DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

Nota Técnica

Assunto: ***Programar os V570 para duas redes Modbus e replicar as MIs.***

Objetivo: Programar os V570 para duas redes Modbus e replicar as MIs..

1. INTRODUÇÃO

Temos uma rede Modbus funcionando Master/Slave, e queremos adicionar mais um V570 na rede e ter as Mis replicadas em todos os V570. Ou seja, ao inserimos um valor em qualquer V570 este valor deverá ser replicados em todos os V570.

2. PROCEDIMENTO

Temos que compartilhar as memórias do V570, mas o protocolo Modbus, mesmo tendo duas redes (vermelho e azul) Modbus o compartilhamento da tabela de endereços é a mesma para as duas redes. Isto causa um problema para a aplicação, não temos com modificar a mesmo endereço de memória para ser replicado nos V570.
Por causa de ter somente uma tabela para as duas redes, terá que ser criada uma alocação de memória conforme a figura 1 abaixo.

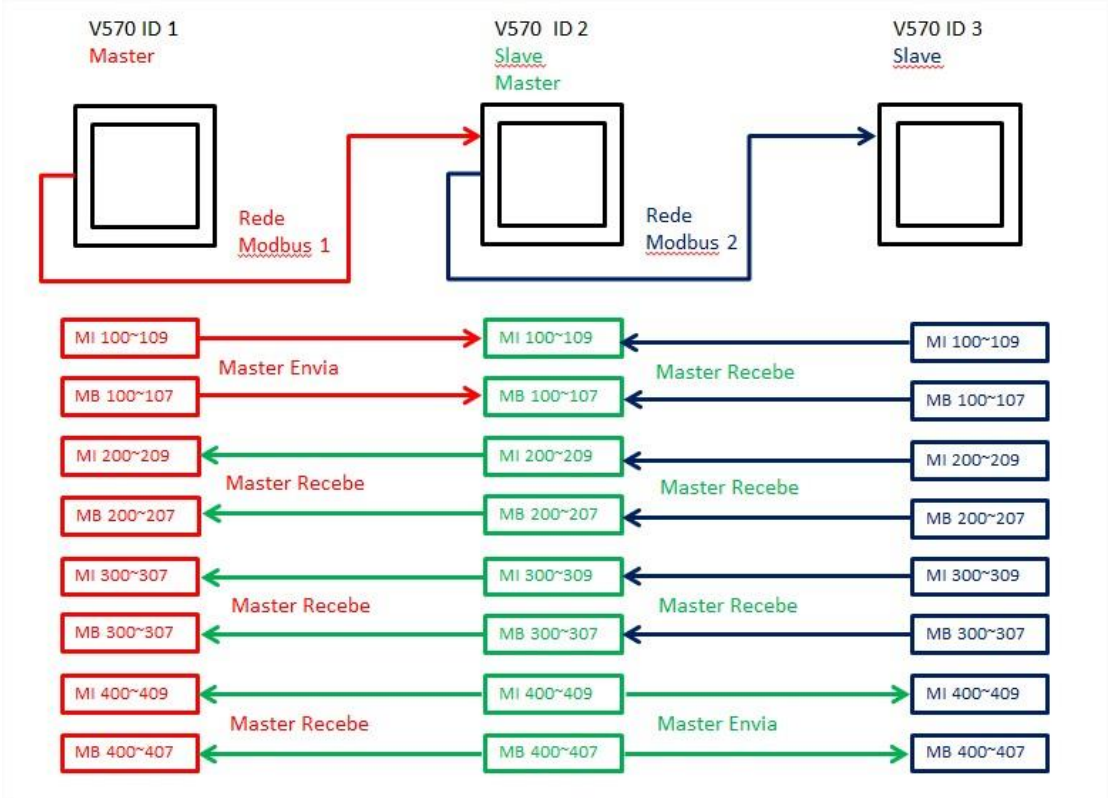


Figura 1

DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

V570	ID 1	V570	ID 2	V570	ID 3
MI 100~109	Lê/Escreve Escreve no ID 2	MI100~109	Recebe do ID 1		
MB100~107	Lê/Escreve Escreve no ID 2	MB100~107	Recebe do ID 1		
		MI200~209	Lê/Escreve		
		MB200~207	Lê/Escreve		
		MI300~309	Lê do ID 3	MI 300~309	Lê/Escreve
		MB 300~307	Lê do ID 3	MB 300~307	Lê/Escreve
MI400~409	Lê de ID 2	MI 400~409	Escreve no ID3	MI 400~409	Recebe do ID 2
MB400~407	Lê de ID 2	MB 400~407	Escreve no ID3	MB 400~407	Recebe do ID 2

Tabela 1

Conforme a tabela acima, definimos as áreas de memória.
O ID 2 concentra todas as memórias da rede Modbus, e distribui para toda a rede.
Desta forma o valor de MI400 e MB400 é compartilhado. Por causa desta solução, as memórias devem ser lidas a todo instante pelos Master, e realizar uma comparação também para atualizar as MIs 400 e MBs400.

Método de leitura e escrita das memórias

Utilizaremos um contador de 100ms e lógica de comparação.

O contador irá controlar a varredura das memórias e o comparador irá definir quando a memória será atualizada.

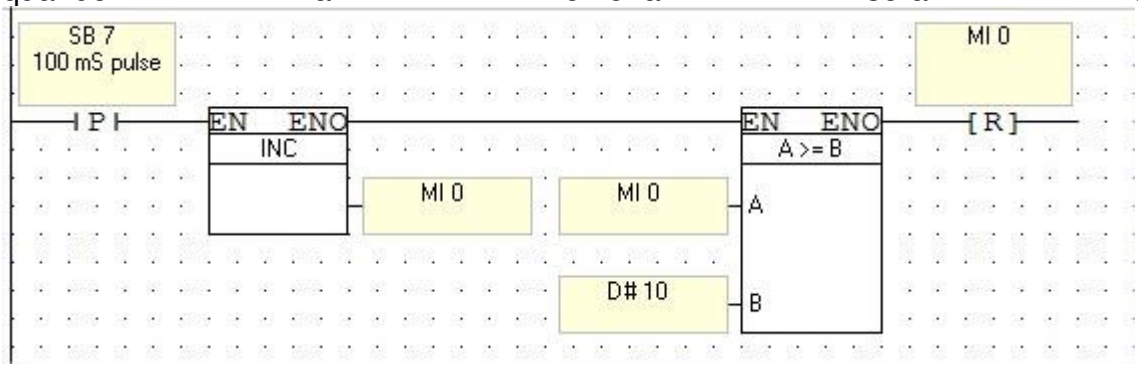


Figura 2

A MI 0 é incrementada até atingir o valor “10” e depois é zerada, e reinicia o ciclo.

Comparador passo 1 – Este comparador fica setado de “0” até “1”, fica habilitado durante 200ms.

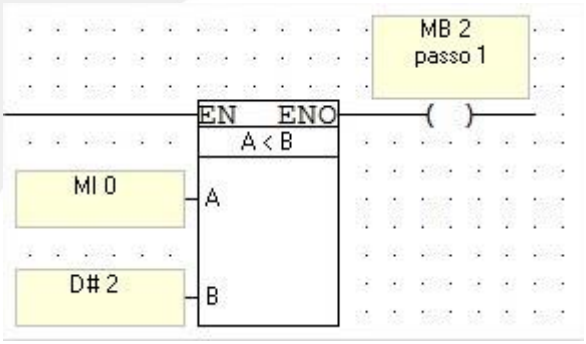


Figura 3

Comparador passo 2 – Este comparador fica setado de “2” até “3”.

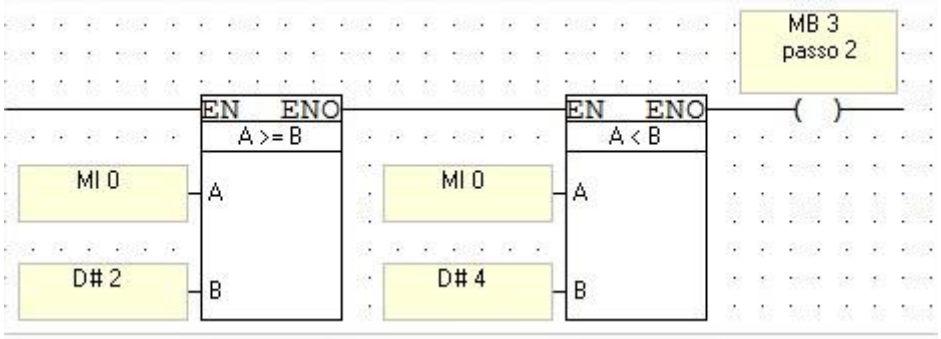


Figura 4

Esta mesma lógica é aplicada até o penúltimo passo (passo 3 MI 0>= “4” e MI 0<”6” e passo 4 MI 0 >= “6” eMI 0 <”8”)
Comparador passo 5 – Este comparador fica setado de “9” até “0”

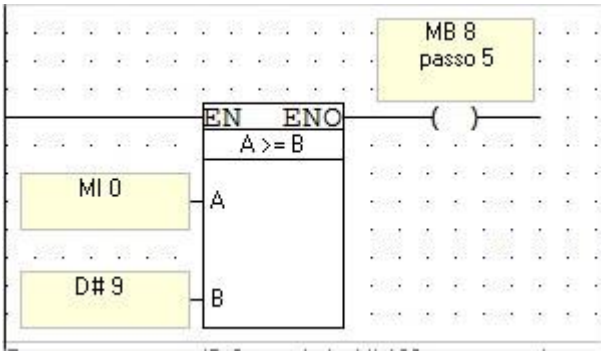


Figura 5

DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

Passo 1 - Escreve no escravo ID 3 a partir da MI 100 e mesmo valor contido na MI100 até MI109 do Master, com tamanho do vetor de 10

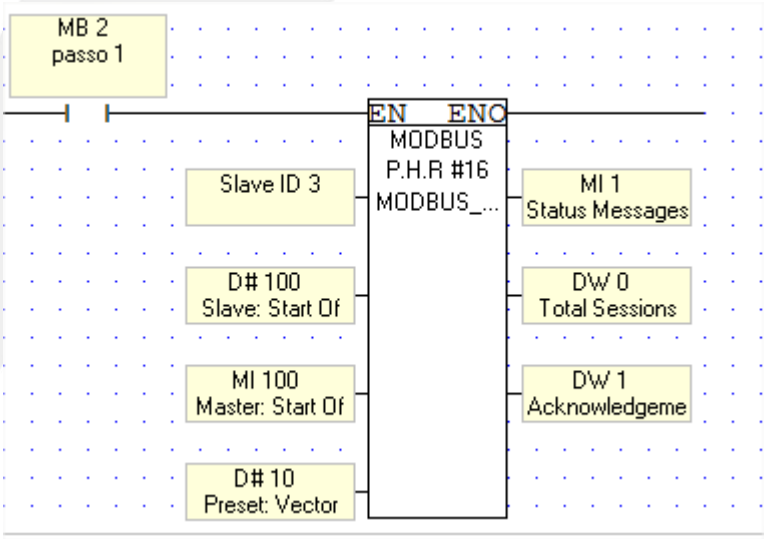


Figura 6

Passo 2 - Escreve no escravo ID 3 a partir da MI 200 e mesmo valor contido na MI200 até MI209 do Master, com tamanho do vetor de 10

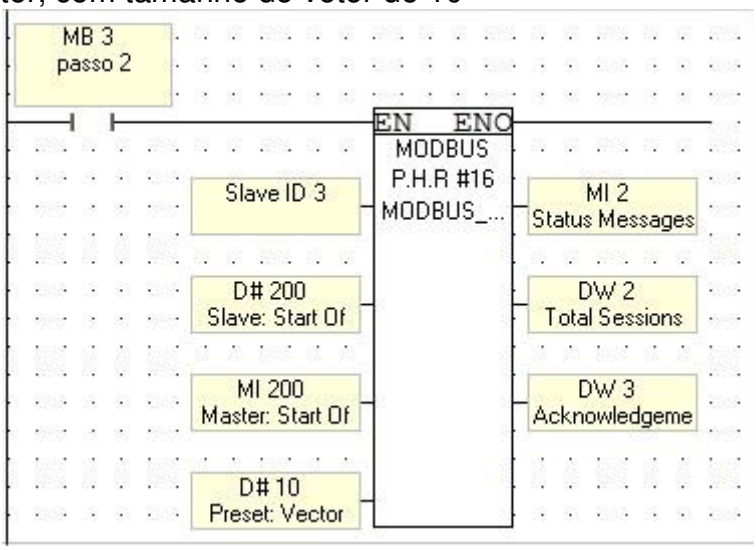


Figura 7

DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

Passo 3 - Lê do escravo ID 3 a partir da MI 300 e escreve no Master a partir da MI 300, com tamanho do vetor de 10

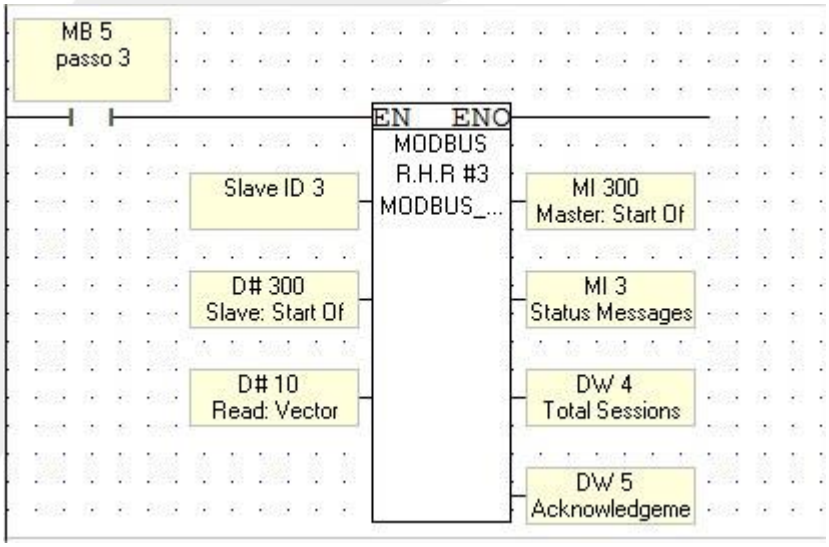


Figura 8

Passo 4 - Lê do escravo ID 3 a partir da MB 300 e escreve no Master a partir da MB 300, com tamanho do vetor de 16.

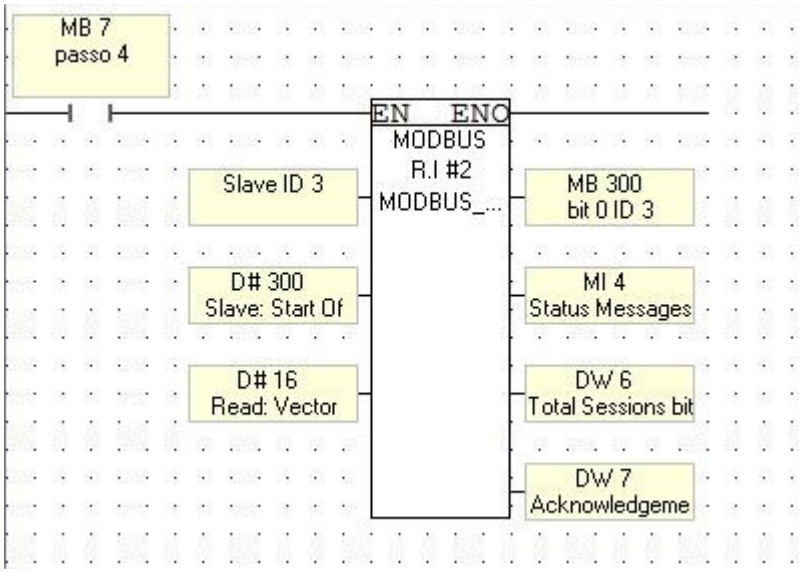


Figura 9

Passo 5 - Escreve no escravo ID3 a partir do MB 400 o mesmo status de MB 400 do Master, tamanho do vetor 16.

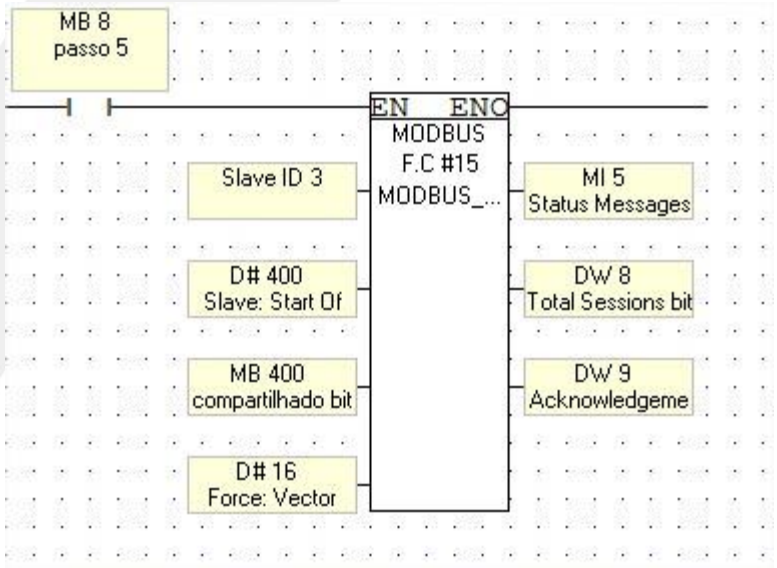


Figura 10

Com as variáveis estão sendo atualizadas a todo instante, devemos verificar qual a última atualização para ser escrita no MI 400. Lembrando que temos 3 MIs que podem atualizar a MI400.

Método para atualizar as variáveis

Novamente iremos utilizar um contador para controlar a atualização. Este contador conta até 12, ou seja, a variável é atualizada cada 600ms.
Observe a lógica abaixo, temos um contador (MI20) em ciclo que conta até 12 pulsos de 100ms e depois é zerado.

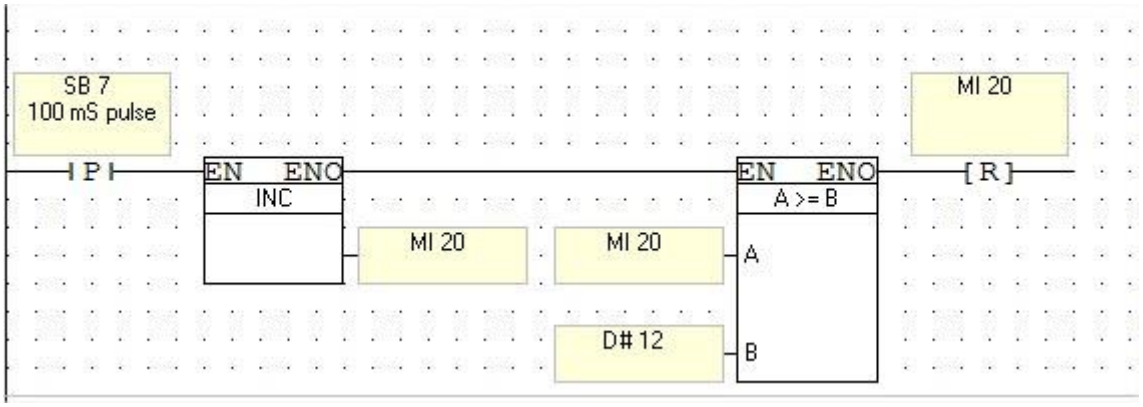


Figura 11

DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

Logicas de comparação, quando o contador MI20 estiver abaixo de “6”, o MB20 estará setado.

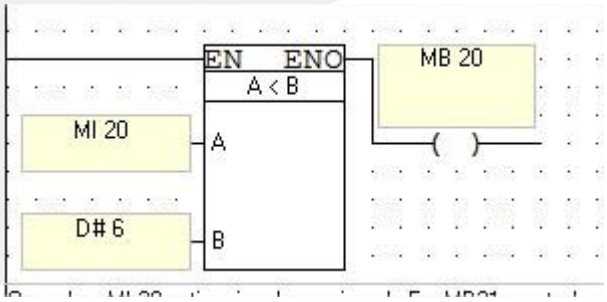
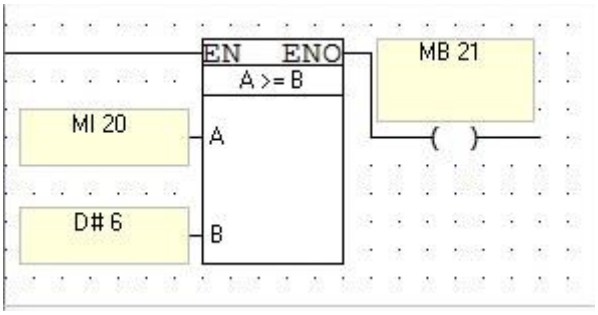


Figura 12

Quando o contador MI20 estiver acima ou igual a 6, o MB21 estará setado.



Com pulso negativo de MB 20, fazermos um “Store Direct” da MI100 para a MI 150. Que é o valor atual.

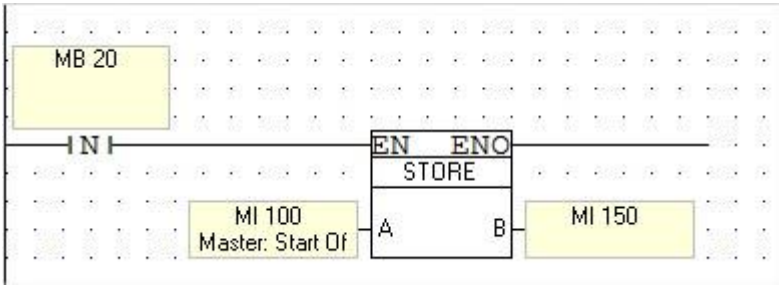


Figura 13

Com o pulso negativo de MB 21, fazemos um “Store Direct” da MI100 para a MI151. Que é o valor atual após 600ms da última gravação.

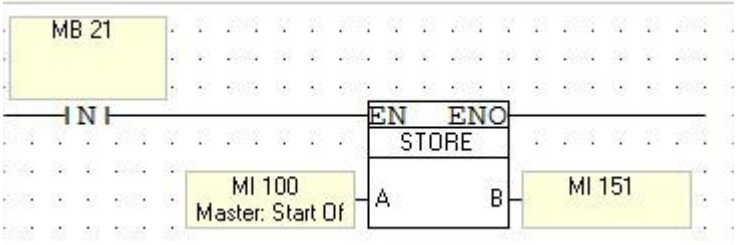


Figura 14

Comparação das variáveis. Verifica se houve alteração das variáveis, que sempre são comparadas em um intervalo de 600ms. Esta mesma lógica se aplica a demais MI200 e 300.

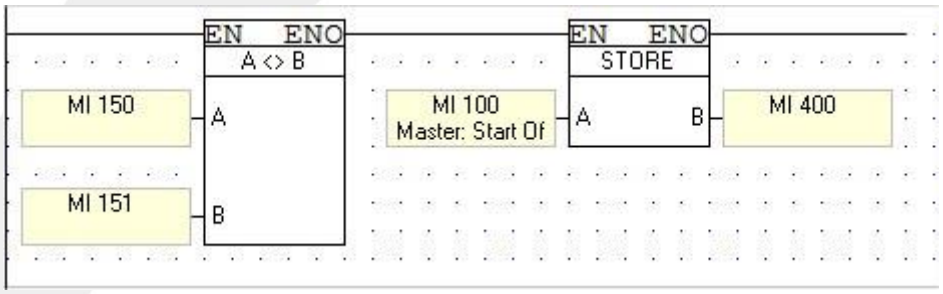


Figura 15

Para os bits também utilizamos o mesma lógica.

Para um pulso vindo do MB 100 (ID 1), MB 200 (ID 2) ou MB 300 (ID 3) seta a MB400.

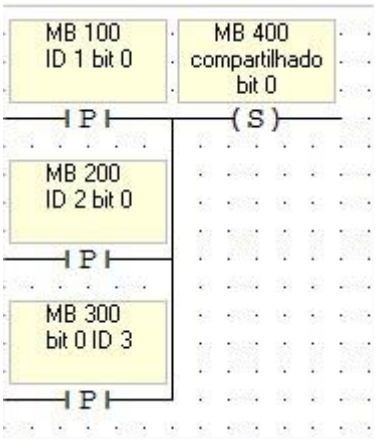


Figura 16

Para um pulso vindo do MB 101 (ID 1), MB 201 (ID 2) ou MB 301 (ID3) reseta a MB400.

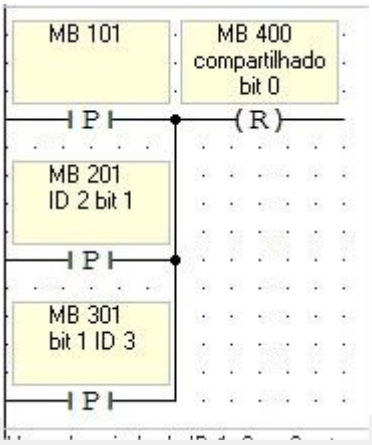


Figura 17

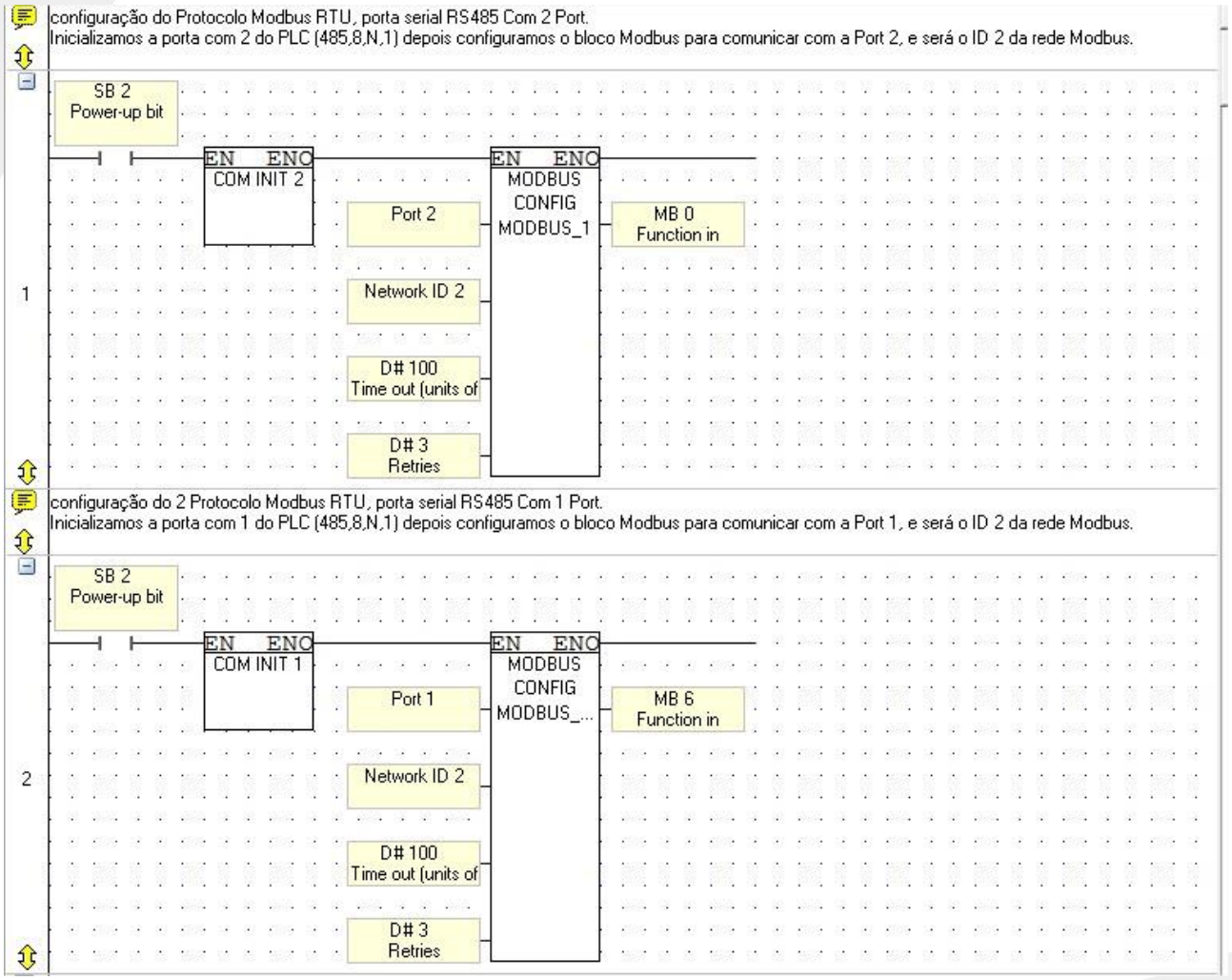
Podemos notar que para um bit compartilhado, devemos utilizar dois bits de controle.

Para escrever as variáveis a serem compartilhadas.

No ID 1 escrevemos na MI100 e temos o retorno através da MI400. Setamos a MB100 e resetamos pela MB101 e temos o retorno através da MB400.
No ID 2 escrevemos na MI200 e temos o retorno através da MI400. Setamos a MB200 e resetamos pela MB201 e temos o retorno através da MB400.
No ID 3 escrevemos na MI300 e temos o retorno através da MI400. Setamos a MB300 e resetamos pela MB301 e temos o retorno através da MB400.

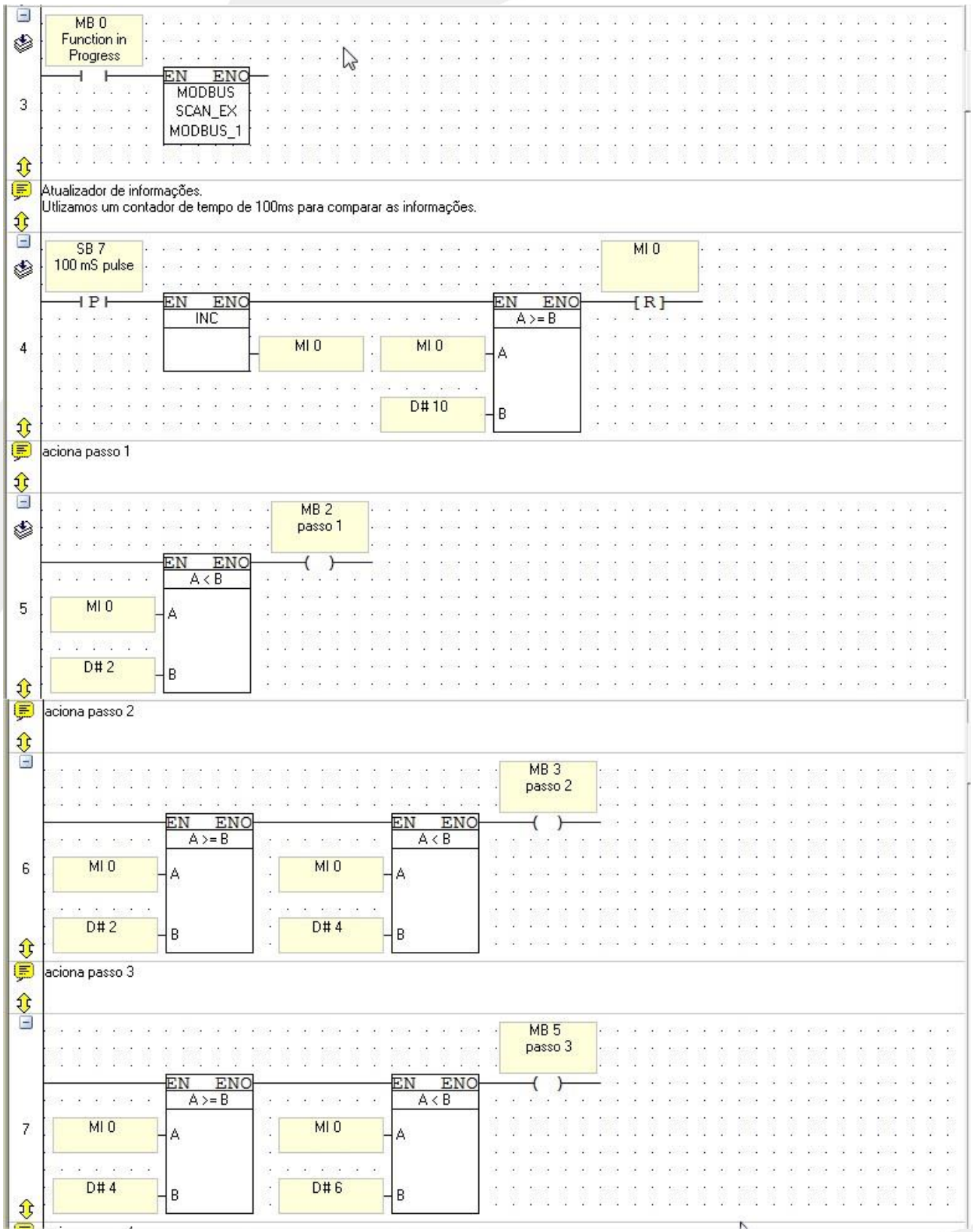
Programa para o ID 2 (Master/Slave)

Segue o programa para a ID 2



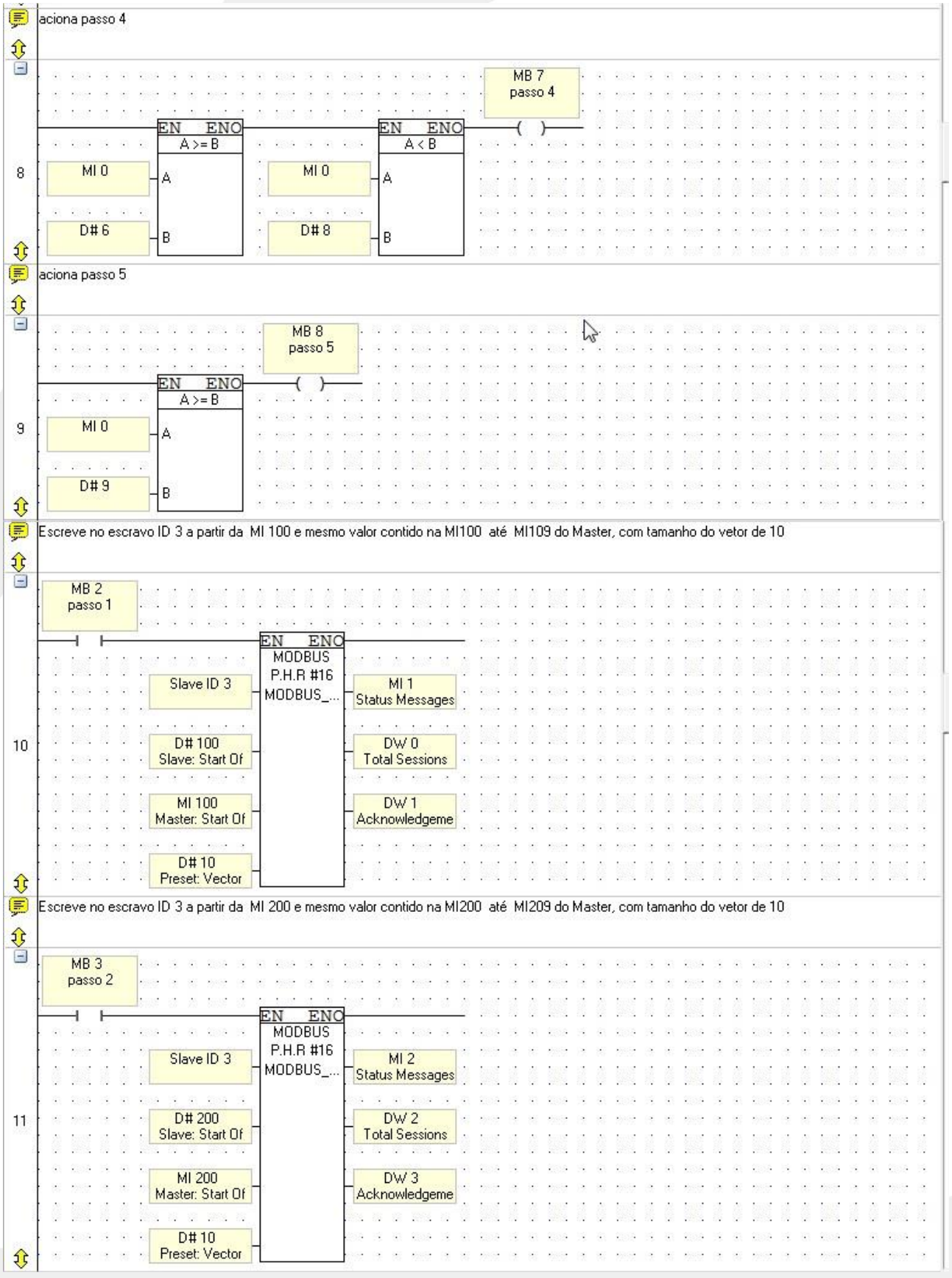
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



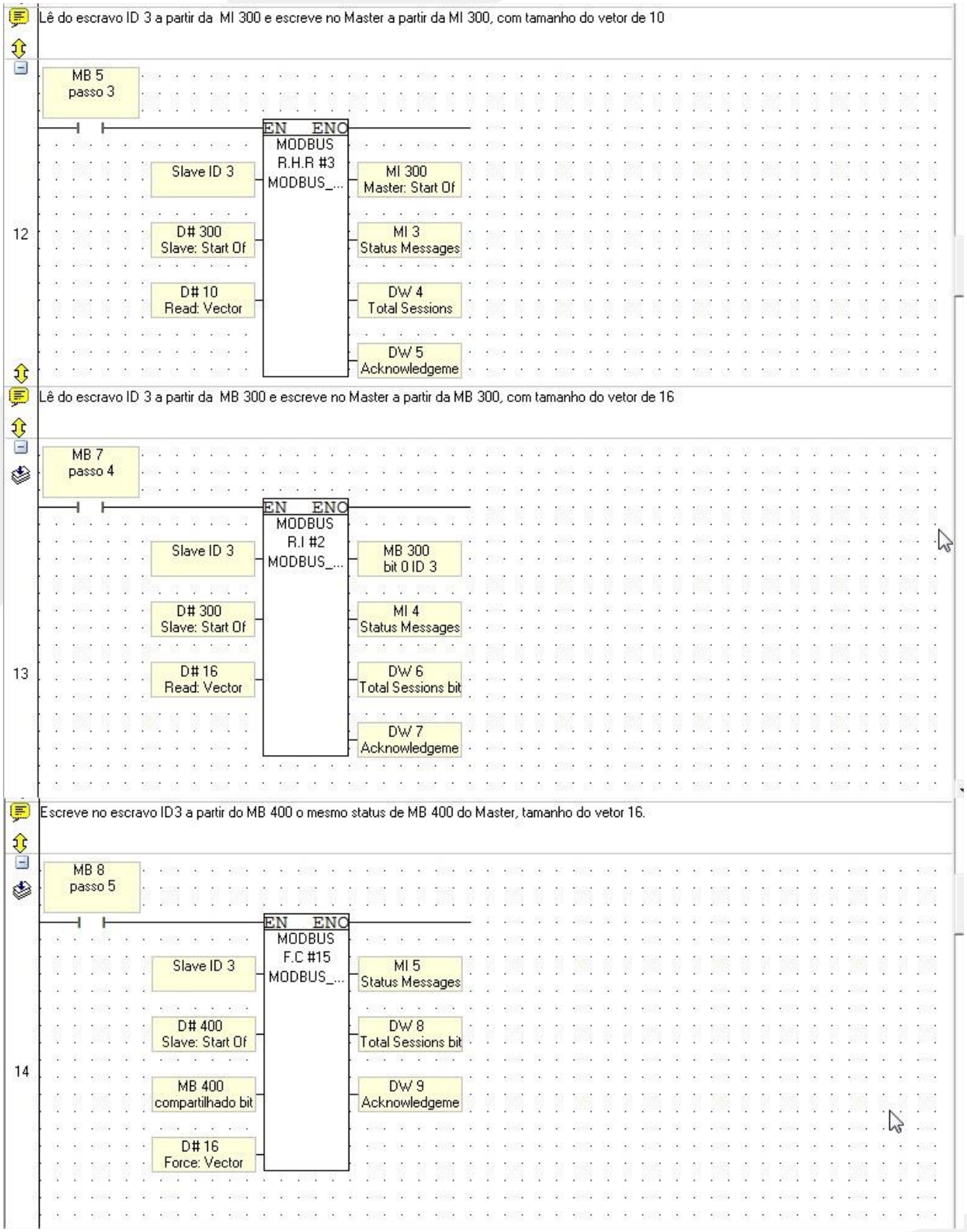
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



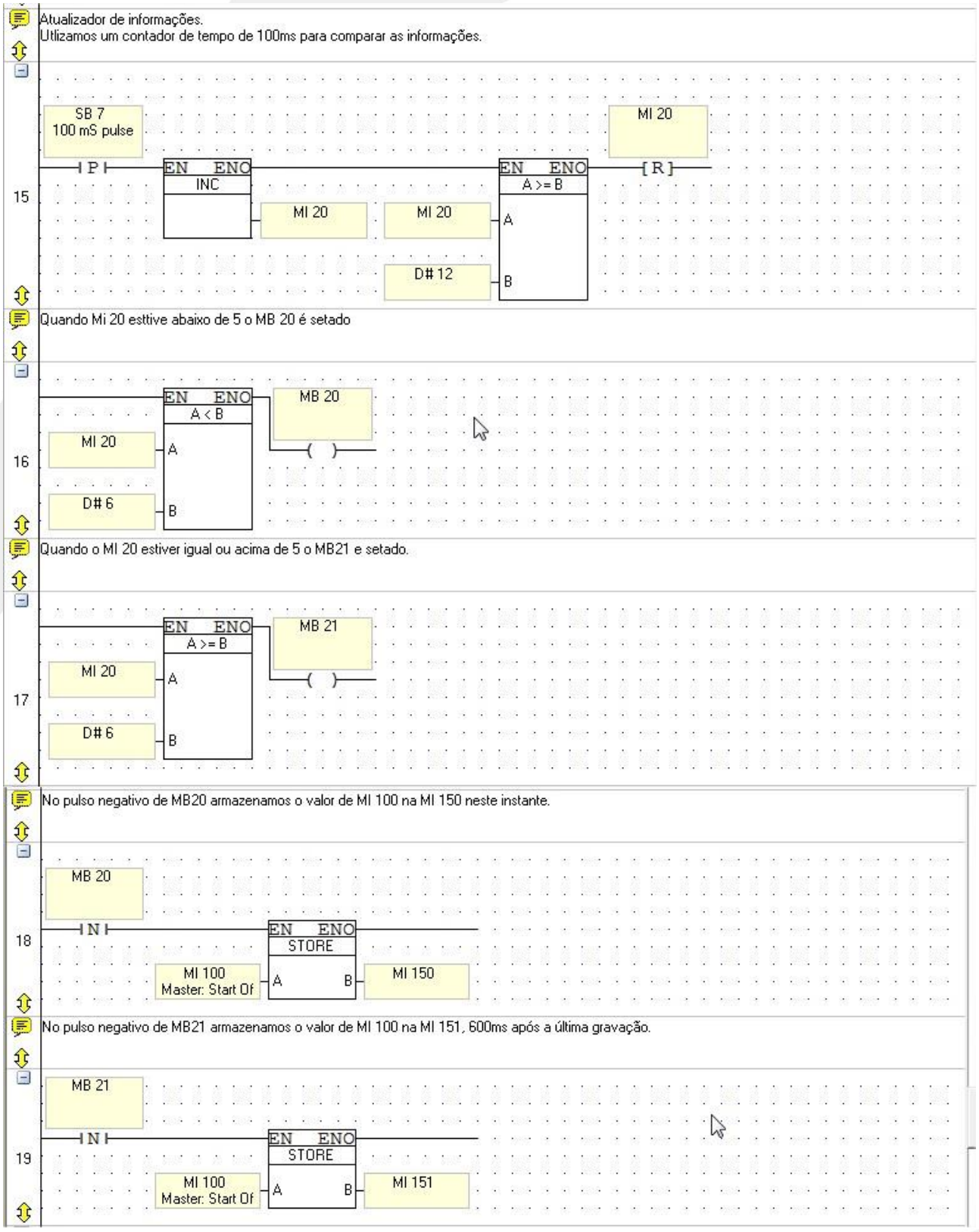
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



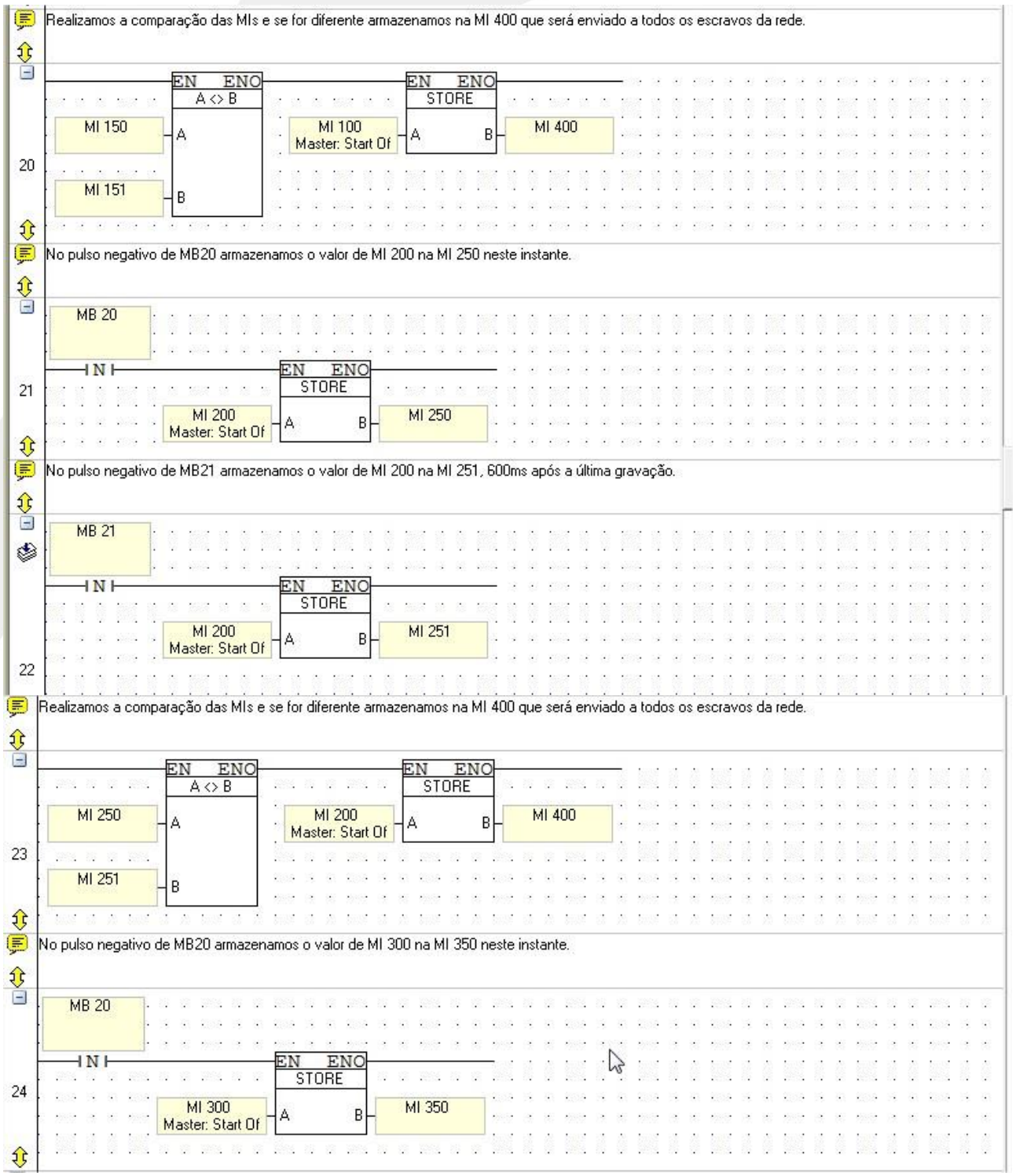
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



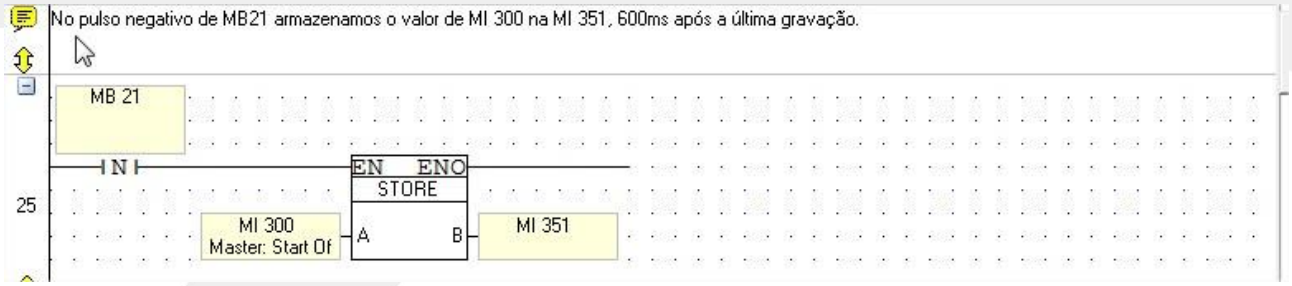
DAKOL
Soluções em Automação

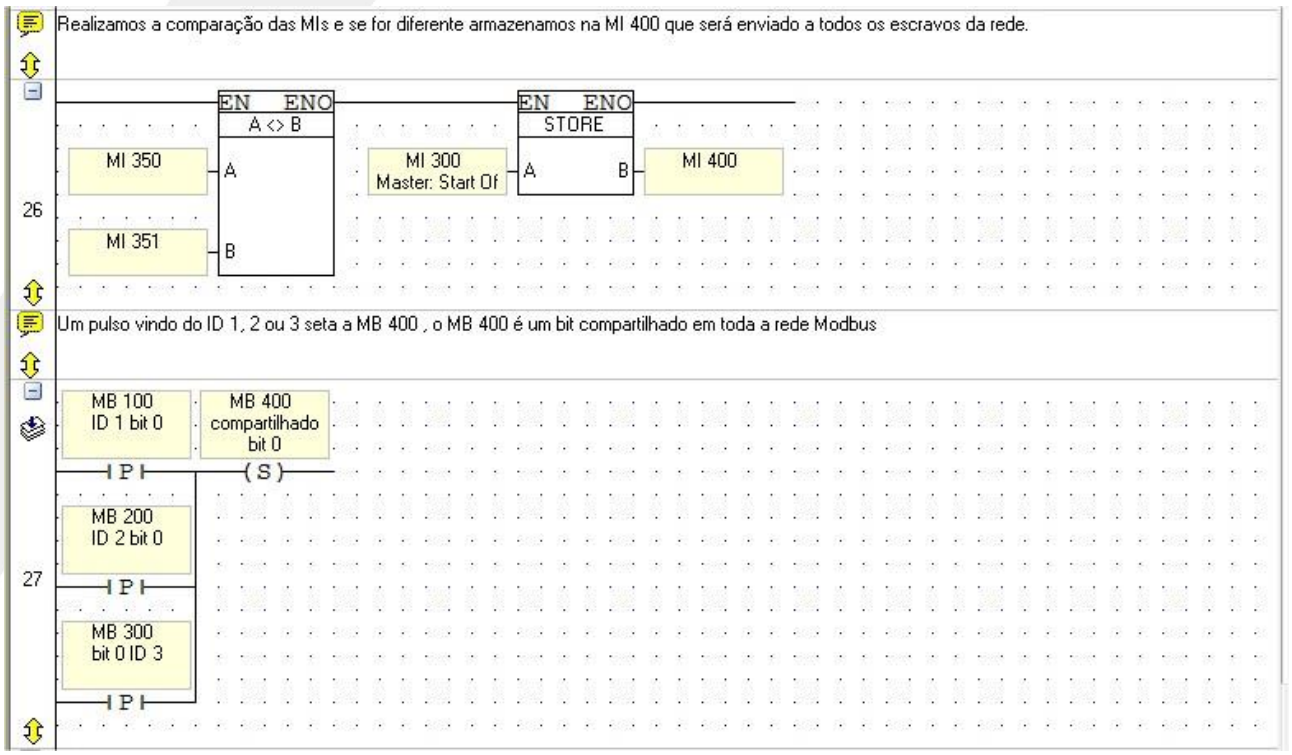
Data: 02/09/2013 Versão 0
Autor: JNohara



DAKOL
Soluções em Automação

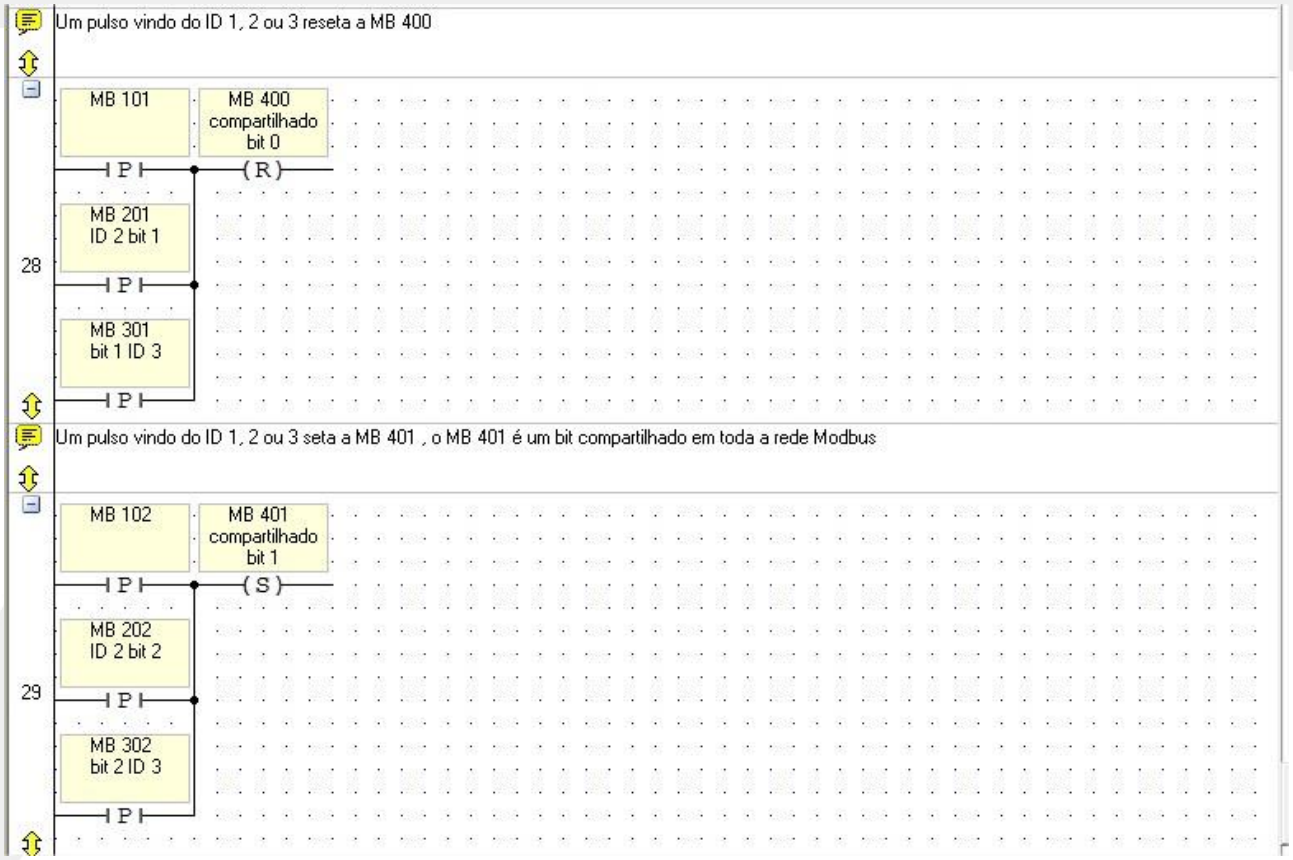
Data: 02/09/2013 Versão 0
Autor: JNohara





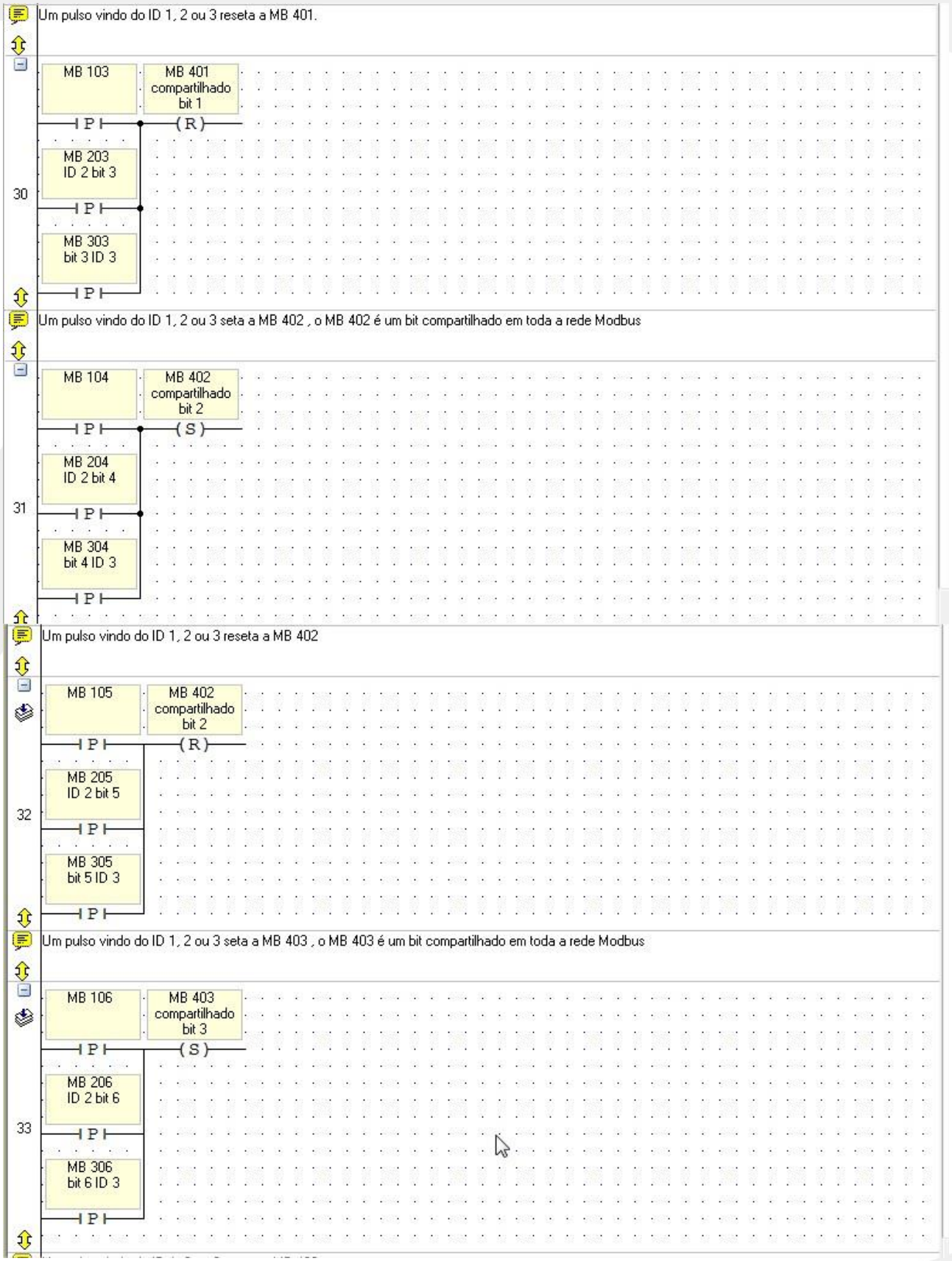
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



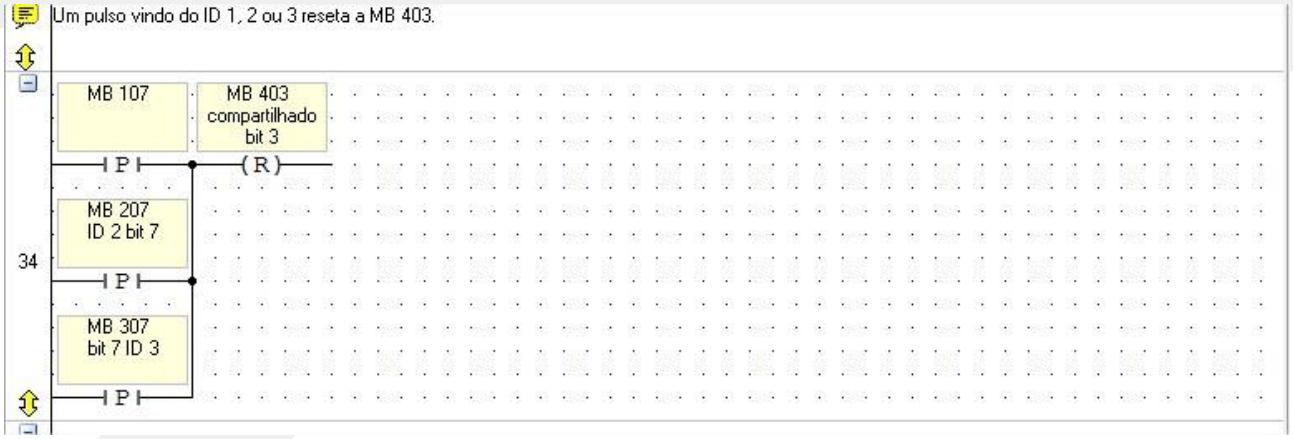
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

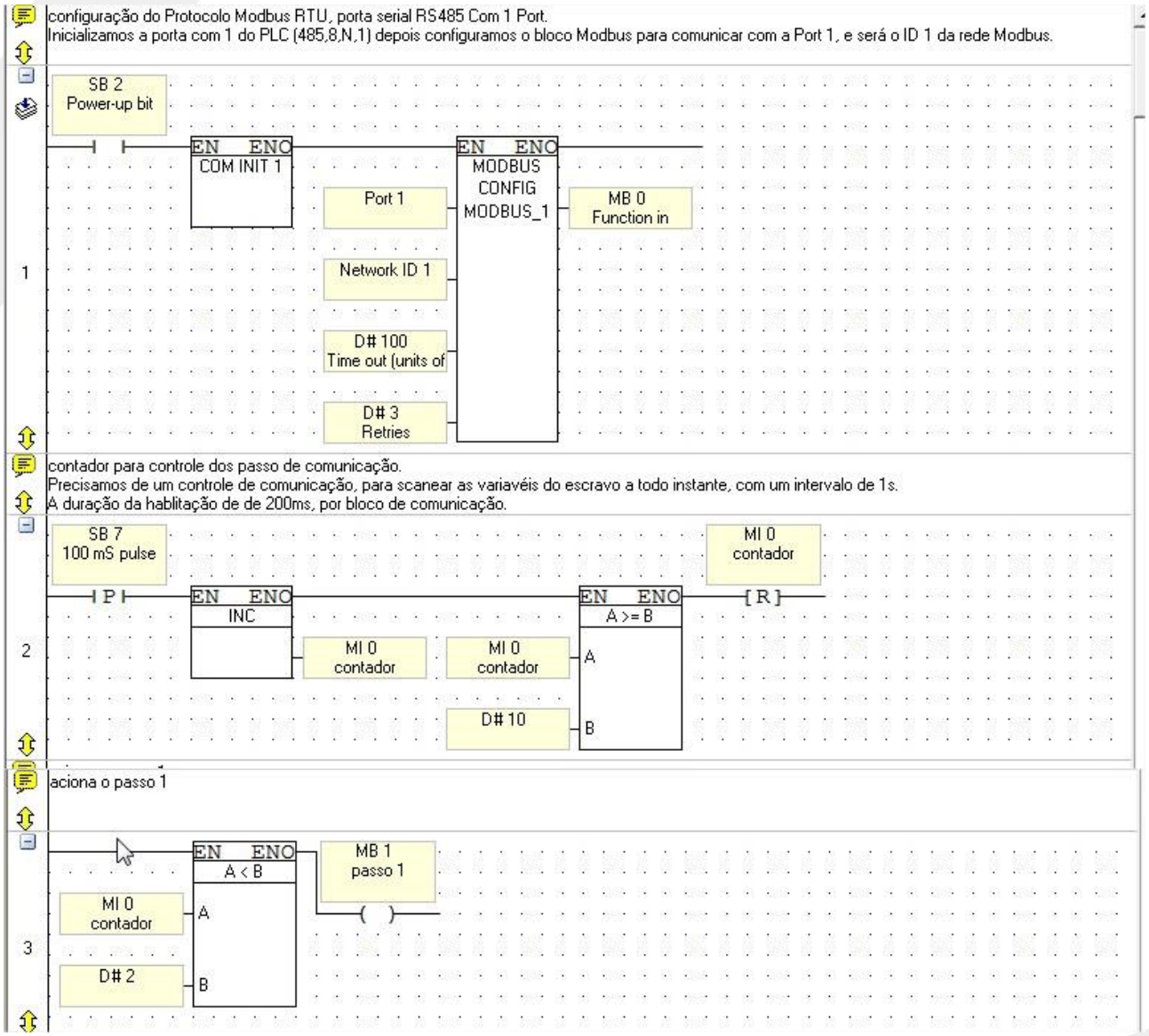


DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

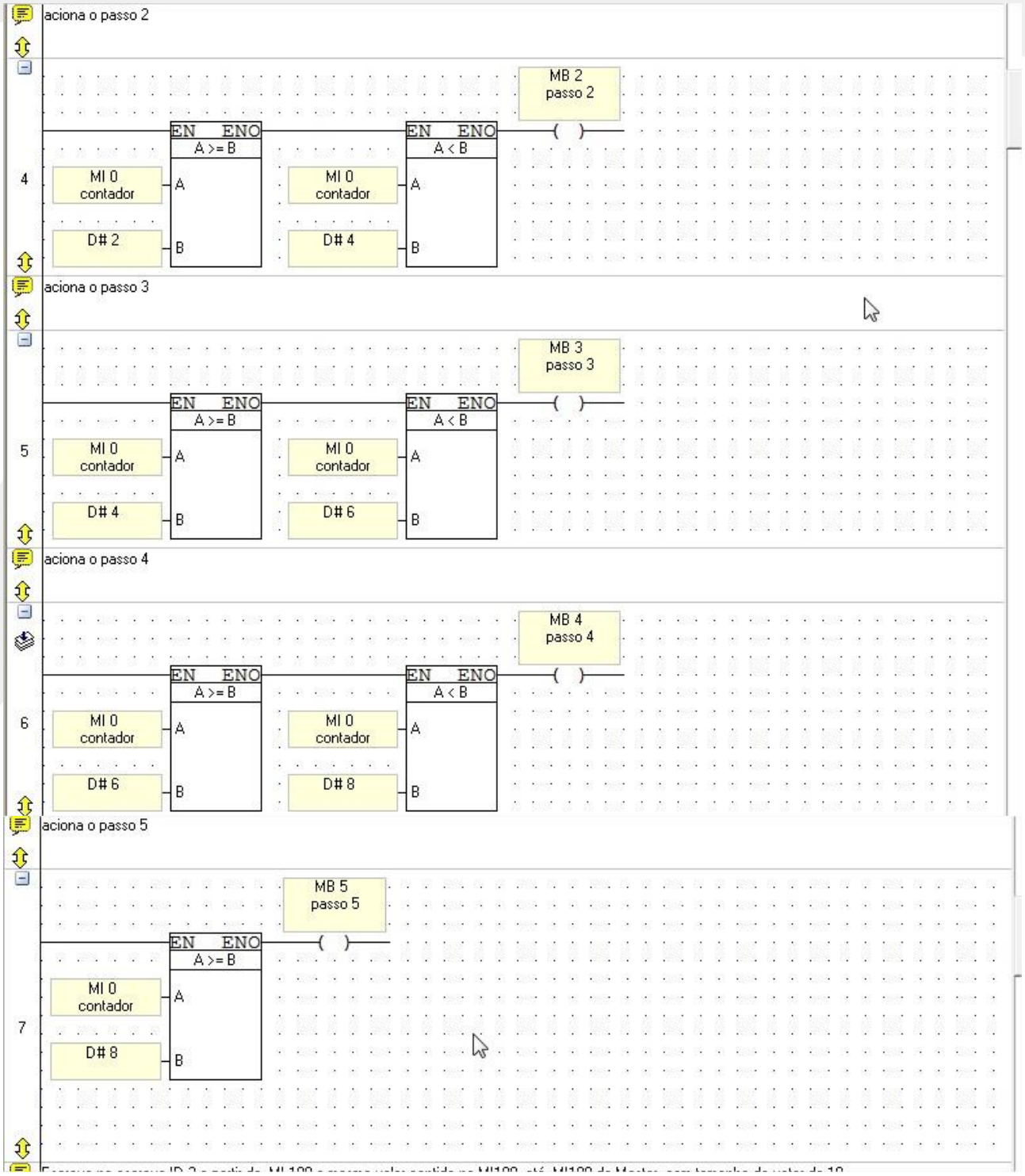


Programa para o ID 1 (Master)



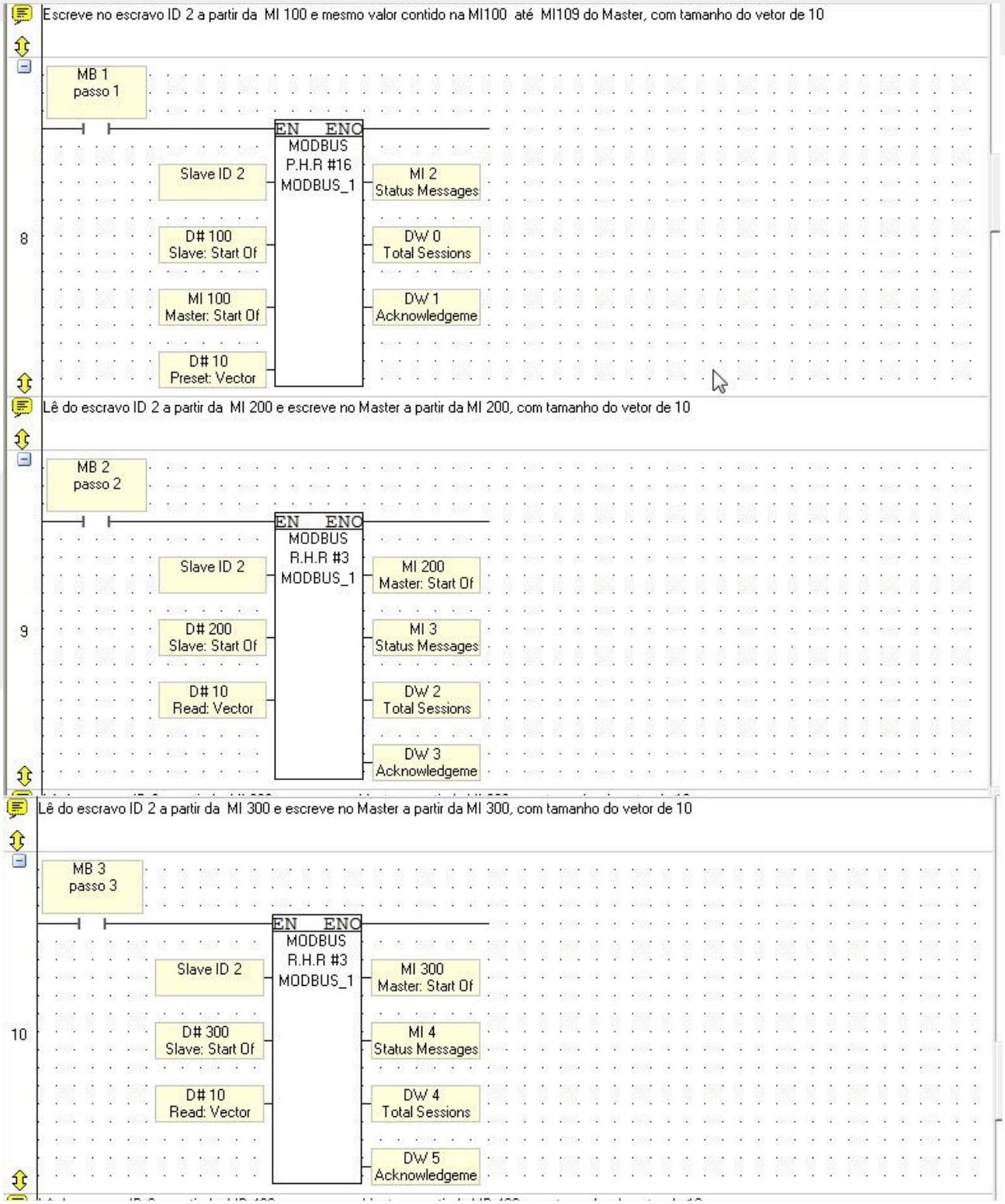
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



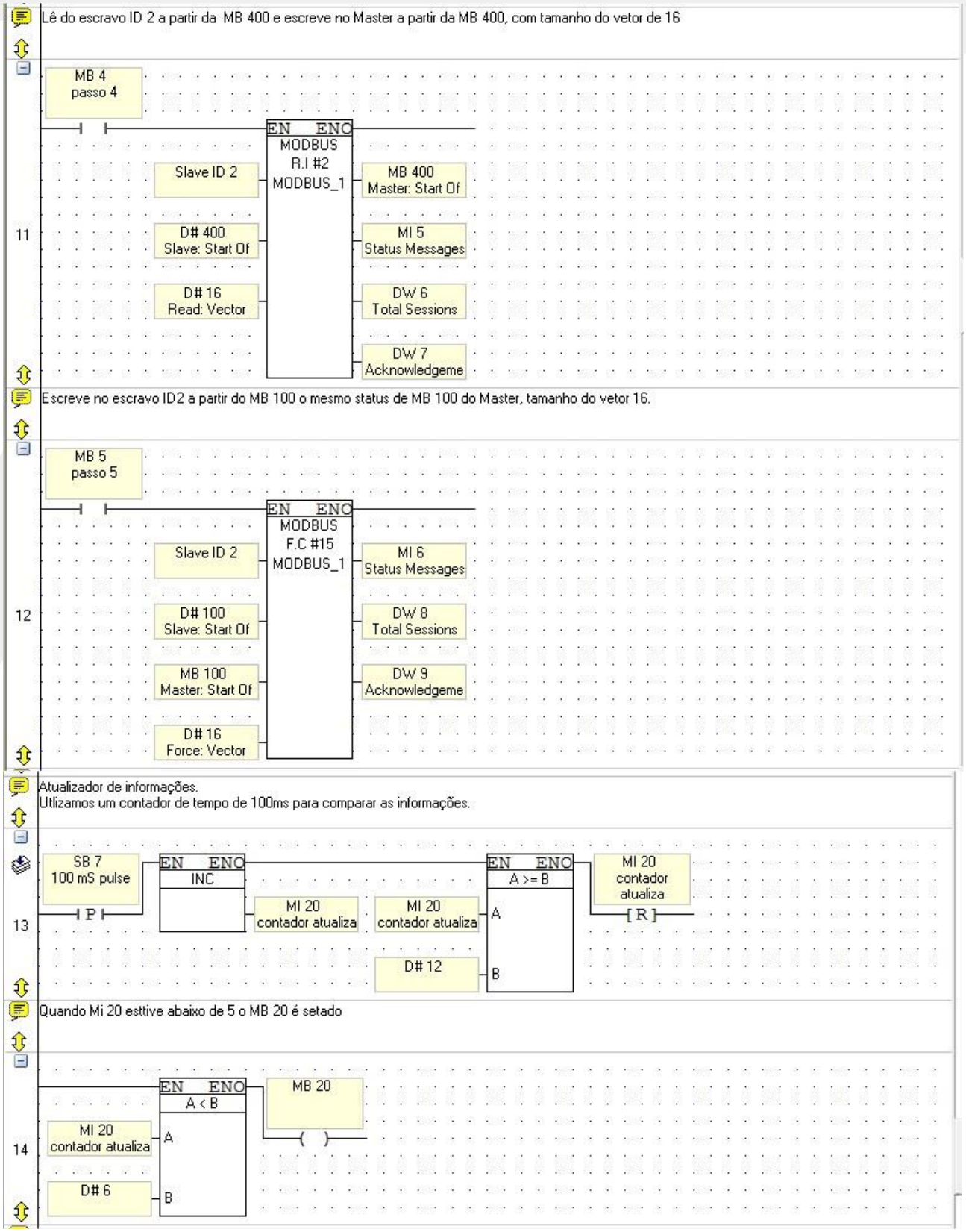
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



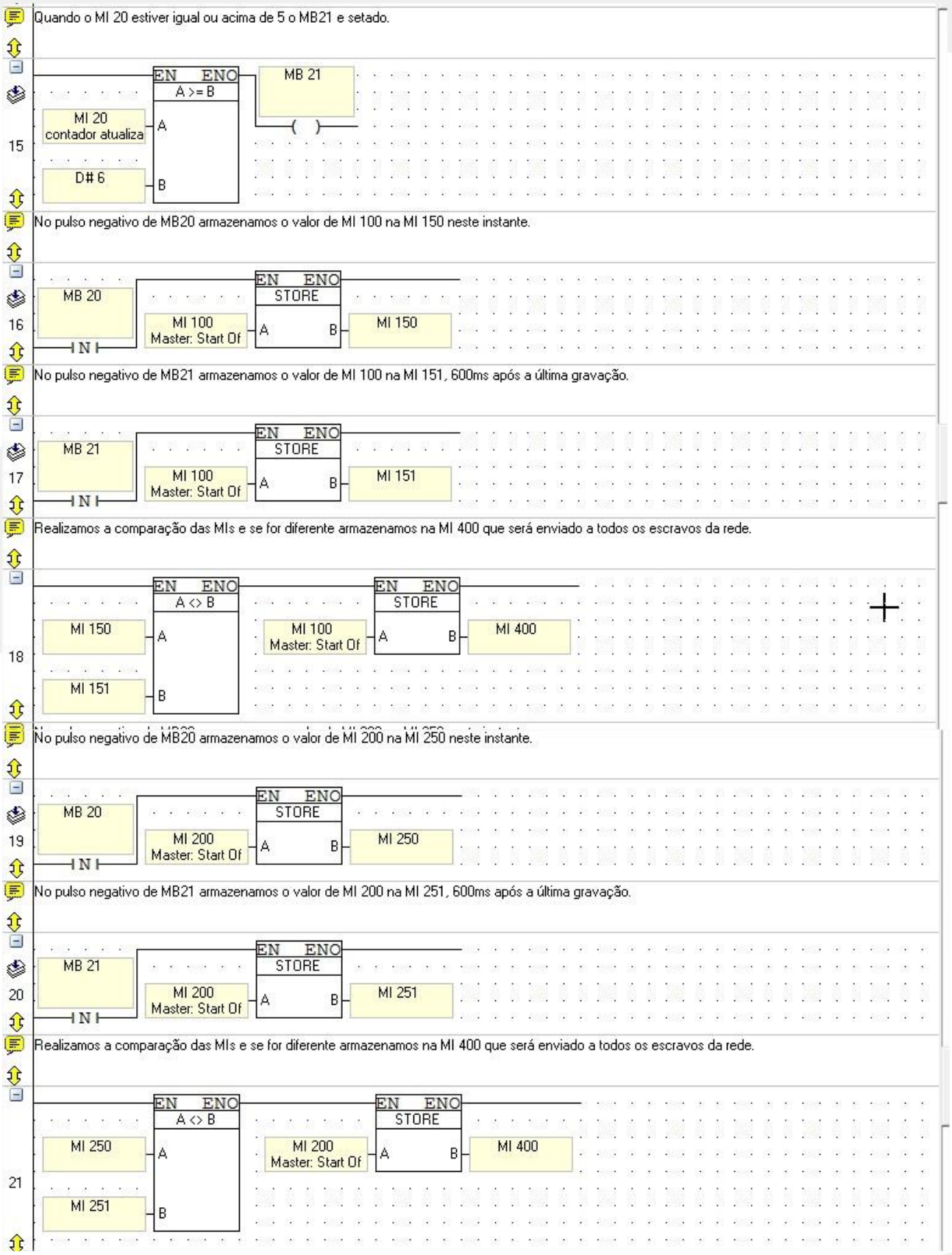
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



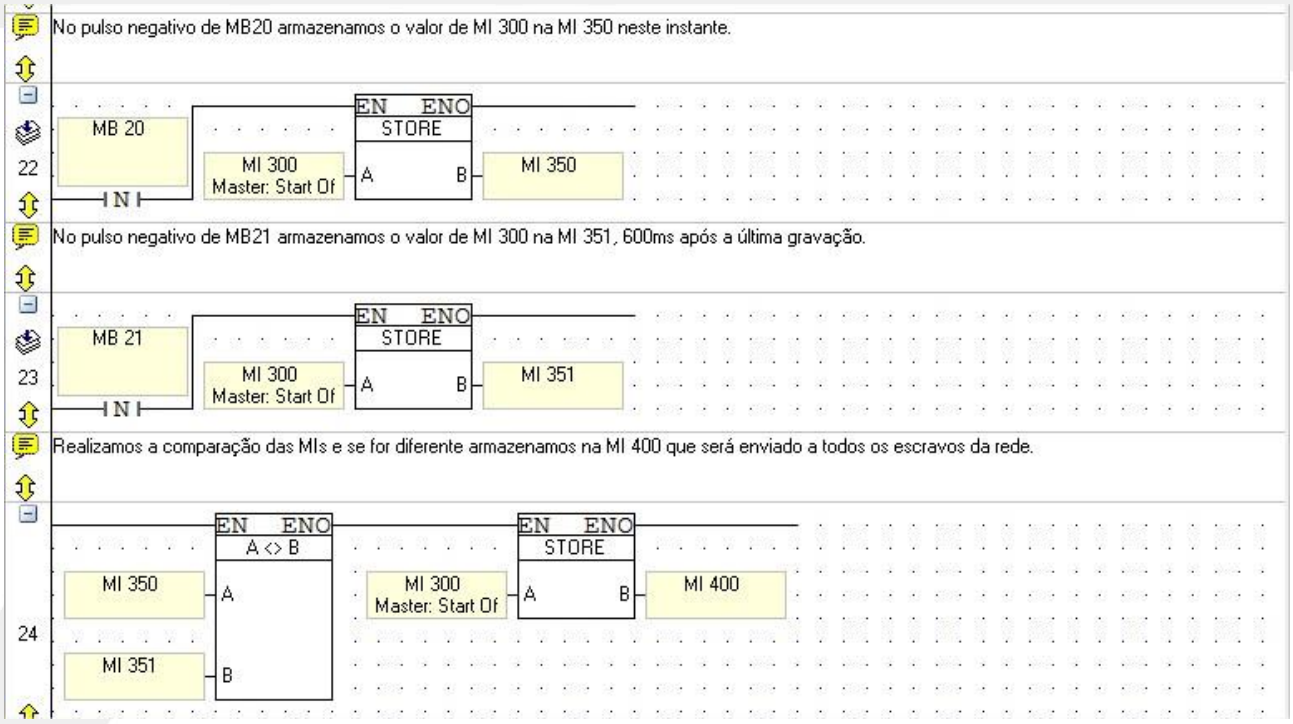
DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

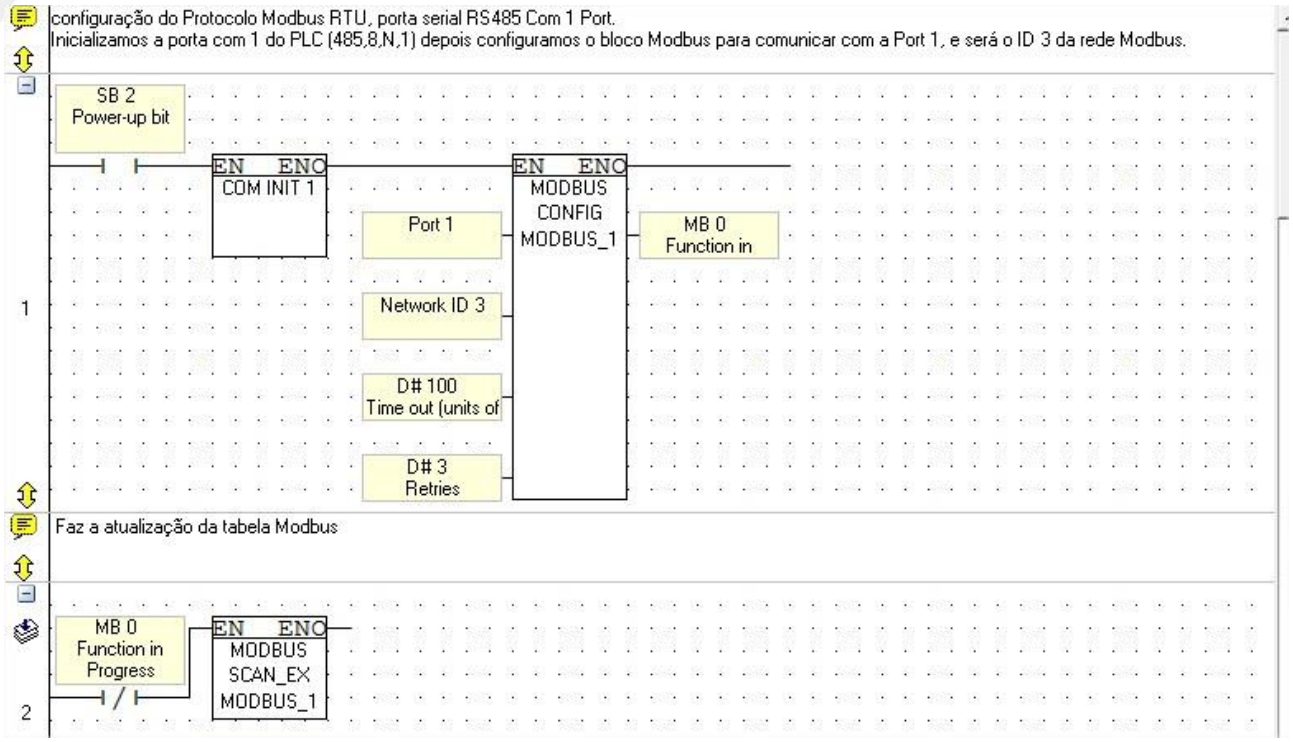


DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara

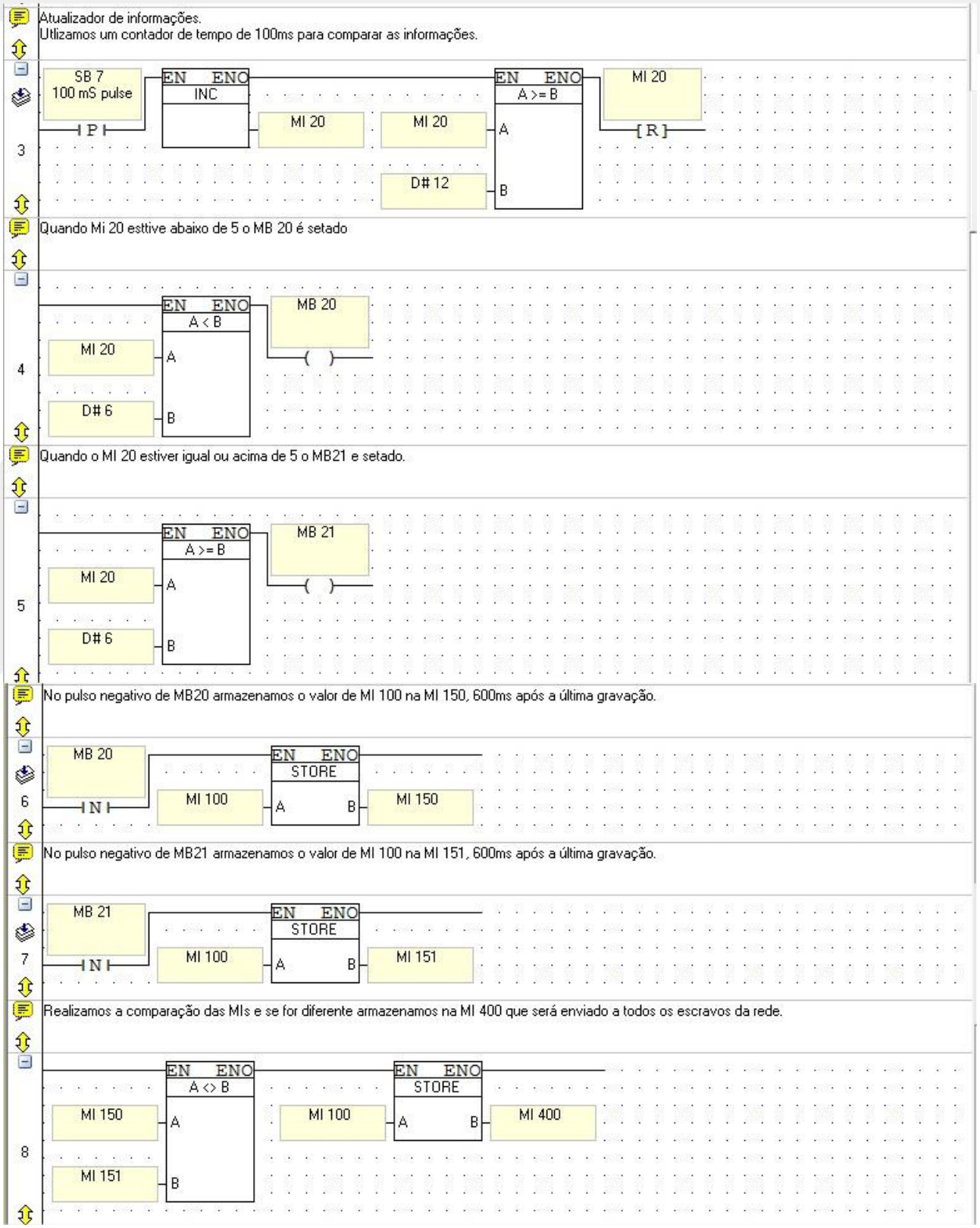


Programa para o ID 3 (Slave)



DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara



DAKOL
Soluções em Automação

Data: 02/09/2013 Versão 0
Autor: JNohara