

### 3. Configuração dos instrumentos

#### 3.1. Introdução

A configuração dos instrumentos FIELDBUS consiste basicamente no interligamento lógico dos diversos blocos funcionais implementados em cada device da rede através de um software configurador (SYSCON) além da definição dos parâmetros de controle de cada bloco. São definidos também as conexões indicadoras de alarme e diversos eventos que podem ocorrer na malha de controle (ver Figura 3.1)

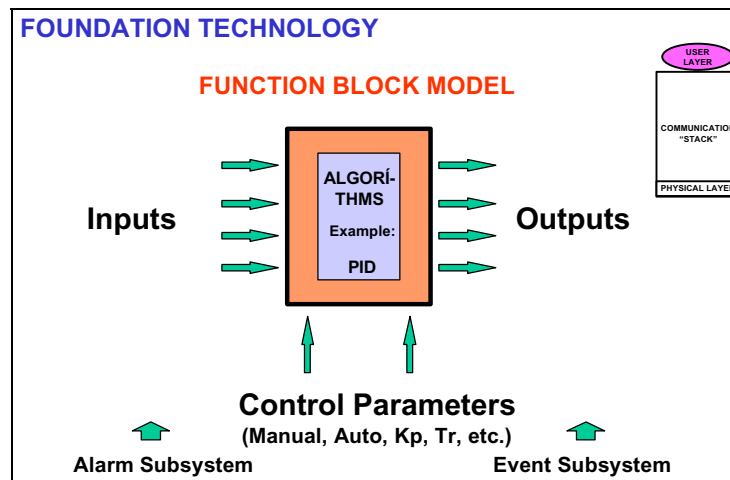


Figura 3.1 - Diagrama Geral de um bloco de função

Na figura acima temos a representação genérica de um bloco funcional que possui suas entradas, saídas e parâmetros de controle; fazendo uma referência aos níveis de protocolo (Figura 1.6), observa-se que este tipo de configuração envolve apenas a camada superior, ou seja, a camada do usuário; nenhuma preocupação é tomada com relação à forma como se procederá a comunicação (solicitação de token, etc.) nem tampouco como será entregue a comunicação ao nível físico.

O protocolo de comunicação é tratado internamente pelos próprios blocos funcionais que são responsáveis tanto pelas comunicações cíclicas (publicações de parâmetros para supervisão e links) quanto pelas comunicações acíclicas (notificações de alarmes/eventos, informações de diagnóstico e de display, etc.) - ver Figura 3.2.

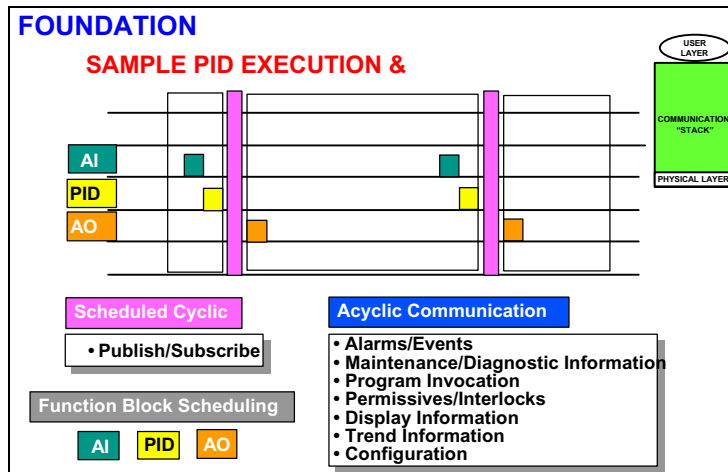


Figura 3.2 - Sequência no tempo de comunicação cíclica e acíclica

A conexão entre blocos de diferentes devices é definida como um *link* na rede. Os links são comuns nas malhas de controle FIELDBUS e imperceptível a quem esteja configurando o sistema (protocolo a nível de usuário) já que a comunicação é tratada diretamente pelos blocos funcionais (ver Figura 3.3).

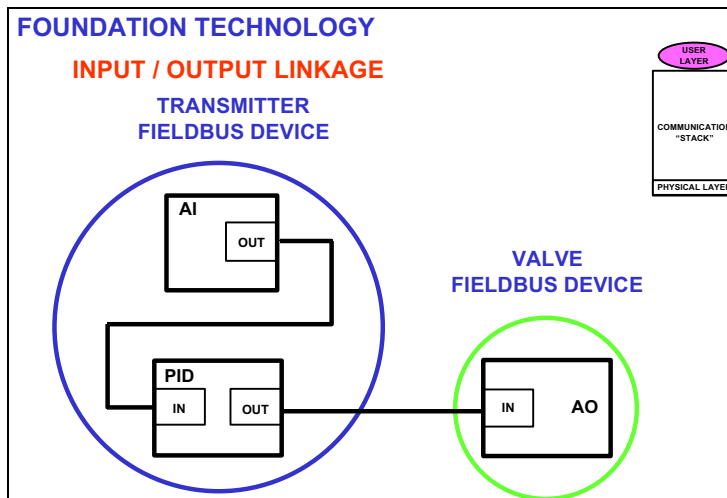


Figura 3.3 - Exemplo de conexão de blocos

### 3.2. Estudo dos blocos funcionais

Na Figura 3.4 são mostrados exemplos de blocos funcionais que implementam algumas funções básicas para a configuração FIELDBUS.

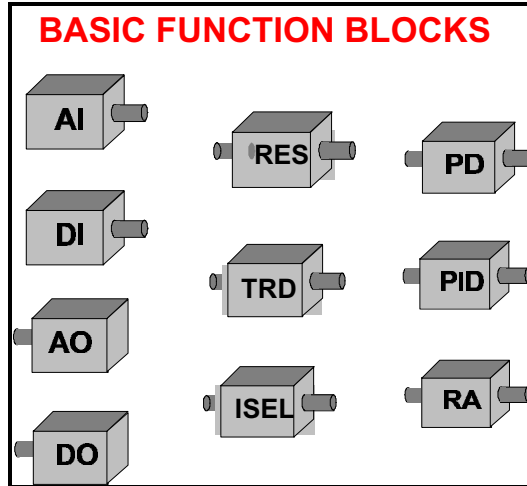


Figura 3.4 - Blocos de funções básicas

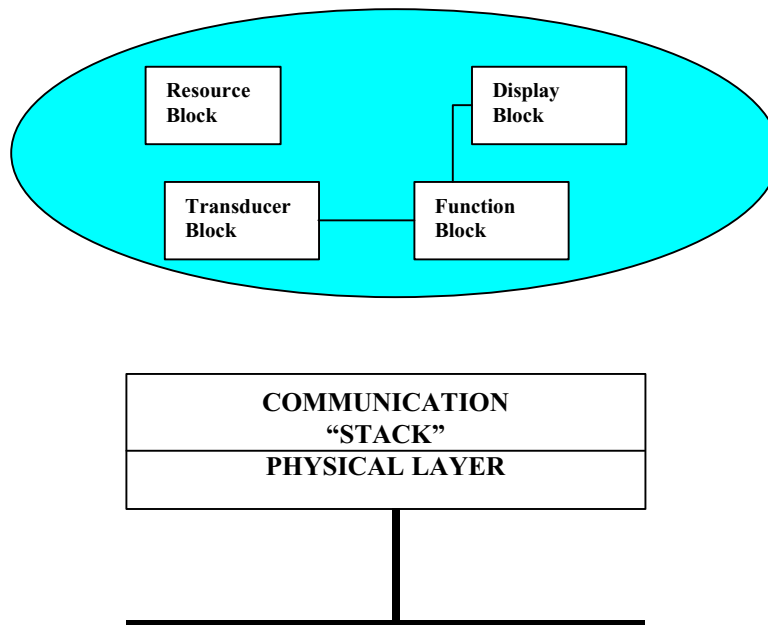


Figura 3.5 – Modelo de blocos para equipamentos de campo

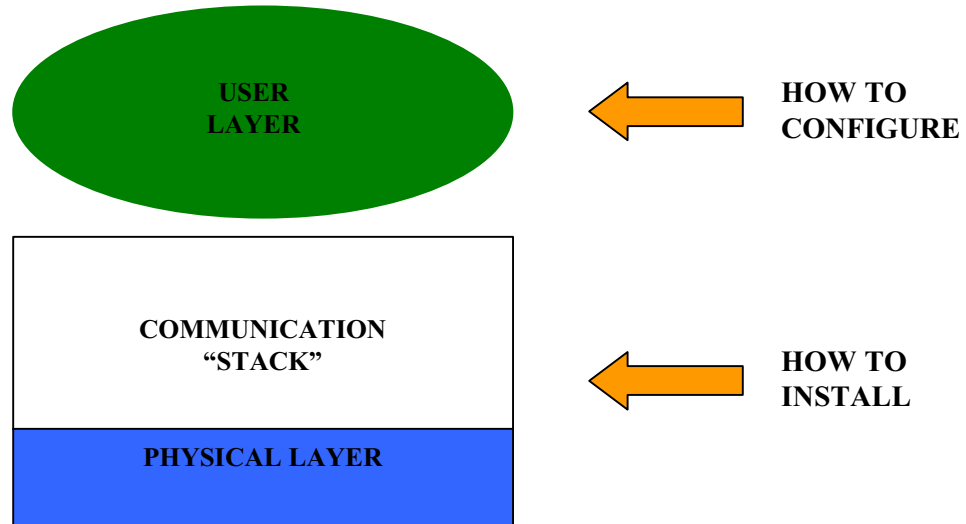


Figura 3.6 – Interface simples com o usuário

A combinação dos blocos (básicos e avançados) para a montagem da malha de controle pode ser feita intuitivamente, de forma simples, como sugere a Figura 3.7 que compara a elaboração de uma malha de controle em cascata com um simples jogo de encaixar.

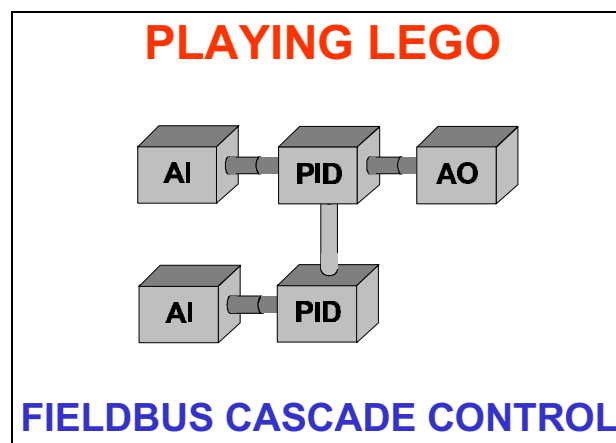


Figura 3.7 - Conexões de blocos de funções

Quando o usuário está configurando o sistema através do software SYSCON, é possível a visualização esquemática interna dos blocos funcionais, o que ajuda no entendimento interno do bloco facilitando configuração dos parâmetros internos bem como a interligação das entradas e saídas dos blocos funcionais (Figura 3.8)

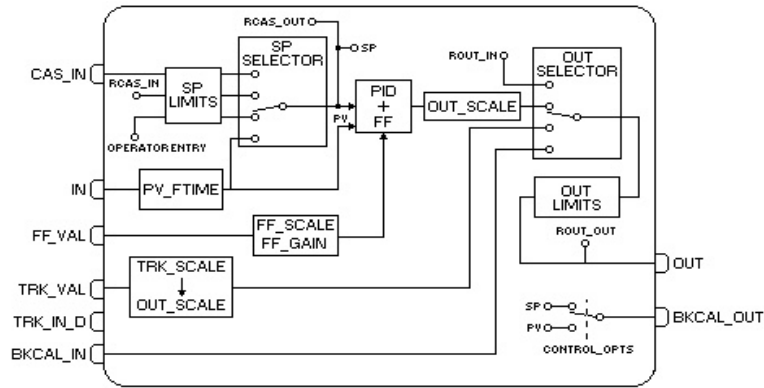


Figura 3.9- Diagrama de bloco de controle PID

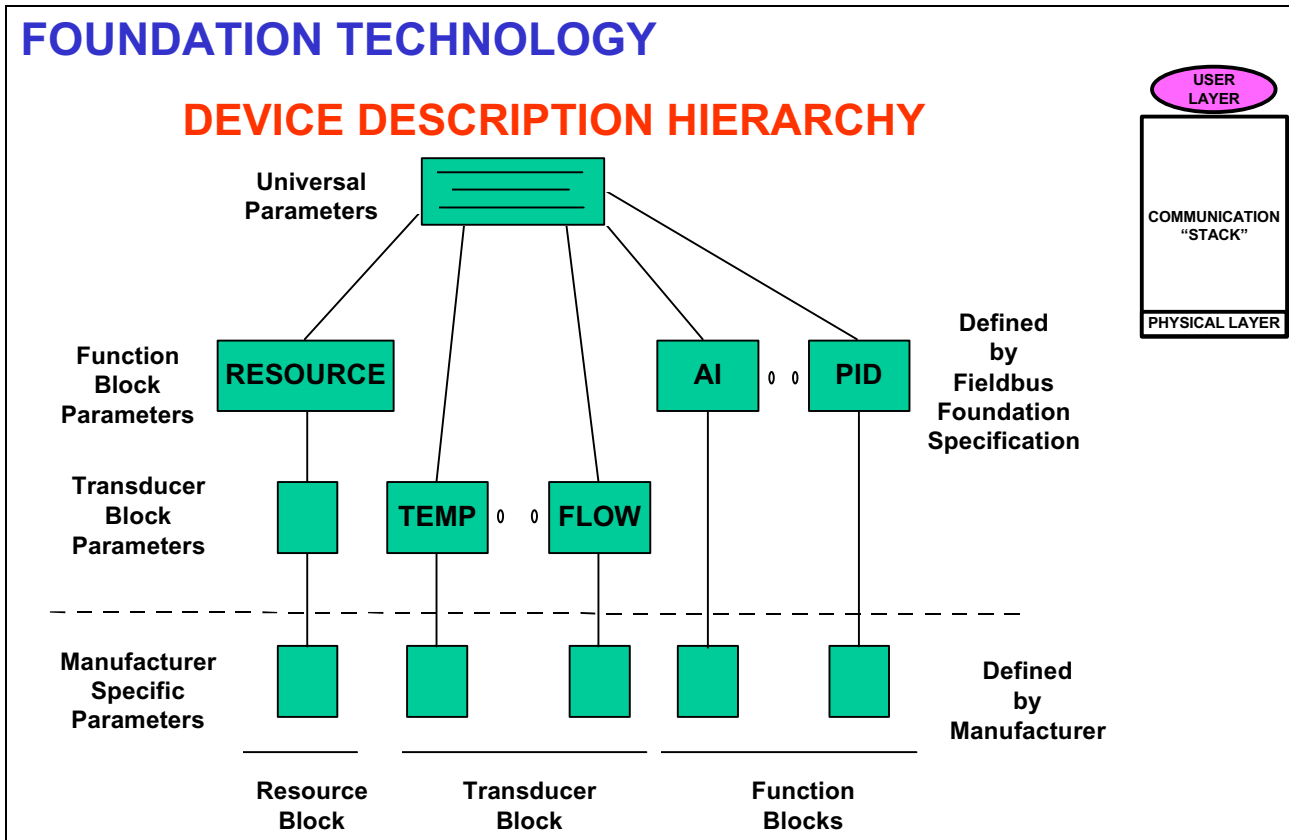


Figura 3.10 - Hierarquia do Device Description

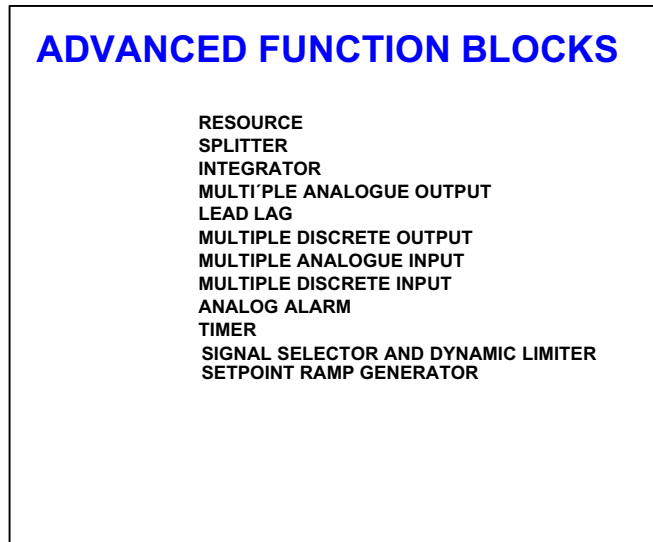


Figura 3.11 - Blocos de função avançados

### Desempenho comparativo entre o loop de controle analógico e de controle FIELDBUS

Através das duas figuras a seguir, pode-se levantar uma comparação de desempenho entre o sistema de controle tradicional e o FIELDBUS; No controle com instrumentação operando em corrente (padrão 4-20 mA), há a necessidade de duas conversões A/D e duas D/A para que o sinal possa ser transmitido do elemento sensor até o atuador (Figura 3.12- o sinal caminha pela rede de forma analógica) com o que obtem-se um atraso (tempo morto) muitas vezes bastante significativo.

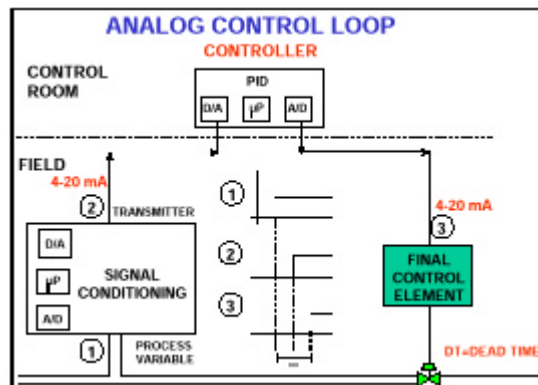


Figura 3.12 - Malha com controle analógico

Na instrumentação FIELDBUS este tipo de problema é minimizado pois tem-se apenas uma conversão A/D e uma D/A minimizando também o erro decorrente às conversões (Figura 3.13). Neste sistema, o transmissor converte o sinal do elemento sensor para a forma digital e o transmite diretamente ao atuador nesta mesma forma (digital)

onde ocorre uma conversão para um sinal analógico de acionamento do atuador. Neste caso, o tempo morto é reduzido possibilitando níveis de controle mais refinados.

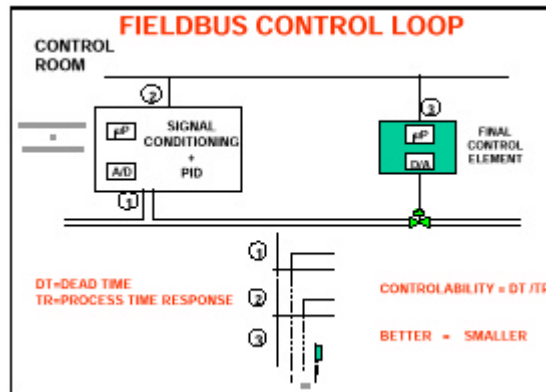


Figura 3.13 - Malha com controle FIELD BUS



### 3.3. Definições foundation fieldbus

Existem algumas definições da Fieldbus Foundation que temos que conhecer, em seguida temos as principais:

#### 3.3.1 Parâmetros dos Blocos.

Existem 3 parâmetros de blocos:

- **Dinâmicos** : ele é alterado a cada ciclo do bloco;
- **Não-Volátil** : ele é calculado pelo algoritmo;
- **Estáticos** : ele é alterado somente quando ocorrer mudanças no valor.

Ver no manual de Function Blocks a partir da página 80 as tabelas dos diversos parâmetros, e seus respectivos tipos na coluna **Store**, **D** (Dinâmicos), **N** (Não-Voláteis) e **S** (Estáticos).

#### 3.3.2 Tipos de parâmetros.

Existem 3 tipos de parâmetros:

- **Entrada** : fornece o valor e o status;
- **Saída** : fornece o valor e o status;
- **Contained** : são parâmetros internos.

#### 3.3.3 Mode BLK .

Existem 5 tipos de modos , sendo que somente 4 deles serão mostrados pelo Syscon :

1. **TARGET** - Este modo é ajustado pelo operador, entre um dos modos permitidos pelo “MODE\_BLK.Permitted”.

2. **ACTUAL** - Indica o modo que o bloco está sendo executado, podendo ser diferente do modo Target devido às condições de processo. O valor deste modo é calculado como parte de execução do bloco.

3. **PERMITTED** - Define os modos que são permitidos para determinado bloco. É configurado baseado na aplicação.

4. **NORMAL** - Este modo deve ser usado durante as condições normais de operação. Este parâmetro pode ser configurado e lido pôr uma Interface, mas não é usado no algoritmo do bloco.



**5. SUPPORTED** - Todos os modos que o tipo de função suporta. Não é visualizado pelo Syscon. O Permitted é um sub-conjunto deste modo.

**TABELA DE PRIORIDADES.**

Os modos obedecem a uma prioridade que é definida pela tabela a seguir.

O conceito de prioridade é seguido quando o algoritmo deve ajustar um modo com valor diferente do definido no modo TARGET.

MODE	DESCRIPTION	PRIORITY	ACTUAL	TARGET
ROut	Remote Output	0 - lowest	1	09,13
Rcas	Remote Cascade	1	2	10,14
Cas	Cascade	2	4	12
Auto	Automatic	3	8	8
Man	Manual	4	16	16
LO	Local Override	5	32	--
IMan	Initialization Manual	6	64	--
O/S	Out of Service	7 - hidhest	128	128

**NOTA :** IMAN e LO não podem ser ajustados pelo operador como MODE\_BLK.Target.

Modo tipo	Fonte do SP	Fonte do OUT
O/S	Usuário	Usuário
Iman	Usuário	Outro bloco de função – seguindo parâmetro BKCAL_IN
LO	<ul style="list-style-type: none"> <li>PID / EPID / APID : Usuário</li> <li>AO / DO : Estado de falha (último valor ou FSTATE_VAL)</li> </ul>	<ul style="list-style-type: none"> <li>PID / EPID / APID : Outro bloco de função – seguindo parâmetro TRK_VAL</li> <li>AO / DO : Estado de falha (último valor ou FSTATE_VAL)</li> </ul>
Man	Usuário	Usuário
Auto	Usuário	Bloco algoritmo
Cas	Outro bloco de função – Seguindo parâmetro CAS_IN	Bloco algoritmo
Rcas	Controle de aplicação rodando no dispositivo interface	Bloco algoritmo
Rout	Bloco mantém último valor	Controle de aplicação rodando no dispositivo interface

**MUDANÇA DE MODOS.**

<b>MODO O/S</b>	<ol style="list-style-type: none"> <li>TARGET = O/S</li> <li>RS_STATE = STANDBY</li> </ol>																																																												
<b>MODO IMAN</b>	<ol style="list-style-type: none"> <li>BKCAL_IN_STATUS = GOOD.FSA</li> <li>BKCAL_IN_STATUS = GOOD.NI</li> <li>BKCAL_IN_STATUS = GOOD.IR</li> <li>BKCAL_IN_STATUS = GOOD.LO</li> <li>BKCAL_IN_STATUS = BAD.X</li> <li>BKCAL_IN_STATUS = UNCERTAIN E UNCERTAIN AS BAD</li> </ol>																																																												
<b>MODE LO</b>	<ol style="list-style-type: none"> <li>PID – CONTROL_OPTS.TRK ENABLE = TRUE</li> <li>PID – TRK_IN_D = TRUE/GOOD</li> <li>PID – TRK_VAL = GOOD</li> <li>PID – CONTROL_OPTS.TRK IN MANUAL = FALSE</li> </ol> <div style="margin-left: 400px;"> <table style="border: none;"> <tr> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="font-size: 3em; vertical-align: middle;">{</td> <td style="font-size: 3em; vertical-align: middle;">{</td> </tr> <tr> <td></td> <td>TARGET =</td> <td>O/S</td> <td>IMAN</td> <td>ACTUAL</td> <td>MANTÉM</td> </tr> <tr> <td></td> <td>TARGET =</td> <td>AUTO</td> <td>CAS</td> <td>LO e aceita</td> <td>mudança</td> </tr> <tr> <td></td> <td></td> <td>RCAS</td> <td>ROUT</td> <td>do target</td> <td>p/ man</td> </tr> <tr> <td></td> <td></td> <td>MAN</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>TRUE</td> <td>ACTUAL =</td> <td>MAN</td> <td>LO e não</td> <td>aceita</td> </tr> <tr> <td></td> <td></td> <td></td> <td>AUTO</td> <td>mudança</td> <td>p/ man</td> </tr> <tr> <td></td> <td></td> <td></td> <td>CAS</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>RCAS</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>RQUT</td> <td></td> <td></td> </tr> </table> </div>	{	{	{	{	{	{		TARGET =	O/S	IMAN	ACTUAL	MANTÉM		TARGET =	AUTO	CAS	LO e aceita	mudança			RCAS	ROUT	do target	p/ man			MAN					TRUE	ACTUAL =	MAN	LO e não	aceita				AUTO	mudança	p/ man				CAS						RCAS						RQUT		
{	{	{	{	{	{																																																								
	TARGET =	O/S	IMAN	ACTUAL	MANTÉM																																																								
	TARGET =	AUTO	CAS	LO e aceita	mudança																																																								
		RCAS	ROUT	do target	p/ man																																																								
		MAN																																																											
	TRUE	ACTUAL =	MAN	LO e não	aceita																																																								
			AUTO	mudança	p/ man																																																								
			CAS																																																										
			RCAS																																																										
			RQUT																																																										
	<ol style="list-style-type: none"> <li>AO QUANDO RECEBE IFS NO CAS_IN</li> </ol>																																																												
<b>MODO MAN</b>	<ol style="list-style-type: none"> <li>IN.STATUS { BAD UNCERTAIN E STATUS_OPTS UNCERTAIN AS GOOD = FALSE</li> <li>TARGET { RCAS - IN.STATUS = BAD ROUT - IN.STATUS = BAD</li> <li>SHED_OPT { SHED TO MAN / NORMAL RET – TARGET MANTÉM, MAS ACTUAL VAI P/ MAN SHED TO MAN / NO RET – TARGET / ACTUAL = MAN</li> </ol>																																																												
<b>MODO AUTO</b>	<ol style="list-style-type: none"> <li>CAS_IN – STATUS = BAD E TARGET = CAS OU UNCERTAIN COMO BAD</li> <li>TARGET = CAS</li> <li>RCAS_IN.STATUS = BAD OU UNCERTAIN COMO BAD</li> <li>SHED_OPTS = SHED TO AUTO / NORMAL RET SHED TO AUTO / NO RET</li> </ol>																																																												

<b>MODO CAS</b>	<ol style="list-style-type: none"> <li>1. TARGET = RCAS</li> <li>2. RCAS_IN.STATUS = BAD OU UNCERTAIN COMO BAD</li> <li>3. SHED_OPTS = SHED TO AUTO / NORMAL RET SHED TO AUTO / NO RET</li> <li>4. CAS_IN.STATUS = GOOD</li> </ol>
<b>MODO RCAS</b>	<ol style="list-style-type: none"> <li>1. TARGET = RCAS</li> <li>2. RCAS_IN.STATUS = GOOD</li> </ol>
<b>MODO ROUT</b>	<ol style="list-style-type: none"> <li>1. TARGET = ROUT</li> <li>2. ROUT_IN.STATUS = GOOD</li> </ol>

### 3.3.4 STATUS

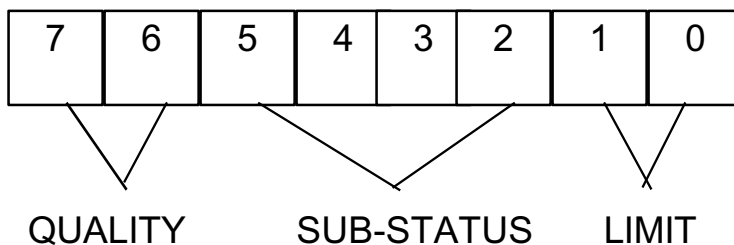
***ESTRUTURA DOS PARÂMETROS É COMPOSTO POR :***

- VALUE ( 4 BYTE)
- STATUS ( 1 BYTE) - Informação sobre a qualidade do valor medido.

***O STATUS pode ser transmitido de :***

- Um bloco para outro.
- Para o HMI( Human Machine Interface).
- Para Histórico.

#### **MODELO DE STATUS.**



## STATUS FORWARD

As seguintes condições são

- 1 - *Limited Higher* - Não pode gerar valor mais alto pois existe
- 2 - *Limited Lower* - Não pode gerar valor mais baixo pois existe
- 3 - Valor não pode mudar, pois está constante
- 4 - O valor vem de um bloco que está em falha.
- 5 - Falha na comunicação
- 6 - Valor UNCERTAIN

## STATUS BACKWARD

As seguintes condições são

- 1 - *Limited Higher* - Não pode gerar valor mais alto pois existe limite.
- 2 - *Limited Lower* - Não pode gerar valor mais baixo pois existe limite
- 3 - Valor não pode mudar, pois está
- 4 - O valor vem de um bloco que está em
- 5 - O valor é de um bloco cuja saída que foi localmente ajustado para Local ou Fail Safe ou pôr Interlock Logic.
- 6 - Falha na comunicação
- 7 - Valor de um Control Selector cuja saída correspondente não foi selecionada

## DEFINIÇÃO DO STATUS

- A definição do STATUS ATTRIBUTES é a mesma para todos os parâmetros (INPUT, OUTPUT and CONTAINED).
- Existem 4 STATUS de QUALITY sendo que para cada um são definidos 16 Sub-Status.

- 1 - BAD** - O valor não deve ser usado
- 2 - UNCERTAIN** - A qualidade do valor é menor que o normal, mais ainda pode ser usado.
- 3 - Good(Non Cascade)** - A qualidade do valor é boa, pode haver, indicação de alarmes através do sub-status.
- 4 - Good(Cascade)** - O valor pode ser usado em controle.

Ver na tabela da norma os Sub-Status.

### **3.3.5 PARÂMETROS DE OPÇÃO.**

#### **3.3.5.1 STATUS\_OPTS**

Ver página 103 do manual de Function Blocks.

#### **3.3.5.2 IO\_OPTS**

Ver página 101 do manual de Function Blocks.

#### **3.3.5.3 CONTROL\_OPTS**

Ver página 102 do manual de Function Blocks.

### 3.4. Configuração dos dispositivos fieldbus

Neste item estaremos apresentando os passos necessários para a configuração de uma rede Fieldbus, utilizando o configurador Syscon.

Para um melhor entendimento deste estudo, sugerimos que se faça antes a leitura do item 2.3.8. onde são abordados as características de cada um dos instrumentos que serão utilizados para exemplificação nesta seção.

Na figura abaixo vemos uma tela típica deste configurador onde são apresentadas duas janelas: uma com as configurações dos instrumentos e outra com as conexões entre os blocos funcionais.

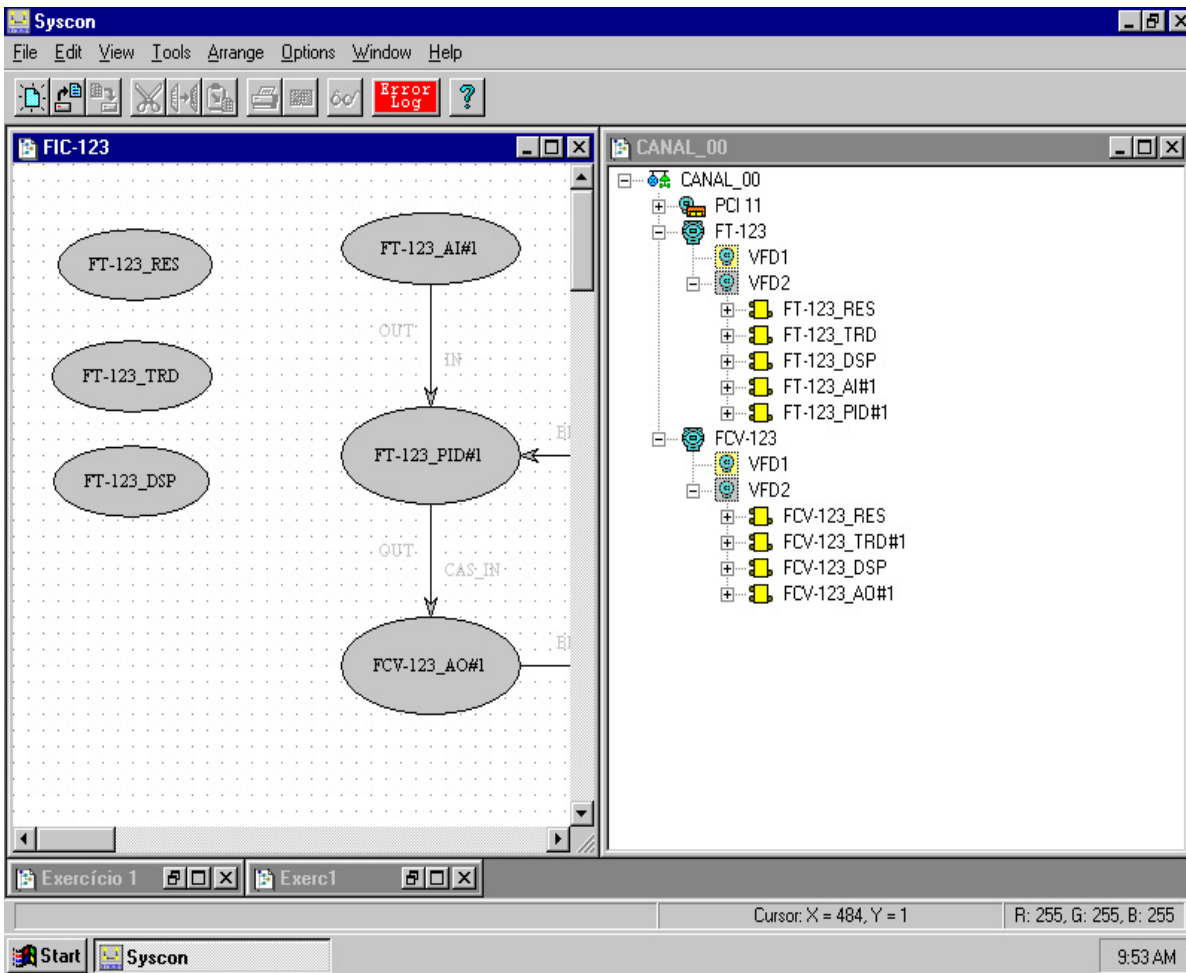


Figura 3.14 - Tela típica do configurador SYSCON

### 3.4.1. Escalonamento de Blocos de Função

O Trabalho de configuração dos instrumentos Fieldbus, consiste dentre outras coisas, na parametrização dos blocos funcionais que serão utilizados pelos instrumentos.

Em especial, todo bloco funcional que possui uma entrada para algum tipo de variável (seja de um elemento sensor ou simplesmente uma variável de saída de outro bloco), possui dois parâmetros que podem causar alguma confusão. Trata-se dos parâmetros de escalonamento da variável de entrada (PV\_SCALE) e de saída (OUT\_SCALE) presente no bloco. Estes parâmetros fazem a conversão de valores de entrada ou saída para valores em unidades de engenharia.

Os exercícios abaixo ajudam a fixar estes conceitos; para cada figura apresentada abaixo, preencha os campos com as informações solicitadas.

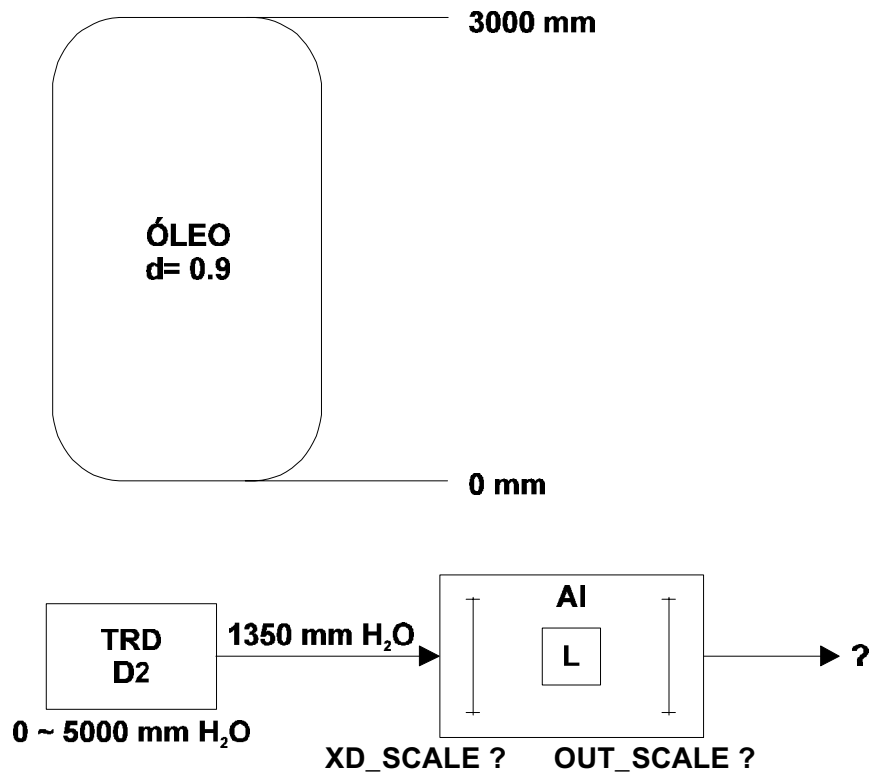
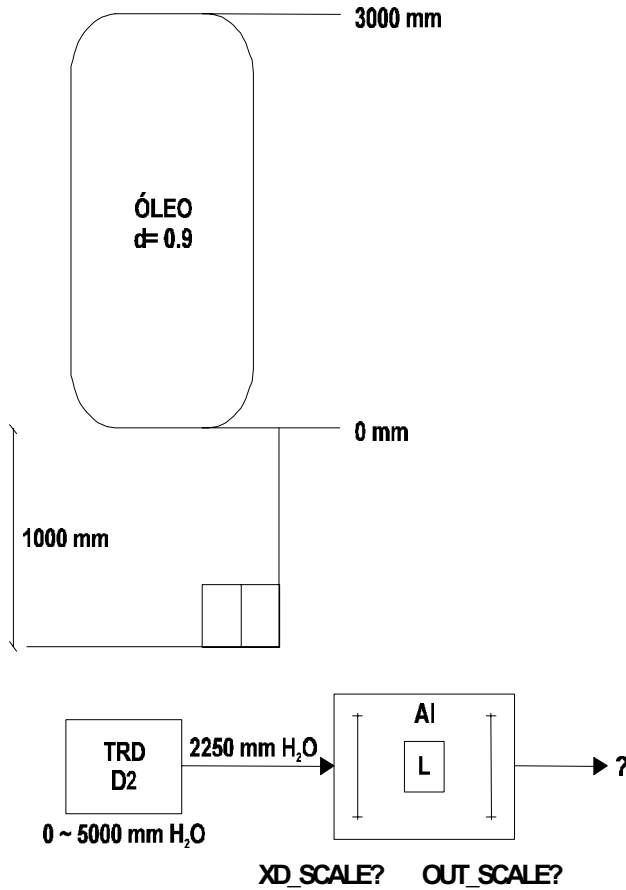


Figura 3.15 - Exercício 1

<b>R:</b>	<b>XD_SCALE:</b> 0% = 0	<b>OUT_SCALE:</b> 0% = 0
	100% = 2700	100% = 3000
	U.Eng = mmH <sub>2</sub> O	U.Eng = mm



### Expressões

IN<sub>0</sub>  
IN<sub>100</sub> > ESCALAS DE ENTRADA

IN<sub>0</sub>  
IN<sub>100</sub> > ESCALAS DE SAÍDAS

$$PV\% = \frac{IN_{(IN_0)}}{(IN_{100}) - (IN_0)} \times 100, \text{ ONDE IN = ENTRADA}$$

$$OUT = \frac{PV\%}{100} [(OUT_{100}) - (OUT_0)] + (OUT_0)$$

MTRE\_19

Figura 3.16 - Exercício 2



**R:** XD\_SCALE: 0% = 900  
 100% = 3600  
 U.Eng = mmH<sub>2</sub>O

OUT\_SCALE: 0% = 0  
 100% = 3000  
 U.Eng = mm

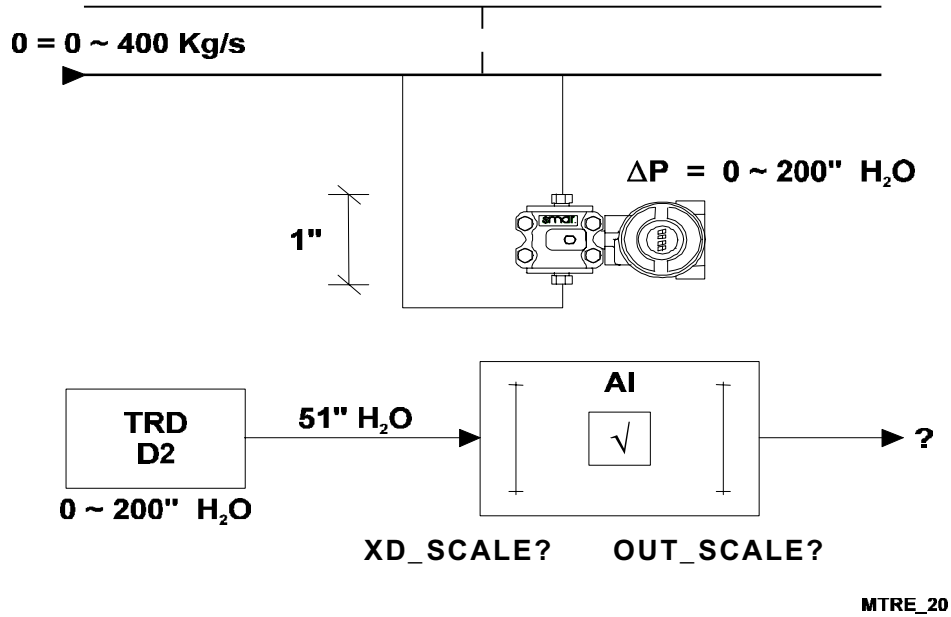


Figura 3.17 - Exercício 3

**R:** XD\_SCALE: 0% = 0  
 100% = 200  
 U.Eng = "H<sub>2</sub>O

OUT\_SCALE: 0% = 0  
 100% = 400  
 U.Eng = Kg/s

### 3.4.2. Configuração da rede Fieldbus - Exercício 1

Efeturemos o passo a passo para configuração de um primeiro exercício, que terá a função de modelo para os demais exercícios.

Para os exercícios apresentados a seguir configure, a partir dos diagramas de processo convencionais as soluções de automação em Fieldbus Foundation, utilizando como recurso gráfico o software Syscon.

#### Exercício 1

Montar uma configuração FIELDBUS que realize a lógica de controle conforme indicado na Figura 3.18 (controle PID simples de vazão).

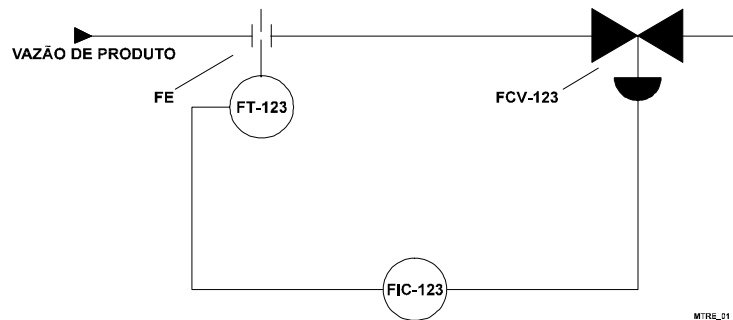


Figura 3.18 - Malha de controle proposta para o exercício 1

Para este exercício utilizaremos apenas dois instrumentos FIELDBUS: um transmissor de pressão diferencial (LD302) que irá ler a vazão da tubulação, e um conversor de sinal fieldbus para corrente (IF302) cujo sinal de saída (4 a 20mA) irá alimentar um atuador de campo (uma válvula) que fará o controle da vazão.

Para implementar este controle, utilizaremos a estratégia da Figura 3.24.

Na figura abaixo temos a representação das interligações/topologia utilizada para este exercício.

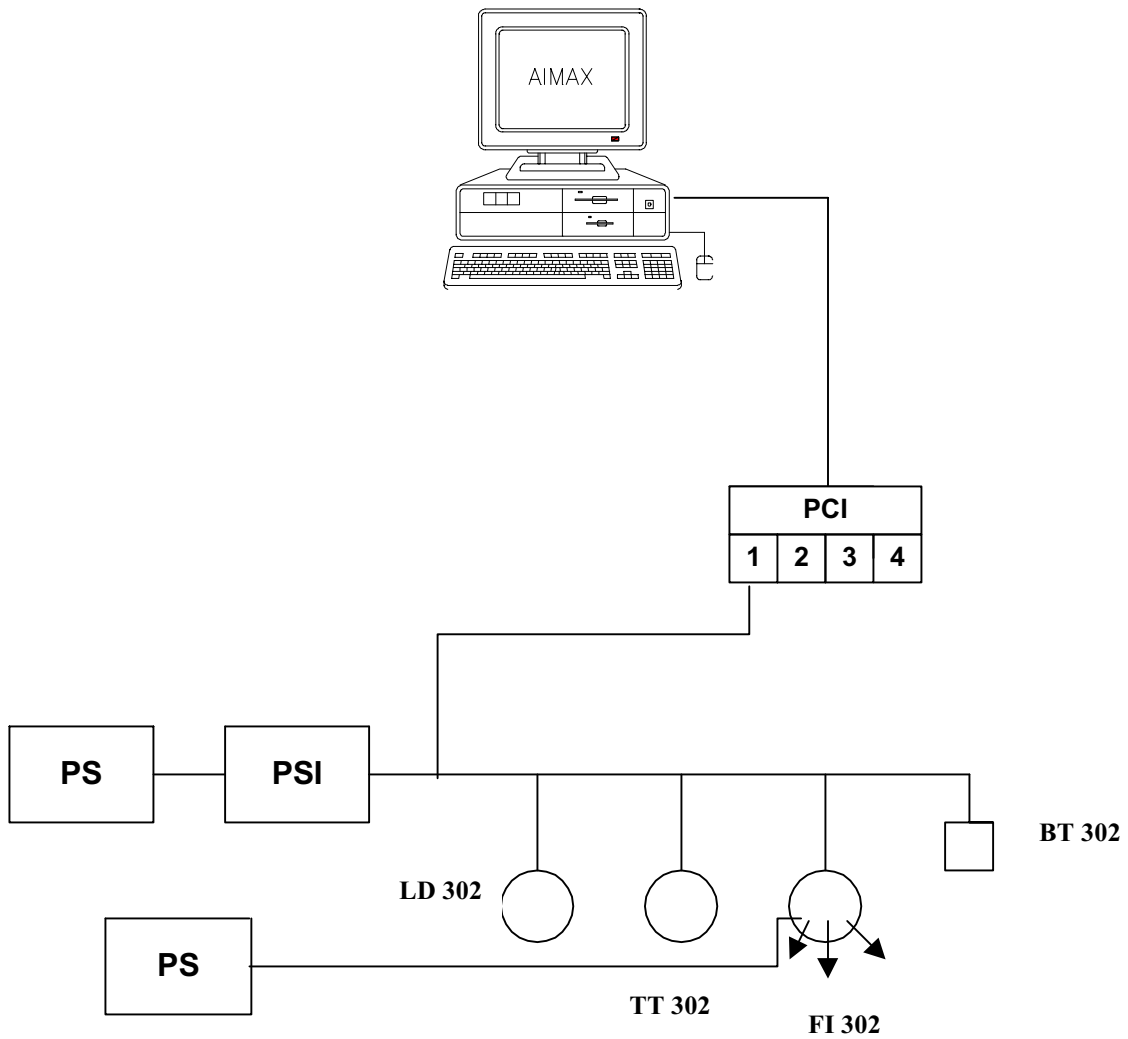
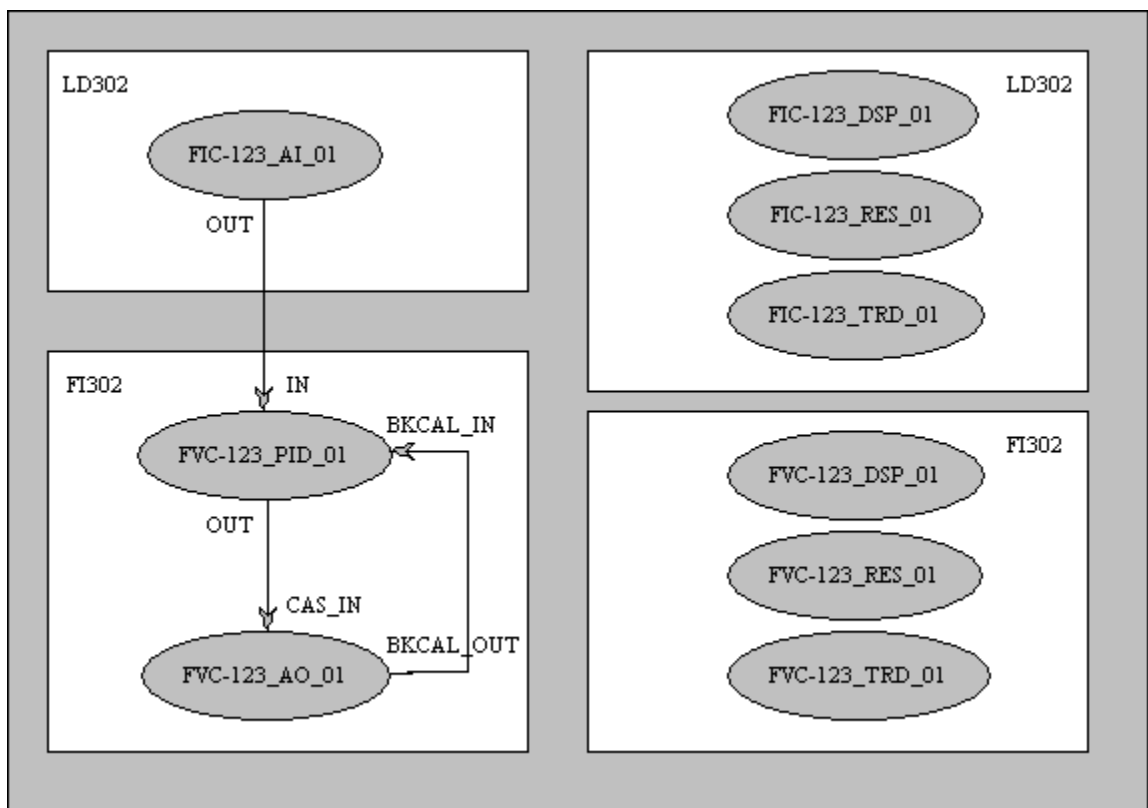


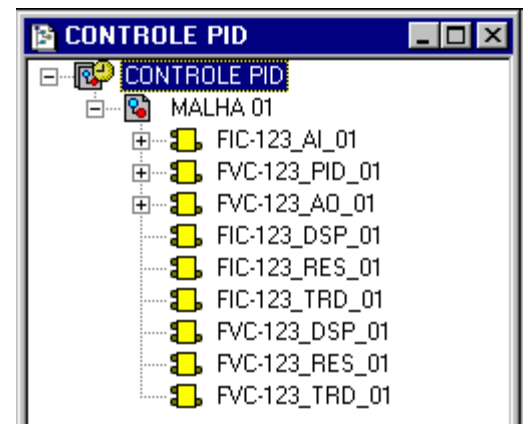
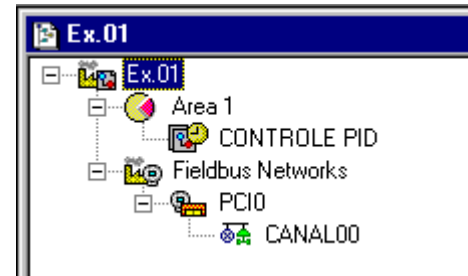
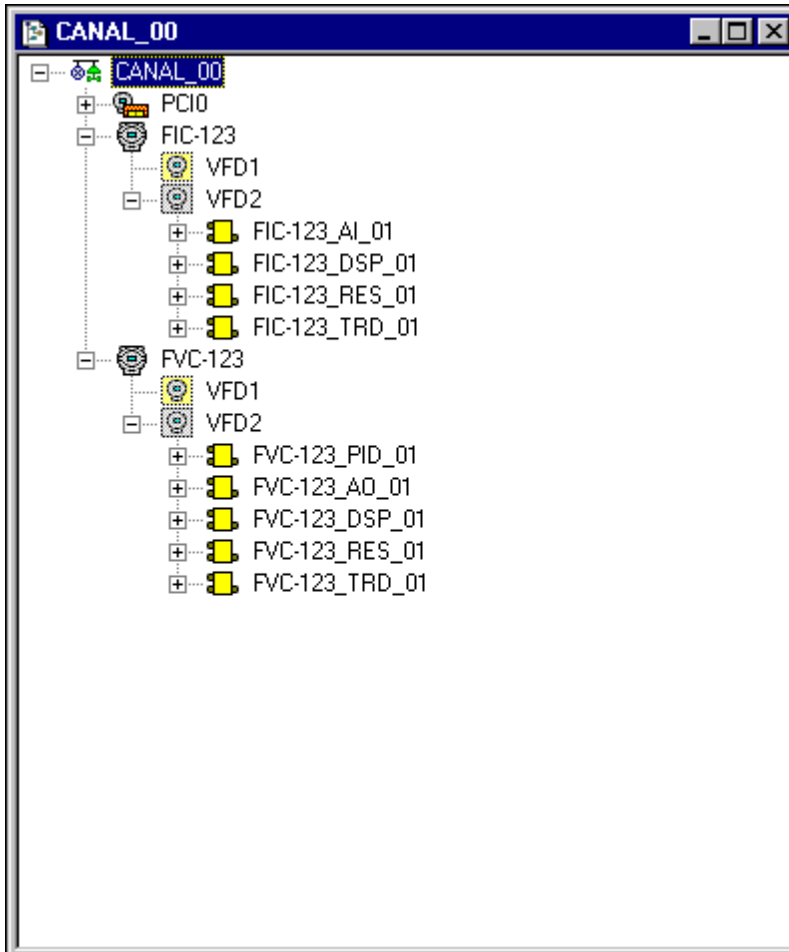
Figura 3.19 – Exercício 1 – Topologia

a) Arquivo de conexões:

Para este controle, utilizaremos a estratégia de controle sugerida abaixo:



## CONFIGURAÇÃO



PARAMETRIZAÇÃO

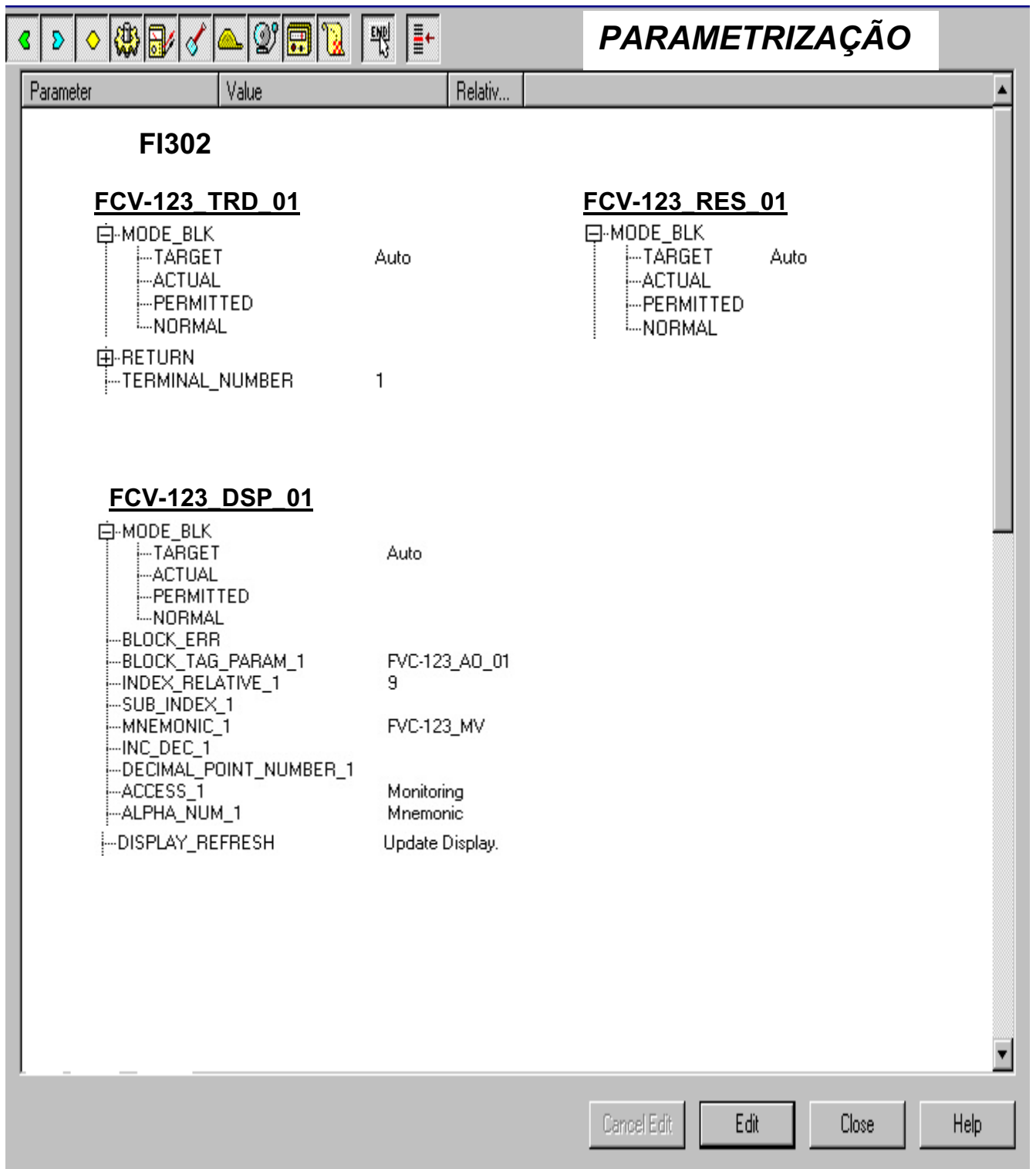
Parameter	Value	Relativ...
<b>LD302</b>		
<b><u>FIC-123 TRD 01</u></b>		
[-] MODE_BLK		
-- TARGET	Auto	
-- ACTUAL		
-- PERMITTED		
-- NORMAL		
<b><u>FIC-123 RES 01</u></b>		
[-] MODE_BLK		
-- TARGET	Auto	
-- ACTUAL		
-- PERMITTED		
-- NORMAL		
<b><u>FIC-123 DSP 01</u></b>		
[-] MODE_BLK		
-- TARGET	Auto	
-- ACTUAL		
-- PERMITTED		
-- NORMAL		
-- BLOCK_ERR		
-- BLOCK_TAG_PARAM_1	FIC-123_AI_01	
-- INDEX_RELATIVE_1	8	
-- SUB_INDEX_1		
-- MNEMONIC_1	FIC-123_PV	
-- INC_DEC_1		
-- DECIMAL_POINT_NUMBER_1		
-- ACCESS_1	Monitoring	
-- ALPHA_NUM_1	Mnemonic	
-- DISPLAY_REFRESH	Update Display.	
<b><u>FIC-123 AI 01</u></b>		
[-] MODE_BLK		
-- TARGET	Auto	
-- ACTUAL		
-- PERMITTED		
-- NORMAL		
-- BLOCK_ERR		
[-] PV		
[-] OUT		
[-] SIMULATE		
[-] XD_SCALE		
-- EU_100	200	
-- EU_0	0	
-- UNITS_INDEX	inH2O (4°C)	
-- DECIMAL		
[-] OUT_SCALE		
-- EU_100	150	
-- EU_0	0	
-- UNITS_INDEX	m³/h	
-- DECIMAL		
[-] GRANT_DENY		
-- ID_OPTS		
-- STATUS_OPTS	Propagate Fail Fwd	
-- CHANNEL	1	
-- L_TYPE	Indirect Sq Root	

Cancel Edit Edit Close Help

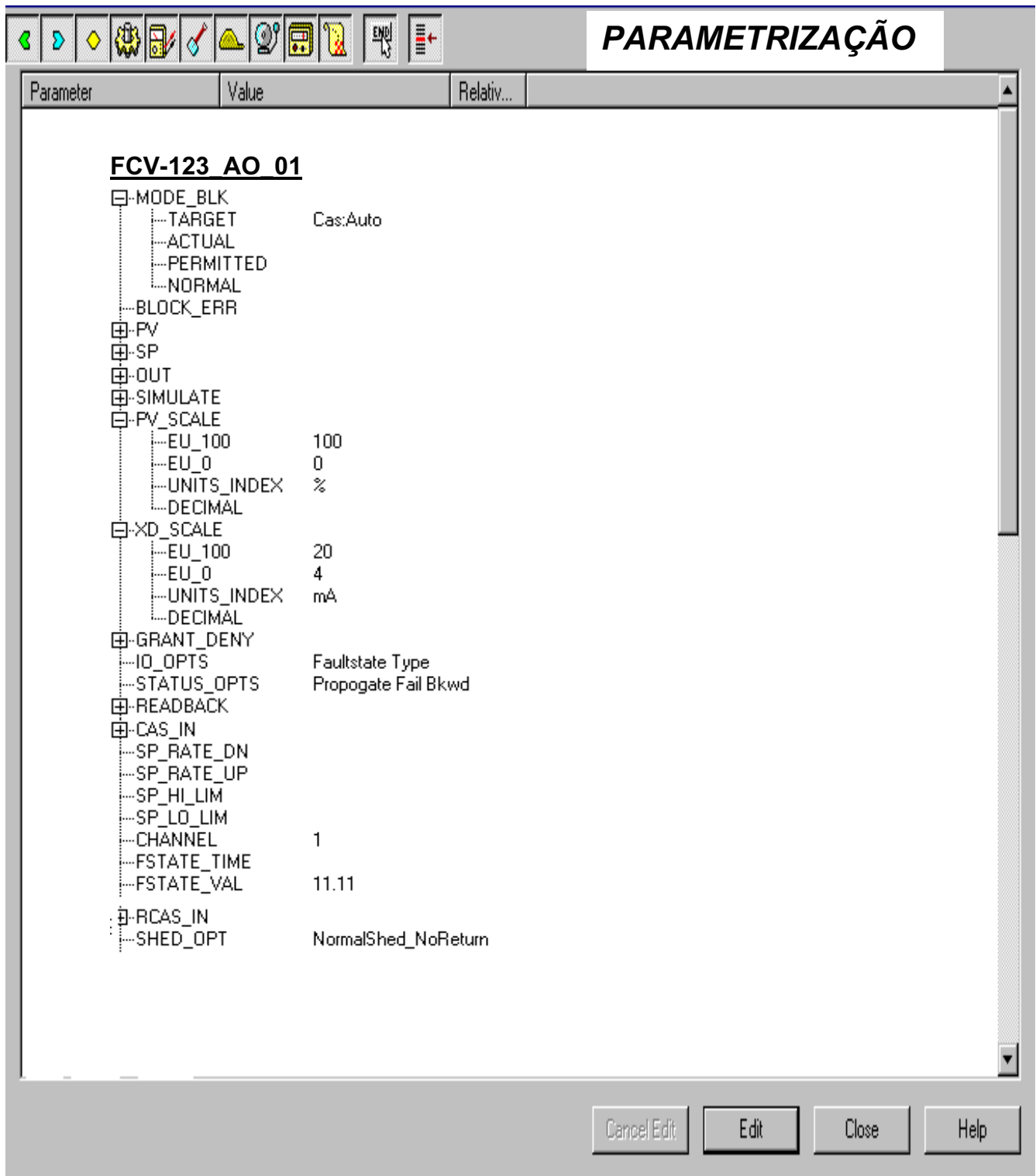
PARAMETRIZAÇÃO

Parameter	Value	Relativ...
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><b>FI302</b></p> <p><b><u>FVC-123 PID 01</u></b></p> </div> <div style="width: 70%;"> <ul style="list-style-type: none"> <li>[-] MODE_BLK                             <ul style="list-style-type: none"> <li>--- TARGET      Auto</li> <li>--- ACTUAL</li> <li>--- PERMITTED</li> <li>--- NORMAL</li> </ul> </li> <li>--- BLOCK_ERR</li> <li>[+] PV</li> <li>[+] SP</li> <li>[+] OUT</li> <li>[-] PV_SCALE                             <ul style="list-style-type: none"> <li>--- EU_100      150</li> <li>--- EU_0        0</li> <li>--- UNITS_INDEX      m<sup>2</sup>/h</li> <li>--- DECIMAL</li> </ul> </li> <li>[-] OUT_SCALE                             <ul style="list-style-type: none"> <li>--- EU_100      100</li> <li>--- EU_0        0</li> <li>--- UNITS_INDEX      %</li> <li>--- DECIMAL</li> </ul> </li> <li>[+] GRANT_DENY</li> <li>--- CONTROL_OPTS      Track enable</li> <li>--- STATUS_OPTS      IFS if Bad IN:Uncertain as Good</li> <li>[+] IN                             <ul style="list-style-type: none"> <li>--- PV_FTIME</li> <li>--- BYPASS          Off</li> </ul> </li> <li>[+] CAS_IN                             <ul style="list-style-type: none"> <li>--- SP_RATE_DN</li> <li>--- SP_RATE_UP</li> <li>--- SP_HI_LIM      150</li> <li>--- SP_LO_LIM</li> <li>--- GAIN            1</li> </ul> </li> <li>--- RESET            1</li> <li>--- BAL_TIME</li> <li>--- RATE            0</li> <li>[+] BKCAL_IN                             <ul style="list-style-type: none"> <li>--- OUT_HI_LIM</li> <li>--- OUT_LO_LIM</li> <li>--- BKCAL_HYS</li> </ul> </li> <li>[+] BKCAL_OUT</li> <li>[+] RCAS_IN</li> <li>[+] ROUT_IN</li> <li>--- SHED_OPT          NormalShed_NoReturn</li> </ul> </div> </div>		

Cancel Edit Edit Close Help





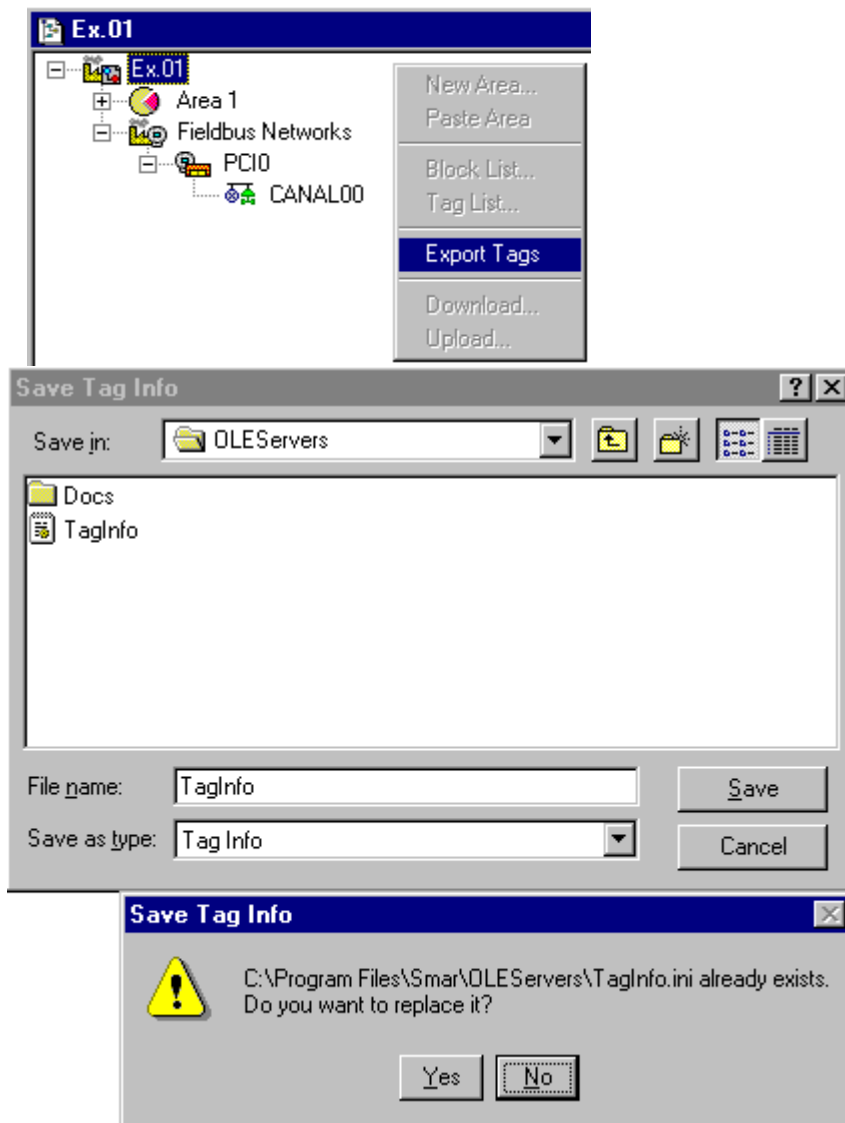


## **1. EXECUTAR O “DEFAULT” NOS TRANSMISSORES**

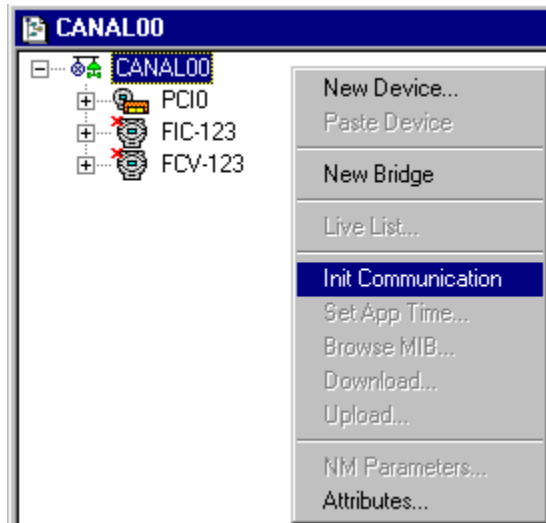
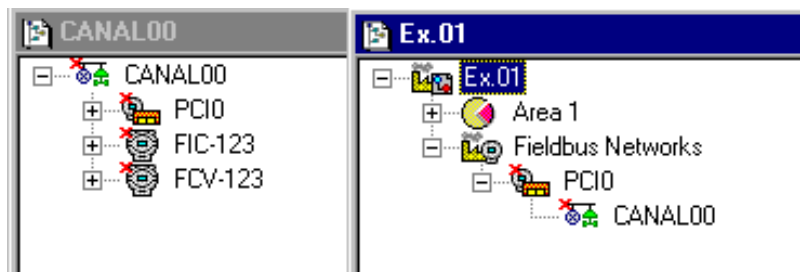
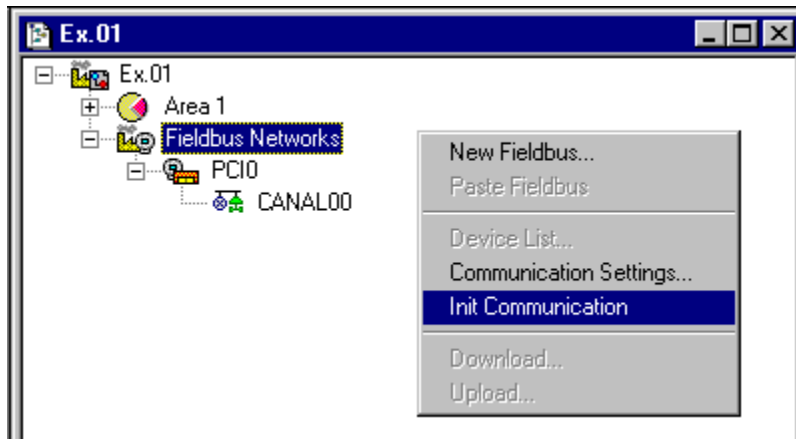
Inserir as duas chaves magnéticas, uma no orifício “S” e a outra no “Z”. Desligar o transmissor da fonte, e religar novamente, visualizando o display até que apareça “Fact”. Em seguida, retire as chaves magnéticas e aguarde o salvamento(visualizando o display até que desapareça o “5”).

Repetir a operação para os demais transmissores.

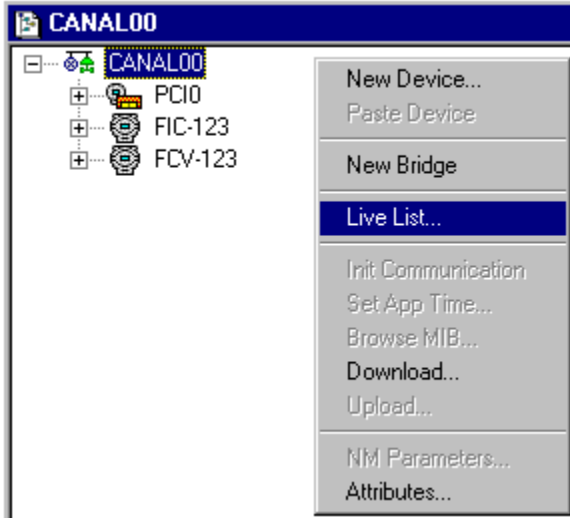
## **1.EXPORT TAGS**



## 2. EXECUTAR O “INIT COMMUNICATION”



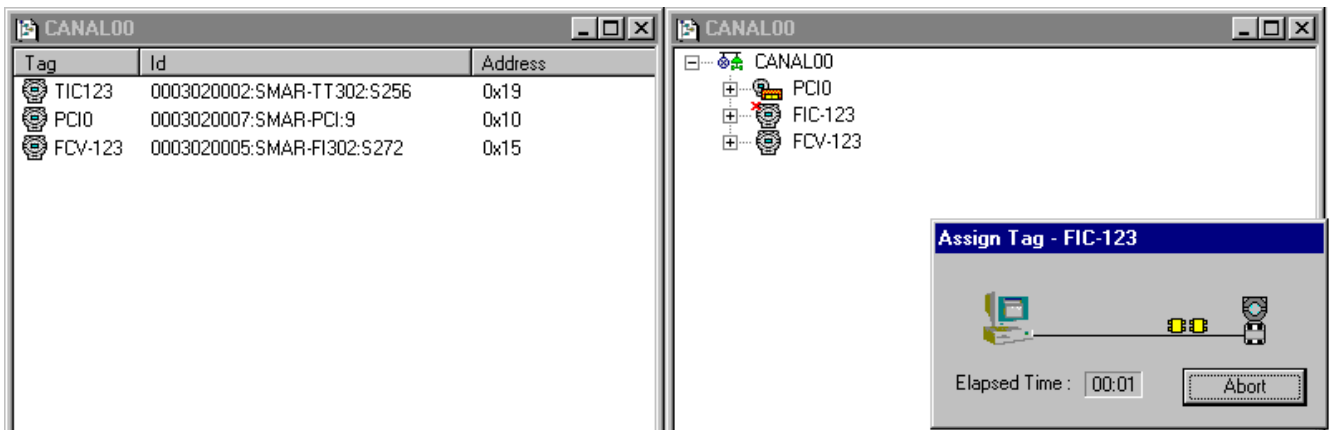
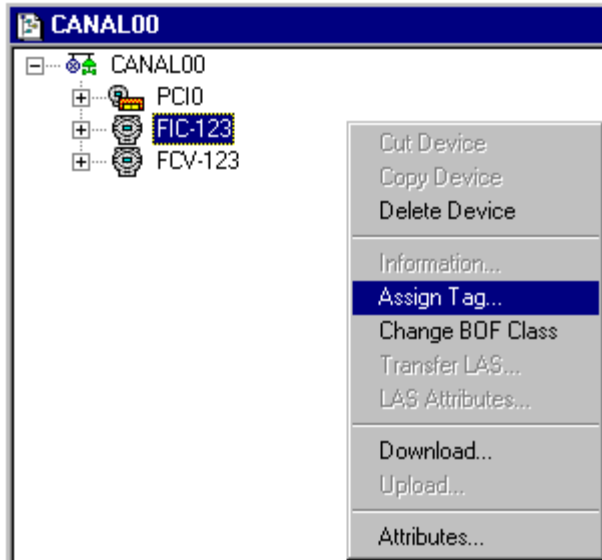
### 3. ABRIR A “LIVE LIST”



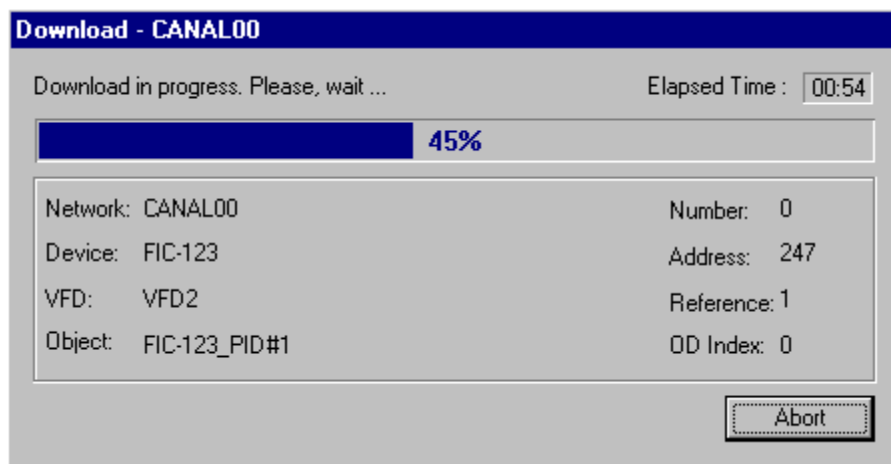
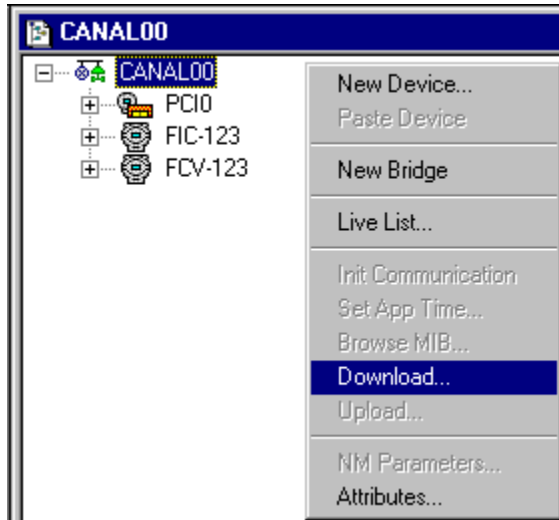
The screenshot shows the 'Live List' window in the CANAL00 software. It contains a table with three columns: 'Tag', 'Id', and 'Address'. The table lists four devices:

Tag	Id	Address
FV124	0003020005:SMAR-FI302:S272	0x15
TIC123	0003020002:SMAR-TT302:S256	0x19
FIC124	0003020001:SMAR-LD302:S180	0xF7
PCI0	0003020007:SMAR-PCI:9	0x10

#### 4. PROCEDER O “ASSIGN TAG”



## 5. EM SEGUIDA , EXECUTAR O “DOWNLOAD”



Após finalizar o “Download”, ir em cada um dos blocos dos TRM’s, abrir “On – Line Characterization” e verificar as condições de operação da malha de controle.

### 3.4.3 Exercícios de configuração

#### Exercício 2

No exercício 1 realizamos um controle simples de vazão. Utilize o mesmo exercício como referência e incremente a função de totalização da vazão disponibilizando-a no display do transmissor de pressão.

#### Exercício 3

Neste exercício mostraremos a independência de malhas de controle distintas que podem ser configuradas em uma mesma configuração FIELDBUS. Utilizando um transmissor de temperatura (TT302) ligado à mesma configuração dos exercícios anteriores, faça a indicação de um ponto de temperatura proveniente de um sensor tipo Pt-100.

#### Exercício 4

Neste exercício mostraremos como a indicação de uma determinada variável pode ser efetuada em qualquer instrumento da rede FIELDBUS. Utilizando a configuração dos exercícios anteriores, efetue também a indicação da temperatura no transmissor de pressão (LD302).

#### Exercício 5

Neste exercício mostraremos o recurso de *mestre-backup* onde um instrumento fica “responsável” pelo gerenciamento da comunicação caso a interface controladora PCI seja desconectada da rede FIELDBUS. Configure o transmissor de temperatura para que ele seja o *mestre-backup* do canal.

Para os diagramas e os esquemas das redes FIELDBUS a seguir, crie as configurações utilizando o configurador Syscon e as informações obtidas no exercício 1. Após a configuração, proceda também com a parametrização dos blocos, conforme lista ao final de cada exercício.

**Observação:** Na seqüência temos exercícios mais complexos cuja solução completa se encontra no manual do SYSCON incluído no anexo “A” desta apostila.

### 3.5. Programação do Controlador Lógico Programável

#### 3.5.1. Uso do Configurador

##### The Configurator

The user prepares the plant strategy using the classic LADDER logic network, combining Relays, Coils, User-Functions and Built-in-Functions.

User-Functions is a Boolean function prepared by the user that can take part of the Ladder logic network. It is edited using our exclusive Click-and-Write technique. With the Click-and-Write the user can write a full Boolean equation using all of the resources of the PLC without touching the keyboard!

The user can create as many Ladder Networks as the memory space allows and can have the execution order changed at any time.

Each Network is composed by a set of 15X16 cells. The cells can represent lines, Relays, Coils, User-Functions, Built-in-Functions, Jumpers and/or Returns.

##### The Conf-700 Menu

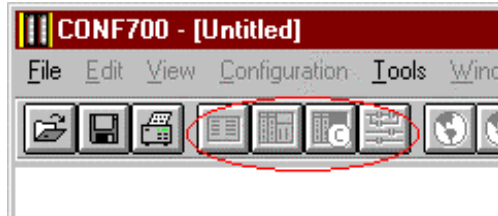




## Basic Steps

The user needs basically to navigate through the “Configuration Pages”. Configuration Pages are selected using dedicated buttons in the button bar or using the pull down menu under Configure.

Buttons to access the Pages:



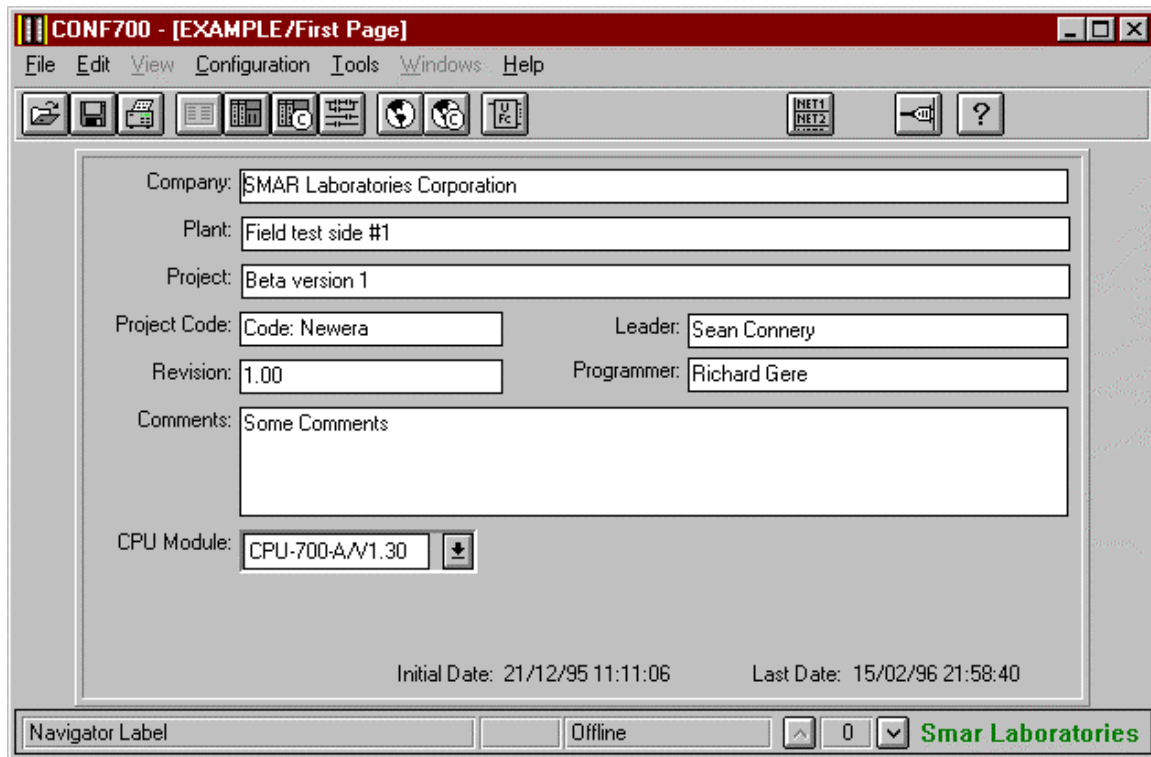
The program has four “Configuration Pages”;

- Information Page
- Hardware Definition Page
- Virtual Variables Page
- Ladder Logic Page

## Information Page

When you Open or ask for a New configuration the program shows the first “Page” of a PLC configuration. This “Page” is called the Information Page.

The Information Page expects the user to enter all the text information on the particular project he is working on as well as selecting the CPU Version. Once he selects the CPU version the software automatically select the compatible set of Modules, Functions and related features to work with.



### 3.5.1.1. Definição do hardware

#### Hardware Definition Page

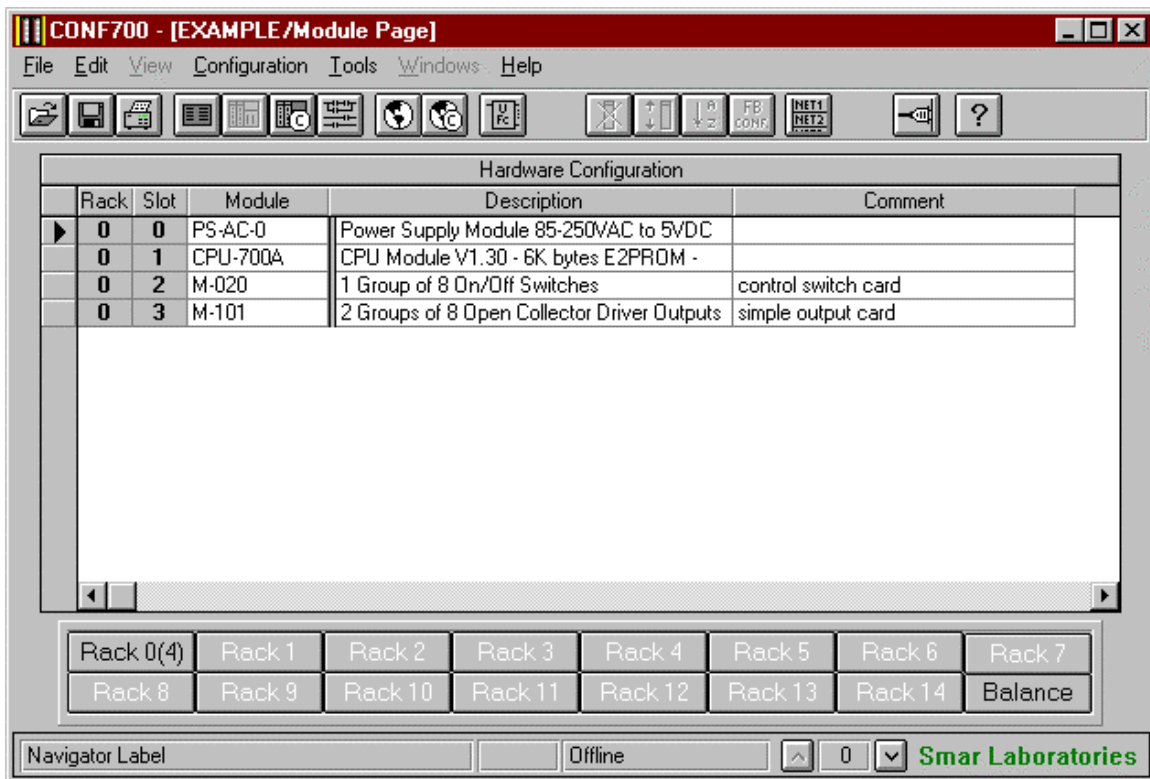
This is the place where the user defines the Racks and Modules that will compose the PLC-700 system for his specific application. This page gives the user a complete picture of the hardware.

Every time the user adds a Input or Output Module the configurator allocates “Global Variables” directly related with I/O points of the Module and a “Default” names for them. Besides the default attributions the user can also create his own Tags for each “Group” and “Point” related with any used Module.

To add a Rack click on an available button marked with a faded Rack number right under the list of modules. A dialog box will appear. Mark the box that says “In Use”. See The Rack Dialog Box for details.

To add a Module just click on an empty line in the Module column. A look up list of available Modules will appear for selection. Click on the desired Module and one or more Global Variable will be automatically created in reference to the inputs and/or outputs of the Module.

The Fieldbus Module requires an extra step of informing how many of each type of I/O blocks have being used.

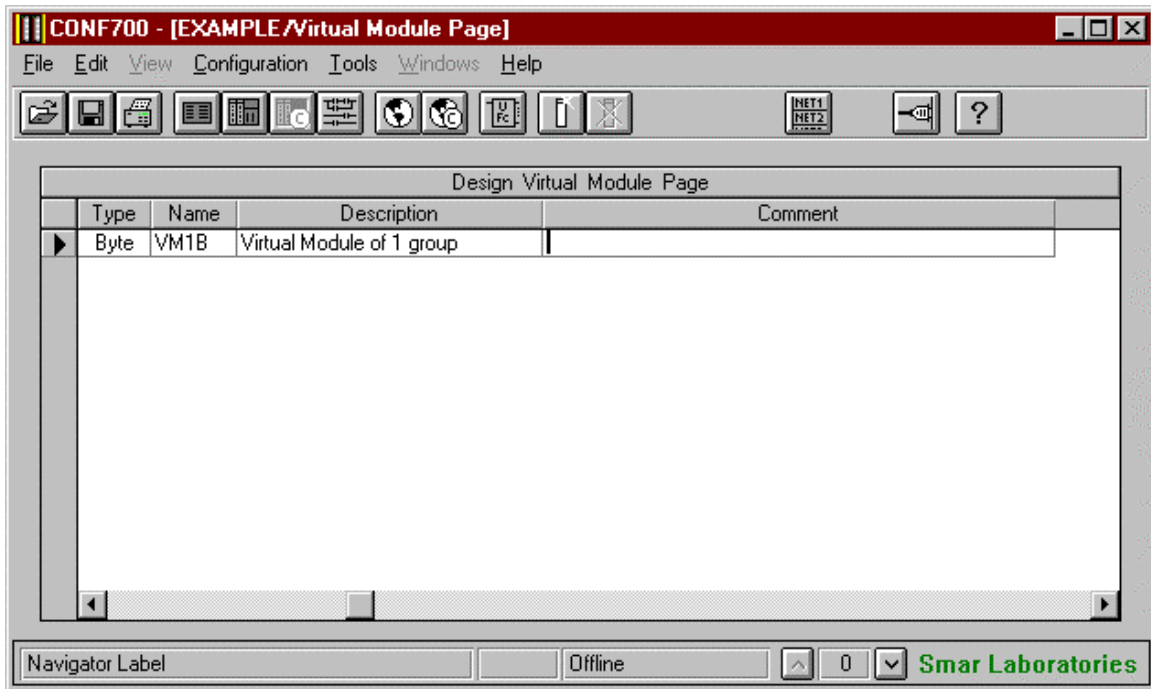


### 3.5.1.2. Definição de variáveis reais e virtuais

#### Virtual Variables

Virtual Variables are auxiliary variables that can be used in any part of the logic program, either in the “Ladder Logic Network” or in the writing of a “User Function”. They are not physically related with any of the physical I/O points of the Modules but only internal variables. Their values can also be changed and read by the communication port.

You add Virtual Variables when you create a Virtual Module. On the list of Virtual Modules you can see which are the groups of Virtual Variables that are going to be created. The memory allocation for those Virtual Variables is also automatically done by the configurator.

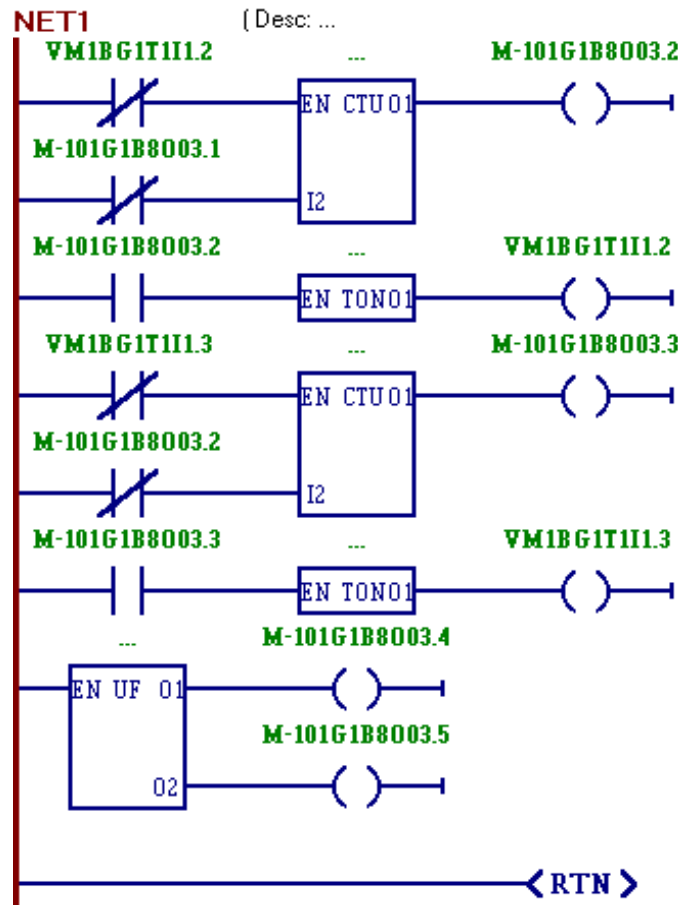


### 3.5.1.3. Configuração do ladder

#### Ladder Configuration

This is the Page where the user writes the logic strategy for his application. In this Page the user defines Networks (NET) and fill them with all the necessary elements ( Relays, Coils, User Functions and Built-In Functions, Jumps, Returns and Lines) to accomplish an application.

The PLC-700 will execute the Networks in the order the were created, but the execution order can be change by the configurator. Networks can be deleted or inhibited of execution as well.



#### Network Management

Click this button (BUTTON) or call Network Management from the menu under Configure. This dialog box is self explanatory. The description text will instruct you on real time! In this Dialog Box you can:

- Change the execution order of the Networks.
- Delete or Stop\Enable the execution of Networks. Note that a Network that has being stopped will still occupie space in the PLC memory, if you do not need it anymore you may want to delete it.
- Send changes when you are directly connected with the PLC-700.

### 3.5.1.4. Configuração de blocos FIELDBUS

#### Fieldbus Configuration






The Fieldbus module blocks are linked with other blocks of instruments in the field through the SMAR SYSCOM.

For the PLC-700 it is necessary to know the quantity and type of Fieldbus blocks used per channel of Fieldbus Modules. When configuring the PLC-700 this information must be entered. The configurator will allocate the correct space in the memory automatically.

When a Communication Input Block (CIxx) or a Communication Output Block (COxx) is used (linked) the PLC-700 will allocate room in the CPU Module for all its corresponding inputs or output. Non-used blocks in the Fieldbus Module are completely ignored.

Once these I/Os values are in the CPU Module memory the user can have access to them through the EIA-485 or EIA-232C channels.

#### Table of Fieldbus blocks per channel.

Block Diagram	Description	Quantity/ Channel	Input Type	Output Type	Memory
	<b>ALARM</b> - The input signal will be examined and an appropriate output will be set in order to reflect its condition.	8	FP	Digital	0
	<b>Comm. Input Digital Data</b> - 8 Digital signals from the input will be available for the PLC.	4	Digital	-	8 Bits in the Digital I/O area.
	<b>Comm. Input Analog Data</b> - 8 Analog Signals from the input will be available for the PLC.	2	FP	-	Take the place of 16 Analog Signals
	<b>Comm. Output Digital Data</b> - 8 Digital signals from the PLC can be sent to the network.	3	-	Digital	8 Bits in the Digital I/O area
	<b>Comm. Output Analog Data</b> - 8 Analog signals from the PLC can be sent to the network.	2	-	Analog	Take the place of 16 Analog Signals

### **3.5.2. Exercício 1 - Acionamento de motor com proteção**

Desenvolver a configuração do controlador lógico para um intertravamento simples de motor.

### **3.5.3. Exercício 2 - Link com FIELDBUS; alarme de alta temperatura acionando uma saída do LC700**

Desenvolver a configuração do controlador lógico para o acionamento de uma saída digital sinalizando um alarme em uma aplicação FIELDBUS.