

57-1500

MA-BHA

Siemens Automation Cooperates with Education | 05/2017

Módulo TIA Portal 032-100 Fundamentos da programação FC com SIMATIC S7-1500

SIEMENS



Gratuito para o uso em centros de treinamento / pesquisa e desenvolvimento. © Siemens AG 2017. Todos os direitos reservados.

Pacotes de treinamento SCE apropriados a este tutorial

Comandos SIMATIC

- SIMATIC ET 200SP Open Controller CPU 1515SP PC F e HMI RT SW Nº de referência: 6ES7677-2FA41-4AB1
- SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety Nº de referência: 6ES7512-1SK00-4AB2
- SIMATIC CPU 1516F PN/DP Safety
 Nº de referência: 6ES7516-3FN00-4AB2
- SIMATIC S7 CPU 1516-3 PN/DP
 Nº de referência: 6ES7516-3AN00-4AB3
- SIMATIC CPU 1512C PN com software e PM 1507
 Nº de referência: 6ES7512-1CK00-4AB1
- SIMATIC CPU 1512C PN com software, PM 1507 e CP 1542-5 (PROFIBUS) Nº de referência: 6ES7512-1CK00-4AB2
- SIMATIC CPU 1512C PN com software Nº de referência: 6ES7512-1CK00-4AB6
- SIMATIC CPU 1512C PN com software e CP 1542-5 (PROFIBUS) Nº de referência: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software para treinamento

- SIMATIC STEP 7 Professional V14 SP1 licença individual Nº de pedido: 6ES7822-1AA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 6 licenças para sala de aula Nº de pedido: 6ES7822-1BA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 6 licenças para upgrade Nº de pedido: 6ES7822-1AA04-4YE5
- SIMATIC STEP 7 Professional V14 SP1 20 licenças para estudantes Nº de pedido: 6ES7822-1AC04-4YA5

Por favor, note que os pacotes de treinamento podem ser substituídos por pacotes atualizados, quando necessário.

Um resumo dos pacotes SCE atualmente disponíveis pode ser encontrado em: <u>siemens.com/sce/tp</u>

Treinamentos avançados

Para treinamentos regionais avançados SCE Siemens, entre em contato com o parceiro SCE da sua região <u>siemens.com/sce/contact</u>

Outras informações sobre SCE

siemens.com/sce

Nota sobre o uso

A documentação de treinamento SCE para a solução de automação convencional Totally Integrated Automation (TIA) foi elaborada para o programa "Siemens Automation Cooperates with Education (SCE)" especificamente para fins educacionais em instituições públicas de ensino F&E. A Siemens AG não assume responsabilidade sobre o conteúdo.

Esta documentação somente deve ser utilizada para a formação inicial nos produtos/sistemas da Siemens. Isto é, ela pode ser copiada total ou parcialmente e ser entregue aos estudantes para a utilização no âmbito de seu treinamento. A distribuição, a reprodução desta documentação e a divulgação do seu conteúdo são permitidas apenas para fins educacionais nas instituições públicas de ensino.

As exceções demandam a aprovação por escrito do representante da Siemens AG: Roland Scheuerer <u>roland.scheuerer@siemens.com</u>.

As violações estão sujeitas a indenização por danos. Todos os direitos, inclusive da tradução, são reservados, particularmente para o caso de registro de patente ou marca registrada.

A utilização em cursos para clientes industriais é expressamente proibida. O uso comercial dos documentos não é autorizado.

Agradecemos à TU Dresden, principalmente ao Prof. Dr. Eng. Leon Urbas e à da empresa Michael Dziallas Engineering e a todos os outros envolvidos pelo auxílio durante a elaboração desta documentação de treinamento SCE.

Diretório

1	Oł	ojetivo	5
2	Re	equisito	5
3	Ha	ardwares e softwares necessários	6
4	Te	eoria	7
	4.1	Sistema operacional e programa de aplicativo	7
	4.2	Módulos organizacionais	8
	4.3	Imagem do processo e processamento cíclico do programa	9
	4.4	Funções	11
	4.5	Blocos de função e blocos de dados de instância	12
	4.6	Blocos de dados globais	13
	4.7	Blocos de código habilitados para biblioteca	14
	4.8	Linguagens de programação	15
5	De	efinição da tarefa	16
6	Pla	anejamento	16
	6.1	Parada de emergência	16
	6.2	Operação manual – Motor da correia em comando por impulsos	16
7	In	strução passo a passo estruturada	17
	7.1	Desarquivar um projeto existente	17
	7.2	Criação de uma nova tabela de variáveis	18
	7.3	Criação de novas variáveis dentro da tabela de variáveis	20
	7.4	Importação da "Tag table_sorting station" (Tabela de variáveis_sistema de classificação)	21
	7.5	Criação da função FC1 "MOTOR_MANUAL" para o motor da correia em comando por impulsos	25
	7.6	Determinar a interface da função FC1 "MOTOR_MANUAL"	27
	7.7	Programação do FC1: MOTOR_MANUAL	30
	7.8	Programação do módulo organizacional OB1 – Comando do funcionamento da correia para frente em operação manual	37
	7.9	Programação do módulo organizacional OB1 – Comando do funcionamento da correia para trás em operação manual	42
	7.10	Salvar e compilar o programa	44
	7.11	Carregar o programa	45
	7.12	Observar os módulos do programa	46
	7.13	Arquivamento do projeto	48
8	Lis	sta de verificação	49
9	Ex	ercício	50
	9.1	Definição da tarefa – Exercício	50
	9.2	Planejamento	50
	9.3	Lista de verificação – Exercício	51
1() Inf	formação adicional	52

FUNDAMENTOS DA PROGRAMAÇÃO FC

1 Objetivo

Neste capítulo irá conhecer os elementos fundamentais de um programa de comando – os *Módulos organizacionais (OB)*, as *Funções (FC)*, os *Módulos de função (FB)* e os *Módulos de dados (DB)*. Além disto, lhes será apresentada a programação de funções e de módulos de função *habilitadas para bibliotecas*. Irá conhecer a linguagem de programação *Plano de funções (FBD)* e a utilizará para a programação de uma função FC1 e um módulo organizacional OB1.

Os comandos SIMATIC S7 listados no capítulo 3 podem ser utilizados.

2 Requisito

Este capítulo baseia-se na configuração de hardware da SIMATIC S7 CPU1516F-3 PN/DP, porém também pode ser realizada com outras configurações de hardware, que possuam os cartões de entrada e saída digitais. Para a execução deste capítulo poderá, por ex., recorrer ao seguinte projeto:

SCE_PT_012_101_Configuração de hardware_CPU1516F.zap13

3 Hardwares e softwares necessários

- 1 Engineering Station: Pré-requisitos são hardware e sistema operacional (outras informações, vide Readme nos DVDs TIA Portal Installations)
- 2 Software SIMATIC STEP 7 Professional no TIA Portal a partir de V13
- Comando SIMATIC S7-1500/S7-1200/S7-300, p. ex. CPU 1516F-3 PN/DP a partir de Firmware V1.6 com Memory Card e 16DI/16DO assim como 2AI/1AO Nota: As entradas digitais deverão ser executadas em um painel de controle.
- 4 Conexão Ethernet entre Engineering Station e comando



4 Teoria

4.1 Sistema operacional e programa de aplicativo

O *sistema operacional* está contido em cada comando (CPU) e organiza todas as funções e processos da CPU, que não estão relacionados a uma tarefa específica de comando. As tarefas do sistema operacional incluem, por ex.:

- Execução do reinício (inicialização a quente)
- Atualização da imagem do processo das entradas e da imagem do processo das saídas
- Chamada cíclica do programa de aplicativo
- Registro dos alarmes e chamada dos OBs de alarme
- Identificação e tratamento das falhas
- Gestão dos espaços de memória

O sistema operacional é parte integrante da CPU e já está instalado nesta por ocasião do fornecimento.

O **Programa de aplicativo** contém todas as funções, que são necessárias para o processamento de sua tarefa específica de automação. As tarefas do programa de aplicativo incluem:

- Verificação dos pré-requisitos para um reinício (inicialização a quente) com o auxílio dos OBs de inicialização
- Processamento dos dados de processo, isto é, ativação dos sinais de saída dependendo dos estados dos sinais de entrada
- Resposta aos alarmes e entradas de alarme
- Processamento das falhas no decurso normal do programa

4.2 Módulos organizacionais

Os módulos organizacionais (OB) formam a interface entre o sistema operacionais do comando (CPU) e o programa de aplicativo. Eles são chamados pelo sistema operacional e controlam as seguintes operações:

- Processamento cíclico do programa (por ex., OB1)
- Comportamento de inicialização do comando
- Processamento do programa controlado por alarme
- Tratamento das falhas

Em um projeto deve existir pelo menos *um módulo organizacional para o processamento cíclico do programa.* Um módulo organizacional é chamado por um *evento de partida* como apresentado na Figura 1. Nesta ocasião, os OBs individuais possuem prioridades determinadas, para que, por ex., um OB82 possa interromper o OB1 cíclico para o tratamento da falha.



Figura 1: Eventos de partida no sistema operacional e chamada OB

Após a ocorrência de um evento de partida, são possíveis as seguintes respostas:

- Se o evento tiver sido atribuído a um OB, este evento inicia a execução do OB atribuído.
 Se a prioridade do OB atribuído for mais alta do que a prioridade dos OBs em execução no momento, este será executado imediatamente (interrupção). Se este não for o caso, primeiro será esperado até que o OB com a prioridade mais alta seja executado.
- Se o evento não tiver sido atribuído a nenhum OB, será executada a resposta previamente configurada do sistema.

A Tabela 1 apresenta alguns exemplos de eventos de partida para uma SIMATIC S7-1500, cujo(s) possível(eis) número(s) de OB e a resposta previamente configurada do sistema do módulo organizacionais não esteja presente no comando.

Evento de partida	Possível número OB	Resposta previamente configurada do sistema
Inicialização	100, ≥ 123	Ignorar
Programa cíclico	1, ≥ 123	Ignorar
Alarme de horário	10 até 17, ≥ 123	-
Atualização do alarme	56	Ignorar
O tempo de monitoramento do ciclo foi excedido uma vez	80	STOP
Alarme de diagnóstico	82	Ignorar
Erro de programação	121	STOP
Erro de acesso periférico	122	Ignorar

Tabela 1: Números OB para os diferentes eventos de partida

4.3 Imagem do processo e processamento cíclico do programa

Se no programa de aplicativo cíclico as entradas (E) e saídas (A) forem ativadas, então, geralmente os estados de sinal não são consultados diretamente pelos módulos de entrada/saída, mas serão acessados um espaço de memória da CPU. Este espaço de memória contém uma imagem dos estados de sinal e é designado como **Imagem do processo**.

O processamento cíclico do programa é realizado na seguinte sequência:

 No início do programa cíclico é consultado, se as entradas individuais conduzem ou não a tensão. Este status das entradas é salvo na **imagem de processo das entradas (PAE)**.
 Nisto, a informação 1 ou "High" é definida para as entradas conduzindo tensão e a informação 0 ou "Low" é definida para as entradas não conduzindo tensão.

2. O processador executa então o programa salvo no módulo organizacional cíclico. Nesta ocasião, é acessada a **imagem de processo das entradas (PAE**) lida anteriormente para a informação de entrada necessária e os resultados da interligação são gravados em uma assim chamada **Imagem de processo das saídas (PAA**).

3. No final do ciclo, a **Imagem de processo das saídas** (**PAA**) é transmitida como estado de sinal aos módulos de saída e estes são ligados ou desligados. Em seguida, continua-se novamente com o ponto 1.

1. Salvar o status das entradas em PAE.



3. Transmitir o status de PAA para as saídas.

Figura 2: Processamento cíclico do programa

Nota: O tempo que o processador necessita para esta sequência é chamado de tempo de ciclo. Este, por sua vez, depende da quantidade e do tipo de instruções e do desempenho do processador do comando.

4.4 Funções

Funções (FCs) são blocos de código sem memória. Você *não possui nenhuma memória de dados*, na qual possam ser salvos os parâmetros dos módulos. Por isto, no acesso a uma função, todos os parâmetros de interfaces devem ser conectados. Para salvar os dados de forma permanente, devem ser sido criados antes os blocos de dados globais.

Uma função contém um programa, que é executado sempre que a função for chamada por outro bloco de código.

As funções podem ser aplicadas, por exemplo, para as seguintes finalidades:

- As funções matemáticas que, dependendo dos valores de entrada fornecem um resultado.
- As funções de tecnologia, como ativações individuais com interligações binárias

Uma função também pode ser chamada diversas vezes em diferentes locais dentro de um programa.



Figura 3: Função com chamada a partir do módulo organizacional Main[OB1]

4.5 Blocos de função e blocos de dados de instância

Os blocos de função são blocos de código, que salvam suas variáveis de entrada, as variáveis de saída, as variáveis de passagem e também as variáveis estatísticas nos blocos de dados de instância de forma permanente, de modo que eles também **estejam disponíveis após a execução do bloco**. É por isto que eles também são chamados de blocos com "memória".

Os blocos de função também podem trabalhas com variáveis temporárias. Porém, as variáveis temporárias não são salvas no bloco de dados de instância, mas encontram-se disponíveis somente durante um ciclo.

Os blocos de função são usados em tarefas que não podem ser realizadas com funções:

- Sempre quando no bloco forem necessários os tempos e contadores.
- Sempre que uma informação tiver que ser armazenada no programa. Por exemplo, uma pré-seleção do modo de operação com um botão.

Os blocos de função somente são executados, quando um bloco de função for acessado por outro bloco de código. Um bloco de função também pode ser chamado diversas vezes em diferentes locais dentro de um programa. Desta forma, eles facilitam a programação de funções complexas, que se repetem com frequência.

A chamada de um bloco de função é denominada como instância. A cada instância de um bloco de função é atribuído um espaço de memória, que irá conter os dados com os quais o bloco de função irá trabalhar. Esta memória é disponibilizada pelos blocos de dados, que são automaticamente criados pelo software.

Também é possível disponibilizar a memória para diversas instâncias em um bloco de dados na forma de **instância múltipla**. O tamanho máximo dos blocos de dados de instância varia de acordo com a CPU. As variáveis declaradas no bloco de função determinam a estrutura do bloco de dados de instância. As variáveis declaradas no bloco de função determinam a estrutura do bloco de dados de instância.



Figura 4: Bloco de função e instância com chamada a partir do módulo organizacional Main[OB1]

4.6 Blocos de dados globais

Ao contrário dos blocos de código, os blocos de dados não contém nenhuma instrução, mas são usados para o armazenamento dos dados de usuário.

Portanto, nos blocos de dados estão contidos os dados variáveis, com os quais o programa de aplicativo trabalha. A estrutura dos blocos de dados globais pode ser livremente definida.

Os blocos de dados globais registram os dados, que podem ser utilizados *por todos os outros blocos* (consultar a figura 5). O bloco de dados de instância somente deve ser acessado pelo bloco de função pertinente. O tamanho máximo dos blocos de dados varia dependendo da CPU.



Figura 5: Diferença entre o bloco de dados global e o bloco de dados de instância.

Os exemplos de aplicação para os blocos de dados globais são:

- Armazenamento das informações de um sistema de armazenamento. "Que produto encontra-se onde?"
- Armazenamento de receitas referentes a determinados produtos.

4.7 Blocos de código habilitados para biblioteca

A criação de um programa de aplicativo pode ser realizada de modo linear ou estruturada. A **programação linear** grava todo o programa de aplicativo no OB do ciclo, porém, é adequado somente para os programas muito simples, os quais são utilizados, neste meio tempo, em outros sistemas de comando, de custos mais baixos, por ex., LOGO!.

No caso dos programas mais complexos, recomenda-se sempre uma **programação estruturada**. Aqui, a completa tarefa de automação pode ser dividida em pequenas tarefas parciais, para então solucioná-las em funções e blocos de função.

Nesta ocasião, deveriam ser criados preferencialmente os blocos de código habilitados para biblioteca. Isto é, que os parâmetros de entrada e de saída de uma função ou de um bloco de função sejam determinados de modo geral e somente com a utilização do bloco, sejam equipados com as variáveis globais atuais (entradas/saídas).



Figura 6: Função habilitada para biblioteca com chamada em OB1

4.8 Linguagens de programação

Para a programação das funções encontram-se disponíveis as linguagens de programação do plano de funções (FBD), plano de contato (LAD), lista de instruções (AWL) e Structured Control Language (SCL). Para os blocos de função, além disto, existe a linguarem de programação GRAPH, para a programação das cadeias de passos gráficas.

A seguir será apresentada a linguagem de programação Plano de funções (FBD).

FBD é uma linguagem de programação gráfica. A apresentação é sentida nos sistemas de circuito eletrônico. O programa é ilustrado nas redes. Uma rede recebe um ou vários atalhos de interligação. Os sinais binários e analógicos são interligados entre si pelas caixas. Para a apresentação da lógica binária são utilizados os símbolos lógicos gráficos conhecidos da álgebra booleana.

Com as funções binárias poderá consultar os operadores binários e interligar seus estados de sinal. Um exemplo das funções binárias são as instruções "Interligação E", "Interligação OU" e "Interligação EXCLUSIVA OU", como apresentado na Figura 7.



Figura 7: Funções binárias no FBD e tabela lógica pertinente

Com instruções simples poderá, por ex., controlar as saídas binárias, avaliar os flancos ou executar as funções de salto no programa.

As instruções complexas disponibilizam os elementos de programa, como por ex., os períodos IEC e o contador IEC.

A caixa vazia serve como curinga, na qual poderá selecionar a instrução desejada.

Mecanismo da entrada de liberação EN (enable) / saída de liberação ENO (enable output):

- Uma instrução sem o mecanismo EN/ENO e executada na caixa de entradas, independente do estado do sinal.
- As instruções com o mecanismo EN/ENO somente são executadas, quando a entrada de liberação "EN" apresentar o estado de sinal "1". No caso de execução correta da caixa, a saída de liberação "ENO" apresenta o estado de sinal "1". Assim que surgir um erro durante a execução, a saída de liberação "ENO" será zerada. Se a entrada de liberação EN não for conectada, a caixa sempre ser executada.

5 Definição da tarefa

Neste capítulo, as seguintes funções da descrição do processo do sistema de classificação serão planejadas, programadas e testadas:

Operação manual – Motor da correia em comando por impulsos

6 Planejamento

A programação de todas as funções em OB1 não é recomendadas por razões de clareza e de reutilização. O código do programa, por isto, é armazenado em grande parte nas funções (FCs) e nos blocos de função (FBs). Esta decisão, quais funções serão armazenadas nas FCs e quais o serão no OB1, será planejado a seguir.

6.1 Parada de emergência

A PARADA DE EMERGÊNCIA não necessita de nenhuma função própria. De mesmo modo que o modo de operação, o estado atual do relé da PARADA DE EMERGÊNCIA pode ser utilizado diretamente nos módulos.

6.2 Operação manual – Motor da correia em comando por impulsos

O comando por impulsos do motor da correia deve ser encapsulado em uma função (FC) "MOTOR_MANUAL". Deste modo, por um lado é fomentada a clareza em OB1, por outro lado, no caso de um ampliação do sistema em uma outra correia de transporte, a reutilização será possível. Na Tabela 2 são relacionados os parâmetros planejados.

Input	Tipo de dados	Comentário
Operação manual_ativa	BOOL	Modo da operação manual ativado
Botão_comando por impulsos	BOOL	Botão para ligar o motor da correia em comando por impulsos
Liberação_OK	BOOL	Todas as condições de liberação atendidas
Desativação de proteção_ativa	BOOL	Desativação de proteção ativa, por ex., parada de emergência acionada
Output		
Motor da correia_comando por impulsos	BOOL	Ativar o motor da correia em comando por impulsos

Tabela 2: Parâmetros para FC "MOTOR_MANUAL"

A saída motor da correia_comando por impulsos está ligada, enquanto o botão_comando por impulsos for pressionado, o modo de operação manual estiver ativado, a liberação tiver sido concedida, e a desativação de proteção não estiver ativa.

7 Instrução passo a passo estruturada

A seguir, encontrará uma instrução, como poderá implementar o planejamento. Se já tiver conhecimento suficiente, os passos numerados já serão suficientes para o processamento. Caso contrário, simplesmente siga os seguintes passos detalhados na instrução.

7.1 Desarquivar um projeto existente

→ Ante que possa começar com a programação da função (FC) "MOTOR_MANUAL", necessitará de um projeto com uma configuração de hardware. (z.B. SCE_PT_012-101_Configuração de hardware_S7-1516F_R1502.zap) Para desarquivar um projeto existente, deverá pesquisar a partir da visualização do projeto, em →Project (Projeto) →Retrieve (Extrair), pelo respectivo arquivo. Confirme sua seleção em seguida com Open (Abrir). (→ Project (Projeto) → Retrieve (Extrair) →Selection of a .zap archive (Seleção de uma arquivo .zap) → Open (Abrir).)

Project	Edit	View	Insert	Online	Options						
New.	 I	ect			Ctrl+0						
Close	ic proj	c c c c c c c c c c c c c c c c c c c			Ctrl+W						
Save Save	Save Ctrl+S Save as Ctrl+Shift+S										
Delet Archi	e proje ve	ct			Ctrl+E						
Retrie	eve										
Card	Reader ory car	/USB mi d file	emory		;						
Upgr	ade										
D:lAu D:lAu D:lVo D:lli D:lAu	tomati tomati rlagen 032-20 tomati	_101_CPU _101_CPU \Tank_V1 erung_S7- 11500_V13	314C 1516F 3_SP1 314 3_SP1								
Exit				121172							

 → Como próximo passo, pode ser selecionado o diretório de destino, no qual o projeto desarquivado deverá ser salvo. Confirme sua seleção com "OK". (→ Target directory (Diretório de destino) →OK)

7.2 Criação de uma nova tabela de variáveis

→ Navegue na visualização do projeto até variantes → PLC de seu controller e crie uma tabela de variantes, clicando duas vezes em → Add new tag table (Adicionar nova tabela de variantes) "clique duplo".

VA	Siemens - D:\Automation\012_101_CPU1516F\01	2_101_	CPU1	516F				
Pro	oject Edit View Insert Online Options Too P 🎦 🗔 Saveproject 进 🐰 🏥 🗊 🗙 🍤 🖢 🕻	ls Wini ≝≛≛	dow	Help	RT .	🖋 Go onlin	e 🖋	Go
	Project tree							
	Devices							
	B 0 0	🔲 🔿						
	▼ 7 012_101_CPU1516F	~						
ŧ.	Add new device							
Sta	品 Devices & networks							
	CPU_1516F [CPU 1516F-3 PN/DP]							
	Device configuration							
	😧 Online & diagnostics							
	🕨 🚘 Program blocks							
- 22	🕨 🚂 Technology objects	=						
	🕨 词 External source files							
	🔻 🞑 PLC tags							
	a Show all tags							
	🗳 Add new tag table							
	🝯 Standard-Variablentabelle [54]							
	PLC data types							
	Watch and force tables							
	🕨 🙀 Online backups							
	🕨 🔄 Traces							
	🔤 Program info							
	Device proxy data							
	🖂 PLC alarms							
	Text lists							
	Local modules		l.					
	🕨 🚺 Common data					I' Elect		

→ Nomeie a tabela de variantes recém-criada como "Tag table_sorting station" (Tabela de variantes_sistema de classificação). (→ Clique no lado direito em "Tag table_1" (Tabela de variantes_1) → "Rename" (Renomear) → Tag table_sorting station (Tabela de variantes_sistema de classificação))

M Siemens - D:\Automation\012_101_CPU1516	F/012_101_CPU1516F
Project Edit View Insert Online Options 📑 🎦 🔒 Save project ا 🐰 🗐 🗐 🗙 🏷	Tools Window Help 🛨 (🍽 🗄 🔃 🗓 🖳 🞇 💋 Goonline 💋 Gooffi
Project tree Devices	
 ✓ ☐ 012_101_CPU1516F ≦ Add new device 	<u>^</u>
Devices & networks CPU_1516F [CPU 1516F-3 PN/DP] Device configuration Online & diagnostics Program blocks	≡
Figure Technology objects Figure Technology Figur	
Add new tag table Standard-Variablentabelle [54]	
Cata types Galaxy Watch and force tables Galaxy Online backups Galaxy Traces More Records in force	

→ Abra-a em seguida com um clique duplo. (→ Tag table_sorting station (Tabela de variantes_sistema de classificação))

Siemens - D:\Automation\012_101_CPU1516F	N012_101_CPU1	1516F										3	-
Project Edit View Insert Online Options	Tools Window	Help								Tot	ally Int	egrated Automation	
🌁 🎦 🔒 Save project 🔳 🐰 💷 🗐 🗙 🍤 🗄	e 🗠 🗄 🗓	694 💋	Go online 🖉 Go off	ine 🗛 🖪							any me	PORT	ΤAL
Project tree		2_101_CPU1516	F ► CPU_1516F	[CPU 1516F-	-3 PN/DP] > PLC	tags ▶ Ta	g table_	_sorting	station [0]			_ # #	×
Devices										-	Tags	User constants	s
1900		2 🖻 🕆 🕦	:									E	4
		Tag table sortin	ng station										
▼ 1 012 101 CPU1516F		Name		Data type	Address	Retain	Visibl	Acces	Comment				
Add new device	1	<add net<="" td=""><td>N></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></add>	N>										
A Devices & networks					(mail)			~					
CPU 1516F [CPU 1516F-3 PN/DP]													
Device configuration													
V. Online & diagnostics													
Program blocks	=												
Technology objects													
External source files													
👻 🚂 PLC tags													
Show all tags													
Add new tag table													
💥 Standard-Variablentabelle [54]													
Tag table_sorting station [0]													
City PLC data types													
Watch and force tables								G	Properties	*i Info	& Dia	anostics 🛛 🗖 🗕	
Online backups													
🕨 📴 Traces		seneral											-
Be Program info		ag	Tar										1
Device proxy data													• 4
PLC alarms			G	eneral									
🗐 Text lists													
Local modules			1		Name:								
🕨 🙀 Common data					Data type:								
Documentation.cettings	~		f		Address								
✓ Details view						Retained							
					Comment:								
Data type Comment Name													V

7.3 Criação de novas variáveis dentro da tabela de variáveis

 → Acrescente o nome Q1 e confirme a entrada com a tecla Enter. Se você não criar nenhuma outra variante, o TIA Portal irá fornecer automaticamente o tipo de dado "Bool" e o endereço %E0.0 (I 0.0). (→ <Add new> (<Adicionar>) →Q1 → Enter)

	ogran	nming • CPU_1	STOP [CPU	1516F-3 PN/	UP] V	PLC tag	s 🕨 Tag	able_sorting st	User o	constants
Ŷ	Tag t	De conting et	ation							
	ray t	Name	Data type	Address	Retain	Visibl	Acces	Comment		
1	-	Q1	Bool 🔳	%10.0						
2		<add new=""></add>								

→ Altere o endereço para %A0.0 (Q0.0), especificando-o diretamente ou por meio de clique sobre a seta Dropdown, abrir o menu para o endereçamento, ali alterar a identificação do operador para A e confirmar com Enter ou um clique sobre o sinal. (→ %E0.0 → Operand identifier (Indicação de operação)→ A →

rog	gran	nming + CPU_1	516F [CI	PU	1516F-3	PN/	DP] 🕨	PLC tage	s 🕨 Tag	g table_sorting station [1] 🛛 🗖 🔳 🗙
										Tags 🔳 User constants
1 1 1	j.	🔿 😳 🖬								
Т	ag t	able_sorting sta	ation							
		Name	Data ty	pe	Address		Retain	Visibl	Acces	Comment
1	-	Q1	Bool		%10.0	-				
2		<add new=""></add>				c	operand Oper Bi	identifier: rand type: Address: t number:	I Q M	

→ Faça o comentário para a variante "conveyor motor -M1 forwards fixed speed" (motor de correia -M1 velocidade fixa de movimento para frente).

ro	gran	nming > CPU_1	516F [CPU	1516F-3 PN	/DP] ►	PLC tag	s ▶ Taç	g table_sorting st	ation [1]	_ • • ×
								- Tags	User (constants
#		🖻 😤 🕅								
1	lag t	table_sorting st	ation							
1000		Name	Data type	Address	Retain	Visibl	Acces	Comment		
1	-	Q1	Bool 🔳	%Q0.0 -				conveyor motor -M1	forwards fixe	ed speed
2		<add new=""></add>				1	 Image: A start of the start of			

 → Acrescente na linha 2 uma nova variável Q2. TIA Portal concedeu automaticamente o mesmo tipo de dado como na linha 1 e incrementou o endereço em 1 para %A0.1 (Q0.1).
 Faça o comentário "conveyor motor -M1 backwards fixed speed" (motor de correia -M1 velocidade fixa de movimento para trás).

 $(\rightarrow$ <Add new> (<Adicionar>) \rightarrow Q2 \rightarrow Enter \rightarrow Comentário \rightarrow conveyor motor -M1 backwards fixed speed (motor de correia -M1 velocidade fixa de movimento para trás)

								Tags	User constants
猎	- E	🕈 🙄 🕅 🗶							8
1	ag ta	ble_sorting st	ation						
	h	lame	Data type	Address	Retain	Visibl	Acces	Comment	
	-	Q1	Bool	%Q0.0				conveyor motor -M1	forwards fixed speed
	9	Q1 Q2	Bool Bool	%Q0.0 %Q0.1				conveyor motor -M1 conveyor motor -M1	forwards fixed speed backwards fixed speed

7.4 Importação da "Tag table_sorting station" (Tabela de variáveis_sistema de classificação)

→ Para inserir uma tabela de símbolos existente, clique com o botão direito do mouse sobre um espaço vazio da "Tag table_sorting station" (Tabela de variáveis_sistema de classificação) aplicada. No menu de contexto, selecione "Import file" (Arquivo de importação).

(→ Clique no lado direito em um espaço vazio da tabela de variáveis →Import file (Arquivo de importação))

Devices Image: Devices Image: Devices <td< th=""><th>E 🔺re</th><th>gramming 🕨 CPU_1516F [CPU 1516F-3 PN/DP] 🔸 PLC tags 🔸 Tag table_sorting station [2] 🛛 🗕 🖬 🗮 🗙</th><th></th></td<>	E 🔺re	gramming 🕨 CPU_1516F [CPU 1516F-3 PN/DP] 🔸 PLC tags 🔸 Tag table_sorting station [2] 🛛 🗕 🖬 🗮 🗙	
Image: Second		Tags 🗉 User constants	Options
Image: Disc product of the second of the	🗐 🔿 🥩	🖉 🖻 🥶 int	
032_100_FC-Programming Add new device Comment		Tag table sorting station	 Find and replace
Image: Constraint of the backups Image: Constraint of	ing A	Name Data type Address Retain Visibl Acces Comment	This and replace
Concept m. Concep	1	GI 01 Bool %00.0 C conveyor motor MI forwards fixed speed	Find:
Concentration Concent	2	a Q2 Bool %Q0.1 🔽 🔽 conveyor motor -M1 backwards fixed speed	
Im Device configuration Wonite & diagnostics Wonite & diagnostics Program blocks Cut Cut+C Im Extennology objects Im Extennology obj	16F-3 PN/DP] 3	Add news	D Whole works calls
Wonline & diagnostics Image: Add row Image: Add row Image: Add row Image: Add row row row row row Image: Add row Image: Add row	tion =	gir Insert row	
Cott Critic C	tics	≓ [#] Add row	Match case
Image: Technology objects Image: Technology objects Image: Technology objects Image: Technology objects <td></td> <td>X Cut Ctrl+X</td> <td>Find in substructures</td>		X Cut Ctrl+X	Find in substructures
Import file Import file <t< td=""><td>ts</td><td>Copy Ctrl+C</td><td>Find in hidden texts</td></t<>	ts	Copy Ctrl+C	Find in hidden texts
Concept and for the set of t	les	Paste Ctrl+V	11se wildcards
Show all tags Rename F2 Standard Variablentabelle [S4] Cross-reference information Shifs-F11 Tag table_sorting station [2] "Monitor all Import Size Export Size Import Size Export Size Import Size Export Size Import Size Import Size <t< td=""><td></td><td>X Delete Del</td><td></td></t<>		X Delete Del	
Add new tag table Standard Variablemabelle [54] Sandard Variablemabelle [55] Sanda		Rename F2	Use regular expressions
Standardvariablentabelle [54] Cost-reference mormation Sinte-F1 Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting station [2] Image: Sorting stati	able	and the second se	O Whole document
type Comment Name conveport of Q1	ablentabelle [54]	LT0534Elefence information Shill+Fill	Emmonstrate acceltant
Import file Export file <t< td=""><td>ting station [2]</td><td>Monitor all</td><td>O rion current postada</td></t<>	ting station [2]	Monitor all	O rion current postada
Watch and force tables Export file Details view type Comment Name Conveyor O		Import file	Oselection
Image: Second	tables	Export file	Down
Its fraces Its fraces <td></td> <td>G Properties</td> <td>Que</td>		G Properties	Que
Image: Strength and Strengt			0.00
Details view Re type Comment I convejorm Q	~		Find
t type Comment Name			Replace with:
atype Comment Name .			
ype connegor	Diama		Deplace Deplace all
	a 01		
< III >		< m >	

→ Selecione a tabela de símbolo desejada (por ex., no formato .xlsx) e confirme a seleção com "Open" (Abrir).

 $(\rightarrow SCE_PT_020-100_tabela de variáveis_sistema de classificação... \rightarrow Open (Abrir))$

→ Se a importação tiver sido concluída, receberá uma janela de confirmação com a possibilidade de visualizar o arquivo de protocolo referente à importação. Clique aqui sobre → OK.

Import co	ompleted with warnings (0032:000031) $ imes$
	Import completed with warnings.
_	Detailed information is shown in the import log file.
	Click here to view the log file.
	ОК

- → Poderá ver, que alguns endereços foram destacados em laranja. Este existem em duplicidade e os nomes das variáveis pertinentes foram numerados automaticamente, para evitar ambiguidades.
- → Exclua as variáveis existentes em duplicidade, marcando as linhas e pressionar a tecla Delete no seu teclado ou selecionando no menu de contexto o ponto Excluir.

 $(\rightarrow$ Clique no lado direito nas variáveis marcadas \rightarrow Delete (Excluir))



→ Agora terá diante de si uma tabela de símbolos completa das entradas e saídas digitais. Salve o seu projeto pelo nome 032-100_FC-Programação.

 $(\rightarrow$ Project (Projeto) \rightarrow Save as ... (Salvar em ...) \rightarrow 032-100_Programação-FC \rightarrow Save (Salvar))

Siemens - D:\Automation\012_101_	CPU151	6F\012_10	1_CPU	1516F									- "
Project Edit View Insert Online	Options	Tools V	Vindow	Help								Totally Integrated Automat	ion
ir New	2	± (²⁴ ±	副 🖪		🛛 🖬 💋 (io online 🚀 G	o offline	?	×			PO	RTAL
Y Open	Ctrl+O		01	2 101	CPU1516	F ▶ CPU 15	16F [CPU	1516F-3	PN/DP1	▶ PICt	ans Tag table sorting station [28]		Ξ×
Mgrate project					_0101510		tor [ci o		T WETT		age rag table_serting station (co)		
Close	Ctn+w											Tags 🗉 User consta	nts
- Save	Ctrl+S		1	1	🔿 😳 🕅								
Save as Ctrl+5	hift+S			Tanta	able sortin	a station							
Delete project	Ctrl+E			rug a	Name	Data type	Address	Retain	Visibl	Acces	Comment		
Archive			1	a	-A1	Bool	SID 0	-			return signal emergency stop ok (nc)		
Retrieve			2		-*0	Bool	\$40.1				main switch ON" (no)		-
Card Beader/USB memory			3	-00	-50	Bool	\$10.7				mode selector manual(0) (automatic(1)		
Memory card file			4	-	-51	Bool	950.3				pushbutton automatic start (no)		
			5	-	.52	Bool	510.4				pushbuttop automatic stop (nc)		
Upgrade			= 6	-111	-81	Bool	510.5				sensor cylinder -M4 retracted (no)		
Print	Ctrl+P		7	-	-82	Bool	%10.6				sensor cylinder -M4 extended (nc)		
Print preview			8		-83	Bool	%10.7				sensor motor -M1 actice (pulse signal for		-
D:lAutomation\012_10\012_101_CPU15	516F		9	-01	-84	Bool	%11.0				sensor part at slide (no)		
C:lUsers\spe\Do\032_100_FC-Programm	ning		10	-070	-85	Bool	%11.1				sensor metal part (no)		_
D:\Automation\013_10\013_101_CPU31	14C		11	-	-86	Bool	%11.2				sensor part in front of cylinder -M4 (no)		
D:\Vorlagenprojekt_Webserv\Tank_V13	SP1		12	-01	-87	Bool	%1.3				sensor part at end of conveyor (no)		
D:l\032-200_FB-Programmierung_S7-31	14		13	-00	-53	Bool	%11.4				pushbutton manual mode conveyor -M1		-
D:\Automatisi\012-100_CPU1500_V13_	SP1		14	-01	-54	Bool	%11.5				pushbutton manual mode conveyor -M1		- 11
Exit			15	-0	-55	Bool	%11.6				pushbutton manual mode cylinder -M4 re		
Online backups			16	-00	-56	Bool	%11.7				pushbutton manual mode cylinder -M4 ex		
Fraces			17	-01	-01	Bool	%00.0				conveyor motor -M1 forwards fixed speed		
Program info			18	-00	-02	Bool	%00.1				conveyor motor -MI backwards fixed speed		
Device proxy data			19	-00	-03	Bool	%00.2				conveyor motor -M1 variable speed		
PLC alarms			20		-M2	Bool	%Q0.3				cylinder -M4 retract		- 11
Text lists			21	-	-M3	Bool	%Q0.4				cylinder -M4 extend		
Local modules			22	-00	-P1	Bool	%Q0.5				display main switch on		
Common data			23	-00	-P2	Bool	%Q0.6				display "manual mode"		
Documentation retting			* 24	-	-P3	Bool	%Q0.7				display "automatic mode"		
✓ Details view			25	-00	-P4	Bool	%Q1.0				display "emergency stop activated"		
			26	-01	-P5	Bool	\$011				display, automatic mode started"		~
Data type Comment Name											Q Properties	Info 🚯 🖏 Diagnostics	
A Destal stress	. 1	Tentil	bla.ec										

7.5 Criação da função FC1 "MOTOR_MANUAL" para o motor da correia em comando por impulsos

→ Clique no Portal, na seção Programação PLC em "Add new block" (Adicionar novo bloco), para criar aqui uma função.

FC)

 $(\rightarrow$ PLC programming (Programação PLC) \rightarrow Add new block (Adicionar novo bloco) \rightarrow

mens - D:\Automa	ation\032_100_FC-Programming\032_100_FC-P	rogramming	- *
			Totally Integrated Automation PORTAL
	Device: CPU_1516F	Add new block	
Devices & networks	Show all objects Add new block	Name: Block_1	
PLC programming	٠	Language: SCL Number: 1	
Motion & echnology	*	Organization block O Automatic	
arameterizatio	on Show cross-reference	s Description: Functions are code blocks or subroutines without dedicated in	nemory.
isualization	Show program structu		-
Diagnostics		Function	
	Melp		
		Data block More	
		Additional information	

 → Denomine seu novo módulo com o nome: "MOTOR_MANUAL", comute a linguagem para FBD e deixe conceder o número automaticamente. Ative a marca de verificação "Add new and open" (Adicionar novo bloco e abrir), e automaticamente você chegará na visualização do projeto do seu bloco de função criado.Clique em "Add" (Adicionar).

 $(\rightarrow$ Name (Nome): MOTOR_manual \rightarrow Language (Idioma): FBD \rightarrow Number (Número): automaticamente, \rightarrow Add new and open (Adicionar novo bloco e abrir) \rightarrow Add (Adicionar))

Name:			
MOTOR_MANUAL	-		
	Language:	FBD	
OB	Number:	1	
Organization		🔿 Manual	
DIOCK		 Automatic 	
FB	Description:		
Function block	Functions are o	code blocks or subroutines without dedicated mem	iory.
FC			
Function			
Function			
Function			
Punction			
Data block			

7.6 Determinar a interface da função FC1 "MOTOR_MANUAL"

- → Clicando em "Add new and open" (Adicionar novo bloco e abrir)", a visualização do projeto abre com uma janela para configuração do bloco recém-criado.
- → Na seção superior de sua vista da programação poderá encontrar a descrição da interface de sua função.

in the second se	ы	x # # % E 🗆 🖬 🗐		୧୦ _{ଦେ} ଏଥି ହଥ	I I I 0 0°	
	MC	OTOR_MANUAL				
		Name	Data type	Default value	Comment	
1	-	🔹 Input				
2		Add new>				
3	-	 Output 				
4		Add new>				
5	-	InOut ▼ InOut				
6		Add new>				
7	-	Temp				
8		Add new>				
9	-	Constant				
10		Add new>				
11	-	🔻 Return				
12	-	MOTOR_MANUAL	Void			
	۲					>
•	Blo	ock title:	here of here			~
C	om	nment				
•	1	Network 1:				
	C	Comment				_
-				(a) (

→ Para a ativação do motor da correia é necessário um sinal de saída binário. Por isso, criamos primeiro a variável output #Conveyor_motor_manual_mode (Motor de correia_comando por impulsos) do tipo "Bool". Aos parâmetros efetue o comentário "Control of the conveyor motor in manual mode (Acionar motor de correia no comando por impulsos).

(\rightarrow Output: Conveyor_motor_manual_mode (Motor de correia_comando por impulsos \rightarrow Bool \rightarrow Control of the conveyor motor in manual mode (Acionar motor de correia no comando por impulsos))

(C-Pro	ogi	ramming + CPU_1516F [C	PU 1516F-3 PN/DP]	 Program bl 	ocks • MOTOR_MANUAL [FC1]	_ # = ×
ið	13	1	🖗 🔹 💺 🖿 🚍 💬	2 ± 2 ± 🖃 😥	¢° 6₀ ₫ 6≣	⊉ I ₌ ¹ = 0, 05 Ⅲ	
	MC	T	DR_MANUAL				
		Na	me	Data type	Default value	Comment	
1	-00	•	Input				^
2			<add new=""></add>				
3	-	-	Output				
4	-00		Conveyor_motor_manual	Bool		Control of the conveyor motor in manual	mode 📃 🗐
5			<add new=""></add>				
6	-	•	InOut				
7			<add new=""></add>				
8	-	•	Temp				
9			<add new=""></add>				
10		*	Constant				
11			<add new=""></add>				
12	-	•	Return				
13	-		MOTOR MANUAL	Void			~
	1 < 1				111		>

 → Acrescente como interface de entrada em Input primeiro o parâmetro #Operação manual_ativa e confirme com a entrada com a tecla Enter ou saindo do campo de entrada. Será concedido automaticamente o tipo de dados "Bool". Este será mantido. Faça em seguida o devido comentário "Manual mode activated" (Tipo de operação manual ativado).

 $(\rightarrow Manual_mode_active (Operação_manual_ativo \rightarrow Enter \rightarrow Bool \rightarrow Manual mode activated (Tipo de operação manual ativado))$

→ Acrescente em Input, como outro parâmetro de entrada binário #Botão_Comando por impulsos, #Liberação_OK e #Desativação de proteção_ativa e verifique seus tipos de dados. Complemente com comentários razoáveis.

	C-Pro	ogr	amming + CPU_1516F [Cl	PU 1516F-3 PN/DP]	Program bl	ocks 🕨 MOTOR_MANUAL [FC1] 💦 🖃 🗮 🗙
13		1	2 2 k 🐂 🎮 📟 💬 :	君 : 요 : 日 診	eo 65 en 68	
6.4	MC	т	OR MANUAL			• = = • > = =
		Na	me	Data type	Default value	Comment
1	-	•	Input			
2	-		Manual_mode_active	Bool		Manual mode activated
3	-		Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
4	-		Enable_OK	Bool		All enable conditions OK
5	-		Safety_shutoff_active	Bool 🔳		Safety shutoff active e.g. emergency stop operated
6			<add new=""></add>			
7	-	•	Output			
8	-		Conveyor_motor_manual	Bool		Control of the conveyor motor in manual mode
9			<add new=""></add>			
10		•	InOut			
11			<add new=""></add>			
12	-	•	Temp			
13			<add new=""></add>			
14	-	•	Constant			
15			<add new=""></add>			
16	-	•	Return			
17	-		MOTOR_MANUAL	Void		

→ Para a documentação do programa conceda o título do módulo, um comentário sobre o módulo e para a rede 1 um título de rede útil.

(\rightarrow Block title (Título do bloco): Conveyor motor in manual mode (Acionamento de motor em operação manual) \rightarrow Network 1 (Rede 1): Ativar o motor da correia em comando por impulsos)

	C-Pr	ogr	ammii	ng≯	CPU_	_1516	F [CI	PU 1516F-3	PN/DP]	 Program bl 	ocks → MOTOR_MANUAL[FC1] _ I	
ю	н	X =	¢ ₹	B 20			9	8 ± 2 ± (:=) 🎲	¢° ⊊₀ ¢≣ ⊊≣	⊉ I _≣ % 0, ∞ ∐	
	M	ото	R_MA	NUA	L							
		Nar	me					Data type		Default value	Comment	
1	-00	•	Input									^
2	-00		Ma	nual_	mode_	active		Bool			Manual mode activated	1
3	-00		Pus	hbutt	ton_ma	anual_	mode	Bool			Pushbutton manual mode conveyor on	
4	-00		Ena	able_(ОК			Bool			All enable conditions OK	
5	-00		Sat	fety_sl	hutoff_	active		Bool			Safety shutoff active e.g. emergency stop ope	rated
6			<ac< td=""><td>id nev</td><td>N></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ac<>	id nev	N>							
7	-00	•	Outpu	t								
8	-		Con	nveyo	r_moto	r_man	ual	Bool			Control of the conveyor motor in manual mod	e
•	1	-								101		×
	1.									• 1 • • 1		
		> = 1	7?	ч	-01	↦	-[-]					
•	Blo	ock 1	title:									^
•	Conv Conv	/eyo litior /eyo	r motor ns are g r_moto	r in mi grante r_mai	anual r ed and nual_m	node: the sa node is	lf the fety sl activ	pushbutton_i nutoff is not a ated	manual_r ctivated	node is operated. the output	the enable	
•		Net	work 1	: Co	ontrol o	f the c	onvey	or motor in n	nanual m	ode		
	0	Com	ment									-

7.7 Programação do FC1: MOTOR_MANUAL

→ Abaixo da descrição das interfaces poderá ver na janela de programação uma barra de símbolos com diversas funções lógicas e, entre elas, uma área com redes. Ali nós já determinamos o título do módulo e o título para a primeira rede. No interior das redes, a programação é realizada utilizando-se os módulos lógicos individuais. A divisão em várias redes serve, nesta ocasião, para preservar a clareza. As diversas possibilidades para acrescentar módulos lógicos serão apresentadas a seguir.

→ No lado direito de sua janela de programação poderá ver uma lista de instruções, que poderá utilizar no programa. Busque em → Basic instructions (Instruções simples) → conexões de bit conforme a função –[=] (Atribuição) e arraste estas para sua rede 1 (linha verde aparece, mouse com símbolo +).

(\rightarrow Instructions (Instruções) \rightarrow Basic instructions (Instruções simples) \rightarrow Bit logic operation (Conexão bit) $\rightarrow -[=]$)

					Options	
a 🛪 🔿 🔿 🍬 📰 🗖 🗖 🖗) # ± # ± 🖂 🗊	ලට ද ු දිසි ද ස	15 L L & Th		· · ·	1111
MOTOR MANUAL					> Favorites	
Name	Data type	Default value	Comment		✓ Basic instructions	_
🤕 🔻 Input				^	Name	
- Manual_mode_active	Bool		Manual mode activated		General	
📹 = Pushbutton_manual_mod	le Bool		Pushbutton manual mode conveyor on	-	Bit logic operations	
📹 = Enable_OK	Bool		All enable conditions OK		IT &	
	Bool		Safety shutoff active e.g. emergency stop operated		FT >=1	
🕣 🔻 Output					Fix	
	Bool		Control of the conveyor motor in manual mode			- 1
2		IIII		Ť		
	1	A 19 19 19			ET -[8]	
>=1 [??] -1 -011	-1					
	-			_	I SET BE	18
Block title:				^	RESET BE	
Conveyor motor in manual mode: If th	e pushbutton_manual_r	mode is operated,	, the enable		I SR	
convevor motor manual mode is act	ivated	the output				
Network 1: Control of the conv	eyor motor in manual m	ode		=		18
Commont						
comment						
Ê						1
~				-		
Č.						v
					< III 13	>
					> Extended instructions	
						_
2.20				V	I echnology	

→ Arraste para o seu parâmetro Output #Motor da correia_Comando por impulsos por meio de Drag and Drop para <??.> sobre o seu bloco recém inserido. Poderá selecionar mais facilmente um parâmetro na descrição de interfaces, segurando-o pelo símbolo azul

 $(\rightarrow \blacksquare$ Conveyor_motor_manual_mode (Motor de correia_comando por impulsos))

2000	R	掌掌喙Ѐ⊟ੋ፟፟፟፟፟	웹: ֎: 드 🗊	°° €₀ (≣ ¶	
M	отс	R_MANUAL			-
-	Na	me	Data type	Default value	Comment
1	-	Input	Deal		Manual marks and marks
		Manual_mode_active	Bool		Manual mode activated
	-	Fushbutton_manual_mode	Bool		All onable conditions OK
	-	Enable_OK	Bool		All enable conditions OK
	-	Output	5001		Salety shuton active e.g. emergency stop operated
-50		Conveyor motor manual	Bool		Control of the conveyor motor in manual mode
	-				
				×	
8	>=1	1771ol L+ -[=]			
•	> = 1	[??] → -0 ↦ -[=]			
Blo	>=1	1 [7] <mark> 01 → -[=]</mark> title:			
Blo	>=1	title:	pushbutton_manual_r	node is operated	, the enable
Blo	>=1 ck /eyo litio	Image: The second s	pushbutton_manual_i nutoff is not activated ated	node is operated the output	, the enable
Blc Con Con Con	>=1 ock /eyo litio /eyo	Image: Tree in the second	pushbutton_manual_ nutoff is not activated ated	node is operated the output	, the enable
Blc Conv Conv Conv	>=1 ock /eyo litio /eyo	Image: Tree intermediate Image: Tree intermediate Image: Tree intermediate r motor in manual mode: If the intermediate If the intermediate r motor intermediate Image: Tree intermediate Image: Tree intermediate r motor intermediate Image: Tree intermediate Image: Tree intermediate r motor intermediate Image: Tree intermediate Image: Tree intermediate r motor intermediate Image: Tree intermediate Image: Tree intermediate work 1: Control of the convey	pushbutton_manual_n utoff is not activated ated or motor in manual m	node is operated the output ode	, the enable
Blc Cont Cont Cont	>=1 ock /eyo litio /eyo Net	Image: Transmission of the sector Image: Transmission of the sector r motor in manual mode: If the ns are granted and the safety sl r_motor_manual_mode is activ r_motor_manual_mode is activ work 1: Control of the convey	pushbutton_manual_ nutoff is not activated ated or motor in manual m	node is operated the output ode	, the enable
Blc Cont Cont Cont	>=1 ck /eyo litio /eyo Net	Image: Transmission of the convey ment	pushbutton_manual_ nutoff is not activated ated or motor in manual m	node is operated the output ode	, the enable

→ Deste modo é determinado, que o parâmetro #Motor da correia_Comando por impulsos seja gravado através deste bloco. No entanto, ainda faltam as condições de entrada, para que isto realmente aconteça. Arraste o parâmetro de input #Manual_mode_active (Operação manual_ativo) e solte "..." no lado esquerdo do bloco de alocação.

OI IOI	9 9 % E E E E D	월: 월: 달 달 달	Cº 60 CE 93	
MOTO	DR_MANUAL			
Na	ame	Data type	Default value	Comment
<u>-</u>	Input			
<u>.</u>	Manual_mode_active	Bool		Manual mode activated
•	Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
-	Enable_OK	Bool		All enable conditions OK
	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operate
	Output			
	Conveyor_motor_manual	Bool		Control of the conveyor motor in manual mode
<				
Block Conveyo conditio Conveyo	title: or motor in manual mode: If the p ons are granted and the safety sh or_motor_manual_mode is activ	pushbutton_manual_r hutoff is not activated ated	mode is operated, the output	the enable
Net	twork 1: Control of the convey	or motor in manual m	ode	

(→
Manual_mode_active (Operação manual_ativo))

 → A entrada do bloco de atribuição deve ser interligado adicionalmente por e com os outros parâmetros. Para tal, clique primeiro sobre a entrada do bloco, na qual já esta conectado #Operação manual_ativa, de modo que o traço de entrada apresenta um fundo azul.

→ Clique sobre o símbolo a na sua barra de símbolos lógicos, para inserir uma interligação E entre as variáveis #Operação manual_ativa e o módulo de ramificação.

8	> = 1	·??	٦	-01	↦	-[=]	
	AND	ogic op	erati	on [Shi	ift+F2]		
Col Col	nveyor ndition nveyor	motor s are g _motor	in ma rante _mar	anual r d and nual_m	node: the sa iode is	the pushbutton_manual_mode is operated, ty shutoff is not activated the output ctivated	the enable
	Netv	vork 1	Co	ntrol o	f the c	nveyor motor in manual mode	
	Comn	nent					
				#I mot	Conveg or_ma mode	r_ ual_	
	#Mar	iual_m a	ode_ ctive	_	/=	_	

→ Efetue um clique duplo sobre a segunda entrada da interligação & <??.?> e, no campo exibido a seguir, digite a letra "T", para visualizar uma lista das variáveis disponíveis, que começam com a letra "T".Clique na variável #Pushbutton_manual_mode (Tecla_comando por impulsos) e confirme com → Enter.

 $(\rightarrow$ Block & (Bloco &) $\rightarrow <??? \rightarrow T \rightarrow #Pushbutton_manual_mode (Tecla_comando por impulsos) <math>\rightarrow$ Enter)

C-	Pro	ogramming + CPU_1516F [CF	PU 1516F-3 PN/DP]	 Program blo 	ocks → MOTOR_MANUAL [FC1] 🛛 🗕 🖬	∎ ×
ю,	K		2 ± 2 ± = 😥	¢° ⊊₀ ¢≣ %≣	A I [≡] x [≡] 0 ⁰ ∞ Π ³	
	n C	Name	Data type	Default value	Comment	
		▼ Input	Data type	Delautevalue	comment	
		 Manual mode active 	Bool		Manual mode activated	
	01	 Pushbutton manual mode 	Bool		Pushbutton manual mode conveyor on	1
	01	Enable_OK	Bool		All enable conditions OK	
	01	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop opera	ted
4	01	 Output 				
-	01	Conveyor_motor_manual	Bool		Control of the conveyor motor in manual mode	
Ľ	<			10		>
	nv nd nv	ck title: eyor motor in manual mode: If the itions are granted and the safety sh eyor_motor_manual_mode is activ letwork 1: Control of the convey	pushbutton_manual_r nutoff is not activated t ated or motor in manual m	node is operated, the output ode	the enable	
	C	omment	45			
F	+	Manual_mode	#Conveyor_ motor_manual_ mode =	-		
		#Pushbutton_manual_mode B	ool	Pushbu	tton ma	~

Nota: No caso desta variante de atribuição de variáveis há o risco de uma confusão com as variáveis globais da tabela de variáveis. Por isto, deve ser dada a preferência para a variante apresentada antes com o Drag and Drop a partir da descrição das interfaces.

→ Para que a saída somente possa ser ativada, quando tiver sido concedida a liberação e a desativação de proteção não estiver ativa, as variáveis de entrada #Liberação_OK e
 #Desativação de proteção_ativa devem receber adicionalmente a interligação E. Clique duas vezes sobre a estrela amarela ³ de seu elo E para inserir duas outras entradas.

→ Acrescente a suas entradas recém criadas do elo E, as variáveis de entrada #Liberação_OK e #Desativação de proteção_ativa.

→ Negue a entrada conectada com o parâmetro #Desativação de proteção_ativa,

marcando-o e clicando em seguida sobre

→ Não se esqueça de clicar sobre 🔚 Save project. A função concluída de

"MOTOR_MANUAL [FC1] em FBD é apresentada a seguir.

LX	1 1 1 kg		2±	2 ± 🖃 😥	00 60 cm 6		1_ %_	0	00	113			1	
MOT	OR MANUA		-			- •		88.370						
N	lame		Data t	vpe	Default value	Con	ment							
•	Input													
•	Manual_	mode_active	Bool			Man	ual mod	e acti	vated	1				
-	Pushbutt	on_manual_mode	Bool			Pus	hbutton r	manu	almo	de cor	iveyor	on		
-	Enable_0	ок	Bool			Alle	nable co	nditio	ons O	К				
•	Safety_sl	nutoff_active	Bool			Saf	ty shuto	ffacti	e e.g	. eme	rgency	y stop o	perate	ed
•	Output													
-							trol of the							
- 10	Conveyo	_motor_manual	Bool			Cor	aororan	econ	/eyor	motor	in ma	nual m	ode	
< Block Shoc	k title: yor motor in mi ons are grante yor_motor_mai	motor_manual anual mode: if the d and the safety sh nual_mode is activ	Bool pushbu hutoff is ated	utton_manual_ s not activated	mode is operate the output	Cor	nable		veyor	motor	in ma	inual m	node	>
Block onvey onditionvey Ne Cor Cor	k title: vor motor in mu vor motor in mu vor_motor_mai vor_motor_mai etwork 1: Comment	_motor_manual inual mode: If the d and the safety si nual_mode is activ ntrol of the convey	Bool pushbu hutoff is ated yor mot	utton_manual_ s not activated or in manual m	mode is operate the output	Cor	nable		/eyor	motor	in ma	nual m	node	>

→ Em propriedades do bloco, você pode mudar no item "General" (Gerais) o "Language"
 (Idioma) em LAD (plano de contato). (→ Properties (Propriedades) → General (Gerais)
 → Language (Idioma): LAD)

MOTOR_MANU/	AL [FC1]	S Pro	perties	🛄 Info	追 🗓 Dia	agnostics	
General							
General Information	General						
Compilation		Name:	MOTOR_N	VANUAL			
Attributes		Type: Language:	FC FBD				-
		Number:	LAD FBD				
			 autom 	al			

 \rightarrow Em LAD, o programa apresenta a seguinte aparência.

MC	TO			C 40 Cm 42	
IVIC	Na	me	Data type	Default value	Comment
-00	•	Input			
-00		Manual_mode_active	Bool		Manual mode activated
-00		Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
-00		Enable_OK	Bool		All enable conditions OK
-		Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated
-00	•	Output			
		Conveyor_motor_manual	Bool		Control of the conveyor motor in manual mode
6	-			1.0.0	
1000					1
Blo	ck eyd	title: or motor in manual mode: If the	pushbutton_manual_r	mode is operated	d, the enable
Blo Conv cond Conv	ck eyc itio eyc Net	title: rr motor in manual mode: If the ns are granted and the safety sl rr_motor_manual_mode is activ swork 1: Control of the convey ment	pushbutton_manual_ nutoff is not activated ated or motor in manual m	mode is operated the output	d, the enable

7.8 Programação do módulo organizacional OB1 – Comando do funcionamento da correia para frente em operação manual

→ Antes da programação do módulo organizacional "Main[OB1]", converteremos ali a linguagem de programação para FBD (plano de função). Clique primeiro aqui no lado esquerdo do mouse, na pasta "Program blocks" (Blocos de programa) em "Main[OB1)".

 $(\rightarrow CPU_1516F[CPU 1516F-3 PN/DP \rightarrow Program blocks" (Blocos de programa) \rightarrow Main [OB1] \rightarrow Switch programming language (Linguagem de programação) <math>\rightarrow$ FBD)

 \rightarrow Abra agora o módulo organizacional "Main [OB1]" com um clique duplo.

→ Nomeie a rede 1 como "Control conveyor motor forwards in manual mode" (Acionamento movimento da correia para frente no manual-/comando por impulsos).

(→Network 1:... (Rede 1:...) → Control conveyor motor forwards in manual mode (Acionamento movimento da correia para frente no manual-/comando por impulsos))

?	rogi	ran	nming • CPU_1516F [CPU	1516	F-3	PN/DP] > Pr	ogram blocks 🔸 Main [OB1] 🛛 🗕 🖬 🗮 🗙
ю	ю	{]	0 🕫 🐁 🖿 🚍 💬	8± .	2 ±	= 🐲 🥲	6 @ @ ♥ != % @ ™ □
	Ma	in					
		Na	me	Data ty	pe	Default value	Comment
1	-00	•	Input				
2	-		Initial_Call	Bool			Initial call of this OB
3	-		Remanence	Bool			=True, if remanent data are available
4		•	Temp				
5			<add new=""></add>				
6		•	Constant				
7			<add new=""></add>				
	<					1111	>
_	Die	ak	titles "Main Preason Susan /	Curles)*		Instead Instant	
	DIO	CK	uue: wain riogram sweep (cycle)			
	20100	ne	11.				
•	١	let	work 1: Control conveyor mo	tor forw	ards	in manual mo	de
	C	om	ment				

→ Arraste agora sua função "MOTOR_MANUAL [FC1]" através do Drag and Drop para a rede 1 sobre a linha verde.

Project Edit View Insert Online Options To	ols Window Hel	p 🔟 🔯 Go online 🖉					
Project tree	III (?rogra	mming > CPU_1516F	[CPU 1516F-3	PN/DP] → Pr	ogram blocks	Main [OB1]	# # # ×
Devices							
B 0 0	Кя 🕅 🖆	9 9 6 2 8 3	🗩 🗐 📲 ± 🖓 ±	= 🕸 🥐	6 🖉 🖼 🍄	I = X = @ 000	
7.	Mair	1					
▼ 3032_100_FC-Programming	A N	lame	Data type	Default value	Comment		
Add new device	1 🚳 🗖	nput					
Devices & networks	2 🕣 🗉	Initial Call	Bool		Initial call of this	OB	
CPU_1516F [CPU 1516F-3 PN/DP]	3 🔊 🗉	Remanence	Bool		=True, if remaner	nt data are available	e
Device configuration	4 - 51 -	Temp					
V. Online & diagnostics	5	<add new=""></add>					
- Regram blocks	≡ 6 - • • •	Constant					
Add new block	7	<add new=""></add>					
Amin [OB1]							107
MOTOR MANUAL [FC1]							
Technology objects	▼ Block	k title: "Main Program Si	veep (Cycle)"				
External source files	Comm	ent					
Add new external file					120		_
T PLC tags	Ne	etwork 1: Control conve	yor motor forwards	in manual mo	de		
Show all tags	Cor	mment					
Add new tag table							
Standard-Variablentabelle [54]							
Tag table sorting station [28]	100	-MOTOR_1	ANUAL [FC1]				
PIC data types			No.				
Watch and force tables							
Online backups	100						
Traces							
Program info							
Device providata					1		
	~ <	III.			3 100%	1	
✓ Details view	Main [0	DB1]	9	Properties	i Info i	2 Diagnostics	
	Gene	ral					
	-						

→ Um bloco com a interface determinada por você e as conexões EN e ENO será inseridos na rede 1.

→ Para adicionar um E antes do parâmetro de entrada "Enable_OK" (Liberação_OK), marque esta entrada e adicione com um clique no símbolo [▲] in na sua barra de ferramentas lógica. (→ [▲])

8	>=1	??	н	-01	↦	-[=]
5	AND	logic o	perati	on [Sh	ift+F2]	p (Cycle)*
	Netv	vork 1	: Co	ntrol c	onvey	or motor forwards in manual mode
	Comn	nent				
					۲	%FC1 MOTOR_MANUAL*
			 27.7>	— EN Ma	nual_ ode_ac	tive
			?.?	Pu: ma mo	shbutt nual_ ode	on_ Conveyor_ motor_
			<11.7> <11.7>	En Sa sh	able_0 fety_ utoff_a	OK manual_ mode — ?.?

→ Clique sobre a estrela amarela 🌞 de seu elo E para inserir uma outra entrada. (→ 🧚)

- → Para ligar o bloco com o variáveis globais da "Tag table_sorting station" (Tabela de variantes_sistema de classificação), temos 2 possibilidades:
- → Ou selecionado a "Tag table_sorting station" (Tabela de variantes_sistema de classificação) na árvore do projeto e arrastando a variável global da visualização detalhada na interface do FC1 (→ Tag table_sorting station (Tabela de variantes_sistema de classificação) → Details view (Visualização detalhada) → -S0→ Manual_mode_active (Operação manual_ativo))

▼ 🔁 PLC tags	Network 1: Control conveyor motor forwards in manu	al mode
a Show all tags	Comment	
🗳 Add new tag table	Comment	
💐 Default tag table [54]		NECA
ag table_sorting station [28]		Yor-C.1
PLC data types		"MOTOR_MANUAL"
Watch and force tables		— EN
🕨 🙀 Online backups 🛛 🗸		Manual mode
✓ Details view	&	< <u></u> active
	?.? —	Pushbutton_
Name Data ty Comment	?.? —	?.? — manual_mode
-Q3 Bool conveyor motor -M1 variable s	?.? — *	Enable OK Convoyor motor
-SO Bool 🔳 mode selector manual(0) / auto.		Conveyor_mode
-S1 Bool pushbutton automatic start (no)		Safety_shutoff
-S2 Bool pushbutton automatic stop (nc)		// /</td
-S3 Bool pushbutton manual mode con		

→ Ou inserindo em <??.?> as letras iniciais (ex.: "-S") da variável global desejada e escolhendo da lista exibida, a variável de entrada global "-S0" (% E0.2).

 $(\rightarrow Manual_mode_active \ (Operação \ manual_ativo) \rightarrow \text{-}S \rightarrow \text{-}S0)$

		_									
8	>=1	177	-	-01	↦	-[=]					
► E	llock ti	itle: "	Main	Progra	m Swe	eep (Cyc	ile)*				
- 6	Netv	work 1	: Co	ntrol c	onvey	or motor	r forwards in manual me	ode			
	Comm	nent									
								%FC1			
								MOTOR_MANU	AL"		
								EN			
								Manual			
							-so*	Bool	%10.2	mode selector	^
					&		*•• *•• *•	Bool	%10.3	pushbutton aut	
			<11.7>	-			🖅 "-S2"	Bool	%10.4	pushbutton aut	-
		5	<11.7>	-*			-S3 "	Bool	%11.4	pushbutton ma	
							- 54"	Bool	%11.5	pushbutton ma	
							"-S5"	Bool	%11.6	pushbutton ma	*

→ Adicione as outras variáveis de entrada "-S3", "-K0", "-B1", "-S4" e "-A1" e na saída "Conveyor_motor_manual_mode" (Motor de correia_comando por impulsos) a variável de saída "-Q1" (%A0.0).

→ Negue as consultas das variáveis de entrada "-S0", "-S4" e "-A1", marcando-as e

7.9 Programação do módulo organizacional OB1 – Comando do funcionamento da correia para trás em operação manual

→ Conceda à rede 2 o nome de "Control conveyor motor backwards in manual mode" (Ativação do funcionamento da correia para trás em operação manual/comando por impulsos) e acrescente, como já executado na rede 1, a sua função "MOTOR_MANUAL [FC1]" por meio de Drag and Drop.

→ Conecte sua função, como apresentado aqui. Na linguagem de programação FBD (plano de função) receberá o seguinte resultado.

→ Na linguagem de programação LAD (plano de contato), o resultado apresenta a seguinte aparência.

7.10 Salvar e compilar o programa

→ Para salvar o seu projeto, no menu selecione o botão Save project. Para verter todos os módulos, clique na pasta "Program blocks" (Blocos do programa) e selecione o ícone
 no menu. (→ Save project → Program blocks (Blocos do programa) →)

M Siemens - D:\Automation\032_100_FC-Programming\032_100_FC-Progr Project Edit View Insert Online Options Tools Window Help 📑 📴 🔚 Save project 📕 🐰 🏥 🚡 🗙 🏷 🛨 (주士 🐻 🛄 🌆 🚆 Compile 0_FC Devices B 0 0 ы ы 🖗 🗄 032_100_FC-Programming ~ >=1 ? 8 Add new device Devices & networks Block title CPU_1516F [CPU 1516F-3 PN/DP] Network Device configuration & Online & diagnostics Commen 10 Program blocks Add new block Main [OB1] MOTOR_MANUAL [FC1] Technology objects

→ Na área "Info" "Compile" (Verter) será mostrado em seguida, qual bloco pôde ser vertido com êxito.

		Q, F	roperties	1.1	nfo	& Diag	nostics	
General Cross-references	Compile	Syntax						
🕄 🚹 🚺 Show all messages								
Compiling completed (errors: 0; warr	nings: 0)							
! Path 👻	Description			Go to	?	Errors	Warnings	Time
✓ ▼ CPU_1516F				7		0	0	12:16:44 PM
📀 🔻 Program blocks				7		0	0	12:16:44 PM
MOTOR_MANUAL (FC1)	Block was succes	ssfully compile	ed.	~				12:16:44 PM
Main (OB1)	Block was succes	ssfully compile	ed.	7				12:16:44 PM
Ø	Compiling compl	leted (errors: (); warnings: 0))				12:16:44 PM

7.11 Carregar o programa

→ Após uma compilação bem sucedida, o completo comando pode ser carregado com o programa criado, como foi já descrito nos módulos referentes à configuração de hardware. (→ □□)

M Siemens - D:\Automation\032_100_FC-Progra	amming\032_100_FC-Programming	
Project Edit View Insert Online Options	Tools Window Help	Totally Integrated Automation
Project A M A	호 (여 호 🕤 🔃 🕼 📕 🔂 🍠 Go online 🧬 Go offline 🔥 🖪 📑 🛪 🚽 🛛	PORTAL
Project tree	A D2 100 FC Programming & CPU 1516F (FUL) 1516F 3 DM(DD) & Program Marke & Main (DB1)	2 = X
Tiblecturee	Download to device	
Devices		
1900	표 22 22 22 22 22 22 22 22 22 22 22 22 22	
2	Block interface	
▼ 3032_100_FC-Programming	I A I P 40	
Add new device	a >=1 [??? → -ol ↦ -[=]	
🛔 Devices & networks	Block title: "Main Program Super (Orle)"	
CPU_1516F [CPU 1516F-3 PN/DP]	Diock due. main rogian sheep (cycle)	
Device configuration	 Network 1: Control conveyor motor forwards in manual mode 	
😨 Online & diagnostics	Comment	
Program blocks		
Fechnology objects	%FC1	
External source files	"MOTOR_MANUAL"	
🕨 🎑 PLC tags	&— EN	
Complexity PLC data types	190.1 190.2 Manual	=
Watch and force tables	"-K0" — "-S0" — mode_active	
Online backups	590.5 Pushbutton	
🕨 🔛 Traces	"-81"	
Program info	11.5 "-S3" — mode Conveyor_ motor	
Device proxy data	*54* - + + + Enable_OK manual_ %Q0.0	
PLC alarms	\$10.0 Safetymode	
Text lists	*-A1* -•• shutoff_active ENO	
Local modules		
Common data		
Documentation settings	 Network 2: Control conveyor motor backwards in manual mode 	
Languages & resources	Comment	
Online access		
Card Reader/USB memory	%FC1	
✓ Details view	& EN	~
	100	<i>?</i> ⁶
Name	S. Properties	Info 🚯 🖞 Diagnostics 👘 🗋 🗕 🔶
Portal view Dverview	S Main	ne project 032, 100, FC-Programming

7.12 Observar os módulos do programa

 \rightarrow Para a observação do programa carregado, o módulo desejado deve ter sido aberto.

Com um clique sobre o símbolo \square a observação pode ser ligada/desligada. (\rightarrow Main

Nota: A observação é realizada aqui com relação ao sinal e em função do comando. Os estados de sinal nos bornes são exibidos com TRUE ou FALSE.

→ A função solicitada no bloco de organização "Main [OB1]", "MOTOR_manual" [FC1] pode ser selecionada com um clique no lado do mouse direto em "Open and monitor" (Abrir e observar). (→ "MOTOR_manual" [FC1] → Open and monitor (Abrir e observar))

Nota: A observação é realizada aqui com relação à função e independente do comando. O acionamento do encoder ou o estado do sistema são apresentados aqui com TRUE ou FALSE.

→ Se um determinado uso da função "MOTOR_MANUAL" [FC1] não for observado, é possível selecionar o ambiente da solicitação através do símbolo . (→ . (→ Call environment (Ambiente da solicitação) → OK)

D	ependency structure	1 Address	Details
1	🖀 Main	OB1	Main NW1 (Co.
2	🖀 Main	OB1	Main NW2 (Co
		Transfer to *adjus	ted manually"

7.13 Arquivamento do projeto

→ Finalmente, ainda pretendemos arquivar o completo projeto. Por favor, selecione no menu → "Project" (Projeto), o item → "Archive ..." (Arquivar ...). Selecione uma pasta, na qual se quer arquivar o projeto e salve como tipo de arquivo "TIA Portal project archives" (TIA Portal-Arquivos de projeto). (→ Project (Projeto) → Archive (Arquivar) →TIA Portal project archives" (TIA Portal-Arquivos de projeto) → 032-100_Programação-FC....→ Save (Salvar))

8 Lista de verificação

Nº	Descrição	Verificado
1	Compilação bem sucedida e sem mensagem de erro	
2	Carregamento bem sucedido e sem mensagem de erro	
3	Ligar o sistema (-K0 = 1) Cilindro recolhido / Mensagem de retorno ativada (-B1 = 1) DESLIGA EMERGÊNCIA (-A1 = 1) não ativado Modo de operação MANUAL (-S0 = 0) Ativar o comando por impulsos para avançar a correia (-S3 = 1) então o motor da correia avança em rotação fixa (-Q1 = 1)	
4	Como em 3, mas ativar DESLIGA EMERGÊNCIA (-A1 = 0) \rightarrow -Q1 = 0	
5	Como em 3, mas modo de operação AUTOMÁTICO (-S0 = 1) \rightarrow -Q1 = 0	
6	Como em 3, mas desligar o sistema (-K0 = 0) \rightarrow -Q1 = 0	
7	Como em 3, mas cilindro não recolhido (-B1 = 0) \rightarrow -Q1 = 0	
8	Ligar o sistema (-K0 = 1) Cilindro recolhido / Mensagem de retorno ativada (-B1 = 1) DESLIGA EMERGÊNCIA (-A1 = 1) não ativado Modo de operação MANUAL (-S0 = 0) Ativar o comando por impulsos para retroceder a correia (-S4 = 1) então o motor da correia retrocede em rotação fixa (-Q2 = 1)	
9	Como em 8, mas ativar DESLIGA EMERGÊNCIA (-A1 = 0) \rightarrow -Q2 = 0	
10	Como em 8, mas modo de operação AUTOMÁTICO (-S0 = 1) \rightarrow -Q2 = 0	
11	Como em 8, mas desligar o sistema (-K0 = 0) \rightarrow -Q2 = 0	
12	Como em 8, mas cilindro não recolhido (-B1 = 0) \rightarrow -Q2 = 0	
13	Como em 8, mas também ativar o comando por impulsos para avançar a correia (-S3 = 1) \rightarrow -Q1 = 0 e também -Q2 = 0	
14	Projeto arquivado com sucesso	

9 Exercício

9.1 Definição da tarefa – Exercício

Neste exercício, as seguintes funções da descrição do processo do sistema de classificação serão planejadas, programadas e testadas:

- Operação manual estender o cilindro
- Operação manual recolher o cilindro

Nota: Preste a atenção nesta ocasião, sobre a reutilização ou a blindagem das funções.

9.2 Planejamento

Agora, planeja de modo autônomo a implementação da definição da tarefa.

Nº	Descrição	Verificado
1	Função FC: CILINDRO_MANUAL criada	
2	Interfaces definidas	
3	Função programada	
4	Função FC2 integrada na rede 3 do OB1	
5	Variáveis de entrada para o cilindro recolhido conectadas	
6	Variáveis de saída para o cilindro recolhido conectadas	
7	Compilação bem sucedida e sem mensagem de erro	
8	Função FC2 integrada na rede 4 do OB1	
9	Variáveis de entrada para o cilindro estendido conectadas	
10	Variáveis de saída para o cilindro estendido conectadas	
11	Compilação bem sucedida e sem mensagem de erro	
12	Carregamento bem sucedido e sem mensagem de erro	
13	Ligar o sistema (-K0 = 1) Cilindro recolhido / Mensagem de retorno ativada (-B1 = 1) DESLIGA EMERGÊNCIA (-A1 = 1) não ativado Modo de operação MANUAL (-S0 = 0) Não ativar a recolha do cilindro (-S5 = 0) Ativar a extensão do cilindro (-S6 = 1) então estender o cilindro (-M3 = 1) com sucesso	
14	Ligar o sistema (-K0 = 1) Cilindro estendido / mensagem de retorno ativada (-B2 = 0) DESLIGA EMERGÊNCIA (-A1 = 1) não ativado Modo de operação MANUAL (-S0 = 0) Não ativar a extensão do cilindro (-S6 = 0) Ativar a recolha do cilindro (-S5 = 1) então recolher o cilindro (-M2 = 1) com sucesso	
15	A recolha e a extensão do cilindro não podem ser ativadas simultaneamente	
16	Projeto arquivado com sucesso	

9.3 Lista de verificação – Exercício

10Informação adicional

Para um início de treinamento ou um aprofundamento, é possível encontrar informações adicionais para auxílio de orientação, como, por ex.: Getting Started, vídeos, tutoriais, aplicativos, manuais, guias de programação e software/firmware para teste, no seguinte link:

www.siemens.com/sce/s7-1500