



Comércio e Manutenção de Produtos Eletrônicos

Manual CP-WS1

**Mapeamento de memória e conexões do
Controlador CP-WS11/4DO4DI-USB**

PROXSYS

Versão 1.5

Setembro -2018

Controlador Industrial CP-WS1

1- Configurações de Hardware

O controlador CP-WS11/4DO4DI-USB, foi desenvolvido para atender aos requisitos de pequenas aplicações de controle industriais envolvendo entradas e saídas digitais e também entradas analógicas.

Código CLP	Entradas digitais CC 12 – 30 Vcc PNP	Saídas digitais Transistor	Saídas digitais Rele	Sensor DS18B20
CP-WS11/4DO4DI-USB – 2 RELE	4	0	2	0
CP-WS11/4DO4DI-USB – Saída Transistor	4	4	0	0
CP-WS11/4DO4DI-USB – 4 RELE	4	0	4	2

Tabela 1 – Configuração do CP-WS11/4DO4DI-USB

Outras configurações estão disponíveis sob consulta. A programação do controlador é realizada através de linguagem ladder através do editor ladder SCPws1.

No SCPws1, deve-se escolher no menu Arquivo >> Configurações de hardware o controlador programável CP-WS11/4DO4DI-USB, como mostra a figura 1.

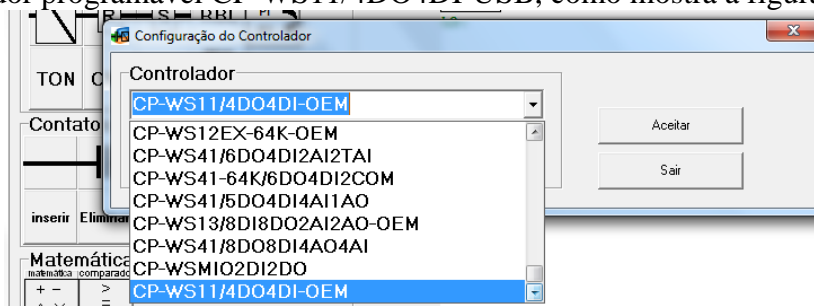


Figura 1 – Seleção do CP-WS11/4DO4DI-USB no software SCPws1

A figura 2 mostra o aspecto do CP-WS11/4DO4DI-USB com 4 saídas a transistor que é uma placa que pode ou não ser montada em suporte para trilho TS-35.



Figura 2 – Controlador CP-WS11/4DO4DI-USB – Saídas Transistor

A figura 3 mostra o CP-WS11/4DO4DI-USB com 2 saídas a rele montada em suporte para trilho TS-35.



Figura 3 – Controlador CP-WS11/4DO4DI-USB – Saídas Rele

A figura 4 mostra o CP-WS11/4DO4DI-USB com 4 saídas a rele montada em suporte para trilho TS-35.

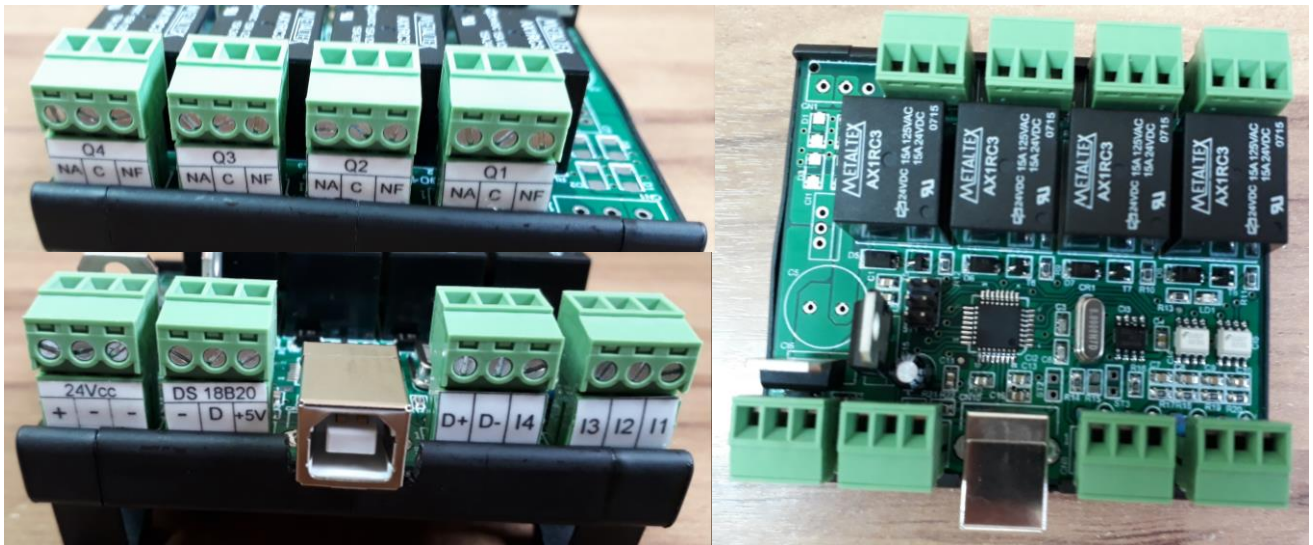


Figura 4 – Controlador CP-WS11/4DO4DI-USB – 4 Saídas Rele sensor DS18B20

2 – Mapeamento de memória

O controlador CP-WS11/4DO4DI-USB conta com operadores do tipo R, M, T, C, I, Q, L. As quantidades e funções de cada um destes elementos é mostrada na tabela 2.

Tipo de elemento	Quantidade	Faixa	Função
Operador R	50	R1 a R50	Rele auxiliar R22-R23-R24 gerador pulsos motor passo
Operador M	2	M1-M2	Memória Inteira
Operador M	3	M3-M5	Sensor temperatura DS18B20 M3=quantidade(até 2 sensores) M4 e M5= Val. Temperatura
Operador M	4	M6-M9	Memória Inteira
	1	M10	Contagem rápida entrada I1
	1	M11	Contagem rápida entrada I2
	1	M12	Contagem rápida entrada I3
Operador M	1	M13	Contagem rápida entrada I4
Operador M	3	M14-M16	Memória Inteira
Operador M	1	M17	Largura pulso PWM – Q4
Operador M	1	M18	Largura pulso PWM – Q3
Operador M	1	M19	Largura pulso PWM – Q1
Operador M	1	M20	Memória Inteira
	15	M21-M35	Memória Inteira retentiva EEPROM . Obs.1 M27 – Frequência PWM
Operador M	44	M36-M80	Memória inteira uso geral
Operador M	1	M150	Memória inteira controle Modbus porta USB não existe fisicamente só pra receber status usa ou não na E28
Operador L	1	L1	Memória Long – 32 bits uso geral
Operador L	1	L2	Memória Long – 32 bits Contagem rápida I1
Operador L	3	L3-L5	Memória Long – 32 bits uso geral
Operador T1	10	T1-T10	Contato Saída temporizador
Operador C1	10	C1-C10	Contato saída contador
Operador CR	10	CR1 – CR10	Bobina reset contador
Operador Q – Saída Rele	2	Q1- Q4	Saída Digital – Transistor ou rele dependendo modelo
Operador Q	1	Q5	Led uso geral placa
Operador I	2	I1-I4	Entrada Digital PNP

Tabela 2 – Mapa de memórias do CP-WS11/4DO4DI-USB

Obs. 1 : As memórias de M21 ao M35 são recuperadas da memória EEPROM quando o controlador é inicializado. Podem ser alterados com um programa disponível **CLP_proxsys_util_V6.exe** como mostra a figura 3.

3 – Conexões elétricas

As entradas digitais do CP-WS11/4DO4DI-USB podem receber sinal de tensão contínua na faixa de 12 a 30 Vcc. A alimentação elétrica também é em tensão contínua em 24 Vcc ou 12Vcc . As entradas I1 a I4 são referenciadas ao negativo da fonte utilizada (GND). Os terminais +Vcc junto as saídas digitais Q1, Q2, Q3 e Q4 devem ser alimentados com o positivo da fonte de alimentação do controlador no caso de saídas a transistor.

5- Protocolo MODBUS

O controlador CP-WS11/4DO4DI-USB, opera como escravo com protocolo Modbus/RTU e permite as operações de leitura de bloco de memórias inteiras, escrita de memória inteira entre outros comandos listados no item 6. No protocolo Modbus RTU equivalem aos comandos 03(mestre solicita valor de bloco de memórias e escravo responde, comando 06(mestre solicita escrita de um único registro). Existe um software específico para configurar o controlador para operar em protocolo MODBUS. Para gravar o programa através do SCPWS1, o protocolo MODBUS deve estar desabilitado ou então deve-se utilizar a opção Recuperar Firmware. Para configuração do protocolo modbus do CP-WS11/4DO4DI-USB, deve-se utilizar o software de Calibração e ajuste do CP-WS12/OEM que está disponível no instalador **CLP_proxsys_util_V6.exe**. A visão geral deste software está na figura 5. Ao instalar este executável será criado uma entrada no menu iniciar do Windows.

Neste programa é possível ajustar o endereço para o protocolo modbus da porta USB e da porta RS485 bem como o Baud Rate da porta RS485. Ao fechar o programa pode-se escolher em ativar ou não o protocolo MODBUS para a porta USB.

Além deste programa de calibração e ajuste, está disponível também no instalador **CLP_proxsys_util_V6.exe** um programa para ajuste das memórias não voláteis EEPROM do CP-WS11/4DO4DI-USB, chamado **Ajuste memórias EEPROM**. O aspecto deste programa pode ser visto na figura 6.

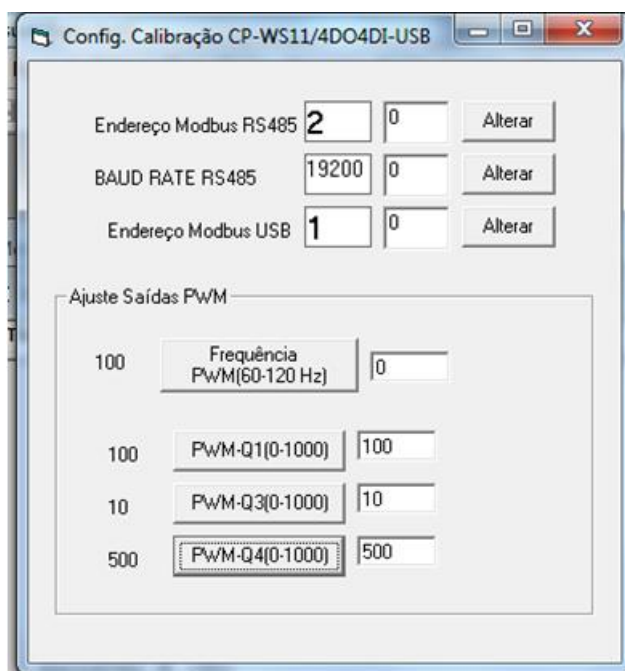


Figura 5 – Tela software de configuração e ajuste do CP-WS11/4DO4DI-USB

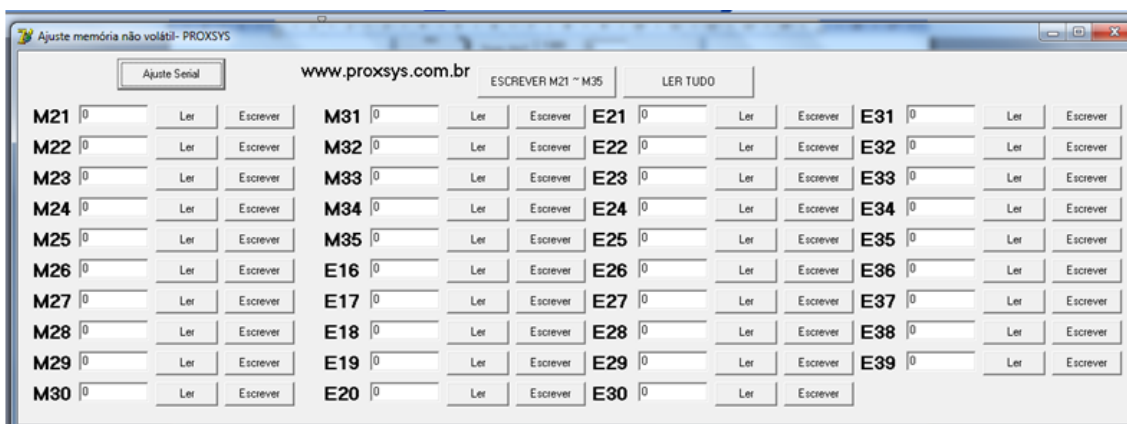


Figura 6 – Visão geral do programa Ajuste memórias EEPROM

Para o CP-WS11/4DO4DI-USB estão disponíveis as memórias de M21 a M35, conforme descrito nas tabelas 2. Todas as vezes que este programa for executado, será necessário ajustar a porta serial para o numero da porta COM que está em uso com o controlador.

6 – Comandos Protocolo Modbus suportados pelo CP-WS11/4DO4DI-USB

6.1 – Comando 01 - READ COILS

O comando Read Coils permite ler o estado das saídas Q do controlador. Cada bit da resposta indicará o estado de uma saída Q, sendo 1=ON e 0=OFF. A primeira saída requisitada será colocada no bit 0. Os bits não usados estarão em “0”.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	01
Endereço de Início	2 bytes	De 1 a 4 (Q1 a Q4)
Quantidade de saídas	2 bytes	De 1 a 4 (máximo 16)
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	01
Número de bytes	1 byte	N – Número de bytes de dados
Estados das saídas Q	N bytes	Estados das saídas
CRC16	2 bytes	

6.2 – Comando 02 - READ DISCRET INPUTS

O comando Read Discret Inputs permite ler o estado das entradas I do controlador. Cada bit da resposta indicará o estado de uma entrada I, sendo 1=ON e 0=OFF. A primeira entrada requisitada será colocada no bit 0. Os bits não usados estarão em “0”.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	02
Endereço de Início	2 bytes	De 1 a 4 (I1 a I4)
Quantidade de entradas	2 bytes	De 1 a 4 (máximo 16)
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	02
Número de bytes	1 byte	N – Número de bytes de dados
Estados das entradas I	N bytes	Estados das entradas
CRC16	2 bytes	

6.3 – Comando 03 - READ HOLDING REGISTER

O comando Read Holding Register permite ler o valor das memórias M do controlador. Será enviado como resposta o valor da memória solicitada e as subsequentes conforme a quantidade de registros solicitados. Caso seja solicitado uma quantidade acima do máximo permitido, a resposta será a quantidade máxima de registros possíveis.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	03
Número da memória inicial	2 bytes	Conforme controlador
Quantidade de memórias	2 bytes	Conforme controlador
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	03
Número de bytes	1 byte	N – Número de bytes de dados
Valor das memórias solicitada	N bytes	Valor de M[x]
CRC16	2 bytes	

6.4 – Comando 05 - WRITE SINGLE COIL

O comando Write Single Coil permite definir o valor de uma saída Q do controlador. A saída será desligada caso o valor enviado for “0000” ou ligada se o valor for “FF00” (valores em hexadecimal).

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	05
Número da saída	2 bytes	1 a 4
Valor para ligar/desligar	2 bytes	0x0000 ou 0xFF00
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	05
Número da saída	2 bytes	1 a 4
Valor para ligar/desligar	2 bytes	0x0000 ou 0xFF00
CRC16	2 bytes	

6.5 – Comando 06 - WRITE SINGLE REGISTER

O comando Write Single Register permite definir o valor de uma memória M do controlador. É informado a memória e o valor a ser colocado. Observe que as memórias são numeradas iniciando em M[1], mas para o comando a saída M[1] é a de número “0”.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	06
Número da memória	2 bytes	Conforme controlador
Valor	2 bytes	Valor a ser colocado em M[x]
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	06
Número da memória	2 bytes	Conforme controlador
Valor	2 bytes	Valor a ser colocado em M[x]
CRC16	2 bytes	

6.6 – Comando 15 - WRITE MULTIPLE COILS

O comando Write Multiple Coils permite definir o valor de várias saídas Q do controlador em um único comando. A saída será desligada caso o valor do bit correspondente for “0” ou ligada se o valor for “1”. É informado a saída inicial e a quantidade de saídas a serem alteradas. Observe que as saídas são numeradas iniciando em Q[1], mas para o comando a saída Q[1] é a de número “0”.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	15
Saída inicial	2 bytes	Número da saída - 1
Quantidade de saídas	2 bytes	Conforme controlador
Número de Bytes	1 byte	N
Valor para saídas	N bytes	
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	15
Saída inicial	2 bytes	Número da saída - 1
Quantidade de saídas	2 bytes	Conforme controlador
CRC16	2 bytes	

6.7 – Comando 15 - WRITE MULTIPLE REGISTER

O comando Write Multiple Register permite definir o valor de um bloco de memória M do controlador em um único comando. É informado a memória inicial, a quantidade de memórias a serem alteradas e um bloco com os valores. Observe que as memórias são numeradas iniciando em M[1], mas para o comando a saída M[1] é a de número “0”.

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	15
Memória inicial	2 bytes	Número memória - 1
Quantidade de memórias	2 bytes	Conforme controlador
Número de Bytes	1 byte	N
Valor para memórias	N*2 bytes	
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	15
Memória inicial	2 bytes	Número memória - 1
Quantidade de memórias	2 bytes	Conforme controlador
CRC16	2 bytes	

6.8 – Comando 16 - MASK WRITE REGISTER

O comando Mask Write Register permite alterar o valor de uma memória M utilizando funções lógicas AND, OR e NOT, deste modo permite se alterar o valor de apenas 1 bit de dado da memória.

O valor final da memória será:

$$\text{Valor_final} = (\text{Valor_atual AND Máscara_and}) \text{ OR } (\text{Máscara_or AND (NOT Máscara_and)})$$

Requisição do mestre:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	16
Memória	2 bytes	Número memória - 1
Máscara AND	2 bytes	0x0000 a 0xFFFF
Máscara OR	2 bytes	0x0000 a 0xFFFF
CRC16	2 bytes	

Resposta do controlador:

Número do controlador escravo	1 byte	De 1 a 254
Comando	1 byte	16
Memória	2 bytes	Número memória - 1
Máscara AND	2 bytes	0x0000 a 0xFFFF
Máscara OR	2 bytes	0x0000 a 0xFFFF
CRC16	2 bytes	

7 – Saídas a transistor PWM

Estão disponíveis até 3 saídas PWM no controlador CP-WS11/4DO4DI-USB. Estas saídas PWM podem operar com frequência ajustável na faixa de 60 a 120 Hz. O ajuste de frequência pode ser realizado alterando o valor da memória inteira M27 no programa ladder ou através do programa de ajuste memórias EEPROM mostrado na figura 4.

Fisicamente estas saídas estão mapeadas e identificadas como Q1, Q3 e Q4 e estão disponíveis na versão com saídas digitais a transistor mostrado na figura 2. O dutycycle do PWM pode ser ajustado através das memórias M19 para Q1, M18 para Q3 e M17 para Q4. Além da operação com PWM, as saídas podem operar normalmente com saídas digitais comuns deixando os valores de M17=0, M18=0 e M19=0 de forma fixa. Deve-se ter cuidado com o uso deste recurso de PWM na versão com saídas a rele mostrada na figura 3. No caso de saídas a rele, obrigatoriamente deve-se utilizar M17=0, M18=0 e M19=0, para manter o controle das saídas pelos respectivos operadores Q.

8- Utilização do protocolo MODBUSRTU com supervisórios ou controladores Mestres

Devido as limitações de tipos de dados que podem ser lidos/escritos com o protocolo modbus pode ser necessário lançar mão de alguns artifícios para obter o status de entradas digitais ou escrever nas saídas do controlador CP-WS11/4DO4DI-USB.

A figura 7, mostra o método utilizado para transferir o status das entradas digitais para a memória M14. Cada entrada digital, dependendo do seu estado carrega

seu bit correspondente em M14. A memória inteira M14 é uma memória com capacidade de 16 bits. Cada bit de M14, no programa ladder representa o status de uma entrada digital de acordo com seu peso binário.

2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
X	X	X	X	X	X	X	X	X	X	X	X	I4	I3	I2	I1

Figura 7 – Distribuição das entradas digitais em M14 para o programa exemplo

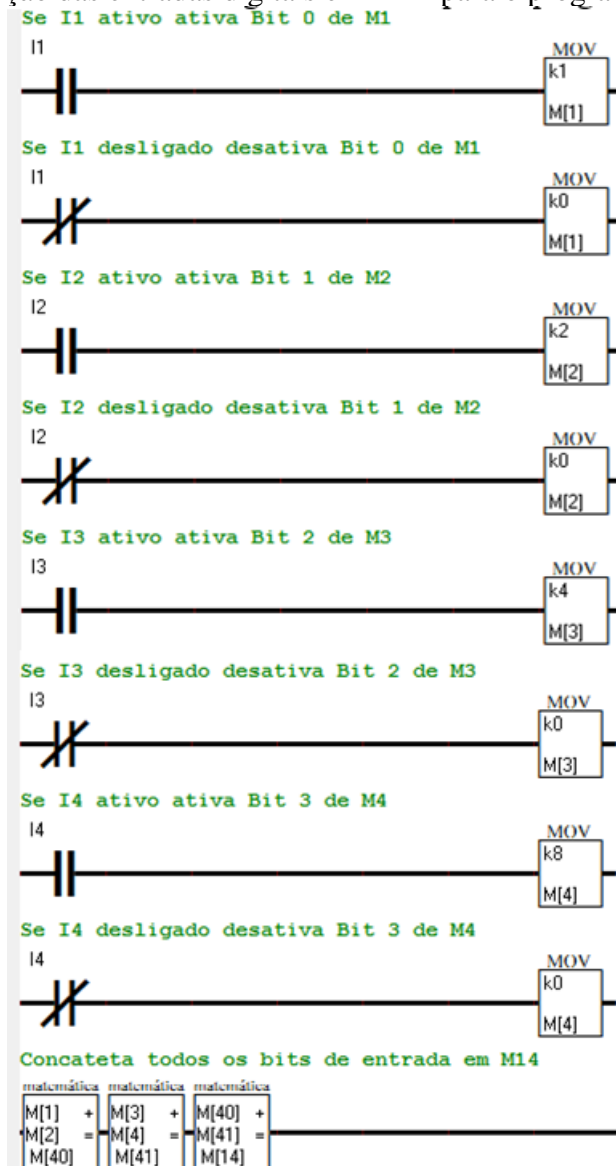


Figura 8 – Trecho de programa para aquisição entradas digitais

Assim, quando I1 ligado deve-se ativar o bit 0 com peso binário igual a 1, quando I2 ligado deve-se ativar o bit 1 com peso binário igual a 2, quando I3 ligado deve-se ativar o bit 2 com peso binário igual a 4 e quando I4 ligado deve-se ativar o bit 3 com peso binário igual a 8. Quando as respectivas entradas estiverem desligadas os pesos binários devem ser zerados.

Utiliza-se no ladder, um memória inteira para cada entrada digital, ativando e desativando o respectivo bit e em seguida todos os resultados são concatenados em M14 através de operação de soma, como mostra a figura 8.

Depois de aquiritar as entradas digitais, é necessário criar uma forma de receber os valores que irão ativar / desativar os reles auxiliares R1 e R2, utilizando para isso a memória inteira M15. Da mesma forma como foi feito para M14, cada bit de M15 será responsável pelo controle de um determinado rele auxiliar. Neste caso, vamos utilizar também um recurso disponível nas memórias do tipo L (long de 32 bits) disponível no CP-WS12/OEM. As memórias do tipo L, podem endereçar contatos no programa ladder, utilizando o endereçamento L1/1, L1/2, L1/3 e assim sucessivamente. Neste caso, L1/1 representa o bit 0 da memória L1, L1/2 o bit 1 e assim sucessivamente. Basta então mover a memória M15 para L1 e ai então endereçar os contatos com os bits correspondentes. A figura 9 mostra como é a disposição dos bits em M15 e L1.

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	R2	R1

Disposição dos bits em M15

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	L1/2	L1/1

Disposição dos bits em L1

Figura 9 – Distribuição dos reles auxiliares em M15 para o programa exemplo

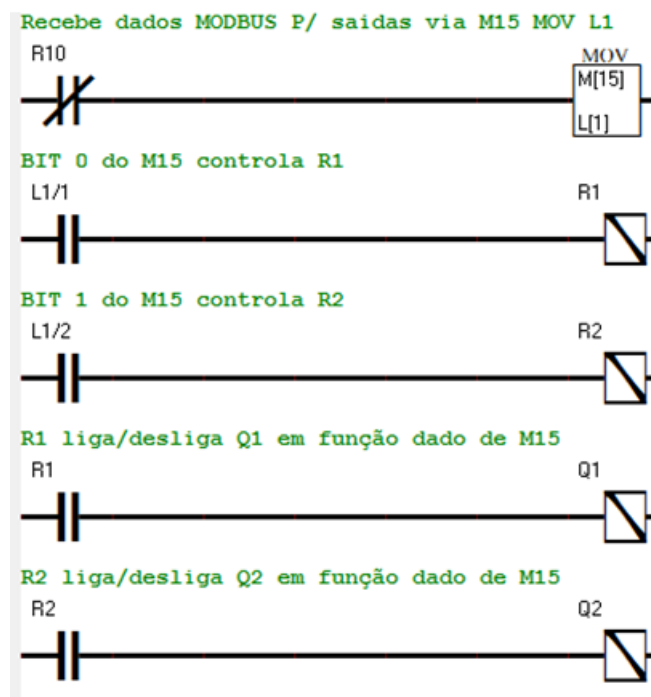


Figura 10 – Trecho programa tratamento recepção de dados recebidos por MODBUS

A figura 10, mostra o trecho de programa necessário para esta operação. Primeiro o valor recebido pelo CP-WS11/4DO4DI-USB proveniente do supervisor ou CLP mestre através de M15 é movimentado para L1, neste caso é obrigatório o uso de um contato lógico antes do bloco MOV, utilizou-se R10 sempre fechado, para executar esta operação. Com o valor transferido para L1, pode-se utilizar cada bit de L1 em contatos abertos ou fechados para execução de operações no diagrama ladder.

9 – Geração de pulsos para driver de motor de passo

O controlador CP-WS11/4DO4DI-USB, tem a possibilidade de configuração para geração de pulsos para driver de motor de passo. Os pulsos são gerados através da saída digital PNP- Q3 (24 Vcc – ver recomendações do driver do motor), que pode operar tanto como uma saída comum quanto uma saída de geração de pulsos. A forma como a saída Q3 opera é configurada pelo programa Ajuste memórias EEPROM mostrado na figura 6 disponível no instalador **CLP_proxsys_util_V5.exe**, mostrado no item 5.

O método de operação da saída Q3 é definida através do valor ajustado na memória M22. Para M22=0 a saída Q3 tem operação normal como saída digital. Para M22=1 a saída Q3 opera gerando pulsos para drive de motor de passo de acordo com o ajustado em M24.

Existem duas formas de gerar pulsos para a saída Q3 e estas formas são configuradas através da memória inteira M24. São elas :

- M24 = 0 :Fazendo um SET da saída Q3, pulsos são gerados até que uma borda de descida na entrada I1 do controlador seja detectada. Este método de operação é indicado para máquinas rotuladoras para rótulo auto-adesivos por exemplo. Neste caso a entrada I1 recebe sinal de um sensor que percebe o intervalo entre os rótulos no liner.

-M24= 1 : Acionando a saída Q3, uma quantidade determinada pelo valor ajustado em L4 de pulsos é gerada na saída Q3. Antes de iniciar a geração de pulsos deve-se zerar o valor de contagem de pulsos através de breve acionamento de R23.

A seguir estas formas de operação serão detalhadas.

9.1- Geração de pulsos controlada por sensor M24=0

Como já comentado, esta forma de operação é obtida com a memória M24=0. As memórias envolvidas nesta operação são:

M21 – Rotação no eixo do motor com resolução de décimos de RPM com base em um driver + motor ajustado para 400 pulsos por volta. Neste caso para o motor operar a 300 RPM, deve-se ajustar M21=3000.

O valor máximo para ajuste deste parâmetro é 6000 (que equivale a 600 RPM) para um conjunto driver + motor ajustado para 400 pulsos por volta.

Um trecho de programa típico para a operação deste modo de operação pode ser visto na figura 11.

No exemplo apresentado na figura 11, um pulso na entrada I2, que pode ser um sensor de presença por exemplo, seta a saída Q3. Neste instante pulsos são gerados na saída Q3, com frequência de acordo com o ajustado em M21.

Quando uma transição de nível alto para nível baixo (borda de descida) na entrada I1 é detectada, a geração de pulsos é interrompida. Observe que o número de rotações por minuto (RPM) do eixo é ajustado através da movimentação de um valor constante para M21 no caso 200, equivalente a 20 RPM.

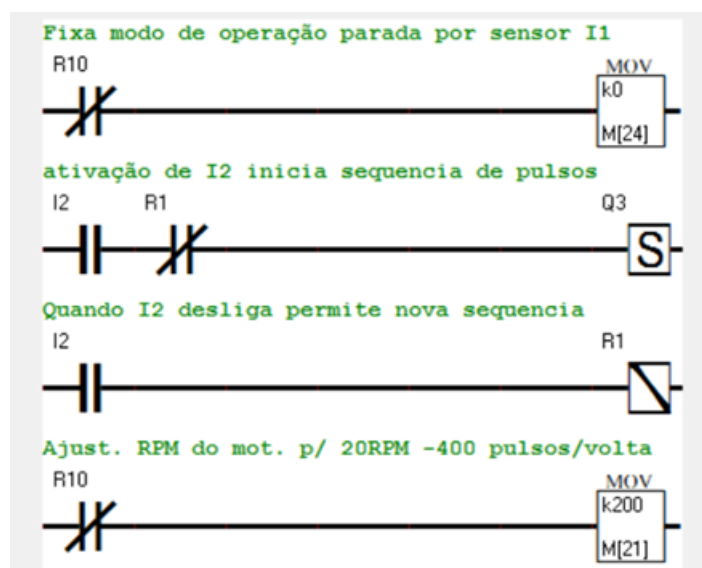


Figura 11 – Geração de pulsos controlada por sensor

O valor máximo permitido para M21 é 6000 equivalendo a 600 RPM.

9.2- Geração de número pré-determinado de pulsos M24=1

Neste modo de operação, são gerados pulsos de acordo com a quantidade de pulsos ajustados na memória do tipo LONG L4, ajustado via movimentação no programa ladder. Da mesma forma que no modo de operação 9.1, a memória M21, define o número de RPM de rotação do eixo do motor. Em L4 deve-se definir quantos pulsos ou quantos passos o motor deve girar. Isso vai depender de como o conjunto Motor + driver está ajustado em termos de resolução. Vamos supor que um conjunto motor + driver esteja ajustado para 400 pulsos/volta. Neste caso são necessários 400 pulsos para gerar uma volta completa no eixo do motor. É muito importante, neste caso, garantir que o contador de pulsos gerador esteja em um valor conhecido ou zerado. Para zerar o contador de pulsos é necessário ativar R23. Enquanto R23 ativo a contagem de pulsos permanecerá zerada. Um exemplo de programa para esta forma de operação pode ser visto na figura 12.

No exemplo da figura 12, as entradas I1 e I2 são utilizadas para gerar pulsos para o driver de motor de passo na saída Q3. Com a particularidade que I2 além de iniciar a geração de pulsos ativa também a saída Q2 que pode ser utilizado para acionar a entrada de controle de direção do driver. Assim, um acionamento de I1 faz o motor girar no sentido horário e o acionamento de I2 faz o motor girar no sentido anti-horário. Observe também que um breve pulso de I1 irá zerar o contador de pulsos através do acionamento de R23 e fará o motor girar 200 pulsos. Enquanto I1 estiver acionado o controlador ficará gerando pulsos para a saída Q3, visto que R23 permanecerá acionado. Quando I1 for desativo o motor ainda irá girar por mais 200 pulsos ajustados em L4. Um funcionamento semelhante é utilizado na entrada I2 com a diferença do acionamento da saída Q2 que pode controlar a direção de giro do motor.

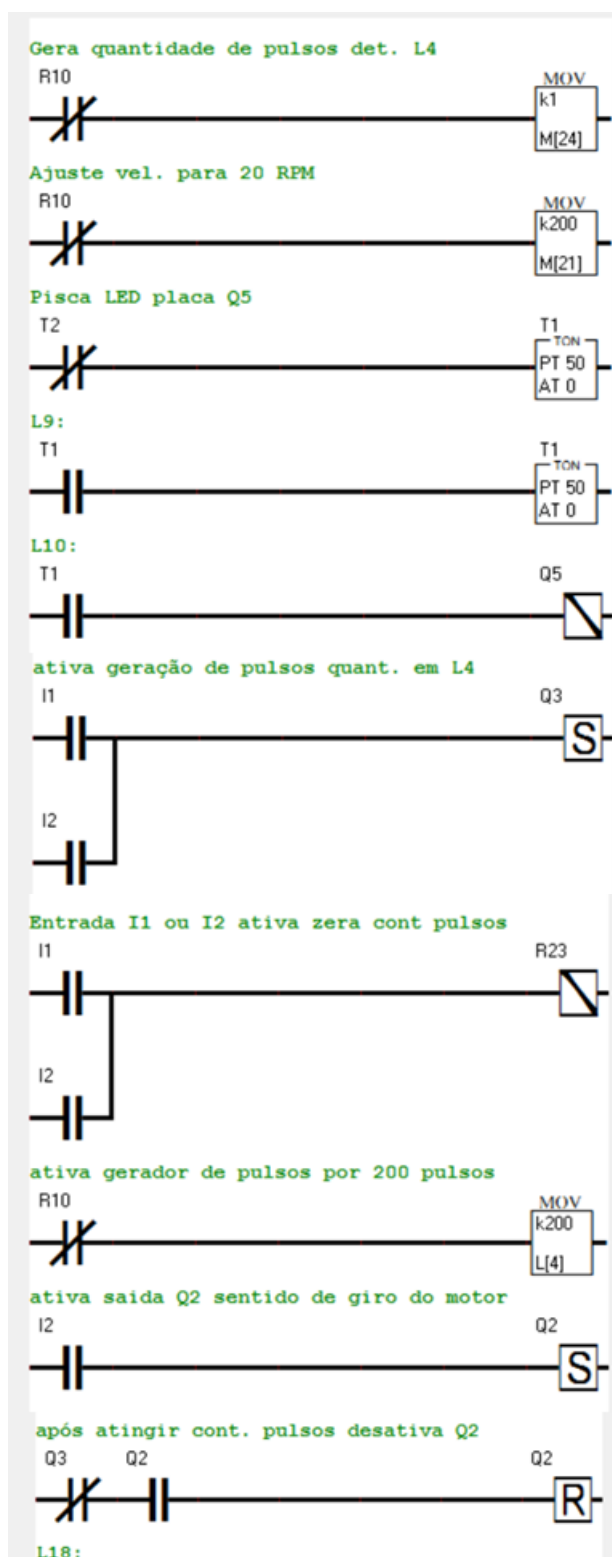


Figura 12 – Operação com geração pulsos determinados por L4

10- Salvando valores não voláteis de M na memória EEPROM

Existe a possibilidade de salvar o valor corrente das memórias M na faixa de M21 a M35 na memória EEPROM. Deve-se tomar extremo cuidado com isso pois existe um limite máximo de 100.000 vezes para a gravação destas memórias. Para proteger estas memórias de gravação acidental criou-se um procedimento que depende

do estado no rele auxiliar R50. Desta forma para que o valor corrente das memórias de M21 a M35 seja gravada na memória não volátil é necessário ativar a bobina R50 por um ciclo de scan. Esta funcionalidade é útil quando se deseja por exemplo utilizar entradas digitais do controlador para setar valores através de pulsos UP e DOWN. No exemplo da figura 12 as entradas I1 e I2 ajustam o valor da memória M21 na faixa de 0 a 50 através de impulsos de UP (I1) e Down (I2). Nesta lógica após um período de 3 segundos de inatividade nas entradas I1 e I2 o novo valor de M21 é salvo na memória EEPROM. Para visualizar o efeito utilizamos a saída Q5 que é um led montado no controlador.

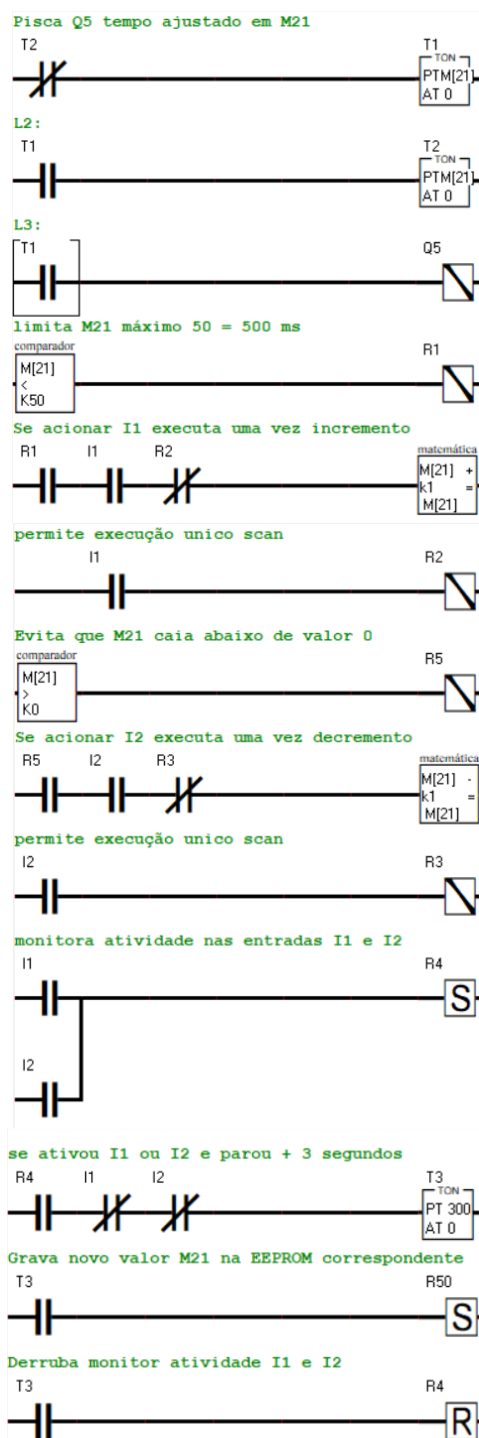


Fig. 13 – Programa exemplo armazenagem EEPROM Ladder