



Direto ao **Ponto**

**Comunicação CC-Link IE Field-
Basic com servo MR-JE-C
usando Function Block de
comando**

Rev. A





Revisões

Data da Revisão	Nome do Arquivo	Revisão
Dez/2020 (A)	DAP-PIQF-09(A)_CC- Link_ IEF_Basic_FX5_MR-JE-C	Primeira edição



1. Objetivo

O objetivo desse documento é explicar como configurar a comunicação via CC-Link IE Field Basic entre o PLC FX5 e um servo modelo MR-JE-C através do uso de um Function Block e um programa de exemplo, desenvolvido pela MEB.

Para baixar os programas de exemplo, por favor, entre em contato com a MEB:

cat@mitsubishielectric.com.br ou (11) 4689-3000, opção 2.

2. Software

- GX Works 3
- MR-Configurator 2

3. Documentação para Referência

Manual Usuário JE-C

Manual CC-Link IE Field Basic JE-C

Manual Profile Mode JE-C

4. Hardware

- CPU – FX5U
- Servo MR-JE-C

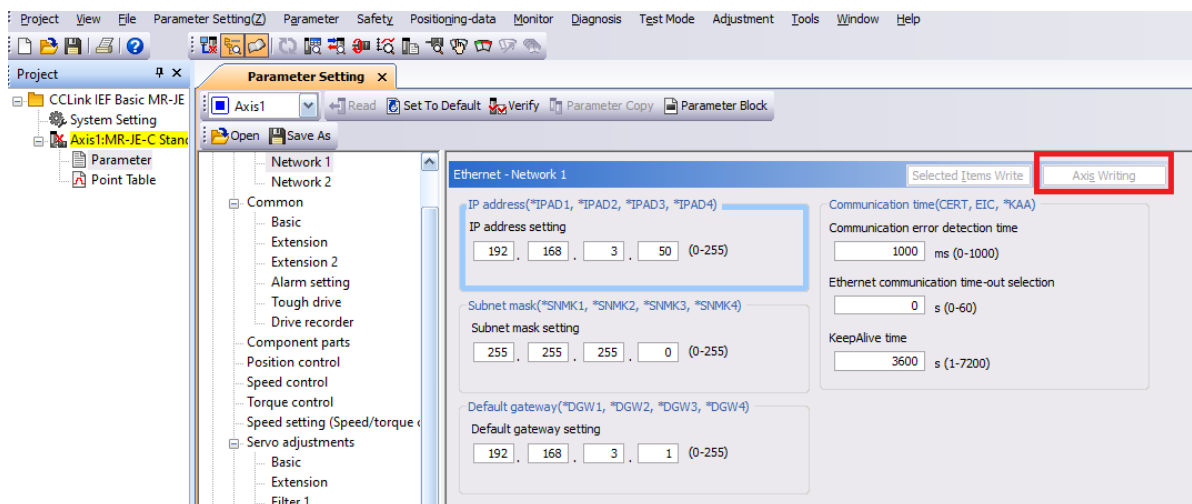
5. Configuração do Servo

Os servos Mitsubishi da série MR-JE-C possuem em sua porta Ethernet nativa o protocolo CC-Link IE Field Basic, podendo ser conectado facilmente a uma rede Ethernet via hub ou diretamente a outro dispositivo (ponto a ponto), usando um cabo CAT 5 ou superior comum.

Para configurar o servo, abra o arquivo **CCLink IEF Basic MR-JE Eixo_n.mrc2**.

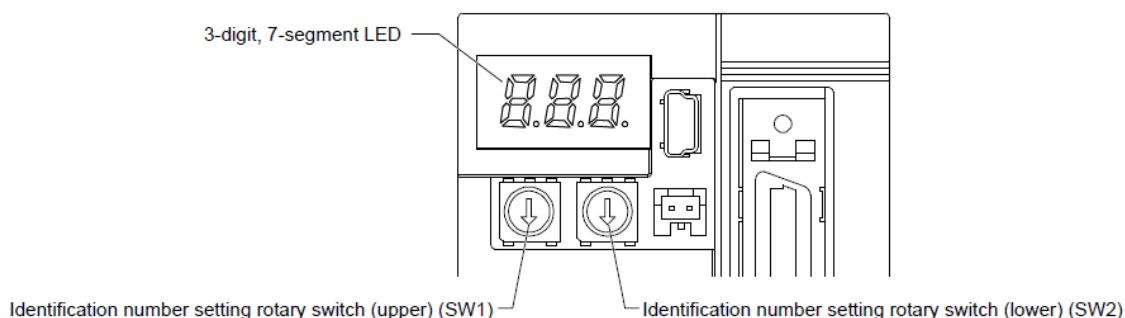
O “n” do **Eixo_n** corresponde ao número do eixo que o usuário está editando. Nesse programa, temos até 4 eixos criados como exemplo. Se houver necessidade, o usuário poderá manualmente adicionar mais eixos.

Com o computador conectado ao amplificador e o arquivo aberto, clique na aba “Parameter” na janela de navegação e clique em “**axis writing**” para escrever os parâmetros. Repita esse procedimento em todos os eixos.



Certifique-se de que as chaves do servo SW1 e SW2 estão em “Zero” ou na posição em HEXADECIMAL correspondente ao último octeto do endereço de IP atribuído ao amplificador.

The following figure shows the identification number setting rotary switch.



As posições das chaves anulam as configurações de IP feitas via parâmetros em “Network 1”.



6. Configuração do CLP FX5U

O CLP FX5U possui uma porta Ethernet nativa, que serve para programar a CPU e para estabelecer conexão com redes e equipamentos externos. É por meio dela que nos conectaremos à rede CC-Link IE Field Basic.

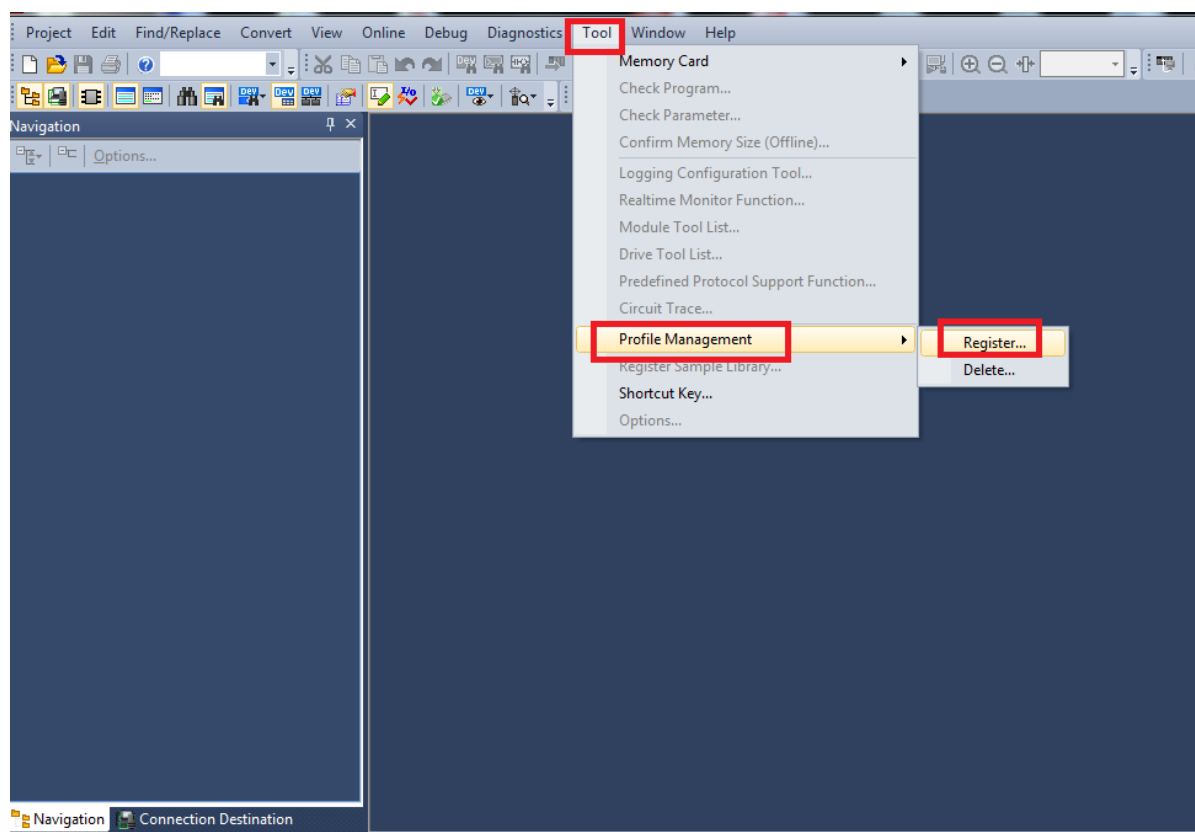
Conecte um cabo CAT 5 (ou equivalente) à porta Ethernet da CPU e conecte a outra ponta do cabo a um HUB, switch ou roteador de sua escolha. Conecte também os inversores ao mesmo equipamento de gerenciamento de rede.

Use o software GX Works 3 para efetuar as seguintes configurações.

Obs.: Carregando o programa de exemplo **CCLink IEF Basic iQ-F_MR-JE_Structured_DT+FB_4_Eixos.gx3** na CPU FX5U, as configurações seguintes serão automaticamente carregadas.

Primeiramente é necessário registrar o *Profile* do servo para que o FX5 consiga identificar os servos na rede, para baixar o arquivo necessário, entre em contato com a Mitsubishi Electric Brasil.

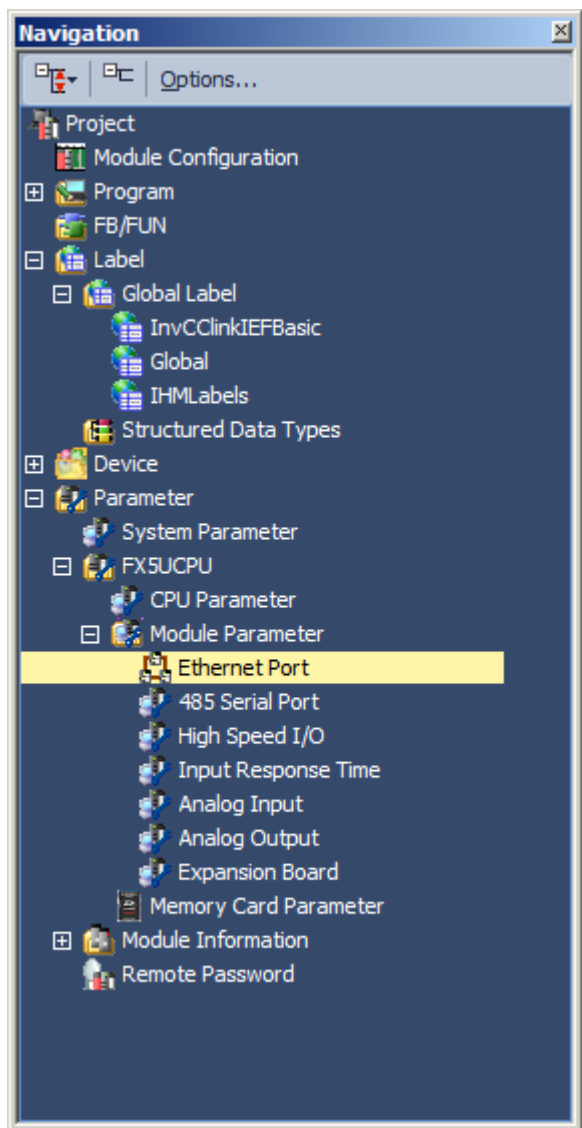
Com o arquivo do *Profile* baixado, abra o GXWorks3 sem nenhum projeto criado vá em “Tool” > “Profile Management” > “Register”, e indique o local do arquivo.



Uma vez indicado o *Profile* será instalado e estará pronto para usar.



Depois do projeto criado e *Profile* instalado, abra um novo projeto e vá até o menu de configuração “*Ethernet Port*”, conforme a figura a seguir:



Na janela de configuração, habilite a função CC-Link IEF Basic no menu “*To use or not to use CC-link IEF Basic Setting*” selecionando a opção “*enable*”, conforme a figura a seguir;



Module Parameter Ethernet Port

Setting Item List

- Basic Settings
- Own Node Settings
- CC-Link IEF Basic Settings
- External Device Configuration
- Application Settings

Setting Item

Item	Setting
Own Node Settings	
IP Address	192.168.3.250
Subnet Mask	
Default Gateway	
Communication Data Code	Binary
CC-Link IEF Basic Setting	
To Use or Not to Use CC-Link IEF Basic Setting	Enable
Network Configuration Settings	<Detailed Setting>
Refresh Settings	<Detailed Setting>
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set to determine how many bits of the IP address are used as the network address, which is used to identify the network.
Masked bit has been specified from top between the top bit and bit 2.
For example, set '255.255.255.0' to assign the upper 24 bits of IP address to the subnet mask.
[Setting range]
- Empty
- 0.0.0.1 to 255.255.255.255 (in decimal)

Item List Find Result

Check Restore the Default Settings

Apply

Após isso, clique no item seguinte, “*Network Configuration Settings*”, para configurar as estações conectadas à CPU.

CC-Link IEF Basic Configuration

CC-Link IEF Basic Configuration Edit View Close with Discarding the Setting Close with Reflecting the Setting

Detect Now Link Scan Setting

Connected Count 4

No.	Model Name	STA#	Station Type	RX/RX Setting			RWw/RWv Setting			Group No.	RSVD STA	IP Address	Subnet Mask	MAC Address	Comment
				Points	Start	End	Points	Start	End						
0	Host Station	0	Master Station								192.168.3.250				
1	MR-JE-C	1	Slave Station	54 (1 Occupied Station)	0000	003F	32	0000	001F	1	No Setting	192.168.3.50	255.255.255.0		
2	MR-JE-C	2	Slave Station	54 (1 Occupied Station)	0040	007F	32	0020	003F	1	No Setting	192.168.3.51	255.255.255.0		
3	MR-JE-C	3	Slave Station	54 (1 Occupied Station)	0080	00BF	32	0040	005F	1	No Setting	192.168.3.52	255.255.255.0		
4	MR-JE-C	4	Slave Station	54 (1 Occupied Station)	00C0	00FF	32	0060	007F	1	No Setting	192.168.3.53	255.255.255.0		

Clique no botão “*Detect Now*” para encontrar as estações que estão conectadas à rede. Depois de detectado clique em “***Close with Reflecting Setting***” para salvar a configuração.

Na tela anterior clique em “***Refresh Setting***” para configurar os registradores do CLP que vão comandar o CLP.



Em “Device Name”, temos RX, RY, RWr, RWw, que são os registradores que irão comandar o servo e receber informações dele, já descritos nesse tutorial.

Setting Item									
Link Side					CPU Side				
Device Name	Points	Start	End		Target	Device Name	Points	Start	End
RX	256	00000	000FF	↔	Specify Devi	W	16	00000	0000F
RY	256	00000	000FF	↔	Specify Devi	W	16	00010	0001F
RWr	128	00000	0007F	↔	Specify Devi	W	128	00020	0009F
RWw	128	00000	0007F	↔	Specify Devi	W	128	000A0	0011F

Lembre-se de clicar em “**Apply**” após concluir essas configurações.

Dessa forma, teremos a seguinte configuração pronta em nosso programa de exemplo;

Bits de Leitura (RX)							
Servo 1		Servo 2		Servo 3		Servo 4	
De	Até	De	Até	De	Até	De	Até
W0.0	W3.F	W4.0	W7.F	W8.0	WB.F	WC.0	WF.F

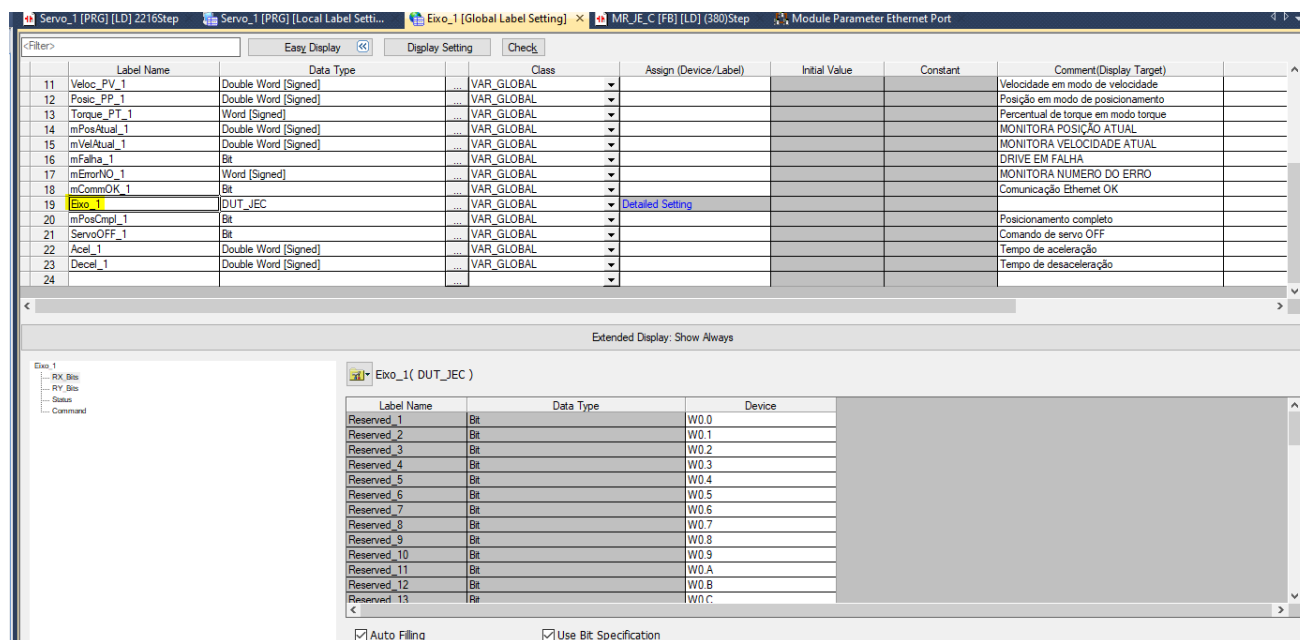
Bits de Escrita (RY)							
Servo 1		Servo 2		Servo 3		Servo 4	
De	Até	De	Até	De	Até	De	Até
W10.0	W13.F	W14.0	W17.F	W18.0	W1B.F	W1C.0	W1F.F

Palavras de Leitura (RWr)							
Servo 1		Servo 2		Servo 3		Servo 4	
De	Até	De	Até	De	Até	De	Até
W20	W3F	W40	W5F	W60	W7F	W80	W9F

Palavras de Escrita (RWw)							
Servo 1		Servo 2		Servo 3		Servo 4	
De	Até	De	Até	De	Até	De	Até
WA0	WBF	WC0	WDF	WE0	WFF	W100	W11F



As mesmas configurações descritas na tabela acima devem ser feitas nos **Labels “Eixo_n”**, em cada tabela de Label de cada eixo. Na figura abaixo, podemos ver que, para o Label Eixo_1, as variáveis RX Bits começam a partir do W0.0, conforme a tabela acima nos mostrou.



Label Name	Data Type	Class	Assign (Device/Label)	Initial Value	Constant	Comment(Display Target)
11 Veloc_PV_1	Double Word (Signed)	VAR_GLOBAL				Velocidade em modo de velocidade
12 Posic_PP_1	Double Word (Signed)	VAR_GLOBAL				Posição em modo de posicionamento
13 Torque_PT_1	Word (Signed)	VAR_GLOBAL				Percentual de torque em modo torque
14 mPosAtual_1	Double Word (Signed)	VAR_GLOBAL				MONITORA POSIÇÃO ATUAL
15 mVelAtual_1	Double Word (Signed)	VAR_GLOBAL				MONITORA VELOCIDADE ATUAL
16 mFalha_1	Bit	VAR_GLOBAL				DRIVE EM FALHA
17 mErrorNO_1	Word (Signed)	VAR_GLOBAL				MONITORA NUMERO DO ERRO
18 mCommOK_1	Bit	VAR_GLOBAL				Comunicação Ethernet OK
19 Eixo_1	DUT_JEC	VAR_GLOBAL	Detailed Setting			
20 mPosCmpl_1	Bit	VAR_GLOBAL				Posicionamento completo
21 ServoOFF_1	Bit	VAR_GLOBAL				Comando de servo OFF
22 Acel_1	Double Word (Signed)	VAR_GLOBAL				Tempo de aceleração
23 Decel_1	Double Word (Signed)	VAR_GLOBAL				Tempo de desaceleração
24						

Label Name	Data Type	Device
Reserved_1	Bit	W0.0
Reserved_2	Bit	W0.1
Reserved_3	Bit	W0.2
Reserved_4	Bit	W0.3
Reserved_5	Bit	W0.4
Reserved_6	Bit	W0.5
Reserved_7	Bit	W0.6
Reserved_8	Bit	W0.7
Reserved_9	Bit	W0.8
Reserved_10	Bit	W0.9
Reserved_11	Bit	W0.A
Reserved_12	Bit	W0.B
Reserved_13	Bit	W0.C

Essa tabela irá mudar de acordo com a necessidade de o usuário incrementar eixos. Se for necessário reduzir o número de eixos, use os exemplos prontos para menos eixos.



7. Utilizando o programa de exemplo (CCLink IEF Basic iQ-F_MR-JE_Structured_DT+FB_4_Eixos.gx3)

O programa de exemplo já vem com rotinas criadas para 4 eixos de servos. As rotinas são basicamente iguais, mudando apenas o número do eixo _1, _2, _3 e _4.

The screenshot displays the Mitsubishi GX Developer software interface. On the left, the 'Navigation' pane shows the project structure, including 'Project', 'Module Configuration', 'Program', 'Initial', 'Scan', 'MAIN', 'Servo_1', 'Local Label', 'ProgramBody', 'Servo_2', 'Servo_3', 'Servo_4', 'Fixed Scan', 'Event', 'Standby', 'No Execution Type', 'Unregistered Program', 'FB/FUN', 'FBFILE', 'MR_JE_C', 'Label', 'Global Label', 'Eixo_1', 'Eixo_2', 'Eixo_3', 'Eixo_4', 'Structured Data Types', 'Device', 'Parameter', 'System Parameter', 'FX5UCPU', 'CPU Parameter', 'Module Parameter', 'Ethernet Port', '485 Serial Port', 'High Speed I/O', 'Input Response Time', 'Analog Input', 'Analog Output', and 'Expansion Board'. The main workspace shows the ladder logic for the 'MR_JE_C' program. The first step (Step 1) is selected, showing a network with a coil (0) and a contact labeled 'MR_JE_C.1 (MR_JE_C)'. The right side of the workspace shows the variable declaration for the network, including 'DUT: DUT', 'DUT: DUT', 'DUT: DUT', 'DUT: DUT', and various servo-related variables like 'TorqueLi', 'CommOK', 'CurrentP', 'ActualSp', 'ErrorNo', 'StopAxis', 'Fault', 'ResetFault', 'Enabled', 'HomeOK', 'PosCompl', 'PositSp', 'CommPo', 'ABS_Pos', 'INC_Pos', 'PV_Speed', 'SpeedMo', 'TorqueC', and 'TorqueM'.



O usuário deverá criar a lógica de sua máquina **adicionando uma ou mais novas rotinas, com os nomes desejados** e manter no programa as rotinas criadas pela MEB para cada servo. Dessa forma, o usuário poderá usar em seu programa as variáveis de entrada e saída dos Function Blocks de cada eixo em sua lógica para executar o controle dos servos.

As variáveis disponíveis para uso do usuário são as seguintes:

Label	Descrição	Formato
ServoON_n	Sinal de servo ON	Bit
Stop_n	Sinal de STOP	Bit
MoveABS_n	Movimenta em ABS	Bit
MoveINC_n	Movimenta em INC	Bit
Pausa_n	Pausa movimento	Bit
Reset_n	Reset de Erros	Bit
Home_n	Home do eixo	Bit
MoveSpd_n	Move em Speed Mode	Bit
MoveTorq_n	Move em torque mode	Bit
Veloc_PP_n	Velocidade em modo de posicionamento	Double Word [Signed]
Veloc_PV_n	Velocidade em modo de velocidade	Double Word [Signed]
Posic_PP_n	Posição em modo de posicionamento	Double Word [Signed]
Torque_PT_n	Percentual de torque em modo torque	Word [Signed]
mPosAtual_n	MONITORA POSIÇÃO ATUAL	Double Word [Signed]
mVelAtual_n	MONITORA VELOCIDADE ATUAL	Double Word [Signed]
mFalha_n	DRIVE EM FALHA	Bit
mErrorNO_n	MONITORA NÚMERO DO ERRO	Word [Signed]
mCommOK_n	Comunicação Ethernet OK	Bit
mPosCmpl_n	Posicionamento completo	Bit
ServoOFF_n	Comando de servo OFF	Bit
Acel_n	Tempo de aceleração	Double Word [Signed]
Decel_n	Tempo de desaceleração	Double Word [Signed]

OBS.: Substitua “_n” pelo número do eixo “_1”, “_2”, etc.

Para executar servo ON

- Mantenha ligado o bit **ServoON_n**.

Para executar um HPR (Home Position Return)

- Ligue o bit **Home_n**

Obs.: O servo está configurado para o método de HPR “data set”. **Quando o bit Home_n for ligado, o servo interpretará que a posição atual será o ZERO.**

Para executar um movimento incremental

- Mova o valor desejado de posição para a variável **Posic_PP_n** (Double Word);
- Mova o valor desejado de velocidade em r/min para a variável **Veloc_PP_n** (Double INT)
- Ligue o bit **MoveINC_n**



Para executar um movimento absoluto

- Mova o valor desejado de posição para a variável **Posic_PP_n** (Double Word);
- Mova o valor desejado de velocidade em r/min para a variável **Veloc_PP_n** (Double INT)
- Ligue o bit **MoveABS_n**

Para rodar em modo velocidade

- Mova o valor desejado de velocidade em r/min para a variável **Veloc_PV_n** (Double INT)
(note que não é a mesma dos movimentos de posicionamento)
- Ligue o bit **MoveSPD_n**

Para rodar em modo torque

- Mova o valor desejado de torque em percentual para a variável **Torque_PT_n** (INT)
(note que não é a mesma dos movimentos de posicionamento)
- Ligue o bit **MoveTorq_n**

Para alterar rampas de aceleração e desaceleração

- Utilize as variáveis **Acel_1** e **Decel_1** (Double INT) em “ms”.

As variáveis que começam com “m” minúsculo são apenas para monitoramento

mPosAtual_n	MONITORA POSIÇÃO ATUAL	Double Word [Signed]
mVelAtual_n	MONITORA VELOCIDADE ATUAL	Double Word [Signed]
mFalha_n	DRIVE EM FALHA	Bit
mErrorNO_n	MONITORA NUMERO DO ERRO	Word [Signed]
mCommOK_n	Comunicação Ethernet OK	Bit
mPosCmpl_n	Posicionamento completo	Bit