



**GUIA RÁPIDO PARA  
COMUNICAÇÃO CANopen ENTRE**



---

**CANopen**®

## Sumário

<b>1. INTRODUÇÃO</b> .....	3
<b>2. CONFIGURAÇÃO DO HARDWARE</b> .....	4
2.1 Ligação Física dos Servos (CN3) .....	4
2.2 Configuração de Baud Rate (Velocidade de Transmissão) .....	5
2.3 Topologia da Rede .....	6
2.4 Resistor de Terminação .....	7
2.4 Cabos CANopen .....	8
2.5 Resumo das Precauções.....	8
<b>3. PARAMETRIZAÇÃO NO DRIVE</b> .....	8
3.1 Parâmetros no Drive – Configuração do Protocolo.....	8
<b>4. WORD DE CONTROLE E STATUS</b> .....	9
4.1 Control Word (6040h) .....	9
4.2 Status Word (6041h).....	10
<b>5. MODOS DE OPERAÇÃO</b> .....	12
<b>5.1 Modo Posição</b> .....	12
5.1.1 Procedimento de Operação (Posição) .....	12
5.1.2 Lista de Objetos relacionados para o Modo Posição .....	12
5.1.3 Control Word (6040h) para o Modo Posição.....	13
5.1.4 Status Word (6041h) para o Modo Posição.....	13
5.1.5 Exemplo de Aplicações com Modo Posição.....	14
<b>5.2 Modo Velocidade</b> .....	16
5.2.1 Procedimento de Operação (Velocidade).....	16
5.2.2 Lista de Objetos relacionados para o Modo Velocidade .....	16
5.2.3 Exemplo de Aplicação com Modo Velocidade.....	17
<b>5.3 Modo Torque</b> .....	17
5.3.1 Procedimento de Operação (Torque).....	17
5.3.2 Lista de Objetos relacionados para o Modo Torque.....	17
5.3.3 Exemplo de Aplicação com Modo Torque .....	18
<b>5.4 Modo Torque</b> .....	18
5.4.1 Procedimento de Operação (Torque).....	18
5.4.2 Lista de Objetos relacionados para o Modo Torque.....	18
5.4.3 Exemplo de Aplicação com Modo Torque .....	19
<b>6. CONFIGURAÇÃO DE SOFTWARE</b> .....	19
<b>6.1 Servo Plover</b> .....	20
<b>6.2 CANopen Builder</b> .....	21
6.2.1 Upload do Mapa EDS no Software .....	22
6.2.2. Configuração do Mapa EDS (PDO's) .....	26
<b>6.3 ISPSOft</b> .....	29
<b>7. FLUXOGRAMA</b> .....	31
<b>8. CONSIDERAÇÕES FINAIS</b> .....	32

## 1. INTRODUÇÃO

Nesse guia rápido, vamos criar um passo a passo de como configurar e manusear o servo motor da INVT com o CLP Delta (linha DVP) utilizando o protocolo CANopen.

Sabemos que o protocolo CANopen é muito utilizado em aplicações de posicionamento do servo motor e feedback de posição. A vantagem da utilização desse protocolo é a economia dos meios físicos (cabos) para acionamento dos motores e a velocidade de transmissão de dados, que pode chegar até 1Mbps.

A rede CANopen também facilita o controle de dois ou mais eixos, onde podemos controlar os motores através dos endereços fixos de cada servo (nó). A topologia linha, como veremos adiante neste documento, sintetiza a ligação física de fios e permite uma organização mais limpa do painel, além da facilidade no controle de variáveis.

Os servos motores INVT suportam o protocolo CAN em seus drives, porém, é ideal a análise e codificação correta dos modelos de drives. Os modelos da linha DA180 e DA200 Standard não possuem esse recurso para comunicação, mesmo obtendo a porta RJ45 incorporada, com isso, é necessário a validação do código com o time técnico da Kalatec.

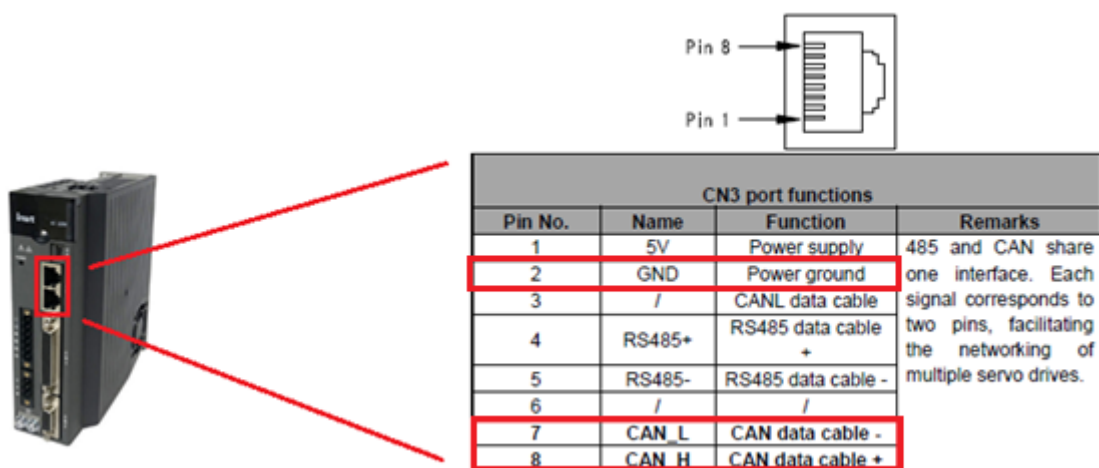
Os CLP's abordados neste documento serão os CLP's da Delta, linha DVP. Usamos para teste de comunicação a CPU DVP28SV2 e o módulo de expansão para rede CANopen DVPCOPM. Os CLP's da linha DVP são modelos Slim, compactos e de baixo custo, por isso, devemos sempre adequar ao projeto a expansão DVPCOPM para montar uma rede CAN.

Utilizaremos softwares gratuitos (disponíveis no site da Kalatec Automação) para configuração da rede. As informações foram retiradas dos manuais dos fabricantes INVT e Delta para montar esse Quick Start. Para maiores informações, entrar em contato com a engenharia da Kalatec Automação. O contato do nosso suporte técnico e vendas se encontra no final desse documento.

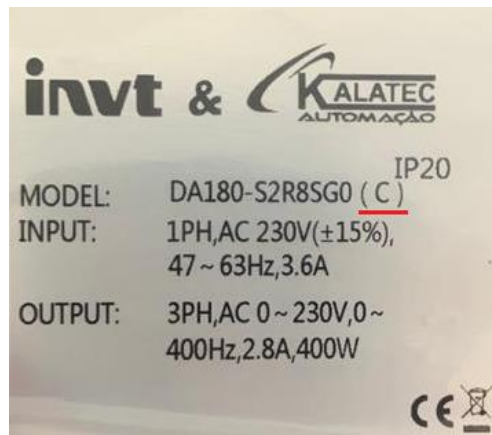
## 2. CONFIGURAÇÃO DO HARDWARE

### 2.1 Ligação Física dos Servos (CN3)

A ligação física da rede CANopen é simples, usamos o CAN\_LOW (-) e o CAN\_HIGH (+) para a transmissão de informações, e o GND para proteção. Usualmente, a conexão é feita em um soquete RJ-45 nos periféricos. A porta de comunicação dos servos INVT é o CN3. Nesse conector estão presentes os protocolos disponíveis no drive: Modbus e CANopen. Para realizar a ligação física do CANopen, devemos conectar os pinos 7 e 8 (CAN\_L e CAN\_H, respectivamente) e o pino 2 (GND), todos do conector RJ-45. Segue abaixo a figura ilustrando:



**OBS:** Para os modelos DA180, devemos verificar se no código do produto ele vem com a letra "C" no final do código. Exemplo: DA180-S2R8SG0 (C). Esse final "C" significa que o driver é Modbus e CANopen. Caso não tenha essa letra, o driver não é compatível com o protocolo CAN.



## 2.2 Configuração de Baud Rate (Velocidade de Transmissão)

O Baud Rate é a velocidade de transmissão dos dados na comunicação CAN. Podemos configurá-lo de acordo com a necessidade. Quanto mais rápido a velocidade de transmissão, menor é a distância entre os dispositivos. Segue tabela abaixo:

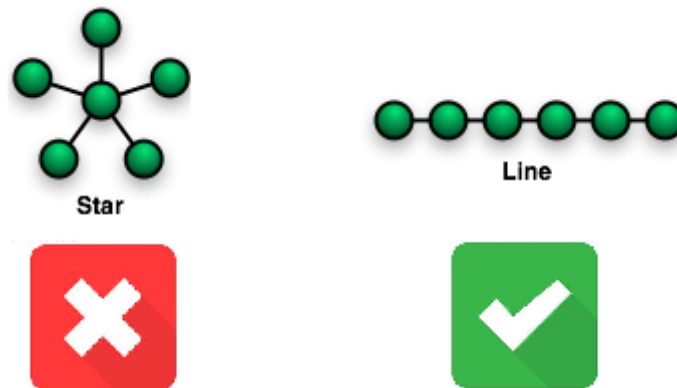
Communication baud rate	Communication distance
1Mbit/s	25m
500kbit/s (Default)	100m
250kbit/s	250m
125kbit/s	500m
50kbit/s	1000m
20kbit/s	2500m

A taxa de transmissão mais convencional é 500kbit/s. Inclusive, com os CLP's da Delta, a taxa de transmissão **DEVE SER CONFIGURADA** para **500kbit/s** ou menor.

Devemos configurar o mesmo Baud Rate para todos os dispositivos da rede. No caso dos servos INVT, alteramos o parâmetro P4.02 para 1 (500kbit/s).

## 2.3 Topologia da Rede

Sabemos que as redes industriais possuem topologias distintas. Para cada rede de protocolo diferente, uma arquitetura de dispositivos é indicada. Para a rede CAN não é diferente. Todos os dispositivos da rede devem ser ligados EM SÉRIE. Não deve ser usada a ligação em estrela.



Podemos exemplificar também uma rede CANopen com servos conforme desenho abaixo:

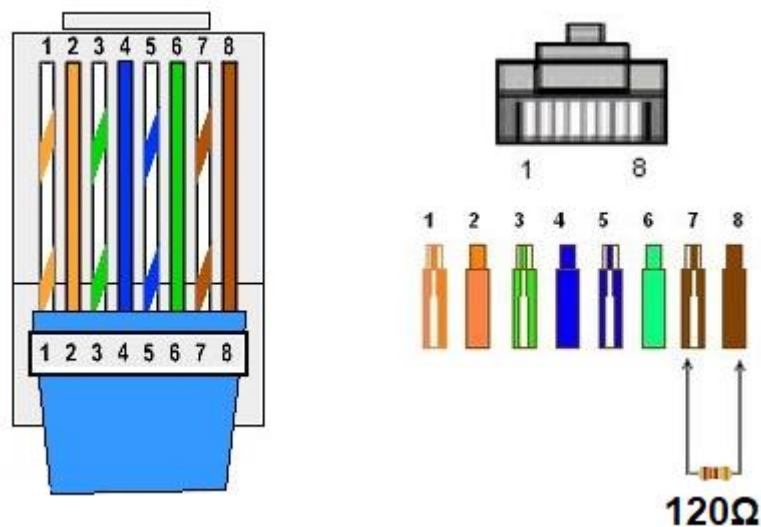


Ao final de cada servo, devemos colocar um resistor de terminação. Isso delimita o final da rede. A seguir abordaremos um pouco mais sobre esse resistor.

## 2.4 Resistor de Terminação

Os resistores de terminação são imprescindíveis em uma rede de comunicação. Esse resistor de  $120\Omega$  é colocado no **último dispositivo** da rede CAN a fim de delimitar o final da rede e evitar reflexões de sinal e desvio de corrente referente a esse sinal de comunicação.

Alguns dispositivos já fornecem na caixa do produto esse resistor, outros já têm incorporado no próprio periférico. Já no caso dos servos da INVT, esse resistor deve ser confeccionado pelo próprio usuário. Abaixo segue a ligação física:



O resistor de  $120\Omega$  deve ser colocado nos pinos 7 e 8 do conector RJ45. Justamente no CAN (-) e no CAN (+).

## 2.4 Cabos CANopen

Para a rede CAN pode ser usado cabos de rede convencionais RJ45, porém não é o mais indicado. O mais indicado é o cabo RJ45 blindado com proteções contra ruído e de dimensões no tamanho ideal para a ramificação da rede.



## 2.5 Resumo das Precauções

1. Todos os dispositivos da rede CANopen devem ser conectados em série. Nunca realizar a conexão em estrela.
2. Um resistor de 120Ω deve ser conectado no último escravo na rede CANopen.
3. O ponto de amostra da comunicação CAN da estação mestre deve ser definido para 80%.
4. Para evitar interferência, é recomendável utilizar pares trançados blindados (STP) como cabos para conexão CAN.
5. Um cabo de conexão mais longo requer um chip CAN com maior capacidade de unidade.

## 3. PARAMETRIZAÇÃO NO DRIVE

### 3.1 Parâmetros no Drive – Configuração do Protocolo

Para configurar o CANopen nos drives precisamos alterar apenas três parâmetros. Utilizamos o software ServoPlover V4.18 para configuração. Seguem os parâmetros abaixo:

1. **P0.03** (Control Mode Selection) para valor = **7** (CANopen Mode);
2. **P4.02** (CAN communication Baud Rate) para valor (**0**: 1Mbps; **1**: 500kbps; **2**: 250kbps; **3**: 125kbps; **4**: 50kbps; **5**: 20kbps).



3. **P4.05** (CAN communication node) para valor **(1~127)**. Usualmente utilizamos o nó 1 para o mestre (CLP) e os demais dispositivos CAN, usamos a partir do nó 2.

Notas:

- Esses três parâmetros são validados somente após o servo ser reenergizado.
- Os endereços dos servos (nós) não podem se repetir ou ser o mesmo nó que o mestre.

## 4. WORD DE CONTROLE E STATUS

### 4.1 Control Word (6040h)

A Control Word é um registro responsável por controlar o funcionamento do motor. Dentro desse registro você pode deixar o motor em Servo On/Off, habilitar para começar o movimento, Reset de alarmes, entre outros. Segue abaixo a tabela exemplificando a funcionalidade da Control Word.

A Control Word é composta por 16bits.

15 ~ 11	10 ~ 9	8	7	6 ~ 4	3	2	1	0
Específico do Fabricante	Reservado	Halt	Reset Alarme	Específico do Modo de Operação	Habilitar Operação	Stop	Habilitar Tensão	Switch On
O	O	O	M	O	M	M	M	M

MSB

LSB

**MSB:** Most Significant Bit (Bit mais significativo);


**LSB:** Least Significant Bit (Bit menos significativo);


**O:** Optional (Opcional);

**M:** Mandatory (Obrigatório);

Temos a relação abaixo dos 16 bits composto em tabelas para exemplificar o funcionamento de cada grupo.

**Bits 0~3 e bit 7 são bits para controle de Status do Servo. Segue tabela com algumas funções:**

COMANDO	Bits da Control Word				
	Reset Alarme	Habilitar Operação	Stop	Habilitar Tensão	Switch On
Desligar	0	X	1	1	0
Switch On	0	0	1	1	1
Switch On	0	1	1	1	1
Desabilitar Tensão	0	X	X	0	X
Stop	0	X	0	1	X
Desabilitar Operação	0	0	1	1	1
Habilitar Operação	0	1	1	1	1
Reset Alarme		X	X	X	X

Onde X indica que a operação não está envolvida e  indica borda de subida.

Bits 4, 5, 6 e 8 são bits relacionados ao modo de controle. Segue tabela com algumas funções:

Bit	Modo Velocidade	Modo Posição	Modo Homing	Modo Interpolação
4	rfg enable	Novo Set-point	Start Home	Habilitar modo Interpolação
5	rfg unlock	Mudar set-point imediatamente	Reservado	Reservado
6	rfg use ref	absoluto/relativo	Reservado	Reservado
8	Halt	Halt	Halt	Halt

Bits 9 e 10: Reservados;

Bits 11~15: Definidos pelo fabricante;

Esses são os bits da word de controle (control word) onde comandamos o servo. No próximo tópico, veremos os bits de status (status word) onde conseguimos monitorar o funcionamento do servo.

## 4.2 Status Word (6041h)

A Status Word é um registro responsável por monitorar o funcionamento do motor. Dentro desse registro você pode averiguar informações como: status de servo on/off, status de alarme, servo ready, entre outras informações. Segue abaixo a tabela exemplificando a funcionalidade da Status Word.

A status word é composta por 16bits:

Bit	Descrição	M/O
0	Pronto para operar	M
1	Switch On	M
2	Operação habilitada	M
3	Alarme	M
4	Tensão habilitada	M
5	Stop	M
6	Switch On desabilitado	M
7	Aviso	O
8	Específico do fabricante	O
9	Remoto	M
10	Término do posicionamento	M
11	Fim de curso ativado	M
12~13	Modo de operação específico	O
14~15	Específico do fabricante	O

**O:** Optional (Opcional);

**M:** Mandatory (Obrigatório);

**Bits 0~3 e 6 são bits relacionados ao modo de status. Segue tabela com algumas funções:**

Valor (Binário)	Estado
xxxx xxxx x0xx 0000	Não está pronto para operar
xxxx xxxx x1xx 0000	Switch On desabilitado
xxxx xxxx x01x 0001	Pronto para operar
xxxx xxxx x01x 0011	Switch On
xxxx xxxx x01x 0111	Operação habilitada
xxxx xxxx x00x 0111	Stop ativo
xxxx xxxx x0xx 1111	Reação de falha ativa
xxxx xxxx x0xx 1000	Alarme

Onde X indica que o bit não está envolvido.

## 5. MODOS DE OPERAÇÃO

### 5.1 Modo Posição

Um servo drive (escravo) recebe um comando de posição transmitido por uma unidade mestre (CLP). A posição obtida pela conversão de comando de posição com base na relação do *Electronic Gear* (engrenagem eletrônica) é usada como *Target Position* (posição alvo) no controle de posição interno.

#### 5.1.1 Procedimento de Operação (Posição)

1. Defina **6060h: Mode of Operations** para **1** (Modo Posição).
2. Defina **6081h: Profile Velocity** para alterar a velocidade (unidade: RPM).
3. Defina **6083h (Acc)** e **6084h (Dcc): Profile Acceleration/Deceleration** para alterar a aceleração e desaceleração (unidade: ms).
4. Defina **Sub-1** e **Sub-2** do **6093h: Position Factor** para alterar o *Electronic Gear* (engrenagem eletrônica) do drive. Nesses parâmetros que definimos a resolução do servo. **Sub-1** indica o numerador e o **Sub-2** indica o denominador, correspondentes aos parâmetros **P0.25** e **P0.26** do driver, respectivamente.
5. Defina **607Ah: Target Position** para alterar a posição em que o servo irá movimentar (unidade: definida pelo próprio usuário).
6. Defina **6040h: Control Word** para habilitar o servo e dar o trigger para o motor movimentar na posição programada. 0x0F indica habilitado. Para mais detalhes, volte para seção XXXXXXXX.
7. Leia **6064h: Position Actual Value** para obter o feedback de leitura da posição atual do servo.
8. Leia **6041h: Status Word** para obter o feedback dos status do servo (entradas acionadas, alarmes, servo ready, entre outras...).

#### 5.1.2 Lista de Objetos relacionados para o Modo Posição

Index	Nome	Tipo	Atribuição
6040h	Control Word	UNSIGNED16	RW
6041h	Status Word	UNSIGNED16	RO
6060h	Modes of Operation	INTEGER8	RW
6061h	Modes of Operation Display	INTEGER8	RO
6062h	Position Demand Value	INTEGER32	RO
6063h	Position Actual Value*	INTEGER32	RO
6064h	Position Actual Value	INTEGER32	RO
6065h	Following Error Window	UNSIGNED32	RW

6067h	Position Window	UNSIGNED32	RW
607Ah	Target Position	INTEGER32	RW
6081h	Profile Velocity	UNSIGNED32	RW
6093h	Position Factor	UNSIGNED32	RW
6083h	Profile Acceleration	UNSIGNED32	RW
6084h	Profile Deceleration	UNSIGNED32	RW
60F4h	Following Error Actual Value	INTEGER32	RO
60FCh	Position Demand Value	INTEGER32	RO

**Nota:** Na lista acima estão presentes os principais objetos para realizar a operação do modo posição. Caso a aplicação exija o controle de outras variáveis/objetos, consultar manual completo CANopen INVT, onde está presente a lista completa de objetos.

### 5.1.3 Control Word (6040h) para o Modo Posição

15 ~ 9	8	7	6	5	4	3 ~ 0
(Veja seção 4.1)	Halt	(Veja seção 4.1)	Abs / Rel	Mudar set-point imediatamente	Novo set-point	(Veja seção 4.1)

MSB LSB

Nome	Valor	Descrição
Novo set-point	0	Não assume uma posição desejada
	1	Assume a posição desejada
Mudar set-point imediatamente	0	Termina a posição atual e depois inicia a próxima posição
	1	Interrompe a posição atual e inicia a próxima posição
Abs / Rel	0	Posicionamento em modo absoluto
	1	Posicionamento em modo relativo
Halt	0	Executa posicionamento
	1	Stop no eixo com desaceleração

### 5.1.4 Status Word (6041h) para o Modo Posição

15~14	13	12	11	10	9 ~ 0
(Veja seção 4.2)	Erro	Reconhecimento do set point	(Veja seção 4.2)	Posição finalizada	(Veja seção 4.2)

MSB LSB

Nome	Valor	Descrição
Posição finalizada	0	Halt = 0 posição alvo não atingida Halt = 1 eixo desacelera
	1	Halt = 0 posição alvo atingida Halt = 1 velocidade do eixo é 0 (zero)
Reconhecimento do set point	0	O gerador de trajetória não assumiu o valor de posição (ainda)
	1	O gerador de trajetória assumiu o valor de posição
Erro	0	Executa posicionamento
	1	Stop no eixo com desaceleração

## 5.1.5 Exemplo de Aplicações com Modo Posição

### 5.1.5.1 Modo Single Set Point

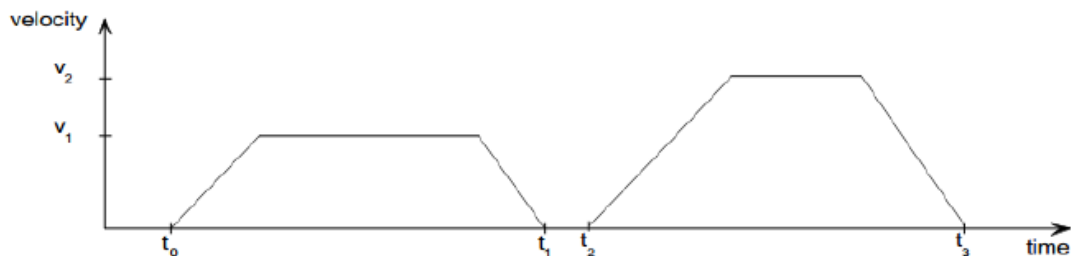


Diagrama do modo single set point.

Se você estiver trabalhando no modo **INCREMENTAL**, você deve seguir os seguintes passos:

1. Defina **6040h** para **0x4F** (do qual o bit 6 é usado para definir o modo incremental e os bits 3 a 0 são usados para habilitar o driver).
2. Defina **607Ah** para o comando de posição.
3. Defina **6040h** para **0x5F** para acionar o comando de posição (o comando de posição é habilitado quando o valor de bit4 muda de 0 para 1).
4. O drive retorna **bit12 (6041h)** quando o **bit4** muda de **0** para **1 (6040h)** e a estação mestre exclui o valor de **bit 4 (6040h)** após receber **6041h** para se preparar para transmitir o próximo comando de posição.

Se você estiver trabalhando no modo **ABSOLUTO**, você deve seguir os seguintes passos:

1. Defina **6040h** para **0x0F**.
2. Defina **607Ah** para o comando de posição.
3. Defina **6040h** para **0x1F** para habilitar o modo de posição.

4. O drive retorna **bit12 (6041h)** quando o **bit4** muda de **0** para **1 (6040h)** e a estação mestre exclui o valor de **bit 4 (6040h)** após receber **6041h** para se preparar para transmitir o próximo comando de posição.

#### 5.1.5.2 Modo Multi-Setpoint alterado imediatamente.

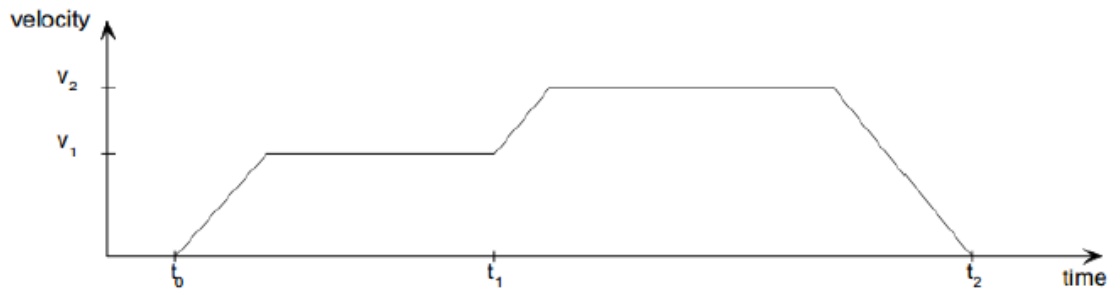


Diagrama do Modo Multi-Setpoint alterado imediatamente.

Se você estiver trabalhando no modo **INCREMENTAL**, você deve seguir os seguintes passos:

1. Defina **6040h** para **0x6F** (do qual o bit 6 é usado para definir o modo incremental, o bit5 é usado para definir o modo de efeito imediato e os bits 3 a 0 são usados para habilitar o driver).
2. Defina **607Ah** para o comando de posição.
3. Defina **6040h** para **0x7F** para acionar o comando de posição (o comando de posição é habilitado quando o valor de bit4 muda de 0 para 1).
4. O drive retorna **bit12 (6041h)** quando o **bit4** muda de **0** para **1 (6040h)** e a estação mestre exclui o valor de **bit 4 (6040h)** após receber **6041h** para se preparar para transmitir o próximo comando de posição.

Se você estiver trabalhando no modo **ABSOLUTO**, você deve seguir os seguintes passos:

1. Defina **6040h** para **0x2F** (do qual o bit5 é usado para definir o modo de efeito imediato e os bits 3 a 0 são usados para habilitar o driver).
2. Defina **607Ah** para o comando de posição.
3. Defina **6040h** para **0x3F** para acionar o comando de posição.
4. O drive retorna **bit12 (6041h)** quando o **bit4** muda de **0** para **1 (6040h)** e a estação mestre exclui o valor de **bit 4 (6040h)** após receber **6041h** para se preparar para transmitir o próximo comando de posição.
5. Se várias posições de destino serão transmitidas, repita o passo 2.

**OBS:** Os servos drives SV-DA200 suportam armazenamento em cache interno de 08 etapas de posições de destino.

## 5.2 Modo Velocidade

No modo velocidade, o drive recebe um comando de velocidade de rotação transmitido pela estação mestre e planeja as velocidades internas de acordo com as configurações dos parâmetros de planejamento de aceleração.

### 5.2.1 Procedimento de Operação (Velocidade)

1. Defina **6060h: Modo de Operação** para **3** (Modo velocidade).
2. Defina **6083h** para ajustar a aceleração (unidade: ms).
3. Defina **6084h** para ajustar a desaceleração (unidade: ms).
4. Defina **6040h: Control word** para habilitar o drive e acionar o motor.
5. Defina **60FFh: Target Velocity** para definir a velocidade do motor (unidade: RPM).
6. Leia **6041h: Status Word** para ter o feedback de velocidade, posição, status do servo, entre outras informações.
7. Leia **606Ch: Velocity Actual Value** para obter o feedback da velocidade atual (unidade: RPM).

### 5.2.2 Lista de Objetos relacionados para o Modo Velocidade

Index	Nome	Tipo	Atribuição
6040h	Control Word	UNSIGNED16	RW
6041h	Status Word	UNSIGNED16	RO
6060h	Modes of Operation	INTEGER8	RW
6061h	Modes of Operation Display	INTEGER8	RO
6069h	Velocity Sensor Actual Value	INTEGER32	RO
606Bh	Velocity Demand Value	INTEGER32	RO
606Ch	Velocity Actual Value	INTEGER32	RO
606Dh	Velocity Window	UNSIGNED16	RW
606Fh	Velocity Threshold	UNSIGNED16	RW
6083h	Profile Acceleration	UNSIGNED32	RW
6084h	Profile Deceleration	UNSIGNED32	RW
60F8h	Max Slippage	INTEGER32	RW
60FFh	Target Velocity	INTEGER32	RW

**Nota:** Na lista acima estão presentes os principais objetos para realizar a operação do modo posição. Caso a aplicação exija o controle de outras variáveis/objetos, consultar manual completo CANopen INVT, onde está presente a lista completa de objetos.



### 5.2.3 Exemplo de Aplicação com Modo Velocidade

Ao usar o modo velocidade, você precisará realizar os seguintes passos:

1. Defina **6060h** para **3** (modo velocidade).
2. Defina **6040h** para **0x0F** para habilitar o drive e **0x0** para desabilitar o drive.
3. Defina **60FFh** para alterar a velocidade do motor.
4. Defina **6083h** e **6084h** para modificar a aceleração e desaceleração do motor, respectivamente (em ms).

### 5.3 Modo Torque

No modo torque, o drive recebe um comando de torque transmitido pela estação mestre e planeja os torques internos de acordo com as configurações dos parâmetros de planejamento do torque.

#### 5.3.1 Procedimento de Operação (Torque)

1. Defina **6060h: Modo de Operação** para **4** (Modo torque).
2. Defina **6087h: Torque Slope** para ajustar a rampa de torque (unidade: ms). Indica o tempo que leva para aumentar o torque de 0 ~ 100% do torque nominal.
3. Defina **6040h: Control word** para habilitar o drive e acionar o motor.
4. Defina **6071h: Target Torque** para definir o torque do motor (unidade: 0,1%).
5. Leia **6041h: Status Word** para ter o feedback de velocidade, torque, status do servo, entre outras informações.

#### 5.3.2 Lista de Objetos relacionados para o Modo Torque

Index	Nome	Tipo	Atribuição
6040h	Control Word	UNSIGNED16	RW
6041h	Status Word	UNSIGNED16	RO
6060h	Modes of Operation	INTEGER8	RW
6061h	Modes of Operation Display	INTEGER8	RO
6071h	Target Torque	INTEGER16	RO
6072h	Max Torque	UNSIGNED16	RW
6073h	Max Current	UNSIGNED16	RO
6074h	Torque Demand Value	INTEGER16	RO
6075h	Motor Rated Current	UNSIGNED32	RO
6076h	Motor Rated Torque	UNSIGNED32	RO
6077h	Torque Actual Value	INTEGER16	RO

6078h	Current Actual Value	INTEGER16	RO
6079h	DC Link Circuit Voltage	UNSIGNED32	RO
6087h	Torque Slope	UNSIGNED32	RW

**Nota:** Na lista acima estão presentes os principais objetos para realizar a operação do modo posição. Caso a aplicação exija o controle de outras variáveis/objetos, consultar manual completo CANopen INVT, onde está presente a lista completa de objetos.

### 5.3.3 Exemplo de Aplicação com Modo Torque

Ao usar o modo torque, você precisará realizar os seguintes passos:

1. Defina **6060h** para **4** (modo torque).
2. Defina **6040h** para **0x0F** para habilitar o drive e **0x0** para desabilitar o drive.
3. Defina **6071h** para **alterar o torque do motor**.
4. Defina **6087h** para alterar o Torque Slope.

## 5.4 Modo Torque

No modo torque, o drive recebe um comando de torque transmitido pela estação mestre e planeja os torques internos de acordo com as configurações dos parâmetros de planejamento do torque.

### 5.4.1 Procedimento de Operação (Torque)

1. Defina **6060h: Modo de Operação** para **4** (Modo torque).
2. Defina **6087h: Torque Slope** para ajustar a rampa de torque (unidade: ms). Indica o tempo que leva para aumentar o torque de 0 ~ 100% do torque nominal.
3. Defina **6040h: Control word** para habilitar o drive e acionar o motor.
4. Defina **6071h: Target Torque** para definir o torque do motor (unidade: 0,1%).
5. Leia **6041h: Status Word** para ter o feedback de velocidade, torque, status do servo, entre outras informações.

### 5.4.2 Lista de Objetos relacionados para o Modo Torque

Index	Nome	Tipo	Atribuição
6040h	Control Word	UNSIGNED16	RW
6041h	Status Word	UNSIGNED16	RO

6060h	Modes of Operation	INTEGER8	RW
6061h	Modes of Operation Display	INTEGER8	RO
6071h	Target Torque	INTEGER16	RO
6072h	Max Torque	UNSIGNED16	RW
6073h	Max Current	UNSIGNED16	RO
6074h	Torque Demand Value	INTEGER16	RO
6075h	Motor Rated Current	UNSIGNED32	RO
6076h	Motor Rated Torque	UNSIGNED32	RO
6077h	Torque Actual Value	INTEGER16	RO
6078h	Current Actual Value	INTEGER16	RO
6079h	DC Link Circuit Voltage	UNSIGNED32	RO
6087h	Torque Slope	UNSIGNED32	RW

**Nota:** Na lista acima estão presentes os principais objetos para realizar a operação do modo posição. Caso a aplicação exija o controle de outras variáveis/objetos, consultar manual completo CANopen INVT, onde está presente a lista completa de objetos.

#### 5.4.3 Exemplo de Aplicação com Modo Torque

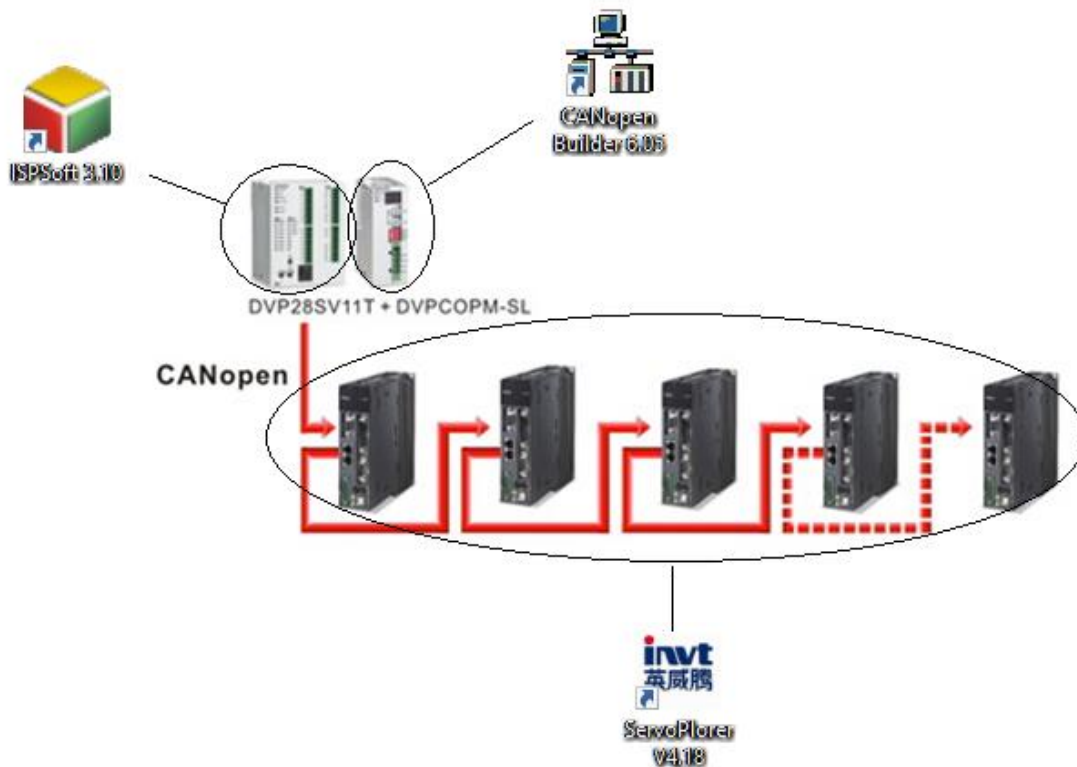
Ao usar o modo torque, você precisará realizar os seguintes passos:

1. Defina **6060h** para **4** (modo torque).
2. Defina **6040h** para **0x0F** para habilitar o drive e **0x0** para desabilitar o drive.
3. Defina **6071h** para **alterar o torque do motor**.
4. Defina **6087h** para alterar o Torque Slope.

## 6. CONFIGURAÇÃO DE SOFTWARE

Vamos utilizar três softwares para configuração da rede CANopen, sendo eles:

- **Servo Plover V4.18** (ou acima) → Configuração dos Servos Motores INVT;
- **CANopen Builder V6.05** (ou acima) → Configuração da expansão DVPCOPM-SL;
- **ISPSOFT V3.10** (ou acima) → Configuração dos CLP's da linha DVP;



## 6.1 Servo Plover

Para a configuração do Servo Plover V4.18 é necessário analisarmos os parâmetros que vamos alterar para trabalhar com a rede CAN. Esses parâmetros estão presentes no Capítulo 3 desse manual (veja 3.1).

- Parâmetro P0.03 = CANopen mode;
- Parâmetro P4.02 = 5000Kbps;
- Parâmetro P4.05 = Nó do servo na rede;

Parameter Setting

P0	P1	P2	P3	P4	P5	P6	PtP0	PtP1	PtP2	Different parameter	Common parameter	
Function Code	Parameter Name						Current Value	*	Unit	Min	Max	Default
P0.00	Motor Type						0		-	0	9999999	236
P0.01	Encoder type selection						unknown		-	1	13	2500 line standard in
P0.02	Motor Forward Direction						CCW		-	0	1	CCW
P0.03	Control Mode Selection						CANopen mode		-	0	9	Position mode
P0.04	Internal servo enbaling						Disable		-	0	1	Disable
P0.05	Jog speed						0		r/min	0	1000	200
P0.06	Numerator of encoder pulse output						0		-	0	2147483647	10000

Parameter Setting

P0	P1	P2	P3	P4	P5	P6	PtP0	PtP1	PtP2	Different parameter	Common parameter	
Function Code	Parameter Name						Current Value	*	Unit	Min	Max	Default
P4.01	485 Local communication address						0		-	1	255	1
P4.02	Can baudrate selection						500K		-	0	5	500K
P4.03	Communication baudrate selection						9600		-	0	3	19200
P4.04	Communication parity mode						N 8 1		-	0	5	N 8 1

Parameter Setting

P0	P1	P2	P3	P4	P5	P6	PtP0	PtP1	PtP2	Different parameter	Common parameter	
Function Code	Parameter Name						Current Value	*	Unit	Min	Max	Default
P4.01	485 Local communication address						0		-	1	255	1
P4.02	Can baudrate selection						1M		-	0	5	500K
P4.03	Communication baudrate selection						9600		-	0	3	19200
P4.04	Communication parity mode						N 8 1		-	0	5	N 8 1
P4.05	Can Local communication address						2		-	1	127	1
P4.06	485 Local communication address						Keep fault		-	0	1	Auto d
P4.07	EtherCAT synchronisation period						250us		-	0	3	1ms

## 6.2 CANopen Builder

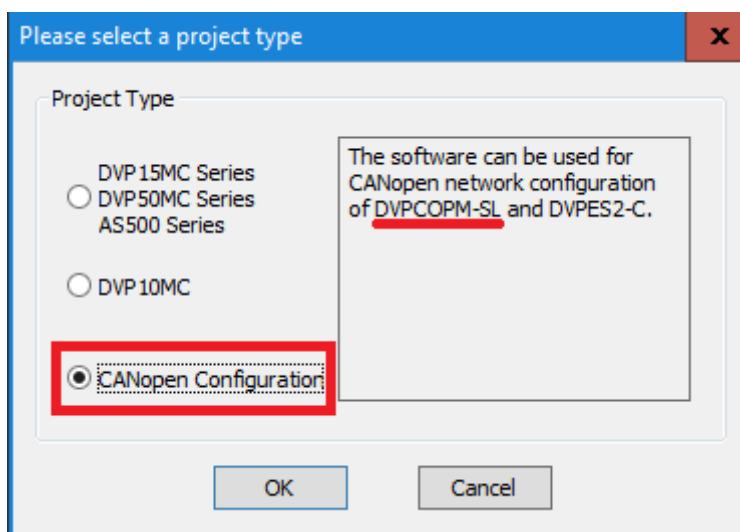
O software CANopen Builder é responsável por reconhecer os servos na rede e entrelaçar os endereços do CLP e do servo. Para configurar esses endereços no CANopen Builder é necessário ter o arquivo EDS (Electronic Data Sheet) do servo motor. Esse arquivo é onde está configurado todo dicionário de PDO's, ou seja, nesse arquivo está presente todo o endereçamento de memórias presentes no servo motor. No CANopen Builder podemos realizar o upload desse

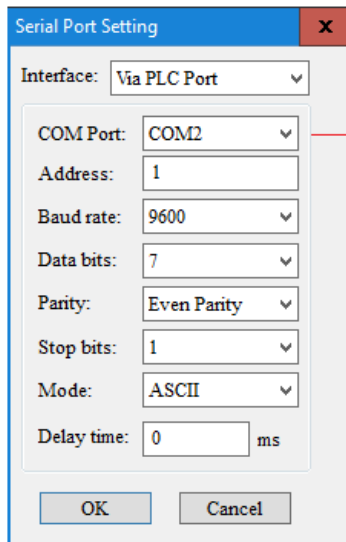
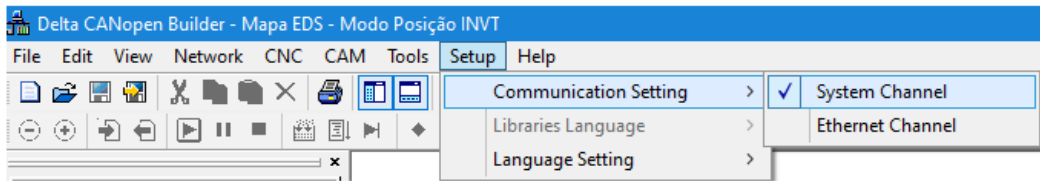
arquivo e trabalhar apenas com os endereços do CLP que estarão entrelaçados com esse mapa EDS.

Segue abaixo o passo a passo do CANopen Builder para:

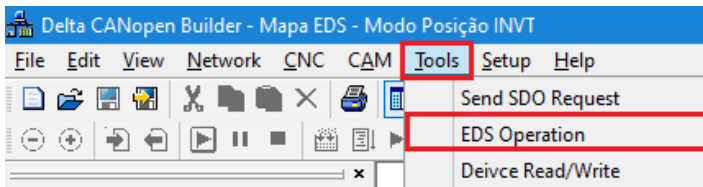
- Reconhecimento dos servos na rede;
- Upload do mapa EDS;
- Configuração do entrelaçamento de endereços;

### 6.2.1 Upload do Mapa EDS no Software

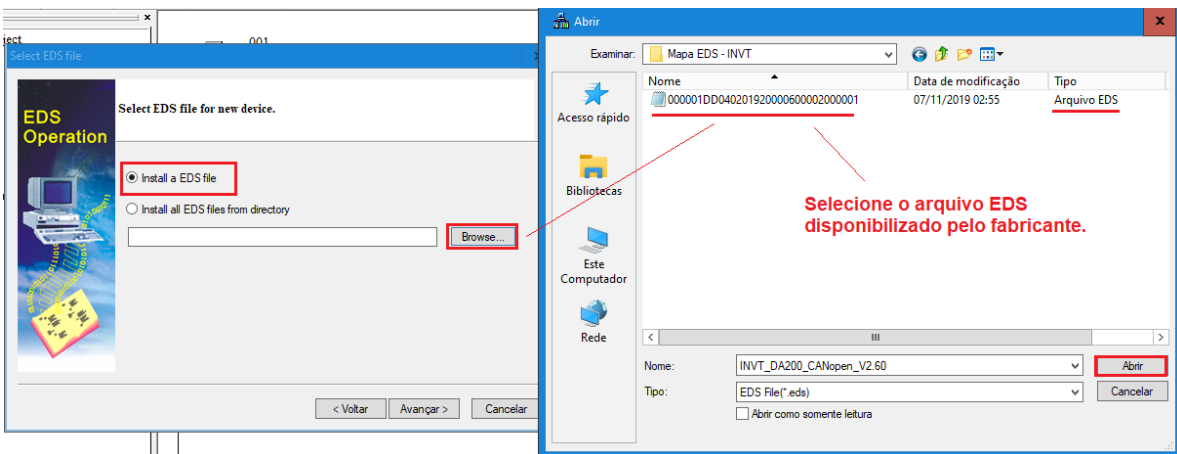
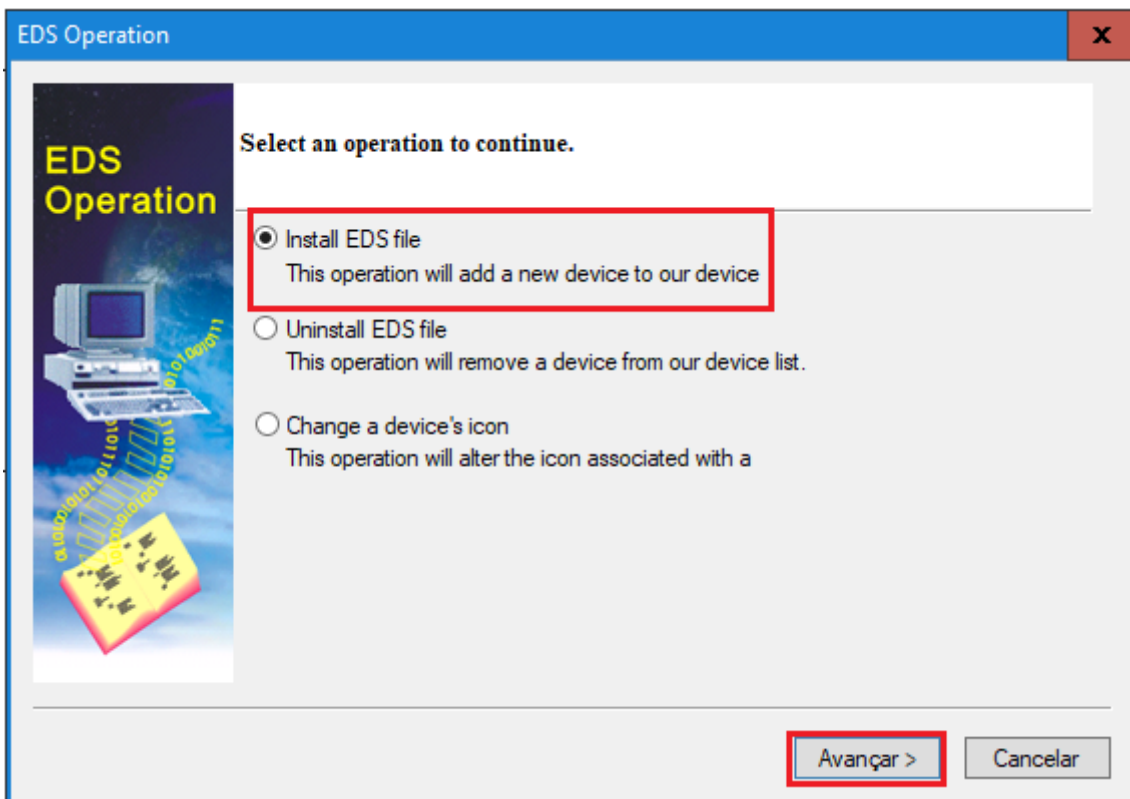




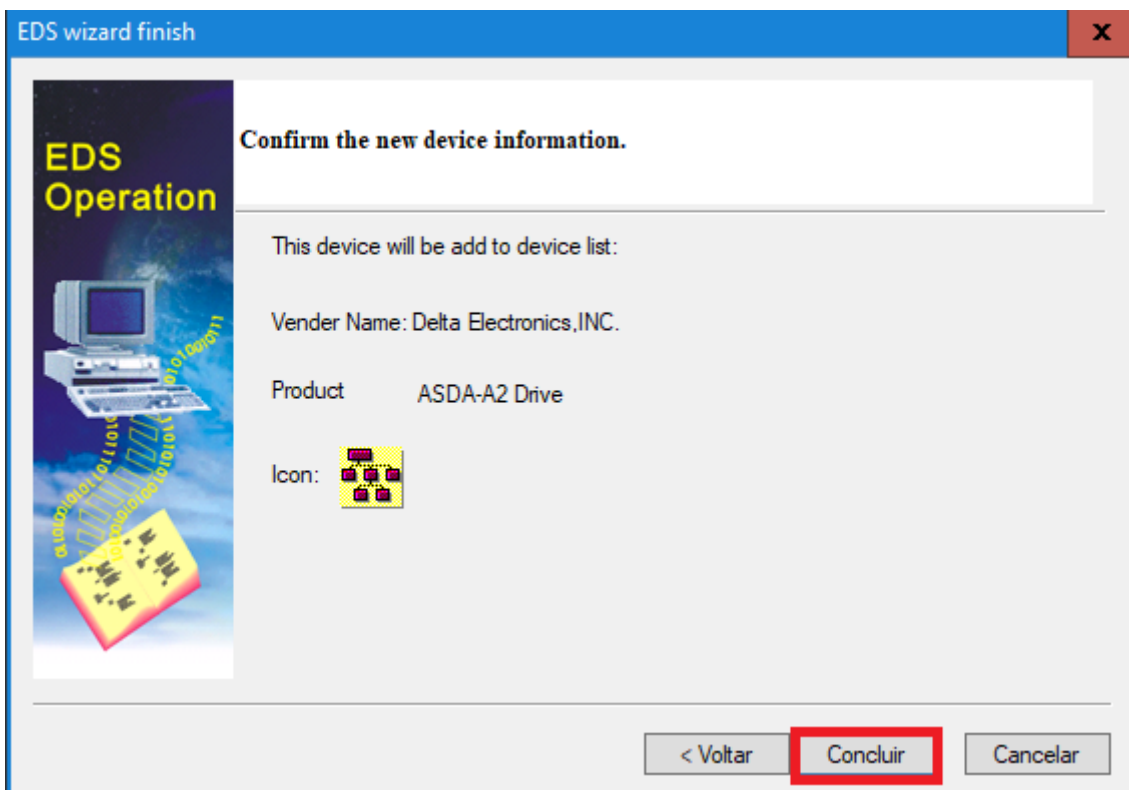
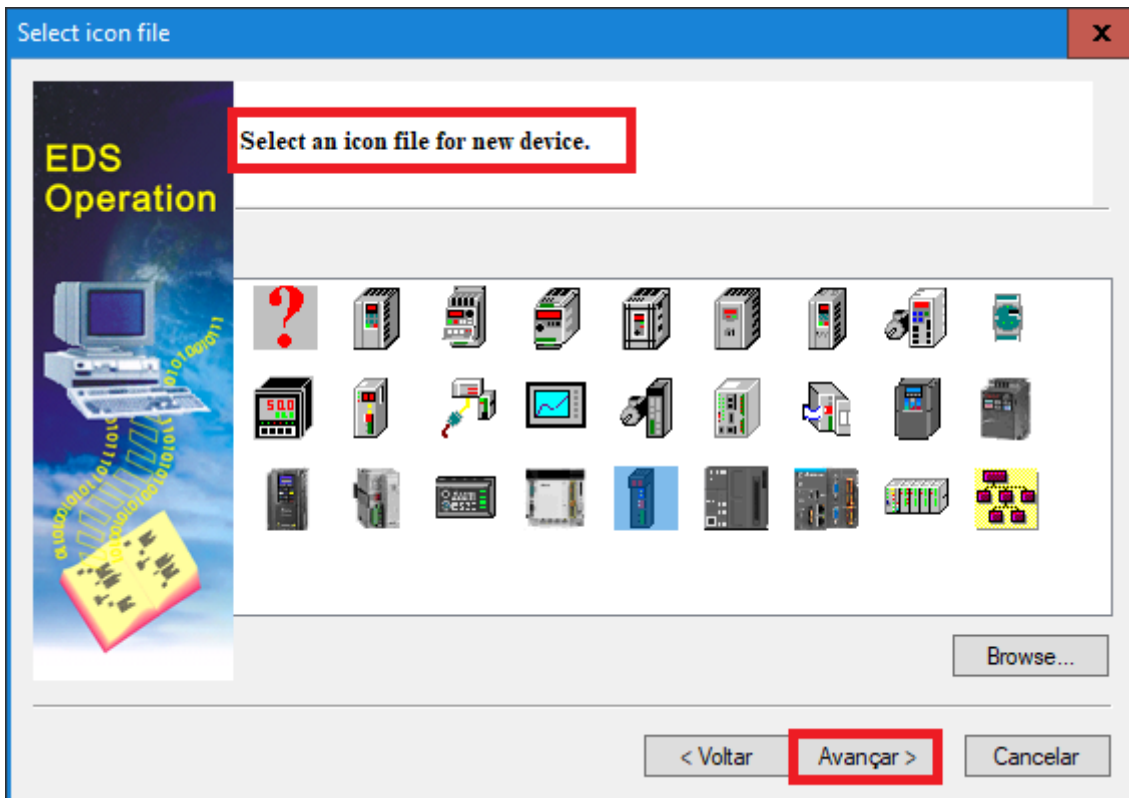
Abrir gerenciador de dispositivos e verificar qual porta de comunicação está sendo utilizada. Pode ser utilizado o COMMGR também para identificação da porta



Clique em Tools > EDS Operation

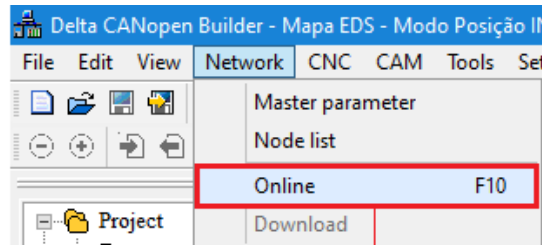






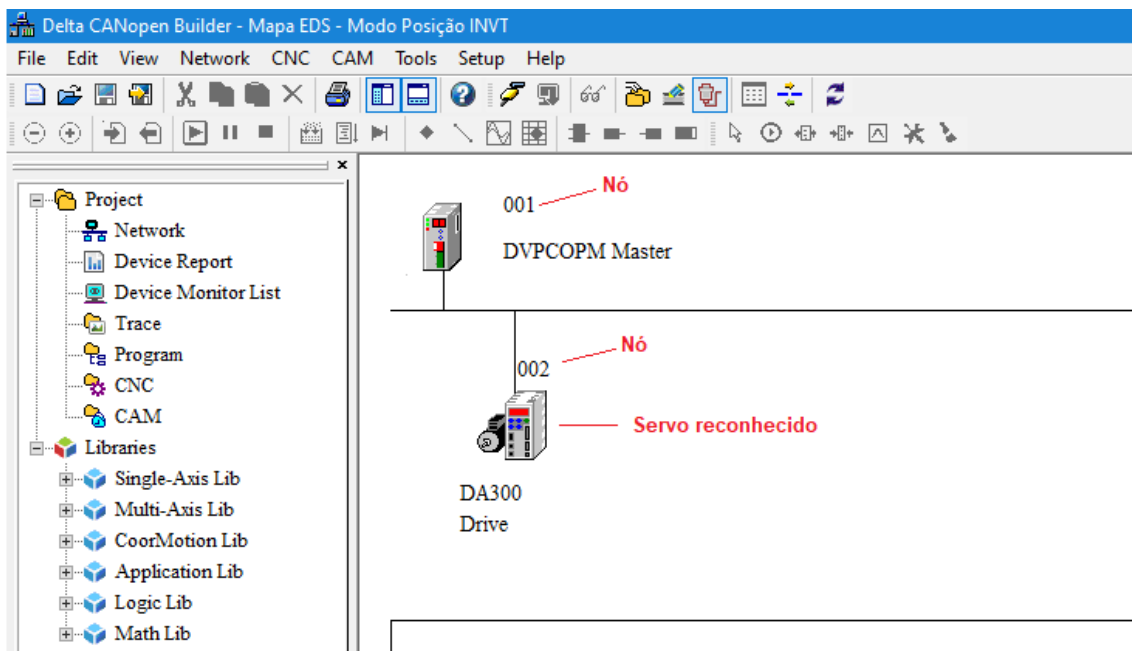
## 6.2.2. Configuração do Mapa EDS (PDO's)

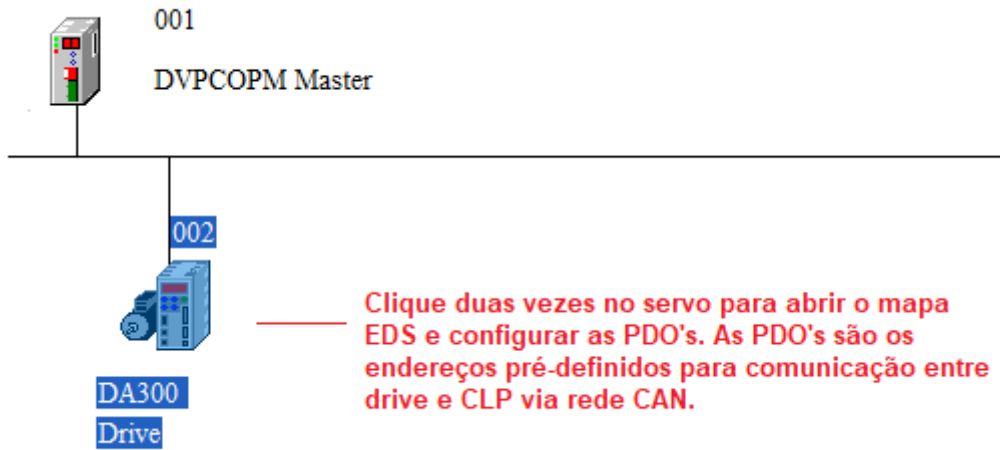
Segue abaixo o procedimento para reconhecimento do servo na rede CANopen através do CANopen Builder.



Caso os servos já estejam configurados e parametrizados para o modo CANopen, você consegue reconhecê-los diretamente no CANopen Builder.

Basta ir em Network > Online e esperar ele achar todos os nós da rede. Com isso, o próprio software já montará a estrutura, conforme imagem ao lado.





Clique duas vezes no servo para abrir o mapa EDS e configurar as PDO's. As PDO's são os endereços pré-definidos para comunicação entre drive e CLP via rede CAN.

Node Configuration...

Node-Id: 2    Name: DA300 Drive

Node Information(Hex)

Vendor Id: 00000003    Error Control Protocol

Device Type: 04020200    Auto SDO Configuration

Product Code: 00000010    Emergency COB-ID: 82

Revision: 00000074    Nodeguard COB-ID: 702

PDO from EDS file

Index	PDO Name	Type	Inhibit	Event	
1400	Receive PDO Communic...	254	-	-	
1401	Receive PDO Communic...	254	-	-	
1402	Receive PDO Communic...	254	-	-	
1403	Receive PDO Communic...	254	-	-	

Copy EDS file

Add

Delete

Define PDO

Configured PDO

Index	COB-ID	R/T	Len	Type	Description

OK

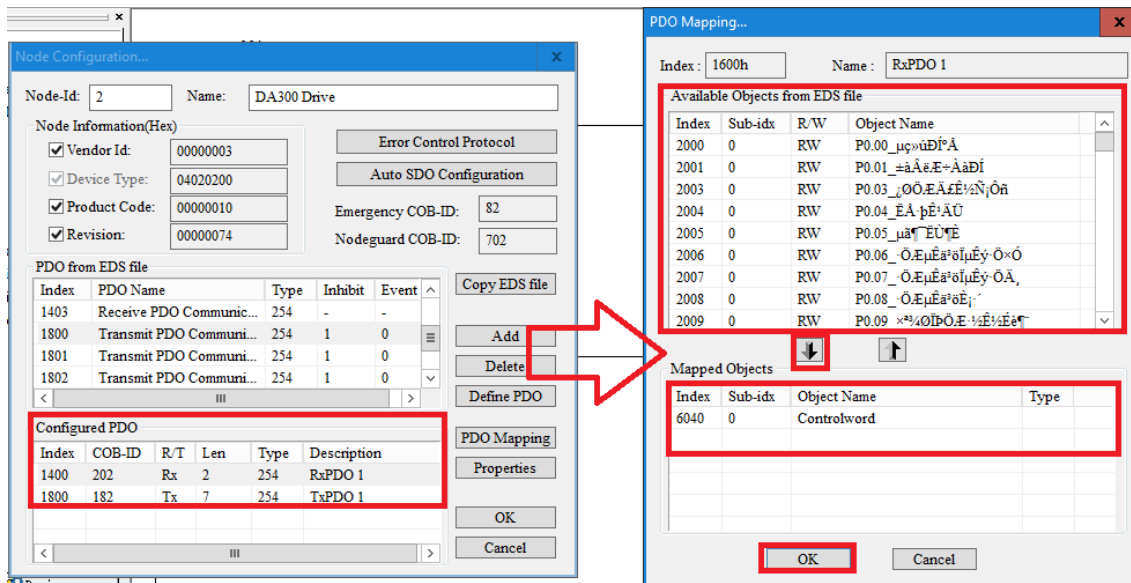
Cancel

Essa tabela mostra as PDO's do mapa EDS do servo, onde:

Receive PDO = Endereço que será RECEBIDO pelo servo (escrita no CLP).

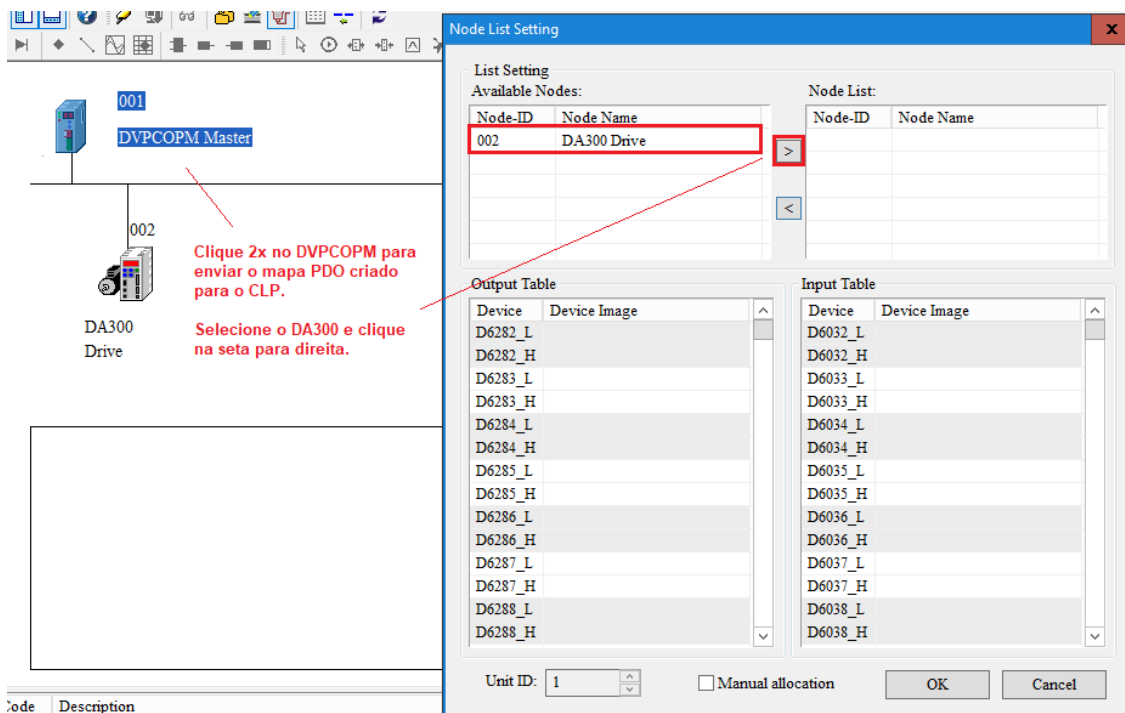
Transmit PDO = Endereço que será ENVIADO pelo servo (leitura no CLP).

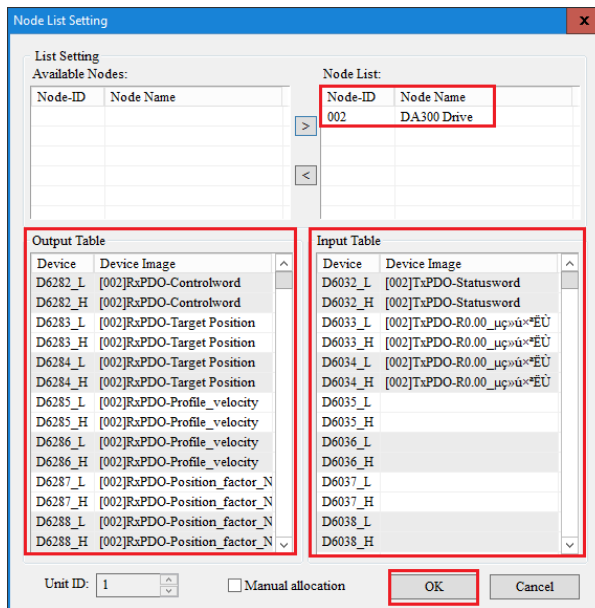
Na tabela de "Configured PDO" é o local onde configuramos os endereços PDO's para ler/escrever no CLP. Basta selecionar quantos registros de leitura/escrita, clicar em "Add" e depois configurar os endereços.



Depois de configurar a PDO, clique 2x no Index e selecione qual parâmetro você quer ler ou escrever em PDO Mapping. Clique na seta para baixo para mover o endereço a ser lido/escrito e depois clique em Ok.

Faça isso para todos os endereços que você quer ler/escrever e crie seu próprio mapa de memórias para comunicação.





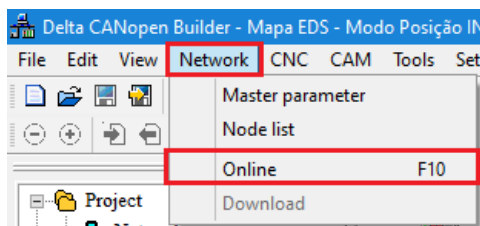
Feito o procedimento acima, podemos visualizar todo o nosso mapa criado. Em Output Table são os endereços em que vamos escrever no servo e Input Table são os endereços que vamos ler do servo.

Observamos que esses endereços já vêm configurados com os registros "D" do CLP. Ou seja, os registradores do CLP já estão atrelados aos endereços do servo. Por isso devemos respeitar esse endereçamento.

Outro ponto importante é a divisão dos registros. Podemos observar que temos Dxxx\_L e Dxxx\_H, ou seja, Low word e High Word. Nunca devemos colocar um endereço sobreposto a outro. Exemplo

D6282\_L - Control Word  
D6282\_H - Target Position

Observamos que há dois endereços diferentes do servo dentro de um mesmo registro do CLP. Isso ocasionará conflito e não conseguiremos comunicar de maneira correta. Para reparar esse erro, devemos realocar as memórias do mapa PDO descrito nos passos anteriores.



Após configurado o mapa PDO, podemos realizar o download para a expansão clicando em Network > Online > Download

### 6.3 ISPSOft

O ISPSOft V3.10 (ou acima) tem por objetivo criar a lógica de programação para trabalhar com o servo motor. Nesse software que vamos ler/escrever dados do servo e realizar o controle de variáveis e intertravamento. Como mostrado no capítulo anterior (ver 6.2), já temos todos os endereços configurados e atrelados a variáveis do servo. Agora só temos que implementar esses endereços no próprio ISPSOft.

É necessário a criação de um "Monitor Table" para facilitar a visualização e acionamento dos registradores, conforme exemplo abaixo:

CANopen\_INVT\_ModoVelocidade(1) - Delta ISPSoft - [Monitor Table]

File Edit View Compile PLC Tools Wizard Window Help

Project	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Radix	Comment...
NWCONFIG	D6032					Binary	Status Word
Project [C:\Gustavo\	D6033					Signed Decimal	Leitura RPM
Device Comm	D6034					Signed Decimal	Leitura RPM
Used Device R	D6035					Signed Decimal	
SV2	D6036					Signed Decimal	
Tasks	D6282					Binary	Control Word
Global Symbol	D6283					Signed Decimal	Target Velocity
Main Tabl	D6284					Signed Decimal	Target Velocity
Programs	D6285					Signed Decimal	Aceleração (ms)
Main [PRC	D6286					Signed Decimal	Desaceleração (ms)
Function Block	D6287					Signed Decimal	Modo de Operação
Device Monitor							
Monitor Te							
APIs							

Os endereços acima (D – registradores) foram retirados do mapa PDO criado no software CANopen Builder. Para exemplificar, segue abaixo um mapa PDO criado (modo velocidade) para comparação:

Node List Setting

List Setting

Available Nodes:

Node-ID	Node Name

Node List:

Node-ID	Node Name
002	DA300 Drive

Output Table

Device	Device Image
D6282_L	[002]RxPDO-Controlword
D6282_H	[002]RxPDO-Controlword
D6283_L	[002]RxPDO-Target_velocity
D6283_H	[002]RxPDO-Target_velocity
D6284_L	[002]RxPDO-Target_velocity
D6284_H	[002]RxPDO-Target_velocity
D6285_L	[002]RxPDO-v1_velocity_accele
D6285_H	[002]RxPDO-v1_velocity_accele
D6286_L	[002]RxPDO-v1_velocity_decel
D6286_H	[002]RxPDO-v1_velocity_decel
D6287_L	[002]RxPDO-Modes of operatic
D6287_H	
D6288_L	
D6288_H	

Input Table

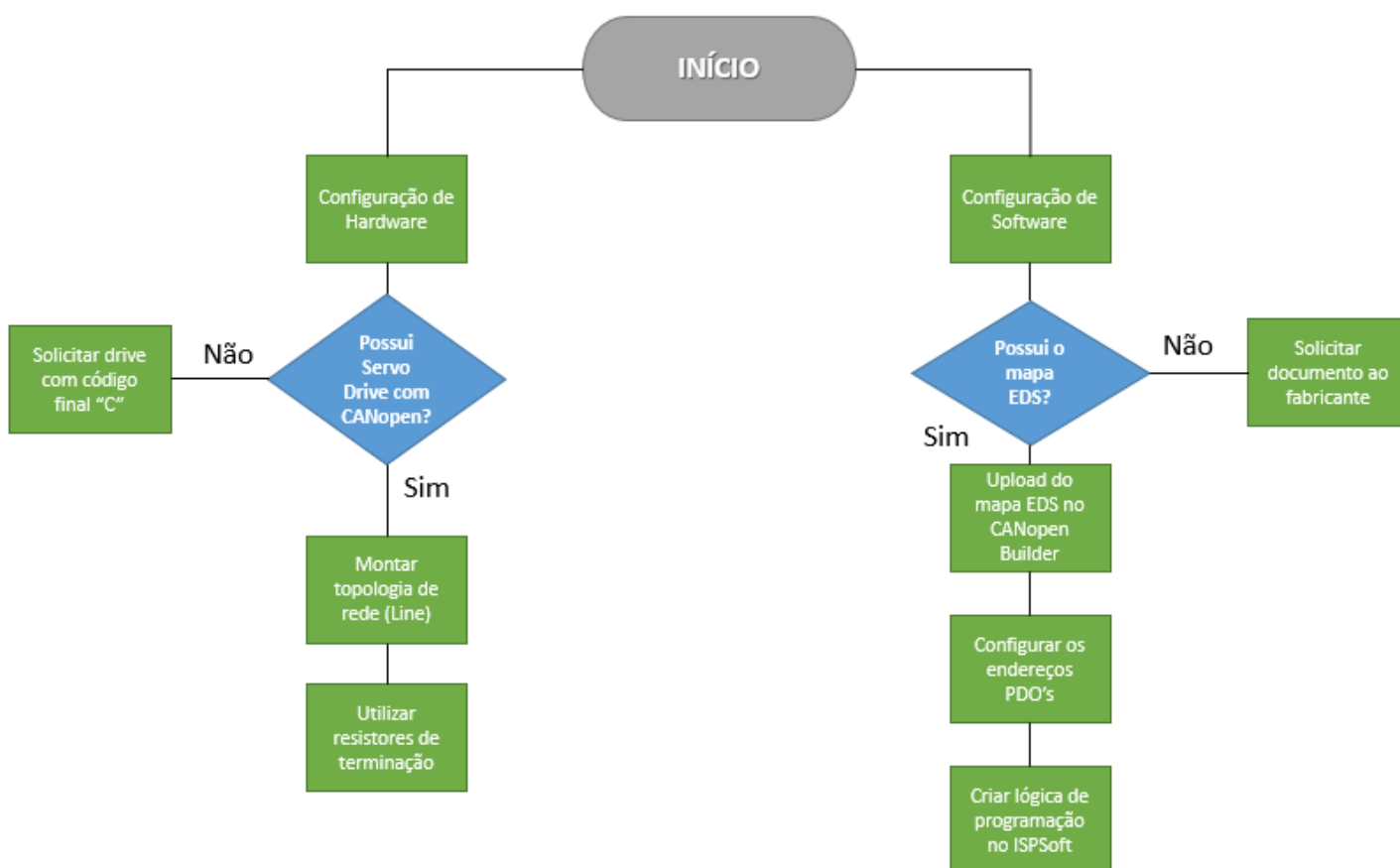
Device	Device Image
D6032_L	[002]TxPDO-Statusword
D6032_H	[002]TxPDO-Statusword
D6033_L	[002]TxPDO-R0.00_μç»ú*ÈÛ
D6033_H	[002]TxPDO-R0.00_μç»ú*ÈÛ
D6034_L	[002]TxPDO-R0.00_μç»ú*ÈÛ
D6034_H	[002]TxPDO-R0.00_μç»ú*ÈÛ
D6035_L	
D6035_H	
D6036_L	
D6036_H	
D6037_L	
D6037_H	
D6038_L	
D6038_H	

Unit ID: 1  Manual allocation

Com isso, já identificamos todos os endereços. Basta criar a lógica de programação, realizar o download para o CLP e manusear as Words de acordo com cada modo de operação do servo.

**OBS:** Lembrando que o passo a passo para os modos de operação do servo está presente no capítulo 5 desse documento.

## 7. FLUXOGRAMA



## 8. CONSIDERAÇÕES FINAIS

Com a introdução sobre o conceito da rede CAN, conseguimos entender um pouco mais sobre a aplicação e o funcionamento desse protocolo. Para ter um bom desenvolvimento é necessário um conhecimento básico no protocolo CANopen, porém vimos que não é difícil a comunicação.

Outra vantagem excelente dessa expansão MA-8X8YT é que conseguimos agrega-la com qualquer outro fabricante de CLP, seja Delta, Siemens, Rockwell, Schneider, entre outros, desde que o CLP possua o protocolo CANopen.

Qualquer dúvida pertinente entre em contato com o time técnico da Kalatec para solucioná-las. Para terem acesso a programação usada nesse documento, vídeo-aulas e manuais, entre em contato conosco também que fornecemos essa documentação. Abaixo segue os contatos em relação a cada região:

### Matriz Campinas – SP

Segunda à Quinta das 07h40 às 17h30  
Sexta das 08h00 às 17h00  
Telefone: (19) 3045-4900  
Atende Brasil inteiro e Interior de São Paulo.

### Filial São Paulo – SP

Segunda à Quinta das 07h40 às 17h30  
Sexta das 08h00 às 17h00  
Telefone: (11) 5514-7680  
Atende Grande São Paulo e São Paulo Capital.

### Filial Joinville – SC

Segunda à Quinta das 07h40 às 17h30  
Sexta das 08h00 às 17h00  
Telefone: (47) 3425-0042  
Atende a Região Sul do Brasil.

**SITE KALATEC AUTOMAÇÃO**

