



SIEMENS



# Documentação de treinamento SCE

Siemens Automation Cooperates with Education | 05/2017

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

Cooperates  
with Education

Automation

SIEMENS

## 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:

[siemens.com/sce/tp](https://www.siemens.com/sce/tp)

## Treinamentos avançados

Para treinamentos regionais avançados SCE Siemens, entre em contato com o parceiro SCE da sua região [siemens.com/sce/contact](https://www.siemens.com/sce/contact)

## Outras informações sobre SCE

[siemens.com/sce](https://www.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 [roland.scheuerer@siemens.com](mailto:roland.scheuerer@siemens.com).

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	Objetivo.....	5
2	Requisito.....	5
3	Hardwares e softwares necessários.....	6
4	Teoria.....	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	Definição da tarefa.....	16
6	Planejamento.....	16
6.1	Parada de emergência.....	16
6.2	Operação manual – Motor da correia em comando por impulsos.....	16
7	Instrução 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	Lista de verificação.....	49
9	Exercício.....	50
9.1	Definição da tarefa – Exercício.....	50
9.2	Planejamento.....	50
9.3	Lista de verificação – Exercício.....	51
10	Informação 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.

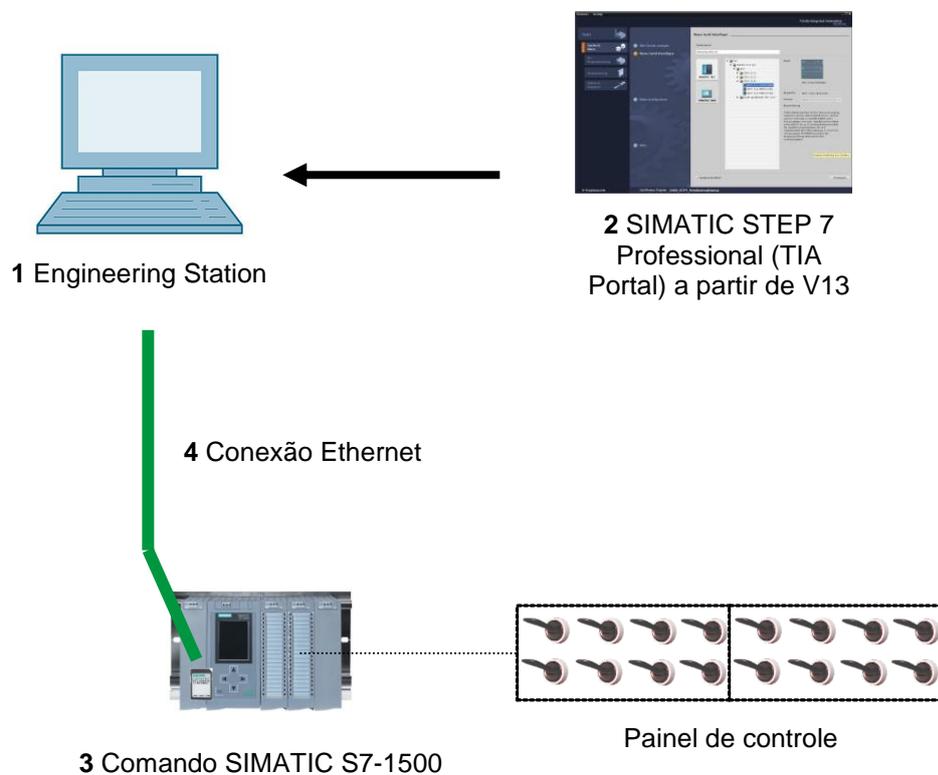
## 2 Requisito

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
- 3 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.

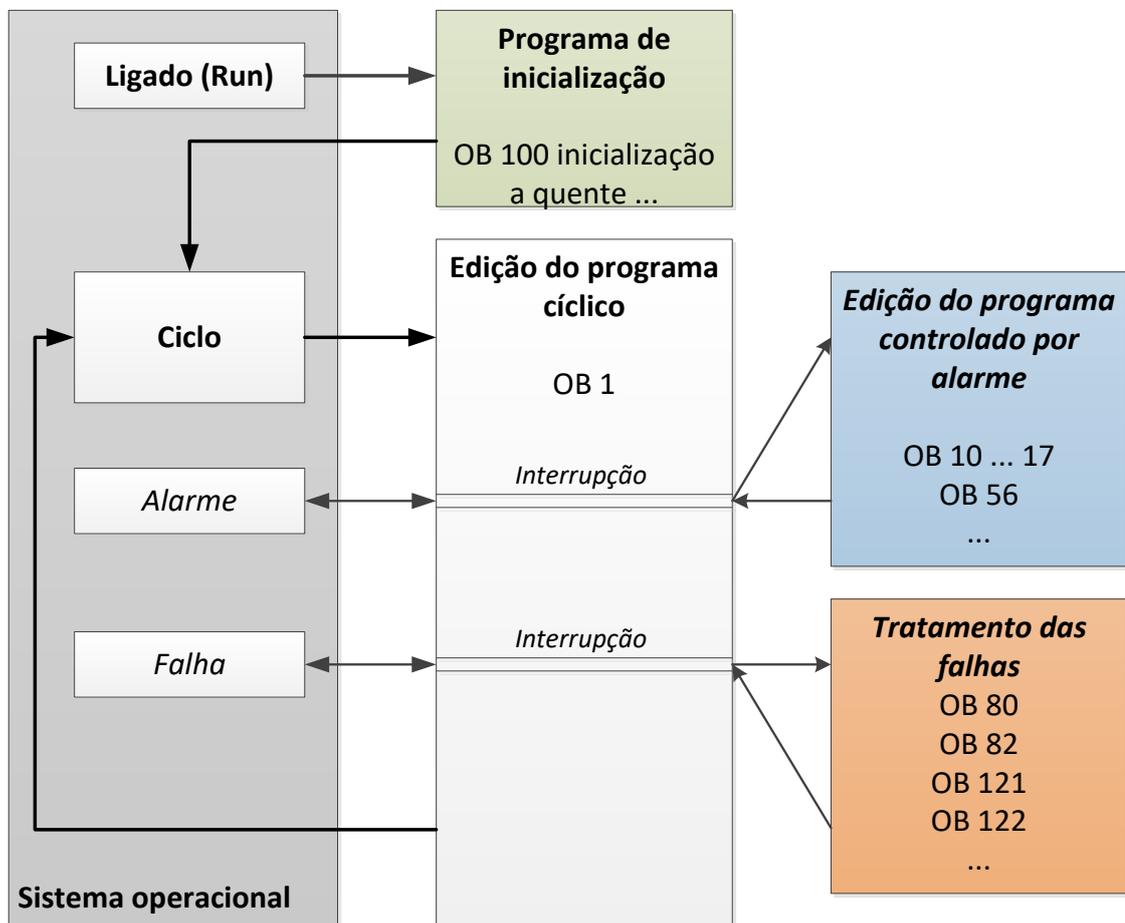


Figura 1: Eventos de partida no sistema operacional e chamada OB

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, $\geq$ 123	Ignorar
<b>Programa cíclico</b>	1, $\geq$ 123	Ignorar
Alarme de horário	10 até 17, $\geq$ 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

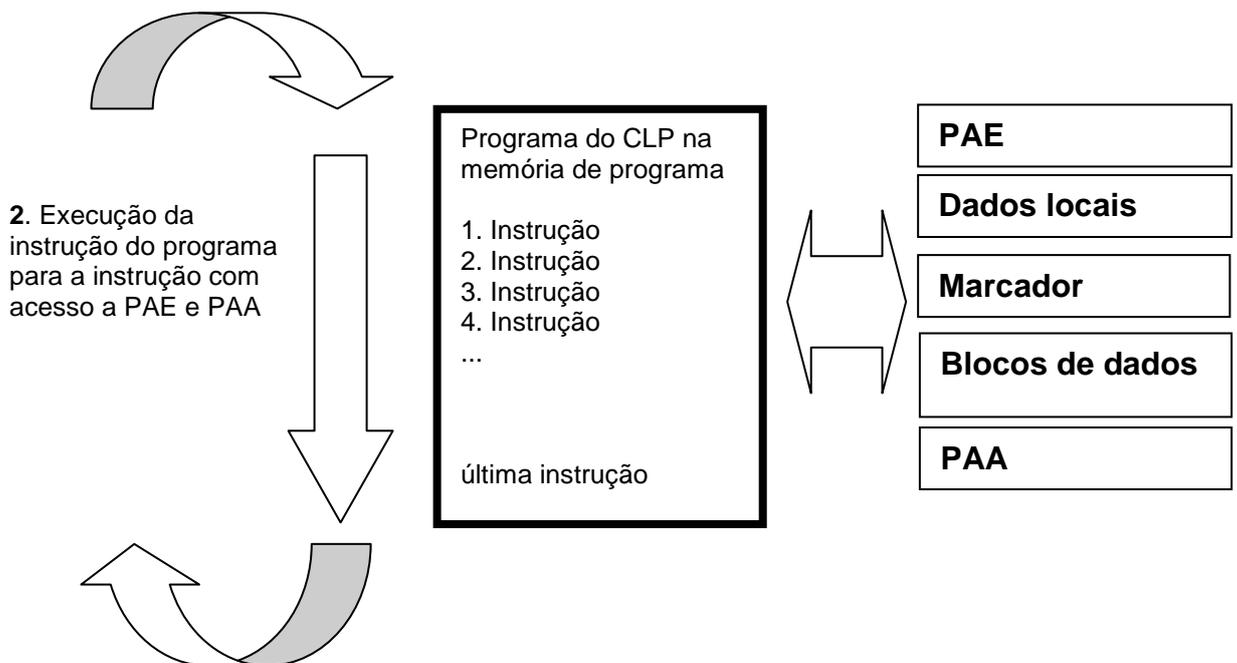
### 4.3 Imagem do processo e processamento cíclico do programa

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:

1. 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 sidos 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.

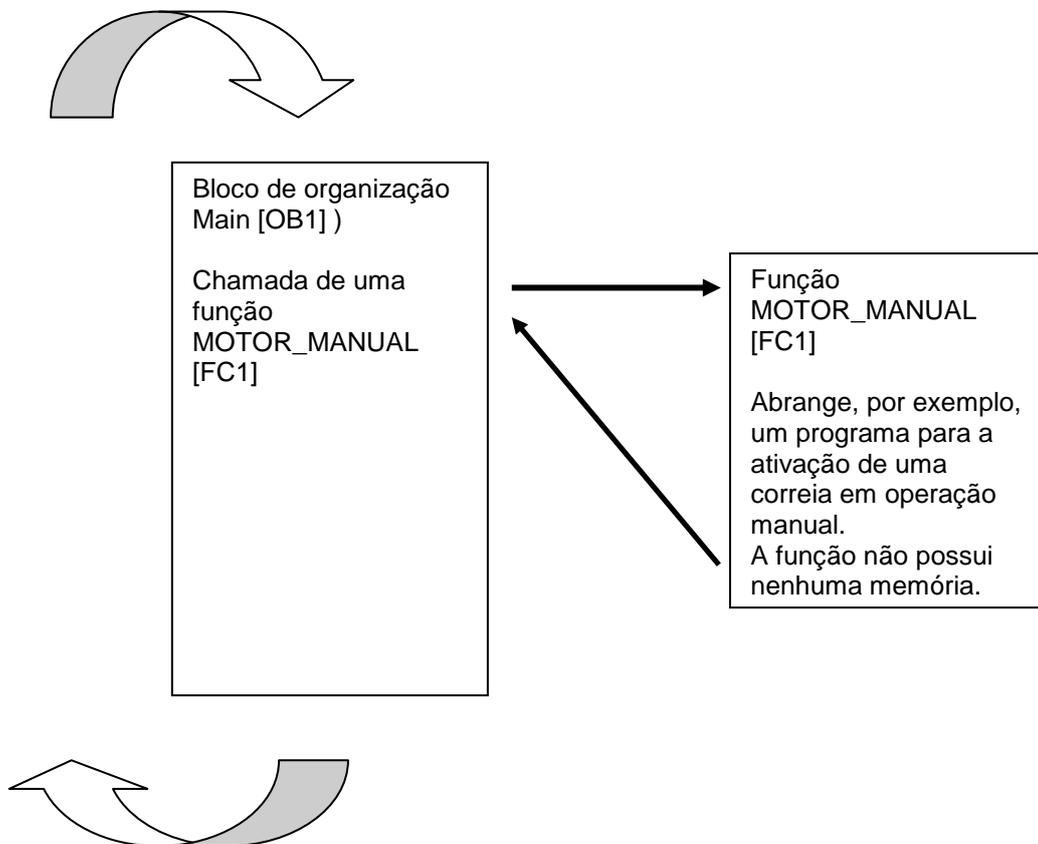


Figura 3: Função com chamada a partir do módulo organizacional Main[OB1]

## 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 trabalhar 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.

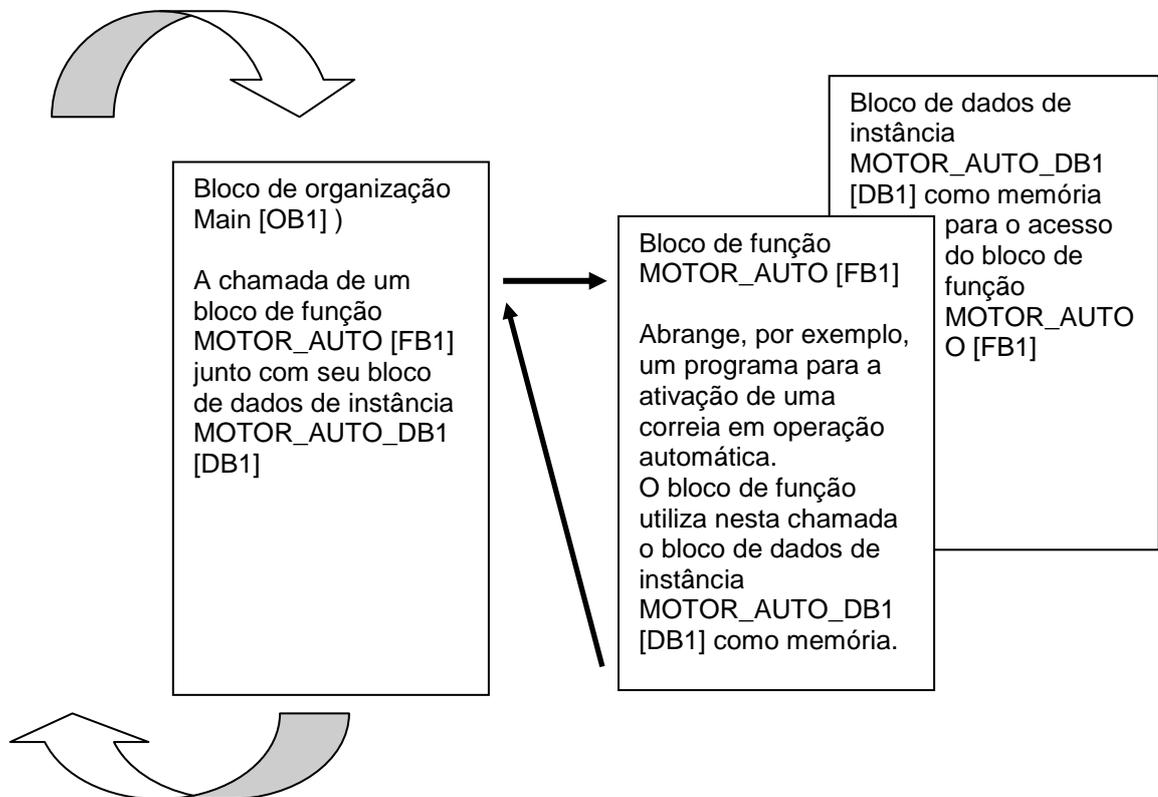


Figura 4: Bloco de função e instância com chamada a partir do módulo organizacional Main[OB1]

## 4.6 Blocos de dados globais

Ao contrário dos blocos de código, os blocos de dados não contêm 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.

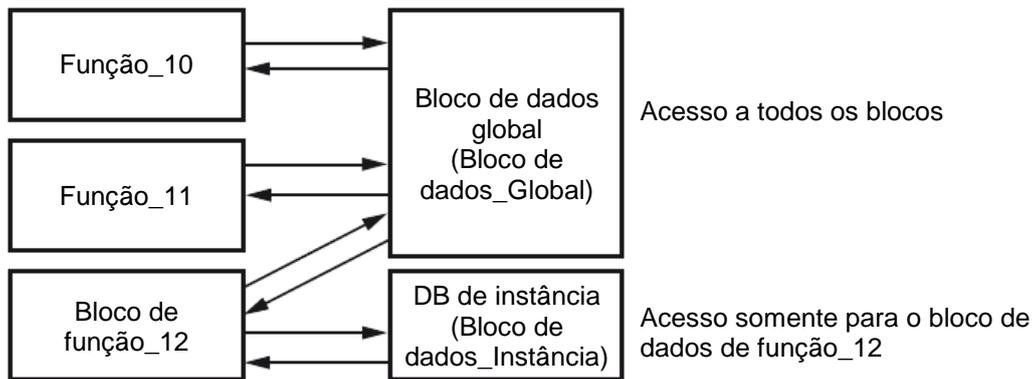


Figura 5: Diferença entre o bloco de dados global e o bloco de dados de instância.

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).

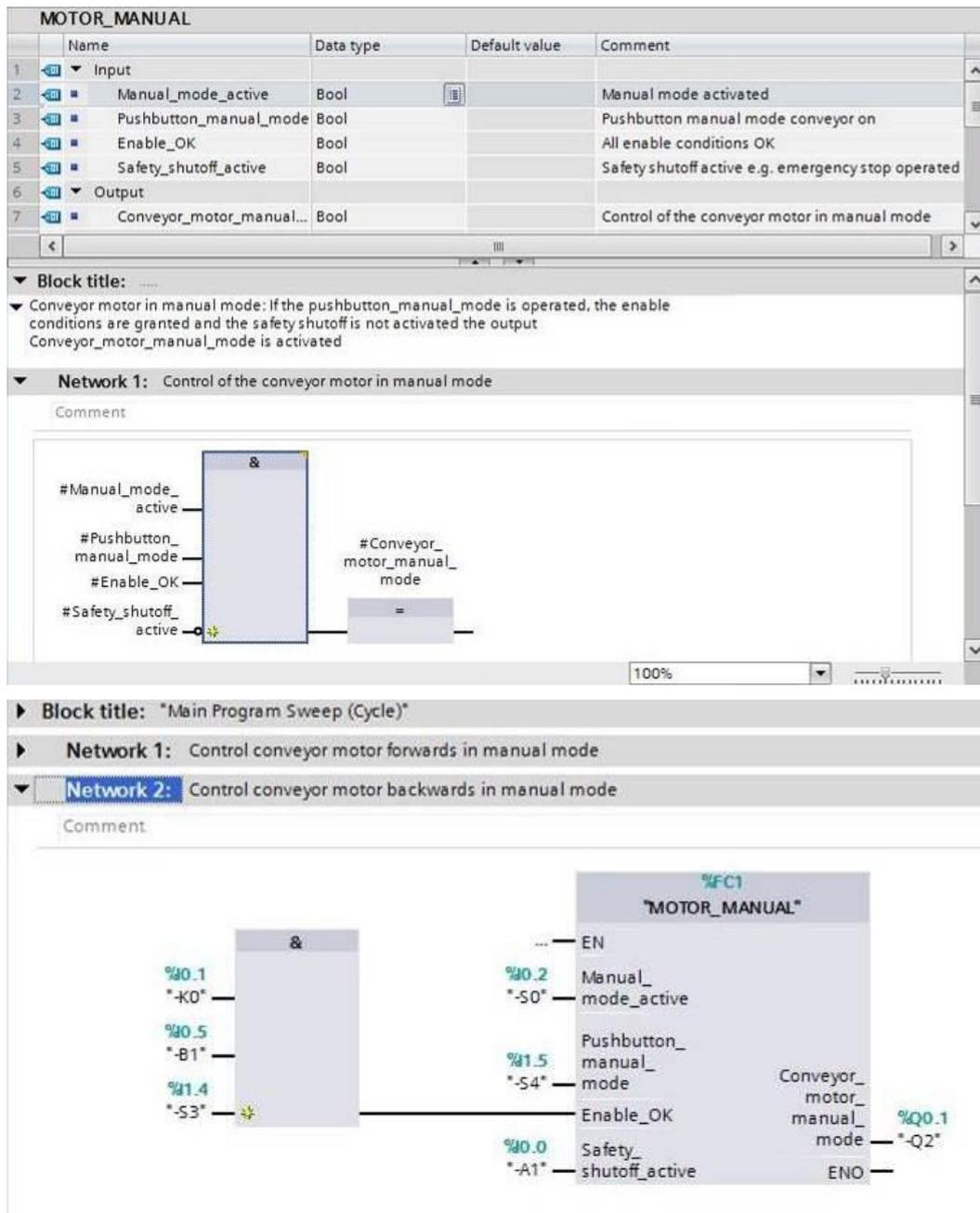


Figura 6: Função habilitada para biblioteca com chamada em OB1

## 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 linguagem 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.

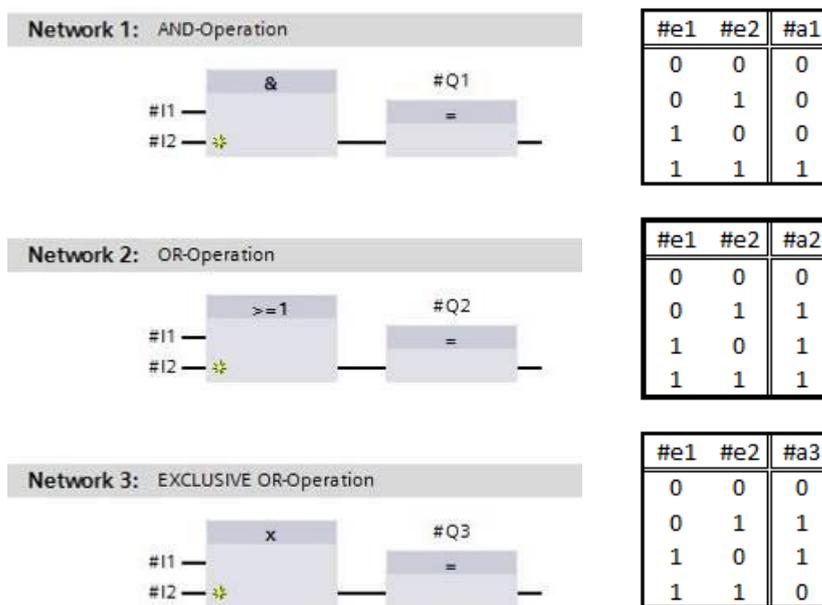


Figura 7: Funções binárias no FBD e tabela lógica pertinente

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 recomendada 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 uma 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
<b>Output</b>		
Motor da correia_comando por impulsos	BOOL	Ativar o motor da correia em comando por impulsos

Tabela 2: Parâmetros para FC "MOTOR\_MANUAL"

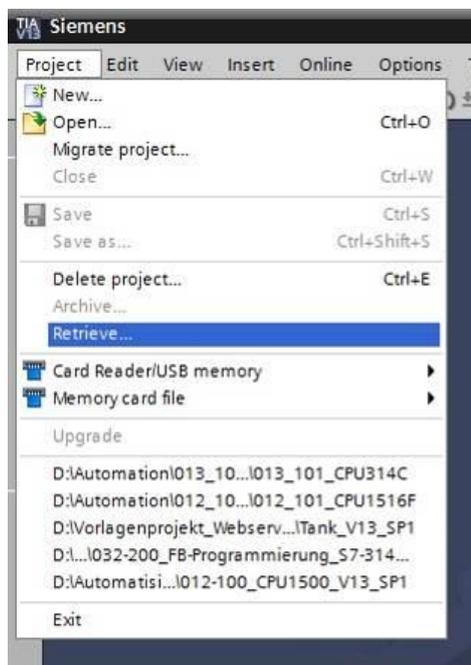
A saída motor da correia\_comando por impulsos está ligada, enquanto o botão\_comando por impulsos for pressionado, o modo de operação manual estiver ativado, a liberação tiver sido concedida, e a desativação de proteção não estiver ativa.

## 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

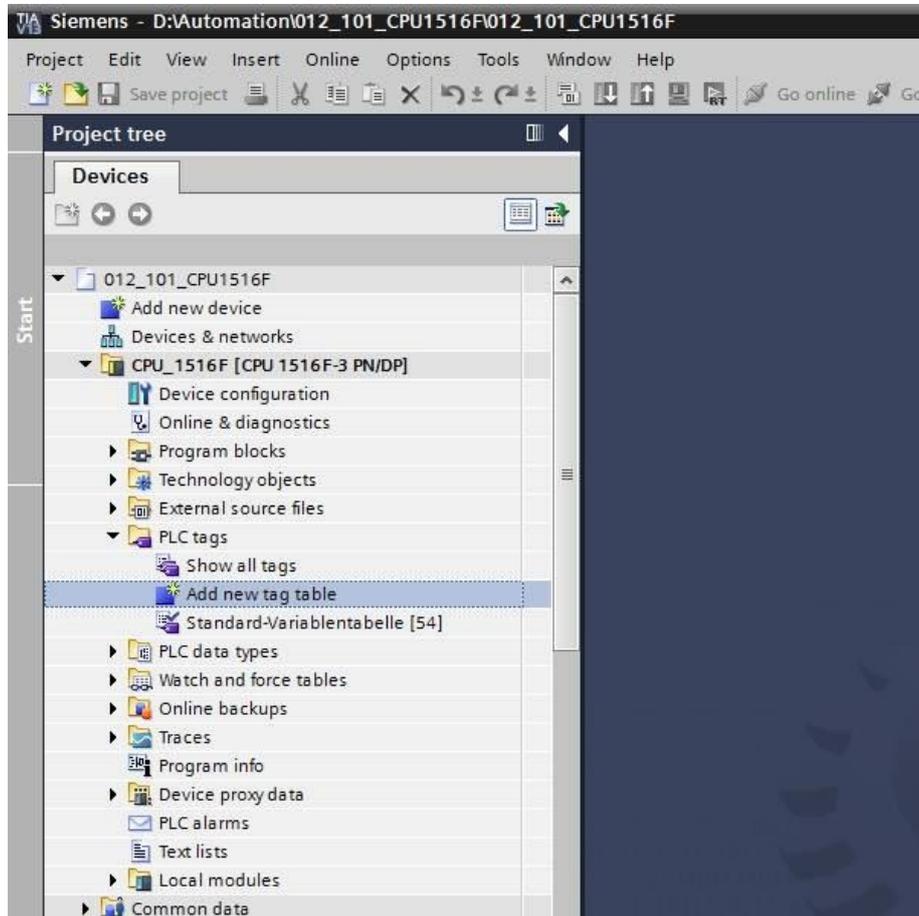
- Ante que possa começar com a programação da função (FC) "MOTOR\_MANUAL", necessitará de um projeto com uma configuração de hardware. (z.B. SCE\_PT\_012-101\_Configuração de hardware\_S7-1516F\_R1502.zap) Para desarmarquivar um projeto existente, deverá pesquisar a partir da visualização do projeto, 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).)



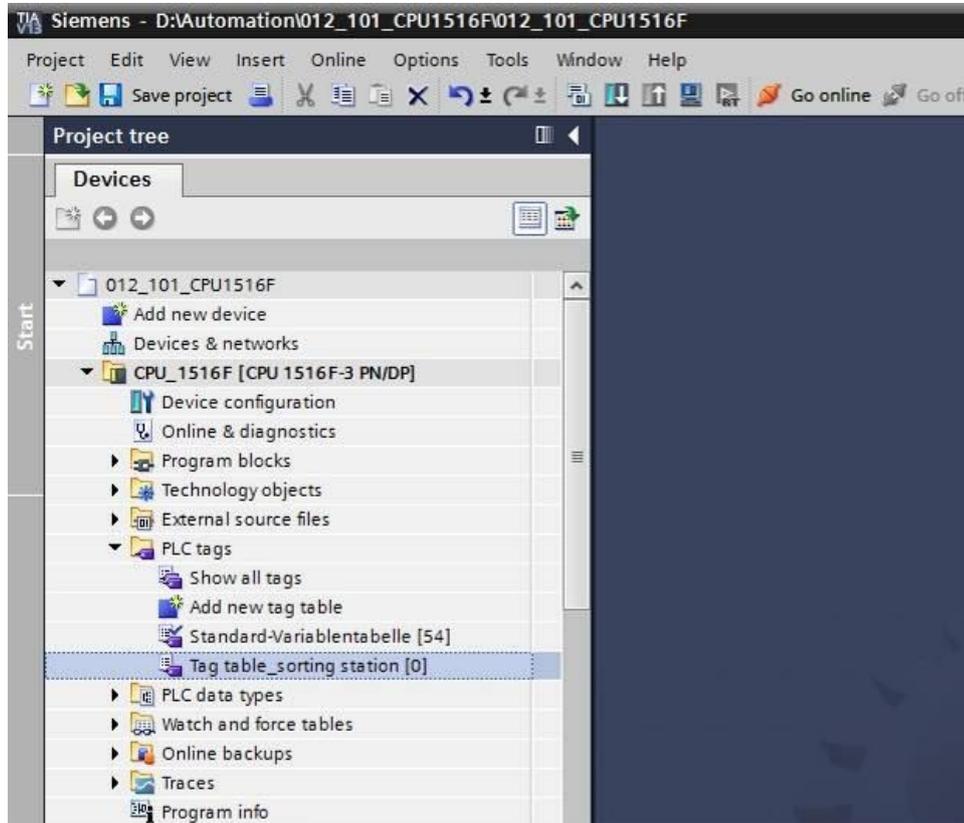
- Como próximo passo, pode ser selecionado o diretório de destino, no qual o projeto desarmarquivado deverá ser salvo. Confirme sua seleção com "OK". (→ Target directory (Diretório de destino) →OK)

## 7.2 Criação de uma nova tabela de variáveis

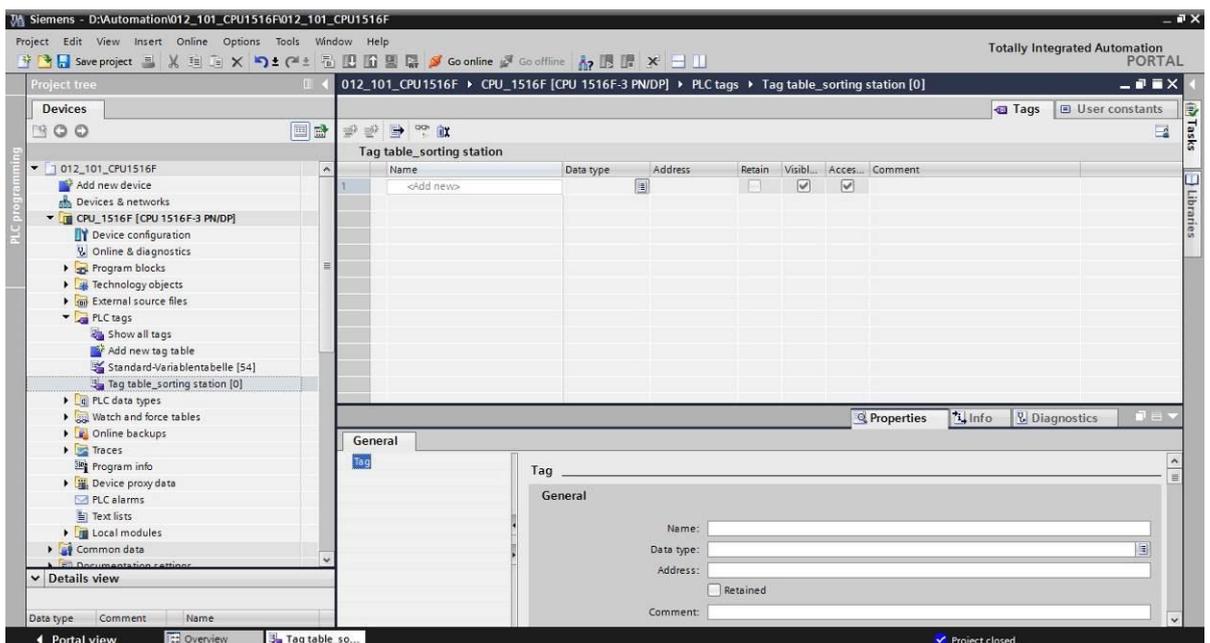
- Navegue na visualização do projeto até variantes → PLC de seu controller e crie uma tabela de variantes, clicando duas vezes em → Add new tag table (Adicionar nova tabela de variantes) "clique duplo".



- 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))

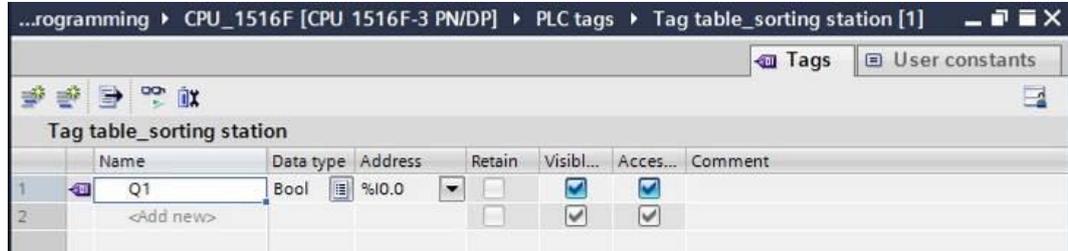


- Abra-a em seguida com um clique duplo. (→ Tag table\_sorting station (Tabela de variantes\_sistema de classificação))

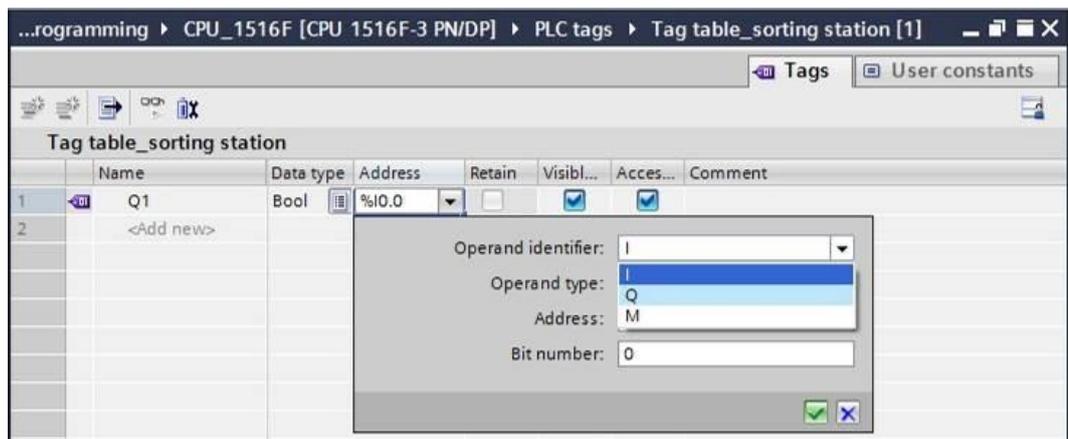


### 7.3 Criação de novas variáveis dentro da tabela de variáveis

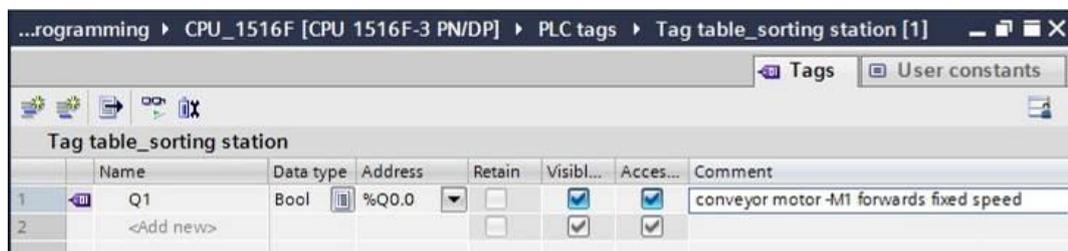
- Acrescente o nome Q1 e confirme a entrada com a tecla Enter. Se você não criar nenhuma outra variante, o TIA Portal irá fornecer automaticamente o tipo de dado "Bool" e o endereço %E0.0 (I 0.0). (→ <Add new> (<Adicionar>) →Q1 → Enter)



- 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 → )

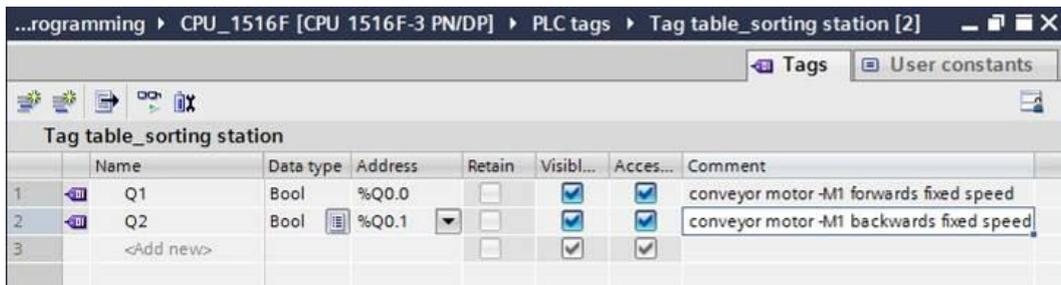


- Faça o comentário para a variante "conveyor motor -M1 forwards fixed speed" (motor de correia -M1 velocidade fixa de movimento para frente).



→ 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 %Q0.1 (Q0.1). Faça o comentário "conveyor motor -M1 backwards fixed speed" (motor de correia -M1 velocidade fixa de movimento para trás).

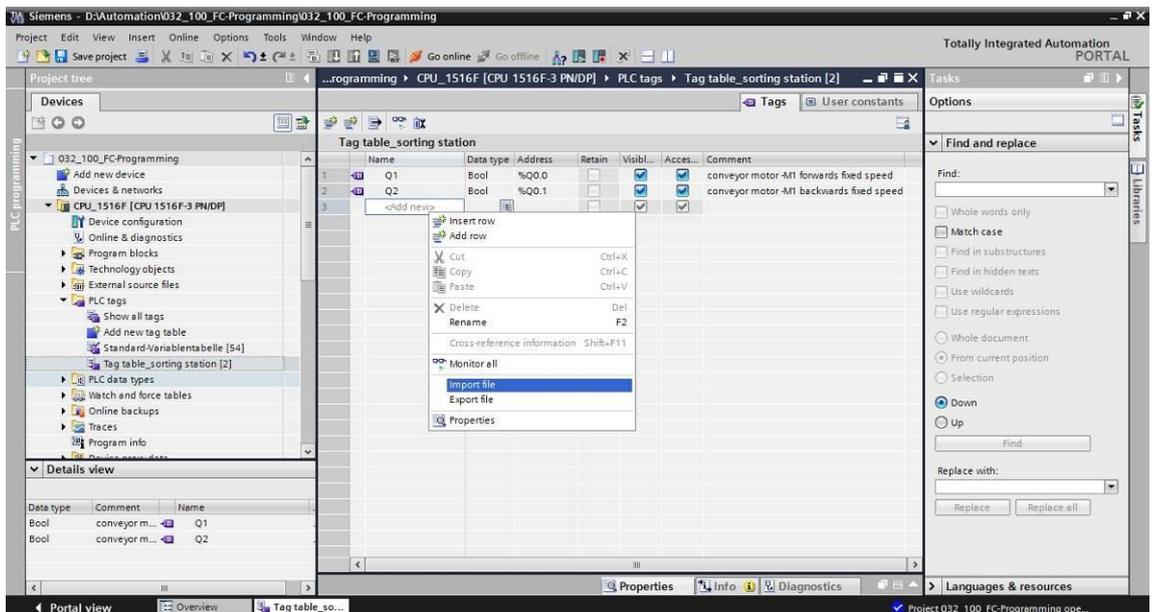
(→ <Add new> (<Adicionar>) → Q2 → Enter → Comentário → conveyor motor -M1 backwards fixed speed (motor de correia -M1 velocidade fixa de movimento para trás)



## 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).

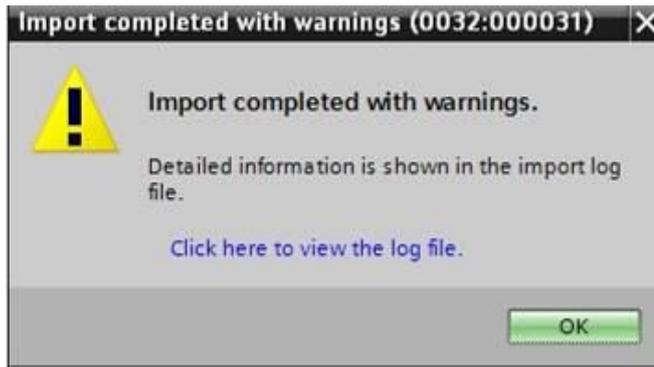
(→ Clique no lado direito em um espaço vazio da tabela de variáveis →Import file (Arquivo de importação))



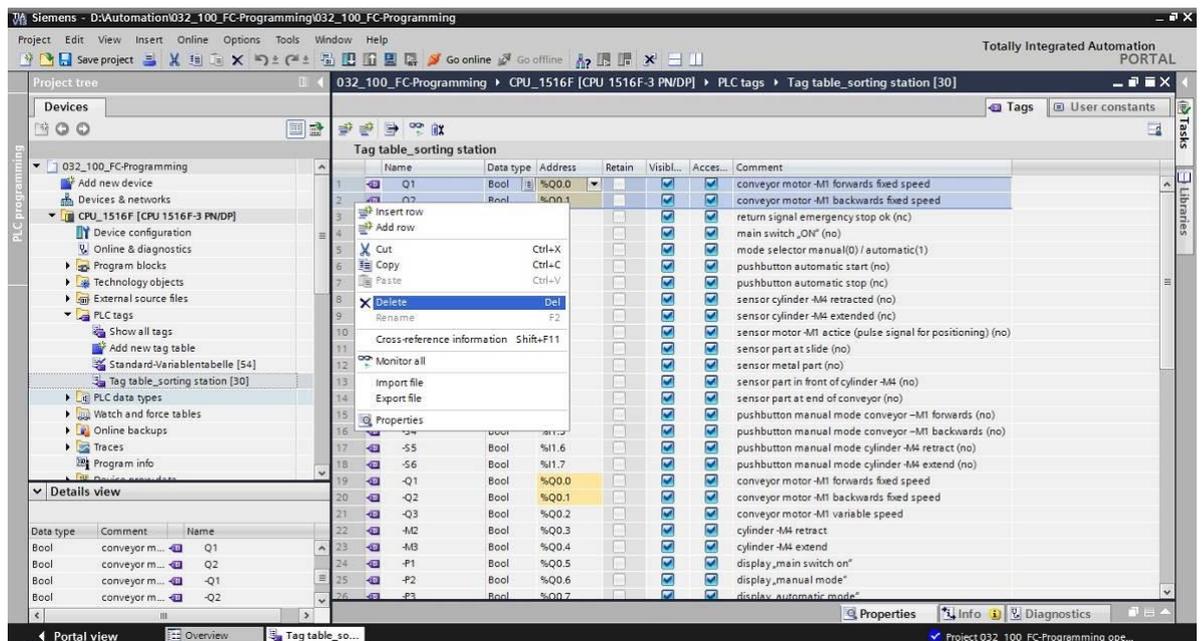
→ Selecione a tabela de símbolo desejada (por ex., no formato .xlsx) e confirme a seleção com "Open" (Abrir).

(→ SCE\_PT\_020-100\_tabela de variáveis\_sistema de classificação... → 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.

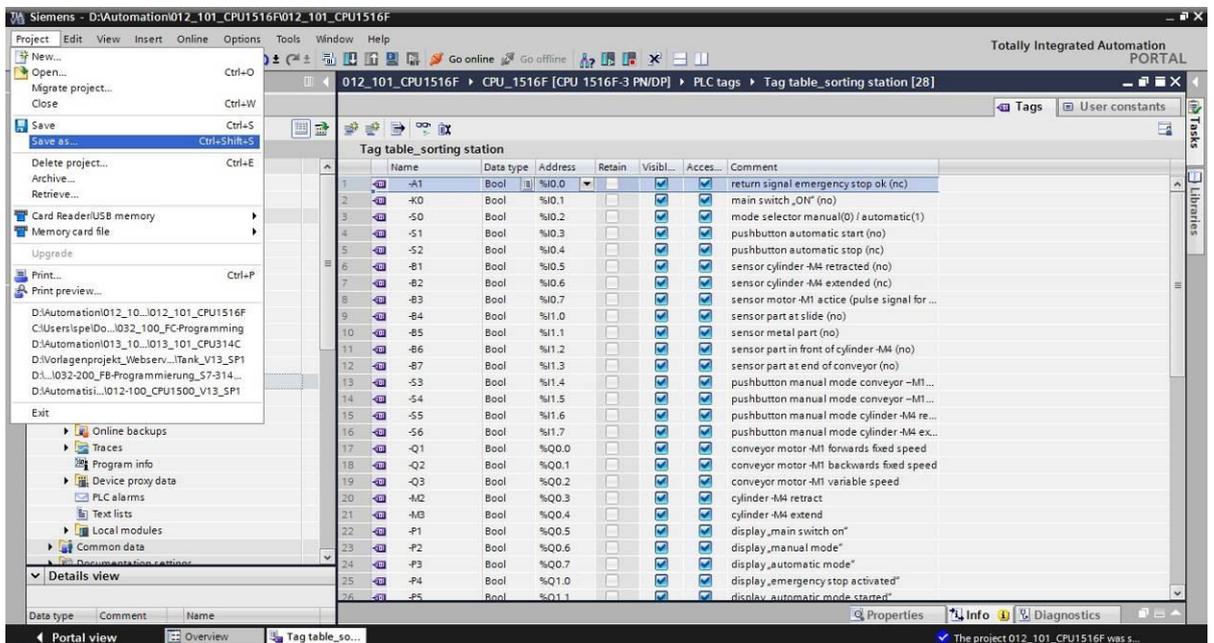


- Poderá ver, que alguns endereços foram destacados em laranja. Este existem em duplicidade e os nomes das variáveis pertinentes foram numerados automaticamente, para evitar ambiguidades.
- Exclua as variáveis existentes em duplicidade, marcando as linhas e pressionar a tecla Delete no seu teclado ou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

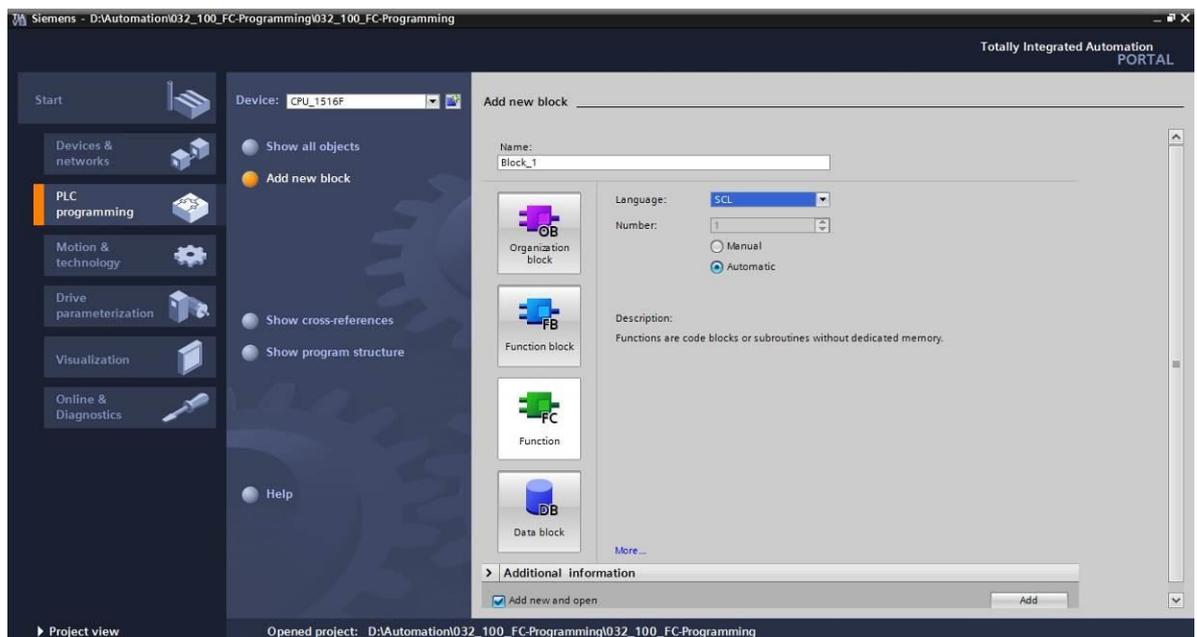
(→ Project (Projeto) → Save as ... (Salvar em ...) → 032-100\_Programação-FC → Save (Salvar))



## 7.5 Criação da função FC1 "MOTOR\_MANUAL" para o motor da correia em comando por impulsos

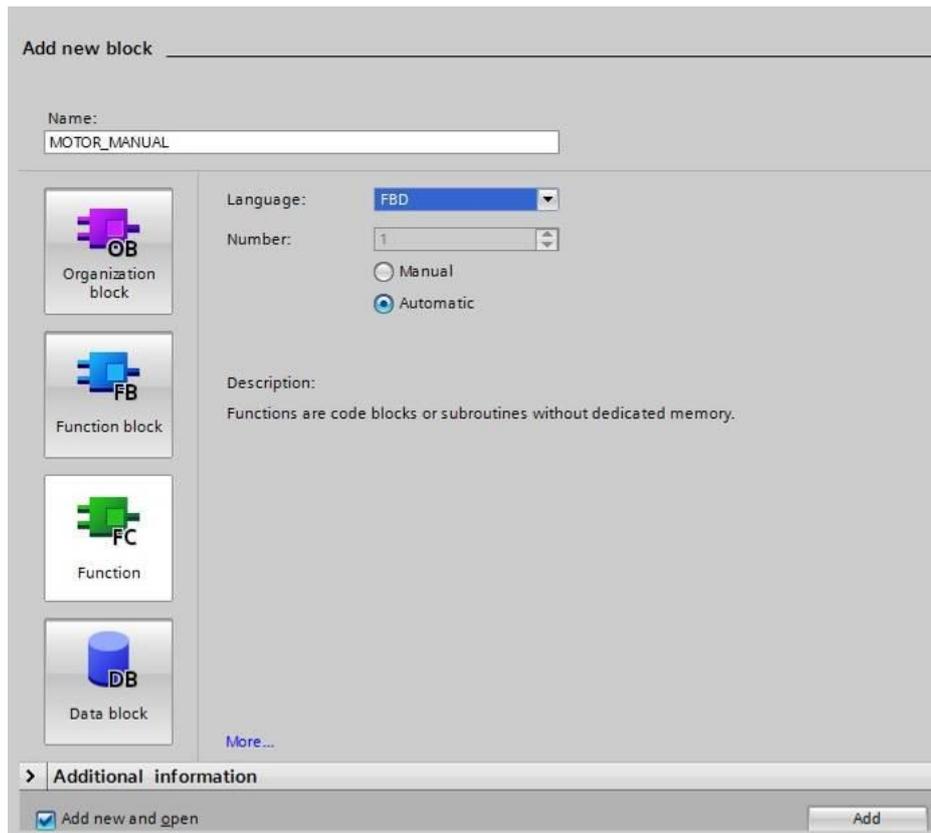
→ Clique no Portal, na seção Programação PLC em "Add new block" (Adicionar novo bloco), para criar aqui uma função.

(→ PLC programming (Programação PLC) → Add new block (Adicionar novo bloco) →



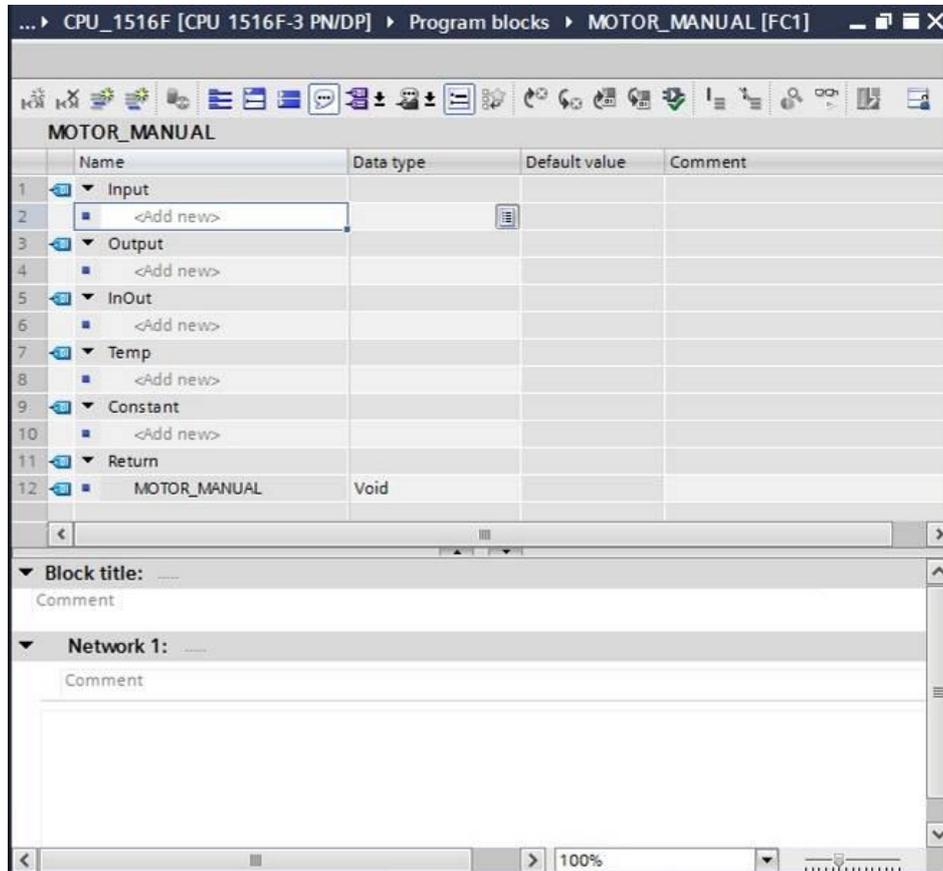
→ Denomine seu novo módulo com o nome: "MOTOR\_MANUAL", comute a linguagem para FBD e deixe conceder o número automaticamente. Ative a marca de verificação "Add new and open" (Adicionar novo bloco e abrir), e automaticamente você chegará na visualização do projeto do seu bloco de função criado. Clique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

(→Name (Nome): MOTOR\_manual→ Language (Idioma): FBD → Number (Número): automaticamente, →  Add new and open (Adicionar novo bloco e abrir) → Add (Adicionar))



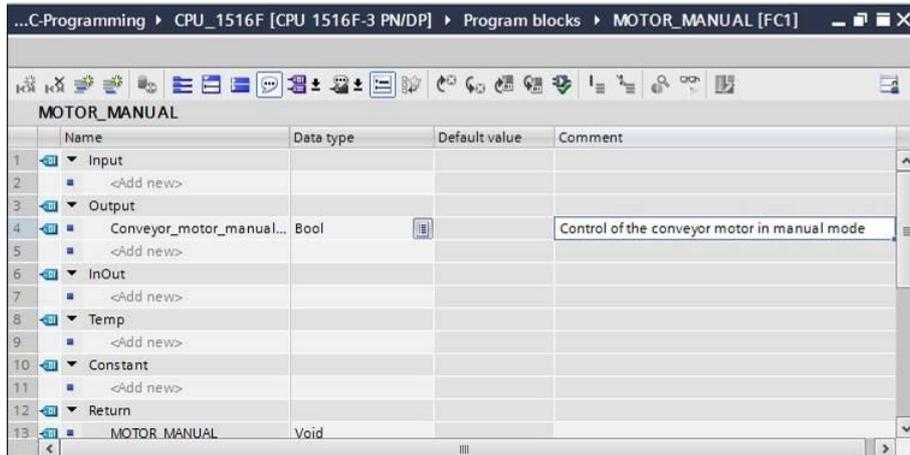
## 7.6 Determinar a interface da função FC1 "MOTOR\_MANUAL"

- 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.



→ 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)".

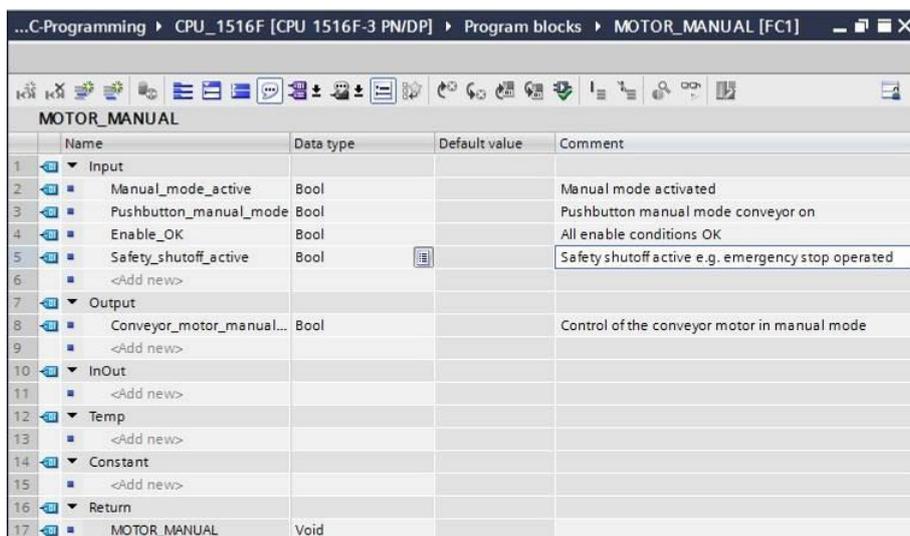
(→ Output: Conveyor\_motor\_manual\_mode (Motor de correia\_comando por impulsos → Bool → Control of the conveyor motor in manual mode (Acionar motor de correia no comando por impulsos))



→ 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).

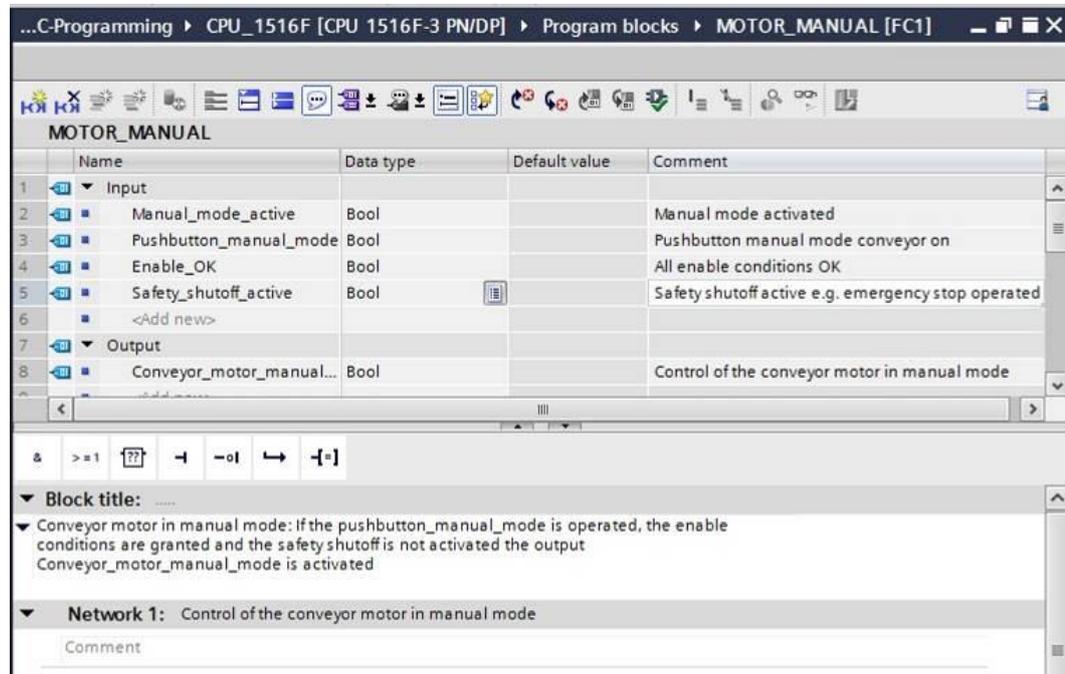
(→ Manual\_mode\_active (Operação\_manual\_ativo → Enter → Bool → Manual mode activated (Tipo de operação manual ativado))

→ 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.



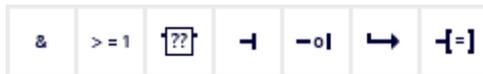
→ Para a documentação do programa conceda o título do módulo, um comentário sobre o módulo e para a rede 1 um título de rede útil.

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



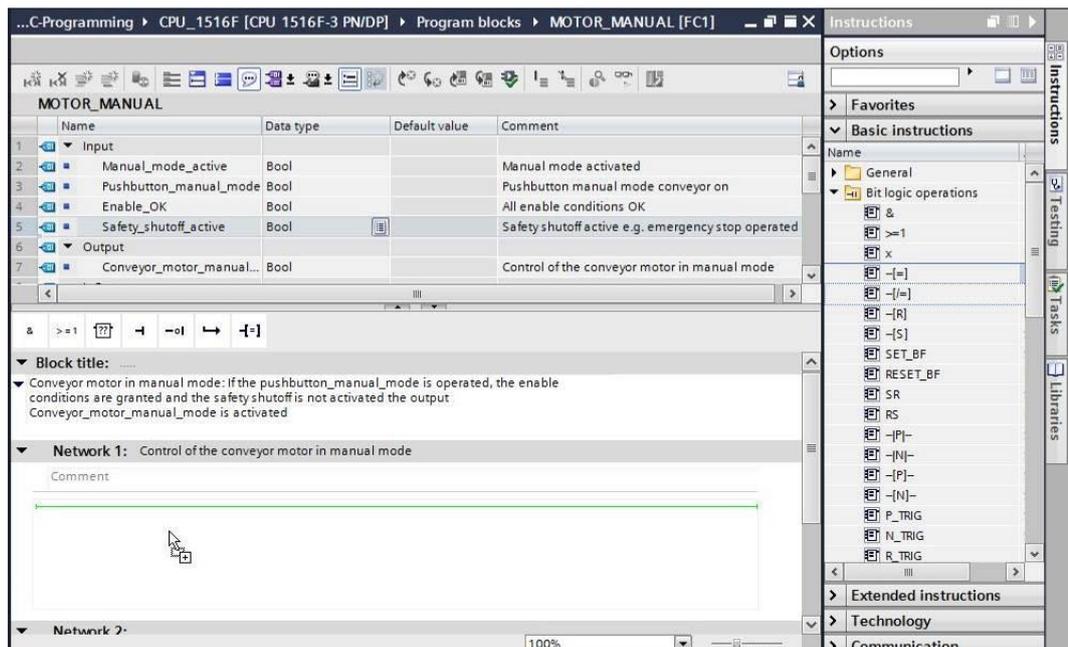
## 7.7 Programação do FC1: MOTOR\_MANUAL

→ 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.



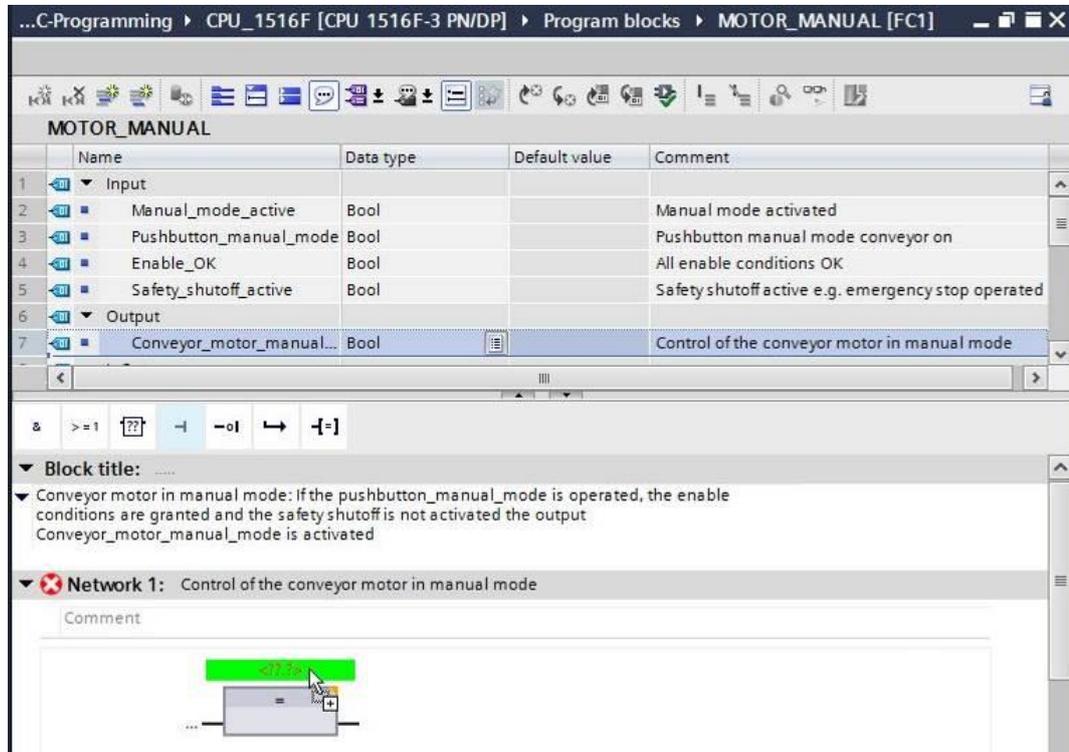
→ No lado direito de sua janela de programação poderá ver uma lista de instruções, que poderá utilizar no programa. Busque em → Basic instructions (Instruções simples) → conexões de bit conforme a função –[=] (Atribuição) e arraste estas para sua rede 1 (linha verde aparece, mouse com símbolo +).

(→ Instructions (Instruções) → Basic instructions (Instruções simples) → Bit logic operation (Conexão bit) → –[=])



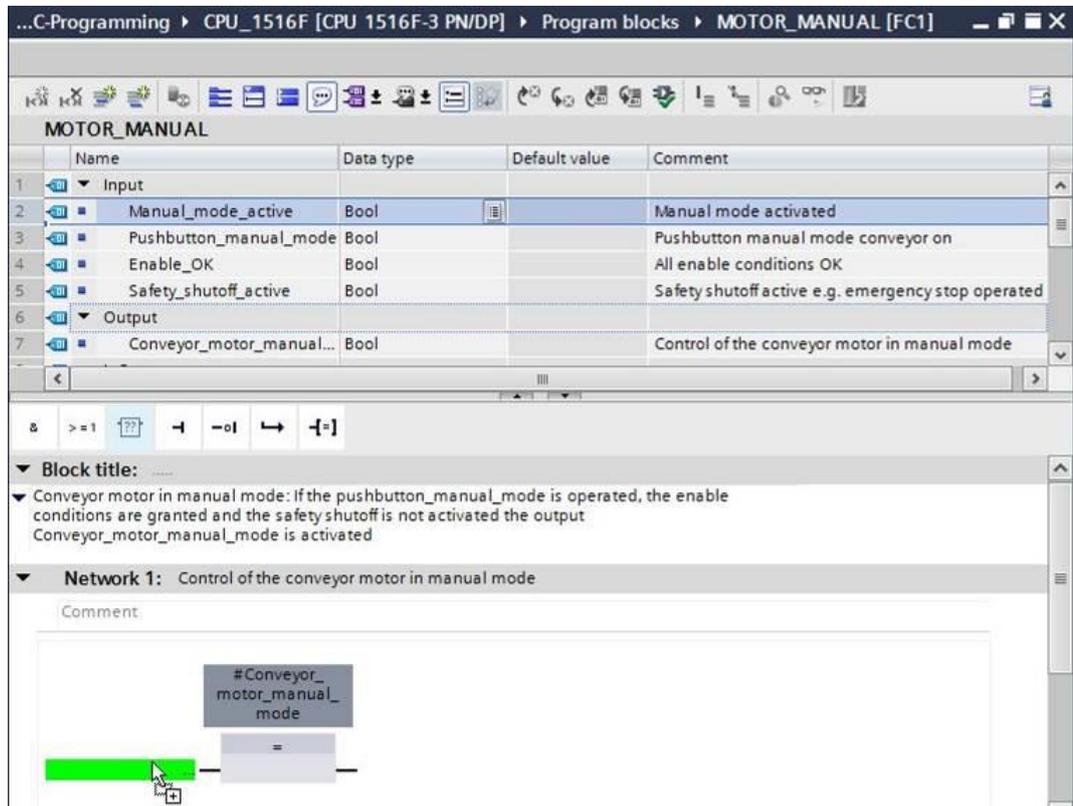
→ Arraste para o seu parâmetro Output #Motor da correia\_Comando por impulsos por meio de Drag and Drop para <??.?> sobre o seu bloco recém inserido. Poderá selecionar mais facilmente um parâmetro na descrição de interfaces, segurando-o pelo símbolo azul

(→ Conveyor\_motor\_manual\_mode (Motor de correia\_comando por impulsos))



→ 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.

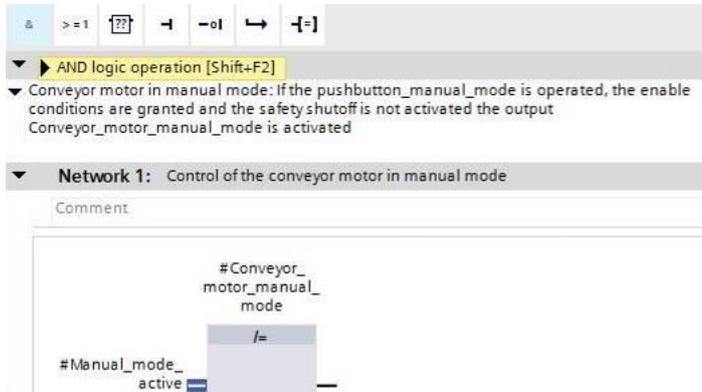
(→  Manual\_mode\_active (Operação manual\_ativo))



→ 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.

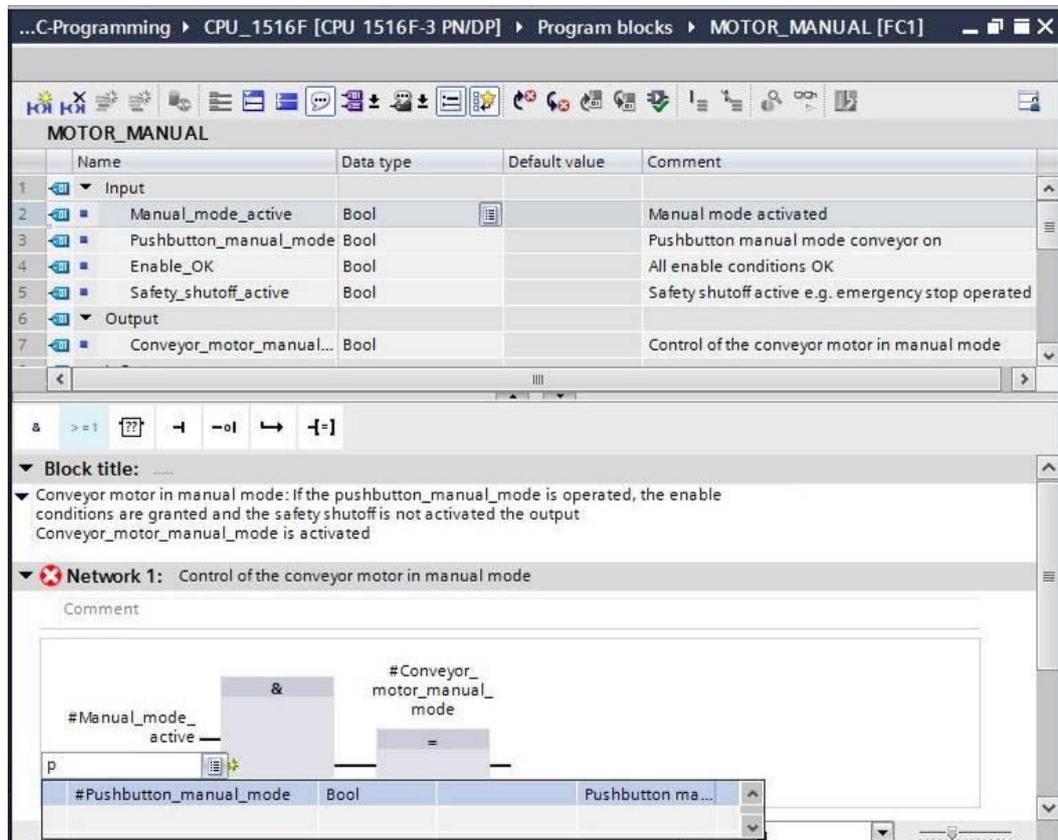


→ Clique sobre o símbolo  na sua barra de símbolos lógicos, para inserir uma interligação E entre as variáveis #Operação manual\_ativa e o módulo de ramificação.



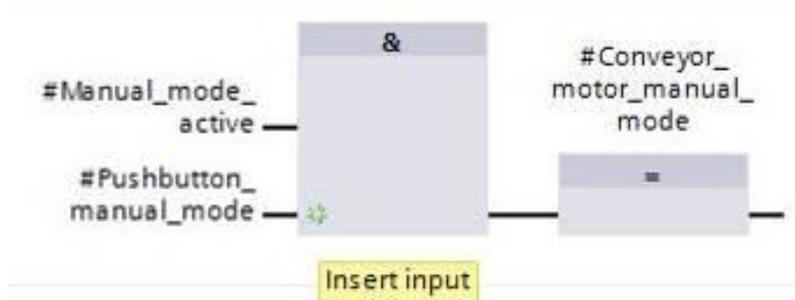
→ Efetue um clique duplo sobre a segunda entrada da interligação &  e, no campo exibido a seguir, digite a letra "T", para visualizar uma lista das variáveis disponíveis, que começam com a letra "T". Clique na variável #Pushbutton\_manual\_mode (Tecla\_comando por impulsos) e confirme com → Enter.

(→ Block & (Bloco &) →  → T → #Pushbutton\_manual\_mode (Tecla\_comando por impulsos) → Enter)

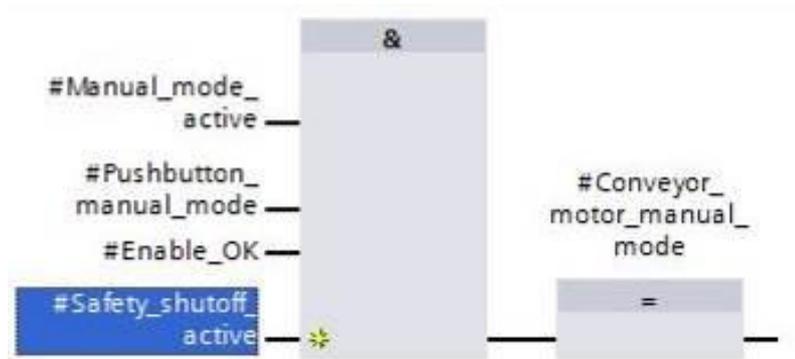


**Nota:** No caso desta variante de atribuição de variáveis há o risco de uma confusão com as variáveis globais da tabela de variáveis. Por isto, deve ser dada a preferência para a variante apresentada antes com o Drag and Drop a partir da descrição das interfaces.

→ 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  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.



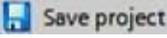
→ Negue a entrada conectada com o parâmetro #Desativação de proteção\_ativa, marcando-o e clicando em seguida sobre .

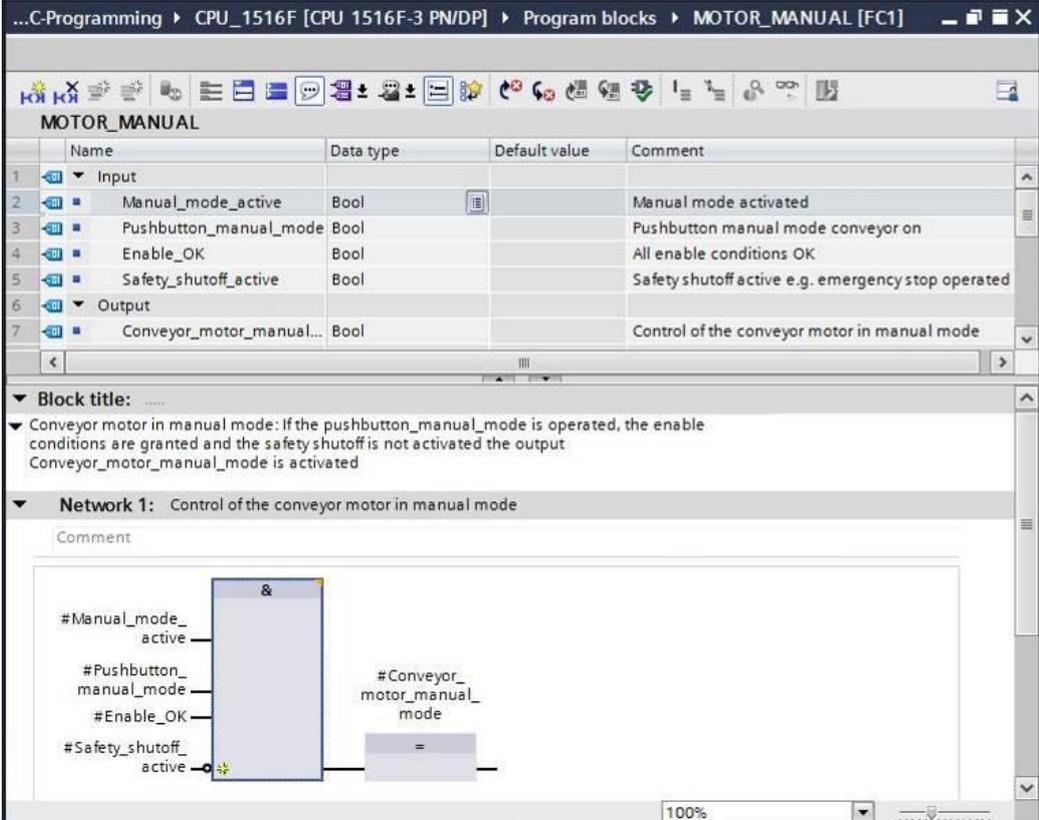
**Block title:** Invert RLO [Ctrl+Shift+4]

Conveyor motor in manual mode: when the pushbutton\_manual\_mode is operated, the enable conditions are granted and the safety shutoff is not activated the output Conveyor\_motor\_manual\_mode is activated

**Network 1:** Control of the conveyor motor in manual mode

Comment

→ Não se esqueça de clicar sobre . A função concluída de "MOTOR\_MANUAL [FC1] em FBD é apresentada a seguir.



The screenshot displays the TIA Portal interface for the MOTOR\_MANUAL [FC1] block. It is divided into two main sections: a variable declaration table and a logic network.

	Name	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	Output			
7	Conveyor_motor_manual...	Bool		Control of the conveyor motor in manual mode

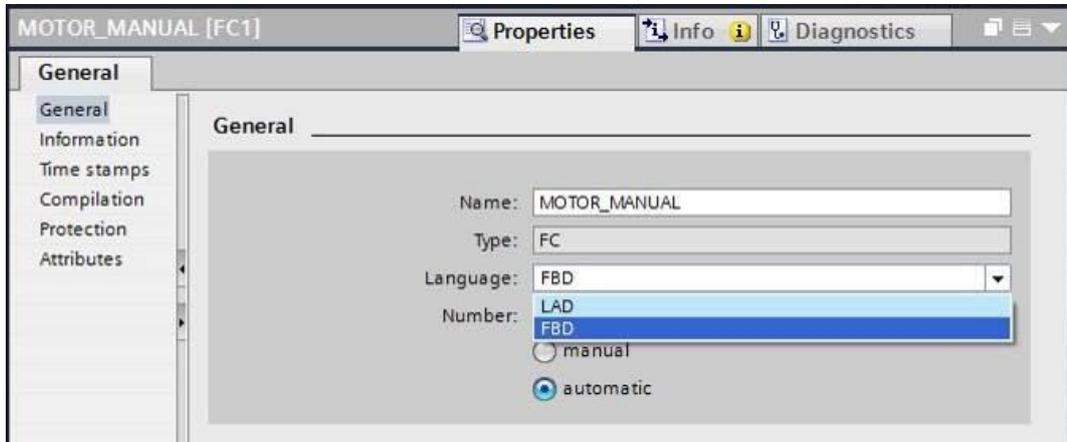
**Block title:** .....

**Conveyor motor in manual mode:** If the pushbutton\_manual\_mode is operated, the enable conditions are granted and the safety shutoff is not activated the output Conveyor\_motor\_manual\_mode is activated

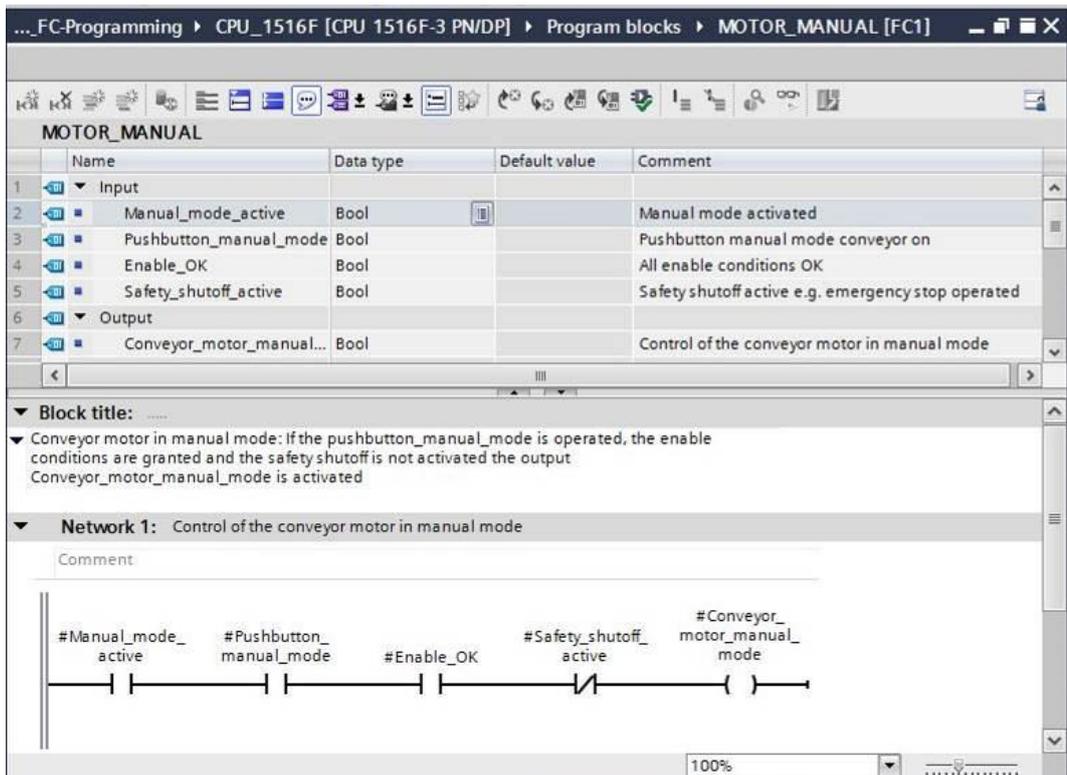
**Network 1:** Control of the conveyor motor in manual mode

**Logic Network 1:** This is a normally open contact network. It consists of four input variables connected in series to an AND gate (represented by a box with '&'). The inputs are: #Manual\_mode\_active, #Pushbutton\_manual\_mode, #Enable\_OK, and #Safety\_shutoff\_active. The output of the AND gate is connected to an assignment coil (represented by a box with '=') for the output variable #Conveyor\_motor\_manual\_mode.

- 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)



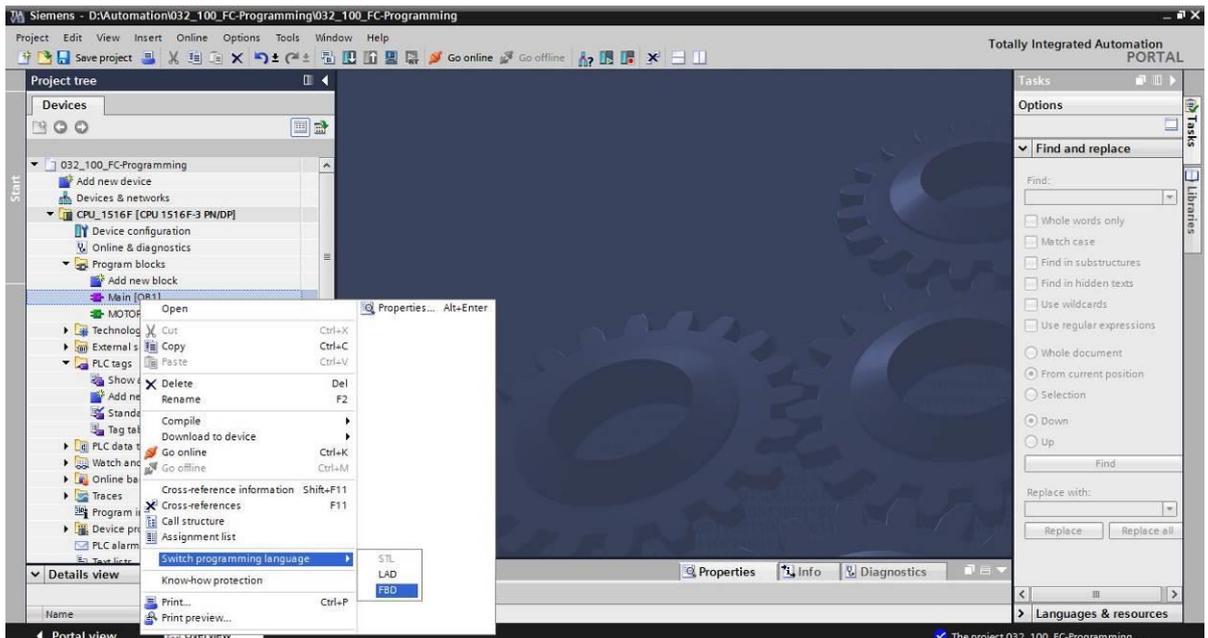
- Em LAD, o programa apresenta a seguinte aparência.



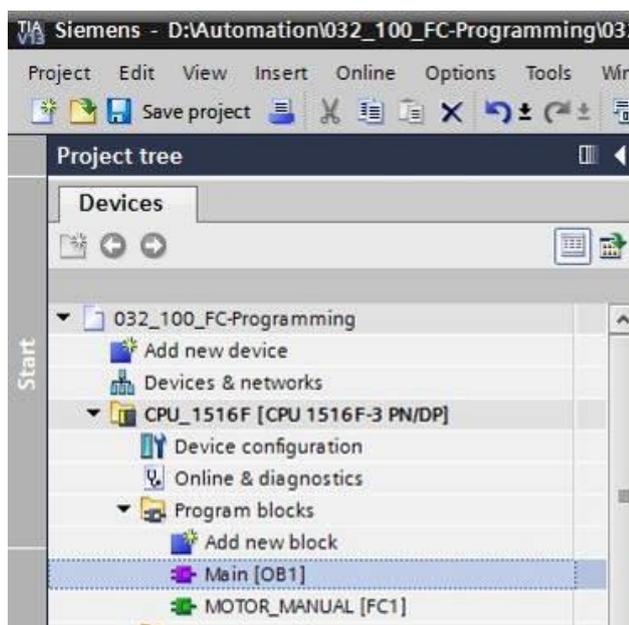
## 7.8 Programação do módulo organizacional OB1 – Comando do funcionamento da correia para frente em operação manual

→ 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]".

(→ CPU\_1516F[CPU 1516F-3 PN/DP → Program blocks" (Blocos de programa) →Main [OB1] → Switch programming language (Linguagem de programação) →FBD)

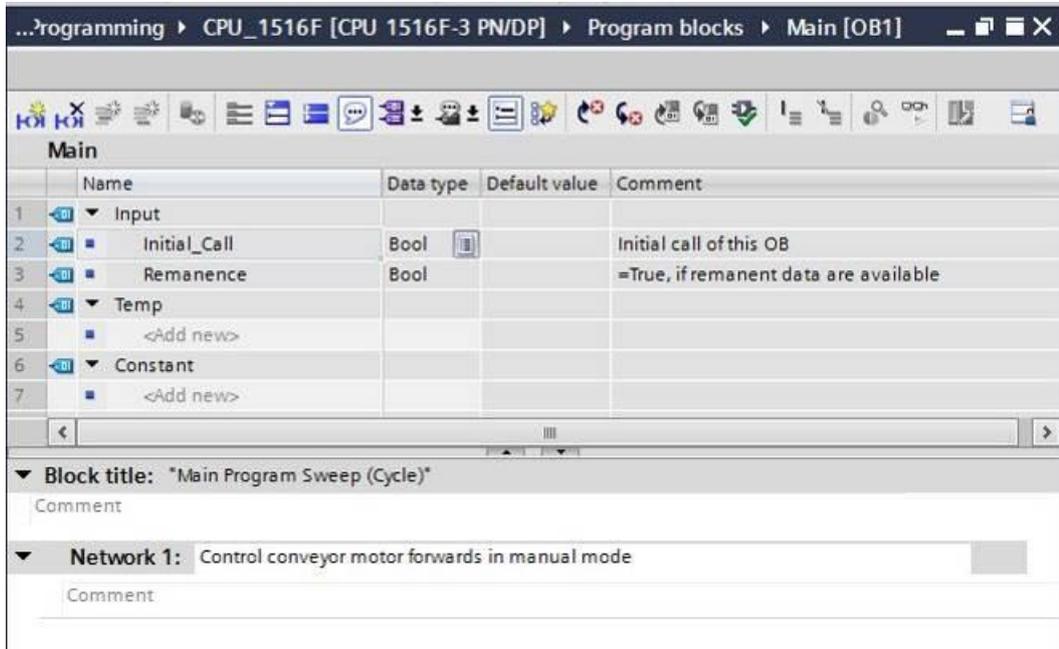


→ Abra agora o módulo organizacional "Main [OB1]" com um clique duplo.

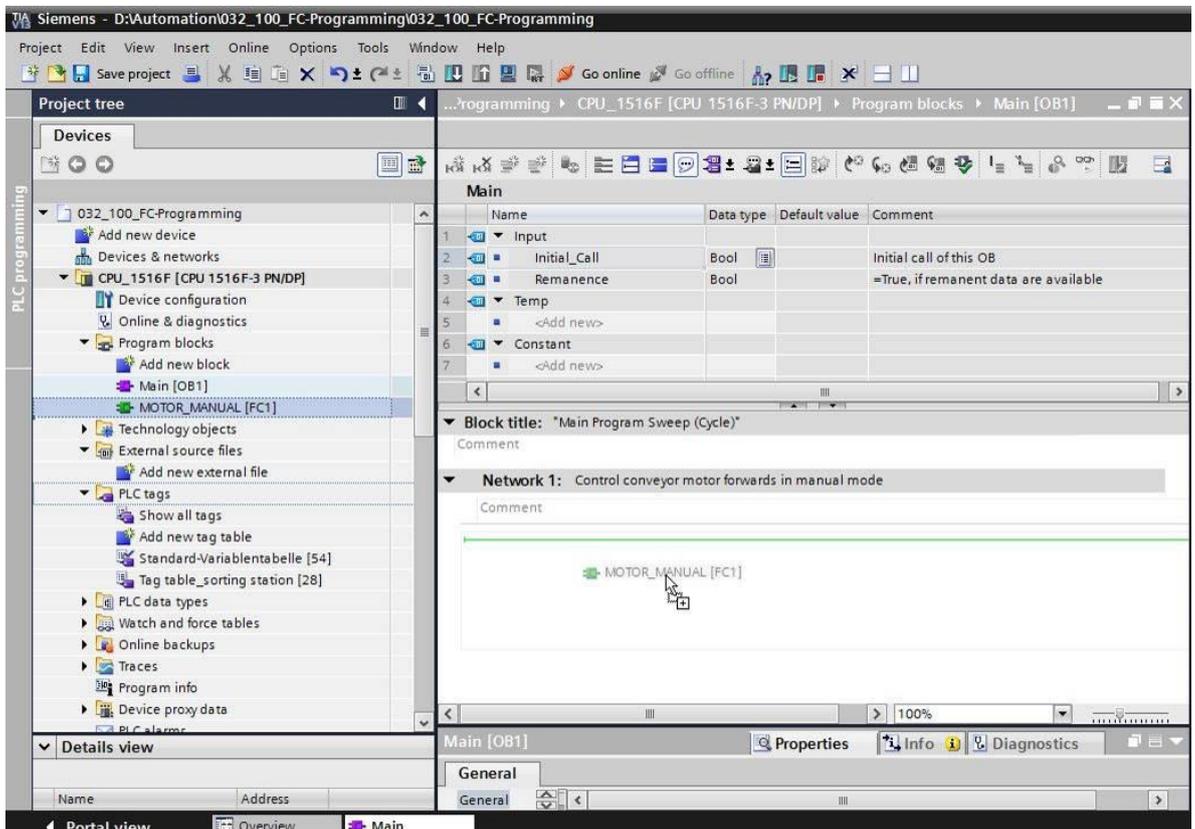


→ Nomeie a rede 1 como "Control conveyor motor forwards in manual mode" (Acionamento movimento da correia para frente no manual-/comando por impulsos).

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



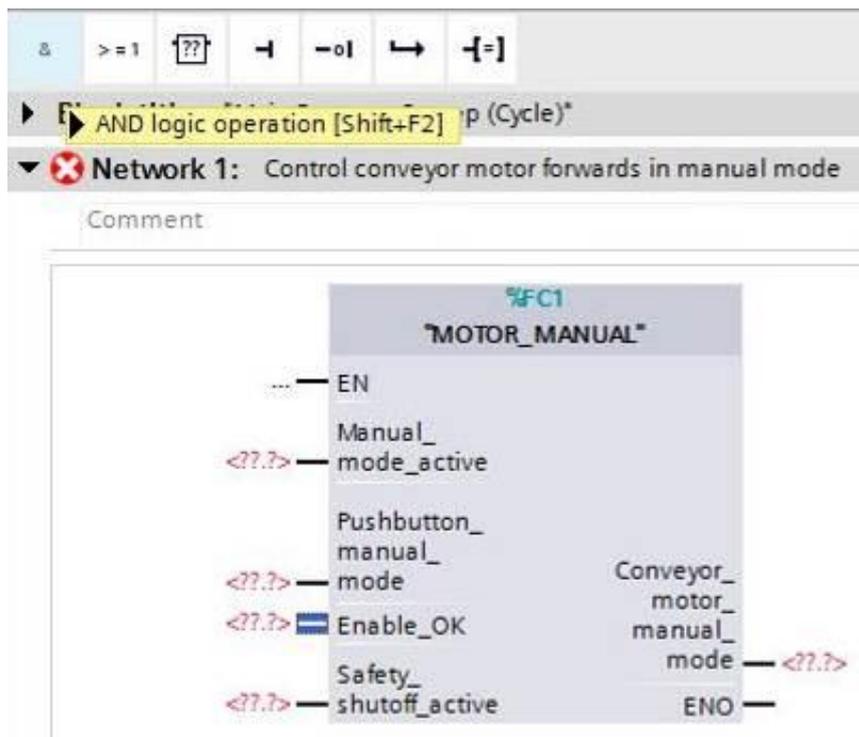
→ Arraste agora sua função "MOTOR\_MANUAL [FC1]" através do Drag and Drop para a rede 1 sobre a linha verde.



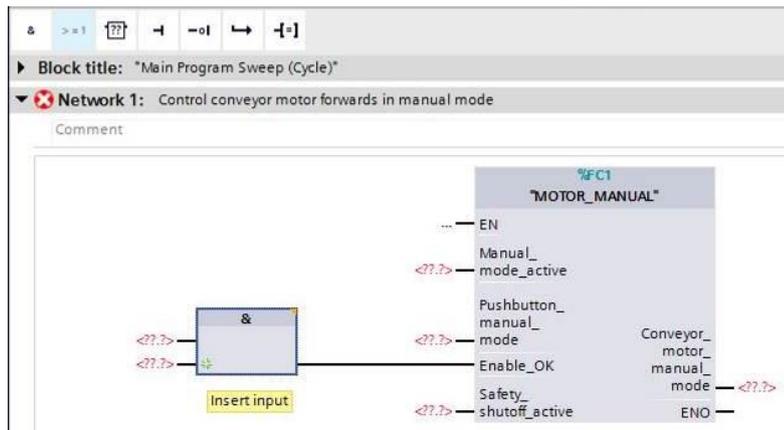
→ Um bloco com a interface determinada por você e as conexões EN e ENO será inseridos na rede 1.



→ 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. (→  )

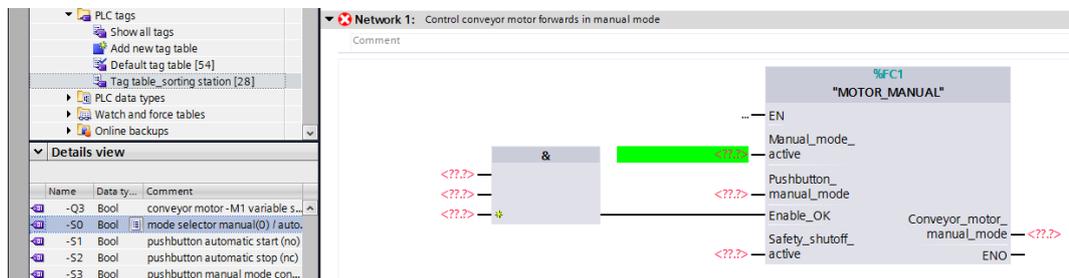


→ Clique sobre a estrela amarela  de seu elo E para inserir uma outra entrada. (→ )

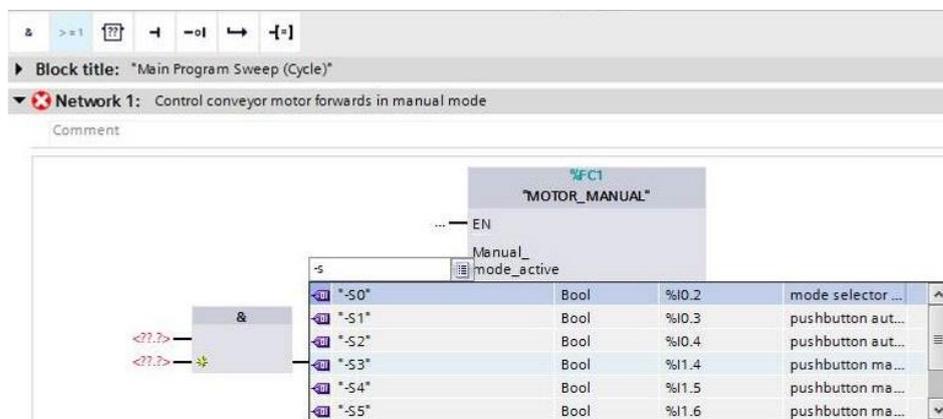


→ Para ligar o bloco com o variáveis globais da "Tag table\_sorting station" (Tabela de variantes\_sistema de classificação), temos 2 possibilidades:

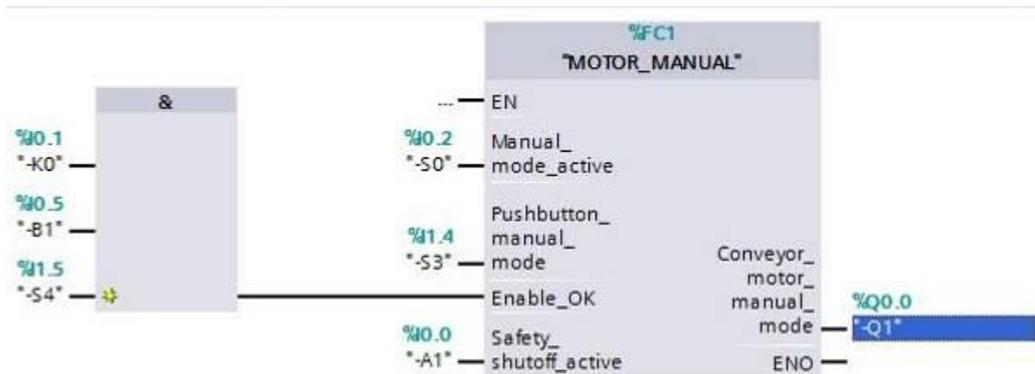
→ Ou selecionado a "Tag table\_sorting station" (Tabela de variantes\_sistema de classificação) na árvore do projeto e arrastando a variável global da visualização detalhada na interface do FC1 (→ Tag table\_sorting station (Tabela de variantes\_sistema de classificação) → Details view (Visualização detalhada) → -S0 → Manual\_mode\_active (Operação manual\_ativo))



→ 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). (→ Manual\_mode\_active (Operação manual\_ativo) → -S → -S0)

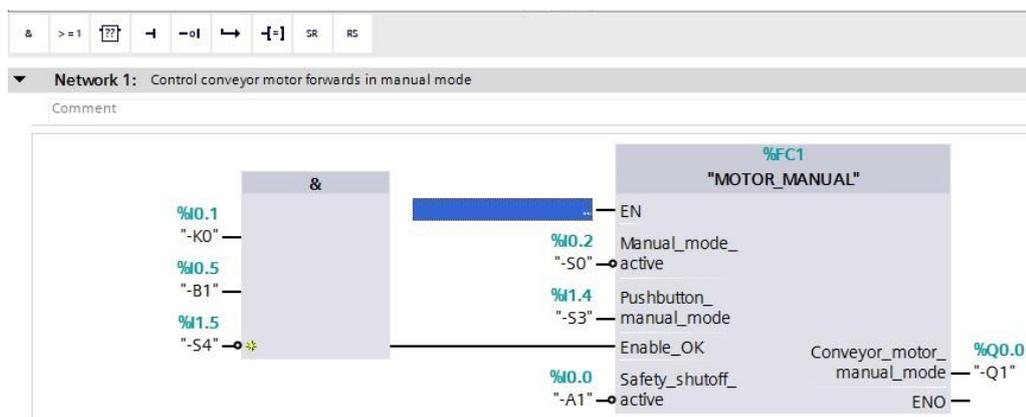


→ Adicione as outras variáveis de entrada "-S3", "-K0", "-B1", "-S4" e "-A1" e na saída "Conveyor\_motor\_manual\_mode" (Motor de correia\_comando por impulsos) a variável de saída "-Q1" (%A0.0).



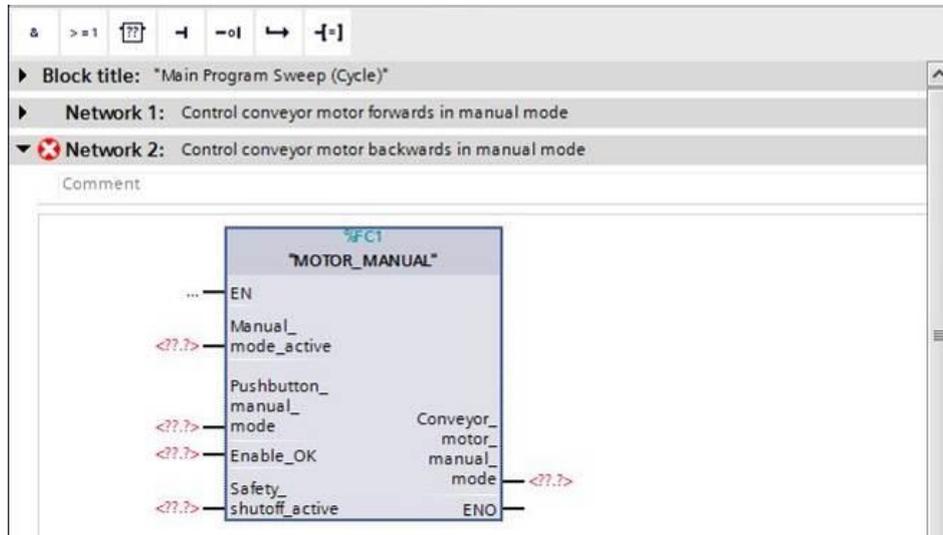
→ Negue as consultas das variáveis de entrada "-S0", "-S4" e "-A1", marcando-as e

clicando em seguida sobre  (→ -S0 →  → -S4 →  → -A1 → )

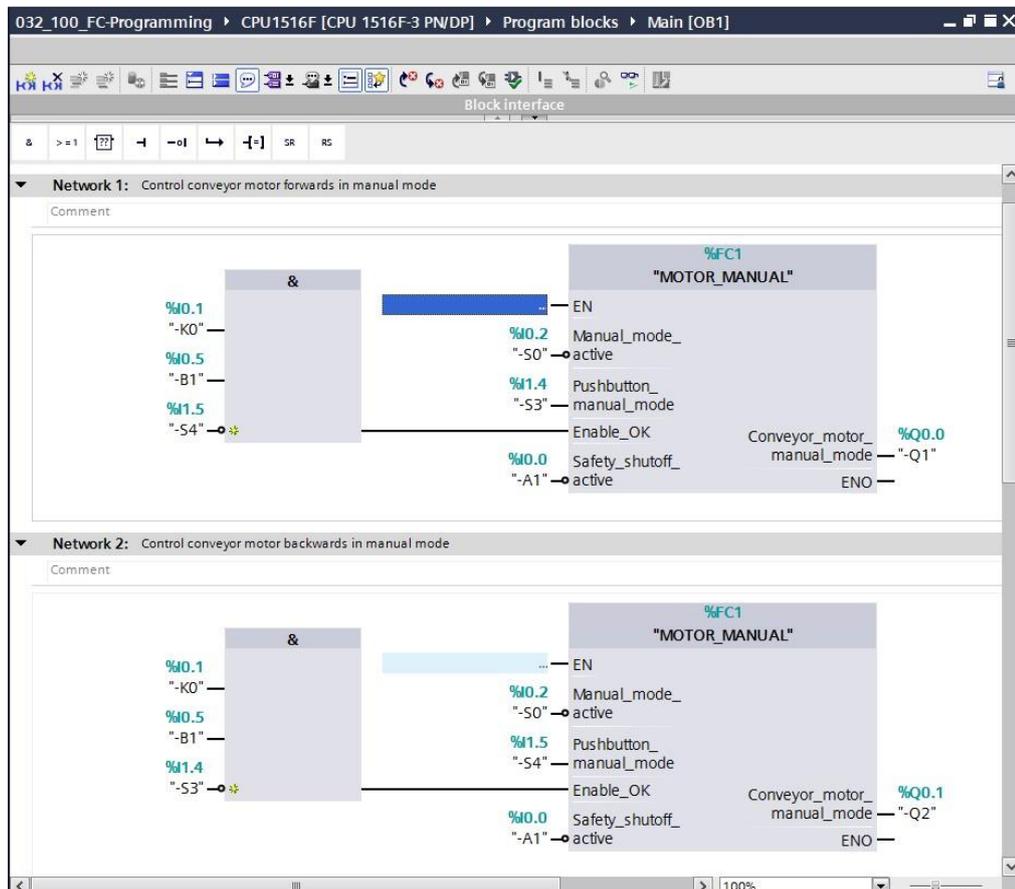


## 7.9 Programação do módulo organizacional OB1 – Comando do funcionamento da correia para trás em operação manual

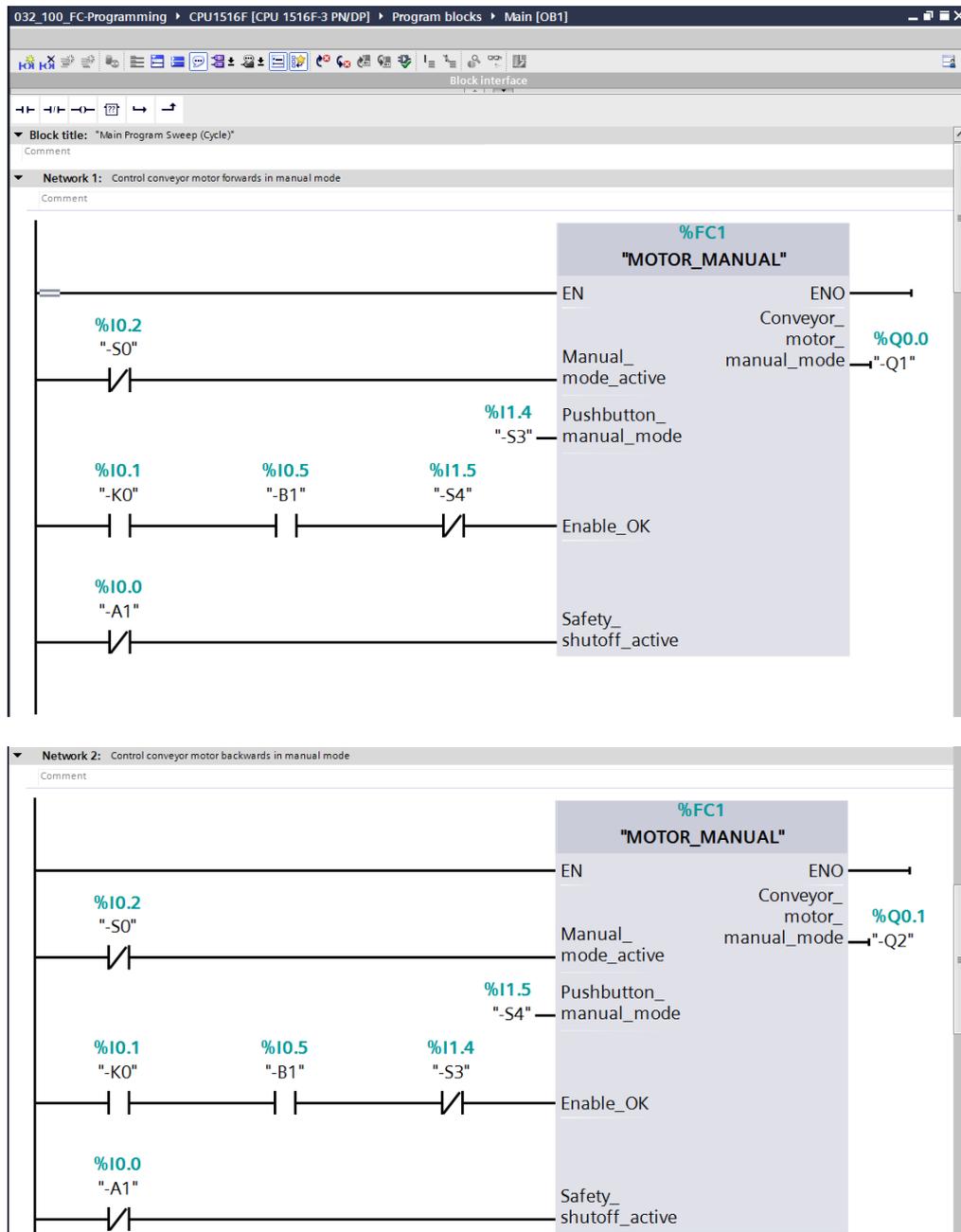
- Conceda à rede 2 o nome de "Control conveyor motor backwards in manual mode" (Ativação do funcionamento da correia para trás em operação manual/comando por impulsos) e acrescente, como já executado na rede 1, a sua função "MOTOR\_MANUAL [FC1]" por meio de Drag and Drop.



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

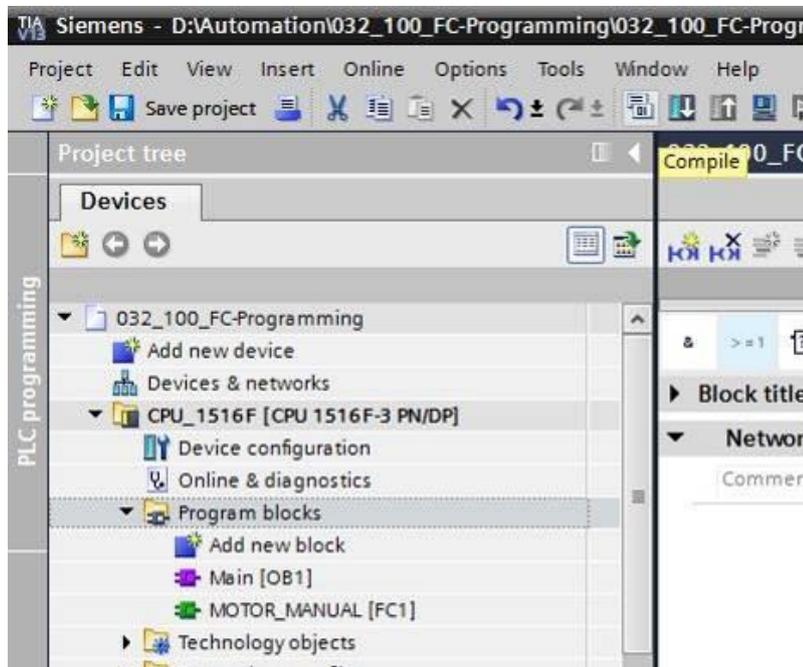


→ Na linguagem de programação LAD (plano de contato), o resultado apresenta a seguinte aparência.

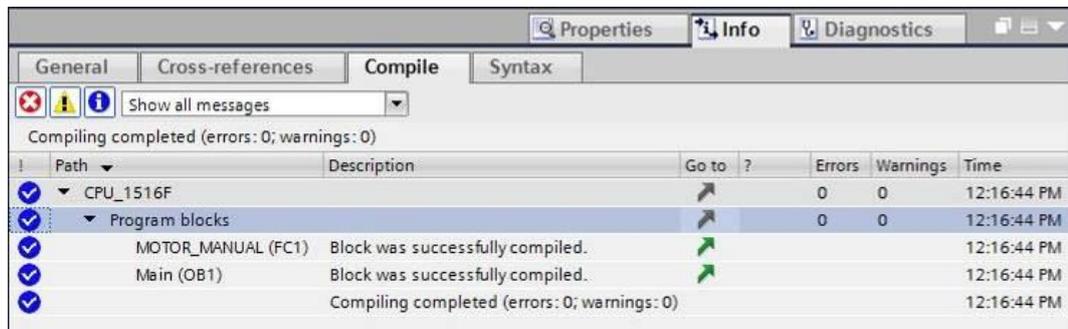


## 7.10 Salvar e compilar o programa

- 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) → )



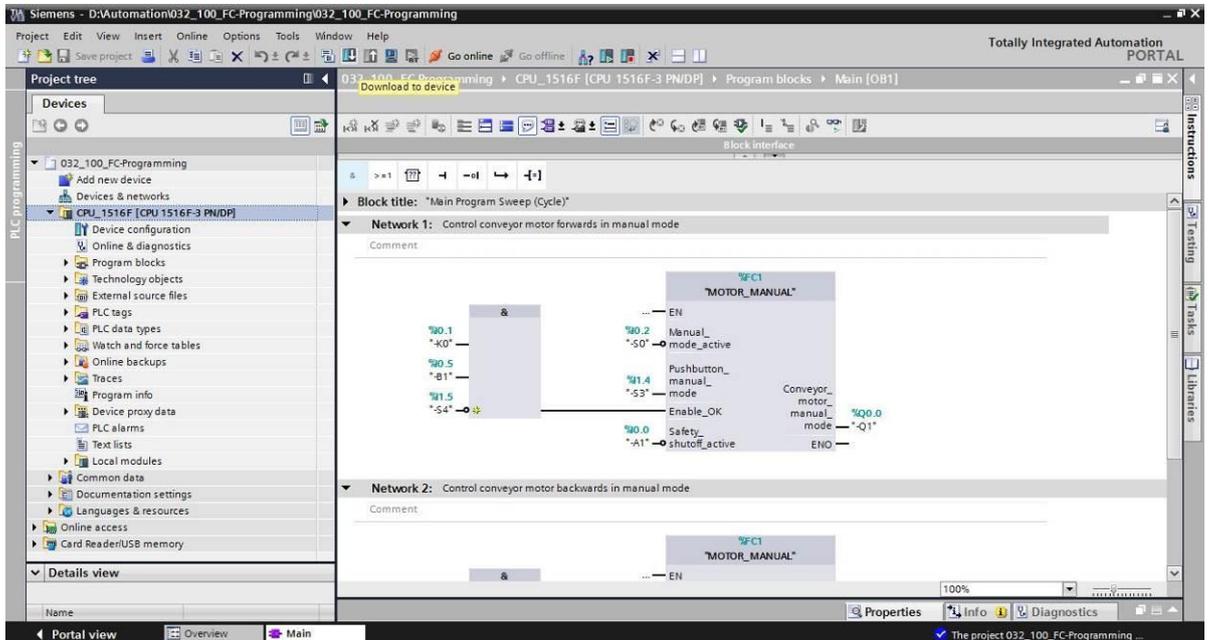
- Na área "Info" "Compile" (Verter) será mostrado em seguida, qual bloco pôde ser vertido com êxito.



Path	Description	Go to	Errors	Warnings	Time
▼ CPU_1516F			0	0	12:16:44 PM
▼ Program blocks			0	0	12:16:44 PM
MOTOR_MANUAL (FC1)	Block was successfully compiled.				12:16:44 PM
Main (OB1)	Block was successfully compiled.				12:16:44 PM
	Compiling completed (errors: 0; warnings: 0)				12:16:44 PM

## 7.11 Carregar o programa

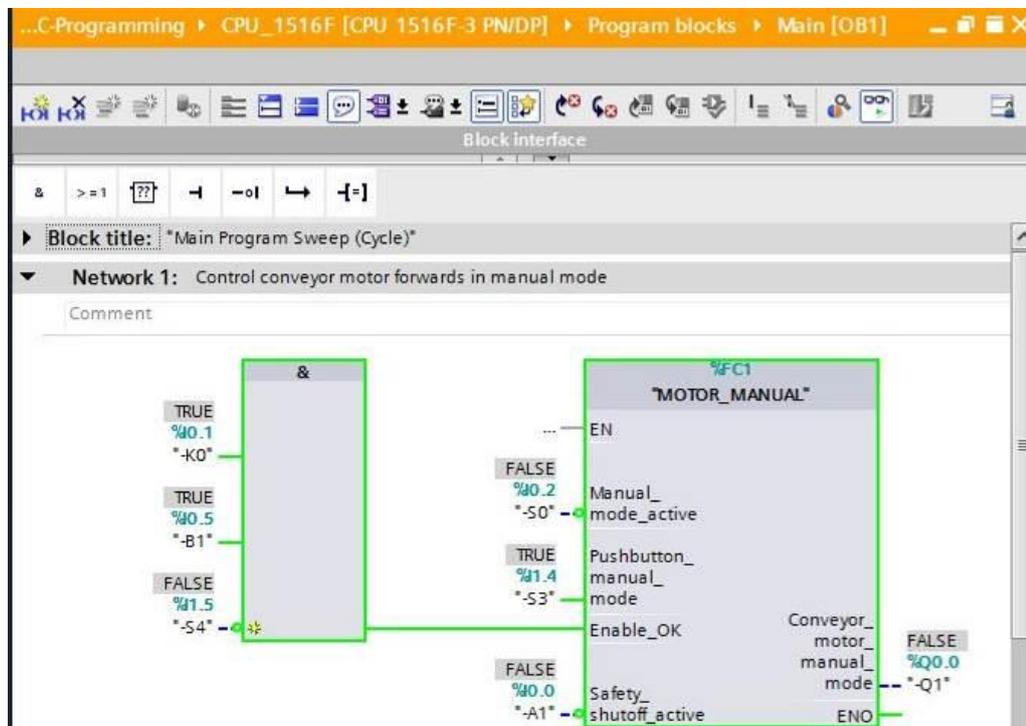
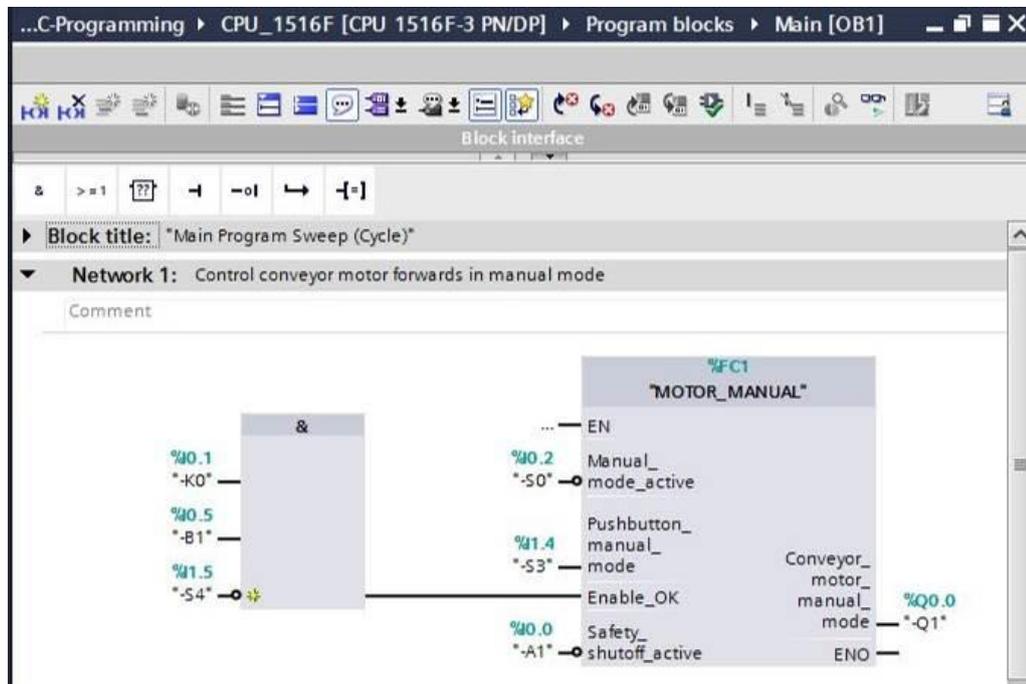
- Após uma compilação bem sucedida, o completo comando pode ser carregado com o programa criado, como foi já descrito nos módulos referentes à configuração de hardware. (→ )



## 7.12 Observar os módulos do programa

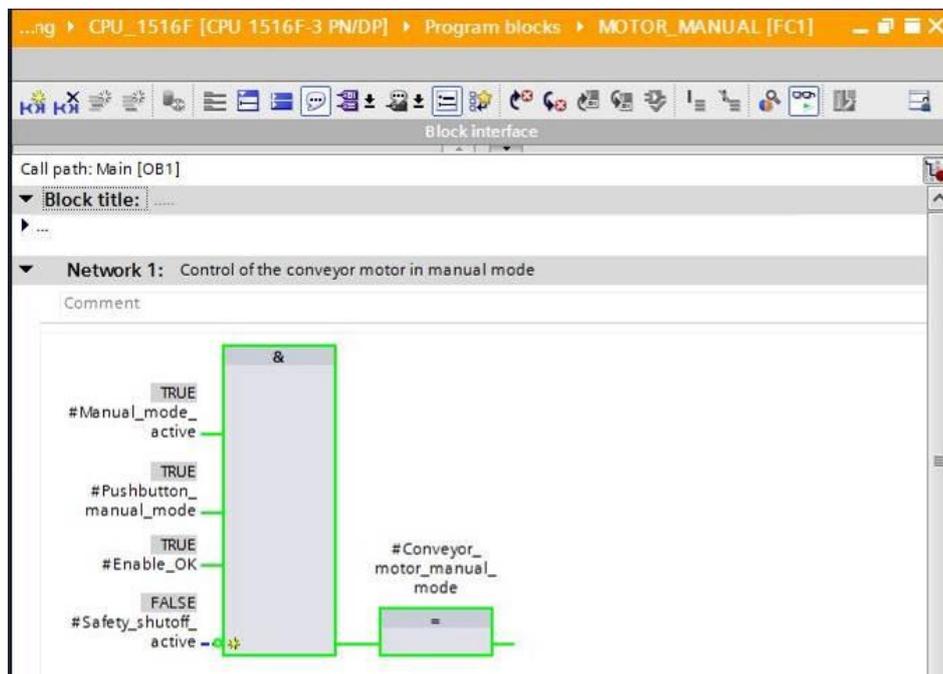
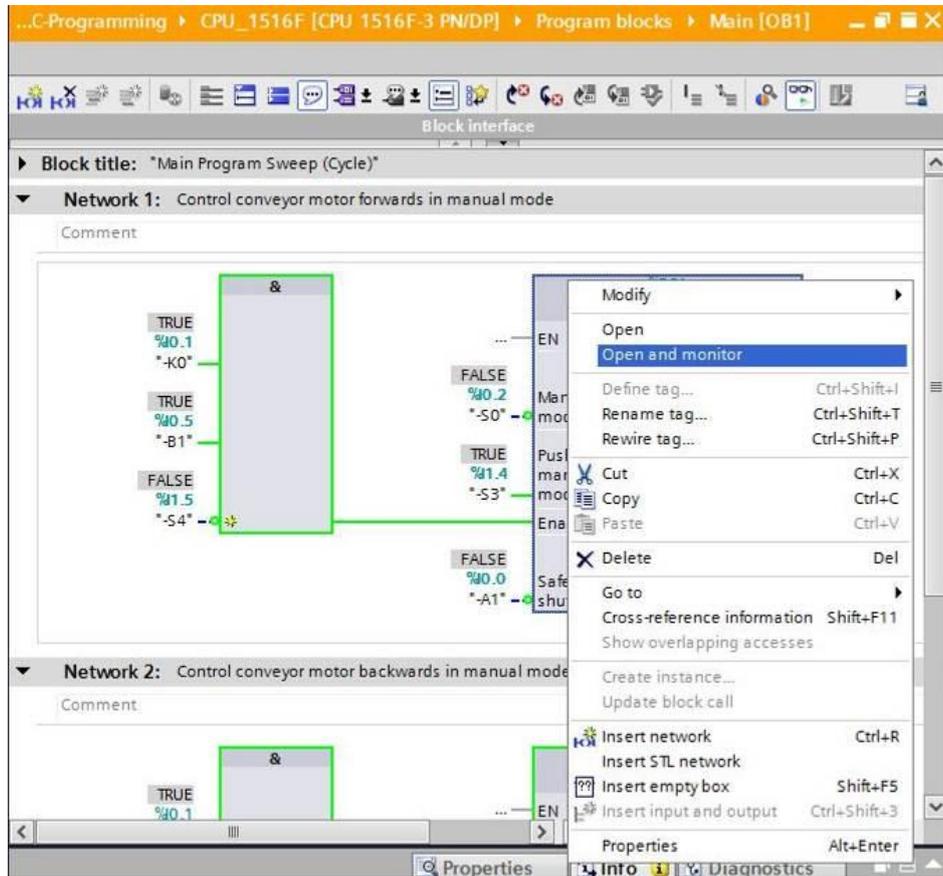
→ Para a observação do programa carregado, o módulo desejado deve ter sido aberto.

Com um clique sobre o símbolo  a observação pode ser ligada/desligada. (→ Main [OB1] → )



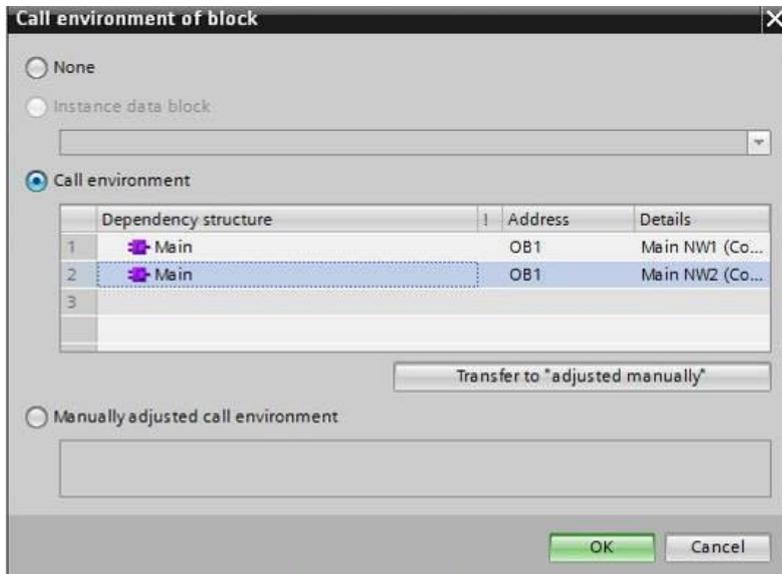
**Nota:** A observação é realizada aqui com relação ao sinal e em função do comando. Os estados de sinal nos bornes são exibidos com TRUE ou FALSE.

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



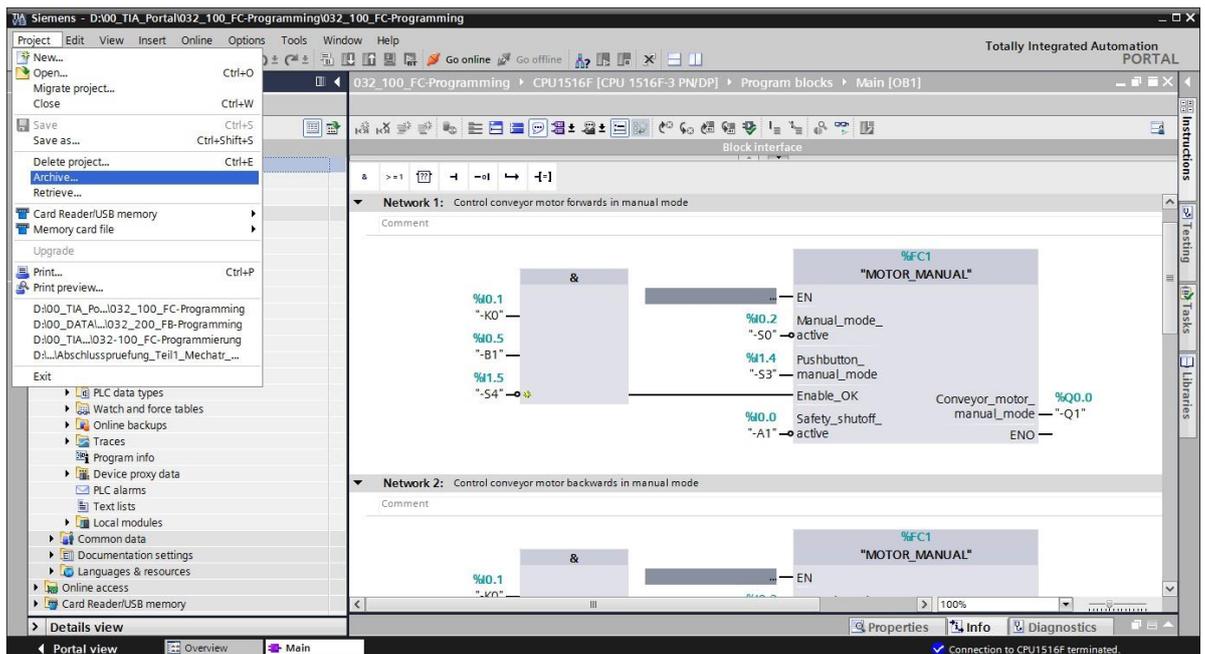
**Nota:** A observação é realizada aqui com relação à função e independente do comando. O acionamento do encoder ou o estado do sistema são apresentados aqui com TRUE ou FALSE.

- 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)



## 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) → -Q1 = 0	
5	Como em 3, mas modo de operação AUTOMÁTICO (-S0 = 1) → -Q1 = 0	
6	Como em 3, mas desligar o sistema (-K0 = 0) → -Q1 = 0	
7	Como em 3, mas cilindro não recolhido (-B1 = 0) → -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) → -Q2 = 0	
10	Como em 8, mas modo de operação AUTOMÁTICO (-S0 = 1) → -Q2 = 0	
11	Como em 8, mas desligar o sistema (-K0 = 0) → -Q2 = 0	
12	Como em 8, mas cilindro não recolhido (-B1 = 0) → -Q2 = 0	
13	Como em 8, mas também ativar o comando por impulsos para avançar a correia (-S3 = 1) → -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.

### 9.3 Lista de verificação – Exercício

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	

## 10 Informação 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](http://www.siemens.com/sce/s7-1500)