



# Direto ao **Ponto**

**Comunicação CC-Link IE Field-Basic  
com Servo JE-C  
Nº. DAP-iQF-04**

*Rev. A*





**Revisões**

Data da Revisão	Nome do Arquivo	Revisão
Dez/2018 (A)	DAP-iQF-04_CTRL JEC CCLINK BASIC	Primeira edição



### 1. OBJETIVO

O objetivo desse documento é explicar como controlar o Servo JE-C para via CC-Link ie Field Basic, com o FX5

### 2. CONCEITO

O servo MR-JE-C possui uma porta ethernet integrada que serve para receber comandos através de diferentes redes baseadas em Ethernet, como CC-Link IE field Basic. No nosso exemplo iremos controlar o servo por essa rede

### 3. DOCUMENTAÇÃO PARA REFERÊNCIA

- 1. Manual Usuário JE-C
- 2. Manual CC-Link IE Filed Basic JE-C
- 3. Manual Profile Mode JE-C

### 4. HARDWARE/SOFTWARE

- 1 PC com SO Windows XP, 7 ou 10 com o software GxWorks-3 instalado;
- 1 Servo JE-C
- 1 FX5U

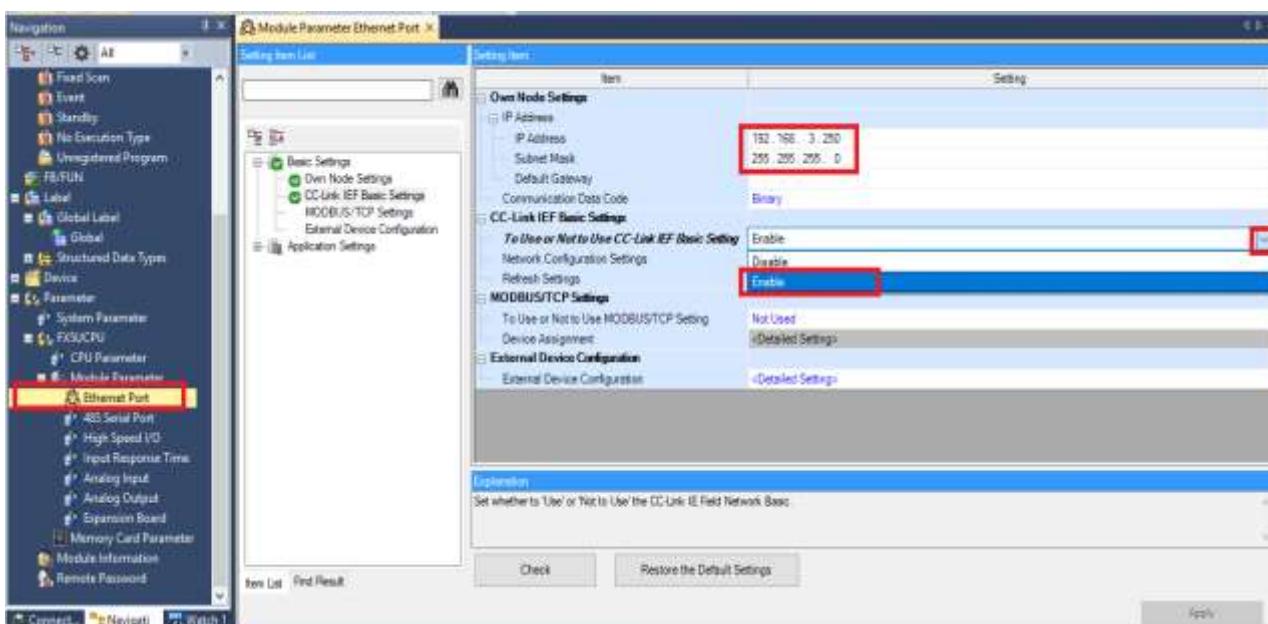
### 5. CONFIGURAR A COMUNICAÇÃO NO SERVO

É necessário configurar o servo para trabalhar corretamente, nesse exemplo utilizaremos o servo com IP 192.168.3.50, para mais detalhes de configuração do servo consultar o **DAP-SV05(A)\_PAR\_JEC\_ETHERNET**

### 6. CONFIGURAR PLC FX5U

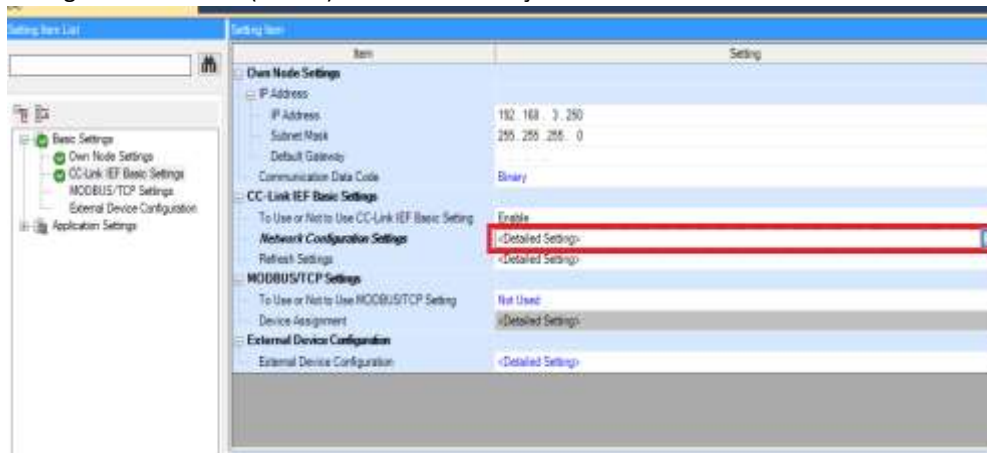
Para configurar o CLP, a primeira coisa, é configurar os parâmetros da rede CC-link Basic, basta seguir os seguintes passos:

- 1. Configurar o IP do CLP e habilitar o CC-Link IE Field Basic

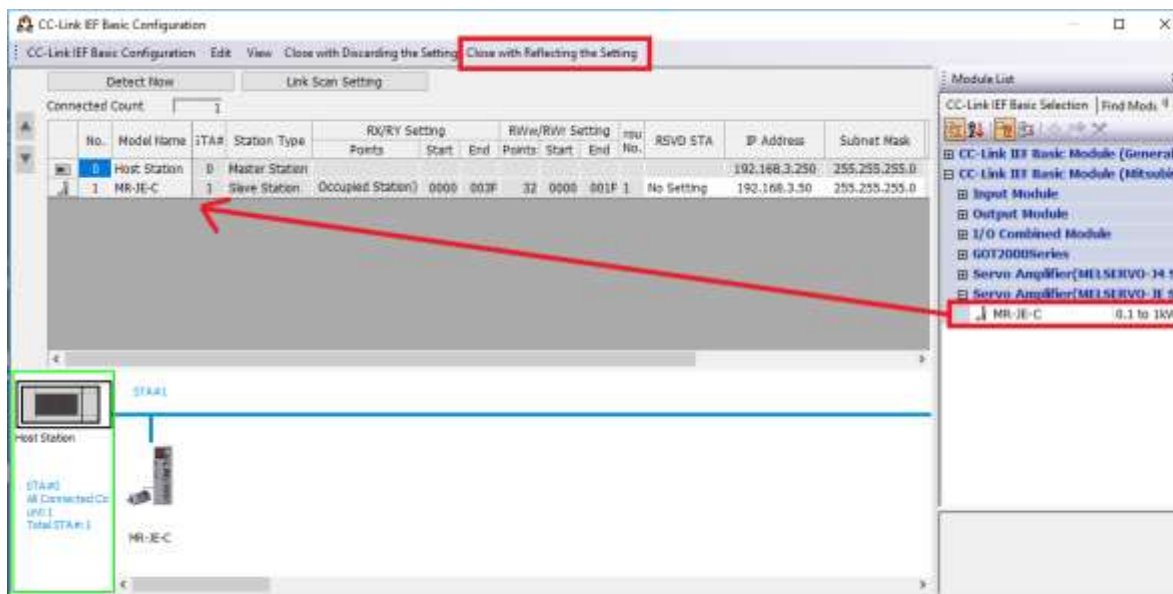




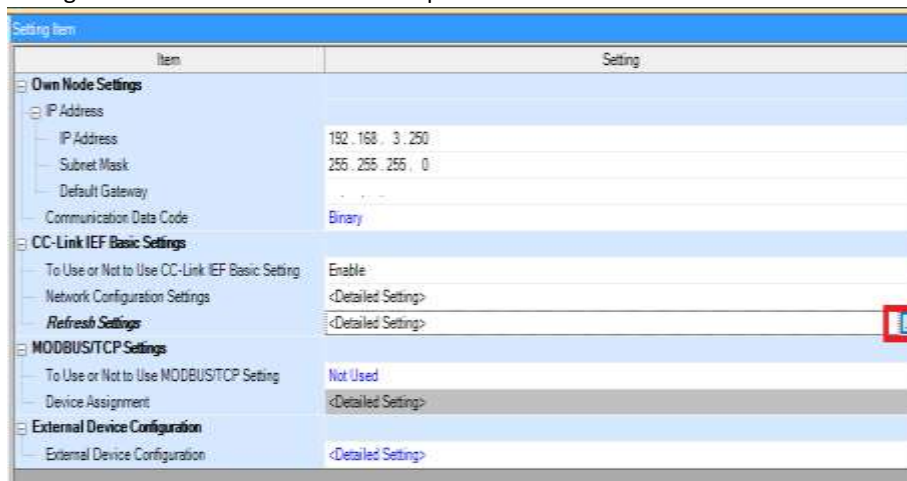
2. Configure os escravos (servos) da sua comunicação



Arraste o servo JE-C do menu, configure o seu endereço IP e em seguida clique em Close with reflecting the setting para confirmar a configuração:



3. Configure a área de memória utilizada pela rede





Aqui devemos lembrar que quatro tipos de dados são trocados na comunicação RX, RY, RWr, RWw.

- RX, são **bits** de **leitura** do CLP, onde ele pode **ler** status e confirmações.
- RY, são **bits** de **escrita** do CLP, onde ele pode **enviar** comandos.
- RWr, são **words** de **leitura** do CLP, onde pode **ler** monitores numéricos.
- RWw, são **words** de **escrita** do CLP, onde ele pode **enviar** comandos numéricos.

Os RX e RY, podem ser atribuídos a qualquer memória de bits, como X, Y, B, M etc...

Os RWr e RWw, podem ser atribuídos a qualquer memória de words como D, W, R etc...

Nesse exemplo foi configurado da seguinte forma:

Link Side					CPU Side				
Device Name	Points	Start	End		Target	Device Name	Points	Start	End
RX	64	00000	0003F	↔	Specify Devi	B	64	00000	0003F
RY	64	00000	0003F	↔	Specify Devi	B	64	00040	0007F
RWr	32	00000	0001F	↔	Specify Devi	W	32	00000	0001F
RWw	32	00000	0001F	↔	Specify Devi	W	32	00040	0005F

Ou seja,

- RX – de B0 a B3F
- RY – de B40 a B7F
- RWr – de W0 a W1F
- RWw – de W40 a W5F

Feito isso, toda nossa parte de configuração da Rede está completa, agora basta um pouco de programação para enviar comandos para o servo.

### 7. PROGRAMAÇÃO CLP

Depois que as áreas de memória foram configuradas basta seguir as tabelas de comandos encontradas nos manuais para comandar o servo

As tabelas são:



Master station → Servo amplifier (RYn)				Servo amplifier → Master station (RXn)			
(Note) Device No.	Device	Symbol	Remark	(Note) Device No.	Device	Symbol	Remark
RYn0 to RY (n + 3) E	Not used			RXn0 to RX (n + 3) E	Not used		
RY (n + 3) F	Cyclic communication ready command	CSR		RX (n + 3) F	Cyclic communication ready	SSR	

Master station → Servo amplifier (RWwn)				Servo amplifier → Master station (RWm)			
(Note) Device No.	Index	Device		(Note) Device No.	Index	Device	
RWwn00	6060	Control mode	Modes of operation	RWm00	6061	Control mode display	Modes of operation display
RWwn01	6040	Control command	Controlword	RWm01			
RWwn02	2D01	Control Input 1	Control DI 1	RWm02	6041	Control status	Statusword
RWwn03	2D02	Control Input 2	Control DI 2	RWm03			
RWwn04	2D03	Control Input 3	Control DI 3	RWm04	6064	Current position (command unit)	Position actual value
RWwn05		Position command (pp)	Target position	RWm05	606C	Current speed	Velocity actual value
RWwn06	607A						
RWwn07		Speed command (pv)	Target velocity	RWm07	60F4	Droop pulses	Following error actual value
RWwn08	60FF						
RWwn09		Speed limit value (tq)	Velocity limit value	RWm09	6077	Current torque	Torque actual value
RWwn0A	2D20						
RWwn0B	6071	Torque command (tq)	Target torque	RWm0A	2D11	Control output 1	Status DO 1
RWwn0C		Command speed (pp)	Profile velocity	RWm0B	2D12	Control output 2	Status DO 2
RWwn0D	6081						
RWwn0E		Acceleration time constant (pp, pv)	Profile acceleration	RWm0C	2D13	Control output 3	Status DO 3
RWwn0F	6083						
RWwn10		Deceleration time constant (pp, pv)	Profile deceleration	RWm0D	2A42	Alarm No.	Current alarm 2
RWwn11	6084						
RWwn12		Amount of torque command change (per second) (tq)	Torque slope	RWm0E	60B9	Touch probe function status	Touch probe status
RWwn13	6087						
RWwn14	60E0	Torque limit value (forward)	Positive torque limit value	RWm0F	60BA	Touch probe 1 Position latched at the rising edge	Touch probe pos1 pos value
RWwn15	60E1	Torque limit value (reverse)	Negative torque limit value	RWm10			
RWwn16				RWm11	60BB	Touch probe 1 Position latched at the falling edge	Touch probe pos1 neg value
RWwn17	60B8	Touch probe function setting	Touch probe function	RWm12			
RWwn18	60F2	Positioning operation setting	Positioning option code	RWm13			
RWwn19	2D05	Control Input 5	Control DI 5	RWm14	2C12	Input device status 1	External Input signal display1
RWwn1A				RWm15			
RWwn1B				RWm16			
RWwn1C				RWm17			
RWwn1D				RWm18			
RWwn1E				RWm19			
RWwn1F				RWm1A			
				RWm1B			
				RWm1C			
				RWm1D			
				RWm1E			
				RWm1F			

A seguir veremos alguns exemplos de como utilizar esses registradores:

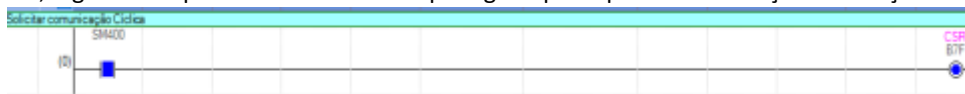




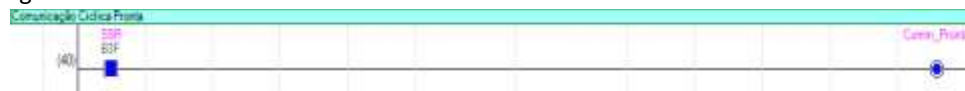
### 1. Habilitar comunicação

Antes de fazer qualquer tipo de comando é necessário habilitar a comunicação cíclica do CC-Link, para isso é necessário acionar o RY03F.

No exemplo, o RY começa no B40, RY00 = B40, então o RY3F = B40 + 3F (somar em hexa) = B7F, logo tenho que deixar esse bit sempre ligado para que a comunicação aconteça:



Com isso feito posso verificar meu sinal de entrada RX, se essa habilitação foi feita com sucesso, o bit que monitora isso é o RX3F, usando a mesma lógica, meu RX começa no B00, logo a minha memória será o B3F:

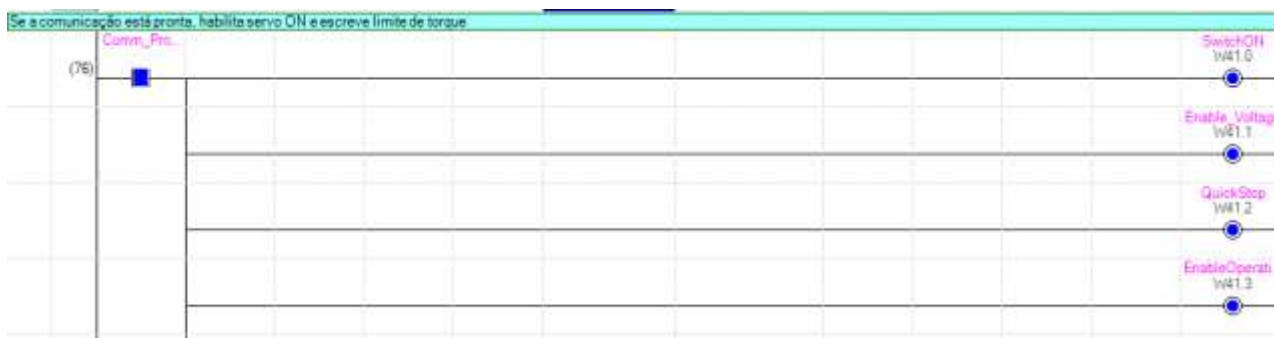


### 2. Ligar o Servo ON

Antes de executar qualquer tipo de movimento, é necessário executar o Servo ON, que habilita o servo e deixar o motor com torque, pronto executar movimentos. Para isso é necessário ligar alguns Bits da Control Word que é a RWw01, no nosso caso seria a W41. Cada bit dessa word tem uma função:

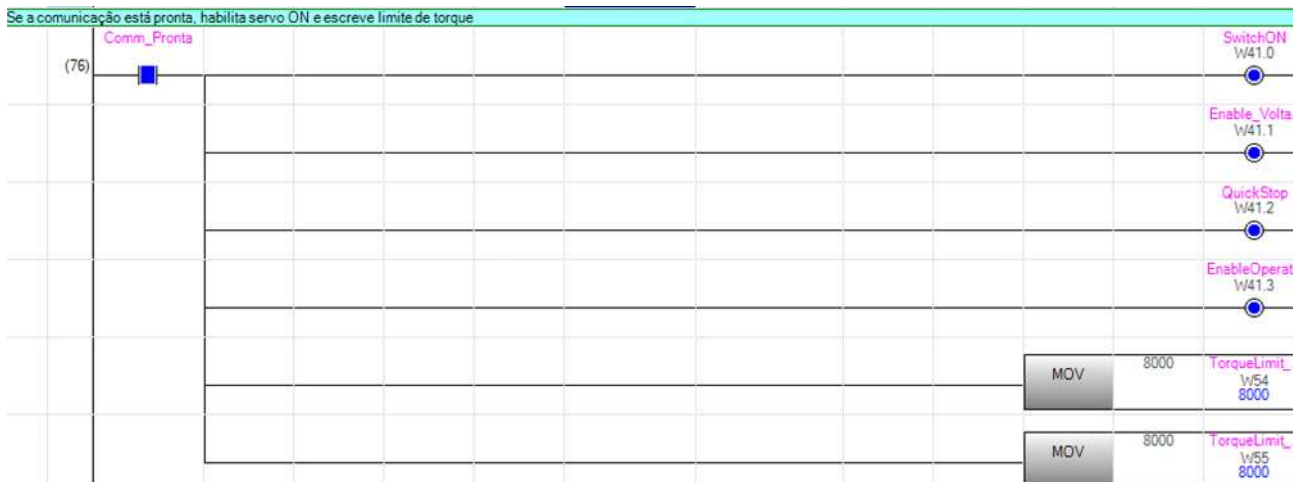
6040	0	Control command (Controlword) Set control commands to control the servo amplifier. Bit 0: switch on Bit 1: enable voltage Bit 2: quick stop Bit 3: enable operation Bit 4 to Bit 6: operation mode specific Bit 7: fault reset Bit 8: halt Bit 9: operation mode specific Bit 10 to Bit 14: reserved Bit 15: operation mode specific
------	---	---

Para Habilitar o servo ON é necessário ligar os 4 primeiros bits:





Outra coisa MUITO importante para ser fazer nesse momento, é colocar limites de torque no servo, pois as memórias vem zeradas, logo o servo fica sem torque, mesmo habilitado. As memórias de limite de torque são RWw14 e RWw15, logo W54 e W55:



### 3. Referenciamento

Normalmente antes de fazer posicionamento com o servo, é definido o seu ponto zero, também conhecido como home.

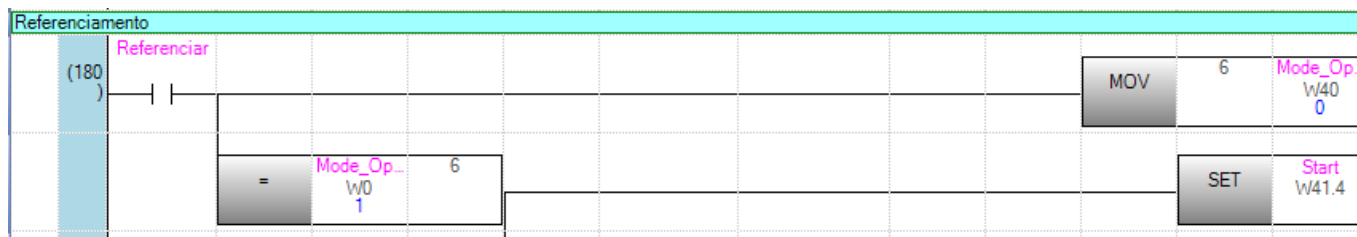
O tipo de home que o servo vai fazer pode ser escolhido pelos parâmetros, no **DAP-SV05(A)\_PAR\_JEC\_ETHERNET**, ensina como escolher o tipo de home.

Para executar essa tarefa vamos necessitar do RWw0, RWw1, que são os W40 e W41 respectivamente

RWw0:

6060	0	Control mode (Modes of operation) Set the control mode. 0: No mode assigned 1: Profile position mode (pp) 3: Profile velocity mode (pv) 4: Profile torque mode (tq) 6: Homing mode (hm) -20: Position control mode -21: Speed control mode -22: Torque control mode
------	---	--

Portanto, precisamos enviar o valor 6 para o W40, e acionar Start na ControlWord, que seria W41.4



No Exemplo acima foi utilizado a W0 em uma comparação, para verificar se o Mode of Operation realmente foi alterado para 6 (que é o valor para executar o home)



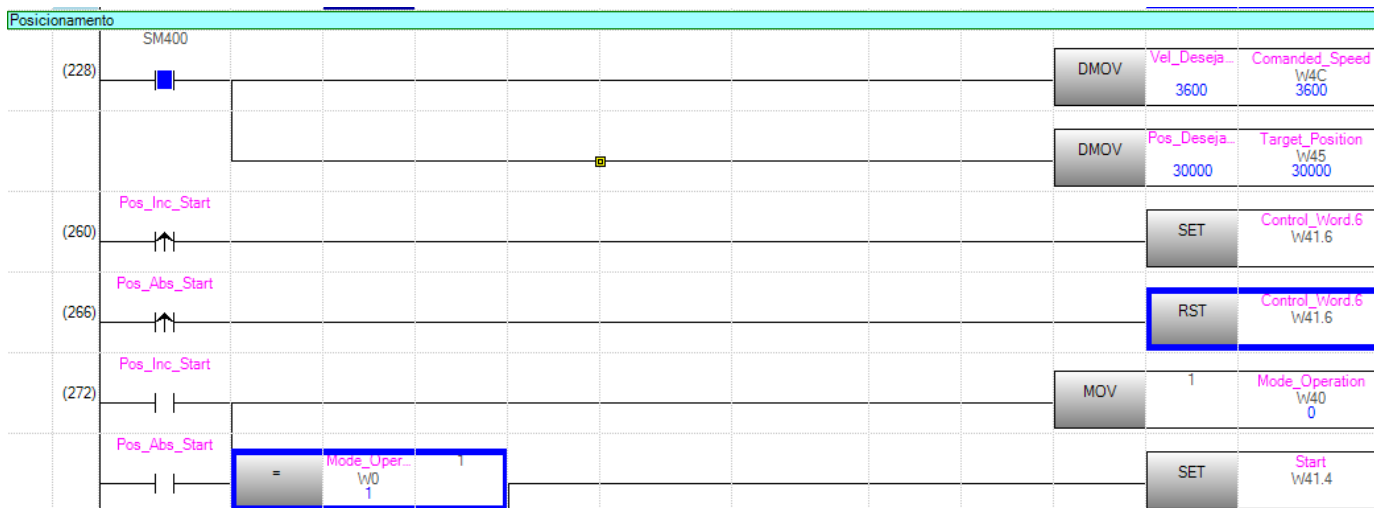


### 4. Posicionamento

Para executar o posicionamento devemos configurar uma serie de parâmetros:

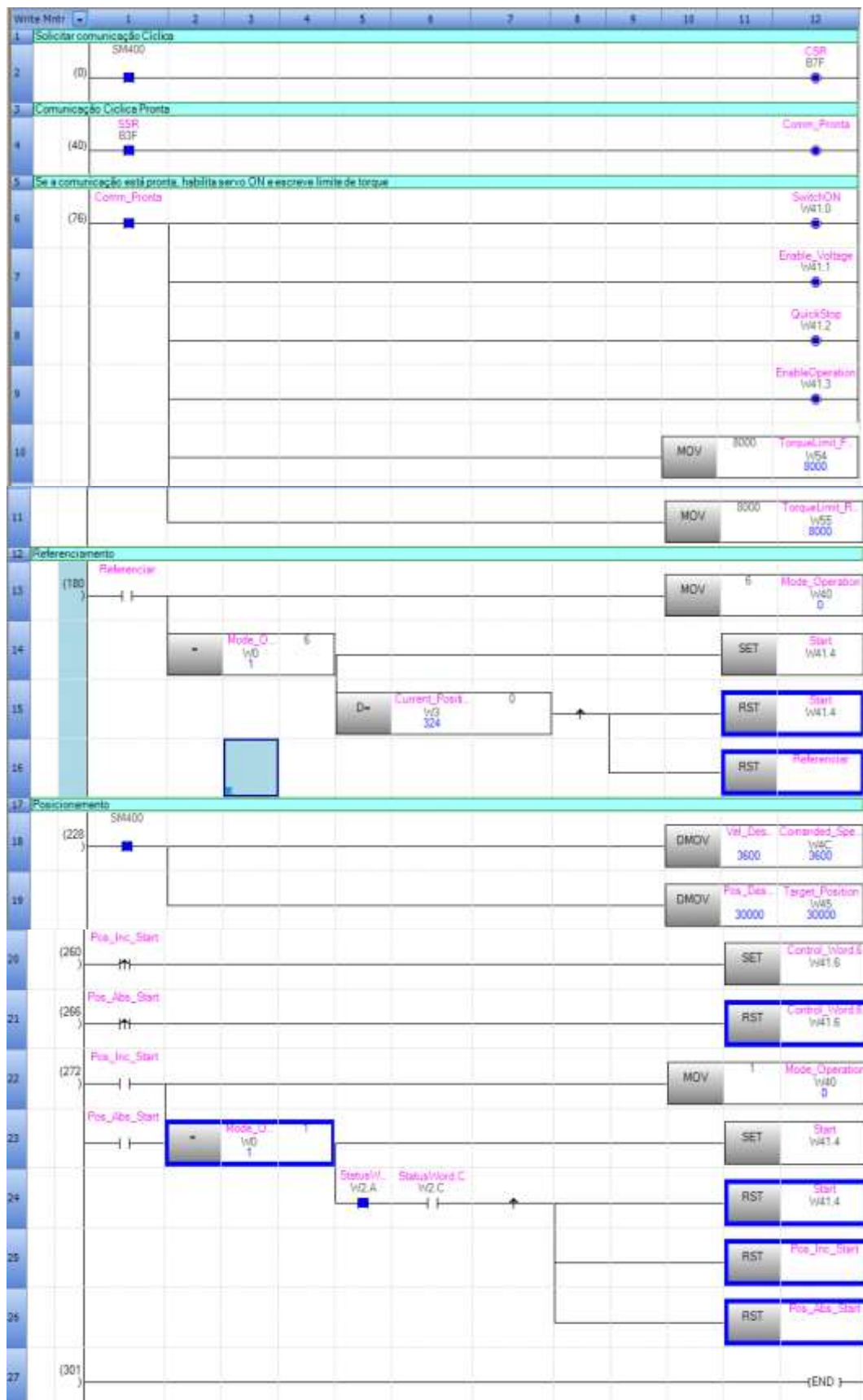
- Tipo de Posicionamento (incremental ou absoluto)( W41.6)
- Posição de destino (W45)
- Velocidade (W4C)
- Mode Operation (W40)
- Start (W41.4)

As unidades de deslocamento estão em unidades de posição configuradas na engrenagem eletrônica do servo, consulte o **DAP-SV05(A)\_PAR\_JEC\_ETHERNET** para mais detalhes, caso nada tenha sido configurado, a unidade estará em pulsos do encoder do servo comandado. Já a velocidade, a unidade é 0.01 rpm.





5. Apêndice – Programa Exemplo





	Label Name	Data Type	Class	Assign (Device/Label)	Initial Value	Constant	Comment
1	CSR	Bit	VAR_GLOBAL	→ B7F			
2	SSR	Bit	VAR_GLOBAL	→ B3F			
3	Mode_Operation	Word [Unsigned]/Bit String [16-bit]	VAR_GLOBAL	→ W40			6060
4	Control_Word	Word [Unsigned]/Bit String [16-bit]	VAR_GLOBAL	→ W41			6040
5	Target_Position	Double Word [Signed]	VAR_GLOBAL	→ W45			607A
6	Acc	Double Word [Signed]	VAR_GLOBAL	→ W4E			6083
7	Decc	Double Word [Signed]	VAR_GLOBAL	→ W50			6084
8	Comanded_Speed	Double Word [Signed]	VAR_GLOBAL	→ W4C			6081
9	TorqueLimit_FWD	Word [Signed]	VAR_GLOBAL	→ W54			
10	TorqueLimit_REV	Word [Signed]	VAR_GLOBAL	→ W55			
11	Reset_Alarm	Bit	VAR_GLOBAL	→ W41.7			
12	Start	Bit	VAR_GLOBAL	→ W41.4			
13	SwitchON	Bit	VAR_GLOBAL	→ W41.0			
14	Enable_Voltage	Bit	VAR_GLOBAL	→ W41.1			
15	QuickStop	Bit	VAR_GLOBAL	→ W41.2			
16	EnableOperation	Bit	VAR_GLOBAL	→ W41.3			
17	Curent_Position	Double Word [Signed]	VAR_GLOBAL	→ W3			
18	Mode_Operation_Display	Word [Unsigned]/Bit String [16-bit]	VAR_GLOBAL	→ W0			
19	StatusWord	Word [Unsigned]/Bit String [16-bit]	VAR_GLOBAL	→ W2			