um comando para movimentar o eixo árvore no sentido horário.

**6.14.4.4 Chamada do bloco**

sbSpindleSingle( Enable, CmdM3, CmdM4, CmdM5, AxisNumber, PlcSpeed, MaxSpeed, UseStdFuncM, SpeedPercToQuit)

### 6.14.4 Eixo Árvore Simples - sbSpindleSimple



#### 6.14.4.1 Funcionamento

Este bloco tem a função de movimentar de forma contínua um eixo do tipo árvore.

O acionamento do spindle pode ser feito através do PLC utilizando as entradas "CmdM3", "CmdM4" e "CmdM5".

O acionamento do spindle também pode ser feito através da execução de programa ISO, onde os códigos das funções M padrões (M3, M4 e M5) serão utilizados se a entrada "UseStdFuncM" estiver em TRUE.

Este bloco pode ser utilizado em máquinas que possuam somente um eixo do tipo árvore.

#### 6.14.4.2 Entradas

##### **Enable (BOOL):**

Esta entrada habilita o bloco para utilização do eixo árvore quando o valor desta entrada for TRUE. Caso o valor desta entrada for FALSE o bloco é desabilitado e todas as saídas recebem valores nulos.

##### **CmdM3 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM4 (BOOL):**

Esta entrada solicita ao bloco a execução do movimento no sentido anti-horário do eixo definido na entrada "AxisNumber". Para interromper este movimento é necessário solicitar o mesmo através da entrada "CmdM5".

##### **CmdM5 (BOOL):**

Esta entrada solicita ao bloco o cancelamento do movimento no eixo definido na entrada "AxisNumber".

##### **AxisNumber (REAL):**

Esta entrada define o número do eixo (1 a 32) que se quer movimentar.

##### **PlcSpeed (REAL):**

Esta entrada define a velocidade de movimento do eixo quando for solicitado um movimento pelas entradas "CmdM3" ou "CmdM4".

##### **MaxSpeed (REAL):**

Esta entrada define a velocidade máxima de movimento do eixo. Essa entrada será utilizada internamente para definir o valor da saída analógica.

##### **UseStdFuncM (BOOL):**

Esta entrada define se o bloco será utilizado para receber códigos de função M através do programa ISO na execução de programas pelo CNC.

##### **QuitSpeed (BOOL):**

Esta entrada define o sinal será utilizada para determinar a continuação da execução do programa ISO, se a entrada "UseInputToQuitFuncM" for TRUE.

**UseInputToQuitFuncM (BOOL):**

Esta entrada define se a entrada "QuitSpeed" será utilizada para determinar a continuação da execução do programa ISO. Se esta entrada for FALSE o CNC continuará a execução normalmente, sem aguardar o spindle. Por segurança uma função de tempo pode ser adicionada no programa ISO para resolver essa situação.

**6.14.4.3 Saídas****M3 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido horário.

**M4 (BOOL):**

Esta saída indica que o eixo está se movimentando no sentido anti-horário.

**Error (BOOL):**

Esta saída indica que um erro aconteceu no eixo.

**ErrorCode (INT):**

Esta saída indica o código do erro que aconteceu no eixo.

**CurrentSpeed (REAL):**

Esta saída indica a velocidade atual que o eixo está se movimentando.

**MinimumSpeedReached (BOOL):**

Esta saída indica que o eixo está se movimentando e já atingiu a porcentagem de velocidade definida na entrada "SpeedPercToQuit".

**Running (BOOL):**

Esta saída indica que o eixo está em movimento.

**Enabled (BOOL):**

Esta saída indica que o bloco está habilitado e pronto para o uso.

**WaitingToQuitProgramFunction (BOOL):**

Esta saída indica que o bloco recebeu um código do programa ISO para movimentar o eixo. O bloco já ligou o eixo e está aguardando atingir a porcentagem de velocidade definida na entrada "SpeedPercToQuit" para quitar a função M e continuar a execução do programa pelo CNC.

**StatusM3 (BOOL):**

Esta saída indica que o CNC recebeu um comando para movimentar o eixo árvore no sentido horário.

**StatusM4(BOOL):**

Esta saída indica que o CNC recebeu um comando para movimentar o eixo árvore no sentido anti-horário.

**6.14.4.4 Chamada do bloco**

sbSpindleSimple( Enable, CmdM3, CmdM4, CmdM5, AxisNumber, PlcSpeed, MaxSpeed, UseStdFuncM, QuitSpeed, UseInputToQuitFuncM)

## 7.0 Exemplo de PLC

A seguir mostramos um trecho de programa de PLC base feito para o CNC.

Este programa está cheio de comentários para uma melhor compreensão de qualquer programador que possa ter a necessidade de entender/alterar o programa.

```
//=====
// INICIO DO PROGRAMA
//=====

//=====
// Leitura dos Parâmetros
//=====

IF svParametersUpdated THEN
    PosicaoTrocaX := sfReadParameter( 0, 0, 2 );
    PosicaoTrocaY := sfReadParameter( 0, 1, 2 );
    PosicaoTrocaZ := sfReadParameter( 0, 2, 2 );
END_IF;

//=====
// Bits de sistema
//=====
//-----
//Emergencia
//-----

//Botoes de emergencia
    svEmergencyCnc := iEmergencia;

//Saída de emergencia
    oEmergencia := svEmergencyOutput;

//Mensagens
    MESSAGE(0, ((NOT svEmergencyCnc) AND svInitializationMode));
    ALARM(0, (NOT svEmergencyOutput AND svCncInitialized));

//-----
//Partida e parada e bloqueios
//-----

//Start para CNC
    svStartCnc := iStart;

//Stop para CNC
    svStopCnc := iStop;

//Aborta execução
    svAbortCnc := ((NOT svEmergencyCnc) OR (NOT iInversorOK) OR (NOT iPressaoDoArOK));

//Bloqueio de partida
    svBlockStart := (NOT iInversorOK) OR (NOT iPressaoDoArOK);

//Mensagens
    TIMED_MESSAGE(0, (svBlockStart AND svStartCnc));
    MESSAGE(2, (NOT svStopCnc));
    MESSAGE(12, (NOT iInversorOK));
    MESSAGE(13, (NOT iPressaoDoArOK));
```

```

//=====
//Movimentos manuais - Jog
//=====

//habilitação para os movimentos de jog
HabJogX := svCncInitialized AND (NOT svExecutionRun) AND (bJogPositivoX OR iJogPosX OR
bJogNegativoX OR iJogNegX);
HabJogY := svCncInitialized AND (NOT svExecutionRun) AND (bJogPositivoY OR iJogPosY OR
bJogNegativoY OR iJogNegY);
HabJogZ := svCncInitialized AND (NOT svExecutionRun) AND (bJogPositivoZ OR iJogPosZ OR
bJogNegativoZ OR iJogNegZ);

//efetua os movimentos de jog (instâncias do bloco sbJog)
JogX(HabJogX, (bJogPositivoX OR iJogPosX) AND NOT FimDeCursoX.PositiveLimit,
(bJogNegativoX OR iJogNegX) AND NOT FimDeCursoX.NegativeLimit, 1, 0);
JogY(HabJogY, (bJogPositivoY OR iJogPosY) AND NOT FimDeCursoY.PositiveLimit,
(bJogNegativoY OR iJogNegY) AND NOT FimDeCursoY.NegativeLimit, 2, 0);
JogZ(HabJogZ, (bJogPositivoZ OR iJogPosZ) AND NOT FimDeCursoZ.PositiveLimit,
(bJogNegativoZ OR iJogNegZ) AND NOT FimDeCursoZ.NegativeLimit, 3, 0);

//Mensagens de erros de jog
MESSAGE(3, JogX.Error);
MESSAGE(4, JogY.Error);
MESSAGE(5, JogZ.Error);

//=====
//Fins de curso
//=====

//Fim de curso para cada eixo (instâncias do bloco sbLimitSwitch)
FimDeCursoX(svCncInitialized, svOnKeyCE, iFimDeCursoPosX OR ReferenciaRunning,
iFimDeCursoNegX, FALSE, TRUE);
FimDeCursoY(svCncInitialized, svOnKeyCE, iFimDeCursoPosY OR ReferenciaRunning,
iFimDeCursoNegY, FALSE, TRUE);
FimDeCursoZ(svCncInitialized, svOnKeyCE, iFimDeCursoPosZ OR ReferenciaRunning,
iFimDeCursoNegZ, FALSE, TRUE);

//aborta execução se tiver algum fim de curso
IF R_EDGE(FimDeCursoX.Alarm OR FimDeCursoY.Alarm OR FimDeCursoZ.Alarm) THEN
    svAbortCnc := TRUE;
END_IF;

ErroFimDeCurso := FimDeCursoX.Alarm OR FimDeCursoY.Alarm OR FimDeCursoZ.Alarm;

//mensagens de fim de curso
MESSAGE(6, FimDeCursoX.PositiveLimit);
MESSAGE(7, FimDeCursoX.NegativeLimit);
MESSAGE(8, FimDeCursoY.PositiveLimit);
MESSAGE(9, FimDeCursoY.NegativeLimit);
MESSAGE(10, FimDeCursoZ.PositiveLimit);
MESSAGE(11, FimDeCursoZ.NegativeLimit);
MESSAGE(14, ErroFimDeCurso);

```

```

//=====
//Referência dos eixos
//=====

//solicita busca de referência
IF sfKeyCode(2041) THEN
  IniciaBuscaReferencia := TRUE;
END_IF;

//desmarca referência dos eixos - ao solicitar cancelar busca ou interromper
IF sfKeyCode(2042) OR sfKeyCode(2044) THEN
  IniciaBuscaReferencia := FALSE;
  ReferenciaEixoX.Done := FALSE;
  ReferenciaEixoY.Done := FALSE;
  ReferenciaEixoZ.Done := FALSE;
END_IF;

//referência para cada eixo (instâncias do bloco sbHoming)
ReferenciaEixoZ(IniciaBuscaReferencia, iFimDeCursoPosZ, FALSE, FALSE, 3);
ReferenciaEixoX(ReferenciaEixoZ.Done, iFimDeCursoPosX, FALSE, FALSE, 1);
ReferenciaEixoY(ReferenciaEixoZ.Done, iFimDeCursoPosY, FALSE, FALSE, 2);

//indicação de referenciamento ok
EixosReferenciados := ReferenciaEixoX.Done AND ReferenciaEixoY.Done AND
ReferenciaEixoZ.Done;
//indicação de referenciamento em andamento
ReferenciaRunning := ReferenciaEixoX.Running OR ReferenciaEixoY.Running OR
ReferenciaEixoZ.Running;

//cancela solicitação de busca, ao término da busca do primeiro eixo
IF R_EDGE(ReferenciaEixoZ.Done) THEN
  IniciaBuscaReferencia := FALSE;
END_IF;

```