

MFI-PBM

Descrição técnica

Departamento de Pesquisa e Desenvolvimento
Versão. 1.0
Data: 02/abril/2008

ÍNDICE

ÍNDICE	2
ÍNDICE DE FIGURAS	3
ÍNDICE DE TABELAS	4
1. Introdução	5
1.1. Sistema Citrino	5
1.2. Sobre o <i>PROFIBUS DP</i>	6
1.3. Características do MFI-PBM	9
2. Dimensionamento	10
2.1. Conexão do módulo na rede e configuração da base BMFI-PBS	11
3. Configuração do módulo MFI-PBM	12
3.1. Incluir o mestre MFI-PBM na configuração utilizando o software SyCon	13
3.2. Configurar os parâmetros <i>PROFIBUS</i> do mestre	14
3.3. Incluir os escravos na configuração do mestre	15
3.4. Gerar o arquivo com extensão .dbm	18
3.5. Importar o arquivo .dbm gerado pelo SyCon	20
3.6. Descarregar via MCPU-1 a configuração no mestre MFI-PBM	23
4. Diagnósticos e Alarmes gerados para o MFI-PBM	25
4.1. Descrição do diagnóstico gerado para o MFI-PBM	25
4.2. Definição dos alarmes gerados para o MFI-PBM	27
4.3. Como armazenar diagnósticos e alarmes em WM	28
5. LEDs do MFI-PBM	31
6. Informações técnicas sobre o MFI-PBM	33
7. Informações adicionais	35
8. Lista de abreviações:	36

ÍNDICE DE FIGURAS

Figura 1.1.1 – Sistema Citrino.....	6
Figura 2.1 – Dimensional mecânico do módulo MFI-PBM	10
Figura 2.1.1 – Exemplo de utilização do terminador	12
Figura 2.1.2 – Terminador, Polarizador e conexão dos fios no conector DB-9.....	12
Figura 3.1.1 – Exemplo de configuração no SyCon (definindo o Mestre)	14
Figura 3.2.1 – Exemplo de parametrização do mestre MFI-PBM.....	15
Figura 3.3.1 – Exemplo de configuração e parametrização no SyCon (instalando GSD)	16
Figura 3.3.2 – Exemplo de configuração e parametrização no SyCon (definindo escravo)	17
Figura 3.3.3 – Exemplo de configuração e parametrização no SyCon (ícone do escravo).....	17
Figura 3.3.4 – Exemplo de configuração e parametrização no SyCon (configuração).....	18
Figura 3.3.5 – Exemplo de configuração e parametrização no SyCon (parametrização)	18
Figura 3.4.1 – Exemplo de configuração no SyCon	19
Figura 3.4.2 – Salvando arquivo com extensão .dbm no SyCon	19
Figura 3.5.1 – Iniciando a página inicial do CitrinoTools	21
Figura 3.5.2 – Montando a arquitetura do sistema com o MFI-PBM.....	21
Figura 3.5.3 – Preparando o CitrinoTools para adicionar a configuração PROFIBUS	22
Figura 3.5.4 – Arquivo .dbm a ser importado pelo CitrinoTools.....	22
Figura 3.5.5 – Anexando as entradas e saídas com relação à variável WM	23
Figura 3.6.1 – Descarregando no MFI-PBM via MCPU-1	24
Figura 3.6.2 – Processo de descarregando no MFI-PBM via MCPU-1	24
Figura 4.3.1 – Habilitando diagnósticos na MCPU-1	29
Figura 4.3.2 – Escolhendo o WM para diagnósticos e alarmes	30
Figura 6.1 – Comandos acíclicos.....	33

ÍNDICE DE TABELAS

Tabela 1.3.1 – Descrição técnica do módulo MFI-PBM	9
Tabela 2.1.1 – Pinagem do conector DB-9 PROFIBUS	11
Tabela 2.1.2 – Configuração da base BMFI-PBS.....	11
Tabela 4.1.1 – Representação do diagnóstico de estação não existente (WM0 escolhido pelo usuário) 26	
Tabela 4.1.2 – Representação do diagnóstico estendido ou de canal (WM0 escolhido pelo usuário)	27
Tabela 4.2.1 – Representação dos alarmes recebidos (WM0 escolhido pelo usuário)	28
Tabela 5.1 – Descrição do significado dos LEDs do MFI-PBM	32
Tabela 6.1 – Descrição dos comandos acíclicos suportados pelo MFI-PBM	34

1. Introdução

O objetivo é descrever o módulo de interface PROFIBUS mestre para o sistema Citrino.

As seguintes informações descrevem o módulo mestre *PROFIBUS* da Fertron, o MFI-PBM (Module Fieldbus Interface - PROFIBUS Master). Este módulo é considerado um mestre Classe-1 e ele contém algumas funções para o funcionamento de um mestre Classe-2 simultaneamente.

Este módulo tem o objetivo de dar conexão remota *PROFIBUS DP* e *PROFIBUS DP-V1*. Ou seja, um escravo *PROFIBUS* de qualquer fabricante (Allen Bradley, Siemens, GE) poderá se comunicar com o módulo MFI-PBM da Fertron e deixar os dados disponíveis na memória do módulo MCPU-1 para o controle da aplicação desejada.

Nos próximos itens serão abordadas descrições técnicas, como devem ser feitas a instalação, configuração e parametrização do módulo MFI-PBM e por fim explicar o diagnóstico, os LEDs e os comandos Classe-1 e Classe-2 suportados.

Para informações mais detalhadas sobre o protocolo, acesse o site da organização *PROFIBUS* (PTO - *PROFIBUS Trade Organization*) no website www.PROFIBUS.com.

1.1. Sistema Citrino

O Sistema Citrino é a mais nova família de CLPs (Controladores Lógicos Programáveis) da Fertron. Este sistema é baseado por CPUs, Módulos de entradas e saídas digitais e analógicas, módulos de interfaces de rede de campo (*fieldbus*) e módulos de fontes.

Portanto o sistema Citrino tem a capacidade de automatizar vários tipos de aplicações distintas dando bastante flexibilidade.

O módulo MCPU-1 é um módulo que tem alta capacidade de processamento e de memória. Ele contém 20 MBytes de memória e a CPU trabalha numa frequência de 500MHz.

Os módulos de entrada e saída (E/S) digitais e analógicas podem ser compostos de vários módulos distintos, como por exemplo M16AO-IV (módulo com 16 saídas analógicas), M32DI-24V (módulo de 32 entradas digitais com acionamento em 24 Volts).

O módulo de interface de rede, é um módulo que pode conter um protocolo específico para controle. O MFI-PBM é um módulo de interface de rede *PROFIBUS* mestre. Este módulo propicia uma conectividade da MCPU-1 tanto com os módulos de entradas e saídas digitais e analógicas do Citrino (utilizando o módulo escravo MFI-PBS como interface escrava no protocolo *PROFIBUS*) quanto os módulos de E/S de outros tipos de fabricantes que contenham um módulo escravo no protocolo *PROFIBUS*.



Figura 1.1.1 – Sistema Citrino

1.2. Sobre o PROFIBUS DP

O protocolo *PROFIBUS* foi criado em 1989 pelo governo alemão em conjunto com vários fabricantes de equipamentos de automação. Ele é um padrão aberto e é reconhecido como o *Fieldbus* mais rápido da atualidade. Ele é baseado no RS-485 em um padrão Europeu EN-50170 (*PROFIBUS -FMS, -DP, -PA e DPV1*) e é orientado no modelo de referência OSI (*Open System Interconnection*). Neste modelo cada camada gerencia precisamente tarefas definidas. A camada 1 deste modelo é a camada física e define as características de transmissões físicas. A camada 2 é a camada de dados e define o acesso ao barramento do protocolo. A camada 7 é a da aplicação e define as funções de aplicações.

O *PROFIBUS DP* define duas classes de mestre. O mestre classe-1 gerencia a comunicação normal ou a troca de dados (*data exchange*) com o escravo designado em seu barramento. Um mestre classe-2 é um dispositivo especial utilizado inicialmente para configurar escravos e para propósitos de diagnósticos. Alguns mestres podem suportar ambas funcionalidades classe-1 e classe-2. O módulo MFI-PBM é um mestre classe-1 e ele suporta vários comandos necessários para o funcionamento de um mestre classe-2, que será explicado posteriormente.

Um mestre classe-1 manda a taxa de transmissão (*baud-rate*) e os escravos auto-detectam esta transmissão. Como já foi explicado anteriormente, ele gerencia o "*data exchange*" com os escravos designados e atua como um controlador principal para a troca de informações de E/S ciclicamente com seus escravos distribuídos de acordo com o ciclo da mensagem definida. Um mestre pode comunicar ativamente com seus escravos designados, mas unicamente comunicações passivas (no requerimento) com outro dispositivo do mestre classe-2.

O mestre classe-2 é normalmente um dispositivo de configuração, talvez um software ou computador, e seu propósito é para comissionamento, manutenção e diagnósticos. Ele atua como um mestre "supervisor" e que pode comunicar ativamente com um mestre classe-1 e seus escravos, em adição com seus próprios escravos. Isto é, um mestre classe-2 pode gerenciar o controle de um escravo que não sejam os seus designados, brevemente e através do outro mestre classe-1. Todas as trocas de informações entre um mestre classe-1 e um mestre classe-2, são iniciadas originalmente no mestre classe-2.

O tamanho dos dados de entrada e saída para serem transferidos de um escravo para um mestre é definido na base de dados do escravo ou arquivo GSD. Os arquivos GSD de cada dispositivo conectado na rede (unicamente mestre classe-1 e escravos) são compilados dentro de uma lista de parâmetros do mestre que contém dados de parametrização e configuração, uma lista de configuração de endereços e parâmetros de barramento para todas as estações conectadas. O mestre utiliza estas informações para iniciar a comunicação com os escravos durante o "startup"

Após o mestre receber sua lista de parâmetros ele estará pronto para iniciar o *data exchange* com seus escravos. Durante o *startup*, após um sistema de *reset* ou no retorno de *power-up*, o mestre tentará restabelecer o contato com todos os escravos designados a ele antes de assumir qualquer troca de dados cíclicos de E/S. Cada escravo deverá ter um único endereço válido de 0 a 125 para comunicar com o mestre. Qualquer escravo que tiver um endereço *default* 126 esperará um comando *Set_Slave_Address* (SSA) de um mestre classe-2 antes de ele ser parametrizado. O módulo MFI-PBM tem suporte ao comando SSA e deverá ser utilizado o CitrinoTools para fazer o comando SSA. Na tentativa de estabelecer conexão o mestre começará com um escravo de menor endereço e finalizará no escravo de maior endereço. O mestre enviará parametrização e configuração para todos os seus escravos designados (um escravo pode unicamente estar acessível via escrita com o mestre designado que o configurou e o parametrizou durante o *startup*). Se um escravo adicional for colocado no barramento da rede e ele não estiver incluído na lista de parâmetros do mestre, será necessária a geração de uma nova configuração para ser descarregado na lista de parâmetros do mestre para a inclusão do *status* do novo escravo.

O PROFIBUS DP é mais comumente utilizado como mestre classe-1 (sistemas mono-mestre) fazendo um "polling" ciclicamente com os escravos distribuídos. Entretanto, o PROFIBUS também permite comunicações acíclicas entre um mestre de classe-2 e os escravos fazendo mais que uma estação ativa ou mestre possíveis. Um mestre classe-1 automaticamente detectará a presença de uma nova estação ativa conectada ao barramento de rede (mestre classe-2). Quando um mestre classe-1 completa o seu ciclo de *polling*, ele passará o *token* para o mestre classe-2 garantindo um acesso temporário no barramento da rede. Comportamentos determinístico são mantidos uma vez que o mestre classe-2 pode unicamente utilizar o tempo permitido por uma via (*gap time*) especificada. Um sistema PROFIBUS pode ter mais que um mestre classe-1, mas as comunicações mestre a mestre não são permitidas exceto para garantir o acesso ao barramento via troca de *token*.

Para ilustrar a idéia de comunicação entre mestres em um sistema PROFIBUS DP, um mestre classe-1 troca dados simultaneamente com todos os seus escravos simultaneamente, um de cada vez, de acordo com sua lista de parâmetros previamente armazenada. No final do ciclo de dados (*gap time*) é permitida a comunicação cíclica entre o mestre classe-2 e os mesmos escravos designados ao mestre classe-1. Durante este tempo, um mestre classe-1 passará o *token* ao mestre classe-2 garantindo o seu acesso ao barramento. O mestre classe-2 com o *token* na mão tem a oportunidade de trocar dados com todos os escravos dentro de um período de tempo específico chamado de *token half-time* ou *token hold-time* (T_H). O mestre classe-2 pode então proceder a leitura de dados ou informações de diagnóstico de quaisquer escravo e então finalizar o ciclo, passando o *token* de volta ao mestre de classe-1.

Normalmente o *gap time* não tem tempo suficiente para finalizar uma troca total de dados, desta maneira este processo pode ser continuado durante vários ciclos pelo mestre classe-2. Quando a transferência da lista de dados finalizar o mestre classe 2

fechará a conexão. Observe que o mestre classe-2 pode unicamente estabelecer conexão com os escravos durante o *gap time*.

Como já foi explicado anteriormente, é possível para um mestre classe-2 tomar o controle de um escravo DP. Durante este tempo o escravo parará o seu *data exchange* com o mestre classe-1. O mestre classe-1 reconhecerá isto e pedirá ciclicamente diagnósticos do escravo, checando o campo de endereço do mestre enquanto estiver presente um outro endereço de mestre válido. Após o mestre classe-2 finalizar a sua comunicação com o escravo, ele seta o campo de endereço do mestre no escravo para um valor inválido (255). Isto fará com que o mestre de classe-1 tente controlar o escravo novamente e ele será reparametrizado e reconfigurado antes de voltar ao *data exchange* com o mestre de classe-1.

MFI-PBS tem um arquivo GSD (FTR_OB2A.gsd) disponível no site da Fertron para *download*.

2. Dimensionamento

O sistema Citrino é um produto com grau de proteção IP-20 e sua utilização mais comum é a instalação em painéis. Na Figura 2.1 pode-se ver o dimensional do equipamento para instalação em painéis.

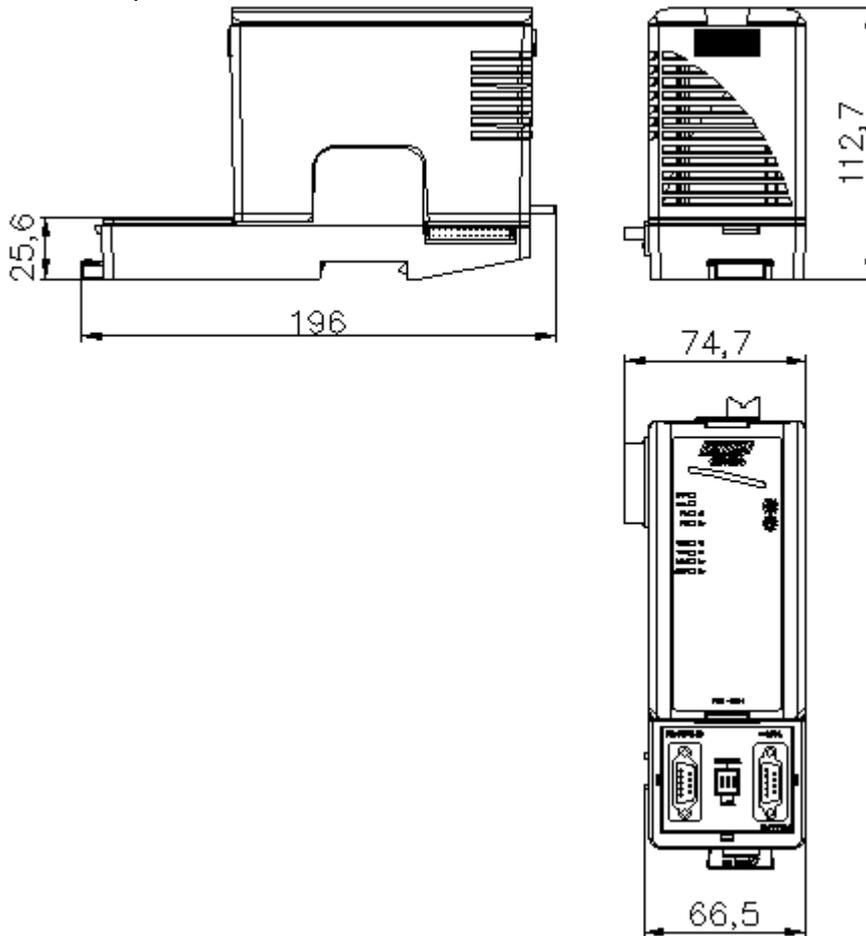


Figura 2.1 – Dimensional mecânico do módulo MFI-PBM

O módulo MFI-PBM deve estar localizado na sua base (BMFI-PBM) situada no mesmo segmento que a MCU-1 (segmento 0). Ou seja, o módulo MFI-PBM nunca poderá ficar nos segmentos que contém módulos de expansão. Será permitido ainda até quatro módulos MFI-PBM, e todos deverão ficar situados no primeiro segmento.

2.1. Conexão do módulo na rede e configuração da base BMFI-PBS

O módulo MFI-PBM contém um conector fêmea padrão DB-9. Os pinos do conector DB-9 são convencionados da seguinte maneira:

Pino	Sinal	Descrição
1	PE	Terra de proteção e/ou blindagem
2	NC	-
3	B-Line	Recepção e transmissão de dados positivos
4	CNTR-P	Sinal de controle para repetidores (direção de controle)
5	DGND	Potencial de referência
6	VP	Alimentação positiva para terminação (+5Vcc)
7	NC	-
8	A-Line	Recepção e transmissão de dados negativos
9	NC	-

Tabela 2.1.1 – Pinagem do conector DB-9 PROFIBUS

Os pinos 5 e 6 não devem ser utilizados, já que o propósito deles é apenas como referência para a polarização dos sinais A-Line e B-Line (*pull-up* e *pull-down*). Esta polarização já se encontra na DIP 2 da base (BMFI-PBM), não havendo a necessidade de colocar no cabo do conector DB-9 esta função.

Como já foi comentado anteriormente, a base do módulo mestre PROFIBUS é chamada de BMFI-PBM. A sua configuração deve respeitar a Tabela 2.1.2.

Segmentos	BMFI-PBM	
	DIP 1	DIP 2
Segmento 0	Slots 0-7 (1 a 4 OFF) se houver outros módulos de I/O posterior aos do mestre; (1-2 OFF 3-4 ON) Se for o ultimo slot do primeiro segmento; (1-2 ON 3-4 OFF) Se for o ultimo slot do único segmento (ou seja, não há outros segmentos);	Para ligar o terminador (2 ON), se ultimo nó ou se primeiro nó; Para ligar os polarizadores (1 e 3 ON);

Tabela 2.1.2 – Configuração da base BMFI-PBS

O PROFIBUS utiliza o terminador para diminuir a reflectância. Esta reflectância é causada pelo aumento gradativo da capacitância ao longo do distância percorrida pelo cabo. Isto faz com que haja espúrios no sinal e o terminador acaba com este efeito fazendo um casamento de impedâncias ao longo da linha. Ele deve ser utilizado em cada início e final de segmento. Um exemplo de utilização do terminador pode ser visto na Figura 2.1.1. Já o polarizador visa estabelecer uma normalização de tensão em toda a linha entre 0V (GND) e 5V.

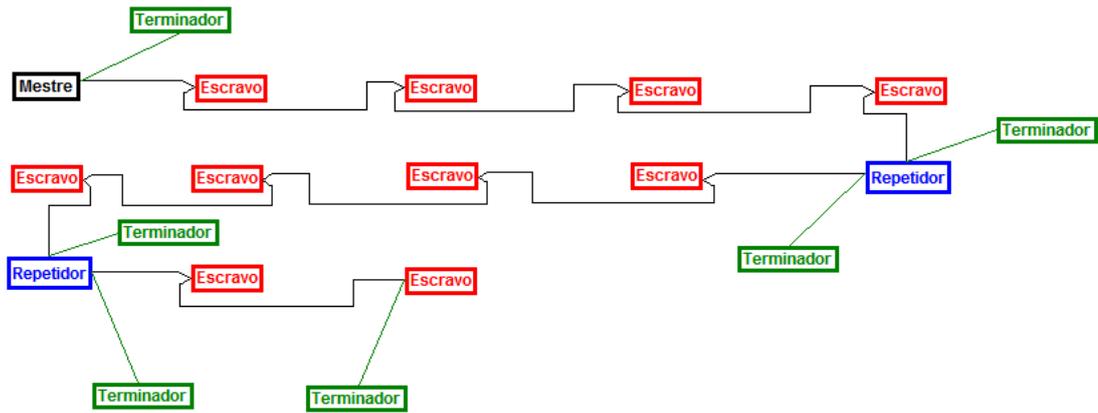


Figura 2.1.1 – Exemplo de utilização do terminador

Na Figura 2.1.2 pode ser visto como deve ser conectado o cabo PROFIBUS em seu conector e também pode ser visto como é a definição elétrica do terminador e do polarizador. Observe que a blindagem do cabo deve ser aterrada em ambas as pontas para a comunicação PROFIBUS.

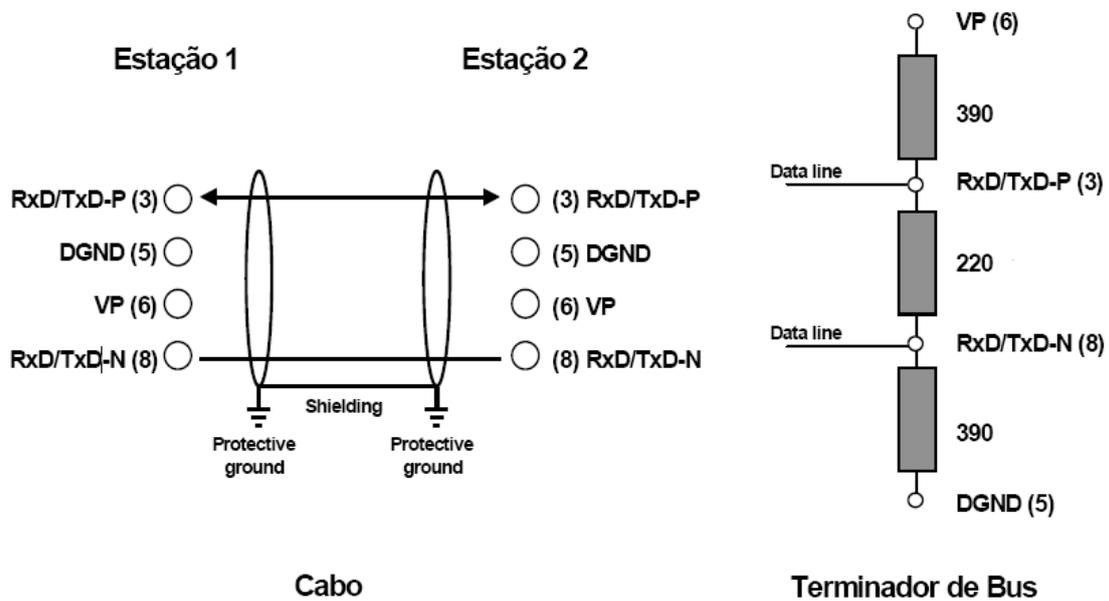


Figura 2.1.2 – Terminador, Polarizador e conexão dos fios no conector DB-9

3. Configuração do módulo MFI-PBM

Para fazer a configuração do mestre MFI-PBM são necessários o Software Configurador CitrinoTools e o Software SyCon da Hilscher para gerar o arquivo com extensão .dbm que posteriormente será importado pelo CitrinoTools.

Para configurar o módulo mestre PROFIBUS, são necessários os seguintes passos:

- 1) Incluir o mestre MFI-PBM na configuração utilizando o software SyCon;

- 2) Configurar os parâmetros *PROFIBUS* do mestre;
- 3) Incluir os escravos na configuração do mestre;
- 4) Parametrizar os escravos;
- 5) Gerar o arquivo com extensão .dbm;
- 6) Abrir o Software CitrinoTools;
- 7) Importar o arquivo .dbm gerado pelo SyCon;
- 8) Escolher a variável que será armazenada e escrita nos dados da rede PROFIBUS (por exemplo WM, DM, RM);
- 9) Descarregar via M_{CPU}-1 a configuração no mestre MFI-PBM;
- 10) Esperar em torno de uns 15 segundos até salvar a configuração em flash;

Todos esses detalhes serão explicados com mais detalhes a seguir. Será considerado que o usuário já terão os Softwares SyCon e CitrinoTools instalados com suas respectivas licenças habilitadas.

3.1. Incluir o mestre MFI-PBM na configuração utilizando o software SyCon

Aqui serão ensinados unicamente como incluir o mestre e os GSDs dos escravos no software SyCon sem entrar muito em detalhes das ferramentas disponíveis no software. O importante para nós será o arquivo exportado para a interpretação no CitrinoTools.

Abaixo está uma breve descrição sobre a inclusão do mestre MFI-PBM no software SyCon.

- 1) Abra o Software configurador SyCon;
- 2) Clique no menu *Insert->Master*;
- 3) No lugar do mouse aparecerá a letra M. Clique na barra vertical que se encontra no software do SyCon. Aparecerá um diálogo;
- 4) Neste diálogo mova a barra de rolagem (*Available Masters*) e clique em COM-C-DPM e logo em seguida em *Add>>* (observe Figura 3.1.1);

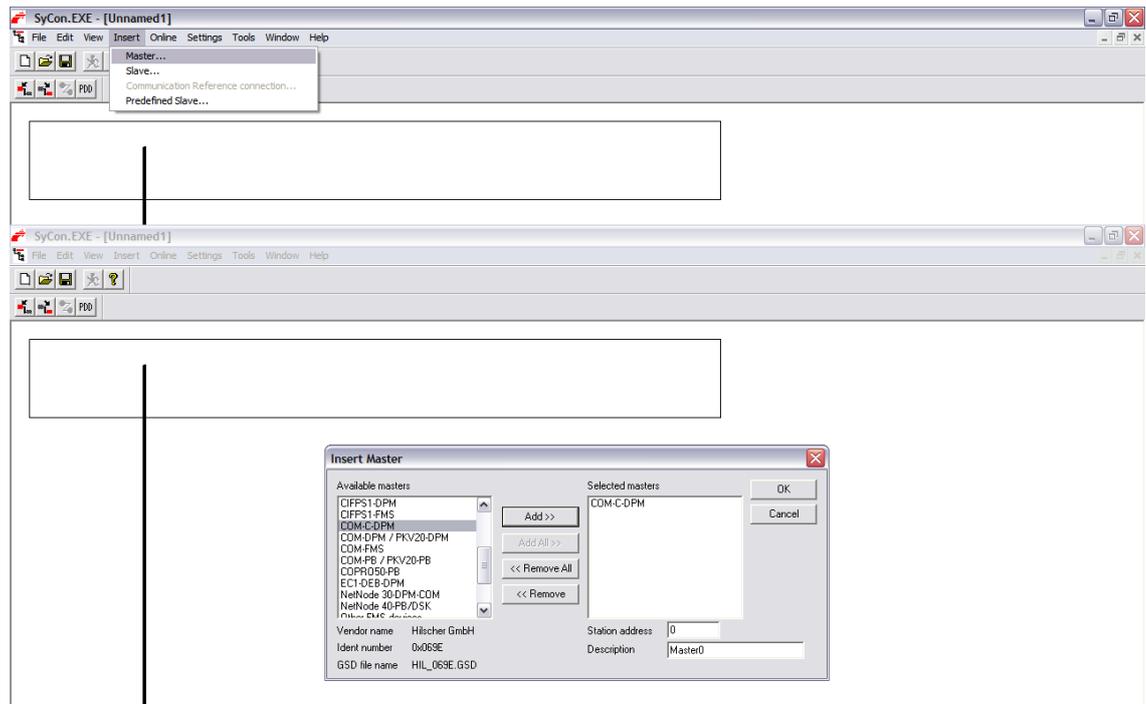


Figura 3.1.1 – Exemplo de configuração no SyCon (definindo o Mestre)

3.2. Configurar os parâmetros **PROFIBUS** do mestre

Abaixo está uma breve descrição sobre como configurar o mestre.

- 1) Com o mestre já definido no passo anterior, dê um duplo clique no ícone do mestre (observe a Figura 3.2.1);
- 2) Aparecerá um diálogo chamado *Master Configuration*;
- 3) Neste diálogo pode-se modificar o endereço (*Station Address*) ou modificar o tag (*Description*);
- 4) Clique em *DP Master Settings...*;
- 5) Os seguintes parâmetros devem estar configurados:
 - a. *Automatic Release of the communication by the device* em *Startup behaviour after system initialization*;
 - b. *Byte addresses* em *Addressing mode*;
 - c. *Big Endian (MSB-LSB)* em *Storage format (word module)*;
 - d. *Buffered, host controlled* em *Handshake of the process data*;
- 6) Clique em OK para confirmar alguma mudança caso seja necessário;

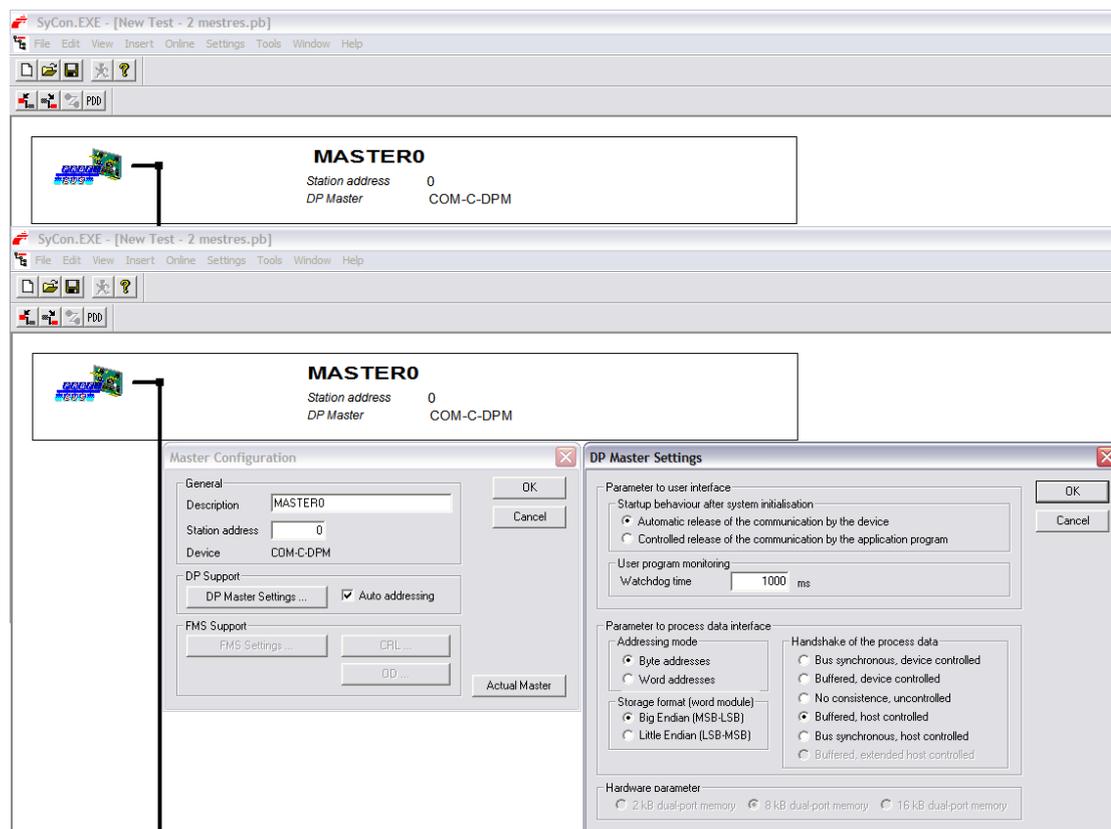


Figura 3.2.1 – Exemplo de parametrização do mestre MFI-PBM

3.3. Incluir os escravos na configuração do mestre

Abaixo tem um exemplo de como importar arquivos GSDs. Neste exemplo, mostrará como importar o arquivo GSD do módulo escravo PROFIBUS do Citrino MFI-PBS.

- 1) Instale o GSD, clicando em *File->Copy GSD*. Em seguida localize o arquivo (FTR_OB2A.gsd) já salvo em seu computador (Figura 3.3.1). Após selecioná-lo em seu diretório, clique em *Open*;
- 2) Clique no menu *Insert->Slave*;
- 3) No lugar do mouse aparecerá a letra S. Clique na barra vertical que se encontra no software do SyCon abaixo do mestre. Aparecerá um diálogo;
- 4) Neste diálogo mova a barra de rolagem (*Available Slaves*) e clique em MFI-PBS e logo em seguida em *Add>>* (observe Figura 3.3.2);
- 5) Pode-se modificar o endereço (como exemplo coloque 5 em *Station Address*) e em *Description* coloque MFI_PBS (Figura 3.3.2);
- 6) Desta maneira aparecerá um ícone como o da Figura 3.3.3;
- 7) Agora a configuração deve ser feita. Apenas como exemplo, coloque 4 módulos de M16AI-IV, 6 módulos M32DI-24V, 4 módulos M16AO-IV e 2 módulos M32DO-TR dando um duplo clique no módulo selecionado na janela *Slave Configuration* (Figura 3.3.4);

- 8) Para parametrizar cada um dos módulos nos slots, clique em *Parameter Data...* e aparecerá uma janela como o da Figura 3.3.5. Se clicarmos no botão *Common*, aparecerão dois parâmetros específicos da interface e não dos módulos nos slots (*Operation Mode* e *Diagnostic Information*). Deixe-os como está. Qualquer dúvida sobre o significado dos parâmetros, leia o manual do módulo escravo PROFIBUS do Citrino (MFI-PBS). Se clicarmos no botão *Module*, pode-se parametrizar um módulo em um determinado slot. Para parametrizar o primeiro M16AI-IV, basta dar um duplo clique no primeiro módulo da janela da Figura 3.3.5 e escolher as opções. Qualquer outra parametrização deverá ser feita utilizando o mesmo raciocínio. Após finalizado clique em OK em todas as janelas;
- 9) Para instalar outros GSDs, volte ao passo 1 e de maneira semelhante aos passos de 2 ao 8, configure e parametrize o escravo com o GSD instalado;

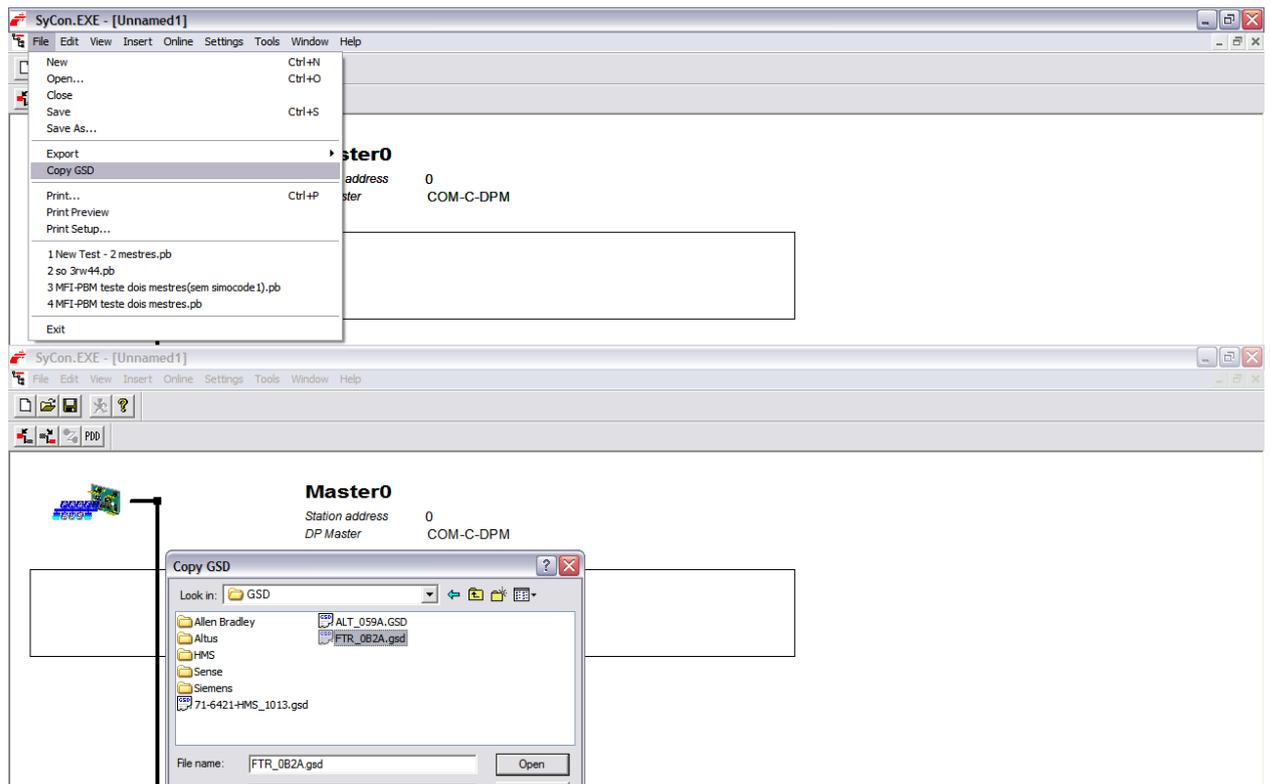


Figura 3.3.1 – Exemplo de configuração e parametrização no SyCon (instalando GSD)

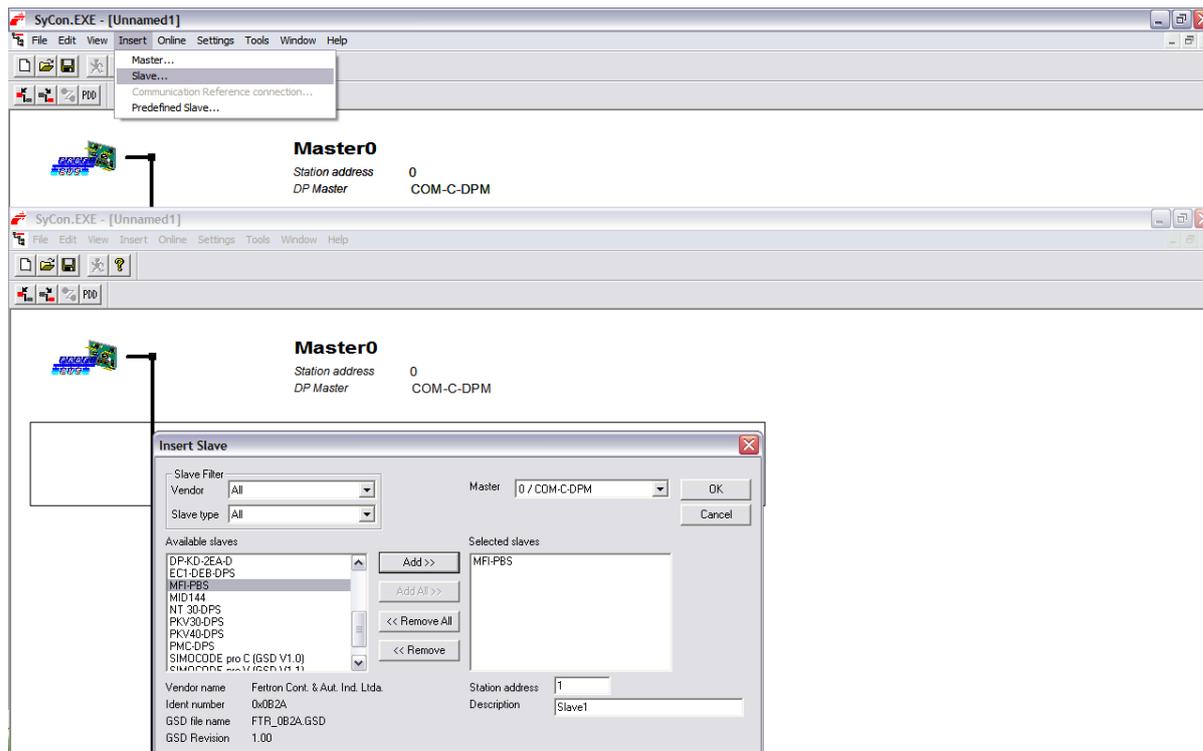


Figura 3.3.2 – Exemplo de configuração e parametrização no SyCon (definindo escravo)

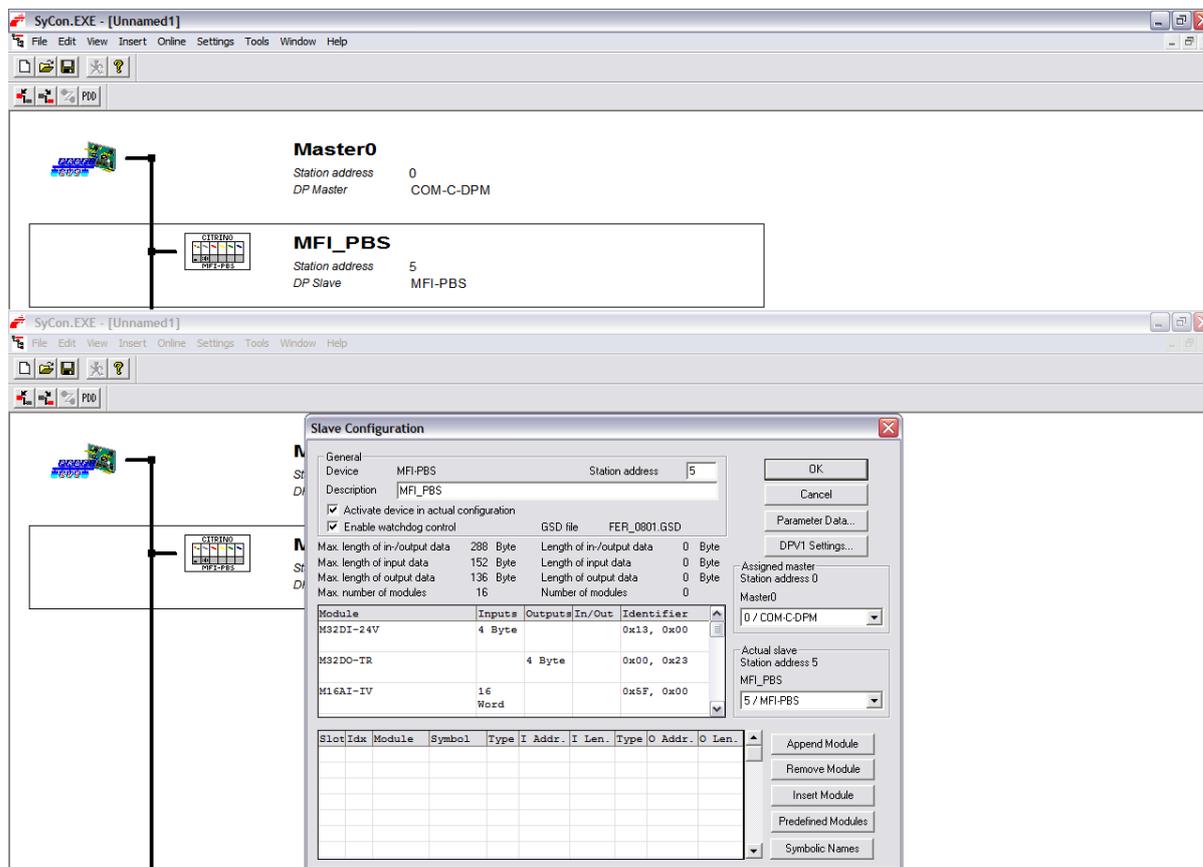


Figura 3.3.3 – Exemplo de configuração e parametrização no SyCon (ícone do escravo)

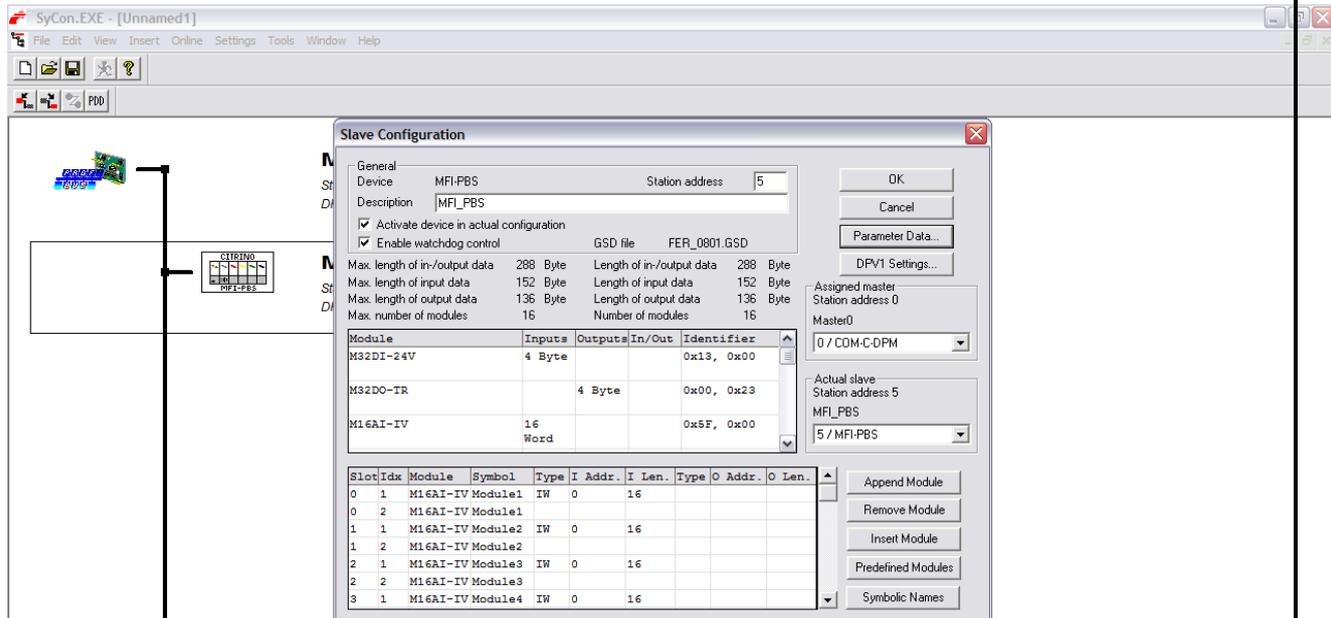


Figura 3.3.4 – Exemplo de configuração e parametrização no SyCon (configuração)

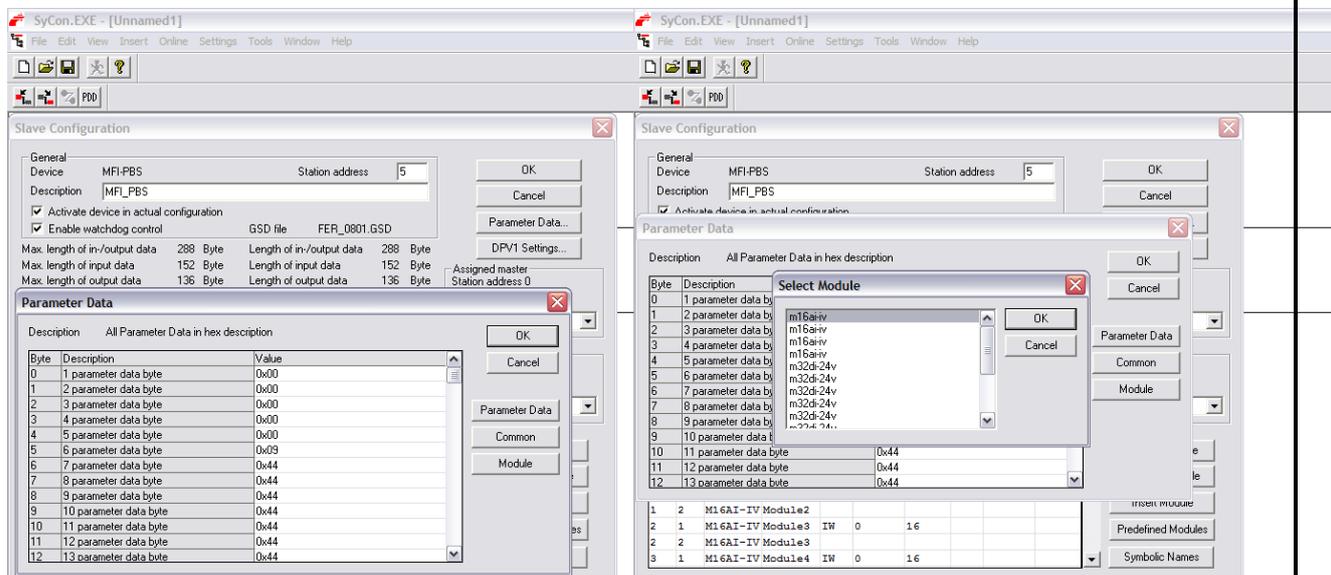


Figura 3.3.5 – Exemplo de configuração e parametrização no SyCon (parametrização)

3.4. Gerar o arquivo com extensão .dbm

Abaixo está um exemplo de como gerar um arquivo com extensão .dbm.

- 1) Abra o arquivo, clicando em *File->Open*. Depois selecione o arquivo com extensão .pb gerado anteriormente e clique em OK. No nosso exemplo, colocaremos uma configuração da Figura 3.4.1;

- 2) Clicando no ícone do mestre, será selecionado o mestre cuja configuração deverá ser exportada (podem existir mais de um mestre na configuração). Agora exporte o arquivo, clicando em *File->Export->DBM*. Este arquivo será gerado no mesmo diretório em que o arquivo .pb tinha sido gerado. No caso do exemplo, este arquivo foi gerado no diretório *Desktop* (Figura 3.4.2);

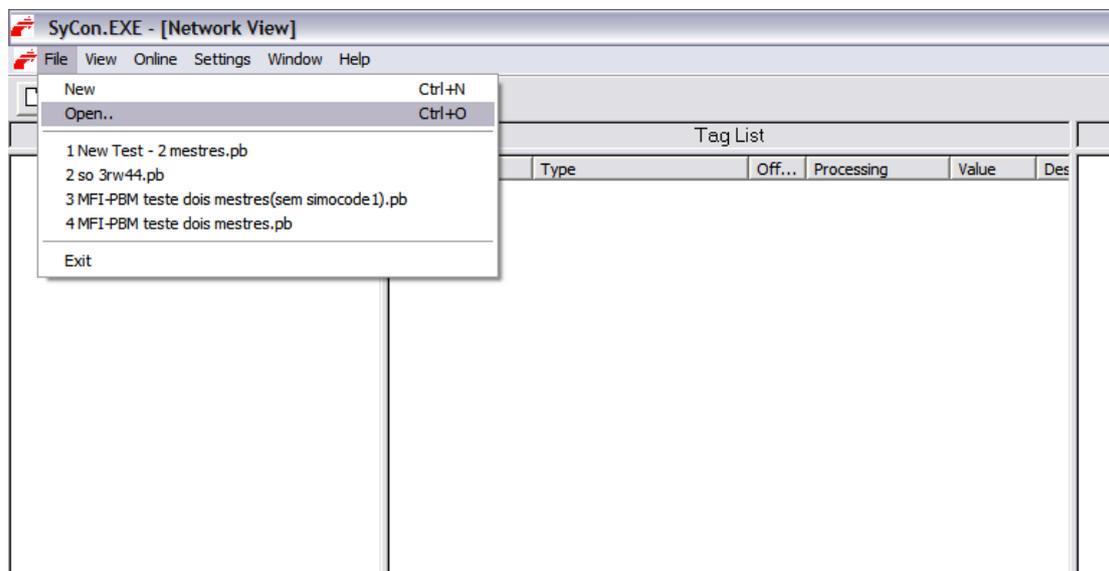


Figura 3.4.1 – Exemplo de configuração no SyCon

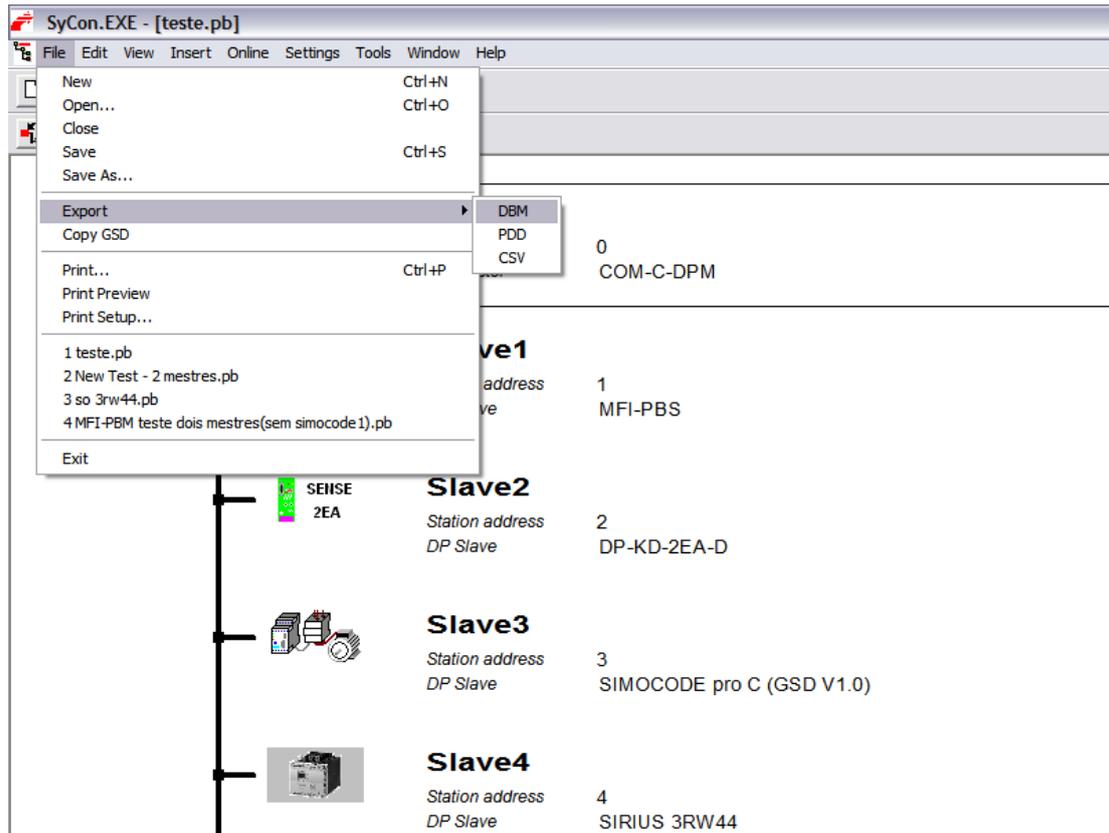


Figura 3.4.2 – Salvando arquivo com extensão .dbm no SyCon

3.5. Importar o arquivo .dbm gerado pelo SyCon

Abaixo está um exemplo de como importar um arquivo com extensão .dbm através do CitrinoTools.

- 1) Com o Software CitrinoTools aberto, clique no ícone Novo (*New*) para começar uma nova configuração (arquitetura do sistema). Observe a Figura 3.5.1;
- 2) Monte a arquitetura do sistema de acordo com a Figura 3.5.2. Conecte na CPU clicando no menu Ethernet->Conectar (*Ethernet->Connect*);
- 3) Agora clique o botão da direita no ícone do MFI-PBM de acordo com a Figura 3.5.3. Em seguida clique em Associar Configuração (*Assign Configuration*) e aparecerá um diálogo como o da Figura 3.5.3;
- 4) No diálogo Associar Configuração (*Assign Configuration*), clique no botão ... (observe a Figura 3.5.4) e aparecerá um novo diálogo para selecionar o arquivo .dbm a ser importado pelo CitrinoTools da configuração feita no SyCon. No nosso exemplo anterior, o arquivo tinha sido exportado para o diretório *Desktop*. Portanto, selecione o arquivo teste.dbm como pode ser visto na Figura 3.5.4 e selecione a opção Abrir (*Open*);
- 5) Após selecionar Abrir (*Open*) no item anterior, o diálogo ficará de acordo com a Figura 3.5.5. Observe que é colocada a quantidade de *words* de entrada (no exemplo 14) e quantidade de *words* de saída (neste exemplo 19). Fica livre para o usuário colocar em quais WMs serão colocadas essas variáveis de entradas e de saídas no PROFIBUS. No nosso exemplo, foi colocado que as variáveis de entrada ocupariam a partir do WM 0 e as variáveis de saídas ocupariam a partir do WM 1000. Clique no botão Aplicar (*Apply*) para confirmar as mudanças. Se o usuário quiser ver como ficaram ocupadas as variáveis de memórias WMs, clique no botão Visualizar Configuração (*View Configuration*) e aparecerá a aba PROFIBUS com os desenhos ilustrativos como o da Figura 3.5.5;

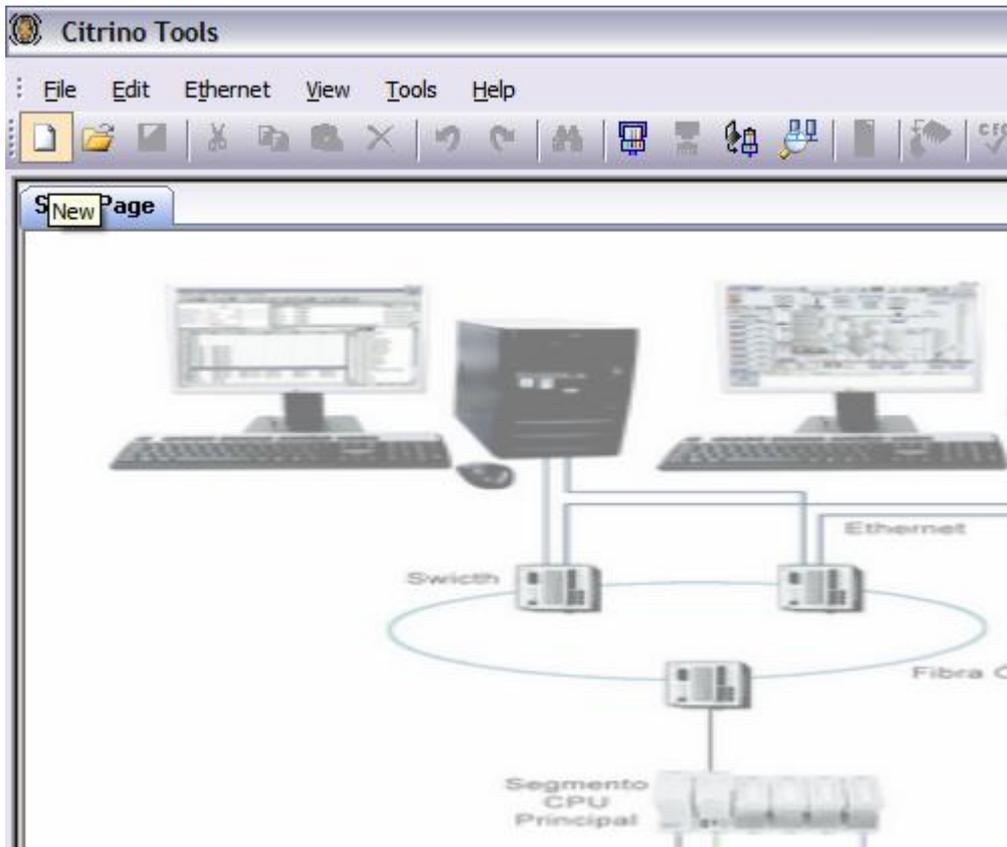


Figura 3.5.1 – Iniciando a página inicial do CitrinoTools

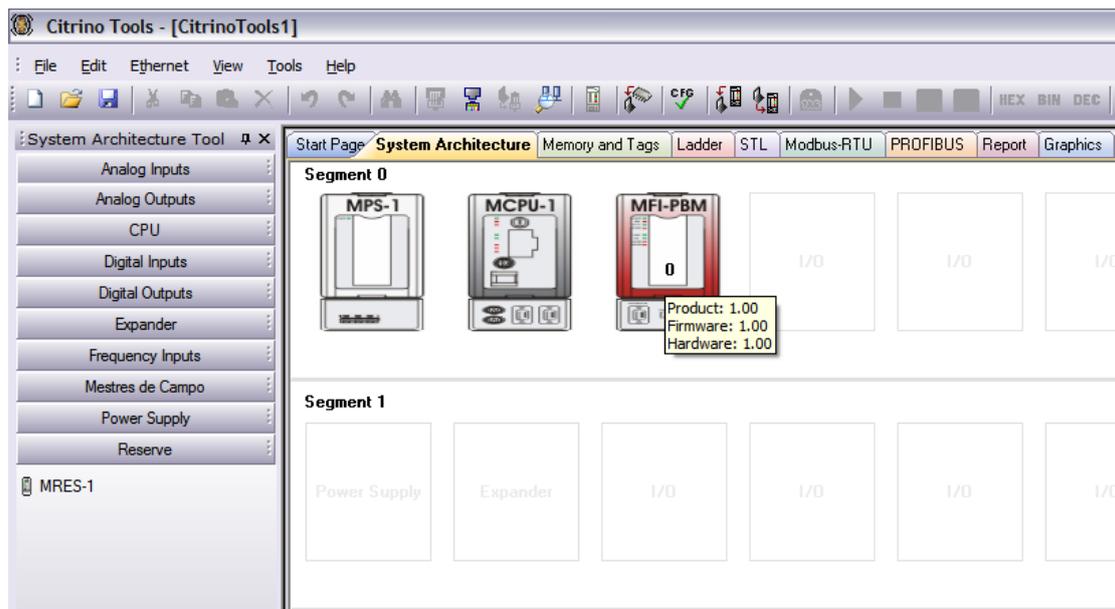


Figura 3.5.2 – Montando a arquitetura do sistema com o MFI-PBM

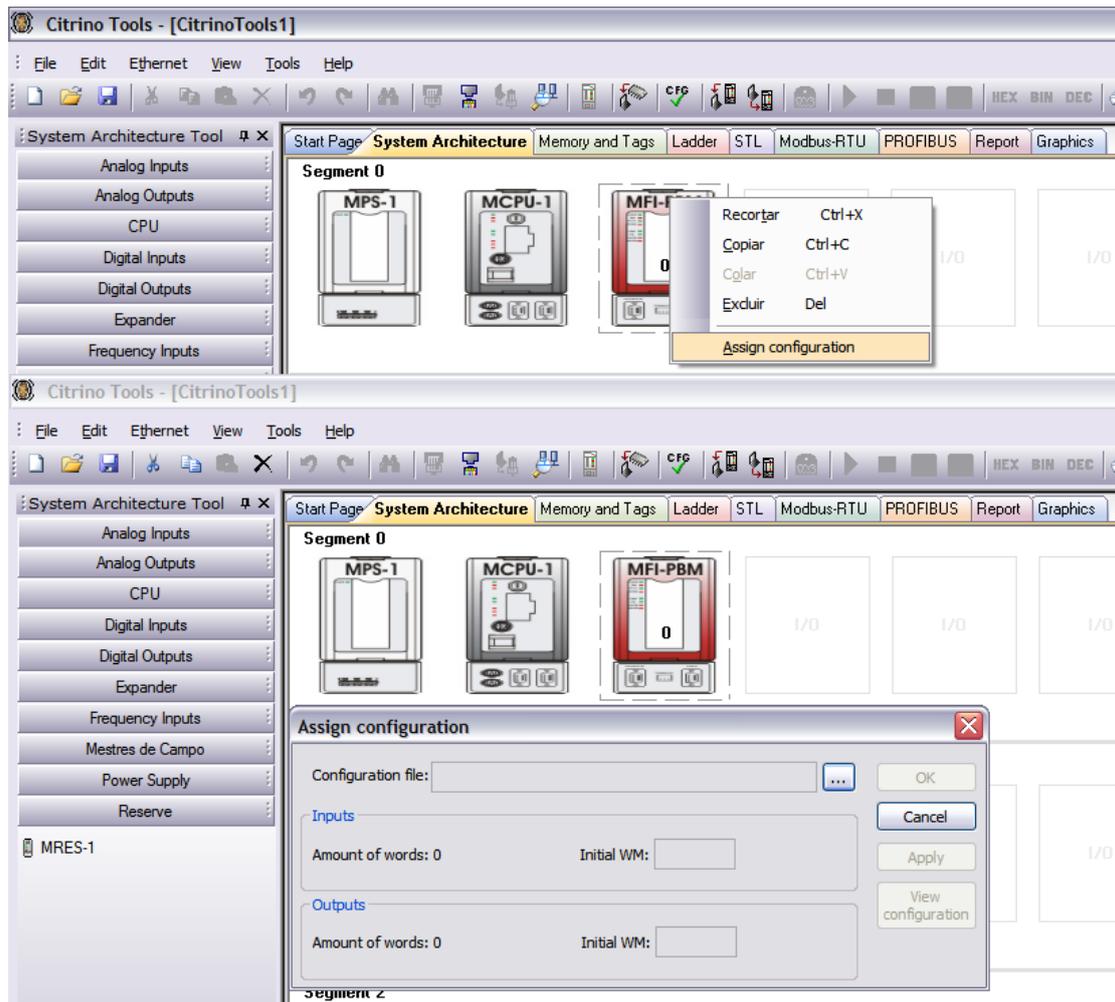


Figura 3.5.3 – Preparando o CitrinoTools para adicionar a configuração PROFIBUS

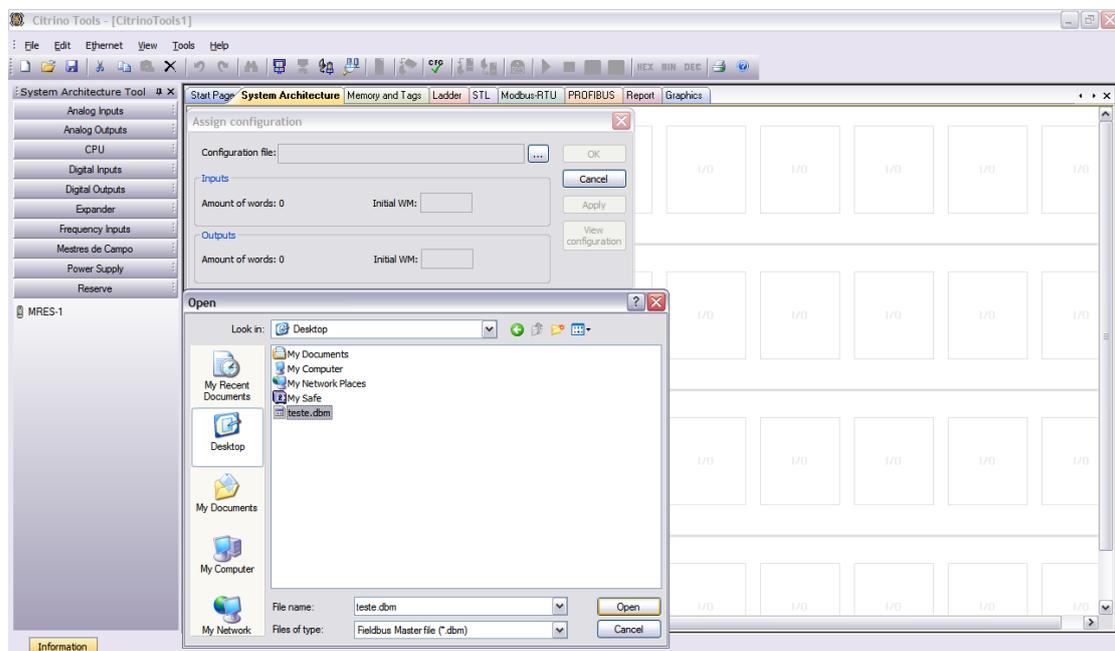


Figura 3.5.4 – Arquivo .dbm a ser importado pelo CitrinoTools

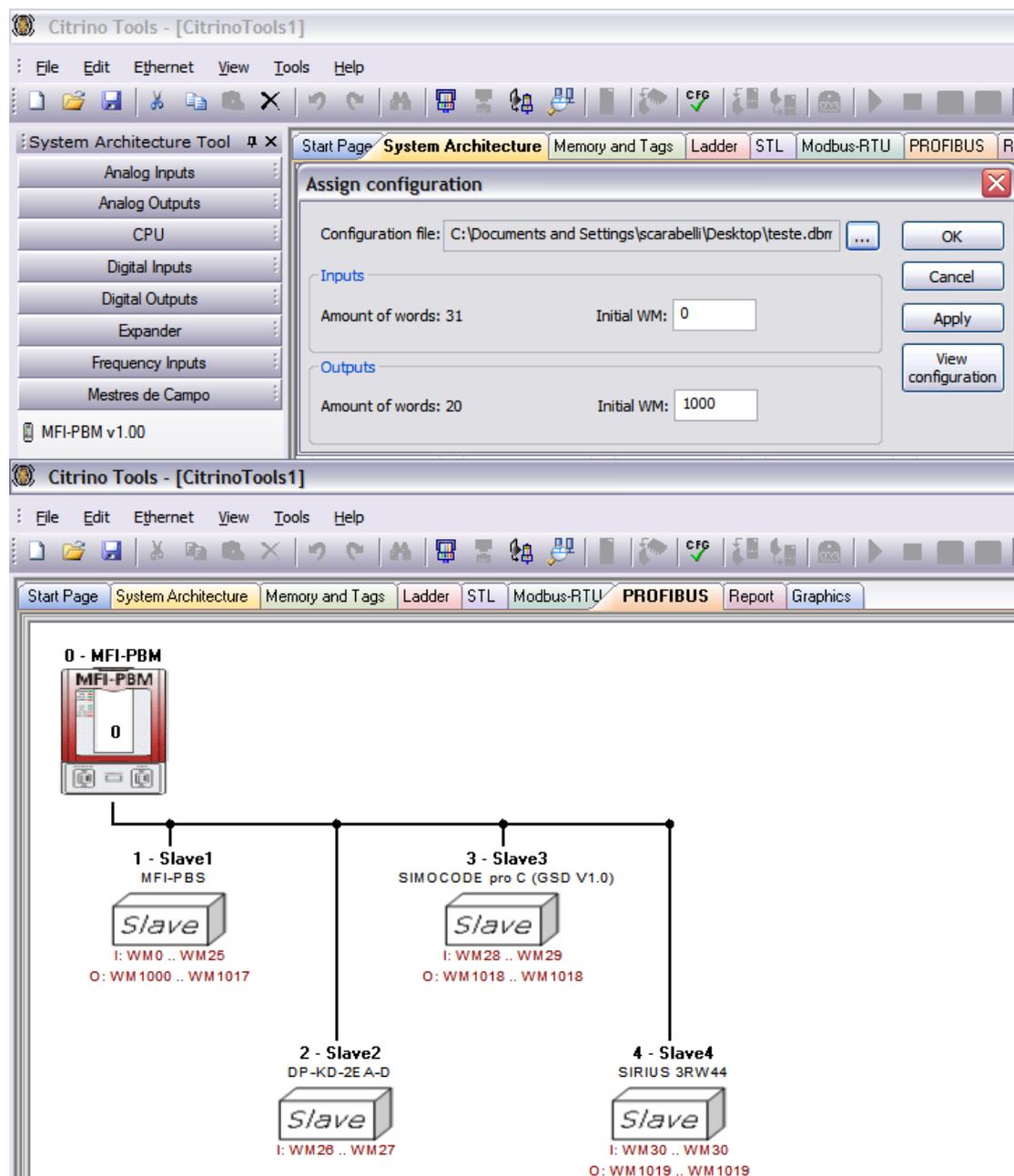


Figura 3.5.5 – Anexando as entradas e saídas com relação à variável WM

3.6. Descarregar via MCPU-1 a configuração no mestre MFI-PBM

Abaixo tem um exemplo de como descarregar a configuração PROFIBUS no mestre MFI-PBM via MCPU-1.

- 1) Continuando do item anterior, clique em Ferramentas->Armazenar->Armazenar Somente Configurações (*Tools->Store->Store Only Configuration*) para descarregar a configuração. Aparecerá um diálogo igual ao da Figura 3.6.1. Observe que o botão Configuração do Módulo MFI-PBM v1.0 (Firmware 1.0)

endereço 0 (*MFI-PBM v1.00 module Address 0 Configuration*) estará desabilitado. Habilite este item clicando no botão verde e selecione o botão *Armazenar (Store)*;

- 2) No diálogo *Communication*, aparecerá uma frase informando que está sendo finalizada a configuração no mestre desejado (*Finishing configuration storage in the PROFIBUS master with address 0...*). Depois que é finalizado o salvamento em flash no módulo MFI-PBM (este processo pode demorar de 15 a 30 segundos), o CitrinoTools envia a configuração restante (Ladder, arquitetura do sistema, Modbus-RTU, etc) para a CPU e finaliza o processo de descarga da configuração. Todo este comentário pode ser observado na Figura 3.6.2;

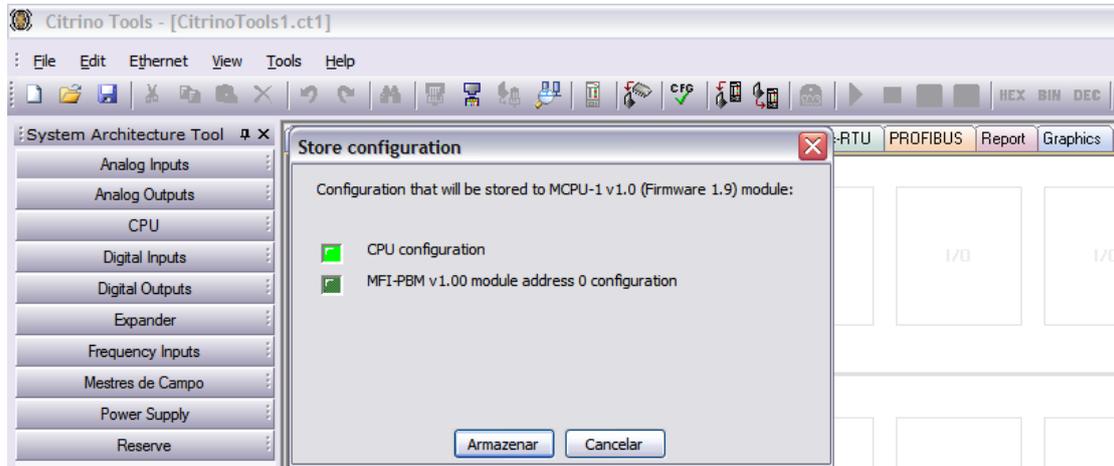


Figura 3.6.1 – Descarregando no MFI-PBM via MCPU-1

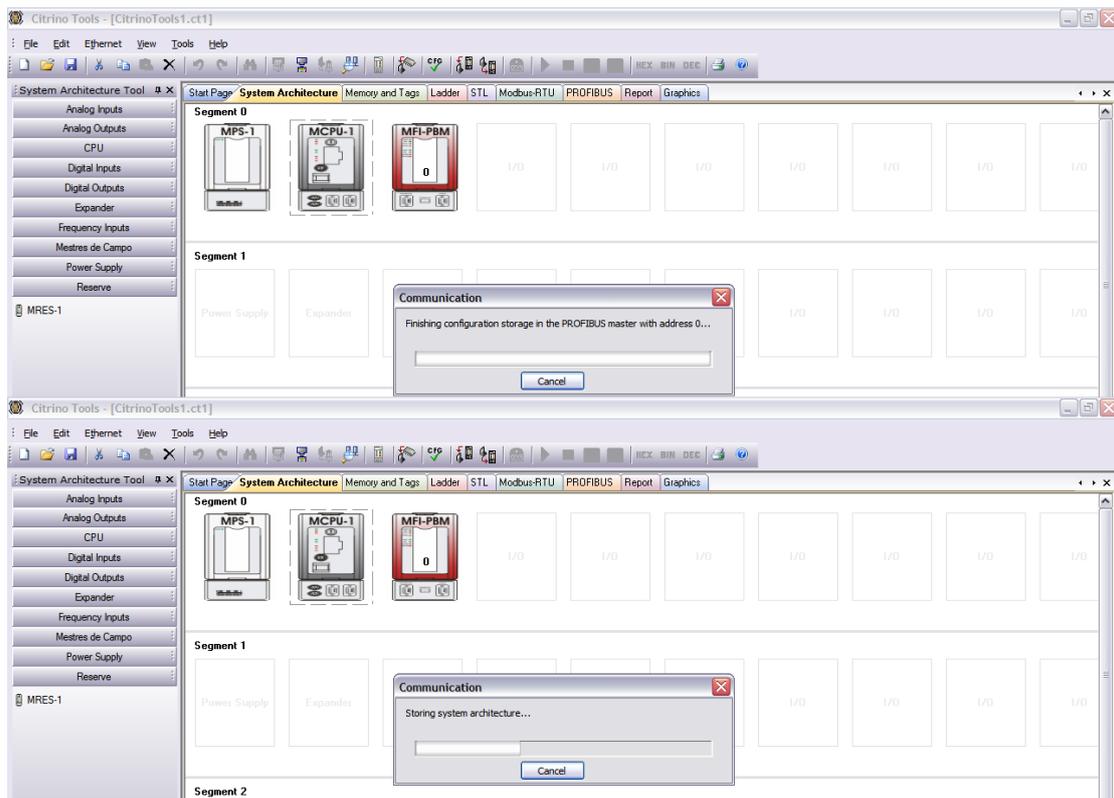


Figura 3.6.2 – Processo de descarregando no MFI-PBM via MCPU-1

4. Diagnósticos e Alarmes gerados para o MFI-PBM

Quando se fala em diagnóstico e alarmes em *PROFIBUS*, significa dizer que esses alarmes e diagnósticos devem estar de acordo com as normas EN 50170 e PROFIBUS-DP Extensions to EN50170 (DPV1). Portanto, é necessário que essas normas sejam consultadas em caso de dúvidas.

Neste capítulo será explicado como deve ser interpretado tanto o diagnóstico quanto os alarmes gerados pelos escravos *PROFIBUS* visto na memória da MCPU-1.

4.1. Descrição do diagnóstico gerado para o MFI-PBM

Quando um escravo *PROFIBUS* gera um diagnóstico, este diagnóstico fica armazenado no módulo MFI-PBM. O módulo então avisa a MCPU-1 através do barramento F-Bus interno que existe pelo menos um diagnóstico a ser lido. A MCPU-1 então lê este diagnóstico e armazena na memória (área de WM) para indicar qual escravo tem diagnóstico na linha *PROFIBUS*.

Este diagnóstico gerado pode ser dividido em três tipos distintos:

1. *Estação (Escravo) não existente ou em falha;*
2. *Diagnóstico estendido ou de canal do escravo;*
3. *Alarmes;*

A área de memória será previamente reservada pelo usuário através do CitrinoTools para os dois tipos, independente da existência ou não dos diagnósticos.

Nos exemplos da Tabela 4.1.1, Tabela 4.1.2 e Tabela 4.2.1 foi dado um exemplo considerando que o usuário escolheu que a partir do WM0 serão salvos os diagnósticos e os alarmes. Então do WM0 ao WM7 estarão os diagnósticos de estação não existente ou em falha. Do WM8 ao WM15 estarão os diagnósticos estendidos ou de canal e do WM16 ao WM23 estarão os alarmes (somente escravos DPV1 geram alarmes).

- Estação não existente ou em falha:

Este tipo de diagnóstico é de formato geral, como por exemplo, escravo ausente ou com erro estático. Se o escravo estiver ausente, este diagnóstico irá setar o bit correspondente do WM escolhido. Se o escravo tiver endereço 0, o bit zero do WM escolhido ficará setado. Se o escravo tiver o endereço 1, o bit 1 do WM escolhido ficará setado e assim por diante. Estes mesmos bits só serão zerados se o diagnóstico do escravo sumir, ou seja, neste exemplo se o escravo comunicar com o seu mestre PROFIBUS na rede o bit dele deverá estar zerado. Vale o mesmo raciocínio caso o escravo esteja em falha. Observe a Tabela 4.1.1 para entender melhor como ficará na memória o diagnóstico dos escravos.

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WM0	B0*	E7	E6	E5	E4	E3	E2	E1	E0†
	B1	E15	E14	E13	E12	E11	E10	E9	E8
WM1	B2	E23	E22	E21	E20	E19	E18	E17	E16
	B3	E31	E30	E29	E28	E27	E26	E25	E24
WM2	B4	E39	E38	E37	E36	E35	E34	E33	E32
	B5	E47	E46	E45	E44	E43	E42	E41	E40
WM3	B6	E55	E54	E53	E52	E51	E50	E49	E48
	B7	E63	E62	E61	E60	E59	E58	E57	E56
WM4	B8	E71	E70	E69	E68	E67	E66	E65	E64
	B9	E79	E78	E77	E76	E75	E74	E73	E72
WM5	B10	E87	E86	E85	E84	E83	E82	E81	E80
	B11	E95	E94	E93	E92	E91	E90	E89	E88
WM6	B12	E103	E102	E101	E100	E99	E98	E97	E96
	B13	E111	E110	E109	E108	E107	E106	E105	E104
WM7	B14	E119	E118	E117	E116	E115	E114	E113	E112
	B15	E127	E126	E125	E124	E123	E122	E121	E120

Tabela 4.1.1 – Representação do diagnóstico de estação não existente (WM0 escolhido pelo usuário)

- Diagnóstico Estendido ou de canal do escravo:

No diagnóstico estendido estão compreendidos os diagnósticos previstos na norma EN50170: *Device Related*, *Identifier Related* e *Channel Related* (que significam respectivamente, Diagnóstico relacionado ao equipamento, diagnóstico relacionado ao Módulo ou *Slot* e diagnóstico relacionado ao canal). Como já foi falado anteriormente, estes diagnósticos não serão comentados aqui, qualquer dúvida a norma deve ser consultada. Se algum desses diagnósticos existir, o bit que representa o endereço do escravo deve ficar setado. Por exemplo, imagine que o usuário tenha escolhido o WM0 para armazenar os diagnósticos. Então para o diagnóstico estendido, os WMs começam a partir do WM8 e terminam no WM15 (observe a Tabela 4.1.2). Agora imagine que o escravo de endereço três tenha algum tipo de diagnóstico estendido. O bit3 do WM8 ficará setado. Para o usuário identificar qual o tipo de diagnóstico estendido que o escravo de endereço 3 tem (*Device Related*, *Identifier Related* ou *Channel Related*), ele deve utilizar o Software CitrinoTools. O Software dependerá do GSD do escravo para indicar no formato texto qual o diagnóstico do escravo. O mestre MFI-PBM tem a capacidade de guardar 100 bytes de diagnósticos por escravo.

* B0 – Byte 0; B1 – Byte 1 e assim por diante

† E0 – Endereço Equipamento 0; E1 – Endereço do Equipamento 1 e assim por diante

		Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
WM8	B16	E7	E6	E5	E4	E3	E2	E1	E0
	B17	E15	E14	E13	E12	E11	E10	E9	E8
WM9	B18	E23	E22	E21	E20	E19	E18	E17	E16
	B19	E31	E30	E29	E28	E27	E26	E25	E24
WM10	B20	E39	E38	E37	E36	E35	E34	E33	E32
	B21	E47	E46	E45	E44	E43	E42	E41	E40
WM11	B22	E55	E54	E53	E52	E51	E50	E49	E48
	B23	E63	E62	E61	E60	E59	E58	E57	E56
WM12	B24	E71	E70	E69	E68	E67	E66	E65	E64
	B25	E79	E78	E77	E76	E75	E74	E73	E72
WM13	B26	E87	E86	E85	E84	E83	E82	E81	E80
	B27	E95	E94	E93	E92	E91	E90	E89	E88
WM14	B28	E103	E102	E101	E100	E99	E98	E97	E96
	B29	E111	E110	E109	E108	E107	E106	E105	E104
WM15	B30	E119	E118	E117	E116	E115	E114	E113	E112
	B31	E127	E126	E125	E124	E123	E122	E121	E120

Tabela 4.1.2 – Representação do diagnóstico estendido ou de canal (WM0 escolhido pelo usuário)

4.2. Definição dos alarmes gerados para o MFI-PBM

Quando um escravo *PROFIBUS* gera um alarme, este alarme também ficará armazenado no módulo MFI-PBM. Assim como no diagnóstico, o módulo então avisa a MCPU-1 através do barramento F-Bus interno que existe pelo menos um alarme a ser lido. A MCPU-1 então lê este alarme e armazena na memória (área de WM) para indicar qual o escravo que tem alarme na linha *PROFIBUS*.

Os alarmes são gerados por escravos que atendam a norma *PROFIBUS-DPV1*. E a norma diz que os alarmes necessitam de reconhecimento e enquanto não houver este reconhecimento, nenhum outro alarme daquele mesmo tipo será gerado pelo escravo. O módulo MFI-PBM tem a capacidade de reconhecer automaticamente o alarme para liberar o acionamento de novos alarmes pelo escravo. Porém esses alarmes serão armazenados na memória do mestre com uma capacidade máxima de armazenamento das últimas 32 ocorrências por escravo. Caso o usuário não leia e existir mais um alarme, o novo alarme gerado sobrescreverá o alarme mais antigo. Para o usuário ler o alarme é necessária a utilização do software *CitrinoTools*.

Apenas como exemplo, imagine que o usuário tenha escolhido o WM0 para armazenar os diagnósticos e alarmes. Então para o alarme, os WMs começam a partir do WM16 e terminam no WM23 (observe a Tabela 4.2.1). Se um escravo de endereço 23 tiver um alarme, o bit 7 do WM17 ficará setado. Para o usuário identificar qual o tipo de alarme que o escravo de endereço 23 tem, ele deve utilizar o Software *CitrinoTools*.

		Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
WM16	B32	E7	E6	E5	E4	E3	E2	E1	E0
	B33	E15	E14	E13	E12	E11	E10	E9	E8
WM17	B34	E23	E22	E21	E20	E19	E18	E17	E16
	B35	E31	E30	E29	E28	E27	E26	E25	E24
WM18	B36	E39	E38	E37	E36	E35	E34	E33	E32
	B37	E47	E46	E45	E44	E43	E42	E41	E40
WM19	B38	E55	E54	E53	E52	E51	E50	E49	E48
	B39	E63	E62	E61	E60	E59	E58	E57	E56
WM20	B40	E71	E70	E69	E68	E67	E66	E65	E64
	B41	E79	E78	E77	E76	E75	E74	E73	E72
WM21	B42	E87	E86	E85	E84	E83	E82	E81	E80
	B43	E95	E94	E93	E92	E91	E90	E89	E88
WM22	B44	E103	E102	E101	E100	E99	E98	E97	E96
	B45	E111	E110	E109	E108	E107	E106	E105	E104
WM23	B46	E119	E118	E117	E116	E115	E114	E113	E112
	B47	E127	E126	E125	E124	E123	E122	E121	E120

Tabela 4.2.1 – Representação dos alarmes recebidos (WM0 escolhido pelo usuário)

IMPORTANTE:

Os bits de diagnósticos serão zerados caso o evento que o geraram não exista mais. Ou seja, se existir um diagnóstico de ausência do endereço 8, o bit 8 do WM escolhido será setado. Quando o equipamento for colocado na rede, este bit será zerado pois não existirá mais o diagnóstico. Já os bits de alarmes serão zerados somente se o alarme for lido pelo Software CitrinoTools.

4.3. Como armazenar diagnósticos e alarmes em WM

Abaixo está um exemplo de como armazenar os diagnósticos e alarmes em WM:

- 1) Observe a configuração da Figura 4.3.1. Clique com o botão da direita do mouse no ícone da MCPU-1 e em seguida clique em Diagnósticos dos Módulos (*Modules Diagnostics*);
- 2) Aparecerá um diálogo chamado Diagnóstico dos Módulos (*Modules Diagnostics*) (Figura 4.3.2). Neste diálogo, observe o item Diagnóstico dos módulos do mestre 1 (endereço 0) (*Master 1 (address 0) modules diagnostics*). Habilite a caixa Usar diagnóstico do Mestre 1 (*Use master 1 diagnostics*). Observe que em Armazenamento dos diagnósticos (*Diagnostics Storage*) é habilitado o local para colocar o valor na variável WM. No exemplo foi colocado o valor 0. Note que automaticamente o Software CitrinoTools já reserva os próximos 128 WMs para diagnósticos. Apesar de o módulo MFI-PBM atualmente só utilizar 24

WMs para diagnósticos e alarmes, os WMs restantes ficam reservados para uso futuro;

IMPORTANTE:

Não existe a obrigação da leitura de diagnósticos. Se o usuário não quiser armazenar as informações de diagnósticos na MCPU-1 basta não habilitar esta função no software. Sem os diagnósticos, o usuário só perderia o debug gráfico dos módulos.

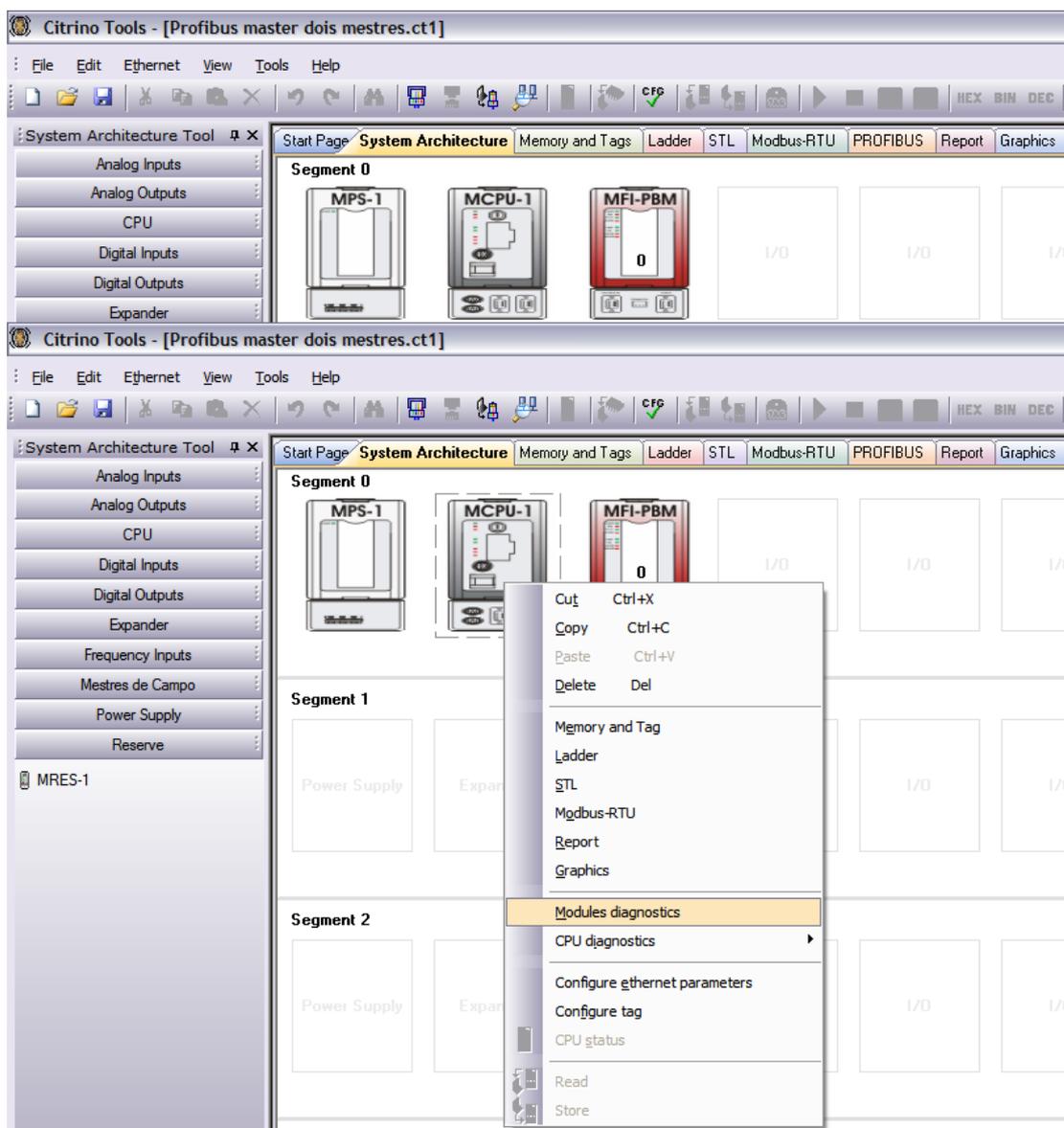


Figura 4.3.1 – Habilitando diagnósticos na MCPU-1

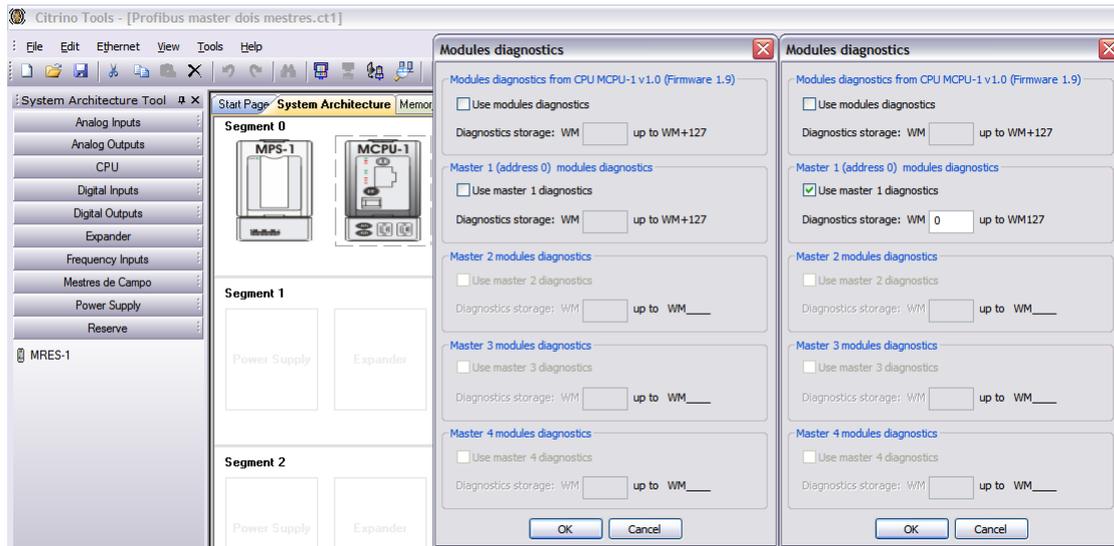


Figura 4.3.2 – Escolhendo o WM para diagnósticos e alarmes

5. LEDs do MFI-PBM

A interface mestre *PROFIBUS* indica diagnósticos e falhas de rede através dos LEDs em seu frontal.

Os diagnósticos dos LEDs visam ajudar o usuário para detectar erros ou problemas na rede *PROFIBUS*, incluindo erros durante o download da configuração e parametrização dos escravos.

Os LEDs PB ST e FB Err indicam diversas situações dependendo do número de vezes que o LED pisca. O LED PB ST indica algum estado relacionado à rede *PROFIBUS* (por exemplo, diagnósticos e alarmes de algum escravo) e o LED FB Err indica algum erro de parametrização ou configuração enviadas a ele.

Quando não existir nenhum tipo de erro, os LEDs devem ficar de acordo com a quarta coluna (Situação Normal) da Tabela 5.1.

O significado de cada um dos LEDs pode ser visto na Tabela 5.1.

LED	Descrição	Indicações	Situação normal
PWR (verde)	Energização do módulo	Aceso: módulo energizado Apagado: módulo desligado	Aceso quando ligado
FAIL (vermelho)	Falha de hardware ou reset	Piscando: reset em andamento Apagado: situação normal	Apagado
PB ST (verde)	Diagnósticos do mestre ou de campo	Piscando: 1x: erro de configuração 2x: diagnóstico presente no mestre ou em pelo menos um escravo Apagado: sem diagnóstico	Apagado
FB Err (vermelho)	Falha de configuração ou de comunicação com CPU	Piscando: 1x: MFI-PBM não parametrizado pela CPU 2x: MFI-PBM não configurado pela CPU 3x: MFI-PBM parametrizado incorretamente pela CPU 4x: MFI-PBM configurado incorretamente pela CPU 5x: MFI-PBM não recebe comunicação da CPU 6x: Erro geral de hardware, manutenção é necessária Apagado: sem falha	Apagado
COMM ST (verde)	Status de comunicação PROFIBUS-DP	Aceso: comunicação normal com escravos PROFIBUS Piscando em 5Hz: sem comunicação ou nenhum escravo comunicando Piscando irregularmente: falha de configuração	Aceso
TOK ST (verde)	Token de comunicação PROFIBUS-DP	Aceso: mestre detém o token da comunicação PROFIBUS-DP Piscando irregularmente: mestre compartilha o token PROFIBUS com outros mestres Apagado: mestre pode ter problemas de comunicação	Aceso ou piscando irregularmente
COMM Err (vermelho)	Erro de comunicação	Aceso: mestre tem problemas de comunicação com pelo menos um escravo Piscando irregularmente: mestre compartilha token PROFIBUS com outros mestres, mas tem problema de comunicação com pelo menos um escravo Apagado: situação normal	Apagado
HDW Err (vermelho)	Erro de hardware	Piscando irregularmente: erro grave de hardware Apagado: situação normal	Apagado

Tabela 5.1 – Descrição do significado dos LEDs do MFI-PBM

6. Informações técnicas sobre o MFI-PBM

O módulo MFI-PBM executa vários tipos de comandos acíclicos previstos na norma *PROFIBUS DP* EN50170. Neste capítulo, mostraremos quais os comandos acíclicos suportados pelo mestre.

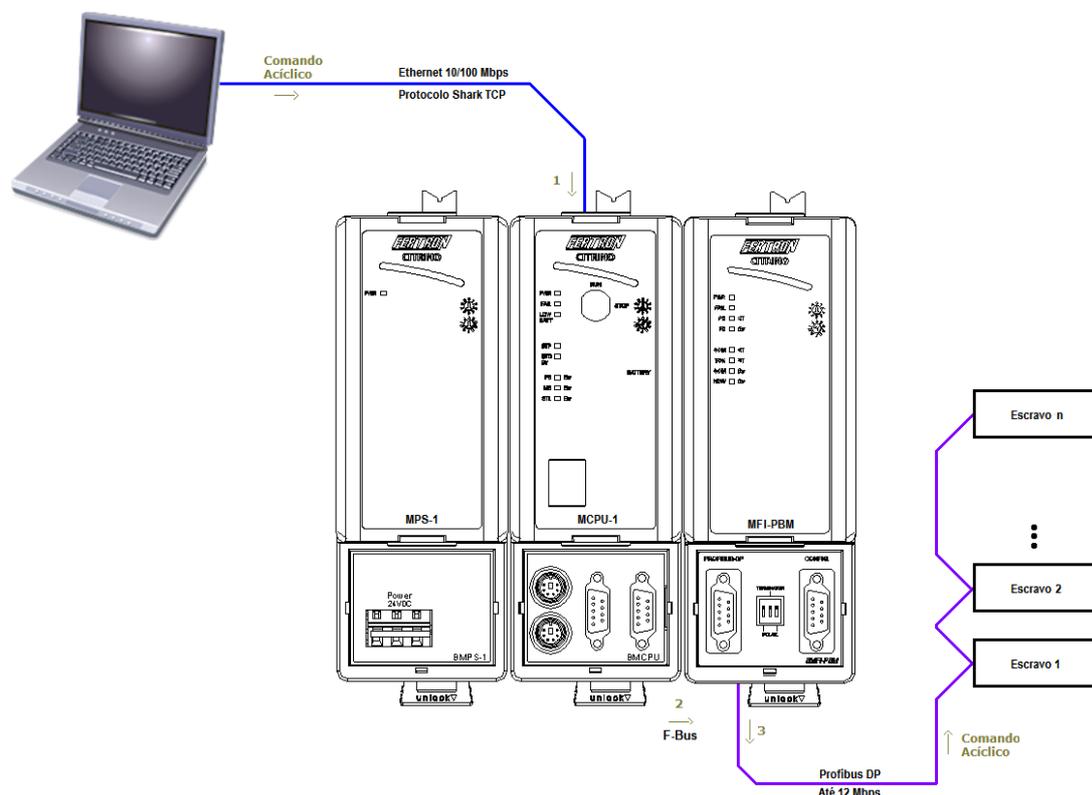


Figura 6.1 – Comandos acíclicos

Para conseguir enviar um comando acíclico é necessário que o Software suporte estes tipos de comandos. As aplicações típicas são a utilização de Softwares para gerenciamento de Ativos como FDT/DTM, EDDL, etc. Observe na Figura 6.1 como é enviado o comando acíclico.

Imagine que um comando de leitura de diagnósticos do escravo no *PROFIBUS DP* (*Get Slave Diagnostic*) seja enviado via Software.

Este comando acíclico é passado via Protocolo Shark/TCP no meio físico Ethernet (cabo azul). Este comando chega até a MCPU-1 através do caminho 1 na Figura 6.1 encapsulado no protocolo Shark/TCP. Este comando por sua vez é passado ao barramento F-Bus e chega ao módulo MFI-PBM através do caminho 2. Agora o módulo interpreta o comando *Get Slave Diagnostic* e envia ao escravo indicado via protocolo *PROFIBUS DP* no meio físico RS-485 (cabo roxo). O escravo então lê seus diagnósticos e responde ao mestre MFI-PBM. O módulo MFI-PBM por sua vez responde de volta para a MCPU-1 através do barramento F-Bus. Por último a MCPU-1 encapsula novamente no protocolo Shark/TCP a resposta do comando pedida pelo Software e a envia de volta ao microcomputador através do meio ethernet (cabo azul).

Todos os comandos acíclicos respeitarão este processo. Na Tabela 6.1 estão os comandos acíclicos atendidos pelo módulo MFI-PBM.

Comandos	Descrições
MSAC1_READ_WRITE	Lê ou escreve dados de forma acíclica em um nó especificado através de um Slot e um Índice
MSAL1M_ALARM	Lê e reconhece o alarme mais antigo de um determinado escravo
MSAC2M_INITIATE	Estabelece uma conexão classe 2 a um escravo específico (usado antes de serviços classe 2 de leitura e escrita)
MSAC2M_READ_WRITE_DATA	Requisita leitura ou escrita de dados através de uma conexão classe 2 a um escravo especificado, operando com slot e índice. Para utilizar este comando, é necessário habilitar a conexão antes (MSAC2M_INITIATE)
MSAC2M_ABORT	Aborta uma conexão classe 2
GET_SLAVE_DIAG	Obtém o diagnóstico do nó PROFIBUS
DDL_M_GLOBAL_CONTROL	Envia um comando para um ou vários escravos para execução de ações como sincronizar entradas, congelar saídas, etc...
DDL_M_SET_PRM	Envia um novo conjunto de parâmetros a um nó especificado. Este comando só pode ser utilizado se o nó em questão já estiver pré-configurado corretamente.
FDL_SEND_DATA_ACK	Permite o envio transparente de um frame SDA de tamanho variável à rede PROFIBUS
FDL_SEND_DATA_NO_ACK	Permite o envio transparente de um frame SDA de tamanho variável à rede PROFIBUS, sem reconhecimento do recebimento dos dados pelas estações remotas
FDL_SEND_DATA_WITH_REPLY	Permite o envio de dados a uma única estação remota e, ao mesmo tempo, o recebimento de dados previamente disponibilizados
FDL_REPLY_UPDATE	Atualiza os dados que serão enviados quando uma solicitação for feita por uma estação remota (por exemplo, via SRD)
FDL_SEND_TIME_SYNC	Envia uma seqüência CS à rede PROFIBUS
GET_LIVE_LIST	Obtém o status de todos os 126 possíveis nós de rede (elemento ativo/passivo/ausente)
FDL_SAP_ACTIVATE	Ativa e configura um SAP local para serviços FDL individuais do tipo SDA e SDN
FDL_RSAP_ACTIVATE	Ativa e configura um SAP remoto para serviços FDL individuais do tipo SRD
FDL_STATUS_REPLY	Obtém o estado atual de uma estação remota
GET_CFG	Lê os dados de configuração do nó especificado.
RD_INPUT	Lê os dados de processo de entrada de um nó conectado, independente de o nó estar comunicando com outro mestre ou não.
RD_OUTPUT	Lê os dados de processo de saída de um nó conectado, independente de o nó estar comunicando com outro mestre ou não.
SET_SLAVE_ADDRESS	Modifica o endereço de um nó PROFIBUS, enviando opcionalmente outros dados ao nó.

Tabela 6.1 – Descrição dos comandos acíclicos suportados pelo MFI-PBM

7. Informações adicionais

Dentre uma variedade de produtos e serviços disponíveis, os usuários podem, a qualquer momento, escolher o melhor produto e com a melhor relação custo-benefício para sua automação. PROFIBUS tem assim comprovado em milhares de aplicações, seja na automação da produção, predial ou na de processos, sucessivamente uma alta economia de gastos, um aumento na flexibilidade associado a uma maior disponibilidade dos sistemas. Estes são com certeza as principais razões que têm levado mais e mais usuários de todo o mundo a decidir em favor do PROFIBUS.

Através de um contínuo desenvolvimento a Fertron está disponibilizando o módulo MFI-PBM com várias funções, que anteriormente podiam ser implementadas somente em equipamentos importados. Para os usuários, isto traz a vantagem de contar com uma solução PROFIBUS nacional em praticamente todos os tipos de aplicações de automação.

8. Lista de abreviações:

DP

Decentralized Periphery

Periferia descentralizada (*Decentralized Periphery*)

DPM1

DP Master Class 1

Mestre DP - Classe 1: O DPM1 é o controlador programável central para o DP.

DPM2

DP Master Class 2

Mestre DP - Classe 2 : O DPM2 é um dispositivo de configuração do DP.

DTM

Device Type Manager

Gerenciador de Tipo de Dispositivos: Parâmetros e opções de um dispositivo de campo, incluindo as características definidas em um arquivo GSD, fornecidas pelo fabricante

EDD

Electronic Device Description

Descrição Eletrônica do Dispositivo

FDL

Fieldbus Data Link

Link de Dados Fieldbus

FDT

Fieldbus Device Type

Tipo de Dispositivo Fieldbus: Método independente do fabricante para descrições de um dispositivo

GSD

GerätSammlungDatei

Arquivo de Base de Dados do Dispositivo Folha de dados eletrônica do dispositivo

PA

Process Automation

Automação de Processo: Perfil PROFIBUS para automação de processo

WM

Word Memory

Variável de memória inteira no formato 16 bits utilizada pela MCPU-1